BIGSdb Documentation

Release 1.44.0

Keith Jolley

Apr 16, 2024

CONTENTS

1 Concepts and terms **3**
- 1.1 BIGSdb . . . 3
- 1.2 Loci . . . 3
- 1.3 Alleles . . . 3
- 1.4 Schemes . . . 4
- 1.5 Profiles . . . 4
- 1.6 Classification groups . . . 4
- 1.7 Sequence tags . . . 4
- 1.8 Sets . . . 5

2 BIGSdb dependencies **7**
- 2.1 Required packages . . . 7

3 Installation and configuration of BIGSdb **9**
- 3.1 Software installation . . . 9
- 3.2 Configuring PostgreSQL . . . 10
- 3.3 Setting global connection parameters . . . 11
- 3.4 Site-specific configuration . . . 12
- 3.5 Setting up the offline job manager . . . 12
- 3.6 Setting up the submission system . . . 13
- 3.7 Setting up a site-wide user database . . . 14
- 3.8 Periodically delete temporary files . . . 15
- 3.9 Prevent preference database getting too large . . . 16
- 3.10 Log file rotation . . . 16
- 3.11 Upgrading BIGSdb . . . 16
- 3.12 Running the BIGSdb RESTful interface . . . 17
- 3.13 Enabling database logging of web and API access . . . 20

4 Database setup **21**
- 4.1 Creating databases . . . 21
- 4.2 Database-specific configuration . . . 22
- 4.3 XML configuration attributes used in config.xml . . . 24
- 4.4 Over-riding global defaults set in bigsdb.conf . . . 39
- 4.5 Over-riding values set in config.xml . . . 40
- 4.6 Setting field validation rules . . . 40
- 4.7 Sparsely-populated fields . . . 43
- 4.8 Kiosk mode . . . 45
- 4.9 User authentication . . . 46
- 4.10 Setting up the admin user . . . 47
- 4.11 Retrieving PubMed citations from NCBI . . . 47

4.12 Configuring access to remote contigs . . . 47
4.13 Setting up front-end and query dashboards . . . 49

5 Administrator's guide **57**
5.1 Types of user . . . 57
5.2 User groups . . . 57
5.3 Curator permissions . . . 57
5.4 Locus and scheme permissions (sequence definition database) . . . 60
5.5 Controlling access . . . 61
5.6 Setting user passwords . . . 61
5.7 Setting the first user password . . . 62
5.8 Enabling plugins . . . 62
5.9 Temporarily disabling database updates . . . 63
5.10 Host mapping . . . 63
5.11 Improving performance . . . 64
5.12 Dataset partitioning . . . 65
5.13 Setting a site-wide users database . . . 68
5.14 Adding new loci . . . 69
5.15 Defining locus extended attributes . . . 79
5.16 Defining locus amino acid variants or single-nucleotide polymorphisms . . . 80
5.17 Defining schemes . . . 82
5.18 Organizing schemes into hierarchical groups . . . 89
5.19 Setting up client databases . . . 91
5.20 Workflow for setting up a MLST scheme . . . 96
5.21 Automated assignment of scheme profiles . . . 97
5.22 Scheme profile clustering - setting up classification schemes . . . 98
5.23 Defining new loci based on annotated reference genome . . . 103
5.24 Setting up LINcode definitions for cgMLST schemes . . . 105
5.25 Genome filtering . . . 108
5.26 Setting locus genome positions . . . 112
5.27 Defining composite fields . . . 114
5.28 Extended provenance attributes (lookup tables) . . . 116
5.29 Sequence bin attributes . . . 118
5.30 Checking external database configuration settings . . . 120
5.31 Exporting table configurations . . . 121
5.32 Authorizing third-party client software to access authenticated resources . . . 124
5.33 BLAST caches . . . 125
5.34 Config-specific file downloads . . . 126

6 Curator's guide **127**
6.1 Adding new sender details . . . 127
6.2 Adding new allele sequence definitions . . . 128
6.3 Updating and deleting allele sequence definitions . . . 135
6.4 Retiring allele identifiers . . . 138
6.5 Un-retiring allele identifiers . . . 138
6.6 Updating locus descriptions . . . 140
6.7 Adding new scheme profile definitions . . . 143
6.8 Updating and deleting scheme profile definitions . . . 146
6.9 Retiring scheme profile identifiers . . . 147
6.10 Un-retiring scheme profile identifiers . . . 148
6.11 Adding isolate records . . . 150
6.12 Updating and deleting single isolate records . . . 154
6.13 Batch updating multiple isolate records . . . 157
6.14 Deleting multiple isolate records . . . 159

6.15 Retiring isolate identifiers . . . 160
6.16 Un-retiring isolate identifiers . . . 160
6.17 Setting alternative names for isolates (aliases) . . . 162
6.18 Linking isolate records to publications . . . 164
6.19 Uploading sequence contigs linked to an isolate record . . . 165
6.20 Batch uploading sequence contigs linked to multiple isolate records . . . 168
6.21 Linking remote contigs to isolate records . . . 171
6.22 Automated web-based sequence tagging . . . 173
6.23 Projects . . . 176
6.24 Isolate record versioning . . . 180
6.25 Populating geographic coordinate lookup values . . . 183

7 Curating data submitted via the automated submission system 185
7.1 Alleles . . . 186
7.2 Profiles . . . 194
7.3 Isolates . . . 200

8 Offline curation tools 205
8.1 Automated offline sequence tagging . . . 205
8.2 Defining exemplar alleles . . . 208
8.3 Automated offline allele definition . . . 209
8.4 Calculating assembly stats . . . 212
8.5 Predicting species based on rMLST analysis . . . 213
8.6 Cleanly interrupting offline curation . . . 214
8.7 Uploading contigs from the command line . . . 214
8.8 Populating geographic point lookup values . . . 215

9 Definition downloads 217
9.1 Allele sequence definitions . . . 217
9.2 Scheme profile definitions . . . 221

10 Data records 223
10.1 Isolate records . . . 223
10.2 Allele definition records . . . 227
10.3 Sequence tag records . . . 229
10.4 Profile records . . . 230
10.5 Sequence bin records . . . 231

11 Querying data 233
11.1 Querying sequences to determine allele identity . . . 233
11.2 Querying multiple sequences to identify allele identities . . . 236
11.3 Searching for specific allele definitions . . . 238
11.4 Browsing scheme profile definitions . . . 245
11.5 Querying scheme profile definitions . . . 246
11.6 Identifying allelic profile definitions . . . 250
11.7 Batch profile queries . . . 254
11.8 Investigating allele differences . . . 256
11.9 Browsing isolate data . . . 259
11.10 Querying isolate data . . . 261
11.11 Bookmarking an isolate query . . . 278
11.12 Retrieving isolates by linked publication . . . 281
11.13 User-configurable options . . . 284

12 User projects 297
12.1 Adding isolates to a user project following a query . . . 298

12.2 Adding or removing isolates belonging to a user project by editing a list . . . 299
12.3 Accessing project isolates . . . 300
12.4 Allowing other users to share your project . . . 301
12.5 Deleting a user project . . . 302

13 Private records **305**
13.1 Uploading private records . . . 305
13.2 Modifying private records . . . 307
13.3 Sharing access to private records . . . 308

14 Data analysis plugins **309**
14.1 BLAST . . . 309
14.2 BURST . . . 315
14.3 Codon usage . . . 319
14.4 Field breakdown . . . 322
14.5 Gene Presence . . . 333
14.6 Genome comparator . . . 338
14.7 GrapeTree . . . 346
14.8 In silico PCR . . . 350
14.9 Interactive Tree of Life (iTOL) . . . 354
14.10 Kleborate . . . 358
14.11 Locus explorer . . . 362
14.12 Microreact . . . 366
14.13 PhyloViz . . . 370
14.14 Polymorphisms . . . 373
14.15 Species identification . . . 375
14.16 ReporTree . . . 379
14.17 Reports . . . 383
14.18 Sequence bin breakdown . . . 383
14.19 Two field breakdown . . . 387
14.20 Unique combinations . . . 390

15 Data export plugins **393**
15.1 Contig export . . . 393
15.2 Isolate record export . . . 397
15.3 Profile export . . . 399
15.4 Sequence export . . . 402

16 Submitting data using the submission system **405**
16.1 Registering a user account . . . 405
16.2 Allele submission . . . 406
16.3 Profile submission . . . 413
16.4 Isolate submission . . . 418
16.5 Genome submission . . . 421
16.6 Assembly submission . . . 422
16.7 Removing submissions from your notification list . . . 425

17 RESTful Application Programming Interface (API) **427**
17.1 Passing additional/optional parameters . . . 427
17.2 Paging using request headers . . . 427
17.3 Resources . . . 428
17.4 Authentication . . . 459

18 Frequently asked questions (FAQs) **465**
18.1 General . . . 465

18.2 Installation 466
18.3 Administration 466

19 Appendix **469**
19.1 Query operators 469
19.2 Sequence tag flags 470
19.3 Allele sequence flags 471

20 Database schema **473**

Index **475**

Gene-by-gene population annotation and analysis

BIGSdb is software designed to store and analyse sequence data for bacterial isolates. Any number of sequences can be linked to isolate records - these can be small contigs assembled from dideoxy sequencing through to whole genomes (complete or multiple contigs generated from parallel sequencing technologies such as Illumina or Oxford Nanopore).

BIGSdb extends the principle of MLST to genomic data, where large numbers of loci can be defined, with alleles assigned by reference to sequence definition databases (which can also be set up with BIGSdb). Loci can also be grouped into schemes so that types can be defined by combinations of allelic profiles, a concept analagous to MLST.

The software has been released under the GNU General Public Licence version 3. The latest version of this documentation can be found at https://bigsdb.readthedocs.org/.

CHAPTER

ONE

CONCEPTS AND TERMS

1.1 BIGSdb

BIGSdb is the software platform - not a specific database. There are many instances of BIGSdb databases, so referring to 'the BIGSdb' is meaningless.

1.2 Loci

Loci are regions of the genome that are identified by similarity to a known sequence. They can be defined by DNA or peptide sequence. They are often complete coding sequences (genes), but may represent gene fragments (such as used in MLST), antigenic peptide loops, or indeed any sequence feature.

In versions of BIGSdb prior to 1.8.0, an isolate record could only have one live *allele* designation for a locus (inactive/pending designations could be stored within the database but were unavailable for querying or analysis purposes). Since biology is rarely so clean, and some genomes may contain more than one copy of a gene, later versions of the software allow multiple allele designations for a locus, all of which can be queried and analysed.

Paralogous loci can be difficult to differentiate by sequence similarity alone. Because of this, loci can be further defined by context, where in silico PCR or hybridization reactions can be performed to *filter the genome* to specific regions based on sequence external to the locus.

1.3 Alleles

Alleles are instances of loci. Every unique sequence, either DNA or peptide depending on the locus, is defined as a new allele and these are defined in a sequence definition database, where they are given an allele identifier. These identifiers are usually integers, but can be text strings. Allele identifiers in text format can be constrained by length and formatting.

When a specific allele of a locus is identified within the sequence data of an isolate record, the allele designation, i.e. identifier, is associated with the isolate record. This efficiently stores the sequence variation found within an isolate. Two isolates with the same allele designation for a locus have identical sequences at that locus. Once the sequence variation within a genome has been reduced to a series of allele designations, genomes can be efficiently compared by identifying which loci vary between them.

It is important to note that allele identifiers are usually arbitrary and are allocated sequentially in the order of discovery. Alleles with adjacent identifiers may vary by a single nucleotide or by many.

1.4 Schemes

Schemes are collections of loci that may be associated with additional field values. At their simplest they just group loci together. Example uses of simple schemes include:

- Antibiotic resistance genes
- Genes involved in specific biochemical pathways
- Antigens
- Vaccine components
- Whole genome MLST (wgMLST)

When schemes are associated with additional fields, one of these fields must be the primary key, i.e. its value uniquely defines a particular combination of alleles at its member loci. The pre-eminent example of this is MLST - where a sequence type (ST) is the primary key that uniquely defines combinations of alleles that make up the MLST profiles. Additional fields can also then be included. The values for these need not be unique. In the MLST example, a field for clonal complex can be included, and the same value for this can be set for multiple STs.

1.5 Profiles

Profiles are instances of *schemes*. A profile consists of a set of *allele identifiers* for the *loci* that comprise the scheme. If the scheme has a primary key field, e.g. sequence type (ST) in MLST schemes, then the unique combination of alleles in a complete profile can be defined by the value of this field.

1.6 Classification groups

Classification groups are a way to cluster scheme profiles using a specified threshold of pairwise allelic mismatches. Currently, single-linkage clustering is supported whereby each member of a group must have no more than the specified number of allelic differences with at least one other member of the group.

1.7 Sequence tags

Sequence tags record locus position within an isolate record's sequence bin. The process of creating these tags, is known as *tag-scanning*. A sequence tag consists of:

- sequence bin id - this identifies a particular contig
- locus name
- start position
- end position
- flag to indicate if sequence is reversed
- flag to indicate if sequence is complete and does not continue off the end of the contig

1.8 Sets

Sets provide a means to take a large database with multiple loci and/or schemes and present a subset of these as though it was a complete database. The loci and schemes chosen to belong to a set can be renamed when used with this set. The rationale for this is that in a database with disparate isolates and a large number of loci, the naming of these loci may have to be long to specify a species name. For example, you may have a database that contains multiple MLST schemes for different species, but since these schemes may use different fragments of the same genes they may have to be named something like 'Streptococcus_pneumoniae_MLST_aroE' to uniquely specify them. If we define a set for 'Streptococcus pneumoniae' we can then choose to only include S. pneumoniae loci and therefore shorten their names, e.g. to 'aroE'.

CHAPTER

TWO

BIGSDB DEPENDENCIES

2.1 Required packages

BIGSdb requires a number of software components to be installed:

2.1.1 Linux packages

- Apache2 web server with mod_perl2
- PostgreSQL database 9.5+
- Perl 5.22+
- BioPerl
- BLAST+
- EMBOSS
  - infoalign - use to extract alignment stats in Genome Comparator.
  - sixpack - used to translate sequences in multiple reading frames.
  - stretcher - used for sequence alignment in allele query.
- Ipcress - part of exonerate package - used to simulate PCR reactions which can be used to filter the genome to predicted amplification products.
- Xvfb - X virtual framebuffer - needed to support SplitsTree in command line mode as used in Genome Comparator.

2.1.2 Perl modules

These are included with most Linux distributions.

- Archive::Zip - Used to upload to iTOL.
- Bio::Biblio - This used to be part of BioPerl but will need to be installed separately if using BioPerl 1.6.920 or later.
- CGI (version 4.04+) - Common Gateway Interface requests and responses (used to be a core module but recently removed).
- Config::Tiny - Configuration file handling.
- Crypt::Eksblowfish::Bcrypt - Used for password hashing.

- Data::Random
- Data::UUID - Globally unique identifer handling for preference storage.
- DBD::Pg - PostgreSQL database driver for DBI.
- DBI - Database independent interface - module used to interact with databases.
- Email::MIME - Used to format E-mail messages.
- Email::Sender - Used to send E-mail messages by submission system.
- Email::Valid - Used to validate E-mails sent by job manager.
- Excel::Writer::XLSX - Used to export data in Excel format.
- Exception::Class - Exception handing.
- File::Map
- File::Type - Used to determine what type of file has been uploaded.
- IO::String
- JSON - Used to manipulate JSON data.
- List::MoreUtils (version 0.28+).
- Log::Dispatch::File - Object for logging to file.
- Log::Log4perl - Configurable status and error logging.
- LWP::UserAgent - Used to upload via API
- Net::Oauth - Required for REST authentication (this needs to be installed even if you are not using REST).
- Parallel::ForkManager - Required for multi-threading tools and plugins.
- PDL - Required for LINcode calculations.
- Template - Required for formatting analysis results.
- Time::Duration - Used by Job Viewer to display elapsed time in rounded units.
- TOML - Used to define dashboard layouts.
- Try::Tiny
- XML::Parser::PerlSAX - part of libxml-perl - Used to parse XML configuration files.

2.1.3 Optional packages

Installing these packages will enable extra functionality, but they are not required by the core BIGSdb package.

- MAFFT 6.8+ - sequence alignment used by some plugins.
- Muscle - sequence alignment used by some plugins.
- Splitstree4 - used by GenomeComparator plugin.
- PostGIS - needed for geographical mapping.
- WeasyPrint - needed to generate PDF files in the Genome Reports plugin.

CHAPTER

THREE

INSTALLATION AND CONFIGURATION OF BIGSDB

3.1 Software installation

BIGSdb consists of two main Perl scripts, bigsdb.pl and bigscurate.pl, that run the query and curator's interfaces respectively. These need to be located somewhere within the web cgi-bin directories. In addition, there are a large number of library files, used by both these scripts, that are installed by default in /usr/local/lib/BIGSdb. Plugin scripts are stored within a 'Plugins' sub-directory of this library directory.

All databases on a system can use the same instance of the scripts, or alternatively any database can specify a particular path for each script, enabling these script directories to be protected by apache htaccess directives.

- *Software requirements*
- Download from SourceForge.net or GitHub.

1. Unpack the distribution package in a temporary directory:

```
gunzip bigsdb_1.x.x.tar.gz
tar xvf bigsdb_1.x.x.tar
```

2. Copy the bigsdb.pl and bigscurate.pl scripts to a subdirectory of your web server's cgi-bin directory. Make sure these are readable and executable by the web server daemon.
3. Copy the contents of the lib directory to /usr/local/lib/BIGSdb/. Make sure you include the Plugins and Offline directories which are subdirectories of the main lib directory.
4. Copy the javascript directory to the root directory of your website, i.e. accessible from http://your_website/javascript/.
5. Copy the css directory to root directory of your website, i.e. accessible from http://your_website/css/.
6. Copy the webfonts directory to the root directory of your website, i.e. accessible from http://your_website/webfonts/.
7. Copy the images directory to the root directory of your website, i.e. accessible from http://your_website/images/.
8. Copy the contents of the conf directory to /etc/bigsdb/. Check the paths of helper applications and database names in the bigsdb.conf file and modify for your system.
9. Create a PostgreSQL database user called apache - this should not have any special priveleges. First you will need to log in as the postgres user:

```
sudo su postgres
```

   Then use the createuser command to do this, e.g.

```
createuser apache
```

From the psql command line, set the apache user password:

```
psql
ALTER ROLE apache WITH PASSWORD 'remote';
```

10. Create PostgreSQL databases called bigsdb_auth, bigsdb_prefs and bigsdb_refs using the scripts in the sql directory. Create the database using the createdb command and set up the tables using the psql command.

```
createdb bigsdb_auth
psql -f auth.sql bigsdb_auth
createdb bigsdb_prefs
psql -f prefs.sql bigsdb_prefs
createdb bigsdb_refs
psql -f refs.sql bigsdb_refs
```

11. Create a writable temporary directory in the root of the web site called tmp, i.e. accessible from http://your_website/tmp.

12. Create a log file, bigsdb.log, in /var/log owned by the web server daemon, e.g.

```
touch /var/log/bigsdb.log
chown www-data /var/log/bigsdb.log
```

(substitute www-data for the web daemon user).

3.2 Configuring PostgreSQL

PostgreSQL can be configured in many ways and how you do this will depend on your site requirements.

The following security settings will allow the appropriate users 'apache' and 'bigsdb' to access databases without allowing all logged in users full access. Only the UNIX users 'postgres' and 'webmaster' can log in to the databases as the Postgres user 'postgres'.

You will need to edit the pg_hba.conf and pg_ident.conf files. These are found somewhere like /etc/postgresql/9.1/main/

3.2.1 pg_hba.conf

```
# Database administrative login by UNIX sockets
local   all         postgres                            ident map=mymap

# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

# "local" is for Unix domain socket connections only
local   all         all                                 ident map=mymap
# IPv4 local connections:
host    all         all         127.0.0.1/32          md5
# IPv6 local connections:
host    all         all         ::1/128               md5
```

3.2.2 pg_ident.conf

```
# MAPNAME      SYSTEM-USERNAME    PG-USERNAME

  mymap        postgres           postgres
  mymap        webmaster          postgres
  mymap        www-data           apache
  mymap        bigsdb             bigsdb
  mymap        bigsdb             apache
```

You may also need to change some settings in the postgresql.conf file. As an example, a configuration for a machine with 16GB RAM, allowing connections from a separate web server may have the following configuration changes made:

```
listen_addresses = '*'
max_connections = 200
shared_buffers = 1024Mb
work_mem = 8Mb
effective_cache_size = 8192Mb
stats_temp_directory = '/dev/shm'
```

Setting stats_temp_directory to /dev/shm makes use of a ramdisk usually available on Debian or Ubuntu systems for frequently updated working files. This reduces a lot of unneccessary disk access.

See Tuning Your PostgreSQL Server for more details.

Restart PostgreSQL after any changes, e.g.

```
/etc/init.d/postgresql restart
```

3.3 Setting global connection parameters

Global database connection parameters can be entered in /etc/bigsdb/db.conf. This allows you to set default values for the host, port, user and password. Default values are as follows:

- dbhost: localhost
- dbport: 5432
- dbuser: apache
- dbpassword: remote

These can all be over-ridden in individual *database configuration config.xml files* using the terms host, port, user, and password.

3.4 Site-specific configuration

Site-specific configuration files are located in /etc/bigsdb by default.

- `bigsdb.conf` - main configuration file
- `logging.conf` - error logging settings. See log4perl project website for advanced configuration details.

Breadcrumb navigation links can be configured with a file called breadcrumbs.conf, placed either in the database configuration directory, the root directory of the website, or in /etc/bigsdb/conf. The file describes links that are higher in the hierarchy than the database index page. The file consists of lines that contain link text separated by a pipe symbol (|) followed by a URL for that link, e.g.

```
Home|/
Organisms|/databases/
```

Global announcments can be made in a banner that appears on each database contents page. This is useful for service announcements such as for planned maintenance. Place a HTML file called announcement.html in /etc/bigsdb including the text that you wish to appear.

3.5 Setting up the offline job manager

To run plugins that require a long time to complete their analyses, an offline job manager has been developed. The plugin will save the parameters of a job to a job database and then provide a link to the job status page. An offline script, run frequently from CRON, will then process the job queue and update status and outputs via the job status page.

1. Create a 'bigsdb' UNIX user, e.g.:

   ```
   sudo useradd -s /bin/sh bigsdb
   ```

2. As the postgres user, create a 'bigsdb' user and create a bigsdb_jobs database using the jobs.sql SQL file, e.g.:

   ```
   createuser bigsdb [no need for special priveleges]
   createdb bigsdb_jobs
   psql -f jobs.sql bigsdb_jobs
   ```

   From the psql command line, set the bigsdb user password::

   ```
   psql
   ALTER ROLE bigsdb WITH PASSWORD 'bigsdb';
   ```

3. Set up the jobs parameters in the /etc/bigsdb/bigsdb.conf file, e.g.:

   ```
   jobs_db=bigsdb_jobs
   max_load=8
   ```

   The jobs script will not process a job if the server's load average (over the last minute) is higher than the max_load parameter. This should be set higher than the number of processor cores or you may find that jobs never run on a busy server. Setting it to double the number of cores is probably a good starting point.

4. Copy the job_logging.conf file to the /etc/bigsdb directory.

5. Set the script to run frequently (preferably every minute) from CRON.

   Copy bigsjobs.pl to /usr/local/bin

You should install xvfb, which is a virtual X server that may be required for third party applications called from plugins. This is required, for example, for calling splitstree4 from the Genome Comparator plugin.

Add the following to /etc/crontab::

```
* * * * * bigsdb xvfb-run -a /usr/local/bin/bigsjobs.pl
```

(set to run every minute from the 'bigsdb' user account).

If you'd like to run this more frequently, e.g. every 30 seconds, multiple entries can be added to CRON with an appropriate sleep prior to running, e.g.:

```
* * * * * bigsdb  xvfb-run -a /usr/local/bin/bigsjobs.pl
* * * * * bigsdb  sleep 30;xvfb-run -a /usr/local/bin/bigsjobs.pl
```

6. Create a log file, bigsdb_jobs.log, in /var/log owned by 'bigsdb', e.g.:

```
sudo touch /var/log/bigsdb_jobs.log
sudo chown bigsdb /var/log/bigsdb_jobs.log
```

3.6 Setting up the submission system

The submission system allows users to submit new data to the database for curation. Submissions are placed in a queue for a curator to upload. All communication between submitters and curators can occur via the submission system.

1. Create a writable submissions directory in the root of the web site called submissions, i.e. accessible from http://your_website/submissions. This is used for file uploads. The directory should be writable by the Apache web daemon (user 'www-data' on Debian/Ubuntu systems). If you are running the *RESTful interface* the directory should also be writable by the bigsdb user. To ensure this, make the directory group-writable and add the bigsdb user to the apache group ('www-data' on Debian/Ubuntu systems). If you will be allowing submissions via the RESTful interface, you should also add the apache user ('www-data' on Debian/Ubuntu systems) to the bigsdb group, e.g.

```
sudo usermod -a -G www-data bigsdb
sudo usermod -a -G bigsdb www-data
```

The actual directory can be outside of the web root and made accessible using a symlink provided your Apache configuration allows this, e.g. the default location is /var/submissions symlinked to /var/www/submissions (assuming your web site is located in /var/www), e.g.

```
sudo touch /var/submissions
sudo chown www-data:www-data /var/submissions
sudo chmod 775 /var/submissions
sudo ln -s /var/submissions /var/www
```

2. Set the submission_dir location in bigsdb.conf.
3. Set the smtp_server in bigsdb.conf to the IP or DNS name of your organisation's SMTP relay. Depending on how your E-mail system is configured, you may be able to use the localhost address (127.0.0.1).
4. Make sure the curate_script and query_script values are set in bigsdb.conf. These point to the web-accessible location of the web scripts and are required to allow curators to be directed between the web interfaces as needed.
5. Set submissions="yes" in the system tag of the *database config.xml file* of each database for which submissions should be enabled.

3.7 Setting up a site-wide user database

A site-wide user database allows users to register themselves for accounts and associate these with specific databases. It means that a single set of log-in credentials can be used across databases, rather than each database maintaining its own.

Users can access/update their account details by calling the bigsdb.pl script without any additional attributes, e.g. http://website/cgi-bin/bigsdb.pl.

Site admins can access administration features by calling the bigscurate.pl script without any additional attributes.

1. Create a user database, e.g. pubmlst_bigsdb_users:

```
createdb pubmlst_bigsdb_users
psql -f users.sql pubmlst_bigsdb_users
```

   Set up sync_user_dbase_users.pl to run every hour as a CRON JOB, e.g. in /etc/crontab, add the following to run this at 5 minutes past each hour

```
05  *  *  *  *  bigsdb   /usr/local/bin/sync_user_dbase_users.pl --user_database␣
↪pubmlst_bigsdb_users
```

   Add the user database details to each database that you want to allow to use it.

   You need to *add the users database details* to each client database that will use it.

2. If you want to allow users to register themselves you need to modify bigsdb.conf.

   You can define multiple user databases (as a comma-separated list) but usually you would have just one. Define this using the site_user_dbs attribute. Use a short domain (site) name separated by a pipe (|) and the name of the database, e.g. add the following to /etc/bigsdb.conf:

```
site_user_dbs=PubMLST|pubmlst_bigsdb_users
```

   Make sure default database connection parameters are set in /etc/bigsdb/db.conf.

3. Set up site admin user in new user database. This has to be done manually - other users will either be able to register themselves or be created by curators from other databases.:

```
psql pubmlst_bigsdb_users
INSERT INTO USERS (user_name,surname,first_name,email,affiliation,
  date_entered,datestamp,status) VALUES ('kjolley','Jolley','Keith',
  'keith.jolley@biology.ox.ac.uk','University of Oxford, UK','now','now',
  'validated');
```

   Set the password for this user using the add_user.pl script (change XXXXXXXX to the password value):

```
add_user.pl -a -d pubmlst_bigsdb_users -n kjolley -p XXXXXXXX
```

   Add specific permissions that this admin user can have by directly adding the following terms to the permissions table:

   - set_site_user_passwords:
     - Allow admin to set user passwords.
   - import_dbase_configs:
     - Allow admin to define which database configurations are made available for registration.

- merge_users
  - Allow admin to merge user accounts.
- modify_users
  - Allow admin to edit user details.

e.g.

```
psql pubmlst_bigsdb_users
INSERT INTO permissions (user_name,permission,curator,datestamp) VALUES
  ('kjolley','import_dbase_configs','kjolley','now');
```

4. Specific *permissions can be set for curators* in individual databases:
   - import_site_users
     - This allows the curator to import site users in to the database.
   - modify_site_users
     - You may not wish to do this! - It allows the curator of any database with this permission to change the details of a user that may be used on other databases on the site.
5. HTML header files can be defined for use when bigsdb.pl or bigscurate.pl are called without a database configuration, such as when a user is registering or modifying their user details. These files, site_header.html, site_footer.html, site_curate_header.html and site_curate_footer.html should be placed in the root directory of the web site.

3.8 Periodically delete temporary files

There are two temporary directories (one public, one private) which may accumulate temporary files over time. Some of these are deleted automatically when no longer required but some cannot be cleaned automatically since they are used to display results after clicking a link or to pass the database query between pages of results.

The easiest way to clean the temp directories is to run a cleaning script periodically, e.g. create a root-executable script in /etc/cron.hourly containing the following::

```
#!/bin/sh
#Remove temp BIGSdb files from secure tmp folder older than 1 week.
find /var/tmp/ -name '*BIGSdb_*' -type f -mmin +10080 -exec rm -f {} \; 2>/dev/null

#Remove .jnlp files from web tree older than 1 day
find /var/www/tmp/ -name '*.jnlp' -type f -mmin +1440 -exec rm -f {} \; 2>/dev/null

#Remove other tmp files from web tree older than 1 week
find /var/www/tmp/ -type f -mmin +10080 -exec rm -f {} \; 2>/dev/null
```

3.9 Prevent preference database getting too large

The preferences database stores user preferences for BIGSdb databases running on the site. Every user will have a globally unique identifier (guid) stored in this database along with a datestamp indicating the last access time. On public databases that do not require logging in, this guid is stored as a cookie on the user's computer. Databases that require logging in use a combination of database and username as the identifier. Over time, the preferences database can get quite large since every unique user will result in an entry in the database. Since many of these entries represent casual users, or even web indexing bots, they can be periodically cleaned out based on their last access time. A weekly CRON job can be set up to remove any entries older than a defined period. For example, the following line entered in /etc/crontab will remove the preferences for any user that has not accessed any database in the past 6 months (the script will run at 6pm every Sunday).

```
#Prevent prefs database getting too large
00   18 *  *  0  postgres    psql -c "DELETE FROM guid WHERE last_accessed < NOW() -␣
↪INTERVAL '6 months'" bigsdb_prefs
```

3.10 Log file rotation

Set the log file to auto rotate by adding a file called 'bigsdb' with the following contents to /etc/logrotate.d:

```
/var/log/bigsdb.log {
  weekly
  rotate 4
  compress
  copytruncate
  missingok
  notifempty
  create 640 root adm
}

/var/log/bigsdb_jobs.log {
  weekly
  rotate 4
  compress
  copytruncate
  missingok
  notifempty
  create 640 root adm
}
```

3.11 Upgrading BIGSdb

Major version changes, e.g. 1.7 -> 1.8, indicate that there has been a change to the underlying database structure for one or more of the database types. Scripts to upgrade the database are provided in sql/upgrade and are named by the database type and version number. For example, to upgrade an isolate database (bigsdb_isolates) from version 1.7 to 1.8, log in as the postgres user and type:

```
psql -f isolatedb_v1.8.sql bigsdb_isolates
```

Upgrades are sequential, so to upgrade from a version earlier than the last major version you would need to upgrade to the intermediate version first, e.g. to go from 1.6 -> 1.8, requires upgrading to 1.7 first.

Minor version changes, e.g. 1.8.0 -> 1.8.1, have no modifications to the database structures. There will be changes to the Perl library modules and possibly to the contents of the Javascript directory, images directory and CSS files.

3.12 Running the BIGSdb RESTful interface

BIGSdb has an Application Programming Interface (API) that allows third-party applications to access the data within the databases. The script that runs this is called bigsrest.pl. This is a Dancer2 application that can be run using a wide range of options, e.g. as a stand-alone script, using Perl webservers with plackup, or from apache. Full documentation for deploying Dancer2 applications can be found online.

The script requires a new database that describes the resources to make available. This is specified in the bigsdb.conf file as the value of the 'rest_db' attribute. By default, the database is named bigsdb_rest.

A SQL file to create this database can be found in the sql directory of the download archive. It is called rest.sql. To create the database, as the postgres user, navigate to the sql directory and type

```
createdb bigsdb_rest
psql -f rest.sql bigsdb_rest
```

This database will need to be populated using psql or any tool that can be used to edit PostgreSQL databases. The database contains three tables that together describe and group the databases resources that will be made available through the API. The tables are:

- **resources**
  - **this contains two fields (both compulsory):**
    - **dbase_config** - the name of the database configuration used with the database. This is the same as the name of the directory that contains the config.xml file in the /etc/bigsdb/dbases directory.
    - **description** - short description of the database.
- **groups (used to group related resources together)**
  - **this contains two fields (compulsory fields shown in bold):**
    - **name** - short name of group. This is usually a single word and is also the key that links resources to groups.
    - **description** - short description of group.
    - long_description - fuller description of group.
- **group_resources (used to add resources to groups)**
  - **this contains two fields (both compulsory)**
    - **group_name** - name of group. This must already exist in the groups table.
    - **dbase_config** - the name of database resource. This must already exist in the resources table.

For example, to describe the PubMLST resources for Neisseria, connect to the bigsdb_rest database using psql,

```
psql bigsdb_rest
```

Then enter the following SQL commands. First add the database resources:

```
INSERT INTO resources (dbase_config,description) VALUES
('pubmlst_neisseria_seqdef','Neisseria sequence/profile definitions');
INSERT INTO resources (dbase_config,description) VALUES
('pubmlst_neisseria_isolates','Neisseria isolates');
```

Then create a 'neisseria' group that will contain these resources:

```
INSERT INTO groups (name,description) VALUES
('neisseria','Neisseria spp.');
```

Finally, add the database resources to the group:

```
INSERT INTO group_resources (group_name,dbase_config) VALUES
('neisseria','pubmlst_neisseria_seqdef');
INSERT INTO group_resources (group_name,dbase_config) VALUES
('neisseria','pubmlst_neisseria_isolates');
```

The REST API will need to run on its own network port. By default this is port 3000. To run as a stand-alone script, from the script directory, as the bigsdb user, simply type:

```
./bigsrest.pl
```

This will start the API on port 3000. You will be able to check that this is running using a web browser by navigating to http://localhost:3000 on the local machine, or using the server IP address from a remote machine. You may need to modify your server firewall rules to allow connection to this port.

Running as a stand-alone script is useful for testing, but you can achieve much better performance using a Perl webserver with plackup. There are various options to choose. PubMLST uses Starman.

To run the API using Starman, type the following as the bigsdb user:

```
plackup -a /var/rest/bigsrest.pl -s Starman -E deployment
```

where the value of -a refers to the location of the bigsrest.pl script. Starman defaults to using port 5000.

Different Linux distributions use different means to control services/daemons. To start the REST interface on system boot on systems using upstart, create a file called bigsdb-rest.conf in /etc/init. The contents of this file should be something like (modify file paths as appropriate):

```
description "Start BIGSdb REST interface"
version "1.0"
author "Keith Jolley"

start on runlevel [12345]

## tell upstart we're creating a daemon
expect fork

script

exec su -s /bin/sh -c 'exec "$0" "$@"' bigsdb -- /usr/local/bin/plackup -a /var/rest/
↪bigsrest.pl -s Starman -E deployment

end script
```

The service will then start automatically on boot or can be manually started by calling:

```
sudo service bigsdb-rest start
```

For systems using systemd, create a file in /etc/systemd/system called bigsdb-rest.service with the following contents (again, modify file paths as appropriate):

```
[Unit]
Description=BIGSdb REST interface
After=network.target

[Service]
User=bigsdb
ExecStart=/usr/bin/plackup -a /var/rest/bigsrest.pl -s Starman -E deployment
Restart=always

[Install]
WantedBy=multi-user.target
```

To start the service automatically on boot you need to enable it:

```
sudo systemctl enable bigsdb-rest.service
```

It can also be manually started by calling:

```
sudo systemctl start bigsdb-rest.service
```

3.12.1 Proxying the API to use a standard web port

Usually you will want your API to be available on the standard web port 80. To do this you will need to set up a virtual host using a different domain name from your web site to proxy the API port. For example, PubMLST has a separate domain 'http://rest.pubmlst.org' for its API. This is set up as a virtual host directive in apache with the following configuration file:

```
<VirtualHost *>
  ServerName rest.pubmlst.org
  DocumentRoot /var/rest
  ServerAdmin keith.jolley@biology.ox.ac.uk
   <Directory /var/rest>
    AllowOverride None
    Require all granted
  </Directory>

  ProxyPass / http://rest.pubmlst.org:5000/
  ProxyPassReverse / http://rest.pubmlst.org:5000/

  <Proxy *>
      Order allow,deny
      Allow from all
  </Proxy>

  ErrorLog  /var/log/apache2/rest.pubmlst.org-error.log
  CustomLog /var/log/apache2/rest.pubmlst.org-access.log common
```

(continues on next page)

(continued from previous page)

```
</VirtualHost>
```

You should also set 'rest_behind_proxy=1' in bigsdb.conf.

3.13 Enabling database logging of web and API access

User access to both the web interface and API can be logged within the bigsdb_auth and bigsdb_rest databases respectively. Each of these contains a table called `log` in which the IP address, username, and page called are recorded with a timestamp. To enable logging, you need to set the following in bigsdb.conf:

```
web_log_to_db=1
rest_log_to_db=1
```

Logging requires writing to the database on each page access so there is a very small performance penalty to enabling this. The tables are however unlogged (i.e. data are not written to the PosgreSQL write-ahead log) which makes them considerably faster than ordinary tables but data in them will be lost in the event of a database crash or unclean shutdown.

As every page access is recorded the log tables will grow in size over time. It is recommended that they are pruned regularly to remove records older than a specified period of time - this may also be required by GDPR! The easiest way to do this is to set up a scheduled CRON job by adding the following to /etc/crontab:

```
0  *  *  *  *  postgres psql -c "DELETE FROM log WHERE timestamp < NOW() - INTERVAL '7␣
↪days'" bigsdb_rest > /dev/null
10 *  *  *  *  postgres psql -c "DELETE FROM log WHERE timestamp < NOW() - INTERVAL '7␣
↪days'" bigsdb_auth > /dev/null
```

CHAPTER

FOUR

DATABASE SETUP

There are two types of BIGSdb database:

- sequence definition databases, containing
  - allele sequences and their identifiers
  - scheme data, e.g. MLST profile definitions
- isolate databases, containing
  - isolate provenance metadata
  - genome sequences
  - allele designations for loci defined in sequence definition databases.

These two databases are independent but linked. A single isolate database can communicate with multiple sequence definition databases and vice versa. Different access restrictions can be placed on different databases.

Databases are described in XML files telling BIGSdb everything it needs to know about them. Isolate databases can have any fields defined for the isolate table,allowing customisation of metadata - these fields are described in the XML file (config.xml) and must match the fields defined in the database itself.

4.1 Creating databases

There are templates available for the sequence definition and isolate databases. These are SQL scripts found in the sql directory.

To create a database, you will need to log in as the postgres user and use these templates. For example to create a new sequence definition database called bigsdb_test_seqdef, navigate to the sql directory and log in as the postgres user, e.g.

```
sudo su postgres
```

then

```
createdb bigsdb_test_seqdef
psql -f seqdef.sql bigsdb_test_seqdef
```

Create an isolate database the same way:

```
createdb bigsdb_test_isolates
psql -f isolatedb.sql bigsdb_test_isolates
```

The standard fields in the isolate table are limited to essential fields required by the system as well as country and year. To add new fields, you need to log in to the database and alter this table. For example, to add fields for age and sex, first log in to the newly created isolate database as the postgres user:

```
psql bigsdb_test_isolates
```

and alter the isolate table:

```
ALTER TABLE isolates ADD age int;
ALTER TABLE isolates ADD sex text;
```

If you want to use the geography_point field type, used for storing and mapping GPS coordinates, then you will need to install the PostGIS module for PostgreSQL and enable this within the database as follows:

```
CREATE EXTENSION postgis;
```

To create a geography_point field for location alter the isolate table as below:

```
ALTER TABLE isolates ADD location geography(POINT,4326);
```

[SRID 4326 represents spatial data using longitude and latitude coordinates on the Earth's surface.]

Fields can also be linked to a GPS lookup table and can then be mapped. If this is enabled then you need to ensure that PostGIS is installed and create the lookup table by running the isolatedb_geocoding.sql script against the isolate database with the following:

```
psql -f isolatedb_geocoding.sql bigsdb_test_isolates
```

Remember that any fields added to the table need to be *described in the config.xml file* for this database.

The xml directory of the software archive contains example XML files for sequence definition and isolate databases (rename these to config.xml). The isolates_config.xml file contains the minimum required isolate table fields and matches the isolate table that will be generated using the isolatedb.sql SQL script.

4.2 Database-specific configuration

Each BIGSdb database on a system has its own configuration directory, by default in /etc/bigsdb/dbases. The database has a short configuration name used to specify it in a web query and this matches the name of the configuration sub-directory, e.g. http://pubmlst.org/cgi-bin/bigsdb/bigsdb.pl?db=pubmlst_neisseria_isolates is the URL of the front page of the PubMLST Neisseria isolate database whose configuration settings are stored in /etc/bigsdb/dbases/pubmlst_neisseria_isolates. This database sub-directory contains a number of optional files:

- `config.xml` - the database configuration file. Fields defined here correspond to fields in the isolate table of the database.
- banner.html - optional file containing text that will appear as a banner within the database index pages. HTML markup can be used within this text.
- header.html - HTML markup that is inserted at the top of all pages. This can be used to set up site-specific menubars and logos.
- footer.html - HTML markup that is inserted at the bottom of all pages.
- curate_header.html - HTML markup that is inserted at the top of all curator's interface pages.
- curate_footer.html - HTML markup that is inserted at the bottom of all curator's interface pages.

- profile_submit.html - HTML markup for text that is inserted in to the submission interface prior to profile submission finalization. This can be used to add specific instructions such as the requirement to make an isolate submission.
- allele_submit.html - HTML markup for text that is inserted in to the submission interface prior to allele submission finalization. This can be used to add specific instructions such as the requirement to attach Sanger trace files.
- isolate_submit.html - HTML markup for text that is inserted in to the submission interface prior to isolate submission finalization. This can be used to add specific instructions such as the request to also make a new profile submission if the isolate has a new profile.
- profile_curate.html - HTML markup for text that is inserted on submission curation page if profile submissions are pending. This can be used to add specific information to curators.
- allele_curate.html - HTML markup for text that is inserted on submission curation page if allele submissions are pending. This can be used to add specific information to curators.
- isolate_curate.html - HTML markup for text that is inserted on submission curation page if isolate submissions are pending. This can be used to add specific information to curators.
- registration_success.txt - Text file containing message content to be used in an automated E-mail when granting access to a user who has requested access to the database using the site-wide account system (where auto-registration is not enabled).
- registration.html - HTML markup for text that will appear on the login page for the current database. This appears right before the "Log in" button.

The header and footer files can alternatively be placed in the root directory of the web site, or in /etc/bigsdb, for site-wide use. If files exist in multiple locations, they are used in the following order of preference: database config directory > web root directory > /etc/bigsdb.

There are four additional files, site_header.html, site_footer.html, curate_site_header.html and curate_site_footer.html which are used when either bigsdb.pl or bigscurate.pl are called without a database configuration. These should be placed in the root directory of the web site or in /etc/bigsdb.

You can also add HTML meta attributes (such as a favicon) by including a file called meta.html in the database configuration directory. For example to set a favicon this file can contain something like the following:

```
<link rel="shortcut icon" href="/favicon.ico" type="image/ico" />
```

These attributes will appear in the <head> section of the HTML page.

4.3 XML configuration attributes used in config.xml

The following lists describes the attributes used in the config.xml file that is used to describe databases.

4.3.1 Isolate database XML attributes

Please note that database structure described by the field elements must match the physical structure of the database isolate table. Required attributes are in **bold**:

```
<db>
```

Top level element. Contains child elements: system and field.:

```
<system>
```

Any value set here can be overridden in a *system.overrides file*.

- **authentication**
  - Method of authentication: either 'builtin' or 'apache'. See *user authentication*.
- **db**
  - Name of database on system.
- **dbtype**
  - Type of database: either 'isolates' or 'sequences'.
- **description**
  - Description of database used throughout interface (see also 'formatted_description').
- align_limit
  - Overrides the sequence export record alignment limit in the Sequence Export plugin. Default: '200'.
- all_plugins
  - Enable all appropriate plugins for database: either 'yes' or 'no', default 'no'.
- alternative_codon_tables
  - Enable alternative codon tables: either 'yes' or 'no'. Set to 'yes' to allow individual isolates to use a different codon table than the default (defined by the 'codon_table' attribute), default is 'no'.
- annotation
  - Semi-colon separated list of accession numbers with descriptions (separated by a |), eg. 'AL157959|Z2491;AM421808|FAM18;NC_002946|FA 1090;NC_011035|NCCP11945;NC_014752|020-06'. Currently used only by Genome Comparator plugin.
- BLAST
  - Enable Blast plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the Blast plugin can be disabled by setting this attribute to 'no'.
- BURST
  - Enable BURST plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the BURST plugin can be disabled by setting this attribute to 'no'.

- cache_schemes
  - Enable automatic refreshing of scheme field caches when batch adding new isolates: either 'yes' or 'no', default 'no'.
  - See *scheme caching*.
- CodonUsage
  - Enable Codon Usage plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the Codon Usage plugin can be disabled by setting this attribute to 'no'.
- codon_table
  - Set the id of the global codon table to use. See https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi for a list with the of ids and their description. This can be overridden on a per-isolate basis if alternative_start_codons is set to 'yes'. Default value is "11".
- codon_usage_limit
  - Overrides the record limit for the Codon Usage plugin. Default: '500'.
- contig_analysis_limit
  - Overrides the isolate number limit for the Contig Export plugin. Default: '1000'.
- ContigExport
  - Enable contig export plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the contig export plugin can be disabled by setting this attribute to 'no'.
- country_field
  - Sets the field in which country is stored. Default: 'country'. This is needed for mapping.
- curate_config
  - The database configuration that should be used for curation if different from the current configuration. This is used when the submission system is being used so that curation links in the 'Manage submissions' pages for curators load the correct database configuration.
- curate_link
  - URL to curator's interface, which can be relative or absolute. This will be used to create a link in the public interface dropdown menu.
- curate_path_includes
  - Partial path of the bigscurate.pl script used to curate the database. See user authentication.
- curate_script
  - Relative web path to curation script. Default 'bigscurate.pl' (version 1.11+).
  - This is only needed if automated submissions are enabled. If bigscurate.pl is in a different directory from bigsdb.pl, you need to include the whole web path, e.g. /cgi-bin/private/bigsdb/bigscurate.pl.
- curators_only
  - Set to 'yes' to prevent ordinary authenticated users having access to database configuration. This is only effective if read_access is set to 'authenticated_users'. This may be useful if you have different configurations for curation and querying with some data hidden in the configuration used by standard users. Default 'no'.
- daily_pending_submissions

  - Overrides the daily limit on pending submissions that a user can submit via the web submission system. Default: '15'.

- daily_rest_submissions_limit
  - Overrides the limit on number of submissions that can be made to the database via the RESTful interface. This is useful to prevent flooding of the submission system by aberrant scripts. Default: '100'.
- default_access
  - The default access to the database configuration, either 'allow' or 'deny'. If 'allow', then specific users can be denied access by creating a file called 'users.deny' containing usernames (one per line) in the configuration directory. If 'deny' then specific users can be allowed by creating a file called 'users.allow' containing usernames (one per line) in the configuration directory. See *default access*.
- default_private_records
  - The default number of private isolate records that a user can upload. The user account must have a status of either 'submitter', 'curator', or 'admin'. This value is used to set the private_quota field when creating a new user record (which can be overridden for individual users). Changing it will not affect the quotas of existing users. Default: '0'.
- default_seqdef_config
  - Isolate databases only: Name of the default seqdef database configuration used with this database. Used to automatically fill in details when adding new loci.
- default_seqdef_dbase
  - Isolate databases only: Name of the default seqdef database used with this database. Used to automatically fill in details when adding new loci.
- default_seqdef_script
  - Isolate databases only: URL of BIGSdb script running the seqdef database (default: '/cgi-bin/bigsdb/bigsdb.pl').
- delete_retire_only
  - Set to 'yes' to retire the id of any isolate that is deleted. This prevents re-use of ids. This setting will override the global setting in bigsdb.conf.
- disable_updates
  - Set to 'yes' to prevent updates. This is useful when moving databases or temporarily running on a backup server.
- disable_update_message
  - Message shown when updates are disabled.
- eav_fields
  - Name to call sparsely-populated fields. Default: 'secondary metadata'.
- eav_field_icon
  - Icon class from FontAwesome to use on isolate info page for sparsely- populated fields. Default 'fas fa-microscope'.
- eav_groups
  - Comma-separated list of category names that sparsely-populated fields can be grouped in to. If this value is set, a category drop-down list will appear when adding or updating sparsely-populated fields. You can add an icon to appear by following the name with a pipe symbol (|) and an icon class from the FontAwesome library, e.g. 'Vaccine reactivity|fas fa-syringe,Risk factors|fas fa-smoking'.

- export_limit
  - Overrides the default allowed number of data points (isolates x columns) to export. Default: '25000000'.
- fast_scan
  - Sets whether fast mode scanning is enabled via the web interface. This will scan all loci together, using exemplar sequences. In cases where multiple loci are being scanned this should be significantly faster than the standard locus-by-locus scan, but it will take longer for the first results to appear. *Allele exemplars* should be defined if you enable this option. Set to 'yes' to enable. Default: 'no'.
- field_groups
  - Comma-separated list of category names that standard isolate fields can be grouped in to in the isolate information page. You can add an icon to appear by following the name with a pipe symbol (|) and an icon class from the FontAwesome library, e.g. 'Antimicrobial resistance|fas fa-capsules'.
- fieldgroup1 - fieldgroup10
  - Allows multiple fields to be queried as a group. Value should be the name of the group followed by a colon (:) followed by a comma-separated list of fields to group, e.g. identifiers:id,strain,other_name.
- formatted_description
  - Markdown formatted description of database. If set, this will be used throughout the HTML interface wherever formatting can be applied (main body of text) and overrides the value set in 'db_description'. Currently only supports **italics** and ****bold****.
- genepresence_record_limit
  - Overrides the record number limit (isolates x loci) for the Gene Presence plugin. Default: 500000 (this can also be set globally in bigsdb.conf).
- genepresence_taxa_limit
  - Overrides the isolate limit for the Gene Presence plugin. Default: 10000 (this can also be set globally in bigsdb.conf).
- GenomeComparator
  - Enable Genome Comparator plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the Genome Comparator plugin can be disabled by setting this attribute to 'no'.
- genome_comparator_limit
  - Overrides the isolate number limit for the Genome Comparator plugin. Default: 1000 (this can also be set globally in bigsdb.conf).
- genome_comparator_max_ref_loci
  - Overrides the limit on number of loci allowed in a reference genome. Default: 10000.
- genome_comparator_threads
  - The number of threads to use for data gathering (BLAST, database queries) to populate data structure for Genome Comparator analysis. You should not set this to less than 2 as this will prevent job cancelling due to the way isolates are queued. Default: '2'.
- genome_submissions
  - Enable genome submissions (automated submission system): either 'yes' or 'no', default 'yes'.
  - To enable, you will also need to set submissions="yes". By default, genome submissions are enabled.
- hide_unused_schemes

  - Sets whether a scheme is shown in a main results table if none of the isolates on that page have any data for the specific scheme: either 'yes' or 'no', default 'no'.

- host
  - Host name/IP address of machine hosting isolate database, default 'localhost'.
- itol_record_limit
  - Overrides the maximum number of records that can be included in an ITOL job. Default: 2000 (this can also be set globally in bigsdb.conf).
- itol_seq_limit
  - Overrides the maximum number of sequeneces (records x loci) that can be included in an ITOL job. Default: 100,000 (this can also be set globally in bigsdb.conf).
- job_priority
  - Integer with default job priority for offline jobs (default:5).
- job_quota
  - Integer with number of offline jobs that can be queued or currently running for this database.
- labelfield
  - Field that is used to describe record in isolate info page, default 'isolate'.
- locus_aliases
  - Display locus aliases and use them in dropdown lists by default: must be either 'yes' or 'no', default 'no'. This option can be overridden by a user preference.
- locus_superscript_prefix
  - Superscript the first letter of a locus name if it is immediately following by an underscore, e.g. f_abcZ would be displayed as fabcZ within the interface: must be either 'yes' or 'no', default 'no'. This can be used to designate gene fragments (or any other meaning you like).
- maindisplay_aliases
  - Default setting for whether isolates aliases are displayed in main results tables: either 'yes' or 'no', default 'no'. This setting can be overridden by individual user preferences.
- max_contigs
  - Number of contigs above which genome submissions will be rejected. Default: 1000
- max_total_length
  - Total length (bp) above which genome submissions will be rejected. Default: 15000000
- Microreact
  - Enable Microreact plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the Microreact plugin can be disabled by setting this attribute to 'no'. Note that for the plugin to be active, a country field containing a defined list of allowed values and an integer year field must be defined in the isolates table. The plugin also requires microreact_token to be provided in bigsdb.conf.
- microreact_country_field
  - Overrides the field in which country is stored. Default: 'country'
- microreact_record_limit

  - Overrides the maximum number of records that can be included in a Microreact job. Default: 2000 (this can also be set globally in bigsdb.conf).

- microreact_seq_limit
  - Overrides the maximum number of sequences (records x loci) that can be included in an Microreact job. Default: 100,000 (this can also be set globally in bigsdb.conf).
- microreact_year_field
  - Overrides the field in which year is stored. Default: 'year'
- min_n50
  - Minimum N50 for genome submissions below which the submission will be rejected. Default: 10000
- min_total_length
  - Minimum total length (bp) for genome submissions below which the submission will be rejected. Default: 1000000.
- min_genome_size
  - Size in bp that is the minimum size of the sequence bin considered to represent a whole genome. This is used in the REST interface to differentiate records with genomes. You can also pass a 'genomes=1' attribute to the an isolate query form and this will populate the appropriate search to return genome records.
- new_version
  - Set to 'no' to prevent copying field value when creating a new version of the isolate record.
- noshow
  - Comma-separated list of fields not to use in breakdown statistic plugins.
- no_publication_filter
  - Isolate databases only: Switches off display of publication filter in isolate query form by default: either 'yes' or 'no', default 'no'.
- only_sets
  - Don't allow option to view the 'whole database' - only list sets that have been defined: either 'yes' or 'no', default 'no'.
- password
  - Password for access to isolates database, default 'remote'.
- pcr_limit
  - Overrides the isolate number limit for the in silico PCR plugin. Default: '10000'.
- PhyloViz
  - Enable third party PhyloViz plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the PhyloViz plugin can be disabled by setting this attribute to 'no'.
- port
  - Port number that the isolate host is listening on, default '5432'.
- privacy
  - Displays E-mail address for sender in isolate information page if set to 'no'. Default 'yes'.
- public_login

  - Optionally allow users to log in to a public database - this is useful as any jobs will be associated with the user and their preferences will also be linked to the account. Set to 'no' to disable. Default 'yes'.

- query_script
  - Relative web path to bigsdb script. Default 'bigsdb.pl' (version 1.11+).
  - This is only needed if automated submissions are enabled. If bigsdb.pl is in a different directory from bigscurate.pl, you need to include the whole web path, e.g. /cgi-bin/bigsdb/bigsdb.pl.
- read_access
  - Describes who can view data: either 'public' for everybody or 'authenticated_users' for anybody who has been able to log in. Default 'public'.
- related_databases
  - Semi-colon separated list of links to related BIGSdb databases on the system. This should be in the form of database configuration name followed by a '|' and the description, e.g. 'pubmlst_neisseria_seqdef|Typing'. This is used to populate the menu items.
- remote_contigs
  - Optionally allow the use of remote contigs. These are stored in a remote BIGSdb database, accessible via the RESTful API. Set to 'yes' to enable.
- rest_kiosk
  - If 'kiosk' attribute is set, then the REST interface will be disabled for the configuration unless a value is set here. The only supported value currently is 'sequenceQuery' which will enable API routes for querying sequences.
- rMLSTSpecies
  - Enable rMLST Species identifier plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the plugin can be disabled by setting this attribute to 'no'.
- script_path_includes
  - Partial path of the bigsdb.pl script used to access the database. See *user authentication*.
- search_sequence_variation
  - If loci are defined that have sequence variants (single amino acid or single nucleotide polymorphisms) defined in their respective typing databases, setting this attribute to 'yes' will enable the isolate database to be searched by whether alleles found in isolates have these variants or not. This is not enabled by default as there is a small, but unnecessary, performance hit in databases that don't have any such loci.
- separate_dataset
  - Treat database configuration as though it was a separate database for the purposes of handling submissions and curators: either 'yes' or 'no', default 'no'. Submissions will be tagged with the configuration name and will only be visible within that configuration. Curators can be limited to specific configurations by populating the curator_configs table. This also affects whether they are notified of submissions.
- SeqbinBreakdown
  - Enable Sequence bin breakdown plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the plugin can be disabled by setting this attribute to 'no'. Note that for the plugin to be active, a country field containing a defined list of allowed values and an integer year field must be defined in the isolates table.
- seq_export_limit

  - Overrides the sequence export limit (records x loci) in the Sequence Export plugin. Default: '1000000'.

- sets
  - Use *sets*: either 'yes' or 'no', default 'no'.
- set_id
  - Force the use of a specific set when accessing database via this XML configuration: Value is the name of the set.
- show_classification_schemes
  - Show similar isolates determined by classification schemes (if defined) on an isolate record page. Set to either 'yes' or 'no', default 'yes'.
- show_lincode_matches
  - Show similar isolates determined by LIN code prefixes (if defined) on an isolate record page. Set to either 'yes' or 'no', default 'yes'.
- start_codons
  - Semi-colon separated list of start codons to allow. Note that this list will replace the built-in defaults of ATG, GTG, and TTG, and is used for all functions that require recognising complete coding sequences, such as automated allele definition.
- start_id
  - Defines the minimum record id to be used when uploading new isolate records. This can be useful when it is anticipated that two databases may be merged and it would be easier to do so if the id numbers in the two databases were different. Default: '1'.
- submissions
  - Enable automated submission system: either 'yes' or 'no', default 'no' (version 1.11+).
  - The curate_script and query_script paths should also be set, either in the bigsdb.conf file (for site-wide configuration) or within the system attribute of config.xml.
- submissions_deleted_days
  - Overrides the default number of days before closed submissions are deleted from the system. Default: '90'.
- TagStatus
  - Enable Tag status plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the plugin can be disabled by setting this attribute to 'no'. Note that for the plugin to be active, a country field containing a defined list of allowed values and an integer year field must be defined in the isolates table.
- tblastx_tagging
  - Sets whether tagging can be performed using TBLASTX: either 'yes' or 'no', default 'no'.
- total_pending_submissions
  - Overrides the total limit on pending submissions that a user can submit via the web submission system. Default: '20'.
- user
  - Username for access to isolates database, default 'apache'.
- user_job_quota

  - Integer with number of offline jobs that can be queued or currently running for this database by any specific user - this parameter is only effective if users have to log in.

- user_projects
  - Sets whether authenticated users can create their own projects in order to group isolates: either 'yes' or 'no', default 'no'.
- view
  - Database view containing isolate data, default 'isolates'.
- views
  - Comma-separated list of views of the isolate table defined in the database. This is used to set a view for a set, or to restrict loci or schemes to a subset of isolate data.
- warn_max_contigs
  - Set a threshold for the number of contigs in a submitted genome assembly to trigger a warning in the submission interface. This value overrides the value set in bigsdb.conf.
- warn_max_total_length
  - Set an upper threshold for the total size of a submitted genome assembly to trigger a warning in the submission interface.
- warn_min_n50
  - Set a threshold for the minimum N50 value in a submitted genome assembly to trigger a warning in the submission interface. This value overrides the value set in bigsdb.conf.
- warn_min_total_length
  - Set a lower threshold for the total size of a submitted genome assembly to trigger a warning in the submission interface.
- webroot
  - URL of web root, which can be relative or absolute. This is used to provide a hyperlinked item in the dropdown menu. Default '/'.
- webroot_label
  - Label text for the breadcrumb link defined by the webroot value. This can be formatted using Markdown. Currently only supports **italics** and ****bold****.

```
<field>
```

Element content: Field name + optional list <optlist> of allowed values, e.g.:

```
<field type="text" required="no" length="40" maindisplay="no"
   web="http://somewebsite.com/cgi-bin/script.pl?id=[?]" optlist="yes">epidemiology
  <optlist>
   <option>carrier</option>
   <option>healthy contact</option>
   <option>sporadic case</option>
   <option>endemic</option>
   <option>epidemic</option>
   <option>pandemic</option>
  </optlist>
</field>
```

- **type**
  - Data type: int, text, float, bool, date, or geography_point.
- allow_submissions
  - Show in submission template and allow data to be submitted even if field is set as 'curate_only'. This has no effect on fields that do not have the 'curate_only' attribute as these fields are included in submissions by default. This attribute will be overridden if the field has the 'no_submissions' attribute set.
- annotation_metric
  - Use field for provenance annotation status metrics. The field should be expected to have a value for records with complete provenance data. Set to 'yes' to include.
- comments
  - Comments about the field. These will be displayed in the field description plugin and as tooltips within the curation interface.
- curate_only
  - Set to 'yes' to hide field unless logged-in user is a curator or admin. Set the 'allow_submissions' attribute to still include the field in the submission template so that it can be included in submissions of new records by standard users.
- default
  - Default value. This will be entered automatically in the web form but can be overridden.
- dropdown
  - Select if you want this field to have its own dropdown filter box on the query page. If the field has an option list it will use the values in it, otherwise all values defined in the database will be included: 'yes' or 'no', default 'no'. This setting can be overridden by individual user preferences.
- geography_point_lookup
  - Set to 'yes' if this field should be linked to a lookup table of GPS coordinates in order to facilitate mapping within the isolate information page and the Field Breakdown plugin. If any fields have this value set, you need to install the lookup tables by running the isolatedb_geocoding.sql script against the isolate database. This also requires that the PostgreSQL PostGIS module is installed.
- group
  - Fields can be grouped in the isolate information page by specifying the group attribute. The group name must be defined in the field_groups system attribute, otherwise the field will not be shown at all. If undefined, the field will be in the default provenance/primary metadata group.
- hide
  - Completely ignore field. This is useful if you access a database using different configuration files and a field is not relevant to a particular instance. See also *Over-riding values*.
- isolate_display
  - Set to 'no' to not show field on the isolate information page.
- length
  - Length of field, default 12.
- log_delete
  - Sets if the field value will be recorded in the log table if the isolate is deleted. Set to 'yes' or 'no', default is 'no'. The id and isolate name are always recorded if deletion is logged.

- maindisplay
  - Sets if field is displayed in the main table after a database search, 'yes' or 'no', default 'yes'. This setting can be overridden by individual user preferences.
- max
  - Maximum value for integer and date types. Special values such as CURRENT_YEAR and CURRENT_DATE can be used.
- min
  - Minimum value for integer and date types.
- multiple
  - Sets if field allows multiple values to be set for it, 'yes' or 'no', default 'no'. If set to 'yes', then the underlying field in the database must be an ARRAY type, e.g. text[].
- no_curate
  - Setting this will hide the field in the curator interface and prevent it from being manually modified. This is useful for fields that are populated by automated scripts or database triggers. Can be 'yes' or 'no', default 'no'.
- no_submissions
  - Setting this will hide the field in the submission template. The field is still available if it is added back to the template manually.
- optlist
  - Sets if this field has a list of allowed values, default 'no'. Surround each option with an <option> tag.
- prefixes
  - Sets the name of a field that this field should be used as a prefix for. That field must be defined. An example of where this would be useful is for defining AMR fields, where one field is a modifier (>,<,=) for a MIC value field. A field with this attribute defined will not be shown as a separate field within the isolate record, but will be displayed as a prefix to the value of the set field. The prefix field will also not be labelled in the curation interface isolate add/update form, but will appear immediately before and inline with the prefixed field.
- query
  - Set to 'no' to exclude field from query drop-down lists.
- regex
  - Regular expression used to constrain field values, e.g. regex="^[A-Z].*$" forces the first letter of the value to be capitalized.
- required
  - Sets if data is required for this field. Allowed values are 'yes', 'no', 'expected', 'genome_required' or 'genome_expected'; default 'yes'. If set to 'expected', the value cannot be left empty when batch adding an isolate record or using the submission system, but a null value can be explicitly set using the value 'null'. The use of this is to encourage submitters to include a value for this field if it is available, while still allowing empty values if it is not. Setting the value to 'genome_required' or 'genome_expected' only affect genome submissions.
- separator
  - Optional string to place between field prefix value and field value if the prefixes attribute is defined.
- suffix

  - Optional string that is displayed after value in isolate information page and curation interface. Useful for adding units for numerical values.

- userfield
  - Select if you want this field to have its own dropdown filter box of users (populated from the users table): 'yes' or 'no', default 'no'.
- web
  - URL that will be used to hyperlink field values. If [?] is included in the URL, this will be substituted for the actual field value.

Special values

The following special variables can be used in place of an actual value:

- CURRENT_DATE: current date in yyyy-mm-dd format
- CURRENT_YEAR: the 4 digit value of the current year

4.3.2 Sequence definition database XML attributes

Required attributes are in **bold**.

<db>

Top level element. Contains child element: system.

<system>

Any value set here can be overridden in a *system.overrides file*.

- **authentication**
  - Method of authentication: either 'builtin' or 'apache'. See *user authentication*.
- **db**
  - Name of database on system.
- **dbtype**
  - Type of database: either 'isolates' or 'sequences'.
- **description**
  - Description of database used throughout interface.
- align_limit
  - Overrides the sequence export record alignment limit in the Sequence Export plugin. Default: '200'.
- allele_comments
  - Enable comments on allele sequences: either 'yes' or 'no', default 'no'.
  - This is not enabled by default to discourage the practice of adding isolate information to allele definitions (this sort of information belongs in an isolate database).
- allele_flags
  - Enable flags to be set for alleles: either 'yes' or 'no', default 'no'.

- alternative_codon_tables
  - Enable alternative codon tables: either 'yes' or 'no', default is 'no'. Set to 'yes' to allow different codon tables to be selected when viewing translated sequences or filtering by CDS when uploading new allele definitions.
- BURST
  - Enable BURST plugin: either 'yes' or 'no'. If no value is set then the plugin will not be available unless the all_plugins attribute is set to 'yes'. If the all_plugins attribute is set to 'yes', the BURST plugin can be disabled by setting this attribute to 'no'.
- codon_table
  - Set the id of the global codon table to use. See https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi for a list with the of ids and their description.
- curate_config
  - The database configuration that should be used for curation if different from the current configuration. This is used when the submission system is being used so that curation links in the 'Manage submissions' pages for curators load the correct database configuration.
- curate_path_includes
  - Partial path of the bigscurate.pl script used to curate the database. See *user authentication*.
- curate_script
  - Relative web path to curation script. Default 'bigscurate.pl' (version 1.11+).
  - This is only needed if automated submissions are enabled. If bigscurate.pl is in a different directory from bigsdb.pl, you need to include the whole web path, e.g. /cgi-bin/private/bigsdb/bigscurate.pl.
- curators_only
  - Set to 'yes' to prevent ordinary authenticated users having access to database configuration. This is only effective if read_access is set to 'authenticated_users'. This may be useful if you have different configurations for curation and querying with some data hidden in the configuration used by standard users. Default 'no'.
- daily_pending_submissions
  - Overrides the daily limit on pending submissions that a user can submit via the web submission system. Default: '15'.
- daily_rest_submissions_limit
  - Overrides the limit on number of submissions that can be made to the database via the RESTful interface. This is useful to prevent flooding of the submission system by aberrant scripts. Default: '100'.
- delete_retire_only
  - Set to 'yes' to retire the id of any allele or profile that is deleted. This prevents re-use of ids. This setting will override the global setting in bigsdb.conf.
- diploid
  - Allow IUPAC 2-nuclotide ambiguity codes in allele definitions for use with diploid typing schemes: either 'yes' or 'no', default 'no'.
- disable_seq_downloads
  - Prevent users or curators from downloading all alleles for a locus (admins always can). 'yes' or 'no', default 'no'.

- exemplars
  - Use exemplar sequences in the BLAST caches used for the sequence query pages. This is useful on larger databases as it speeds up the query significantly. *Exemplar alleles MUST* be defined otherwise sequence queries will fail. 'yes' or 'no', default 'no'.
- formatted_description
  - Markdown formatted description of database. If set, this will be used throughout the HTML interface wherever formatting can be applied (main body of text) and overrides the value set in 'db_description'. Currently only supports **italics** and ****bold****.
- genome_submissions
  - Enable link to genome submissions (automated submission system): either 'yes' or 'no', default 'yes'.
  - To enable, you will also need to set isolate_submissions="yes".
- isolate_database
  - The config name of the isolate database. This is used to provide a link to isolate submissions. You also need to set isolate_submissions="yes".
- isolate_submissions
  - Set to yes to provide a link to isolate submissions. The isolate_database attribute also needs to be set. Default: 'no'.
- job_priority
  - Integer with default job priority for offline jobs (default:5).
- job_quota
  - Integer with number of offline jobs that can be queued or currently running for this database.
- kiosk
  - Set to a page name to restrict configuration to always start on this page, rather than an index page. This faciliates running in a cut-down *kiosk mode* that doesn't allow access to all features. *Currently only 'sequenceQuery' is supported.*
- kiosk_allowed_pages
  - Comma-separated list of pages that the configuration is allowed to show, apart from the page set in the 'kiosk' attribute. Example for a sequence query configuration would be 'sequenceTranslate' to allow access to the translated sequence page following a query.
- kiosk_help
  - URL to context-sensitive help page.
- kiosk_locus
  - Restrict sequence query to a specific locus or scheme. Use either the locus primary name or 'SCHEME_X' where X is the scheme number.
- kiosk_no_genbank
  - Set to "yes" to hide the Genbank accesssion form element in kiosk mode.
- kiosk_no_upload
  - Set to "yes" to hide the sequence file upload in kiosk mode.
- kiosk_simple
  - Remove most explanatory text from kiosk page.

- kiosk_text
  - Alternative text to show on kiosk page.
- kiosk_title
  - Title text to use when running in kiosk mode.
- profile_submissions
  - Enable profile submissions (automated submission system): either 'yes' or 'no', default 'no' (version 1.11+).
  - To enable, you will also need to set submissions="yes". By default, profile submissions are disabled since generally new profiles should be accompanied by representative isolate data, and the profile can be extracted from that.
- public_login
  - Optionally allow users to log in to a public database - this is useful as any jobs will be associated with the user and their preferences will also be linked to the account. Set to 'no' to disable. Default 'yes'.
- query_script
  - Relative web path to bigsdb script. Default 'bigsdb.pl' (version 1.11+).
  - This is only needed if automated submissions are enabled. If bigsdb.pl is in a different directory from bigscurate.pl, you need to include the whole web path, e.g. /cgi-bin/bigsdb/bigsdb.pl.
- read_access
  - Describes who can view data: either 'public' for everybody, or 'authenticated_users' for anybody who has been able to log in. Default 'public'.
- related_databases
  - Semi-colon separated list of links to related BIGSdb databases on the system. This should be in the form of database configuration name followed by a '|' and the description, e.g. 'pubmlst_neisseria_isolates|Isolates'. This is used to populate the menu items.
- script_path_includes
  - Partial path of the bigsdb.pl script used to access the database. See *user authentication*.
- separate_dataset
  - Treat database configuration as though it was a separate database for the purposes of handling submissions and curators: either 'yes' or 'no', default 'no'. Submissions will be tagged with the configuration name and will only be visible within that configuration. Curators can be limited to specific configurations by populating the curator_configs table. This also affects whether they are notified of submissions.
- seq_export_limit
  - Overrides the sequence export limit (records x loci) in the Sequence Export plugin. Default: '1000000'.
- sets
  - Use *sets*: either 'yes' or 'no', default 'no'.
- set_id
  - Force the use of a specific set when accessing database via this XML configuration: Value is the name of the set.
- start_codons

  - Semi-colon separated list of start codons to allow. Note that this list will replace the built-in defaults of ATG, GTG, and TTG, and is used for all functions that require recognising complete coding sequences.

- submissions
  - Enable automated submission system: either 'yes' or 'no', default 'no' (version 1.11+).
  - The curate_script and query_script paths should also be set, either in the bigsdb.conf file (for site-wide configuration) or within the system attribute of config.xml.
- submissions_deleted_days
  - Overrides the default number of days before closed submissions are deleted from the system. Default: '90'.
- total_pending_submissions
  - Overrides the total limit on pending submissions that a user can submit via the web submission system. Default: '20'.
- user_job_quota
  - Integer with number of offline jobs that can be queued or currently running for this database by any specific user - this parameter is only effective if users have to log in.
- webroot
  - URL of web root, which can be relative or absolute. This is used to provide a hyperlinked item in the dropdown menu. Default '/'.
- webroot_label
  - Label text for the breadcrumb link defined by the webroot value. This can be formatted using Markdown. Currently only supports **italics** and ****bold****.

4.4 Over-riding global defaults set in bigsdb.conf

Certain values set in bigsdb.conf can be over-ridden by corresponding values set in a database-specific config.xml file. These can be set within the system tag like other attributes:

- query_script
  - Relative web path to bigsdb script.
- curate_script
  - Relative web path to curation script.
- prefs_db
  - The name of the preferences database.
- auth_db
  - The name of the authentication database.
- tmp_dir
  - Path to the web-accessible temporary directory.
- secure_tmp_dir
  - Path to the web-inaccessible (secure) temporary directory.
- ref_db
  - The name of the references database.

4.5 Over-riding values set in config.xml

Any attribute used in the system tag of the database config.xml file can be over-ridden using a file called **system.overrides**, placed in the same directory as config.xml. This is very useful as it allows you to set up multiple configs for a database, with the config.xml files symlinked so that any changes to one will be seen in each database configuration. An example of why you may wish to do this would be if you create separate public and private views of the isolate table that filters on some attribute. The system.overrides file uses key value pairs separated by = with the values quoted, e.g.

```
view="private"
read_access="authenticated_users"
description="Private view of database"
```

It is also possible to override the allow_submissions, required, maindisplay, default, hide, or curate_only attributes of a particular field using a file called **field.overrides**. The field.overrides file uses the format 'field:attribute="value"' on each line, e.g.

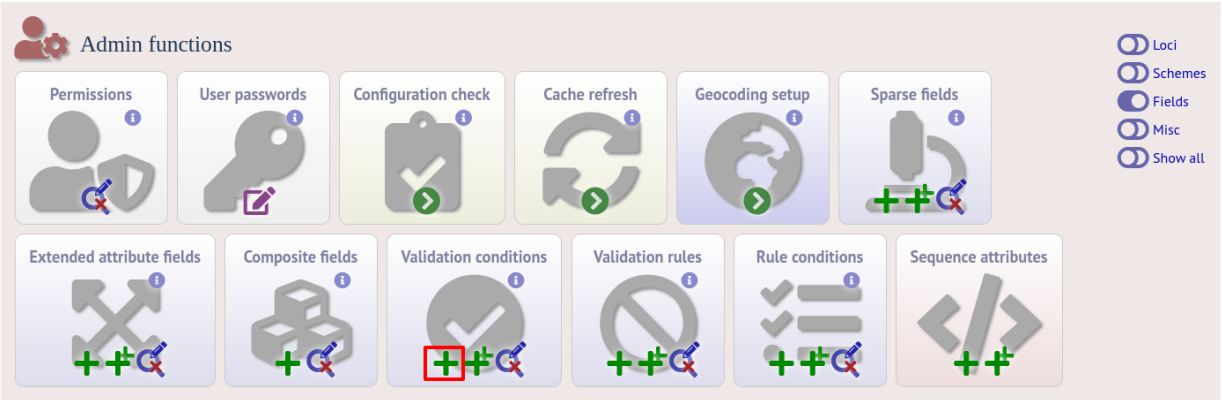
```
date_received:required="yes"
```

4.6 Setting field validation rules

Sometimes it may be necessary to restrict the allowed values in one isolate field depending on the values submitted for another field. It is possible to do this using field validation rules. These combine one or more conditions which all have to match for validation to fail and an isolate record upload to be rejected.

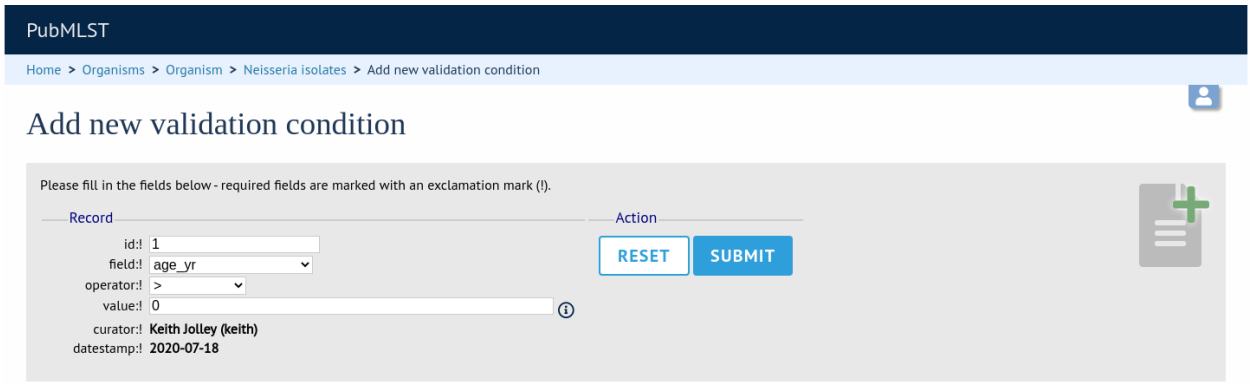
An example of this may be if you have an age_year and an age_month field but you only want age_month to be populated if the subject is less than one year old. You can do this as follows.

As an admin, on the curator interface, click the 'Field' toggle to show the validation table links. Then click 'Add' on the 'Validation conditions' setting:

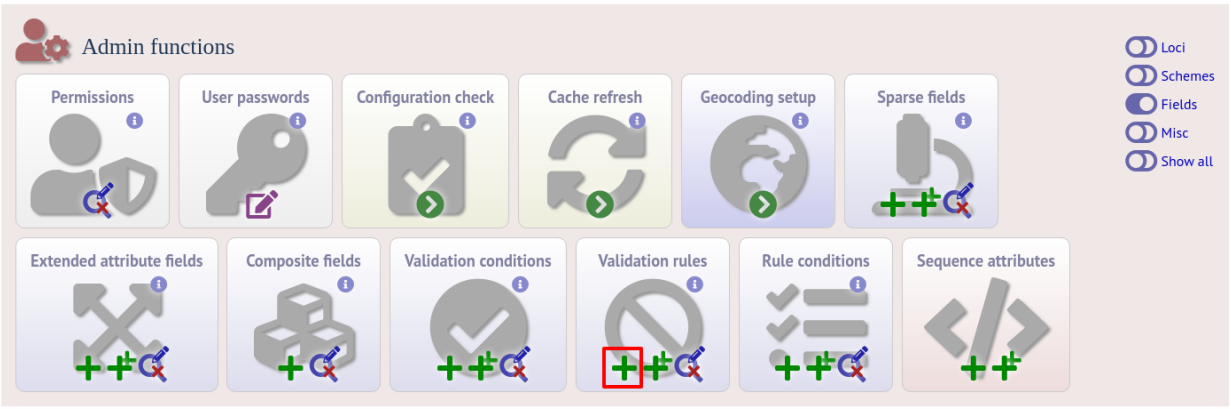


Add the following conditions separately:

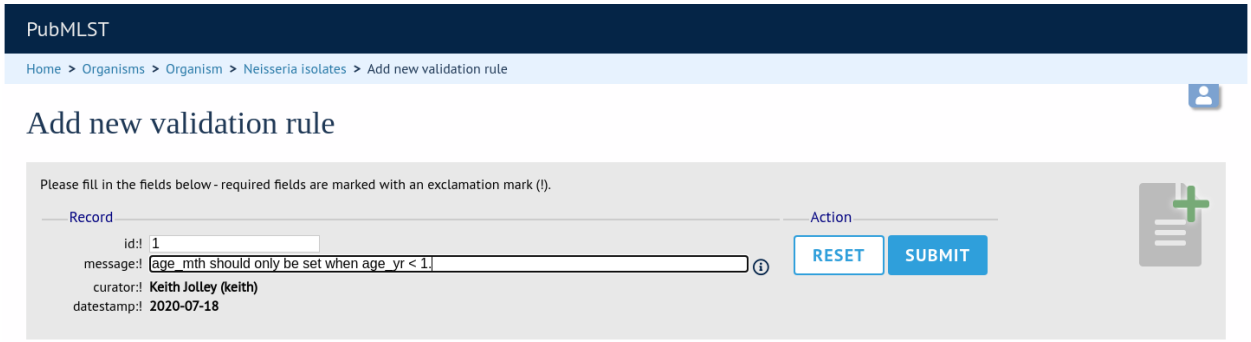
- age_year > 0
- age_month NOT null



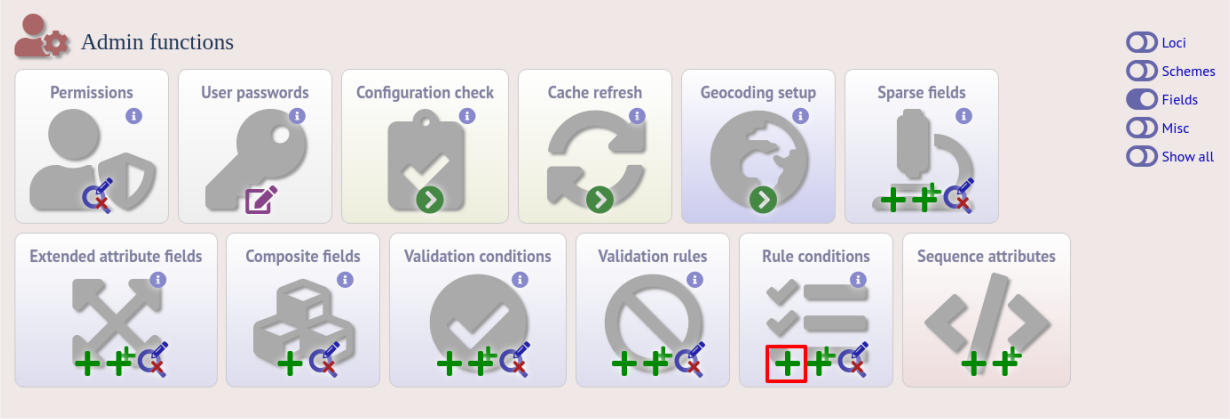
Now add a new 'Validation rule', by clicking 'Add' on the 'Validation rules' setting:



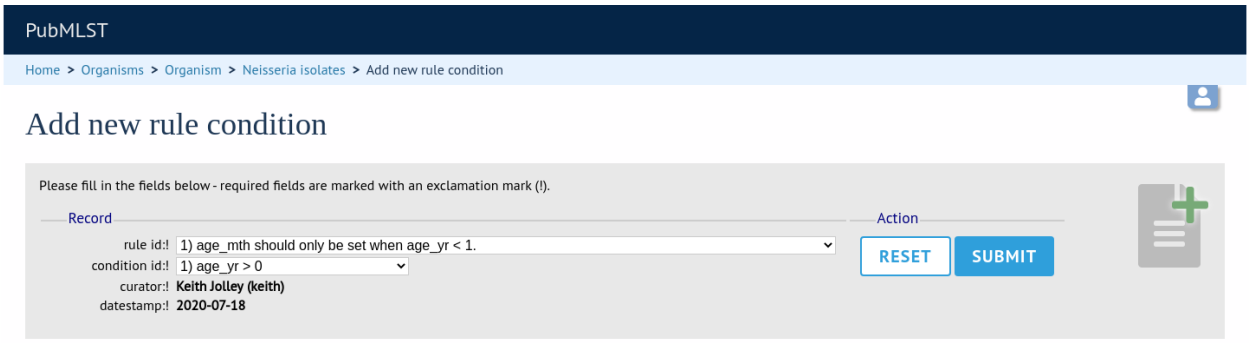
Here you just enter the message that will be returned when the validation fails:



Finally add the conditions to the rule by clicking 'Add' on the 'Rule conditions' setting:



Select the rule message and the condition from the dropdown boxes:



Make sure you do this for each of the conditions that have to match.

Validation checks are performed when adding or updating an isolate record, or when a user submits via the automated submission interface. Currently these checks are not enforced when doing a batch update.

4.6.1 Special condition values

Use the value **null** to indicate that the field is empty, e.g.

- age_month NOT null

Use a field name in square brackets to compare the value in that field, e.g. suppose you have two date fields, 'date_sampled' and 'date_received', and you want to ensure that 'date_received' is not before 'date_sampled'. You can do this with the following condition:

- date_received < [date_sampled]

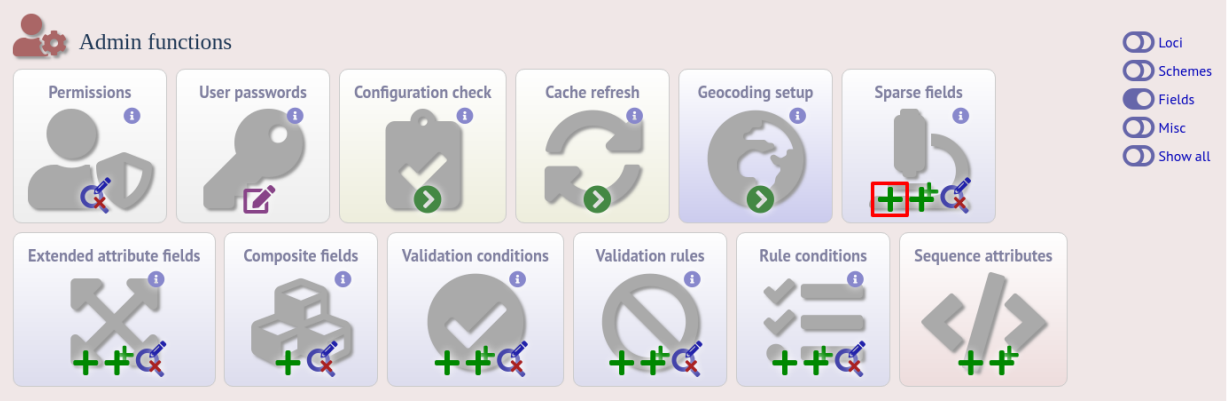
The two fields have to be of the same data type in order to be compared (you cannot compare a text field to an integer field for example).

4.7 Sparsely-populated fields

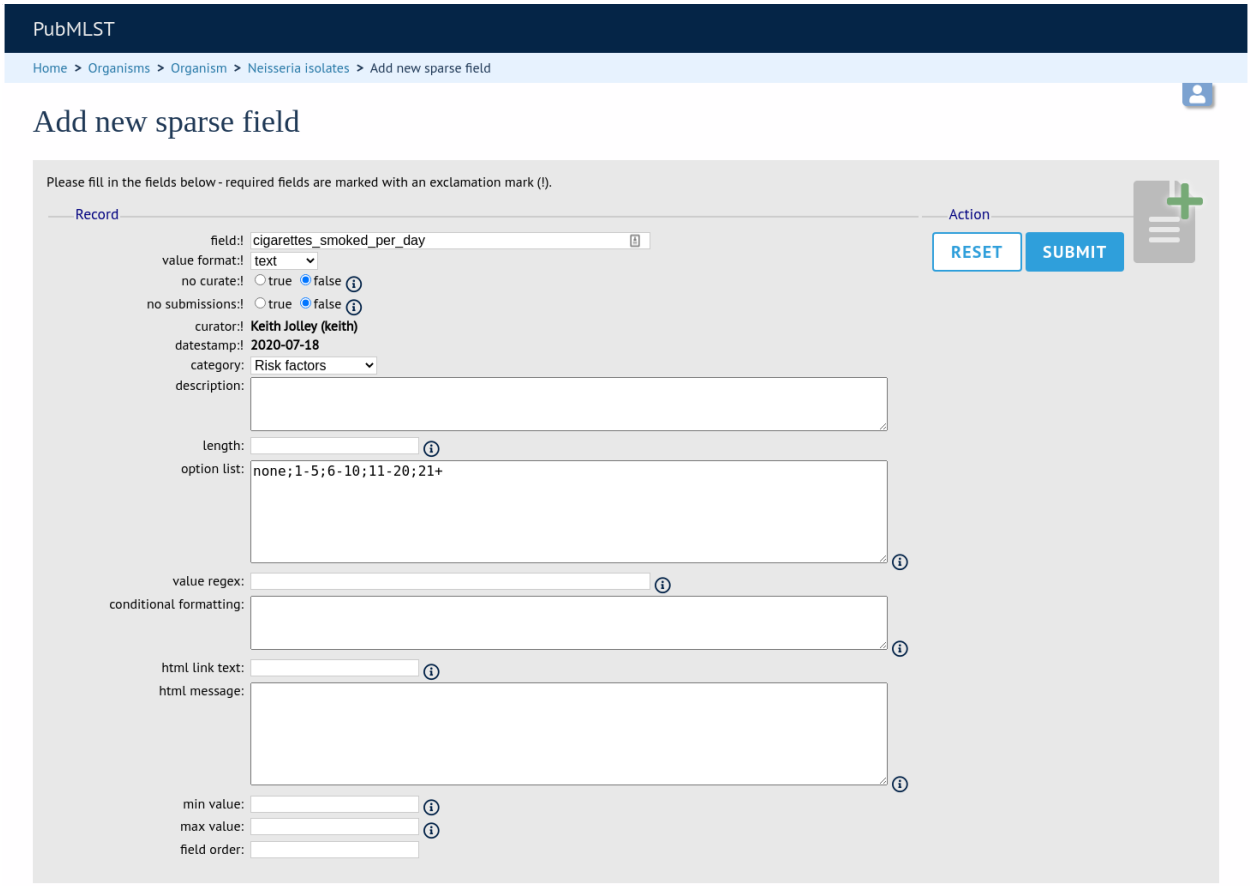
Commonly used isolate fields should be described in the config.xml file and included as columns within the isolates table. Sometimes, however, you may have a need to record information that is only likely to be found in a minority of records. This can be done more efficiently with the use of sparsely-populated fields. These are stored differently in the database (using an entity-attribute-value [EAV] model) but can still be searched and exported in a similar way to normal fields. There is no limit to the number of such fields that can be defined.

The default name for these fields is 'secondary metadata' and this is how they will be grouped in the interface. You can change this by setting the 'eav_fields' attribute in the *system tag of config.xml*. It is also possible to group these fields in to categories - these can be defined with a comma-separated list in the 'eav_groups' attribute in the *system tag of config.xml*.

You will need to be an admin to define sparely-populated fields. Make sure that the 'Fields' toggle is selected on the curators' page. Click the add (+) button on the 'Sparse fields' function.



Fill in the form and click 'Submit'.



Field options are:

- **field**
  - name of field
- **value_format**
  - date type - either integer, float, text, date or boolean.
- **no_curate**
  - Set to true to prevent user updates of fieldThis setting could be used if the value is calculated by an external script rather than entered by a curator.
- **no_submissions**
  - Set to true to prevent the field being listed in the submissions template.
- description
  - Tooltip text that will appear on curator forms.
- length
  - Restrict allowed length of value.
- option_list
  - Semi-colon separated list of allowed values.
- value_regex

  - Regular expression that can constrain allowed values.

- conditional_formatting
  - Semi-colon separated list of values - each consisting of the value, followed by a pipe character (|) and HTML to display instead of the value. If you need to include a semi-colon within the HTML, use two semi-colons (;;) otherwise it will be treated as the list separator.'
- html_link_text
  - This defines the text that will appear on an information link that will trigger a slide-in message (if defined int the next field). Default is 'info'.
- html_message
  - This message will slide-in on the isolate information page when the field value is populated and the information link is clicked. Full HTML formatting is supported.
- min_value
  - Valid for number fields only.
- max_value
  - Valid for number fields only.
- field_order
  - Integer indicating the order that fields should be displayed. If this is not set they will appear alphabetically.

4.8 Kiosk mode

Kiosk mode allows you to run a cut-down interface that offers a single main functionality. Currently, only a sequence query page is supported. The interface is locked down so that only specified functionality is supported and data cannot be exported.

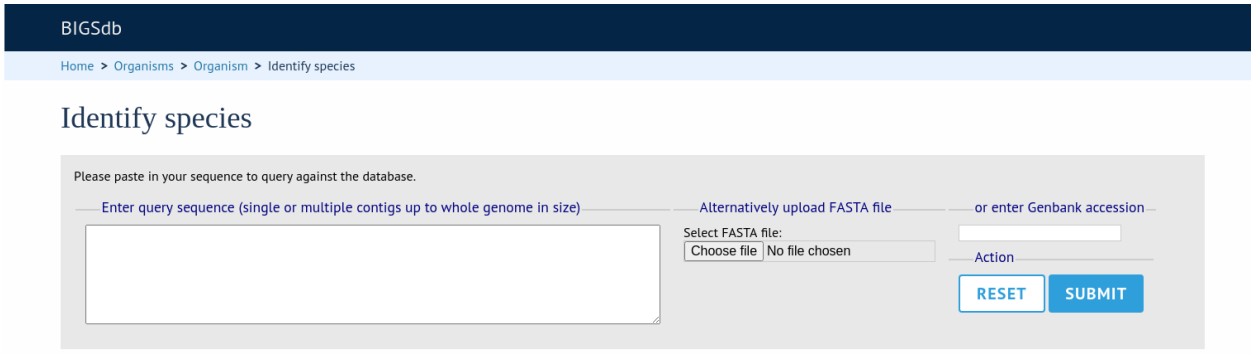
See the *kiosk_* attributes* in config.xml.

As an example, the following settings are used for the rMLST 'Identify species' tool at https://pubmlst.org/rmlst/. The database usually requires a user to log in, but this tool offers a restricted functionality without logging in.

```
kiosk="sequenceQuery"
kiosk_allowed_pages="sequenceTranslate"
kiosk_title="Identify species"
kiosk_locus="SCHEME_1"
kiosk_simple="yes"
kiosk_no_upload="no"
kiosk_no_genbank="no"
rest_kiosk="sequenceQuery"
```

When you go to this example kiosk page you see only the sequence query page and trying to access any other functionality is prevented.

The rest_kiosk attribute enables queries to also be performed using the *RESTful API* which will be similarly locked down.



4.9 User authentication

You can choose whether to allow Apache to handle your authentication or use built-in authentication.

4.9.1 Apache authentication

Using apache to provide your authentication allows a flexible range of methods and back-ends (see the Apache authentication HowTo for a start, or any number of tutorials on the web).

At its simplest, use a .htaccess file in the directory containing the bigscurate.pl (and bigsdb.pl for restriction of read-access) script or by equivalent protection of the directory in the main Apache server configuration. It is important to note however that, by default, any BIGSdb database can be accessed by any instance of the BIGSdb script (including one which may not be protected by a .htaccess file, allowing public access). To ensure that only a particular instance (protected by a specific htaccess directive) can access the database, the following attributes can be set in the system tag of the database XML description file:

- script_path_includes: the BIGSdb script path must contain the value set.
- curate_path_includes: the BIGSdb curation script path must contain the value set.

For public databases, the 'script_path_includes' attribute need not be set.

To use apache authentication you need to set the authentication attribute in the system tag of the database XML configuration to 'apache'.

4.9.2 Built-in authentication

BIGSdb has its own built-in authentication, using a separate database to store password and session hashes. The advantages of using this over many forms of apache authentication are:

- Users are able to update their own passwords.
- Passwords are not transmitted over the Internet in plain text.

When a user logs in, the server provides a random one-time session variable and the user is prompted to enter their username and password. The password is encrypted within the browser using a Javscript one-way hash algorithm, and this is combined with the session variable and hashed again. This hash is passed to the server. The server compares this hash with its own calculated hash of the stored encrypted password and session variable that it originally sent to the browser. Implementation is based on perl-md5-login. Stored passwords are salted and hashed using bcrypt.

To use built-in authentication you need to set the authentication attribute in the system tag of the database XML configuration to 'builtin'.

4.10 Setting up the admin user

The first admin user needs to be manually added to the users table of the database. Connect to the database using psql and add the following (changing details to suit the user).:

```
INSERT INTO users (id, user_name, surname, first_name, email, affiliation, status, date_
↪entered,
datestamp, curator) VALUES (1, 'keith', 'Jolley', 'Keith', 'keith.jolley@biology.ox.ac.uk
↪',
'University of Oxford, UK', 'admin', 'now', 'now', 1);
```

If you are using built-in authentication, set the password for this user using the *add_user.pl* script. This hashes the password and stores this within the authentication database. Other users can be added by the admin user from the curation interface accessible from http://your_website/cgi-bin/private/bigscurate.pl?db=test_db (or wherever you have located your bigscurate.pl script).

4.11 Retrieving PubMed citations from NCBI

Publications listed in PubMed can be associated with individual isolate records, profiles, loci and sequences. Full citations for these are stored within a local reference database, enabling these to be displayed within isolate records and searching by publication and author. This local database is populated by a script that looks in BIGSdb databases for PubMed records not locally stored and then requests the full citation record from the PubMed database.

The script is called retrieve_pubmed_records.pl and can be found in the scripts/maintenance directory.

Simply run the script either as the 'postgres' user or an account that is allowed to connect as the postgres user.

This should be run periodically from a CRON job, e.g. every hour.

4.12 Configuring access to remote contigs

It is possible for isolate records to have contigs in an external BIGSdb database. These must be accessible via the *RESTful API*. The advantage of this is that it enables multiple isolate databases to use the same genome assemblies without having to duplicate the storage of those assemblies. If access to the external database requires authenticated access, OAuth settings can be set to enable contig retrieval.

To enable remote contigs, set the remote_contigs attribute in the *<system>* tag of config.xml, i.e.

```
remote_contigs = "yes"
```

4.12.1 Setting up authentication

A client key for the BIGSdb remote contig manager needs to be generated. This can be done using the *create_client_credentials.pl* script, e.g.

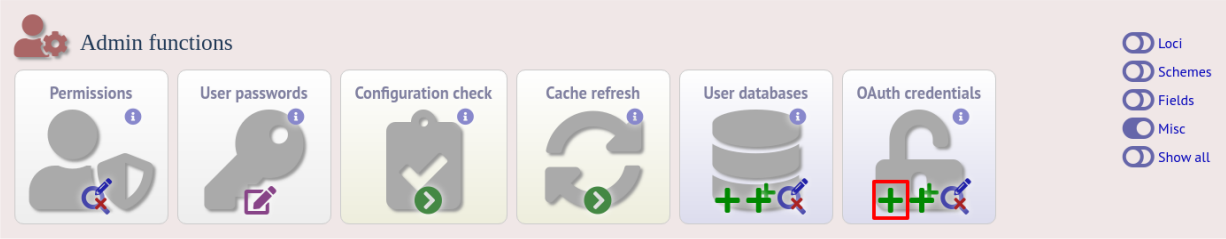
```
create_client_credentials.pl --a 'BIGSdb remote contig manager' --insert
```

This will generate a client id (key) and a client secret and add them to the authentication database.

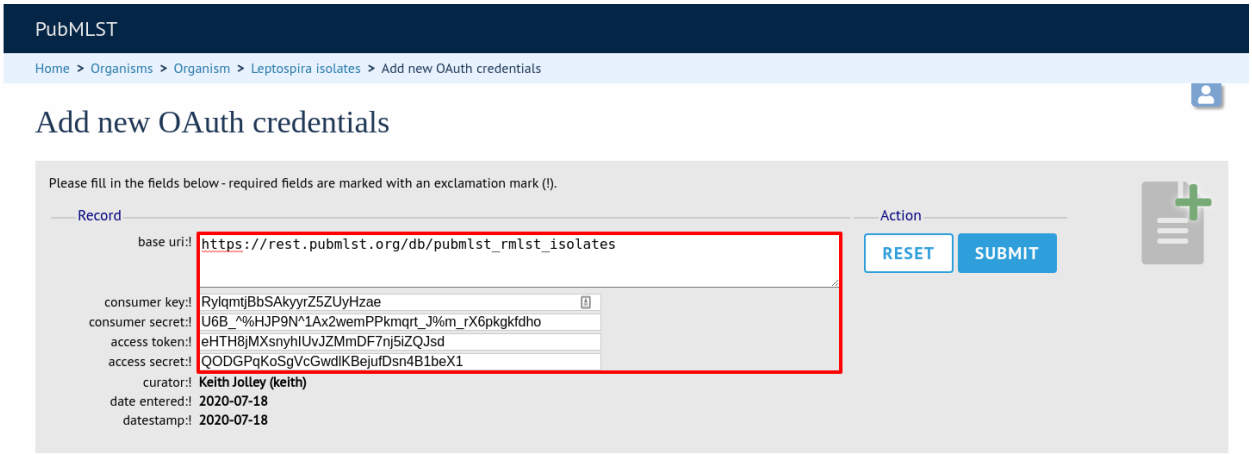
You will then need to obtain an access token and access secret using the client key and secret with the get_oauth_access_token.pl script. You will need to enter the API database URI (e.g. http://rest.pubmlst.org/db/pubmlst_rmlst_isolates) and the web database URL (e.g. https://pubmlst.org/bigsdb?db=pubmlst_rmlst_isolates). You

will then be prompted to follow a link and log in to the database with your user credentials. A verification code will be generated. You need to enter this in to the script when prompted. An access token and secret will be returned to you.

From the curators' page, click the oauth credentials add link in the administrator settings. This function is normally hidden, so you may need to click the 'Misc' toggle to display it.



Populate the OAuth_credentials table with the client key/secret and access token/secret. You should also enter the root REST URI for the database (e.g. http://rest.pubmlst.org/db/pubmlst_rmlst_isolates).



4.12.2 Processing remote contigs

When remote contigs are first linked to a record, the sequences are downloaded in bulk (without their metadata). This allows the sequence lengths to be recorded as this is needed for various queries and outputs. The curator is then given an option to process the contigs, which involves downloading each contig individually to record metadata including the original designation and the sequence platform used. This may take a while so it may be preferable to perform this task offline. This can be done using the process_remote_contigs.pl script found in the scripts/automation directory. Options for using this script are shown below:

```
remote_contigs.pl --help
NAME
    process_remote_contigs.pl
    Download, check length and create checksum contigs stored as URIs

SYNOPSIS
    process_remote_contigs.pl --database NAME [options]

OPTIONS

--database NAME
```

(continues on next page)

(continued from previous page)

```
      Database configuration name.

  --exclude_isolates LIST
      Comma-separated list of isolate ids to ignore.

  --exclude_projects LIST
      Comma-separated list of projects whose isolates will be excluded.

  --help
      This help page.

  --isolates LIST
      Comma-separated list of isolate ids to scan (ignored if -p used).

  --isolate_list_file FILE
      File containing list of isolate ids (ignored if -i or -p used).

  --min ID
      Minimum isolate id.

  --max ID
      Maximum isolate id.

  --projects LIST
      Comma-separated list of project isolates to scan.

  --quiet
      Only display errors.

.. _setup_dashboard:
```

4.13 Setting up front-end and query dashboards

Dashboards can be used as an alternative front-end to isolate databases. They can also be used to summarise the results of a query. In order to enable dashboards for a particular database, they have to be enabled either globally or specifically for the database configuration. If enabled, users will have the option to toggle between the dashboard and the standard index page.

To enable globally and use the front-end dashboard by default, set the following in bigsdb.conf:

```
enable_dashboard=1
default_dashboard_view=1
```

Each of these values can be overridden for a particular database by setting the same attribute in the database config.xml file, with either 'yes' or 'no', i.e. dashboards can be enabled globally but disabled for a particular database configuration, or disabled globally but enabled for a particular database configuration.

To enable query dashboards, set the following in bigsdb.conf:

```
query_dashboard=1
```

Again, this value can be overridden for a particular database by setting the attribute in the database config.xml file, as above. Note that 'enable_dashboard' also needs to be enabled.

4.13.1 Defining default dashboards

A default global front-end dashboard can be set up by placing a dashboard_primary.toml file in /etc/bigsdb. This can be overridden for individual database configurations by adding a TOML file (dashboard_primary.toml), in the same format, to the database configuration directory. Any field defined in the TOML file that does not appear within a particular database is ignored.

A default global query dashboard can be similarly set up with a file called dashboard_query.toml. As this dashboard appears above isolate results tables it is preferred that smaller elements are defined, usually with a height of 1.

An example of the format can be seen below.

```
#Configuration for default front-end dashboard for isolate databases. This
#defines the visual elements that will be included. If field-specific elements
#are defined and that field does not exist in a particular database then it
#will be ignored.

#The default configuration can be overridden for a particular database by
#including a dashboard.toml file, using the same format, in the database
#configuration directory.

#Width can be 1, 2, 3, or 4.
#Height can be 1, 2, or 3.

#Field names have prefixes indicating the field type:
#f_ are standard provenance/primary fields
#e_ are extended attributes with the main field and the attribute separated
#   by ||, e.g. e_country||continent.
#eav_ are sparse fields

elements = [
   {   #Isolate count.
      display           = 'record_count',
      name              = 'Isolate count',
      width             = 2,
      background_colour = '#79cafb',
      main_text_colour  = '#404040',
      watermark         = 'fas fa-bacteria',
      change_duration   = 'month',
      url_text          = 'Browse isolates',
      hide_mobile       = 0
   },
   {  #Genome count (will only display if there are genomes in the database).
      display           = 'record_count',
      name              = 'Genome count',
      genomes           = 1,
      width             = 2,
      background_colour = '#7ecc66',
      main_text_colour  = '#404040',
      watermark         = 'fas fa-dna',
```

(continues on next page)

(continued from previous page)

```
    change_duration   = 'month',
    url_text          = 'Browse genomes',
    post_data         = { genomes = 1 },
    hide_mobile       = 0
  },
  {
    display           = 'field',
    name              = 'Country',
    field             = 'f_country',
    breakdown_display = 'map',
    width             = 3,
    height            = 2,
    hide_mobile       = 1
  },
  {
    #Top 5 list of continents (Geocoding should be set up with default country
     #list linked to continent - see 'Geocoding setup' on admin curator page.
    display           = 'field',
    name              = 'Continent',
    field             = 'e_country||continent',
    breakdown_display = 'top',
    top_values        = 5,
    width             = 2,
    hide_mobile       = 1
  },
  {
    name              = 'Sequence size',
    display           = 'seqbin_size',
    genomes           = 1,
    hide_mobile       = 1,
    width             = 2,
    height            = 1
  },
  {   #Doughnut chart of species.
    display           = 'field',
    name              = 'Species',
    field             = 'f_species',
    breakdown_display = 'doughnut',
    height            = 2,
    width             = 2,
    hide_mobile       = 1
  },
  {   #Treemap of disease.
    display           = 'field',
    name              = 'Disease',
    field             = 'f_disease',
    breakdown_display = 'treemap',
    height            = 2,
    width             = 2,
    hide_mobile       = 1
  },
  {
```

(continues on next page)

(continued from previous page)

```
      #Bar chart of submission years.
      display           = 'field',
      name              = 'Year',
      field             = 'f_year',
      breakdown_display = 'bar',
      width             = 3,
      bar_colour_type   = 'continuous',
      chart_colour      = '#126716',
      hide_mobile       = 1
   },
   {
      #Cumulative chart of submissions by date.
      display           = 'field',
      name              = 'Date entered',
      field             = 'f_date_entered',
      width             = 2,
      breakdown_display = 'cumulative',
      hide_mobile       = 1
   }
]
```

Attributes

The allowed attributes are listed below.

- background_colour
  - RGB hex code for the background colour, e.g. '#79cafb'. This is used only for 'big number' fields, e.g. isolate count.
- bar_colour_type
  - categorical - use contrasting colours for bars.
  - continuous - use the same colour for all bars (set colour use 'chart_colour' attibute).
- breakdown_display - type of visualisation. Allowed values are:
  - bar
    - bar chart - particularly useful for continuous data such as year.
  - cumulative
    - cumulative line chart - used for date_entered or datestamp fields.
  - doughnut
    - doughut chart
  - gps_map
    - GPS map. This can only be used for geography_point fields or fields that are linked to a lookup table of GPS coordinates.
  - pie
    - pie chart
  - top

        * top values list. You can choose the number of values to display by also setting the top_values attributes to either 3, 5, or 10.
    - treemap
        * treemap chart
    - word
        * word cloud. This can only be used for fields that have a defined list of allowed values.
    - map
        * global map. This can only be used for 'country' fields with a defined list allowed values or 'continent' fields which are an extended attribute of country.
- change_duration
    - Show the rate of change, e.g. the number of new records in past month. Used for 'big number' fields, e.g. isolate count or specific value count. Allowed values are 'week', 'month', or 'year'.
- chart_colour
    - RGB hex code for bar or cumulative line charts, e.g. '#126716'.
- display
    - Element type. Allowed values are:
        * field - This is used for most elements.
        * record_count - Used for isolate count fields.
        * seqbin_size - Used to display a histogram of genome sizes.
- field
    - The name of the field to display. Different types of field have different prefixes as follows:
        * Primary isolate field - prefix with 'f_', e.g. 'f_country'.
        * *Secondary metadata fields* - prefix with 'eav_'.
        * *Extended attributes* - prefix with 'e_', followed by the primary field name, followed by '||' and then the extended attribute name, e.g. for continent linked to country you would use 'e_country||continent'.
        * Scheme fields - e.g. clonal complex - prefix with 's_' followed by the scheme id number, then '_', followed by the scheme field name, e.g. for a field called 'clonal_complex' defined for scheme 1, you would use 's_1_clonal_complex'.
- gauge_background_colour
    - RGB hex code, e.g. '#79cafb, for the background colour on a gauge chart.
- gauge_foreground_colour
    - RGB hex code, e.g. '#79cafb, for the foreground colour on a gauge chart.
- geography_view
    - Choice of view for GPS maps, either 'Aerial' or 'Map'. Note that Aerial views can only be used if you have a Bing Maps key set in bigsdb.conf.
- header_background_colour
    - RGB hex code, e.g. '#79cafb#, for the header background for a top values list.
- header_text_colour

  - RGB hex code, e.g. '#79cafb#, for the header text colour for a top values list.
- height
  - Height of element - either 1, 2, or 3.
- hide_mobile
  - Set to 1 to hide element on small mobile devices (width <= 480 pixels).
- main_text_colour
  - RGB hex code, e.g. '#79cafb#, for the colour of the text used in big number elements.
- marker_colour
  - RGB hex code, e.g. '#79cafb#, or HTML colour value, for the colour of the markers on GPS maps.
- marker_size
  - Size of marker on GPS maps. Allowed values are 0-9.
- max_contigs
  - Number of contigs above which a genome will be rejected in the submission interface.
- min_n50
  - N50 value below which a genome will be rejected in the submission interface.
- min_total_length
  - Minimum length in base pairs below which a genome will be rejected in the submission interface.
- name
  - The name used for the title of the element.
- palette
  - ColorBrewer palette used for map displays. Allowed values are:
    * blue
    * green
    * purple
    * orange
    * red
    * blue/green
    * blue/purple
    * green/blue
    * orange/red
    * purple/blue
    * purple/blue/green
    * purple/red
    * red/purple
    * yellow/green
    * yellow/green/blue

        * yellow/orange/brown
        * yellow/orange/red

- post_data
  - Used to pass data attributes for linked queries. Currently only 'genomes' is used to specify that isolates should be filtered to those with genome assemblies, e.g. '{genomes = 1}'.
- specific_value_display
  - Type of display to use for specific values. Allowed values are:
    * gauge - gauge chart
    * number - big number value
- specific_values
  - list of field values to include in count shown in gauge chart or big number display, e.g. '['Neisseria meningitidis']'.
- url_text
  - link text to display when hovering over link leading to data query. Only available for isolate count or specific value charts.
- visualisation_type
  - Either 'breakdown' (default) or 'specific values'. You need to then set the visualisation using either the breakdown_display or specific_value_display attribute.
- warn_max_contigs
  - Number of contigs above which a genome will have a warning message shown in the submission interface.
- warn_min_n50
  - N50 value below which a genome will have a warning message shown in the submission interface.
- warn_min_total_length
  - Minimum length in base pairs below which a genome will have a warning shown in the submission interface.
- watermark
  - FontAwesome icon class used for background watermark on big number charts, e.g. 'fas fa-bacteria'. See https://fontawesome.com/icons?m=free.
- width
  - Width of element - either 1, 2, 3, or 4.

4.13.2 Defining default colours

Sometimes you may wish to maintain consistent colours for specific field values. You can define colours for values by field using an additional configuration file called dashboard_colours.toml that can be placed either in /etc/bigsdb (for global use) or within a database configuration directory. The format is as follows:

```
'eav_Bexsero_reactivity' = {
    'exact match'       = '#2ca02c',
    'cross-reactive'    = '#ff7f0e',
    'none'              = '#d62728',
    'insufficient data' = '#888888',
```

(continues on next page)

(continued from previous page)

```
    'No value'          = '#aaaaaa'
}
'eav_Trumenba_reactivity' = {
    'exact match'       = '#2ca02c',
    'cross-reactive'    = '#ff7f0e',
    'none'              = '#d62728',
    'insufficient data' = '#888888',
    'No value'          = '#aaaaaa'
}
's_1_clonal_complex' = {
    'ST-11 complex'      = 'yellow',
    'ST-41/44 complex'   = 'green'
}
```

Field names are prefixed as follows:

- f_ Standard provenance fields, e.g. f_country
- e_ Extended attribute fields, e.g. e_country||continent (continent attribute linked to country)
- eav_ *Sparely-populated fields*, e.g. eav_Bexsero_reactivity
- s_ Scheme fields, e.g. s_1_clonal_complex (clonal complex field in scheme 1)

This works for pie, doughnut, bar, and pie charts. Note that if you define any values for a field then any value not defined will be shown as light grey in the visualisation.

CHAPTER

FIVE

ADMINISTRATOR'S GUIDE

Please note that links displayed within the curation interface will vary depending on database contents and the permissions of the curator.

5.1 Types of user

There are four types of user in BIGSdb:

- User - can view data but never modify it. Users should be created for every submitter of data so that records can be tracked, even if they do not actually use the database.
- Submitter (isolate databases only) - can add and modify their own isolate data and data submitted by anybody else that is in the same *user group* as them but not anyone elses. A limited range of *Individual permissions* can be set for each submitter, so their roles can be controlled. A submitter with no specific permissions set has no more power than a standard user.
- Curator - can modify data but does not have full control of the database. *Individual permissions* can be set for each curator, so their roles can be controlled. A curator with no specific permissions set has no more power than a standard user.
- Admin - has full control of the database, including setting permissions for curators and setting user passwords if built-in authentication is in use.

5.2 User groups

User groups allow submitter accounts to be grouped such that the submitter can edit isolates where the sender is either themselves or any member of a user group to which they belong.

5.3 Curator permissions

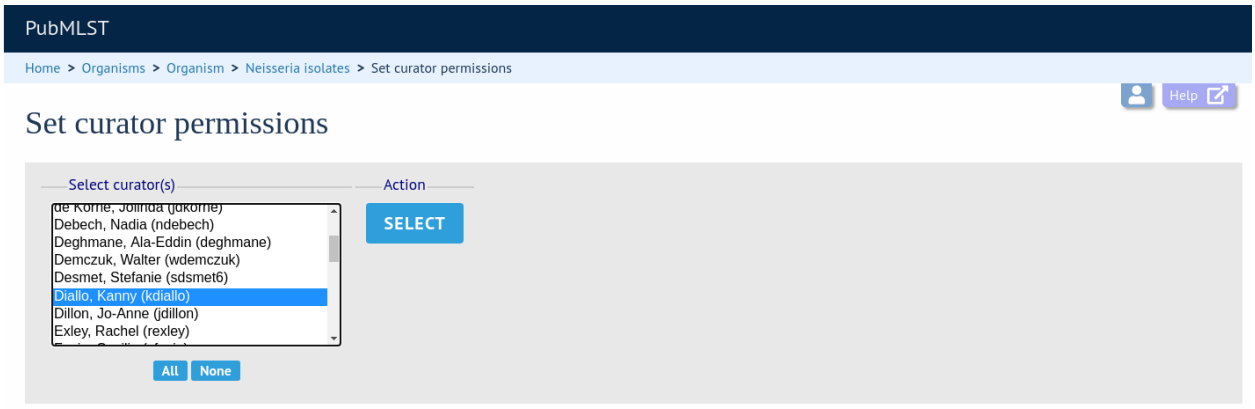
Individual permissions can be set for each curator:

- disable_access - if set to true, this user is completely barred from access.
- query_users - allowed to query and view users registered to the database. This is automatically allowed if permission is set to modify users.
- modify_users - allowed to add or modify user records. They can change the status of users, but can not revoke admin privileges from an account. They can also not raise the status of a user to admin level.
- modify_usergroups - allowed to add or modify user groups and add users to these groups.

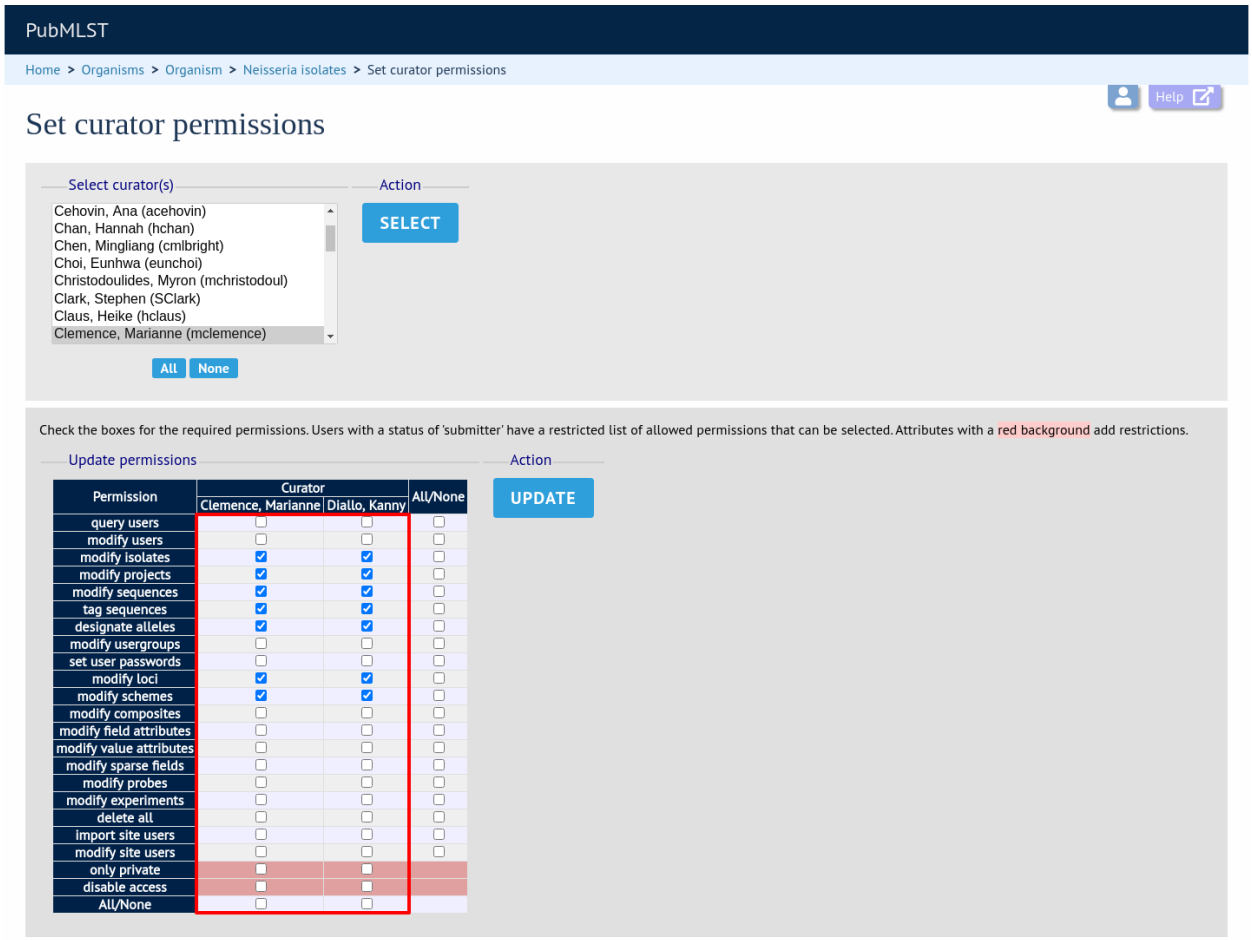
- set_user_passwords - allowed to modify other users' passwords (if built-in authentication is in use).
- modify_loci - allowed to add or modify loci.
- modify_locus_descriptions - allowed to modify the description text and external hyperlinks used. Even with this setting, only loci for which a user is explicitly set as a curator can be modified.
- modify_schemes - allowed to add or modify schemes.
- modify_sequences - allowed to add sequences to the sequence bin (for isolate databases) or new allele definitions (for sequence definition databases).
- modify_isolates - allowed to add or modify isolate records.
- modify_projects - allowed to create projects, modify their descriptions and add or remove isolate records to these.
- modify_composites - allowed to add or modify composite fields (fields made up of other fields, including scheme fields defined in external databases). Composite fields involve defining regular expressions that are evaluated by Perl - this can be dangerous so this permission should be granted with discretion.
- modify_field_attributes - allow user to create or modify secondary field attributes (lookup tables) for isolate record fields.
- modify_value_attributes - allow user to add or modify secondary field values for isolate record fields.
- modify_probes - allow user to define PCR or hybridization reactions to filter tag scanning.
- tag_sequences - allowed to tag sequences with locus information.
- designate_alleles - allowed to manually designate allele numbers for isolate records.
- modify_profiles - allowed to add or modify scheme profiles (only used in a sequence definitions database).
- import_site_users - allowed to import site users in to the database.
- modify_site_users - allowed to modify site user details (you may not want to this! The user account can be used by multiple databases on the site and any changes to user details will be seen throughout the site).
- modify_geopoints - allowed to modify the geography point GPS coordinate lookup table used for mapping.

Permissions can be set by clicking the Update/delete button in the Permissions box in the admin functions area of the curator's interface:

Choose one or more curators from the list (hold down Ctrl to select multiple values). click 'Select'.



Click the appropriate checkboxes to modify permissions. There are also 'All/None' buttons to facilitate quicker selection of options. Click 'Update'.

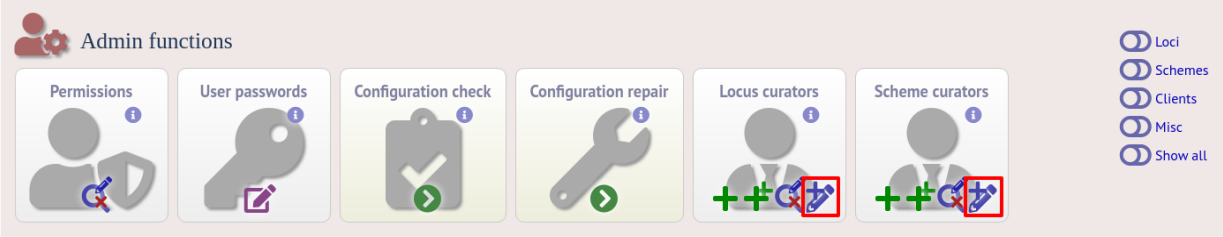


The 'disable access' option provides a quick way to disable access to a curator. This will not be selected by the 'All/None' buttons.

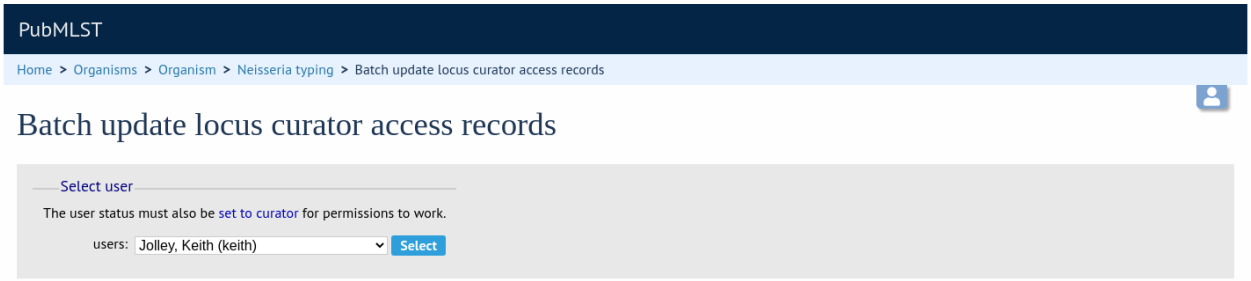
5.4 Locus and scheme permissions (sequence definition database)

To be allowed to define alleles or scheme profiles, curators must be granted specific permission for each locus and scheme by adding their user id number to the 'locus curator' and 'scheme curator' lists.

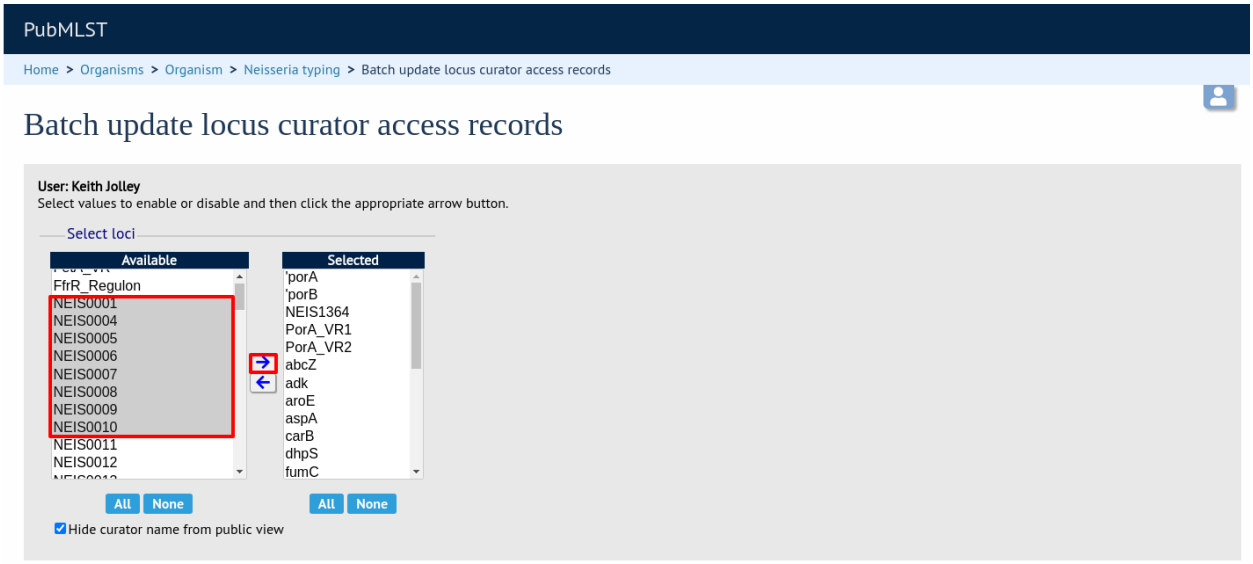
The easiest way to modify these lists is to use the batch update link next to 'locus curator control list' and 'scheme curator control list':



Select the curator from the list:



Then select loci/schemes that the user is allowed to curate in the left hand 'Available' list, and click the right button to move these to the 'Selected' list:



If you uncheck the 'Hide curator name from public view' checkbox, the curator name and E-mail address will appear alongside loci in the download table on the website.

5.5 Controlling access

5.5.1 Restricting particular configurations to specific user accounts

Suppose you only wanted specific users to access a database configuration.

In the config.xml, add the following directive:

```
default_access="deny"
```

This tells BIGSdb to deny access to anybody unless their account name appears within a file called users.allow within the config directory. The users.allow file should contain one username per line. You can also use a usergroups.allow file. This file should contain the names of user groups, the members of which are allowed access. The file should contain one user group name per line.

Alternatively, you can deny access to specific users, while allowing every other authenticated user. In config.xml, add the following directive:

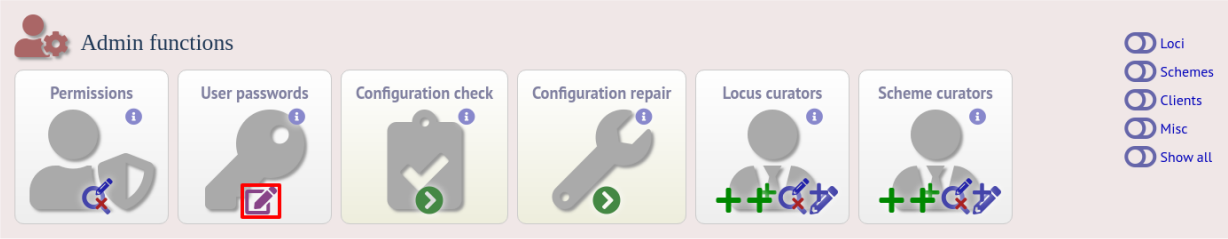
```
default_access="allow"
```

This tells BIGSdb to allow access to anybody unless their account name appears within a file called users.deny within the config directory. The users.deny file should contain one username per line.

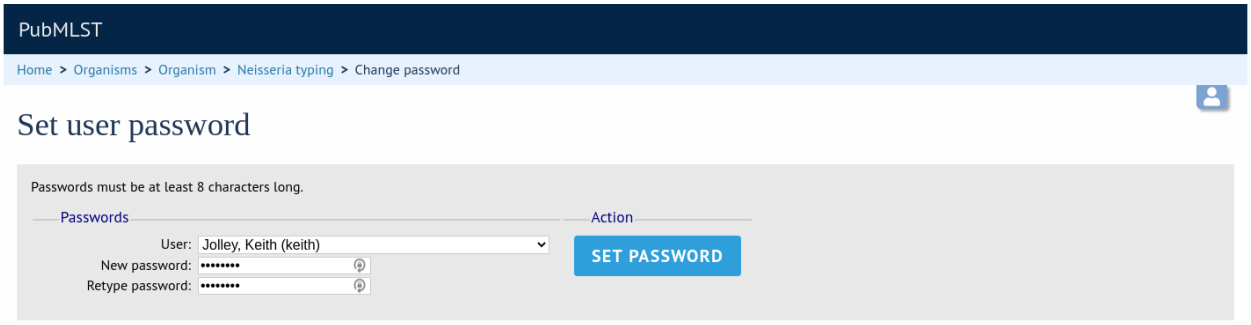
5.6 Setting user passwords

Please note that these instructions only apply if using the built-in BIGSdb authentication system.

If you are an administrator or a curator with specific permission to change other users' passwords, you should see a 'User passwords' box in the admin section of the curator's index page. Click the 'Set passwords' link.



Select the appropriate user from the drop-down list box and enter the new password twice where prompted.



Click 'Set password' and the password will be updated.

5.7 Setting the first user password

To set the first administrator's password for a new database, use the add_user.pl script found in the scripts/maintenance directory:

```
add_user.pl [-a] -d <dbase> -n <username> -p <password>
```

The first user account needs to be added to the database *manually*.

5.8 Enabling plugins

Some plugins can be enabled/disabled for specific databases. If you look in the get_attributes function of the specific plugin file and see a value for system_flag, this value can be used in the system tag of the database configuration XML file to enable the plugin.

For example, the get_attributes function of the BURST plugin looks like:

```
sub get_attributes {
       my %att = (
                name        => 'BURST',
                author      => 'Keith Jolley',
                affiliation => 'University of Oxford, UK',
                email       => 'keith.jolley@biology.ox.ac.uk',
                description => 'Perform BURST cluster analysis on query results query␣
↪results',
                category    => 'Cluster',
                buttontext  => 'BURST',
                menutext    => 'BURST',
                module      => 'BURST',
                version     => '1.0.0',
                dbtype      => 'isolates,sequences',
                section     => 'postquery',
                order       => 10,
                system_flag => 'BURST',
                input       => 'query',
                min         => 2,
                max         => 1000
       );
       return \%att;
}
```

The 'system_flag' attribute is set to 'BURST', so this plugin can be enabled for a database by adding:

```
BURST="yes"
```

to the system tag of the database XML file. If the system_flag value is not defined then the plugin is always enabled if it is installed on the system. If the system_flag value is set to 'no' then the plugin will be disabled even if the all_plugins attribute is set to 'yes'.

5.9 Temporarily disabling database updates

There may be instances where it is necessary to temporarily disable database updates. This may be during periods of server or database maintenance, for instance when running on a backup database server.

Updates can be disabled on a global or database-specific level.

5.9.1 Global

In the /etc/bigsdb/bigsdb.conf file, add the following line:

```
disable_updates=yes
```

An optional message can also be displayed by adding a disable_update_message value, e.g.

```
disable_update_message=The server is currently undergoing maintenance.
```

5.9.2 Database-specific

The same attributes described above for use in the bigsdb.conf file can also be used within the system tag of the database config.xml file, e.g.

```
<system
  db="bigsdb_neisseria"
  dbtype="isolates"
  ...
  disable_updates="yes"
  disable_update_message="The server is currently undergoing maintenance."
```

5.10 Host mapping

During periods of server maintenance, it may be necessary to map a database host to an alternative server. This would allow a backup database server to be used while the primary database server is unavailable. In this scenario, you would probably also want to *disable updates*.

Host mapping can be achieved by editing the /etc/bigsdb/host_mapping.conf file. Each host mapping is placed on a single line, with the current server followed by any amount of whitespace and then the new mapped host, e.g.

```
#Existing_host      Mapped_host
 server1           server2
 localhost         server2
```

[Lines beginning with a hash are comments and are ignored.]

This configuration would use server2 instead of server 1 or localhost wherever they are defined in the database configuration (either host attribute in the database config.xml file, or within the loci or schemes tables).

5.11 Improving performance

5.11.1 Use mod_perl

The single biggest improvement to speed can be obtained by running BIGSdb under mod_perl. There's very little point trying anything else until you have mod_perl set up and running - this can improve start-up performance a hundred-fold since the script isn't compiled on each page access but persists in memory.

5.11.2 Cache scheme definitions within an isolate database

If you have a large number of allelic profiles defined for a scheme, you can cache these definitions within an isolate database to speed up querying of isolates by scheme criteria (e.g. by ST for a MLST scheme).

To do this use the update_scheme_caches.pl script found in the scripts/maintenance directory, e.g. to cache all schemes in the pubmlst_bigsdb_neisseria_isolates database

```
update_scheme_caches.pl --database pubmlst_neisseria_isolates
```

This script creates indexed tables within the isolate database called temp_scheme_X and temp_isolates_scheme_fields_1 (where X is the scheme_id). If these table aren't present, they are created as temporary tables every time a query is performed that requires a join against scheme definition data. This requires importing all profile definitions from the definitions database and determining scheme field values for all isolates. This may sound like it would be slow but caching only has a noticeable effect once you have >5000 profiles.

You are able to update the cache for a single scheme, or a list of schemes, and choose the method of update. For large schemes, such as cgMLST, a full refresh may take a long time, so you may wish to only perform this infrequently (perhaps once a week) with more regular 'daily' or 'daily_replace' updates. A full list of options available are shown by typing

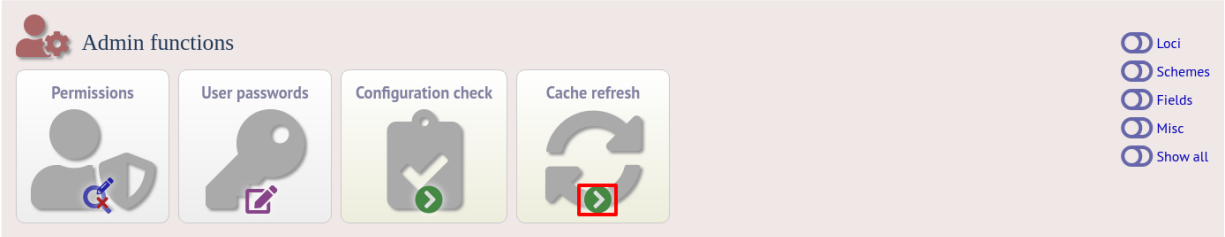
```
update_scheme_caches.pl --help

NAME
    update_scheme_caches.pl - Update scheme field caches

SYNOPSIS
    update_scheme_caches.pl --database NAME [options]

OPTIONS

--database NAME
    Database configuration name.

--help
    This help page.

--method METHOD
    Update method - the following values are allowed:
    full: Completely recreate caches
    incremental: Only add values for records not in cache.
    daily: Only add values for records not in cache updated today.
    daily_replace: Refresh values only for records updated today.

--quiet
```

(continues on next page)

(continued from previous page)

```
    Don't output progress messages.

--schemes SCHEMES
    Comma-separated list of scheme ids to use.
    If left empty, all schemes will be updated.
```

Note that you will need to run this script periodically as a CRON job to refresh the cache. Admins can also refresh the caches manually from a link on the curators' page. This link is only present if the caches have been previously generated.



You can also set cache_schemes="yes" in the system tag of config.xml to enable automatic refreshing of the caches (using the 'daily' method) when batch adding new isolates (you should still periodically run the update_scheme_caches.pl script via CRON to ensure any changes in the sequence definition database are picked up).

If queries are taking longer than 5 seconds to perform and a cache is not in place, you will see a warning message in bigsdb.log suggesting that the caches be set up. Unless you see this warning regularly, you probably don't need to do this.

5.11.3 Use a ramdisk for the secure temporary directory

If you are running BIGSdb on a large server with lots of RAM, you could use some of this as a ramdisk for temporary files. Debian/Ubuntu systems make available up to half the system RAM as a ramdisk mounted under /run/shm (or /dev/shm) by default. Set the secure_tmp_dir to this RAM disk and you should see significant improvement in operations requiring the writing of lots of temporary files, e.g. tag scanning and the Genome Comparator plugin. This is only likely to be appropriate if you have very large amounts of RAM available. As an example, the server hosting the PubMLST databases is a dedicated machine with 1TB RAM with temporary files rarely using more than 50GB space.

5.12 Dataset partitioning

5.12.1 Sets

Sets provide a means to partition the database in to manageable units that can appear as smaller databases to an end user. Sets can include constrained groups of isolates, loci, and schemes from the complete database.

See also:

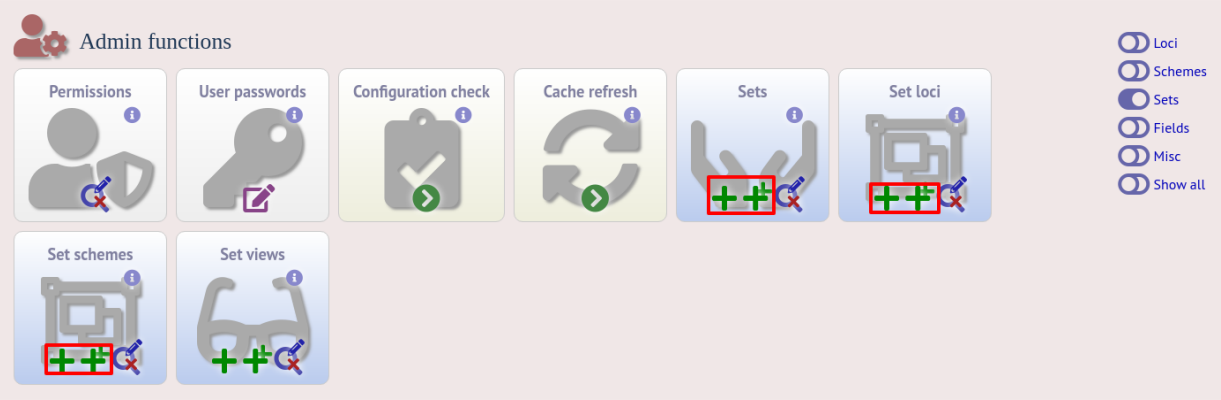
Sets (concept)

5.12.2 Configuration of sets

First sets need to be enabled in the XML configuration file (config.xml) of the database. Add the following attribute to the system tag:

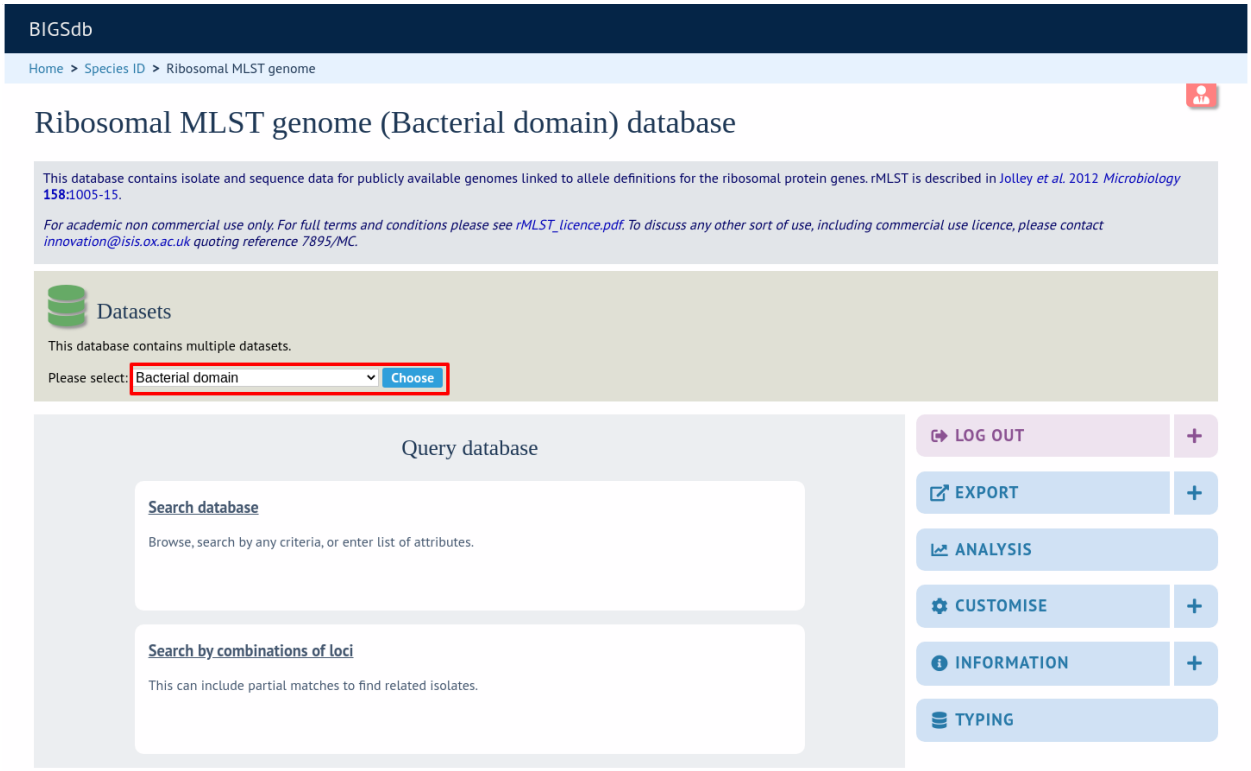
```
sets="yes"
```

With this attribute, the curation interface now has options to add sets, and then add loci or schemes to these sets. These functions are normally hidden, so you may need to click the 'Sets' toggle to display it.



The name of a locus or scheme to use within a set can be defined in the set_name field when adding loci or schemes to a set. Common names can also be set for loci - equivalent to the common name used within the loci table.

Now when a user goes to the contents page of the database they will be presented with a dropdown menu of datasets and can choose either the 'whole database' or a specific set. This selection is remembered between sessions.



Alternatively, a specific set can be selected within the XML config file so that only a specific set is available when

accessed via that configuration. In that case, the user would be unaware that the database contains anything other than the loci and schemes available within the set.

To specify this, add the following attributute to the system tag:

```
set_id="1"
```

where the value is the name of the set.

Note: If the set_id attribute is set, database configuration settings in the curator's interface are disabled. This is because when the configuration is constrained to a set, only loci and schemes already added to the set are visible, so functionality to edit schemes/loci would become very confusing. To modify these settings, you either need to access the interface from a different configuration, i.e. an alternative config.xml with the set_id attribute not set, or temporarily remove the set_id directive from the current config.xml while making configuration changes.

5.12.3 Set views

Finally the isolate record table can be partitoned using database views and these views associated with a set. Create views using something like the following:

```
CREATE VIEW spneumoniae AS SELECT * FROM isolates WHERE species = 'Streptococcus␣
↪pneumoniae';
GRANT SELECT ON spneumoniae TO apache;
```

Add the available views to the XML file as a comma separated list in the system tag 'views' attribute:

```
<system
 .....
 sets="yes"
 views="spneumoniae,saureus"
>
</system>
```

Set the view to the set by using the 'Add set view' link on the curator's page.

5.12.4 Using only defined sets

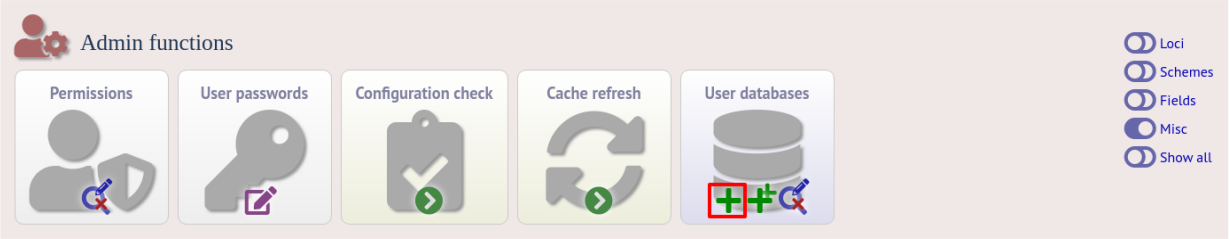
The only_sets attribute can be set to 'yes' to disable the option for 'Whole database' so that only sets can be viewed, e.g.

```
<system
 .....
 sets="yes"
 only_sets="yes"
>
</system>
```

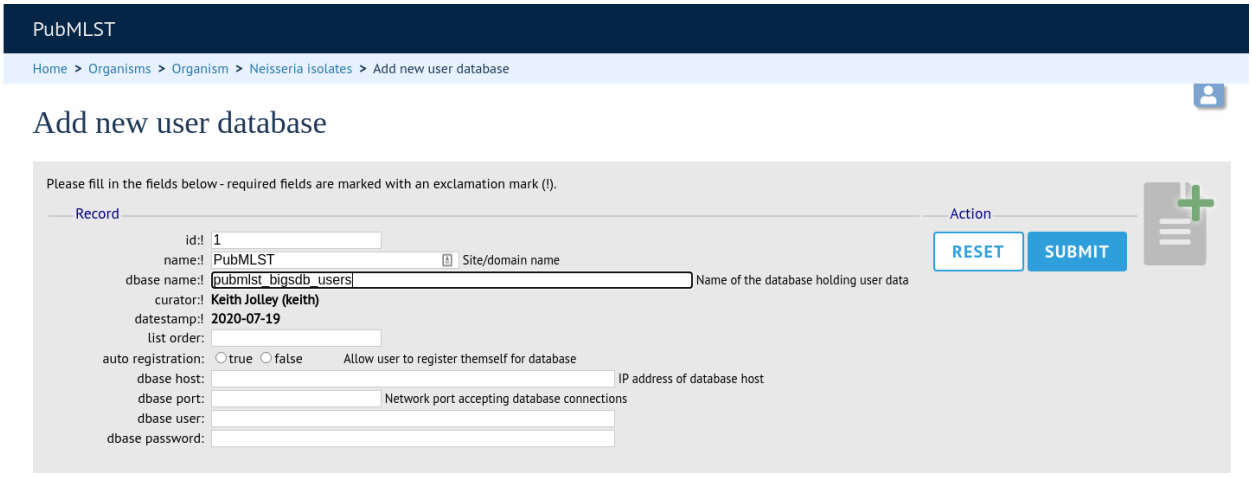
5.13 Setting a site-wide users database

On large sites you may wish to employ a site-wide users database so that user details are kept in a single location and the user can log in to any database using the same credentials.

Once a *site-wide user database has been set up*, this can be defined within each client database as follows. From the curators' contents page, click the add (+) user databases link. This function is normally hidden, so you may need to click the 'Misc' toggle to display it.



Enter the user database details. You only need to enter the full database connection details if these are different from those set in db.conf. Press submit.



Curators will need *specific permissions* set to be able to modify details in, or import users from a site-wide users database.

5.14 Adding new loci

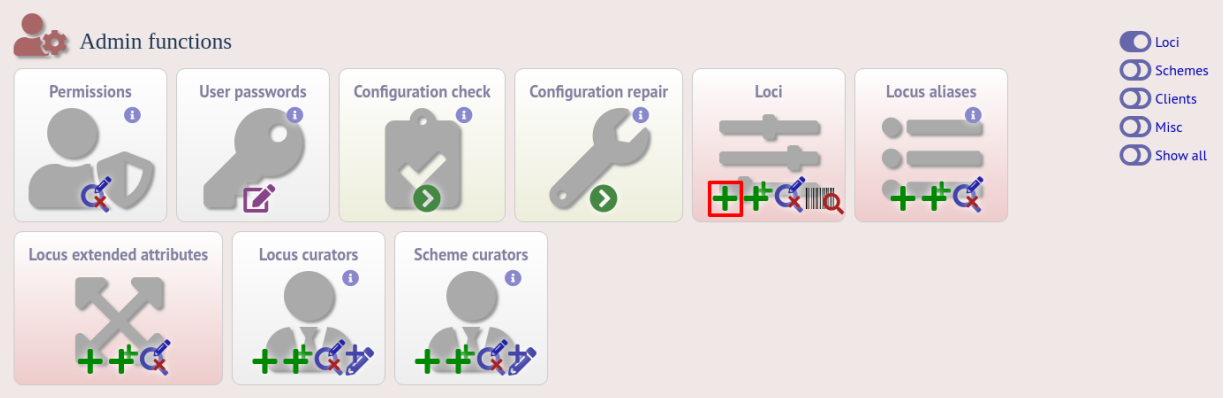
See also:

Loci (concept)

5.14.1 Sequence definition databases

Single locus

Click the add (+) loci link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:

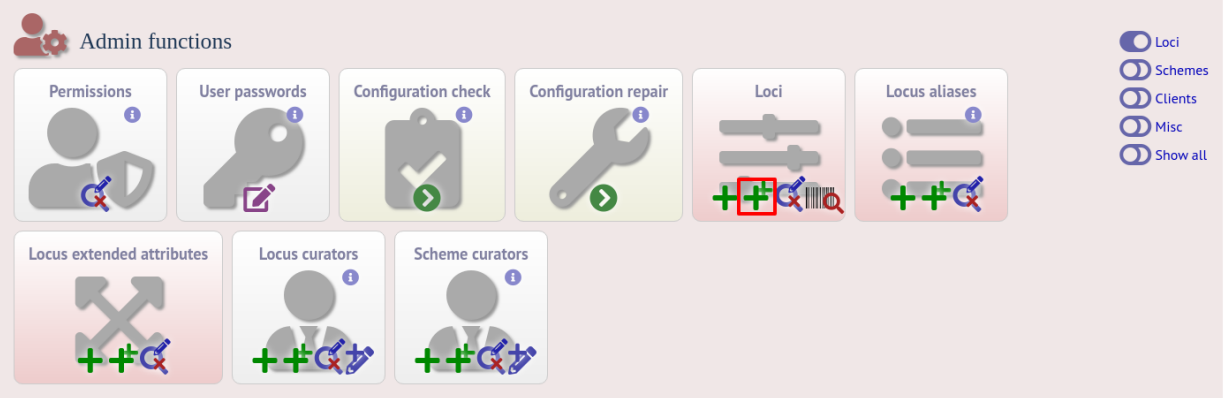
- id - The name of the locus.
  - Allowed: any value starting with a letter, number or underscore.
- data_type - Describes whether the locus is defined by nucleotide or peptide sequence.
  - Allowed: DNA/peptide.
- allele_id_format - The format for allele identifiers.
  - Allowed: integer/text.
- length_varies - Sets whether alleles can vary in length.
  - Allowed: true/false.
- coding_sequence - Sets whether the locus codes for a protein.
  - Allowed: true/false.
- formatted_name - Name with HTML formatting (optional).
  - This allows you to add formatting such as bold, italic, underline and superscripting to locus names as they appear in the web interface.
  - Allowed: valid HTML.
- common_name - The common name for the locus (optional).
  - Allowed: any value.
- formatted_common_name - Common name with HTML formatting (optional).

  - Allowed: valid HTML.
- allele_id_regex - Regular expression to enforce allele id naming (optional).
  - ^: the beginning of the string
  - $:the end of the string
  - d: digit
  - D: non-digit
  - s: white space character
  - S: non white space character
  - w: alpha-numeric plus '_'
  - .: any character
  - *: 0 or more of previous character
  - +: 1 or more of previous character
  - e.g. ^Fd-d+$ states that an allele name must begin with a F followed by a single digit, then a dash, then one or more digits, e.g. F1-12
- length - Standard length of locus (required if length_varies is set to false.
  - Allowed: any integer.
- min_length - Minimum length of locus (optional).
  - Allowed: any integer.
- max_length - Maximum length of locus (optional).
  - Allowed: any integer (larger than the minimum length).
- complete_cds - Whether locus represents a complete coding sequence (optional)
- start_codons - Semi-colon separated list of alternative start codons to allow
  - Note that these are in addition to the built-in defaults of ATG, GTG, TTG.
- orf - Open reading frame of locus (optional).
  - 1-3 are the forward reading frame, 4-6 are the reverse reading frames.
  - Allowed: 1-6.
- genome_position - The start position of the locus on a reference genome (optional).
  - Allowed: any integer.
- match_longest - Specifies whether in a sequence query to only return the longest match (optional).
  - This is useful for some loci that can have some sequences shorter than others, e.g. peptide loci defining antigenic loops. This can lead to instances of one sequence being longer than another but otherwise being identical. In these cases, usually the longer sequence is the one that should be matched.
  - Allowed: true/false.
- full_name - Full name of the locus (optional).
  - Allowed: any value.
- product - Name of gene product (optional).
  - Allowed: Any value.

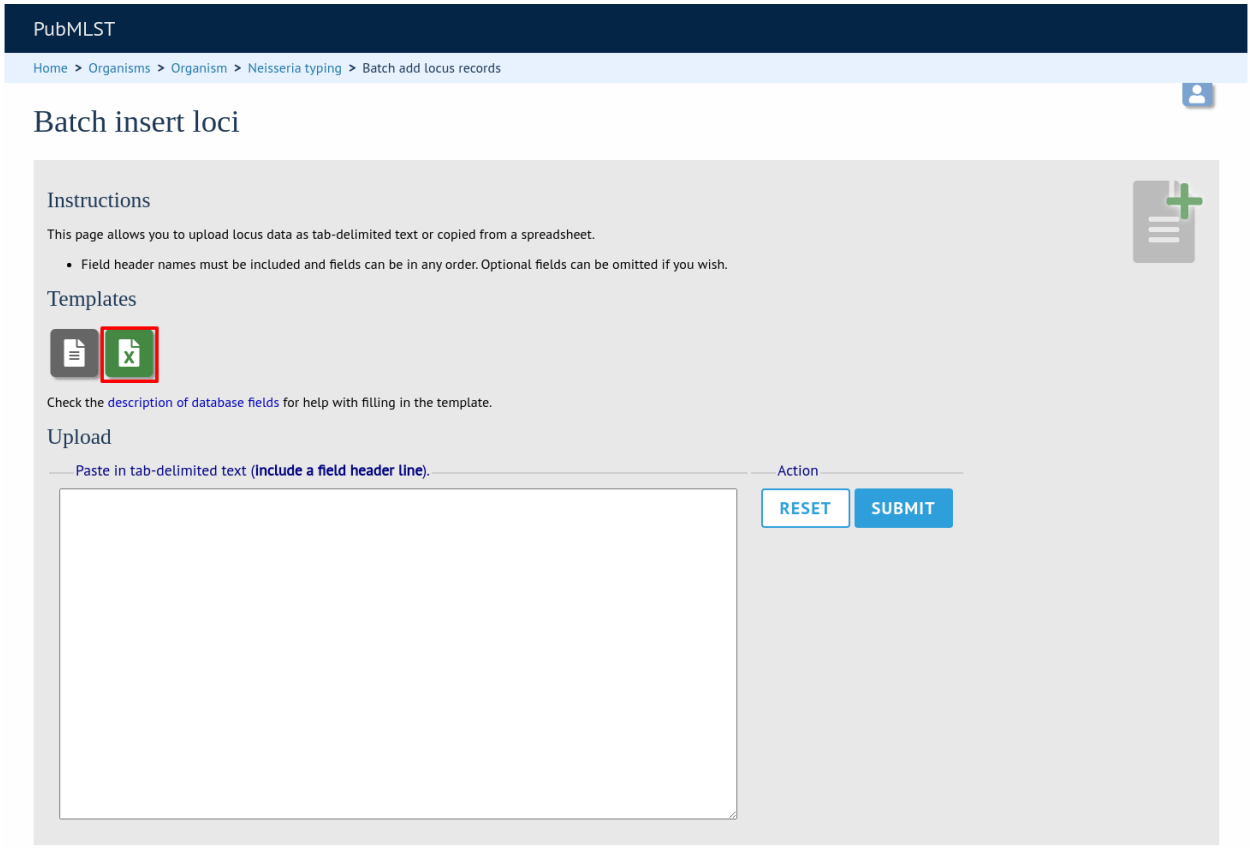
- description - Description of the locus (optional).
  - Allowed: any value.
- aliases - Alternative names for the locus (optional).
  - Enter each alias on a separate line in the text box.
  - Allowed: any value.
- pubmed_ids - PubMed ids of publications describing the locus (optional).
  - Enter each PubMed id on a separate line in the text box.
  - Allowed: any integer.
- links - Hyperlinks pointing to additional resources to display in the locus description (optional).
  - Enter each link on a separate line in the format with the URL first, followed by a | then the description (URL|description).

Batch adding

Click the batch add (++) loci link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



Click the link to download a header line for an Excel spreadsheet:



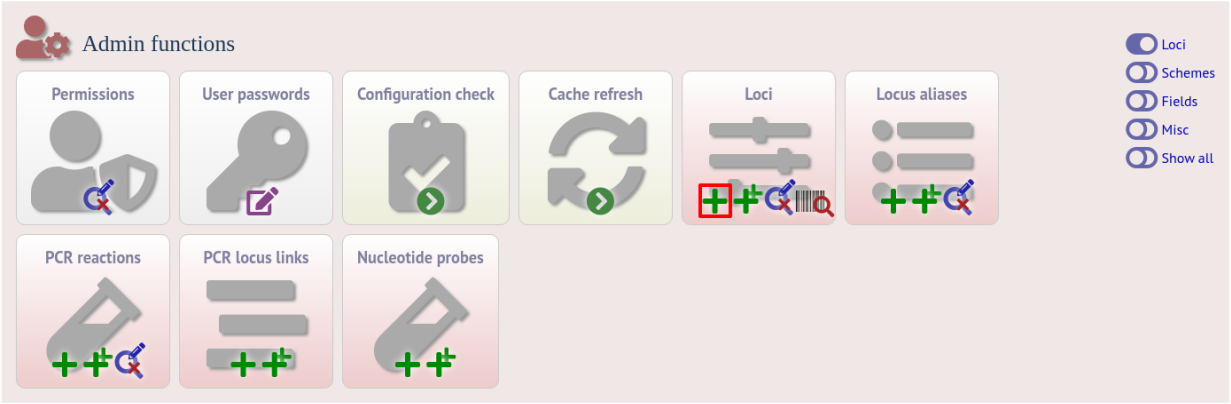
Fill in the spreadsheet using the fields described for *adding single loci*.

Fill in the spreadsheet fields using the table above as a guide, then paste the completed table into the web form and press 'Submit query'.

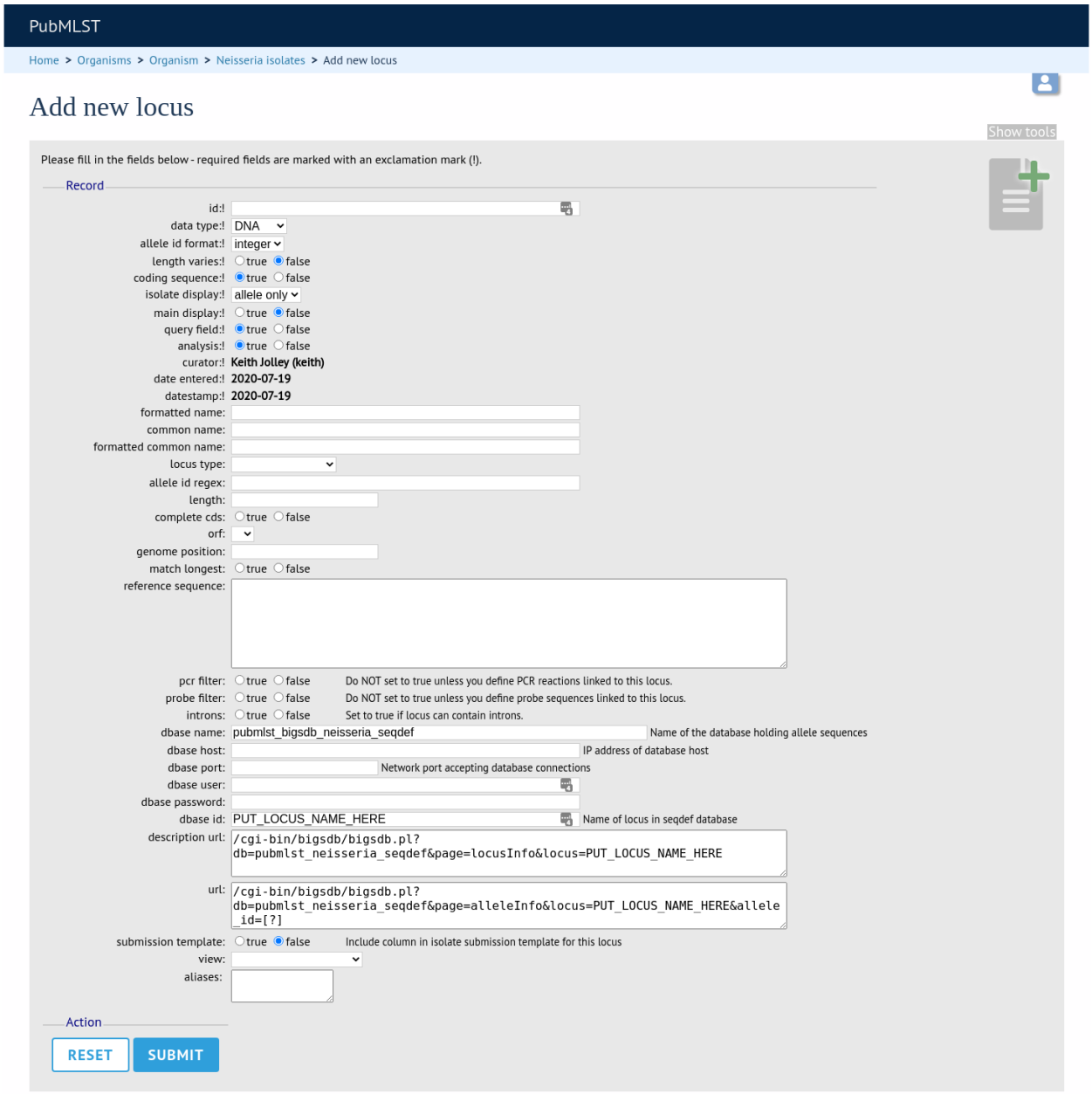
5.14.2 Isolate databases

Single locus

Click the add (+) loci link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:



- id - The name of the locus
  - Allowed: any value starting with a letter or underscore.
- data_type - Describes whether the locus is defined by nucleotide or peptide sequence.
  - Allowed: DNA/peptide.
- allele_id_format - The format for allele identifiers.
  - Allowed: integer/text.
- length_varies - Sets whether alleles can vary in length.
  - Allowed: true/false.
- coding_sequence - Sets whether the locus codes for a protein.

  - Allowed: true/false.

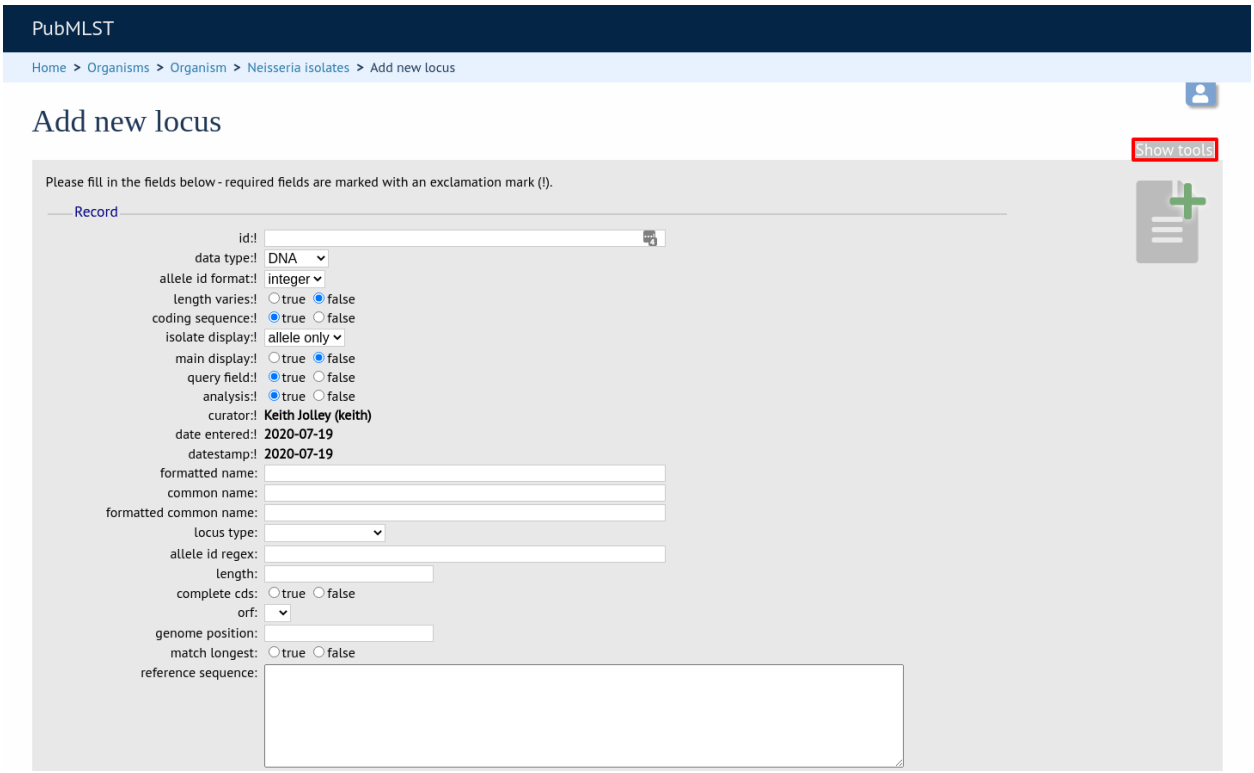
- isolate_display - Sets how alleles for this locus are displayed in a detailed isolate record - this can be overridden by user preference.
  - Allowed: allele only/sequence/hide.
- main_display - Sets whether or not alleles for this locus are displayed in a main results table by default - this can be overridden by user preference.
  - Allowed: true/false.
- query_field - Sets whether or not alleles for this locus can be used in queries by default - this can be overridden by user preference.
  - Allowed: true/false.
- analysis - Sets whether or not alleles for this locus can be used in analysis functions by default - this can be overridden by user preference.
  - Allowed: true/false.
- formatted_name - Name with HTML formatting (optional).
  - This allows you to add formatting such as bold, italic, underline and superscripting to locus names as they appear in the web interface.
  - Allowed: valid HTML.
- common_name - The common name for the locus (optional).
  - Allowed: any value.
- formatted_common_name - Common name with HTML formatting (optional).
  - Allowed: valid HTML.
- allele_id_regex - Regular expression to enforce allele id naming.
  - ^: the beginning of the string
  - $:the end of the string
  - d: digit
  - D: non-digit
  - s: white space character
  - S: non white space character
  - w: alpha-numeric plus '_'
  - .: any character
  - *: 0 or more of previous character
  - +: 1 or more of previous character
  - e.g. ^Fd-d+$ states that an allele name must begin with a F followed by a single digit, then a dash, then one or more digits, e.g. F1-12
- length - Standard length of locus (required if length_varies is set to false).
  - Allowed: any integer.
- complete_cds - Whether locus represents a complete coding sequence (optional)
- start_codons - Semi-colon separated list of alternative start codons to allow

  - Note that these are in addition to the built-in defaults of ATG, GTG, TTG.
- orf - Open reading frame of locus (optional). 1-3 are the forward reading frame, 4-6 are the reverse reading frames.
  - Allowed: 1-6.
- genome_position - The start position of the locus on a reference genome.
  - Allowed: any integer.
- match_longest - Only select the longest exact match when tagging/querying.
  - This is useful for some loci that can have some sequences shorter than others, e.g. peptide loci defining antigenic loops. This can lead to instances of one sequence being longer than another but otherwise being identical. In these cases, usually the longer sequence is the one that should be matched.
  - Allowed: true/false.
- reference_sequence - Sequence used by the automated sequence comparison algorithms to identify sequences matching this locus. **This is only used if a sequence definition database has not been set up for this locus.**
- pcr_filter - Set to true if this locus is further defined by genome filtering using in silico PCR.
  - Allowed: true/false.
- probe_filter - Set to true if this locus is further defined by genome filtering using in silico hybdridization.
  - Allowed: true/false.
- introns - Set to true if locus may contain introns. This setting will only be available if BLAT is configured in bigsdb.conf.
  - Allowed: true/false.
- dbase_name - Name of database (system name).
  - Allowed: any text.
- dbase_host - Resolved name of IP address of database host - leave blank if running on the same machine as the isolate database.
  - Allowed: network address, e.g. 129.67.26.52 or zoo-oban.zoo.ox.ac.uk
- dbase_port - Network port on which the sequence definition database server is listening - leave blank unless using a non-standard port (5432).
  - Allowed: integer.
- dbase_user - Name of user with permission to access the sequence definition database - depending on the database configuration you may be able to leave this blank.
  - Allowed: any text (no spaces).
- dbase_password - Password of database user - again depending on the database configuration you may be able to leave this blank.
  - Allowed: any text (no spaces).
- dbase_id - Name of locus in seqdef database. This is usually the same as the id field.
  - Allowed: any text (no spaces).
- description_url - The URL used to hyperlink to locus information in the isolate information page. This can either be a relative (e.g. /cgi-bin/…) or an absolute (containing http://) URL.
  - Allowed: any valid URL.

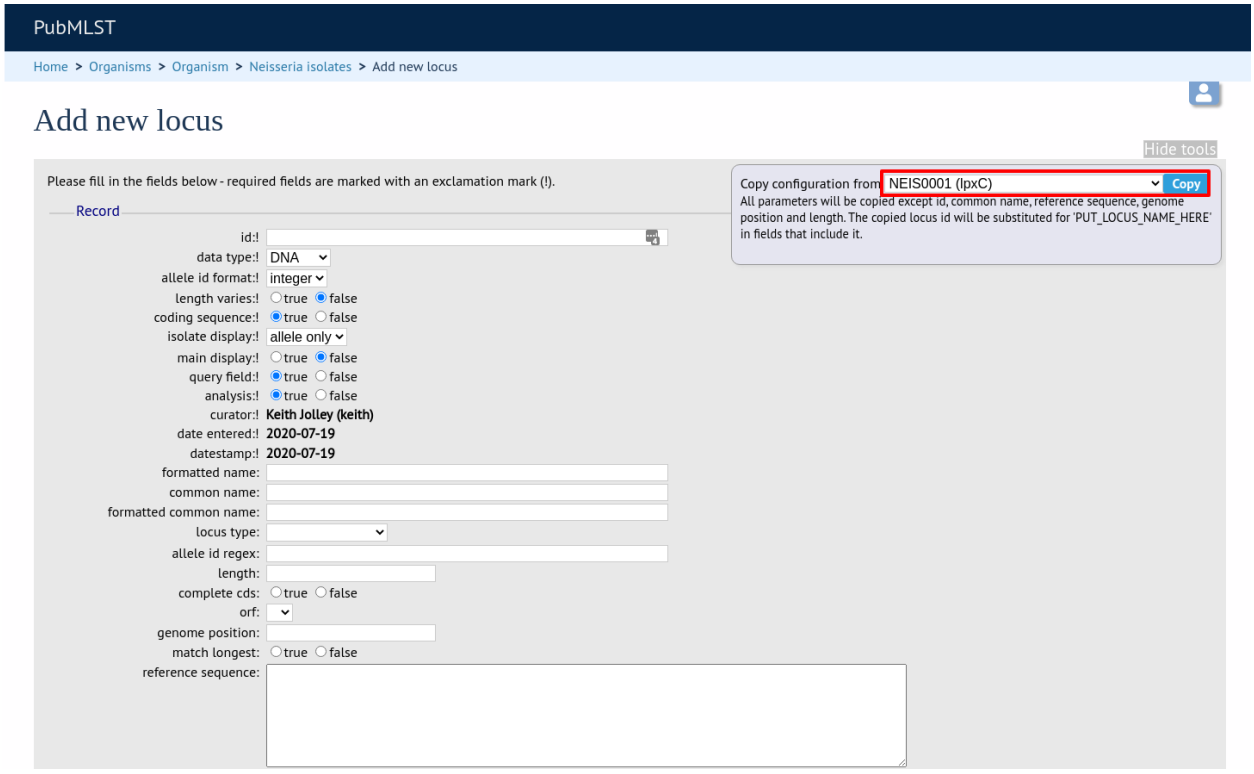
- url - The URL used to hyperlink to information about the allele. This can either be a relative or absolute URL. If [?] (including the square brackets) is included then this will be substituted for the allele value in the resultant URL. To link to the appropiate allele info page on a corresponding seqdef database you would need something like /cgi-bin/bigsdb/bigsdb.pl?db=pubmlst_neisseria_seqdef&page=alleleInfo&locus=abcZ&allele_id=[?].
  - Allowed: any valid URL.
- submission_template - Sets whether or not a column for this locus is included in the Excel submission template.
  - Allowed: true/false (default: false)
- view - Restrict this locus to only isolates contained in the specified database view. This option will only appear if the views attribute is set in the system tag. The view needs to have been defined in the database as a subset of the isolate table (usually filtered by the value of one or more of its fields).

Using existing locus definition as a template

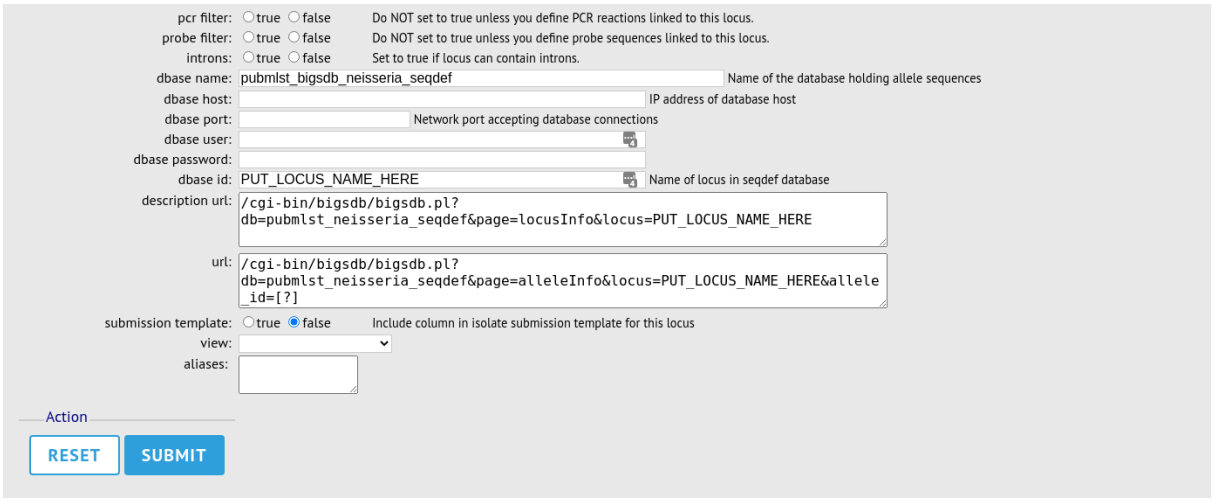
When defining a new locus in the isolate database, it is possible to use an existing locus record as a template. To do this, click the 'Show tools' link in the top-right of the screen:



This displays a drop-down box containing existing loci. Select the locus that you wish to use as a template, and click 'Copy'.



The configuration will be copied over to the web form, with the exception of name fields. Some fields will require you to change the value 'PUT_LOCUS_NAME_HERE' with the value you enter in the id field. These are usually the dbase_id2_value, description_url and url fields:

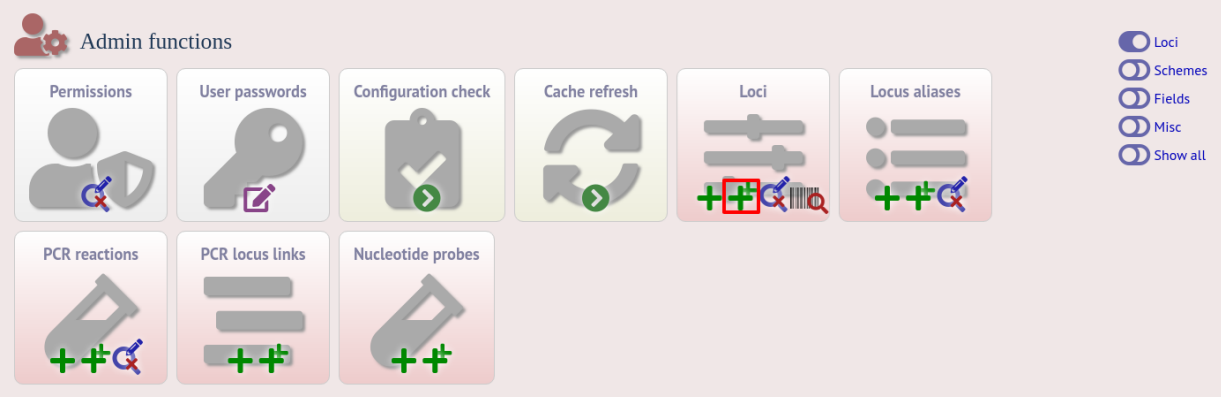


Complete the form and click 'Submit'.

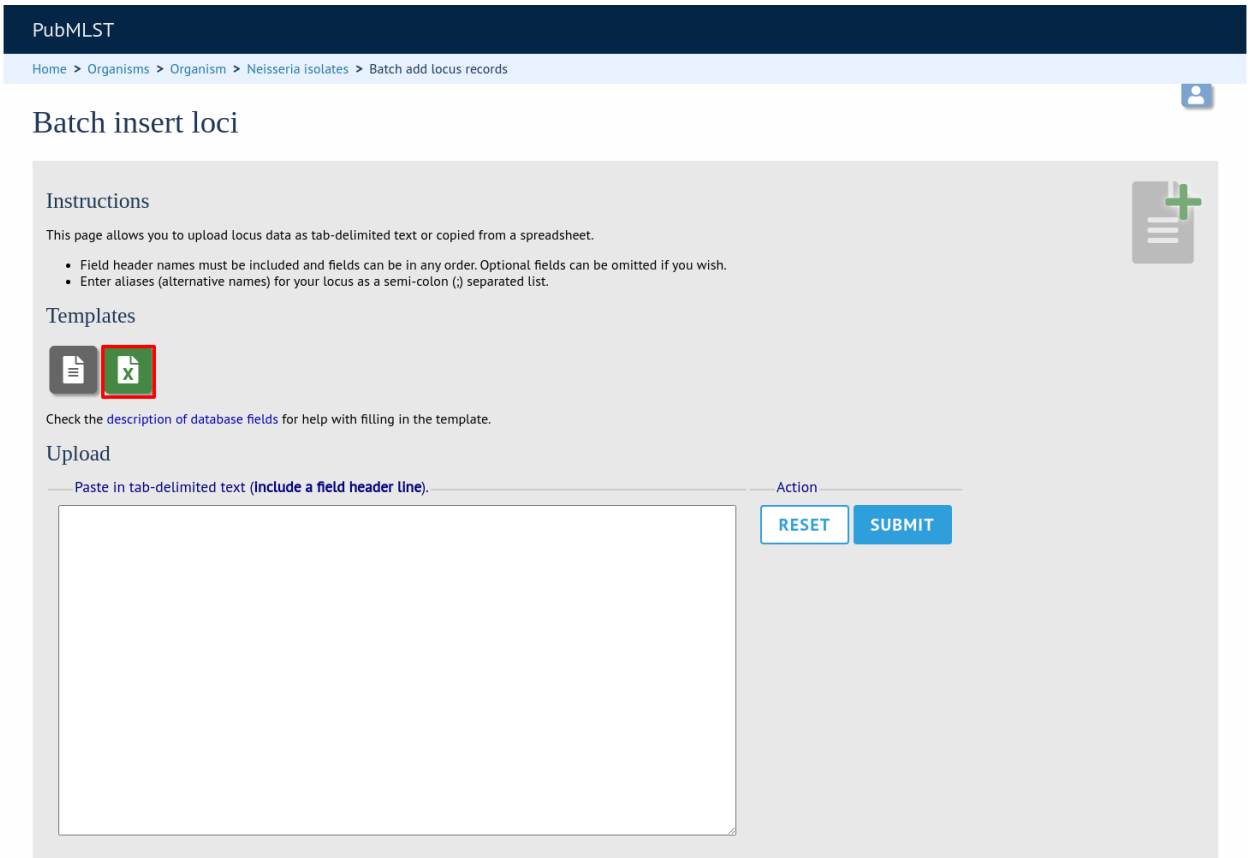
Note: You can also pre-populate the dbase_name, dbase_url and url fields with boilerplate values by setting the default_seqdef_config and default_seqdef_dbase values in the system attribute of the config.xml file.

Batch adding

Click the batch add (++) loci link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



Click the link to download an Excel template:

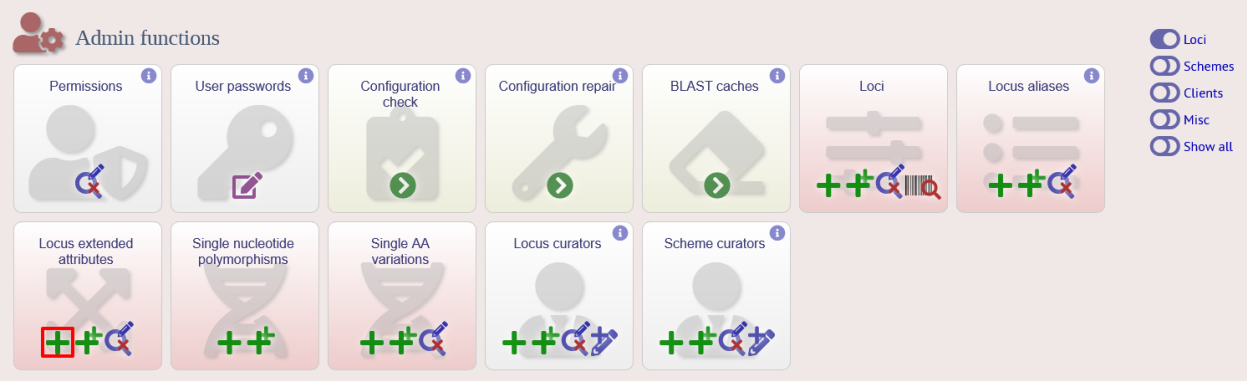


Fill in the spreadsheet fields using the *table above as a guide*, then paste the completed table into the web form and press 'Submit query'.

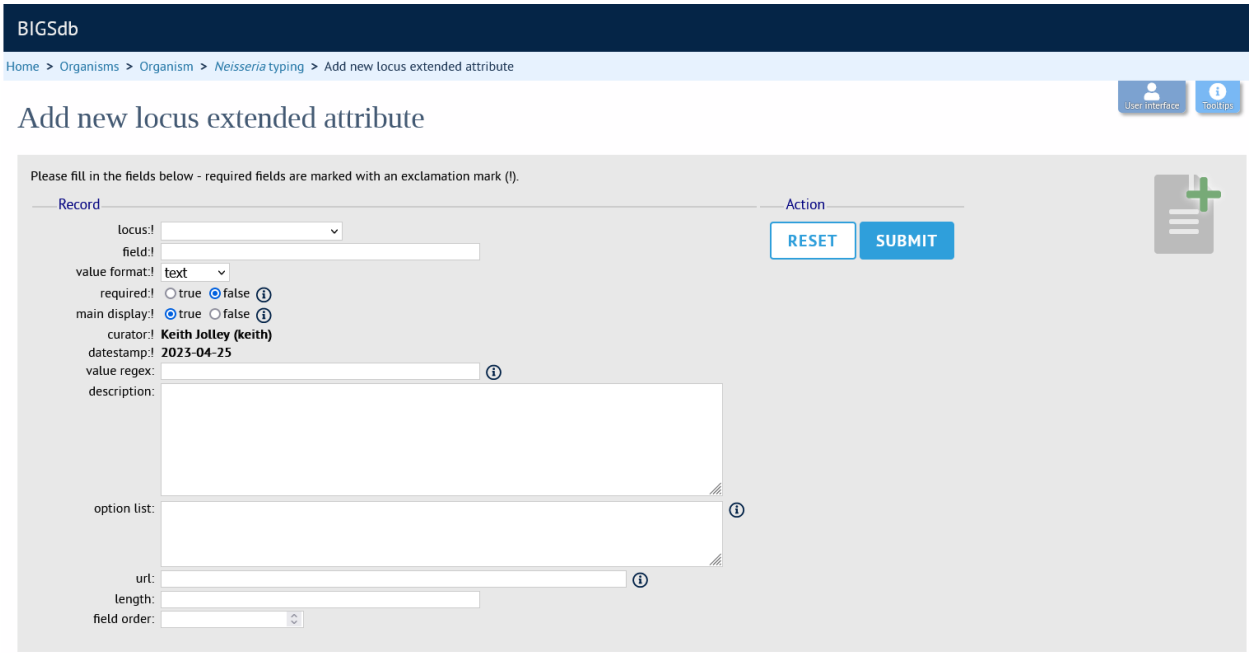
5.15 Defining locus extended attributes

You may want to add additional metadata for the allele definitions of some loci. Since these are likely to be specific to each locus, they cannot be defined generically within the standard locus definition. We can, instead, define extended attributes. Examples of these include higher order grouping of antigen sequences, antibody reactivities, identification of important mutations, or cross-referencing of alternative nomenclatures.

To add extended attributes for a locus, click add (+) locus extended attributes in the sequence definition database curator's interface contents page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:



- locus - Select locus from dropdown box.
  - Allowed: existing locus name.
- field - Name of extended attributes.
  - Allowed: any value.
- value_format - Data type of attribute.

  - Allowed: integer/text/boolean.
- required - Specifies whether the attribute value but be defined when adding a new sequence.
  - Allowed: true/false.
- value_regex - Regular expression to enforce allele id naming (optional).
  - ^: the beginning of the string
  - $:the end of the string
  - d: digit
  - D: non-digit
  - s: white space character
  - S: non white space character
  - w: alpha-numeric plus '_'
  - .: any character
  - *: 0 or more of previous character
  - +: 1 or more of previous character
- description - Description that will appear within the web form when adding new sequences (optional).
  - Allowed: any value.
- option_list - Pipe (|) separated list of allowed values (optional).
- length - Maximum length of value (optional).
  - Allowed: any integer.
- field_order - Integer that sets the position of the field within scheme values in any results (optional).
  - Allowed: any integer.

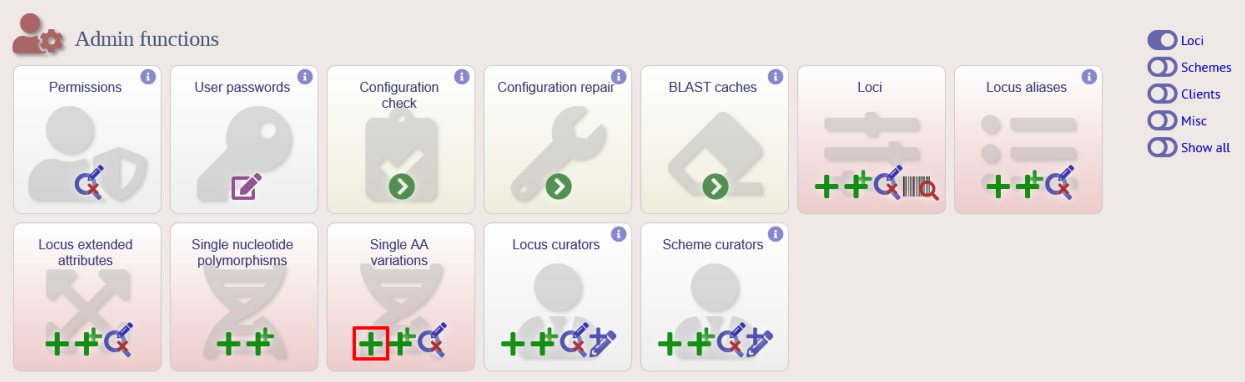
Once extended attributes have been defined, they will appear in the web form when adding new sequences for that locus. The values are searchable when using a *locus-specific sequence query*, and they will appear within query results and allele information pages.

5.16 Defining locus amino acid variants or single-nucleotide polymorphisms

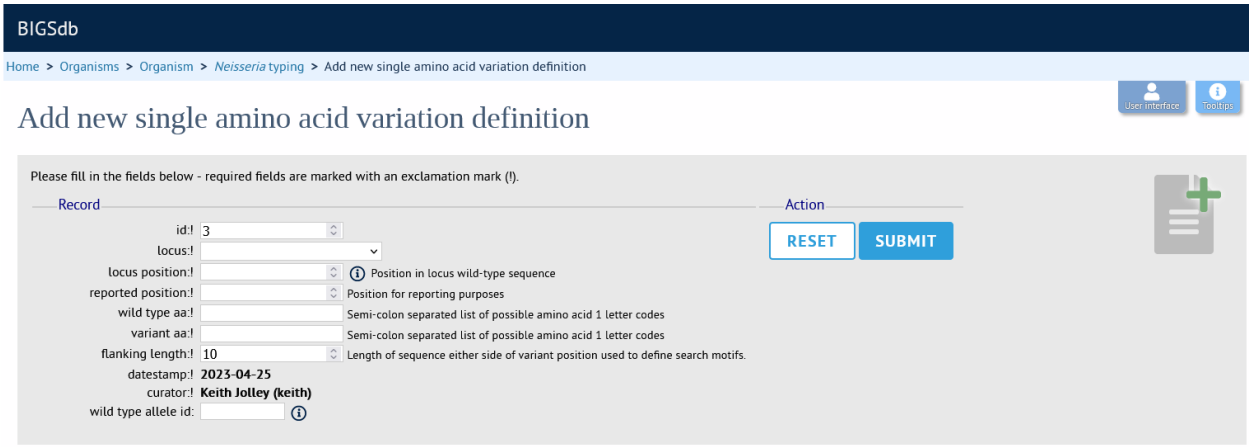
It is possible to annotate allele sequences with specific single amino acid variants (SAAV) or single nucleotide polymorphisms (SNPs). The values are searchable when using a *locus-specific sequence query*, and they will appear within query results and allele information pages.

You can add SAAVs to both DNA and protein loci. If adding to a DNA locus the alleles are translated so that the positions refer to the amino acid sequence.

To add a SAAV for a locus, click add (+) single AA variations in the sequence definition database curator's interface contents page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:



- id - Index number of variant - the next available number will be entered automatically.
  - Allowed: any positive integer.
- locus - Select locus from dropdown box.
  - Allowed: existing locus name.
- locus_position
  - Position in locus wild-type sequence. Position refers to the translated sequence if using a DNA locus.
- reported_position
  - Position to use in the output. This will be the same as the locus_position above if the locus includes the complete coding sequence, but will likely be different if the locus is an internal fragment.
- wild_type_aa
  - Semi-colon (;) separated list of possible wild type amino acids (single letter codes).
- variant_aa
  - Semi-colon (;) separated list of possible variant amino acids (single letter codes).
- flanking_length
  - Length of sequence either side of the variant position used to define search motifs. A default value will be provided automatically, but this can be modified if the search is failing.
- wild_type_allele_id

  - The identifier of a standard length wild-type allele. This can be optionally provided to help define search motifs. If not provided then the search script will attempt to identify a wild-type allele automatically.

Once defined SAAVs can be annotated by running the `scan_mutations.pl` script found in the scripts/automation directory.

SNPs can be defined in the same way by adding values to the 'Single nucleotide polymorphims' table. This has slight differences in the field names (wild_type_nuc and variant_nuc) but the process is otherwise identical. Note that SNPs can only be defined for DNA loci.

5.17 Defining schemes

Schemes are collections of loci that may be associated with particular fields - one of these fields can be a primary key, i.e. a field that uniquely defines a particular combination of alleles at the associated loci.

A specific example of a scheme is MLST - *see workflow for setting up a MLST scheme*.

To set up a new scheme, you need to:

1. Add a new scheme description.
2. Define loci as 'scheme members'.
3. Add 'scheme fields' associated with the scheme.

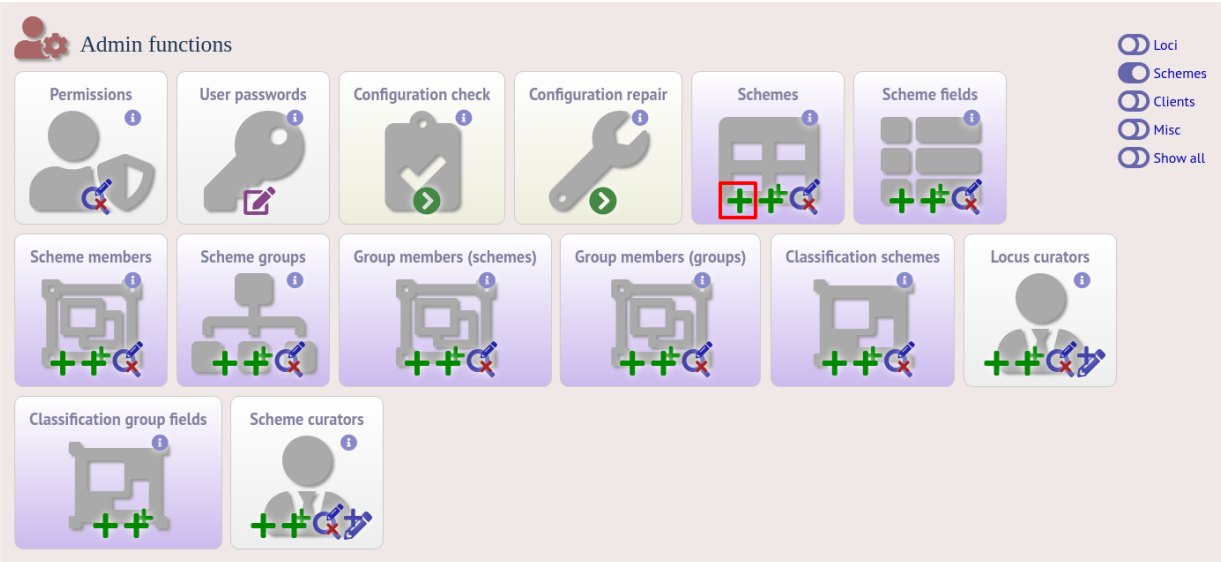
See also:

Schemes (concept)

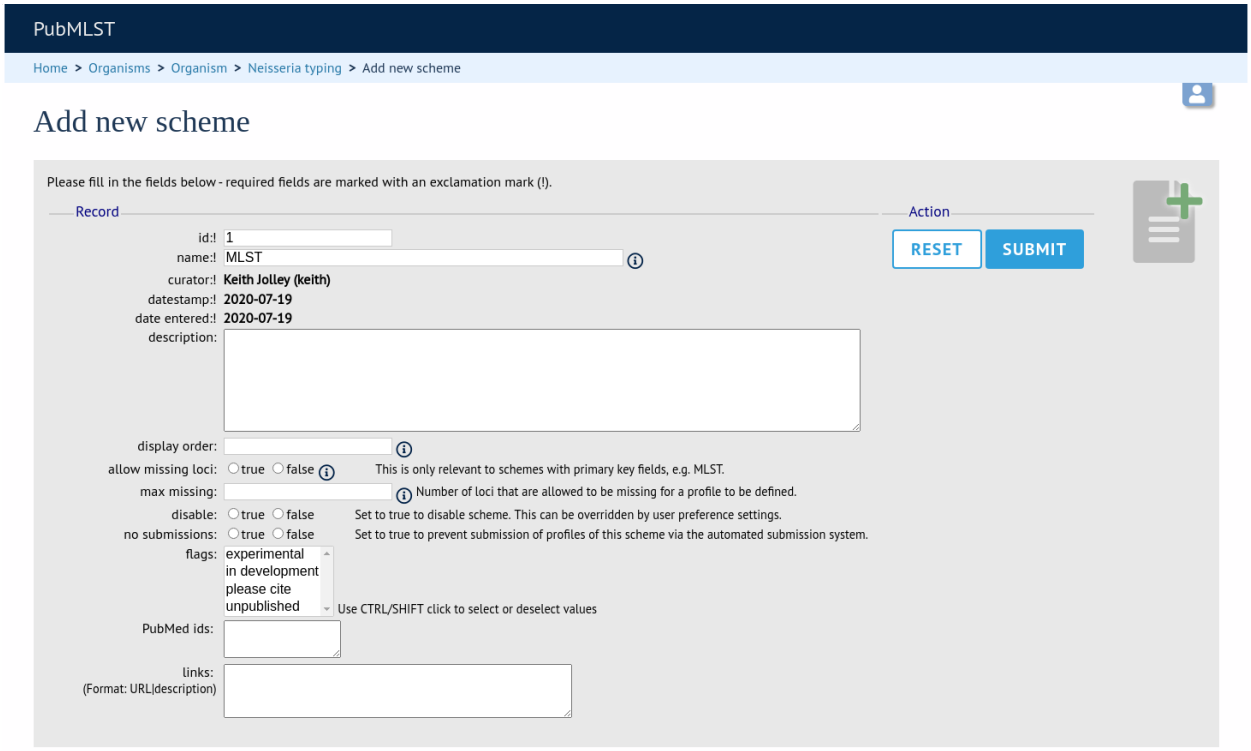
5.17.1 Sequence definition databases

As with all configuration, tables can be populated using the batch interface (++) or one at a time (+). Details for the latter are described below:

Click the add (+) scheme link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it.

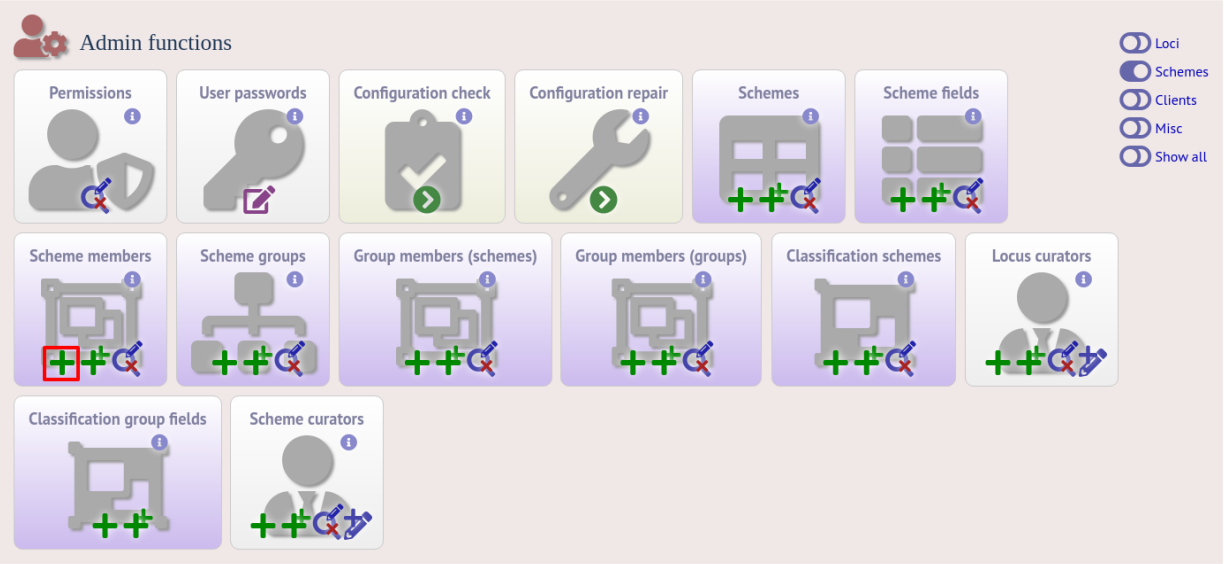


Fill in the scheme description in the web form. The next available scheme id number will be filled in already.

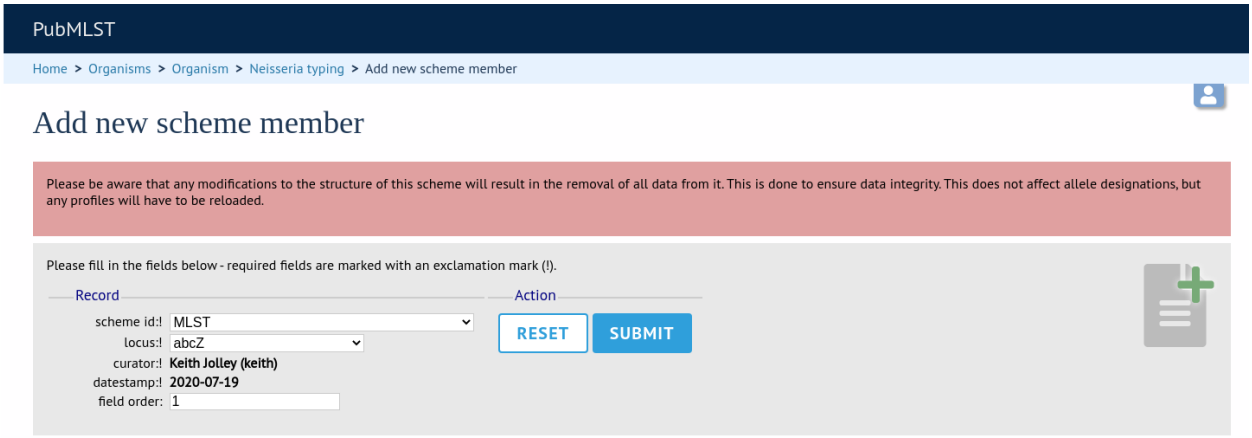


- id - Index number of scheme - the next available number will be entered automatically.
  - Allowed: any positive integer.
- name - Name of scheme.
  - Allowed: any text.
- description - More detailed description of scheme.
  - Allowed: any text.
- display_order - Integer reflecting the display position for this scheme within the interface (optional).
  - Allowed: any integer.
- allow_missing_loci - Allow profile definitions to contain '0' (locus missing) or 'N' (any allele).
- allow_presence - Allow profile definitions to contain 'P' (locus present).
- disable - Disable the scheme so that it is hidden by default. Users can specifically override this to enable the scheme, so it not actually private.
- no_submissions - Do not include scheme in the submission interface for submission of new profiles (only relevant for schemes with primary key fields).

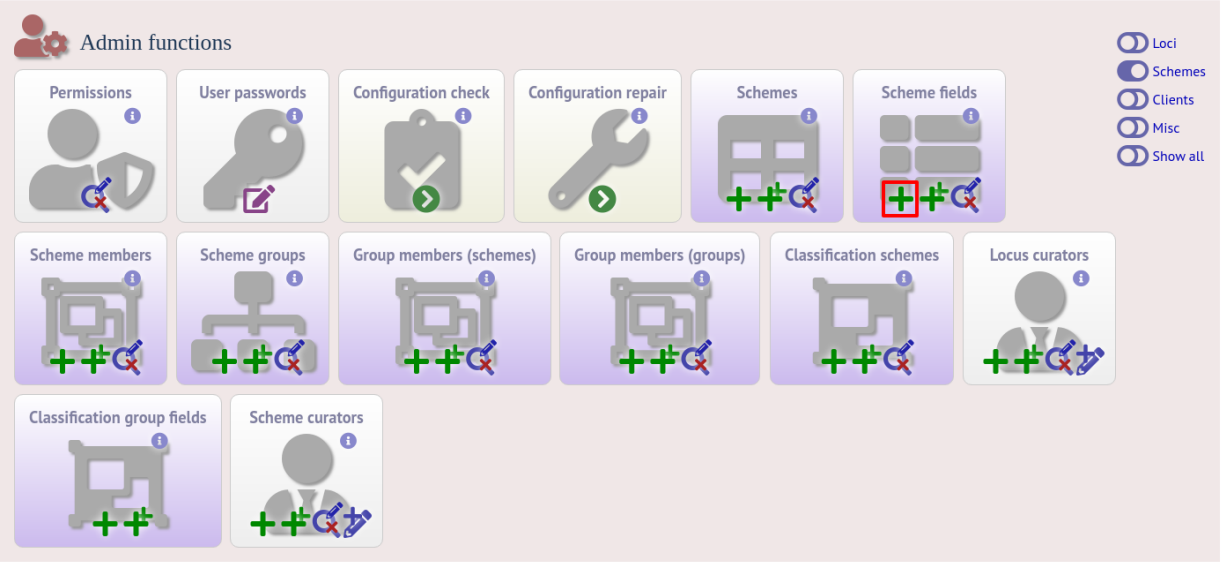
To add loci to the scheme, click the add (+) scheme members link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it.



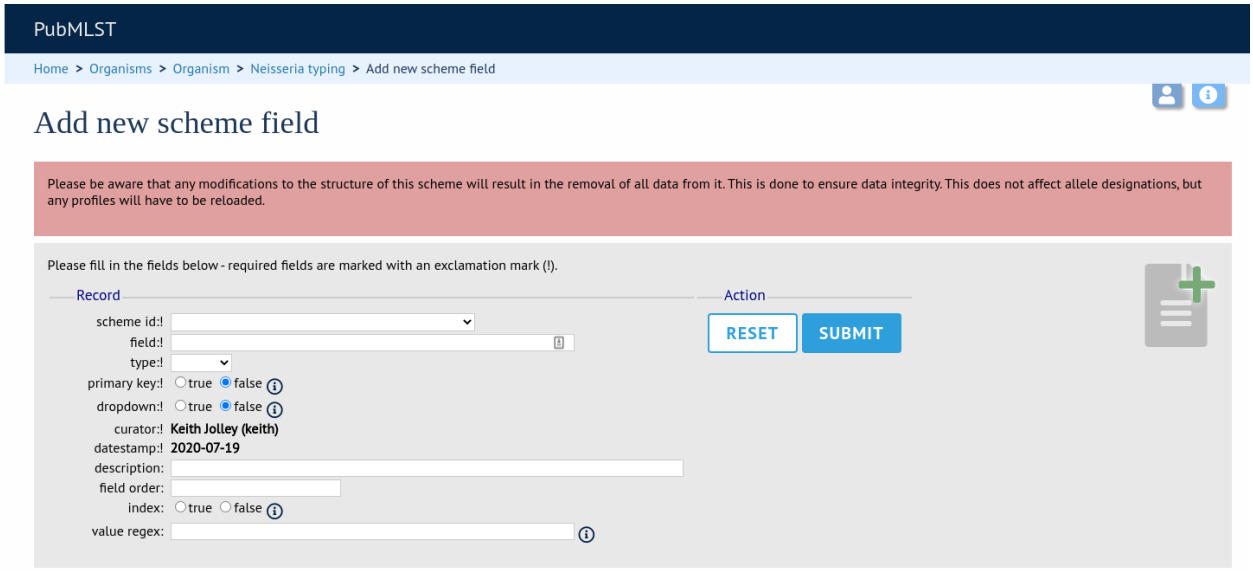
Select the scheme name and a locus that you wish to add to the scheme from the appropriate drop-down boxes. *Loci need to have already been defined*. The field_order field allows you to set the display order of the locus within a profile - if these are left blank that alphabetical ordering is used.



To add scheme fields, click the add (+) scheme fields link on the curator's interface contents page.



Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:



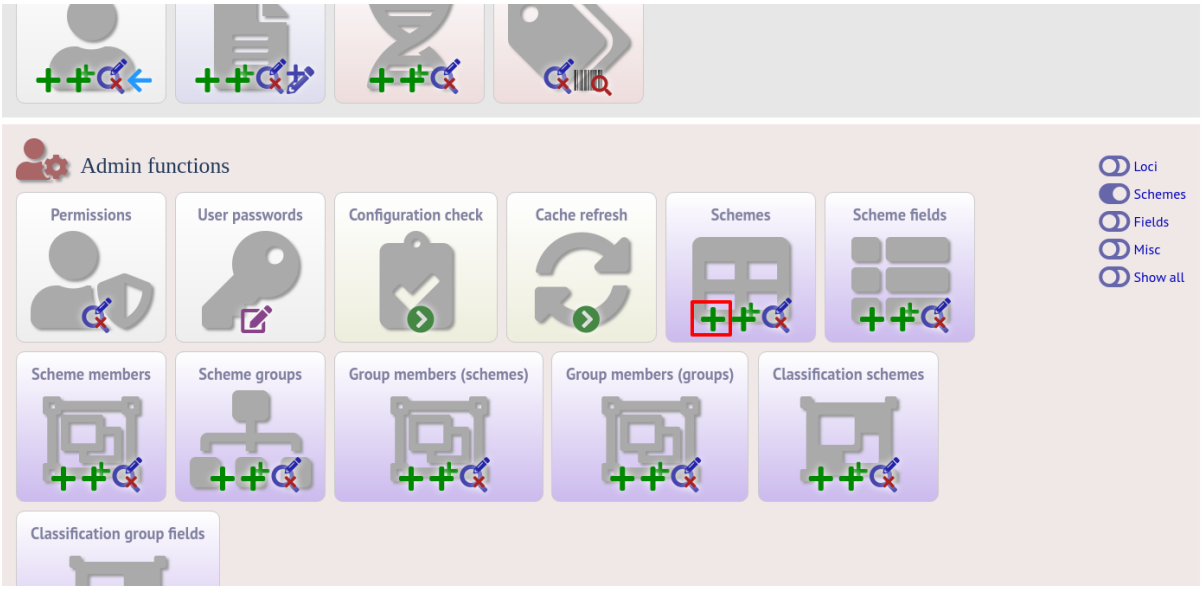
- scheme_id - Dropdown box of scheme names.
  - Allowed: selection from list.
- field - Field name.
  - Allowed: any value.
- type - Format for values.
  - Allowed: text/integer/date.
- primary_key - Set to true if field is the primary key. There can only be one primary key for a scheme.
  - Allowed: true/false.
- dropdown - Set to true if a dropdown box is displayed in the query interface, by default, for values of this field to be quickly selected. This option can be overridden by user preferences.

  - Allowed: true/false.
- description - This field isn't currently used.
- field_order - Integer that sets the position of the field within scheme values in any results.
  - Allowed: any integer.
- value_regex - Regular expression to enforce field values.
  - ^: the beginning of the string
  - $:the end of the string
  - d: digit
  - D: non-digit
  - s: white space character
  - S: non white space character
  - w: alpha-numeric plus '_'
  - .: any character
  - *: 0 or more of previous character
  - +: 1 or more of previous character

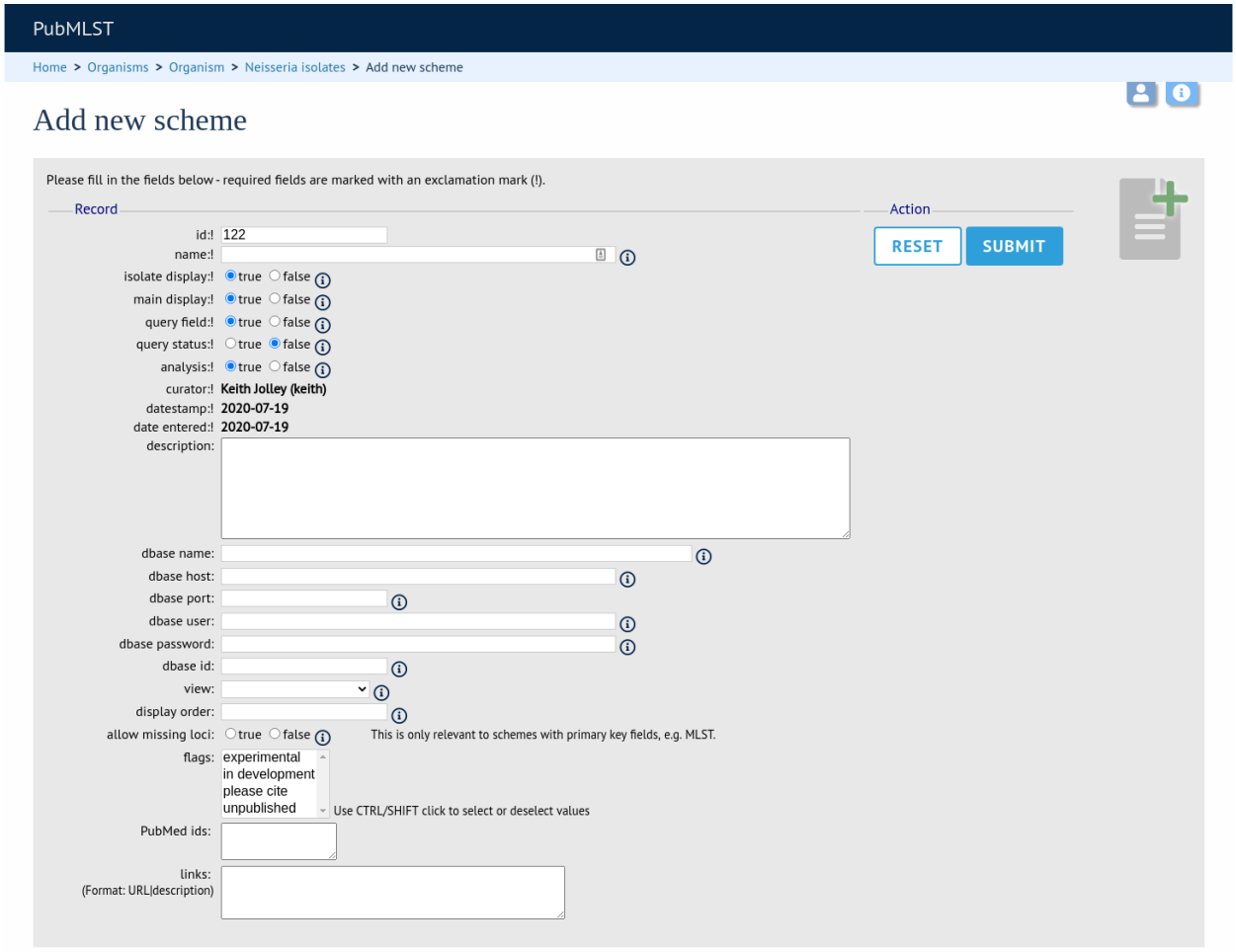
5.17.2 Isolate databases

As with all configuration, tables can be populated using the batch interface (++) or one at a time (+). Details for the latter are described below:

Click the add (+) scheme link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it.



Fill in the scheme description in the web form. Required fields have an exclamation mark (!) next to them:



- id - Index number of scheme - the next available number will be entered automatically.
  - Allowed: any positive integer.
- name - Name of scheme.
  - Allowed: any text.
- description - More detailed description of scheme.
  - Allowed: any text.
- isolate_display - Sets whether or not fields for this scheme are displayed in a detailed isolate record - this can be overridden by user preference.
  - Allowed: allele only/sequence/hide.
- main_display - Sets whether or not fields for this scheme are displayed in a main results table by default - this can be overridden by user preference.
  - Allowed: true/false.
- query_field - Sets whether or not fields for this scheme can be used in queries by default - this can be overridden by user preference.
  - Allowed: true/false.
- query_status - Sets whether a dropdown list box should be displayed in the query interface to filter results based on profile completion for this scheme - this can be overridden by user preference.

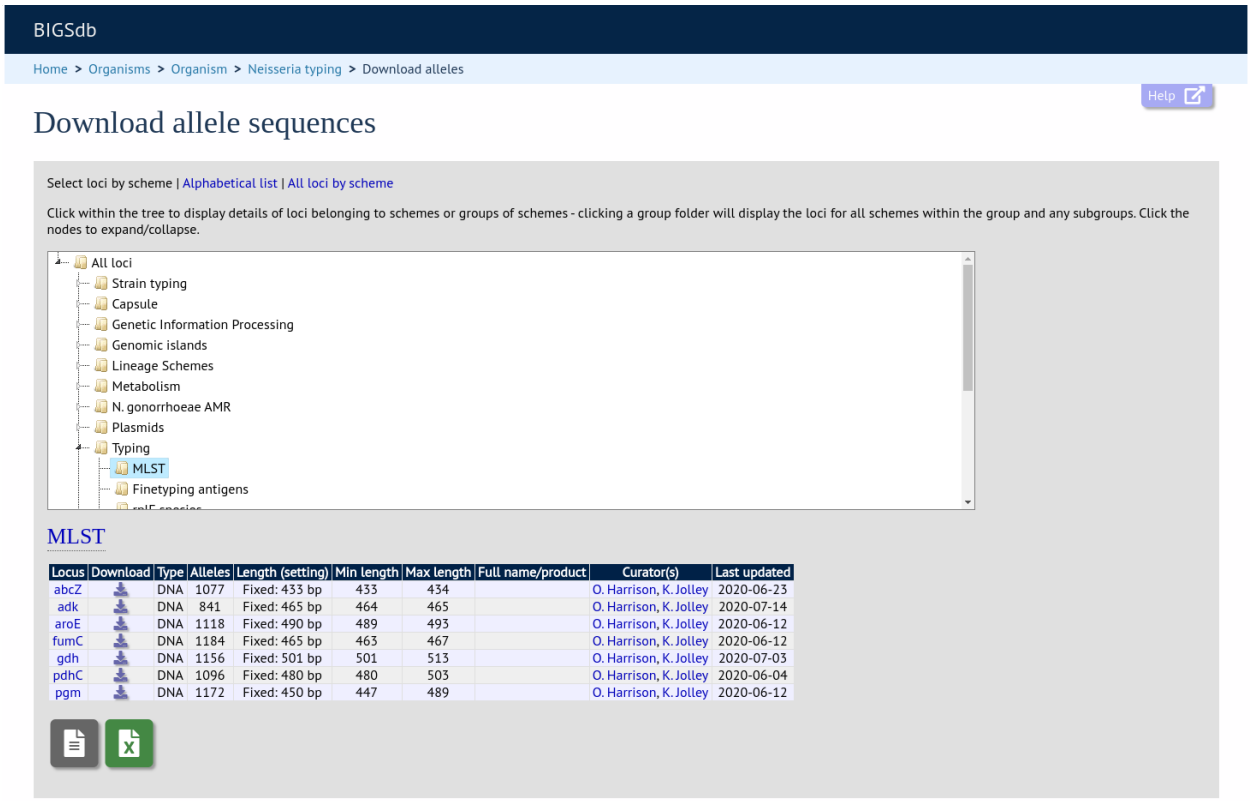
  - Allowed: true/false.

- analysis - Sets whether or not alleles for this locus can be used in analysis functions by default - this can be overridden by user preference.
  - Allowed: true/false.
- recommended - Sets whether the scheme will appear in a list of recommended schemes for use in some analysis plugins. Selecting this option makes the scheme easier to select when there are a lot of schemes defined. It should be used sparingly.
- quality_metric - Sets whether the scheme can be used to help assess the quality of a genome assembly. For a well annotated genome it would be expected for all loci in the scheme to have an allele designated. The annotation status can be searched in an isolate query. This can be used in conjunction with the quality_metric_good and quality_metric_bad attributes that can be used to set the thresholds for what constitutes a good or bad annotation.
- dbase_name - Name of seqdef database (system name) containing scheme profiles (optional).
  - Allowed: any text.
- dbase_host - Resolved name of IP address of database host - leave blank if running on the same machine as the isolate database (optional).
  - Allowed: network address, e.g. 129.67.26.52 or zoo-oban.zoo.ox.ac.uk
- dbase_port - Network port on which the sequence definition database server is listening - leave blank unless using a non-standard port, 5432 (optional).
  - Allowed: integer.
- dbase_user - Name of user with permission to access the sequence definition database - depending on the database configuration you may be able to leave this blank (optional).
  - Allowed: any text (no spaces).
- dbase_password - Password of database user - again depending on the database configuration you may be able to leave this blank (optional).
  - Allowed: any text (no spaces).
- dbase_id - Id of scheme in the sequence definition database.
  - Allowed: any integer.
- quality_metric_good - threshold number of loci that must have allele designations for a genome annotation to be considered good for this scheme. If this isn't set then the number of loci in the scheme is used.
- quality_metric_bad - threshold number of loci that must have allele designations below which a genome annotation is to be considered bad for this scheme. If this isn't set then the value used for quality_metric_good is used (or the number of scheme loci if this also is not set).
- view - Restrict this scheme to only isolates contained in the specified database view. This option will only appear if the views attribute is set in the system tag. The view needs to have been defined in the database as a subset of the isolate table (usually filtered by the value of one or more of its fields).
- display_order - Integer reflecting the display position for this scheme within the interface (optional).
  - Allowed: any integer.
- allow_missing_loci - Allow profile definitions to contain '0' (locus missing) or 'N' (any allele).
- allow_presence - Allow profile definitions to contain 'P' (locus present).

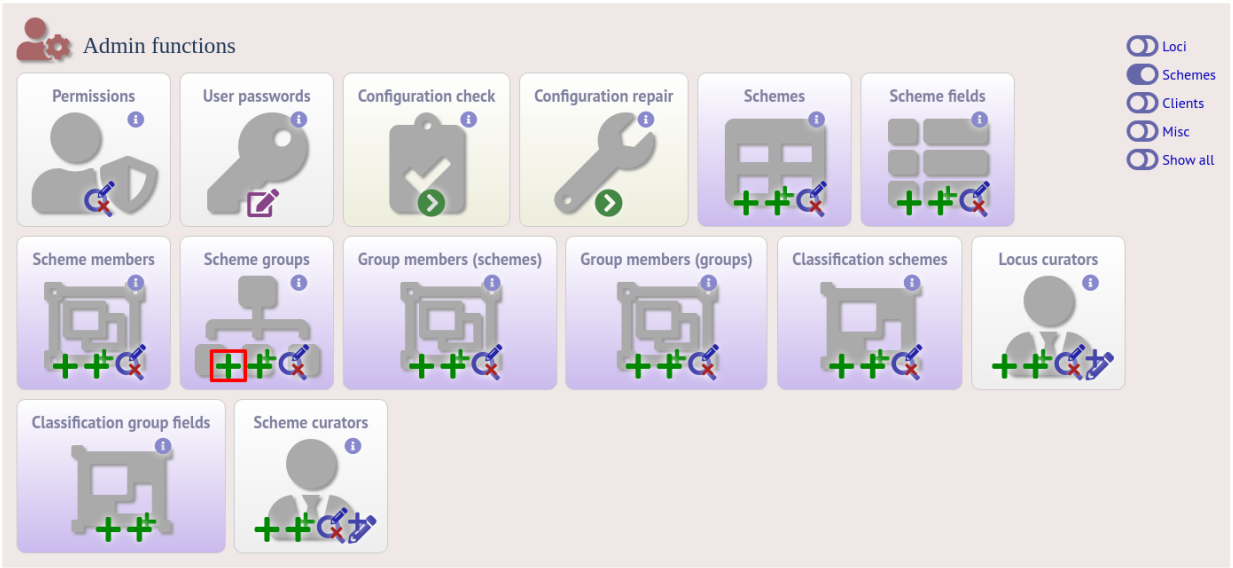
5.18 Organizing schemes into hierarchical groups

Schemes can be organized in to groups, and these groups can in turn be members of other groups. This faciliates hierarchical ordering of loci, but with the flexibility to allow loci and schemes to belong to multiple groups.

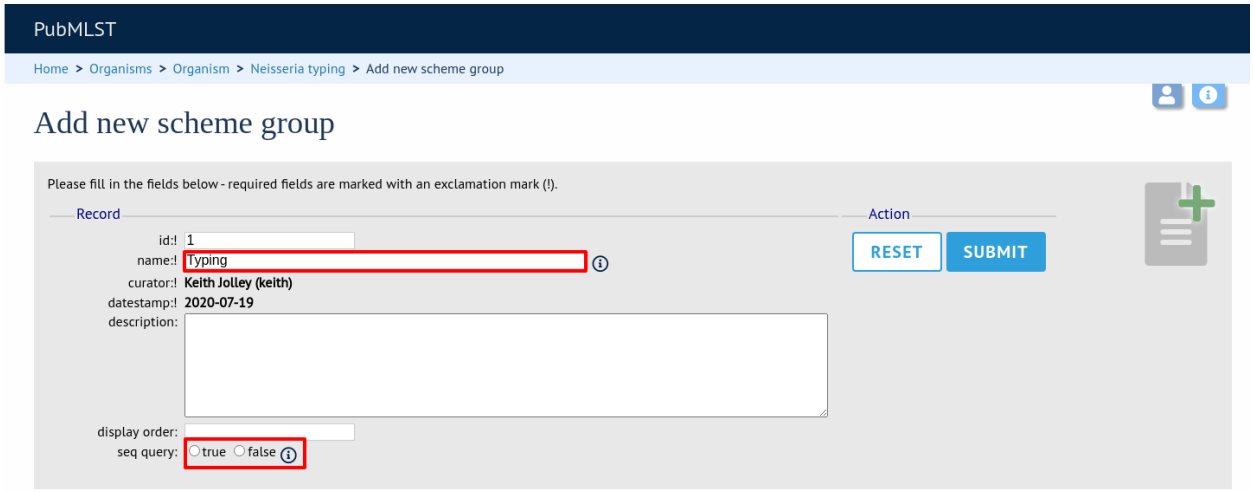
This hierarchical structuring can be seen in various places within BIGSdb, for example the *allele download* page.



Scheme groups can be added in both the sequence definition and isolate databases. To add a new group, click the add (+) scheme group link on the curator's contents page. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it.

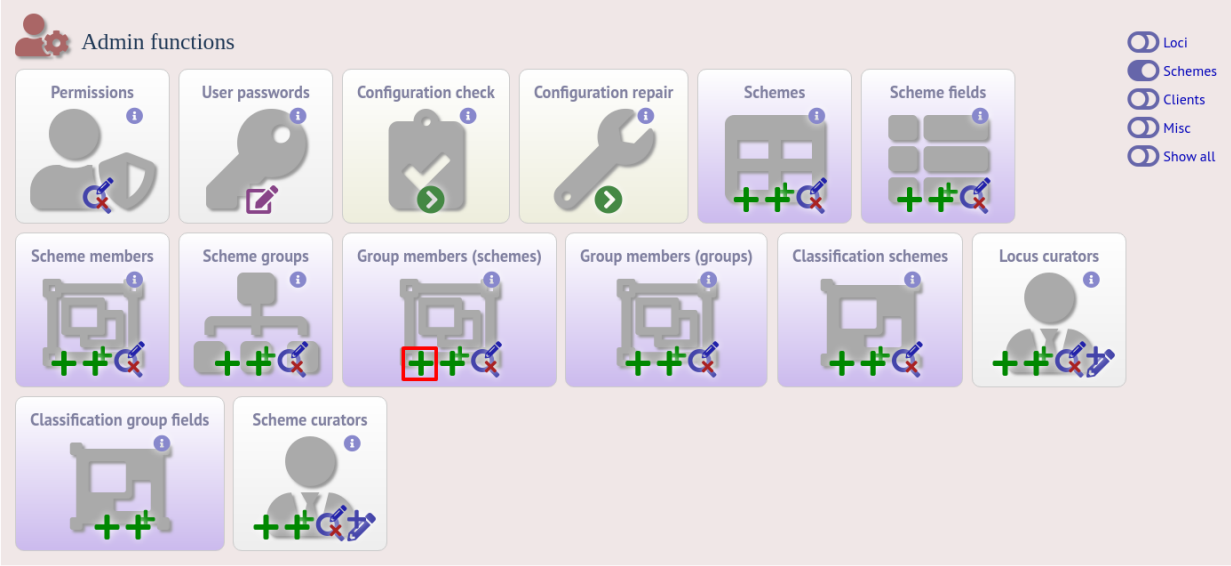


Enter a short name for the group - this will appear within drop-down list boxes and the hierarchical tree, so it needs to be fairly short.

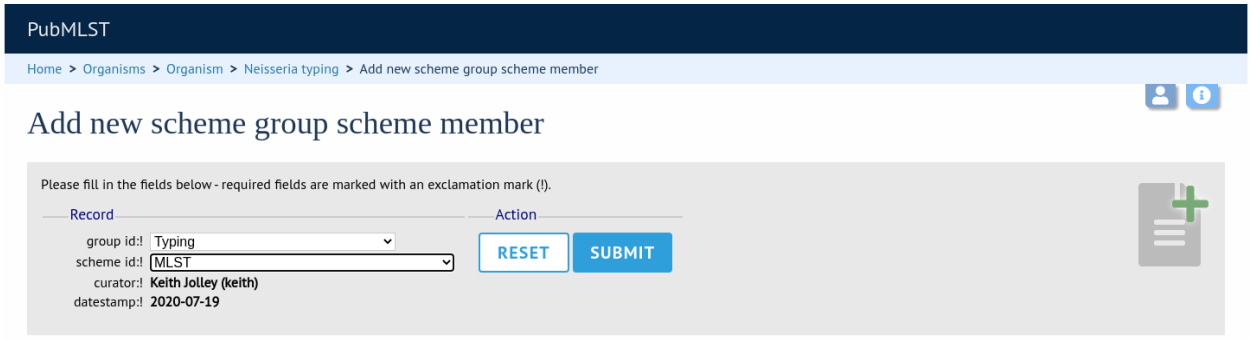


If you are creating a scheme group in the sequence definition database, there is an additional field called 'seq_query'. Set this to true to add the scheme group to the dropdown lists in the *sequence query* page. This enables all loci belonging to schemes within the group to be queried together.

Schemes can be added to groups by clicking the add (+) group members (scheme) link.



Select the scheme and the group to add it to, then click 'Submit'.



Scheme groups can be added to other scheme groups in the same way by clicking the add (+) scheme group group members link.

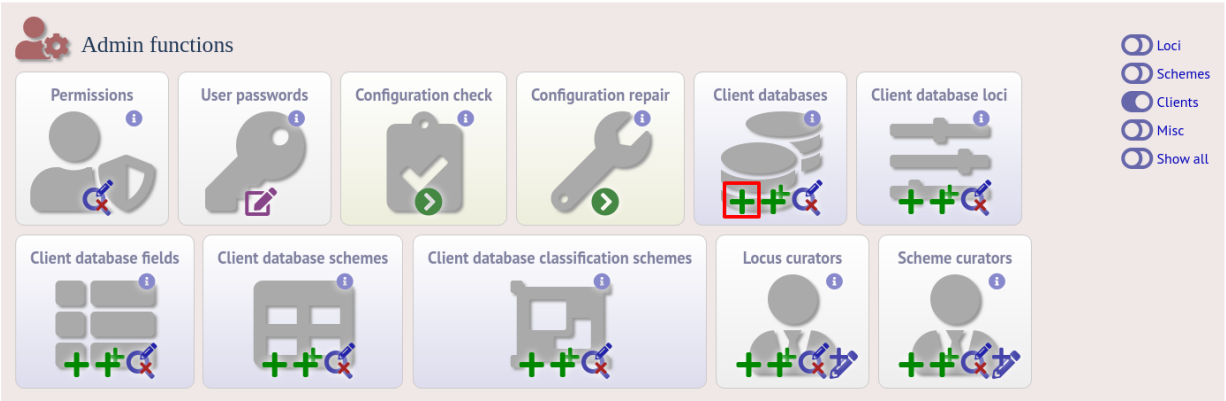
5.19 Setting up client databases

Sequence definition databases can have any number of isolate databases that connect as clients. Registering these databases allows the software to perform isolate data searches relevant to results returned by the sequence definition database, for example:

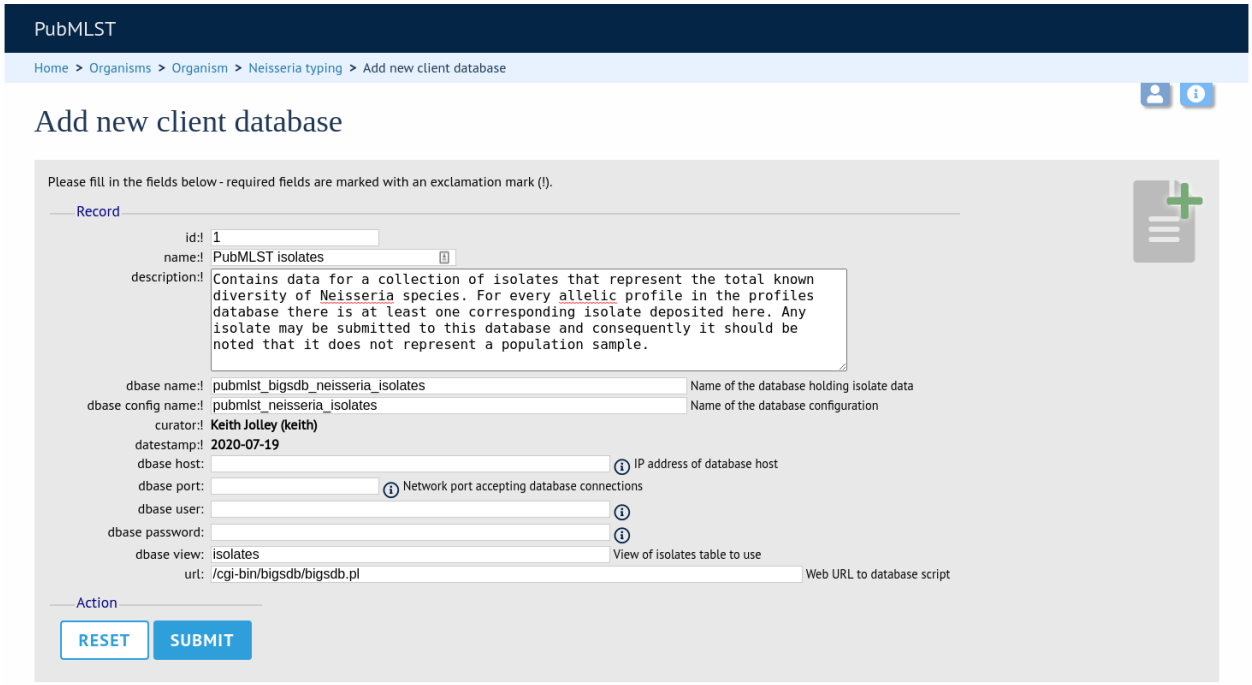
- Determine the number of isolates that a given allele is found in and link to these.
- Determine the number of isolates that a given scheme field, e.g. a sequence type, is found in and link to these.
- Retrieve specific attributes of isolates that have a given allele, e.g. species that have a particular 16S allele, or penicillin resistance given a particular penA allele.

Multiple client databases can be queried simultaneously.

To register a client isolate database for a sequence definition database, click the add (+) client database link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Clients' toggle to display it.



Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:

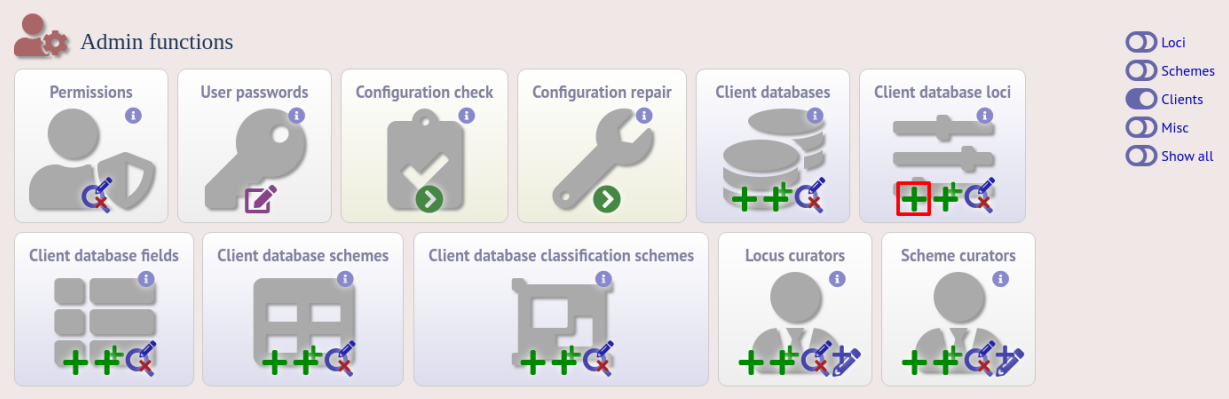


- id - Index number of client database. The next available number is entered automatically but can be overridden.
  - Allowed: any positive integer.
- name - Short description of database. This is used within the interface result tables so it is better to make it as short as possible.
  - Allowed: any text.
- description - Longer description of database.
  - Allowed: any text.
- dbase_name - Name of database (system name).
  - Allowed: any text.
- dbase_config_name - Name of database configuration - this is the text string that appears after the db= part of script URLs.
  - Allowed: any text (no spaces)
- dbase_host - Resolved name of IP address of database host (optional).
  - Allowed: Network address, e.g. 129.67.26.52 or zoo-oban.zoo.ox.ac.uk
  - Leave blank if running on the same machine as the seqdef database.
- dbase_port - Network port on which the client database server is listening (optional).
  - Allowed: integer.
  - Leave blank unless using a non-standard port (5432).
- dbase_user - Name of user with permission to access the client database.
  - Allowed: any text (no spaces).
  - Depending on the database configuration you may be able to leave this blank.
- dbase_password - Password of database user

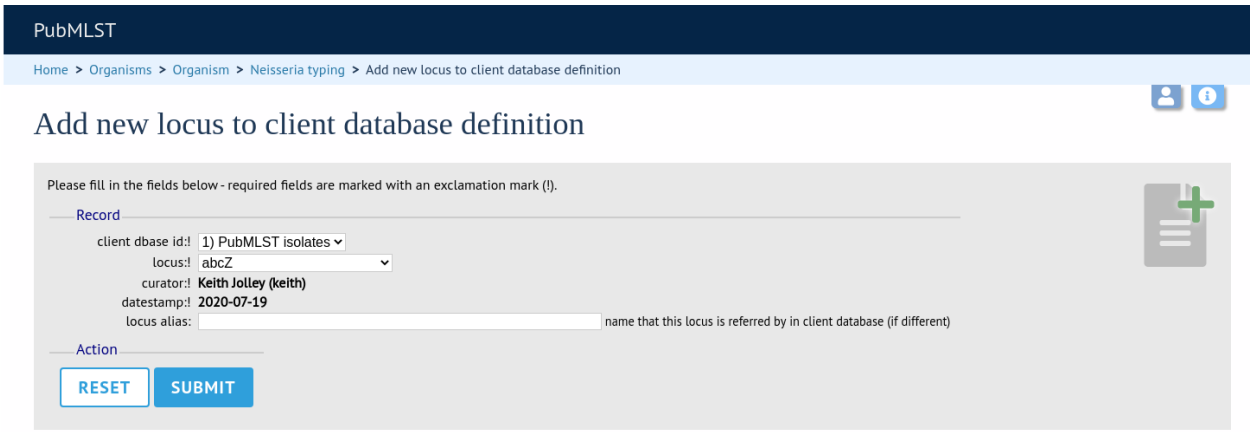
  - Allowed: any text (no spaces).
  - Depending on the database configuration you may be able to leave this blank.
- url - URL of client database bigsdb.pl script
  - Allowed: valid script path.
  - This can be relative (e.g. /cgi-bin/bigsdb/bigsdb.pl) if running on the same machine as the seqdef database or absolute (including http://) if on a different machine.

5.19.1 Look up isolates with given allele

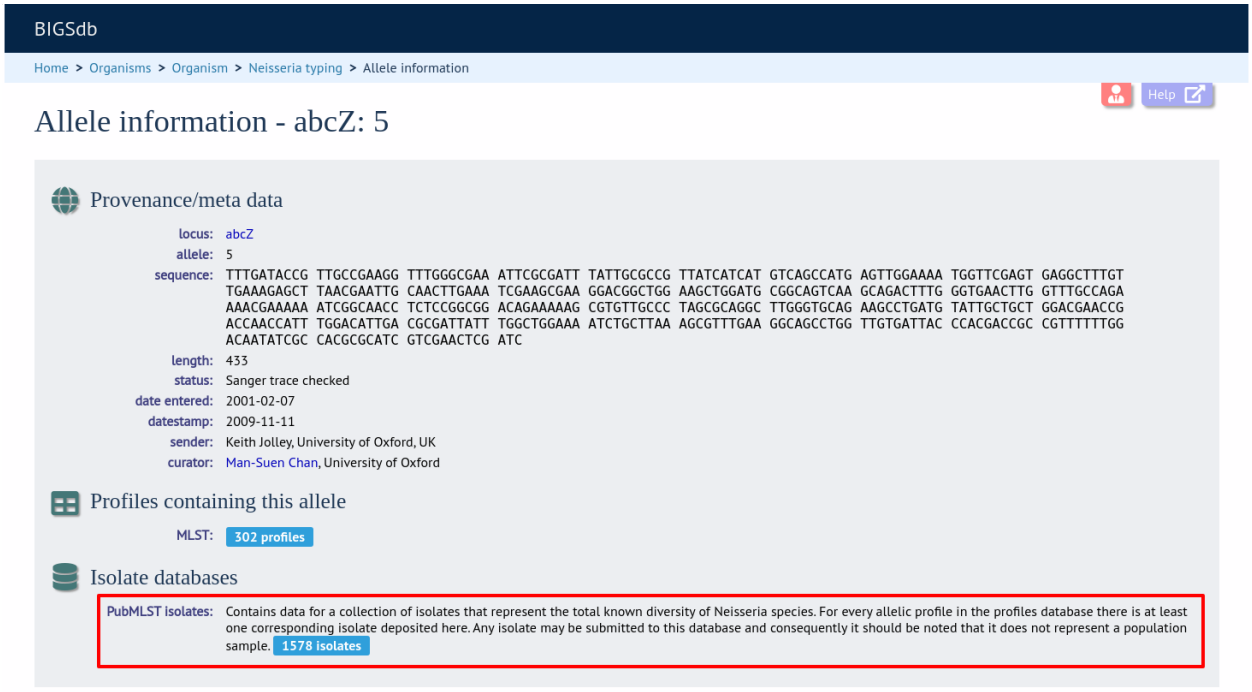
To link a locus, click the add (+) client database loci link on the curator's interface contents page.



Link the locus to the appropriate client database using the dropdown list boxes. If the locus is named differently in the client database, fill this name in the locus_alias.



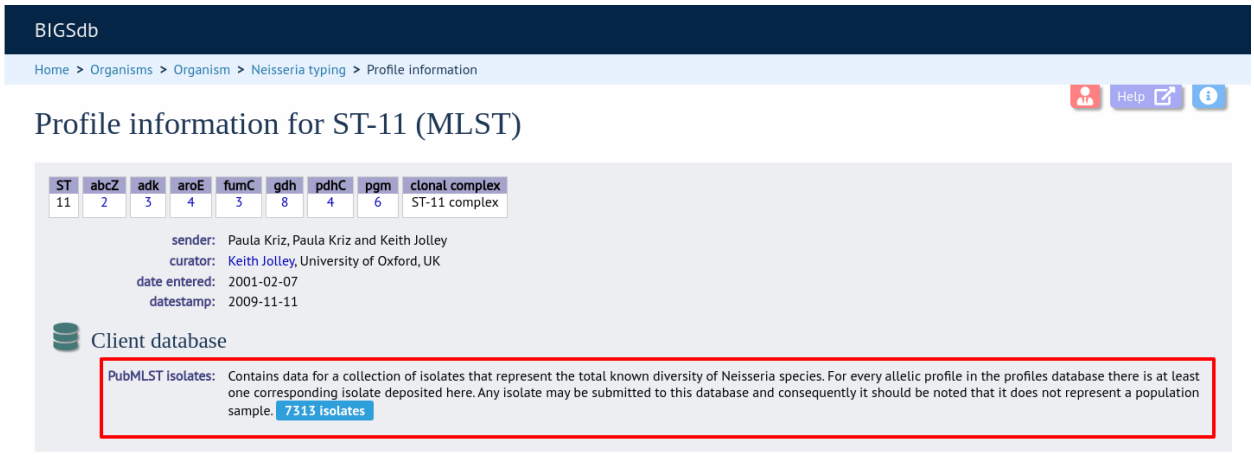
Now when information on a given allele is shown following a query, the software will list the number of isolates with that allele and link to a search on the database to retrieve these.



5.19.2 Look up isolates with a given scheme primary key

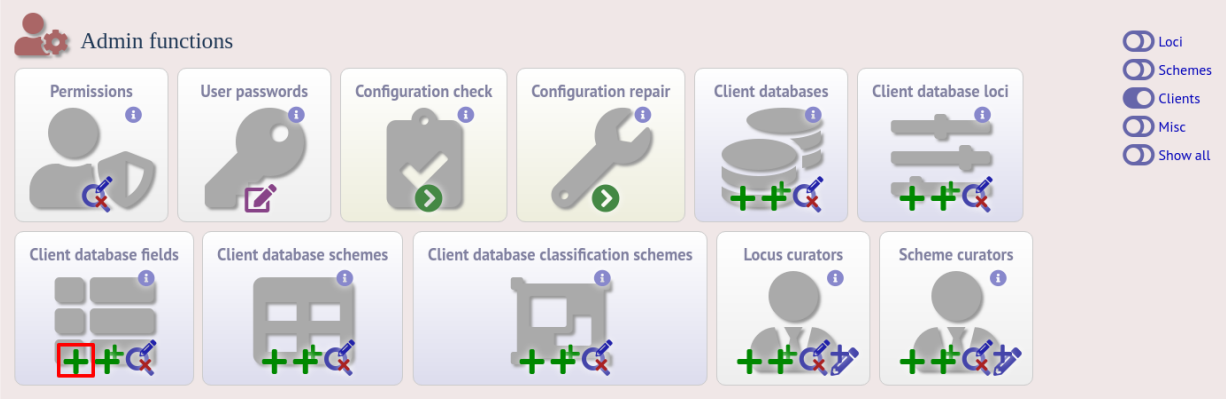
Setting this up is identical to setting up for alleles (see above) except you click on the add (+) client database schemes link and choose the scheme and client databases in the dropdown list boxes.

Now when information on a given scheme profile (e.g. MLST sequence type) is shown following a query, the software will list the number of isolates with that profile and link to a search on the database to retrieve these.

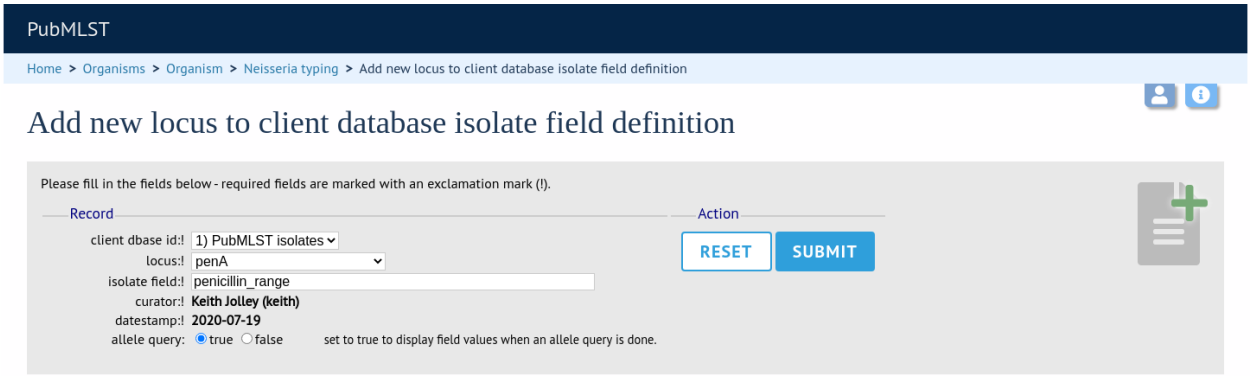


5.19.3 Look up specific isolate database fields linked to a given allele

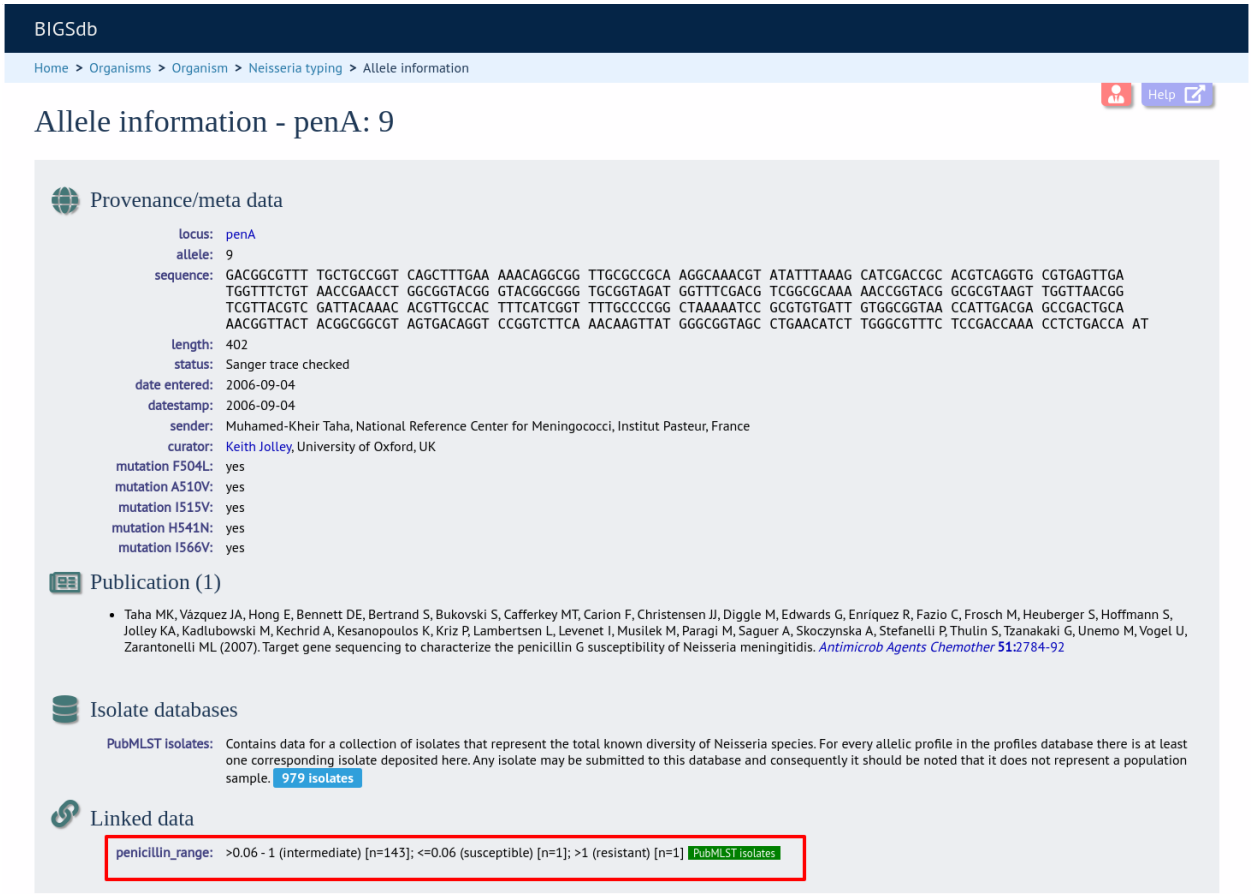
To link an allele to an isolate field, click the add (+) 'client database fields linked to loci' link on the curator's interface contents page.



Select the client database and locus from the dropdown lists and enter the isolate database field that you'd like to link. The 'allele_query' field should be set to true.



Now, in the allele record or following a sequence query that identifies an allele, all values for the chosen field from isolates with the corresponding allele are shown.



5.20 Workflow for setting up a MLST scheme

The workflow for setting up a MLST scheme is as follows (the example seqdef database is called seqdef_db):

Seqdef database

1. Create appropriate loci
2. Create new scheme 'MLST'
3. Add scheme_field 'ST' with primary_key=TRUE (add clonal_complex if you want; set this with primary_key=FALSE)
4. Add each locus as a scheme_member
5. You'll then be able to add profiles

Isolate database

1. Create the same loci with the following additional parameters (example locus 'atpD')

- dbase_name: seqdef_db
- dbase_id: atpD
- url: something like /cgi-bin/bigsdb/bigsdb.pl?db=seqdef_db&page=alleleInfo&locus=atpD&allele_id=[?]

2. Create scheme 'MLST' with:

- dbase_name: seqdef_db
- dbase_id: 1 (or whatever the id of your seqdef scheme is)

3. Add scheme_field ST as before
4. Add loci as scheme_members

5.21 Automated assignment of scheme profiles

It is not practical to define cgMLST profiles via the web interface. A script is provided in the scripts/automation directory of the BIGSdb package called define_profiles.pl that can be used to scan an isolate database and automatically define cgMLST profiles in the corresponding sequence definition database.

The script is run as follows:

```
define_profiles.pl --database <name> --scheme <scheme_id>
```

A full list of options can be found by typing:

```
define_profiles.pl --help
NAME
    define_profiles.pl - Define scheme profiles found in isolate database

SYNOPSIS
    define_profiles.pl --database NAME --scheme SCHEME_ID [options]

OPTIONS

--cache
    Update scheme field cache in isolate database.

--database NAME
    Database configuration name.

--help
    This help page.

--exclude_isolates LIST
    Comma-separated list of isolate ids to ignore.

--exclude_projects LIST
    Comma-separated list of projects whose isolates will be excluded.

--ignore_multiple_hits
    Set allele designation to 'N' if there are multiple designations set for
    a locus. The default is to use the lowest allele value in the profile
    definition.

--isolates LIST
    Comma-separated list of isolate ids to scan (ignored if -p used).

--isolate_list_file FILE
    File containing list of isolate ids (ignored if -i or -p used).
```

(continues on next page)

(continued from previous page)

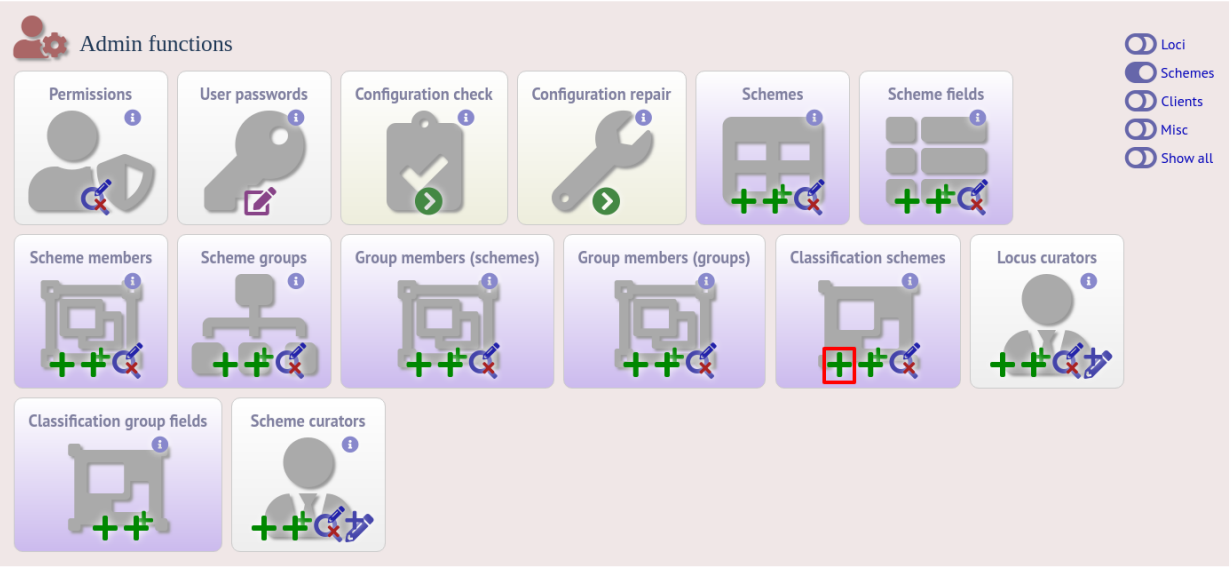
```
--match_missing
    Treat missing loci as specific alleles rather than 'any'. This will
    allow profiles for every isolate that has <= threshold of missing alleles
    to be defined but may result in some isolates having >1 ST.

--max ID
    Maximum isolate id.

--min ID
    Minimum isolate id.

--min_size SIZE
    Minimum size of seqbin (bp) - limit search to isolates with at least this
    much sequence.

--missing NUMBER
    Set the number of loci that are allowed to be missing in the profile. If
    the remote scheme does not allow missing loci then this number will be set
    to 0.  Default=0.

--projects LIST
    Comma-separated list of project isolates to scan.

--scheme SCHEME_ID
    Scheme id number.

--view VIEW
    Limit isolates searched to specified view.
```

5.22 Scheme profile clustering - setting up classification schemes

Classification groups are a way to cluster scheme profiles using a specified threshold of pairwise allelic mismatches. Any number of different classification schemes can sit on top of a standard scheme (such as cgMLST), allowing different similarity thresholds to be pre-determined. Currently, single-linkage clustering is supported whereby each member of a group must have no more than the specified number of allelic differences with at least one other member of the group.

5.22.1 Defining classification scheme in sequence definition database

Once a scheme has been defined, add a classification scheme by clicking the add classification schemes (+) link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it.



Select the underlying scheme and enter a name for the classification scheme, the number of mismatches allowed in order to include a scheme profile in a group, and a description. An example name for such a scheme could be 'Nm_cgc_25' indicating that this is a classification scheme for *Neisseria meningitidis* core genome cluster with a threshold of 25 mismatches.

You can additionally choose whether a relative threshold is used to calculate the number of mismatches to account for missing loci in pairwise comparisons. In this case, in order to be grouped, the number of matching alleles must exceed:

```
(number of common loci x (total loci - defined threshold)) / total loci
```

rather than

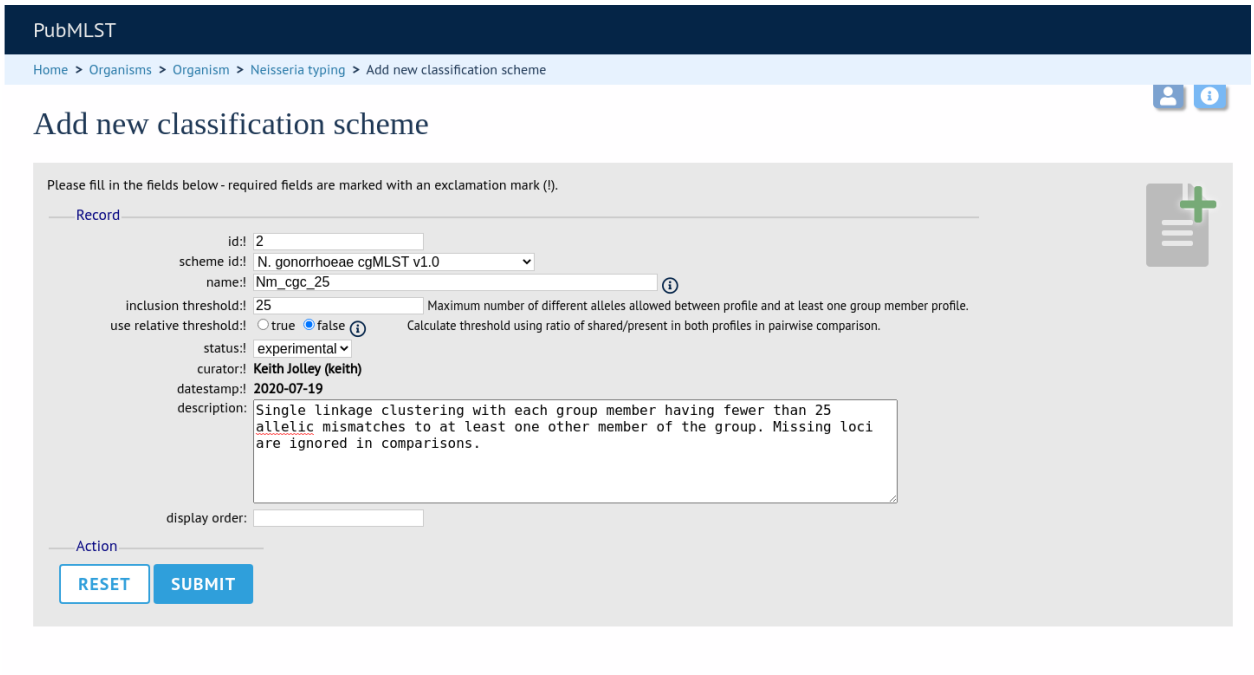
```
total loci - defined threshold
```

when an absolute threshold is used.

As this threshold has to be calculated for each pairwise comparison, clustering using relative thresholds is slower than using an absolute value, and probably makes little real world difference.

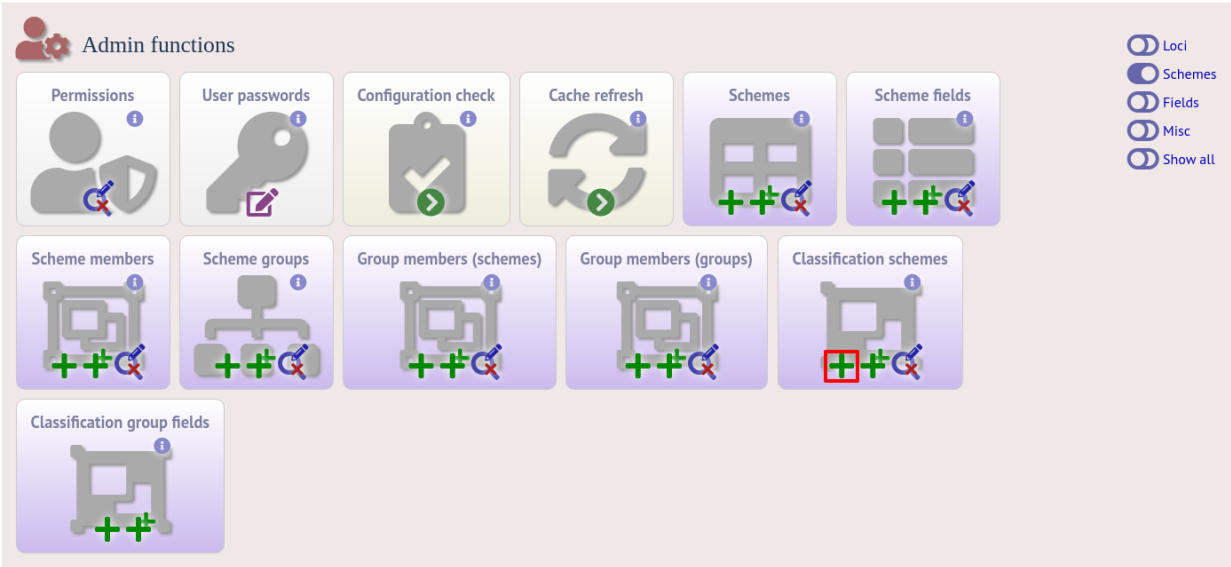
The status can be 'experimental' or 'stable'. The status of a scheme will be shown in the web interface to indicate that any groupings are subject to change and do not form part of the stable nomenclature.

Press 'Submit' to create the classification scheme.



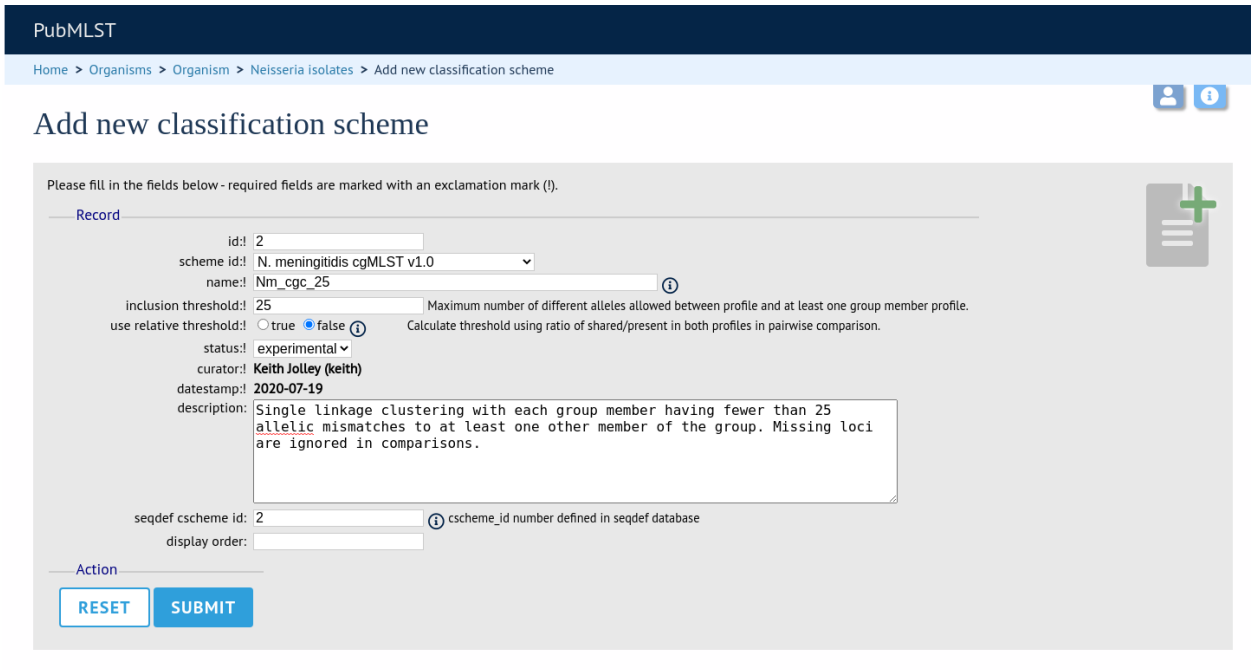
5.22.2 Defining classification scheme in isolate database

Duplicate the scheme definition from the sequence definition database. Click the add classification schemes (+) link on the curator's interface contents page. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it.



Enter the same details used in the sequence definition database. If a different id number is used in the isolate and sequence definition databases, you can set the seqdef id in the seqdef_cscheme_id field (the default is to use the same id).

You can also define a display order - this is an integer field on which the ordering of classification schemes is sorted when displayed in the isolate information page.



It is a good idea to *check the configuration*.

5.22.3 Clustering

Clustering is performed using the cluster.pl script found in the scripts/automation directory of the BIGSdb package. It should be run by the bigsdb user account (or any account with access to the databases).

Currently only single-linkage clustering is supported.

The script is run as follows from the command line:

```
cluster.pl --database <database configuration> --cscheme <classification scheme id>
```

A full list of options can be found by typing:

```
cluster.pl --help
NAME
    cluster.pl - Cluster cgMLST profiles using classification groups.

SYNOPSIS
    cluster.pl --database NAME --cscheme_id SCHEME_ID [options]

OPTIONS

--cscheme CLASSIFICATION_SCHEME_ID
    Classification scheme id number.

--database NAME
    Database configuration name.

--help
    This help page.
```

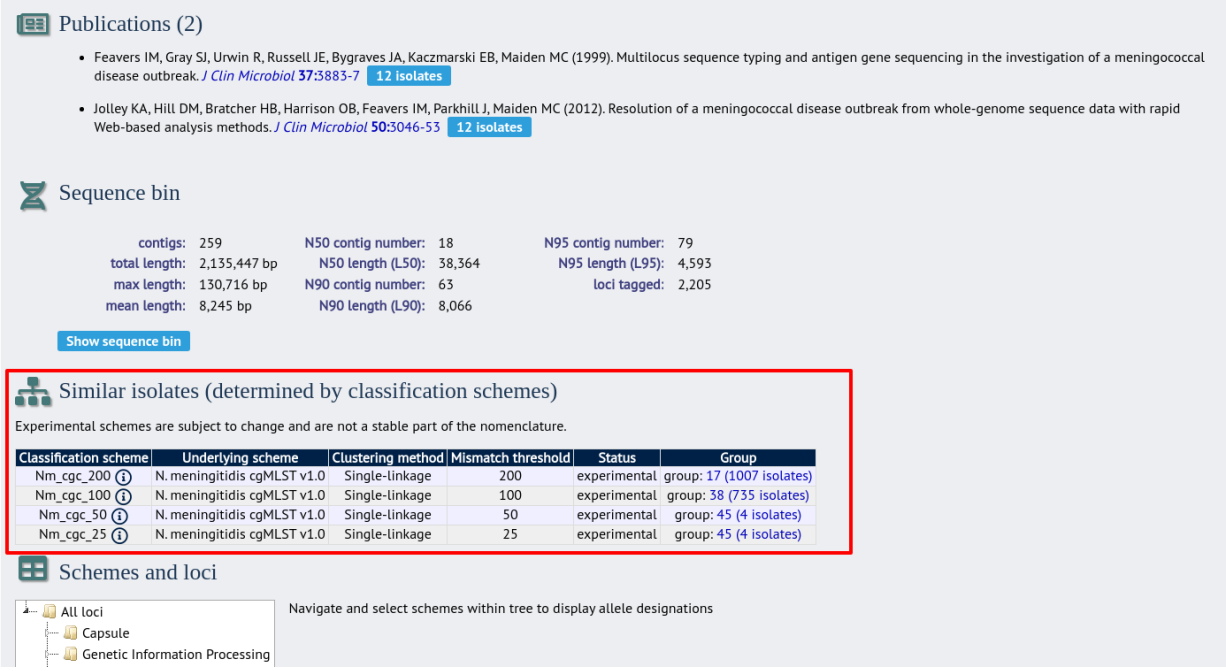
(continues on next page)

(continued from previous page)

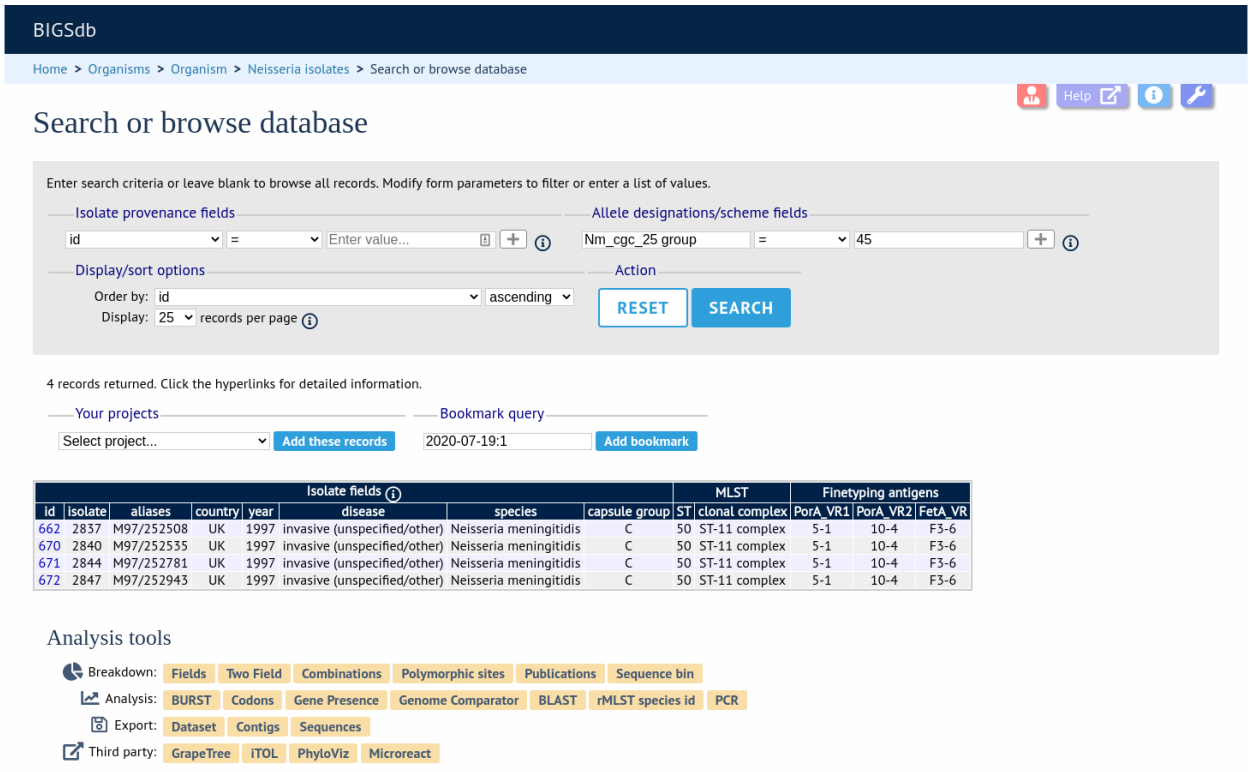
```
--reset
    Remove all groups and profiles currently defined for classification group.
```

Note: Note that for classification schemes to be accessible within the isolate database, *scheme cache tables* must be generated and kept up-to-date.

Where an isolate has been clustered in to a group with other isolates, this information is available in the *isolate information page*.



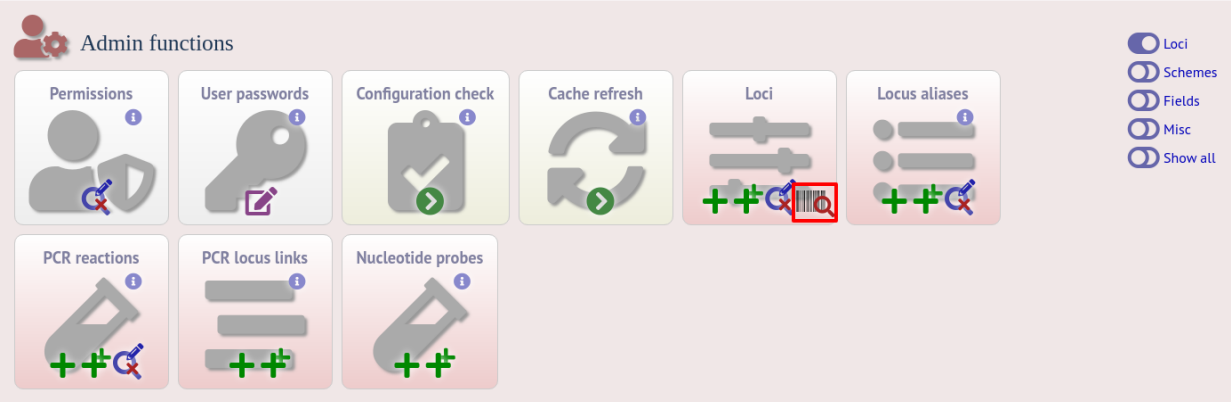
Clicking the hyperlinks will take you to a table containing matching isolates, from where standard analyses can be performed.



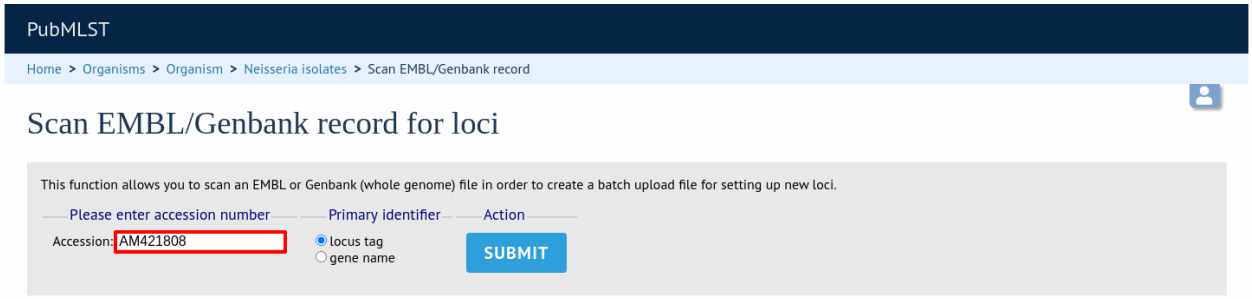
5.23 Defining new loci based on annotated reference genome

An annotated reference genome can be used as the basis of defining loci. The 'Databank scan' function will create an upload table suitable for pasting directly in to the batch locus add form of the *sequence definition* or *isolate* databases.

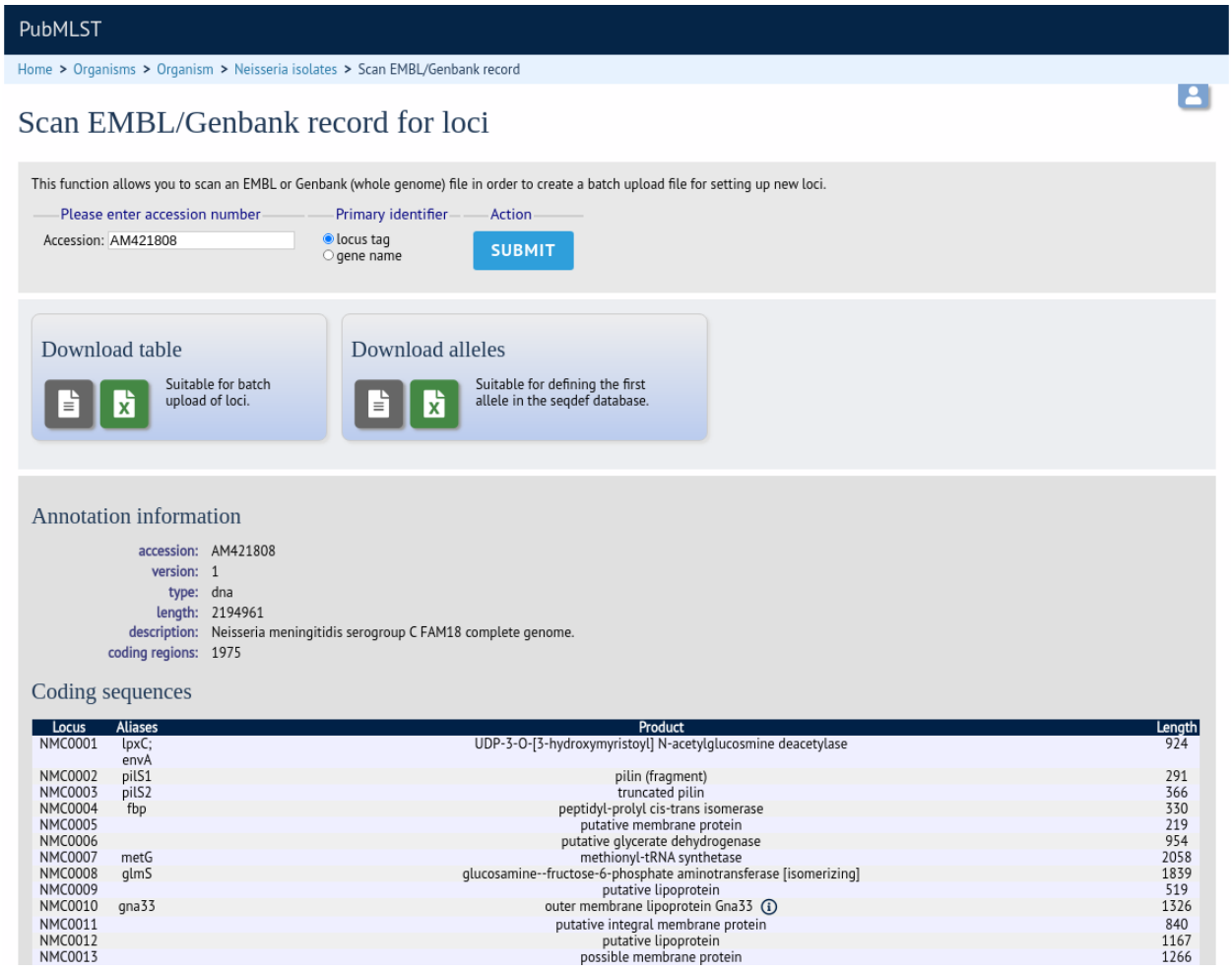
Click 'Database scan' within the 'Loci' group on the curator's contents page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



Enter an EMBL or Genbank accession number for a complete annotated genome and press 'Submit'.



A table of loci will be generated provided a valid accession number is provided.

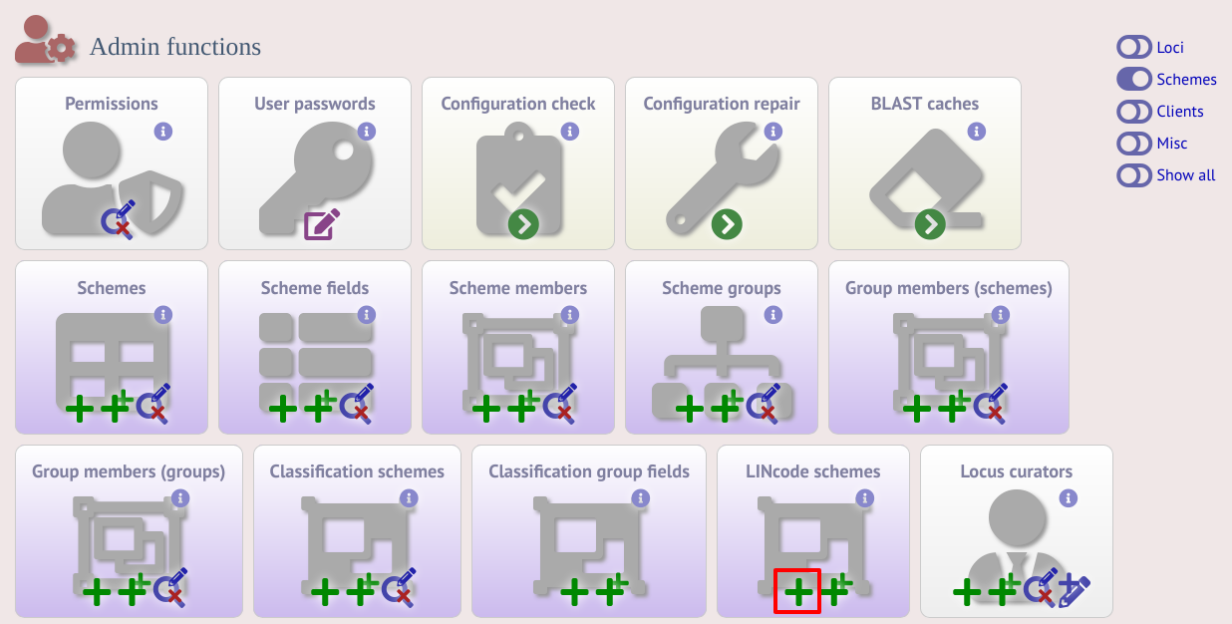


Tab-delimited text and Excel format files will be created to be used as the basis for upload files for the sequence definition and isolate databases. Batch sequence files, in text and Excel formats, are also created for defining the first allele once the locus has been set up in the sequence definition database.

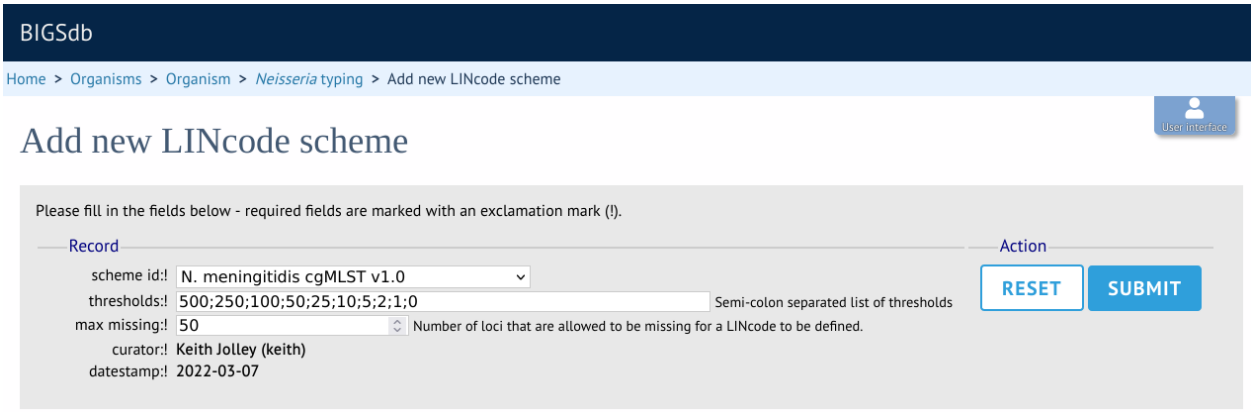
5.24 Setting up LINcode definitions for cgMLST schemes

Note: The idea behind LINcodes is described in Hennart et al. 2021 bioRxiv 2021.07.26.453808.

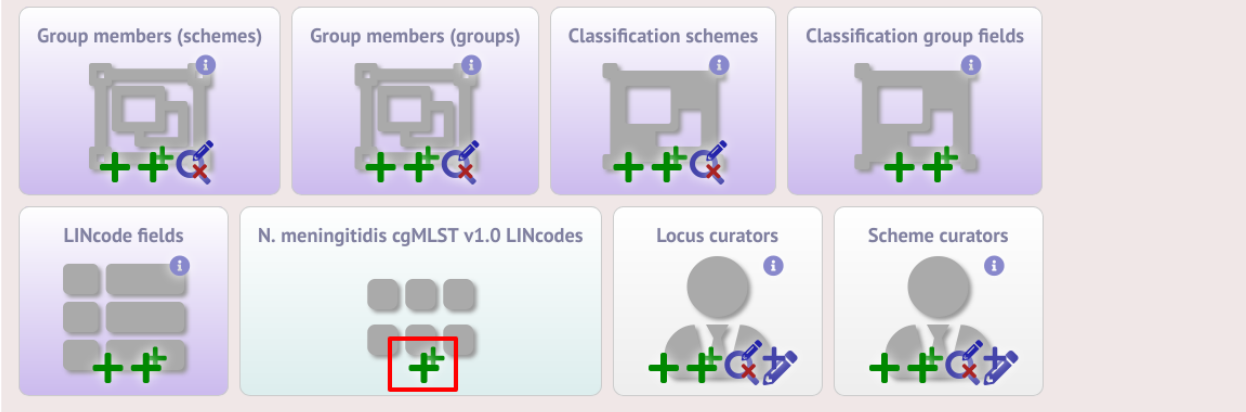
LINcode schemes can be defined by administrators by clicking the add LINcode scheme button within the 'Schemes' group on the curator index page of both the sequence definition and isolate databases. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it. An indexed scheme, e.g. MLST or cgMLST whereby a field defines each unique combination of alleles, needs to be defined before the link will be enabled.



Select the indexed scheme from the dropdown list and enter your locus thresholds in descending order as a semi-colon separated list, e.g. 500;250;100;50;25;10;5;2;1;0. Also enter the number of missing alleles that are allowed within a profile for a LINcode to be assigned. Click 'Submit'.



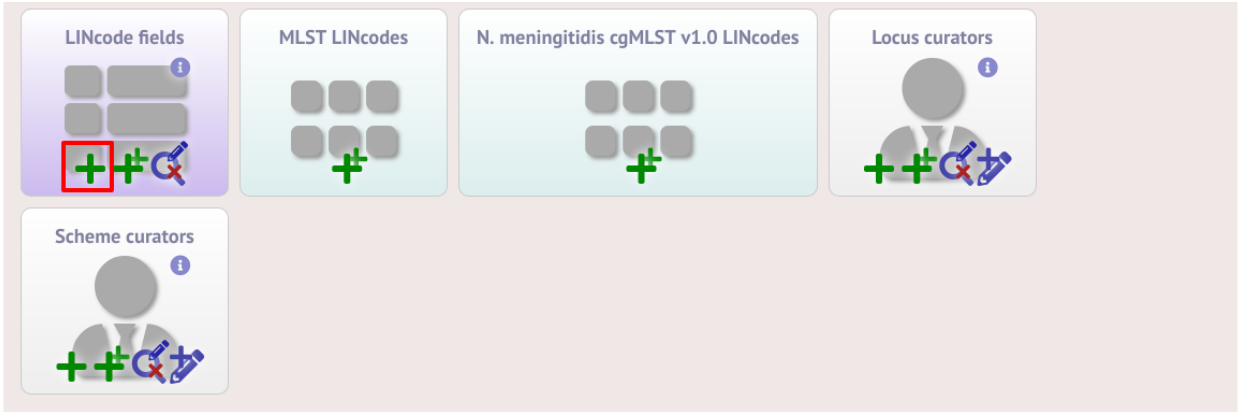
If LINcodes have been previously defined, the existing assignments can be uploaded by clicking the batch add LINcodes button (sequence definition database). This allows you to copy and paste assignments from a spreadsheet template. The template consists of the index field (e.g. cgST), and separate columns for each threshold level.



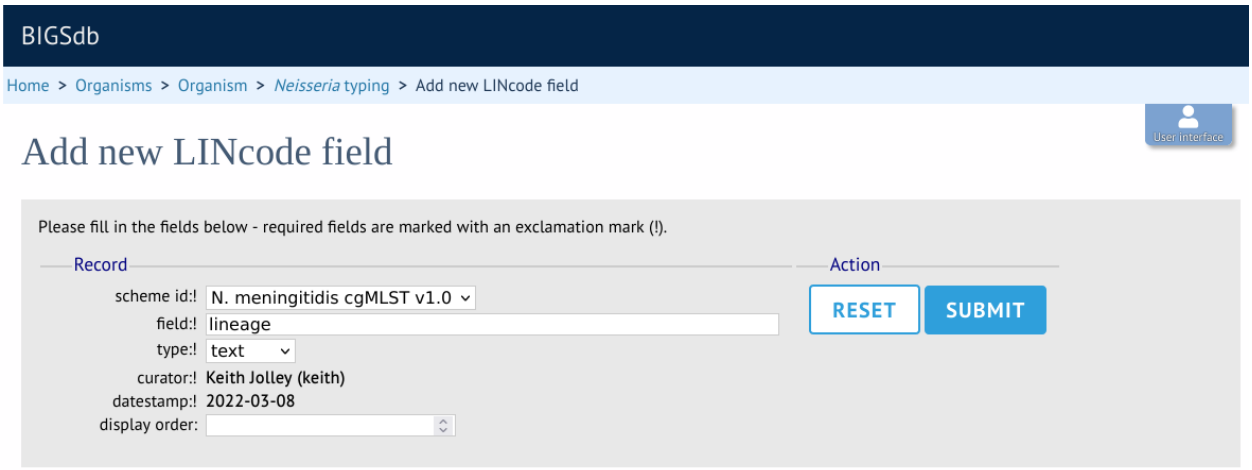
LINcodes can then be assigned automatically using the lincodes.pl script found within the scripts/maintenance directory. A full list of arguments can be found by typing:

```
lincodes.pl --help
NAME
    lincodes.pl - Define LINcodes from cgMLST profiles.

SYNOPSIS
    lincodes.pl --database DB_CONFIG --scheme SCHEME_ID [options]

OPTIONS

--batch_size NUMBER
    Sets a maximum number of profiles to use to initiate assignment order.
    The order of assignment is optimally determined using Prim's algorithm,
    but can take a long time if there are thousands of profiles. Up to the
    number of profiles set here will be ordered and assigned first before
    further batches are ordered and assigned. The default value is 10,000
    but it is recommended that you allow ordering to be determined from all
    defined profiles if LINcodes have not been previously determined, i.e.
    set this value to greater than the number of assigned profiles.

--database DATABASE CONFIG
    Database configuration name. This must be a sequence definition database.

--missing NUMBER
    Set the maximum number of loci that are allowed to be missing in a profile
    for LINcodes to be assigned. If not set, the value defined in the LINcode
    schemes table will be used.

--mmap
    Write distance matrix to disk rather than memory. Use this if calculating a
    very large distance matrix on a machine with limited memory. This may run
    slower.

--quiet
    Only output errors.

--scheme SCHEME ID
    Scheme id number for which a LINcode scheme has been defined.
```

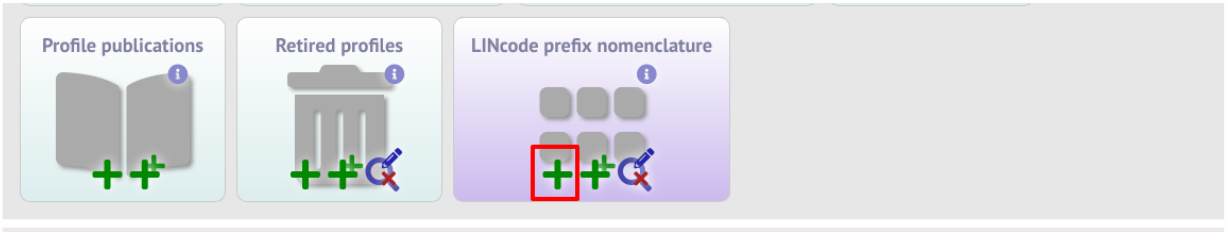
It is also possible to provide nomenclature for specific LINcode prefixes. These can be used to define lineages or sublineages. LINcode fields are first defined by clicking the add lincode field button.



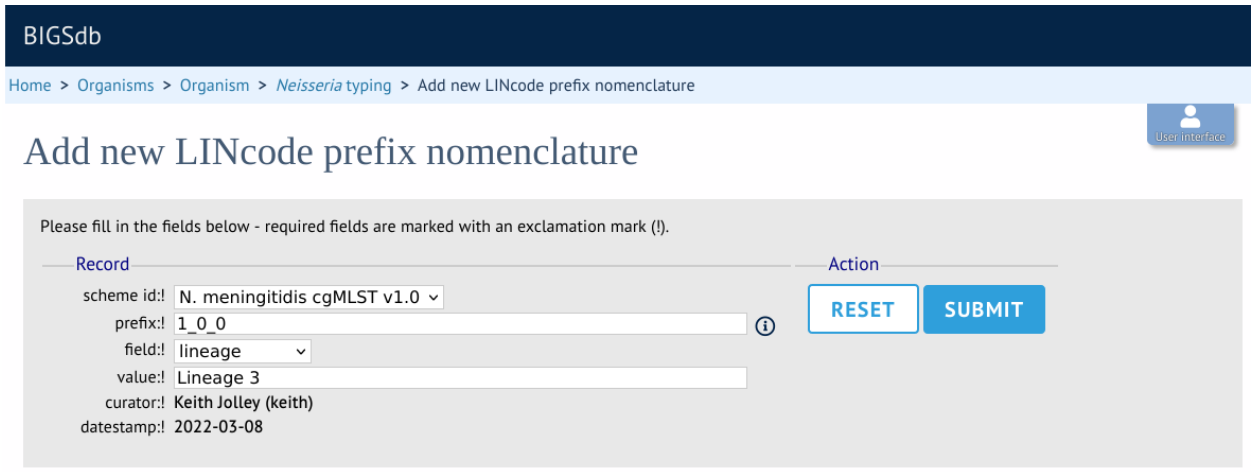
Enter the name of the field you wish to define, e.g. 'lineage' and the data type for any values (either text or integer).



A new menu item called 'LINcode prefix nomenclature' will appear in the curator part of the index page. It is hidden by default so you may need to click the 'Show all' button. Click the add button.



Enter a LINcode prefix (left-hand part of LINcode) and assign a name to it. Make sure you do the same in both the sequence definition and isolate databaase. You will then be able to search isolates based on these field values and they will appear in isolate records.



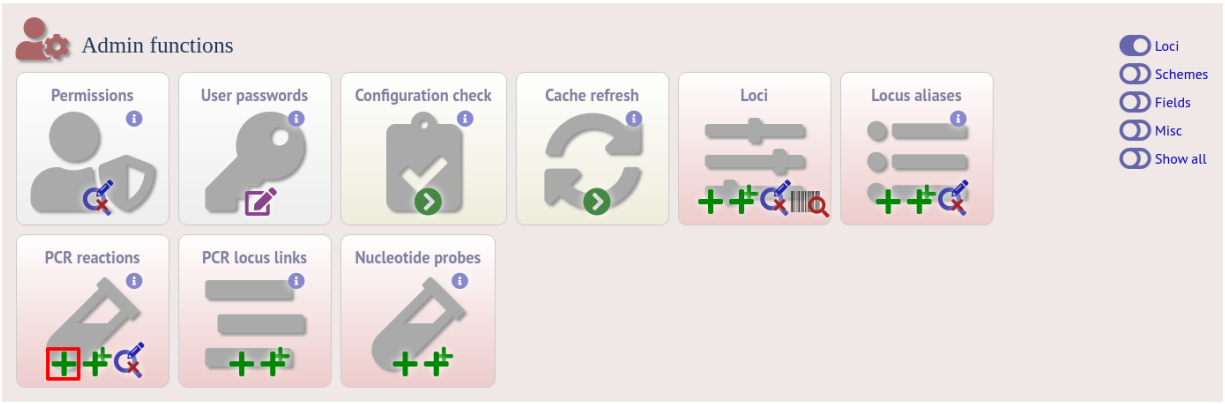
5.25 Genome filtering

Within a genome there may be multiple loci that share allele pools. If an allele sequence is tagged from a genome using only BLAST then there is no way to determine which locus has been identified. It is, however, possible to further define loci by their context, i.e. surrounding sequence.

5.25.1 Filtering by *in silico* PCR

Provided a locus can be predicted to be specifically amplifed by a PCR reaction, the genome can be filtered to only look at regions predicted to fall within amplification products of one or more PCR reactions. Since this is *in silico* we don't need to worry about problems such as sequence secondary structure and primers can be any length.

To define a PCR reaction that can be linked to a locus definition, click the add (+) PCR reaction link on the curator's main page. This function is normally hidden, so you may need to click the 'Loci' toggle to display it.



In the resulting web form you can enter values for your two primer sequences (which can be any length), the minimum and maximum lengths of reaction products you wish to consider and a value for the allowed number of mismatches per primer.

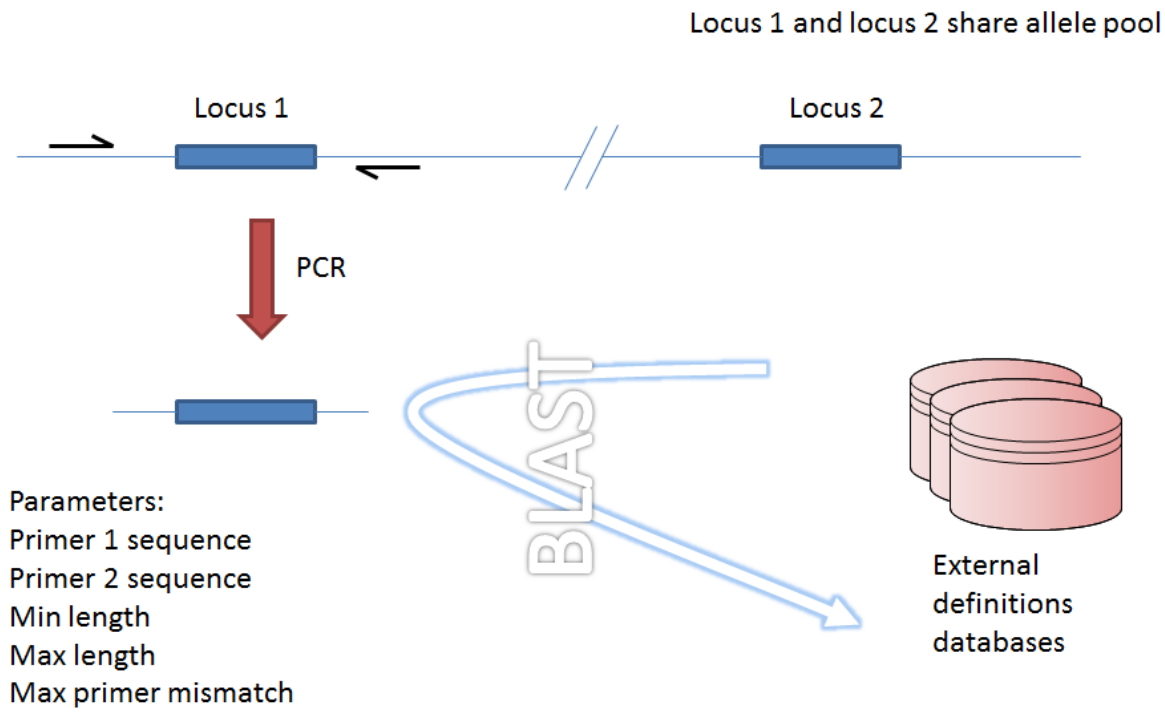
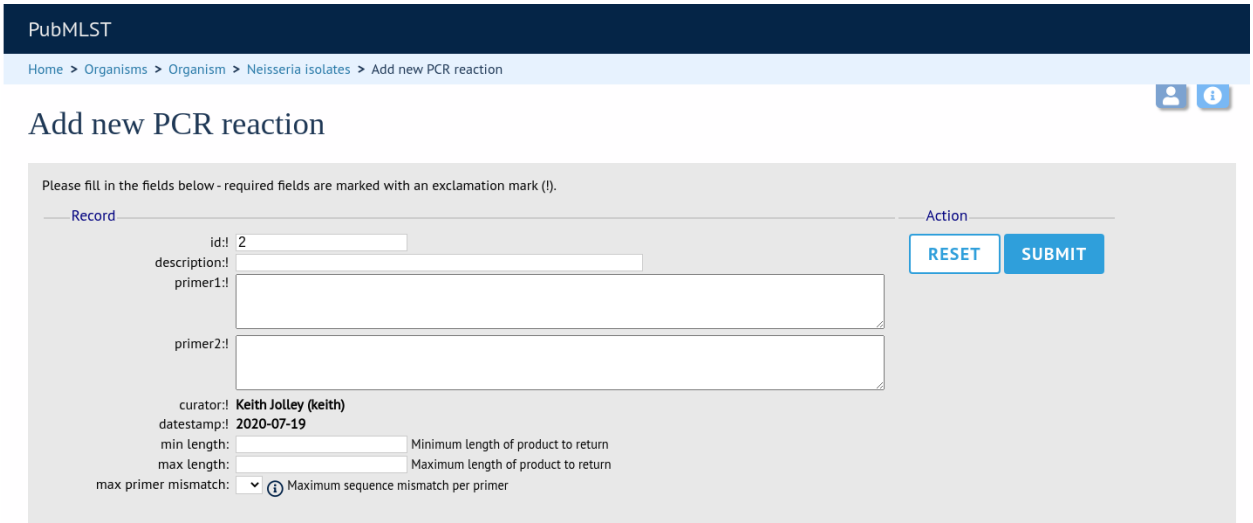


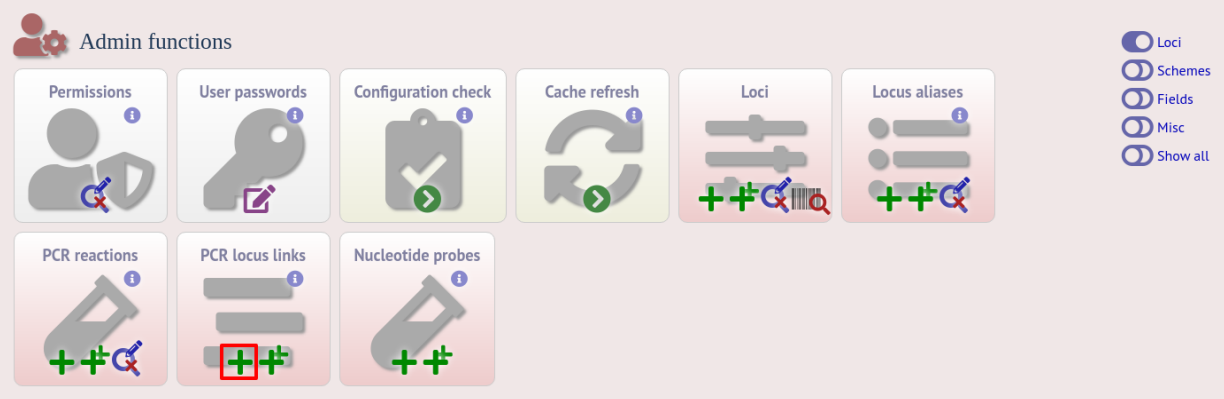
Fig. 1: Genome filtering by *in silico* PCR.



- id - PCR reaction identifier number.
  - Allowed: integer.
- description - Description of PCR reaction product.
  - Allowed: any text.
- primer1 - Primer 1 sequences
  - Allowed: nucleotide sequence (IUPAC ambiguous characters allowed).

- primer2 - Primer 2 sequence.
  - Allowed: nucleotide sequence (IUPAC ambiguous characters allowed).
- min_length - Minimum length of predicted PCR product.
  - Allowed: integer.
- max_length - Maximum length of predicted PCR product.
- max_primer_mismatch - Number of mismatches allowed in primer sequence.
  - Allowed: integer.
  - Do not set this too high or the simulation will run slowly.

Associating this with a particular locus is a two step process. First, create a locus link by clicking the add (+) PCR locus link on the curator's main page. This link will only appear once a PCR reaction has been defined.



Select the locus and PCR reaction name from the dropdown lists to create the link. You also need to edit the locus table and set the pcr_filter field to 'true'.

Now when you next perform *tag scanning* there will be an option to use PCR filtering.

5.25.2 Filtering by *in silico* hybridization

An alternative is to define a locus by proximity to a single probe sequence. This is especially useful if you have multiple contigs and the locus in question may be at the end of a contig so that it doesn't have upstream or downstream sequence available for PCR filtering.

The process is very similar to setting up PCR filtering, but this time click the nucleotide probe link on the curator's content page.

Enter the nucleotide sequence and a name for the probe. Next you need to link this to the locus in question. Click the add (+) probe locus links link on the curator's main page. This link will only appear once a probe has been defined.

Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:

- probe_id - Dropdown list of probe names.
  - Allowed: selection from list.
- locus - Dropdown list of loci.
  - Allowed: selection from list.
- max_distance - Minimum distance of probe from end of locus.
  - Allowed: any positive integer.

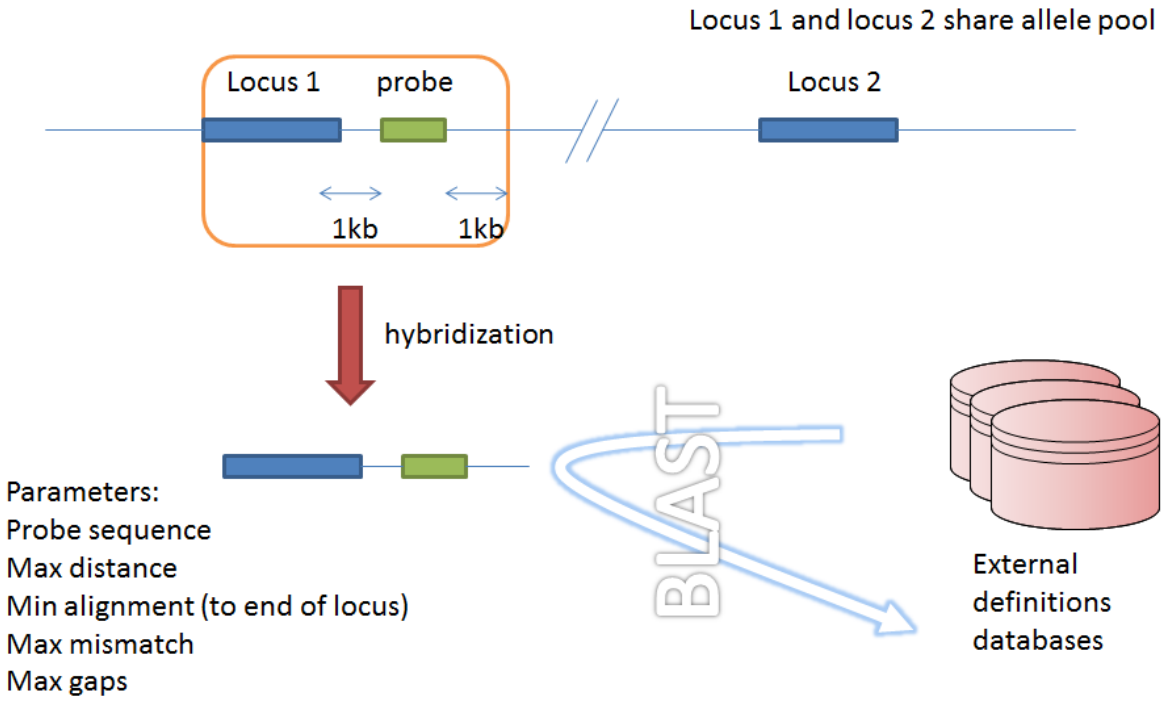
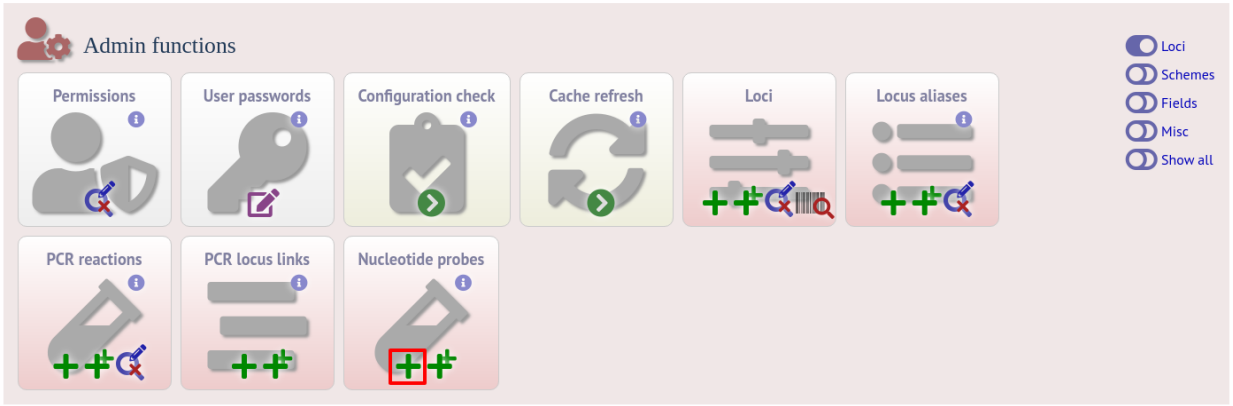
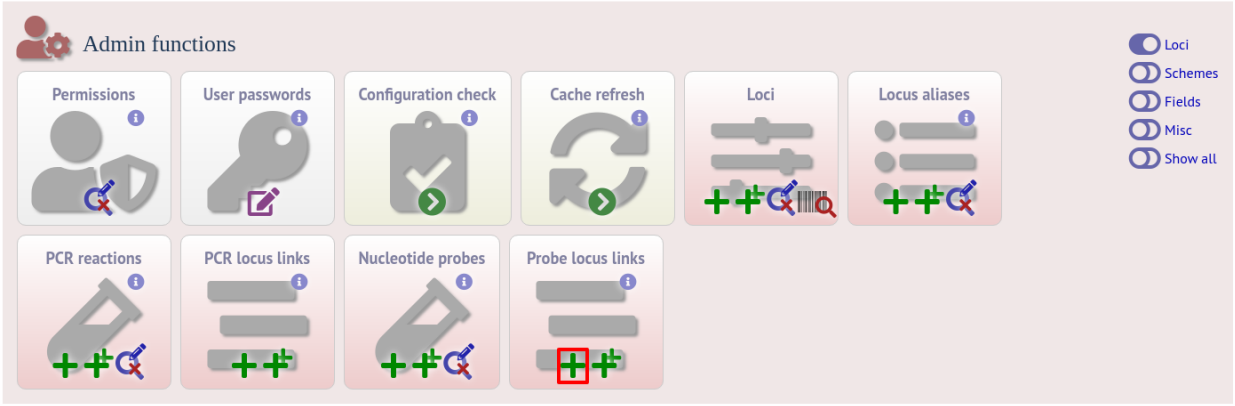


Fig. 2: Filtering by *in silico* hybridization





- min_alignment - Minimum length of alignment allowed.
  - Allowed: any positive integer.
- max_mismatch - Maximum number of mismatches allowed in alignment.
  - Allowed: any positive integer.
- max_gaps - Maximum number of gaps allowed in alignment.
  - Allowed: any positive integer.

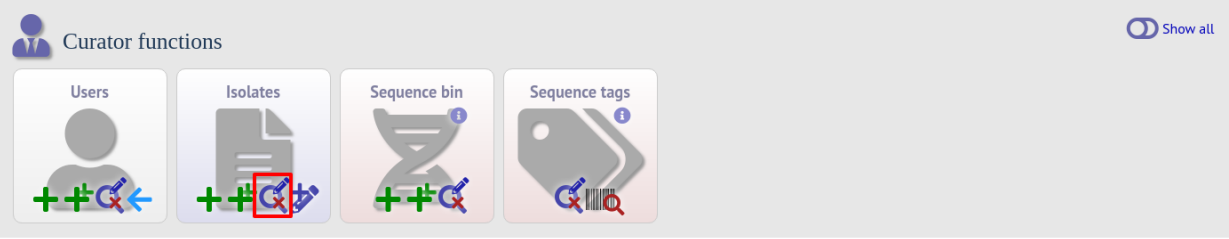
Finally edit the locus table and set the probe_filter field for the specified locus to 'true'.

Now when you next perform *tag scanning* there will be an option to use probe hybridization filtering.

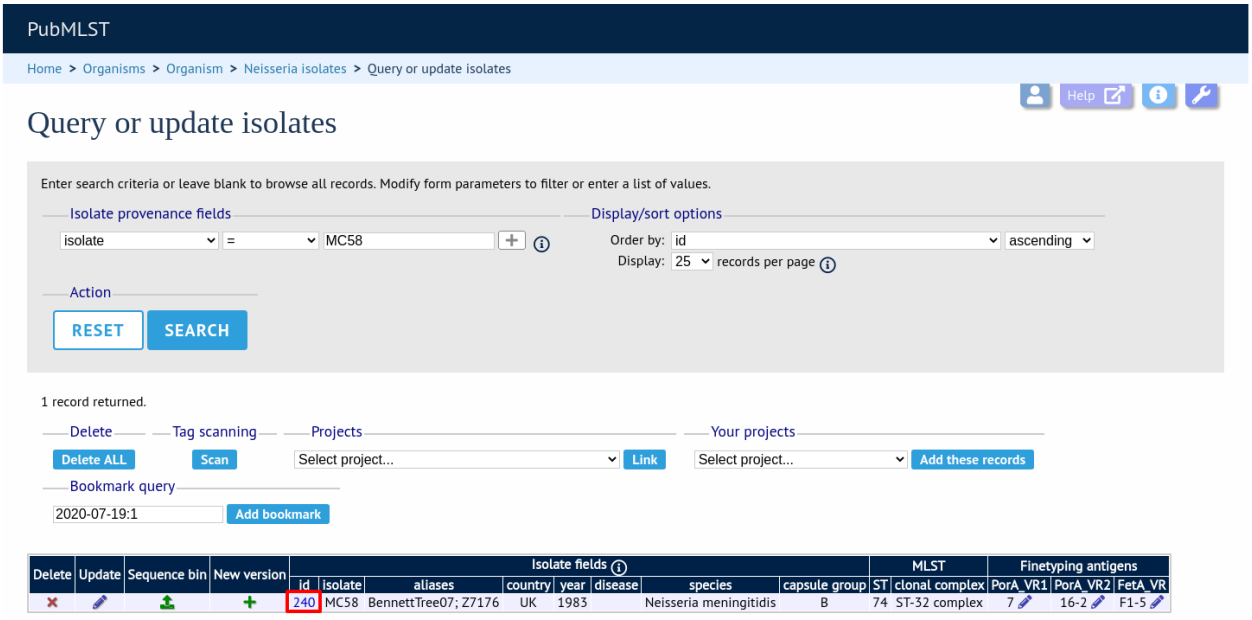
5.26 Setting locus genome positions

The genome position for a locus can be set directly by editing the locus record. To batch update multiple loci based on a tagged genome, however, a much easier way is possible. For this method to work, the reference genome must be represented by a single contig.

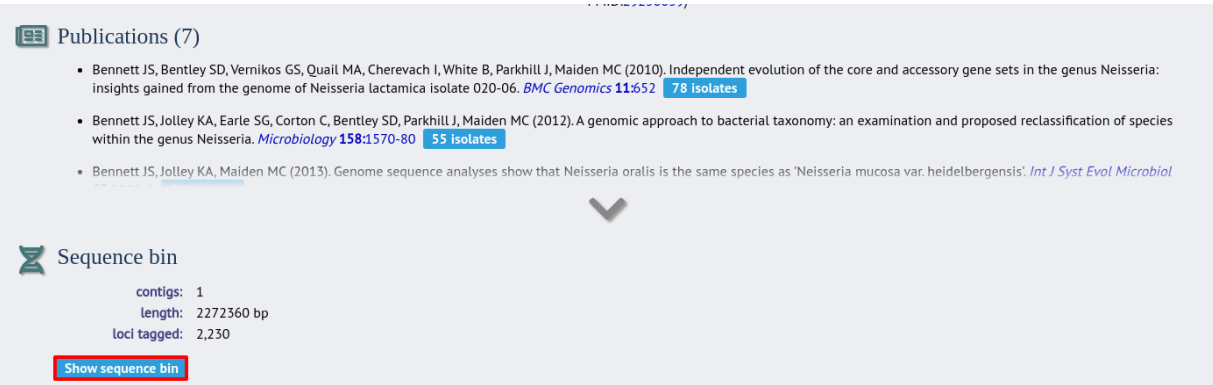
From the curator's main page, you need to do a query to find the isolate that you will base your numbering on. Click 'isolate query' to take you to a standard query form.



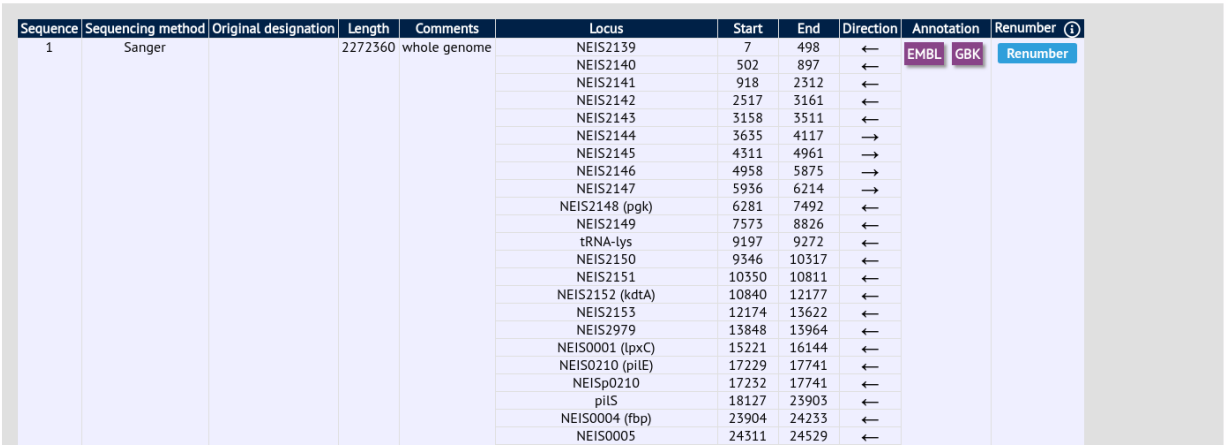
Perform your search and click the hyperlinked id number of the record.



In the isolate record, click the 'Show sequence bin' button to bring up details of the isolate contigs.

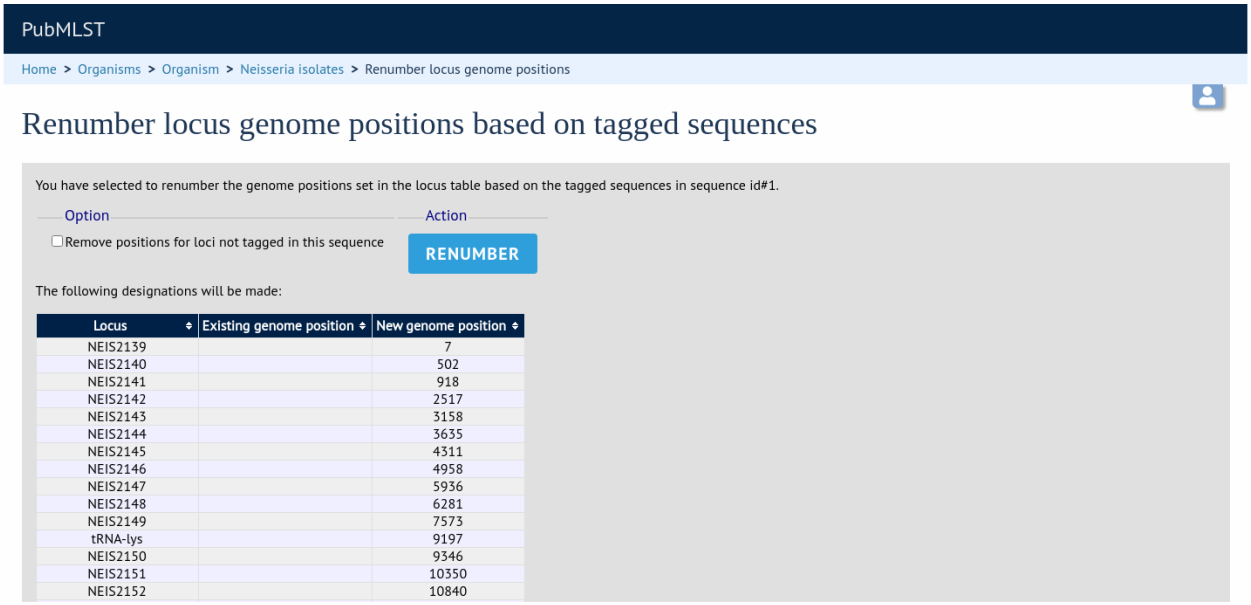


Click the 'Renumber' button:



Sequence	Sequencing method	Original designation	Length	Comments	Locus	Start	End	Direction	Annotation	Renumber
1	Sanger		2272360	whole genome	NEIS2139	7	498	←	EMBL GBK	Renumber
					NEIS2140	502	897	←		
					NEIS2141	918	2312	←		
					NEIS2142	2517	3161	←		
					NEIS2143	3158	3511	←		
					NEIS2144	3635	4117	→		
					NEIS2145	4311	4961	→		
					NEIS2146	4958	5875	→		
					NEIS2147	5936	6214	→		
					NEIS2148 (pgk)	6281	7492	←		
					NEIS2149	7573	8826	←		
					tRNA-lys	9197	9272	←		
					NEIS2150	9346	10317	←		
					NEIS2151	10350	10811	←		
					NEIS2152 (kdtA)	10840	12177	←		
					NEIS2153	12174	13622	←		
					NEIS2979	13848	13964	←		
					NEIS0001 (lpxC)	15221	16144	←		
					NEIS0210 (pilE)	17229	17741	←		
					NEISp0210	17232	17741	←		
					pilS	18127	23903	←		
					NEIS0004 (fbp)	23904	24233	←		
					NEIS0005	24311	24529	←		

A final confirmation screen is displayed with the option to remove existing numbering that doesn't appear within the reference genome. Click 'Renumber'.



Locus	Existing genome position	New genome position
NEIS2139		7
NEIS2140		502
NEIS2141		918
NEIS2142		2517
NEIS2143		3158
NEIS2144		3635
NEIS2145		4311
NEIS2146		4958
NEIS2147		5936
NEIS2148		6281
NEIS2149		7573
tRNA-lys		9197
NEIS2150		9346
NEIS2151		10350
NEIS2152		10840

5.27 Defining composite fields

Composite fields are virtual fields that don't themselves exist within the database but are made up of values retrieved from other fields or schemes and formatted in a particular way. They are used for display and analysis purposes only and can not be searched against.

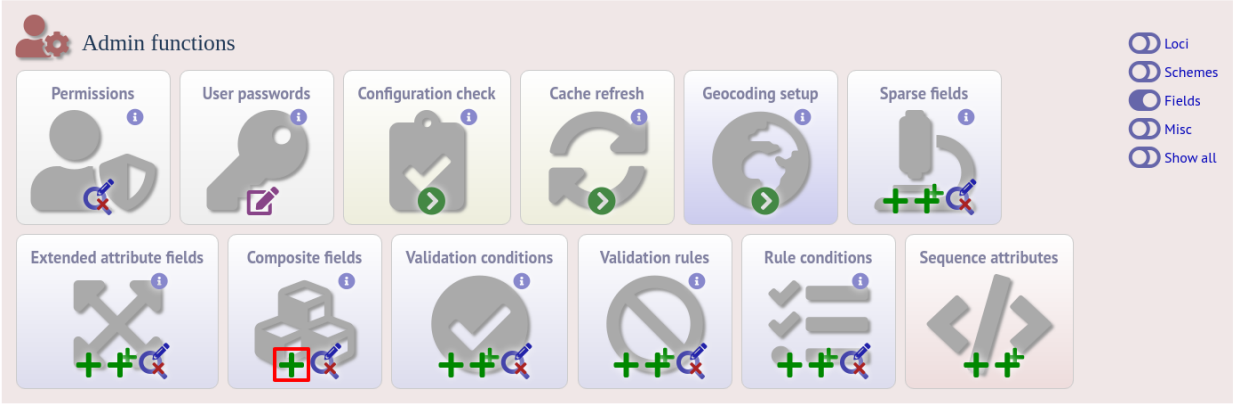
One example of a composite field is used in the Neisseria PubMLST database which has a strain designation composite field made up of serogroup, PorA VR1 and VR2, FetA VR, ST and clonal complex designations in the format:

[serogroup]: P1.[PorA_VR1],[PorA_VR2]: [FetA_VR]: ST-[ST] ([clonal_complex])

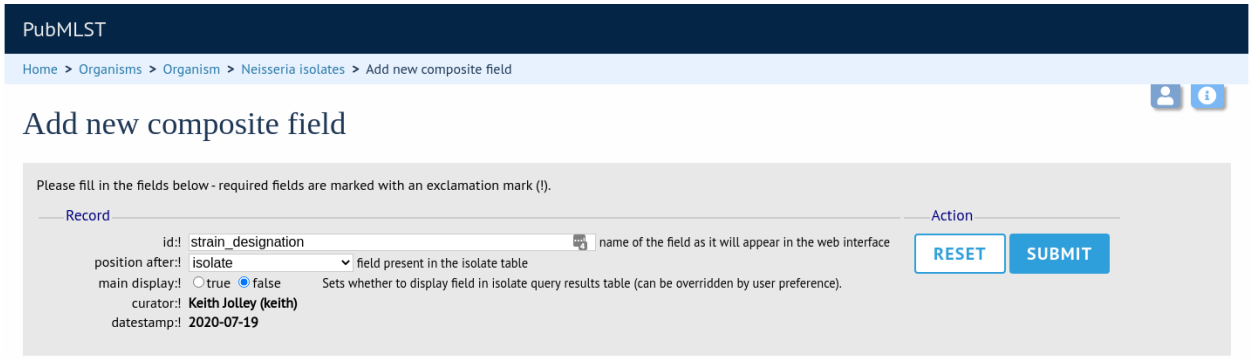
e.g. A: P1.5-2,10: F1-5: ST-4 (cc4)

Additionally, the clonal complex field in the above example is converted using a regular expression from 'ST-4 complex/subgroup IV' to 'cc4'.

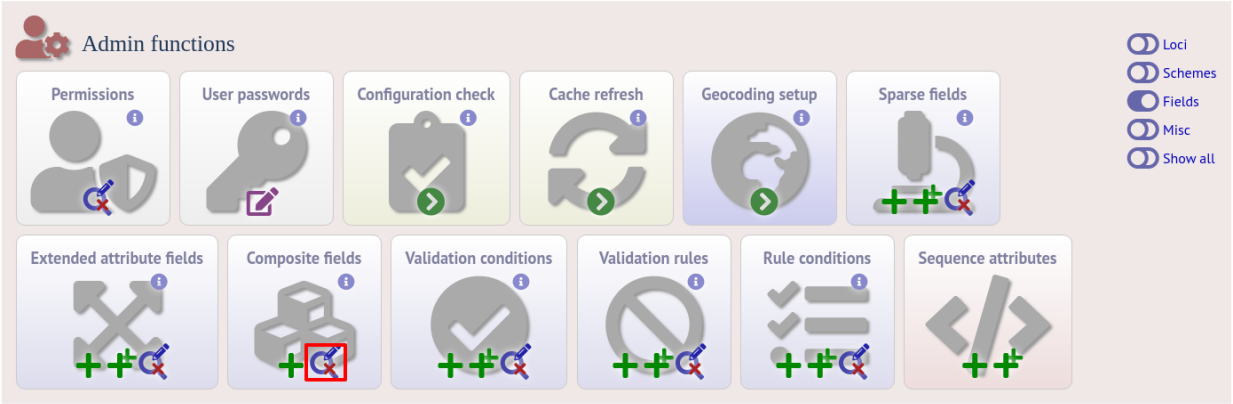
Composite fields can be added to the database by clicking the add (+) composite fields link on the curator's main page. This function is normally hidden, so you may need to click the 'Fields' toggle to display it.



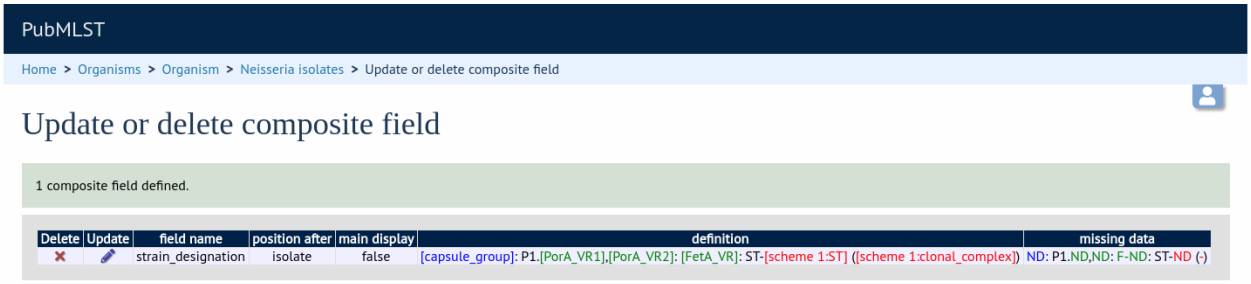
Initially you just enter a name for the composite field and after which field it should be positioned. You can also set whether or not it should be displayed by default in main results tables following a query - this is overrideable by user preferences.



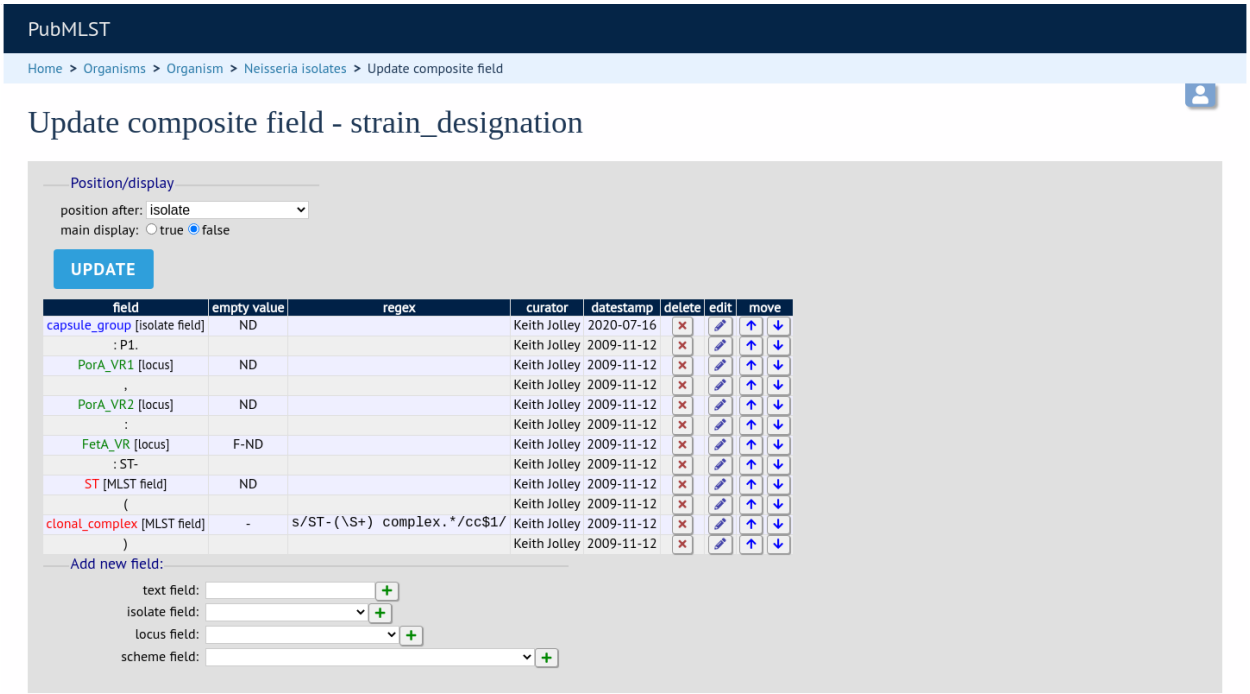
Once the field has been created it needs to be defined. This can be done from query composite field link on the main curator's page.



Select the composite field from the list and click 'Update'.



From this page you can build up your composite field from snippets of text, isolate field, locus and scheme field values. Enter new values in the boxes at the bottom of the page.



Once a field has been added to the composite field, it can be edited by clicking the 'edit' button next to it to add a regular expression to modify its value by specific rules, e.g. in the clonal complex field above, the regular expression is set as:

```
s/ST-(\S+) complex.*/cc$1/
```

which extracts one or more non-space characters following the 'ST-' in a string that then contains the work 'complex', and appends this to 'cc' to produce the final string.

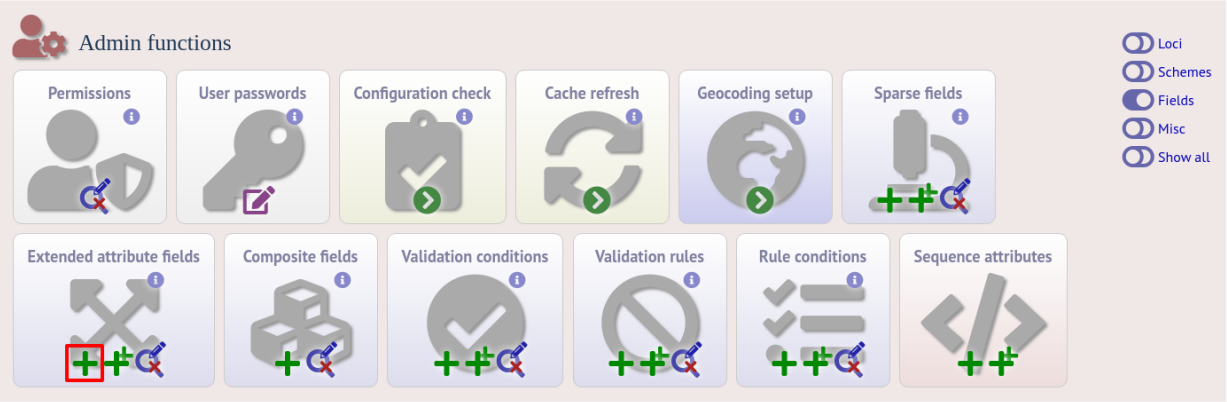
This will convert 'ST-4 complex/subgroup IV' to 'cc4'.

You can also define text to be used for when the field value is missing, e.g. 'ND'.

5.28 Extended provenance attributes (lookup tables)

Lookup tables can be associated with an isolate database field such that the database can be queried by extended attributes. An example of this is the relationship between continent and country - every country belongs to a continent but you wouldn't want to store the continent with each isolate record (not only could data be entered inconsistently but it's redundant). Instead, each record may have a country field and the continent is then determined from the lookup table, allowing, for example, a search of isolates limited to those from Europe.

To set up such an extended attribute, click the add (+) isolate field extended attributes link on the curator's main page. This function is normally hidden, so you may need to click the 'Fields' toggle to display it.



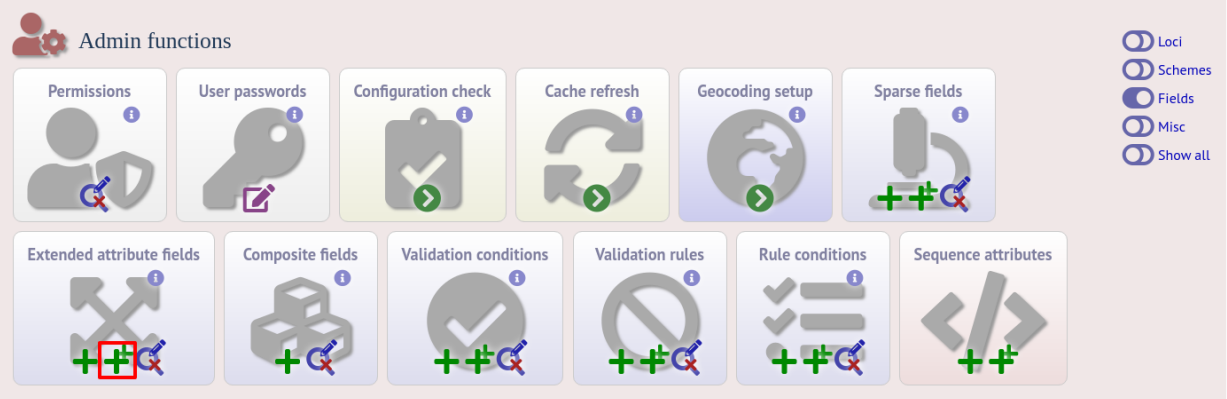
Fill in the web form with appropriate values. Required fields have an exclamation mark (!) next to them:

- isolate_field - Dropdown list of isolate fields.
  - Allowed: selection from list.
- attribute - Name of extended attribute, e.g. continent.
  - Allowed: any text (no spaces).
- value_format - Format for values.
  - Allowed: integer/float/text/date.
- value_regex - Regular expression to enforce allele id naming.
  - ^: the beginning of the string
  - $:the end of the string
  - d: digit
  - D: non-digit
  - s: white space character
  - S: non white space character

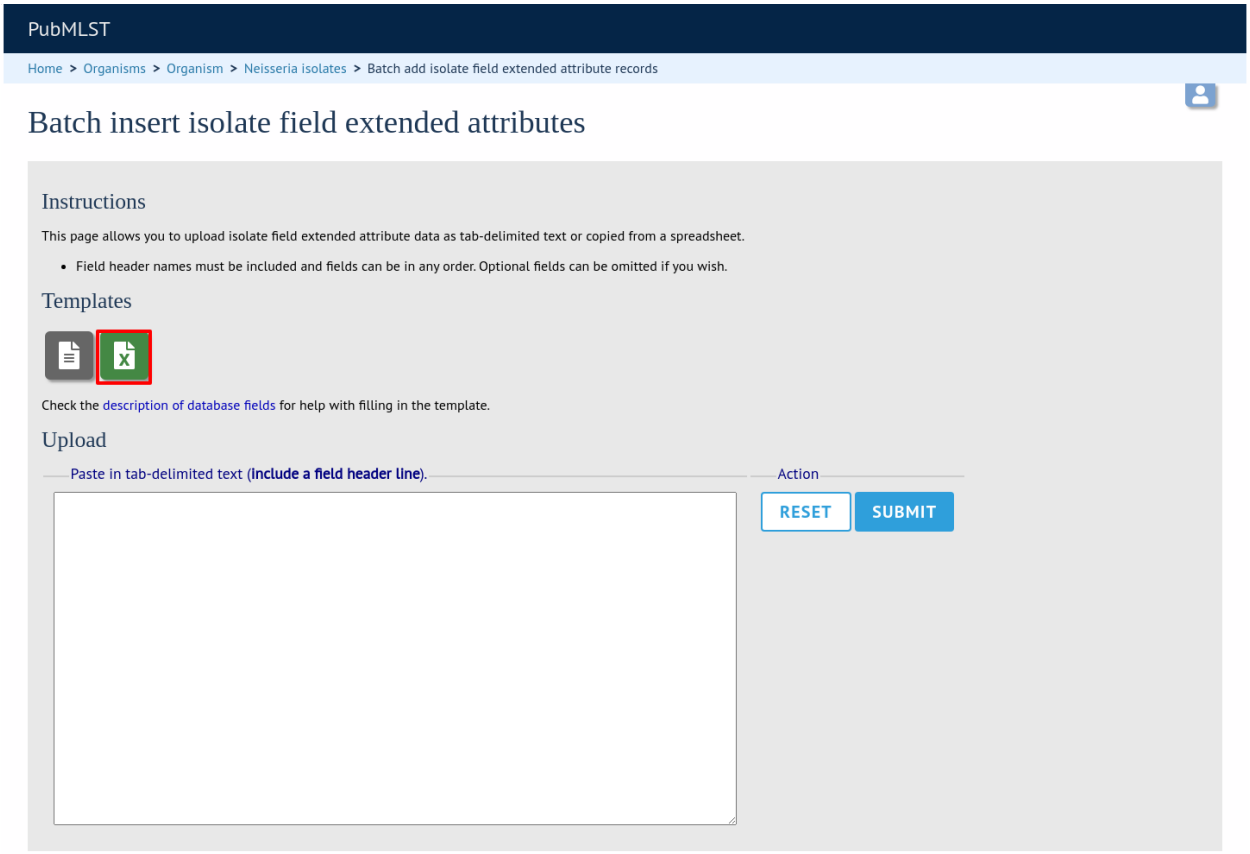
  - w: alpha-numeric plus '_'
  - .: any character
  - *: 0 or more of previous character
  - +: 1 or more of previous character
  - e.g. ^Fd-d+$ states that a value must begin with a F followed by a single digit, then a dash, then one or more digits, e.g. F1-12

- description - Long description - this isn't currently used but may be in the future.
  - Allowed: any text.
- url - URL used to hyperlink values in the isolate information page. Instances of [?] within the URL will be substituted with the value.
  - Allowed: any valid URL (either relative or absolute).
- length - Maximum length of extended attribute value.
  - Allowed: any positive integer.
- field_order - Integer that sets the order of the field following it's parent isolate field.
  - Allowed: any integer.

The easiest way to populate the lookup table is to do a batch update copied from a spreadsheet. Click the batch add (++) isolate field extended attribute values link on the curator's main page (this link will only appear once an extended attribute has been defined). This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Download the Excel template:



Fill in the columns with your values, e.g.

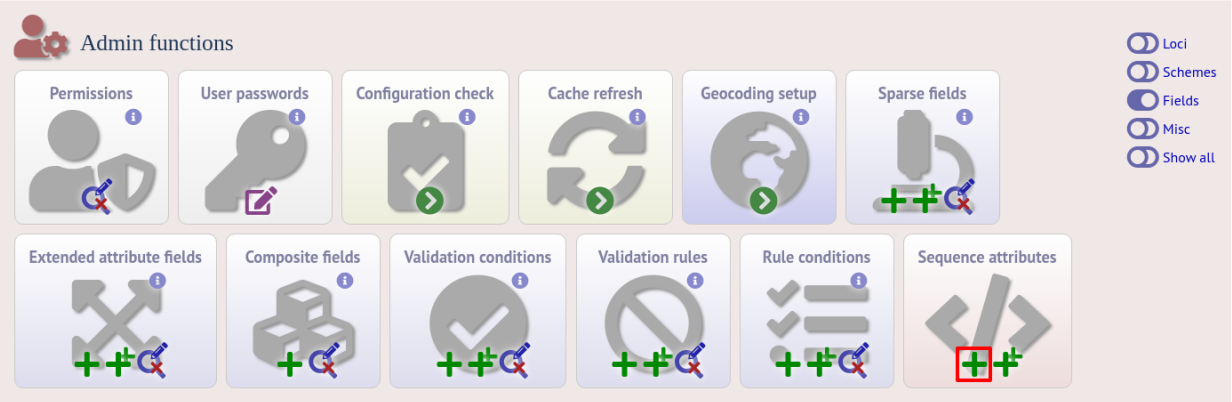
isolate_field	attribute	field_value	value
country	continent	Afghanistan	Asia
country	continent	Albania	Europe
country	continent	Algeria	Africa
country	continent	Andorra	Europe
country	continent	Angola	Africa

Paste from the spreadsheet in to the upload form and click 'Submit'.

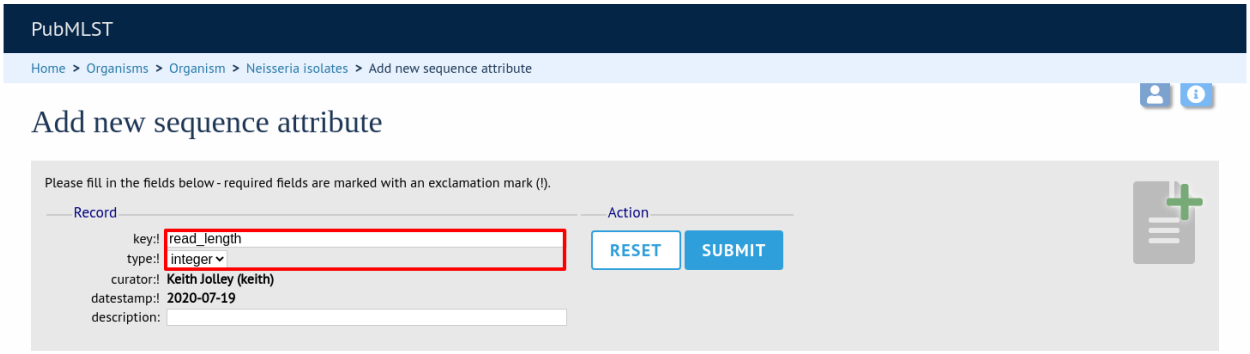
5.29 Sequence bin attributes

It is possible that you will want to store extended attributes for sequence bin contigs when you upload them. Examples may be read length, assembler version, etc. Since there are almost infinite possibilities for these fields, and they are likely to change over time, they are not hard-coded within the database. An administrator can, however, create their own attributes for a specific database and these will then be available in the web form when uploading new contig data. The attributes are also searchable.

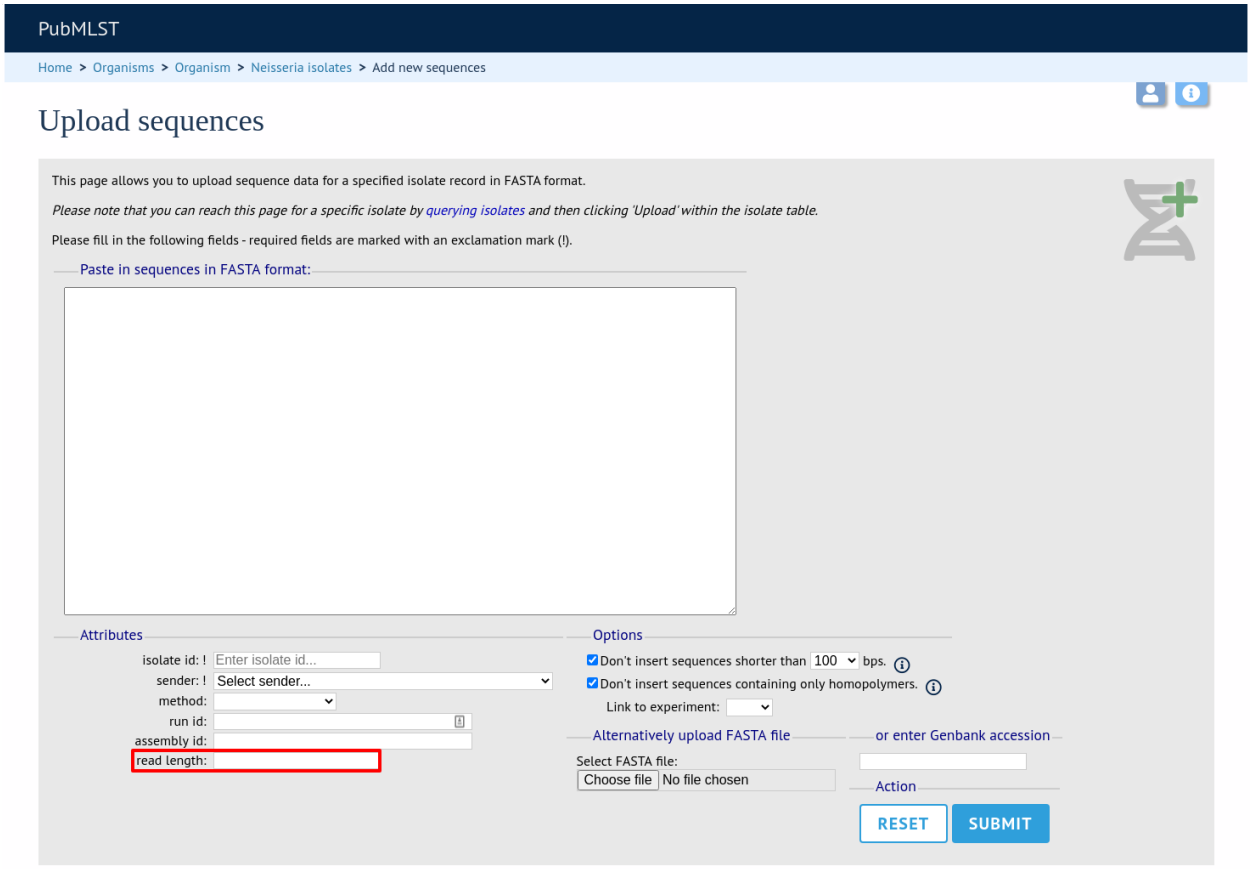
To set up new attributes, click the add (+) 'sequence attributes' link on the isolate database curator's index page. This function is normally hidden, so you may need to click the 'Fields' toggle to display it.



Enter the name of the attribute as the 'key', select the type of data (text, integer, float, date) and an optional short description. Click 'Submit'.



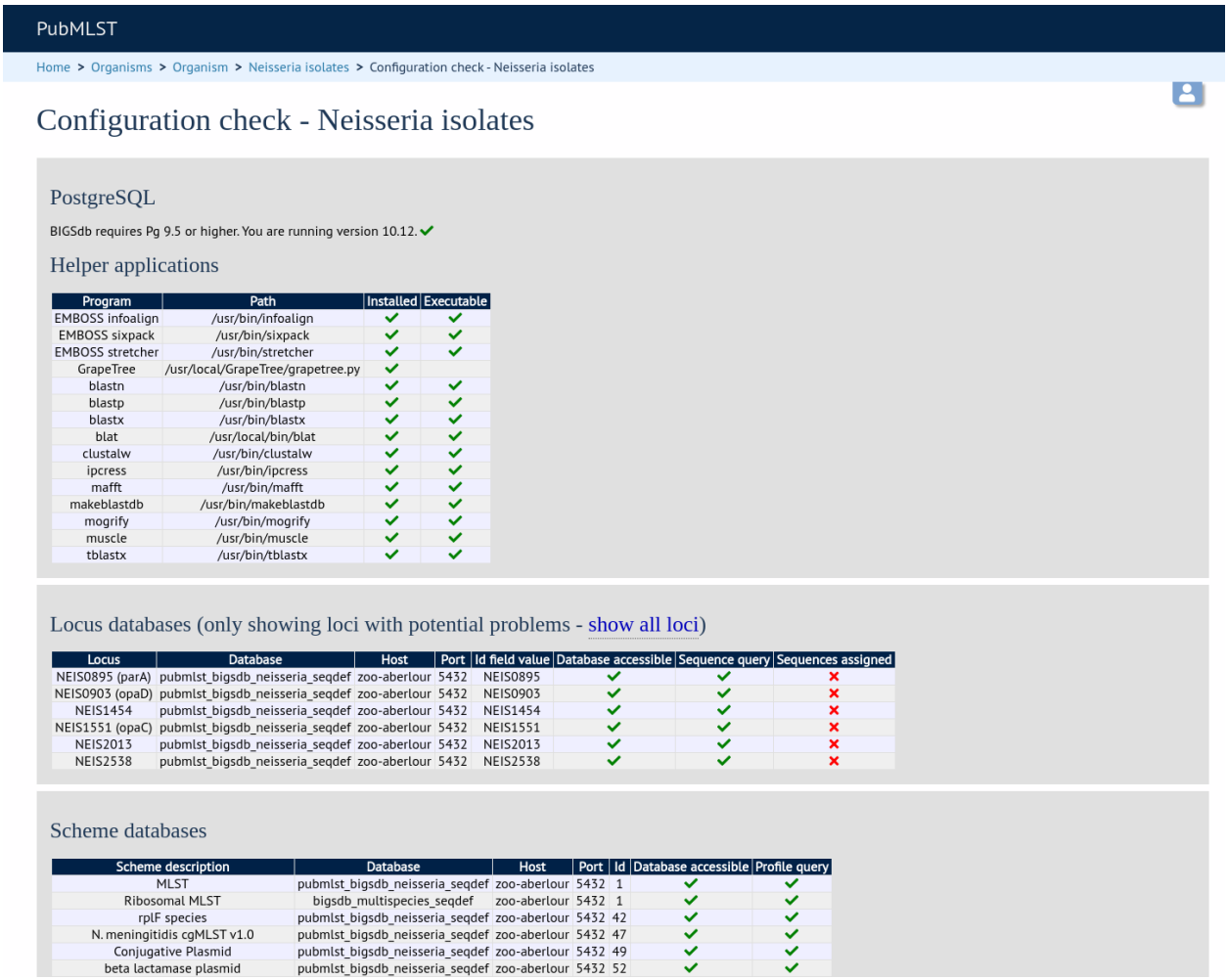
This new attribute will then be available when *uploading contig data*.



5.30 Checking external database configuration settings

Click the 'Configuration check' link on the curator's index page.

The software will check that required helper applications are installed and executable and, in isolate databases, test every locus and scheme external database to check for connectivity and that data can be retrieved. By default, only loci which have an issue will be displayed but you can click the 'show all loci' link to display them all.



PubMLST

Home > Organisms > Organism > Neisseria isolates > Configuration check - Neisseria isolates

Configuration check - Neisseria isolates

PostgreSQL

BIGSdb requires Pg 9.5 or higher. You are running version 10.12. ✔

Helper applications

Program	Path	Installed	Executable
EMBOSS infoalign	/usr/bin/infoalign	✔	✔
EMBOSS sixpack	/usr/bin/sixpack	✔	✔
EMBOSS stretcher	/usr/bin/stretcher	✔	✔
GrapeTree	/usr/local/GrapeTree/grapetree.py	✔	
blastn	/usr/bin/blastn	✔	✔
blastp	/usr/bin/blastp	✔	✔
blastx	/usr/bin/blastx	✔	✔
blat	/usr/local/bin/blat	✔	✔
clustalw	/usr/bin/clustalw	✔	✔
ipcress	/usr/bin/ipcress	✔	✔
mafft	/usr/bin/mafft	✔	✔
makeblastdb	/usr/bin/makeblastdb	✔	✔
mogrify	/usr/bin/mogrify	✔	✔
muscle	/usr/bin/muscle	✔	✔
tblastx	/usr/bin/tblastx	✔	✔

Locus databases (only showing loci with potential problems - show all loci)

Locus	Database	Host	Port	Id field value	Database accessible	Sequence query	Sequences assigned
NEIS0895 (parA)	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	NEIS0895	✔	✔	✘
NEIS0903 (opaD)	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	NEIS0903	✔	✔	✘
NEIS1454	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	NEIS1454	✔	✔	✘
NEIS1551 (opaC)	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	NEIS1551	✔	✔	✘
NEIS2013	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	NEIS2013	✔	✔	✘
NEIS2538	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	NEIS2538	✔	✔	✘

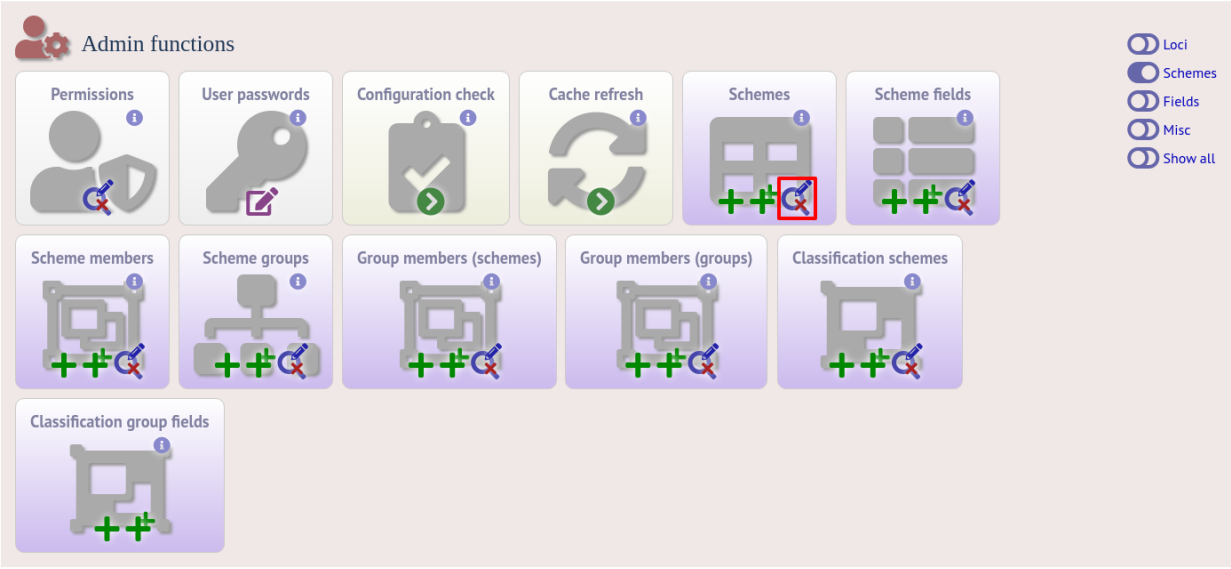
Scheme databases

Scheme description	Database	Host	Port	Id	Database accessible	Profile query
MLST	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	1	✔	✔
Ribosomal MLST	bigsdb_multispecies_seqdef	zoo-aberlour	5432	1	✔	✔
rplF species	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	42	✔	✔
N. meningitidis cgMLST v1.0	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	47	✔	✔
Conjugative Plasmid	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	49	✔	✔
beta lactamase plasmid	pubmlst_bigsdb_neisseria_seqdef	zoo-aberlour	5432	52	✔	✔

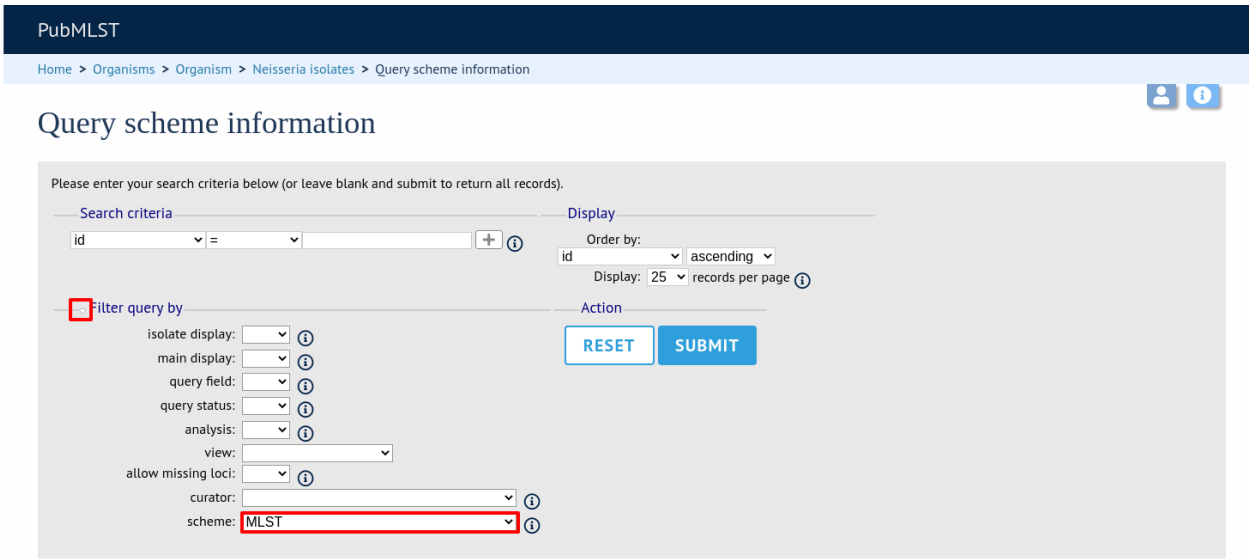
Any problems will be highlighted with a red X.

5.31 Exporting table configurations

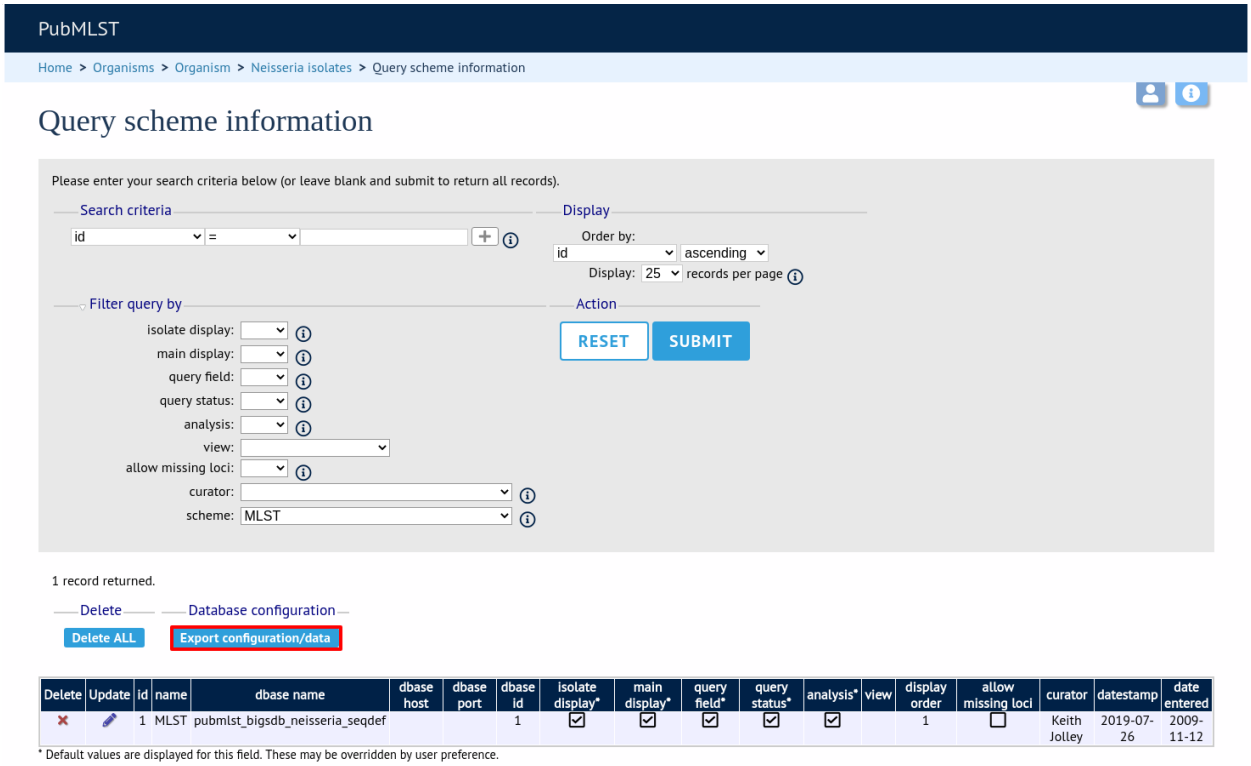
Sometimes it is useful to transfer configurations between different databases or to export a configuration for troubleshooting. Data from most of the tables can be exported in tab-delimited text format suitable for batch uploading. For example, to export scheme configuration data, click the query link (Update or delete) next to schemes in the curator's interface. This function is normally hidden, so you may need to click the 'Schemes' toggle to display it.



Expand the filters and select the required scheme in the dropdown box, then press submit.



Click the button ‘Export configuration/data’.



The three tables that are used to define a scheme (schemes, scheme_members and scheme_fields) are displayed in a format suitable for copy and pasting.

```
schemes
-------
id description dbase_name  dbase_host  dbase_port  dbase_user  dbase_password dbase_
↪table isolate_display   main_display   query_field query_status   analysis display_
↪order  allow_missing_loci   curator  datestamp   date_entered
1  MLST  pubmlst_bigsdb_neisseria_seqdef             mv_scheme_1 1  1  1  1  1  1     2␣
↪ 2012-03-22  2009-11-12

scheme_members
--------------
scheme_id   locus profile_name   field_order curator  datestamp
1  abcZ     1  2  2009-11-12
1  adk      2  2  2009-11-12
1  aroE     3  2  2009-11-12
1  fumC     4  2  2009-11-12
1  gdh      5  2  2009-11-12
1  pdhC     6  2  2009-11-12
1  pgm      7  2  2009-11-12

scheme_fields
-------------
scheme_id   field type  primary_key description field_order url   isolate_display   main_
↪display   query_field dropdown curator  datestamp
1  ST integer  1     1  /cgi-bin/bigsdb/bigsdb.pl?page=profileInfo&db=pubmlst_neisseria_
↪seqdef&scheme_id=1&profile_id=[?]  1  1  1  0  2  2010-01-20
1  clonal_complex text  0     2     1  1  1  1  2  2009-11-16
```

5.32 Authorizing third-party client software to access authenticated resources

If you are running the *RESTful API*, you will need to specifically authorize client software to connect to authenticated resources. This involves creating a client key and a client secret that is used to sign requests coming from the application. The client key and secret should be provided to the application developer.

There is a script to do this in the scripts/maintenance directory of the download archive. The script is called create_client_credentials and should be run by the postgres user. A full list of options can be found by typing:

```
create_client_credentials.pl --help

NAME
    create_client_credentials.pl - Generate and populate
    authentication database with third party application (API client)
    credentials.

SYNOPSIS
    create_client_credentials.pl --application NAME [options]

OPTIONS
-a, --application NAME
    Name of application.

-d, --deny
    Set default permission to 'deny'.  Permissions for access to specific
    database configurations will have to be set.  If not included, the default
    permission will allow access to all resources by the client.

-h, --help
    This help page.

-i, --insert
    Add credentials to authentication database.  This will fail if a matching
    application version already exists (use --update in this case to overwrite
    existing credentials).

-u, --update
    Update exisitng credentials in the authentication database.

-v, --version VERSION
    Version of application (optional).
```

5.33 BLAST caches

Sequence definition databases cache any BLAST databases that they create in order to perform sequence queries. These caches can be found in subdirectories named with the database name in the temp directory defined by the secure_tmp_dir attribute in bigsdb.conf, e.g. /var/tmp/pubmlst_bigsdb_neisseria_seqdef.

These BLAST databases will be marked stale if new alleles are added to the BIGSdb database for any locus covered by the cache. A cache marked stale will be recreated the next time a matching sequence query needs to use it. BLAST databases will also be marked stale if they are older than the cache_days setting in bigsdb.conf (default = 7 days).

It is possible to also manually create and refresh these caches using the update_blast_caches.pl script found in the scripts/maintenance directory.

A full list of options can be found by typing:

```
update_blast_caches.pl --help

NAME
    update_cached_blast_dbs.pl - Refresh BLAST database caches

SYNOPSIS
    update_cached_blast_dbs.pl --database DB_CONFIG [options]

OPTIONS
--all_loci
    Refresh or create cache for all loci.

--database DATABASE CONFIG
    Database configuration name.

--delete_all
    Remove all cache files.

--delete_old
    Remove cache files older than the cache_days setting in bigsdb.conf or
    that have been marked stale.

--delete_single_locus
    Remove caches containing only one locus. There can be many of these and
    they can clutter the cache directory. They are generally quick to recreate
    when needed.

--help
    This help page.

--quiet
    Only show errors.

--refresh
    Refresh existing caches.

--scheme SCHEME_ID
    Refresh or create cache for specified scheme.
```

5.34 Config-specific file downloads

You can make files available on a static website but restrict their access only to users who can authenticate for access to the current database configuration.

This can be done by adding a file called download_files.conf to the database configuration directory within /etc/bigsdb/dbases. This file consists of three columns in tab-delimited format:

- The full path of the file in the file system
- Label which will be used to hyperlink to the file
- A description of the file
- The file type (docx, html, gif,jpg, pdf, png, tar, tar, xlsx currently supported)

The files can be downloaded directly from a BIGSdb URL: /cgi-bin/bigsdb.pl?db=CONFIG&page=downloadFiles&file=LABEL (where CONFIG is the database config name and LABEL is the label used in the download_files.conf file. These URLs can be used as standard links within a web page.

You can also list all available files with the URL: /cgi-bin/bigsdb.pl?db=CONFIG&page=downloadFiles

Navigating to these links will prompt the user to log-in if they are not already (if the database config requires this).

CHAPTER

SIX

CURATOR'S GUIDE

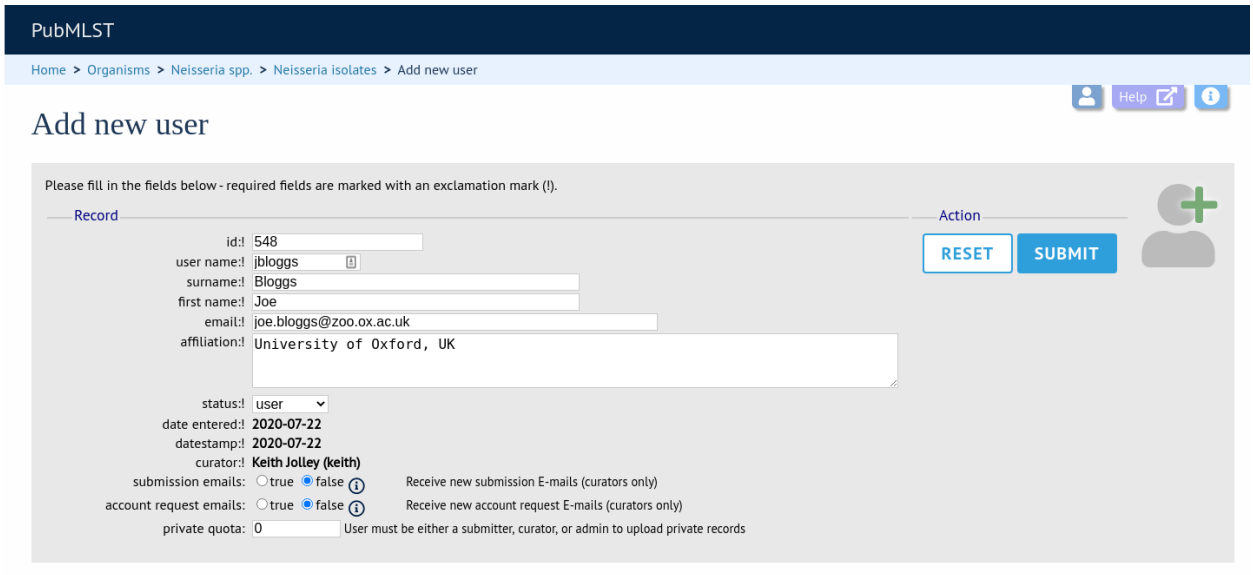
Please note that links displayed within the curation interface will vary depending on database contents and the permissions of the curator. Some infrequently used links are usually hidden by default. These can be enabled by clicking the 'Show all' toggle switch. The admin section has feature- specific toggles as well as a 'Show all' toggle.

6.1 Adding new sender details

All records within the databases are associated with a sender. Whenever somebody new submits data, they should be added to the users table so that their name appears in the dropdown lists on the data upload forms.

To add a user, click the add users (+) link on the curator's contents page.

Enter the user's details in to the form.



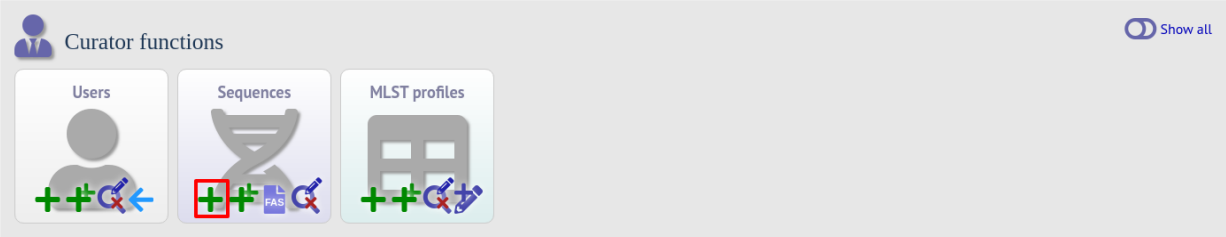
Normally the status should be set as 'user'. Only admins and curators with special permissions can create users with a status of curator or admin.

If the submission system is in operation there will be an option at the bottom called 'submission_emails'. This is to enable users with a status of 'curator' or 'admin' to receive E-mails on receipt of new submissions. It is not relevant for users with a status of 'user' or 'submitter'.

6.2 Adding new allele sequence definitions

6.2.1 Single allele

To add a single new allele, click the sequences add (+) link on the curator's main page.



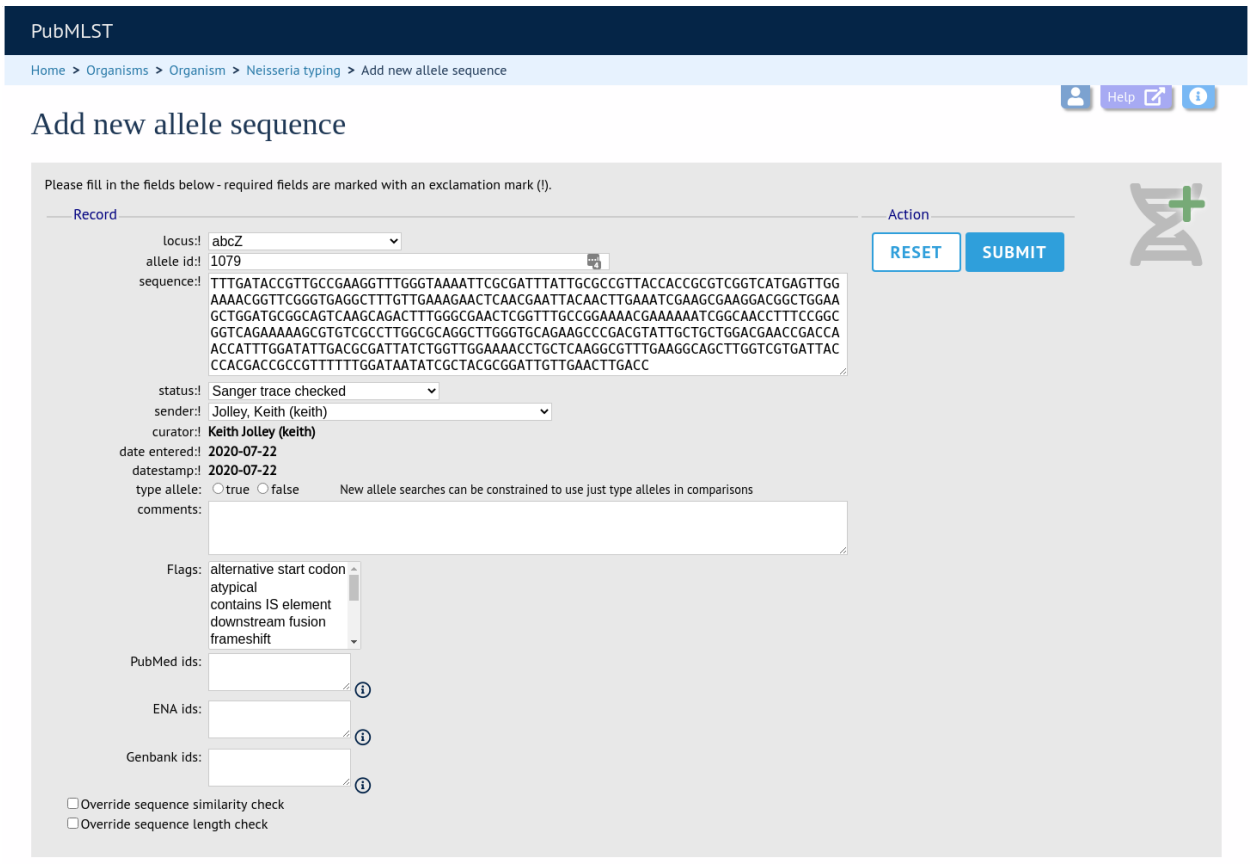
Select the locus from the dropdown list box. The next available allele id will be entered automatically (if the allele id format is set to integer). Paste the sequence in to form, set the status and select the sender name from the dropdown box. If the sender does not appear in the box, you will need to add them to the registered users.

The status reflects the level of curation that the curator has done personally - the curator should not rely on assurances from the submitter. The status can either be:

- Sanger trace checked
  - Sequence trace files have been assembled and inspected *by the curator*.
- WGS: manual extract (BIGSdb)
  - The sequence has been extracted manually from a BIGSdb database *by the curator* . There may be some manual intervention to identify the start and stop sites of the sequence.

- WGS: automated extract (BIGSdb)
  - The sequences have been generated by a BIGSdb tag scanning run and have had no manual inspection or intervention.
- WGS: visually checked
  - Short read data has been inspected visually using an alignment program *by the curator*.
- WGS: automatically checked
  - The sequences have been checked by an automated algorithm that assesses the quality of the data to ensure it meets specified criteria.
- unchecked
  - If none of the above match, then the sequence should be entered as unchecked.

You can also choose whether to designate the sequence as a type allele or not. Type alleles can be used to constrain the sequence search space when defining new alleles using the *web-based scanner* or *offline auto allele definer*.



Press submit. By default, the system will test whether your sequence is similar enough to existing alleles defined for that locus. The sequence will be rejected if it isn't considered similar enough. This test can be overridden by checking the 'Override sequence similarity check' checkbox at the bottom. It will also check that the sequence length is within the allowed range for that locus. These checks can also be overridden by checking the 'Override sequence length check' checkbox, allowing the addition of unusual length alleles.

See also:

allele sequence flags

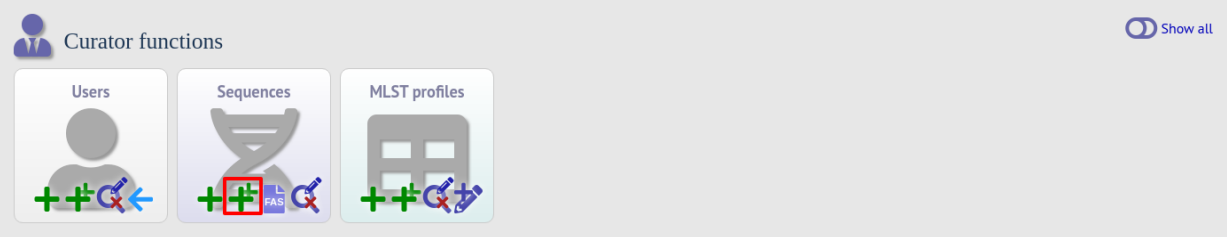
Sequences can also be associated with PubMed, ENA or Genbank id numbers by entering these as lists (one value per line) in the appropriate form box.

6.2.2 Batch adding multiple alleles

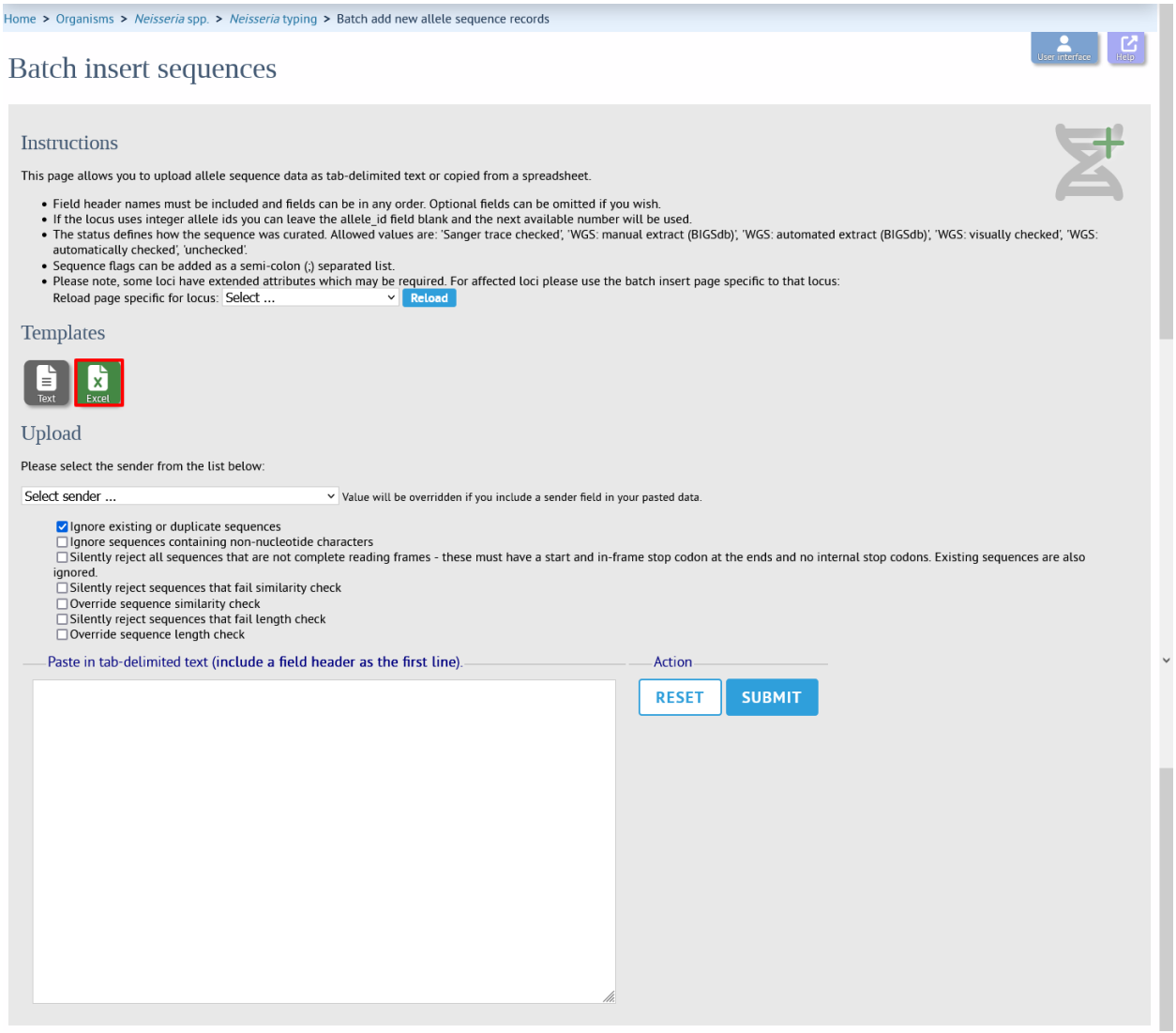
There are two methods of batch adding alleles. You can either upload a spreadsheet with all fields in tabular format, or you can upload a FASTA file provided all sequences are for the same locus and have the same status.

Upload using a spreadsheet

Click the batch add (++) sequences link on the curator's main page.



Download a template Excel file from the following page.



Fill in the spreadsheet. If the locus uses integer allele identifiers, the allele_id can be left blank and the next available number will be used automatically.

The status can be either: 'Sanger trace checked', 'WGS: manual extract (BIGSdb)', 'WGS: automated extract (BIGSdb)', 'WGS: visually checked', 'WGS: automatically checked' or 'unchecked'. See full explanations for these in the *single allele upload* section.

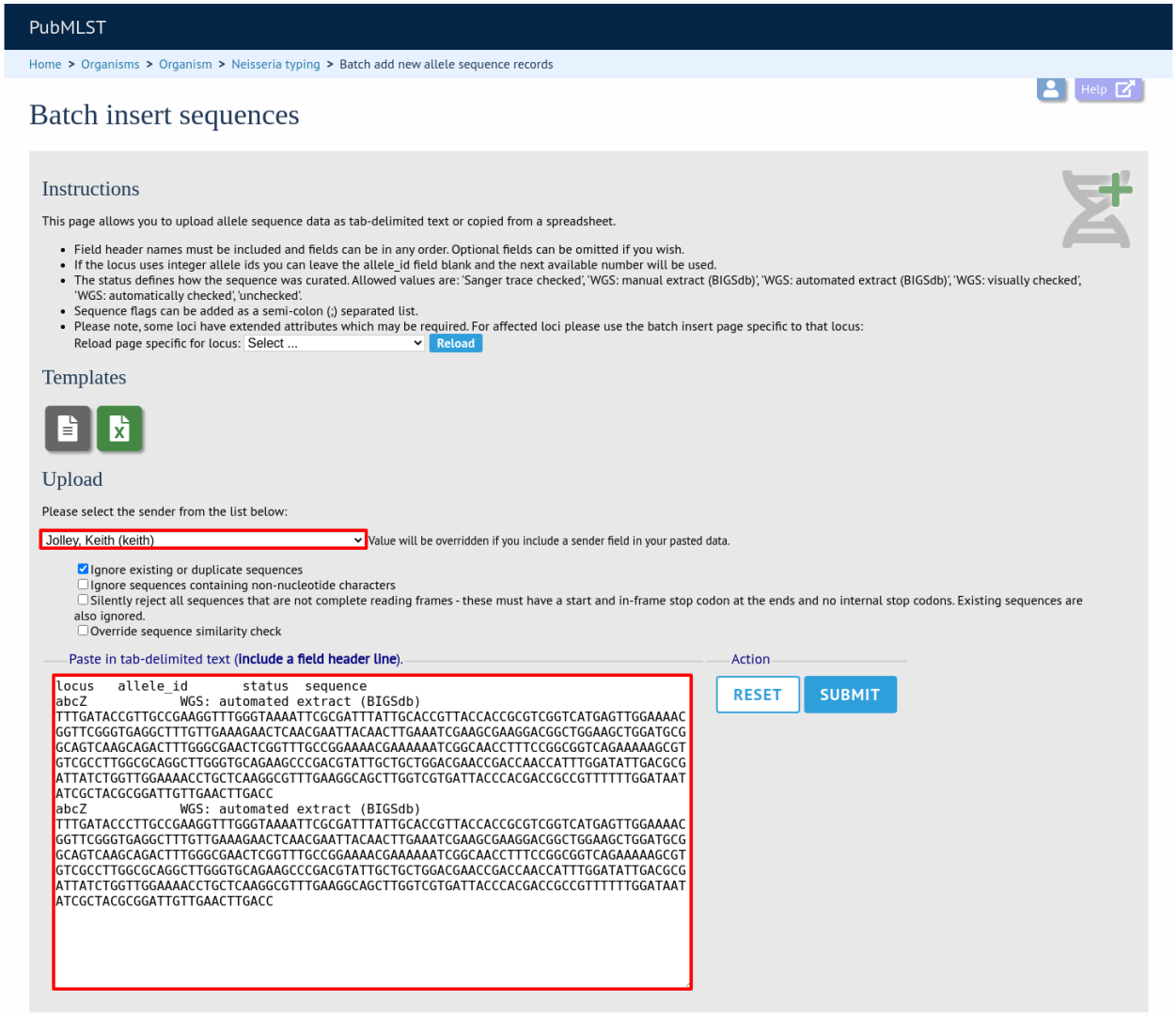
The 'type_allele' field is boolean (true/false) and specifies if the sequence should be considered as a type allele. These can be used to constrain the sequence search space when defining new alleles using the *web-based scanner* or *offline auto allele definer*.

Paste the entire sheet in to the web form and select the sender from the dropdown box.

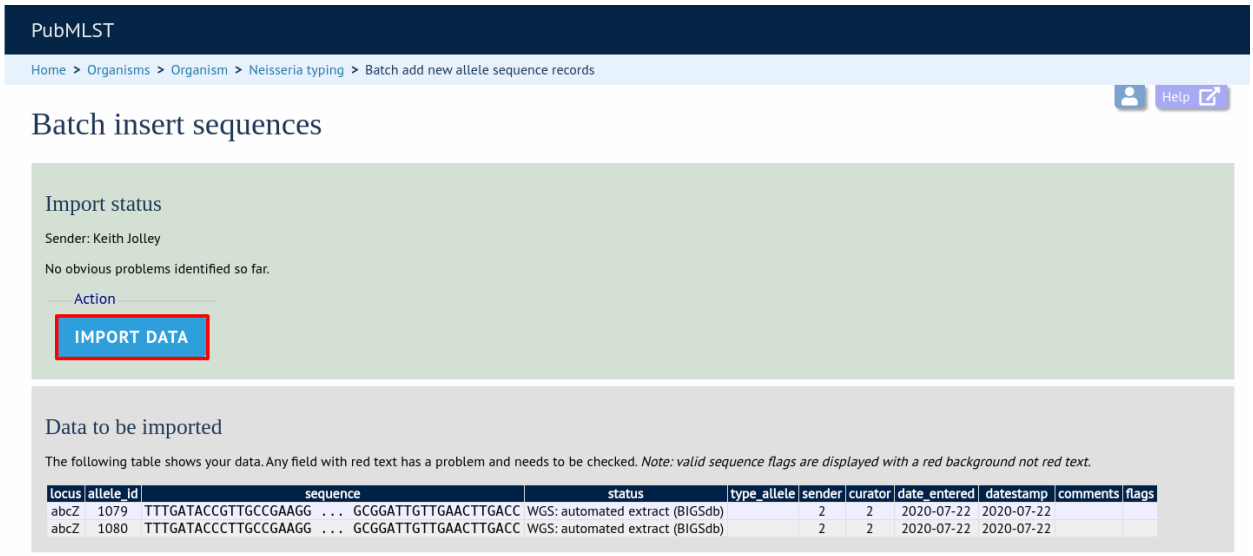
Additionally, there are a number of options available. Some of these will ignore sequences if they don't match certain criteria - this is useful when sequence data has been extracted from genomes automatically. Available options are:

- Ignore existing or duplicate sequences.
- Ignore sequences containing non-nucleotide characters.
- Silently reject all sequences that are not complete reading frames - these must have a start and in-frame stop codon at the ends and no internal stop codons. Existing sequences are also ignored.

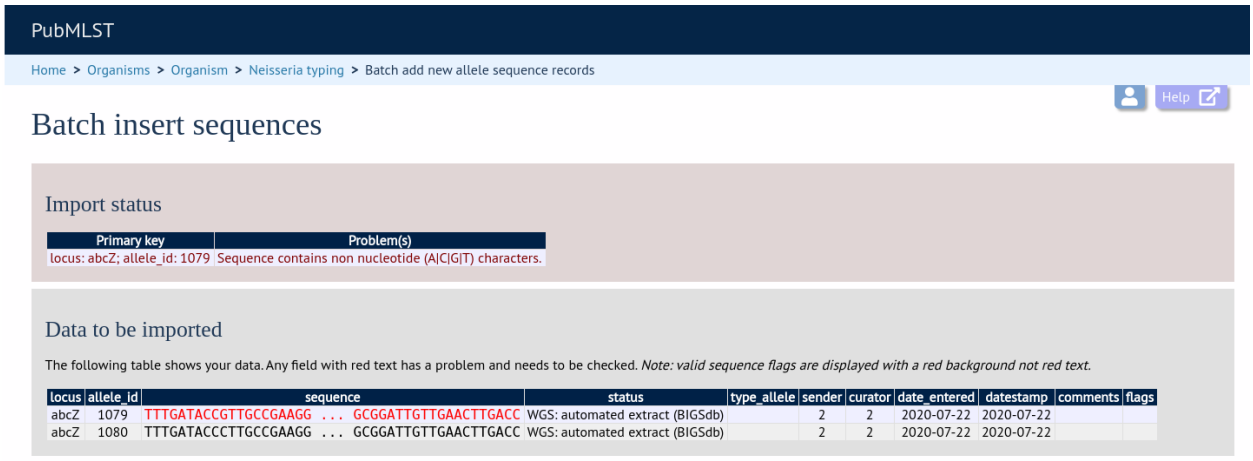
- Silently reject sequences that fail similarity check.
- Override sequence similarity check.
- Silently reject sequences that fail length check.
- Override sequence length check.



Press submit. You will be presented with a page indicating what data will be uploaded. This gives you a chance to back out of the upload. Click 'Import data'.



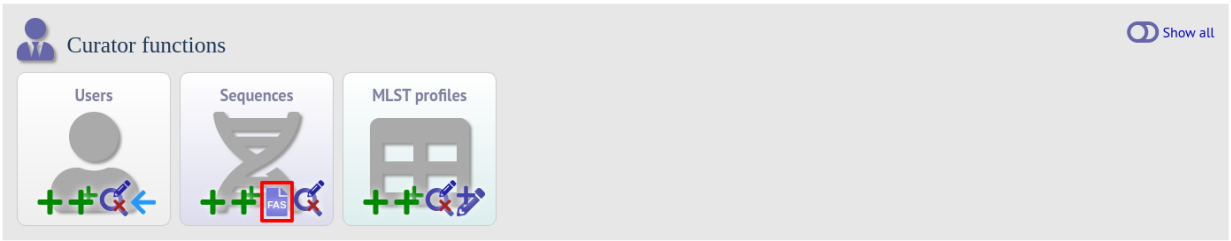
If there are any problems with the submission, these should be indicated at this stage, e.g.:



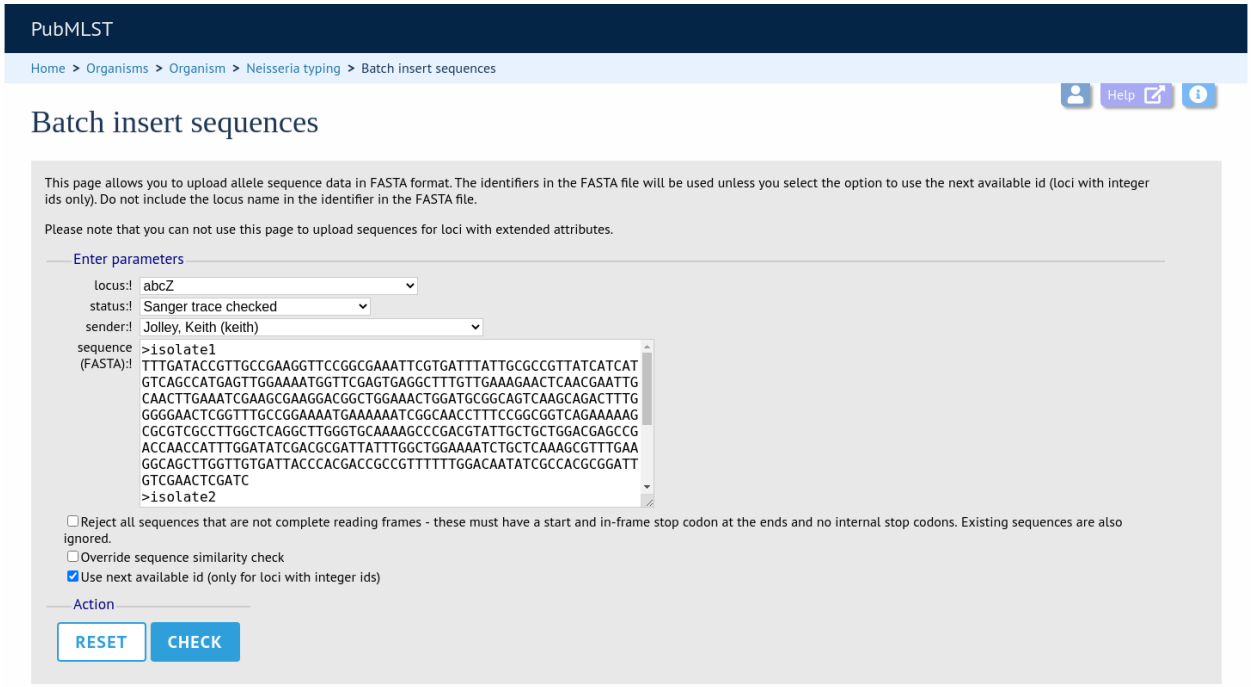
Upload using a FASTA file

Uploading new alleles from a FASTA file is usually more straightforward than generating an Excel sheet.

Click 'FASTA' upload on the curator's contents page.



Select the locus, status and sender from the dropdown boxes and paste in the new sequences in FASTA format.

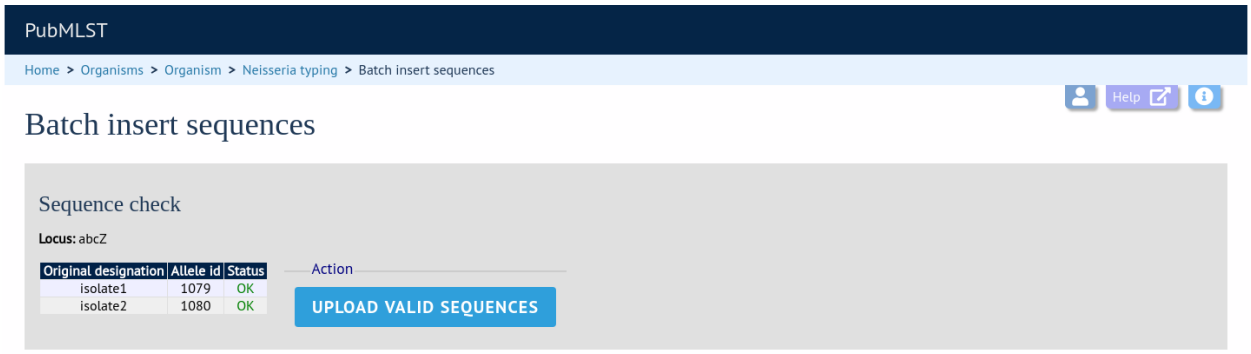


For loci with integer ids, the next available id number will be used by default (and the identifier in the FASTA file will be ignored). Alternatively, you can indicate the allele identifier within the FASTA file (do not include the locus name in this identifier).

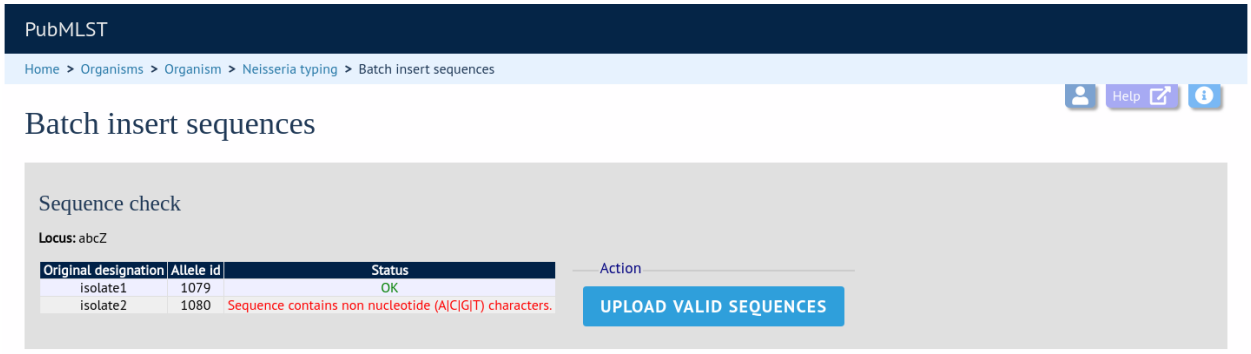
As with the spreadsheet upload, you can select options to ignore selected sequences if they don't match specific criteria.

Click 'Check'.

The sequences will be checked. You will be presented with a page indicating what data will be uploaded. This gives you a chance to back out of the upload. Click 'Upload valid sequences'.



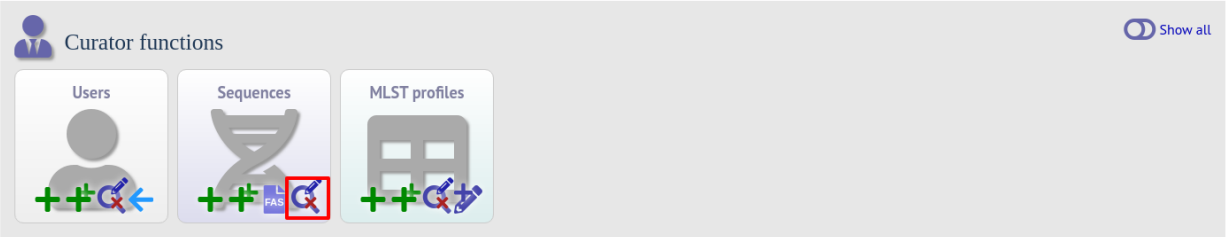
Any invalid sequences will be indicated in this confirmation page and these will not be uploaded (you can still upload the others), e.g.



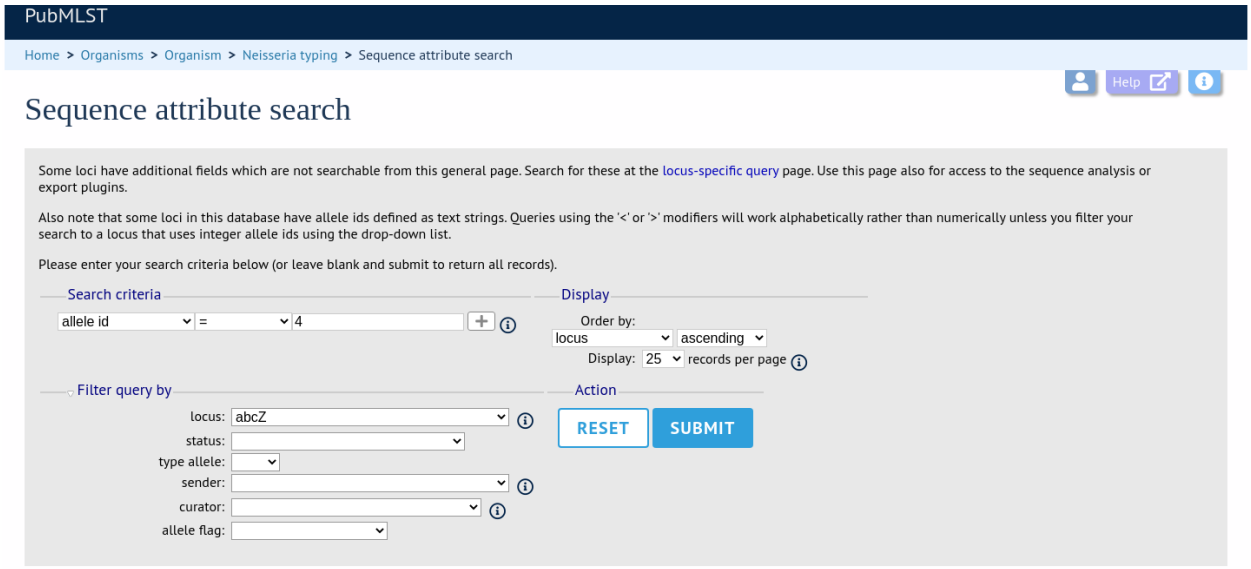
6.3 Updating and deleting allele sequence definitions

Note: You cannot update the sequence of an allele definition. This is for reasons of data integrity since an allele may form part of a scheme profile and be referred to in multiple databases. If you really need to change a sequence, you will have to remove the allele definition and then re-add it. If the allele is a member of a scheme profile, you will also have to remove that profile first, then re-create it after deleting and re-adding the allele.

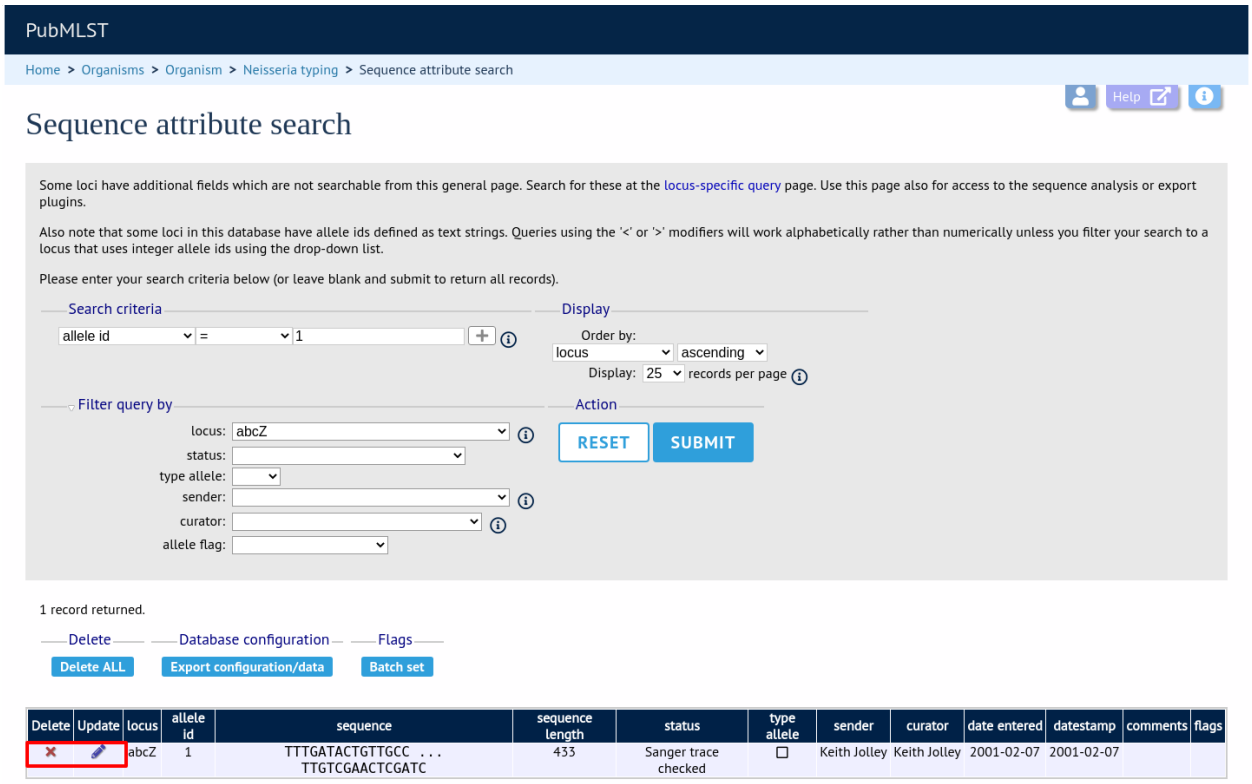
In order to update or delete an allele, first you must select it. Click the update/delete sequences link.



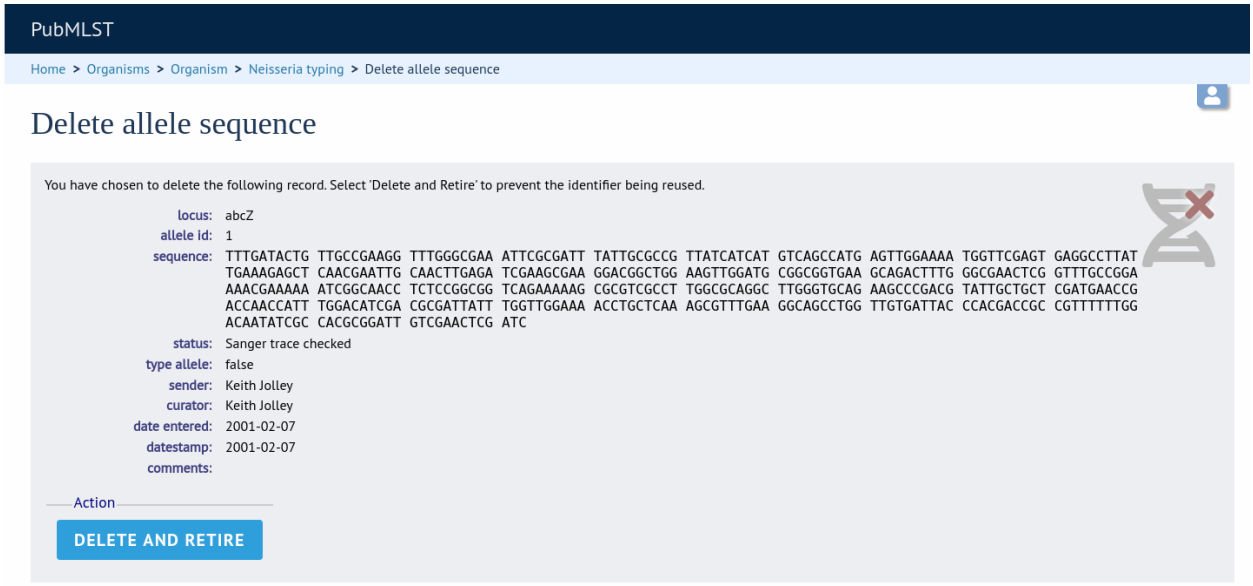
Either search for specific attributes in the search form, or leave it blank and click 'Submit' to return all alleles. For a specific allele, select the locus in the filter (click the small arrow next to 'Filter query by' to expand the filter) and enter the allele number in the allele_id field.



Click the appropriate link to either update the allele attributes or to delete it. If you have appropriate permissions, there may also be a link to 'Delete ALL'. This allows you to quickly delete all alleles returned from a search.

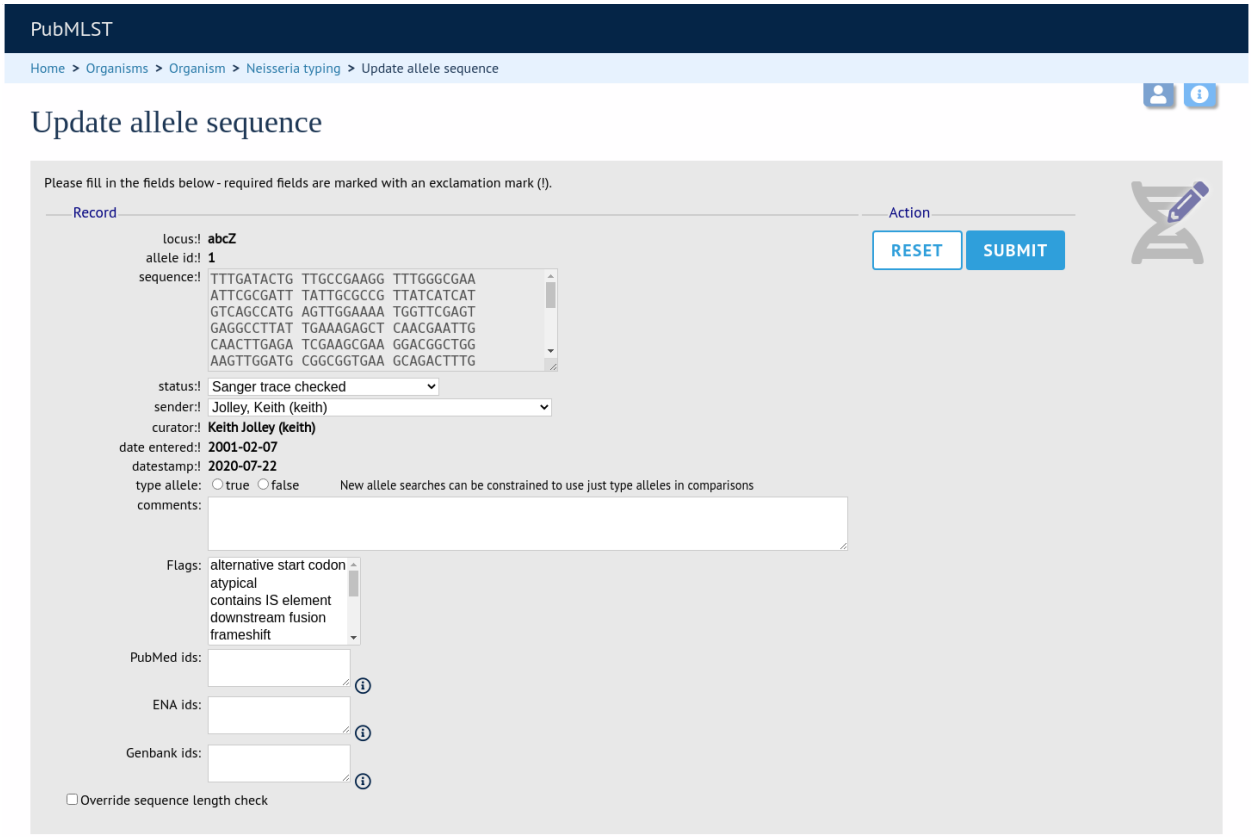


If you choose to delete, you will be presented with a final confirmation screen. To go ahead, click 'Delete'. Deletion will not be possible if the allele is part of a scheme profile - if it is you will need to delete any profiles that it is a member of first. You can also choose to delete and retire the allele identifier. If you do this, the allele identifier will not be re-used. It is possible to set the configuration so that you only have the option to delete and retire.



If instead you clicked 'Update', you will be able to modify attributes of the sequence, or link PubMed, ENA or Genbank records to it. You will not be able to modify the sequence itself.

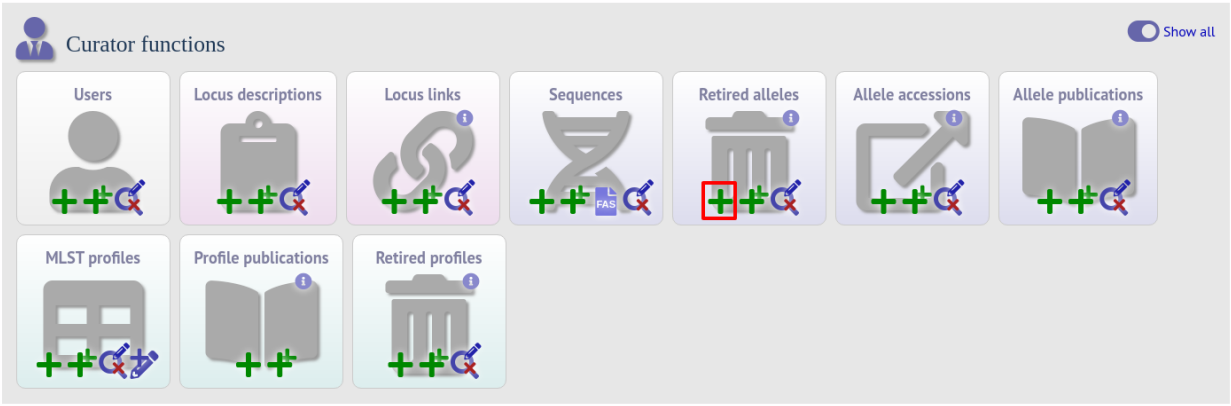
Note: Adding flags and comments to an allele record requires that this feature is enabled in the *database configuration*.



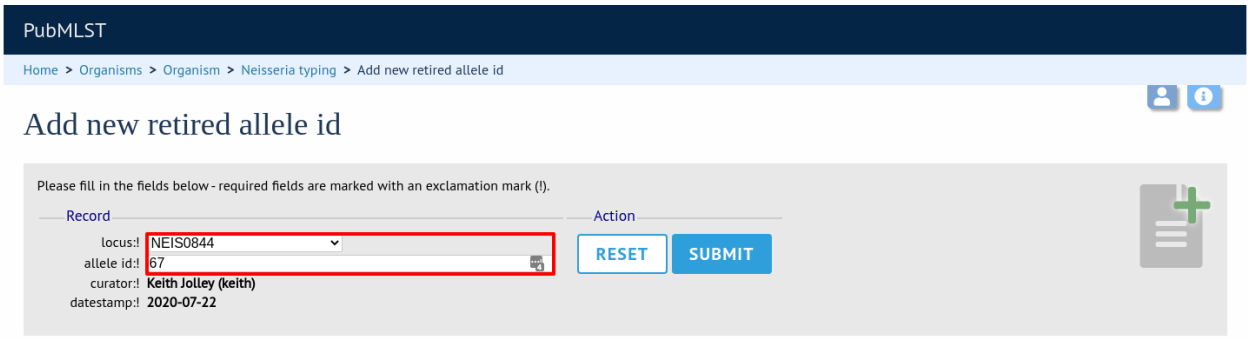
6.4 Retiring allele identifiers

Sometimes there is a requirement to prevent the automated assignment of a particular allele identifier - an allele with that identifier may have been commonly used and has since been removed. Reassignment of the identifier to a new sequence may lead to confusion, so in this instance, it would be better to prevent this.

You can retire an allele identifier by clicking the 'Add' retired allele ids link on the sequence database curators' page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Select the locus from the dropdown list box and enter the allele id. Click 'Submit'.



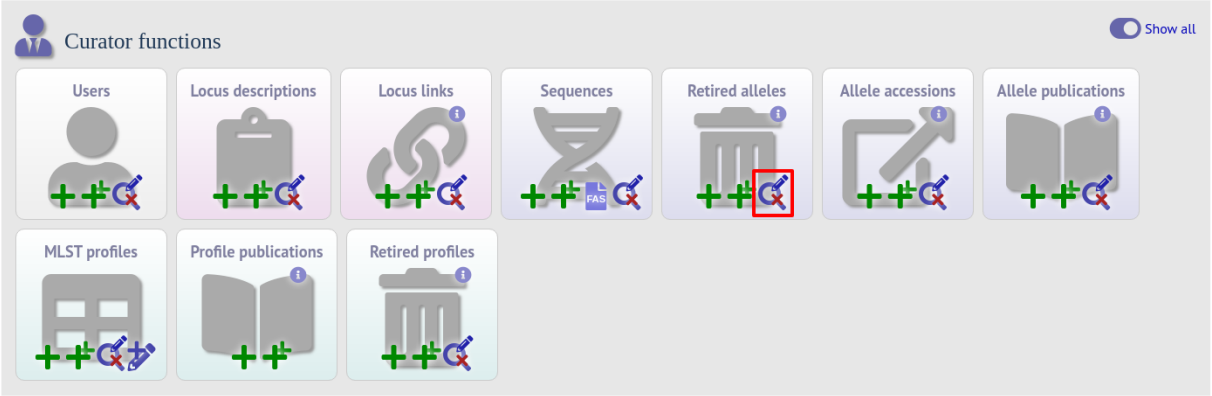
You cannot retire an allele that already exists, so you must delete it before retiring it. Once an identifier is retired, you will not be able to create a new allele with that name.

You can also retire an allele identifier when you *delete an allele*.

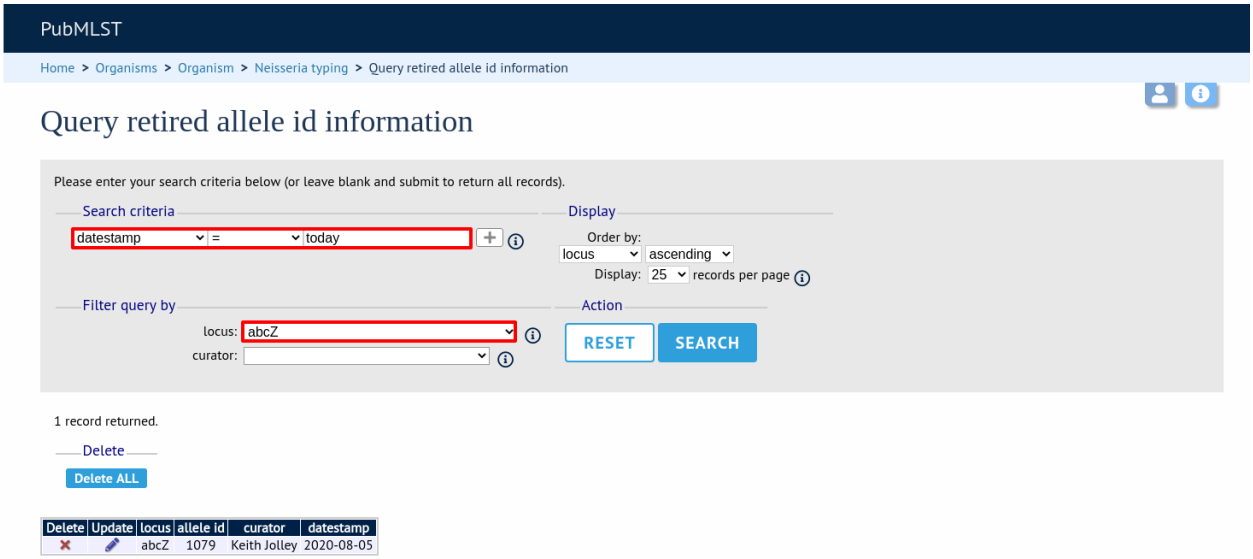
6.5 Un-retiring allele identifiers

If an allele identifier has been retired, it is possible to un-retire it so that it can be re-assigned. To do this, you need to remove the identifier from the retired_alleles table.

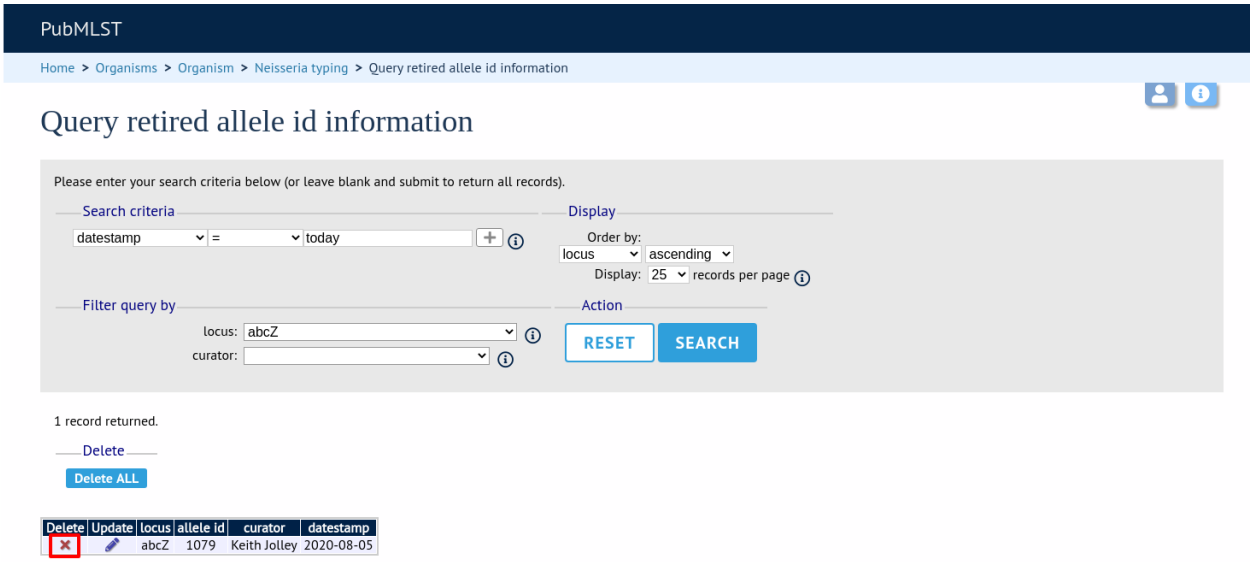
First, find the allele identifier in the retired_alleles table by clicking the 'Update/delete' retired alleles link on the sequence database curators' page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Search by any criteria to find the allele identifier.



Click the delete link on the identifer to be un-retired.



A confirmation page will be displayed. Click 'Delete' to remove the identifier from the retired alleles table.

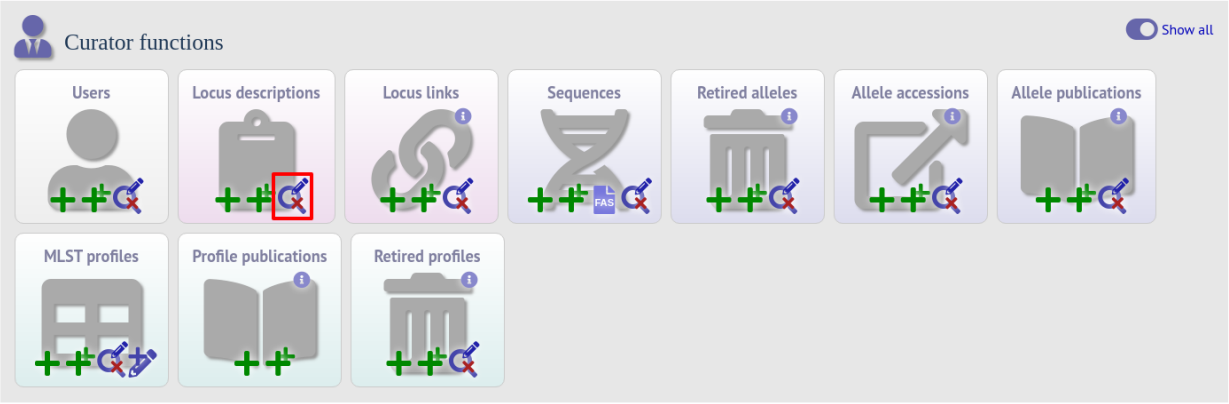
The identifier can now be re-assigned when adding a new allele.

6.6 Updating locus descriptions

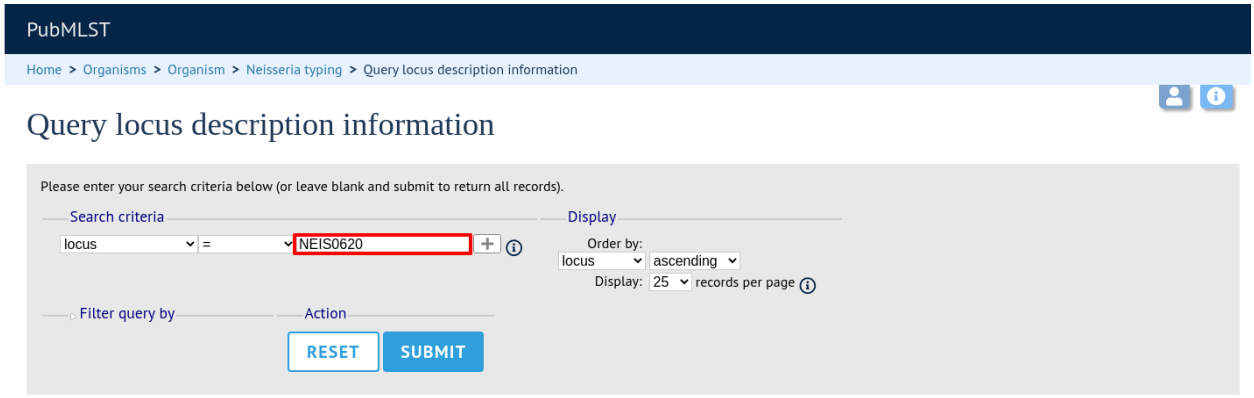
Loci in the sequence definitions database can have a description associated with them. This may contain information about the gene product, the biochemical reaction it catalyzes, or publications providing more detailed information etc. This description is accessible from various pages within the interface such as an *allele information page* or from the *allele download page*.

Note: In recent versions of BIGSdb, a blank description record is created when a new locus is defined. The following instructions assume that this is the case. It is possible for this record to be deleted or it may never have existed if the locus was created using an old version of BIGSdb. If the record does not exist, it can be added by clicking the Add (+) button in the 'locus descriptions' box. Fill in the fields in the same way as described below.

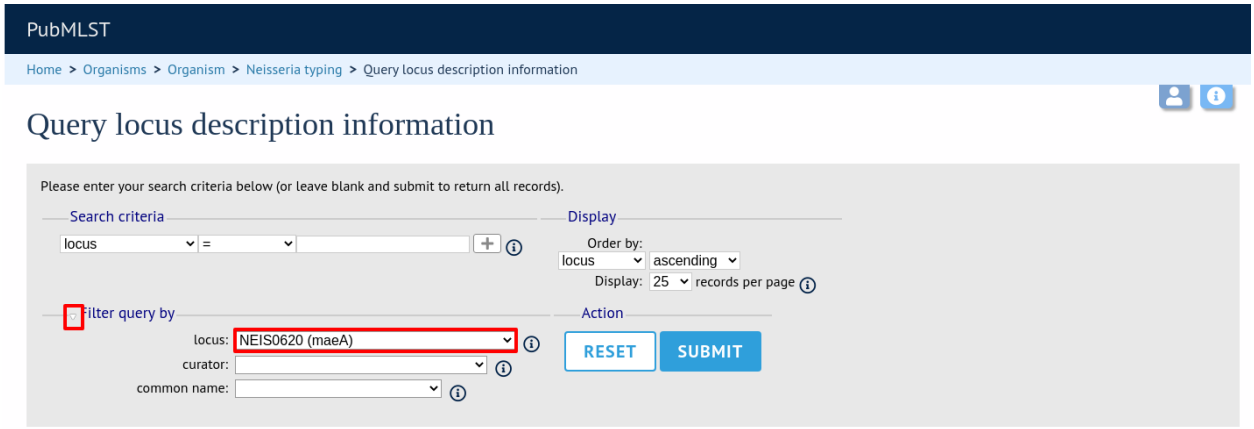
To edit a locus description, first you need to find it. Click the update/delete button in the 'locus descriptions' box on the sequence database curator's page (depending on the permissions set for your user account not all the links shown here may be displayed). This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Either enter the name of the locus in the query box:

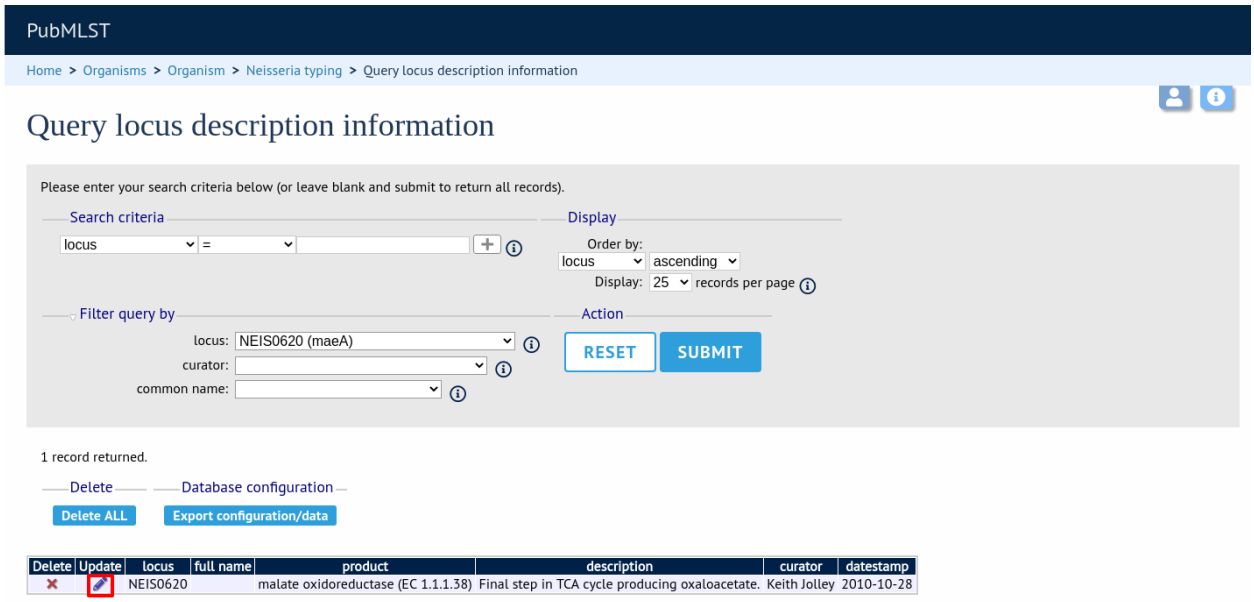


or expand the filter list and select it from the dropdown box:

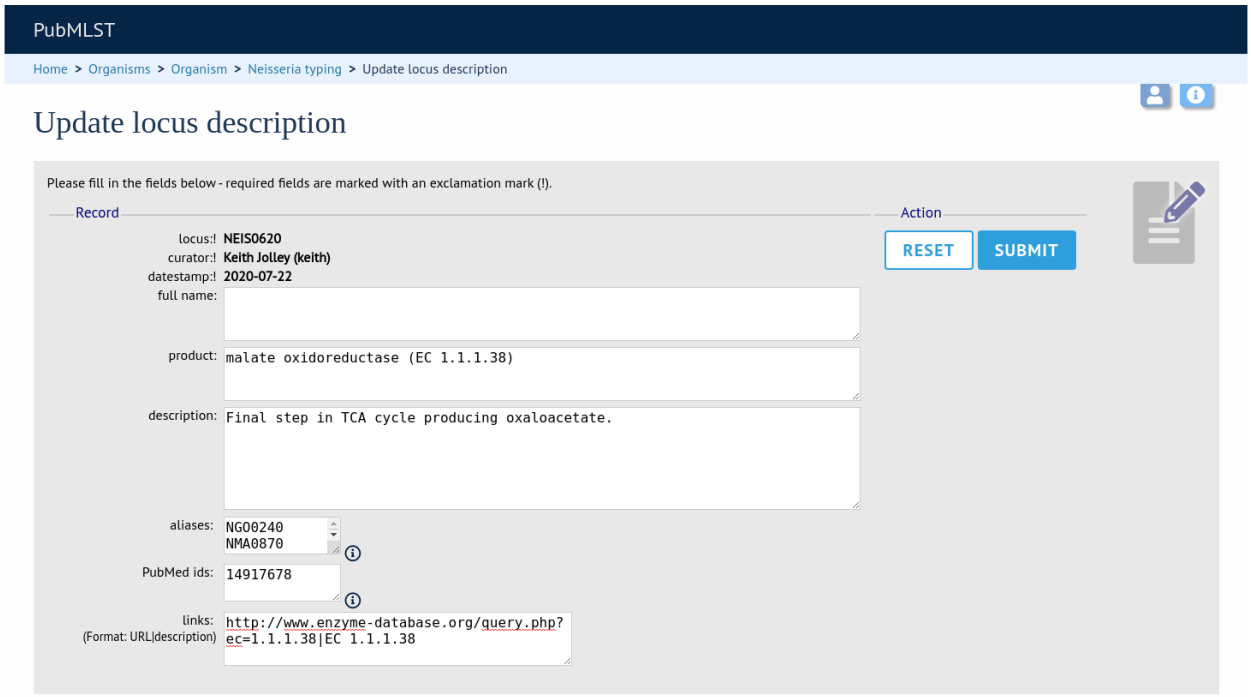


Click 'Submit'.

If the locus description exists, click the 'Update' link (if it doesn't, see the note above).



Fill in the form as needed:



- full_name

  The full name of the locus - often this can be left blank as it may be the same as the locus name. An example of where it is appropriately used is where the locus name is an abbreviation, e.g. PorA_VR1 - here we could enter 'PorA variable region 1'. This should not be used for the 'common name' of the locus (which is defined within the locus record itself) or the gene product.

- product

  The name of the protein product of a coding sequence locus.

- description

  This can be as full a description as possible. It can include the specific part of the biochemical pathway the gene product catalyses or may provide background information, as appropriate.

- aliases

  These are alternative names for the locus as perhaps found in different genome annotations. Don't duplicate the locus name or common name defined in the locus record. Enter each alias on a separate line.

- Pubmed_ids

  Enter the PubMed id of any paper that specifically describes the locus. Enter each id on a separate line. The software will retrieve the full citation from PubMed (this happens periodically so it may not be available for display immediately).

- Links

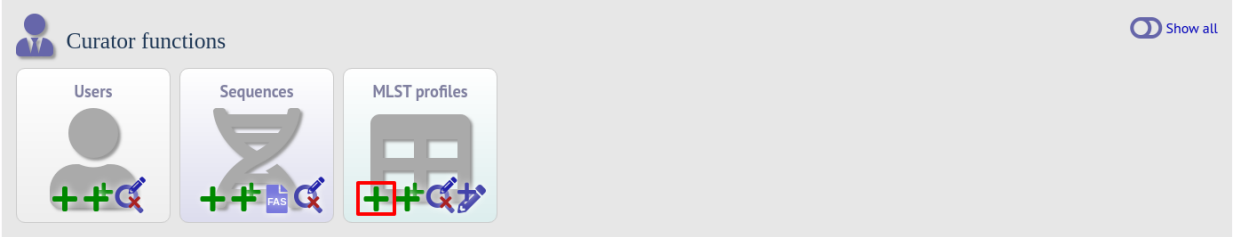
  Enter links to additional web-based resources. Enter the URL first followed by a pipe symbol (|) and then the description.

Click 'Submit' when finished.

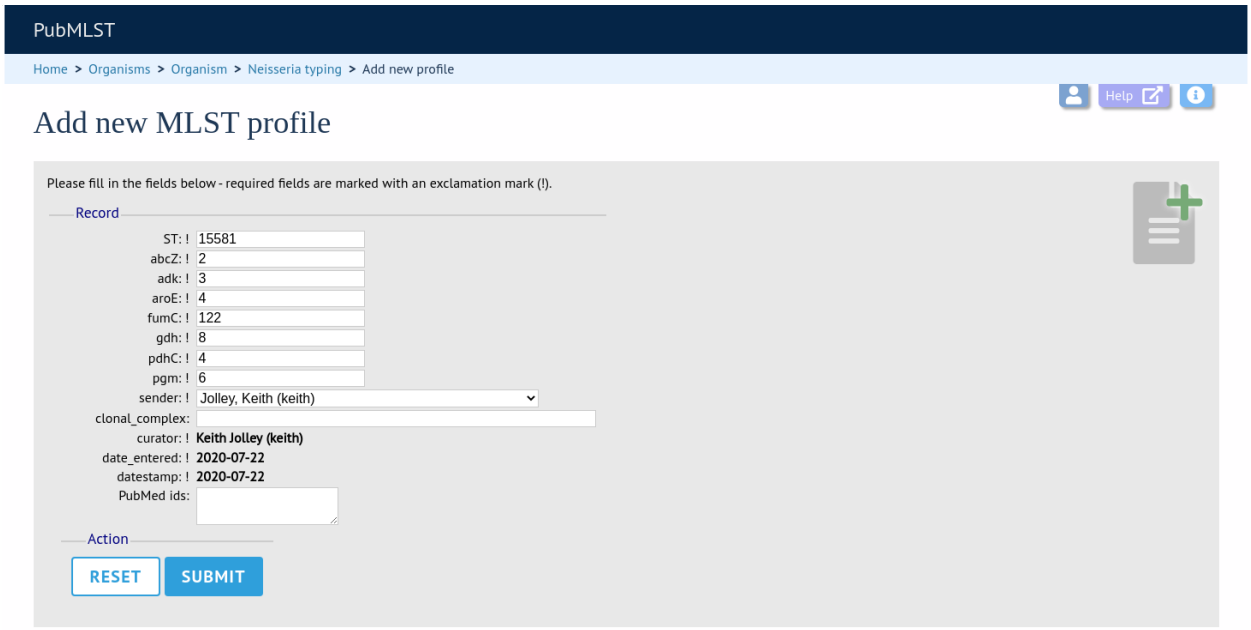
6.7 Adding new scheme profile definitions

Provided a scheme has been set up with at least one locus and a scheme field set as a primary key, there will be links on the curator's main page to add profiles for that scheme.

To add a single profile you can click the add (+) profiles link in the box named after the scheme name (e.g. MLST):

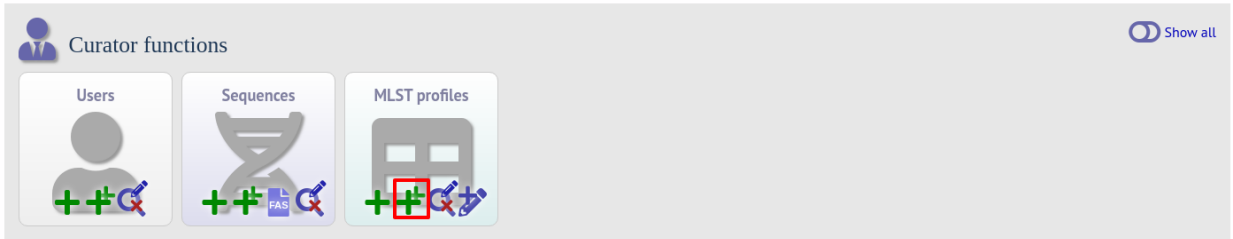


A form will be displayed with the next available primary key number already entered (provided integers are used for the primary key format). Enter the new profile, associated scheme fields, and the sender, then click 'Submit'. The new profile will be added provided the primary key or the profile has not previously been entered.

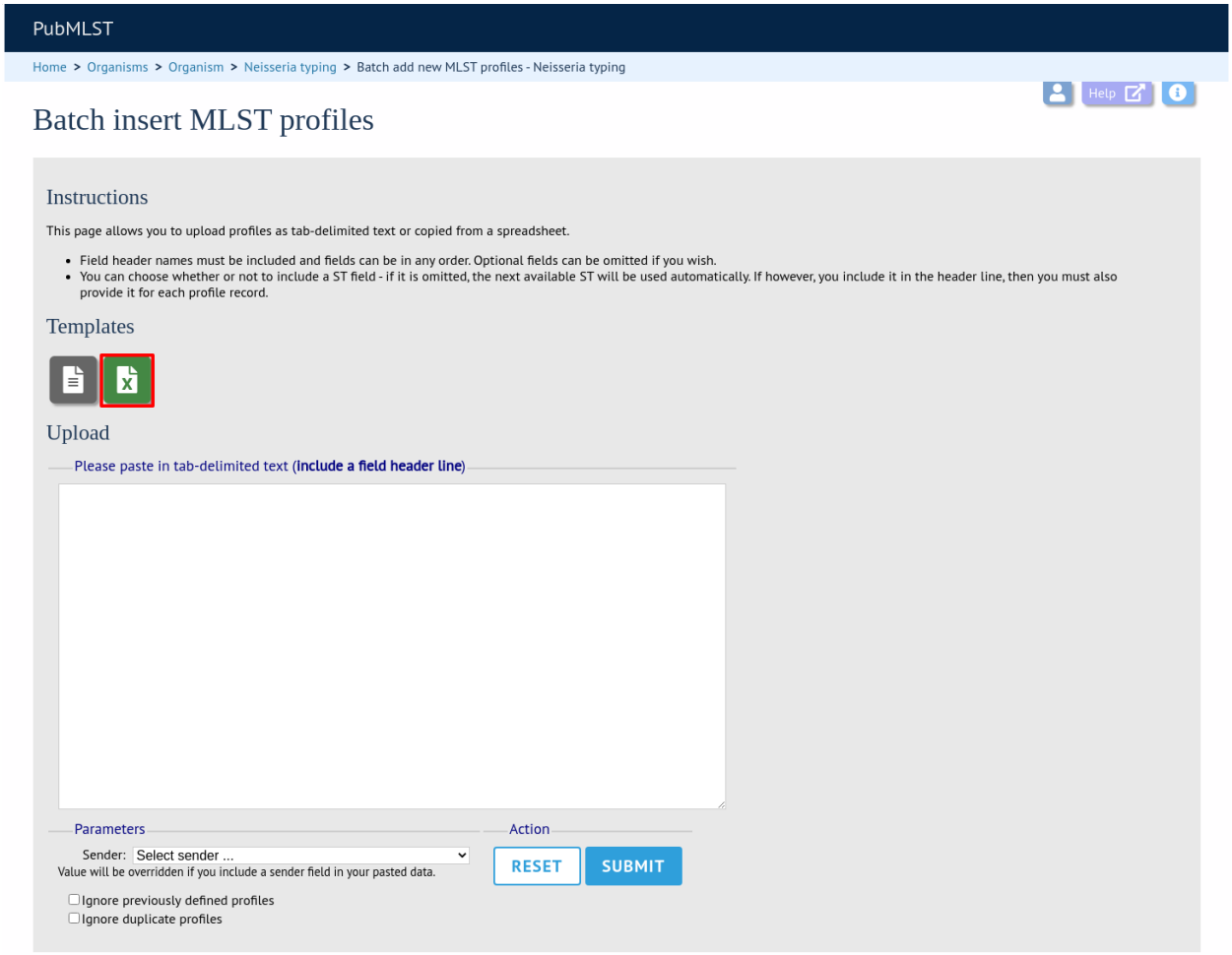


More usually, profiles are added in a batch mode. It is often easier to do this even for a single profile since it allows copying and pasting data from a spreadsheet.

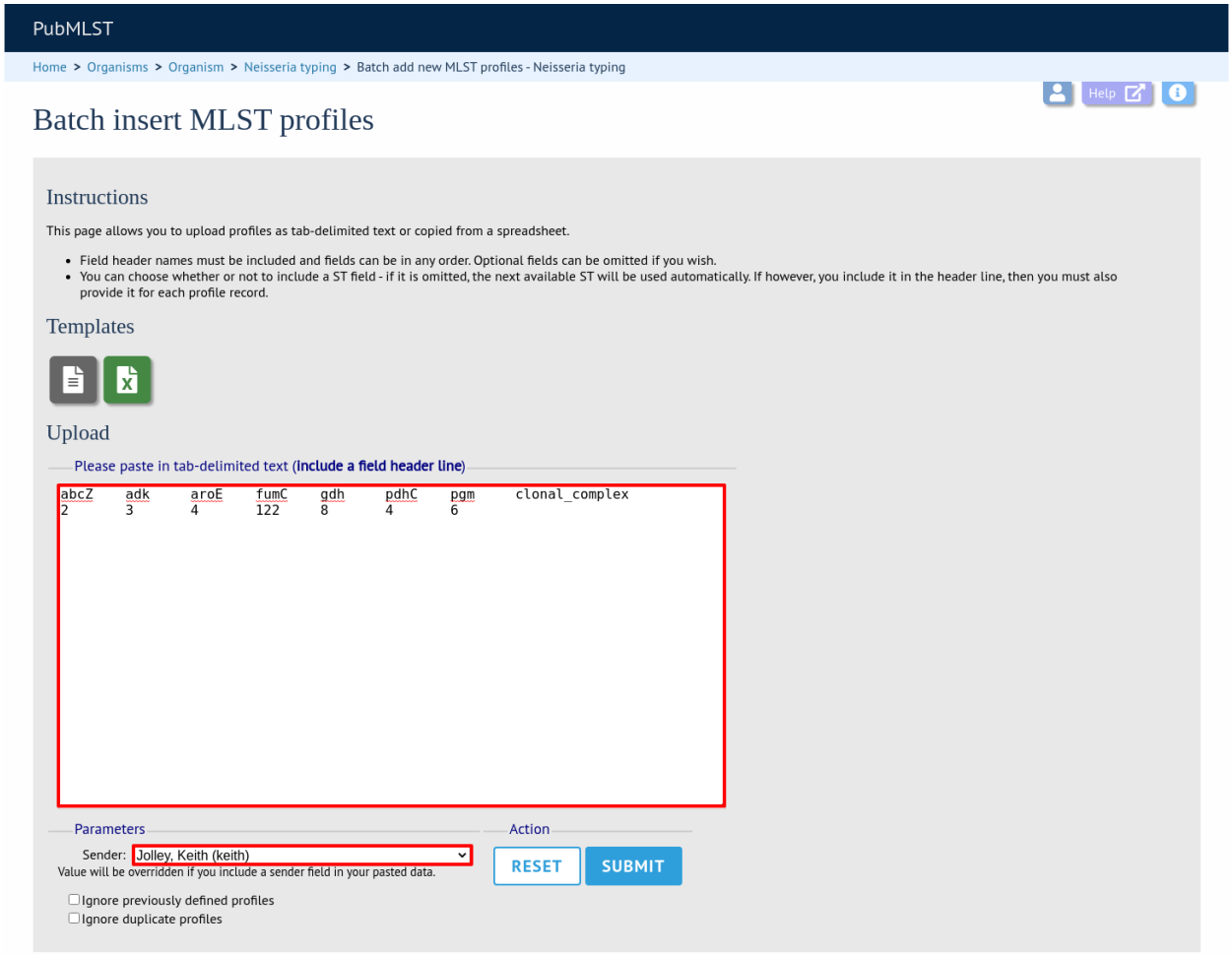
Click the batch add (++) profiles link next to the scheme name:



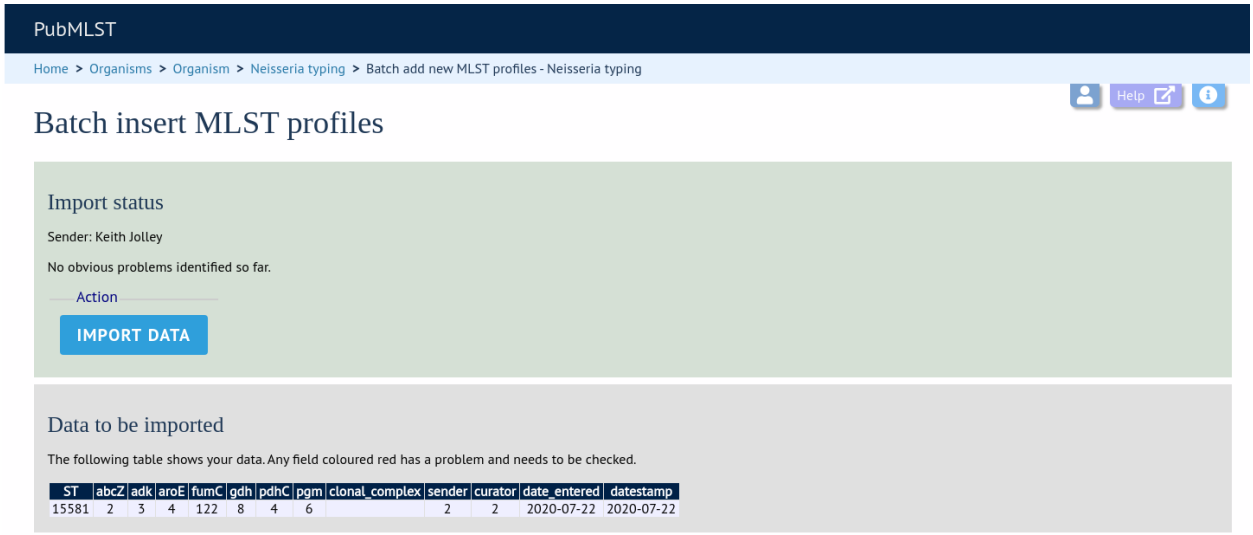
Click the 'Download submission template (xlsx format)' link to download an Excel submission template.



Fill in the spreadsheet using the copied template, then copy and paste the whole spreadsheet in to the large form on the upload page. If the primary key has an integer format, you can exclude this column and the next available number will be used automatically. If the column is included, however, a value must be set. Select the sender from the dropdown list box and then click 'Submit'.

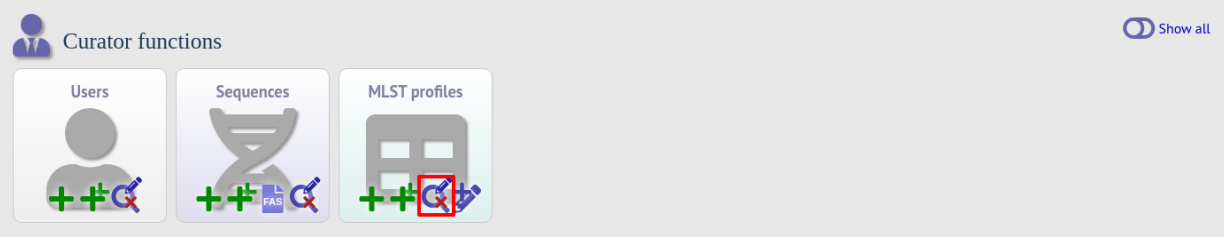


You will be given a final confirmation page stating what will be uploaded. If you wish to proceed with the submission, click 'Import data'.

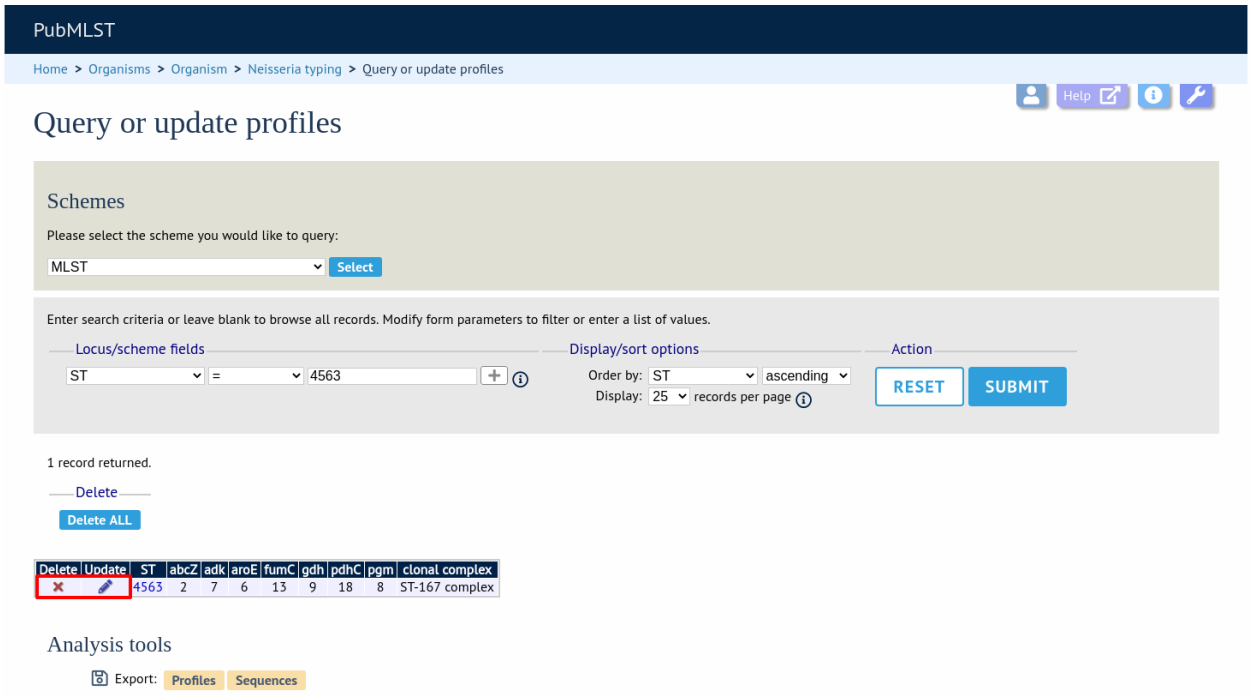


6.8 Updating and deleting scheme profile definitions

In order to update or delete a scheme profile, first you must select it. Click the update/delete profiles link in the scheme profiles box named after the scheme (e.g. MLST):

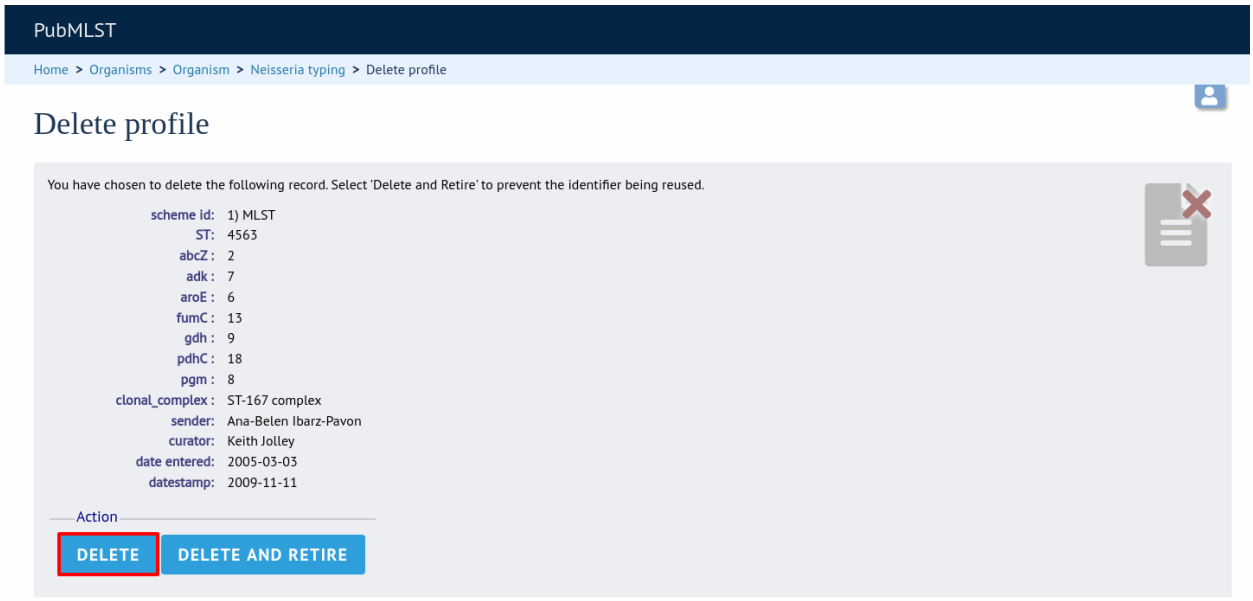


Search for your profile by entering search criteria (alternatively you can use the browse or list query functions).

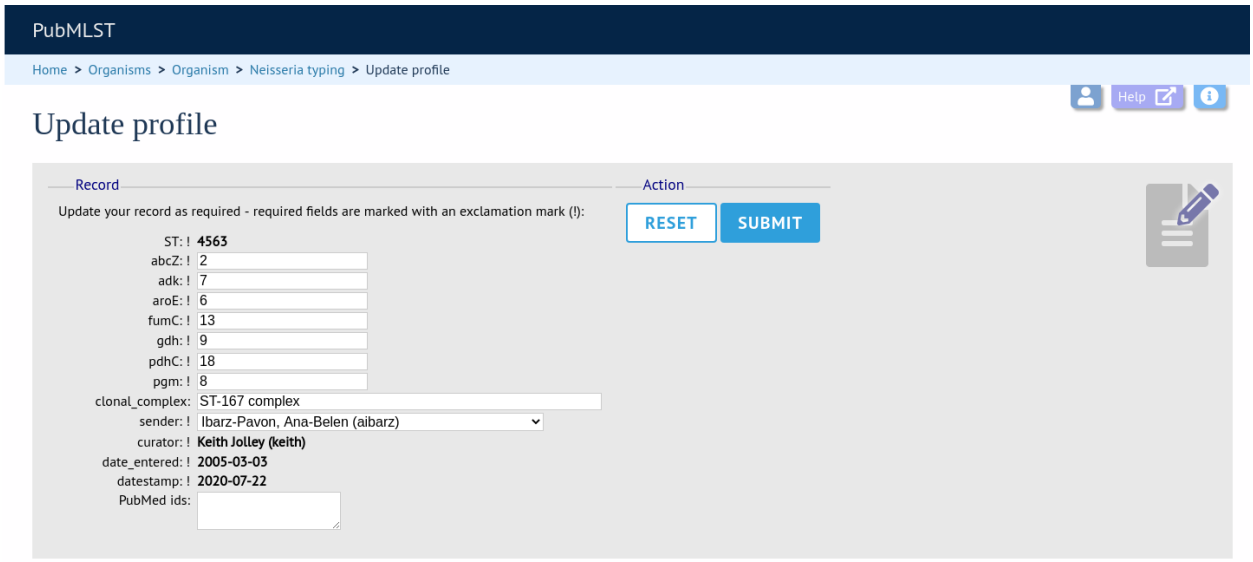


To delete the profile, click the 'Delete' link next to the profile. Alternatively, if your account has permission, you may be able to 'Delete ALL' records retrieved from the search.

For deletion of a single record, the full record will be displayed. Confirm deletion by clicking 'Delete'. You can also choose to delete and retire the profile identifier. If you do this, the profile identifier will not be re-used. The database configuration can be set so that you can only delete and retire.



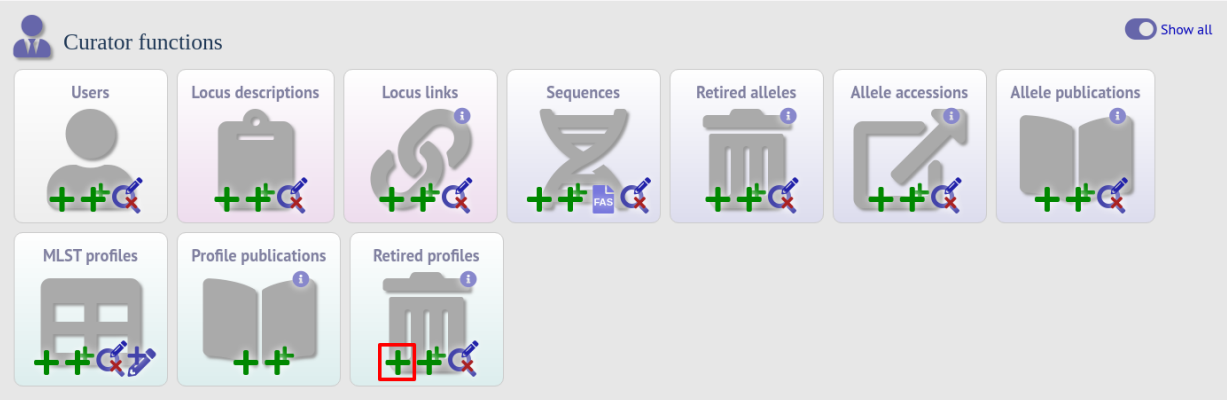
To modify the profile, click the 'Update' link next to the profile following the query. A form will be displayed - make any changes and then click 'Update'.



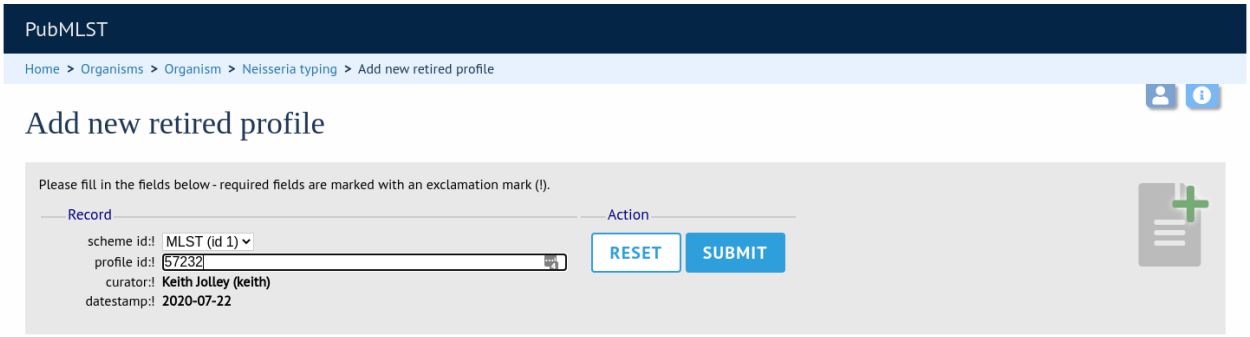
6.9 Retiring scheme profile identifiers

Sometimes there is a requirement to prevent the automated assignment of a particular profile identifier (e.g. ST) - a profile with that identifier may have been commonly used and has since been removed. Reassignment of the identifier to a new profile may lead to confusion, so in this instance, it would be better to prevent this.

You can retire a profile identifier by clicking the 'Add' link in the 'Retired profiles' box on the sequence database curators' page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Select the scheme from the dropdown list box and enter the profile id. Click 'Submit'.



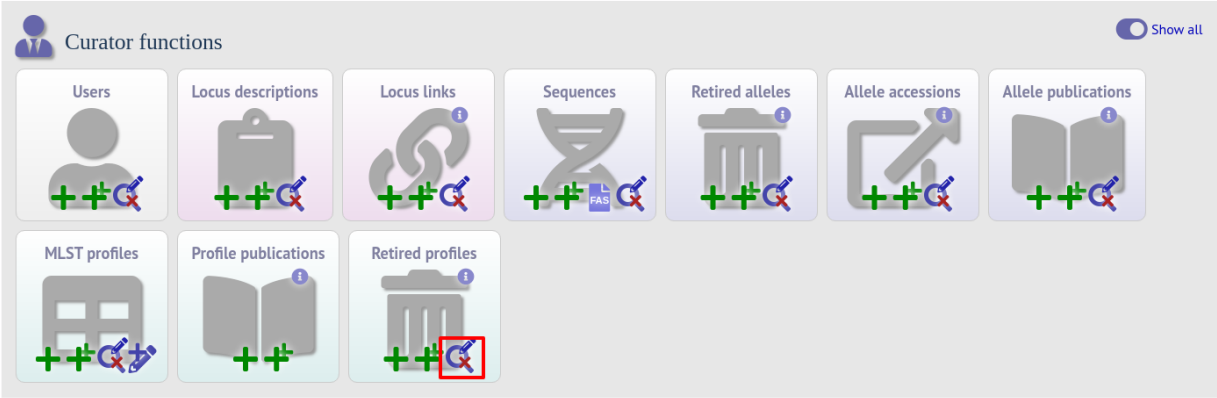
You cannot retire a profile identifier that already exists, so you must delete it before retiring it. Once an identifier is retired, you will not be able to create a new profile with that name.

You can also retire a profile definition when you *delete a profile*.

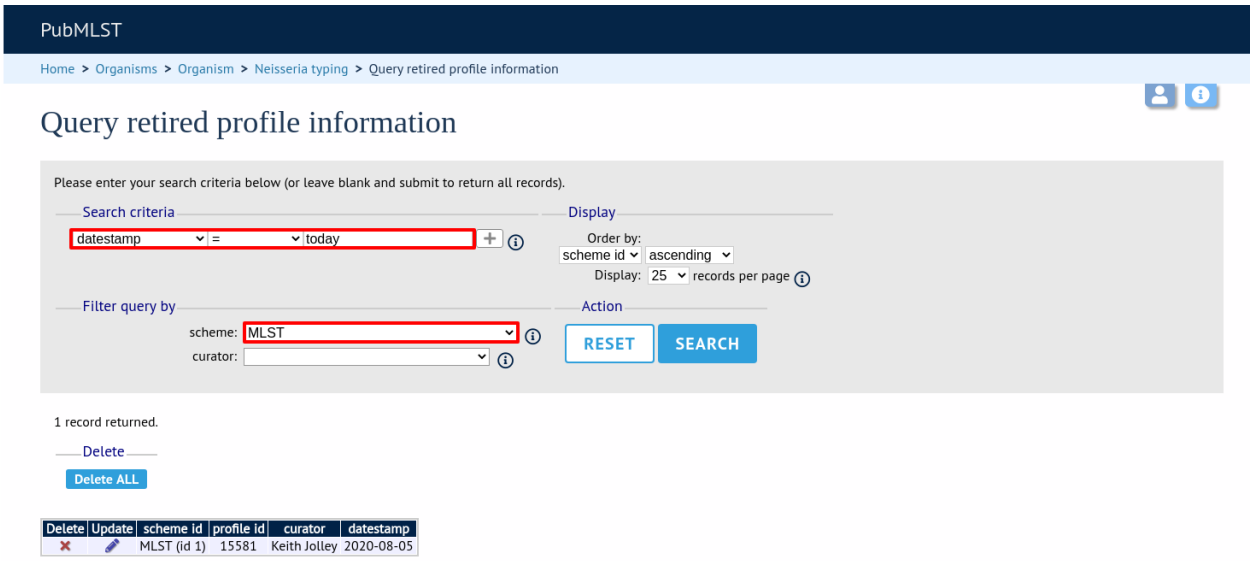
6.10 Un-retiring scheme profile identifiers

If a profile identifier, e.g. ST, has been retired, it is possible to un-retire it so that it can be re-assigned. To do this, you need to remove the identifier from the retired_profiles table.

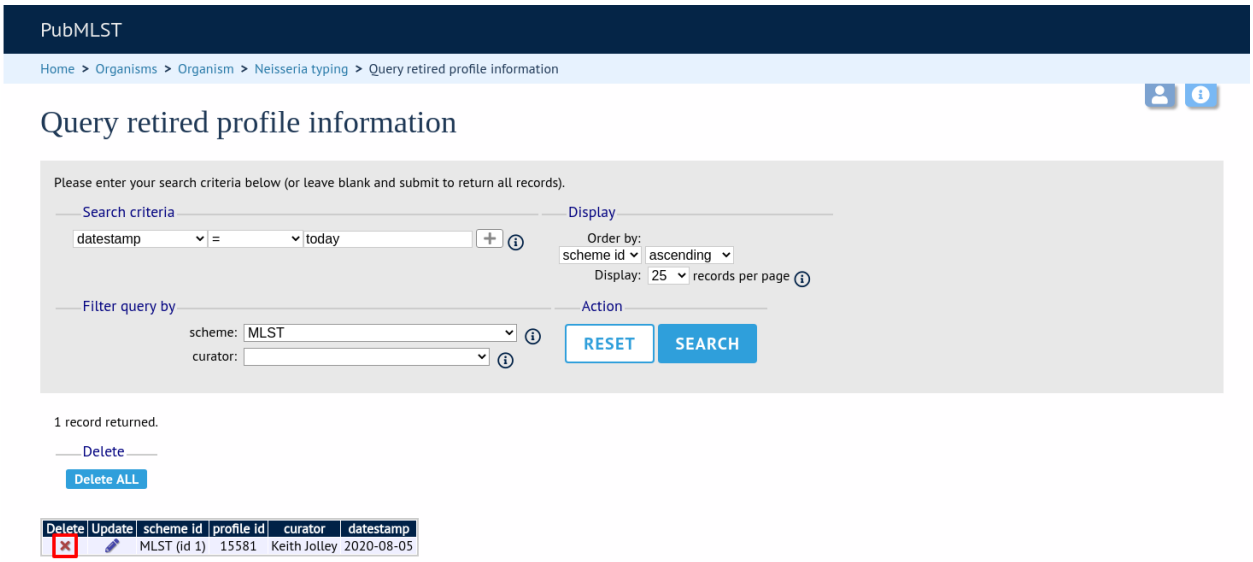
First, find the profile identifier in the retired_profiles table by clicking the 'Update/delete' retired profiles link on the sequence database curators' page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



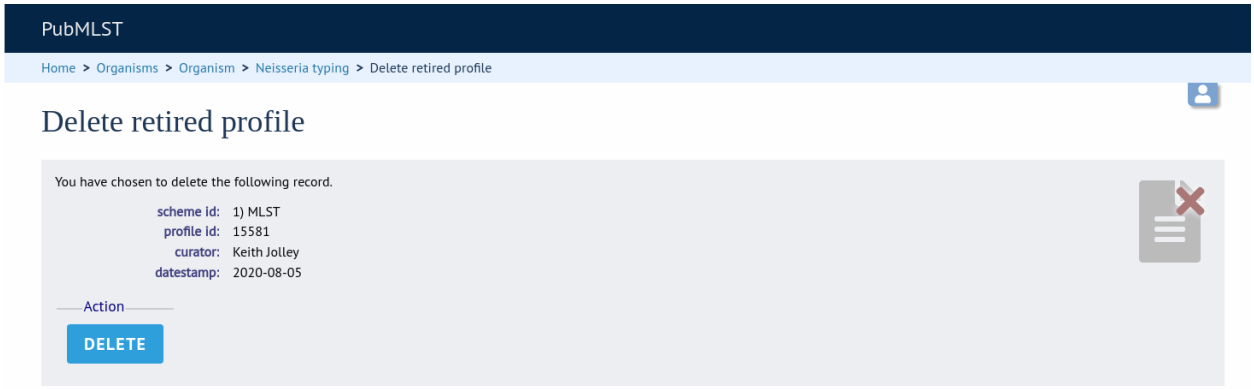
Search by any criteria to find the profile identifier.



Click the delete link on the identifer to be un-retired.



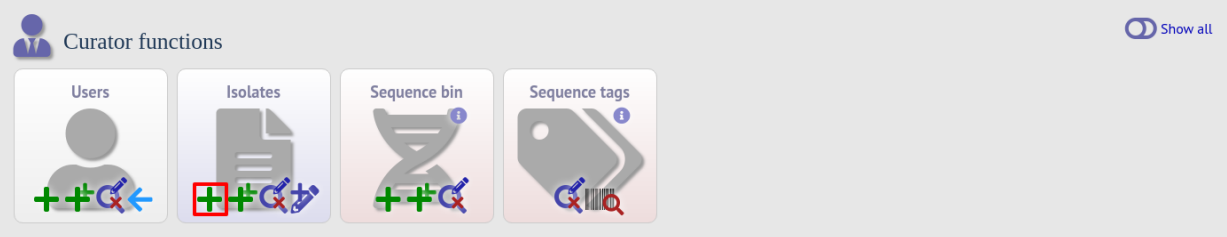
A confirmation page will be displayed. Click 'Delete' to remove the identifier from the retired profiles table.



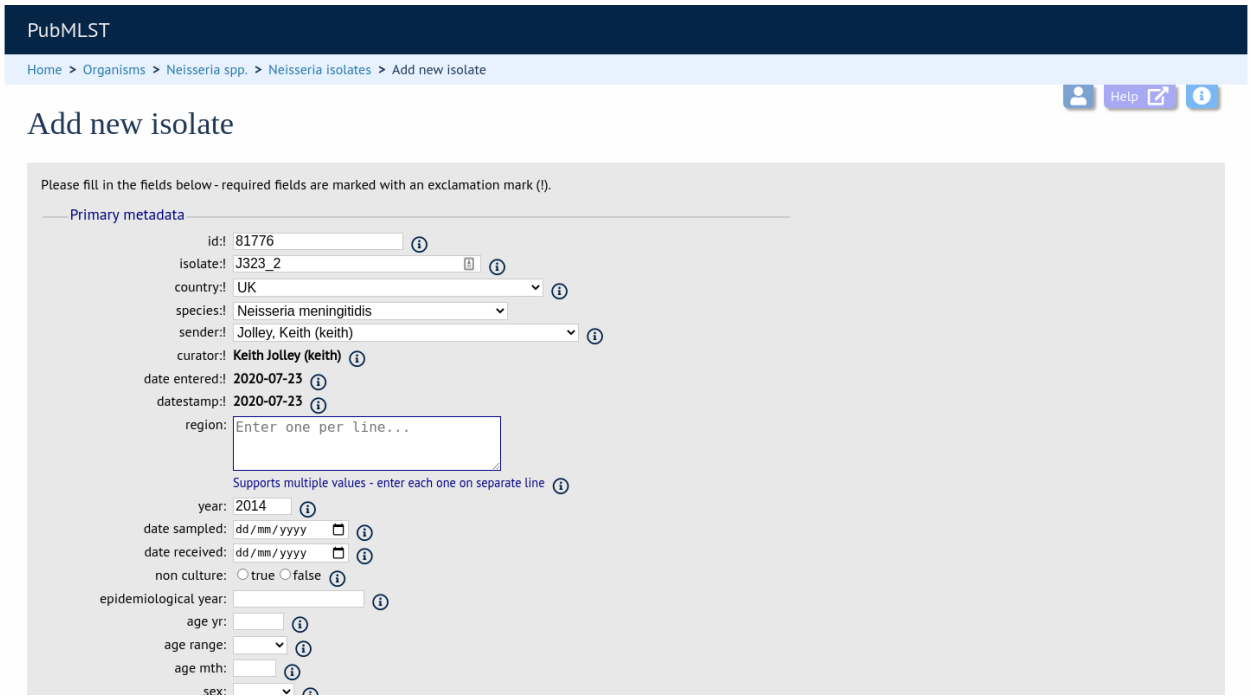
The identifier can now be re-assigned when adding a new profile.

6.11 Adding isolate records

To add a single record, click the add (+) isolates link on the curator's index page.



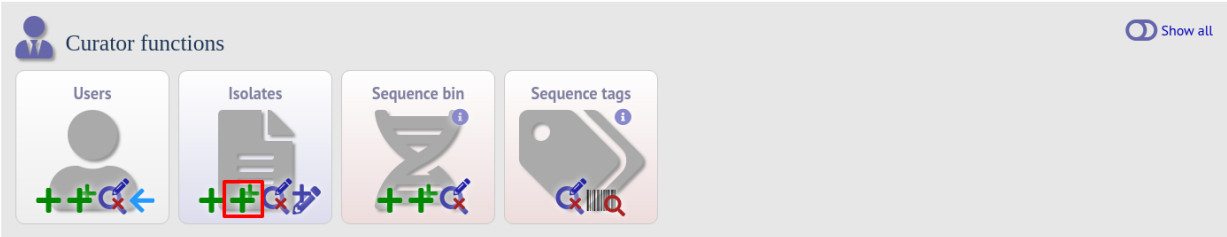
The next available id will be filled in automatically but you are free to change this. Fill in the individual fields. Required fields are listed first and are marked with an exclamation mark (!). Some fields may have drop-down list boxes of allowed values. You can also enter allele designations for any loci that have been defined.



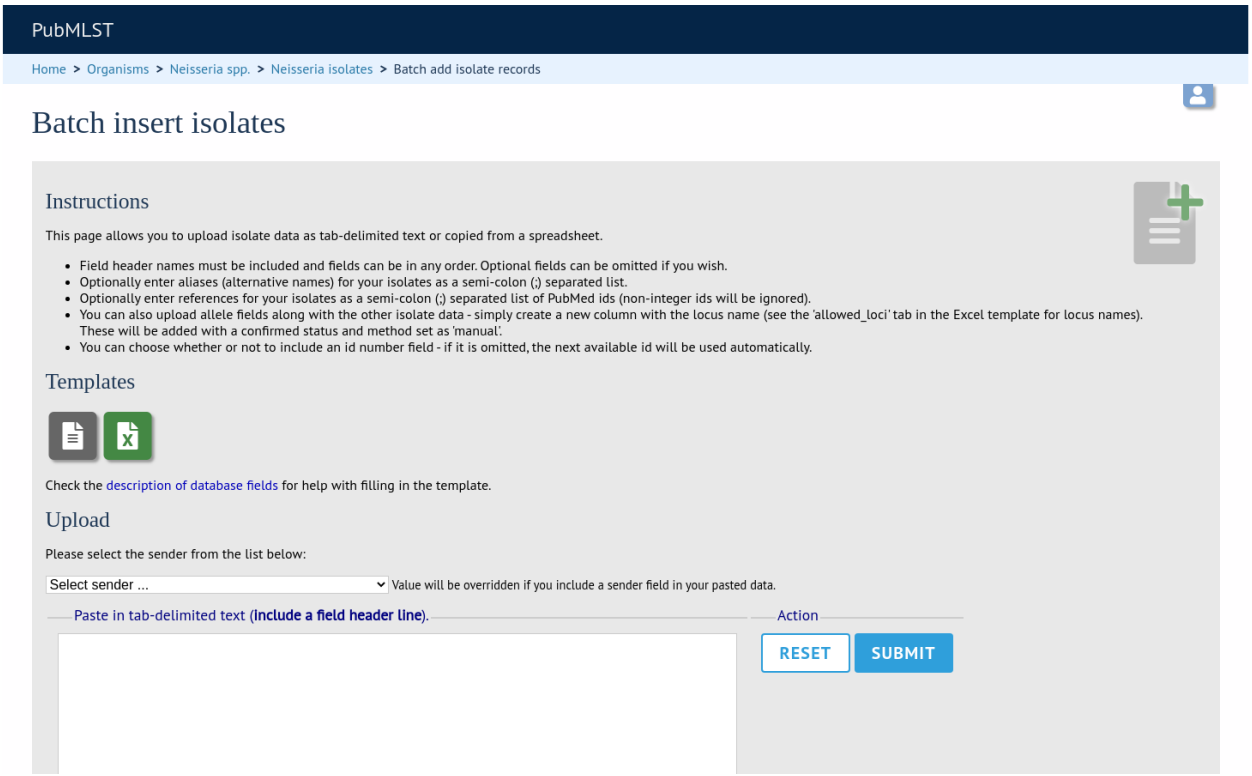
Press submit when finished.

More usually, isolate records are added in batch mode, even when only a single record is added, since the submission can be prepared in a spreadsheet and copied and pasted.

Select batch add (++) isolates link on the curator's index page.

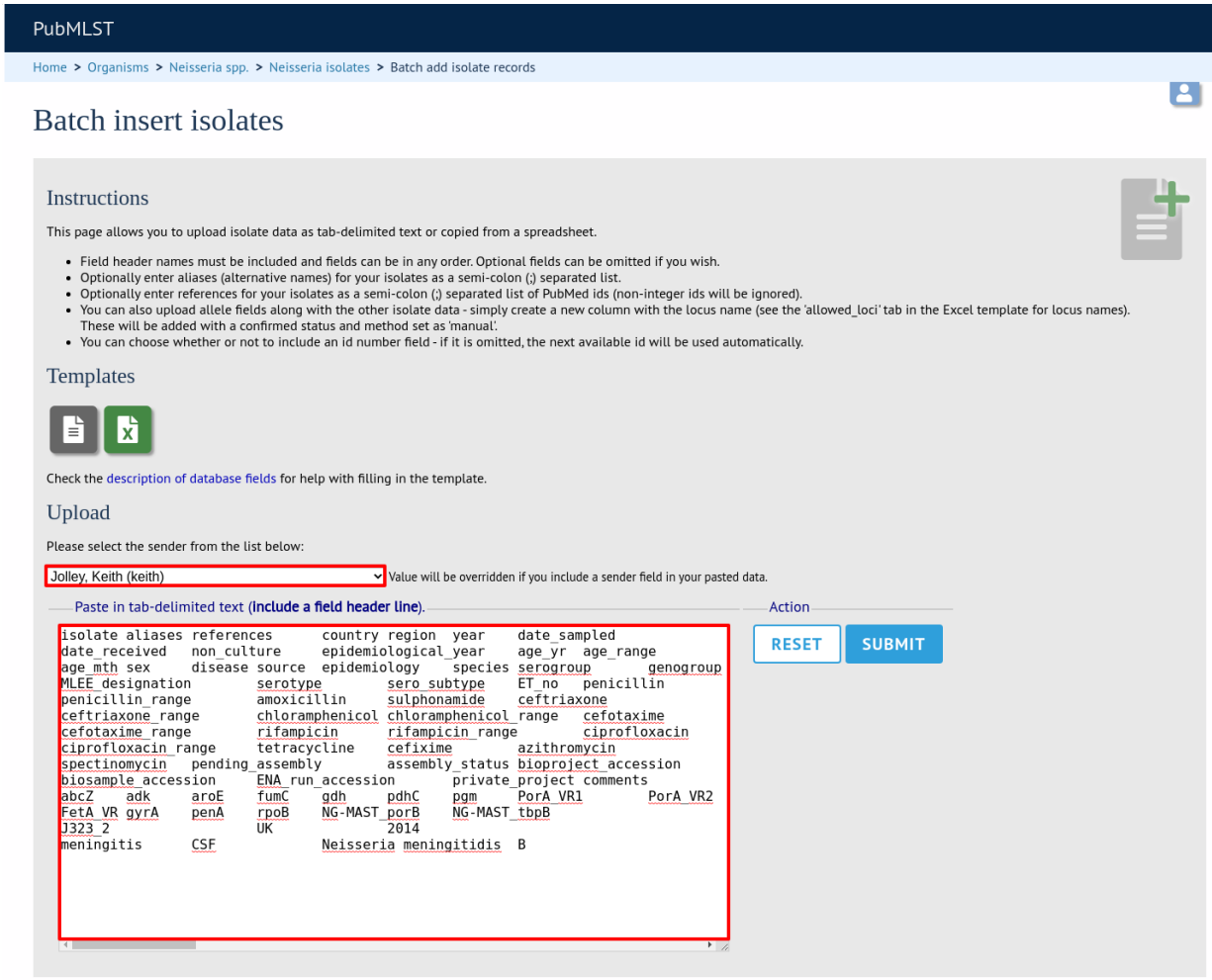


Download a submission template in Excel format from the link.



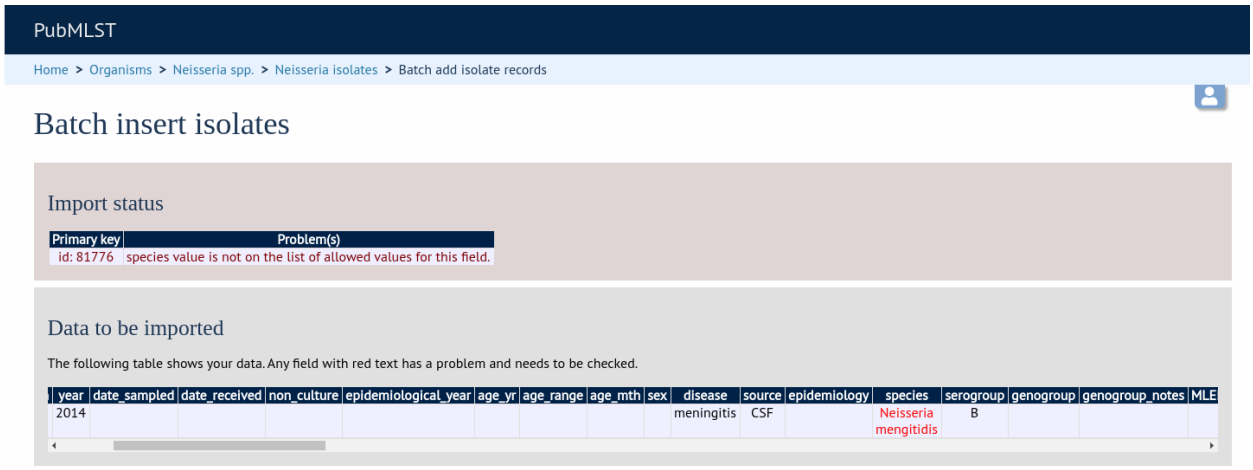
Prepare your data in the spreadsheet - the column headings must match the database fields. In databases with large numbers of loci, there won't be columns for each of these. You can, however, manually add locus columns.

Pick a sender from the drop-down list box and paste the data from your spreadsheet in to the web form. The next available isolate id number will be used automatically (this can be overridden if you manually add an id column).



Press submit. Data are checked for consistency and if there are no problems you can then confirm the submission.

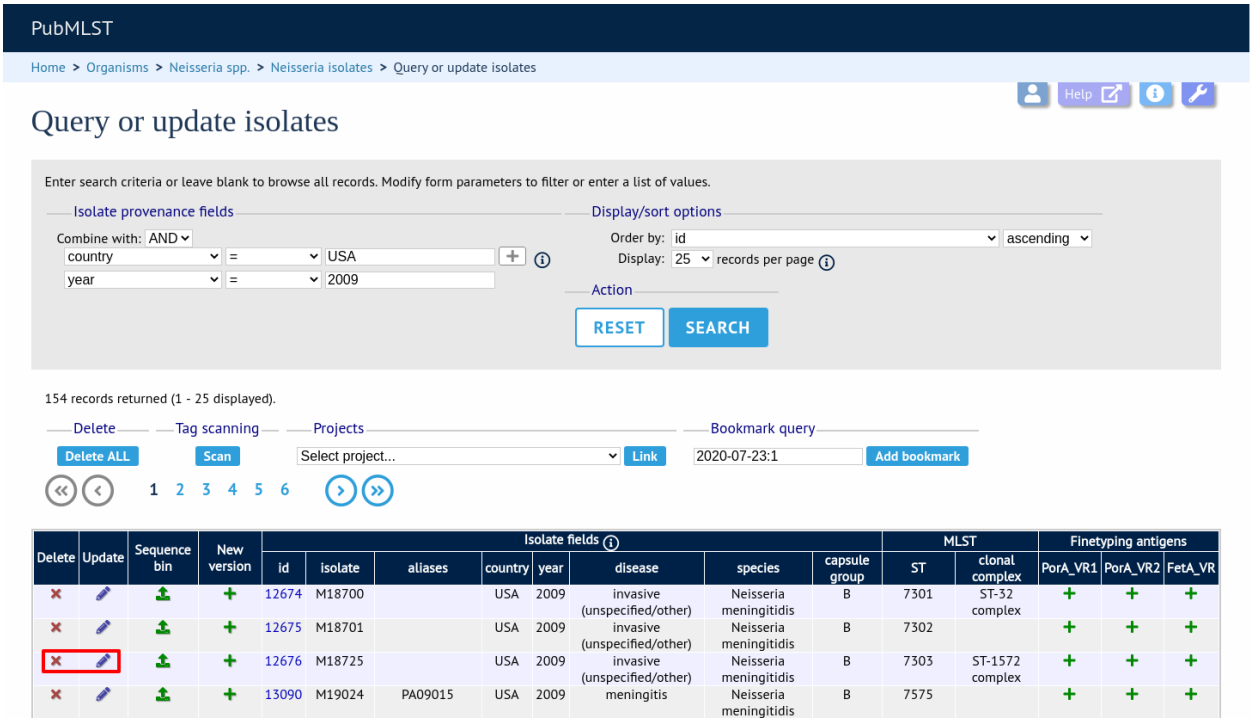
Any problems with the data will be listed and highlighted within the table. Fix the data and resubmit if this happens.



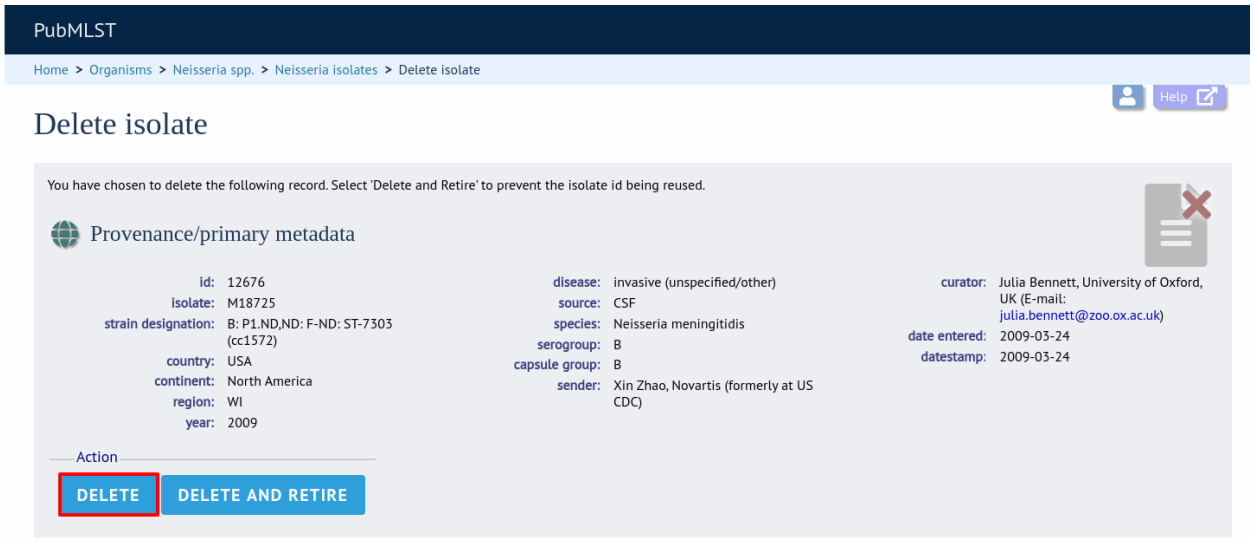
6.12 Updating and deleting single isolate records

First you need to locate the isolate record. You can either browse or use a search or list query.

The query interface is the same as the *public query interface*. Following a query, a results table of isolates will be displayed. There will be delete and update links for each record.

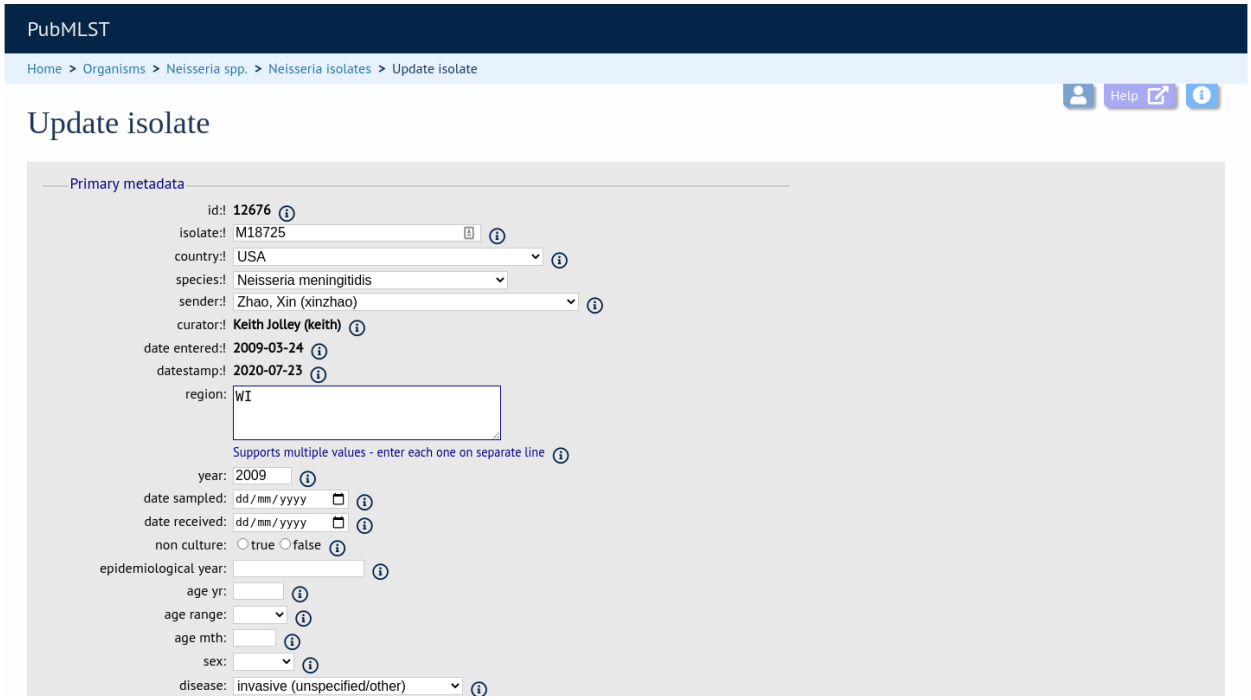


Clicking the 'Delete' link takes you to a page displaying the full isolate record.

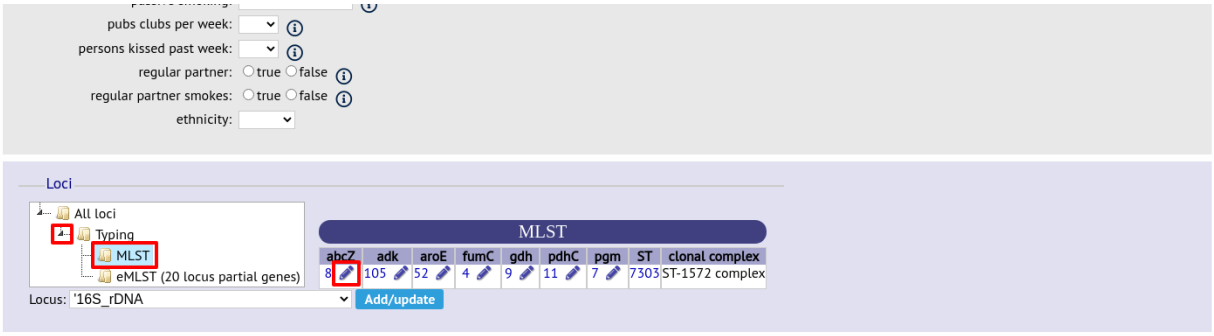


Pressing 'Delete' from this record page confirms the deletion. You can also choose to delete and retire the isolate. If you do this, the isolate id number will not be re-used. It is possible to set the configuration so that you only have the option to delete and retire.

Clicking the 'Update' link for an isolate takes you to an update form. Make the required changes and click 'Update'.



Allele designations can also be updated by clicking within the scheme tree and selecting the 'Add' or 'Update' link next to a displayed locus.



Schemes will only appear in the tree if data for at least one of the loci within the scheme has been added. You can additionally add or update allelic designations for a locus by choosing a locus in the drop-down list box and clicking 'Add/update'.

The allele designation update page allows you to modify an existing designation, or alternatively add additional designations. The sender, status (confirmed/provisional) and method (manual/automatic) needs to be set for each designation (all pending designations have a provisional status). The method is used to differentiate designations that have been determined manually from those determined by an automated algorithm.

6.13 Batch updating multiple isolate records

Select 'batch update' isolates link on the curator's index page.

Prepare your update data in 3 columns in a spreadsheet:

1. Unique identifier field
2. Field to be updated
3. New value for field

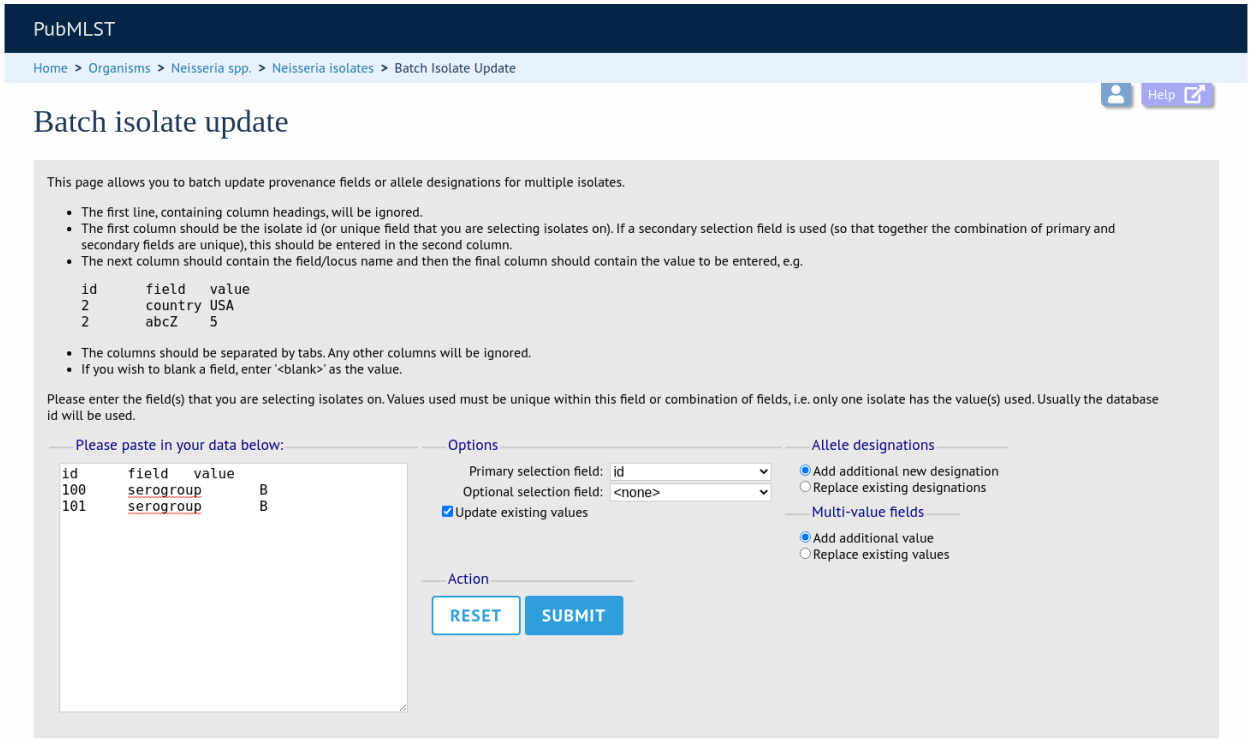
You should also include a header line at the top - this isn't used so can contain anything but it should be present.

Columns must be tab-delimited which they will be if you copy and paste directly from the spreadsheet.

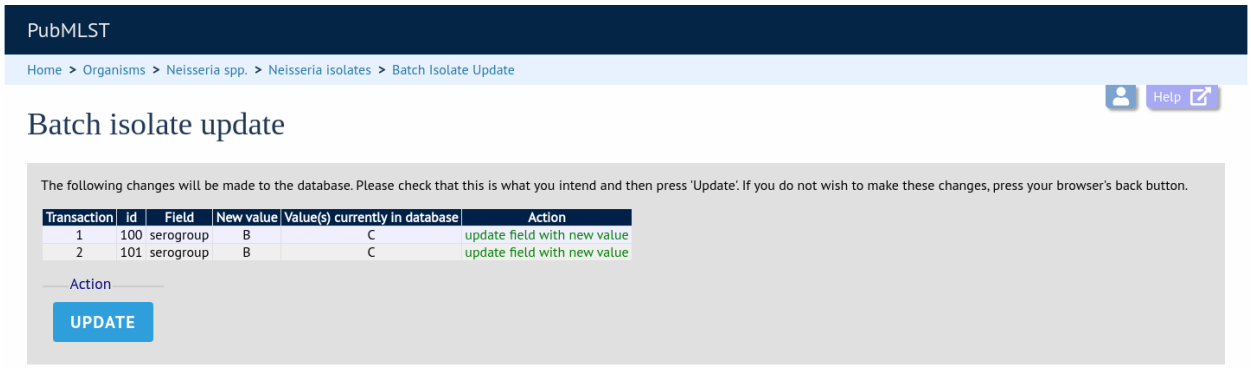
So, to update isolate id-100 and id-101 to serogroup B you would prepare the following:

```
id      field      value
100     serogroup  B
101     serogroup  B
```

Select the field you are using as a unique identifier, in this case id, from the drop-down list box, and paste in the data. If the fields already have values set, you should also check the 'Update existing values' checkbox. Press 'submit'.



A confirmation page will be displayed if there are no problems. If there are problems, these will be listed. Press 'Upload' to upload the changes.



You can also use a secondary selection field such that a combination of two fields uniquely defines the isolate, for example using country and isolate name.

So, for example, to update the serogroups of isolates CN100 and CN103, both from the UK, select the appropriate primary and secondary fields and prepare the data as follows:

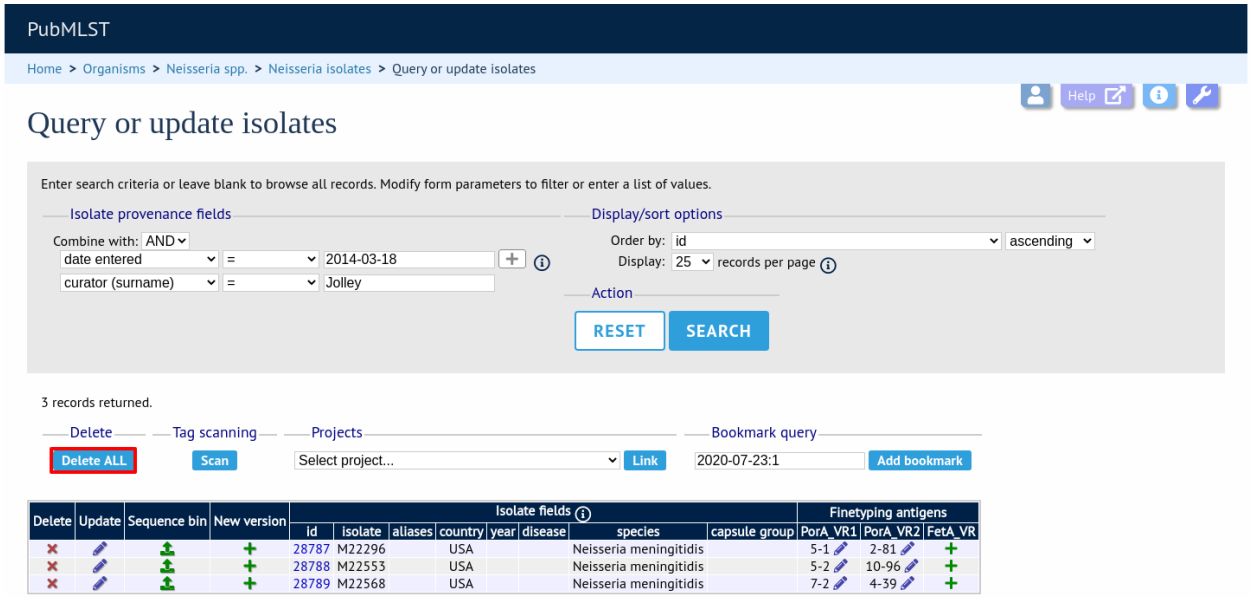
```
isolate     country     field       value
CN100       UK          serogroup   B
CN103       UK          serogroup   B
```

6.14 Deleting multiple isolate records

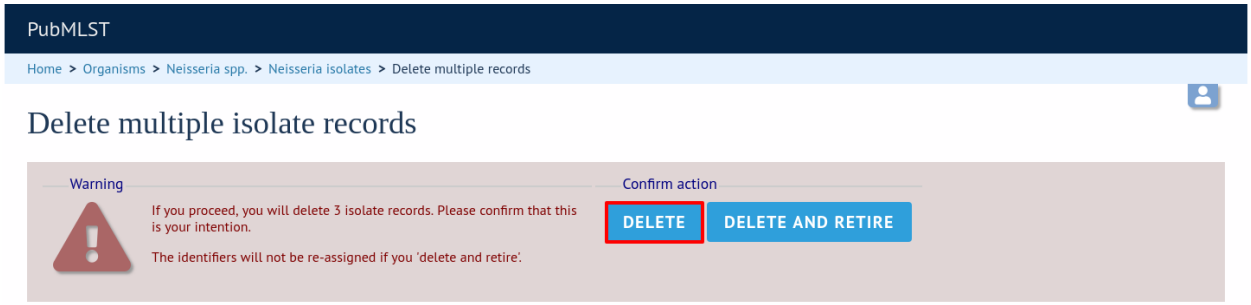
Note: Please note that standard curator accounts may not have permission to delete multiple isolates. Administrator accounts are always able to do this.

Before you can delete multiple records, you need to search for them. From the curator's main page, click the update/delete isolates link:

Enter search criteria that specifically return the isolates you wish to delete. Click 'Delete ALL'.



You will have a final chance to change your mind:

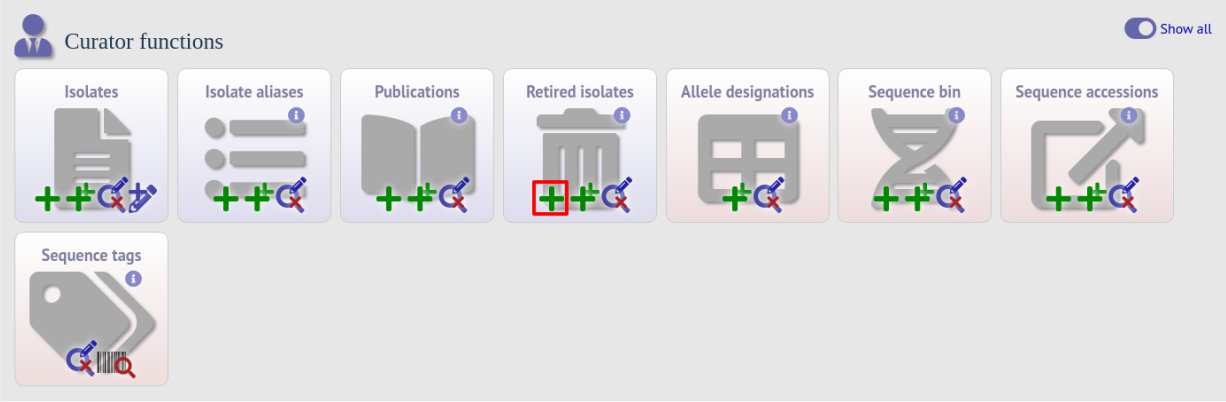


Click 'Delete'. You can also choose to delete and retire the isolate id. If you do this, the id number will not be re-used. It is possible to set the configuration so that you only have the option to delete and retire.

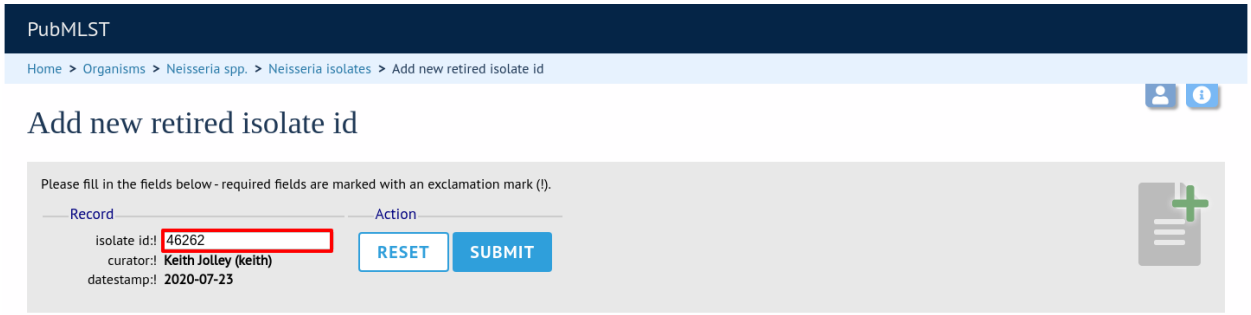
6.15 Retiring isolate identifiers

Sometimes there is a requirement to prevent the automated assignment of a particular isolate identifier number - an isolate with that identifier may have been commonly referred to and has since been removed. Reassignment of the identifier to a new isolate record may lead to confusion, so in this instance, it would be better to prevent this.

You can retire an isolate identifier by clicking the 'Add' retired isolates link on the isolates database curators' page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Enter the isolate id to retire and click 'Submit'.



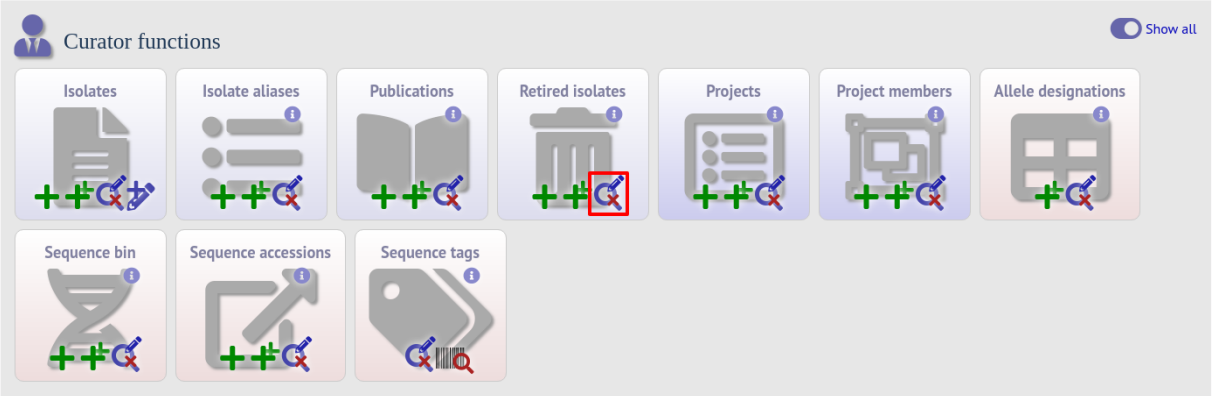
You cannot retire an isolate identifier that already exists, so you must delete it before retiring it. Once an identifier is retired, you will not be able to create a new isolate record using that identifier.

You can also retire an isolate identifier when you *delete an isolate record*.

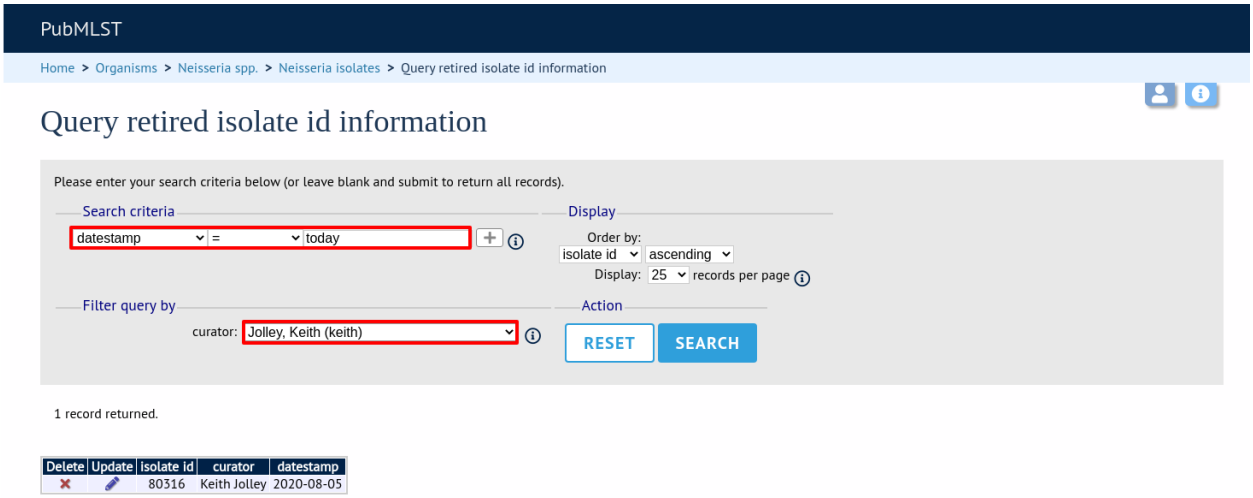
6.16 Un-retiring isolate identifiers

If an isolate identifier has been retired, it is possible to un-retire it so that it can be re-assigned. To do this, you need to remove the identifier from the retired_isolates table.

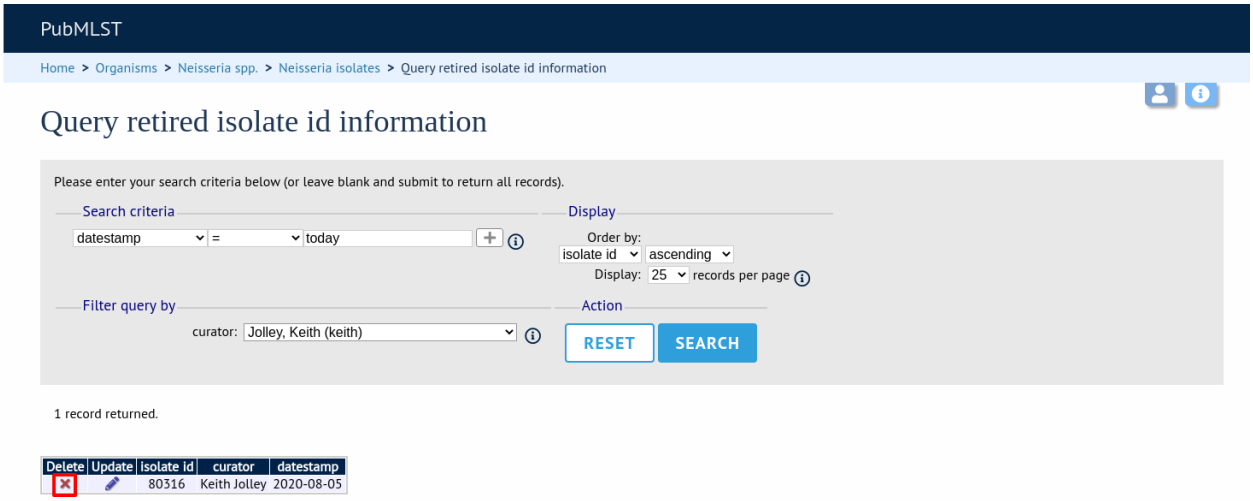
First, find the isolate identifier in the retired_isolates table by clicking the 'Update/delete' retired isolates link on the isolate database curators' page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Search by any criteria to find the isolate identifier.



Click the delete link on the identifer to be un-retired.



A confirmation page will be displayed. Click 'Delete' to remove the identifier from the retired isolates table.

The identifier can now be re-assigned when adding a new isolate record.

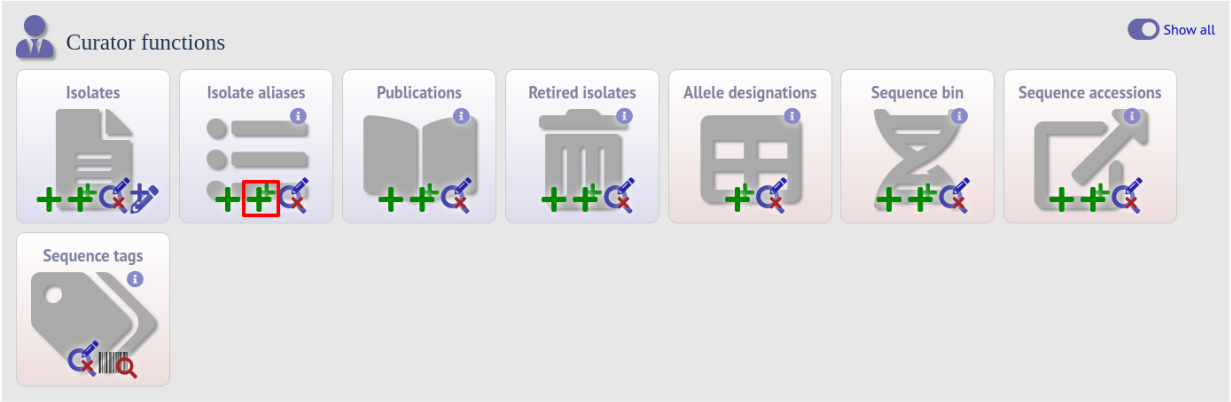
6.17 Setting alternative names for isolates (aliases)

Isolates can have any number of alternative names that they are known by. These isolate aliases can be set when isolates are first added to the database or batch uploaded later. When querying by isolate names, the aliases are also searched automatically.

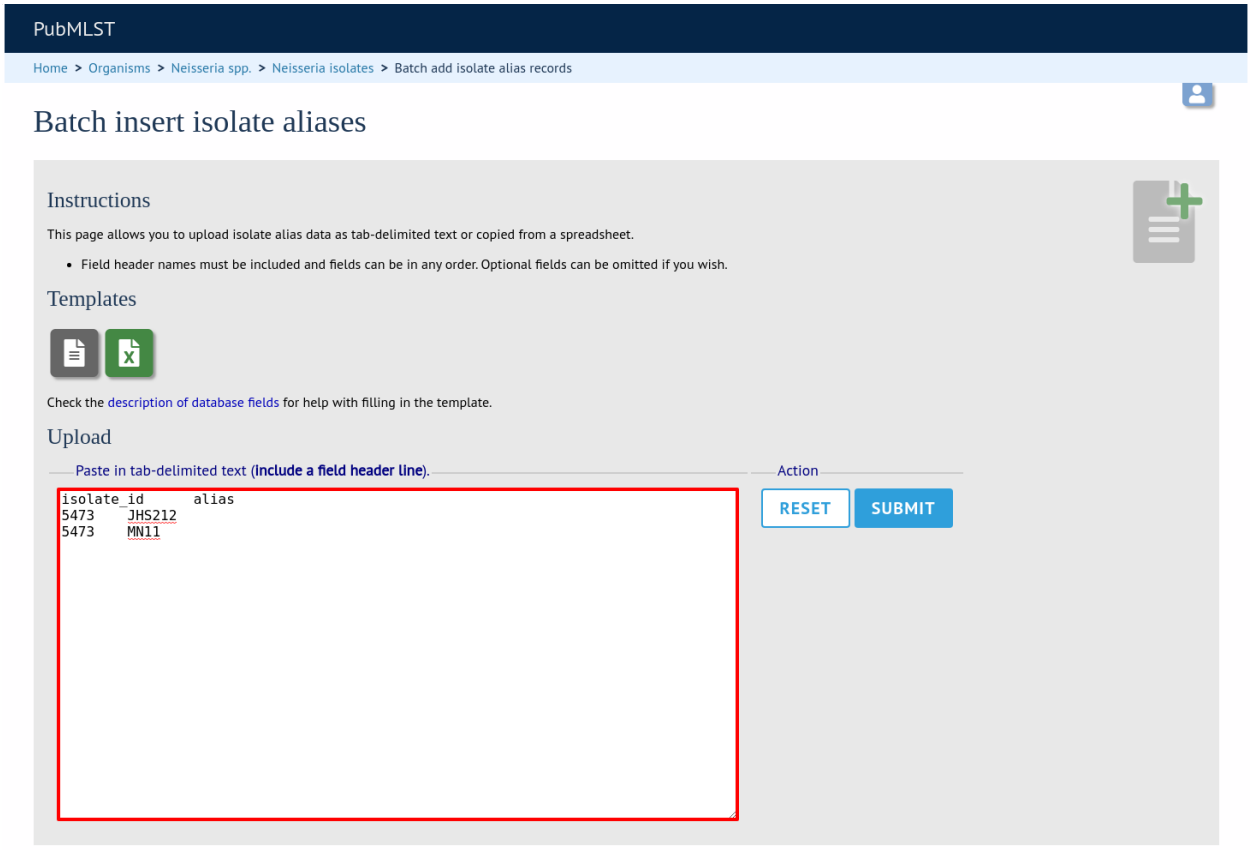
If *adding isolates* singly, add the aliases in to the aliases box (one alias per line):

If batch adding isolates, they can be entered as a semi-colon (;) separated list in the aliases column.

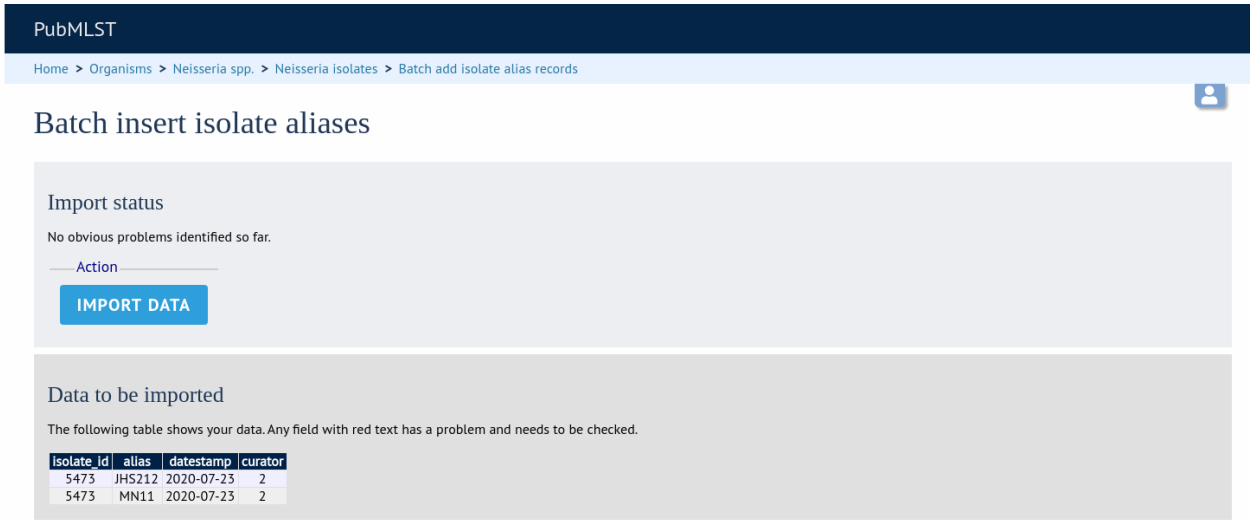
As stated above, aliases can also be batch added. To do this, click the batch add (++) isolate aliases link on the curator's index page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Prepare a list in a spreadsheet using the provided template. This consists of two columns: isolate_id and alias. For example, to add the aliases 'JHS212' and 'NM11' to isolate id 5473, the values to paste in look like:



A confirmation page will be displayed.

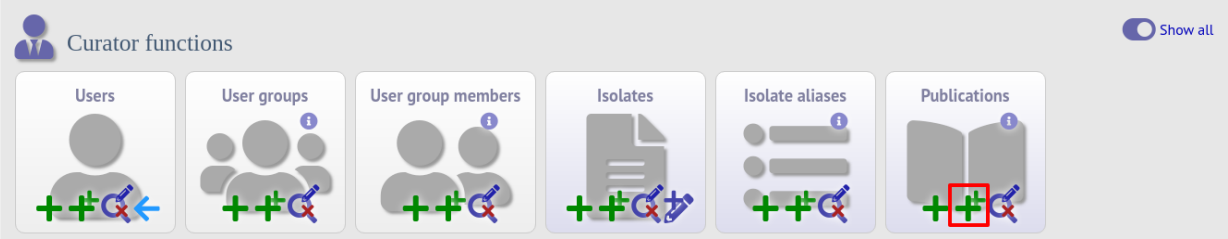


Click ‘Import data’.

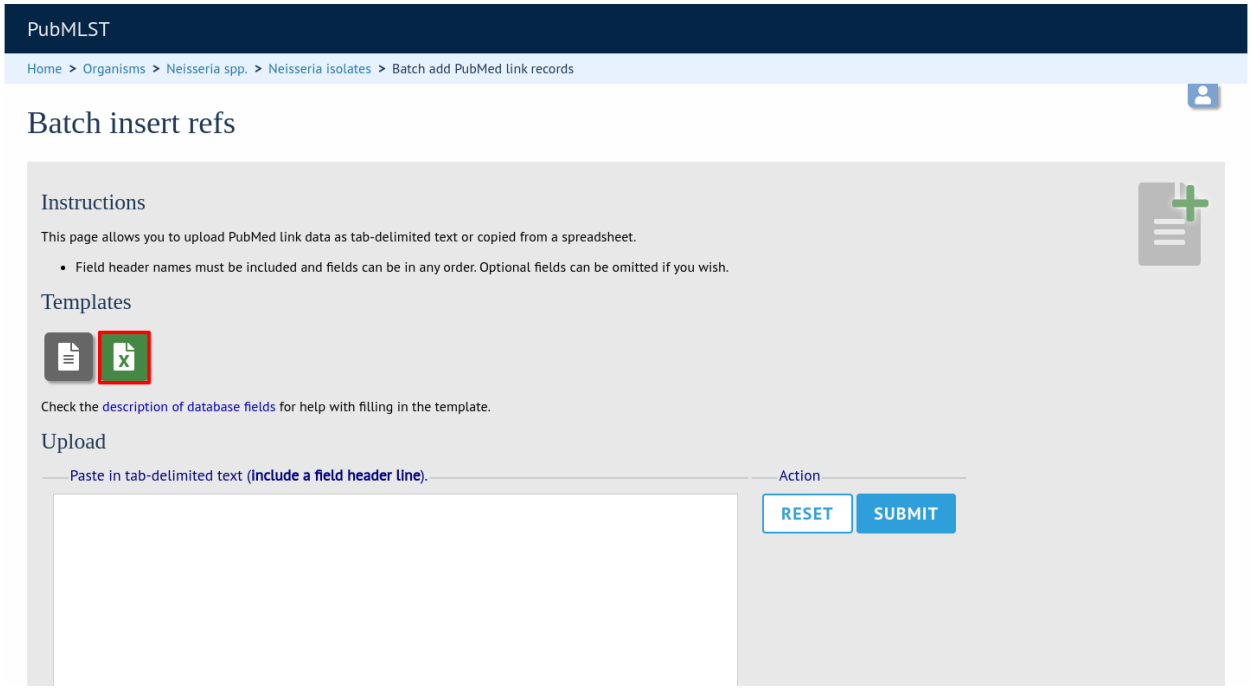
6.18 Linking isolate records to publications

Isolates can be associated with publications by adding PubMed id(s) to the record. This can be done when *adding the isolate*, where lists of PubMed ids can be entered in to the web form.

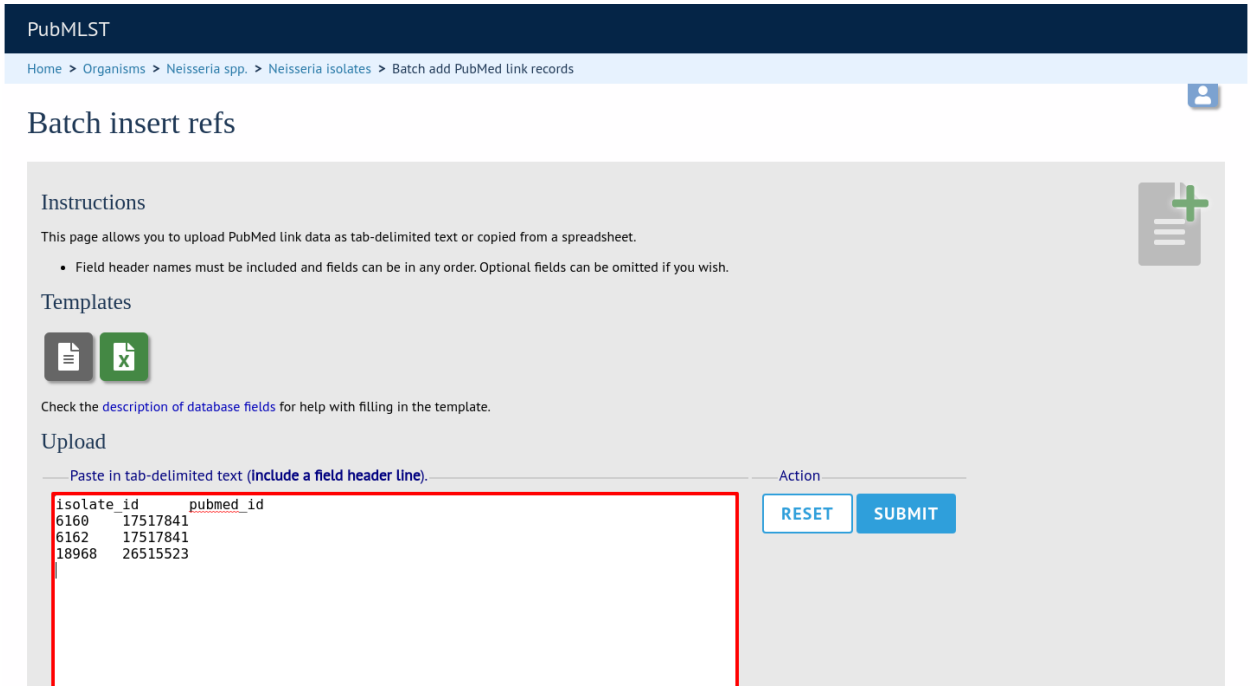
They can also be associated in batch after the upload of isolate records. Click the PubMed batch add (++) link on the curator's main page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Open the Excel template by clicking the link.



The Excel template has two columns, isolate_id and pubmed_id. Simply fill this in with a line for each record and then paste the entire spreadsheet in to the web form and press submit.

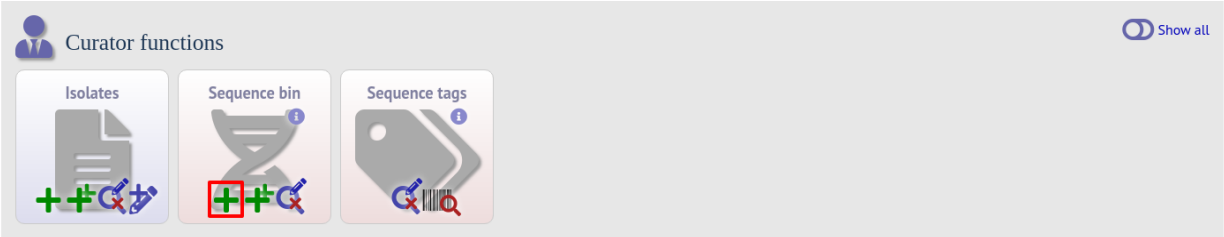


To ensure that publication information is stored locally and available for searching, the references database needs to be *updated regularly*.

6.19 Uploading sequence contigs linked to an isolate record

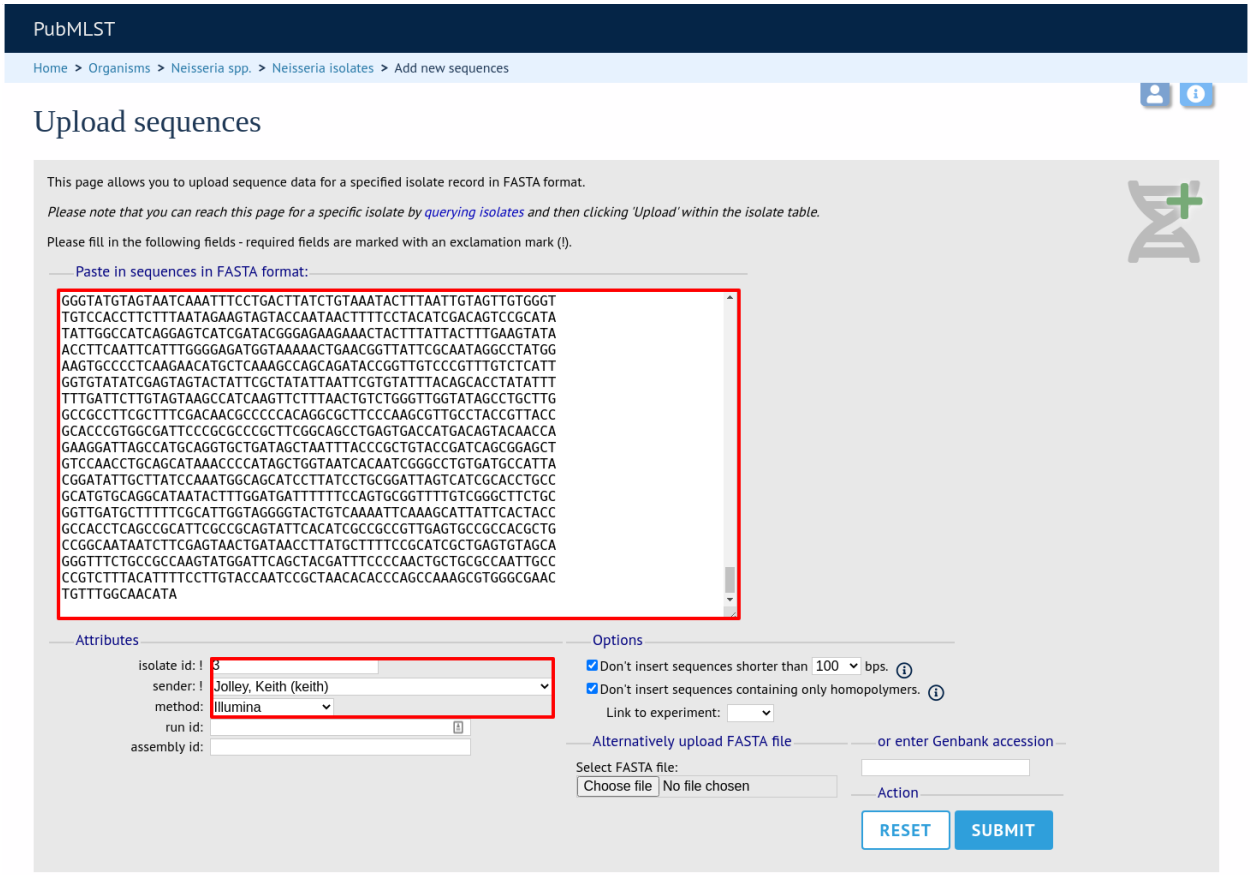
6.19.1 Select isolate from drop-down list

To upload sequence data, click the sequences add (+) sequence bin link on the curator's main page.

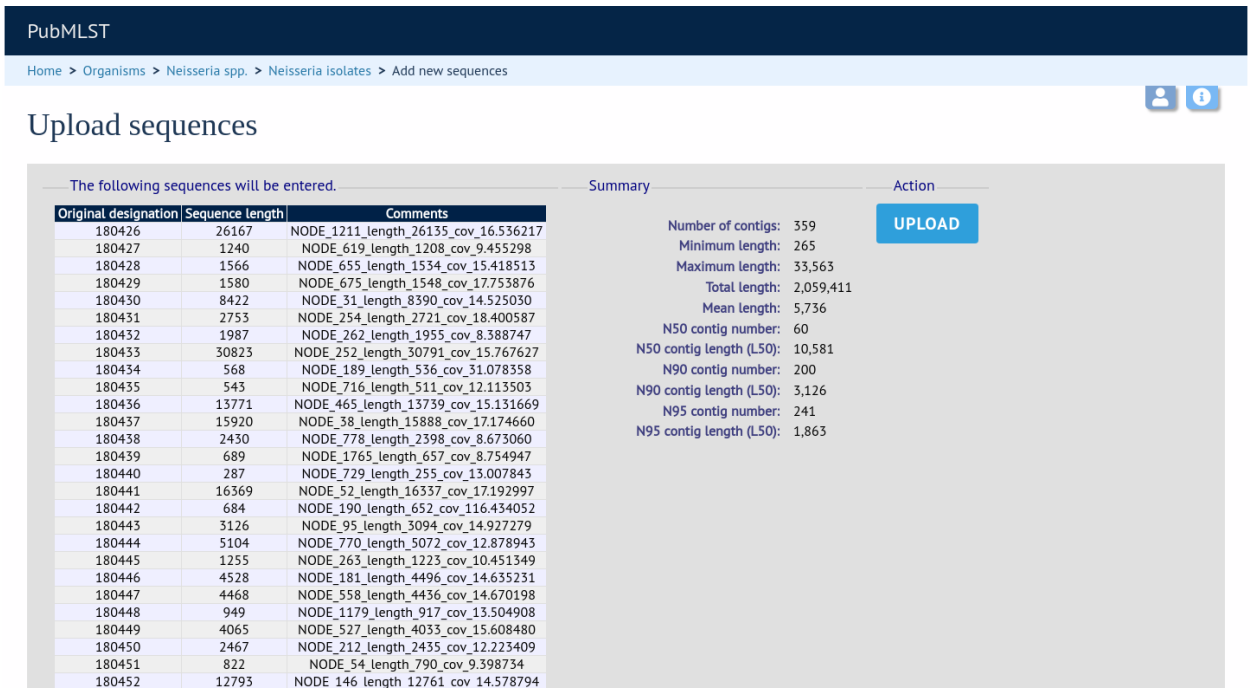


Select the isolate that you wish to link the sequence to from the dropdown list box (or if the database is large and there are too many isolates to list, enter the id in the text box). You also need to enter the person who sent the data. Optionally, you can add the sequencing method used.

Paste sequence contigs in FASTA format in to the form.



Click 'Submit'. A summary of the number of isolates and their lengths will be displayed. To confirm upload, click 'Upload'.

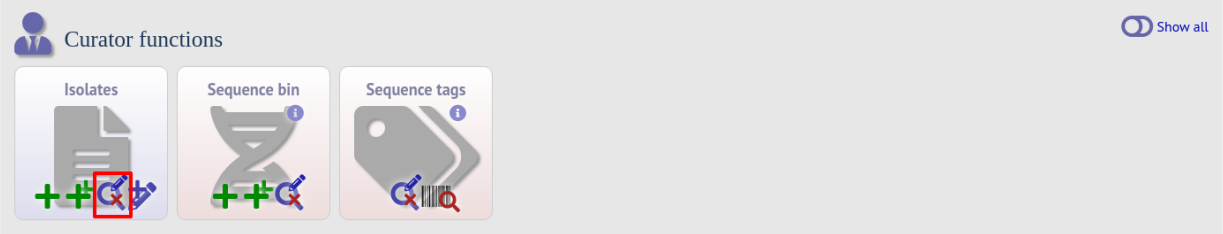


Original designation	Sequence length	Comments
180426	26167	NODE_1211_length_26135_cov_16.536217
180427	1240	NODE_619_length_1208_cov_9.455298
180428	1566	NODE_655_length_1534_cov_15.418513
180429	1580	NODE_675_length_1548_cov_17.753876
180430	8422	NODE_31_length_8390_cov_14.525030
180431	2753	NODE_254_length_2721_cov_18.400587
180432	1987	NODE_262_length_1955_cov_8.388747
180433	30823	NODE_252_length_30791_cov_15.767627
180434	568	NODE_189_length_536_cov_31.078358
180435	543	NODE_716_length_511_cov_12.113503
180436	13771	NODE_465_length_13739_cov_15.131669
180437	15920	NODE_38_length_15888_cov_17.174660
180438	2430	NODE_778_length_2398_cov_8.673060
180439	689	NODE_1765_length_657_cov_8.754947
180440	287	NODE_729_length_255_cov_13.007843
180441	16369	NODE_52_length_16337_cov_17.192997
180442	684	NODE_190_length_652_cov_116.434052
180443	3126	NODE_95_length_3094_cov_14.927279
180444	5104	NODE_770_length_5072_cov_12.878943
180445	1255	NODE_263_length_1223_cov_10.451349
180446	4528	NODE_181_length_4496_cov_14.635231
180447	4468	NODE_558_length_4436_cov_14.670198
180448	949	NODE_1179_length_917_cov_13.504908
180449	4065	NODE_527_length_4033_cov_15.608480
180450	2467	NODE_212_length_2435_cov_12.223409
180451	822	NODE_54_length_790_cov_9.398734
180452	12793	NODE_146_length_12761_cov_14.578794

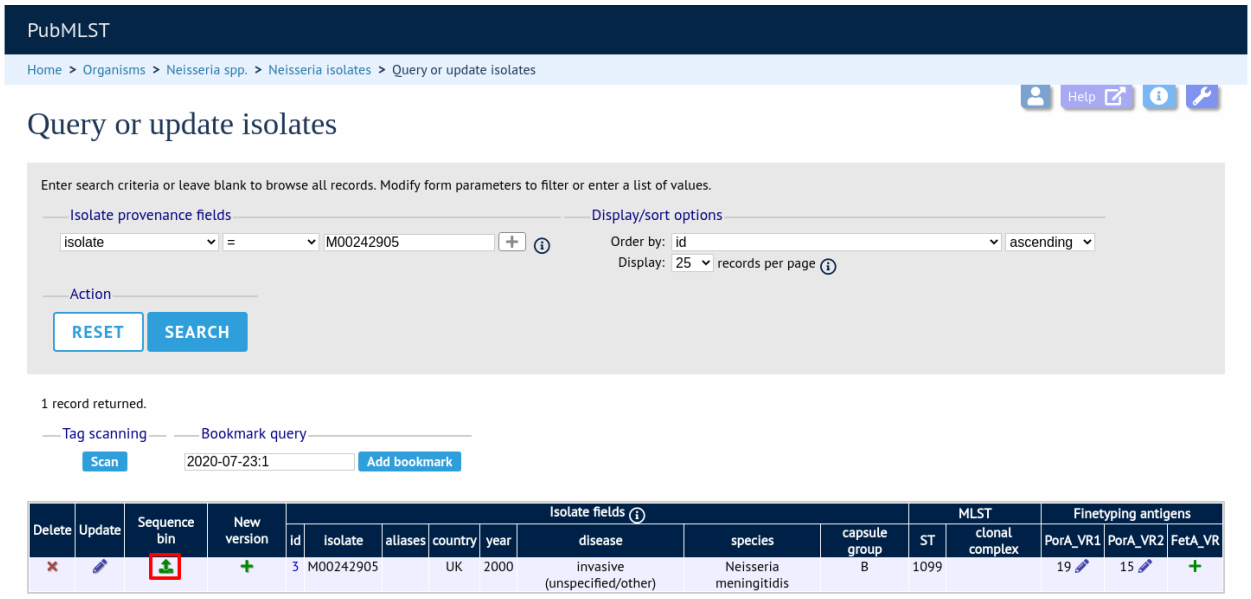
6.19.2 Select from isolate query

As an alternative to selecting the isolate from a dropdown list (or entering the id on large databases), it is also possible to upload sequence data following an isolate query.

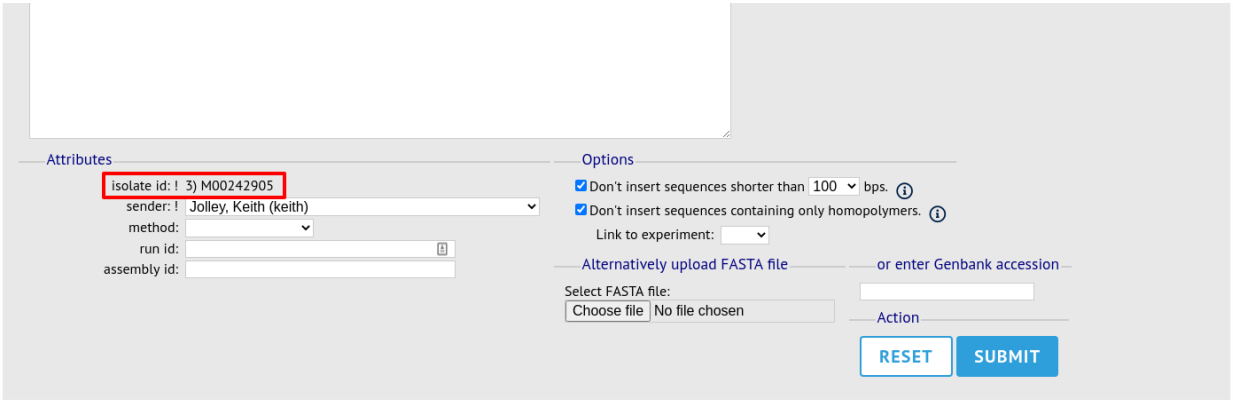
Click the isolate update/delete link from the curator's main page.



Enter your search criteria. From the list of isolates displayed, click the 'Upload' link in the sequence bin column of the appropriate isolate record.

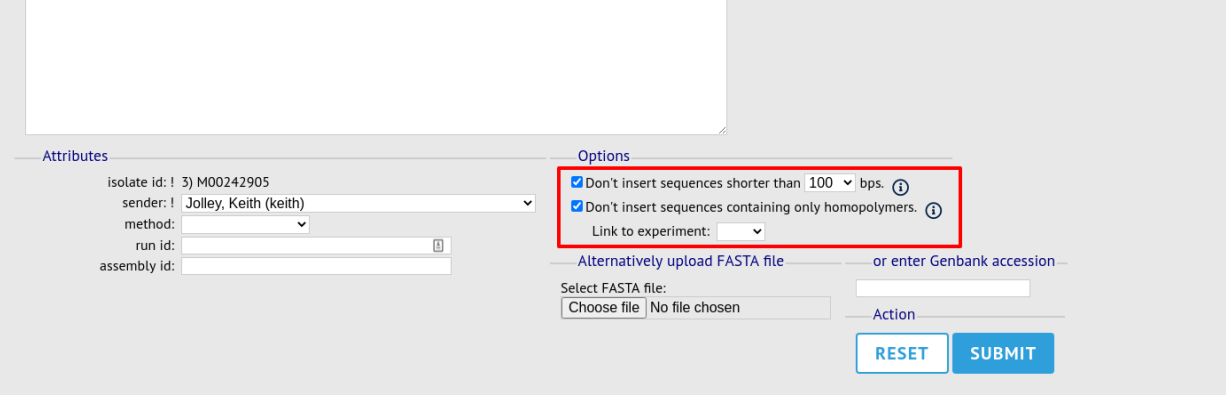


The same upload form as detailed above is shown. Instead of a dropdown list for isolate selection, however, the chosen isolate will be pre-selected.



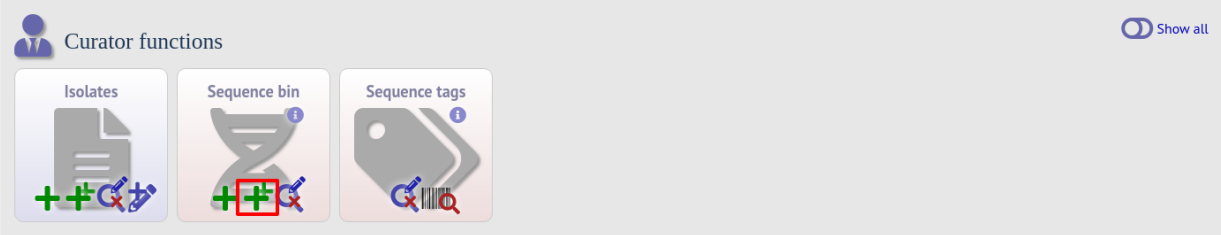
6.19.3 Upload options

On the upload form, you can select to filter out short sequences or those containing only homopolymeric repeats (which can be artefactually produced by some assembler software versions) from your contig list.



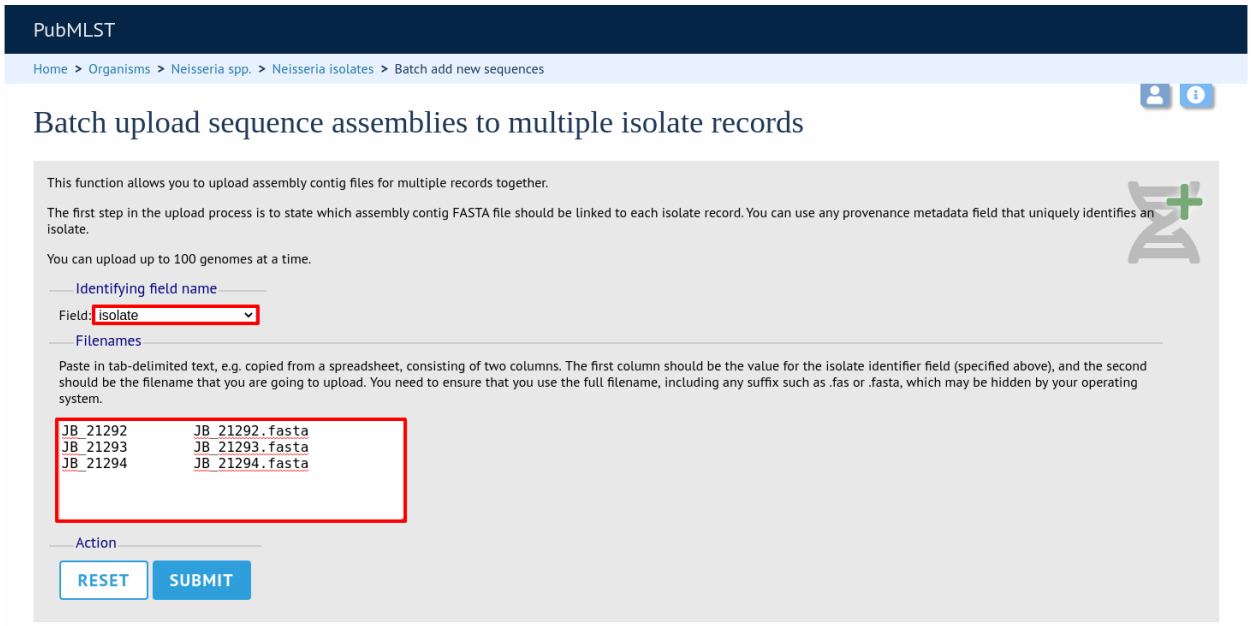
6.20 Batch uploading sequence contigs linked to multiple isolate records

To upload contigs for multiple isolates, click the batch add (++) sequence bin link on the curator's main page.

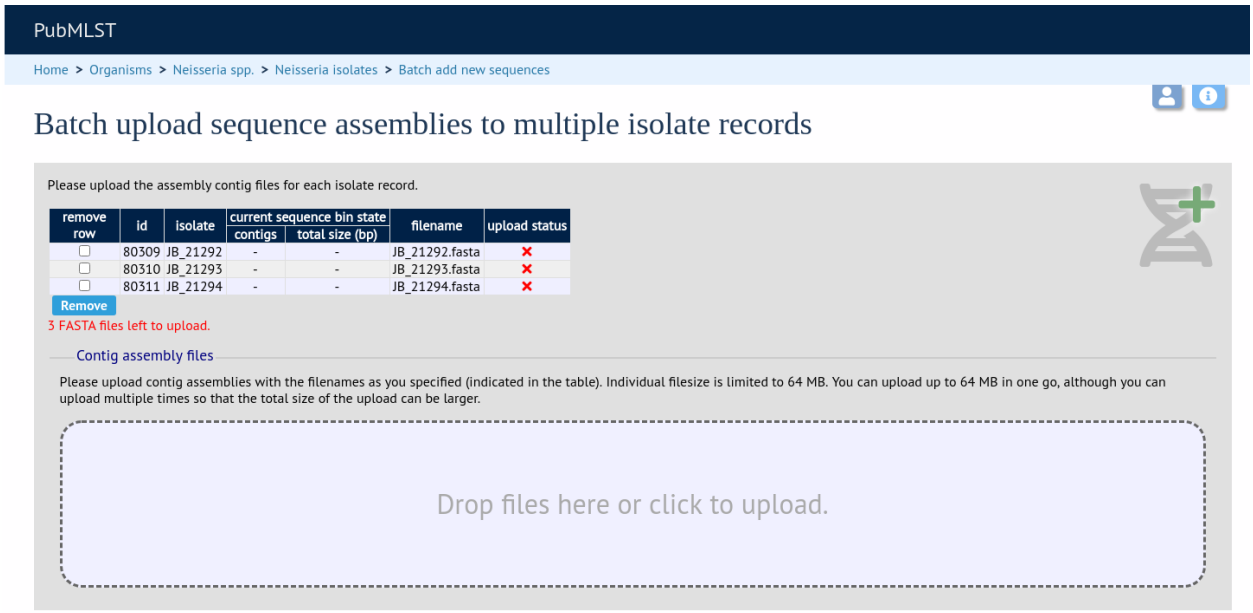


The first step is to upload the name of the contig file that will be linked to each isolate record. This can be done by pasting two columns in tab-delimited text format (e.g. from a spreadsheet) - the first column contains the isolate identifier, the second contains the filename of the contigs file, which should be in FASTA format.

You can choose which field to use for identifying the isolates, e.g. id (database id) or isolate (name of isolate). The value provided for this field needs to uniquely identify the isolate in the database - please note that only id is guaranteed to be unique.

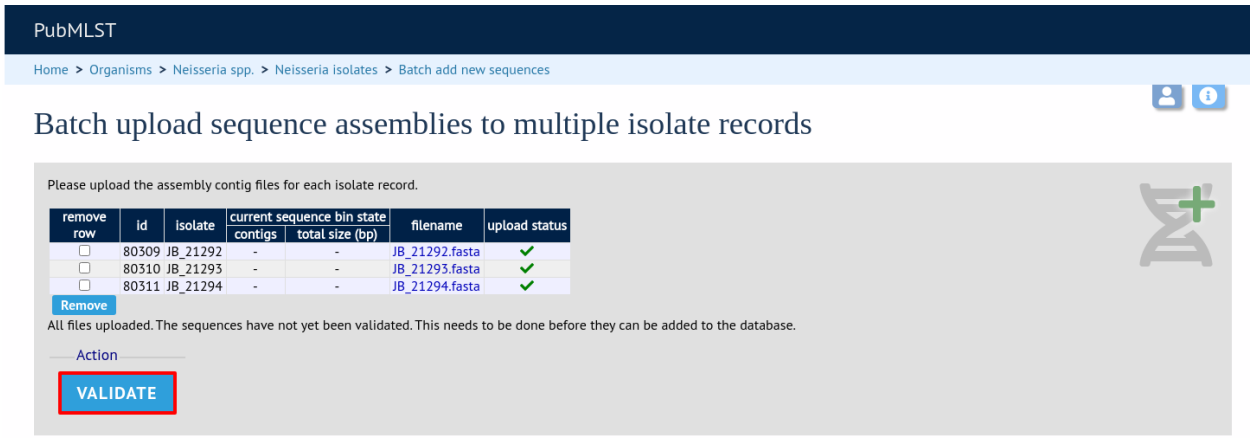


Click Submit. The system will check to make sure that the isolate records are uniquely identified (if not, you will see an error message informing you of this and you will need to use the database id as the identifier). You will then see a file upload form.

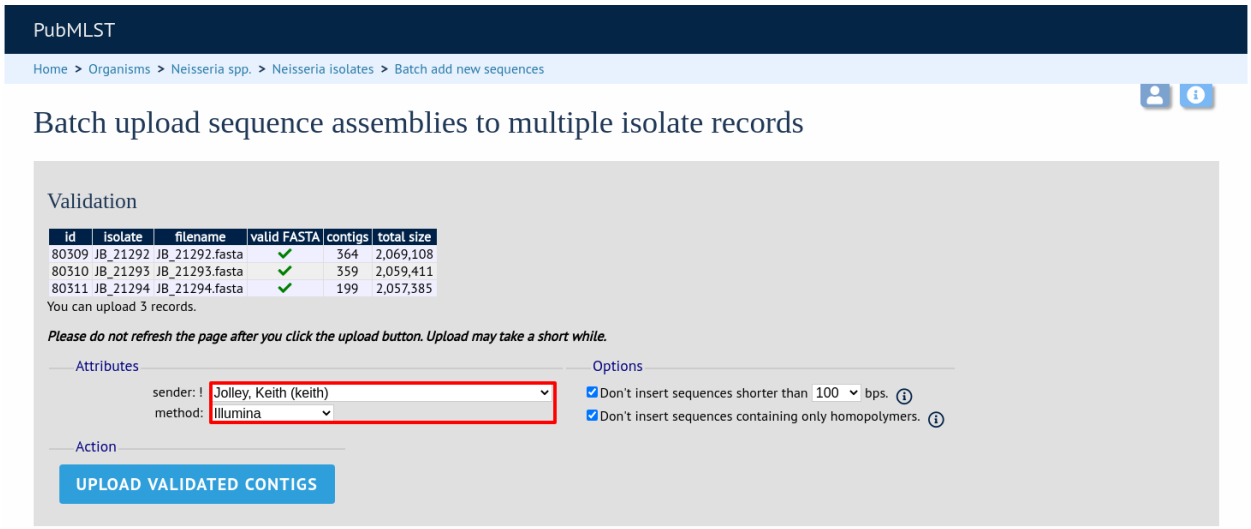


Drag and drop your FASTA format contig files in to the dotted drop area. Provided the filenames exactly match the filename you stated, these will be uploaded to a staging area.

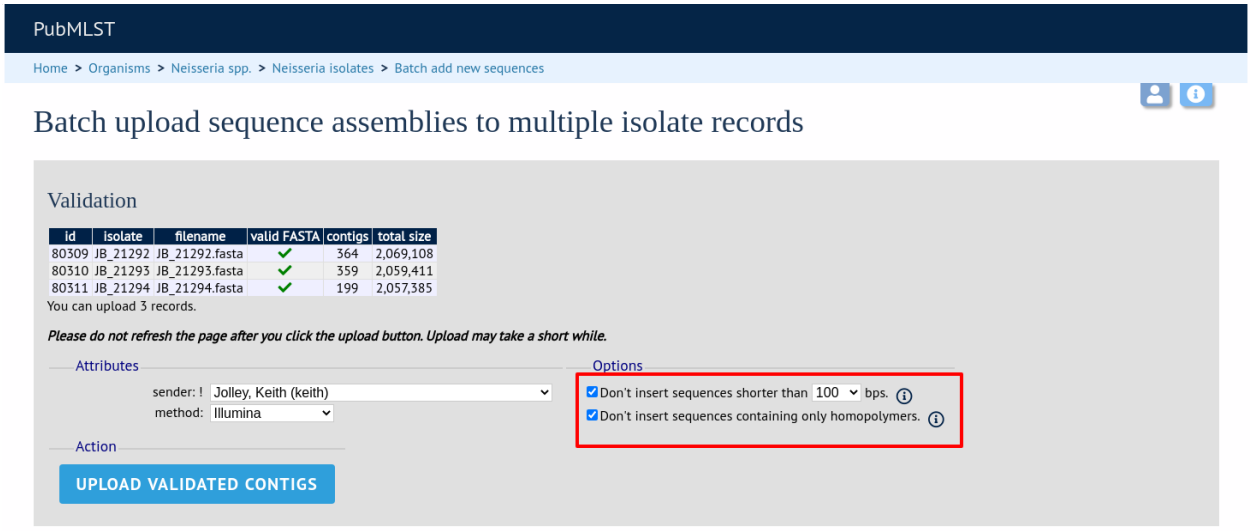
Click 'Validate' to check that these files are valid FASTA format.



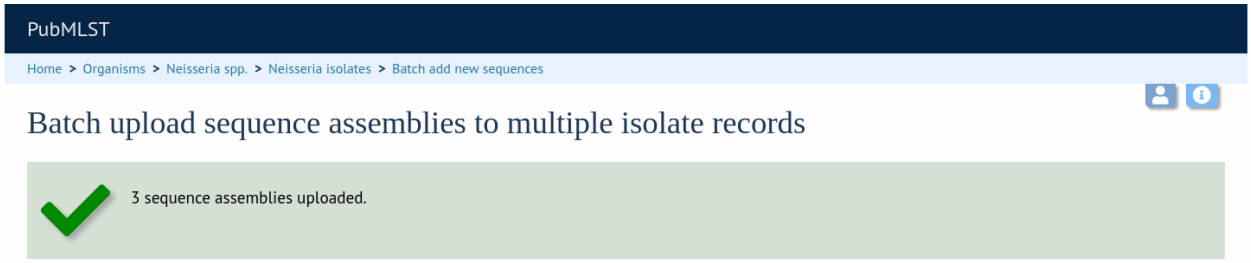
The files will be checked and a table will be displayed showing the total sequence size and number of contigs found. Select the data sender and, optionally the sequencing method from the dropdown lists. Then click 'Upload validated contigs'.



You can also choose to filter out short contigs selecting the checkbox and choosing the minimum length from the dropdown box in the options settings. You can also choose to filter out sequences containing only homopolymeric runs which can be produced artefactually by some assembler versions.



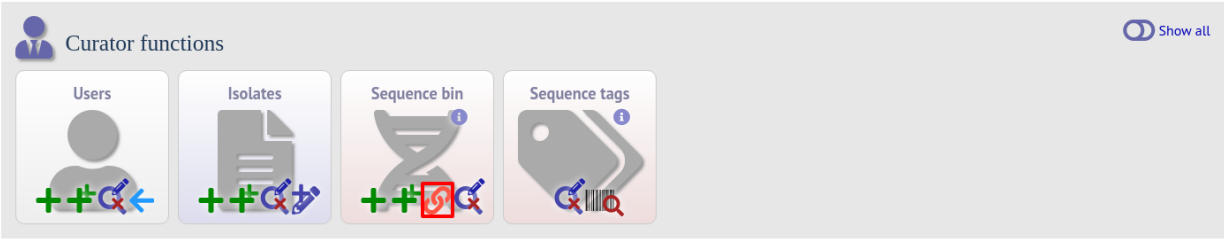
A confirmation message will be displayed after clicking the Upload button.



6.21 Linking remote contigs to isolate records

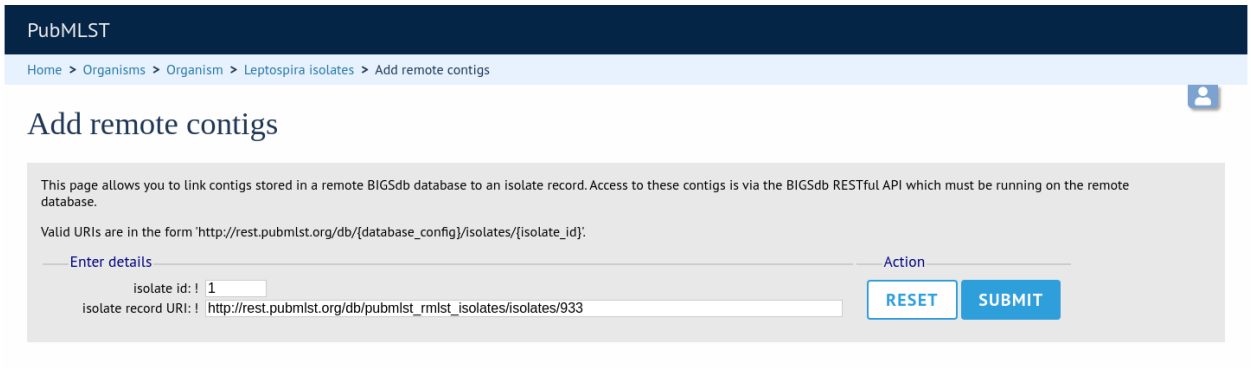
If *remote contigs have been enabled*, isolates can be linked to contigs stored in an external BIGSdb database, rather than directly uploaded. These well then be loaded when needed, for example during scanning or data export. This will be marginally slower than hosting contigs within the same database, but minimises duplication of sequence data and associated storage. Contigs need to be accessible via the BIGSdb *RESTful API*.

Click the sequences link icon on the curator's main page.

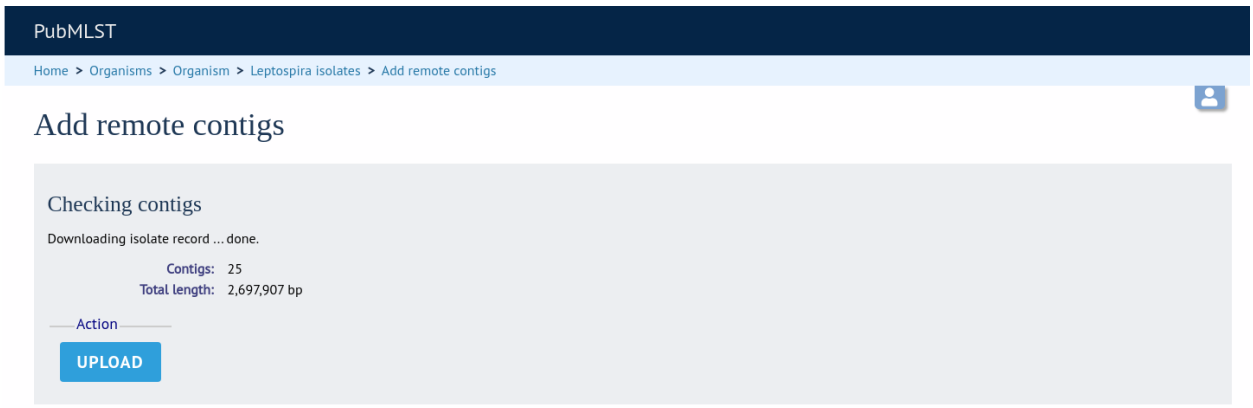


Either select the isolate id from the dropdown list, or enter it manually (list is disabled if there are >1000 records in the database). Enter the URI for the RESTful API of the parent isolate record, e.g. http://rest.pubmlst.org/db/pubmlst_rmlst_isolates/isolates/933. This URI can require authentication if credentials have been *set up*.

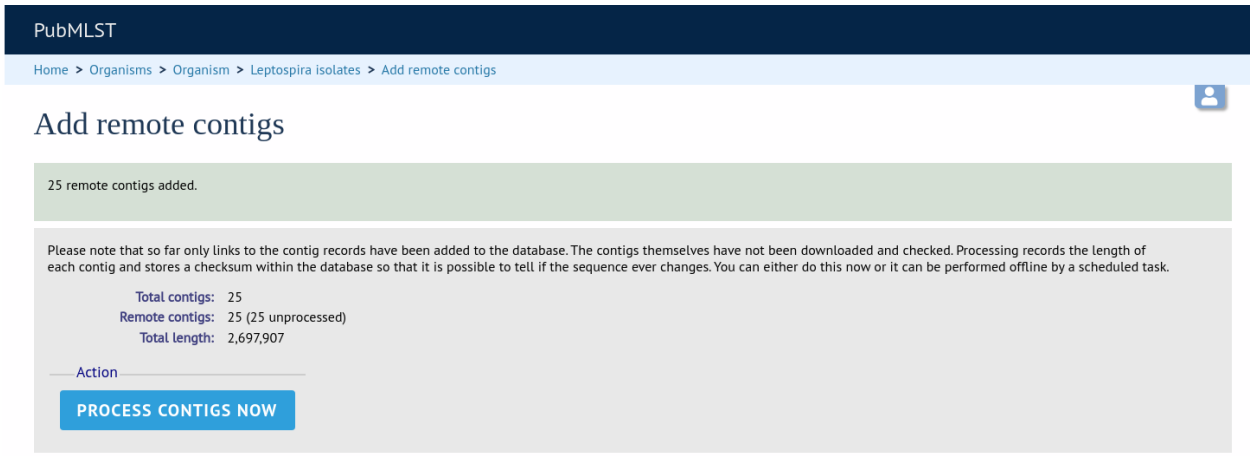
Press submit.



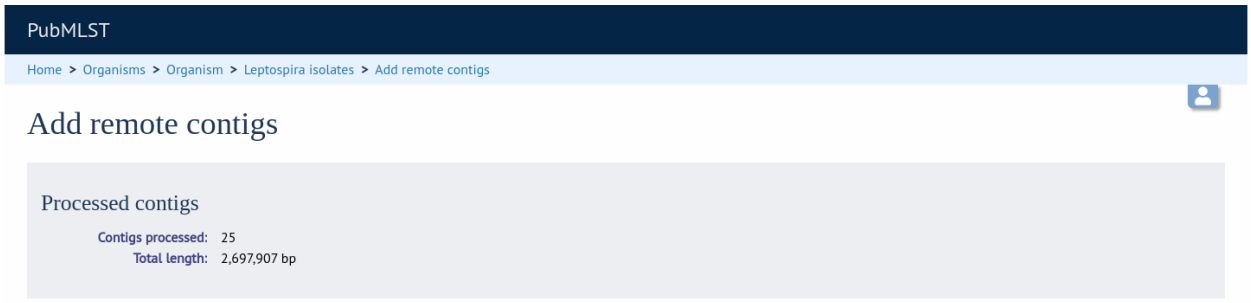
Summary information about the number of contigs and their total length will be downloaded from the remote isolate record. You will then be prompted to upload this information to the database, by clicking the 'Upload' button.



The contigs will be downloaded in bulk in order to determine their lengths. This information is stored within the local database as it is required for various outputs. Full metadata is not stored at this stage.



This is all that is required for the contigs to be used as normal. In order to get the full metadata about the contigs (sequencing platform used, sender and datestamp information), you can choose to process the contigs by clicking the 'Process contigs now' button. This will download each contig in turn, and store its provenance metadata locally.

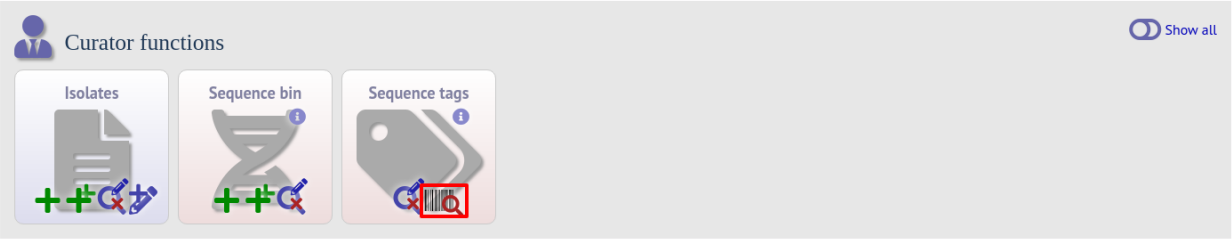


Alternatively, this step can be *performed offline automatically*.

6.22 Automated web-based sequence tagging

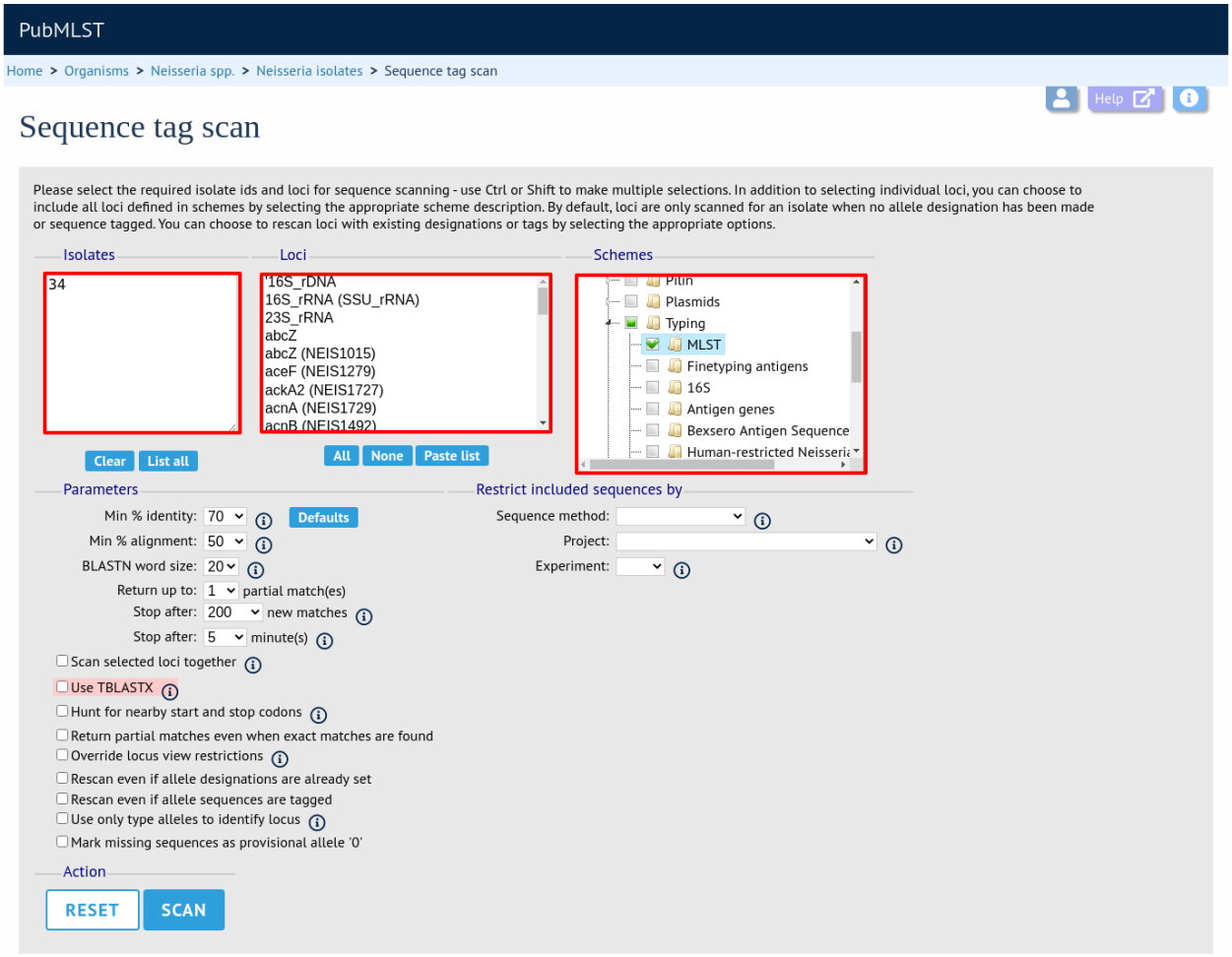
Sequence tagging, or tag-scanning, is the process of identifying alleles by scanning the sequence bin linked to an isolate record. Defined loci can either have a single reference sequence, that is defined in the locus table, or they can be linked to an external database that contains the sequences for known alleles. The tagging function uses BLAST to identify sequences and will tag the specific sequence region with locus information and an allele designation if a matching allele is identified by reference to an external database.

Select 'scan' sequence tags on the curator's index page.



Next, select the isolates whose sequences you wish to scan against. Multiple isolates can be selected by holding down the Ctrl key. All isolates can be selected by clicking the 'All' button under the isolate selection list. On database with a large number of isolates, you will need to enter a list of isolate ids rather than pick from a list.

Select either individual loci or schemes (collections of loci) to scan against. Again, multiple selections can be made.



Choose your scan parameters. Lowering the value for BLASTN word size will increase the sensitivity of the search at the expense of time. Using TBLASTX is more sensitive but also much slower. TBLASTX can only be used to identify the sequence region rather than a specific allele (since it will only match the translated sequence and there may be multiple alleles that encode a particular peptide sequence).

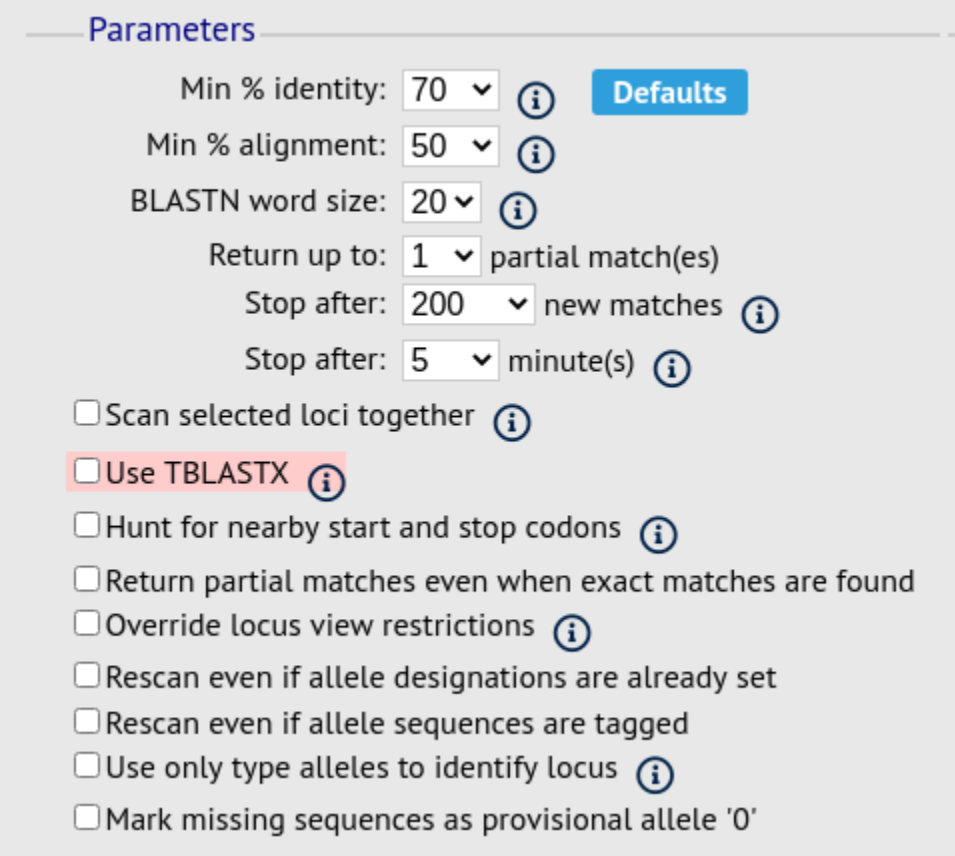
By default, for each isolate only loci that have not had either an allele designation made or a sequence region scanned will be scanned again. To rescan in these cases, select either or both the following:

- Rescan even if allele designations are already set
- Rescan even if allele sequences are tagged

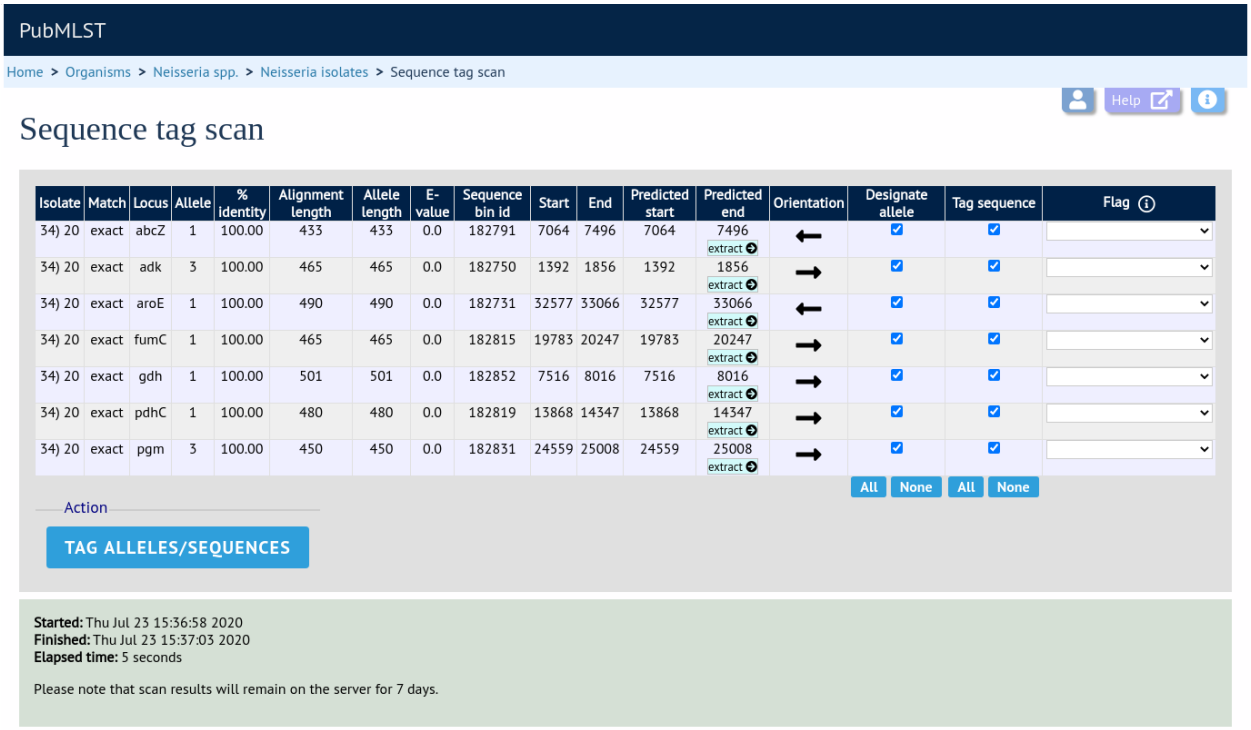
You can select to only use type alleles to identify the locus. This will constrain the search space so that allele definitions don't become more variable over time. If a partial match is found to a type allele then a full database lookup will be performed to identify any known alleles. An allele can be given a status of type allele when *defining*.

If fast scanning is enabled, there will also be an option to 'Scan selected loci together'. This can be significantly quicker than a locus-by-locus search against all alleles but is not enabled by default as it can use more memory on the server and requires *exemplar alleles* to be defined.

Options can be returned to their default setting by clicking the 'Defaults' button.



Press 'Scan'. The system takes approximately 1-2 seconds to identify each sequence (depending on machine speed and size of definitions databases). Alternatively, if 'Scan selected loci together' is available and selected, it may take longer to return initial results but total time should be less (e.g. a 2000 loci cgMLST scheme may be returned in 1-2 minutes). Any identified sequences will be listed in a table, with checkboxes indicating whether allele sequences or sequence regions are to be tagged.



Individual sequences can be extracted for inspection by clicking the 'extract →' link. The sequence (along with flanking regions) will be opened in another browser window or tab.

Checkboxes are enabled against any new sequence region or allele designation. You can also set a flag for a particular sequence to mark an attribute. These will be set automatically if these have been defined within the sequence definition database for an identified allele.

See also:

Sequence tag flags

Ensure any sequences you want to tag are selected, then press 'Tag alleles/sequences'.

If any new alleles are found, a link at the bottom will display these in a format suitable for automatic allele assignment by *batch uploading to sequence definition* database.

See also:

Offline curation tools

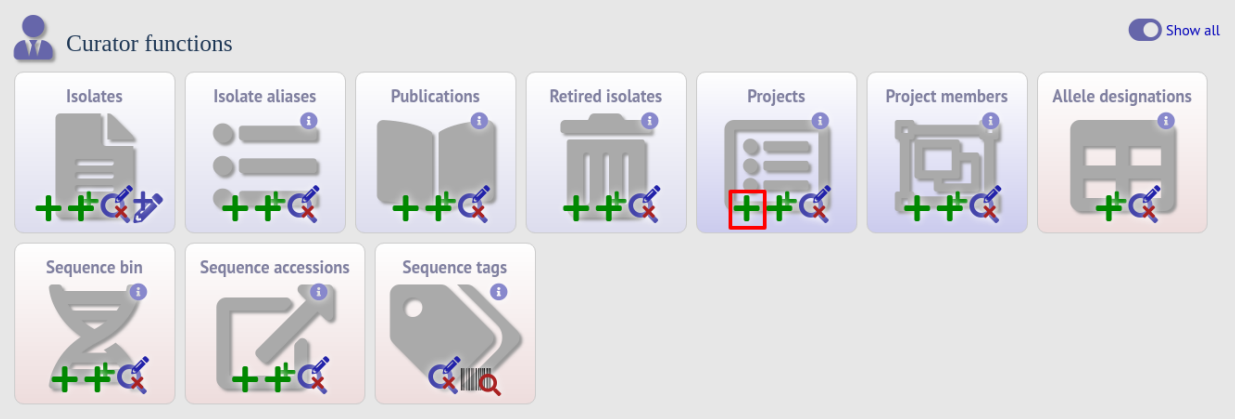
Automated offline sequence tagging

6.23 Projects

6.23.1 Creating the project

The first step in grouping by project is to set up a project.

Click the add (+) project link on the curator's main page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



Enter a short description for the project. This is used in drop-down list boxes within the query interfaces, so make sure it is not too long.

You can also enter a full description. If this is added, the project description can displayed at the top of an isolate information page (but see 'isolate_display' flag below). The full description can include HTML formatting, including image links.

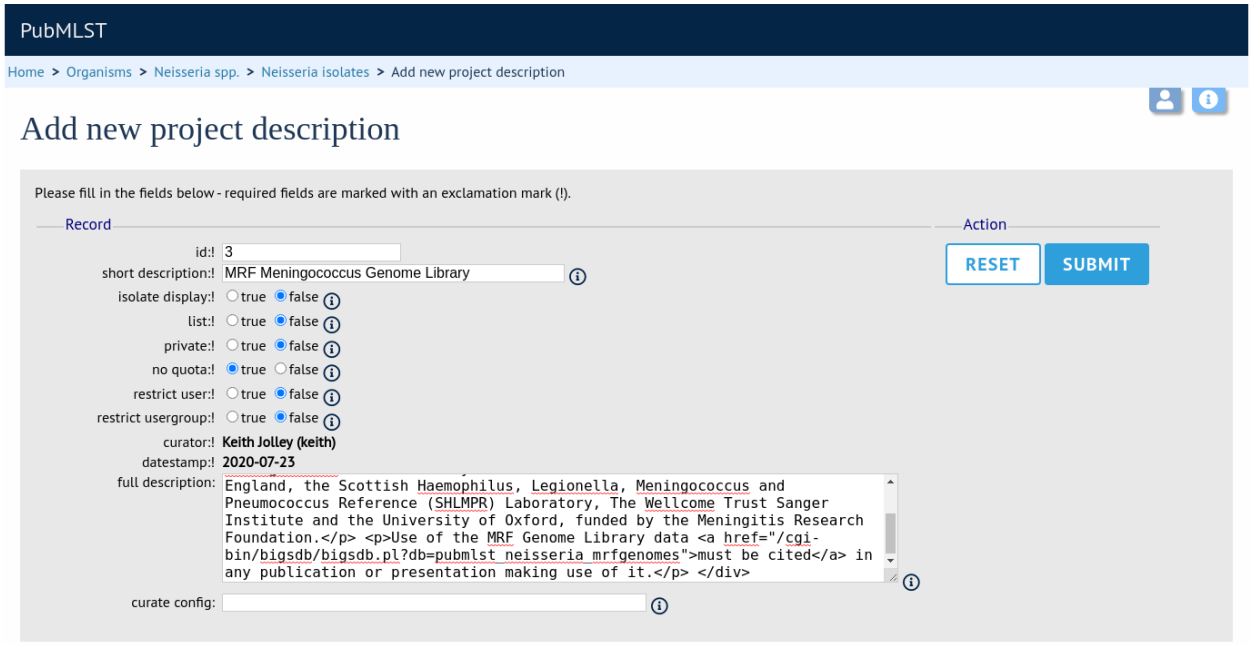
There are additionally two flags that affect how projects are listed:

- isolate_display - Setting this is required for the project and its description to be listed at the top of an isolate record (default: false).
- list - Setting this is required for the project to be listed in a page of projects linked from the main contents page.

There are a further two option flags:

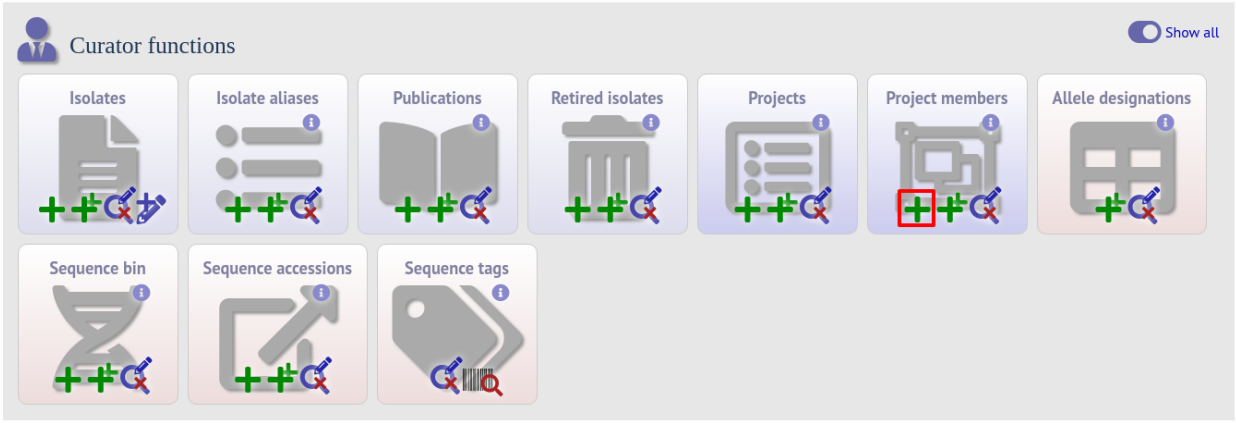
- private - Setting this makes the project a private *user project*. You will be set as the project owner and will be the only user able to access it by default. You can add additional users or user groups who will be able to access and update the project data later.
- no_quota - If set, isolates added to this project will not count against a user's quota of *private records* (only relevant to private projects).

Click 'Submit'.

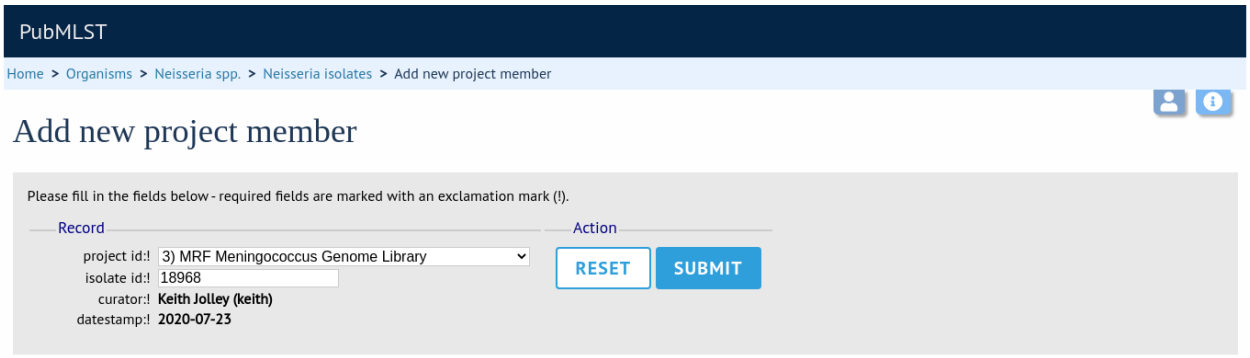


6.23.2 Explicitly adding isolates to a project

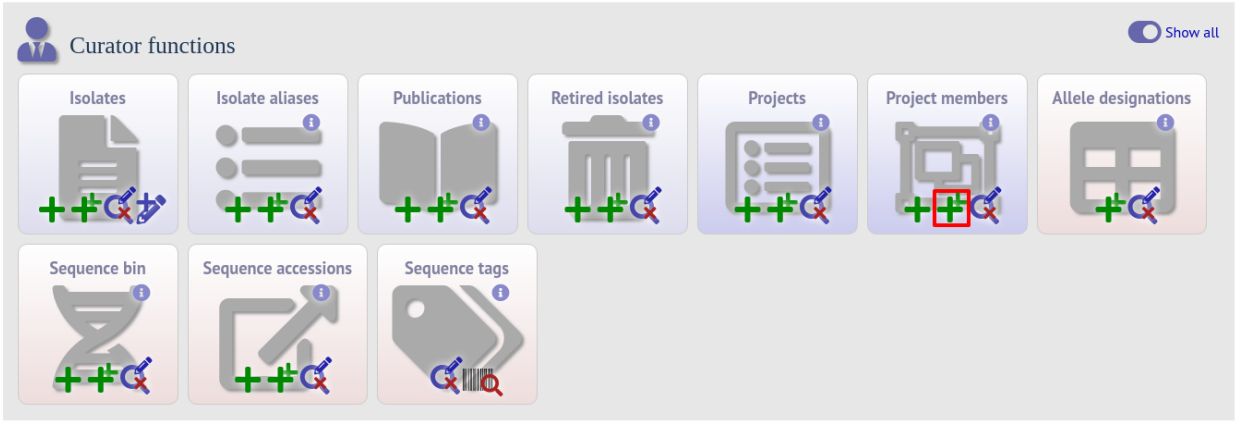
Explicitly adding isolates to the project can be done individually or in batch mode. To add individually, click the add (+) project member link on the curator's main page. This function is normally hidden, so you may need to click the 'Show all' toggle to display it.



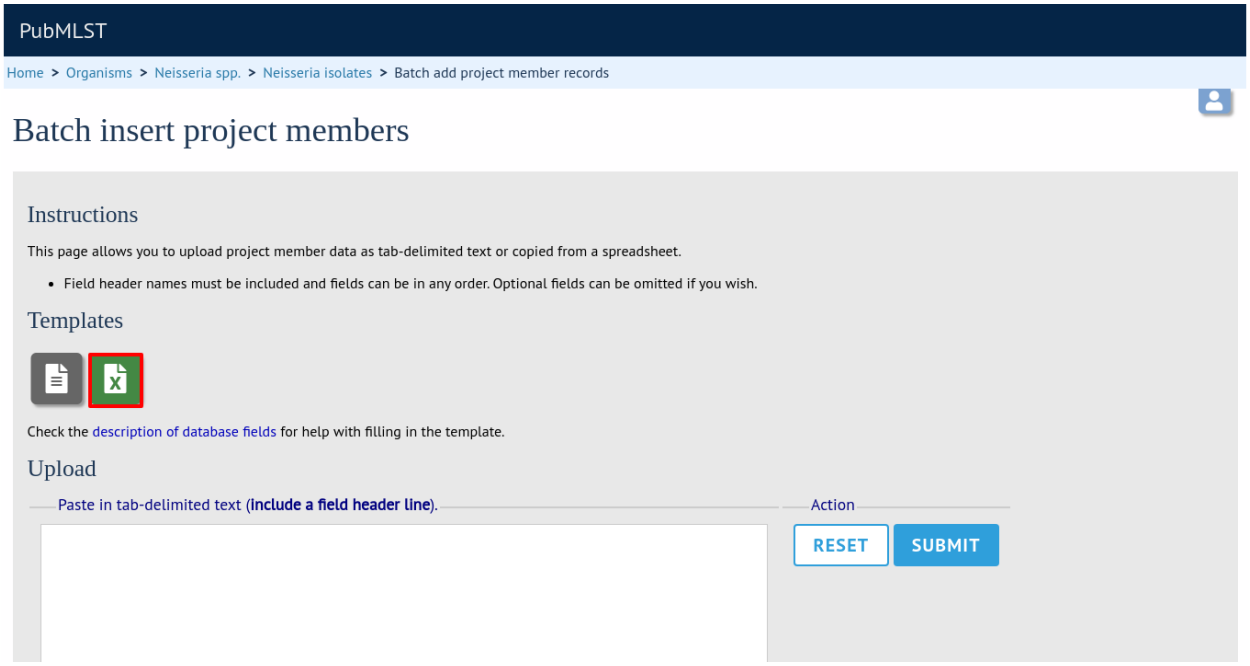
Select the project from the dropdown list box and enter the id of the isolate that you wish to add to the project. Click 'Submit'.



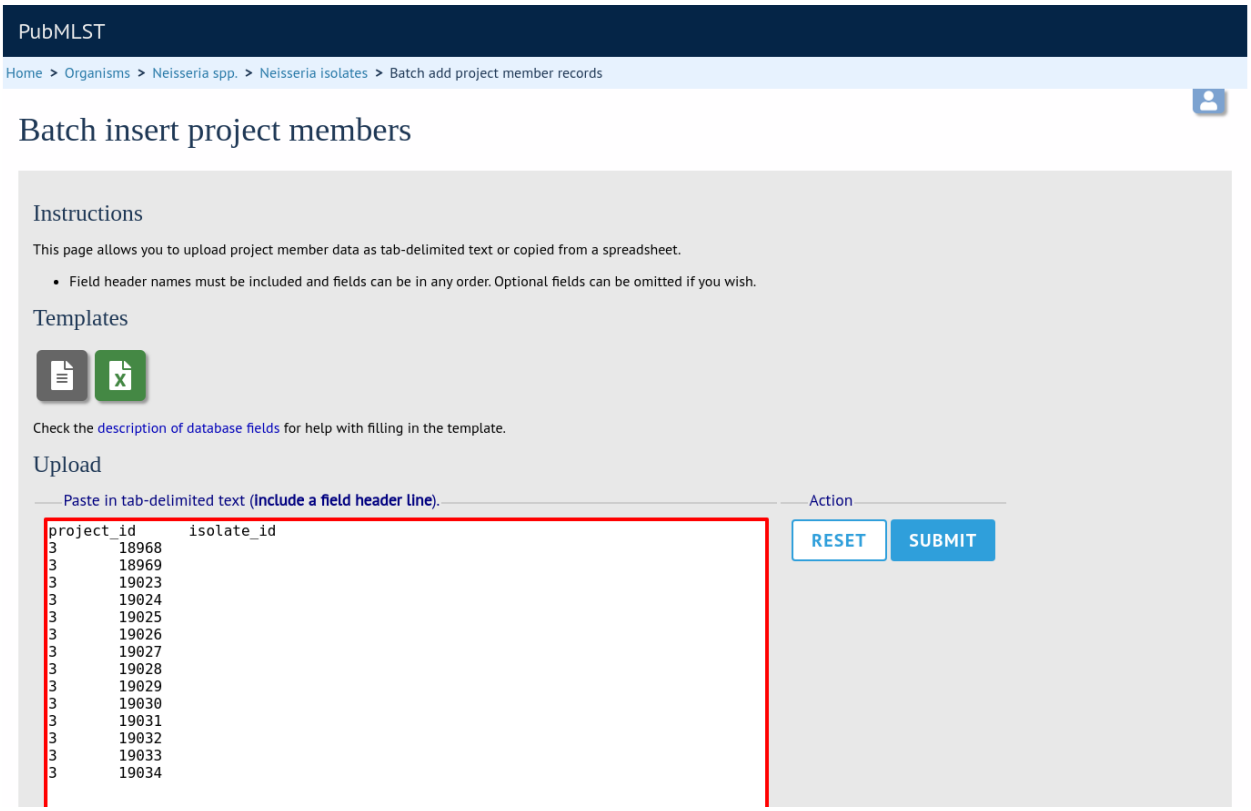
To add isolates in batch mode. Click the batch add (++) project members link on the curator's main page.



Download an Excel submission template:



You will need to know the id number of the project - this is the id that was used when you created the project. Fill in the spreadsheet, listing the project and isolate ids. Copy and paste this to the web upload form. Press 'Submit'.



See also:

Setting up user projects

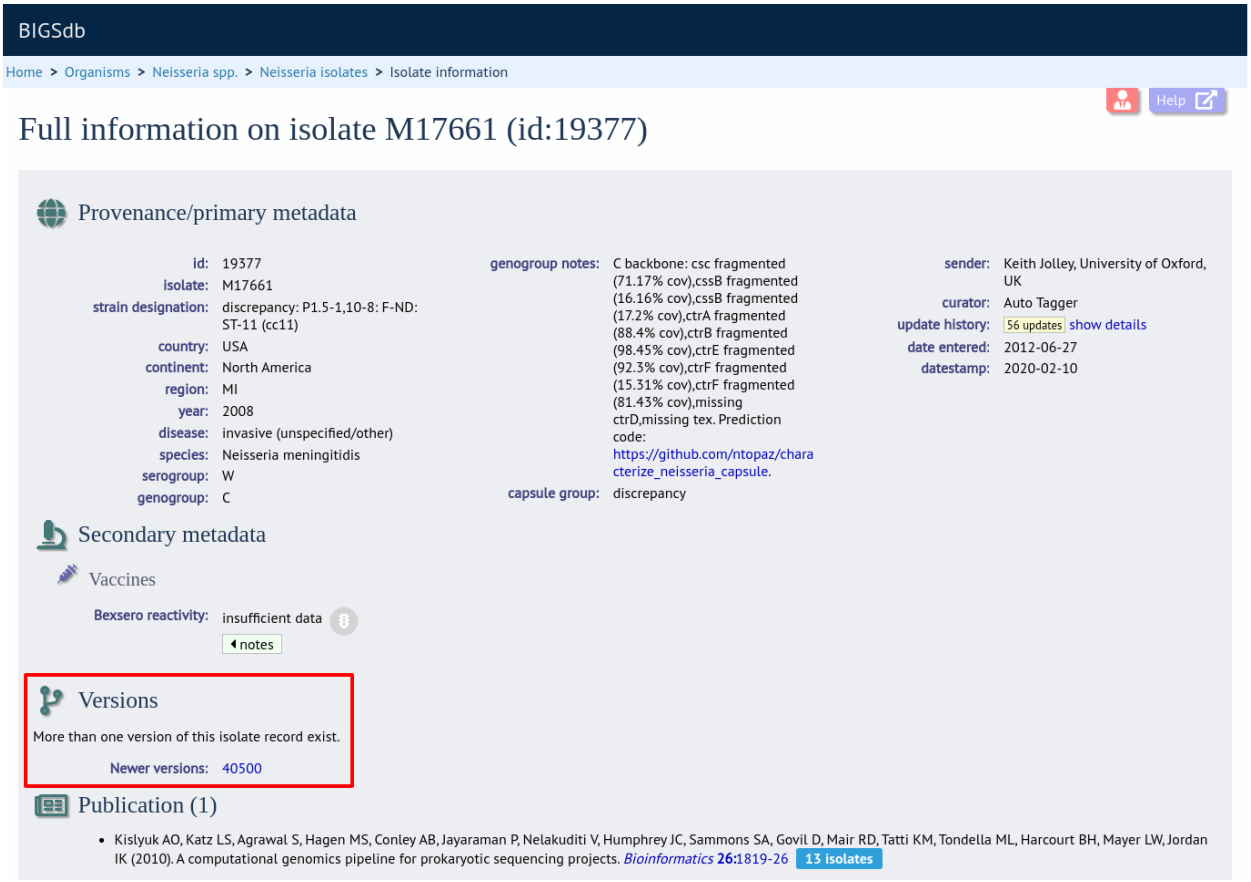
User projects

6.24 Isolate record versioning

Versioning enables multiple versions of genomes to be uploaded to the database and be analysed separately. When a new version is created, a copy of the provenance metadata, and publication links are created in a new isolate record. The sequence bin and allele designations are not copied.

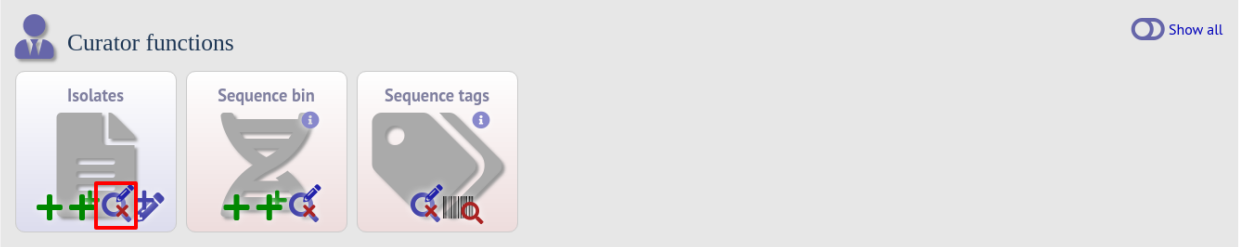
By default, old versions of the record are not returned from queries. Most query pages have a checkbox to 'Include old record versions' to override this.

Links to different versions are displayed within an isolate record:

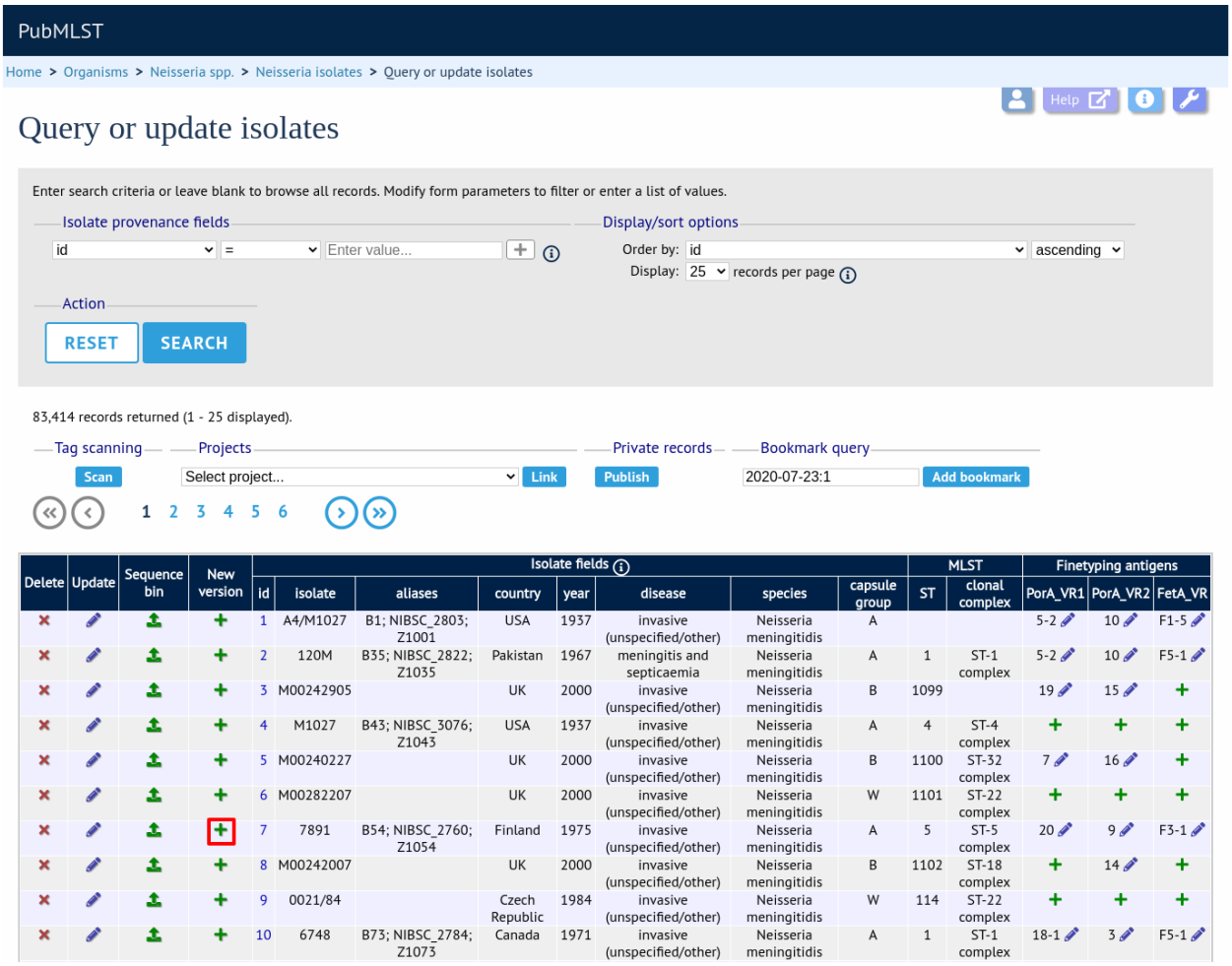


The different versions will also be listed in analysis plugins, with old versions identified with an [old version] designation after their name.

To create a new version of an isolate record, query or browse for the isolate:

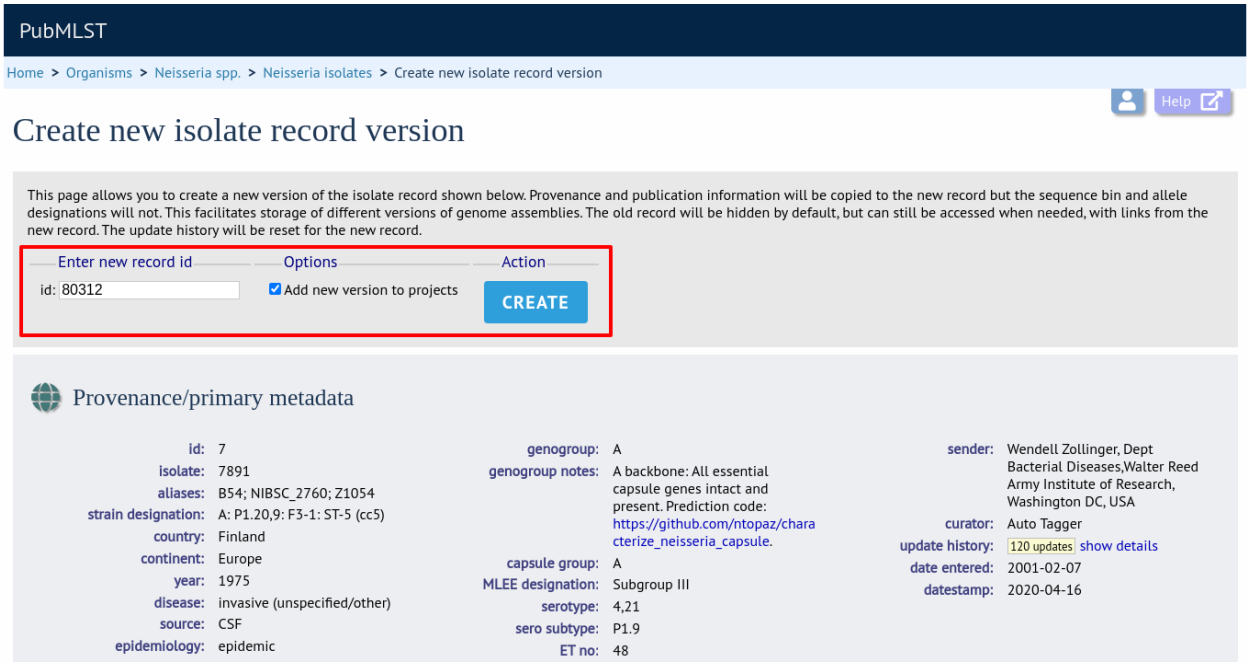


Click the ‘create’ new version link next to the isolate record:



The isolate record will be displayed. The suggested id number for the new record will be displayed - you can change this. By default, the new record will also be added to any projects that the old record is a member of. Uncheck the 'Add new version to projects' checkbox to prevent this.

Click the 'Create' button.

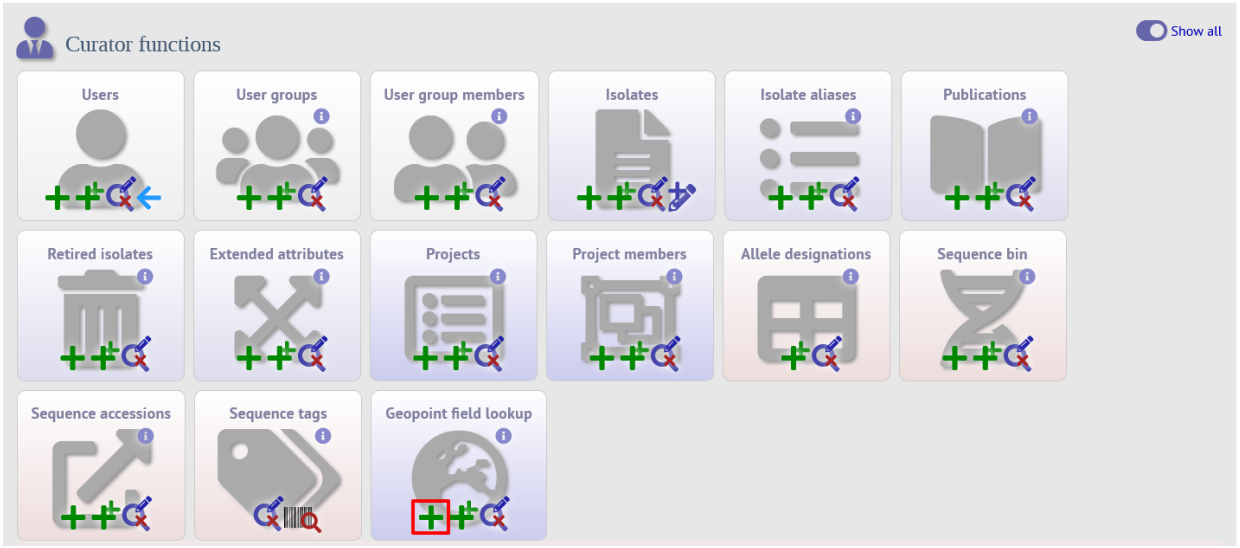


6.25 Populating geographic coordinate lookup values

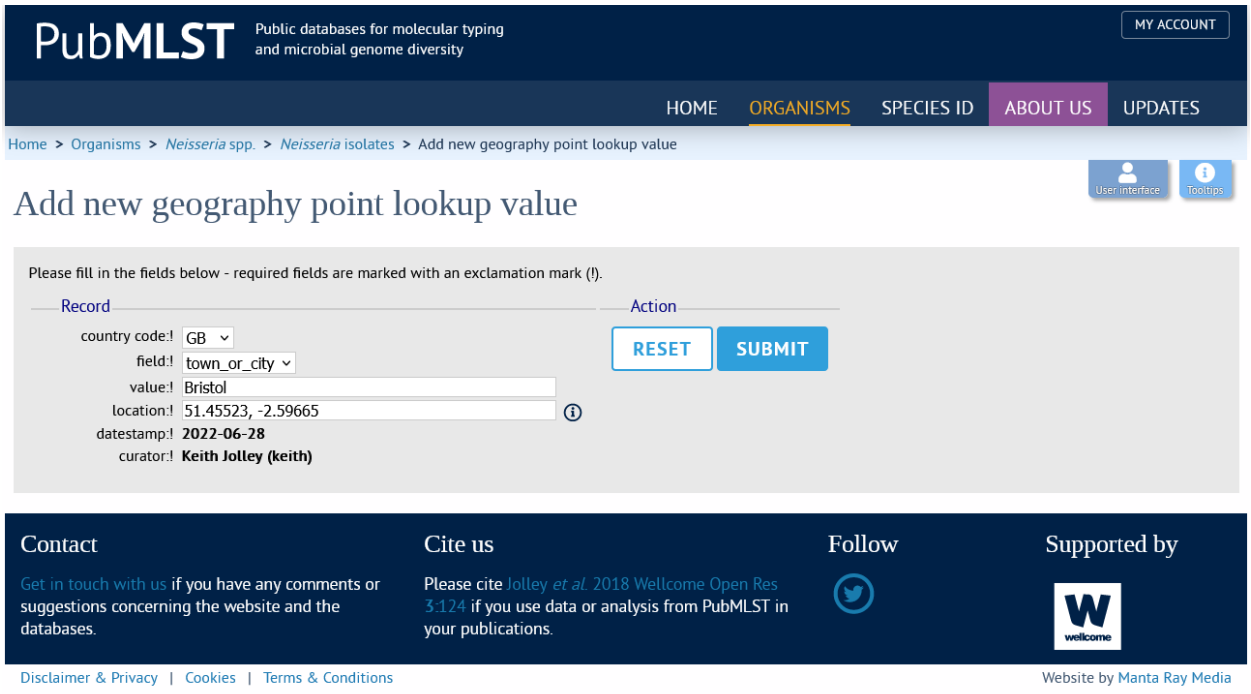
If a field has the geographic_point_lookup attribute set to 'yes' in the *config.xml file*, field values can be mapped to GPS coordinates to facilitate mapping.

If the user has curator privileges with the 'modify_geopoints' permission set, or is an admin, a curator link to modify 'Geopoint field lookup' will be available in the curator interface. This link is normally hidden but can be shown by selecting the 'Show all' toggle.

To add a value, click the 'Add' link:



Select the 2 letter ISO country code and the linked field from the dropdown lists. Enter the town or city in the value field, and the GPS coordinates in LATITUDE, LONGITUDE format.



Now, whenever that specified field value is used, a map will appear in the isolate information page showing the location. In addition, these values will be used in mapping in the Field Breakdown and Microreact plugins.

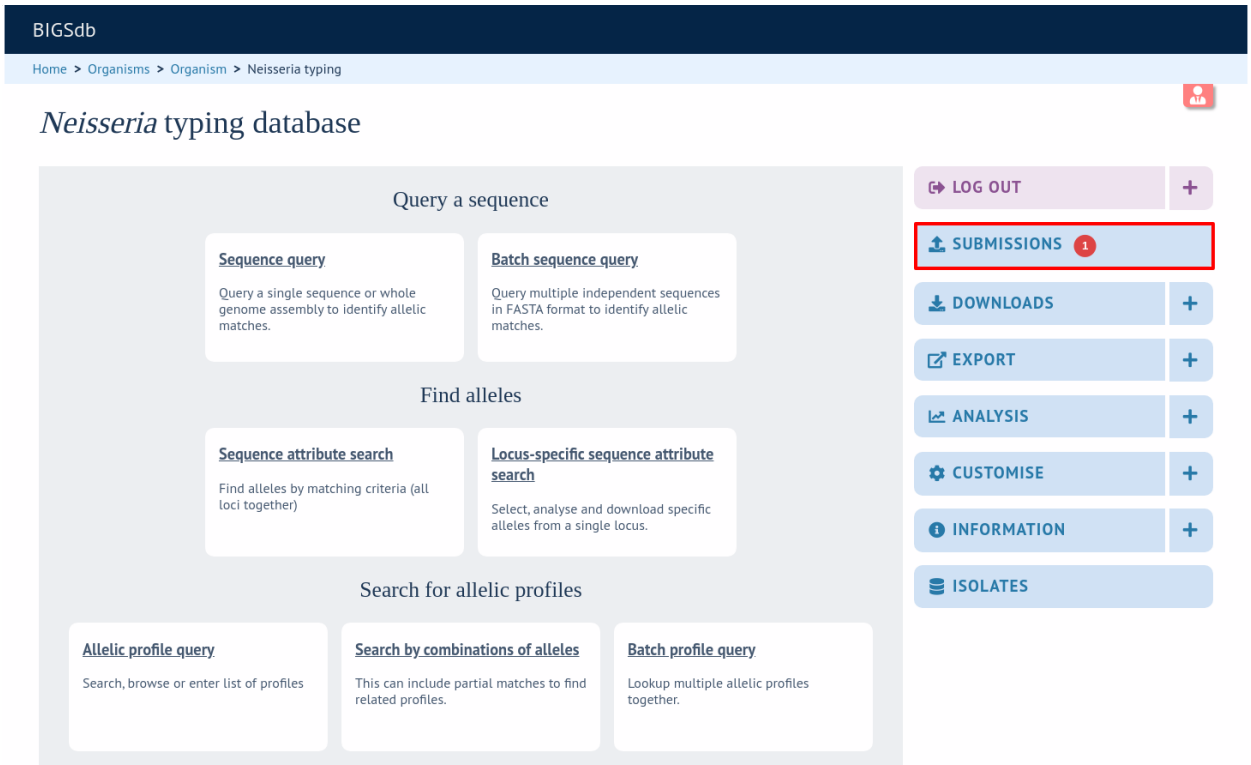
GPS coordinates can be mapped in bulk using the gp_town_lookups.pl script.

CHAPTER

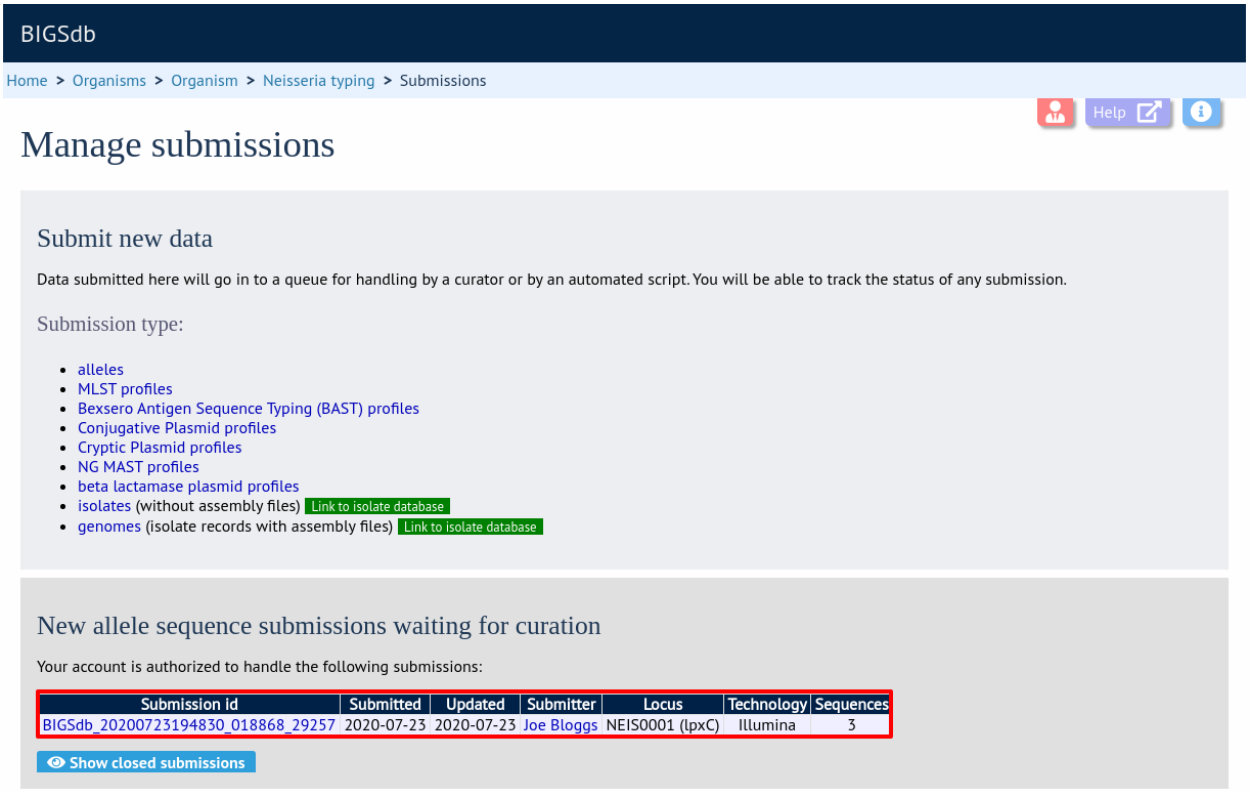
SEVEN

CURATING DATA SUBMITTED VIA THE AUTOMATED SUBMISSION SYSTEM

Data may be submitted by users using the automated submission system if it has been enabled for a specific database. As a curator, you will be notified of pending submissions when you log in to the curator's interface or if you access the 'Manage submissions' links from the standard contents page. Additionally, if your user account has the 'submission_emails' flag set in the users' table you will also receive E-mail notification of new submissions for which you have sufficient privileges to curate.

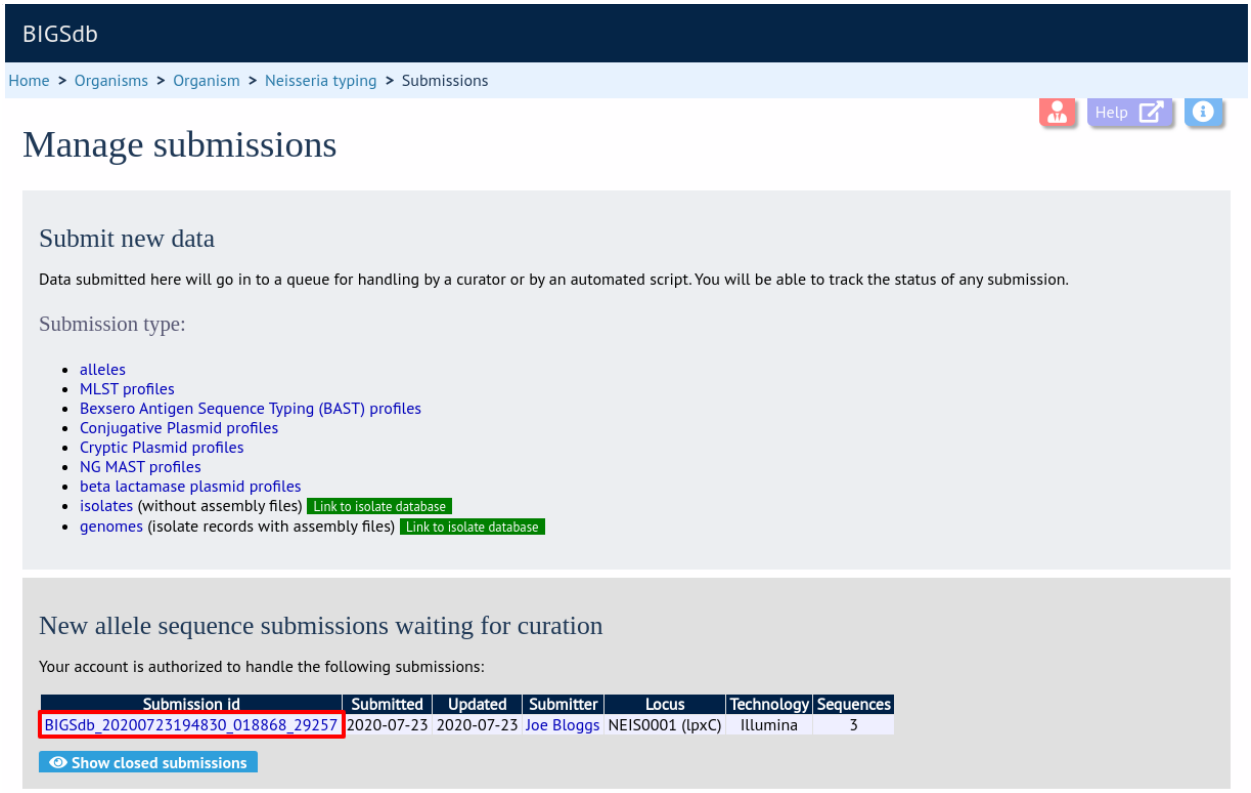


Any submissions for which you have sufficient privileges to curate will be shown.

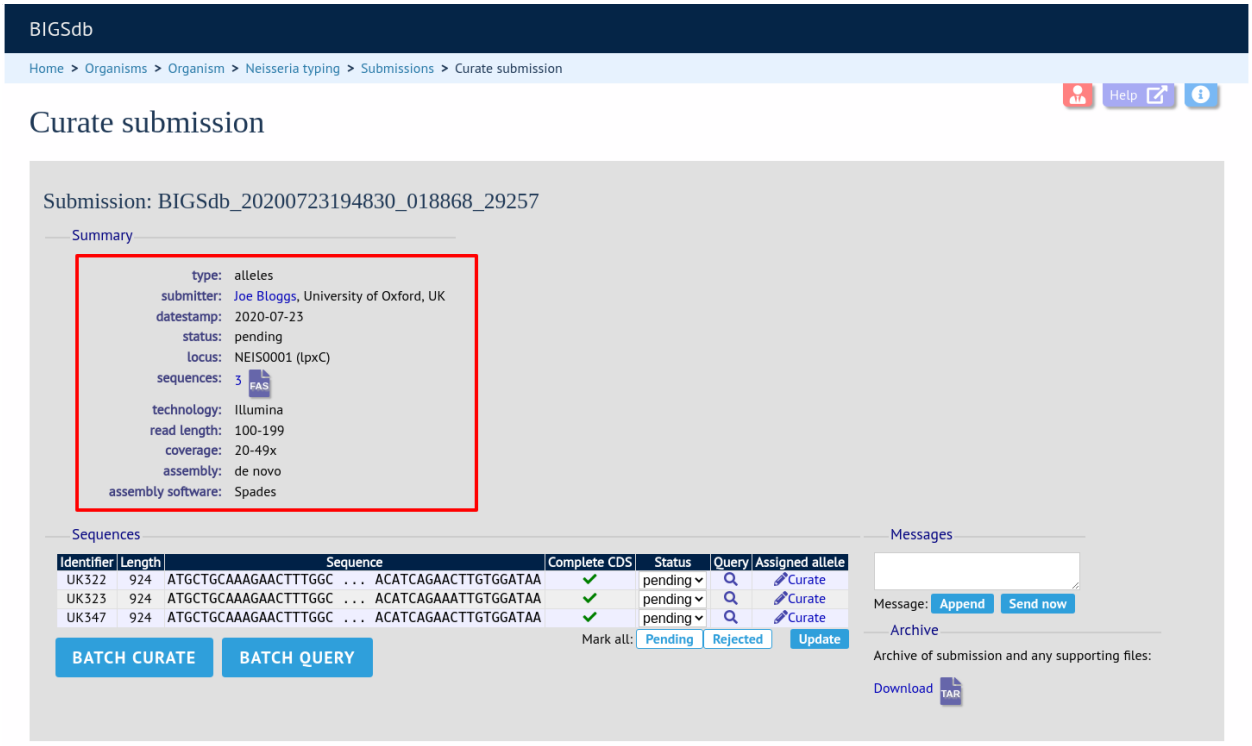


7.1 Alleles

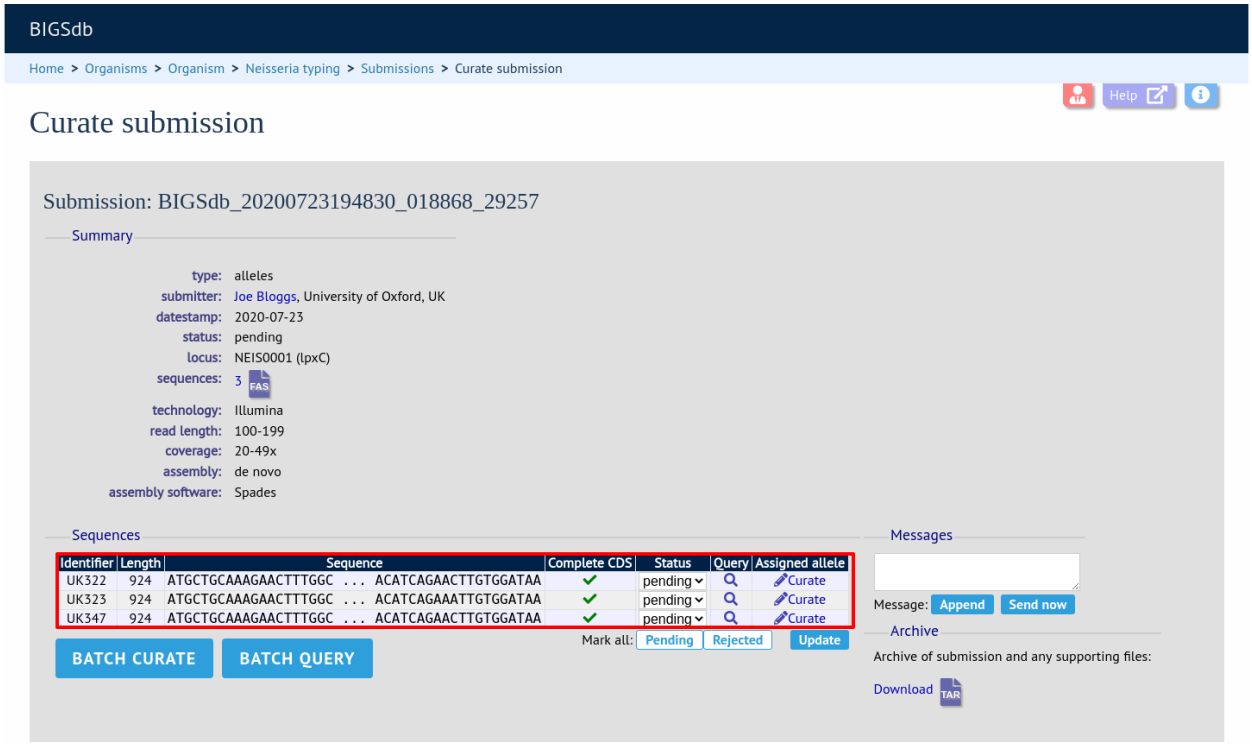
Click the link to the appropriate submission on the 'Manage submissions' page.



You will see a summary section that describes details about how the sequences were obtained. There should also be link here to download all the sequences in FASTA format.

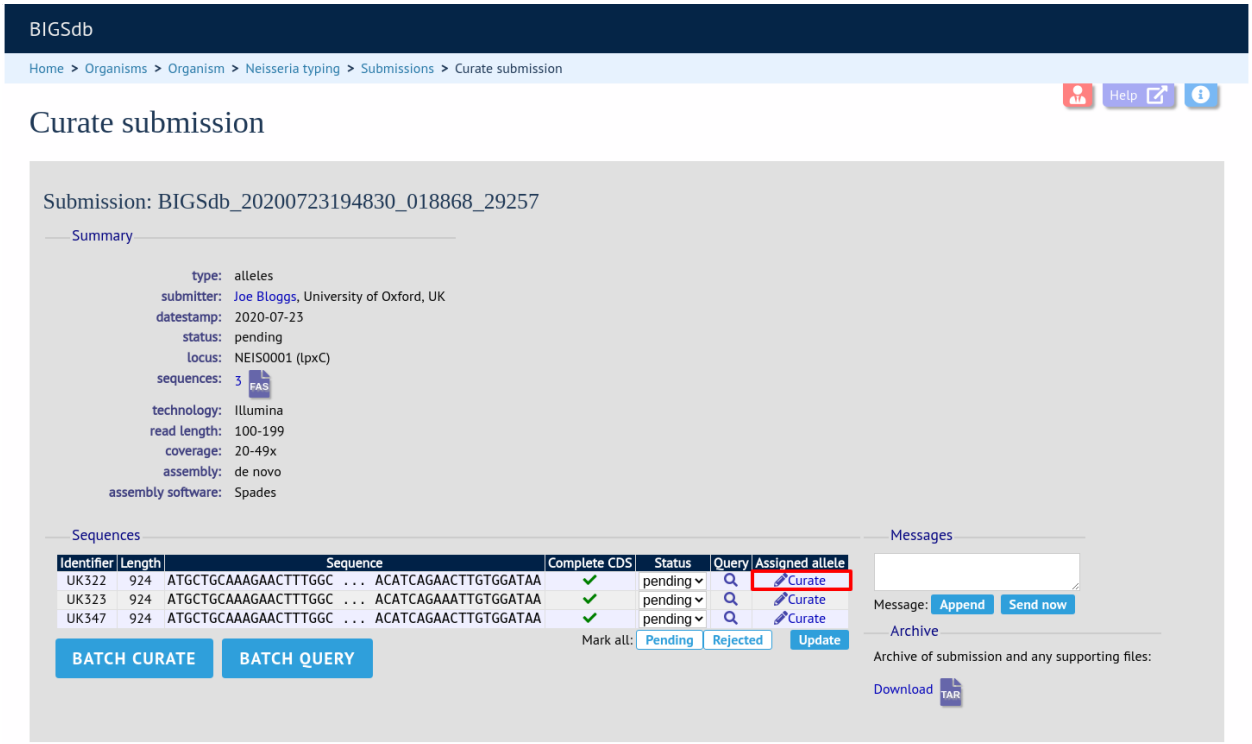


There will also be a table summarizing the sequences in the submission and their current submission status.

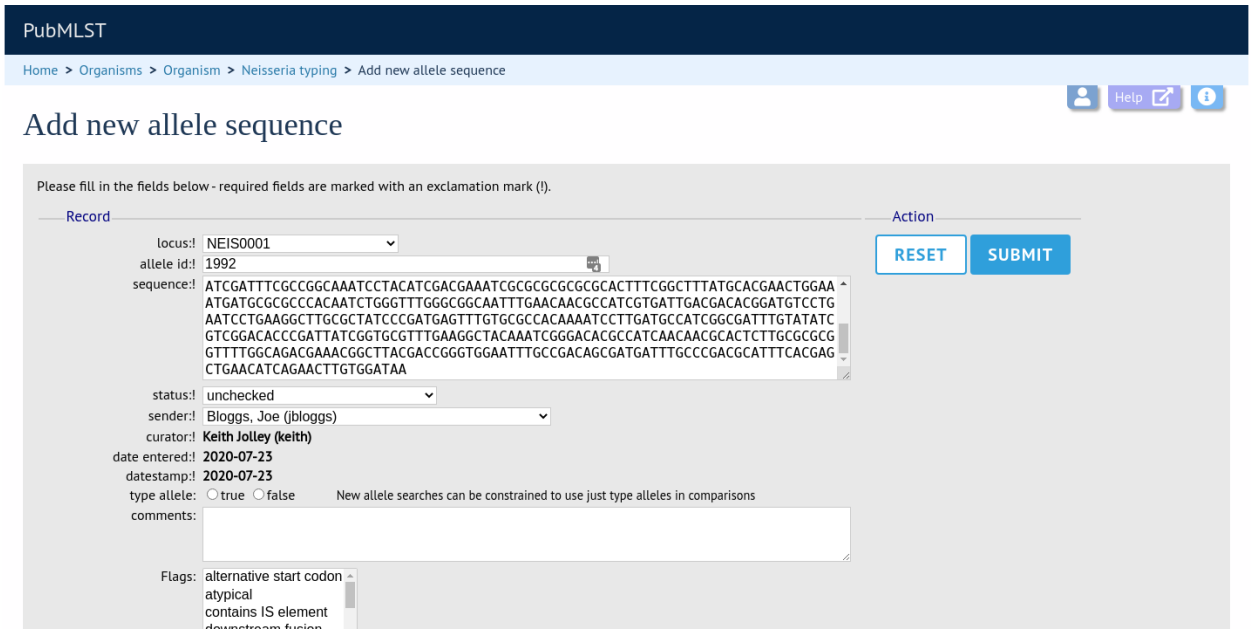


7.1.1 Individual allele curation

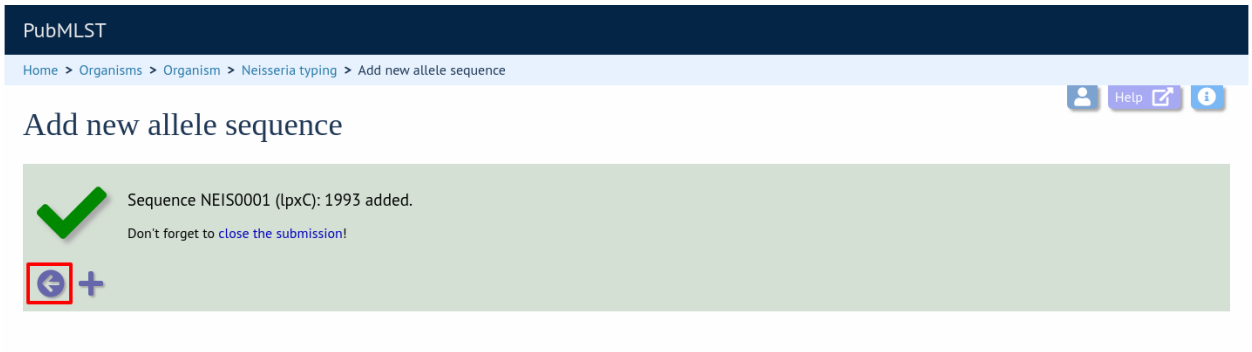
Individual sequences can be curated singly by clicking the 'Curate' links next to the sequence in the table. If you have supporting data attached to the submission, e.g. Sanger trace files then you may need to assess the submission based on the policy of the database.



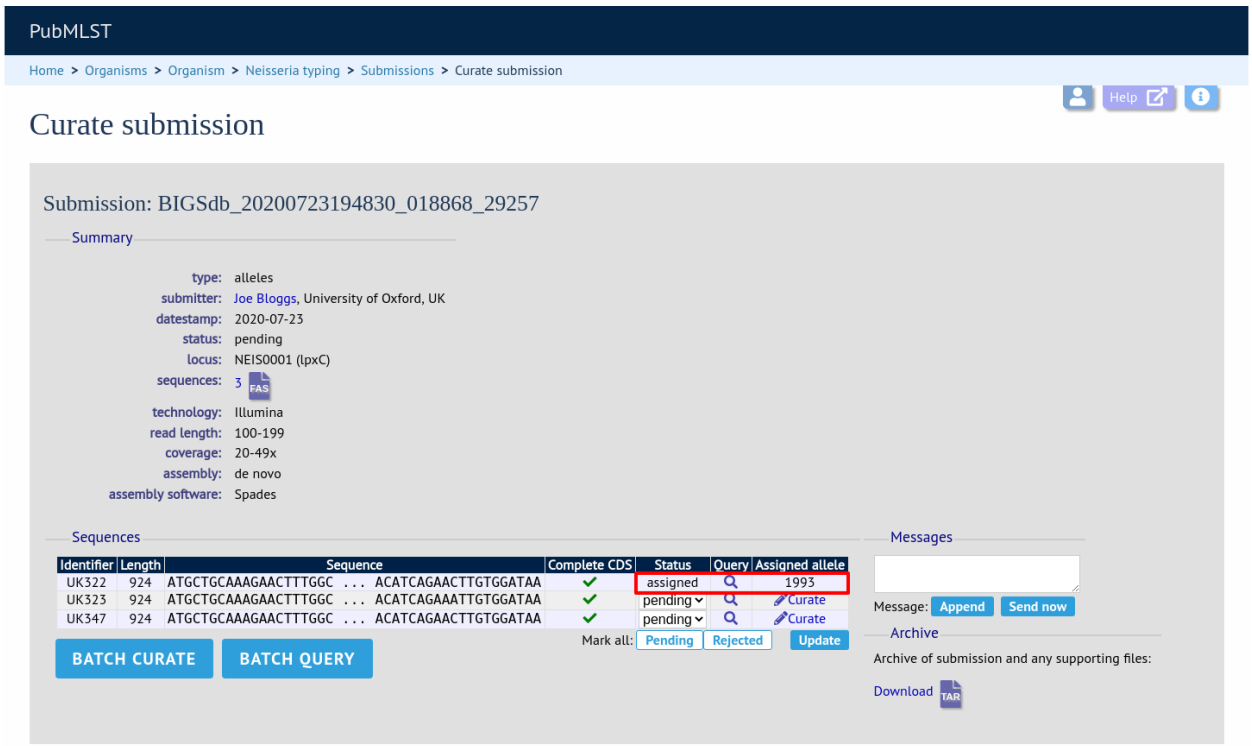
Clicking this link takes you to the curation interface *single sequence upload page*. The upload form will be filled with details from the submission. You may wish to manually change the status from the dropdown list of values.



Clicking 'Submit' from this form will define the new allele and add it to the database. A link on the confirmation page will take you back to the submission management page.



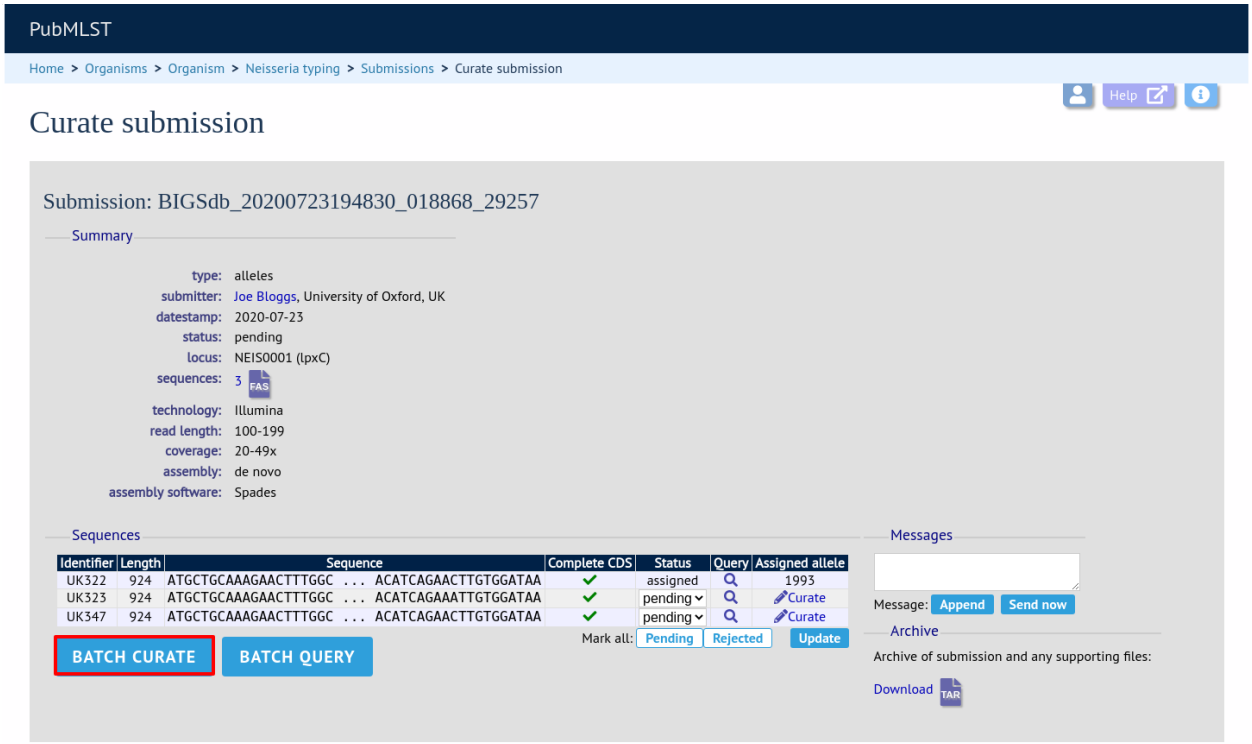
You will find that the status of the newly assigned sequence has changed in the summary table. The assigned value and status are determined on display and should always reflect the live database values.



7.1.2 Batch allele curation

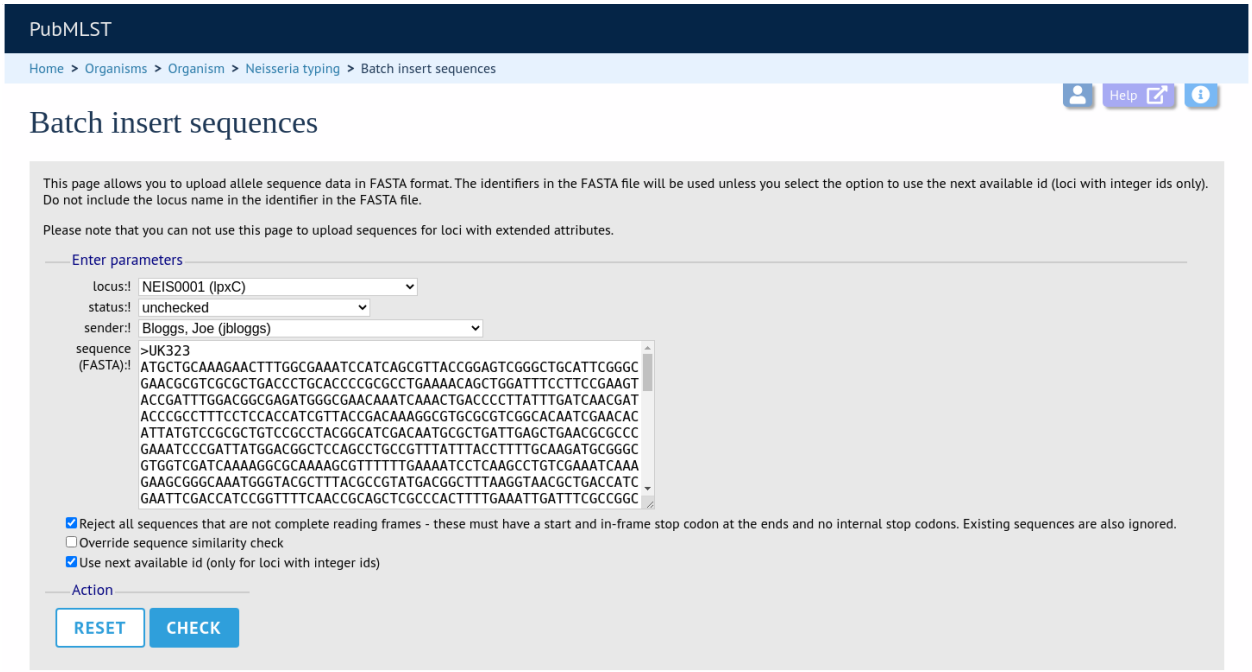
Often, you will want to batch upload submitted sequences. This can be done by clicking the 'Batch curate' button.

Note: Batch curation is only available for loci that do not have extended attributes defined. Entries for these loci require additional values set for these additional fields and so need to be handled individually.

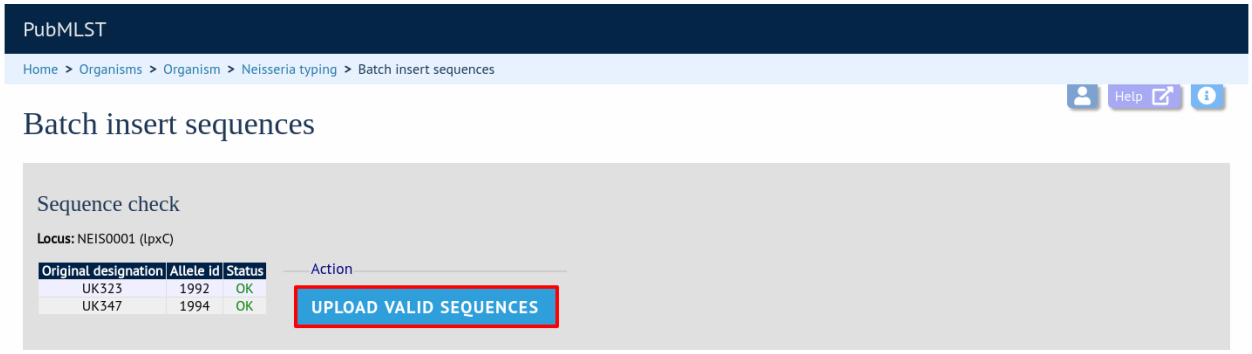


This takes you to the *batch FASTA upload page* in the curators' interface.

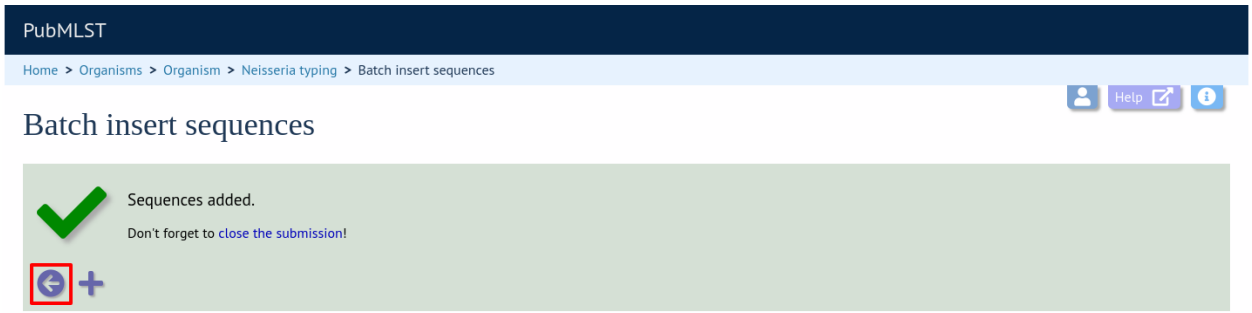
The upload form will be filled with details from the submission. You may wish to manually change the status from the dropdown list of values.



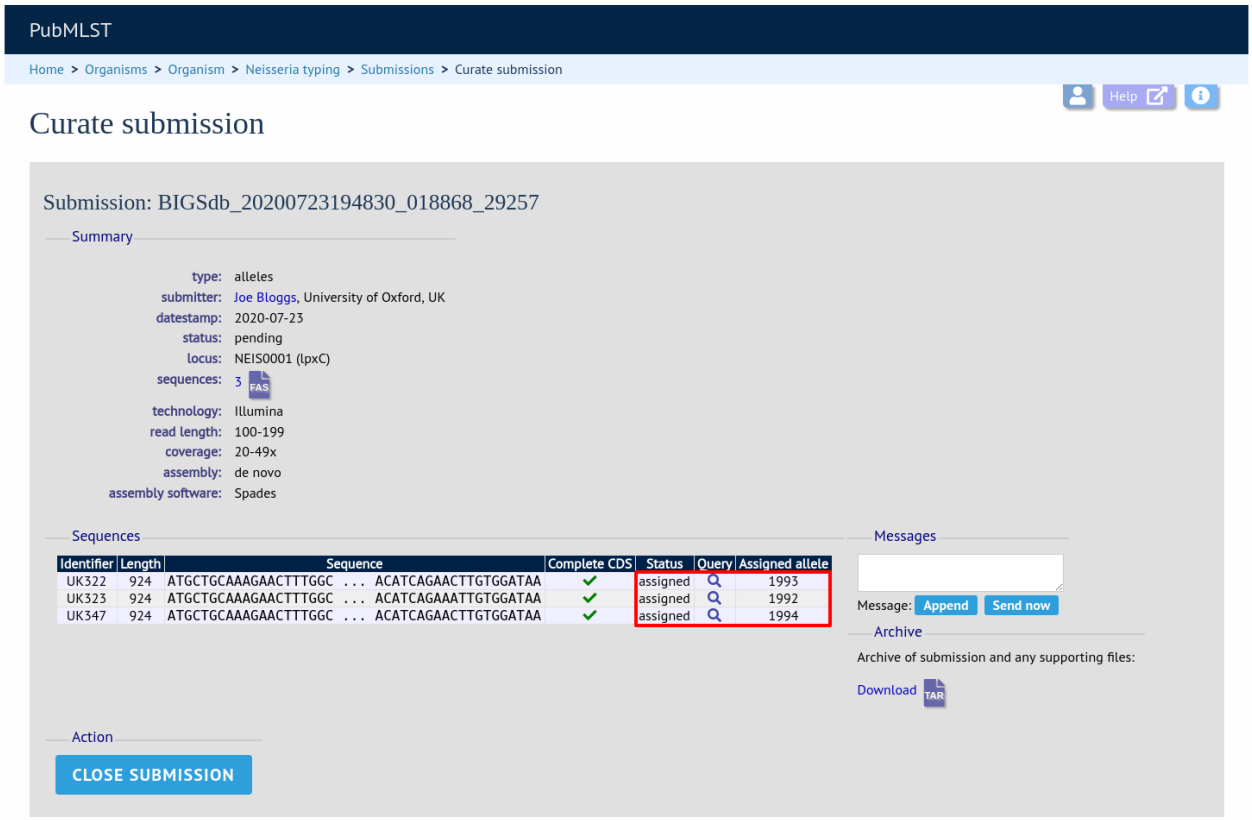
Click 'Check' on this form will perform some standard checks before allowing you to upload the sequences.



A link on the confirmation page will take you back to the submission management page.

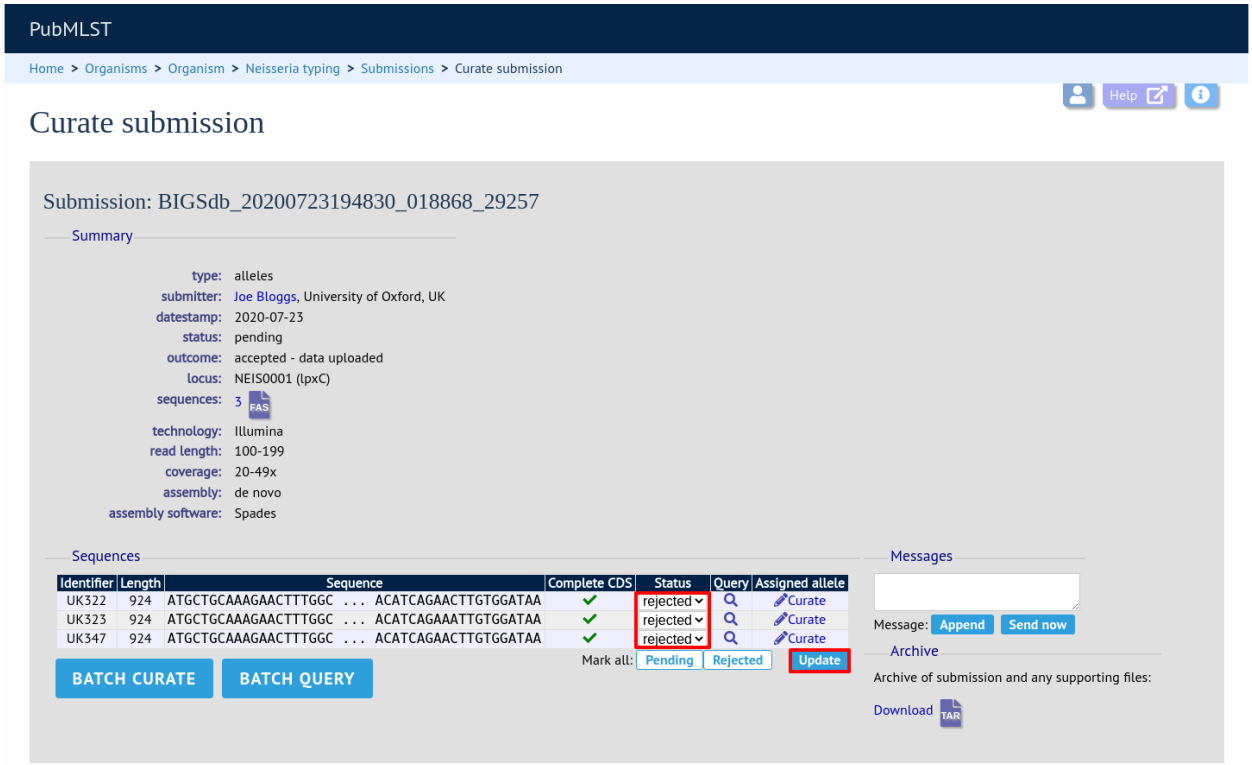


The status of the sequences should reflect their newly assigned status.



7.1.3 Rejecting sequences

Sometimes you may need to reject all, or some of, the sequences in a submission. You can do this by changing the value in the status dropdown box next to each sequence. Click 'Update' to make the change.

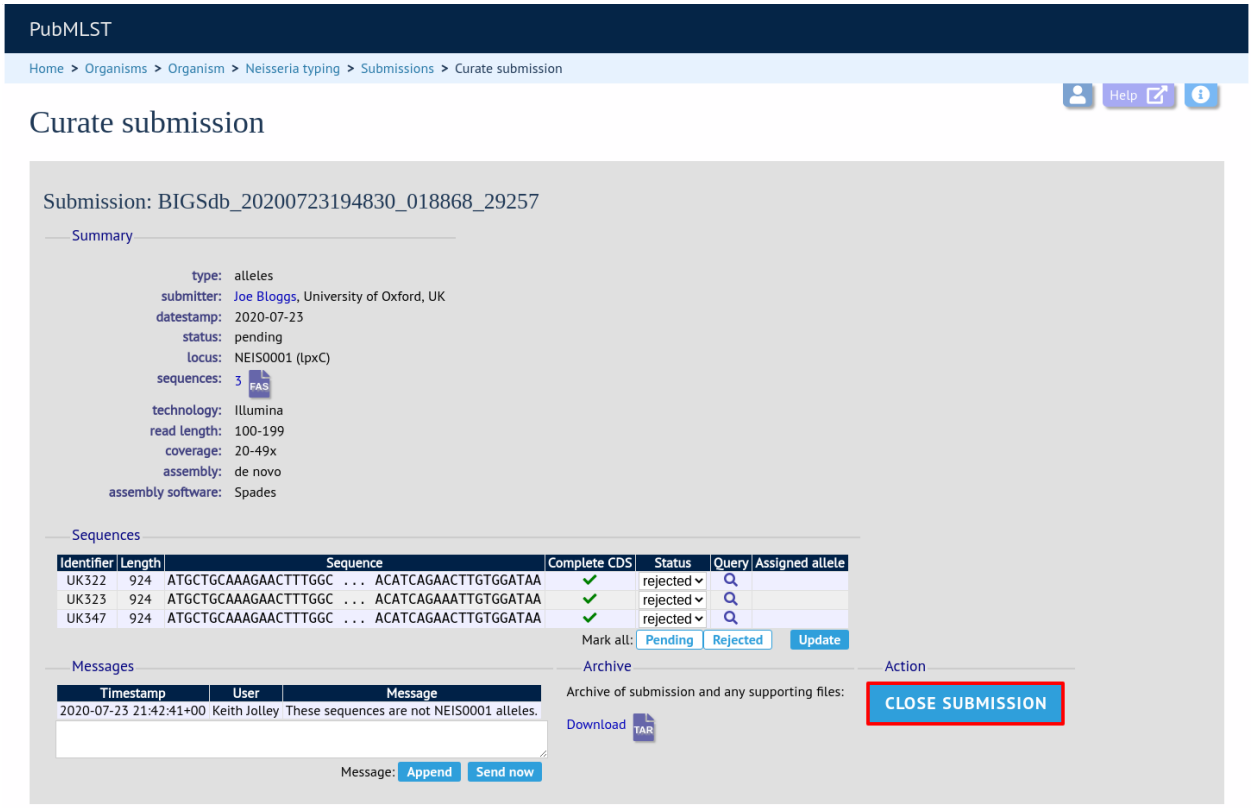


7.1.4 Requesting additonal information

You can send a message to the submitter by entering it in the Messages box and clicking 'Send now'. This will append a message to the submission and send an update to the submitter so that they can respond.

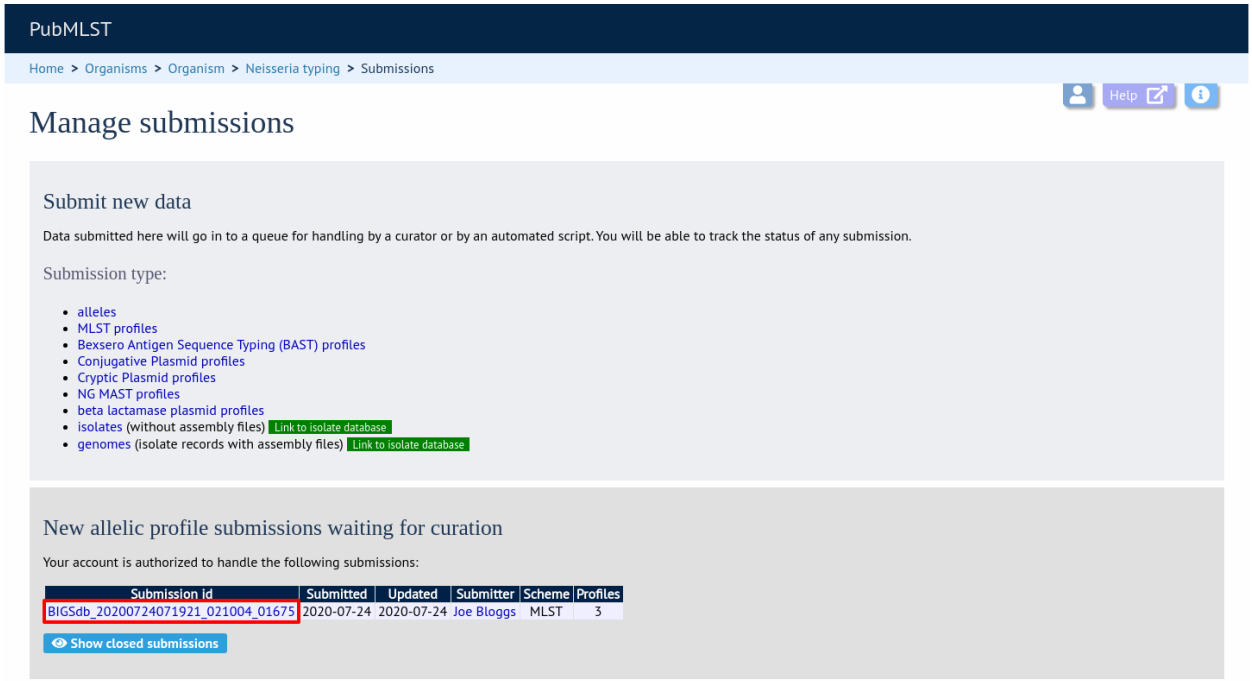
7.1.5 Closing the submission

You can add a message to the submitter by entering it in the message box and clicking 'Append'. Once sequences have all been either assigned or rejected, the 'Close submission' button will be displayed. Click this to close the submission. The submitter will be notified of their submission status.

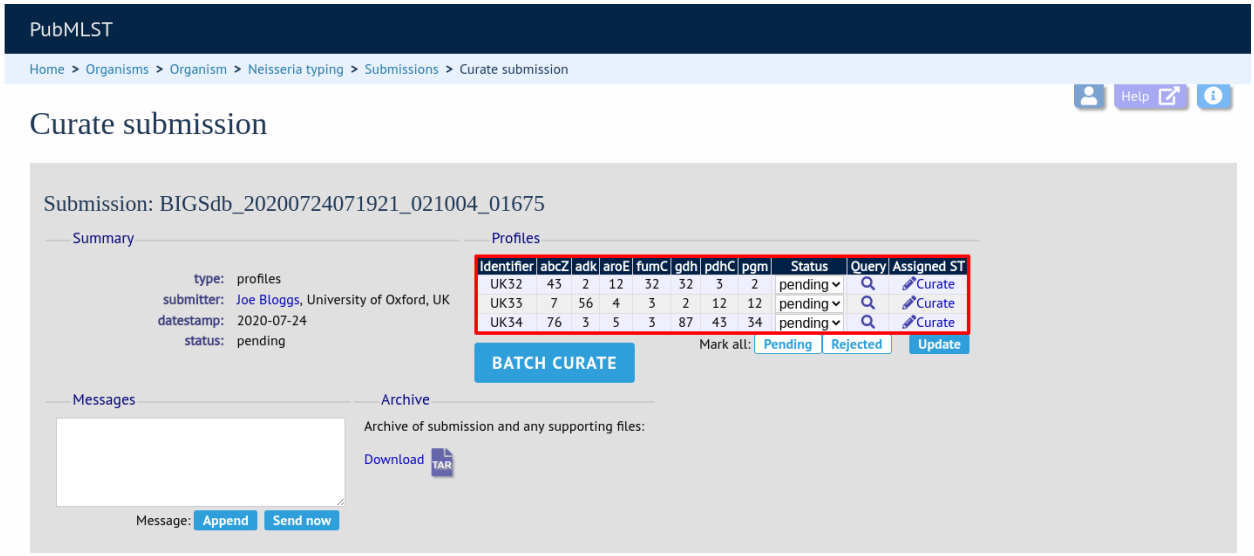


7.2 Profiles

Click the appropriate submission on the 'Manage submissions' page.

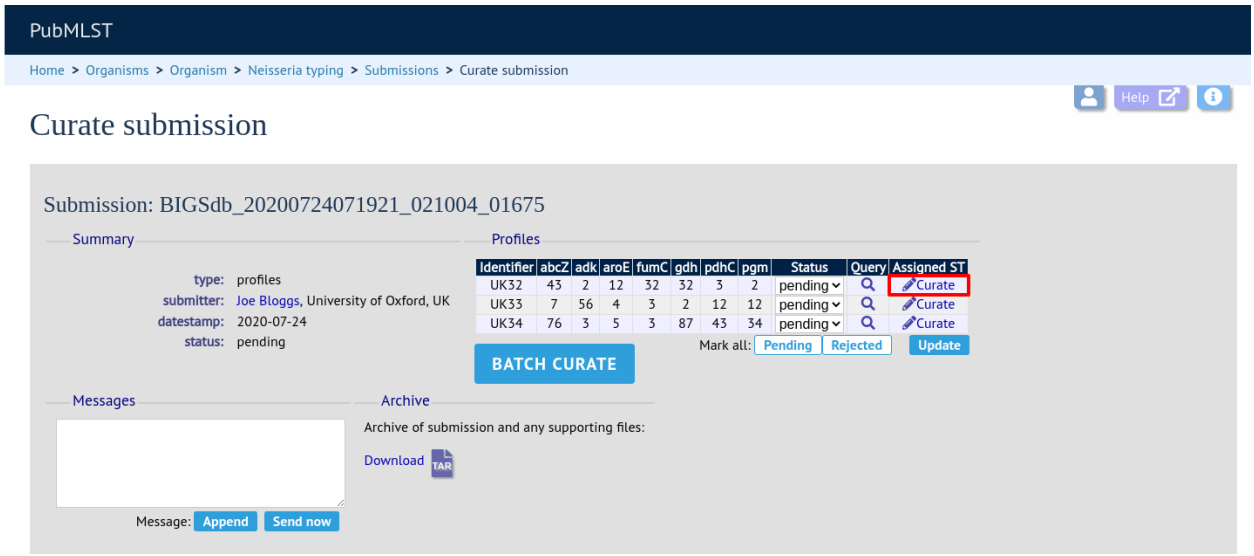


You will see a table summarizing the profiles in the submission and their current status.

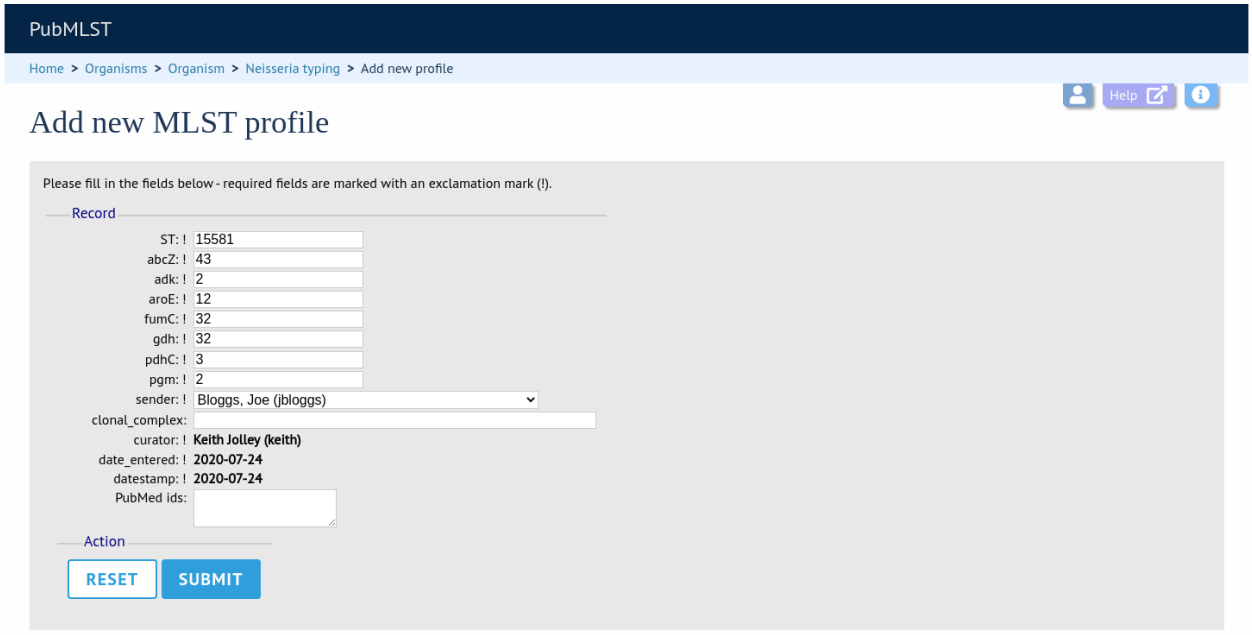


7.2.1 Individual profile curation

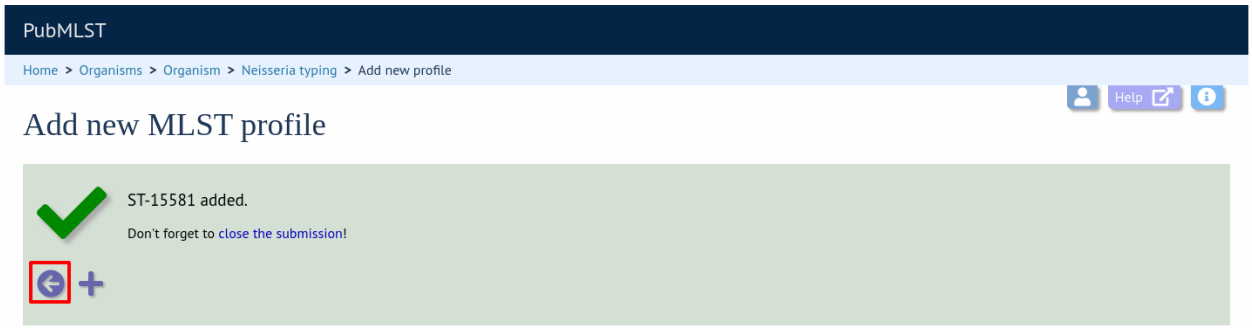
Individual profiles can be curated singly by clicking the 'Curate' links next to the profile in the table.



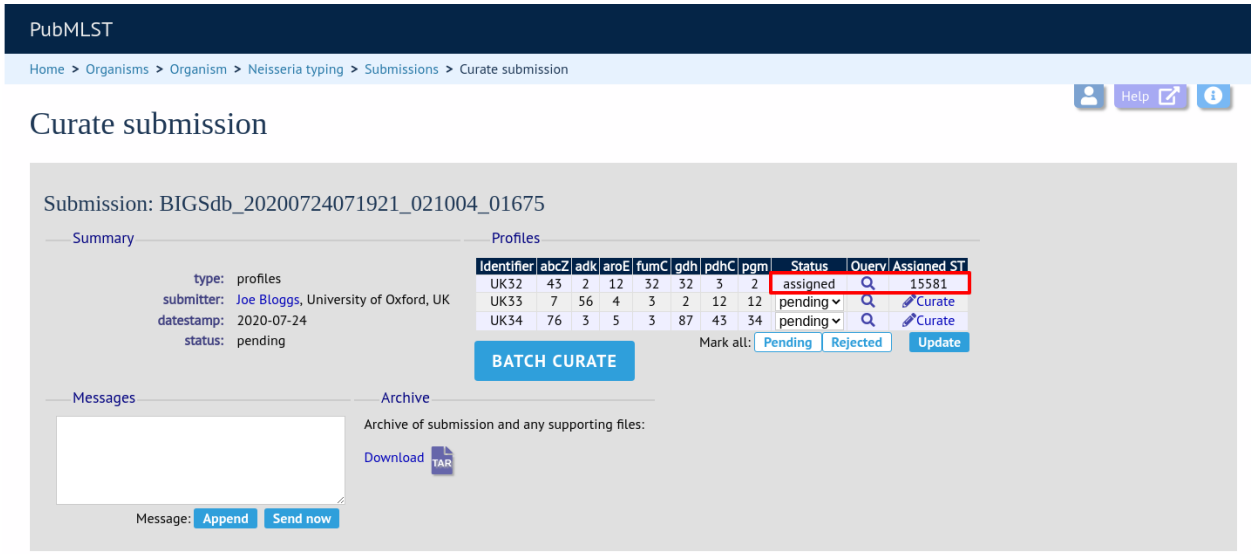
Clicking this link takes you to the curation interface *single profile upload page*. The upload form will be filled with details from the submission.



Clicking 'Submit' from this form will define the new profile and add it to the database. A link on the confirmation page will take you back to the submission management page.

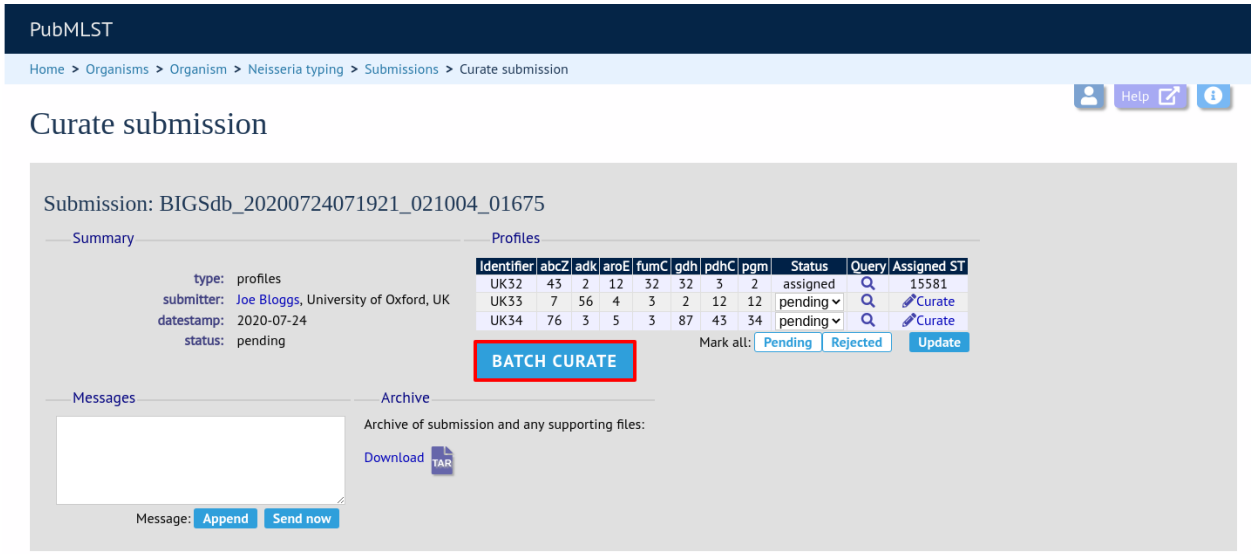


You will find that the status of the newly assigned profile has changed in the summary table. The assigned value and status are determined on display and should always reflect the live database values.



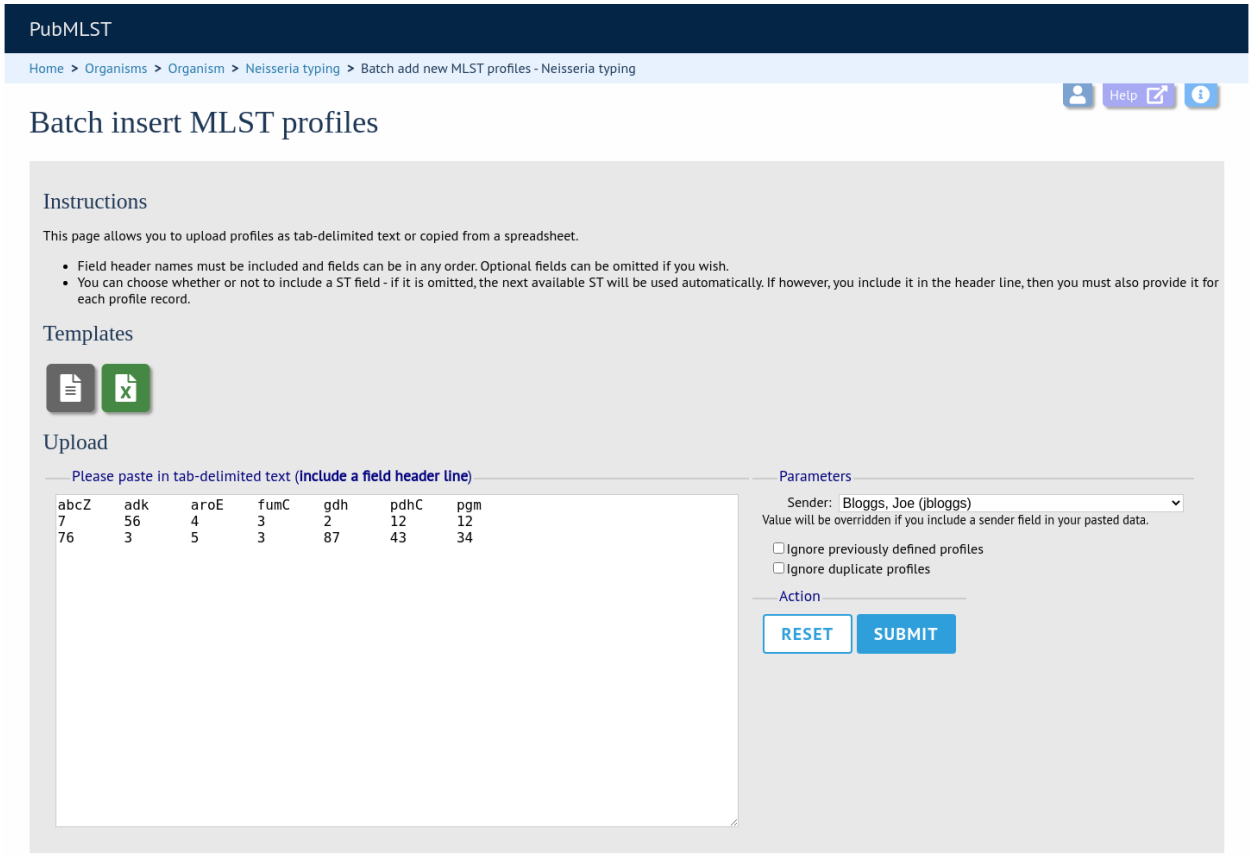
7.2.2 Batch profile curation

Often, you will want to batch upload submitted profiles. This can be done by clicking the 'Batch curate' button.

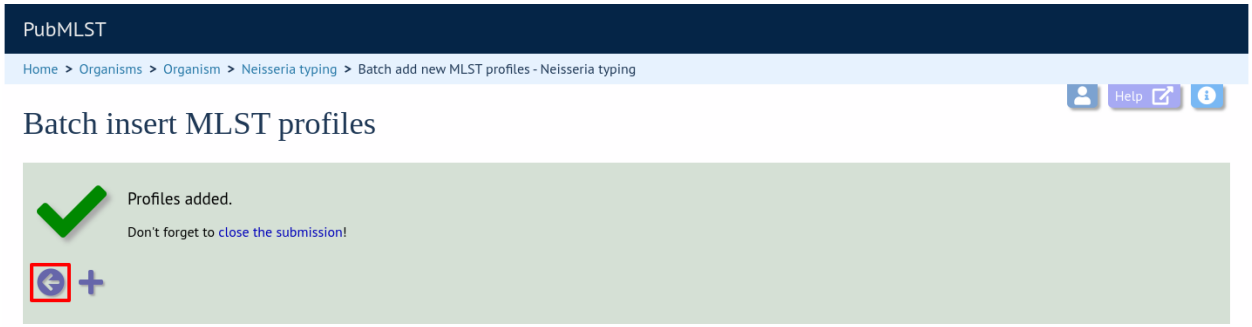


This takes you to the *batch profile upload page* in the curators' interface.

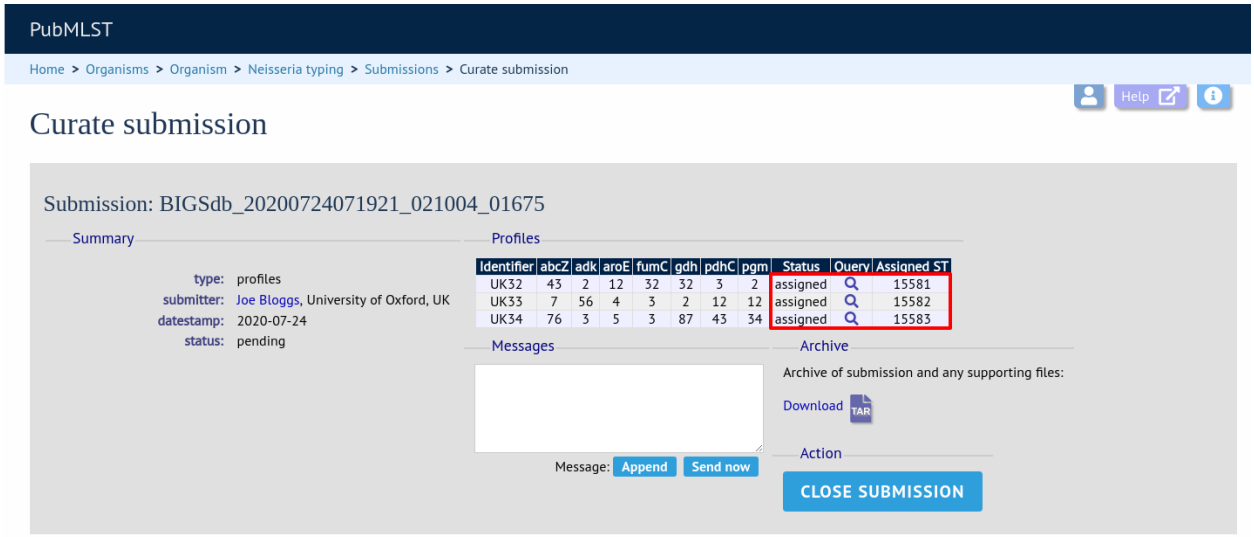
The upload form will be filled with details from the submission.



After upload, a link on the confirmation page leads back to the submission management page.

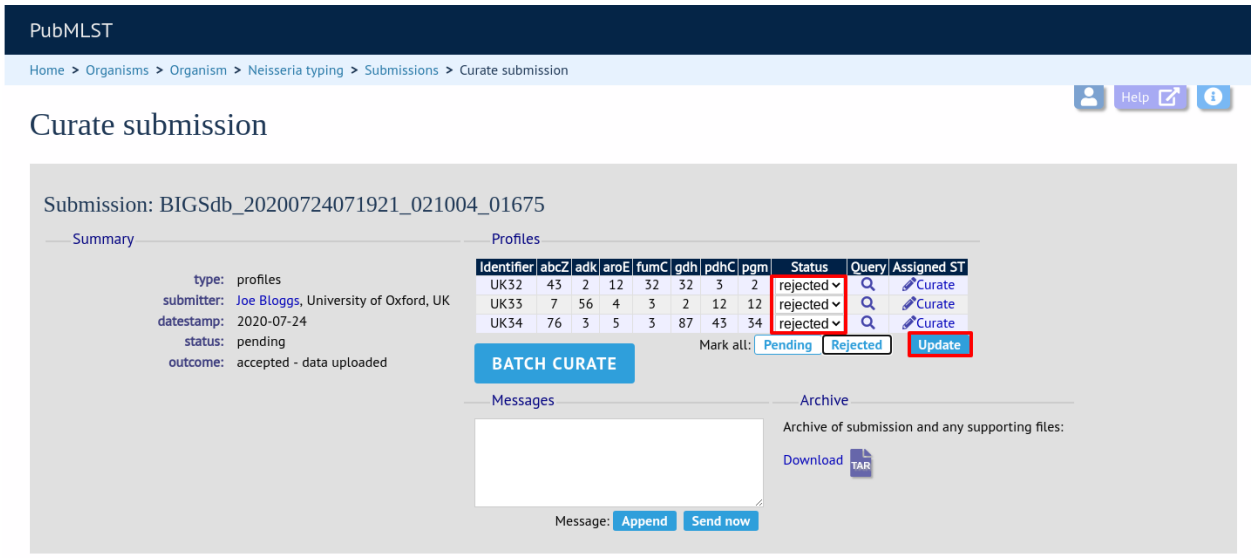


The status of the profiles should reflect their newly assigned status.



7.2.3 Rejecting profiles

Sometimes you may need to reject all, or some of, the profiles in the submission. This may be because isolate data had not been made available, against the policy of the database. You can do this by changing the value in the status dropdown box next to each profile. Click 'Update' to make the change.

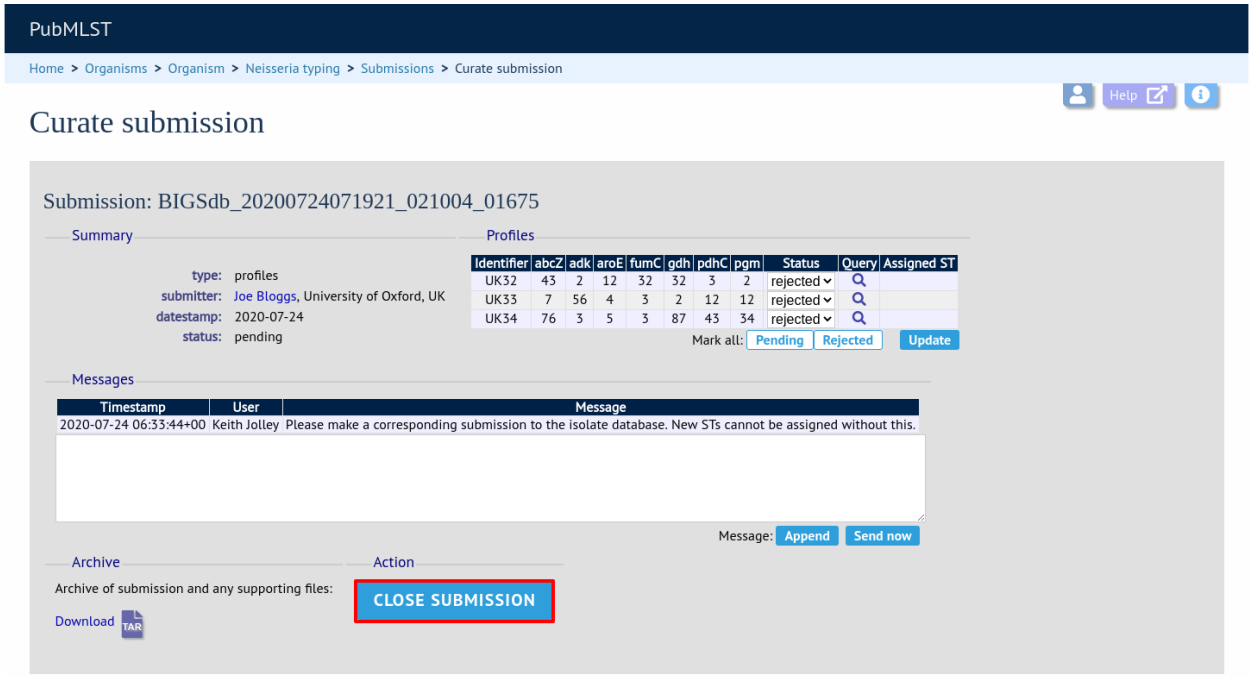


7.2.4 Requesting additonal information

You can send a message to the submitter by entering it in the Messages box and clicking 'Send now'. This will append a message to the submission and send an update to the submitter so that they can respond.

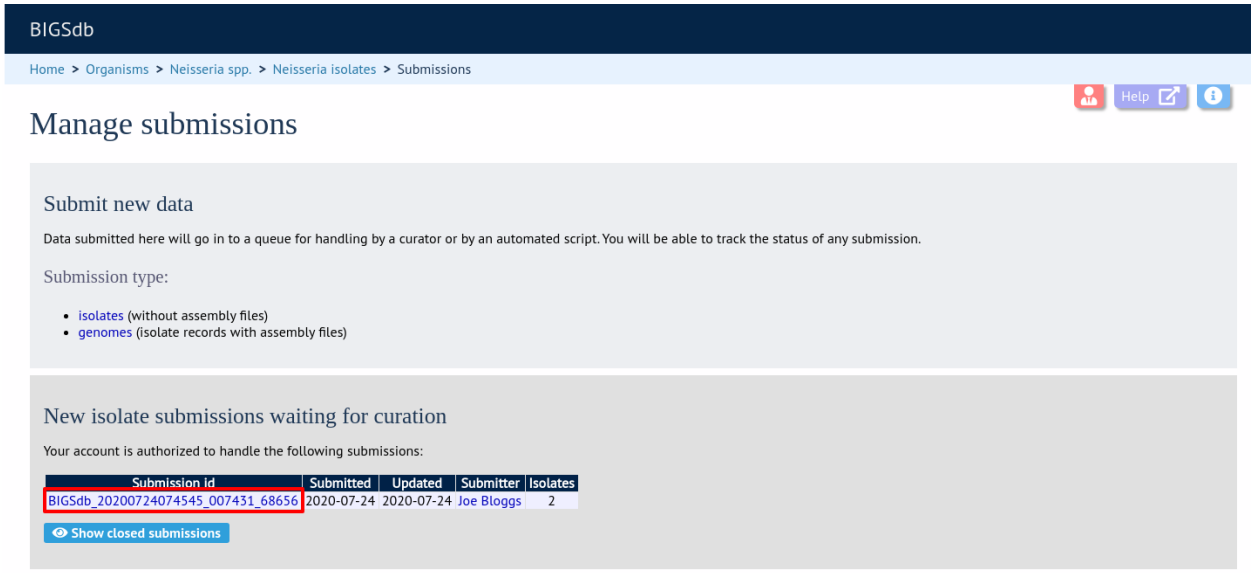
7.2.5 Closing the submission

You can add a message to the submitter by entering it in the message box and clicking 'Append'. Once profiles have all been either assigned or rejected, the 'Close submission' button will be displayed. Click this to close the submission. The submitter will be notified of their submission status.

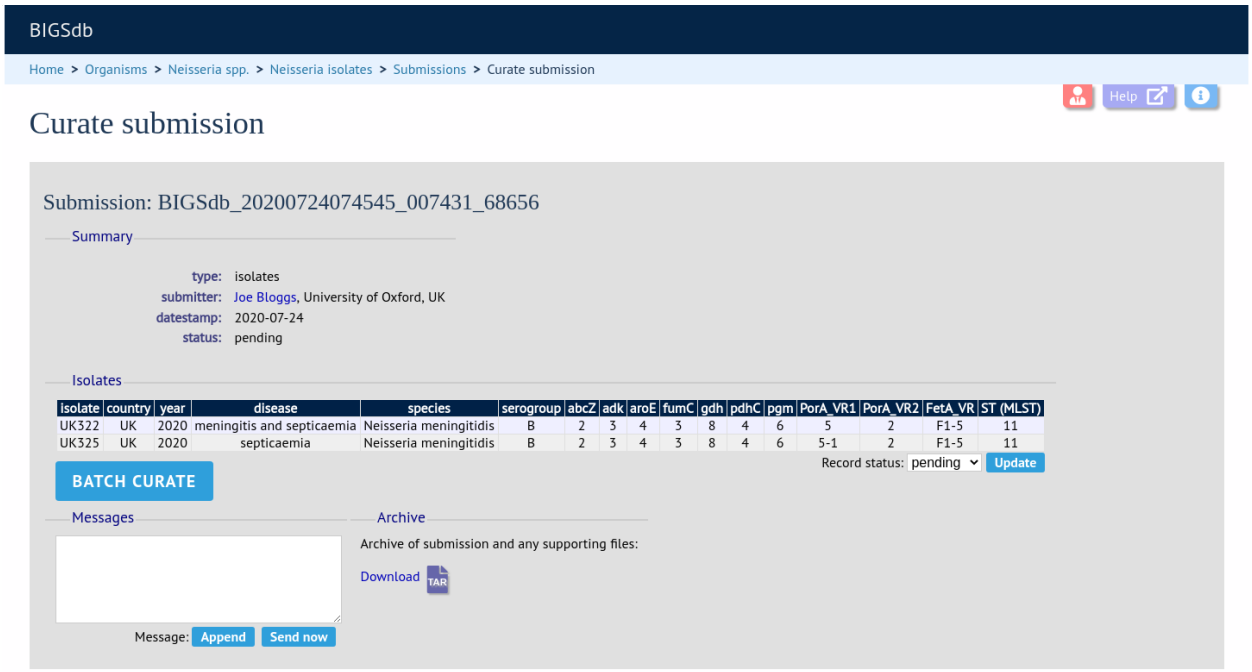


7.3 Isolates

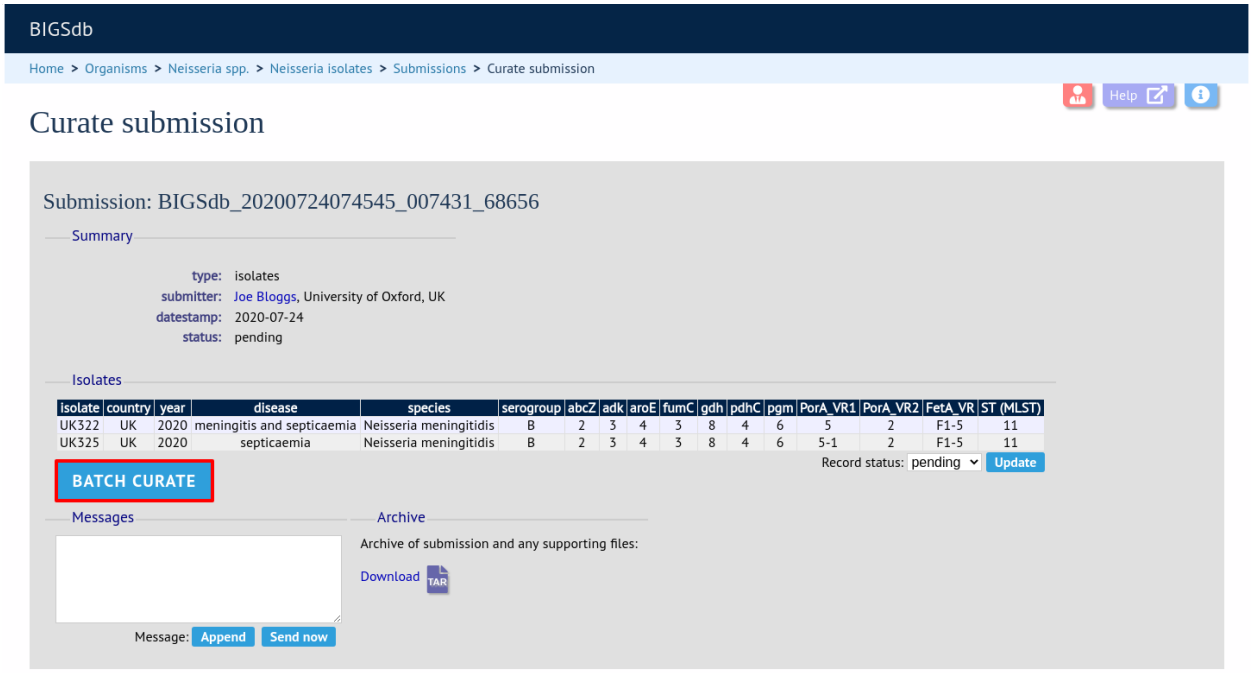
Clicking the appropriate submission on the 'Manage submissions' page.



You will see a table summarizing the submission.

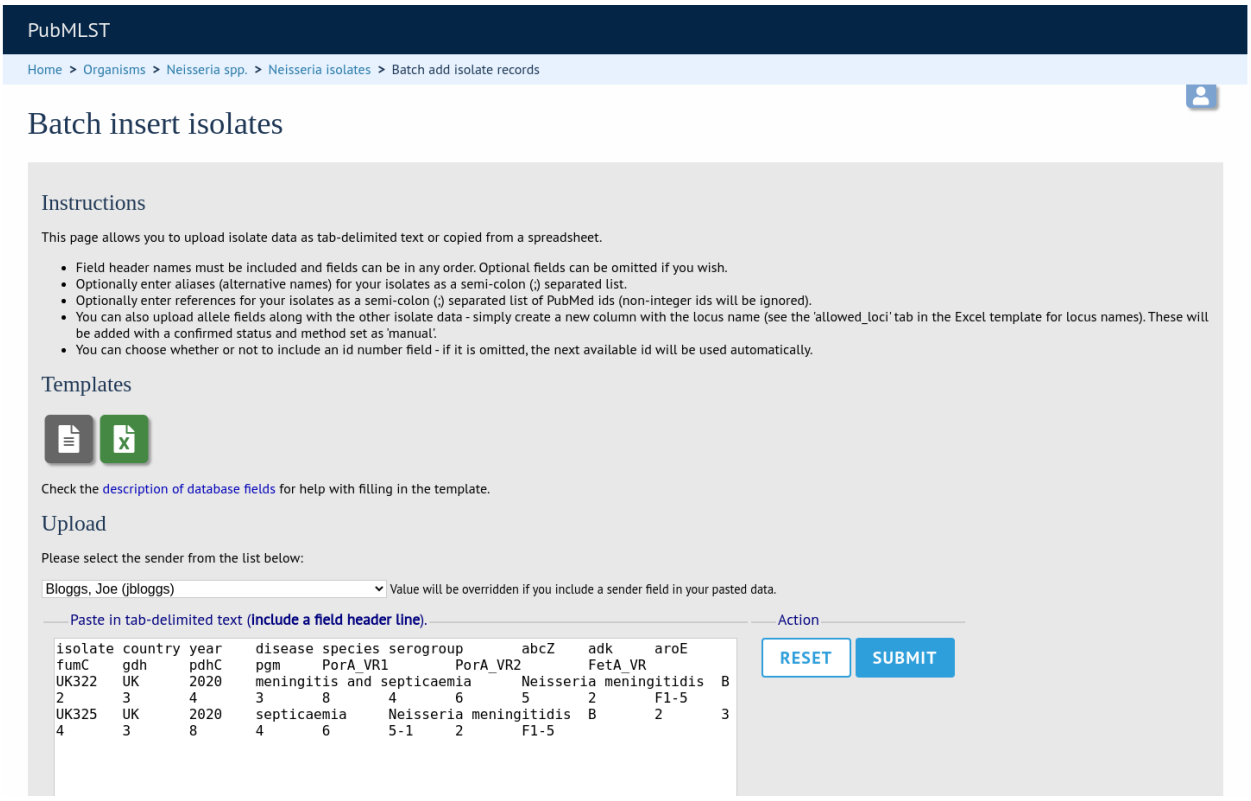


Click the ‘Batch curate’ button.



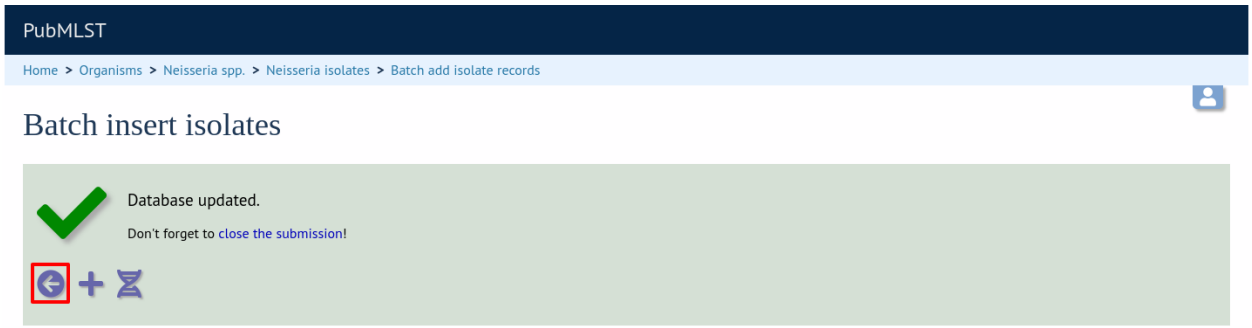
This will take you to the *batch isolate upload page* in the curators' interface.

The upload form will be filled with details from the submission.



Click submit to check and then import if there are no errors.

After upload, a link on the confirmation page leads back to the submission management page.



Note: Depending on the database policy, definitions of new scheme profiles, e.g. for MLST, may require submission of representative isolate records. Where this is the case, the curator will need to extract the new profile from the submitted record. The tab-delimited isolate text file can be downloaded from the archive of supporting files linked to the submission and used directly for *batch adding new profiles*. Alternatively, the curator could use the *Export functionality* of the database to generate the file required for batch profile definition after upload of the isolate data.

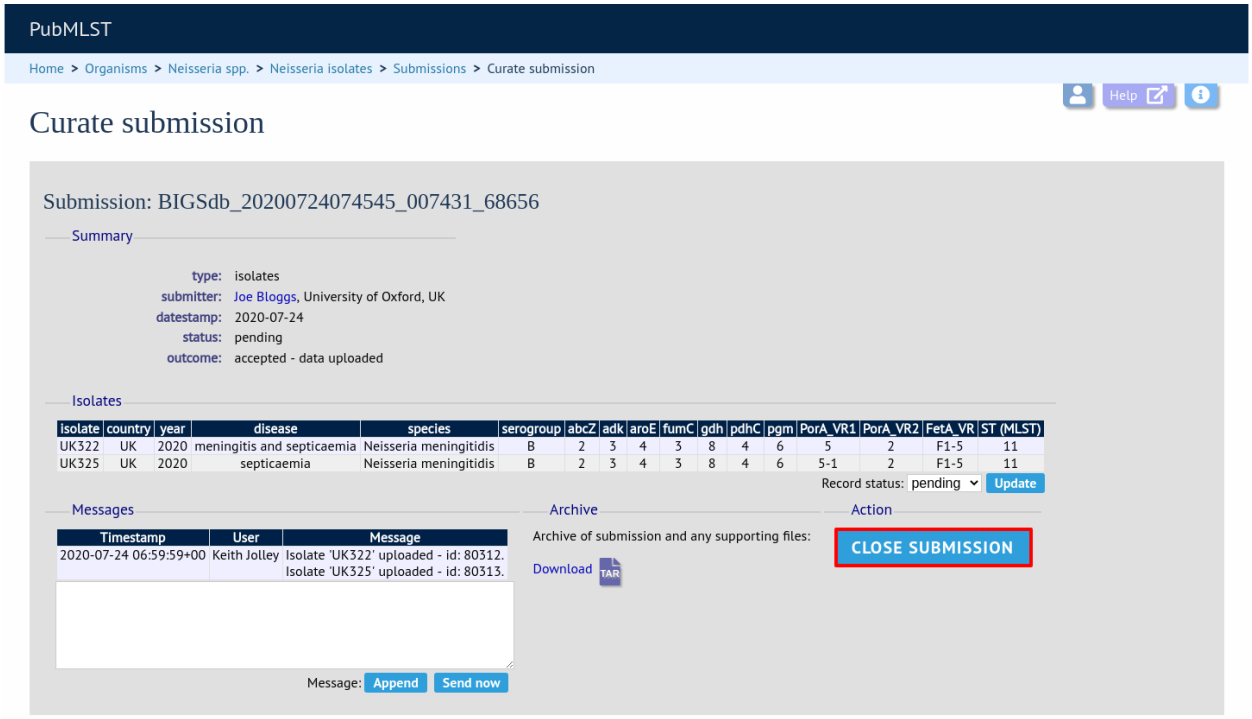
7.3.1 Requesting additonal information

You can send a message to the submitter by entering it in the Messages box and clicking 'Send now'. This will append a message to the submission and send an update to the submitter so that they can respond.

7.3.2 Closing the submission

You can add a message to the submitter by entering it in the message box and clicking 'Append'.

The 'Close submission' button will now appear. Click this to close the submission. The submitter will be notified of their submission status.


PubMLST
Home > Organisms > Neisseria spp. > Neisseria isolates > Submissions > Curate submission
Help
Curate submission
Submission: BIGSdb_20200724074545_007431_68656
Summary
type: isolates
submitter: Joe Bloggs, University of Oxford, UK
datestamp: 2020-07-24
status: pending
outcome: accepted - data uploaded
Isolates
Isolate country year disease species serogroup abcZ adk aroE fumC gdh pdhC pgm PorA_VR1 PorA_VR2 FetA_VR ST (MLST)
UK322 UK 2020 meningitis and septicaemia Neisseria meningitidis B 2 3 4 3 8 4 6 5 2 F1-5 11
UK325 UK 2020 septicaemia Neisseria meningitidis B 2 3 4 3 8 4 6 5-1 2 F1-5 11
Record status: pending
Update
Messages
Timestamp User Message
2020-07-24 06:59:59+00 Keith Jolley Isolate 'UK322' uploaded - id: 80312.
Isolate 'UK325' uploaded - id: 80313.
Message: Append Send now
Archive
Archive of submission and any supporting files:
Download
TAR
Action
CLOSE SUBMISSION

CHAPTER

EIGHT

OFFLINE CURATION TOOLS

8.1 Automated offline sequence tagging

Sequence tagging is the process of identifying alleles by scanning the sequence bin linked to an isolate record. Loci need to be defined in an external sequence definition database that contains the sequences for known alleles. The tagging function uses BLAST to identify sequences and will tag the specific sequence region with locus information and an allele designation if a matching allele is identified by reference to an external database.

There is a script called 'autotag.pl' in the BIGSdb package. This can be used to tag genome sequences from the command line.

Before autotag.pl can be run for the first time, a log file needs to be created. This can be created if it doesn't already exist with the following:

```
sudo touch /var/log/bigsdb_scripts.log
sudo chown bigsdb /var/log/bigsdb_scripts.log
```

The autotag.pl script should be installed in /usr/local/bin. It is run as follows:

```
autotag.pl --database <database configuration>
```

where <database configuration> is the name used for the argument 'db' when using the BIGSdb application.

If you have multiple processor cores available, use the –threads option to set the number of jobs to run in parallel. Isolates for scanning will be split among the threads.

The script must be run by a user that can both write to the log file and access the databases, e.g. the 'bigsdb' user (see 'Setting up the offline job manager').

A full list of options can be found by typing:

```
autotag.pl --help

NAME
    autotag.pl - BIGSdb automated allele tagger

SYNOPSIS
    autotag.pl --database NAME [options]

OPTIONS
-0, --missing
    Marks missing loci as provisional allele 0. Sets default word size to 15.
```

(continues on next page)

(continued from previous page)

```
--curator CURATOR ID
    Curator id to use for updates. By default -1 is used - there should be an
    autotagger account set with this id number.

-d, --database NAME
    Database configuration name.

-e, --exemplar
    Only use alleles with the 'exemplar' flag set in BLAST searches to identify
    locus within genome. Specific allele is then identified using a database
    lookup. This may be quicker than using all alleles for the BLAST search,
    but will be at the expense of sensitivity. If no exemplar alleles are set
    for a locus then all alleles will be used. Sets default word size to 15.

-f --fast
    Perform single BLAST query against all selected loci together. This will
    take longer to return any results but the overall scan should finish
    quicker. This method will also use more memory - this can be used with
    --exemplar to mitigate against this.

-h, --help
    This help page.

-i, --isolates LIST
    Comma-separated list of isolate ids to scan (ignored if -p used).

--isolate_list_file FILE
    File containing list of isolate ids (ignored if -i or -p used).

-I, --exclude_isolates LIST
    Comma-separated list of isolate ids to ignore.

-l, --loci LIST
    Comma-separated list of loci to scan (ignored if -s used).

-L, --exclude_loci LIST
    Comma-separated list of loci to exclude

-m, --min_size SIZE
    Minimum size of seqbin (bp) - limit search to isolates with at least this
    much sequence.

-n, --new_only
    New (previously untagged) isolates only.  Combine with --new_max_alleles
    if required.

--new_max_alleles ALLELES
    Set the maximum number of alleles that can be designated or sequences
    tagged before an isolate is not considered new when using the --new_only
    option.

-o, --order
```

(continues on next page)

(continued from previous page)

```
    Order so that isolates last tagged the longest time ago get scanned first
    (ignored if -r used).

--only_already_tagged
    Only check loci that already have a tag present (but no allele designation).
    This must be combined with the --already_tagged option or no loci will
    match. This option is used to perform a catch-up scan where a curator has
    previously tagged sequence regions prior to alleles being defined, without
    the need to scan all missing loci.

-p, --projects LIST
    Comma-separated list of project isolates to scan.

-P, --exclude_projects LIST
    Comma-separated list of projects whose isolates will be excluded.

-q, --quiet
    Only error messages displayed.

-r, --random
    Shuffle order of isolate ids to scan.

--reuse_blast
    Reuse the BLAST database for every isolate (when running --fast option).
    All loci will be scanned rather than just those missing from an isolate.
    Consequently, this may be slower if isolates have already been scanned,
    and for the first isolate scanned by a thread. On larger schemes, such as
    wgMLST, or when isolates have not been previously scanned, setting up the
    BLAST database can take a significant amount of time, so this may be
    quicker. This option is always selected if --new_only is used.

-R, --locus_regex REGEX
    Regex for locus names.

-s, --schemes LIST
    Comma-separated list of scheme loci to scan.

-t, --time MINS
    Stop after t minutes.

--threads THREADS
    Maximum number of threads to use.

-T, --already_tagged
    Scan even when sequence tagged (no designation).

-v, --view VIEW
    Isolate database view (overrides value set in config.xml).

-w, --word_size SIZE
    BLASTN word size.
```

(continues on next page)

(continued from previous page)

```
-x, --min ID
    Minimum isolate id.

-y, --max ID
    Maximum isolate id.
```

8.2 Defining exemplar alleles

Exemplar alleles are a subset of the total number of alleles defined for a locus that encompass the known diversity within a specified identity threshold. They can be used to speed up *autotagging* as the BLAST queries are performed against exemplars to identify the locus region in the genome followed by a direct database lookup of the sequence found to identify the exact allele found. This is usually combined with the autotagger –fast option.

Once exemplars have been defined you may also wish to set the fast_scan=”yes” option in the config.xml file. This enables their use for scanning within the web curators’ interface.

There is a script called ‘find_exemplars.pl’ in the BIGSdb scripts/maintenance directory.

A full list of options can be found by typing:

```
find_exemplars.pl --help

NAME
    find_exemplars.pl - Identify and mark exemplar alleles for use
    by tagging functions

SYNOPSIS
    find_exemplars.pl --database NAME   [options]

OPTIONS

--database NAME
    Database configuration name.

--datatype DNA|peptide
    Only define exemplars for specified data type (DNA or peptide)

--exclude_loci LIST
    Comma-separated list of loci to exclude

--help
    This help page.

--loci LIST
    Comma-separated list of loci to scan (ignored if -s used).

--locus_regex REGEX
    Regex for locus names.

--schemes LIST
    Comma-separated list of scheme loci to scan.
```

(continues on next page)

(continued from previous page)

```
--update
    Update exemplar flags in database.

--variation DISSIMILARITY
    Value for percentage identity variation that exemplar alleles
    cover (smaller value will result in more exemplars). Default: 10.
```

8.3 Automated offline allele definition

There is a script called 'scannew.pl' in the BIGSdb scripts/automation directory. This can be used to identify new alleles from the command line. This can (optionally) upload these to a sequence definition database.

Before scannew.pl can be run for the first time, a log file needs to be created. This can be created if it doesn't already exist with the following:

```
sudo touch /var/log/bigsdb_scripts.log
sudo chown bigsdb /var/log/bigsdb_scripts.log
```

The scannew.pl script should be installed in /usr/local/bin. It is run as follows:

```
scannew.pl --database <database configuration>
```

where <database configuration> is the name used for the argument 'db' when using the BIGSdb application.

If you have multiple processor cores available, use the –threads option to set the number of jobs to run in parallel. Loci for scanning will be split among the threads.

The script must be run by a user that can both write to the log file and access the databases, e.g. the 'bigsdb' user (see 'Setting up the offline job manager').

A full list of options can be found by typing:

```
scannew.pl --help

NAME
  scannew.pl - BIGSdb automated allele definer

SYNOPSIS
  scannew.pl --database NAME [options]

OPTIONS
-a, --assign
    Assign new alleles in definitions database.

--allow_frameshift
    Allow sequences to contain a frameshift so that the length is not a
    multiple of 3, or an internal stop codon. To be used with
    --coding_sequences option to allow automated curation of pseudogenes.
    New alleles assigned will be flagged either 'frameshift' or 'internal stop
    codon' if appropriate.  Essentially, combining these two options only
    checks that the sequence starts with a start codon and ends with a stop
```

(continues on next page)

(continued from previous page)

```
    codon.

--allow_subsequences
    Allow definition of sub- or super-sequences. By default these will not
    be assigned.

-A, --alignment INT
    Percentage alignment (default: 100).

-B, --identity INT
    Percentage identity (default: 99).

-c, --coding_sequences
    Only return complete coding sequences.

--curator CURATOR ID
    Curator id to use for updates. By default -1 is used - there should be an
    autodefiner account set with this id number.

-d, --database NAME
    Database configuration name.

-h, --help
    This help page.

-i, --isolates LIST
    Comma-separated list of isolate ids to scan (ignored if -p used).

--isolate_list_file FILE
    File containing list of isolate ids (ignored if -i or -p used).

-I, --exclude_isolates LIST
    Comma-separated list of isolate ids to ignore.

-l, --loci LIST
    Comma-separated list of loci to scan (ignored if -s used).

-L, --exclude_loci LIST
    Comma-separated list of loci to exclude.

-m, --min_size SIZE
    Minimum size of seqbin (bp) - limit search to isolates with at least this
    much sequence.

-n, --new_only
    New (previously untagged) isolates only.

-o, --order
    Order so that isolates last tagged the longest time ago get scanned first
    (ignored if -r used).

-p, --projects LIST
```

(continues on next page)

(continued from previous page)

```
    Comma-separated list of project isolates to scan.

-P, --exclude_projects LIST
    Comma-separated list of projects whose isolates will be excluded.

-q, --quiet
    Only error messages displayed.

-r, --random
    Shuffle order of isolate ids to scan.

-R, --locus_regex REGEX
    Regex for locus names.

-s, --schemes LIST
    Comma-separated list of scheme loci to scan.

-t, --time MINS
    Stop after t minutes.

--threads THREADS
    Maximum number of threads to use.

--type_alleles
    Only use alleles with the 'type_allele' flag set to identify locus.
    If a partial match is found then a full database lookup will be performed
    to identify any known alleles. Using this option will constrain the search
    space so that allele definitions don't become more variable over time. Note
    that you must have at least one allele defined as a type allele for a locus
    if you use this option otherwise you will not find any matches!

-T, --already_tagged
    Scan even when sequence tagged (no designation).

-v, --view VIEW
    Isolate database view (overrides value set in config.xml).

-w, --word_size SIZE
    BLASTN word size.

-x, --min ID
    Minimum isolate id.

-y, --max ID
    Maximum isolate id.
```

8.4 Calculating assembly stats

Basic assembly statistics are calculated automatically by the database engine as contigs are added to the sequence bin. These include the number of contigs, total length and the N50 value. Some calculations, such as %GC, number of Ns, and the number of gaps however, require offline analysis since these involve inspecting the nucleotide content of each contig. These can be calculated by running the update_assembly_stats.pl script. You can choose to run this against all databases on the system or against a specific database.

Only one copy of the script can run at a time, but it will stop gracefully if it detects another copy running, so it is recommended that the script is run regularly using a CRON job and the –quiet option. This ensures that records are updated shortly after they have been uploaded.

Once calculated, all assembly statistics can then be *used in isolate queries*.

A full list of options can be found by typing:

```
update_assembly_stats.pl --help

NAME
    update_assembly_stats.pl - Perform/update calculation of
    assembly GC, N and gap stats.

SYNOPSIS
    update_assembly_stats.pl [options]

OPTIONS

--database DATABASE CONFIG
    Database configuration name. If not included then all isolate databases
    defined on the system will be checked.

--exclude CONFIG NAMES
    Comma-separated list of config names to exclude.

--help
    This help page.

--quiet
    Only show errors.

--refresh_days DAYS
    Refresh records last analysed longer that the number of days set. By
    default, only records that have not been analysed will be checked.
```

8.5 Predicting species based on rMLST analysis

The *rMLST plugin* predicts species based on matches to rMLST alleles exclusively found in a particular species. It uses the PubMLST API to query either a genome sequence or rMLST allele designations to identify the species. When the analysis is run using the plugin, the results are also stored with the isolate record and can then be displayed within the isolate information page. This analysis can also be run offline using the update_rmlst_species.pl script.

Only one copy of the script can run at a time, but it will stop gracefully if it detects another copy running, so it is recommended that the script is run regularly using a CRON job and the –quiet option. This ensures that records are updated shortly after they have been uploaded.

A full list of options can be found by typing:

```
update_rmlst_species.pl --help

NAME
    update_rmlst_species.pl - Perform/update species id check

SYNOPSIS
    update_rmlst_species.pl [options]

OPTIONS

--database DATABASE CONFIG
    Database configuration name. If not included then all isolate databases
    defined on the system will be checked.

--exclude CONFIG NAMES
    Comma-separated list of config names to exclude.

--help
    This help page.

--last_run_days DAYS
    Only run for a particular isolate when the analysis was last performed
    at least the specified number of days ago.

--quiet
    Only show errors.

--refresh_days DAYS
    Refresh records last analysed longer that the number of days set. By
    default, only records that have not been analysed will be checked.
```

8.6 Cleanly interrupting offline curation

Sometimes you may wish to stop running autotagger or allele autodefiner jobs as they can be run for a long time and as CRON jobs. If these are running in single threaded mode, the easiest way is to simply send a kill signal to the process, i.e. identify the process id using 'top', e.g. 23232 and then

```
kill 23232
```

The scripts should respond to this signal within a couple of seconds, clean up all their temporary files and write the history log (where appropriate). Do not use 'kill -9' as this will terminate the processes immediately and not allow them to clean up.

If these scripts are running using multiple threads, then you need to cleanly kill each of these. The simplest way to terminate all autotagger jobs is to type

```
pkill autotag
```

The parent process will wait for all forked processes to cleanly terminate and then exit itself.

Similarly, to terminate all allele autodefiner jobs, type

```
pkill scannew
```

8.7 Uploading contigs from the command line

There is a script called upload_contigs.pl in the BIGSdb scripts/maintenance directory. This can be used to upload contigs from a local FASTA file for a specified isolate record.

The upload_contigs.pl script should be installed in /usr/local/bin. It is run as follows:

```
upload_contigs.pl --database <NAME> --isolate <ID> --file <FILE>
    --curator <ID> --sender <ID>
```

The script must be run by a user who has the appropriate database permissions and the local configuration settings should be modified to match the database user account to be used. The default setting uses the 'apache' user which is used by the BIGSdb web interface.

A full list of options can be found by typing:

```
upload_contigs.pl --help

NAME
    upload_contigs.pl - Upload contigs to BIGSdb isolate database

SYNOPSIS
    upload_contigs.pl --database NAME --isolate ID --file FILE
         --curator ID --sender ID [options]

OPTIONS
-a, --append
    Upload contigs even if isolate already has sequences in the bin.

-c, --curator ID
```

(continues on next page)

(continued from previous page)

```
    Curator id number.

-d, --database NAME
    Database configuration name.

-f, --file FILE
    Full path and filename of contig file.

-h, --help
    This help page.

-i, --isolate ID
    Isolate id of record to upload to.

-m, --method METHOD
    Method, e.g. 'Illumina', default 'unknown'.

--min_length LENGTH
    Exclude contigs with length less than value.

-s, --sender ID
    Sender id number.
```

8.8 Populating geographic point lookup values

If a field has the geographic_point_lookup attribute set to 'yes' in the *config.xml file*, field values can be mapped to GPS coordinates to facilitate mapping.

These lookup values can be populated using the gp_town_lookups.pl script found in the scripts/maintenance directory. This script uses the Geonames dataset that contains GPS coordinates for towns and cities with populations of at least 1000 people. The dataset is included in the datasets/Geonames directory.

A full list of options can be found by typing:

```
gp_town_lookups.pl --help

NAME
    geography_point_lookups.pl - Populate geography_point_lookup table
    to set city/town GPS coordinates for mapping.

SYNOPSIS
    geography_point_lookups.pl --database NAME --field FIELD --geodataset DIR

    Run this to populate any unassigned values in the geography_point_lookup
    table.

OPTIONS

--database NAME
    Database configuration name.
```

(continues on next page)

(continued from previous page)

```
--feature_code CODE
    Geonames feature code. See http://www.geonames.org/export/codes.html.
    Default is 'P' (towns/cities).

--field FIELD
    Name of field. This should have the geography_point_lookup attribute set to
    'yes' in config.xml.

--geodataset DIR
    Directory containing the Geonames dataset.

--help
    This help page.

--min_population POPULATION
    Set the minimum population for town to assign. Note that all entries in the
    Geonames database has population, so setting this attribute may result in
    some values not being assigned, but can ensure that only high-confidence
    values are used.

--quiet
    Only show error messages.

--tmp_dir DIR
    Location for temporary files. Defaults to /var/tmp/.
```

CHAPTER

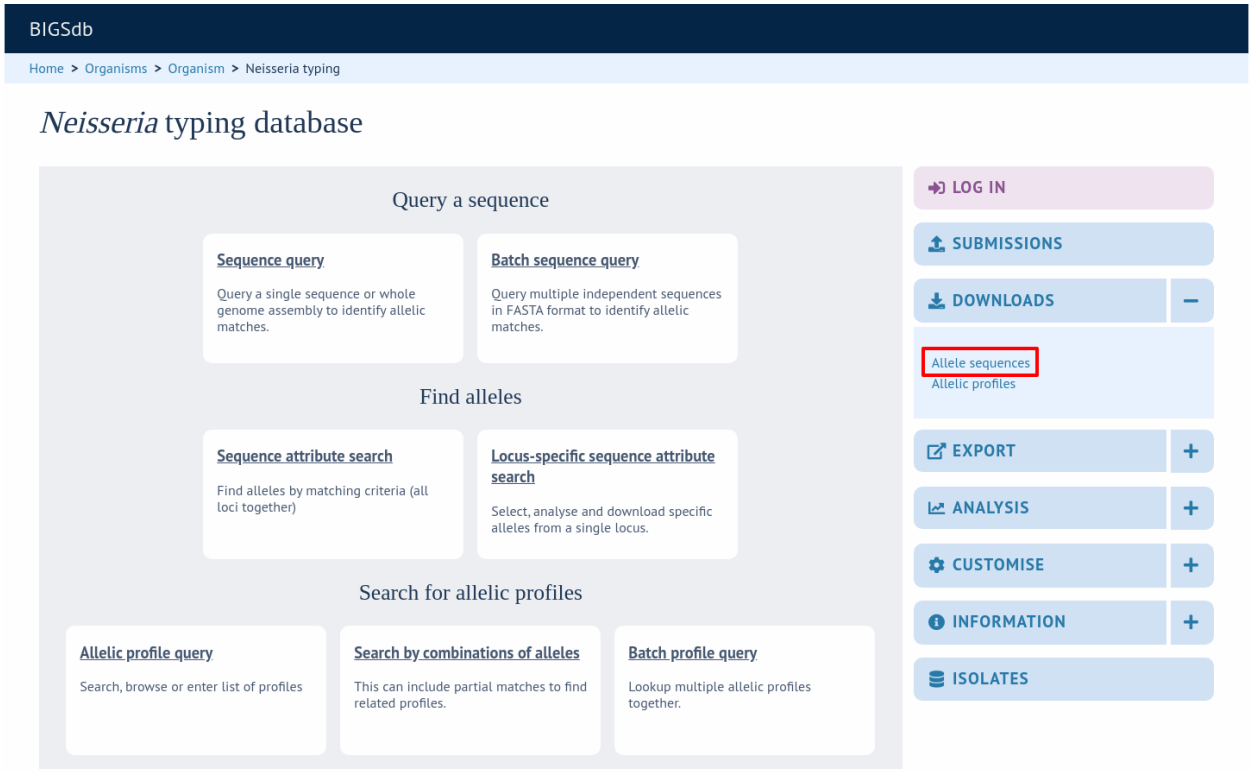
NINE

DEFINITION DOWNLOADS

The sequence definition database defines alleles, i.e. links an allele identifier to a sequence. It also defines scheme, e.g. MLST, profiles.

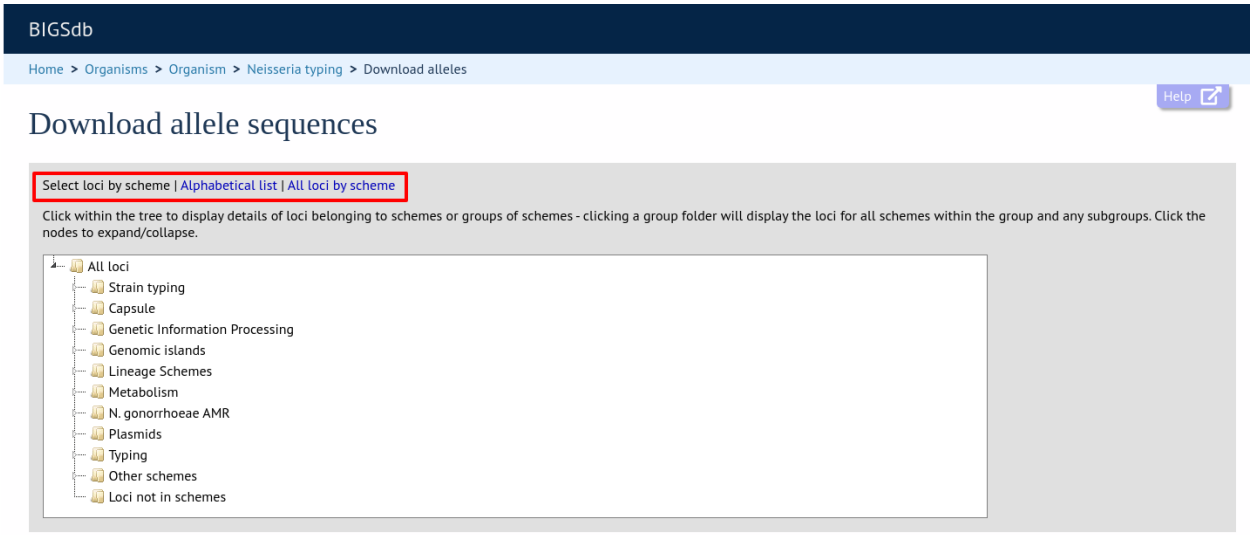
9.1 Allele sequence definitions

Click the 'Allele sequences' link in the 'Downloads' section.

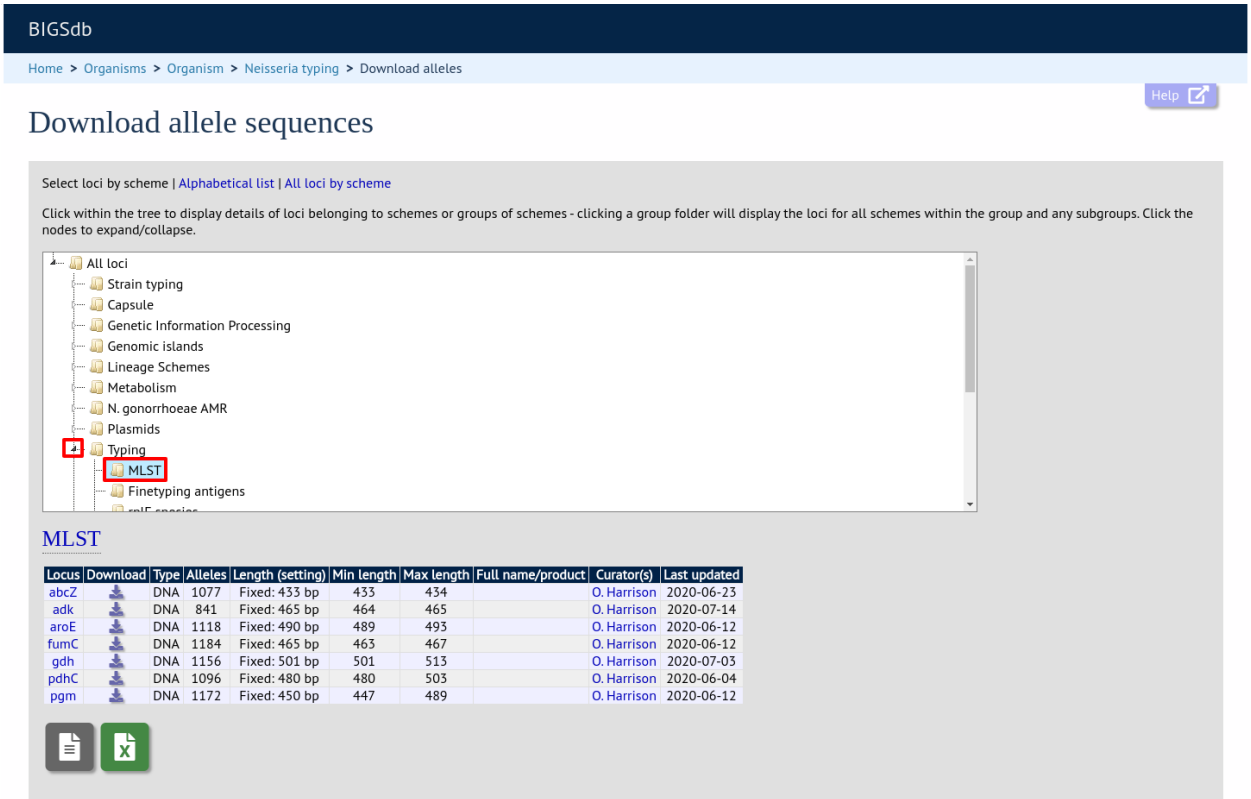


Depending on the database, you may see either a hierarchical scheme tree or a table of loci. You can choose to display links either by scheme using the scheme tree, as an alphabetical list or a page of all schemes, by selecting the approrpiate link at the top of the page.

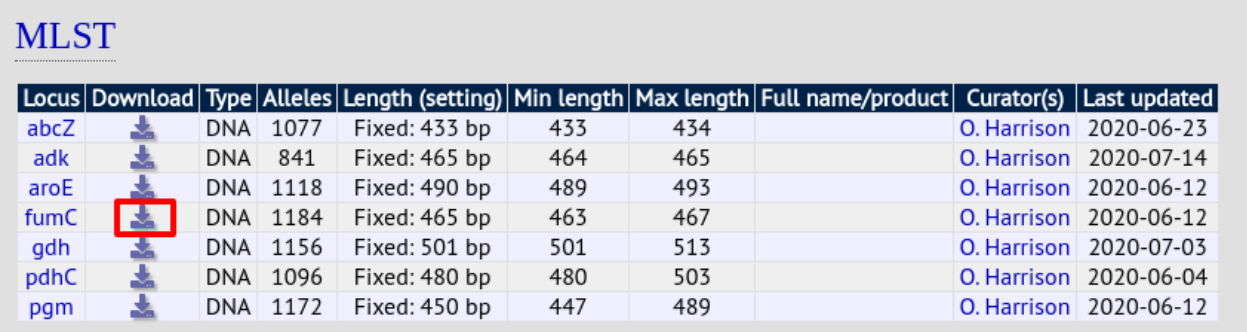
9.1.1 Scheme tree



You can drill down through the tree by clicking branch nodes. Clicking the labels of internal nodes will display tables of all schemes belonging to that scheme group. Clicking the labels of terminal nodes will display that single scheme table.



Click the download link for the required locus



MLST

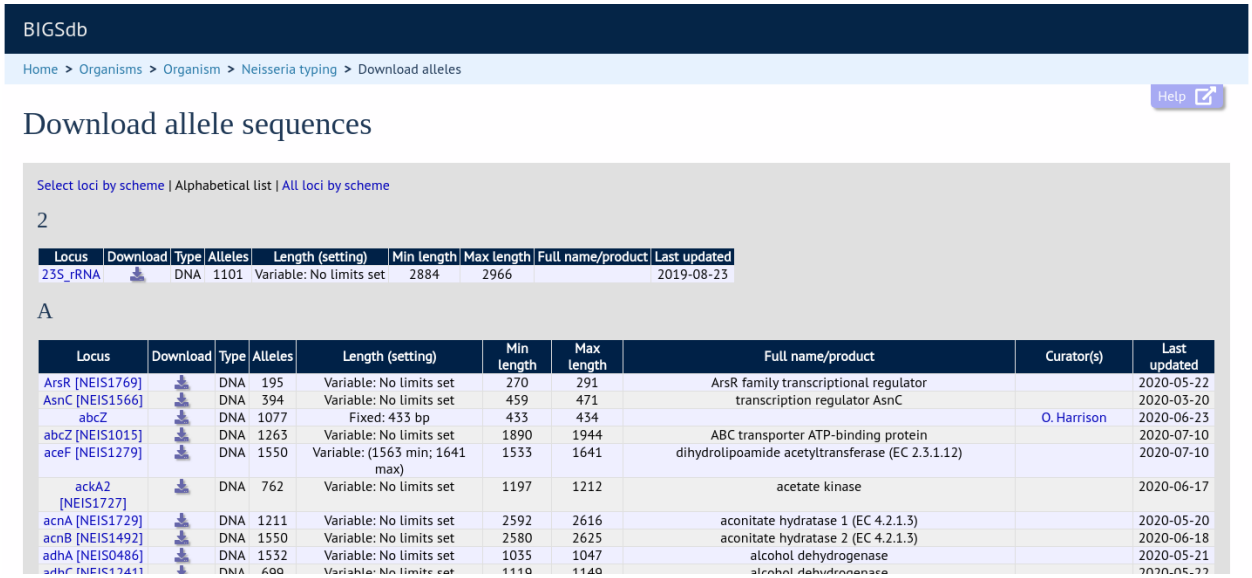
Locus	Download	Type	Alleles	Length (setting)	Min length	Max length	Full name/product	Curator(s)	Last updated
abcZ		DNA	1077	Fixed: 433 bp	433	434		O. Harrison	2020-06-23
adk		DNA	841	Fixed: 465 bp	464	465		O. Harrison	2020-07-14
aroE		DNA	1118	Fixed: 490 bp	489	493		O. Harrison	2020-06-12
fumC		DNA	1184	Fixed: 465 bp	463	467		O. Harrison	2020-06-12
gdh		DNA	1156	Fixed: 501 bp	501	513		O. Harrison	2020-07-03
pdhC		DNA	1096	Fixed: 480 bp	480	503		O. Harrison	2020-06-04
pgm		DNA	1172	Fixed: 450 bp	447	489		O. Harrison	2020-06-12

Alleles will be downloaded in FASTA format, e.g.

```
>fumC_1
GAAGCCTTGGGCGGACGCGATGCCGCCGTTGCCGCTTCGGGCGCATTGAAAACGCTGGCG
GCAAGCCTGAATAAAATCGCCAACGACATCCGCTGGCTGGCAAGCGGCCCGCGCTGCGGT
TTGGGCGAAATCAAAATCCCCGAAAACGAGCCGGGTTCGTCCATCATGCCGGGCAAAGTC
AACCCGACCCAATGCGAAGCGATGACCATGGTGTGCTGCCAAGTGTTCGGCAACGACGTT
ACCATCGGTATGGCGGGCGCGTCGGGCAATTTCGAGCTGAACGTCTATATGCCCGTCATC
GCCTACAACCTCTTGCAATCCATCCGCCTGTTGGGCGACGCGTGCAACAGCTTCAACGAA
CACTGCGCCGTCGGCATTGAACCCGTACCGGAAAAAATCGACTATTTCCTGCACCATTCC
CTGATGCTCGTTACCGCGTTAAACCGCAAAATCGGTTACGAAAAC
>fumC_2
GAAGCCTTGGGCGGACGCGATGCCGCCGTTGCCGCTTCGGGCGCATTGAAAACGCTGGCG
GCAAGCCTGAATAAAATCGCCAACGACATCCGCTGGCTGGCAAGCGGCCCGCGCTGCGGT
TTGGGCGAAATCAAAATCCCCGAAAACGAGCCGGGTTCGTCCATCATGCCGGGCAAAGTC
AACCCGACCCAATGCGAAGCGATGACCATGGTGTGCTGCCAAGTGTTCGGCAACGACGTT
ACCATCGGCATGGCGGGCGCGTCGGGCAATTTCGAGCTGAACGTCTATATGCCCGTTATC
GCCTACAACCTCTTGCAATCCATCCGCCTCTTGGGCGACGCGTGCAACAGCTTCAACGAA
CACTGCGCCATCGGCATCGAACCCGTACCGGAAAAAATCGACTATTTCCTGCACCATTCC
CTGATGCTCGTTACCGCGTTAAACCGCAAAATCGGTTACGAAAAC
>fumC_3
GAAGCCTTGGGCGGACGCGATGCCGCCGTTGCCGCTTCGGGCGCATTGAAAACGCTGGCG
GCAAGCCTGAATAAAATCGCCAACGACATCCGCTGGCTGGCAAGCGGCCCGCGCTGCGGT
TTGGGCGAAATCAAAATCCCCGAAAACGAGCCGGGTTCGTCCATCATGCCGGGCAAAGTC
AACCCGACCCAATGCGAAGCGATGACCATGGTGTGCTGCCAAGTGTTCGGCAACGACGTT
ACCATCGGCATGGCGGGCGCGTCGGGCAATTTCGAGCTGAACGTCTATATGCCCGTTATC
GCCTACAACCTCTTGCAATCCATCCGCCTGTTGGGCGACGCGTGCAACAGCTTCAACGAA
CACTGCGCCGTCGGCATCGAACCCGTACCGGAAAAAATCGACTATTTCCTGCACCATTCC
CTGATGCTGGTTACTGCGTTAAACCGTAAAATCGGCTACGAAAAC
```

9.1.2 Alphabetical list

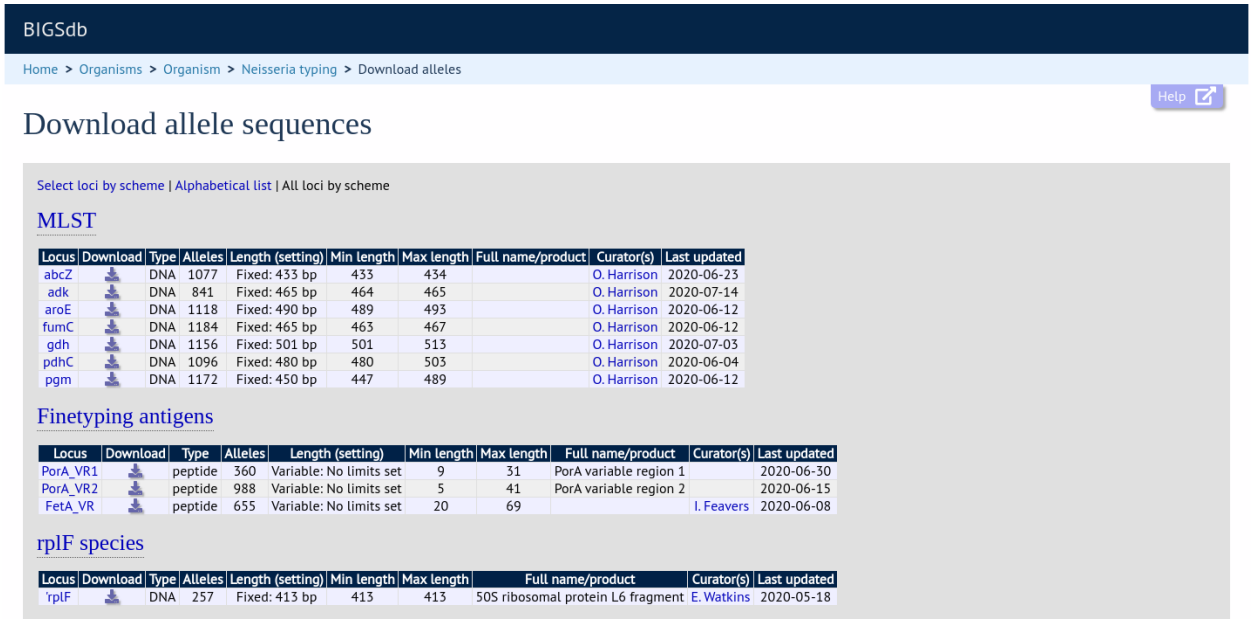
Loci can be displayed in an alphabetical list. Loci will be grouped in to tables by initial letter. If common names are set for loci, they will be listed by both primary and common names.



Click the download links for the required locus.

9.1.3 All loci by scheme

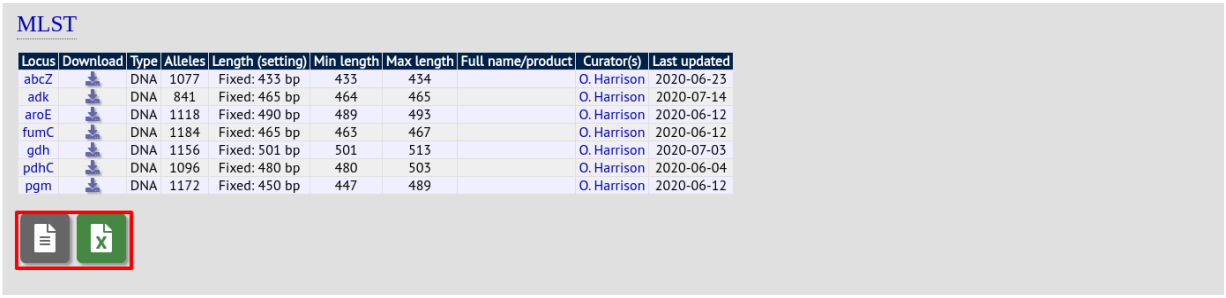
Loci can also be displayed by scheme with all schemes displayed.



Click the green download links for the required locus.

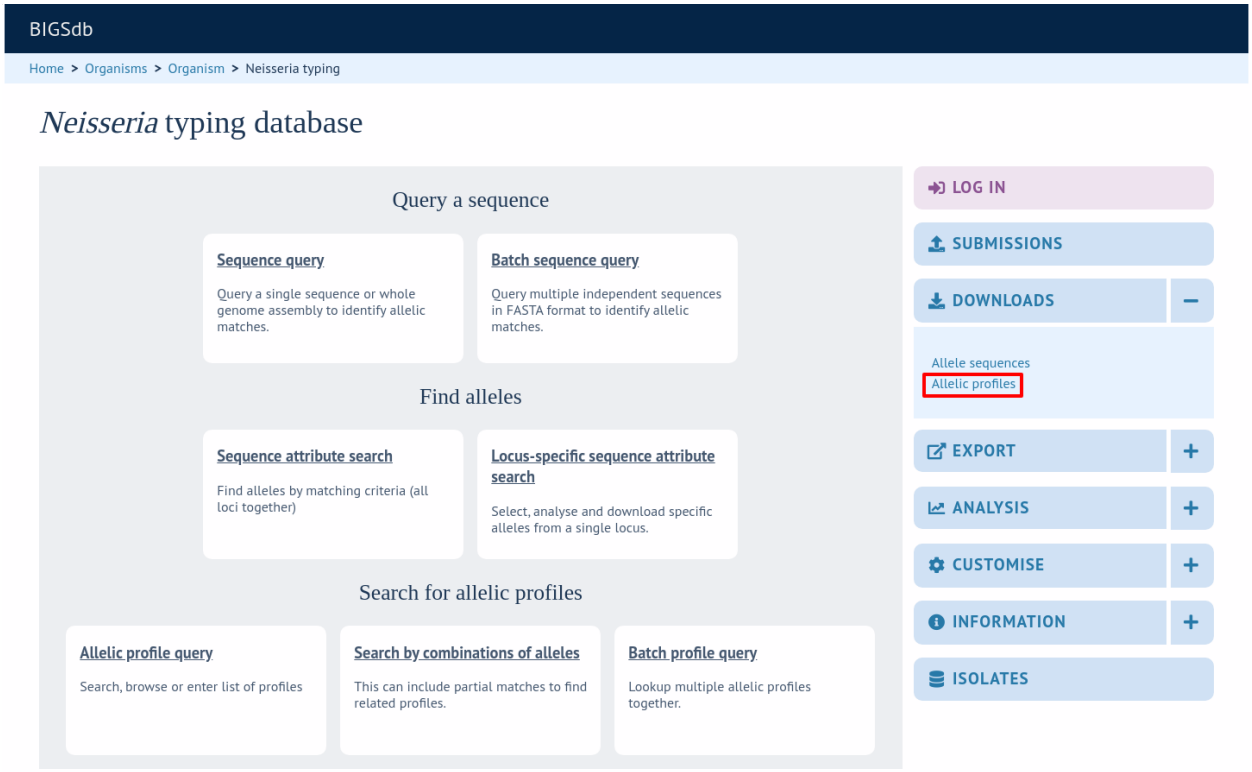
9.1.4 Download locus table

The locus table can be downloaded in tab-delimited text or Excel formats by clicking the links following table display.

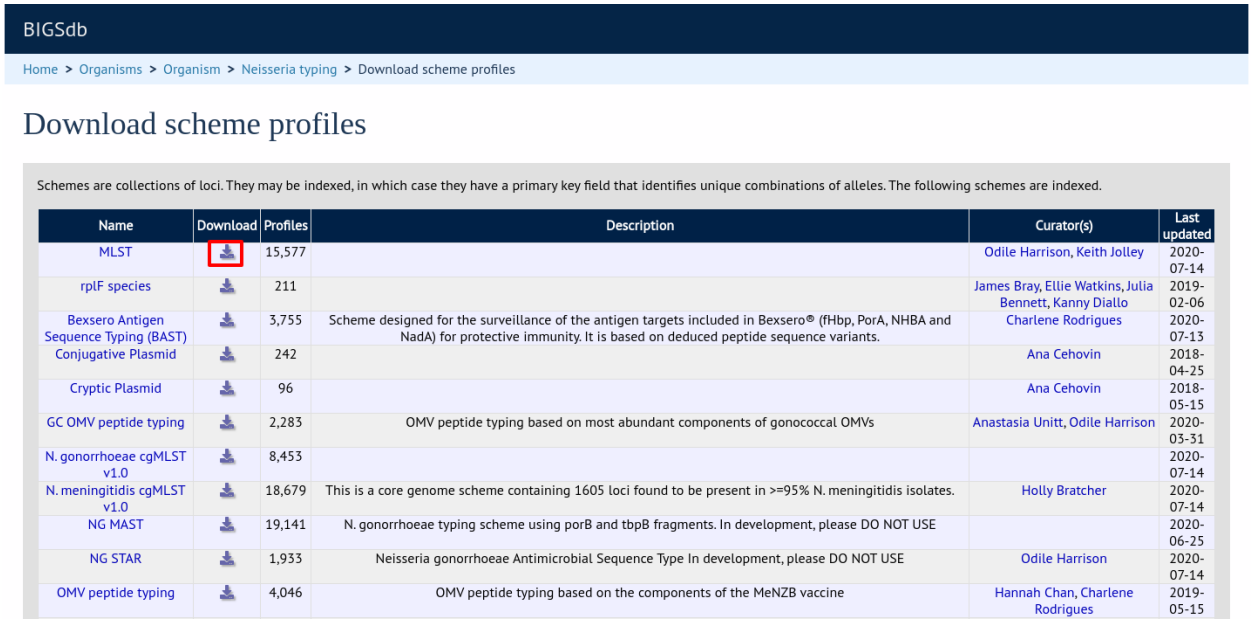


9.2 Scheme profile definitions

Scheme profiles, e.g. those for MLST, can be downloaded by clicking the appropriate link on the contents page.



If there is only one scheme available, the link will directly download the profiles. If multiple schemes are available, the link will take you to an intermediate page from where you can select the scheme to download.



Profiles will be downloaded in tab-delimited format, e.g.

ST	abcZ	adk	aroE	fumC	gdh	pdhC	pgm	clonal_complex
1	1	3	1	1	1	1	3	ST-1 complex/subgroup I/II
2	1	3	4	7	1	1	3	ST-1 complex/subgroup I/II
3	1	3	1	1	1	23	13	ST-1 complex/subgroup I/II
4	1	3	3	1	4	2	3	ST-4 complex/subgroup IV
5	1	1	2	1	3	2	3	ST-5 complex/subgroup III
6	1	1	2	1	3	2	11	ST-5 complex/subgroup III
7	1	1	2	1	3	2	19	ST-5 complex/subgroup III
8	2	3	7	2	8	5	2	ST-8 complex/Cluster A4
9	2	3	8	10	8	5	2	ST-8 complex/Cluster A4
10	2	3	4	2	8	15	2	ST-8 complex/Cluster A4
11	2	3	4	3	8	4	6	ST-11 complex/ET-37 complex
12	4	3	2	16	8	11	20	
13	4	10	15	7	8	11	1	ST-269 complex
14	4	1	15	7	8	11	1	ST-269 complex

CHAPTER

TEN

DATA RECORDS

Record pages for different types of data can be accessed following a query by clicking appropriate hyperlinks.

10.1 Isolate records

An Isolate record page displays everything known about an isolate.

BIGSdb

Home > Organisms > Neisseria spp. > Neisseria isolates > Isolate information

Help

Full information on isolate M10 240474 (id:18968)

Projects

This isolate is a member of the following project:

MRF Meningococcus Genome Library

The MRF Meningococcus Genome Library is a collaboration between Public Health England, the Scottish Haemophilus, Legionella, Meningococcus and Pneumococcus Reference (SHLMPR) Laboratory, The Wellcome Trust Sanger Institute and the University of Oxford, funded by the Meningitis Research Foundation.

Use of the MRF Genome Library data must be cited in any publication or presentation making use of it.

Provenance/primary metadata

id:	18968	disease:	invasive (unspecified/other)	update history:	223 updates show details
isolate:	M10 240474	species:	Neisseria meningitidis	date entered:	2012-02-15
strain designation:	B: P1.19-1,15-11: F3-9: ST-269 (cc269)	serogroup:	B	datestamp:	2020-02-28
country:	UK [England]	genogroup:	B		
continent:	Europe	capsule group:	B		
region:	South East	ENA run accession:	ERR086224 www.ebi.ac.uk		
year:	2010	sender:	Dorothea Hill, University of Oxford, UK		
epidemiological year:	07/2010-06/2011	curator:	Auto Tagger		

Secondary metadata

Vaccines

Bexsero reactivity:	cross-reactive	Trumenba reactivity:	cross-reactive	Trumenba notes:	fHbp_peptide: 15 is cross-reactive to vaccine variant - data derived from MEASURE assays (PMID:29535195), and SBA assays (PMID:22569484, PMID:22718089, PMID:23352429, PMID:26407272, PMID:26707218, PMID:26803328, PMID:26835974, PMID:27745812, PMID:27846061, PMID:28196734, PMID:28566335, PMID:29236639)
	notes		notes		
Bexsero notes:	fHbp_peptide: 15 is cross-reactive to vaccine variant - data derived from MATS assays (PMID:23414709, PMID:23588089, PMID:26686998, PMID:26950303, PMID:27355628, PMID:28366725, PMID:30135218, PMID:30592763, PMID:31770063)				

Publications (4)

- Brehony C, Rodrigues CM, Borrow R, Smith A, Cunney R, Moxon ER, Maiden MC (2016). Distribution of Bexsero® Antigen Sequence Types (BASTs) in invasive meningococcal disease isolates: Implications for immunisation. *Vaccine* **34**:4690-7 **2016 isolates**
- Hill DM, Lucidarme J, Gray SJ, Newbold LS, Ure R, Brehony C, Harrison OB, Bray JE, Jolley KA, Bratcher HB, Parkhill J, Tang CM, Borrow R, Maiden MC (2015). Genomic epidemiology of age-associated meningococcal lineages in national surveillance: an observational cohort study. *Lancet Infect Dis* **15**:1420-8 **899 isolates**
- Rodrigues CMC, Chan H, Vipond C, Jolley K, Harrison OB, Wheeler J, Whiting G, Feavers IM, Maiden MCJ (2018). Typing complex meningococcal vaccines to understand diversity and population structure of key vaccine antigens. *Wellcome Open Res* **3**:151 **3505 isolates**
- Rodrigues CMC, Lucidarme J, Borrow R, Smith A, Cameron JC, Moxon ER, Maiden MCJ (2018). Genomic Surveillance of 4CMenB Vaccine Antigenic Variants among Disease-Causing Neisseria meningitidis Isolates, United Kingdom, 2010-2016. *Emerg Infect Dis* **24**:673-682 **3066 isolates**

Sequence bin

contigs:	275	N50 contig number:	20	N95 contig number:	97
total length:	2,195,045 bp	N50 length (L50):	34,308	N95 length (L95):	3,513
max length:	109,859 bp	N90 contig number:	74	loci tagged:	2,179
mean length:	7,982 bp	N90 length (L90):	6,405		

Show sequence bin

Schemes and loci

- All loci
  - Capsule
  - Genetic Information Processing
  - Genomic islands
  - Lineage Schemes
  - Metabolism
  - Pilin
  - Typing
  - Other schemes
  - Loci not in schemes

Navigate and select schemes within tree to display allele designations

Tools

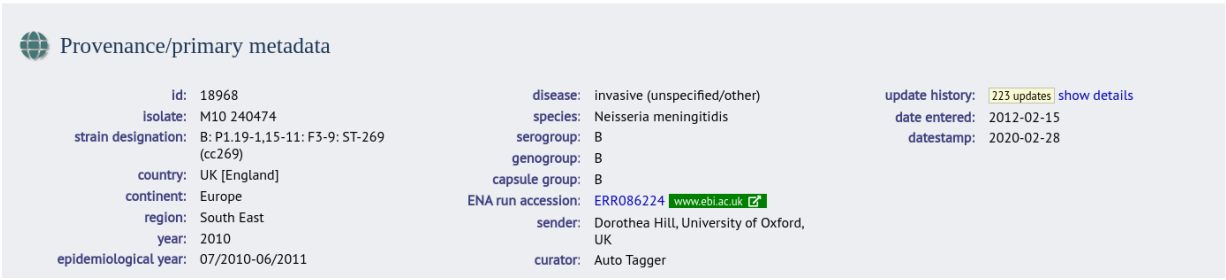
Analysis: rMLST species id PCR

Each record will have some or all of the following sections:

10.1.1 Projects

This displays a list of projects that the isolate is a member of. Only projects that have a full description and the 'isolate_display' flag in their settings will be displayed.

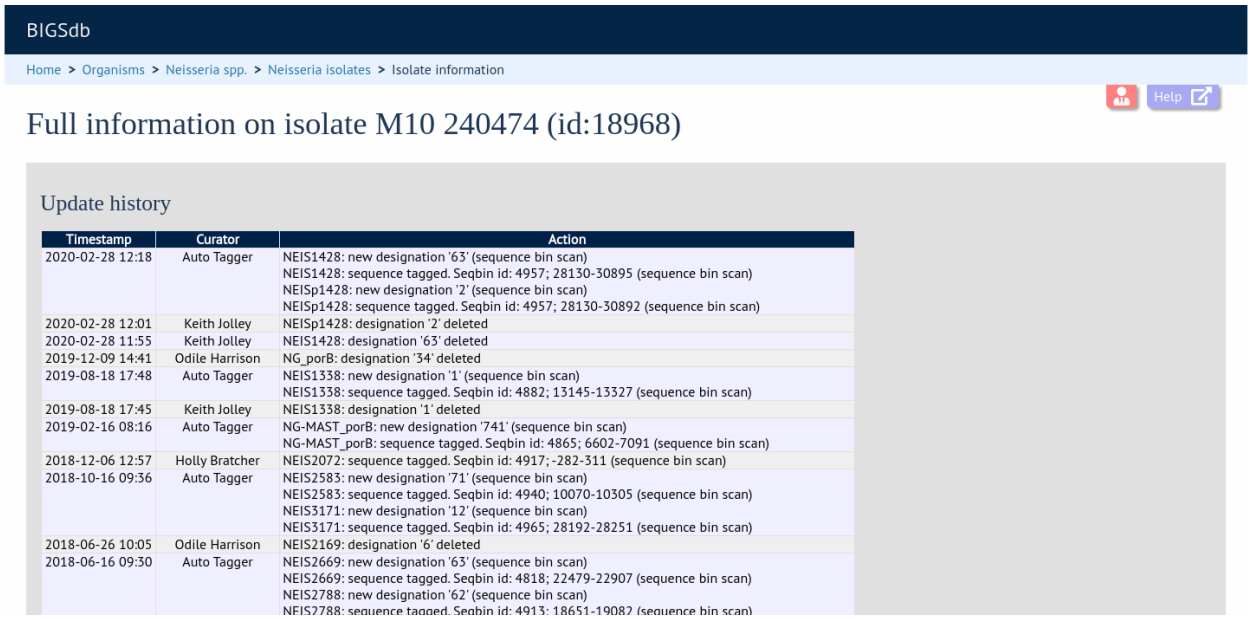
10.1.2 Provenance metadata



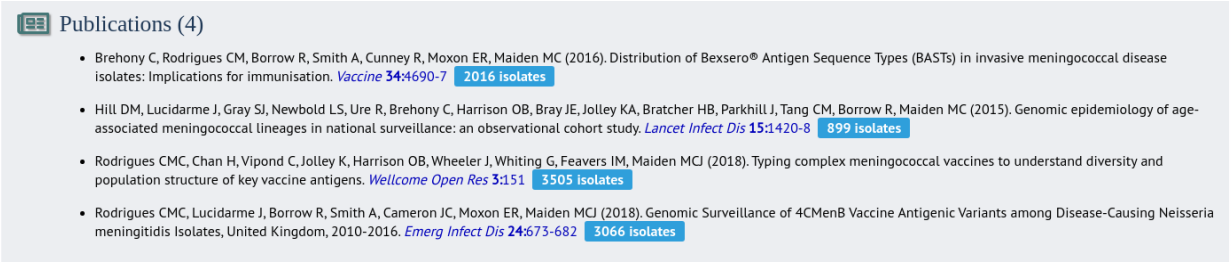
This section includes:

- provenance fields
- housekeeping data
  - who sent the isolate
  - who last curated
  - record creation times
  - last update times
  - links to update history

The update link displays page with exact times of who and when updated the record.



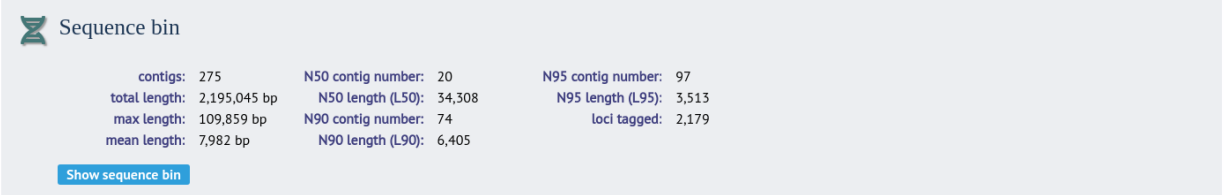
10.1.3 Publications



This section includes full citation for papers linked to the isolate record. Each citation has a button that will return a dataset of all isolates linked to the paper.

If there are five or more references they will be hidden by default to avoid cluttering the page too much. Click the 'Show/hide' button to display them in this case.

10.1.4 Sequence bin summary

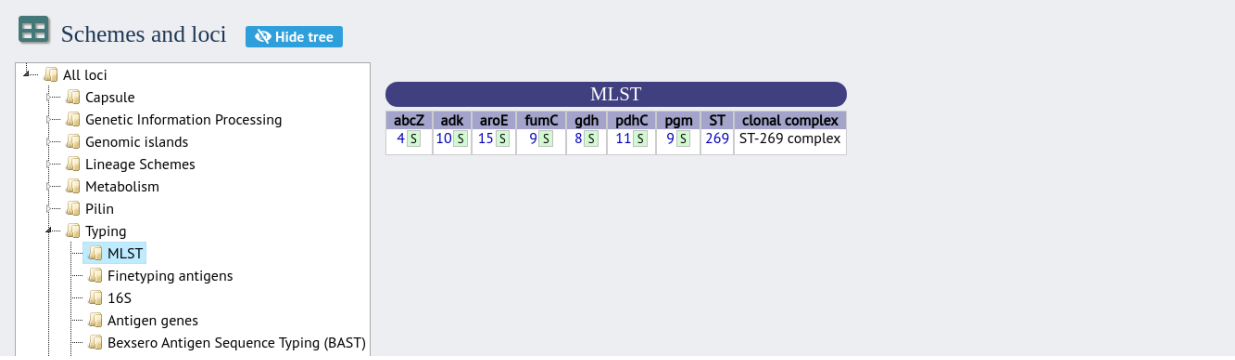


This section contains basic statistics describing the sequence bin. Clicking the 'Show sequence bin' button navigates to the *sequence bin record*.

10.1.5 Scheme and locus data

A hierarchical tree displays available schemes. Click within internal nodes to expand them.

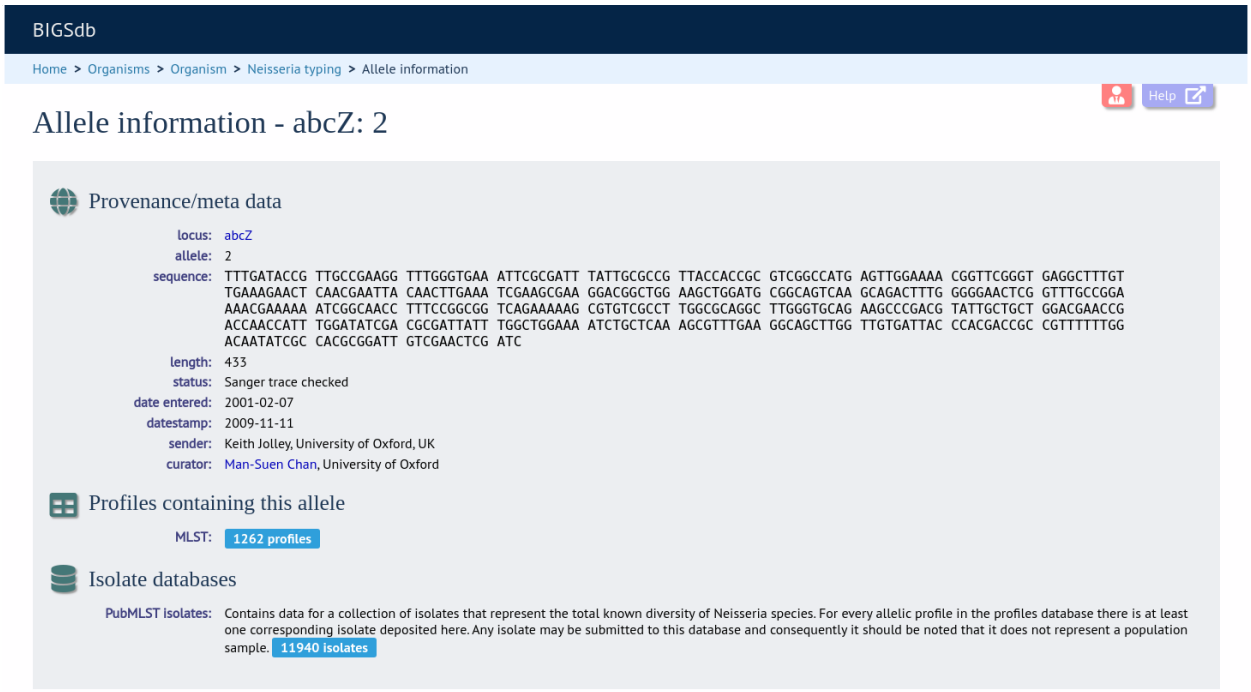
Clicking any terminal node will display data available for a scheme or group of schemes.



Click an allele number within the scheme profile, will display the appropriate *allele definition record*. Clicking the green 'S' link will display the appropriate *sequence tag record*.

10.2 Allele definition records

An allele definition record displays information about a defined allele in a sequence definition database.



If the allele is a member of a scheme profile, e.g. MLST, this will be listed. In this case, there will be a button to display all profiles of that scheme that contain the allele.

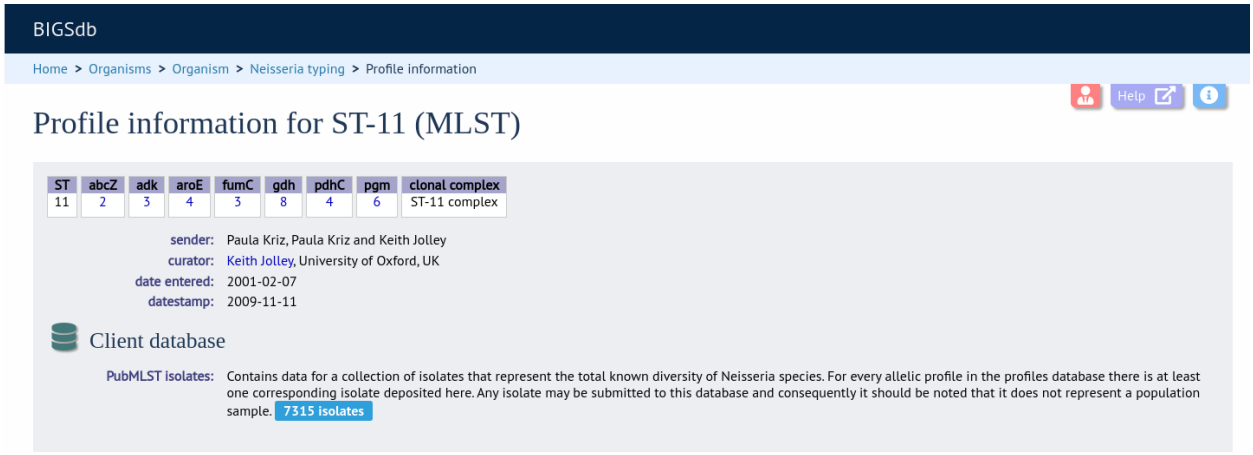
Similarly, if a *client database* has been setup for the database and the allele has been identified in an isolate, there will be a button to display all isolates that have that allele.

10.3 Sequence tag records

A sequence tag record displays information about the location within a contig of a region associated with a locus. The nucleotide sequence will be displayed along with upstream and downstream flanking sequence. The length of these flanking sequences can be modified within the *general options*.

If the tag is for a DNA locus and it is marked as a coding sequence, the three-frame translation will also be displayed.

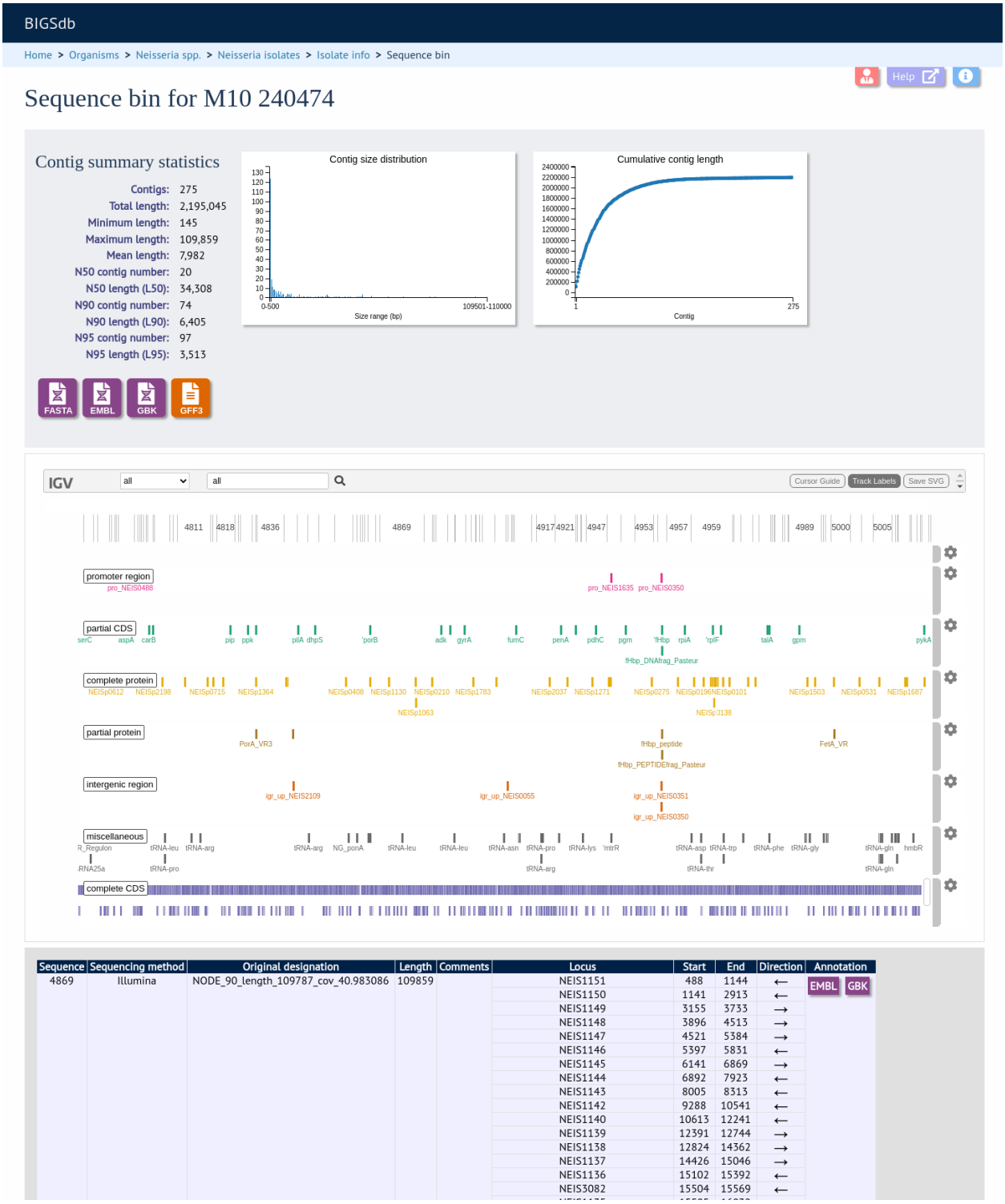
10.4 Profile records



A profile record displays information about a scheme, e.g. MLST, profile. Each allele number within the profile will be hyperlinked. Clicking these will take you to the appropriate *allele definition record*.

If a *client database* has been setup for the database and an isolate has the profile, there will be a button to display all isolates that have the profile.

10.5 Sequence bin records



A sequence bin record contains information about the contigs associated with an isolate record. This includes:

- Number of contigs

- Total length
- Minimum length
- Maximum length
- N50, N90 and N95 values
- Size distribution charts

Charts show the distribution of contig sizes and the cumulative contig length against contig number giving a breakdown indication of contig size.

The record includes an embedded genome browser showing the positions of any loci that have been tagged.

There are also links to download the contigs in FASTA, Genbank or EMBL format, along with annotation in GFF3 format.

Finally there is a table that shows the loci that are tagged on each contig. Individual contigs can also be downloaded in EMBL or Genbank format.

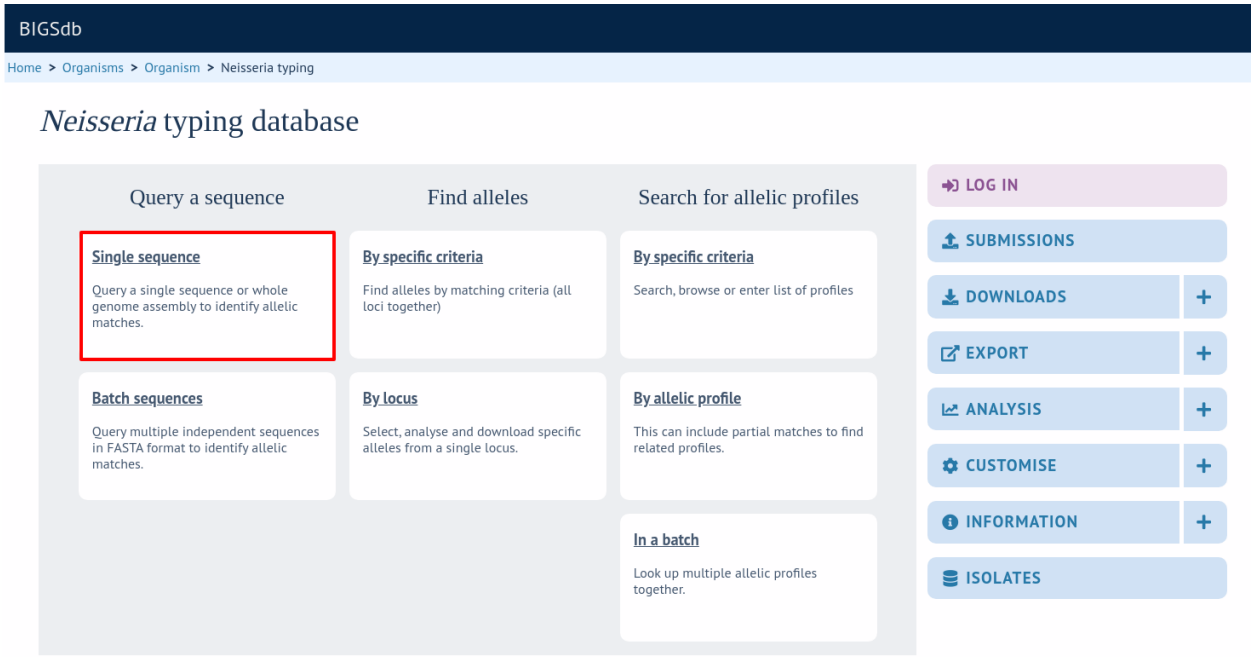
CHAPTER

ELEVEN

QUERYING DATA

11.1 Querying sequences to determine allele identity

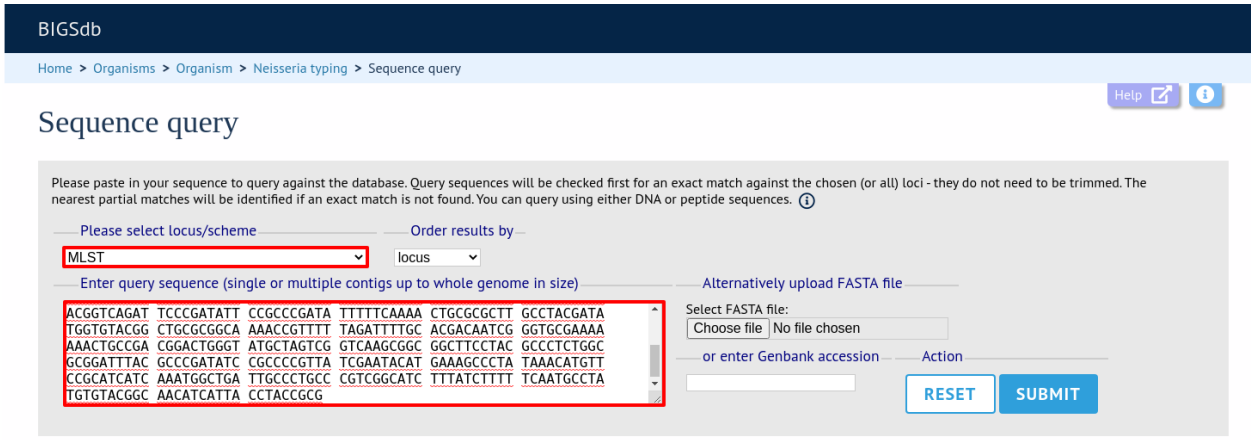
Sequence queries are performed in the sequence definition database. Click 'Single sequence' query from the contents page.



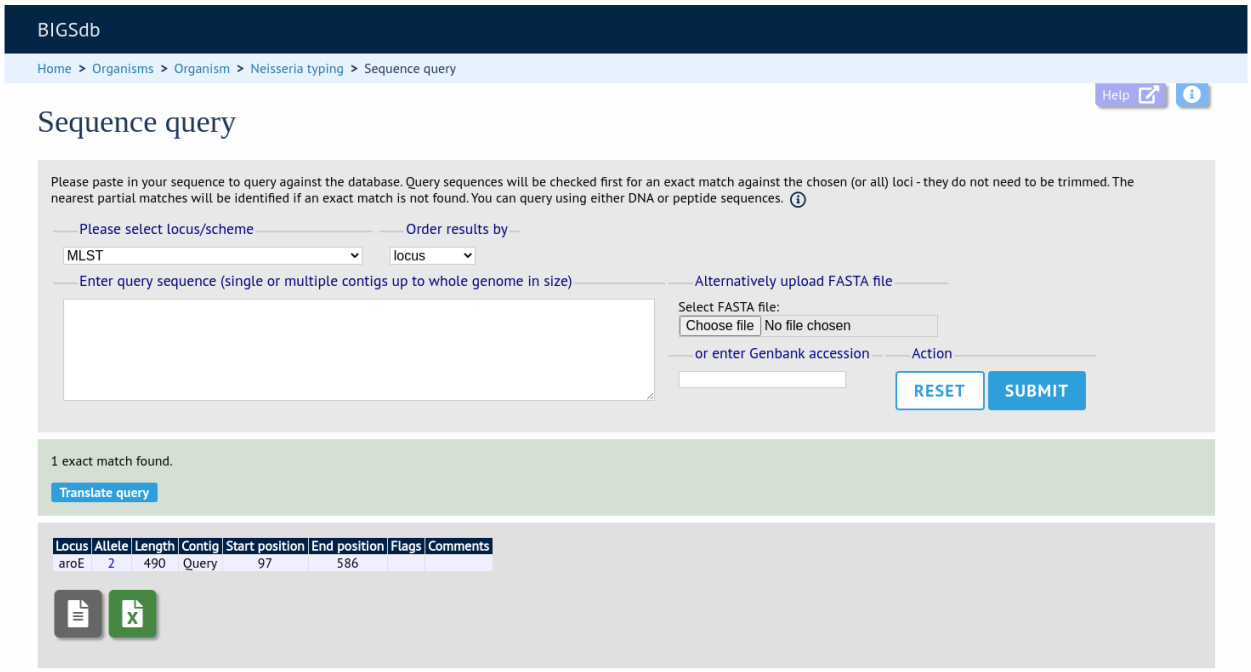
Paste your sequence in to the box - there is no need to trim. Often, you can leave the locus setting on 'All loci' - the software should identify the correct locus based on your sequence. Sometimes, especially in databases with a large number of defined loci, it may be quicker, however, to select the specific locus or scheme (e.g. MLST) that a locus belongs to.

Note: If the locus you are querying is a shorter version of another, e.g. an MLST fragment of a gene where the full length gene is also defined, you will need to select the specific locus or the scheme from the dropdown box. Leaving the selection on 'All loci' will return a match to the longer sequence in preference to the shorter one.

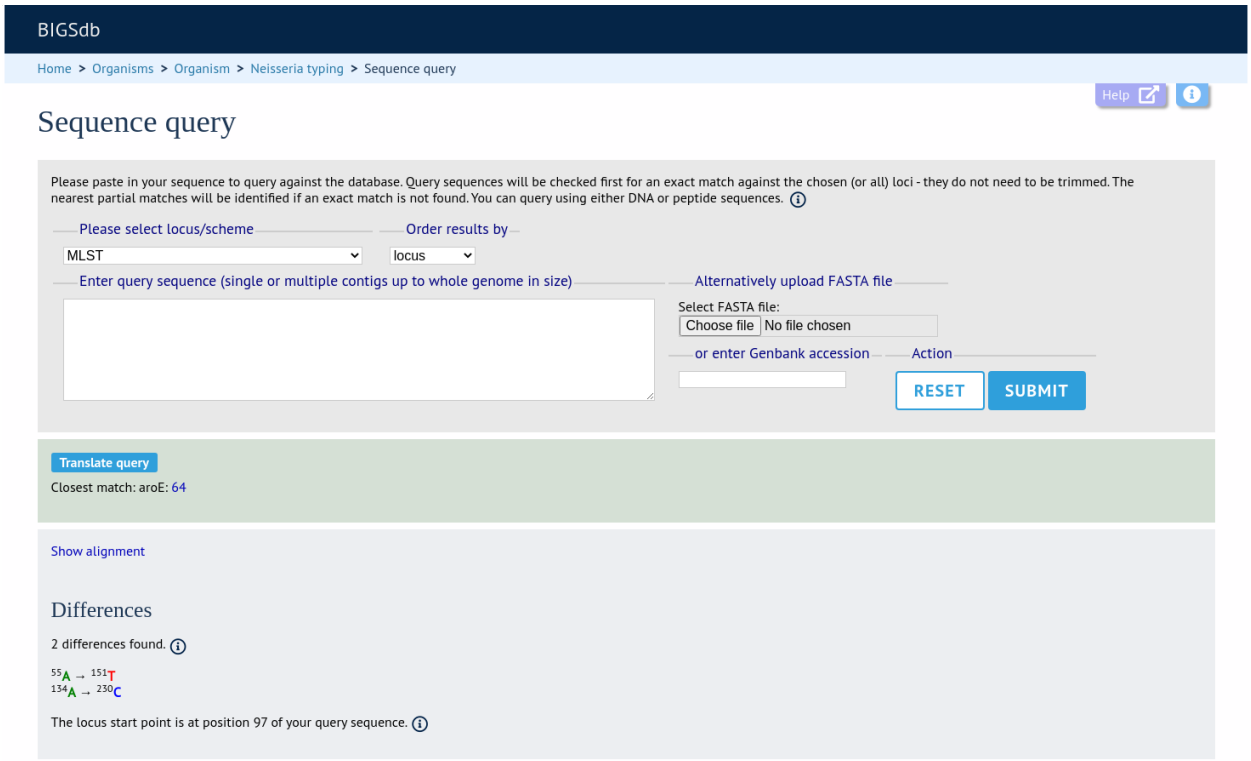
Click 'Submit'.



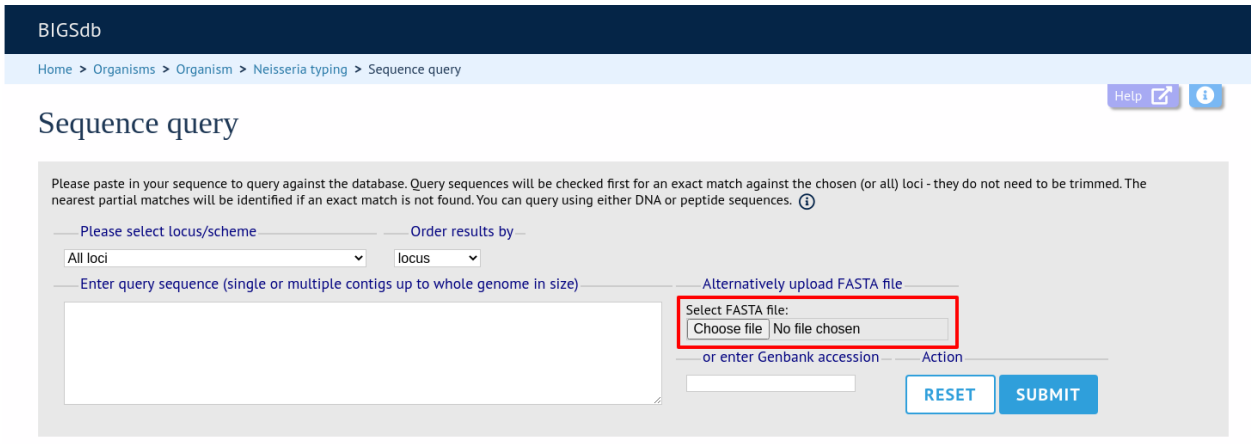
If an exact match is found, this will be indicated along with the start position of the locus within your sequence.



If only a partial match is found, the most similar allele is identified along with any nucleotide differences. The varying nucleotide positions are numbered both relative to the pasted in sequence and to the reference sequence. The start position of the locus within your sequence is also indicated.

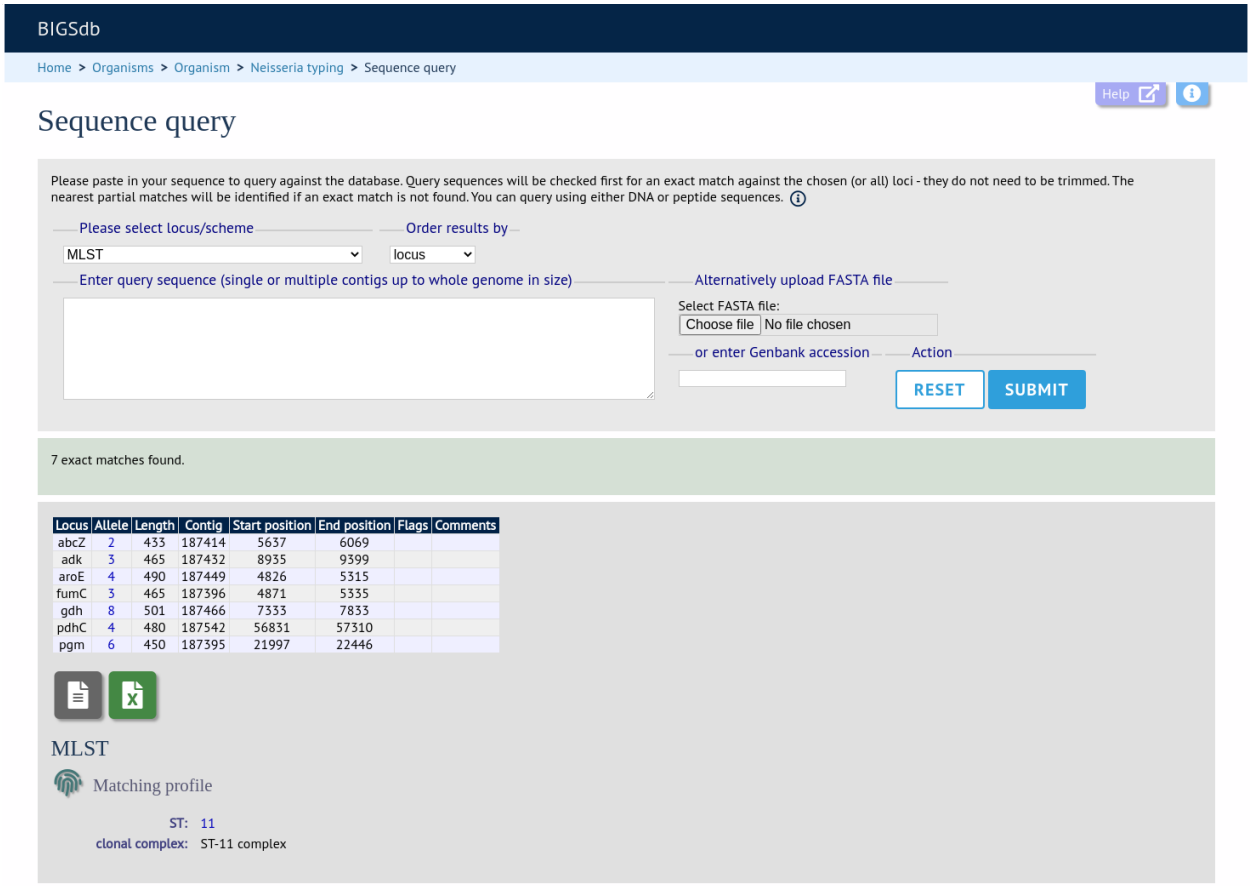


As an alternative to pasting a sequence in to the box, you can also choose to upload sequences in FASTA format by clicking the file browse button.



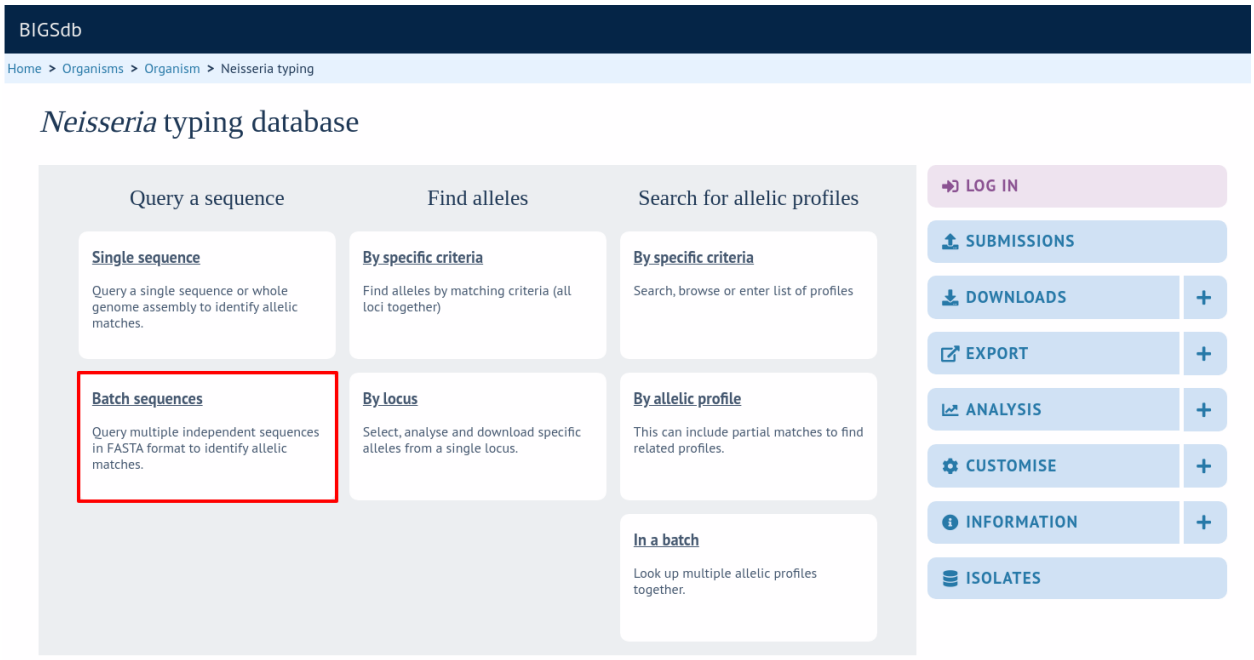
11.1.1 Querying whole genome data

The sequence query is not limited to single genes. You can also paste or upload whole genomes - these can be in multiple contigs. If you select a specific scheme from the dropdown box, all loci belonging to that scheme will be checked (although only exact matches are reported for a locus if one of the other loci has an exact match). If all loci are matched, scheme fields will also be returned if these are defined. This, for example, enables you to identify the MLST sequence type of a genome in one step.

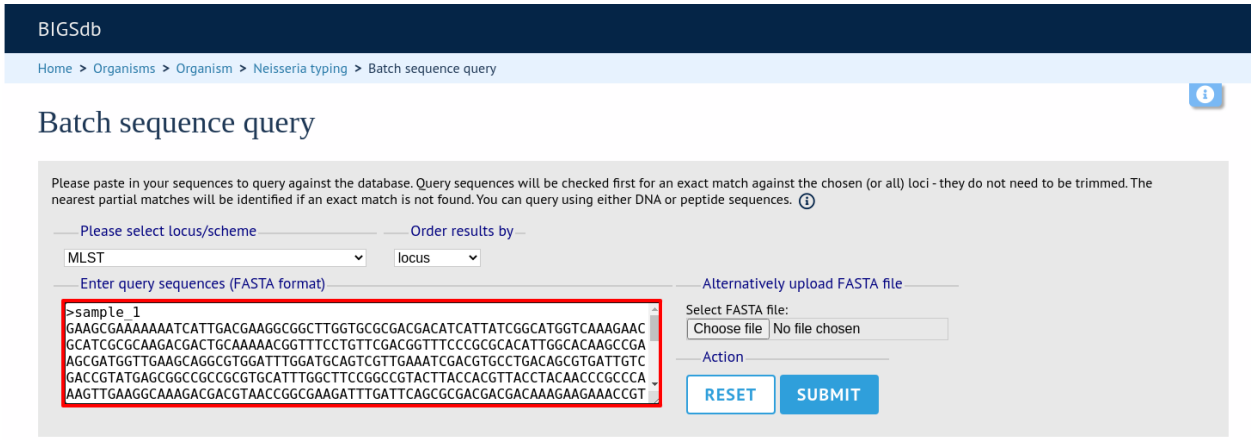


11.2 Querying multiple sequences to identify allele identities

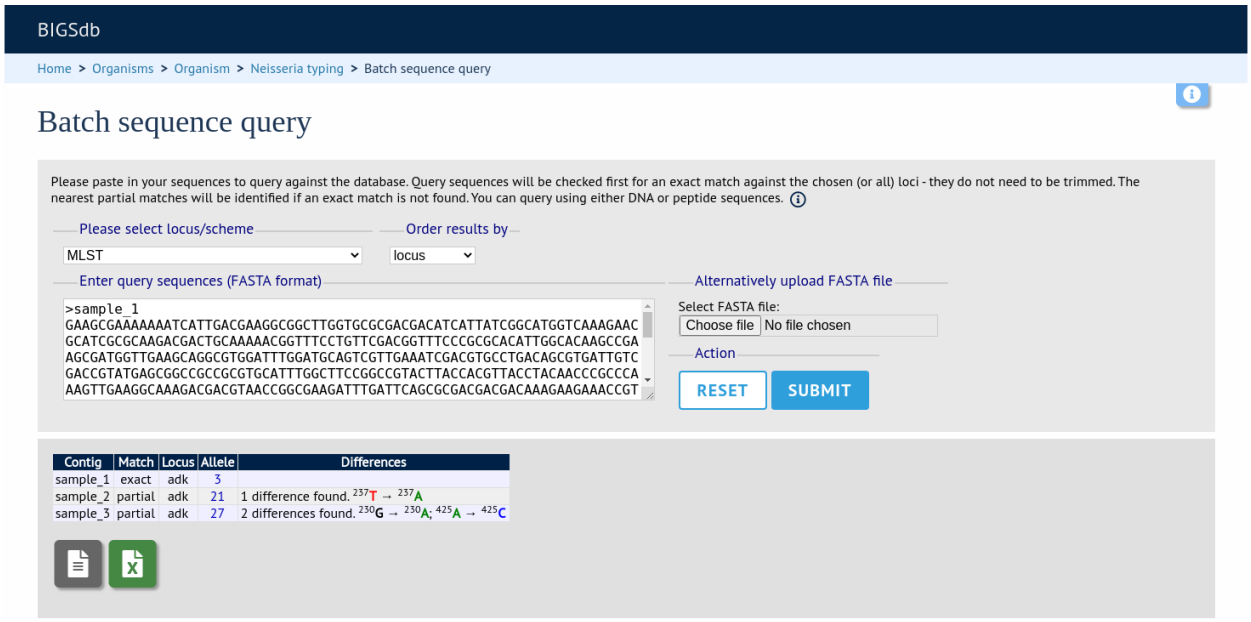
You can also query mutiple sequences together. These should be in FASTA format. Click 'Batch sequences' from the contents page.



Paste your sequences (FASTA format) in to the box. Select a specific locus, scheme or 'All loci'.



The best match will be displayed for each sequence in your file. If this isn't an exact match, the differences will be listed.

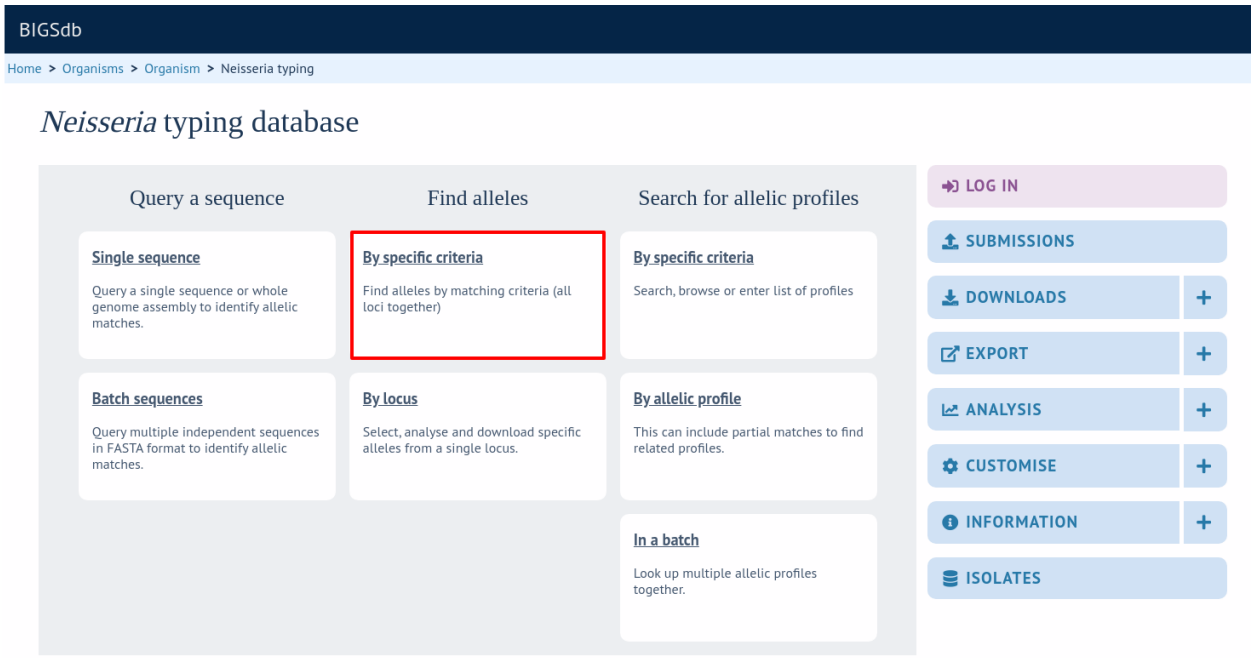


11.3 Searching for specific allele definitions

There are two query pages available that allow searching for specific allele definitions. The first allows querying of all loci together by criteria that are common to all. The second is a locus-specific attribute query that can search on any extended attributes that may be defined for a locus. This locus- specific query also allows you to paste in lists of alleles for download or analysis.

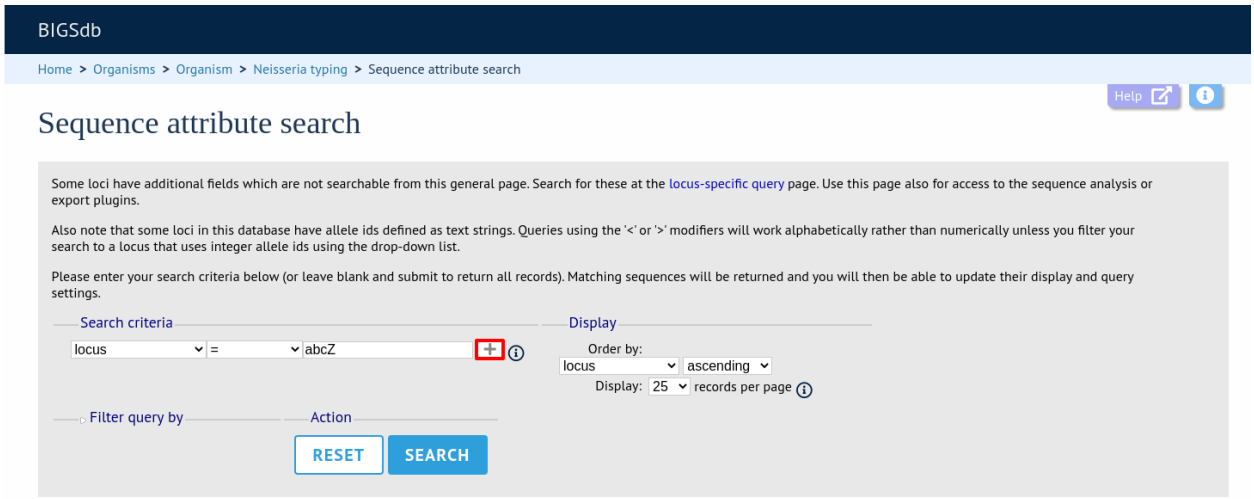
11.3.1 General (all loci) sequence attribute search

To retrieve specific allele designations, click 'By specific criteria' in the 'Find alleles' section on a sequence definition database contents page.

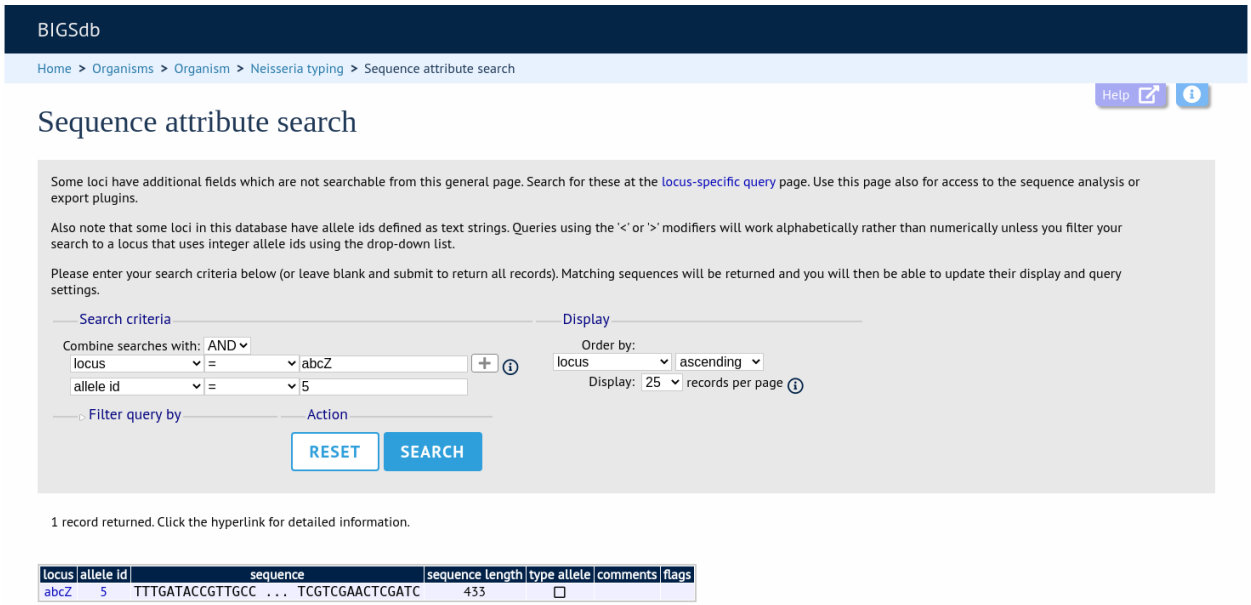


Enter your query using the dropdown search box - additional terms can be added by clicking the '+' button.

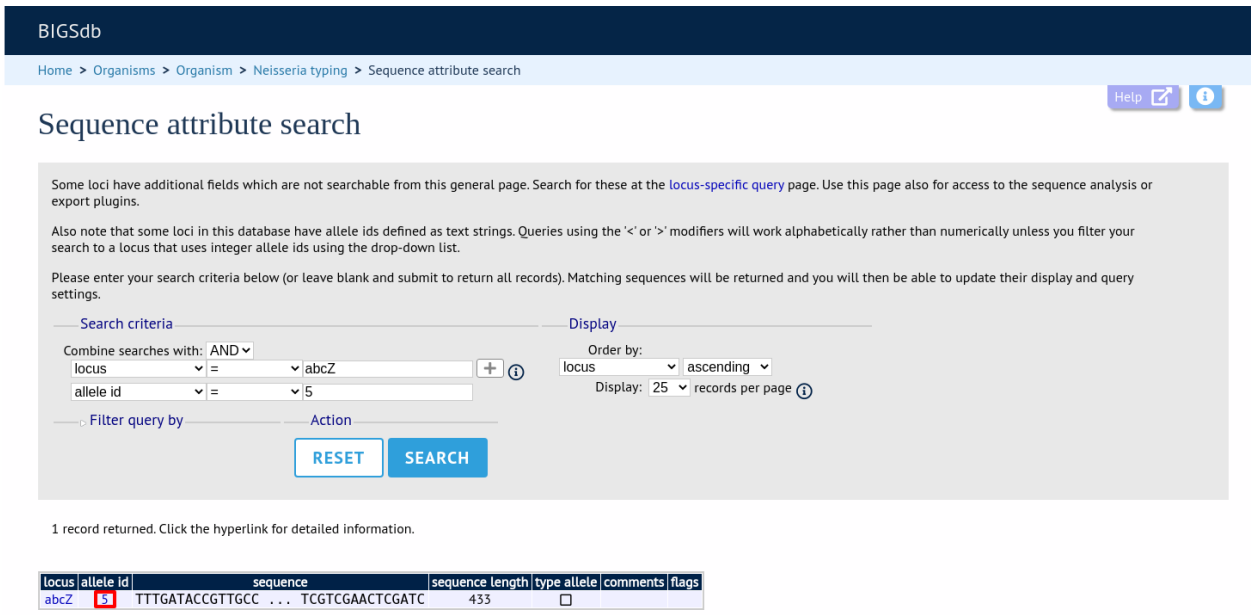
Designations can be queried using *standard operators*.

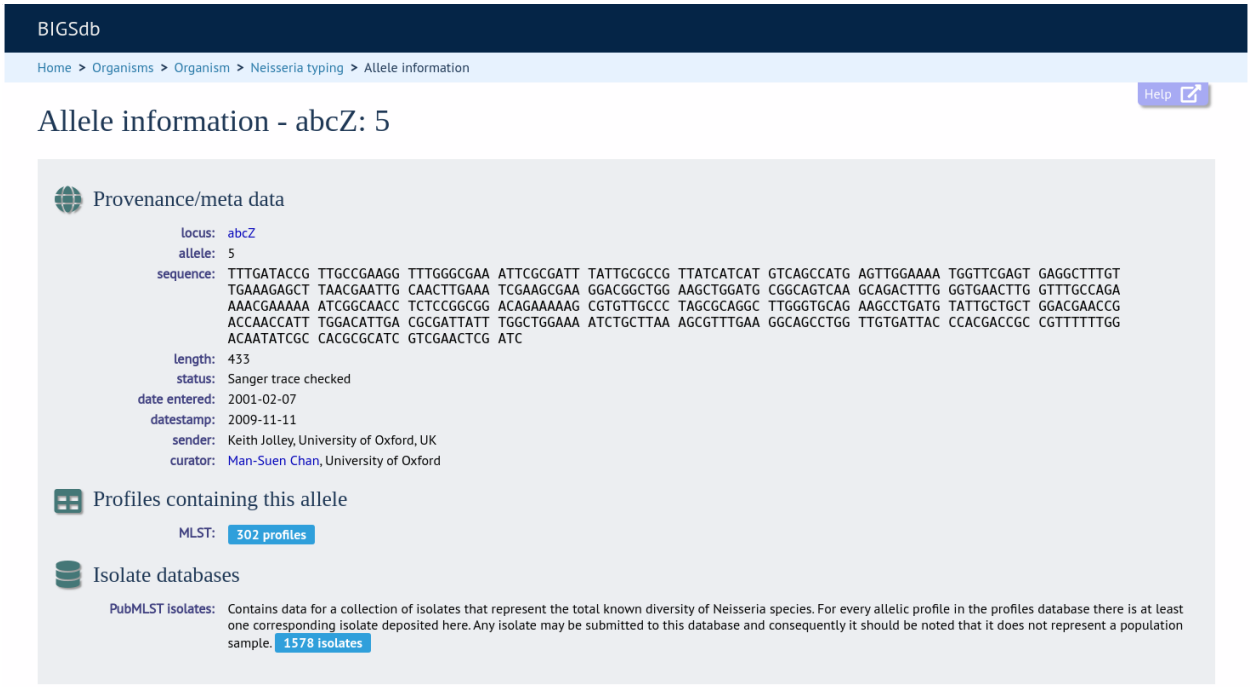


Click 'Search'.

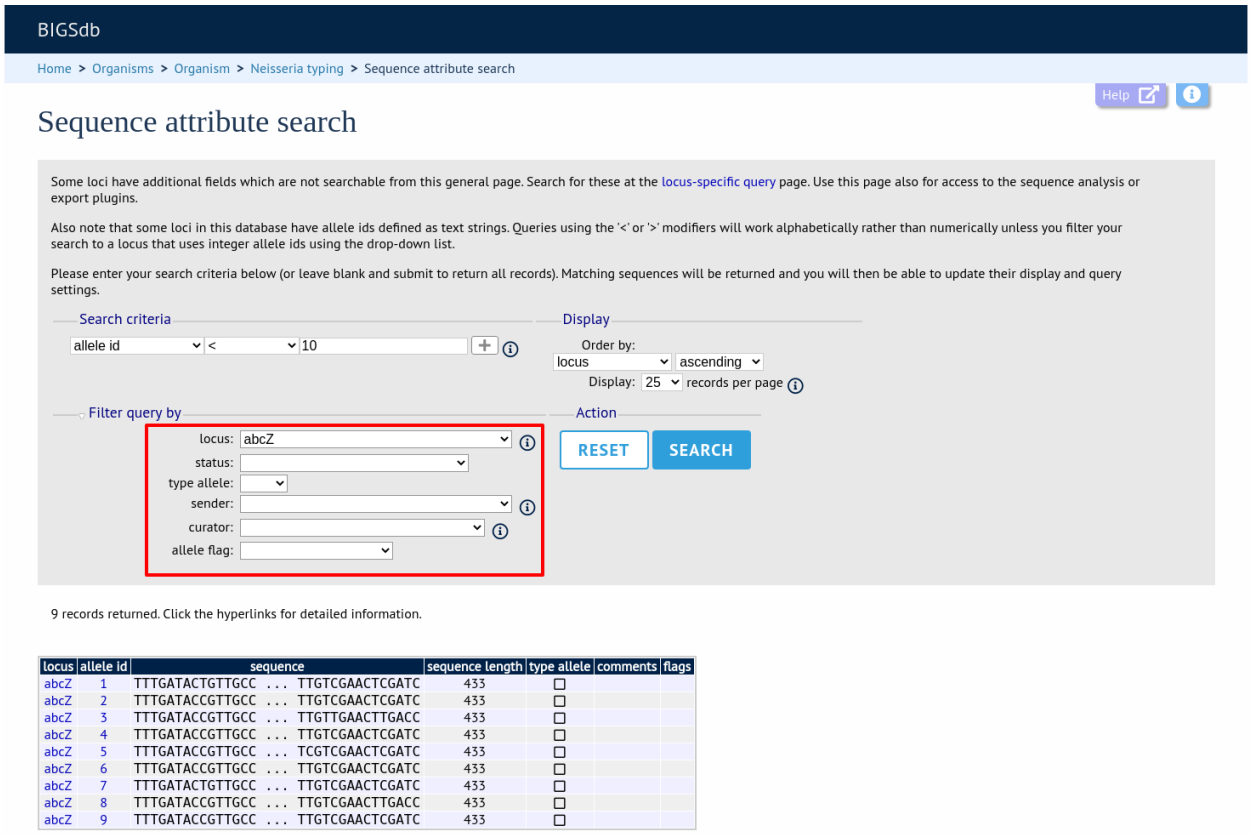


Click the hyperlinked results to display allele records.



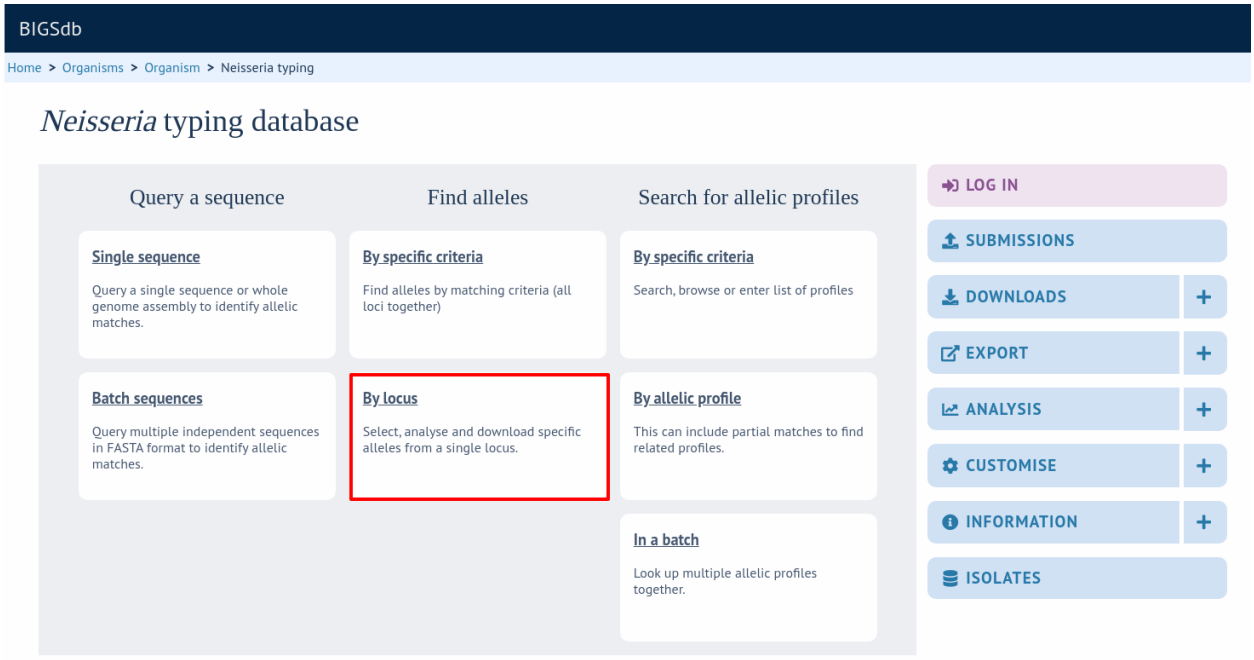


Various search criteria can also be selected by combining with filters. Click the filter heading to display these.

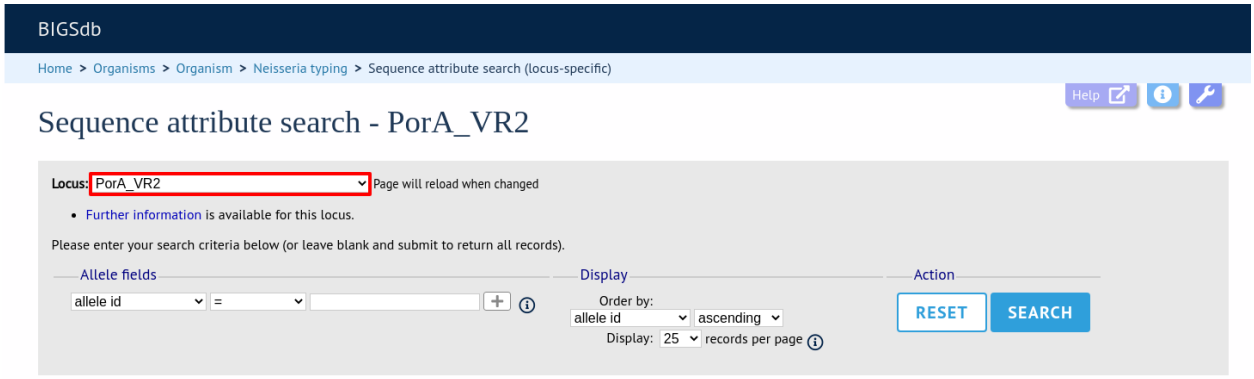


11.3.2 Locus-specific sequence attribute search

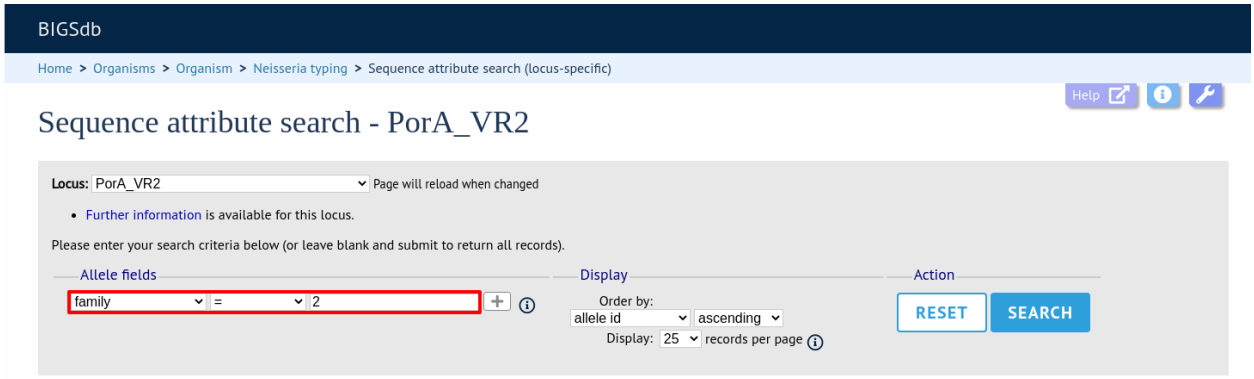
Some loci have *extended attribute fields*. To query these, click 'By locus' in the 'Find alleles' section on a sequence definition database contents page.



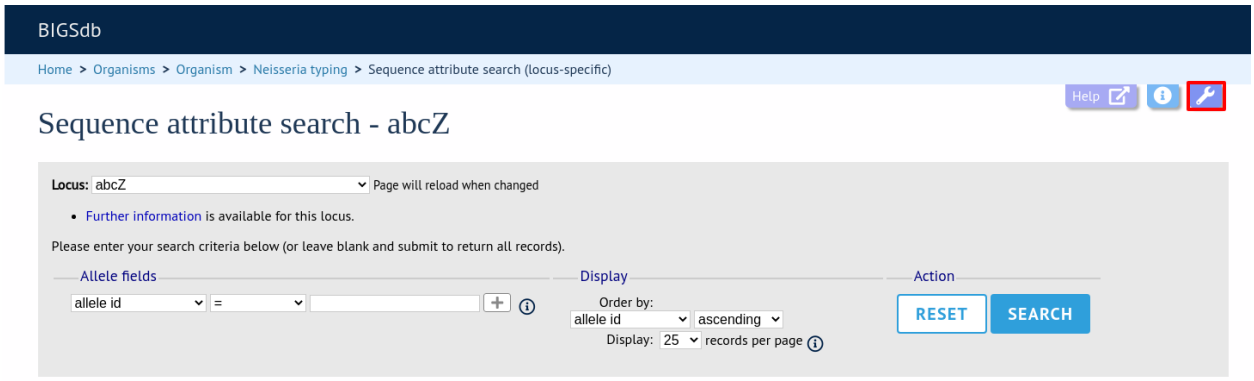
Pick the required locus from the dropdown box.



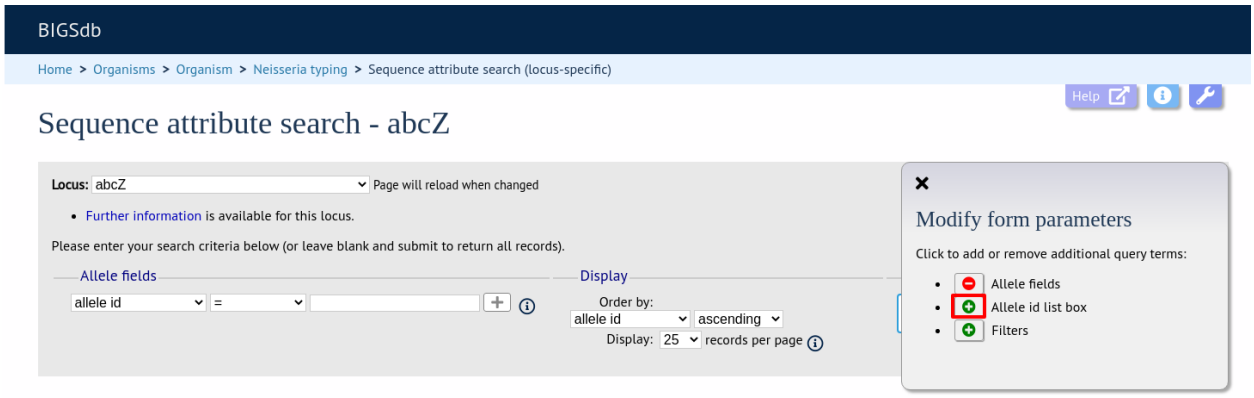
The fields specific for that locus will be added to the dropdown query boxes.



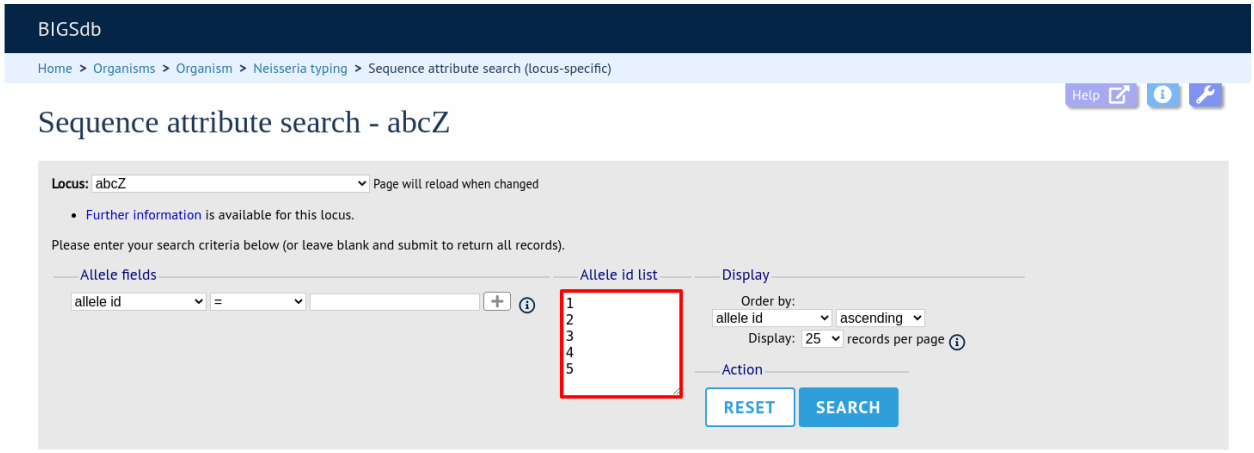
The query form can be modified by clicking the 'Modify form options' tab:



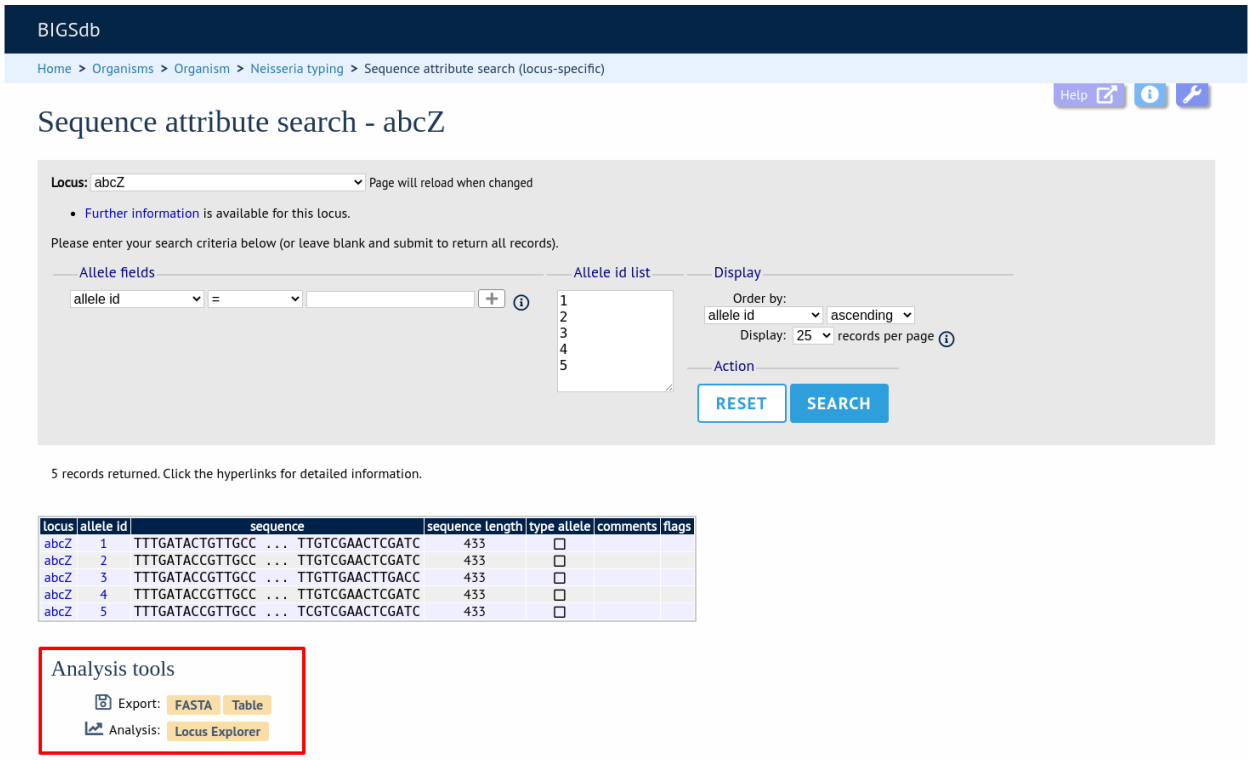
A list box can be added by clicking the 'Show' button for 'Allele id list box'.



Close the form modification tab and you can now enter a list of allele ids for retrieval.

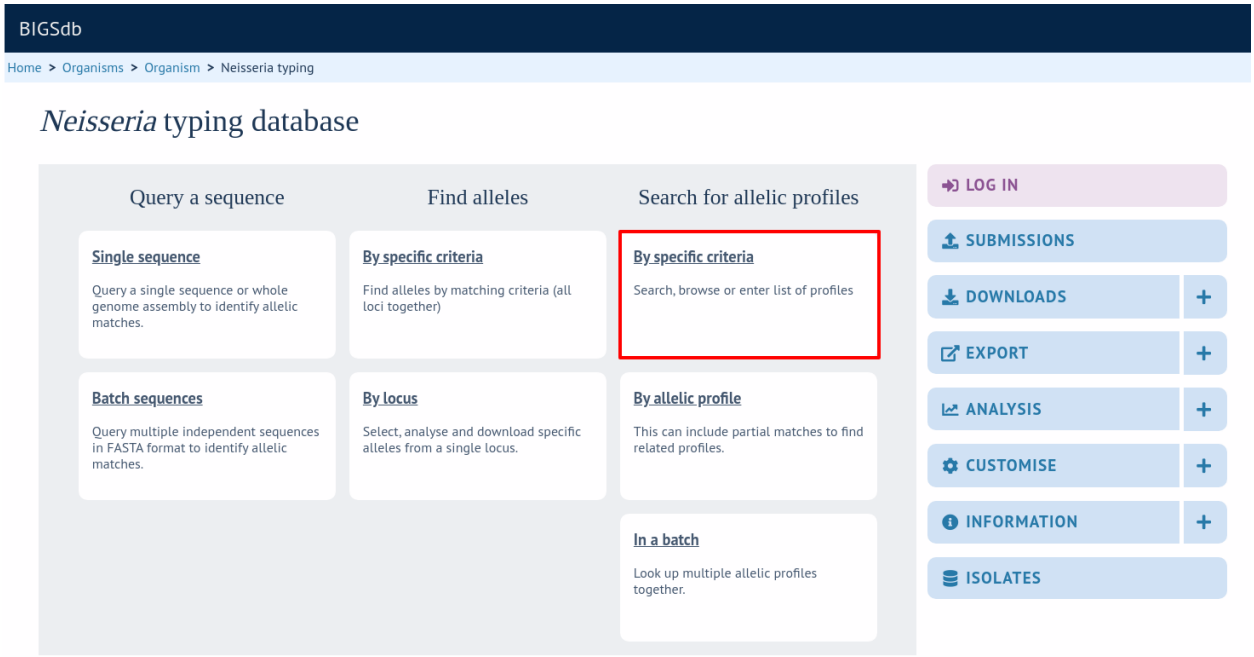


Various analysis and export options will be available for use on the retrieved sequences. These include FASTA output and *Locus Explorer* analysis.

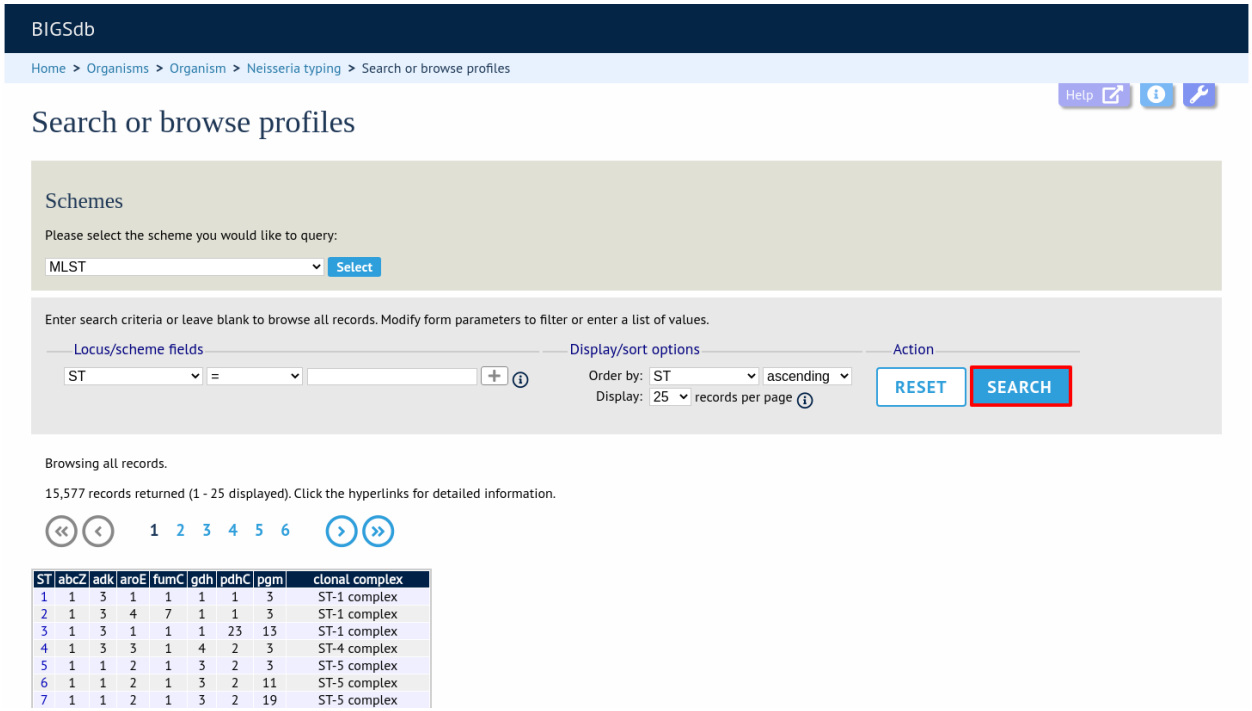


11.4 Browsing scheme profile definitions

If a sequence definition database has schemes defined that include a primary key field, i.e. collections of loci that together create profiles, e.g. for MLST, these can be browsed by clicking the link 'By specific criteria' in the 'Search for allelic profiles' section.

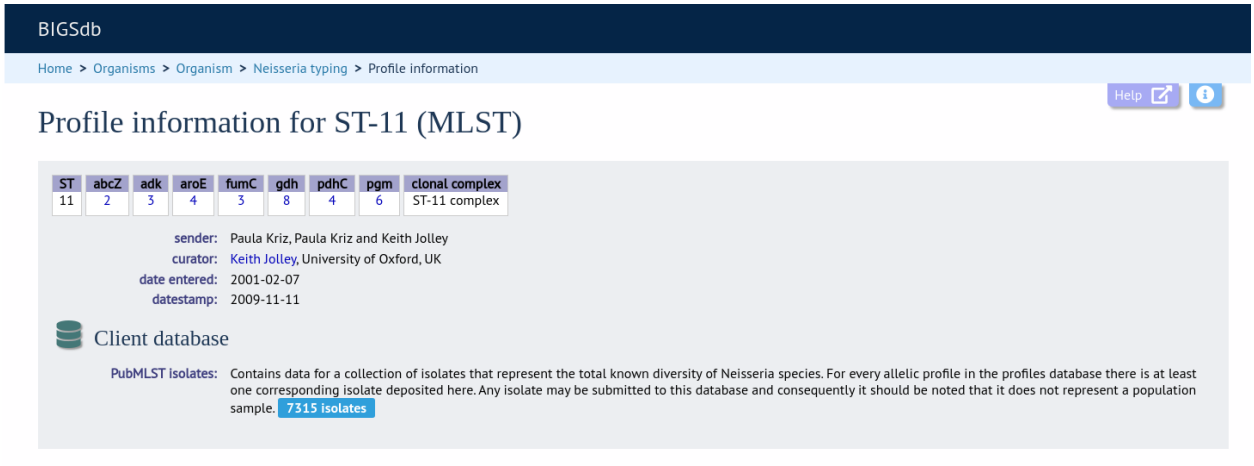


Leave query form fields blank (the display of these may vary depending on modification options set by the user). Choose the field to order the results by, the number of results per page to display, and click 'Search'.



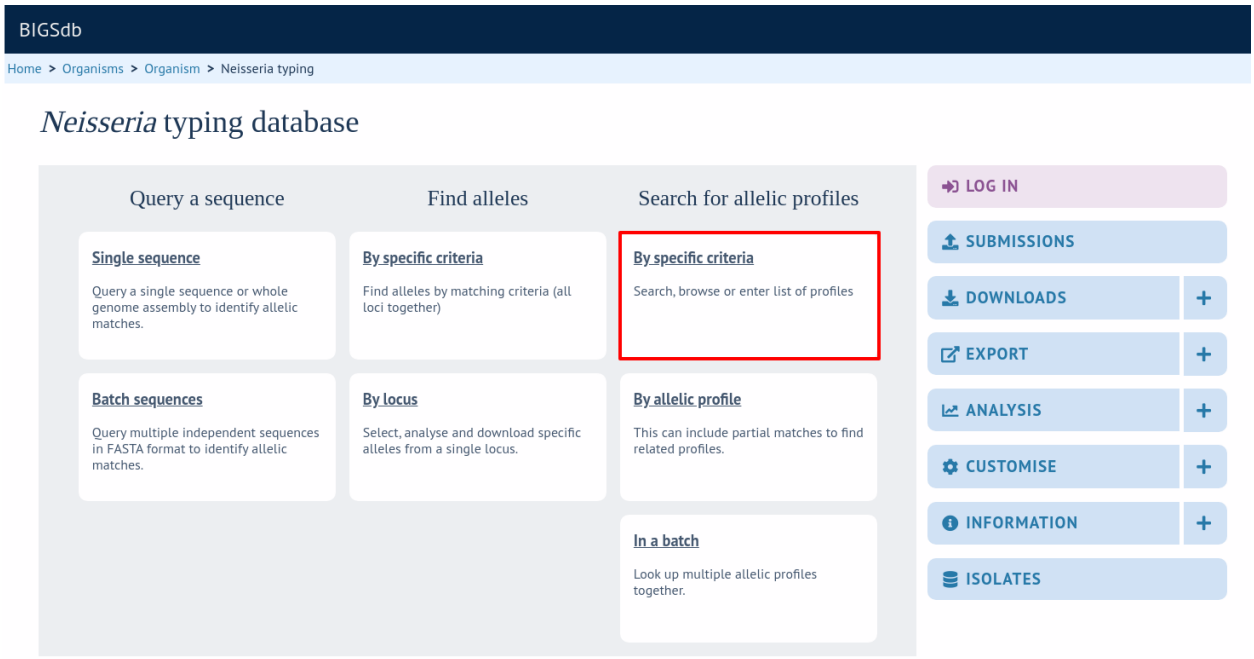
ST	abcZ	adk	aroE	fumC	gdh	pdhC	pgm	clonal complex
1	1	3	1	1	1	1	3	ST-1 complex
2	1	3	4	7	1	1	3	ST-1 complex
3	1	3	1	1	1	23	13	ST-1 complex
4	1	3	3	1	4	2	3	ST-4 complex
5	1	1	2	1	3	2	3	ST-5 complex
6	1	1	2	1	3	2	11	ST-5 complex
7	1	1	2	1	3	2	19	ST-5 complex

Clicking the hyperlink for any profile will display full information about the profile.

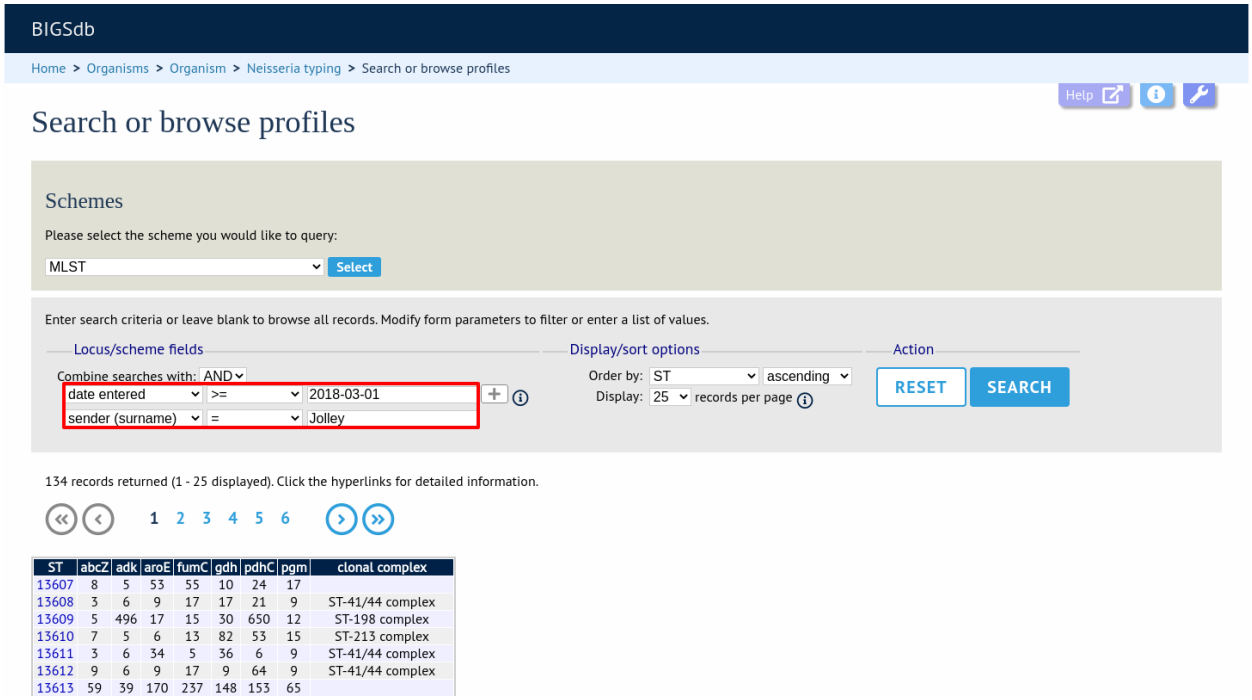


11.5 Querying scheme profile definitions

Click the link to 'By specific criteria' link in the 'Search for allelic profiles' section.

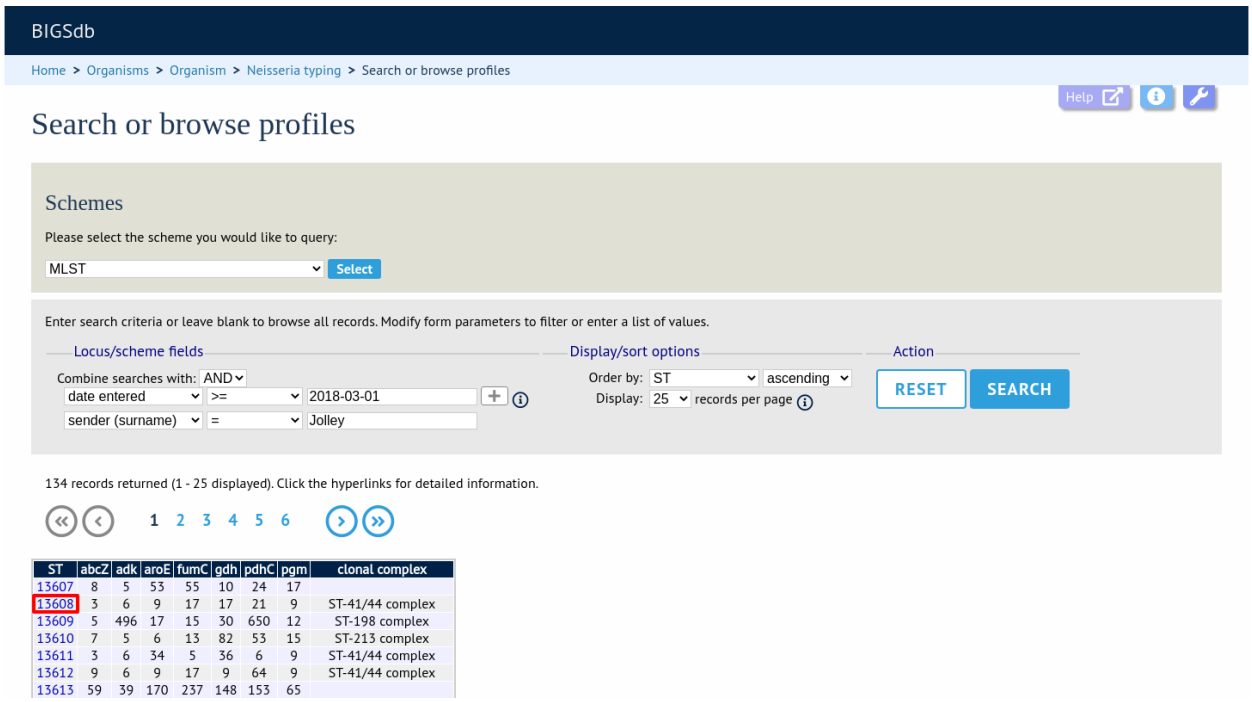


Enter the search criteria you wish to search on. You can add search criteria by clicking the '+' button in the 'Locus/scheme fields' section. These can be combined using 'AND' or 'OR'.

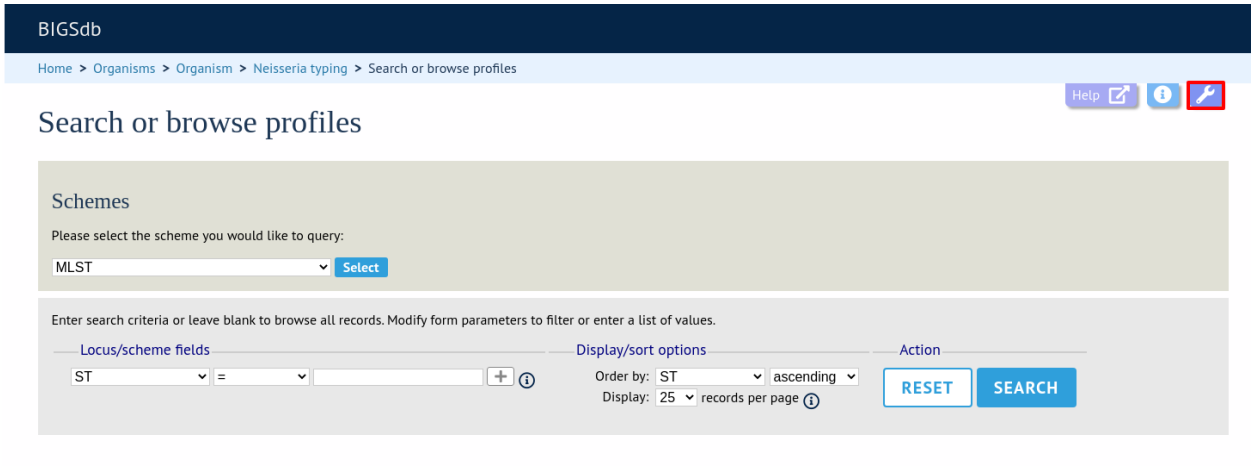


Each field can be queried using *standard operators*.

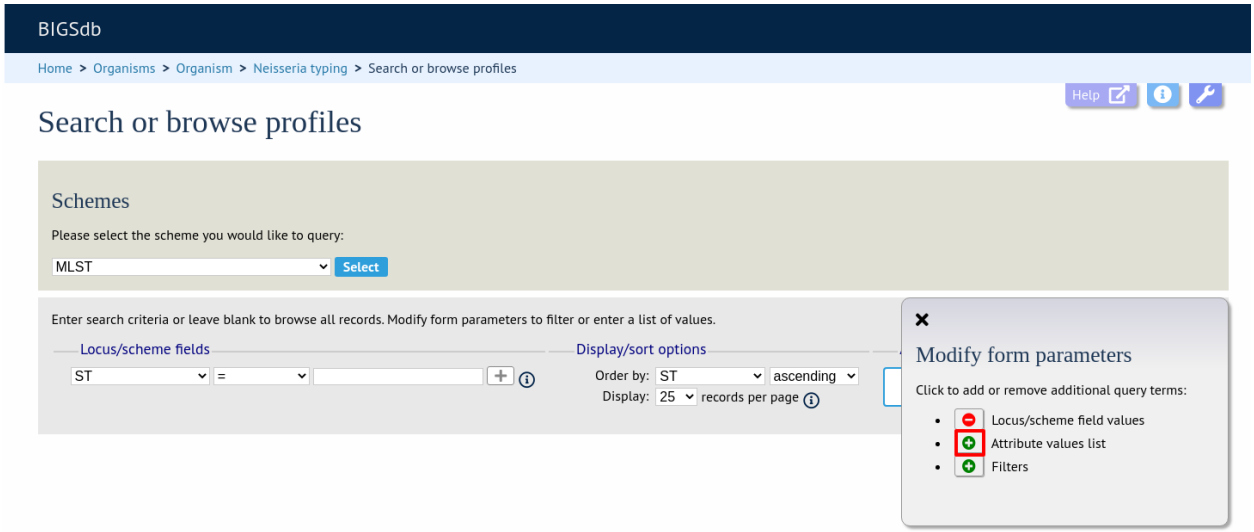
Clicking the hyperlink for any profile will display full information about the profile.



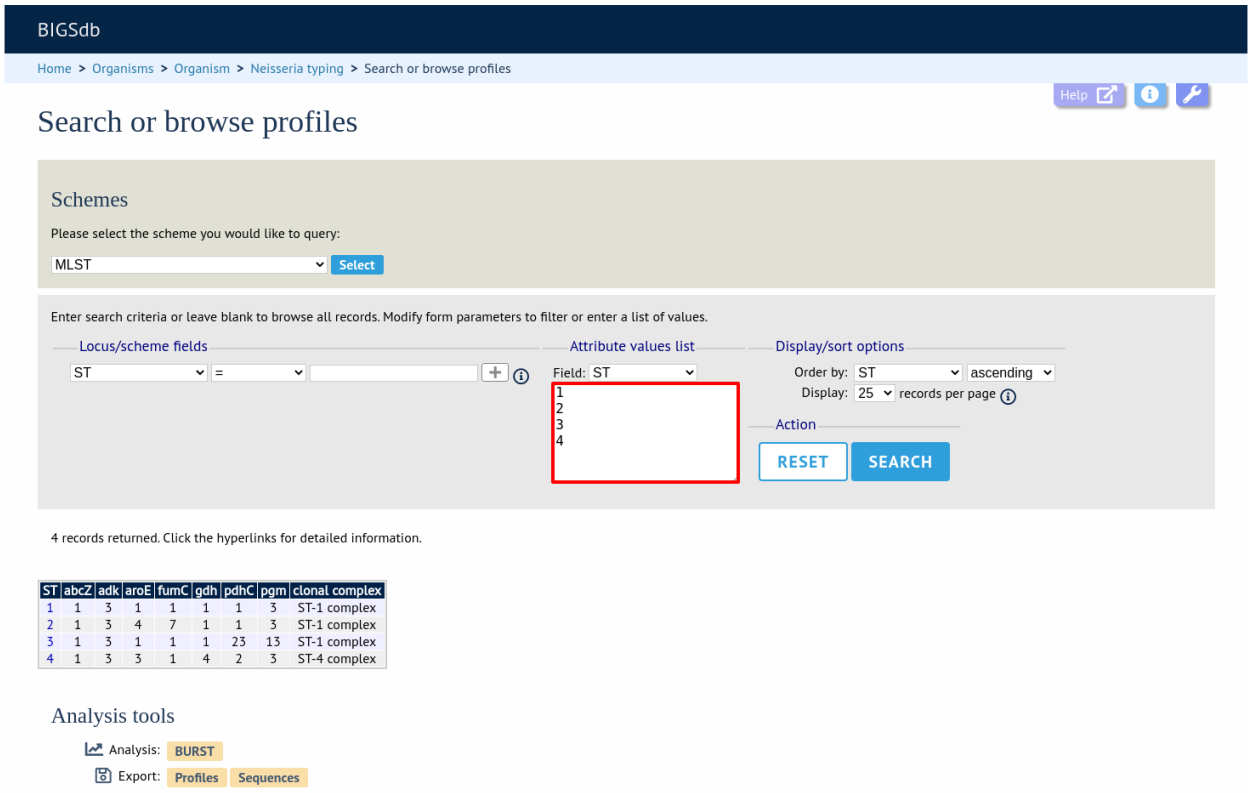
Other query options are available by clicking the 'Modify form options' tab.



For example, you can enter a list of attributes to query on by clicking the 'Show' button next to 'Attribute values list'.

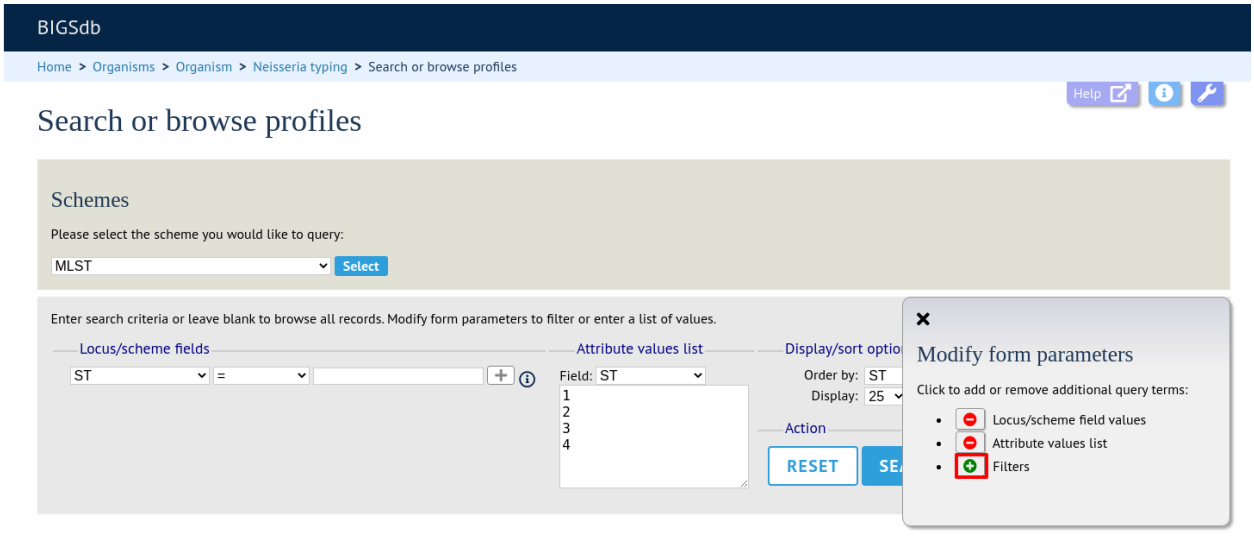


A list box will appear within the page. Hide the form modification tab by clicking the 'X' in the corner or the purple tab again. Now you can choose the attribute to search on along with a list of values.

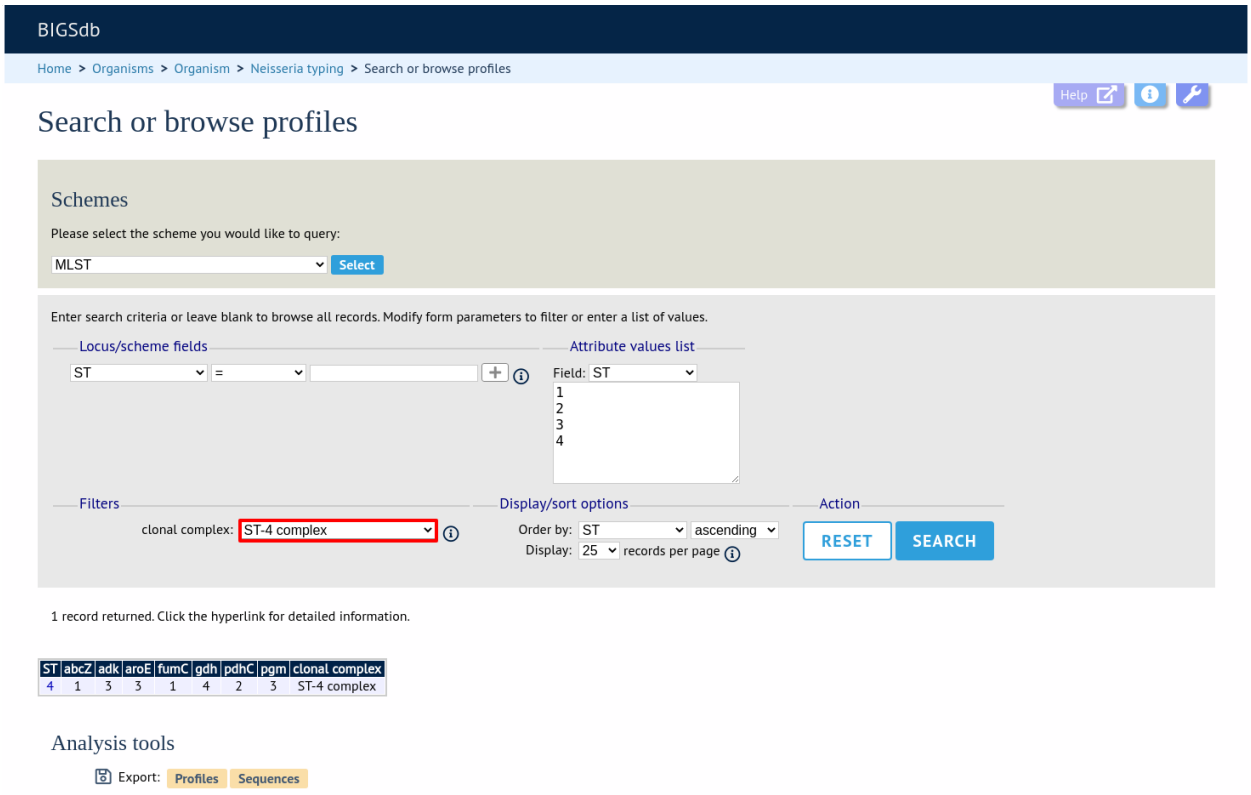


List values will be combined with any other attributes entered in the query form allowing complex queries can be constructed.

You can also add filters to the form by again clicking the 'Modify form options' tab and selecting 'Filters'.



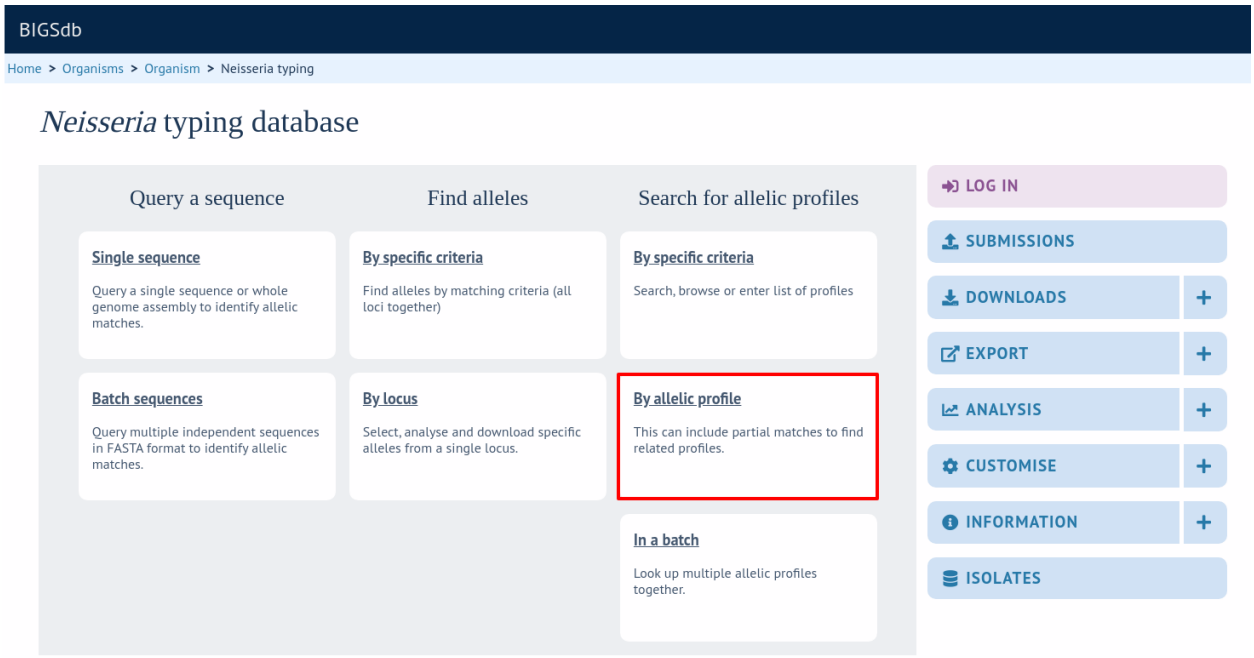
Available filters will vary depending on the database. These will be combined with other query criteria or lists of attributes.



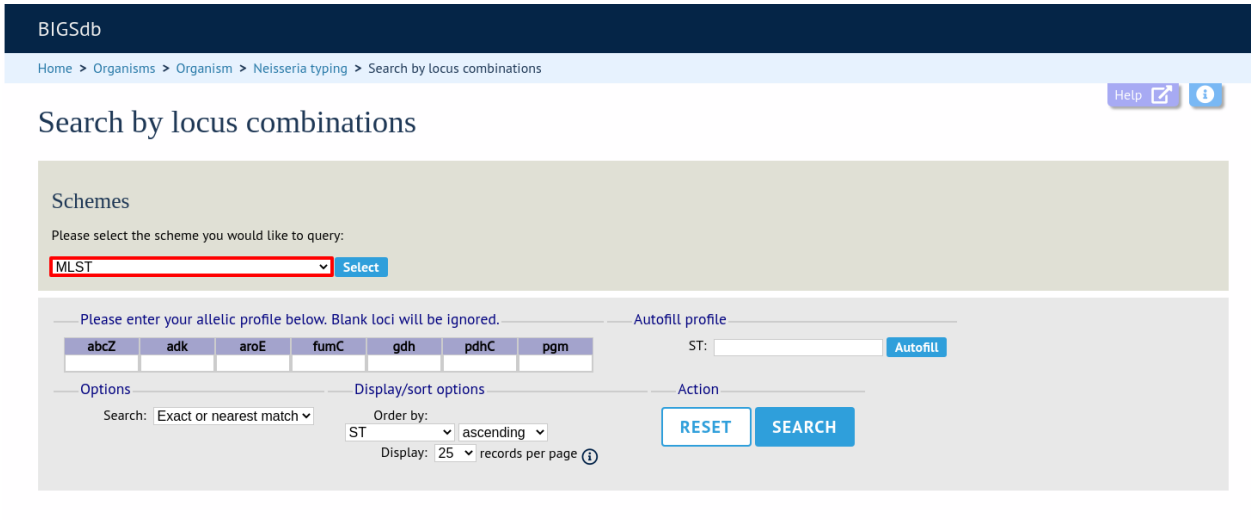
11.6 Identifying allelic profile definitions

For schemes such as MLST you can query allelic combinations to identify the sequence type (or more generically, the primary key of the profile).

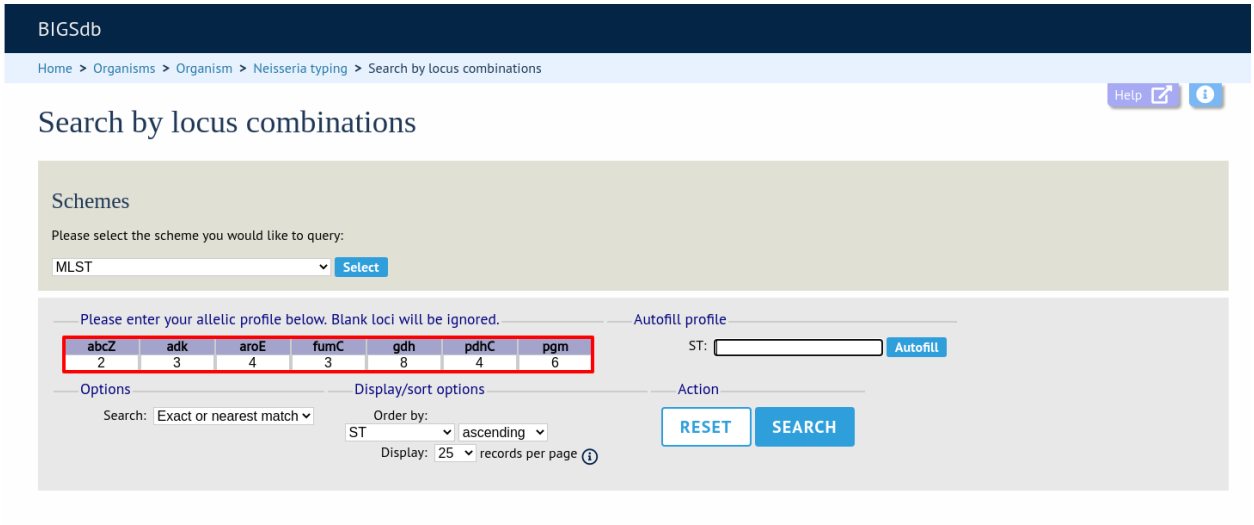
Click the 'By allelic profile' link in the 'Search for allelic profiles' section on a the sequence definition contents page.



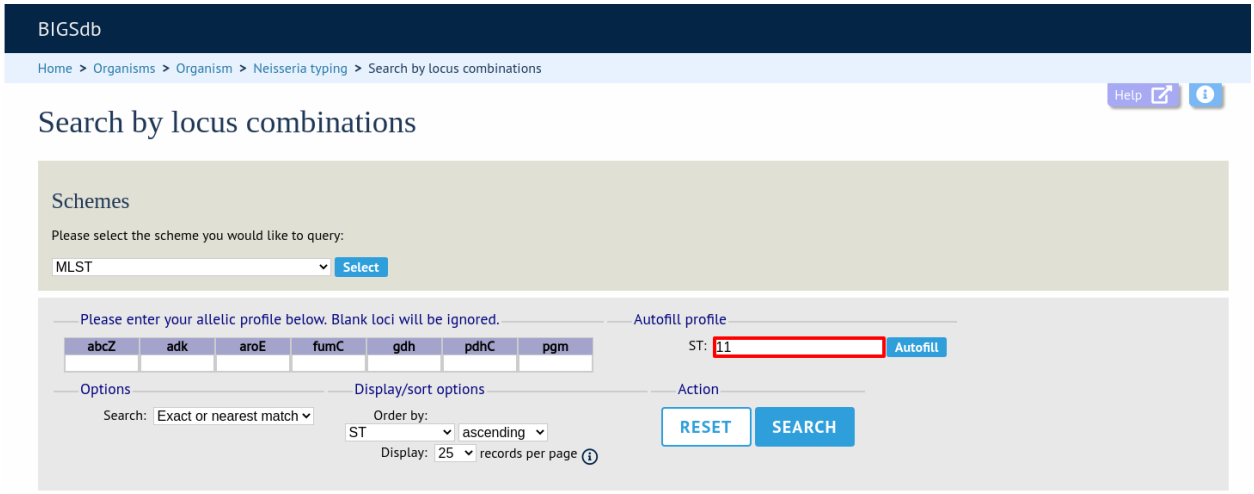
If multiple schemes are defined in the database you should select the scheme you wish to check.



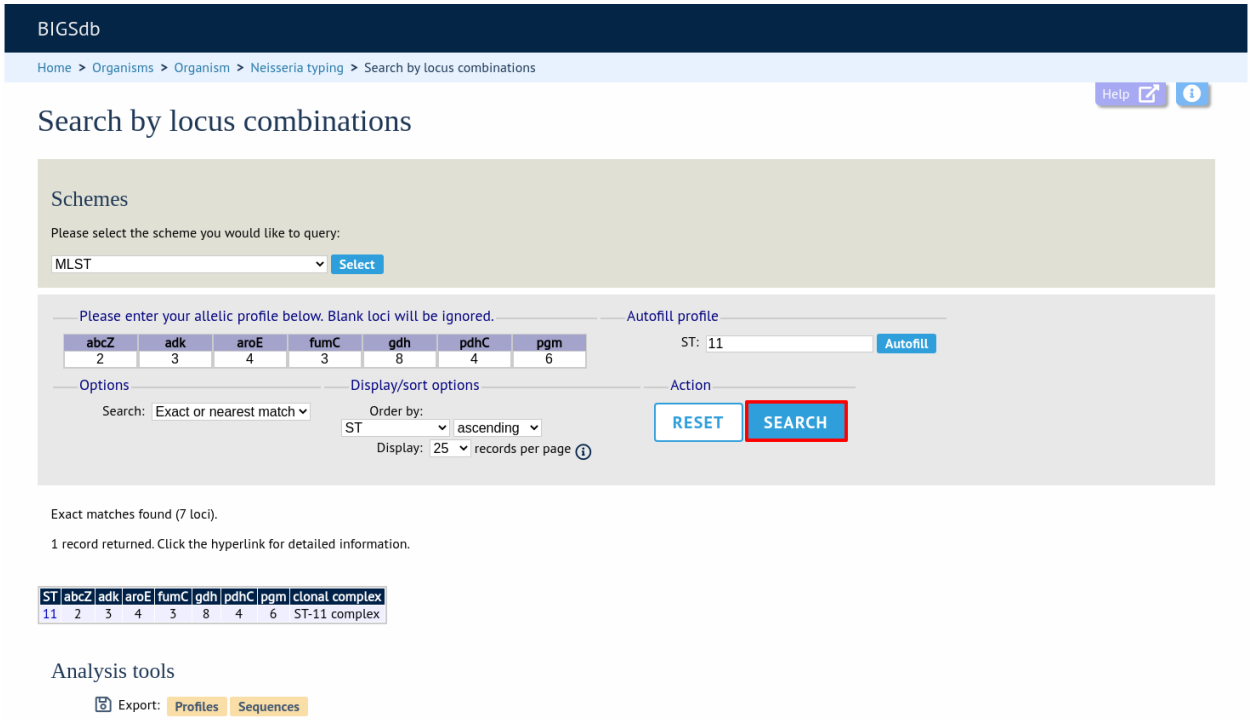
Enter a combination of allelic values (you can enter a partial profile if you wish).



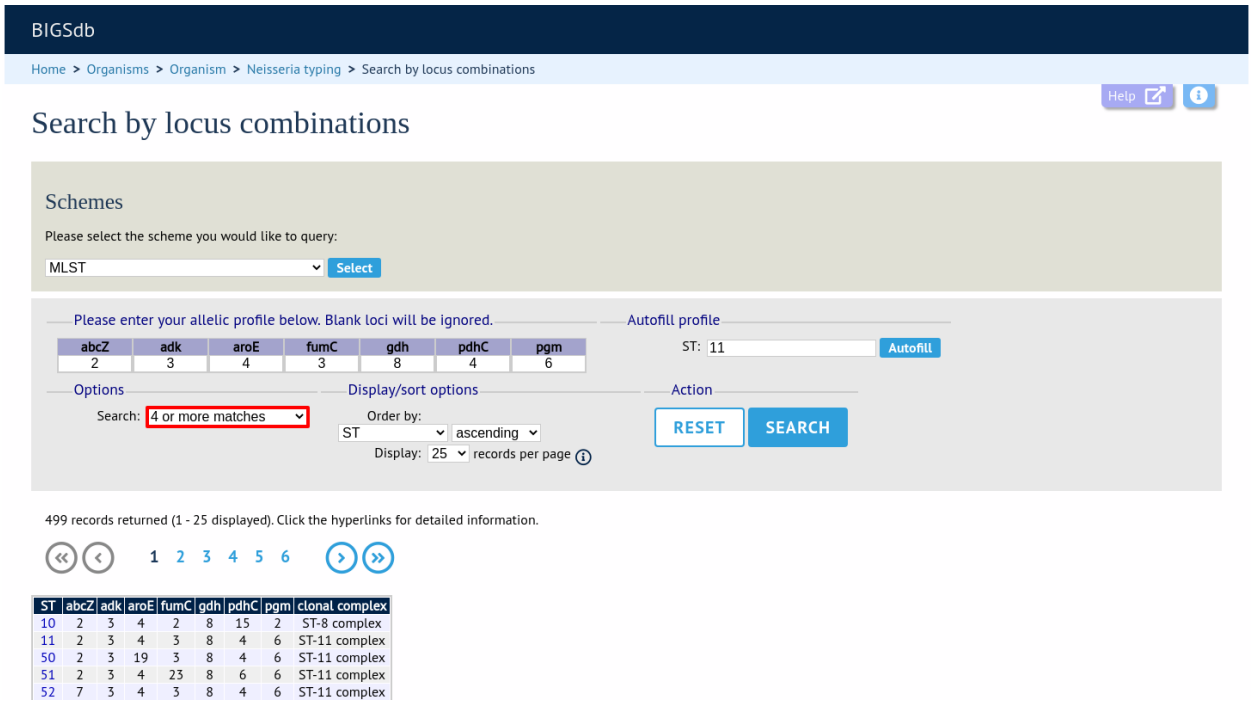
Alternatively, you can automatically populate a profile by entering a value for the scheme primary key field (e.g. ST) and clicking 'Autofill'.



To find the closest or exact match, leave the search box on 'Exact or nearest match' and click 'Submit'. The best match will be displayed.

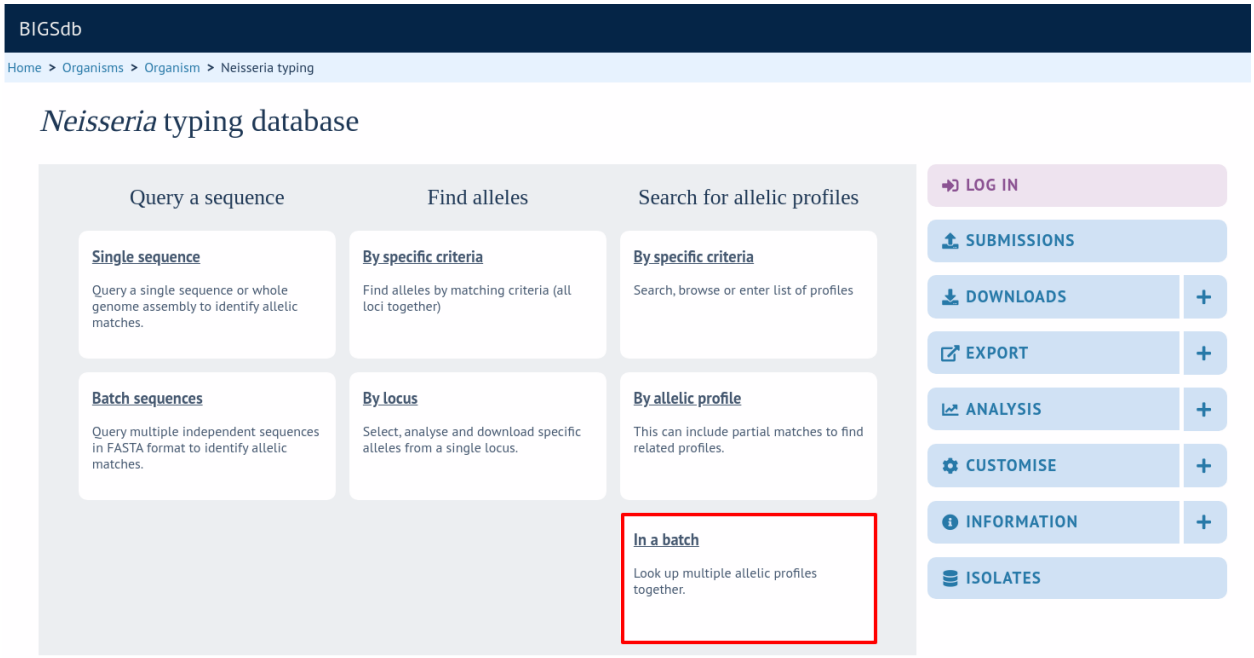


Alternatively, if you wish to find all profiles that match the query profile by at least a set number of loci, select the appropriate value in the search dropdown box, e.g. '4 or more matches' will show related profiles that share at least 4 alleles with the query.

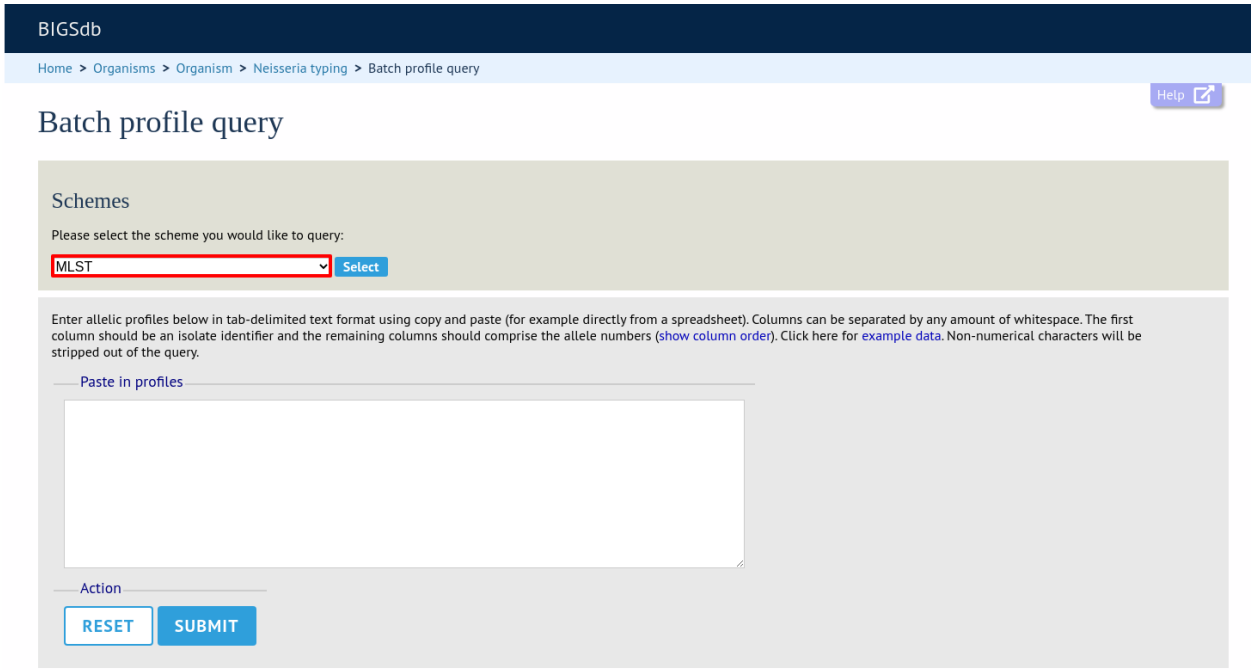


11.7 Batch profile queries

To lookup scheme definitions, e.g. the sequence type for multiple profiles, click 'In a batch' from the 'Search for allelic profiles' section of the sequence definition contents page.

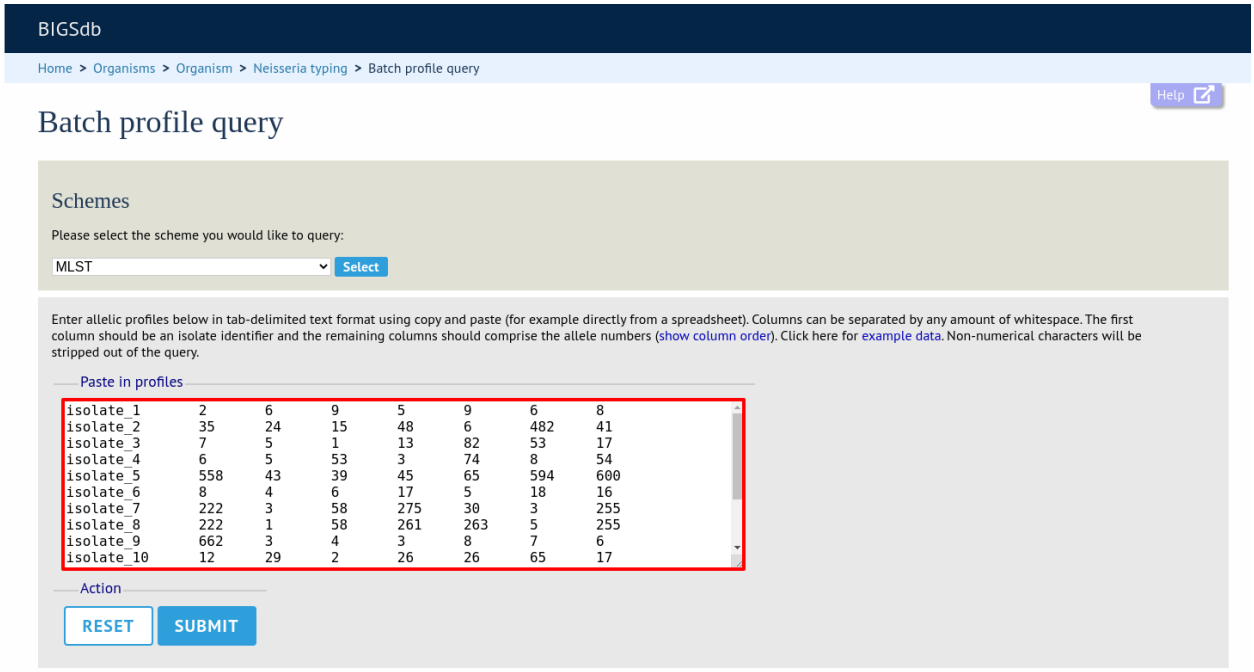


If multiple schemes are defined in the database you should select the scheme you wish to check.

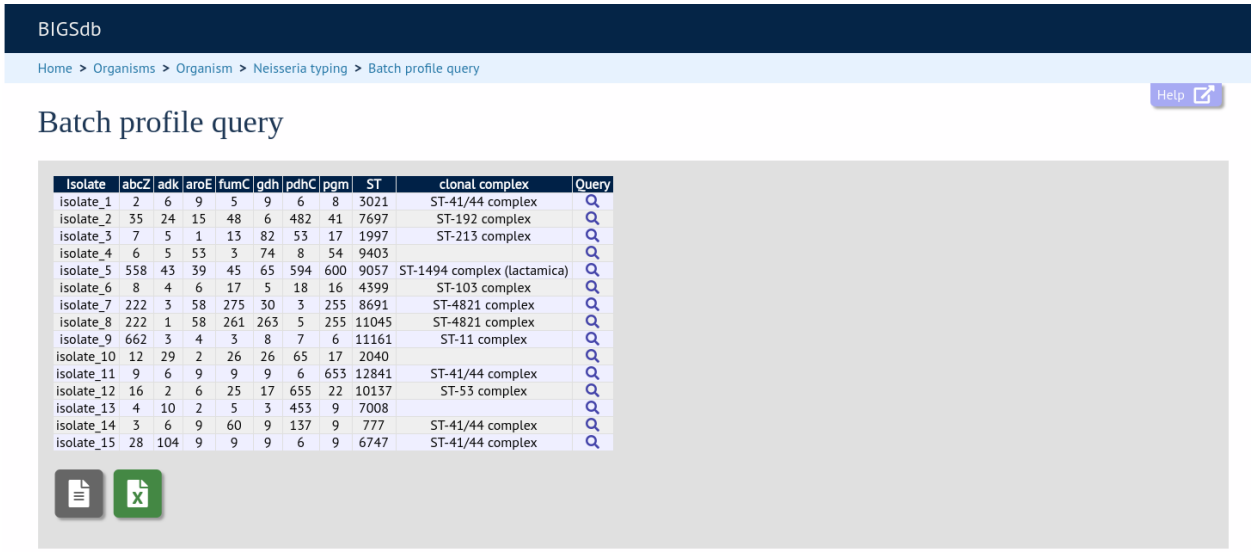


Copy and paste data from a spreadsheet. The first column is the record identifier, and the remaining columns are the alleles for each locus in the standard locus order defined for the scheme. There are links to the column order which can be used as a header line for your spreadsheet and to example data.

Click submit after pasting in the data.



A results table will be displayed.

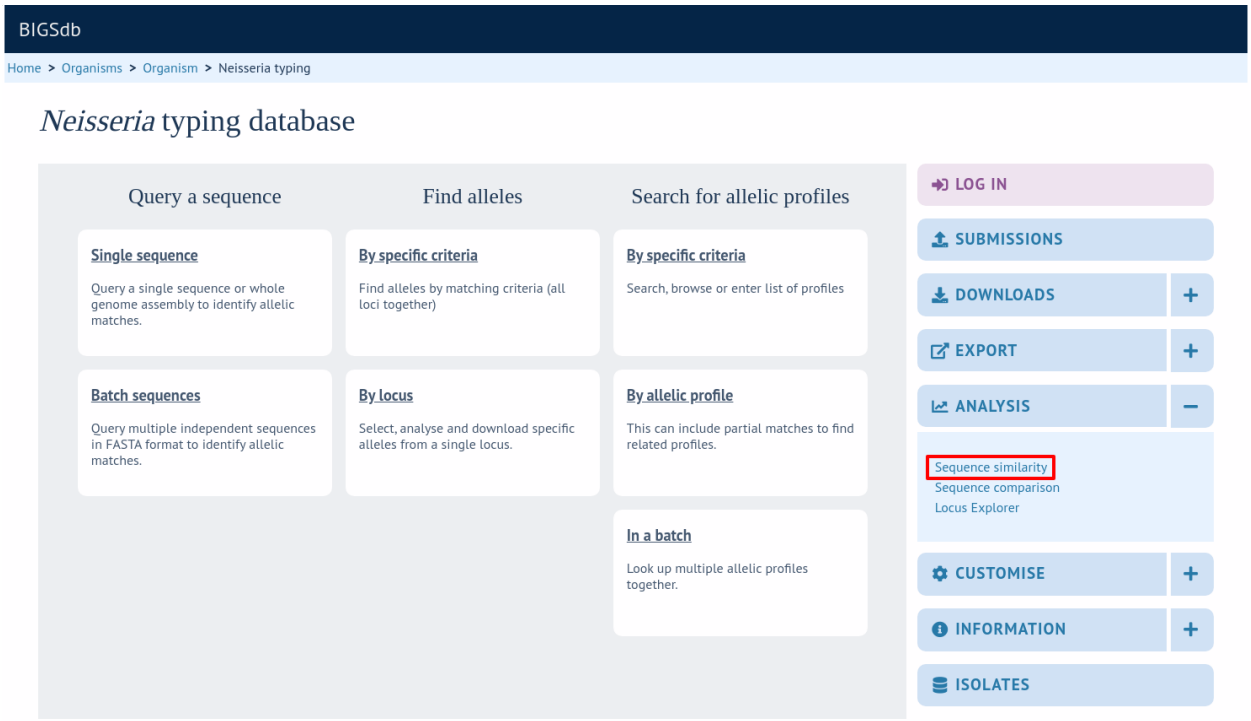


Isolate	abcZ	adk	aroE	fumC	gdh	pdhC	pgm	ST	clonal complex	Query
isolate_1	2	6	9	5	9	6	8	3021	ST-41/44 complex	
isolate_2	35	24	15	48	6	482	41	7697	ST-192 complex	
isolate_3	7	5	1	13	82	53	17	1997	ST-213 complex	
isolate_4	6	5	53	3	74	8	54	9403		
isolate_5	558	43	39	45	65	594	600	9057	ST-1494 complex (lactamica)	
isolate_6	8	4	6	17	5	18	16	4399	ST-103 complex	
isolate_7	222	3	58	275	30	3	255	8691	ST-4821 complex	
isolate_8	222	1	58	261	263	5	255	11045	ST-4821 complex	
isolate_9	662	3	4	3	8	7	6	11161	ST-11 complex	
isolate_10	12	29	2	26	26	65	17	2040		
isolate_11	9	6	9	9	9	6	653	12841	ST-41/44 complex	
isolate_12	16	2	6	25	17	655	22	10137	ST-53 complex	
isolate_13	4	10	2	5	3	453	9	7008		
isolate_14	3	6	9	60	9	137	9	777	ST-41/44 complex	
isolate_15	28	104	9	9	9	6	9	6747	ST-41/44 complex	

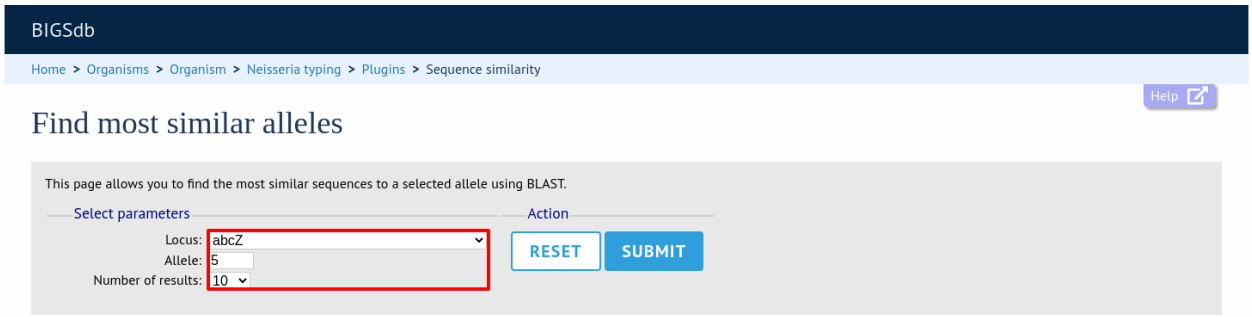
11.8 Investigating allele differences

11.8.1 Sequence similarity

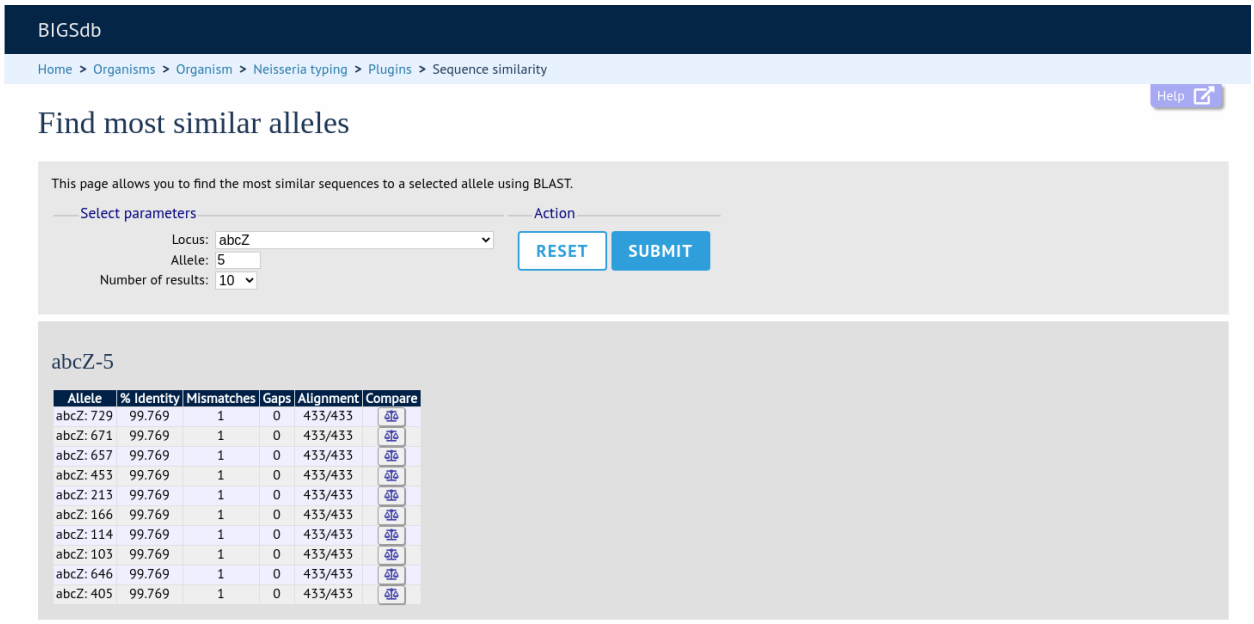
To find sequences most similar to a selected allele within a sequence definition database, expand the 'Analysis' menu item and click 'Sequence similarity' on the contents page.



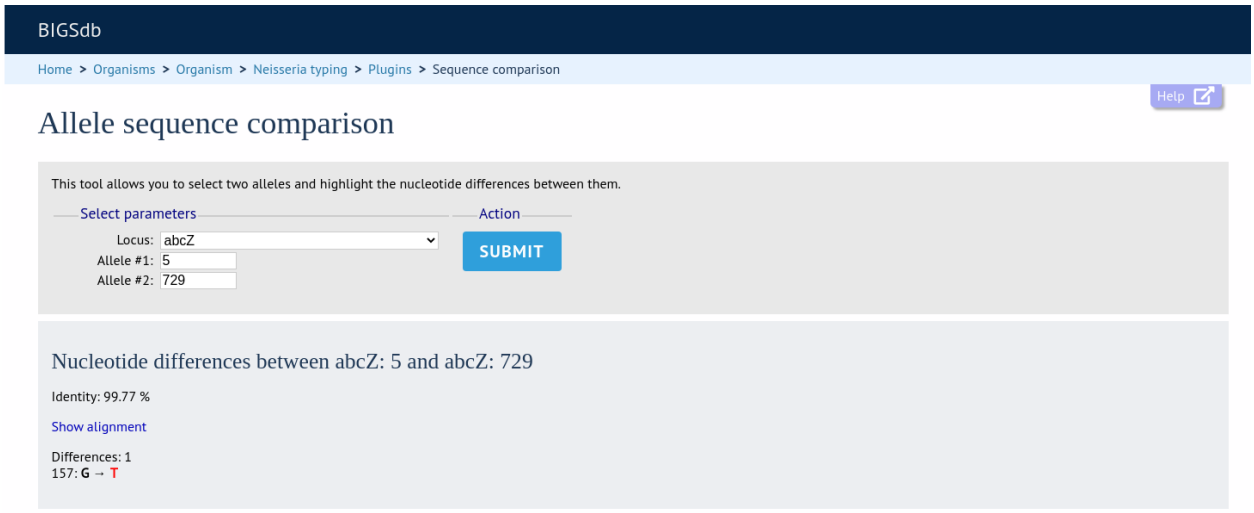
Enter the locus and allele identifer of the sequence to investigate and the number of nearest matches you'd like to see, then press submit.



A list of nearest alleles will be displayed, along with the percentage identity and number of gaps between the sequences.

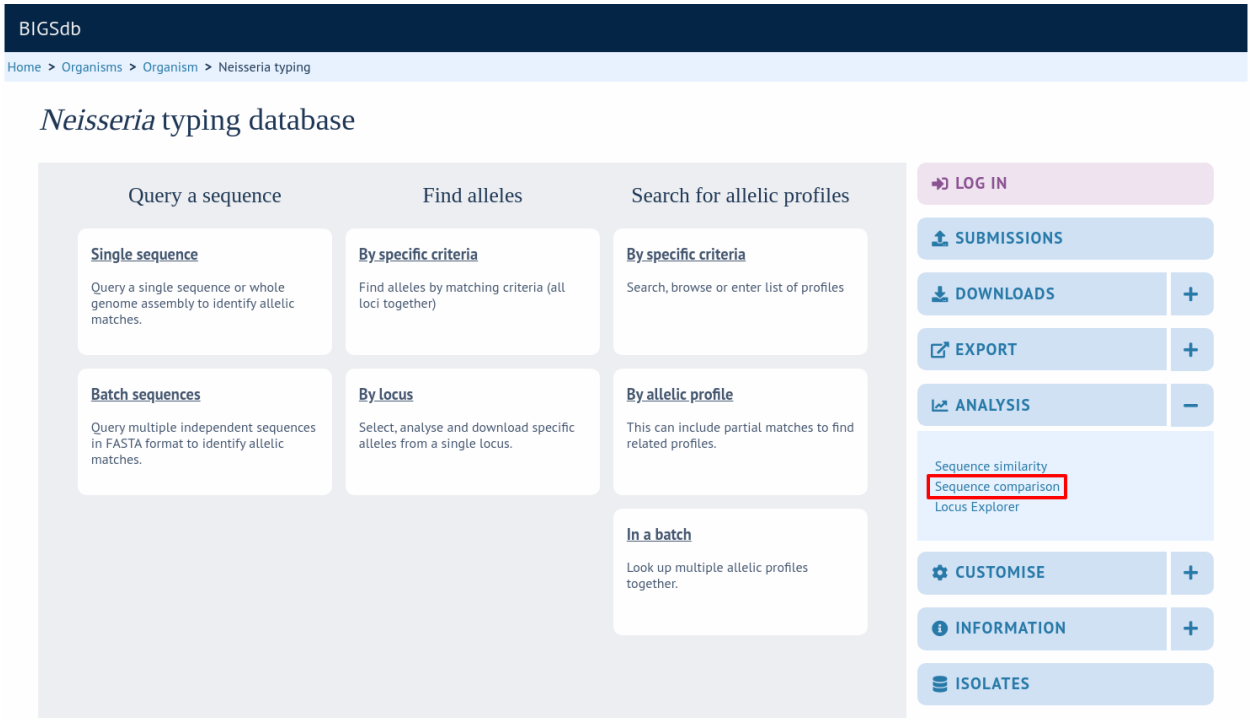


Click the appropriate 'Compare' button to display a list of nucleotide differences and/or a sequence alignment.

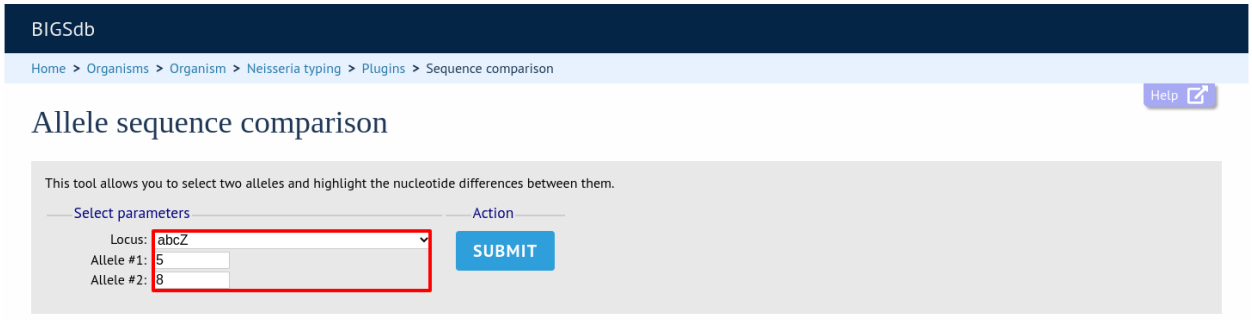


11.8.2 Sequence comparison

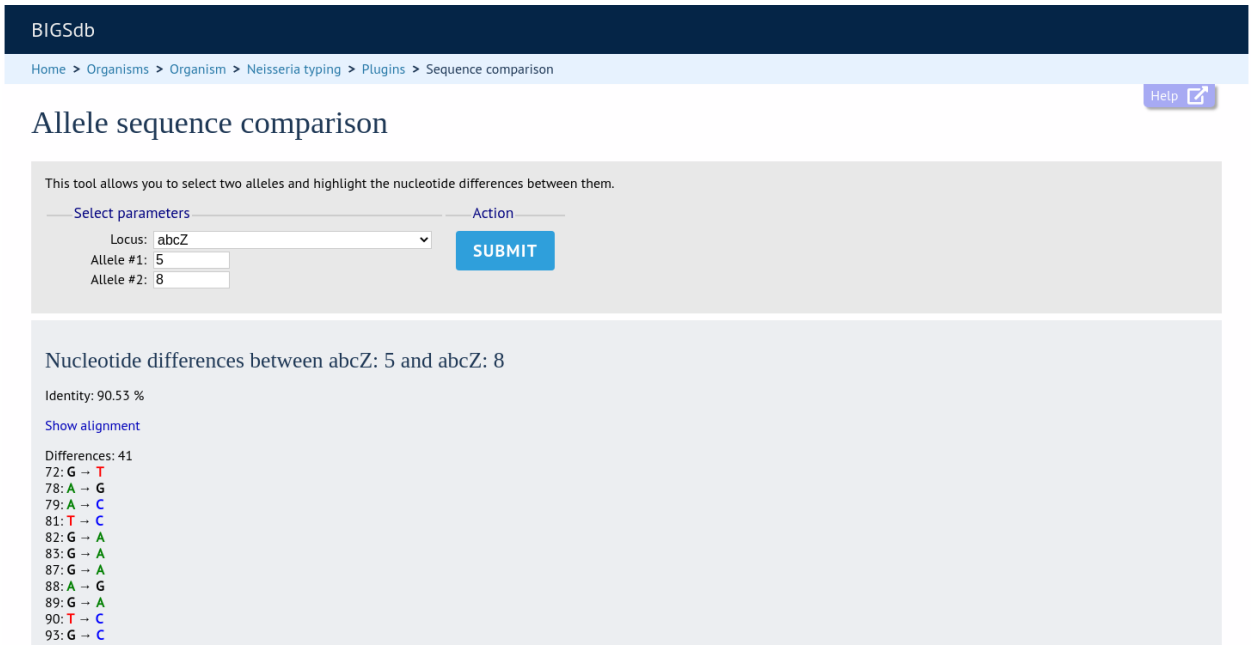
To directly compare two sequences, expand the 'Analysis' section and click 'Sequence comparison' on the contents page of a sequence definition database.



Enter the locus and two allele identifiers to compare. Press submit.



A list of nucleotide differences and/or an alignment will be displayed.

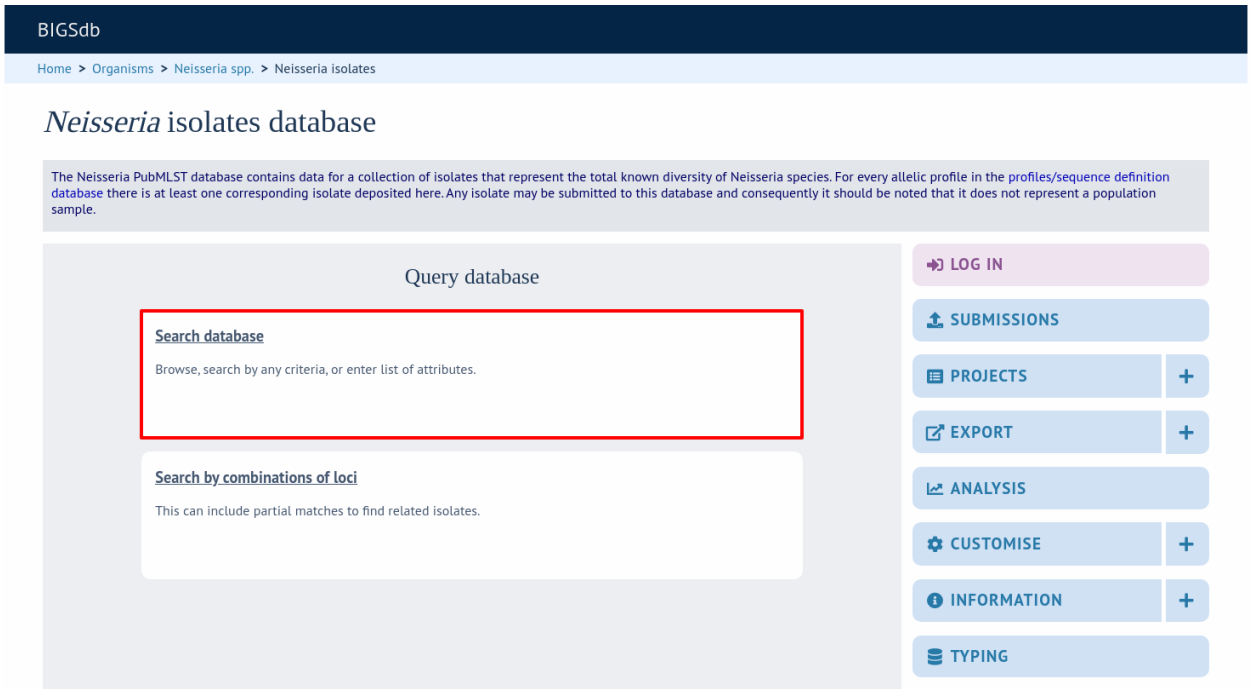


See also:

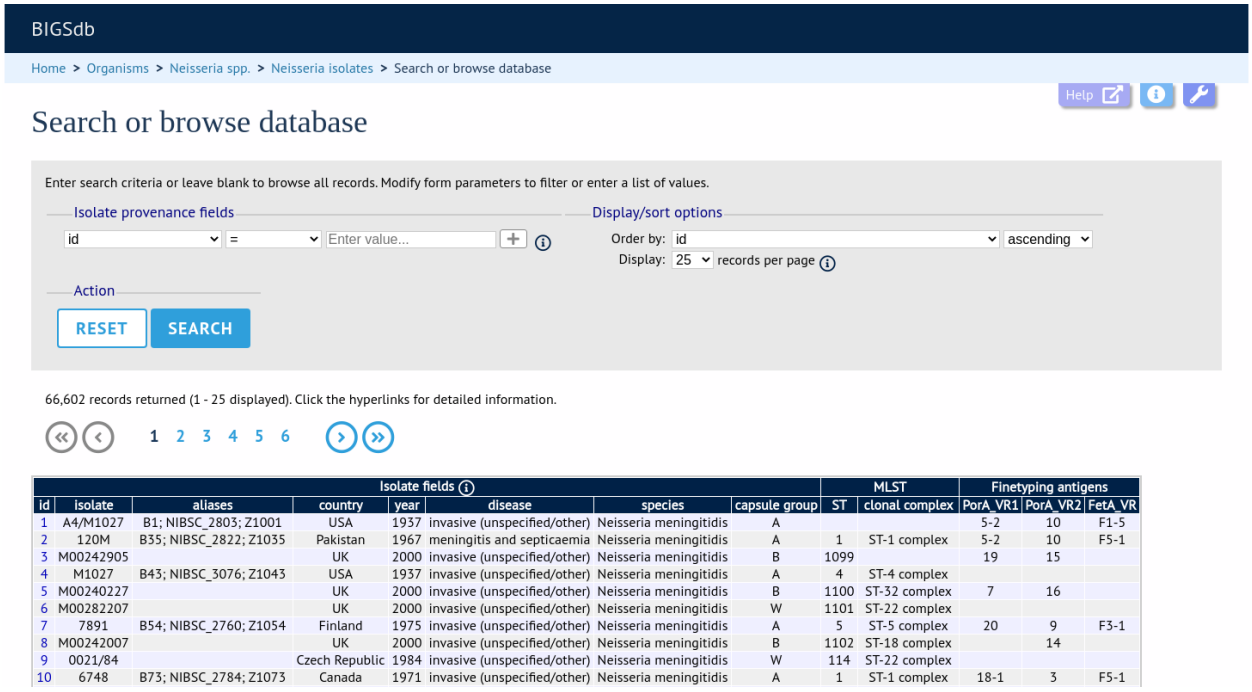
Locus explorer plugin.

11.9 Browsing isolate data

Isolate records can be browsed by clicking the link to 'Search database'.

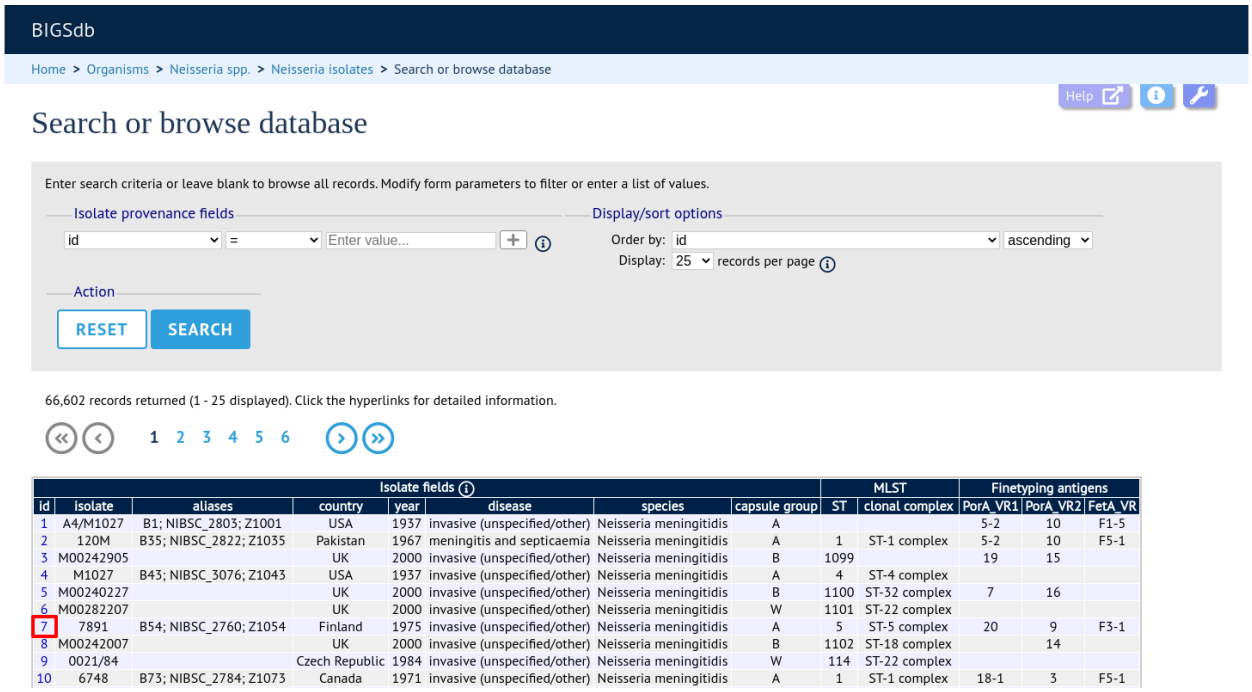


Leave query form fields blank (the display of these may vary depending on modification options set by the user). Choose the field to order the results by, the number of results per page to display, and click 'Search'.



Isolate fields								MLST		Finetyping antigens		
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex	PorA_VR1	PorA_VR2	FetA_VR
1	A4/M1027	B1; NIBSC_2803; Z1001	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-5
2	120M	B35; NIBSC_2822; Z1035	Pakistan	1967	meningitis and septicaemia	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
3	M00242905		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	B	1099		19	15	
4	M1027	B43; NIBSC_3076; Z1043	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex			
5	M00240227		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	B	1100	ST-32 complex	7	16	
6	M00282207		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	W	1101	ST-22 complex			
7	7891	B54; NIBSC_2760; Z1054	Finland	1975	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-1
8	M00240007		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	B	1102	ST-18 complex		14	
9	0021/84		Czech Republic	1984	invasive (unspecified/other)	Neisseria meningitidis	W	114	ST-22 complex			
10	6748	B73; NIBSC_2784; Z1073	Canada	1971	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	18-1	3	F5-1

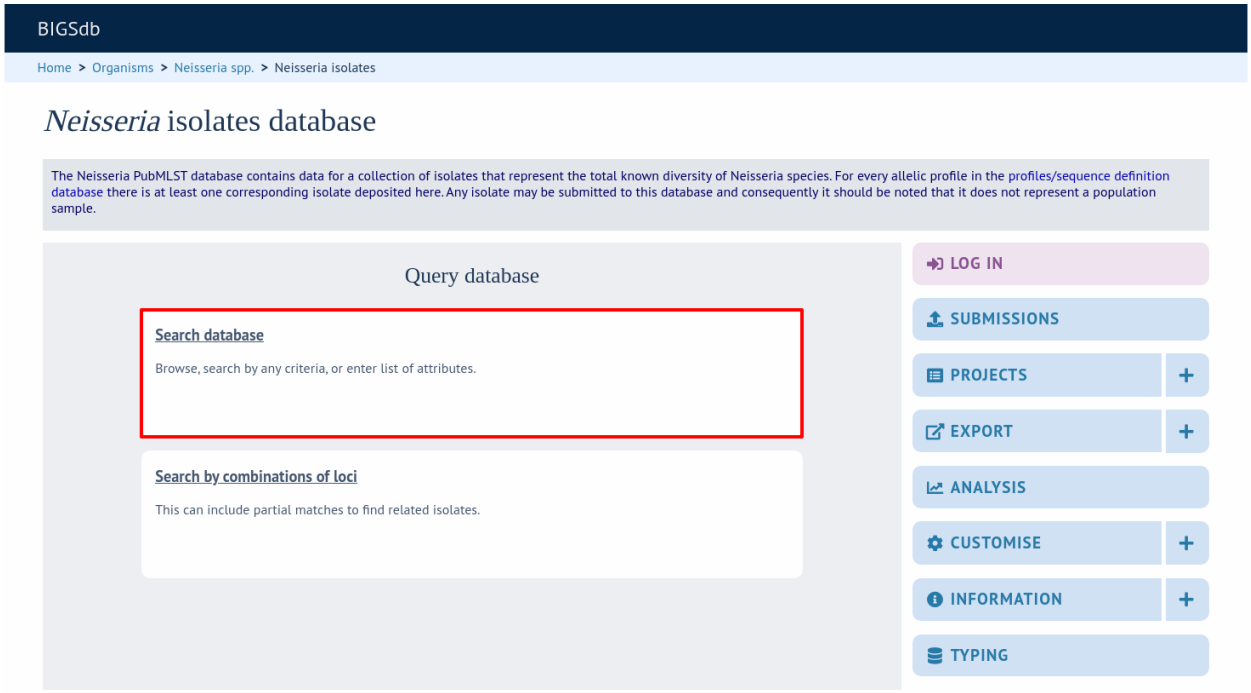
Clicking the hyperlink for any record will display *full information* about the profile.



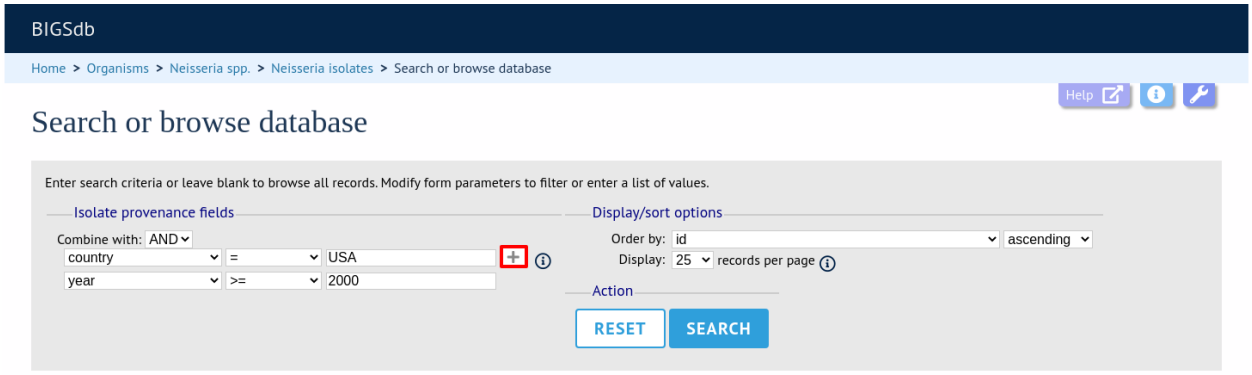
Isolate fields								MLST		Finetyping antigens		
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex	PorA_VR1	PorA_VR2	FetA_VR
1	A4/M1027	B1; NIBSC_2803; Z1001	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-5
2	120M	B35; NIBSC_2822; Z1035	Pakistan	1967	meningitis and septicaemia	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
3	M00242905		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	B	1099		19	15	
4	M1027	B43; NIBSC_3076; Z1043	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex			
5	M00240227		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	B	1100	ST-32 complex	7	16	
6	M00282207		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	W	1101	ST-22 complex			
7	7891	B54; NIBSC_2760; Z1054	Finland	1975	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-1
8	M00242007		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	B	1102	ST-18 complex		14	
9	0021/84		Czech Republic	1984	invasive (unspecified/other)	Neisseria meningitidis	W	114	ST-22 complex			
10	6748	B73; NIBSC_2784; Z1073	Canada	1971	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	18-1	3	F5-1

11.10 Querying isolate data

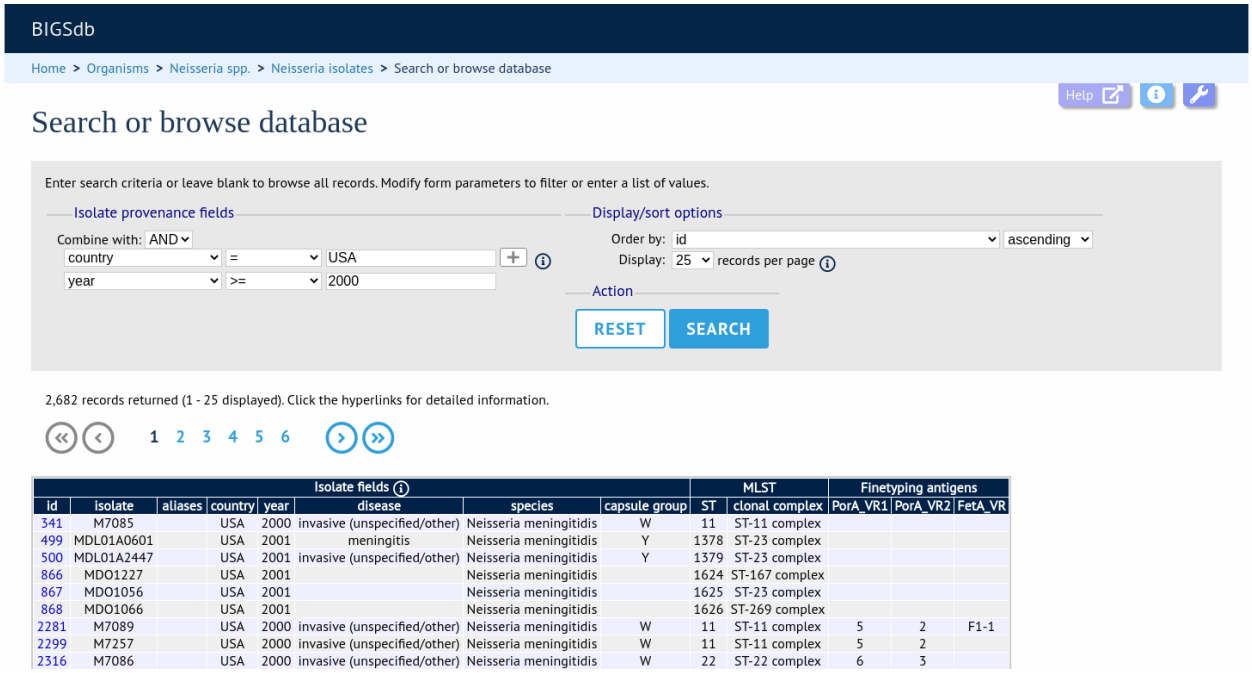
The 'Search database' page of an isolate database allows you to also search by combinations of provenance criteria, scheme and locus data, and more.



To start with, only one provenance field search box is displayed but more can be added by clicking the '+' button (highlighted). These can be linked together by 'and' or 'or'.

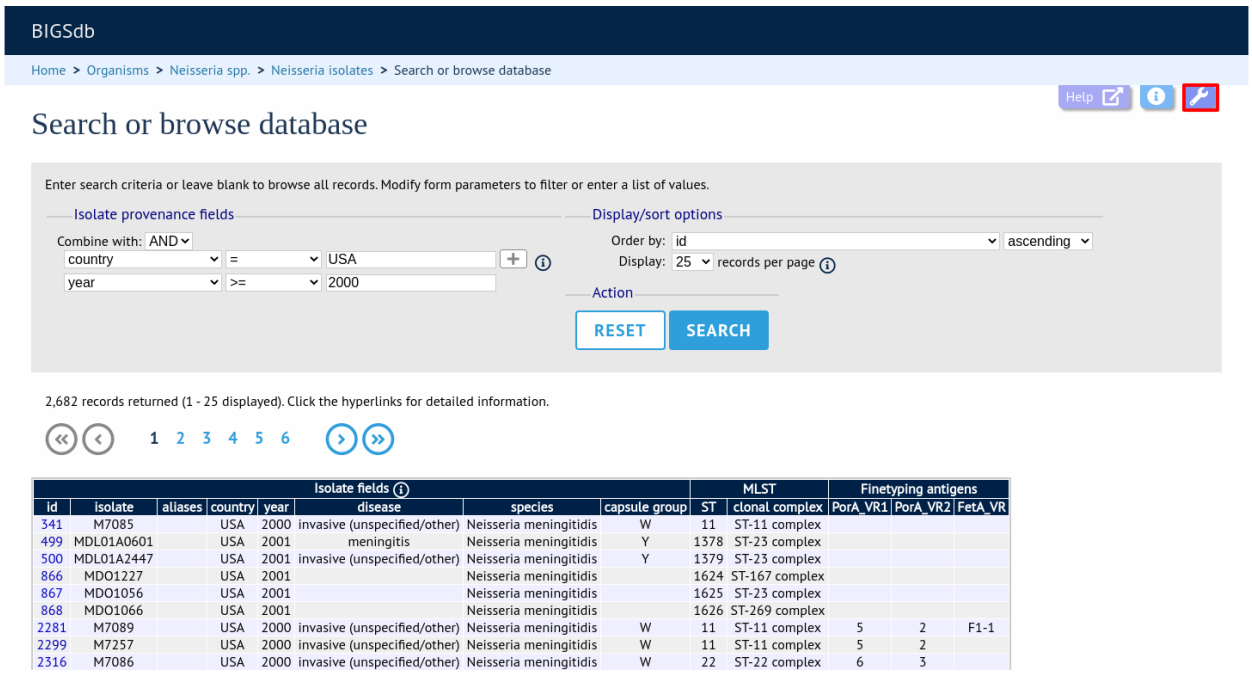


After the search has been submitted, the results will be displayed in a table.



Each field can be queried using *standard operators*.

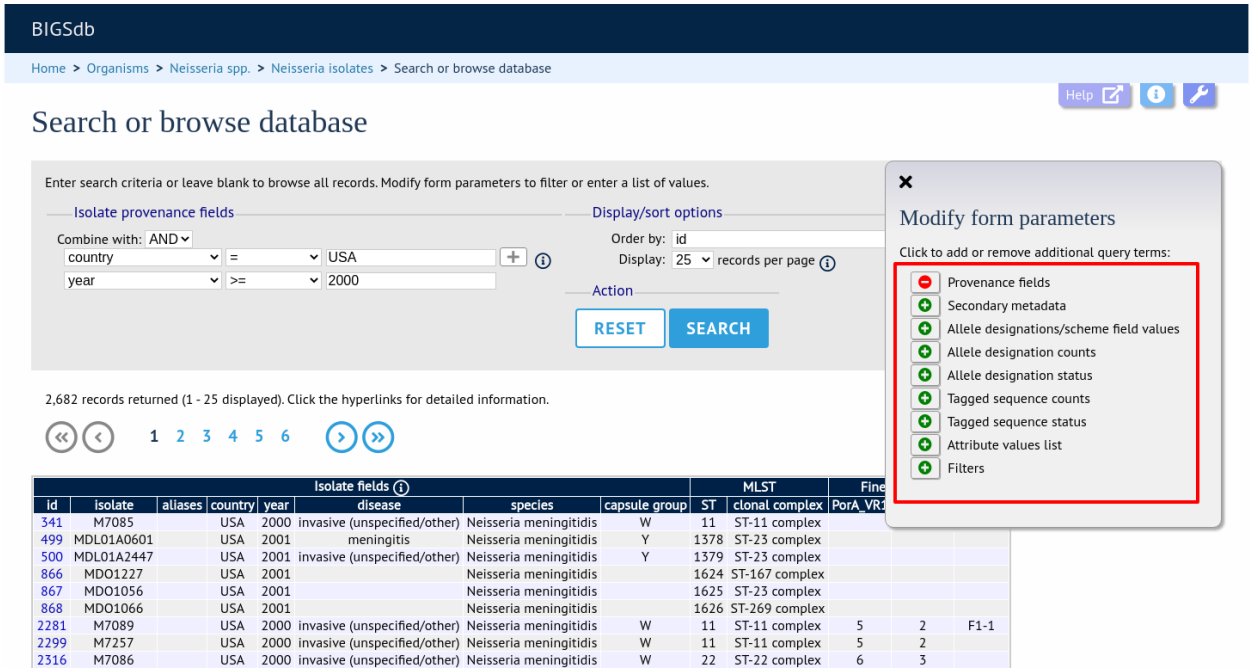
More search features are available by clicking the 'Modify form options' tab on the top of the screen.



A tab will be displayed. Different options will be available here depending on the database. Queries will be combined from the values entered in all form sections. Possible options are:

- Provenance fields
  - Search by combination of provenance field values, e.g. country, year, sender.
- Allele designations/scheme field values
  - Search by combination of allele designations and/or scheme fields e.g. ST, clonal complex information.

- Allele designation status
  - Search by whether allele designation status is confirmed or provisional.
- Tagged sequence status
  - Search by whether tagged sequence data is available for a locus. You can also search by sequence flags.
- Attribute values list
  - Enter a list of values for any provenance field, locus, or scheme field.
- Filters
  - Various filters may be available, including
    - Publications
    - Projects
    - MLST profile completion status
    - Clonal complex
    - Sequence bin size
    - Inclusion/exclusion of *old versions*

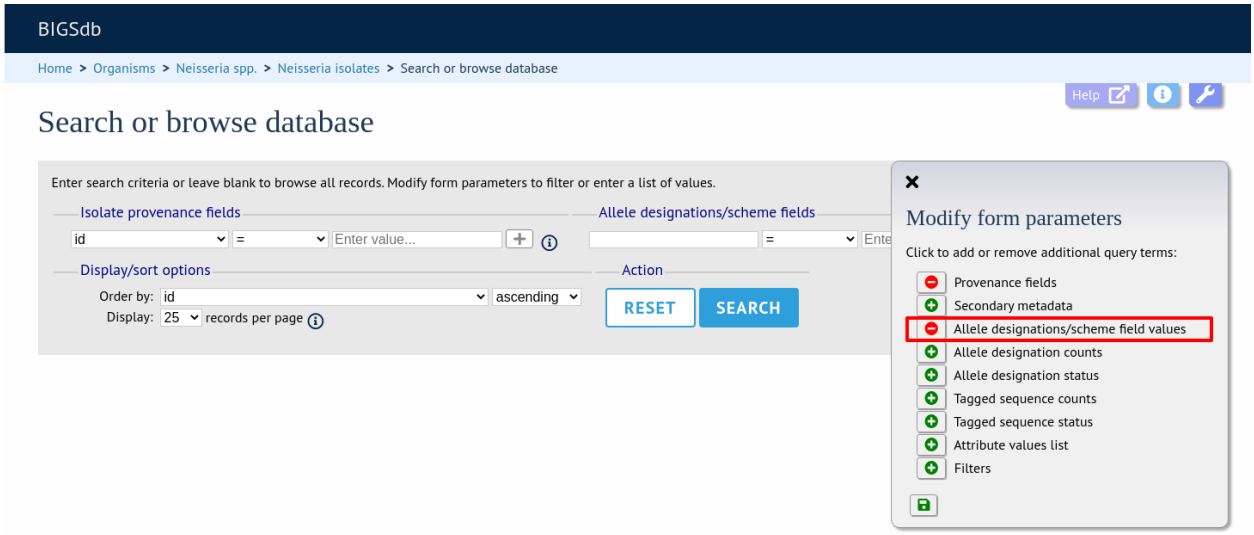


If the interface is modified, a button to save options becomes available within the tab. If this is clicked, the modified form will be displayed the next time you go to the query page.

11.10.1 Query by allele designation/scheme field

Queries can be combined with allele designation/scheme field values.

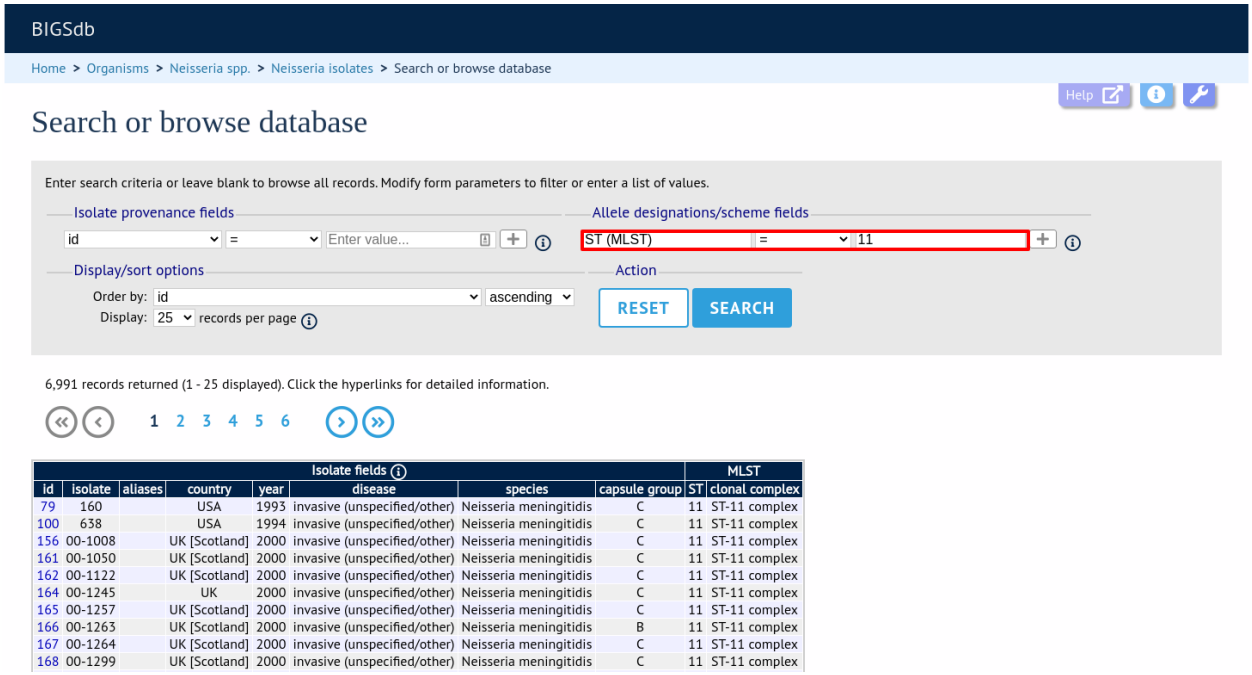
Make sure that the allele designation/scheme field values fieldset is displayed by selecting it in the 'Modify form options' tab.



Designations can be queried using *standard operators*.

Additional search terms can be combined using the '+' button.

Add your search terms and click 'Submit'. Allele designation/scheme field queries will be combined with terms entered in other sections.

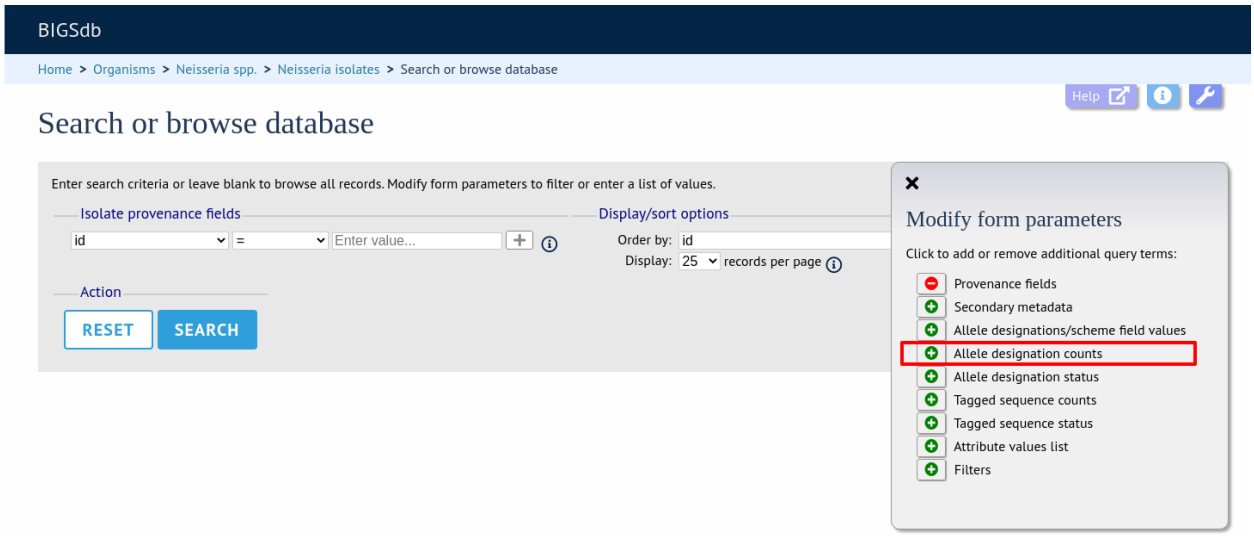


Isolate fields								MLST	
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex
79	160		USA	1993	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
100	638		USA	1994	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
156	00-1008		UK [Scotland]	2000	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
161	00-1050		UK [Scotland]	2000	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
162	00-1122		UK [Scotland]	2000	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
164	00-1245		UK	2000	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
165	00-1257		UK [Scotland]	2000	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
166	00-1263		UK [Scotland]	2000	invasive (unspecified/other)	Neisseria meningitidis	B	11	ST-11 complex
167	00-1264		UK [Scotland]	2000	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex
168	00-1299		UK [Scotland]	2000	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex

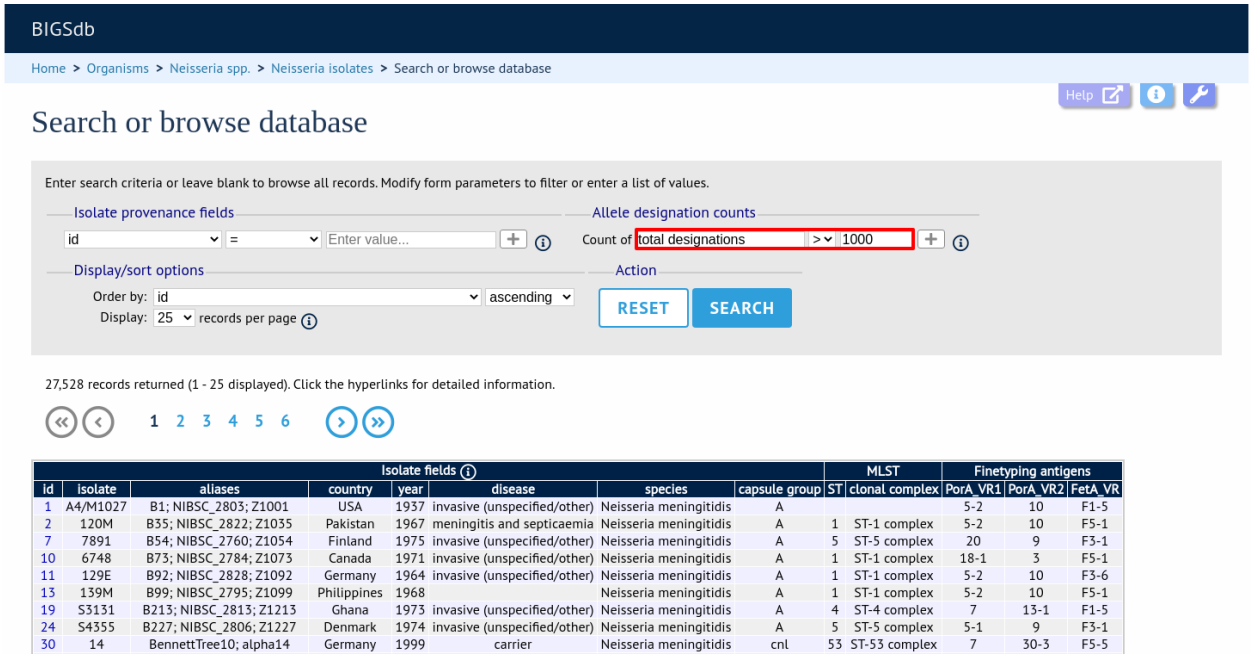
11.10.2 Query by allele designation count

Queries can be combined with counts of the total number of designations or for individual loci.

Make sure that the allele designation counts fieldset is displayed by selecting it in the 'Modify form options' tab.



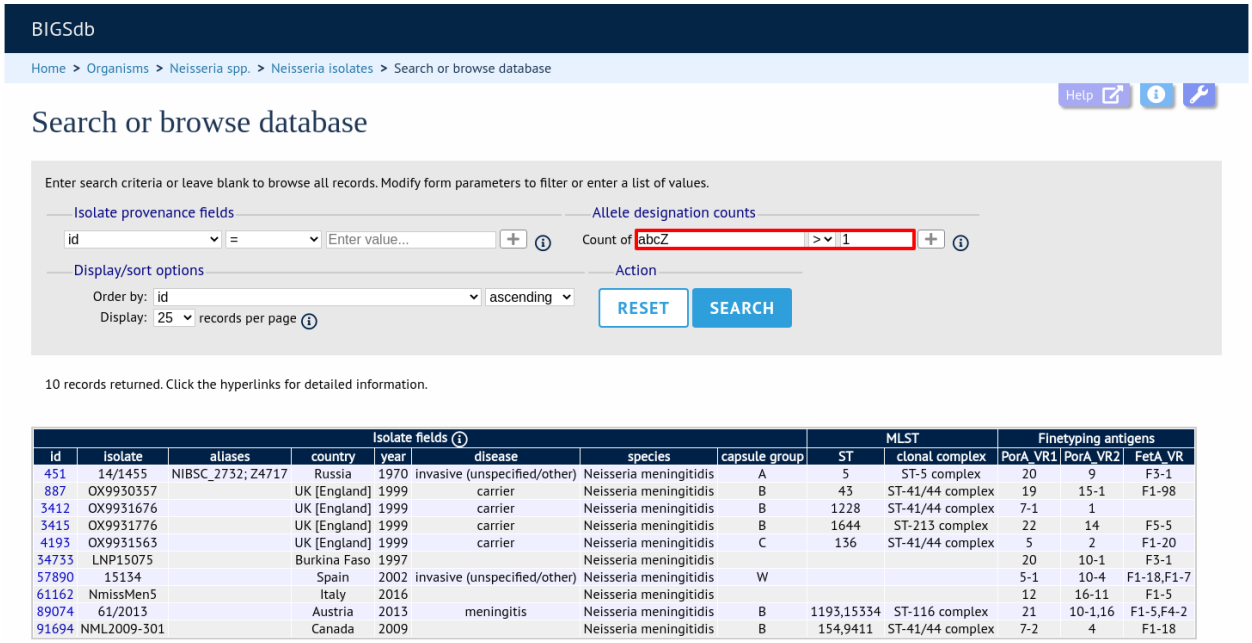
For example, to find all isolates that have designations at >1000 loci, select 'total designations > 1000', then click 'Submit'.



Isolate fields								MLST		Finetyping antigens		
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex	PorA_VR1	PorA_VR2	FetA_VR
1	A4/M1027	B1; NIBSC_2803; Z1001	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-5
2	120M	B35; NIBSC_2822; Z1035	Pakistan	1967	meningitis and septicaemia	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
7	7891	B54; NIBSC_2760; Z1054	Finland	1975	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-1
10	6748	B73; NIBSC_2784; Z1073	Canada	1971	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	18-1	3	F5-1
11	129E	B92; NIBSC_2828; Z1092	Germany	1964	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F3-6
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
30	14	BennettTree10; alpha14	Germany	1999	carrier	Neisseria meningitidis	cnl	53	ST-53 complex	7	30-3	F5-5

You can also search for isolates where any isolate has a particular number of designations. Use the term 'any locus' to do this.

Finally, you can search for isolates with a specific number of designations at a specific locus.



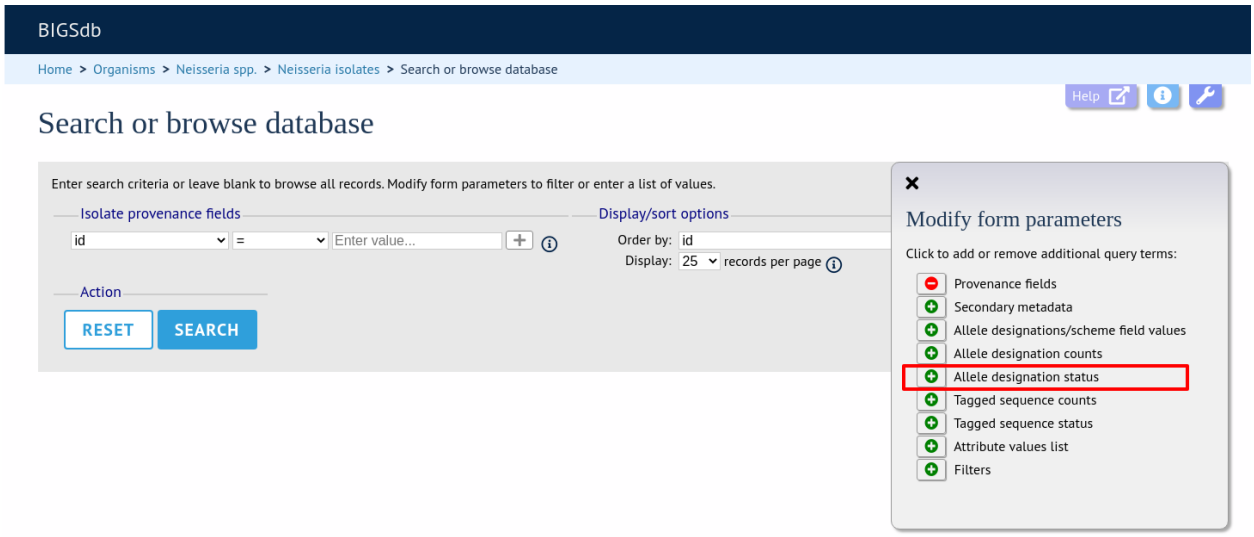
Additional search terms can be combined using the '+' button. Designation count queries will be combined with terms entered in other sections.

Note: Searches for 'all loci' with counts that include zero, e.g. 'count of any locus = 0' or with a '<' operator are not supported. This is because such searches have to identify every isolate for which one or more loci are missing. In databases with thousands of loci this can be a very expensive database query.

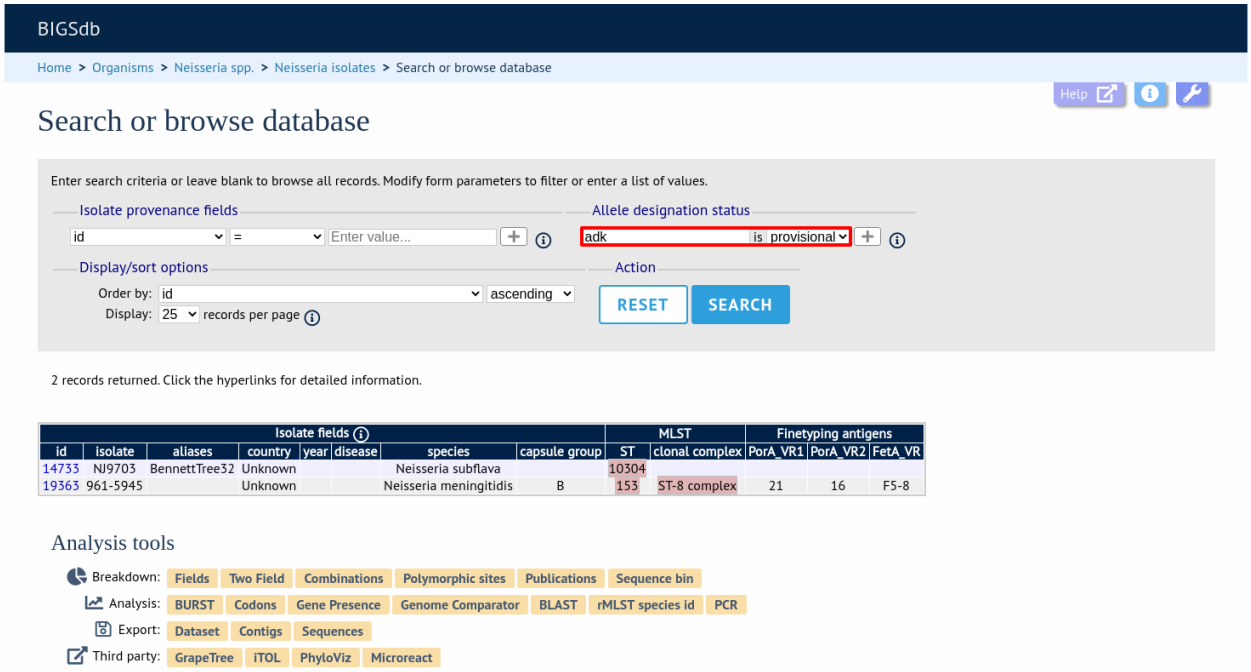
11.10.3 Query by allele designation status

Allele designations can be queried based on their status, i.e. whether they are confirmed or provisional. Queries will be combined from the values entered in all form sections.

Make sure that the allele designation staus fieldset is displayed by selecting it in the 'Modify form options' tab.



Select a locus from the dropdown box and either 'provisional' or 'confirmed'. Additional query fields can be displayed by clicking the '+' button. Click 'Submit'.

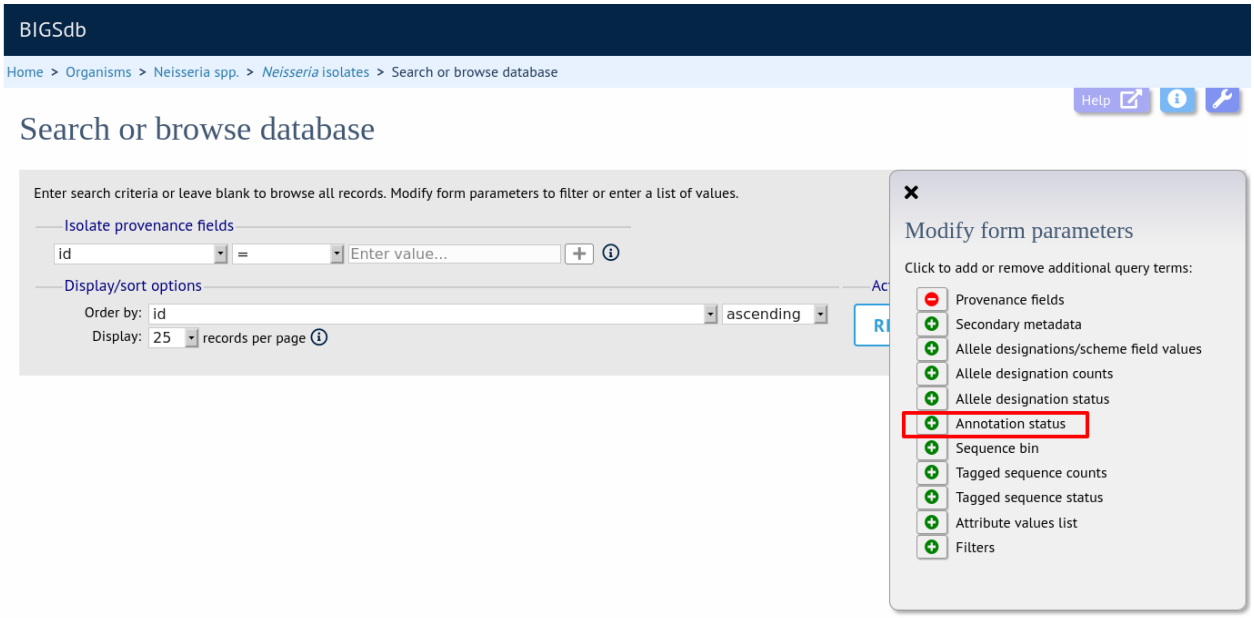


Provisional allele designations are marked within the results tables with a pink background. Any scheme field designations that depend on the allele in question, e.g. a MLST ST, will also be marked as provisional.

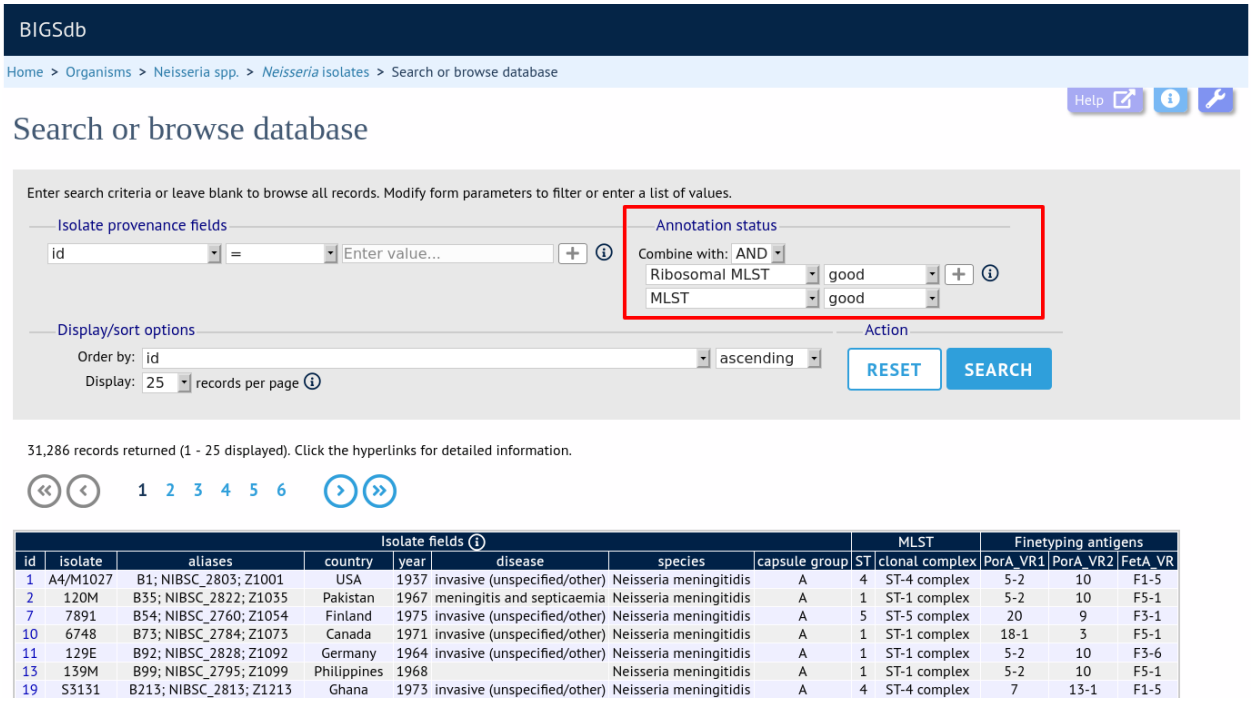
11.10.4 Query by annotation status

Isolates can be queried by the annotation status of particular schemes if these have been set up. The idea is that for a well-annotated record the isolate would be expected to have allele designations for all loci in the scheme. Alternatively, different thresholds for number of loci with allele designations can be set up by the scheme administrator to indicate good or bad quality thresholds.

Make sure that the annotation status fieldset is displayed by selecting it in the 'Modify form options' tab.



Additional search terms can be combined using the '+' button. Annotation status queries will be combined with terms entered in other sections.

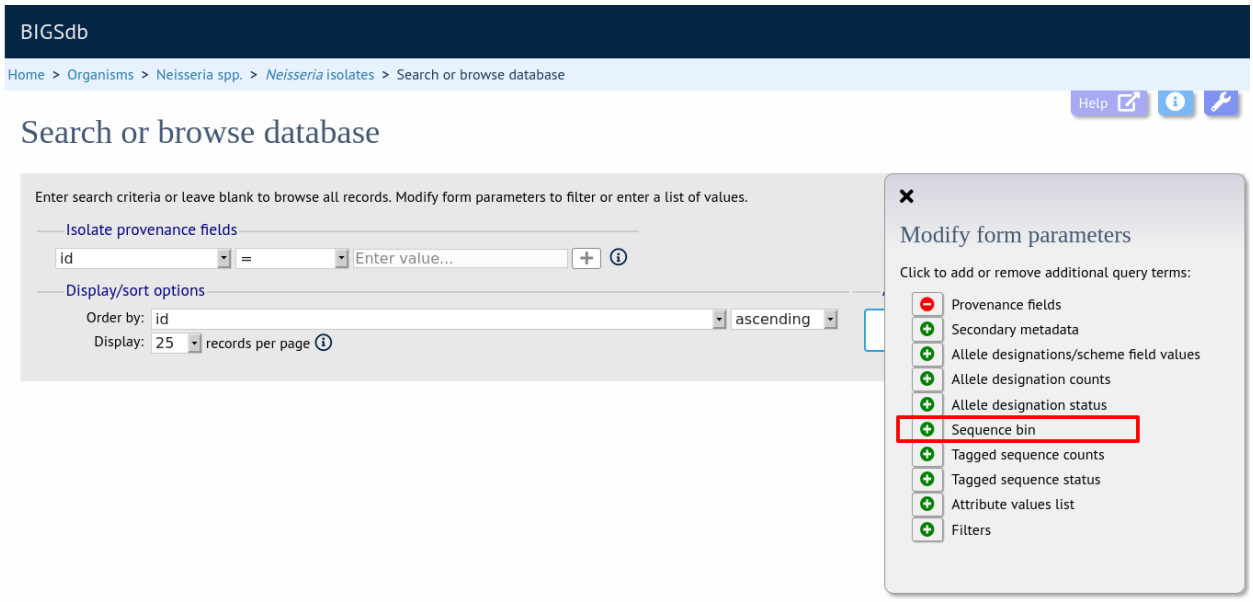


Isolate fields								MLST		Finetyping antigens		
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex	PorA_VR1	PorA_VR2	FetA_VR
1	A4/M1027	B1; NIBSC_2803; Z1001	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	5-2	10	F1-5
2	120M	B35; NIBSC_2822; Z1035	Pakistan	1967	meningitis and septicaemia	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
7	7891	B54; NIBSC_2760; Z1054	Finland	1975	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-1
10	6748	B73; NIBSC_2784; Z1073	Canada	1971	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	18-1	3	F5-1
11	129E	B92; NIBSC_2828; Z1092	Germany	1964	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F3-6
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5

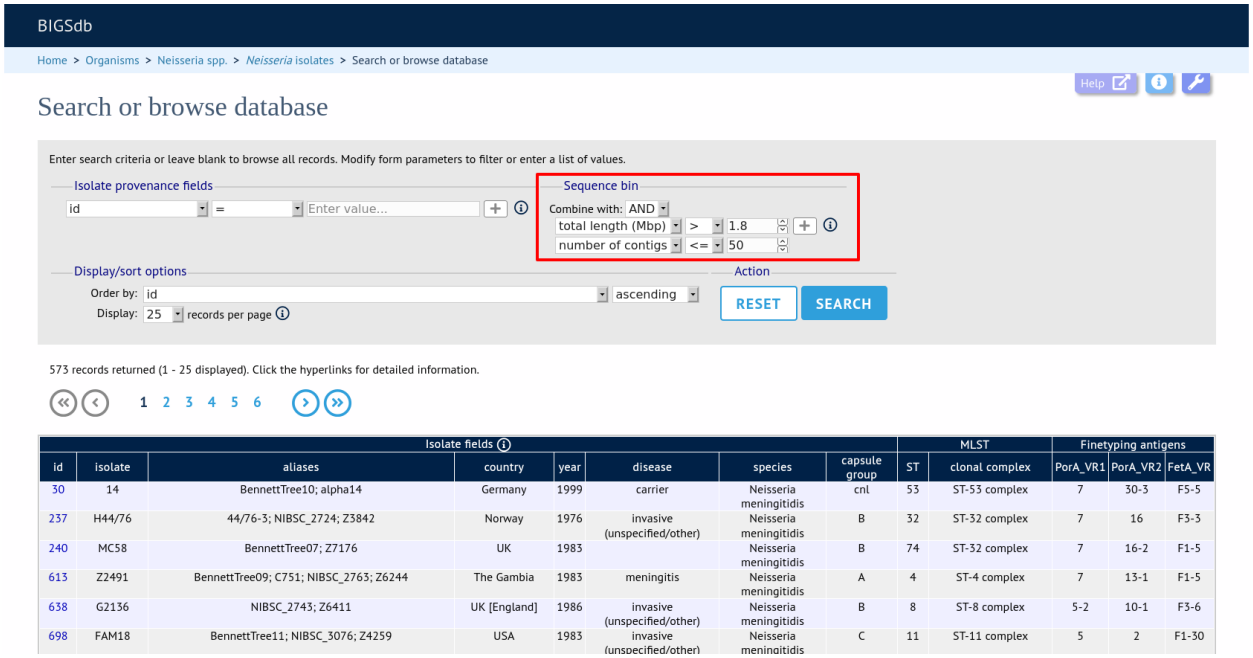
11.10.5 Query by sequence bin size and number of contigs

Isolates can be queried based on the total length of sequences within the sequence bin, the number of contigs, the N50 and/or the L50 values.

Make sure that the sequence bin fieldset is displayed by selecting it in the 'Modify form options' tab.



Additional search terms can be combined using the '+' button. Sequence bin queries will be combined with terms entered in other sections.

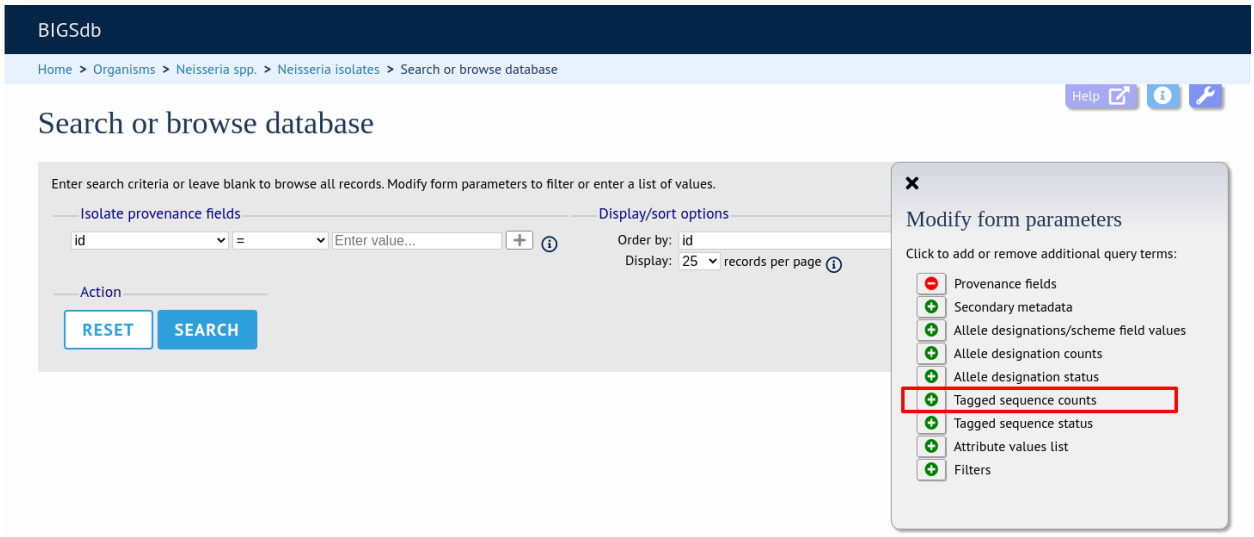


Isolate fields								MLST		Finetyping antigens		
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex	PorA_VR1	PorA_VR2	FetA_VR
30	14	BennettTree10; alpha14	Germany	1999	carrier	Neisseria meningitidis	cnl	53	ST-53 complex	7	30-3	F5-5
237	H44/76	44/76-3; NIBSC_2724; Z3842	Norway	1976	invasive (unspecified/other)	Neisseria meningitidis	B	32	ST-32 complex	7	16	F3-3
240	MC58	BennettTree07; Z7176	UK	1983		Neisseria meningitidis	B	74	ST-32 complex	7	16-2	F1-5
613	Z2491	BennettTree09; C751; NIBSC_2763; Z6244	The Gambia	1983	meningitis	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
638	G2136	NIBSC_2743; Z6411	UK [England]	1986	invasive (unspecified/other)	Neisseria meningitidis	B	8	ST-8 complex	5-2	10-1	F3-6
698	FAM18	BennettTree11; NIBSC_3076; Z4259	USA	1983	invasive (unspecified/other)	Neisseria meningitidis	C	11	ST-11 complex	5	2	F1-30

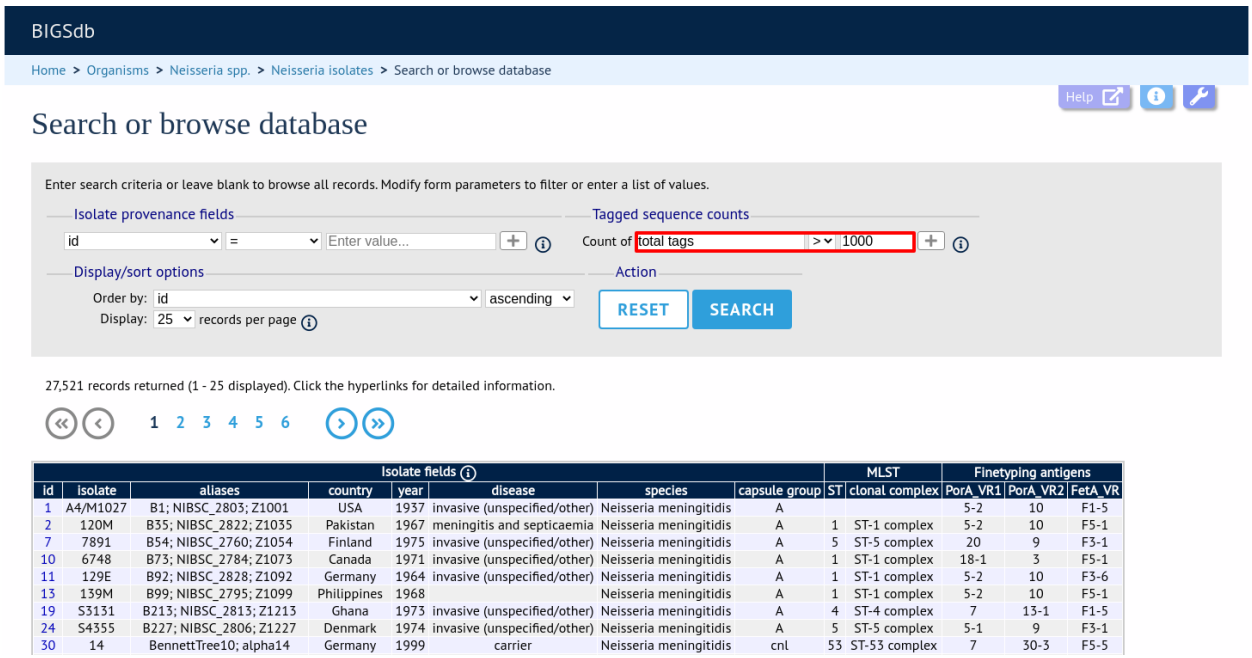
11.10.6 Query by sequence tag count

Queries can be combined with counts of the total number of tags or for individual loci.

Make sure that the tagged sequence counts fieldset is displayed by selecting it in the 'Modify form options' tab.



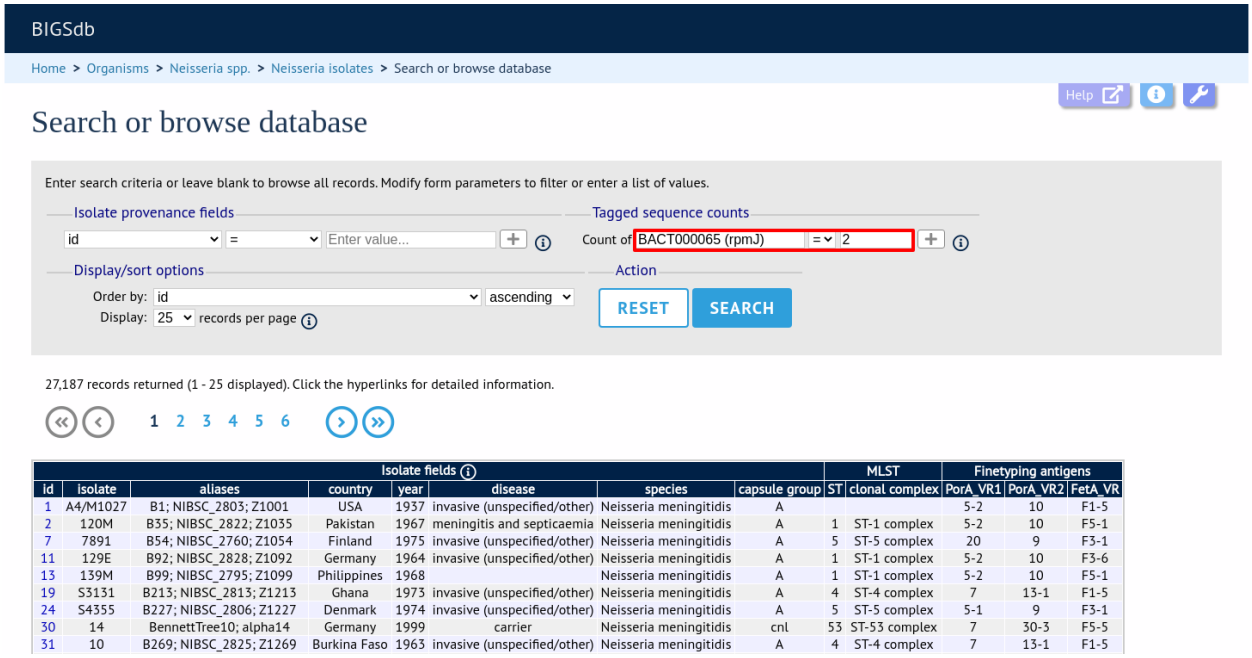
For example, to find all isolates that have sequence tags at >1000 loci, select 'total tags > 1000', then click 'Submit'.



Isolate fields								MLST		Finetyping antigens		
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex	PorA_VR1	PorA_VR2	FetA_VR
1	A4/M1027	B1; NIBSC_2803; Z1001	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-5
2	120M	B35; NIBSC_2822; Z1035	Pakistan	1967	meningitis and septicaemia	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
7	7891	B54; NIBSC_2760; Z1054	Finland	1975	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-1
10	6748	B73; NIBSC_2784; Z1073	Canada	1971	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	18-1	3	F5-1
11	129E	B92; NIBSC_2828; Z1092	Germany	1964	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F3-6
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
30	14	BennettTree10; alpha14	Germany	1999	carrier	Neisseria meningitidis	cnl	53	ST-53 complex	7	30-3	F5-5

You can also search for isolates where any isolate has a particular number of sequence tags. Use the term 'any locus' to do this.

Finally, you can search for isolates with a specific number of tags at a specific locus.



Isolate fields								MLST		Finetyping antigens		
id	isolate	aliases	country	year	disease	species	capsule group	ST	clonal complex	PorA_VR1	PorA_VR2	FetA_VR
1	A4/M1027	B1; NIBSC_2803; Z1001	USA	1937	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-5
2	120M	B35; NIBSC_2822; Z1035	Pakistan	1967	meningitis and septicaemia	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
7	7891	B54; NIBSC_2760; Z1054	Finland	1975	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-1
11	129E	B92; NIBSC_2828; Z1092	Germany	1964	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F3-6
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
30	14	BennettTree10; alpha14	Germany	1999	carrier	Neisseria meningitidis	cnl	53	ST-53 complex	7	30-3	F5-5
31	10	B269; NIBSC_2825; Z1269	Burkina Faso	1963	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5

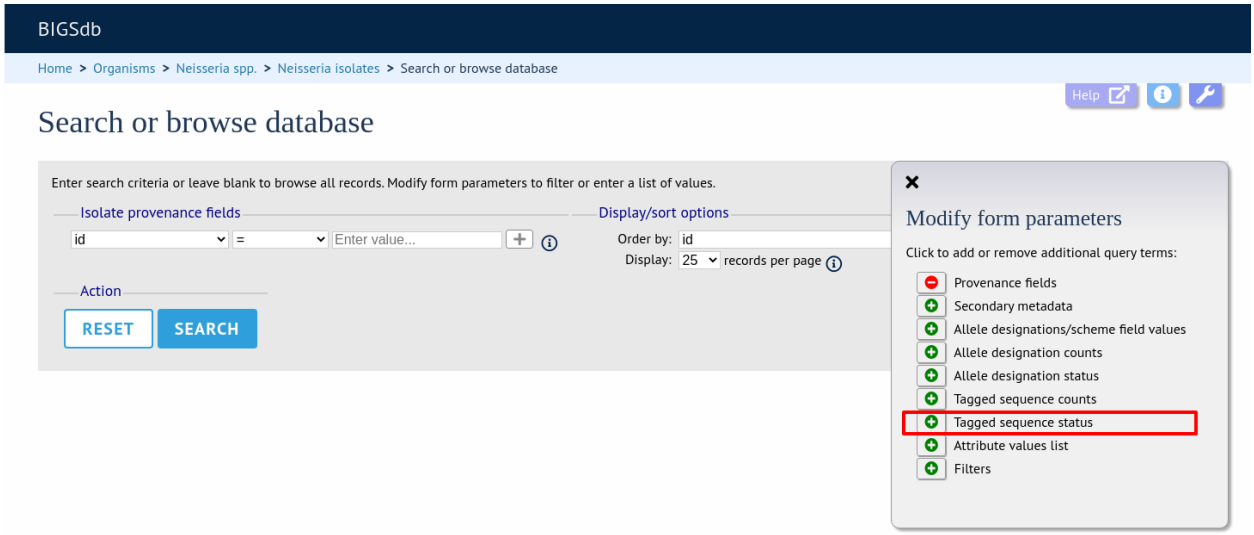
Additional search terms can be combined using the '+' button. Sequence tag count queries will be combined with terms entered in other sections.

Note: Searches for 'all loci' with counts that include zero, e.g. 'count of any locus = 0' or with a '<' operator are not supported. This is because such searches have to identify every isolate for which one or more loci are not tagged. In databases with thousands of loci this can be a very expensive database query.

11.10.7 Query by tagged sequence status

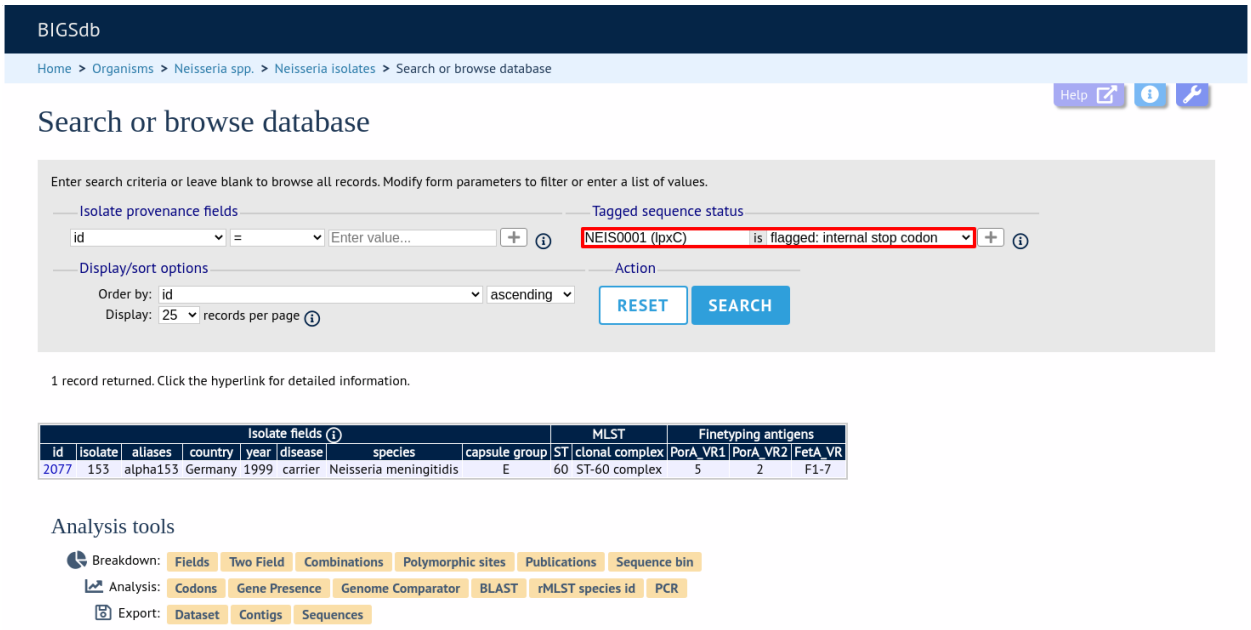
Sequence tags identify the region of a contig within an isolate's sequence bin entries that correspond to a particular locus. The presence or absence of these tags can be queried as can whether or not the sequence has an a flag associated with. These flags designate specific characteristics of the sequences. Queries will be combined from the values entered in all form sections.

Make sure that the tagged sequences status fieldset is displayed by selecting it in the 'Modify form options' tab.



Select a specific locus in the dropdown box (or alternatively 'any locus') and a status. Available status values are:

- untagged
  - The locus has not been tagged within the sequence bin.
- tagged
  - The locus has been tagged within the sequence bin.
- complete
  - The locus sequence is complete.
- incomplete
  - The locus sequence is incomplete - normally because it continues beyond the end of a contig.
- flagged: any
  - The sequence for the locus has a flag set.
- flagged: none
  - The sequence for the locus does not have a flag set.
- flagged: <specific flag>
  - The sequence for the locus has the specific flag chosen.



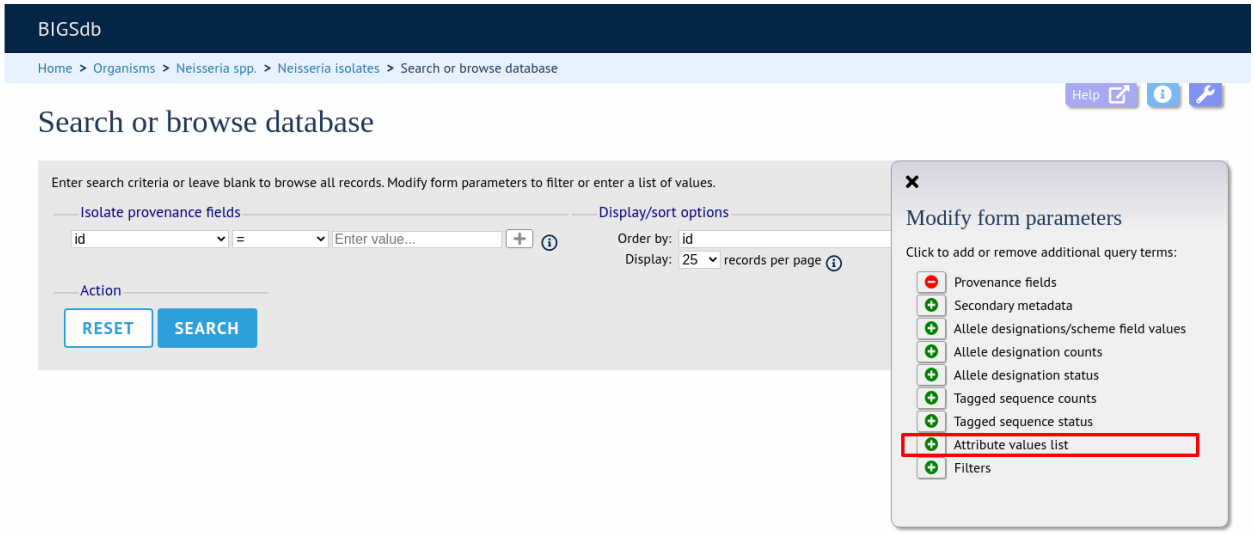
See also:

Sequence tag flags

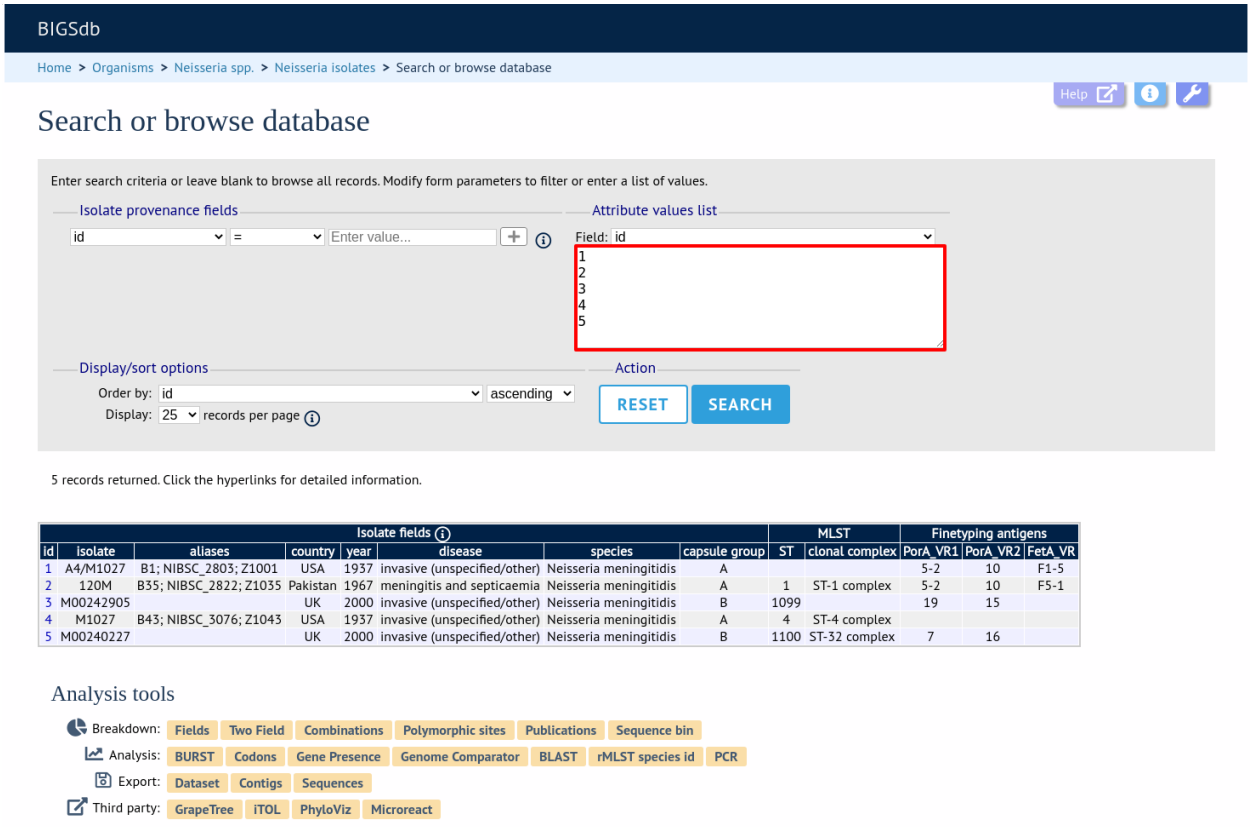
11.10.8 Query by list of attributes

The query form can be modified with a list box in to which a list of values for a chosen attribute can be entered - this could be a list of ids, isolate names, alleles or scheme fields. This list will be combined with any other criteria or filter used on the page.

If the list box is not shown, add it by selecting it in the 'Modify form options' tab.



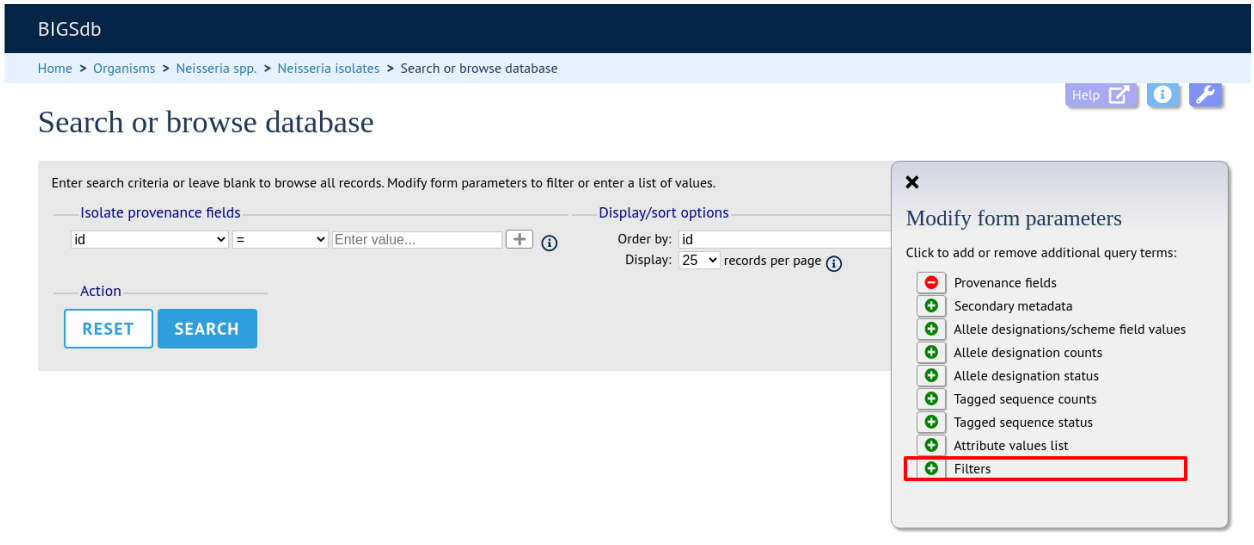
Select the attribute to query and enter a list of values.



11.10.9 Query filters

There are various filters that can additionally be applied to queries, or the filters can be applied solely on their own so that they filter the entire database.

Make sure that the filters fieldset is displayed by selecting it in the 'Modify form options' tab.



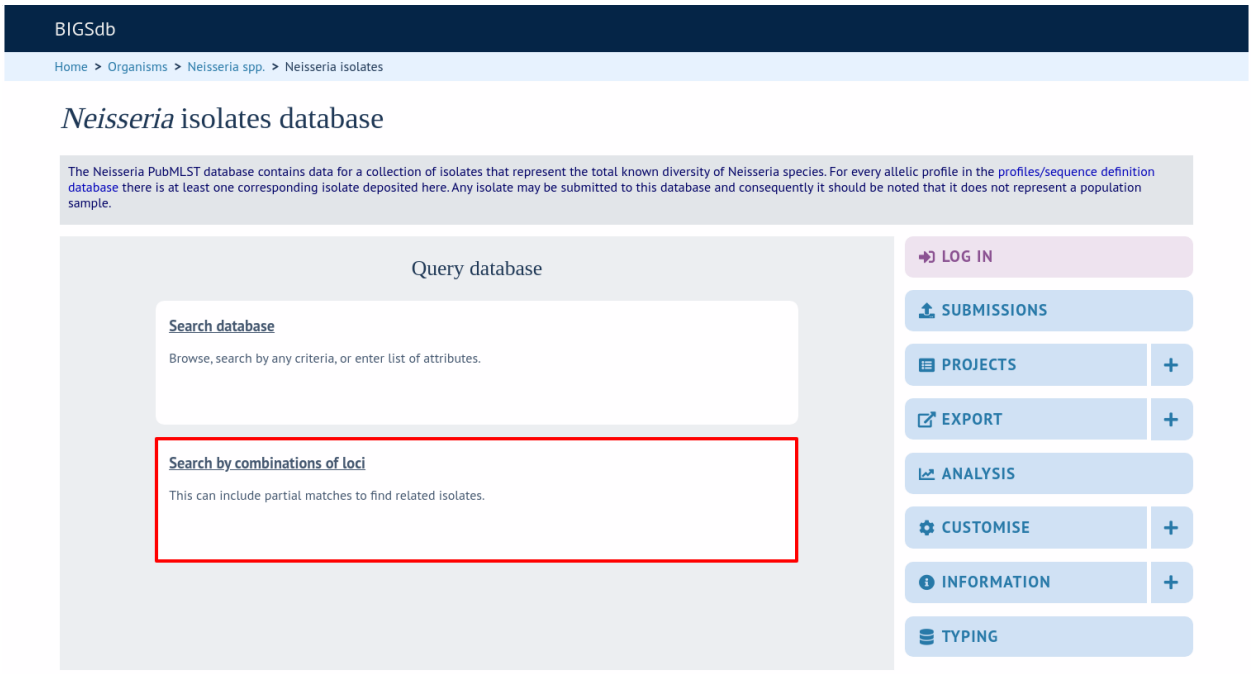
The filters displayed will depend on the database and what has been defined within it. Common filters are:

- Publication - Select one or more publication that has been linked to isolate records.
- Project - Select one or more project that isolates belong to.
- Profile completion - This is commonly displayed for MLST schemes. Available options are:
  - complete - All loci of the scheme have alleles designated.
  - incomplete - One or more loci have not yet been designated.
  - partial - The scheme is incomplete, but at least one locus has an allele designated.
  - started - At least one locus has an allele designated. The scheme mat be complete or partial.
  - not started - The scheme has no loci with alleles designated.
- Provenance fields - Dropdown list boxes of values for specific provenance fields may be present if set for the database. Users can choose to *add additional filters*.
- Old record versions - Checkbox which, if selected, will include all record versions in a query.

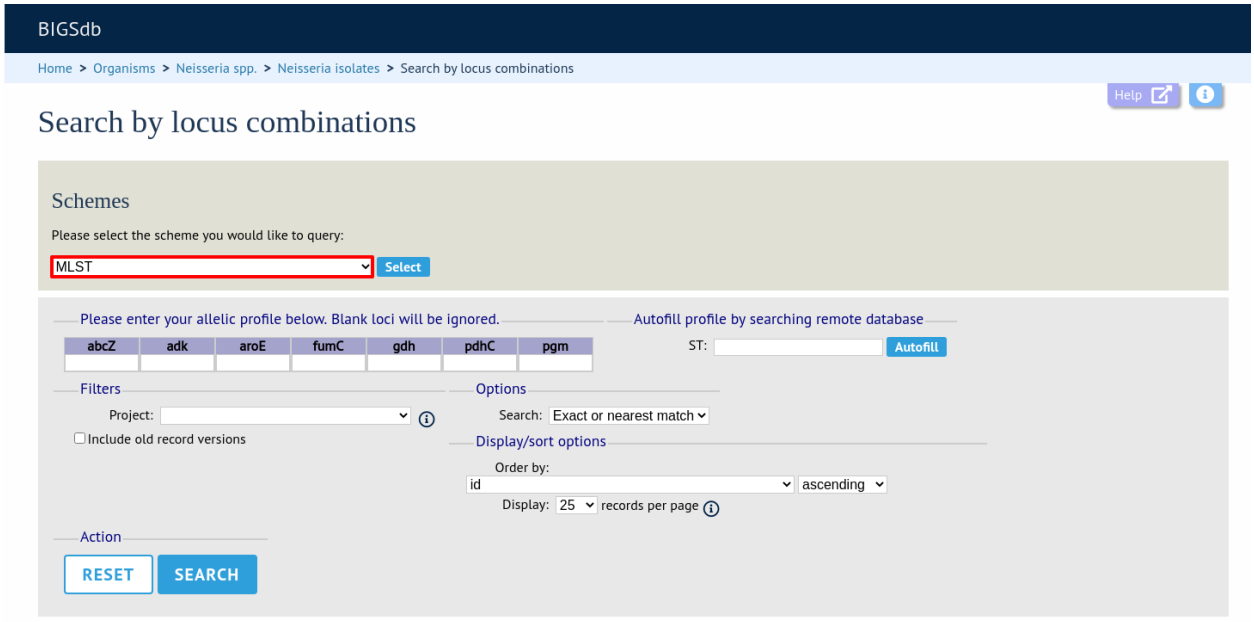
11.10.10 Querying by allelic profile

If a scheme, such as MLST, has been defined for an isolate database it is possible to query the database against complete or partial allelic profiles. Even if no scheme is defined, queries can be made against all loci.

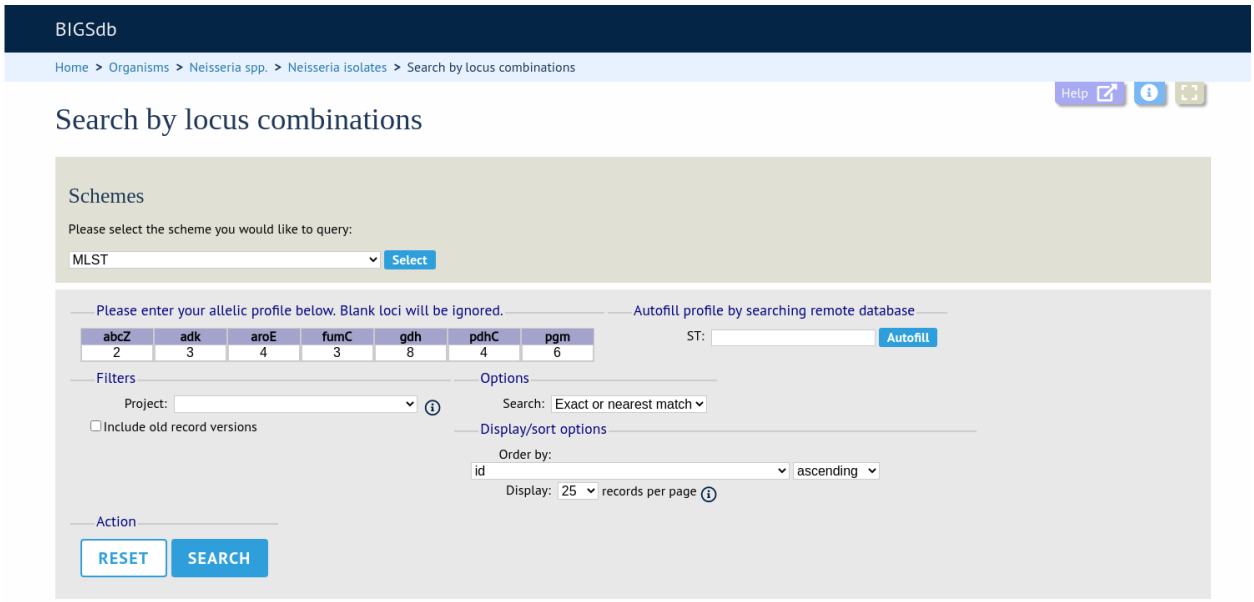
On the index page, click 'Search by combinations of loci (profiles)' for any defined scheme. Enter either a partial (any combination of loci) or complete profile.



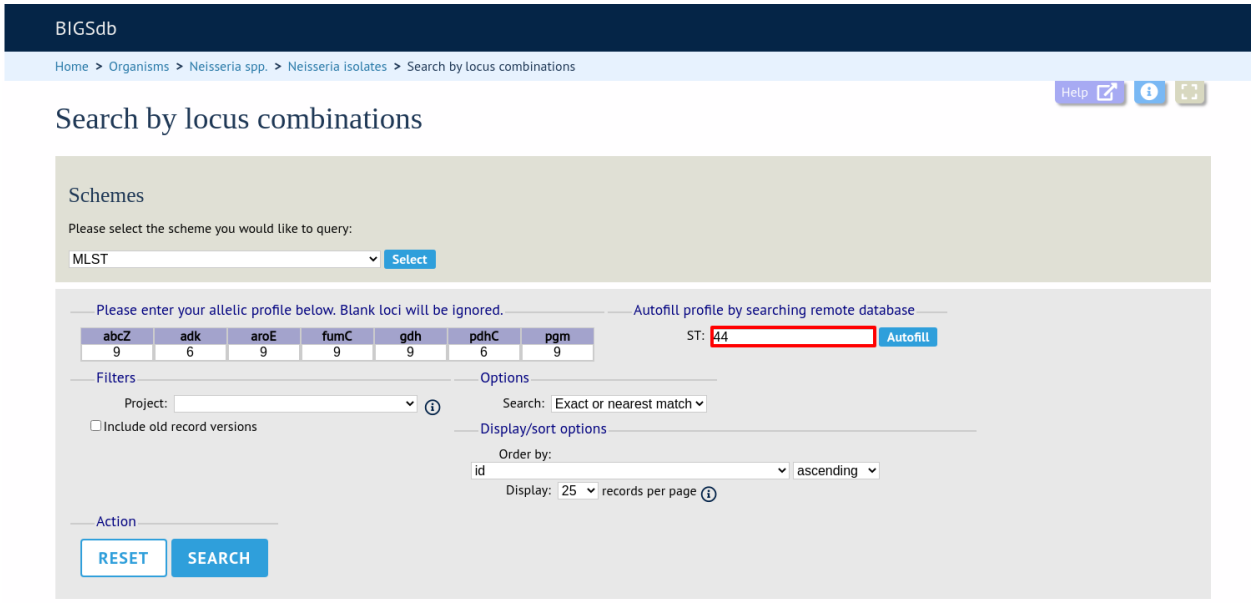
If multiple schemes are defined, you may have to select the scheme you wish to query in the 'Schemes' dropdown box and click 'Select'.



Enter the combination of alleles that you want to query for. Fields can be left blank.



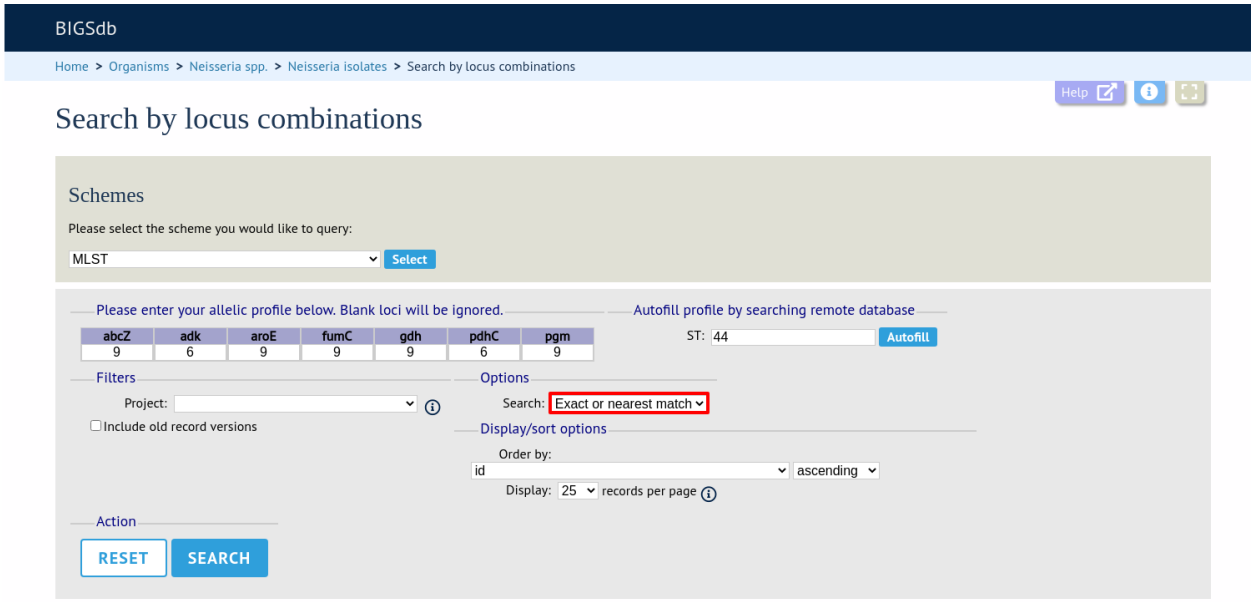
Alternatively, for scheme profiles, you can enter a primary key value (e.g. ST) and select 'Autofill' to automatically fill in the associated profile.



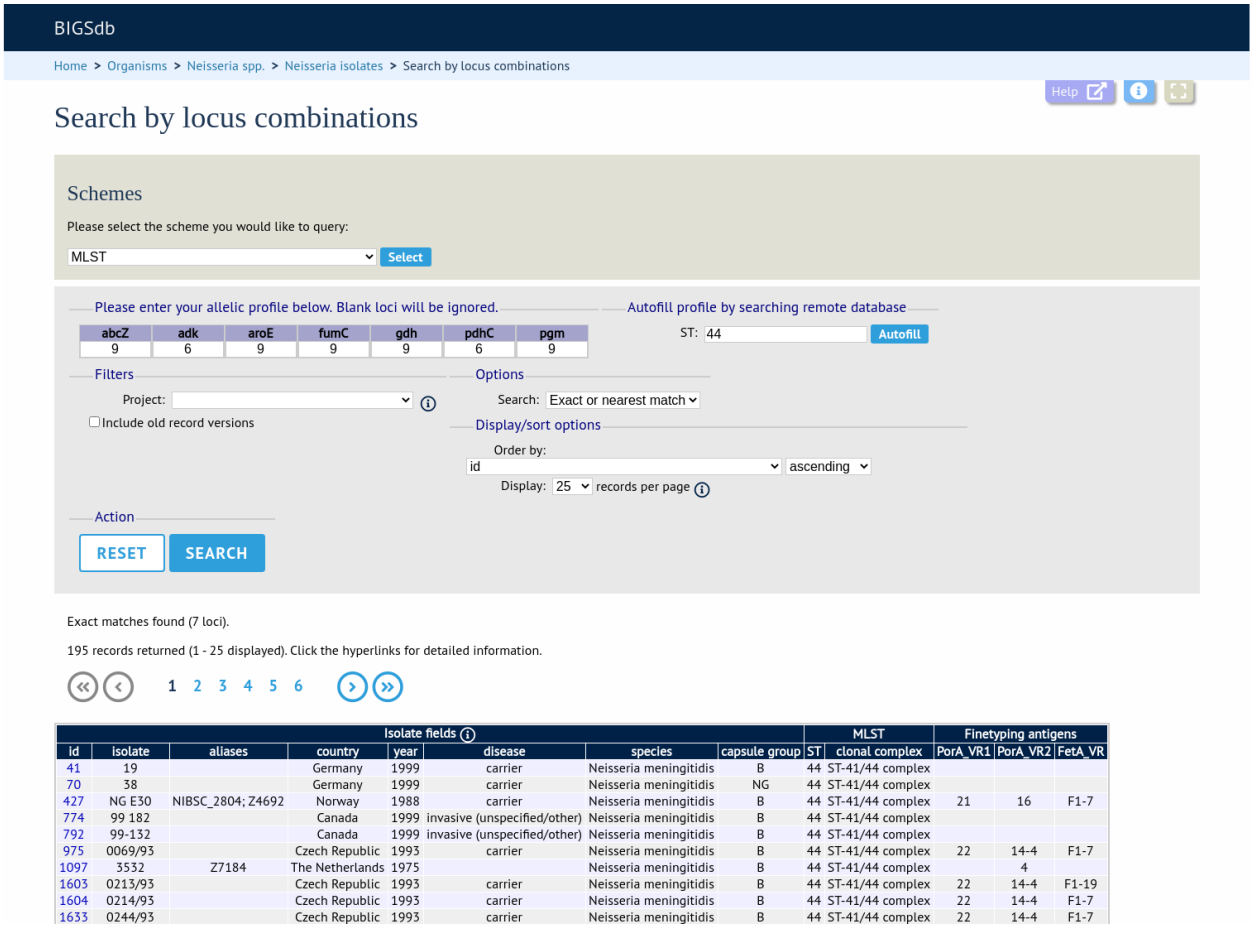
Select the number of loci that you'd like to match in the options dropdown box. Available options are:

- Exact or nearest match
- Exact match only
- x or more matches
- y or more matches
- z or more matches

Where x,y, and z will range from n-1 to 1 where n is the number of loci in the scheme.



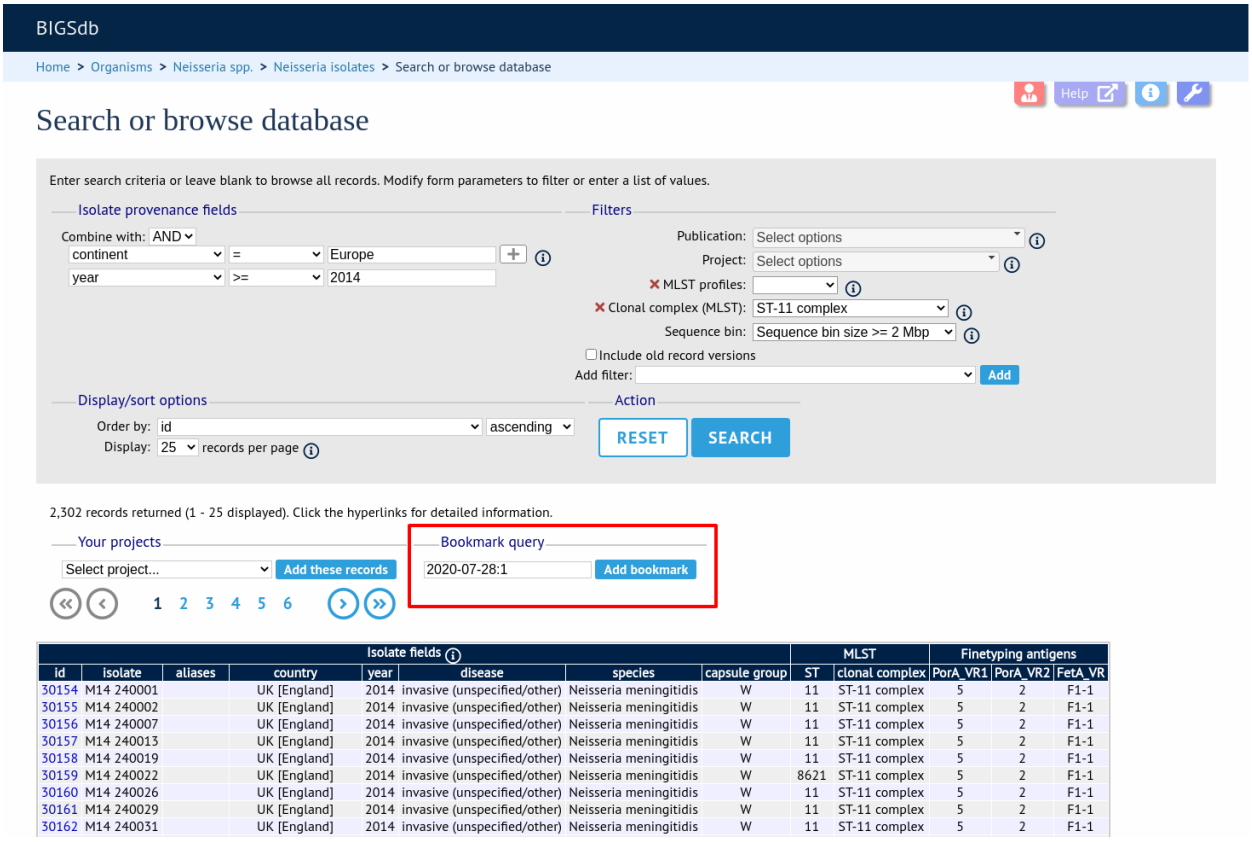
Click 'Search'.



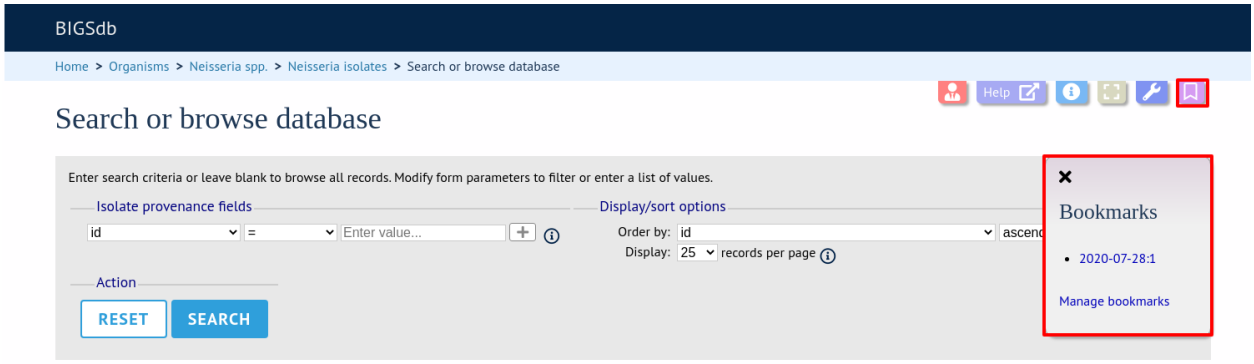
11.11 Bookmarking an isolate query

Once you have made an isolate database query, you can bookmark it so that it can be repeated in the future. You need to have an account and be logged in to the database to be able to bookmark.

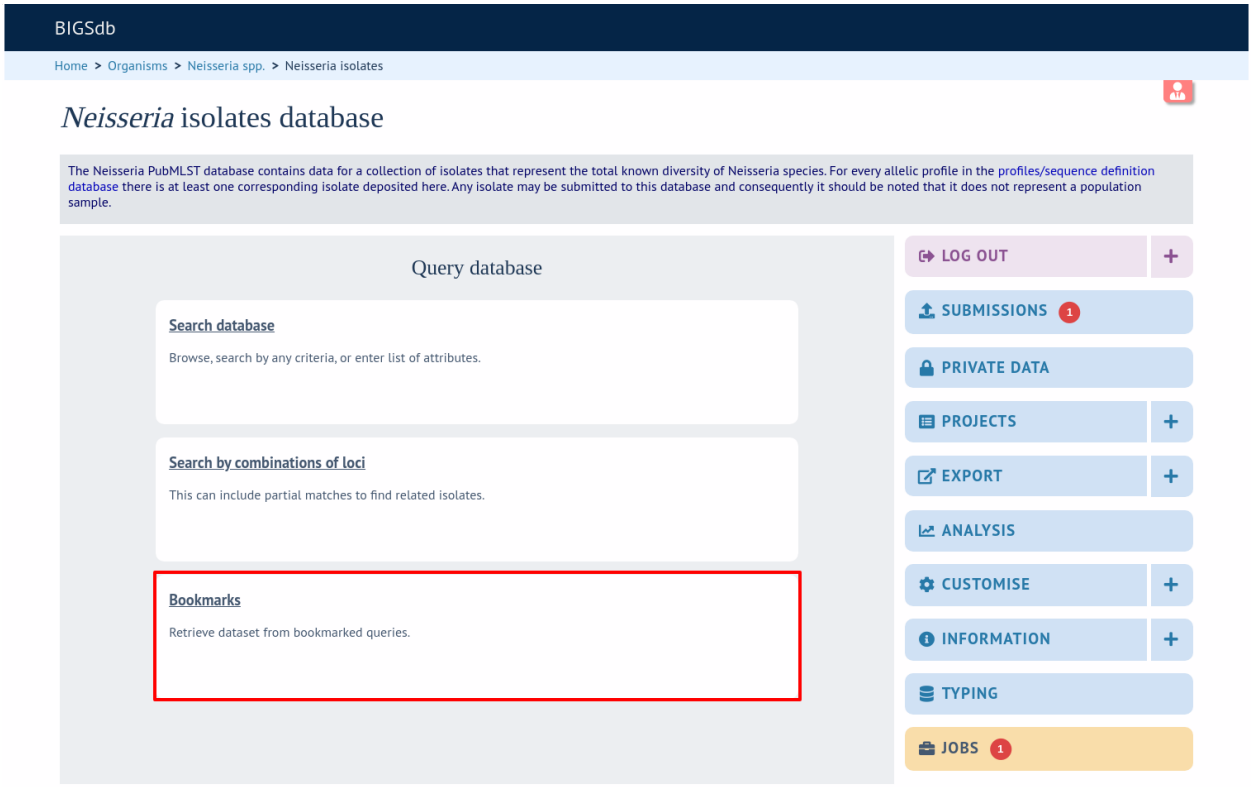
Following a query, there will be a 'Bookmark query' section in the results header section. Enter a name for the bookmark (a default name based on the date will be shown) and click 'Add bookmark'.



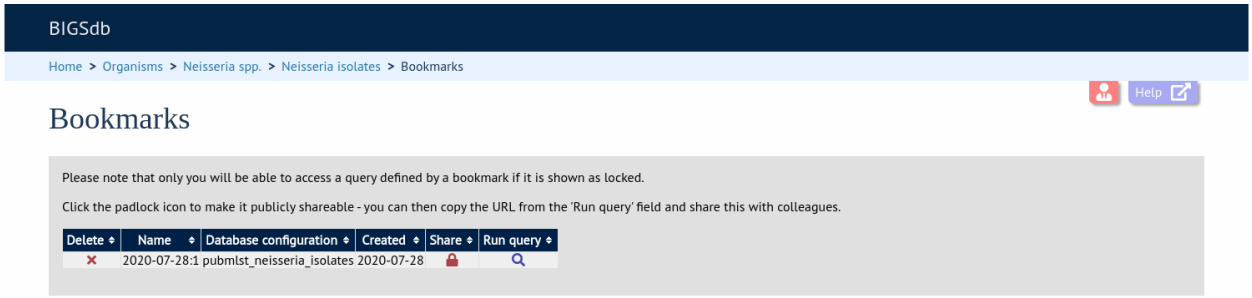
A new top-right tab will appear when you have saved bookmarks. You can click on this to easily access all saved bookmarks.



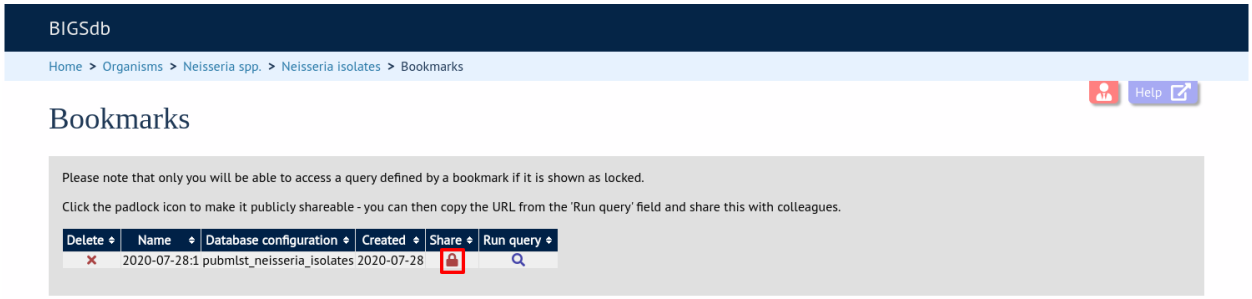
You can also access your bookmarks from the main contents page. A link will appear in the query section once you have saved a bookmark.



This will take you to a page where you can manage your bookmarks.



You can go to a bookmarked search by clicking on the 'Run query' icon. By default, a bookmark can only be used by the logged-in user who created it. This is for privacy reasons to prevent other users from finding out what terms are being used for a search. If, however, you wish to share the URL to the query, you can make it shareable by clicking the padlock icon.

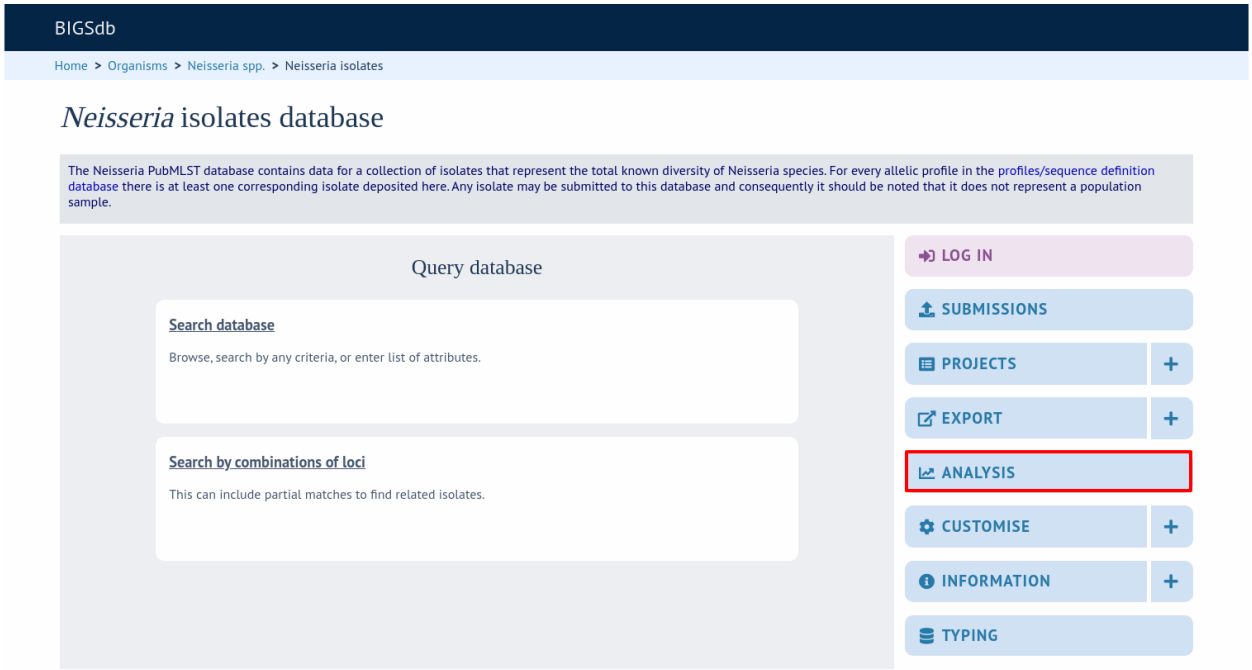


The icon will change to a green open padlock. You can right-click on the 'Run query' link to copy the URL if you wish to share it with others.

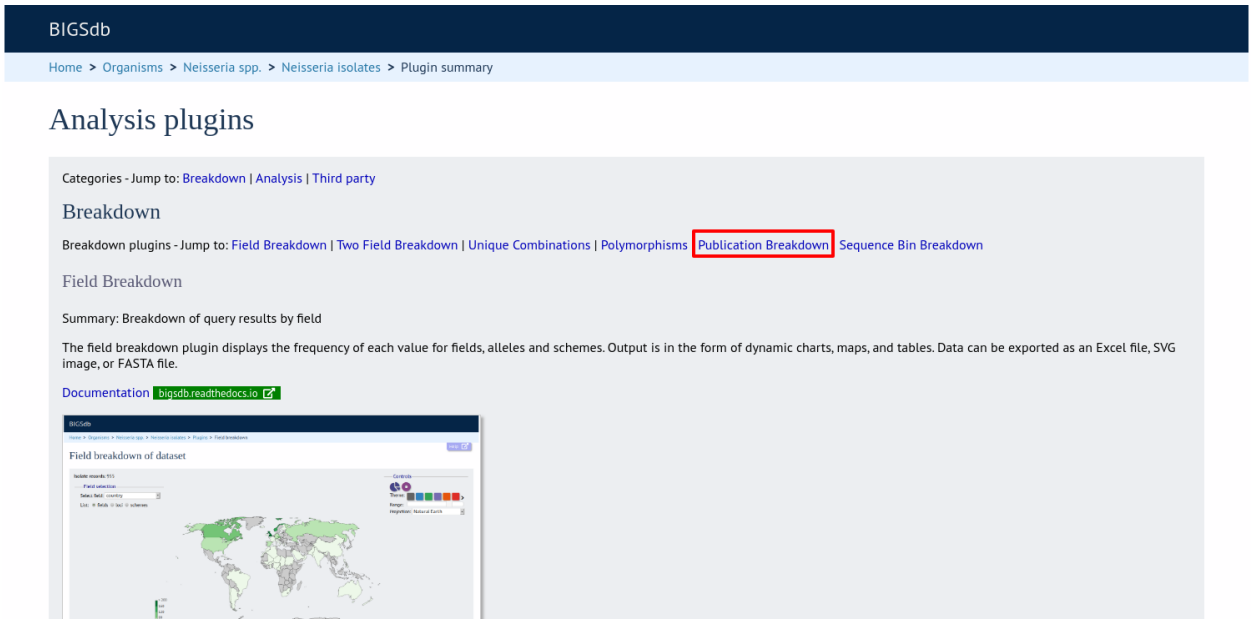
Bookmarks can be deleted by clicking on the delete icon.

11.12 Retrieving isolates by linked publication

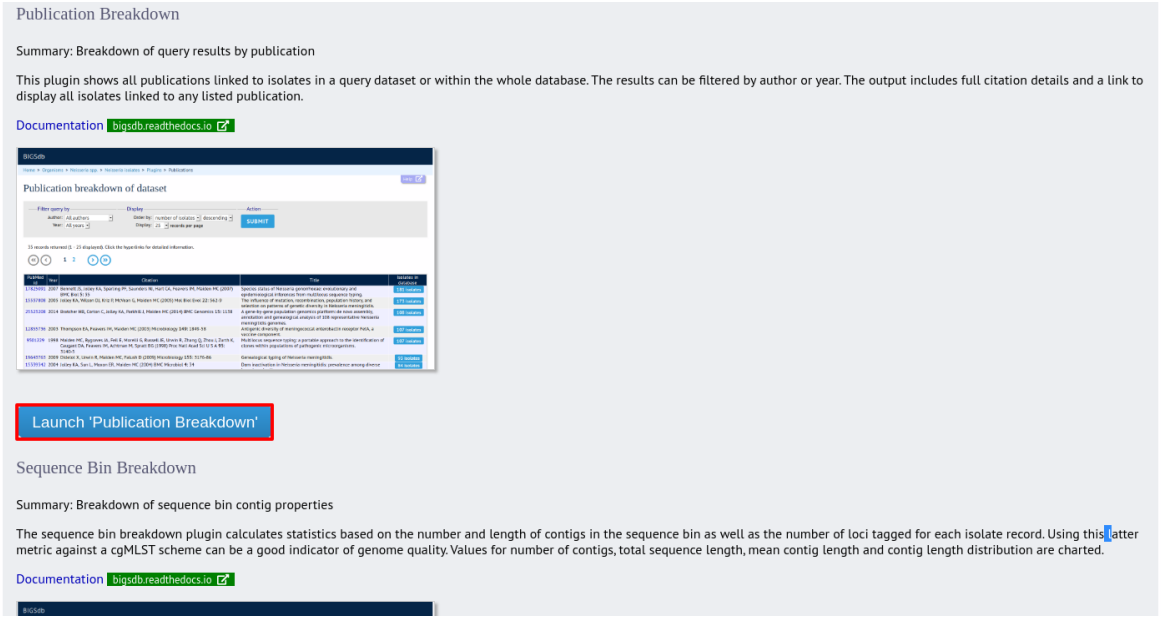
Click 'Analysis' on the contents page.



Click 'Publication breakdown'



Then click 'Launch Publication Breakdown'.

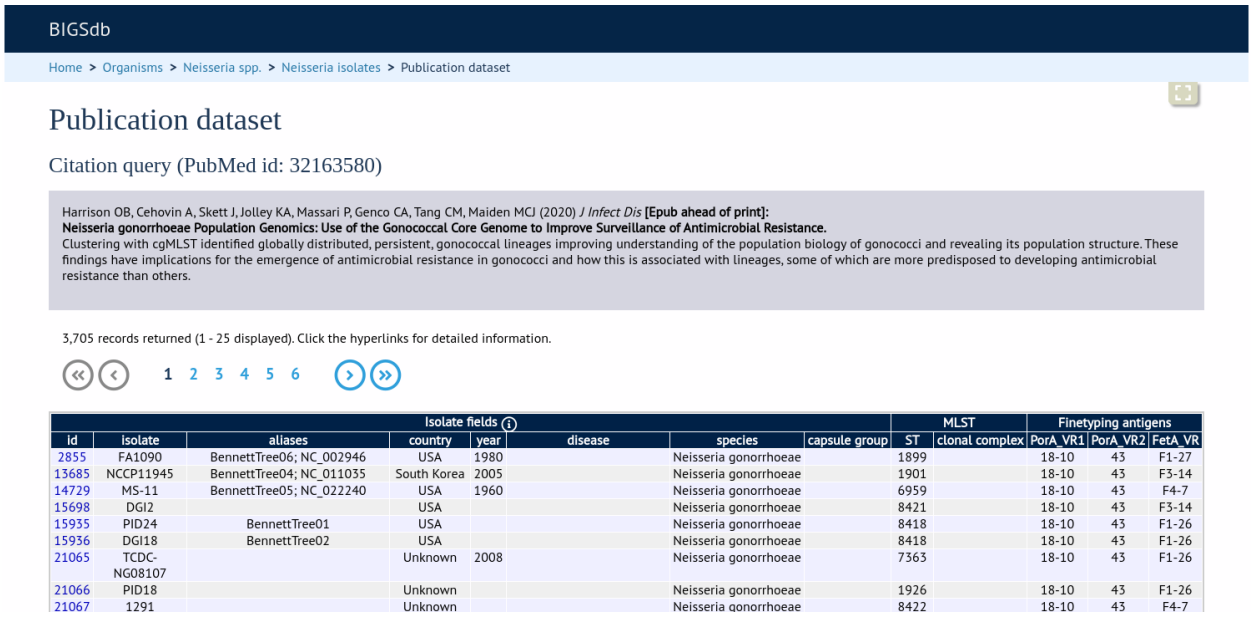


A list of publications linked by isolates within the database will be displayed.

These can be filtered by author and/or year, and the sort order changed.

To display the isolate records for any of the displayed publications, click the button to the right of the citation.

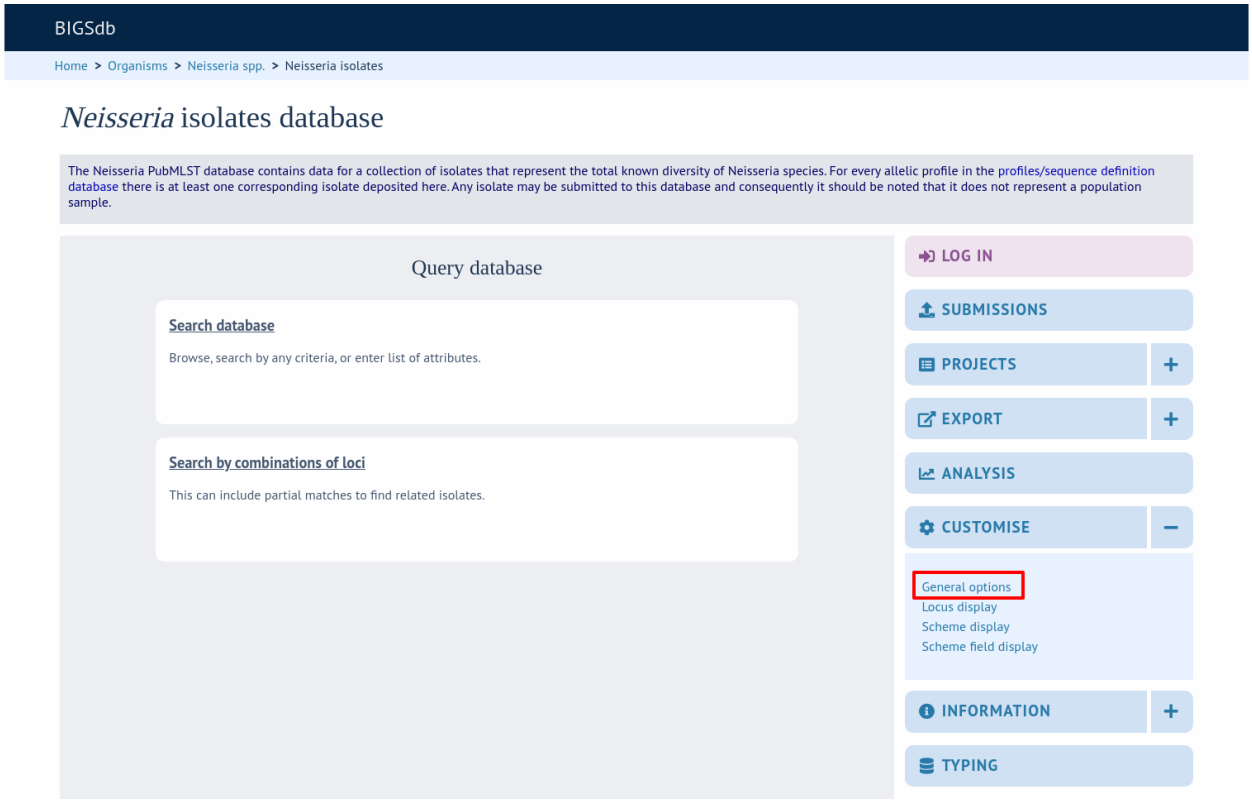
The abstract of the paper will be displayed (if available), along with all isolates linked to it.



11.13 User-configurable options

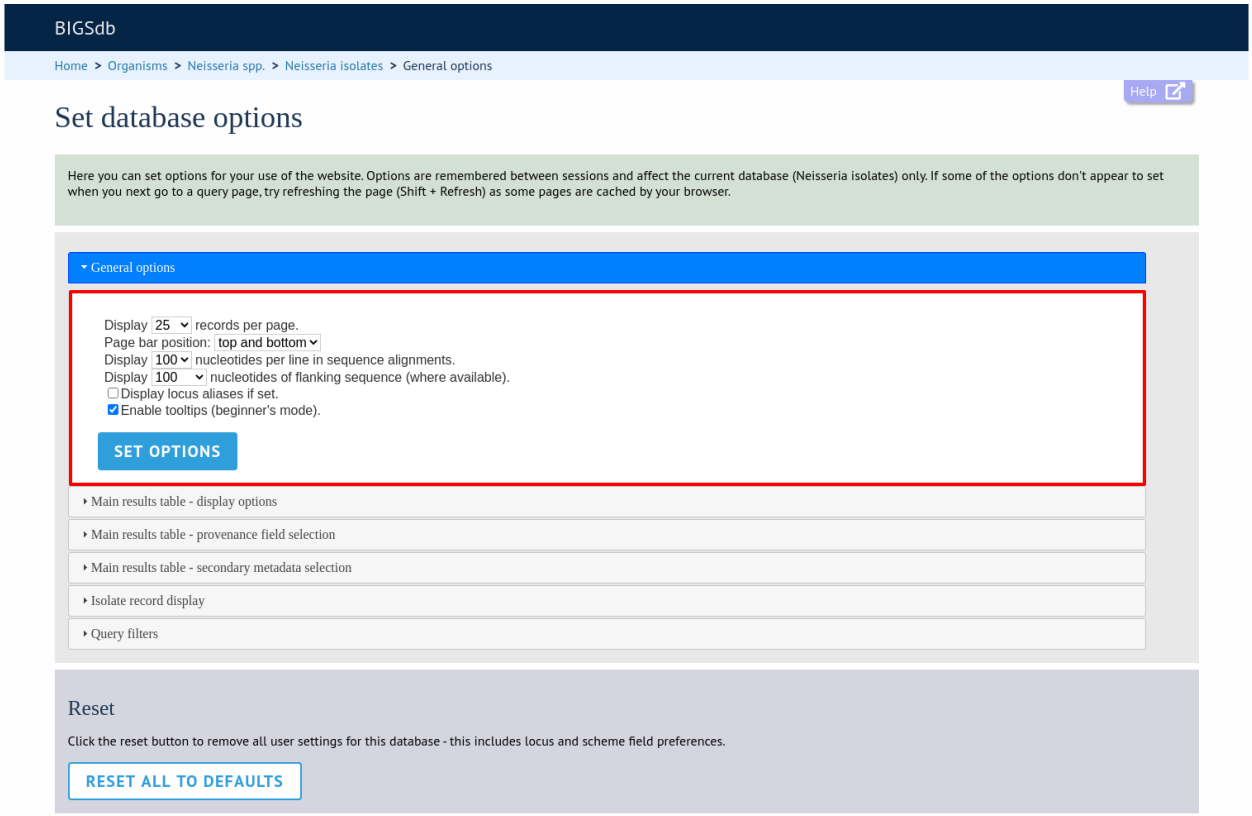
The BIGSdb user interface is configurable in a number of ways. Choices made are remembered between sessions. If the database requires you to log on, the options are associated with your user account, whereas if it is a public database, that you haven't logged in to, the options are associated with a browser cookie so they will be remembered if you connect from the same computer (using the same browser).

Most options are set by clicking the 'Customise' link on the database contents page. Most of the available options are visible for isolate databases, whereas sequence definition databases have fewer available.



11.13.1 General options

The general options tab is displayed by default. If another tab is being shown, click the 'General options' header.



The general tab allows the following options to be modified:

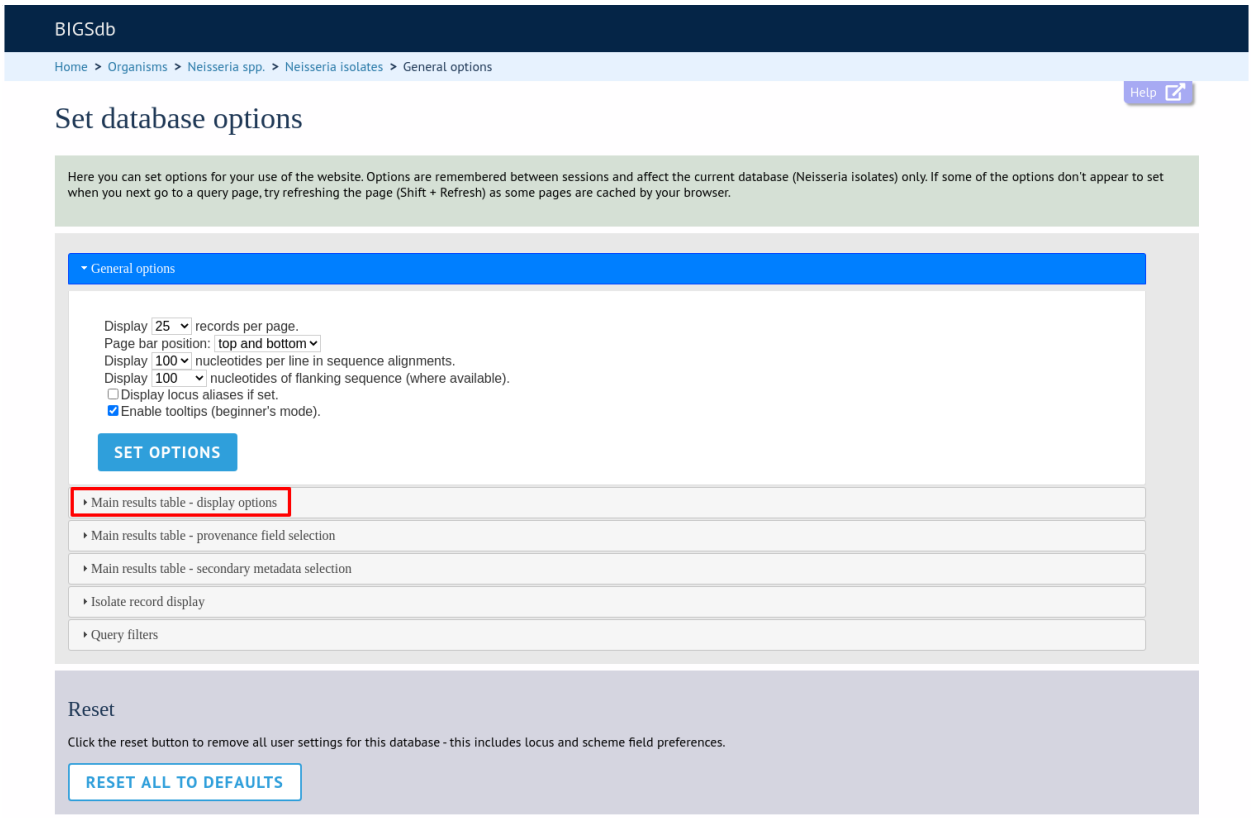
- Records per page
- Page bar position
- Nucleotides per line - Some analyses display sequence alignments. This option allows you to set the width of these alignments so suit your display.
- Flanking sequence length - This sets the length of flanking sequence upstream and downstream of a particular locus that is included whenever a sequence is displayed. Flanking sequences are displayed fainter that the locus sequence.
- Locus aliases - Loci can have multiple names (aliases). Setting this option will display all alternative names in results tables.
- Tooltips (beginner's mode) - Most query forms have help available in the form of information tooltips. These can be switched on/off here. They can also be toggled off by clicking the Toggle: 'i' button at the top-right of the display of some pages.

Click 'Set options' to remember any changes you make.

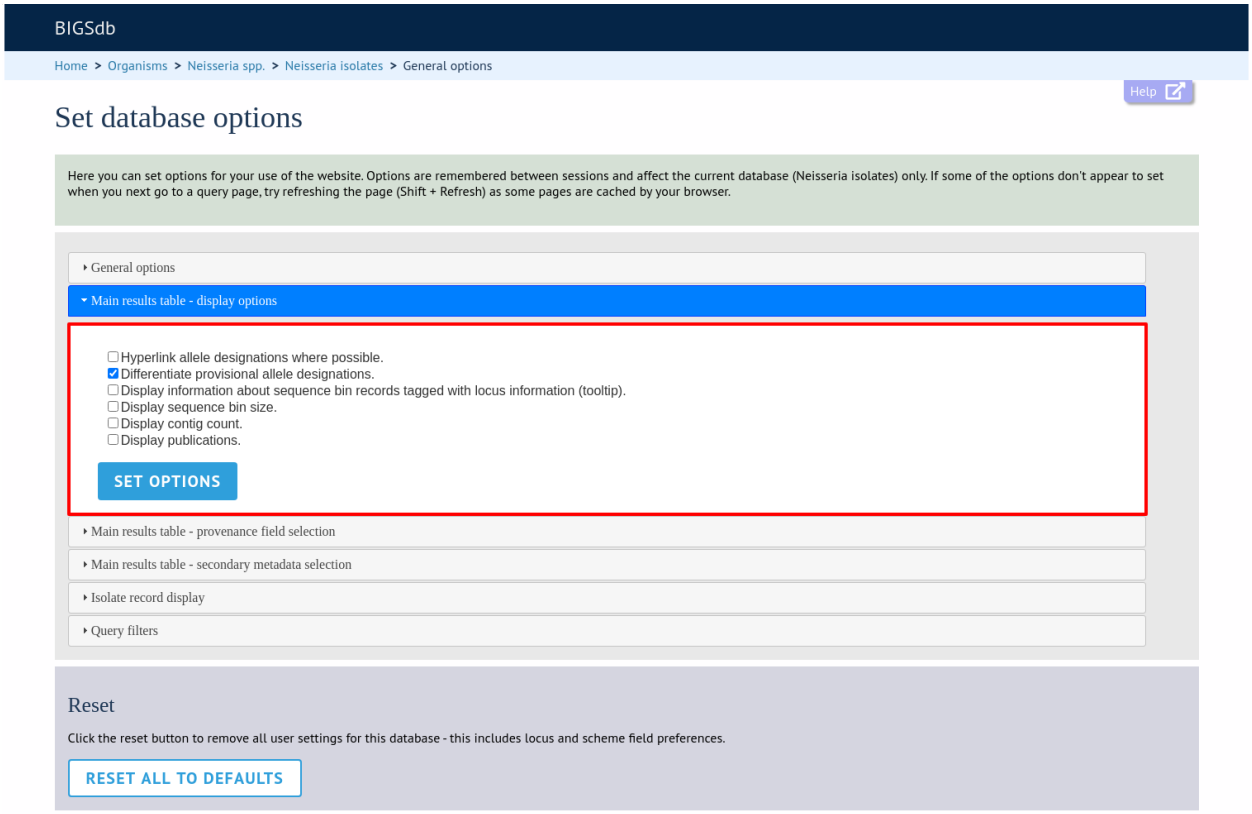
11.13.2 Main results table

The 'main results table' tab contains options for the display of paged results following a query.

Click the 'Main results table' header to display the tab.



The 'main results table' tab will scroll up.



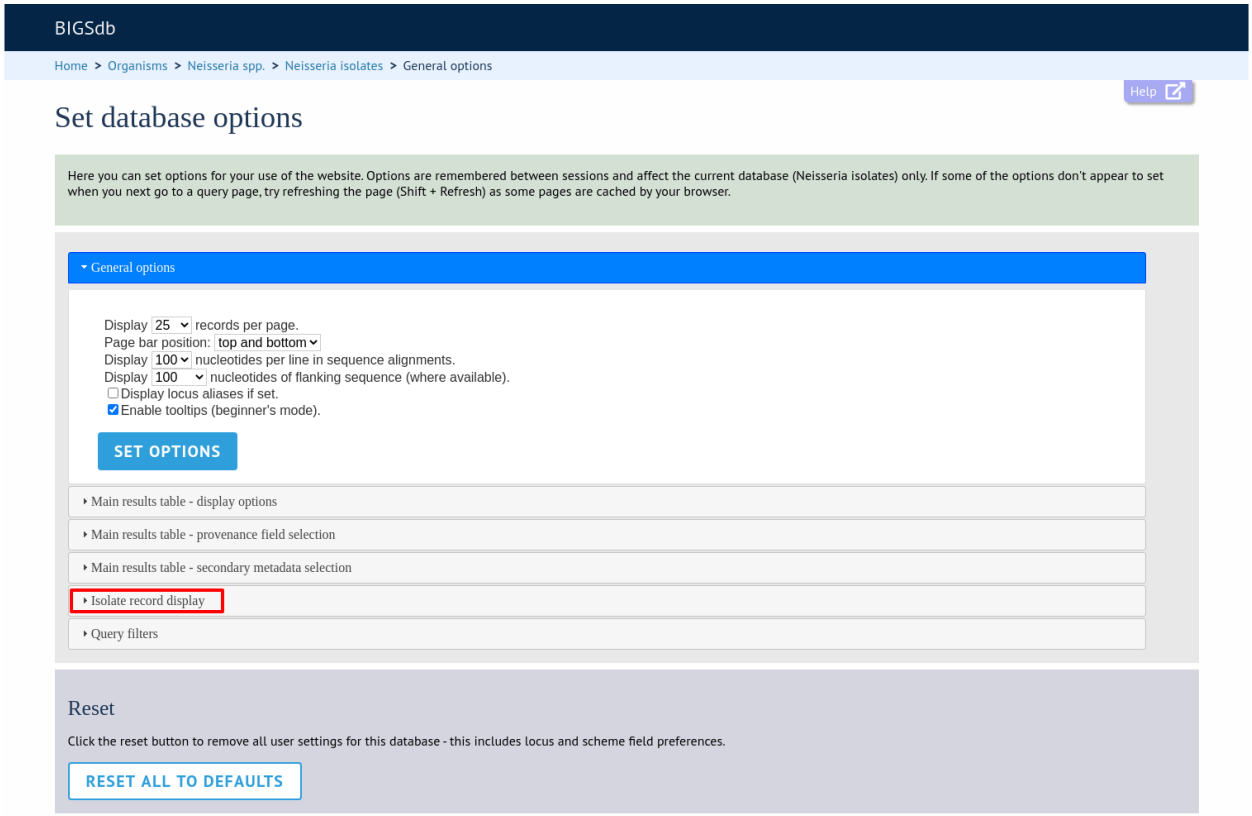
This tab allows the following options to be modified:

- Hyperlink allele designations - Hyperlinks point to an information page about the particular allele definition. Depending on the locus, these may exist on a different website.
- Differentiate provisional allele designations - Allele designations can be set as confirmed or provisional, usually depending on the method of assignment. Selecting this option will display provisional designations in a different colour to confirmed designations.
- Information about sequence bin records - Creates a tooltip that displays details about sequence tags corresponding to a locus.
- Sequence bin records - Displays a tooltip linking to the sequence tag if available.
- Sequence bin size - Displays the size of the sum of all contigs associated with each isolate record.
- Contig count - Displays the number of contigs associated with each isolate record.
- Publications - Displays citations with links to PubMed for each record.

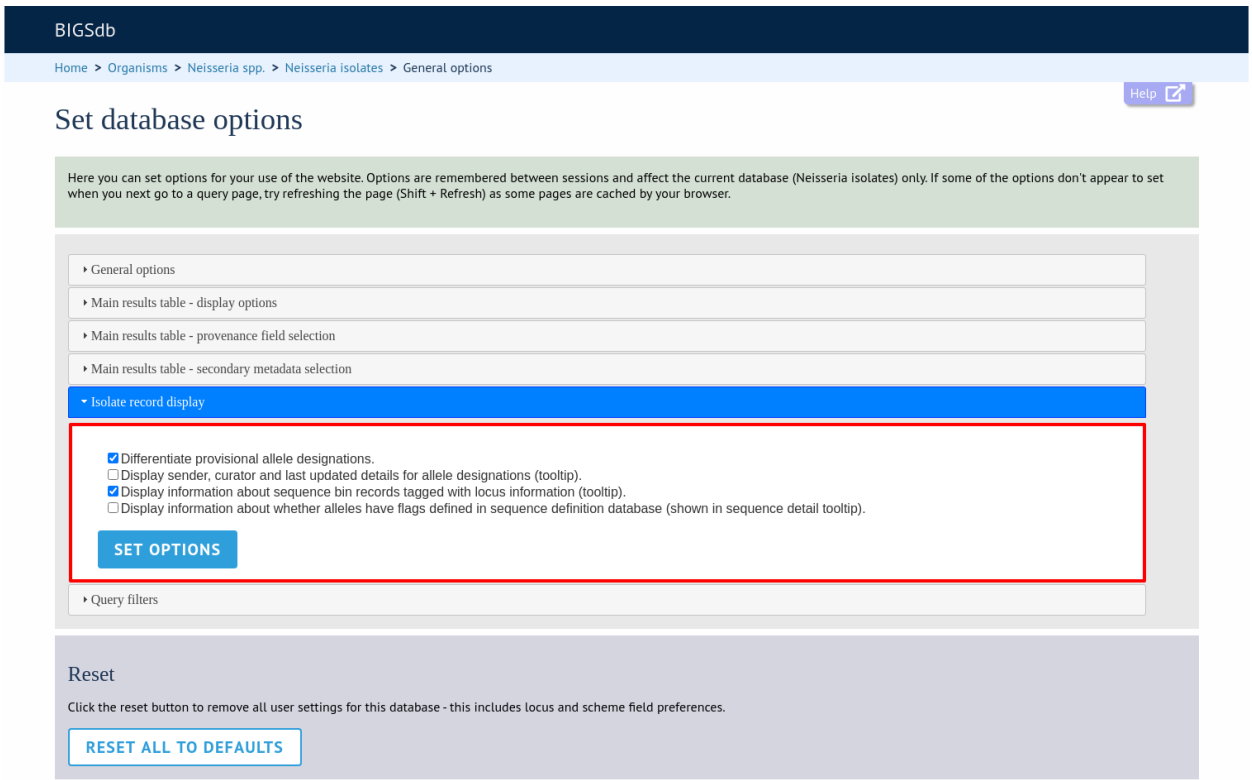
11.13.3 Isolate record display

The 'isolate record display' tab contains options for the display of a full isolate record.

Click the 'Isolate record display' tab to display the tab.



The 'Isolate record display' tab will scroll up.



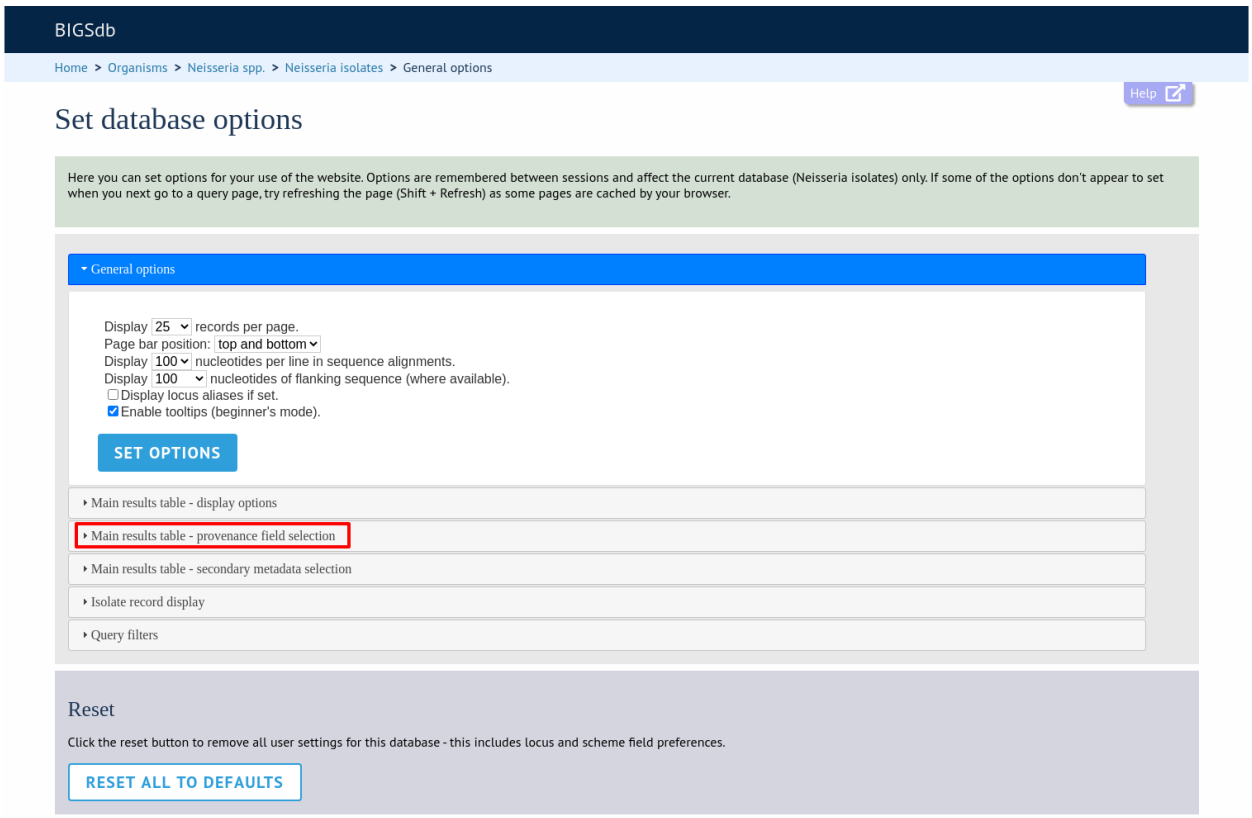
This tab allows the following options to be modified:

- Differentiate provisional allele designations - Allele designations can be set as confirmed or provisional, usually depending on the method of assignment. Selecting this option will display provisional designations in a different colour to confirmed designations.
- Display sender, curator and last updated records - Displays a tooltip containing sender information next to each allele designation.
- Sequence bin information - Displays a tooltip with information about the position of the sequence if tagged within the sequence bin.
- Allele flags - Displays information about whether alleles have flags defined in sequence definition databases.

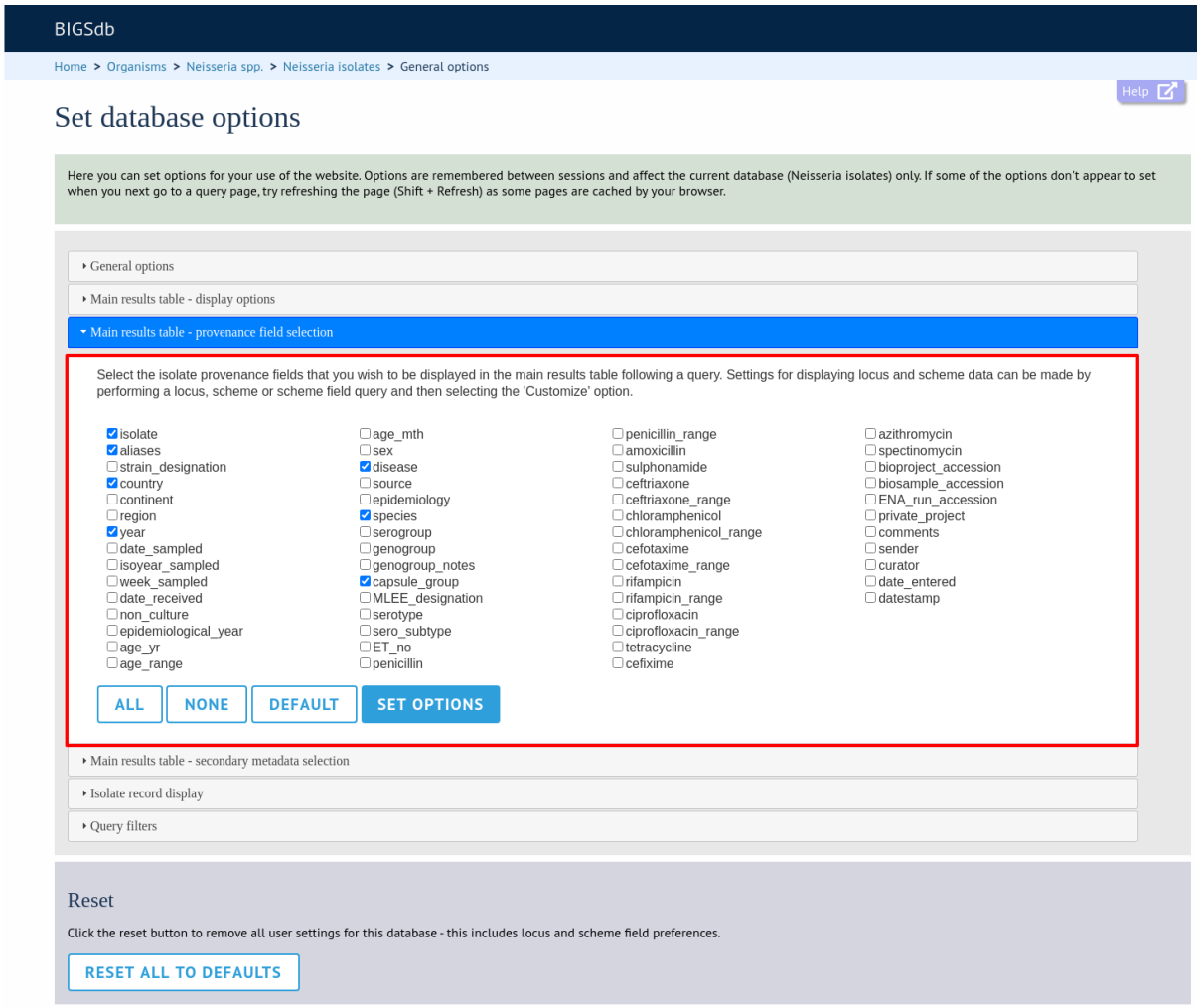
11.13.4 Provenance field display

The 'provenance field display' tab contains checkboxes for fields to display in the main results table.

Click the 'Provenance field display' tab to display the tab.



The 'Provenance field display' tab will scroll up.



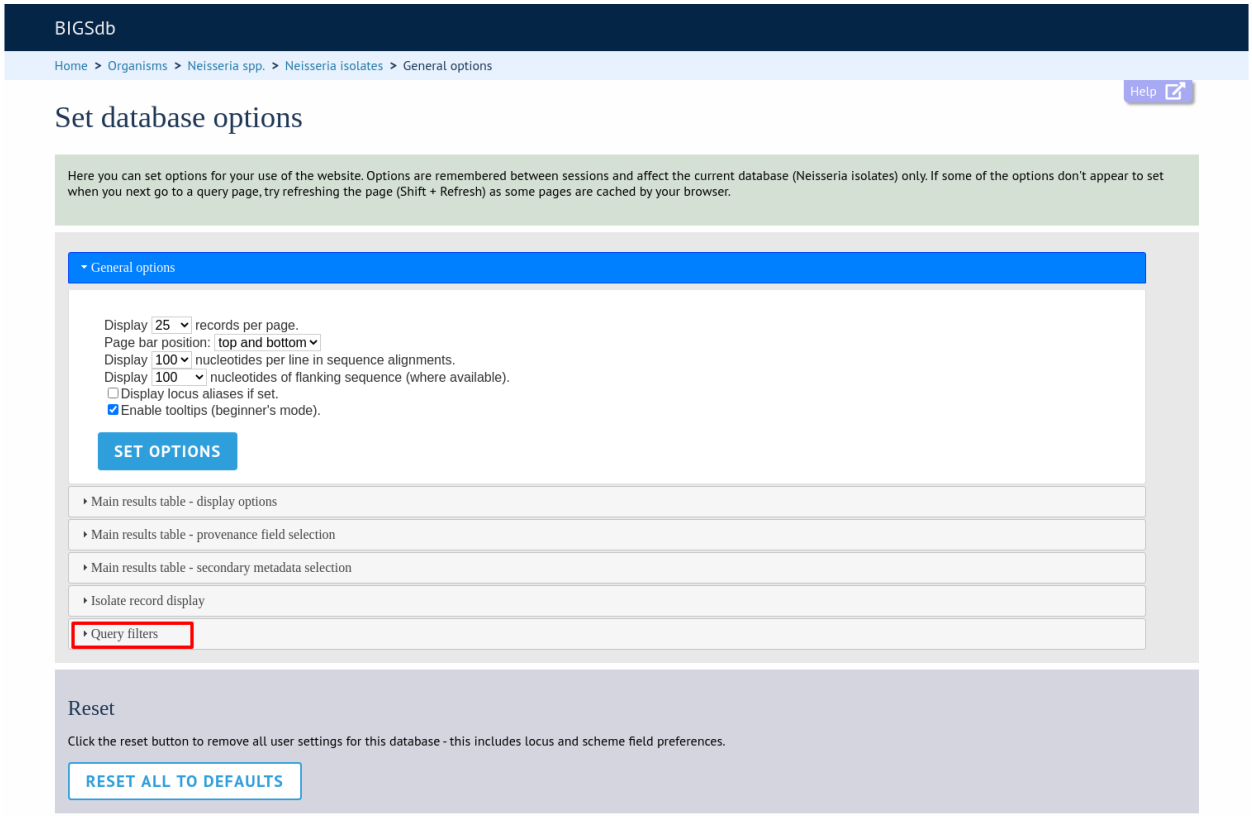
Some fields will be checked by default - these are defined during *database setup* (maindisplay option).

Check any fields that you wish to be displayed and then click 'Set options'. You can return to the default selection by clicking 'Default' followed by 'Set options'.

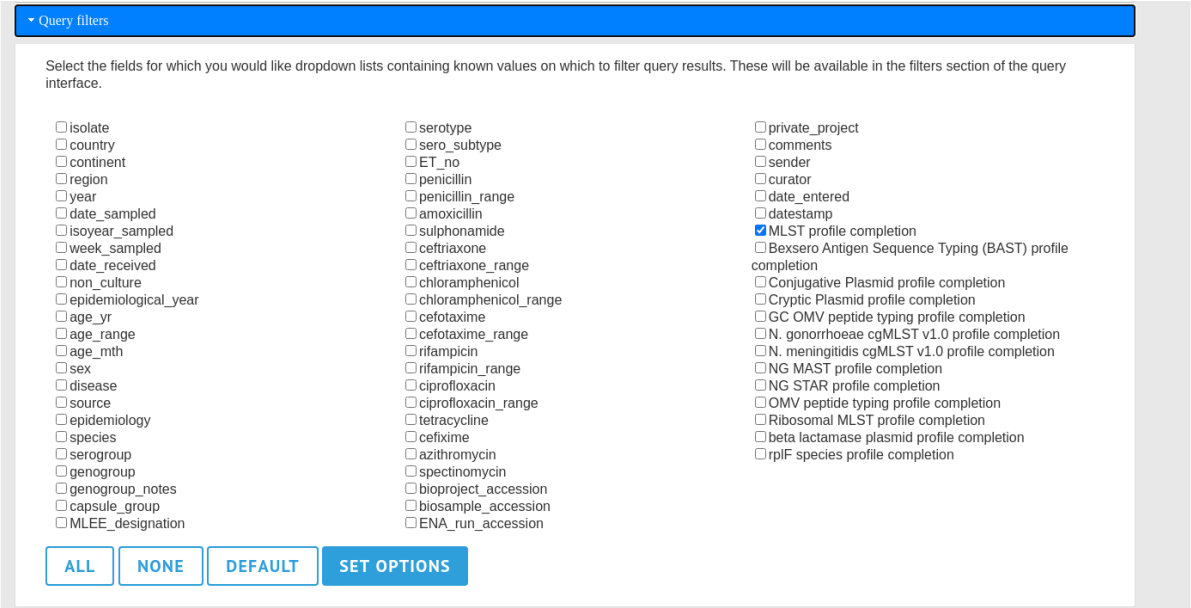
11.13.5 Query filters

The 'query filters' tab contains checkboxes for provenance fields and scheme completion status. Checking these results in drop-down list box filters appearing in the query page *filters fieldset*.

Click the 'Query filters' tab to display the tab.



The 'Query filters' tab will scroll up.



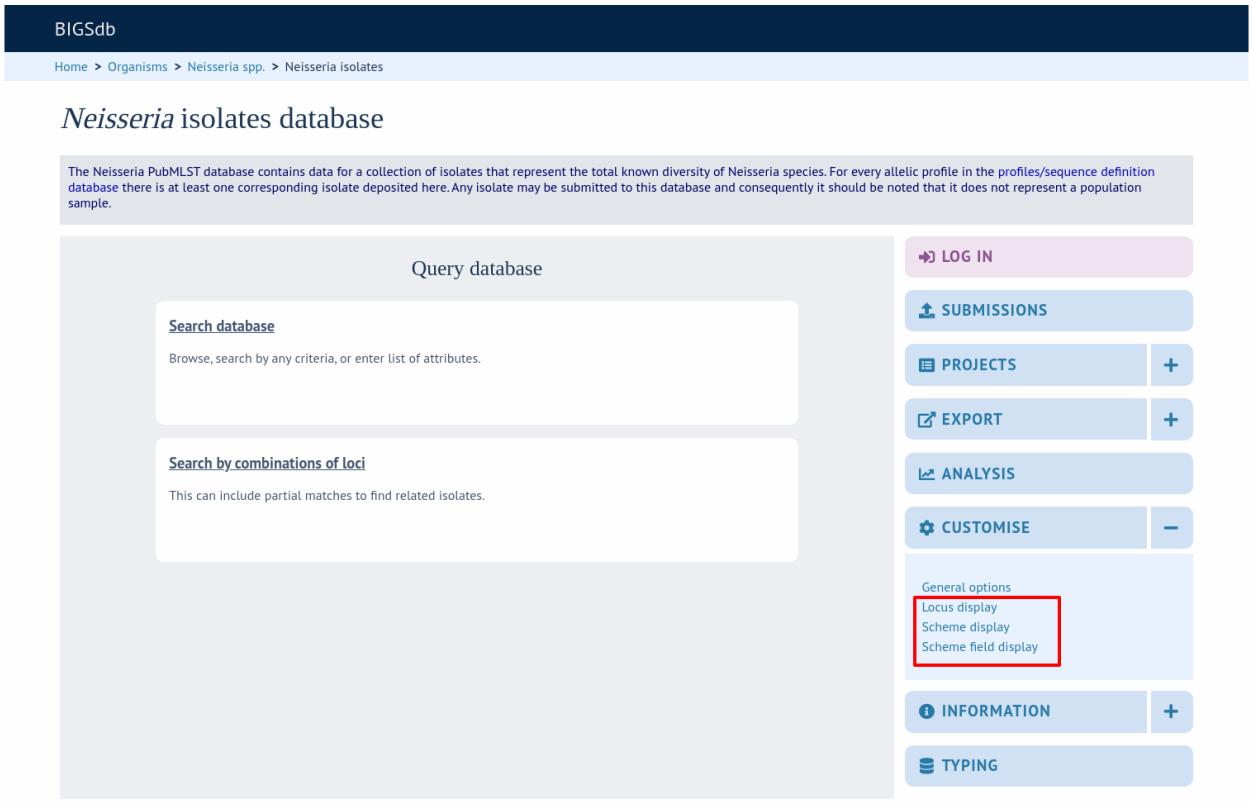
A list of possible filters appears. Click any checkbox for a filter you would like to make available. Click 'Set options' when done. You can return to the default selection by clicking 'Default' followed by 'Set options'.

11.13.6 Modifying locus and scheme display options

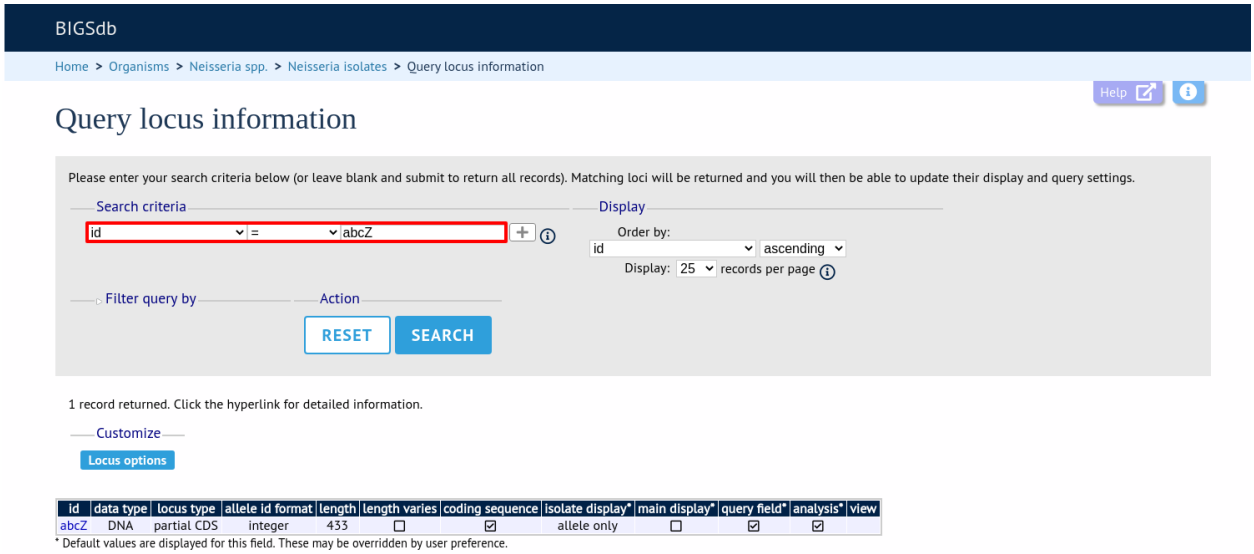
Whether or not loci, schemes or scheme fields are displayed in result tables, isolate records, or within query dropdown boxes can all be set with default options when first defined. These attributes can, however, be overridden by a user, and these selections will be remembered between sessions.

The procedure to modify these attributes is the same for locus, schemes or scheme fields, so the steps for loci will be demonstrated only.

Click the appropriate link in the 'Customise' section on the isolate contents page.

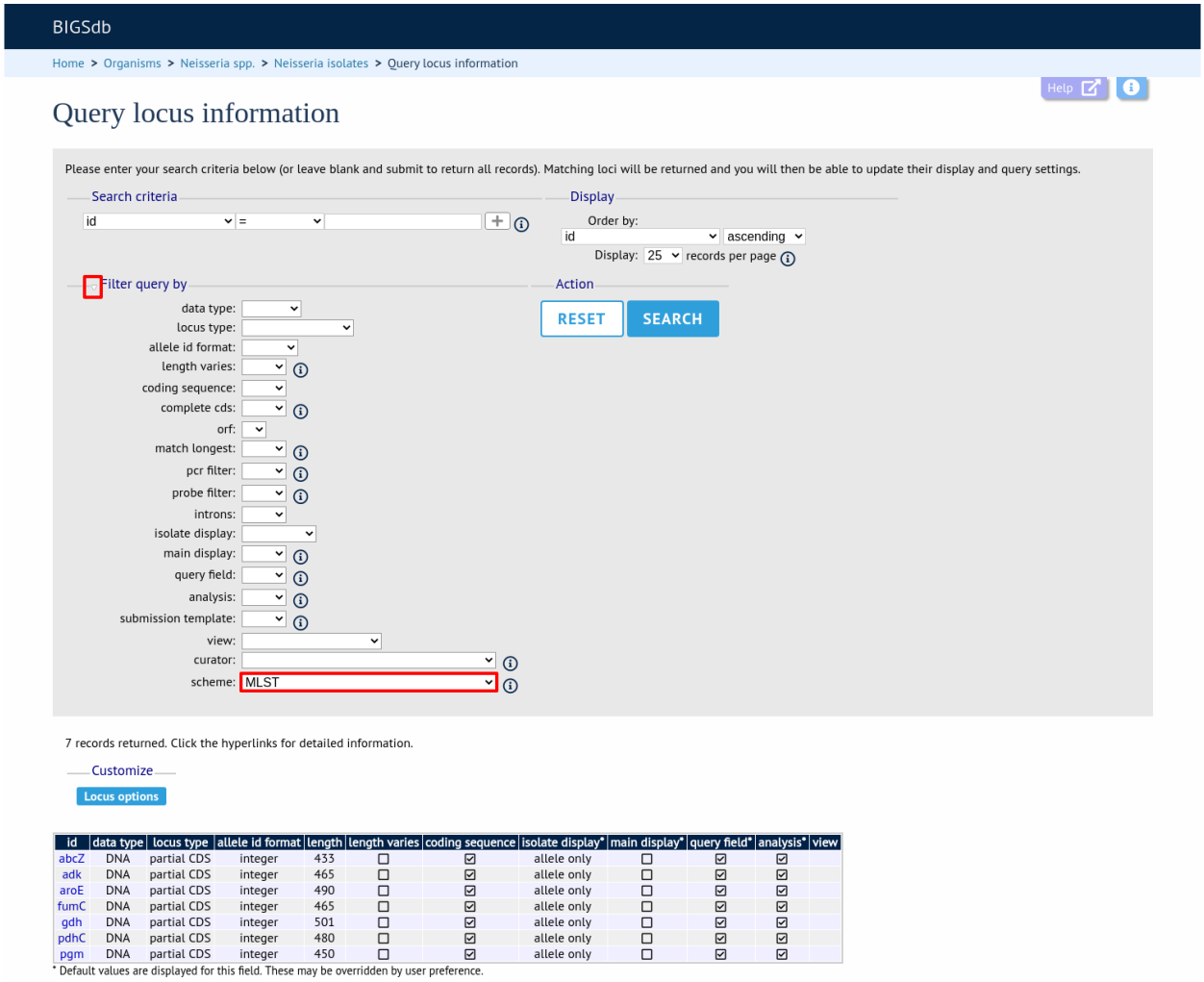


Either select the locus id by querying for it directly.

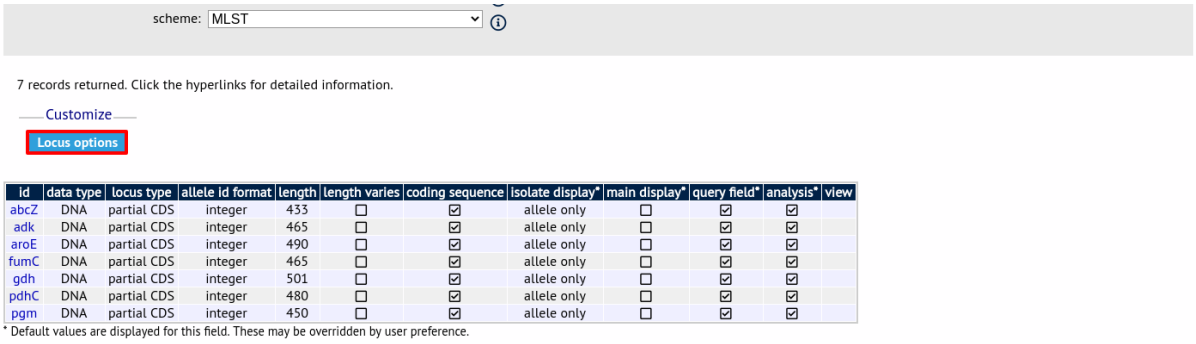


Designations can be queried using *standard operators*.

Alternatively, you can search by filtering loci by schemes. Click the 'Filter query by' header and select the scheme in the dropdown box.



Once loci have been selected, click Customize 'locus options'.



You can then choose to add or remove individual loci from the selection by clicking the appropriate checkboxes. At the bottom of the page are a number of attributes that you can change - clicking 'Change' will affect all selected loci.

Possible options for loci are:

- isolate_display - Sets how the locus is displayed within an isolate record:
  - allele only - display only identifier
  - sequence - display the full sequence

  - hide - don't show at all

- main_display - Sets whether the locus is displayed in the main results table following a query.
- query_field - Sets whether the locus appears in dropdown list boxes to be used within queries.
- analysis - Sets whether the locus can be used in data analysis functions.

Note: Settings for loci can be overridden by those set for schemes that they are members of. For example, if you set a locus to be displayed within a main results table, but that locus is a member of a scheme and you set that scheme not to be displayed, then the locus will not be shown. Conversely, if you set a scheme to be displayed, but set its member locus not to be shown, then that locus will not be displayed (but other loci and scheme fields may be, depending on their independent settings).

CHAPTER

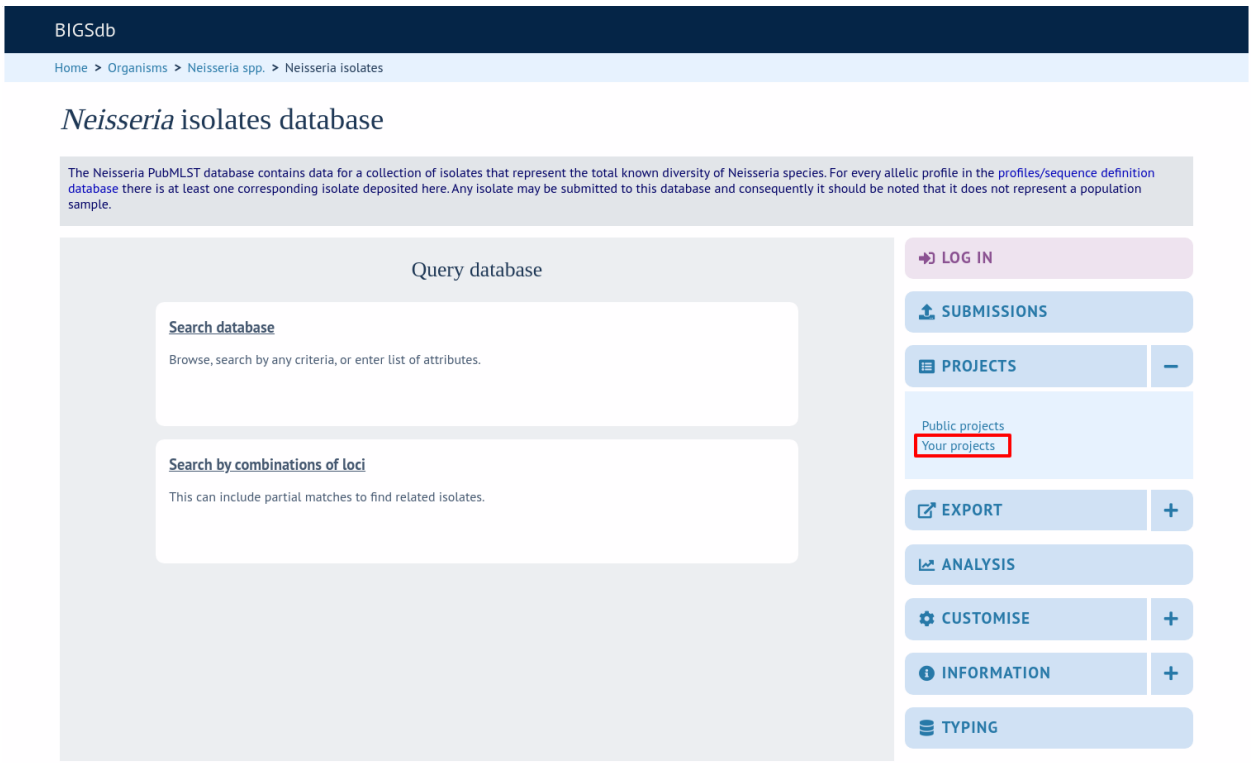
TWELVE

USER PROJECTS

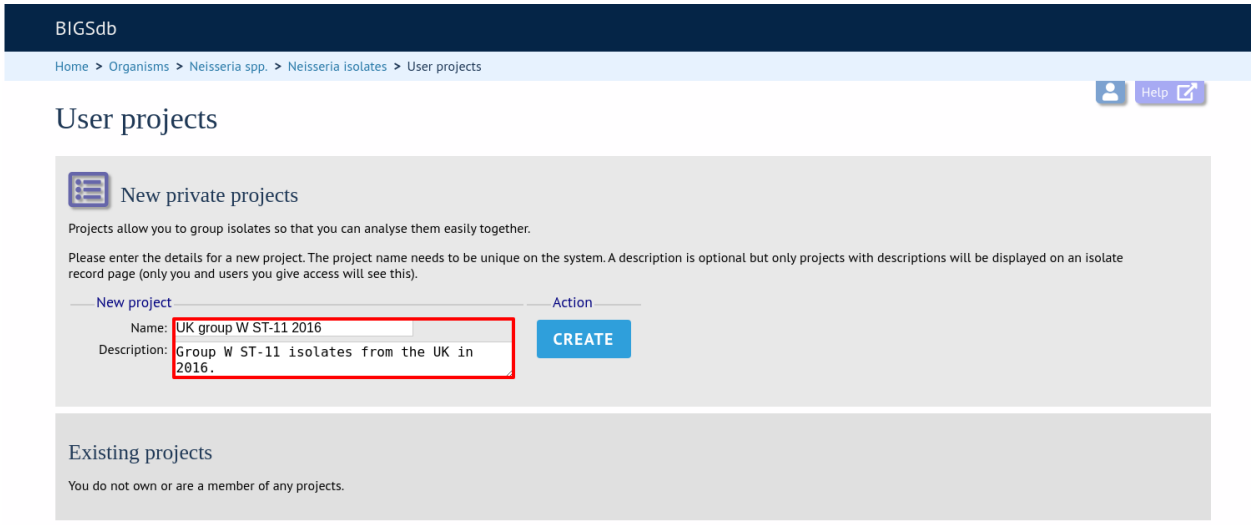
If user projects are enabled, authenticated users are able to set up their own projects in order to group isolates for analysis. If private data and user quotas are enabled, these projects can include private records that can then be shared with other user accounts. Note, that simply adding a record to a user project does not make the record itself private.

Note: User projects can be enabled by an administrator by setting 'user_projects="yes"' in the *config.xml* file for the database.

To create a new project, go to the isolate database contents page and expand the 'Projects' section. Click 'Your projects'.



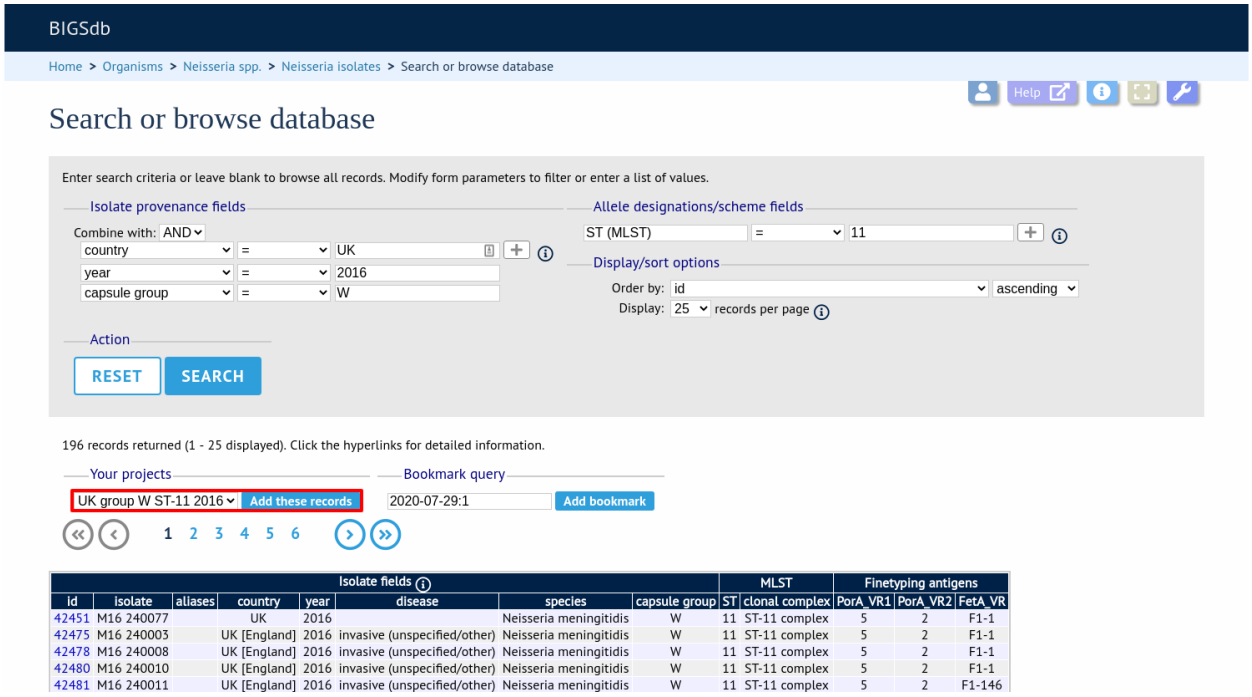
Enter the name of your project (this must be unique on the system - you will be told if the name is already used). You can optionally include a description - if you do this, this will appear within *isolate records* when accessed from your account. Click 'Create'.



You can either add isolates to your project directly following a query or by manually editing a list of ids.

12.1 Adding isolates to a user project following a query

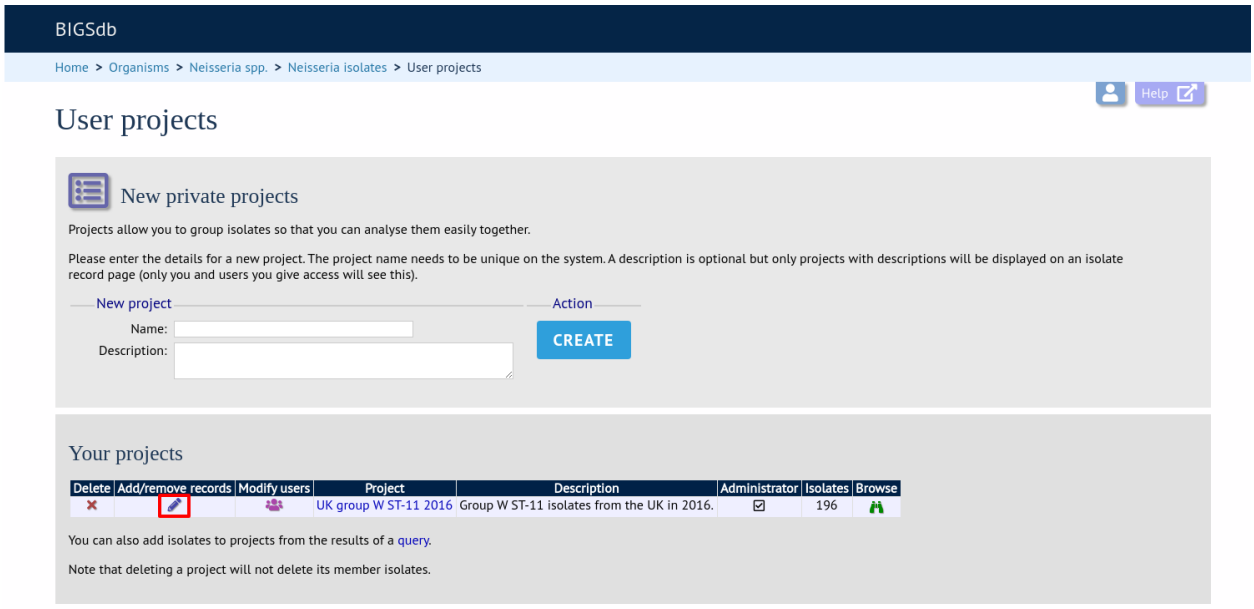
Perform an isolate database query. If you have set up a project, there will be a link in the results header box. Select your project and click 'Add these records'.



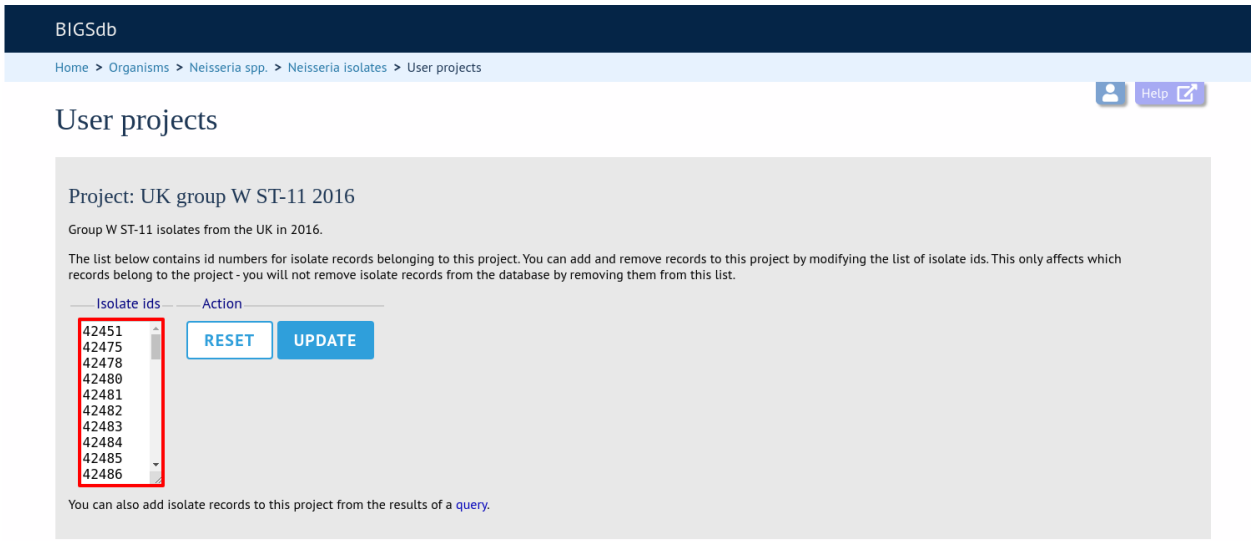
The records will be added to the project. Please note that it doesn't matter whether any of the records have been previously added.

12.2 Adding or removing isolates belonging to a user project by editing a list

From the user projects page, click the 'Add/remove records' link for the project that you wish to modify.

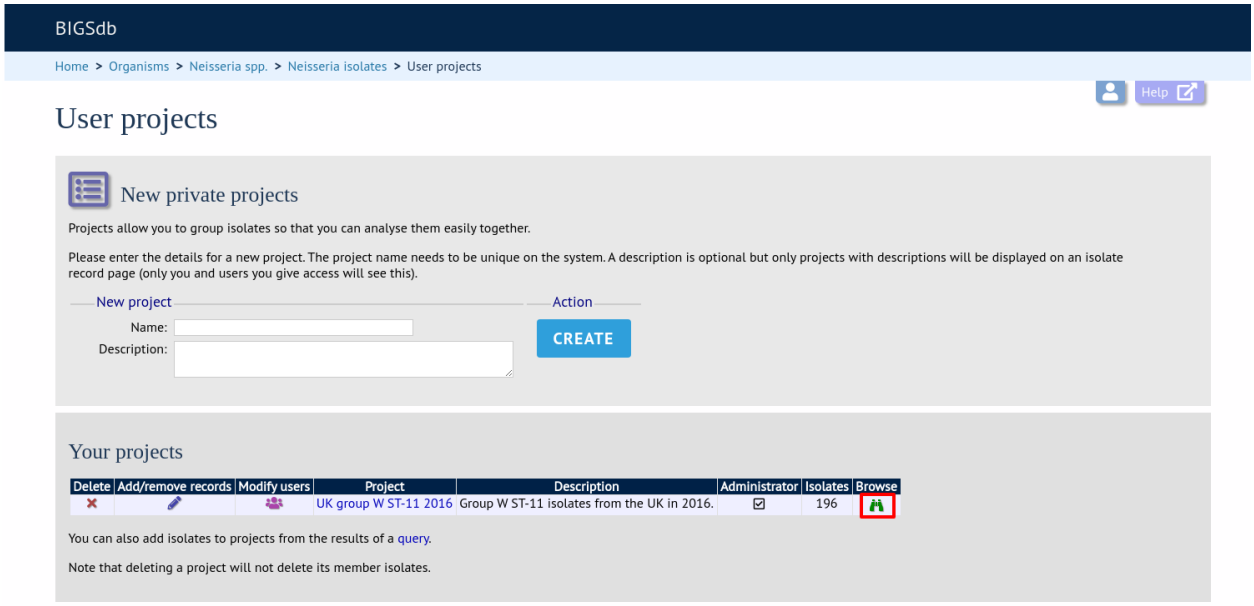


Edit the list of ids of records belonging to the list. You can copy and paste this list if you wish to prepare it in a spreadsheet or text editor. Click 'Update' when finished.

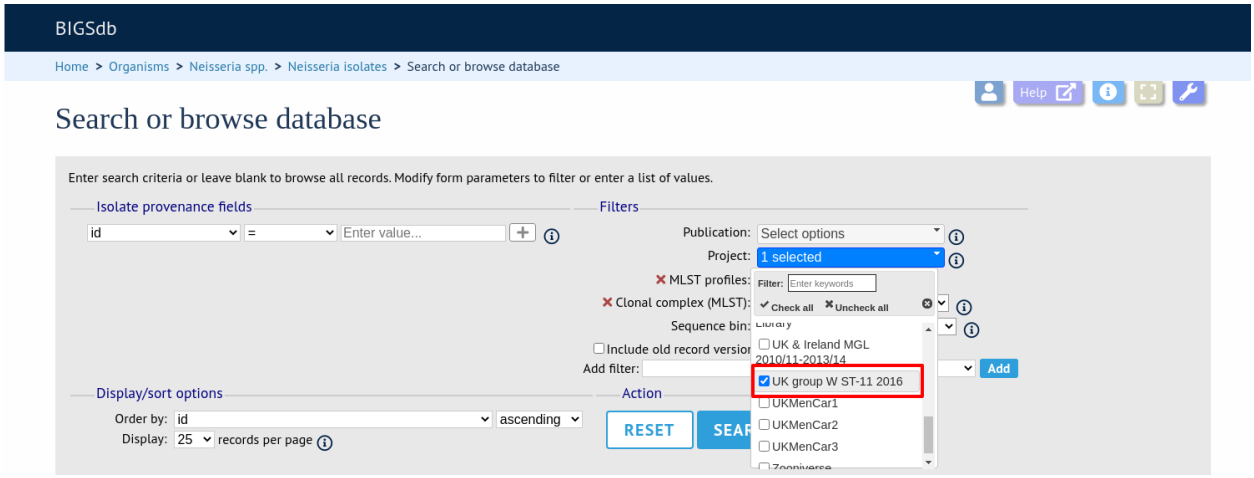


12.3 Accessing project isolates

To browse isolate records belonging to a project, go to the user projects page and click the 'Browse' link for the project.



Alternatively, you can select the project from the *projects filter* on the isolate query page. This would enable you to combine the project with additional search criteria

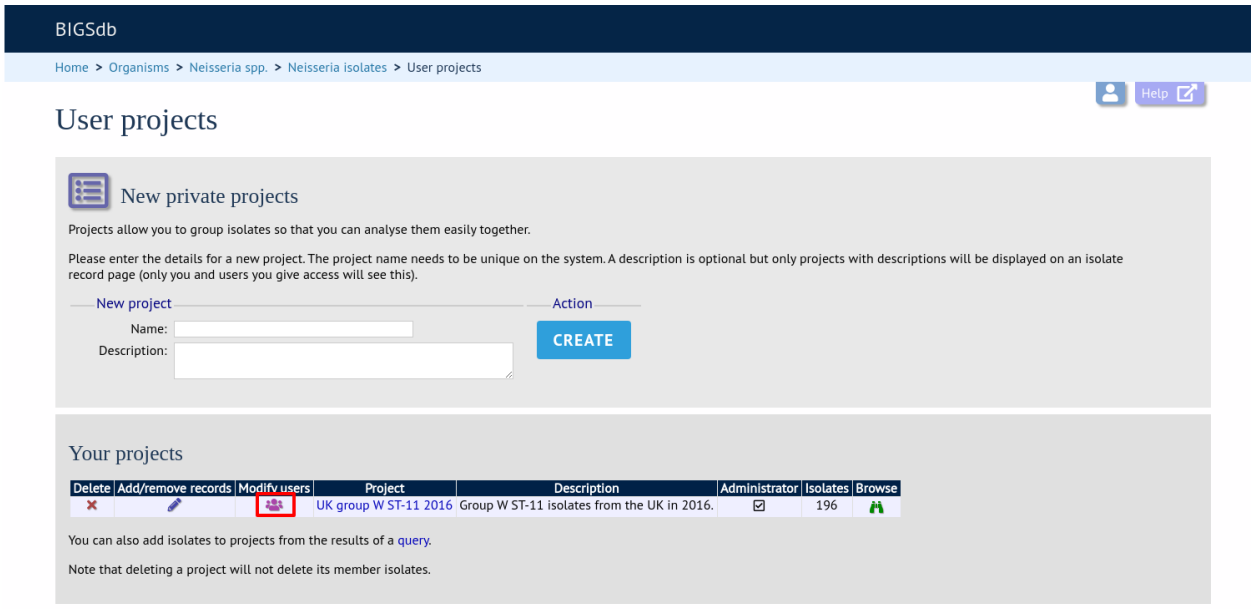


Please note that you will only see your project in the filter list if you are logged in. As a private project, only you will see it.

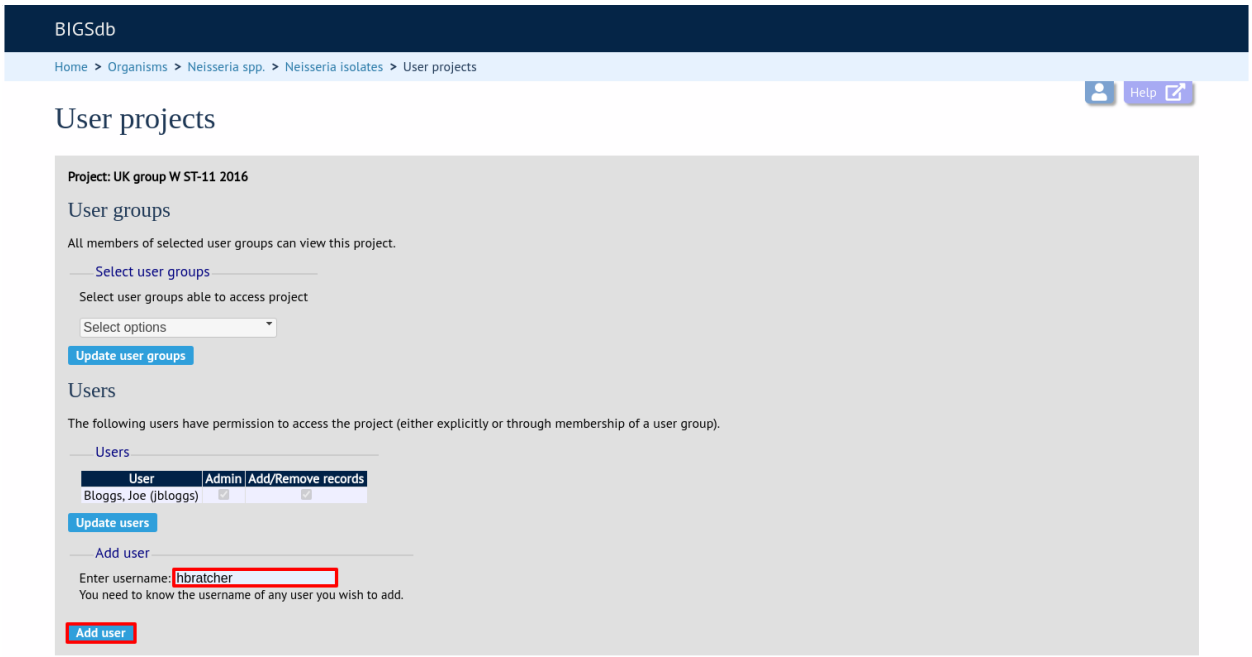
12.4 Allowing other users to share your project

You can share a project that you own with any other user. In order to do this, you must know the username that they use to log in with. They will see the project in their own list of private projects and in the query interface projects filter.

From the user projects page, click the 'Modify users' link for the specified project:

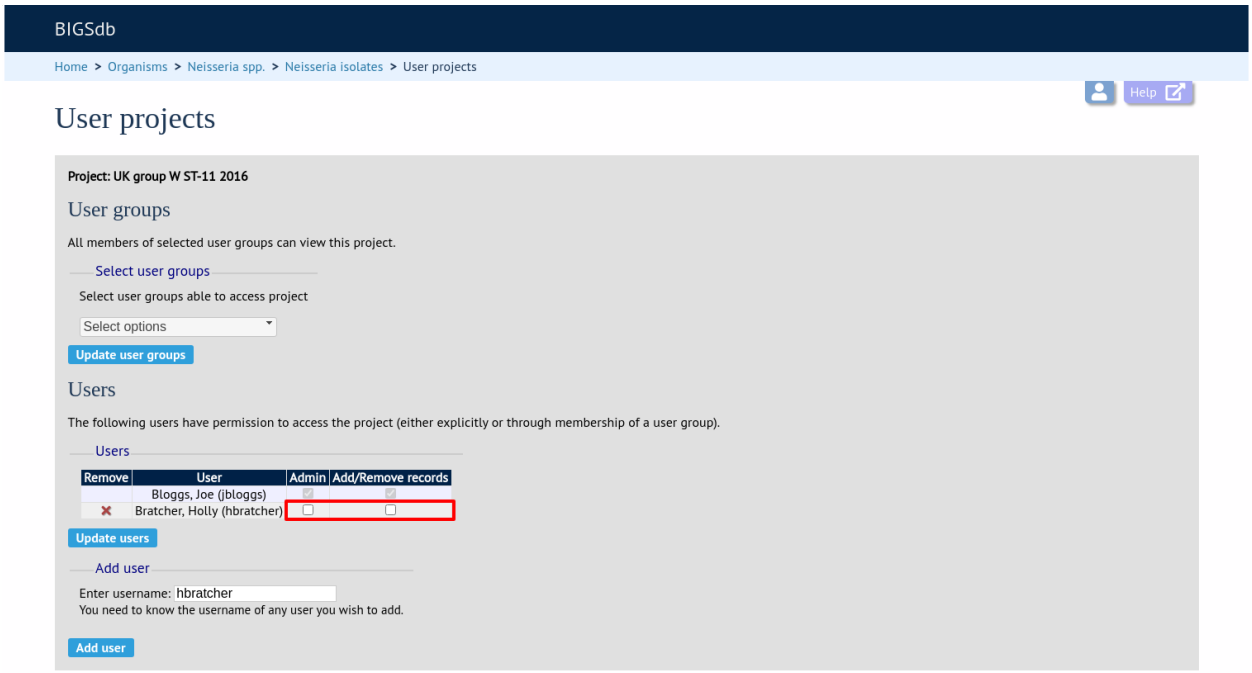


Enter the username of the person you wish to share with and click 'Add user':



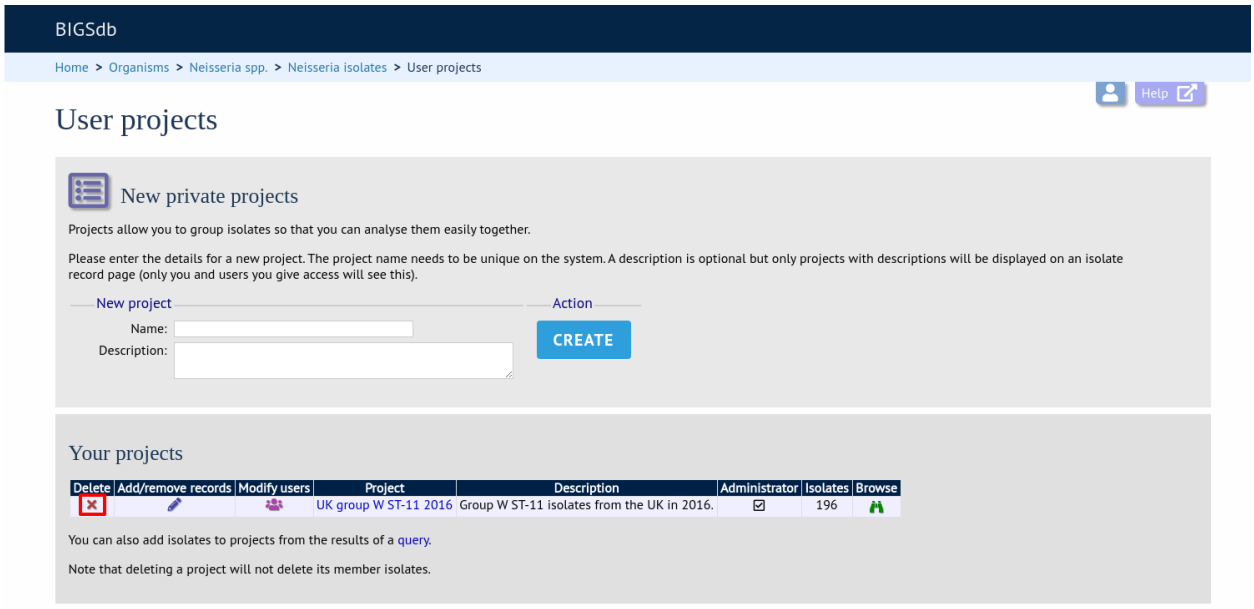
If user groups are in use, you can also share your project with all members of a user group from the same page.

Once a user has been added, you can give them permission to administer (add other users, modify the description or delete) or add/remove records to the project:

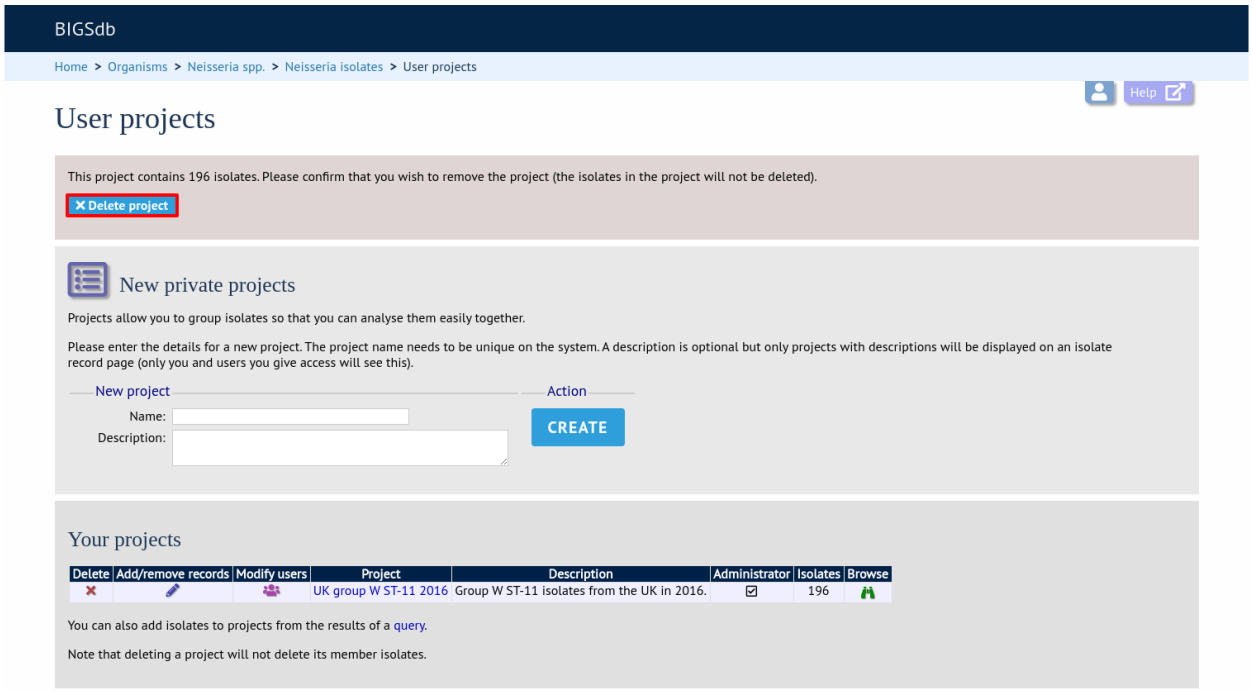


12.5 Deleting a user project

You can delete a project from the user projects page by clicking the 'Delete' link next to the project in question.



If the project contains any isolates you will be asked for confirmation. Click the 'Delete project' button.



If the project contains no isolates, the confirmation page is skipped and the project is removed immediately.

Note: Removing a project does not delete the isolate records belonging to it. A project is simply a means of grouping records.

CHAPTER

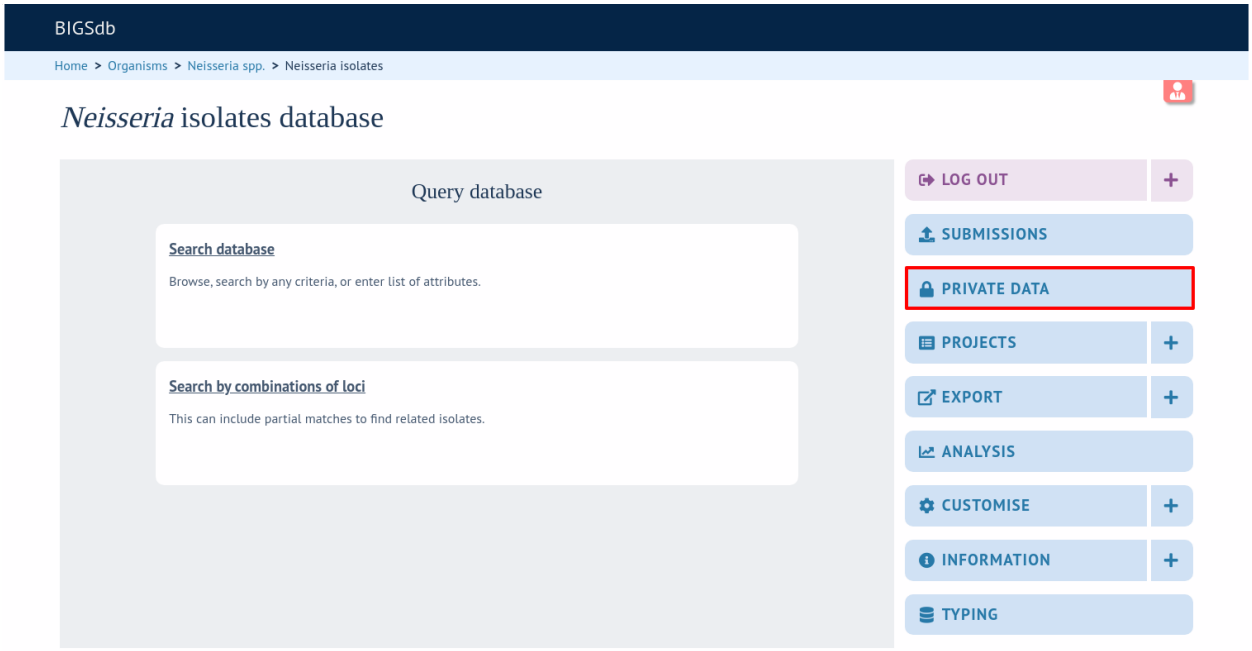
THIRTEEN

PRIVATE RECORDS

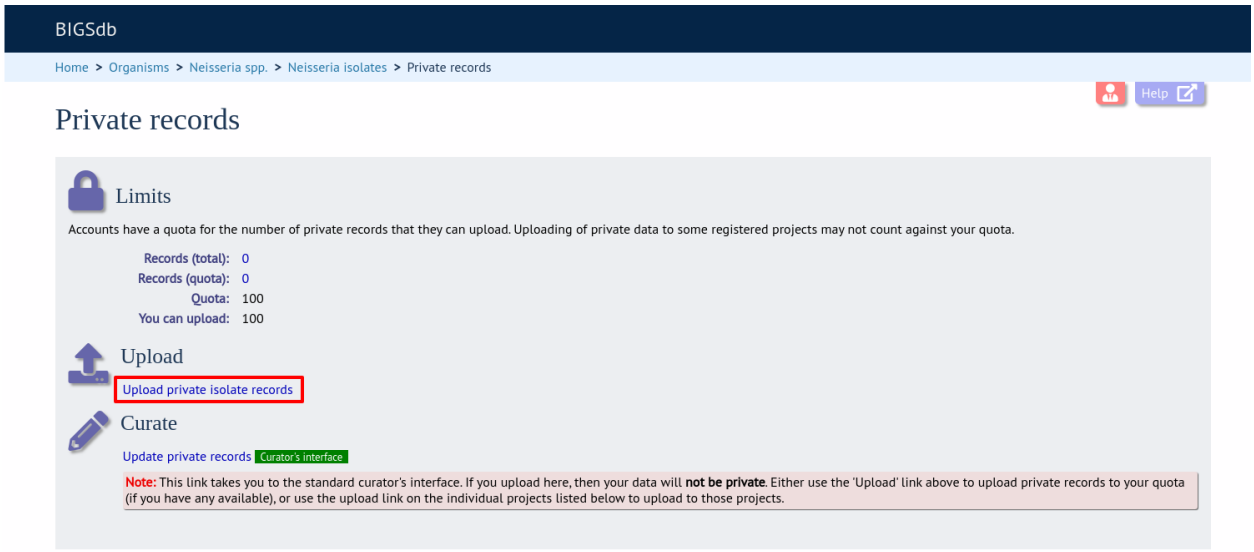
Users with a status of 'submitter', 'curator', or 'admin' can upload private isolate records that will be hidden from public view. A quota needs to be set for the user by an admin before they are able to do this.

13.1 Uploading private records

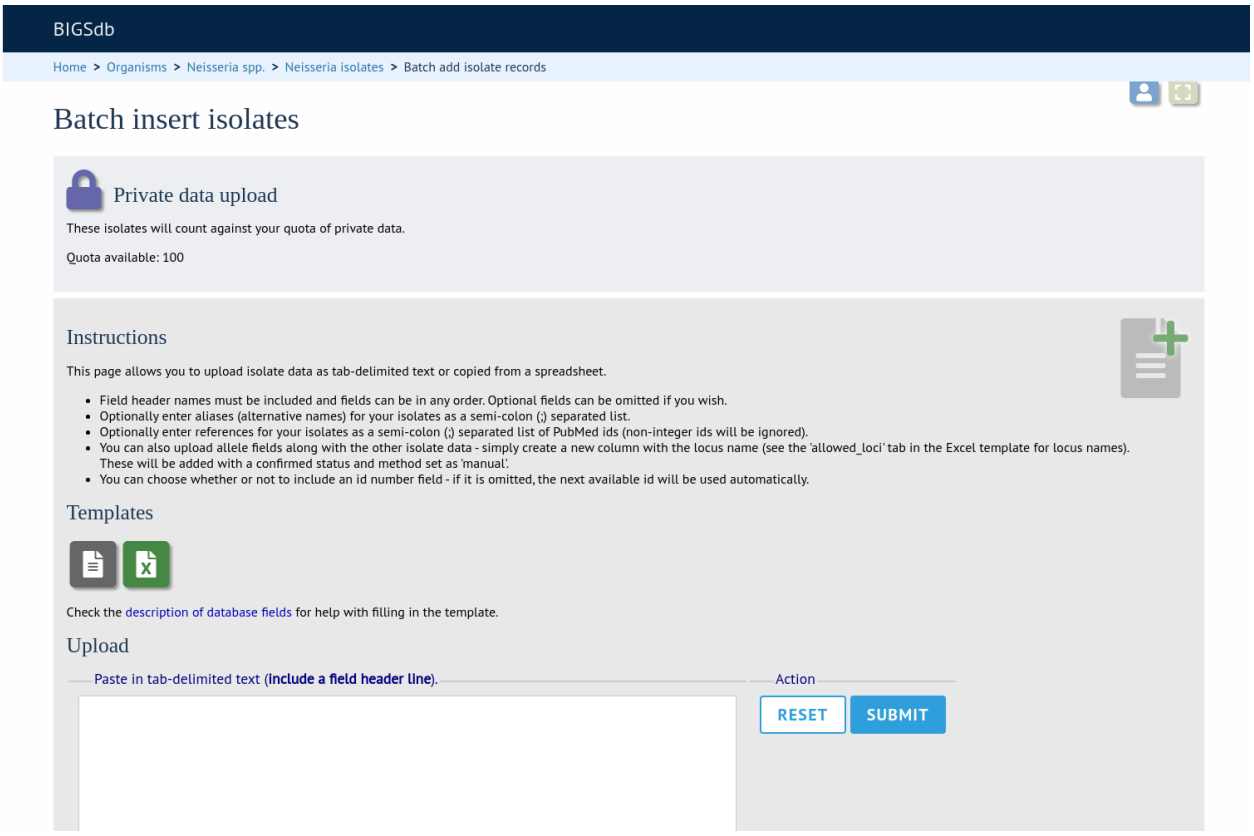
In order to upload private records, you need to make sure that you are logged in. If your account has a quota, there will be a menu item called 'Private data'. Click the 'Upload/manage records' link.



You will see an overview of your quota and links to upload and edit your records. Click the 'Upload private isolate records' link (assuming you have quota available).



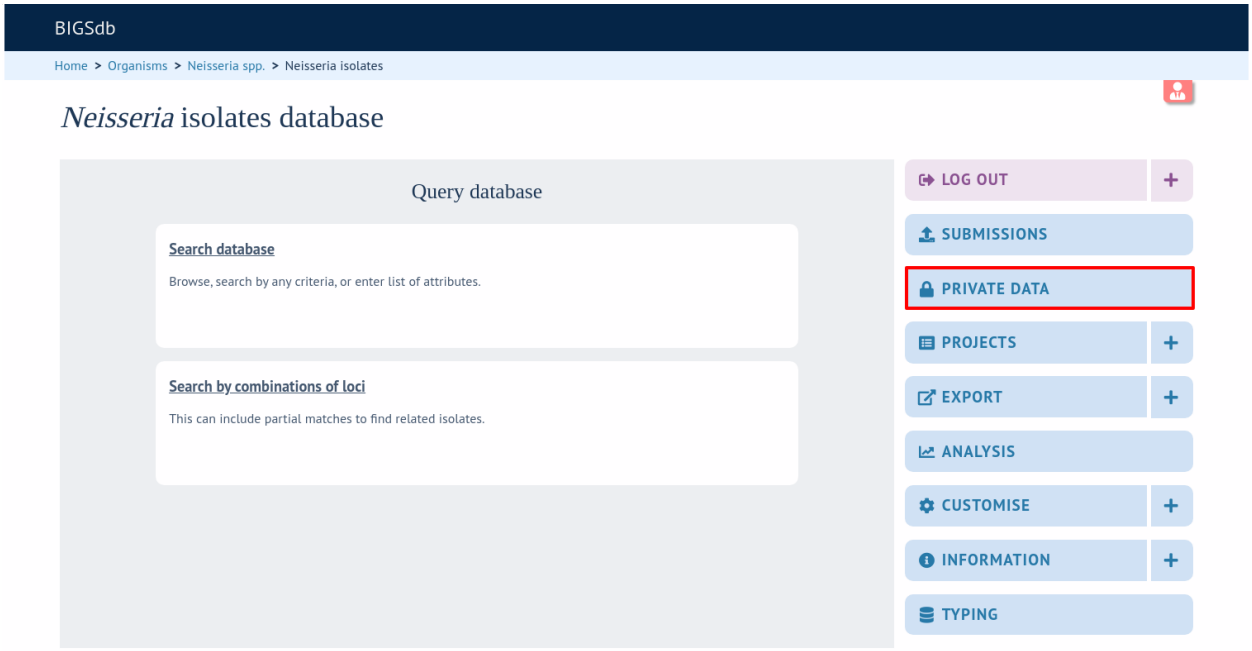
You will then be taken to a curator page that allows you to upload data copy and pasted from a spreadsheet. There is a link to an Excel template that you will need to use to prepare your data - this is the same template used if you were to submit your data to a curator.



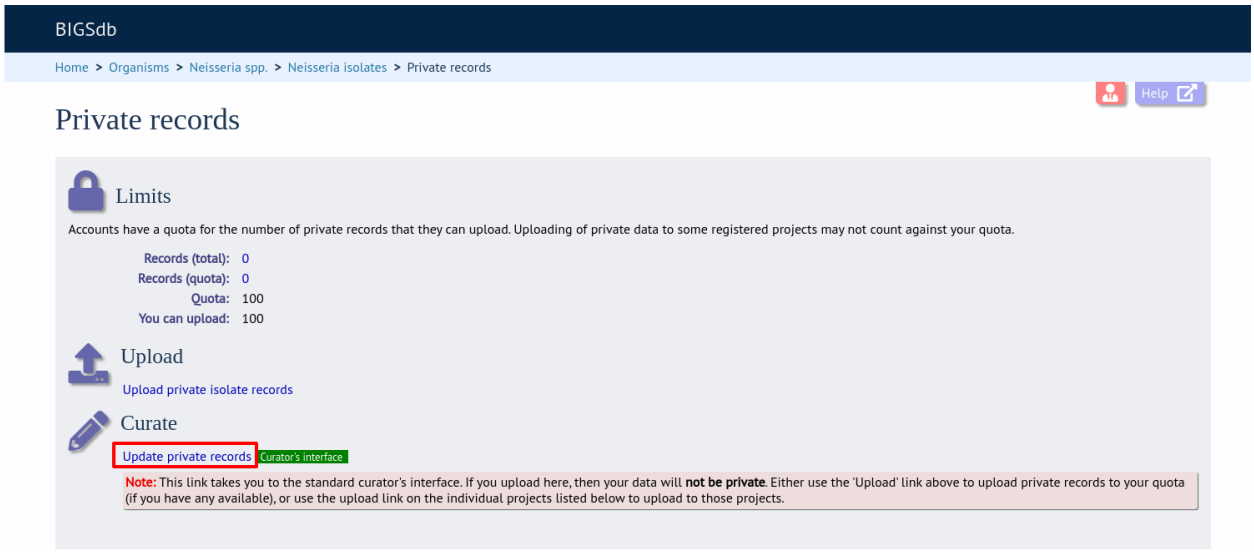
See *batch adding isolate records* for details of the upload process.

13.2 Modifying private records

Click the 'Private data' link on the contents page.



Now click the 'Update private records' link. You will be taken to the *curators' interface*, where you will be able to use the standard curator tools.



Use the *curators' interface* to make any changes to your isolate records, including *uploading genome data*.

13.3 Sharing access to private records

If *user projects* are enabled on the database, you can share access to your private isolates by adding them in to a user project and then sharing this.

See *user projects* for more details.

CHAPTER

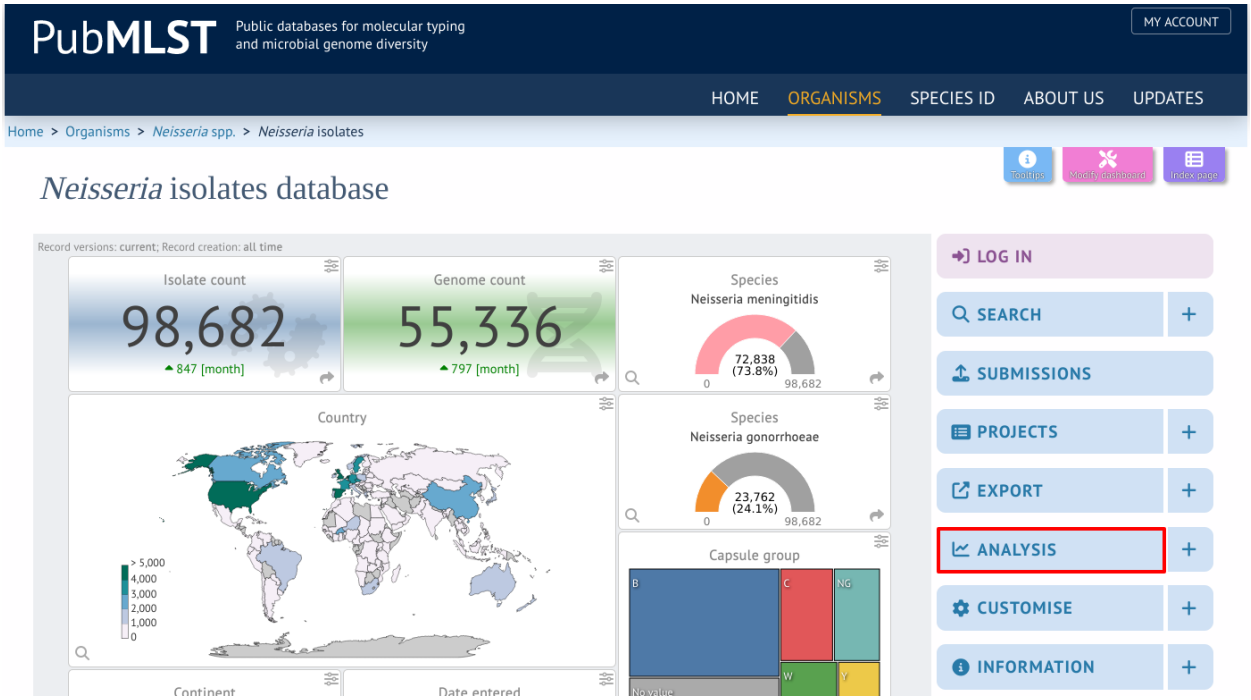
FOURTEEN

DATA ANALYSIS PLUGINS

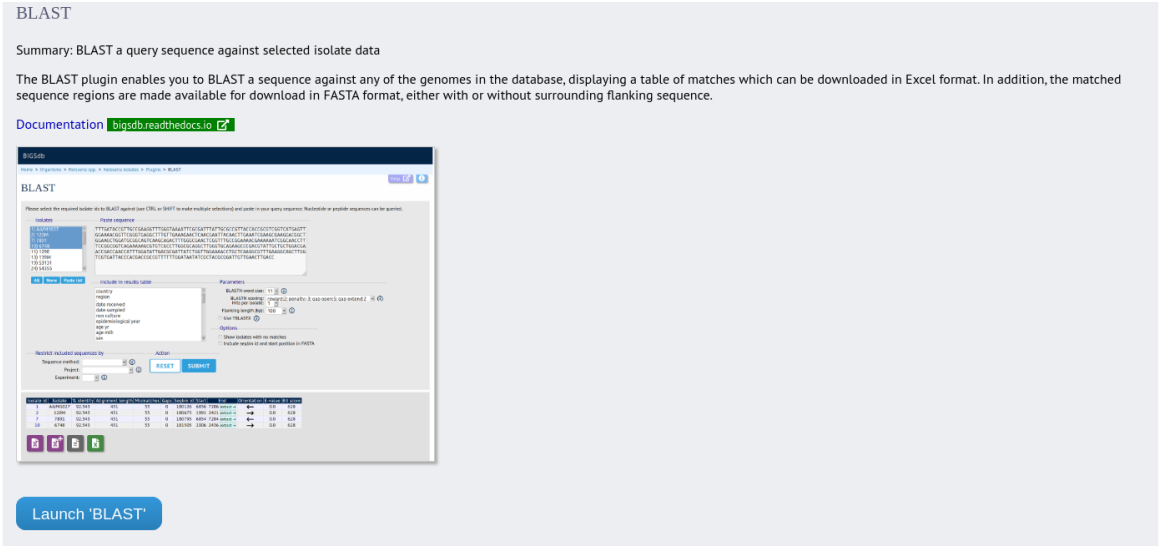
14.1 BLAST

The BLAST plugin enables you to BLAST a sequence against any of the genomes in the database, displaying a table of matches and extracting matching sequences.

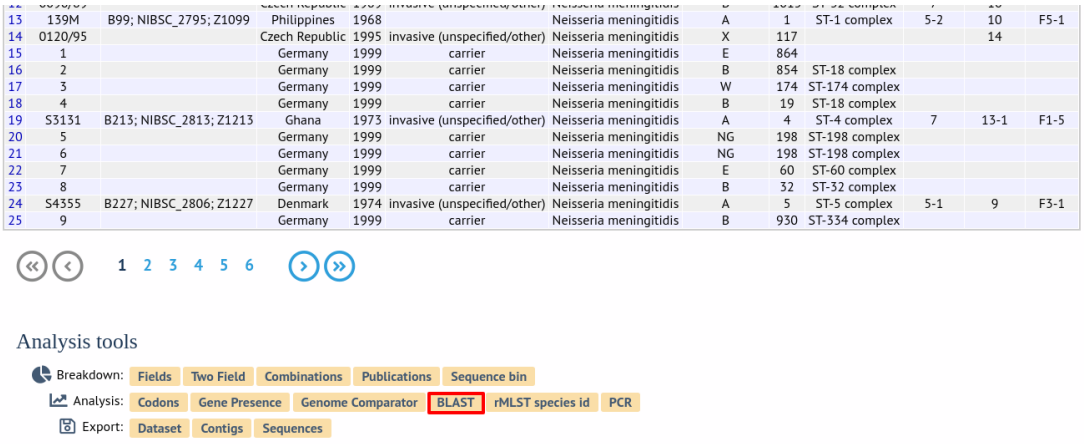
The function can be accessed by selecting the 'Analysis' section on the main contents page.



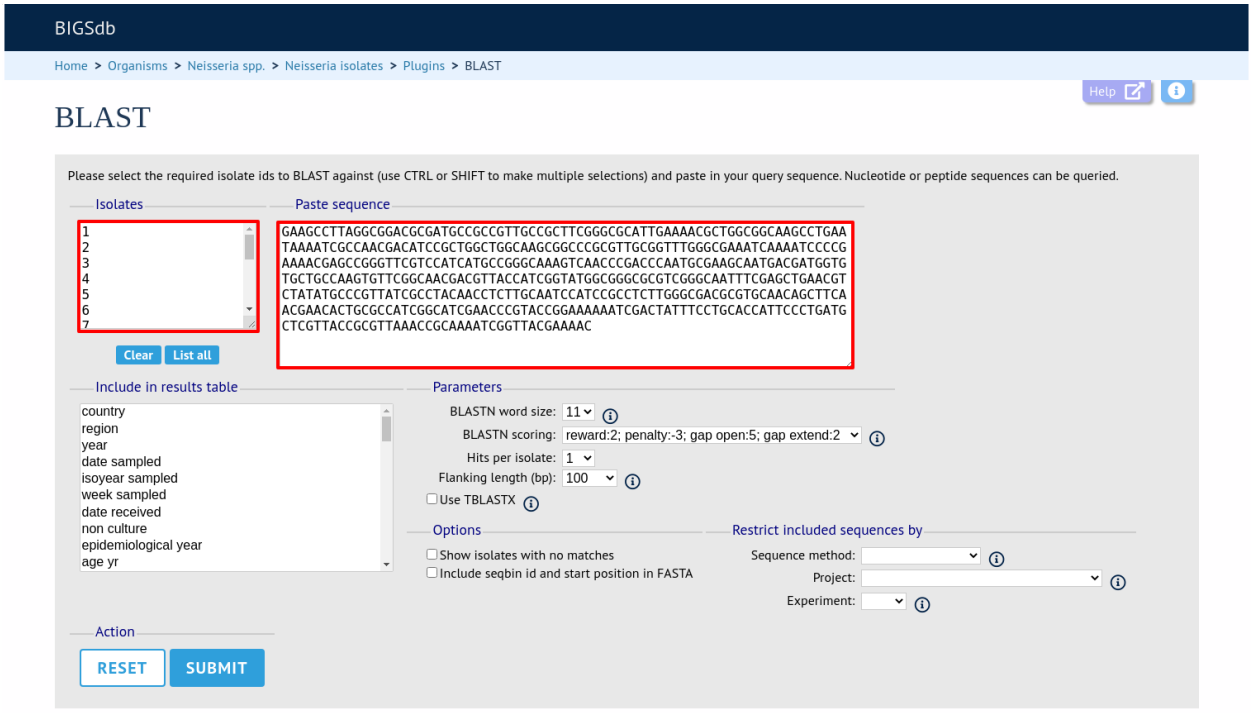
Jump to the 'Analysis' category, follow the link to BLAST, then click 'Launch BLAST'.



Alternatively,it can be accessed following a query by clicking the 'BLAST' button in the Analysis list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.

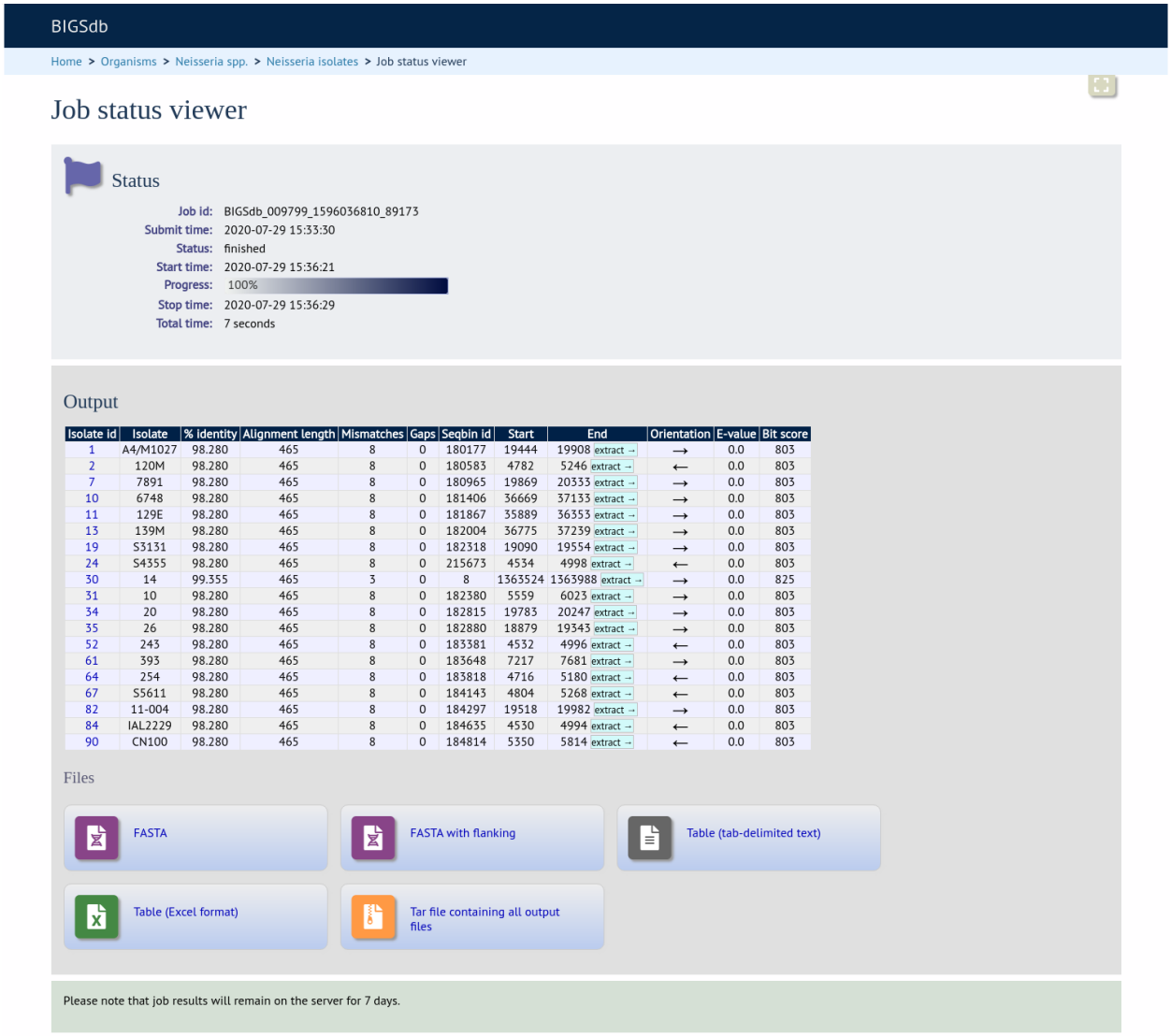


Select the isolate records to analyse (on large databases you will need to enter a list of ids). These will be pre-selected if you accessed the plugin following a query. Paste in a sequence to query - this be either a DNA or peptide sequence.

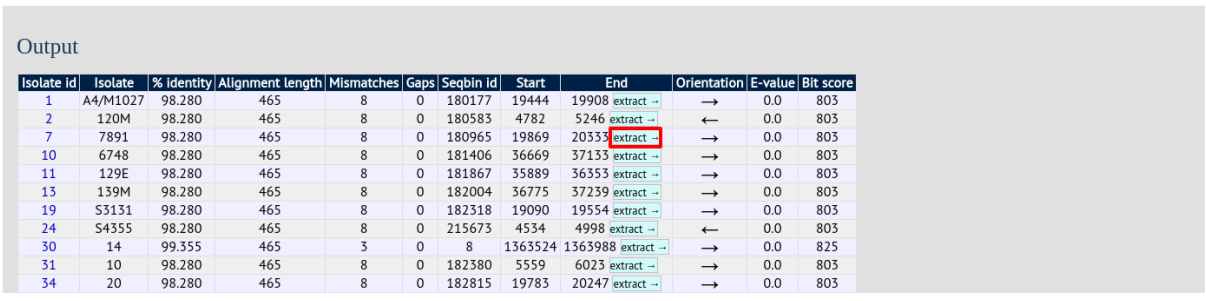


Click submit. If you are querying against 10 or fewer genomes then the results are run immediately, otherwise the job is sent to the job queue.

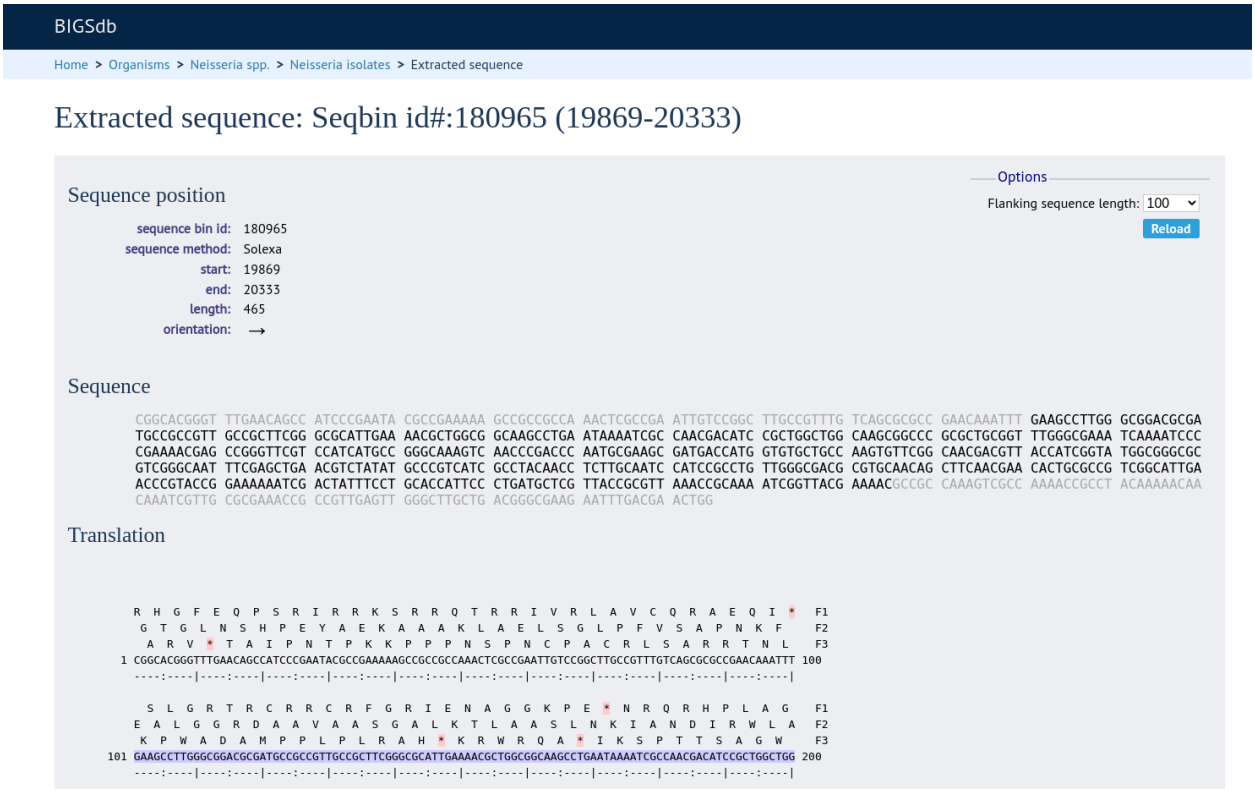
A table of BLAST results will be displayed.



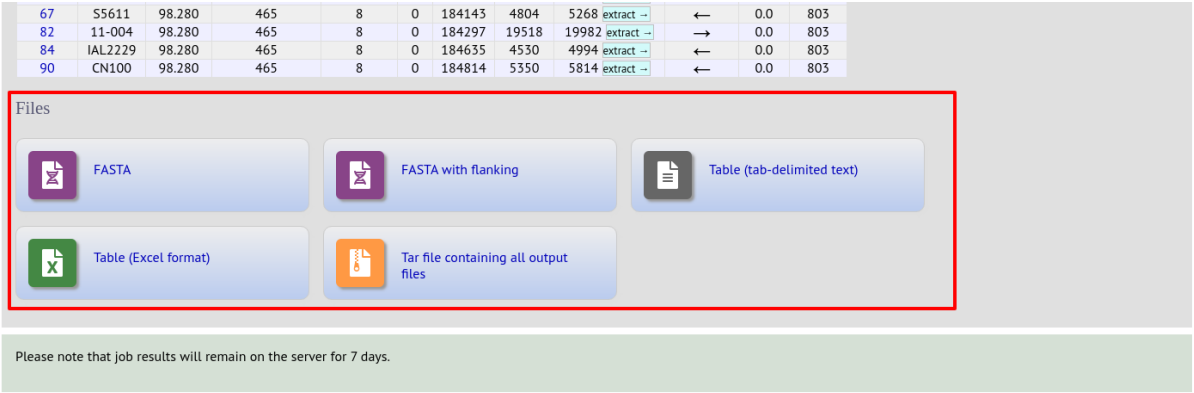
Clicking any of the 'extract' buttons to display the matched sequence.



The extracted sequence is shown along with a translated sequence and flanking sequences.

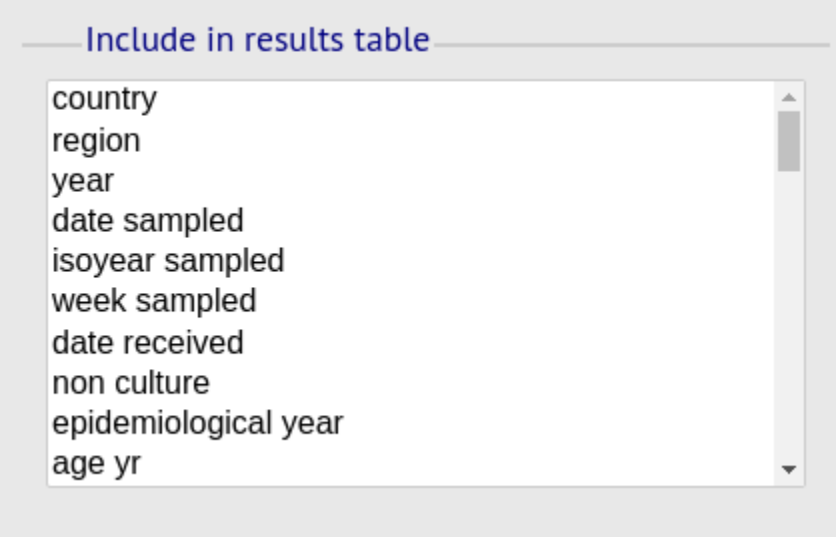


At the bottom of the results table are links to export the matching sequences in FASTA format, (optionall) including flanking sequnces. You can also export the table in tab-delimited text or Excel formats.



14.1.1 Include in results table fieldset

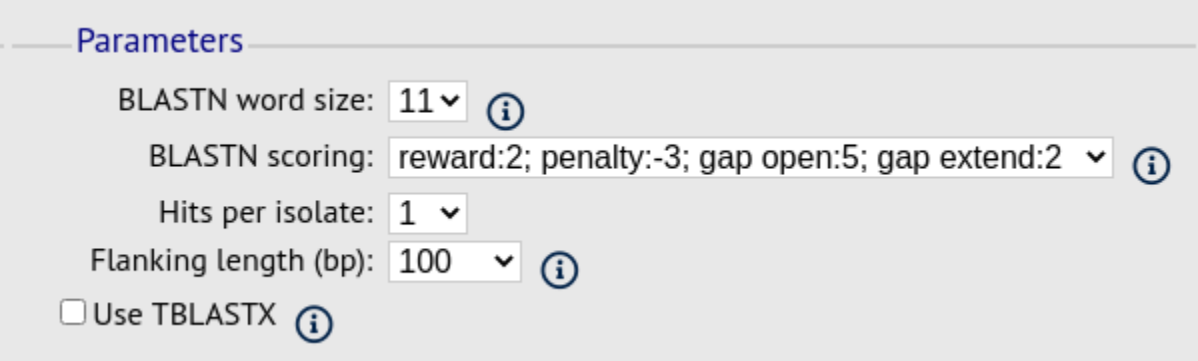
This selection box allows you to choose which isolate provenance fields will be included in the results table.



Multiple values can be selected by clicking while holding down Ctrl.

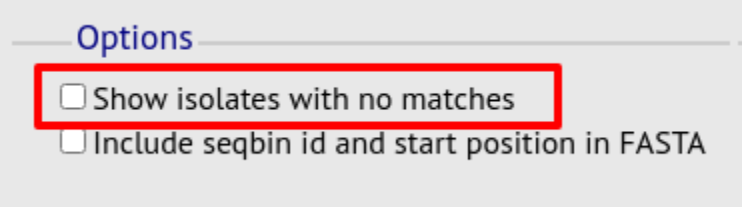
14.1.2 Parameters fieldset

This section allows you to modify BLAST parameters. This affects sensitivity and speed.



- BLASTN word size - This is the length of the initial identical match that BLAST requires before extending a match (default: 11). Increasing this value improves speed at the expense of sensitivity.
- BLASTN scoring - This is a dropdown box of combinations of identical base rewards; mismatch penalties; and gap open and extension penalties. BLASTN has a constrained list of allowed values which reflects the available options in the list.
- Hits per isolate - By default, only the best match is shown. Increase this value to the number of hits you'd like to see per isolate.
- Flanking length - Set the size of the upstream and downstream flanking sequences that you'd like to include.
- Use TBLASTX - This compares the six-frame translation of your nucleotide query sequence against the six-frame translation of the contig sequences. This is significantly slower than using BLASTN.

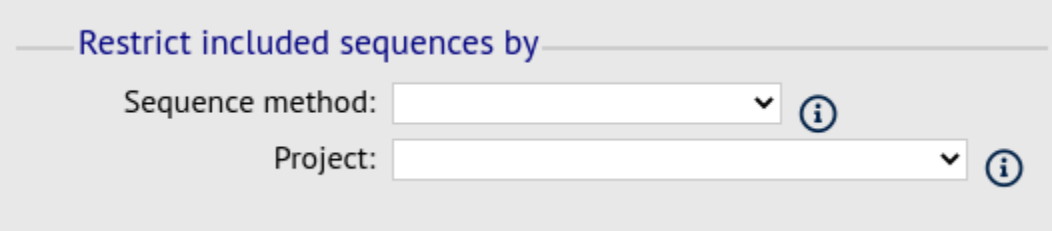
14.1.3 No matches



Click this option to create a row in the table indicating that a match was not found. This can be useful when screening a large number of isolates.

14.1.4 Filter fieldset

This section allows you to further filter your collection of isolates and the contig sequences to include.



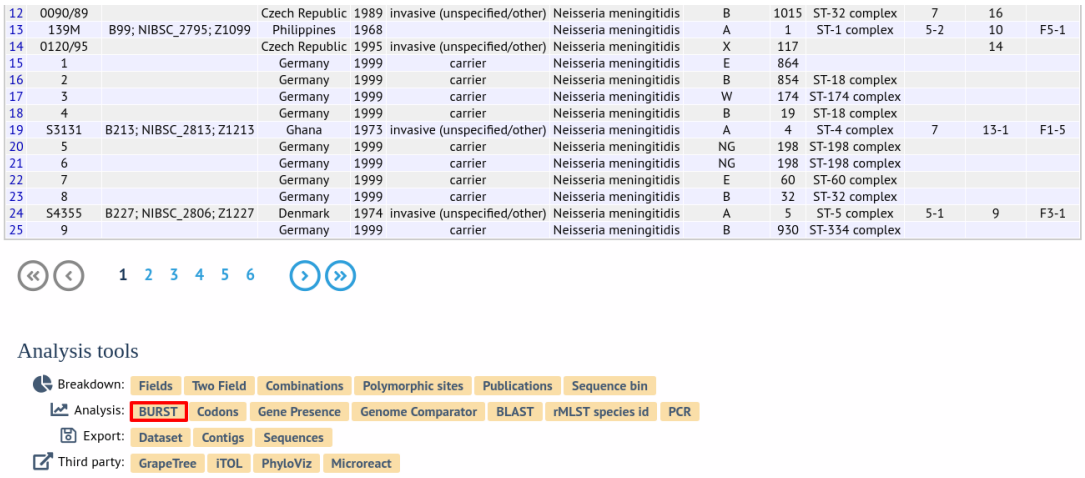
Available options are:

- Sequence method - Choose to only analyse contigs that have been generated using a particular method. This depends on the method being set when the contigs were uploaded.
- Project - Only include isolates belonging to the chosen project. This enables you to select all isolates and filter to a project.

14.2 BURST

BURST is an algorithm used to group MLST-type data based on a count of the number of profiles that match each other at specified numbers of loci. The analysis is available for both sequence definition database and isolate database schemes that have primary key fields set. The algorithm has to be *specifically enabled* by an administrator. Analysis is limited to 1000 or fewer records.

The plugin can be accessed following a query by clicking the 'BURST' button in the Analysis list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.



12	0090/89		Czech Republic	1989	invasive (unspecified/other)	Neisseria meningitidis	B	1015	ST-32 complex	7	16	
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
14	0120/95		Czech Republic	1995	invasive (unspecified/other)	Neisseria meningitidis	X	117			14	
15	1		Germany	1999	carrier	Neisseria meningitidis	E	864				
16	2		Germany	1999	carrier	Neisseria meningitidis	B	854	ST-18 complex			
17	3		Germany	1999	carrier	Neisseria meningitidis	W	174	ST-174 complex			
18	4		Germany	1999	carrier	Neisseria meningitidis	B	19	ST-18 complex			
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
20	5		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
21	6		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
22	7		Germany	1999	carrier	Neisseria meningitidis	E	60	ST-60 complex			
23	8		Germany	1999	carrier	Neisseria meningitidis	B	32	ST-32 complex			
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
25	9		Germany	1999	carrier	Neisseria meningitidis	B	930	ST-334 complex			

1 2 3 4 5 6

Analysis tools

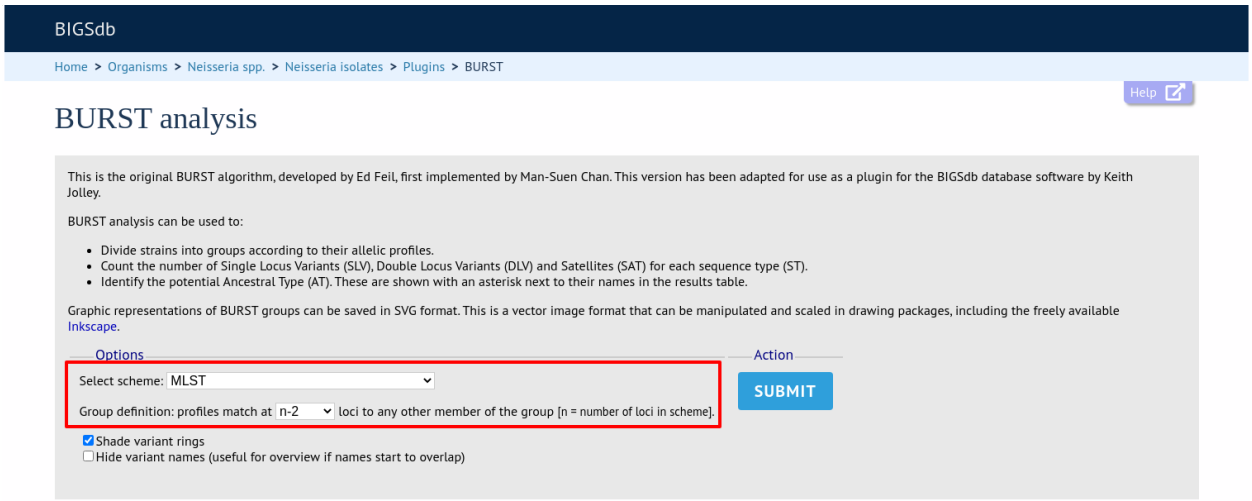
Breakdown: Fields | Two Field | Combinations | Polymorphic sites | Publications | Sequence bin

Analysis: BURST | Codons | Gene Presence | Genome Comparator | BLAST | rMLST species id | PCR

Export: Dataset | Contigs | Sequences

Third party: GrapeTree | iTOL | PhyloViz | Microreact

If there multiple schemes that can be analysed, these can then be selected along with the group definition.

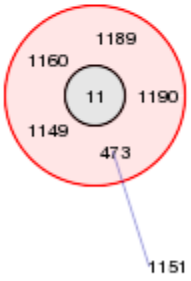


Modifying the group definition affects the size of groups and how they link together. By default, the definition is n-2 (where n is the number of loci), so for example on a 7 locus MLST scheme groups contain STs that match at 5 or more loci to any other member of the group.

Click Submit.

A series of tables will be displayed indicating the groups of profiles. Where one profile can be identified as a central genotype, i.e. the profile that has the greatest number of other profiles that are single locus variants (SLV), double locus variants (DLV) and so on, a graphical representation will be displayed. The central profile is indicated with an asterisk.

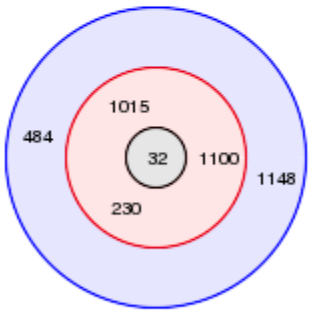
group: 2				
ST	Frequency	SLV	DLV	SAT
11*	37	5		1
473	1	2	4	
1149	19	1	4	1
1151	1	0	1	5
1160	1	2	3	1
1189	1	1	4	1
1190	1	1	4	1



SVG file (right click to save)

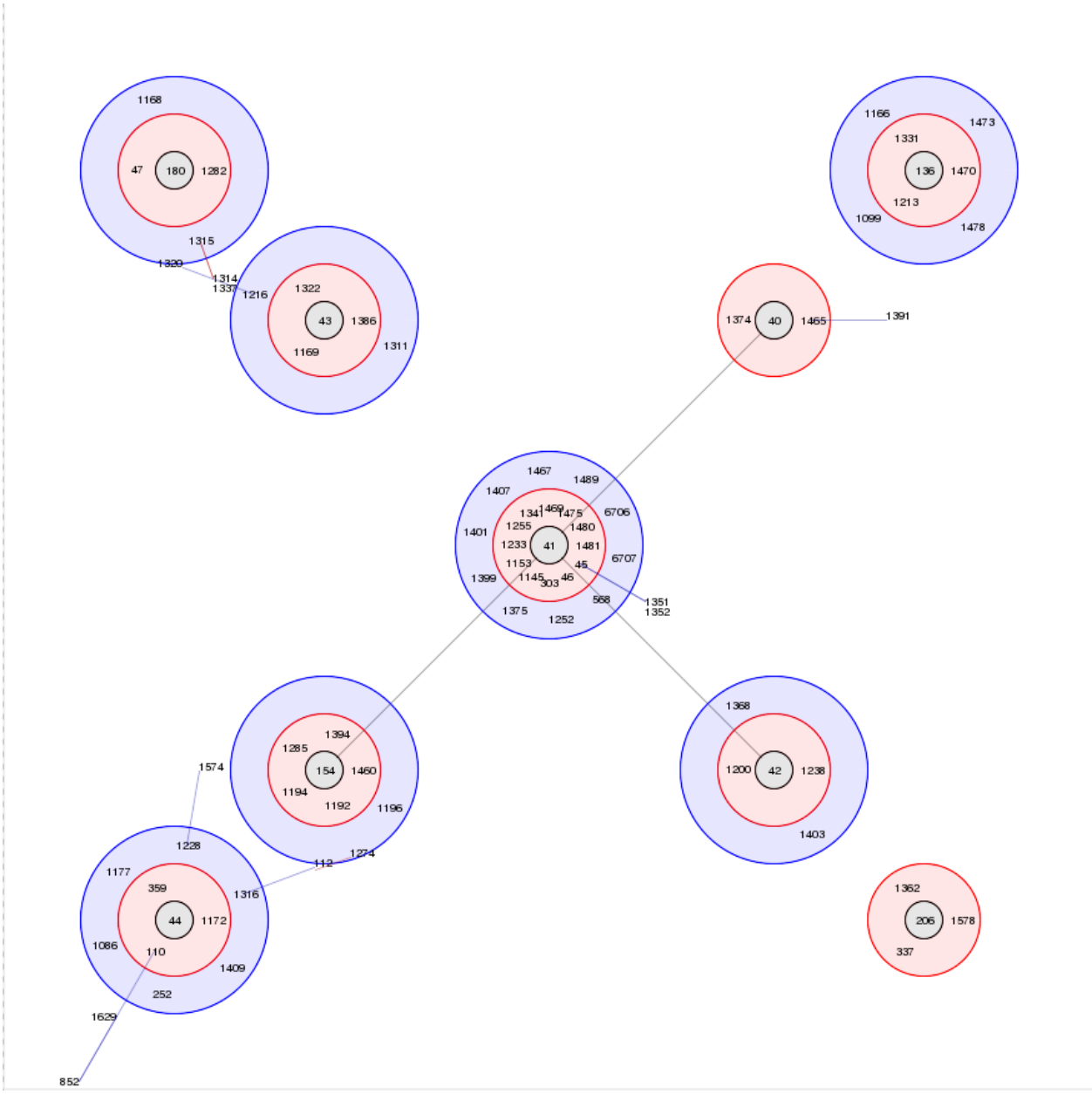
SLV profiles that match the central profile are shown within a red circle surrounding the central profile. Most distant profiles (triple locus variants) may be linked with a line. Larger groups may additionally have DLV profiles. These are shown in a blue circle.

group: 6				
ST	Frequency	SLV	DLV	SAT
32*	2	3	2	
230	1	1	3	1
484	1	0	3	2
1015	1	1	4	
1100	1	1	2	2
1148	1	0	4	1



SVG file (right click to save)

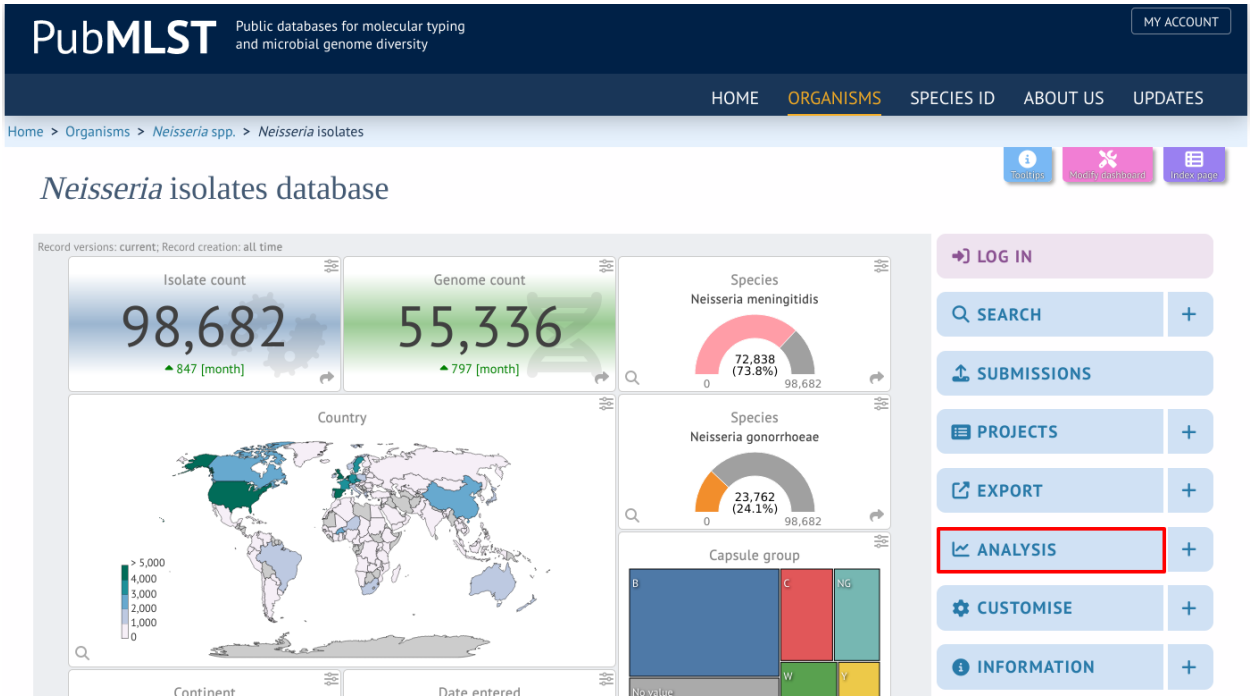
Groups can get very large, where linked profiles form sub-groups and an attempt is made to depict these.



14.3 Codon usage

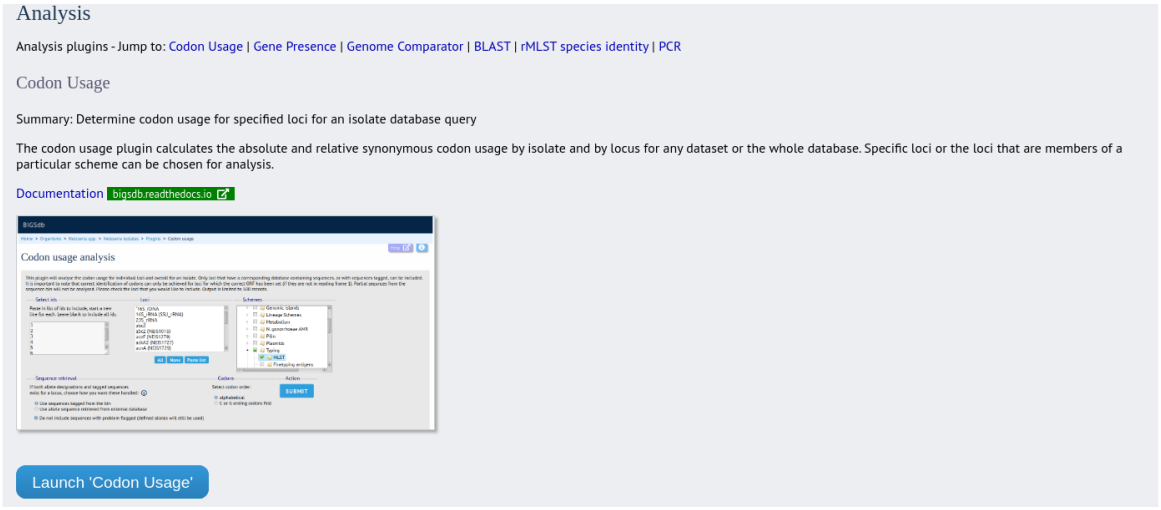
The codon usage plugin for isolate databases calculates the absolute and relative synonymous codon usage by isolate and by locus.

The function can be accessed by selecting the 'Analysis' section on the main contents page.

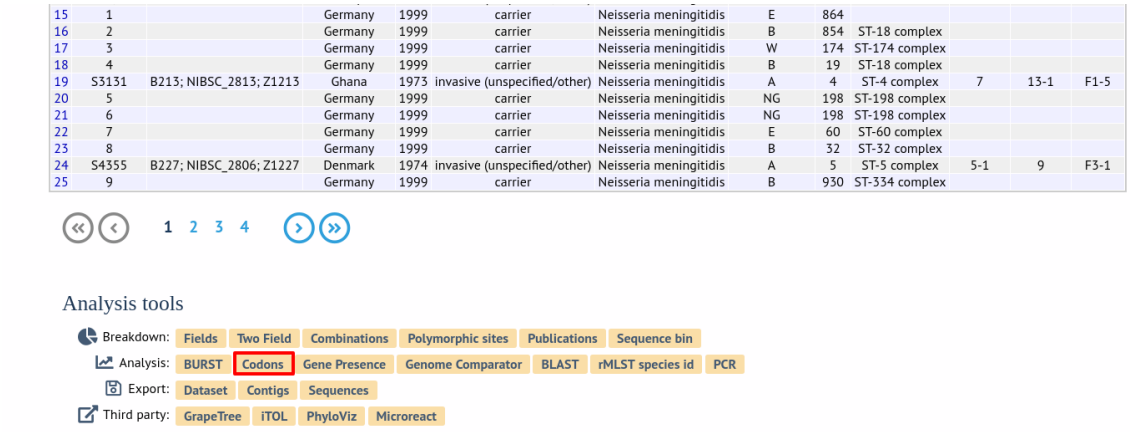


Jump to the 'Analysis' category, follow the link to 'Codon Usage', then click 'Launch Codon Usage'.

The function can be selected by clicking the 'Codon usage' link in the Analysis section of the main contents page.

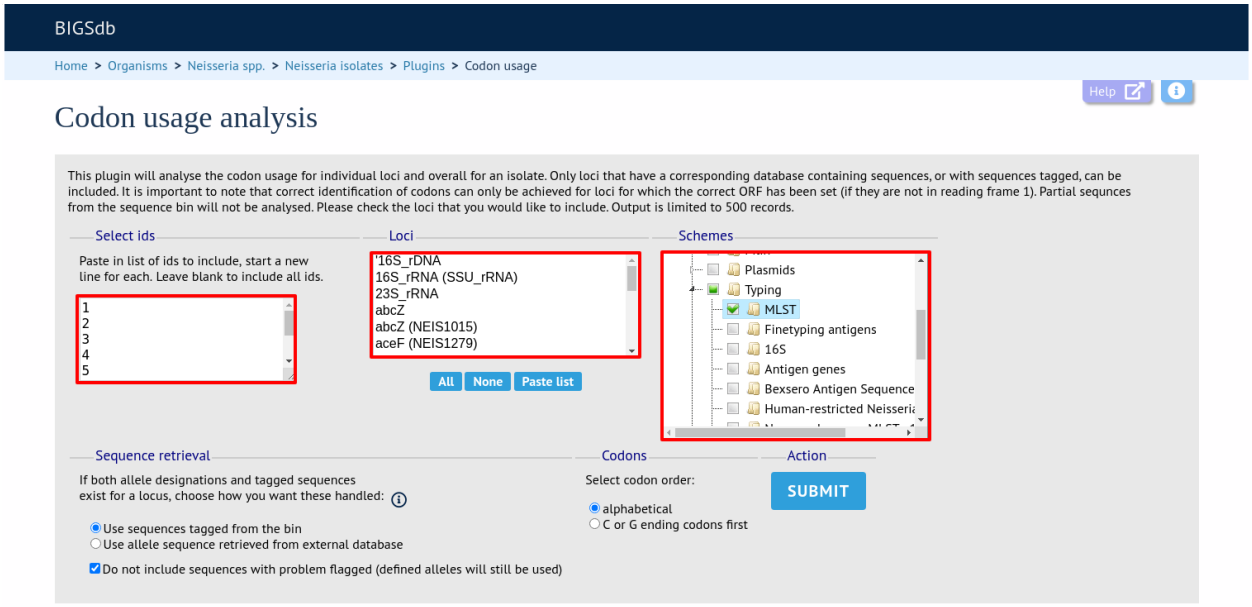


Alternatively, it can be accessed following a query by clicking the 'Codons' button in the Analysis list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.



15	1		Germany	1999	carrier	Neisseria meningitidis	E	864					
16	2		Germany	1999	carrier	Neisseria meningitidis	B	854	ST-18 complex				
17	3		Germany	1999	carrier	Neisseria meningitidis	W	174	ST-174 complex				
18	4		Germany	1999	carrier	Neisseria meningitidis	B	19	ST-18 complex				
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5	
20	5		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex				
21	6		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex				
22	7		Germany	1999	carrier	Neisseria meningitidis	E	60	ST-60 complex				
23	8		Germany	1999	carrier	Neisseria meningitidis	B	32	ST-32 complex				
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1	
25	9		Germany	1999	carrier	Neisseria meningitidis	B	930	ST-334 complex				

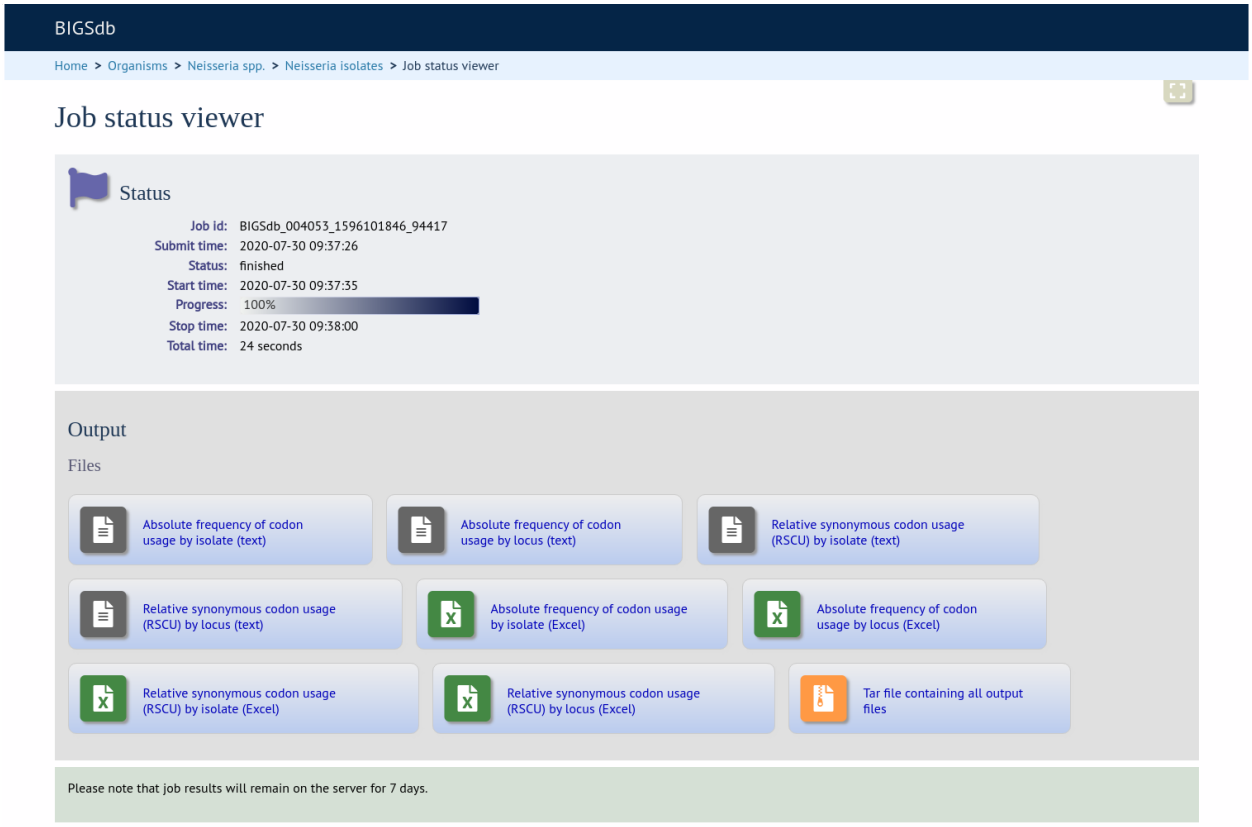
Enter the ids of the isolate records to analyse - these will be already entered if you accessed the plugin following a query. Select the loci you would like to analyse, either from the dropdown loci list, and/or by selecting one or more schemes.



Click submit. The job will be submitted to the queue and will start running shortly.

Output files will be created in both tab-delimited text and Excel formats for the following:

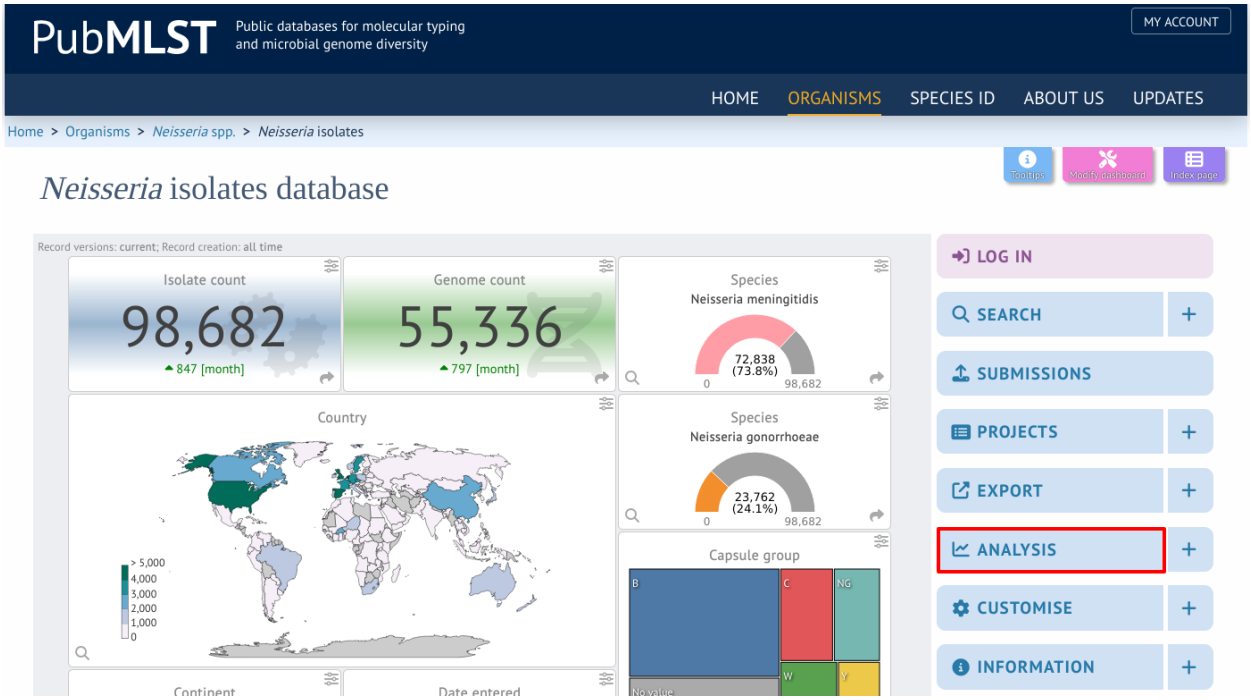
- Absolute frequency of codon usage by isolate
- Absolute frequency of codon usage by locus
- Relative synonymous codon usage by isolate
- Relative synonymous codon usage by locus



14.4 Field breakdown

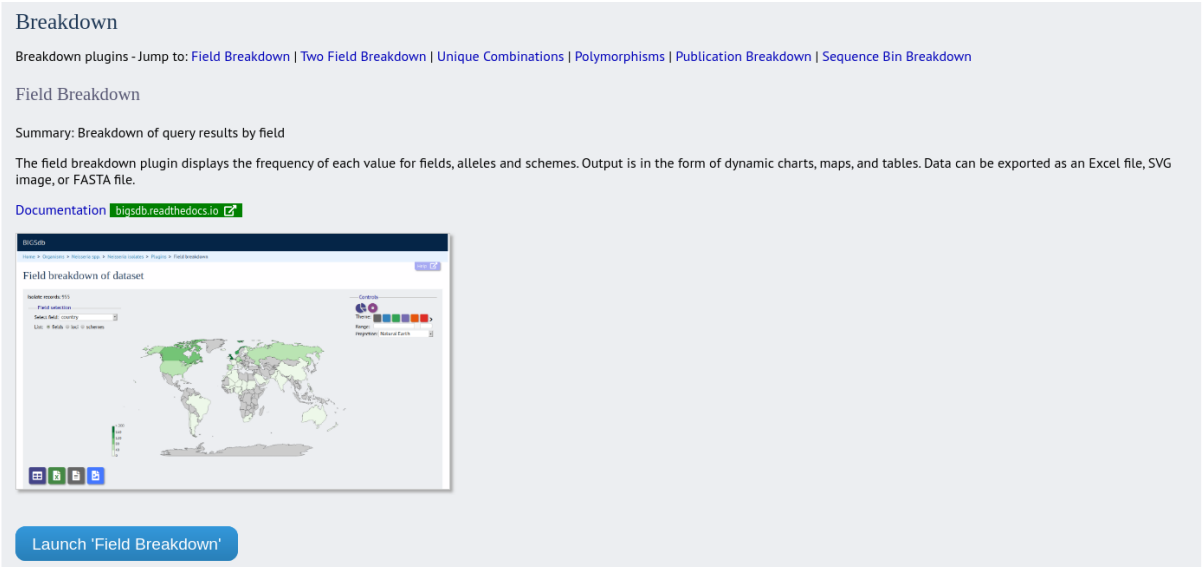
The field breakdown plugin for isolate databases displays the frequency of each value for fields, alleles and schemes.

The function can be accessed by selecting the 'Analysis' section on the main contents page.

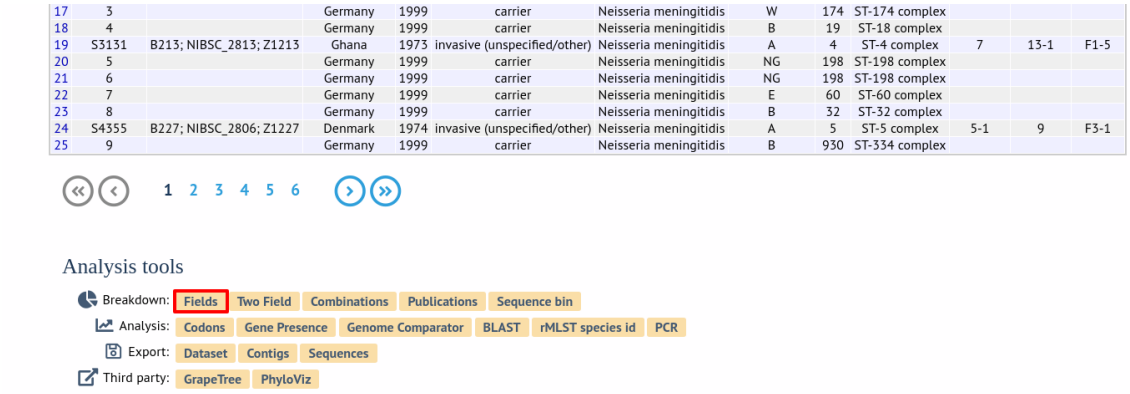


In the 'Breakdown' category, follow the link to 'Field Breakdown', then click 'Launch Field Breakdown'.

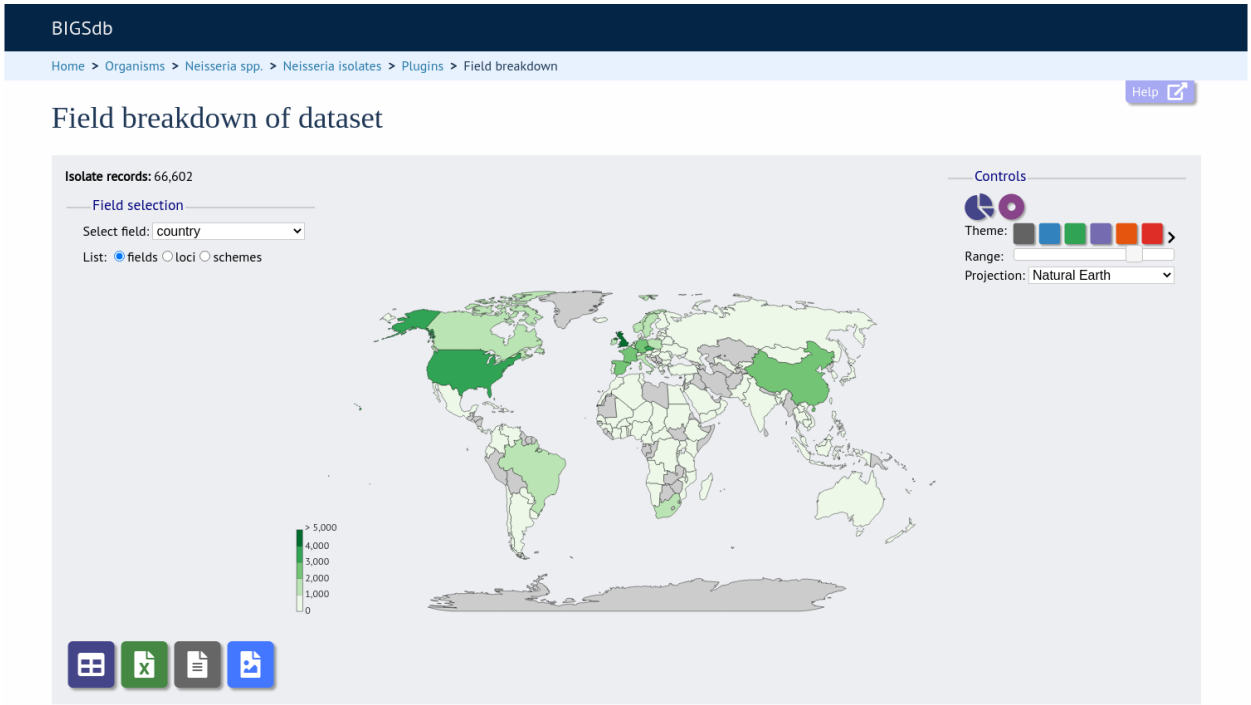
The breakdown function can be selected for the whole database by clicking the 'Single field' link in the Breakdown section of the main contents page.



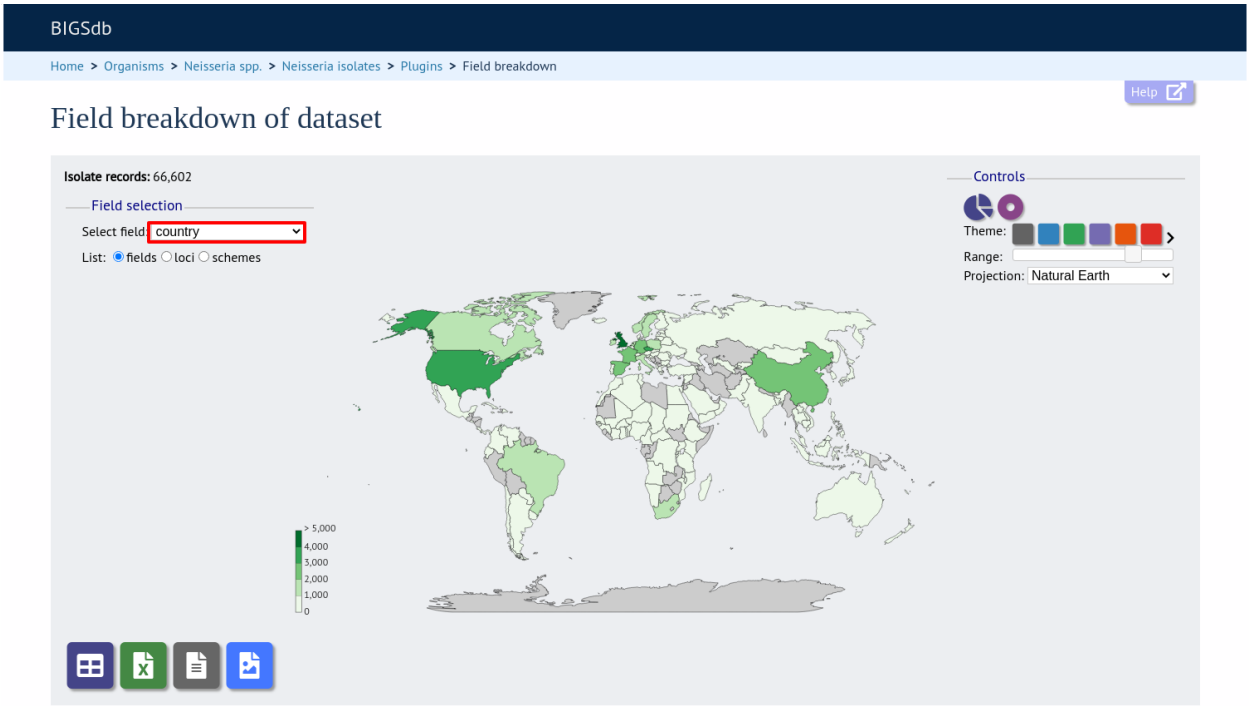
Alternatively, a breakdown can be displayed of the dataset returned from a query by clicking the 'Fields' button in the Breakdown list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.



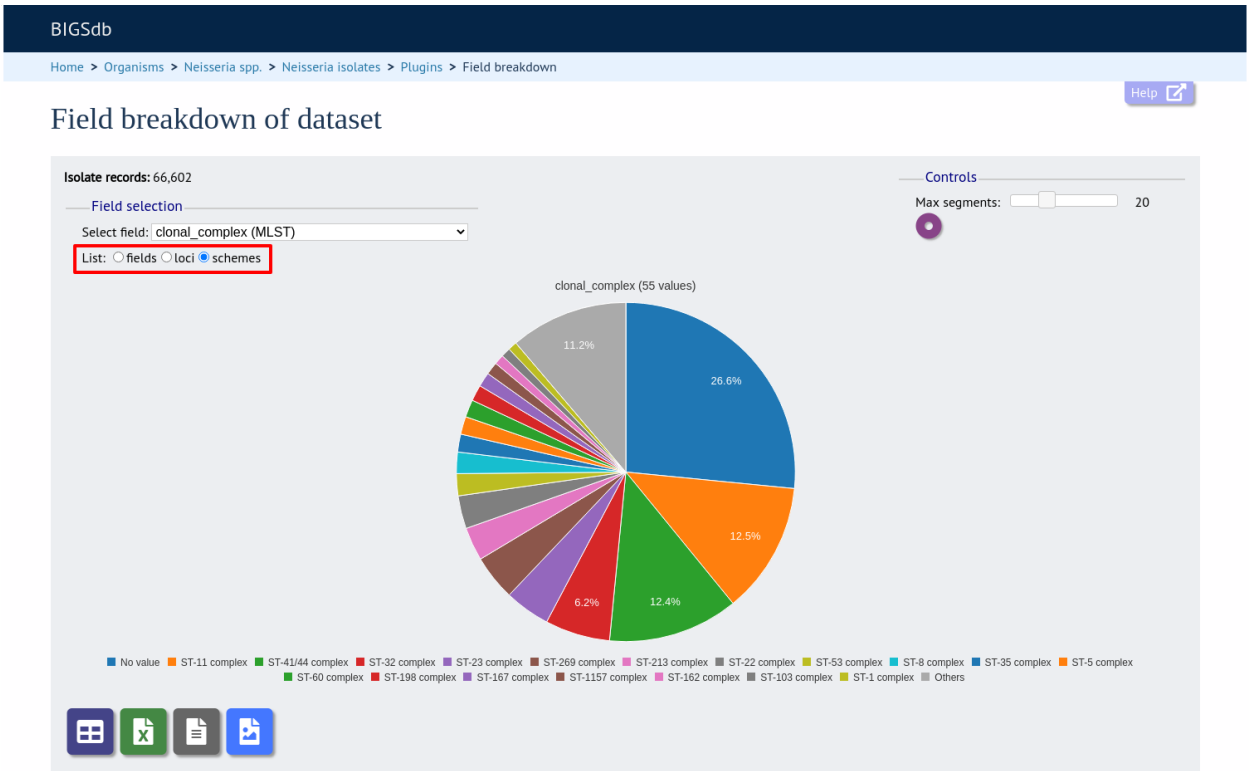
A chart will be displayed for the first field. Depending on the field type, this may be either a world map (for country or continent fields), pie chart, or bar chart.



Other fields can be chosen by selecting them in the dropdown list box.



You can also breakdown loci and schemes by clicking the appropriate button. This will re-populate the dropdown list.

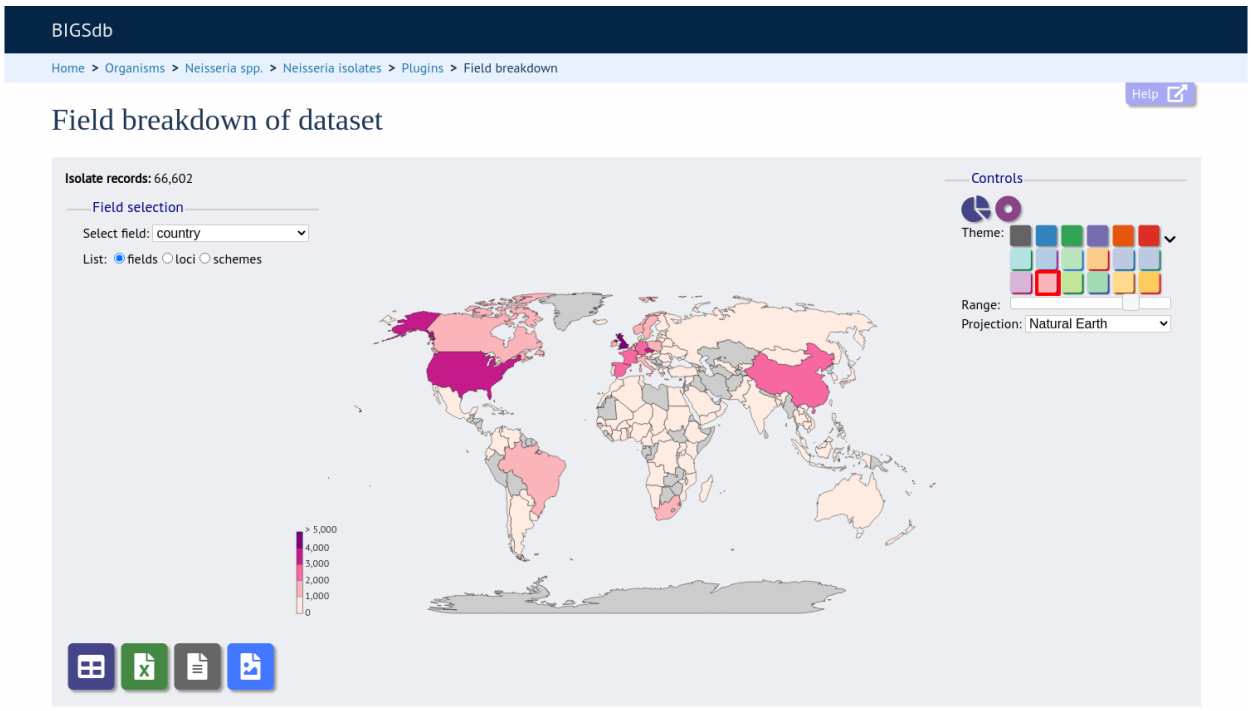


The charts are dynamic and you can manipulate some aspects of them using controls shown on the screen.

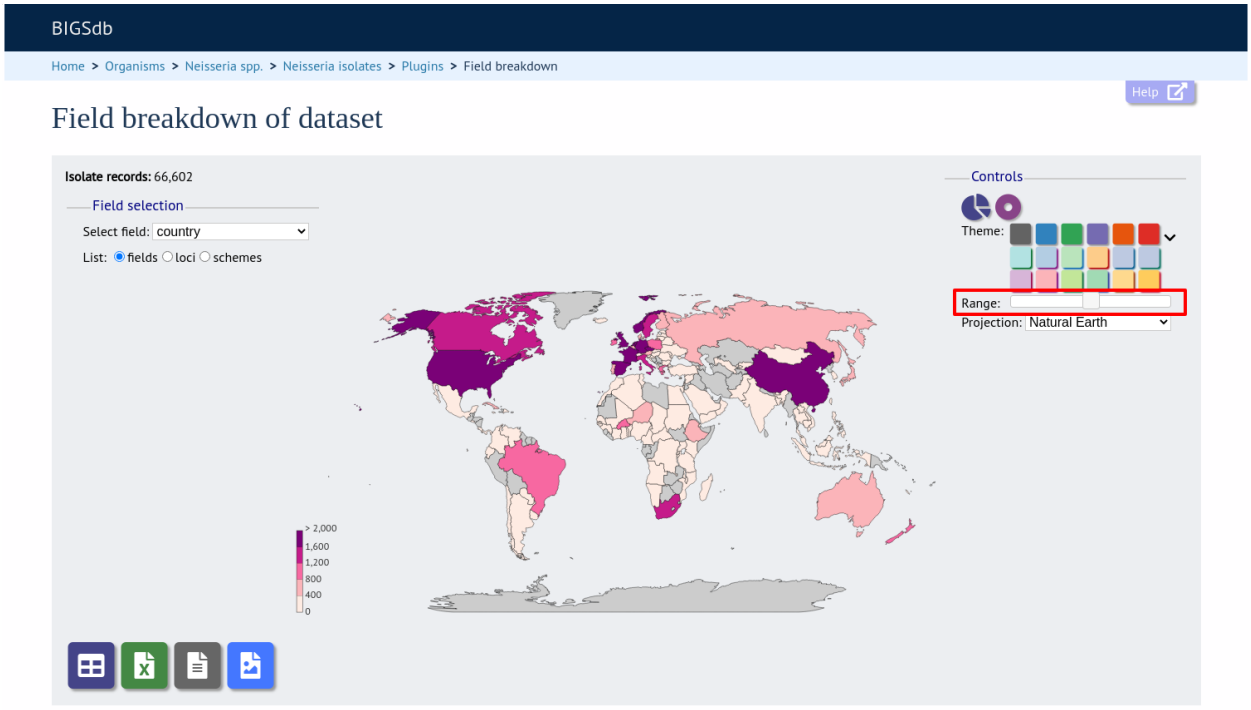
14.4.1 Maps

World maps are shown for country and continent fields (provided standardized country names are used in the database). The maps can be modified in a number of ways.

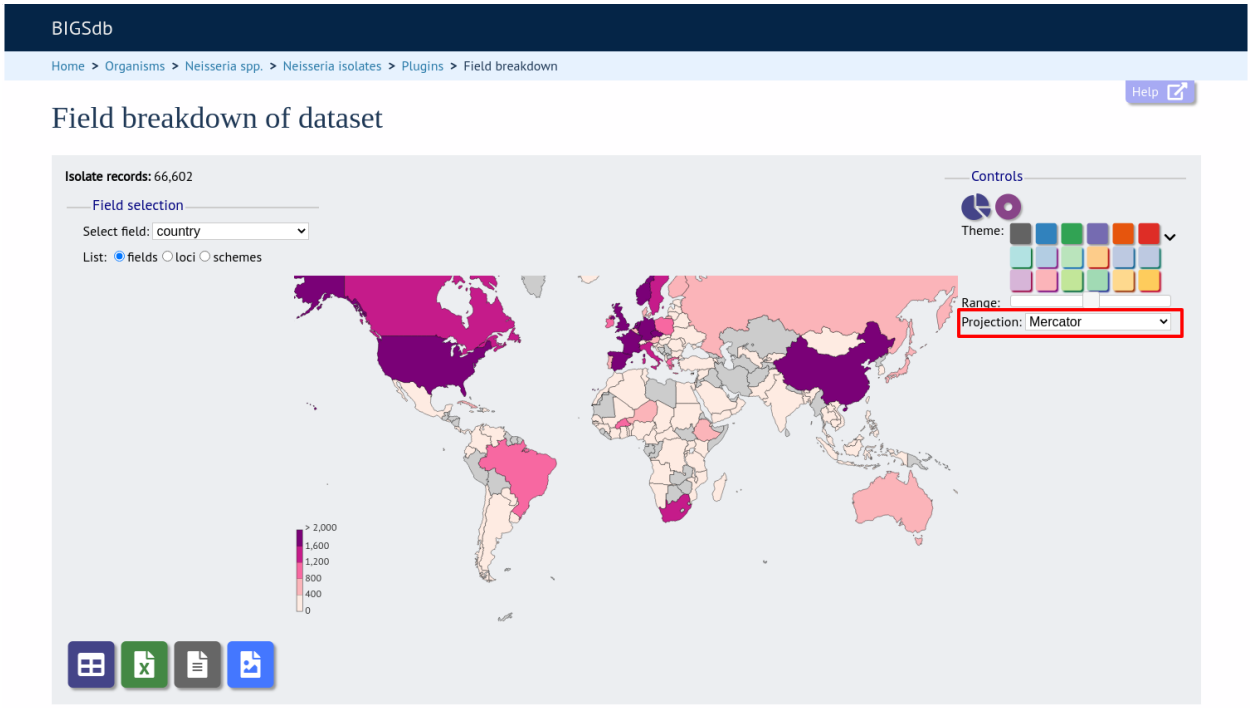
The colour theme can be changed by clicked the appropriate coloured square. Additional themes are available by clicking the '>' link.



The range that is used to decide the colour boundaries can be changed by using the range slider.



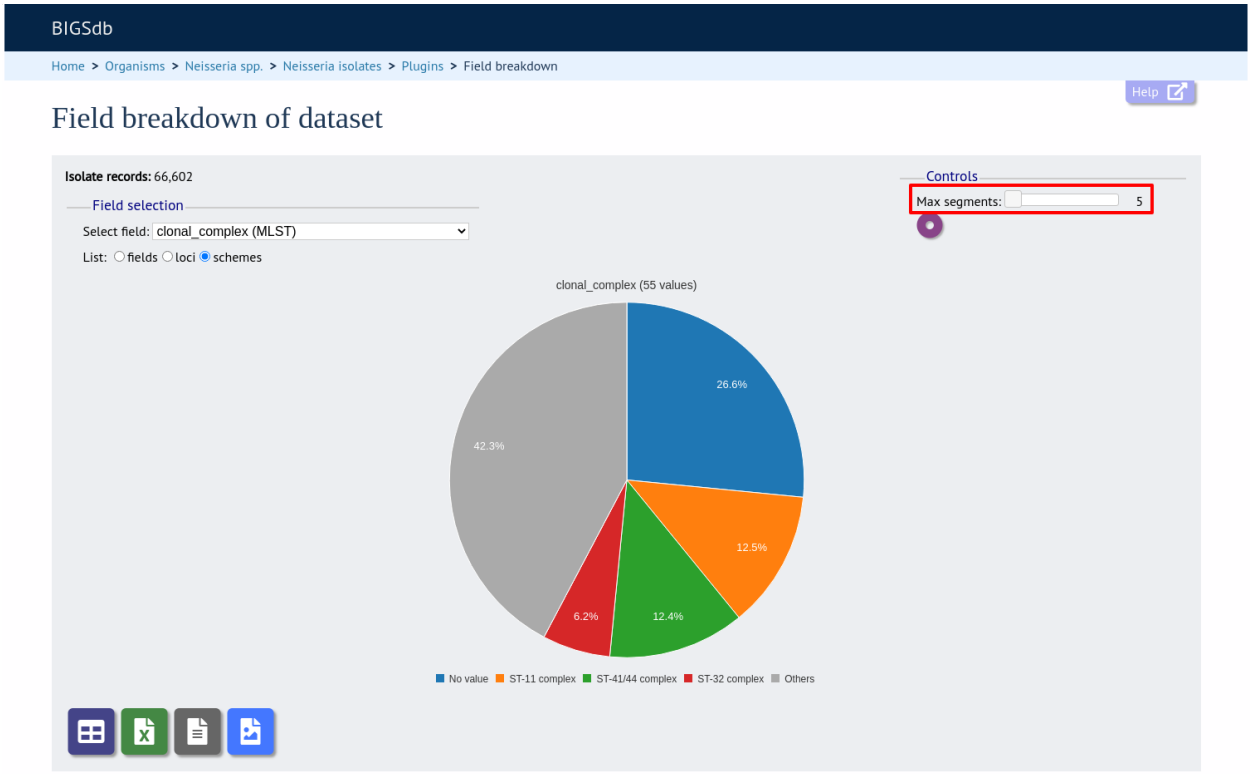
Finally the map projection can be changed. The default 'Natural Earth' provides a reasonable display for most latitudes but you may prefer others such as 'Mercator'.



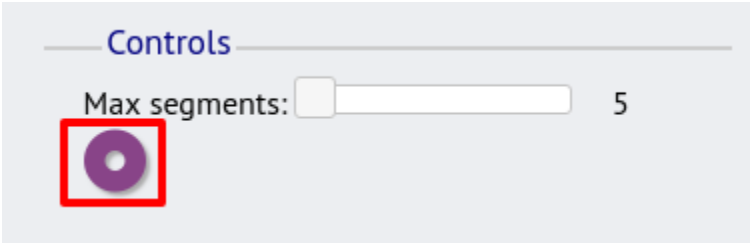
To see the same data as a pie chart, click the 'Pie' or 'Donut' icons.

14.4.2 Pie charts

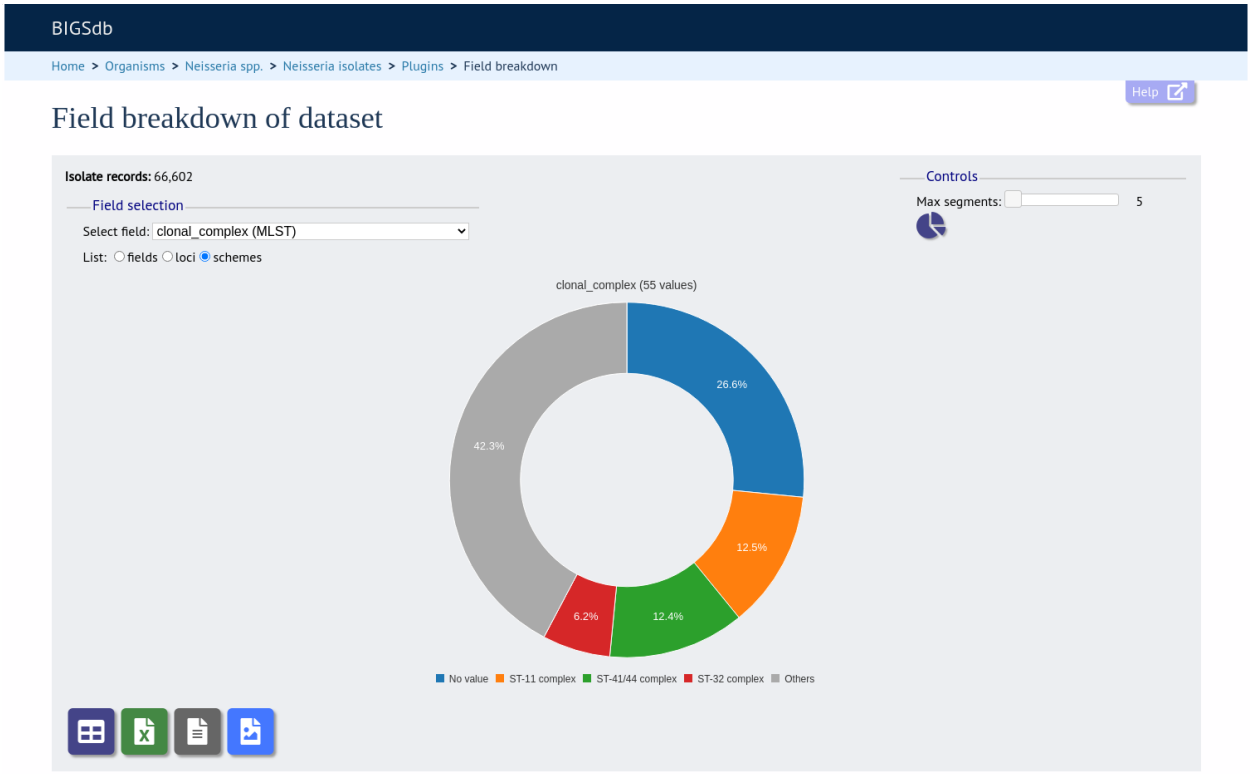
The maximum number of segments shown can be modified by sliding the 'Max segments' control. Low frequency values will be grouped in to a segment called 'Others'.



The chart can be transformed in to a donut chart by clicking the donut icon.



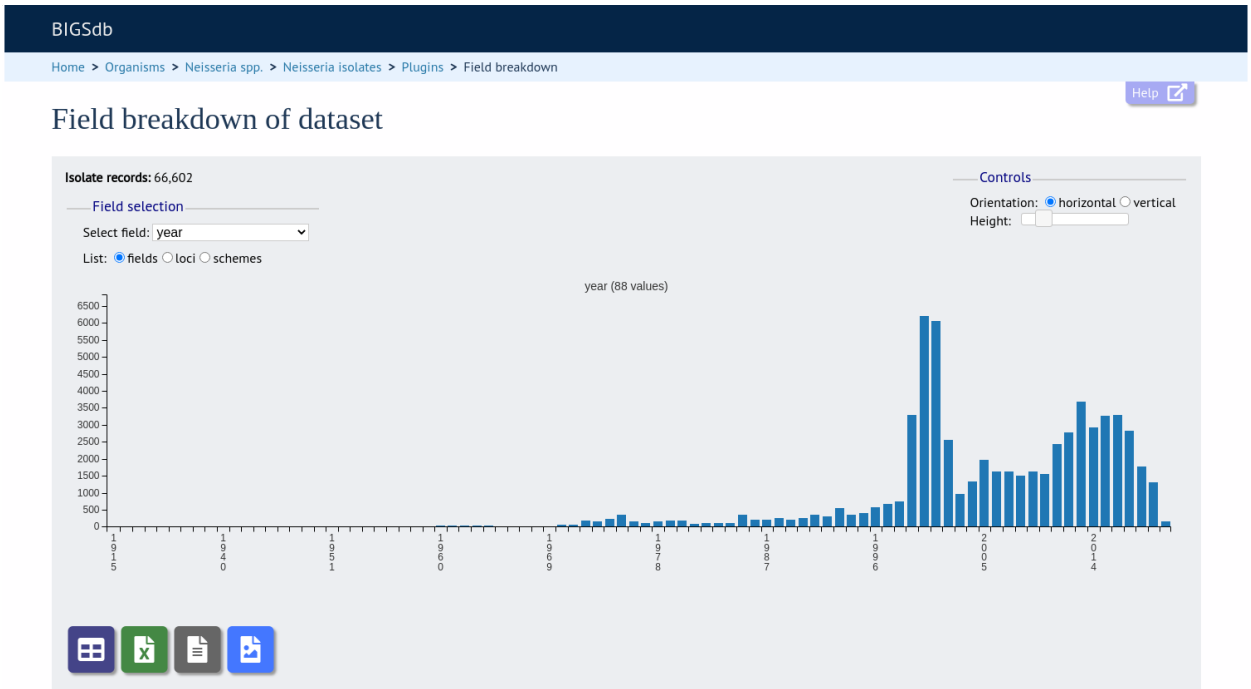
The icon changes to a pie chart image (clicking this will return to the pie chart).



Values can also be removed from the analysis by clicking their label in the legend below the chart. The percentages of the other values will be recalculated. Clicking the label again will re-add the value.

14.4.3 Bar charts

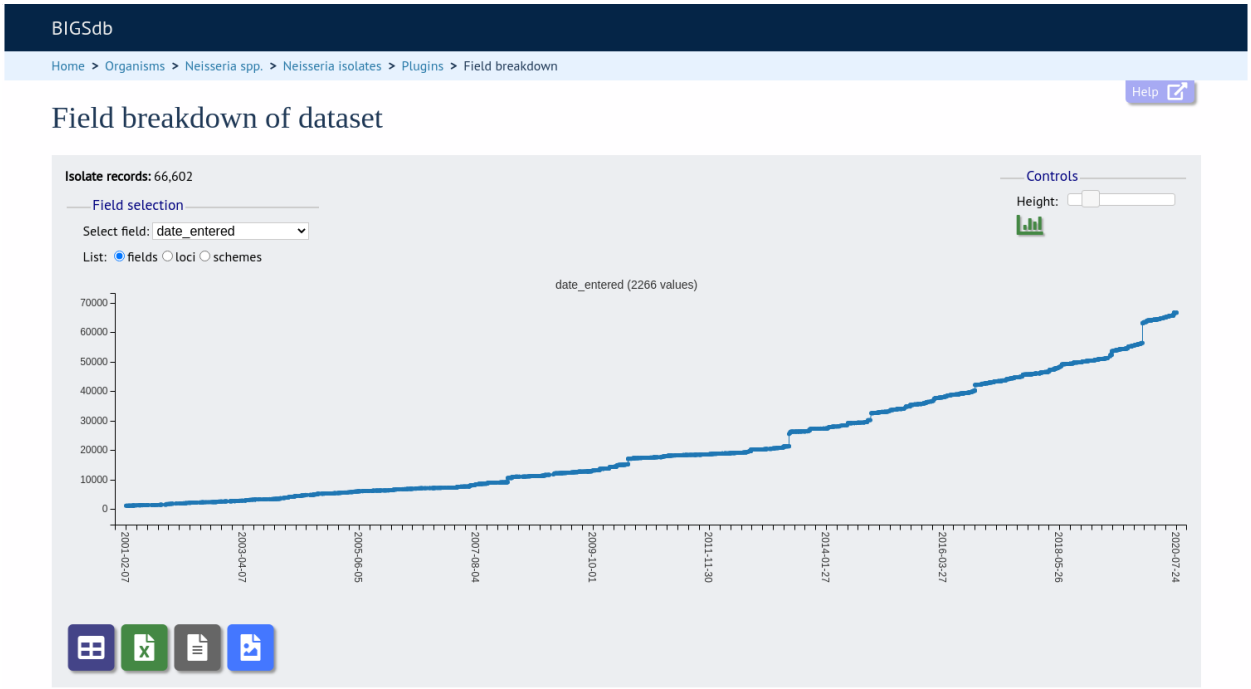
Integer fields will be displayed as a bar chart.



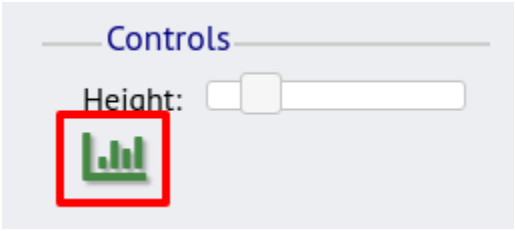
You can modify the height and the orientation of the chart using the controls.

14.4.4 Line charts

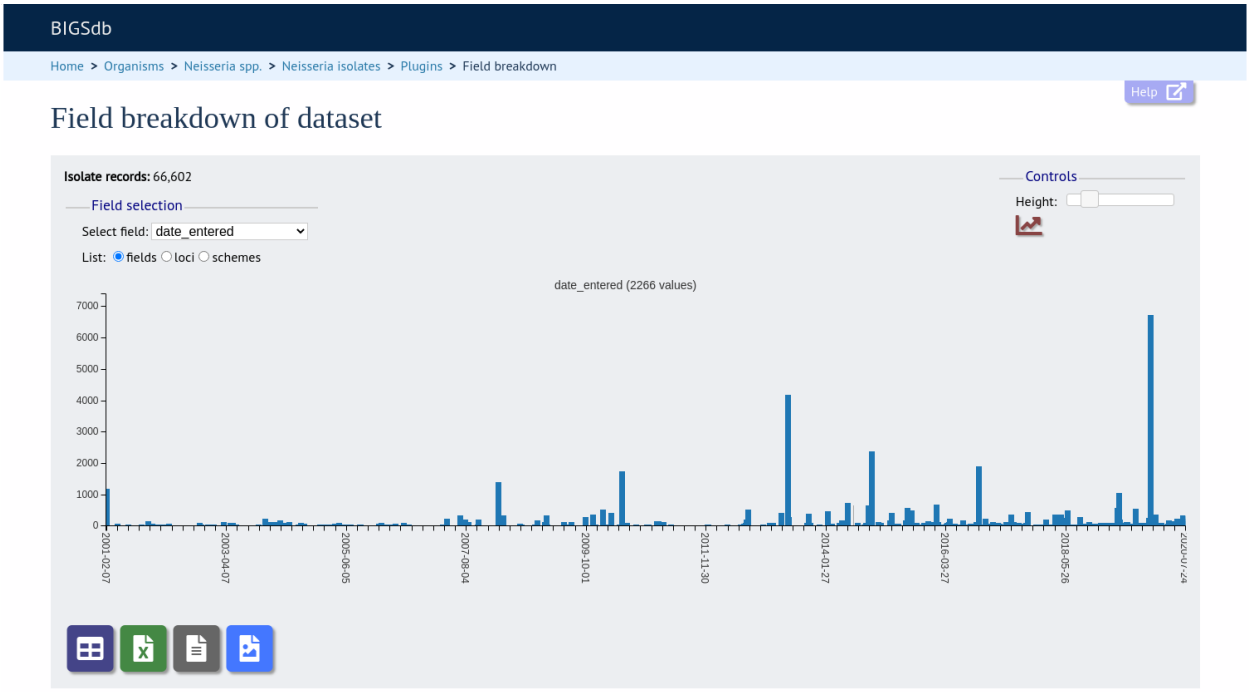
Date fields will be displayed as a line chart. By default this shows the cumulative values.



The chart can be converted in to a bar chart showing discrete values by clicking the bar chart icon.



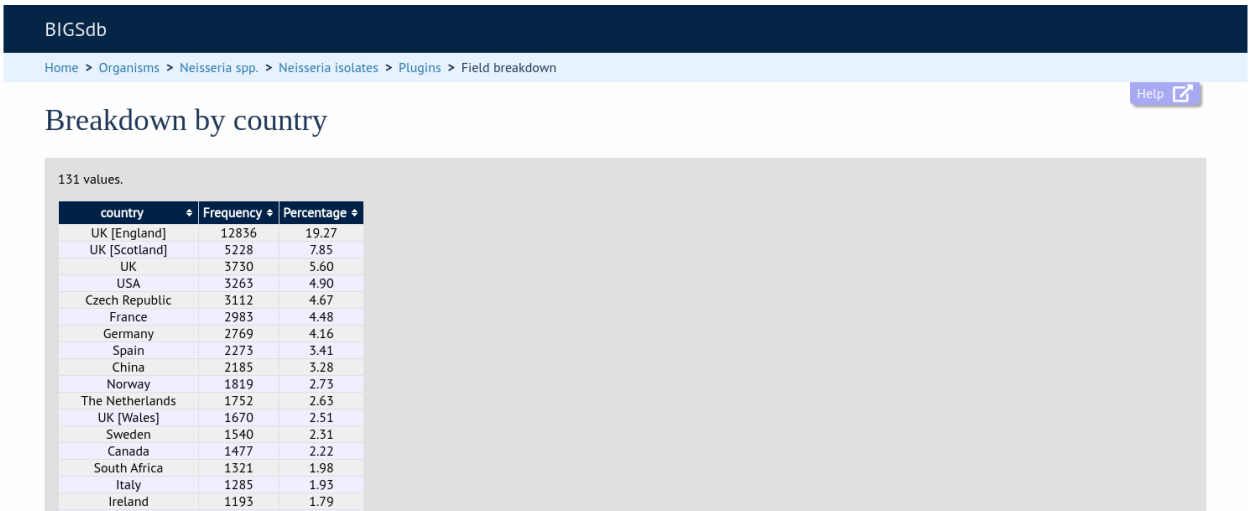
The icon changes to a line chart image (clicking this will return to the line chart).



14.4.5 Summary tables

The field breakdown can be displayed as a summary table containing values and percentages of all values. This can be selected by clicking the table icon below the displayed chart.

The table can be re-ordered by clicking any of the headings.



country	Frequency	Percentage
UK [England]	12836	19.27
UK [Scotland]	5228	7.85
UK	3730	5.60
USA	3263	4.90
Czech Republic	3112	4.67
France	2983	4.48
Germany	2769	4.16
Spain	2273	3.41
China	2185	3.28
Norway	1819	2.73
The Netherlands	1752	2.63
UK [Wales]	1670	2.51
Sweden	1540	2.31
Canada	1477	2.22
South Africa	1321	1.98
Italy	1285	1.93
Ireland	1193	1.79

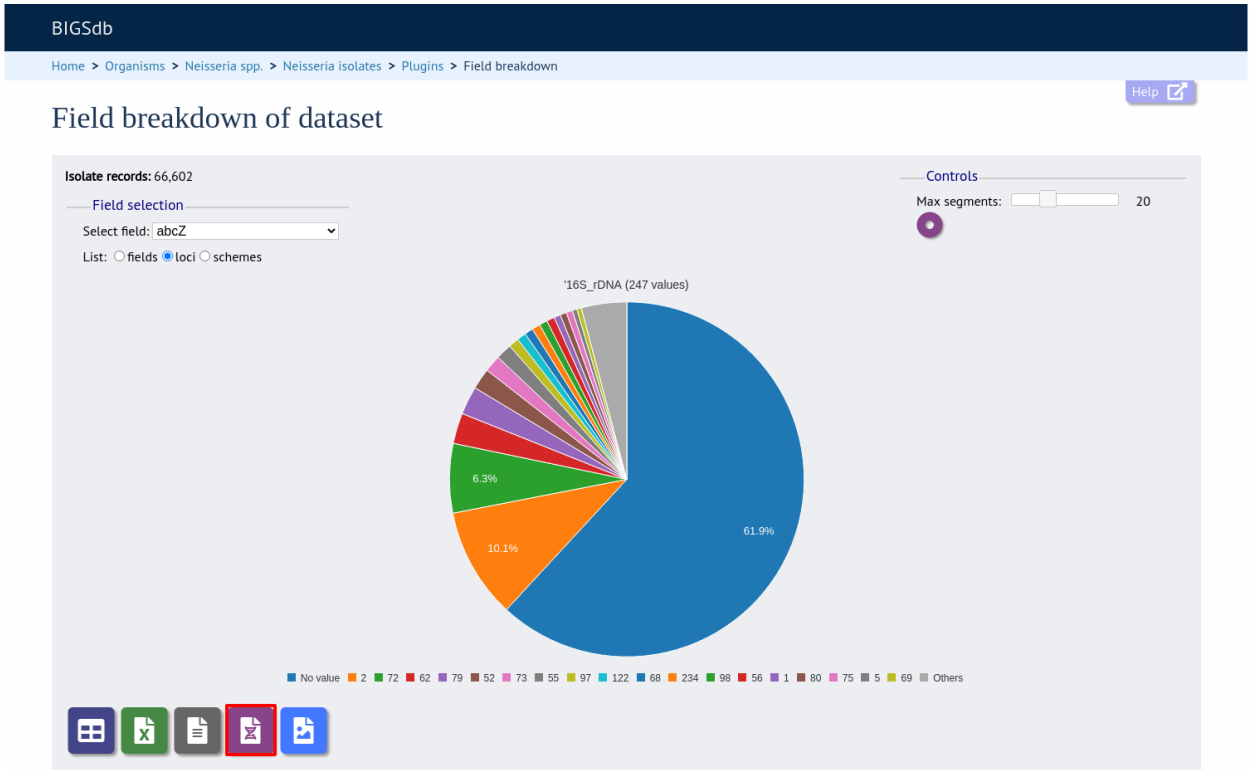
The same table can be exported as an Excel file by clicking the Excel icon.

Alternatively, it can be exported as a tab-delimited text file by clicking the text file icon.

The chart image can also be saved as a SVG file, suitable for manipulation in a graphics program.

14.4.6 Exporting allele sequences

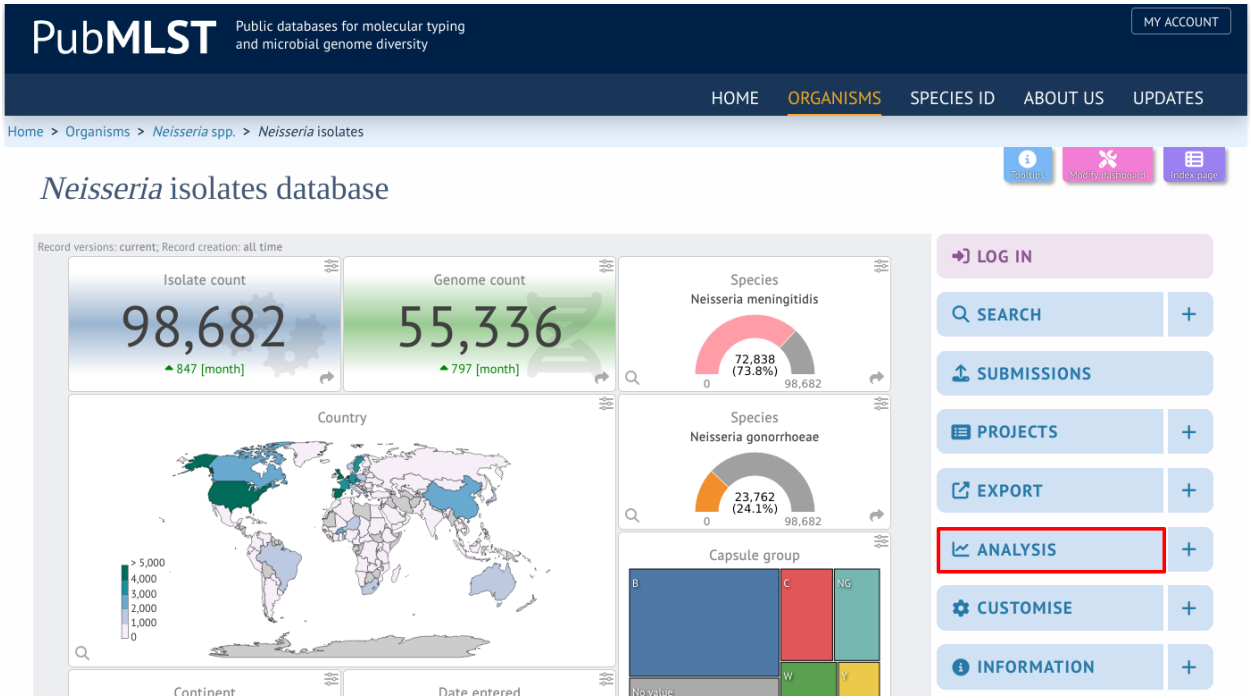
If a locus breakdown is being display, you can choose to export the allele sequences in FASTA format by clicking the FASTA file icon.



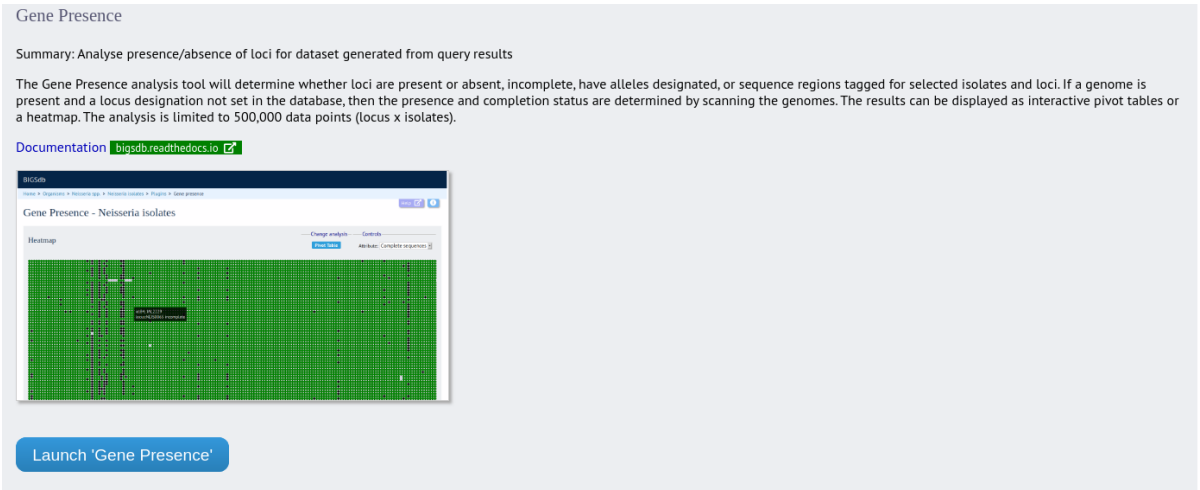
14.5 Gene Presence

The Gene Presence analysis tool will determine whether loci are present or absent, incomplete, have alleles designated, or sequence regions tagged for selected isolates and loci. If a genome is present and a locus designation not set in the database, then the presence and completion status are determined by scanning the genomes. The results can be displayed as interactive pivot tables or a heatmap. The analysis is limited to 500,000 data points (locus x isolates).

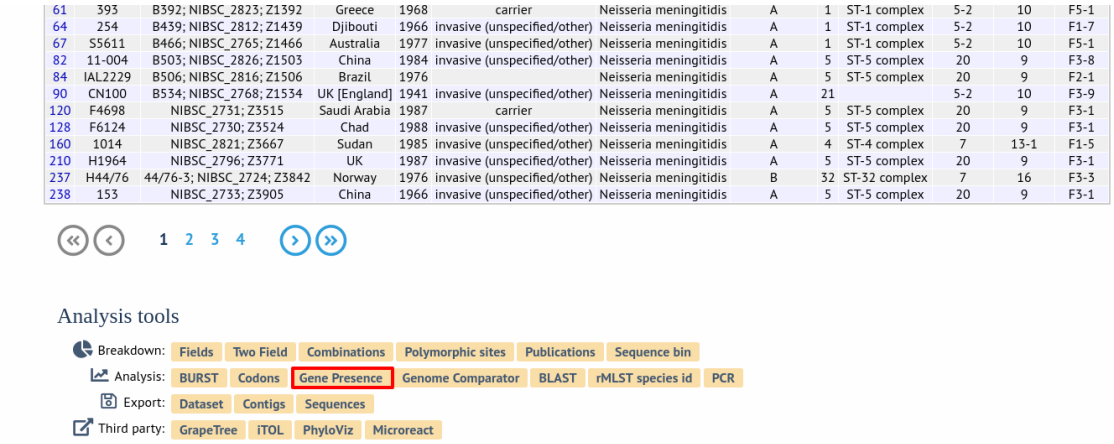
The function can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Analysis' category, follow the link to 'Gene Presence', then click 'Launch Gene Presence'.

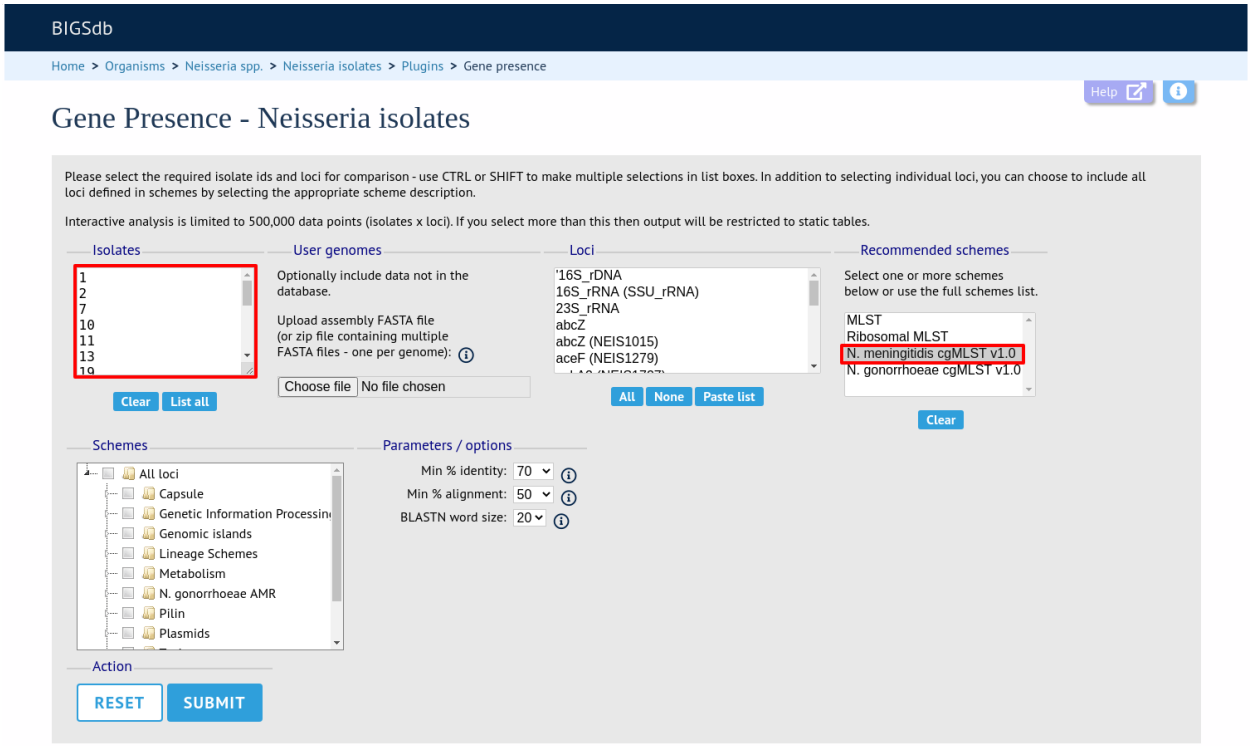


Alternatively, it can be accessed following a query by clicking the 'Gene Presence' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the plugin interface.

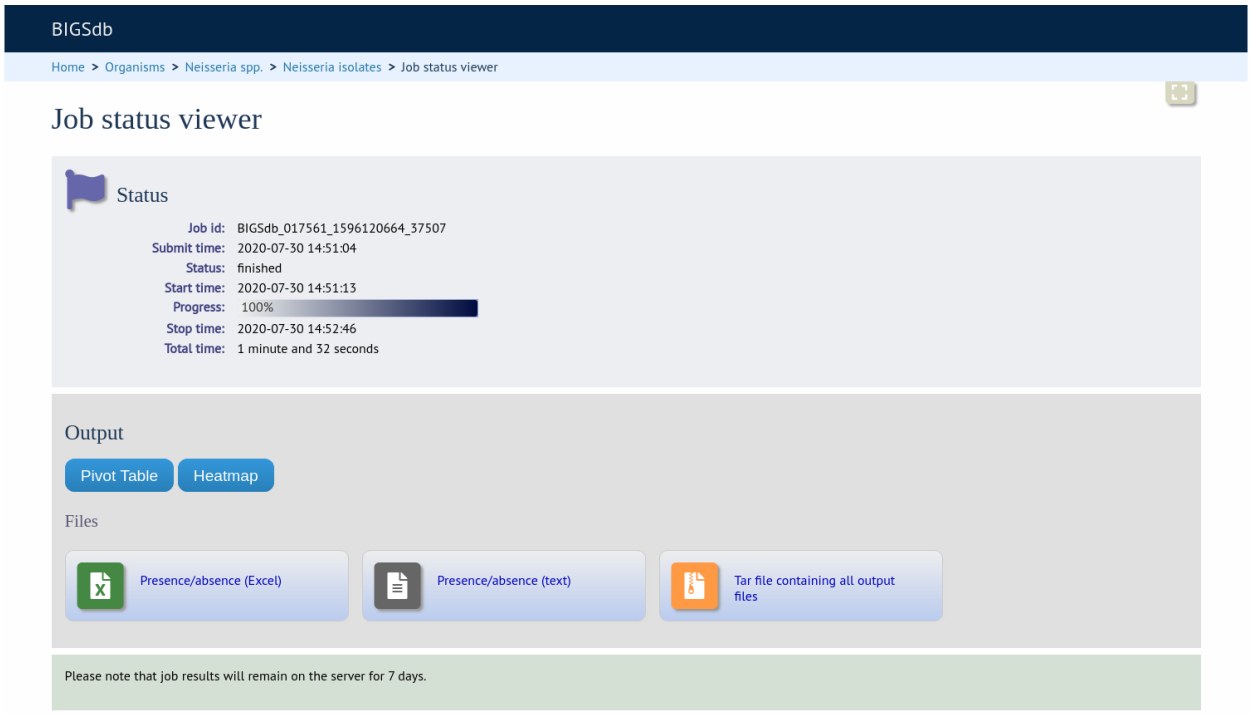


Select the isolates to include. Analysis can be performed on any selection of loci, or more conveniently, you can select a scheme in the scheme selector to include all loci belonging to that scheme. You can also select a recommended scheme if these have been defined.

The parameters of the BLAST query used to determine presence or absence can be modified, but in most cases the default options should work well. Click 'Submit' to start the analysis.

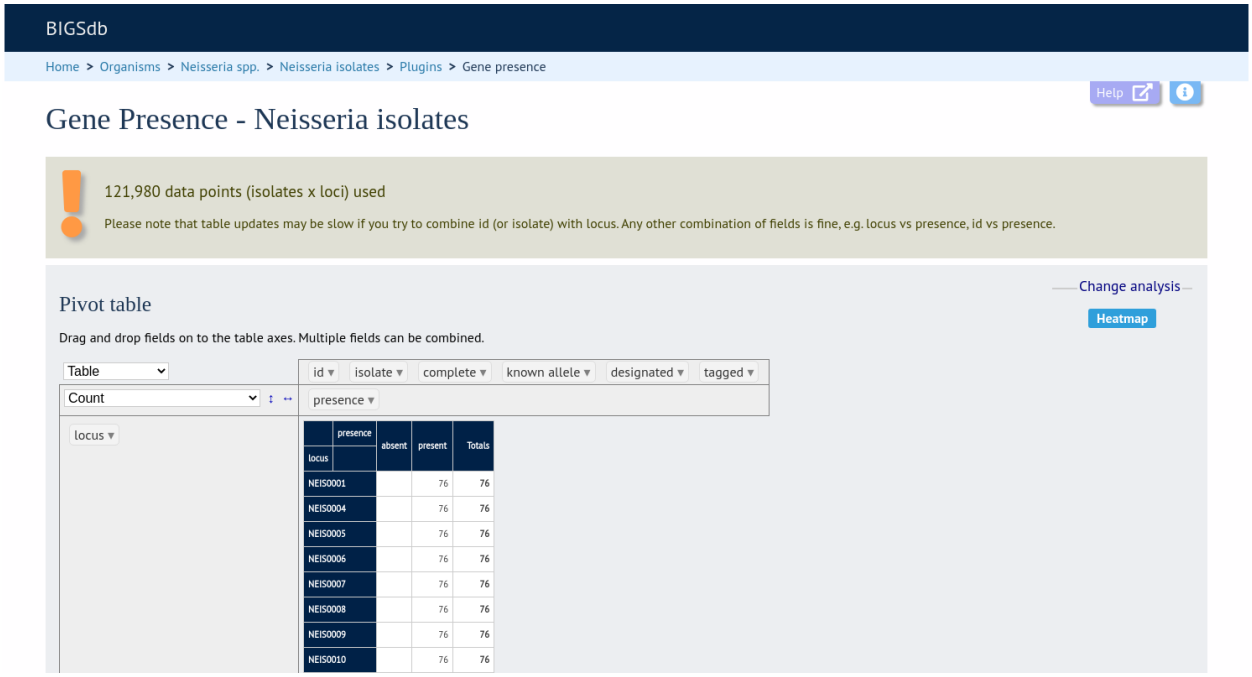


The job will be sent to the job queue. When it has finished, you will have two options to display the output: 'Pivot Table' or 'Heatmap'.

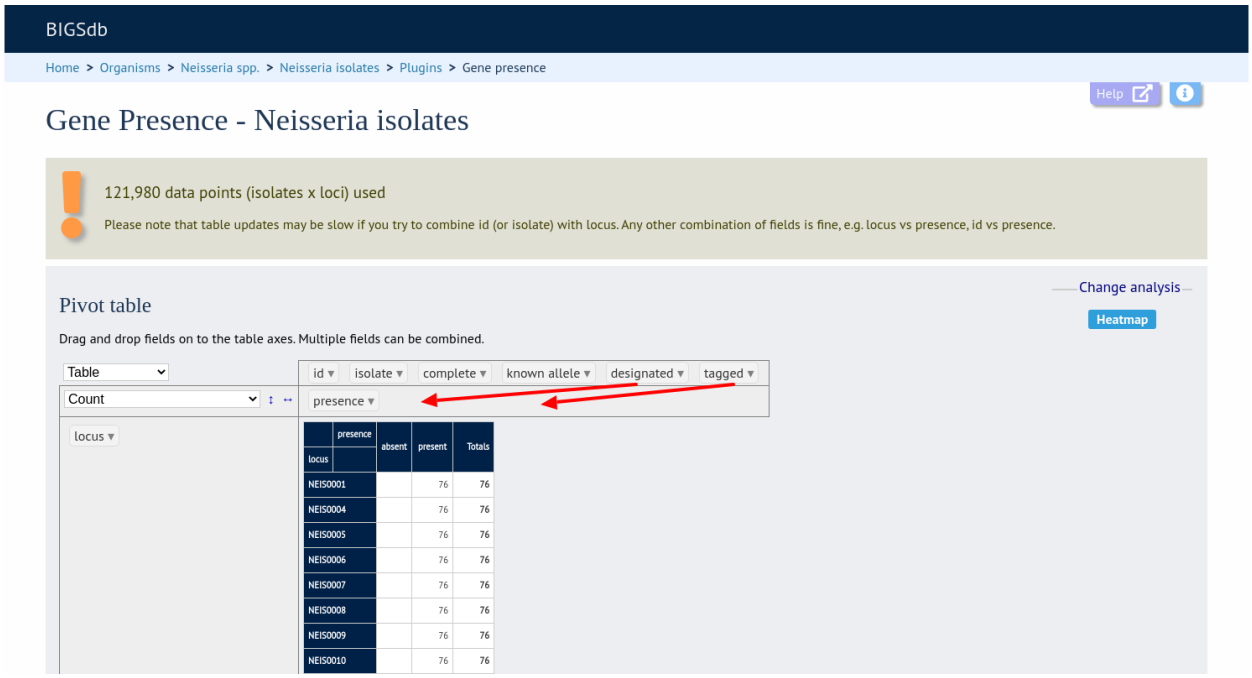


14.5.1 Pivot Table

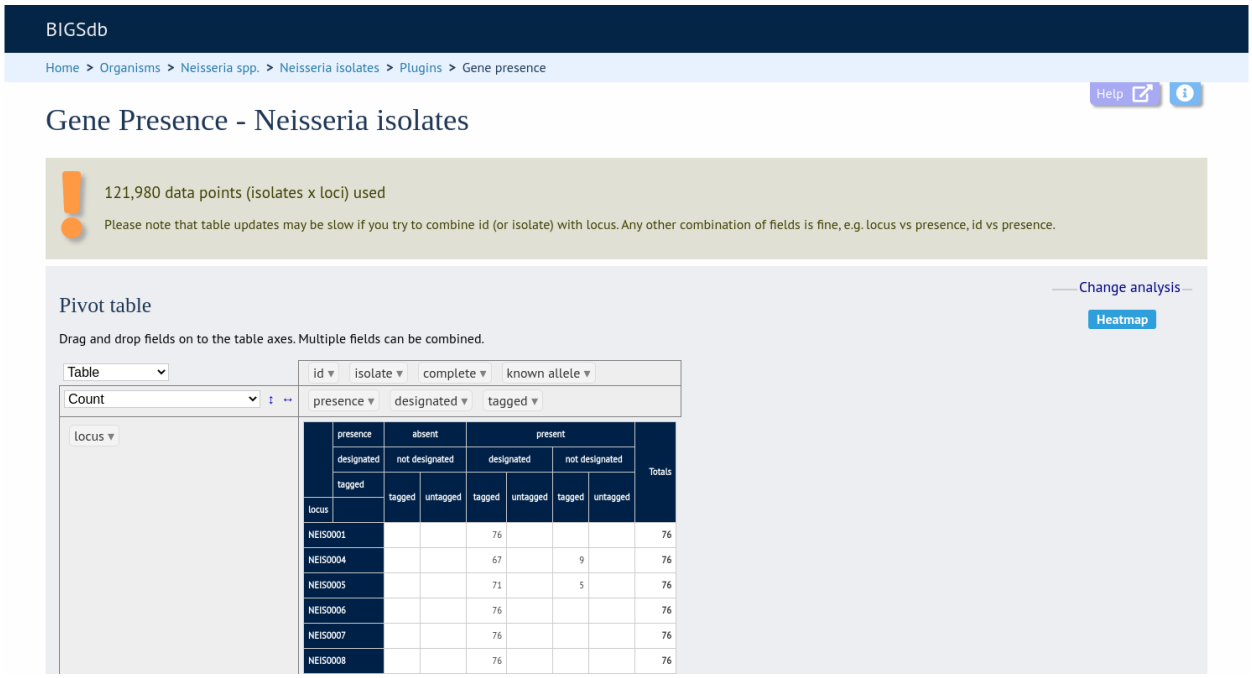
Clicking the 'Pivot Table' button will display an interactive pivot table. The default display shows the number of isolates for which each locus is present or absent.



You can break down any combination of fields by dragging them from the field area at the top of the table to either of the axes. For example, to show how many isolates have alleles designated and sequence regions tagged for each locus, drag the 'designated' and 'tagged' fields to the x-axis selector.



The table will be re-drawn including these fields.

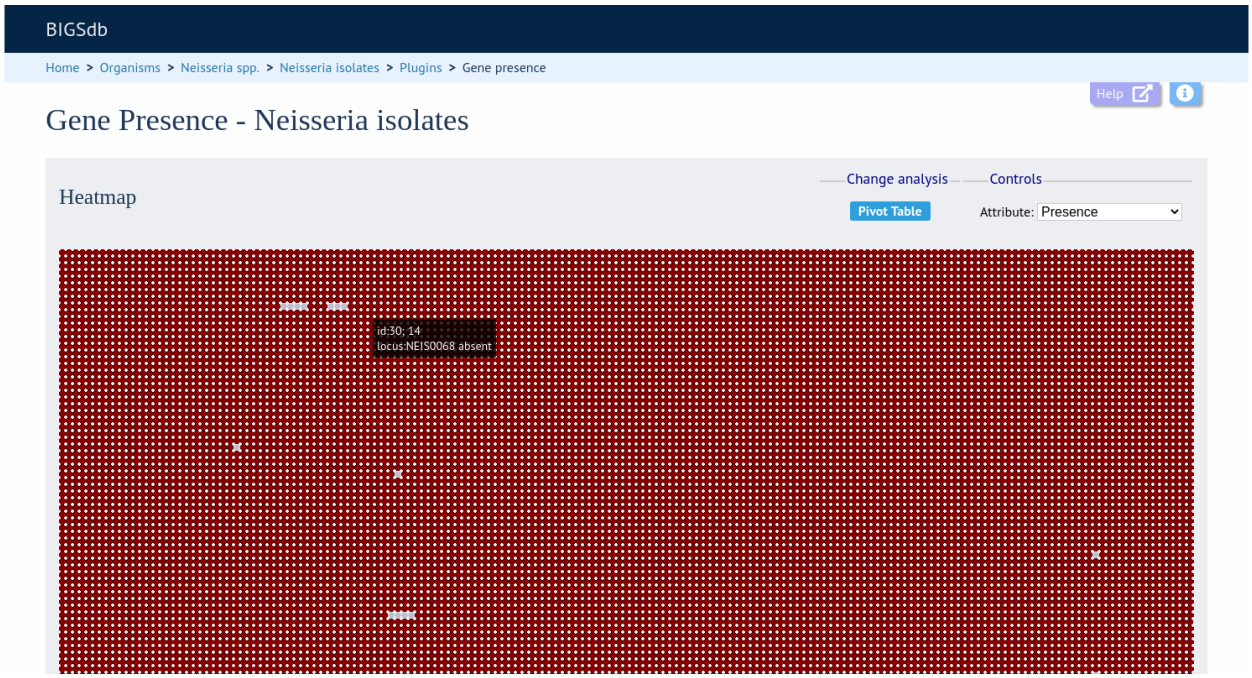


Note: If your dataset has more than 100,000 data points (locus x isolates), then be aware that combining both id (or isolate) and locus within the table will result in sluggish performace. Any other combination of fields should be fine.

14.5.2 Heatmap

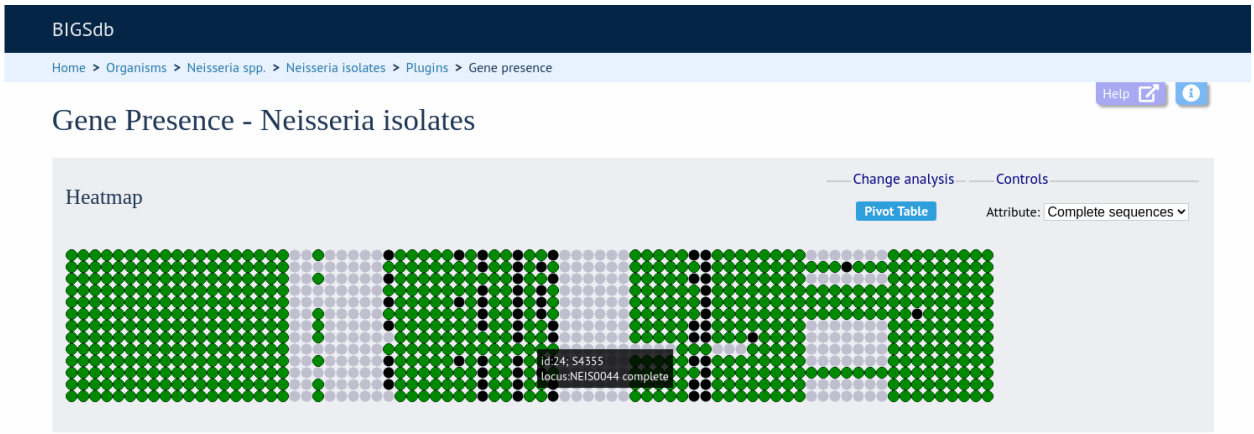
Clicking the 'Heatmap' button will display an interactive heatmap. By default the display shows the presence or absence of a locus for each isolate.

Hovering the mouse cursor or touching a region will identify the isolate and locus in a tooltip.



Change the attribute that is displayed by changing the selection in the attribute dropdown box:

The heatmap does scale to the number of records required to be displayed. If you find individual points to be too small, then choose a smaller subset of data to display:

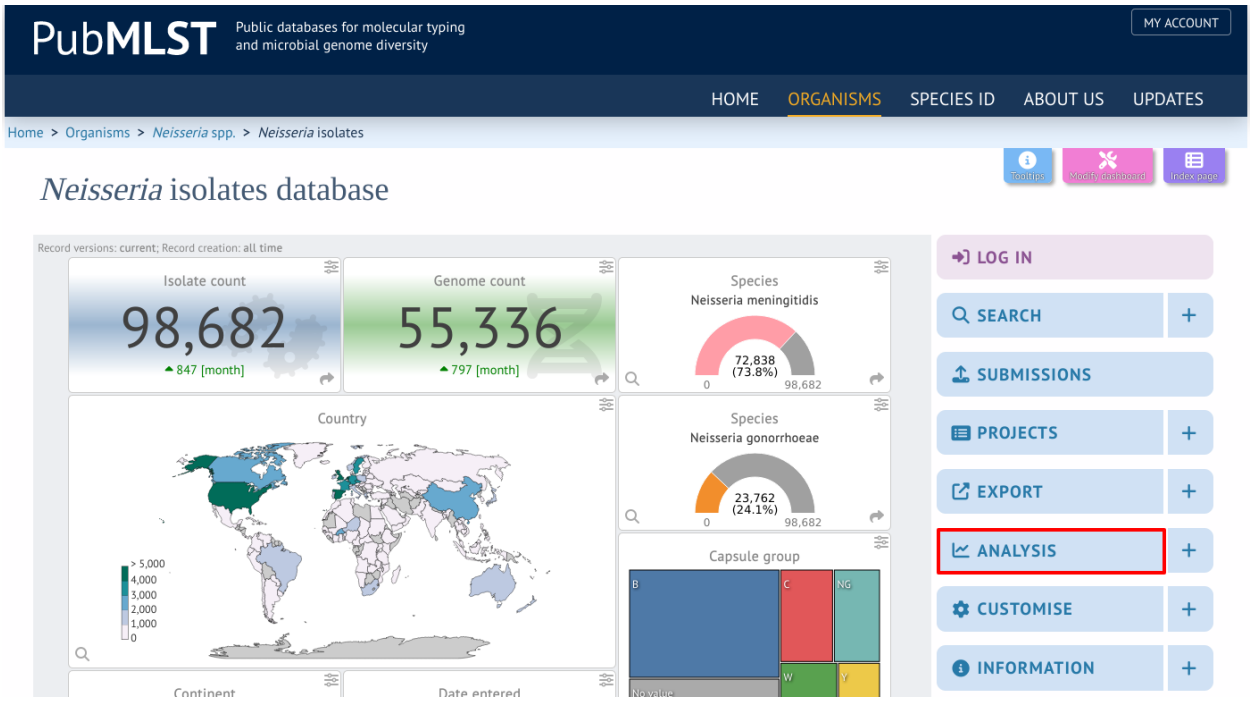


14.6 Genome comparator

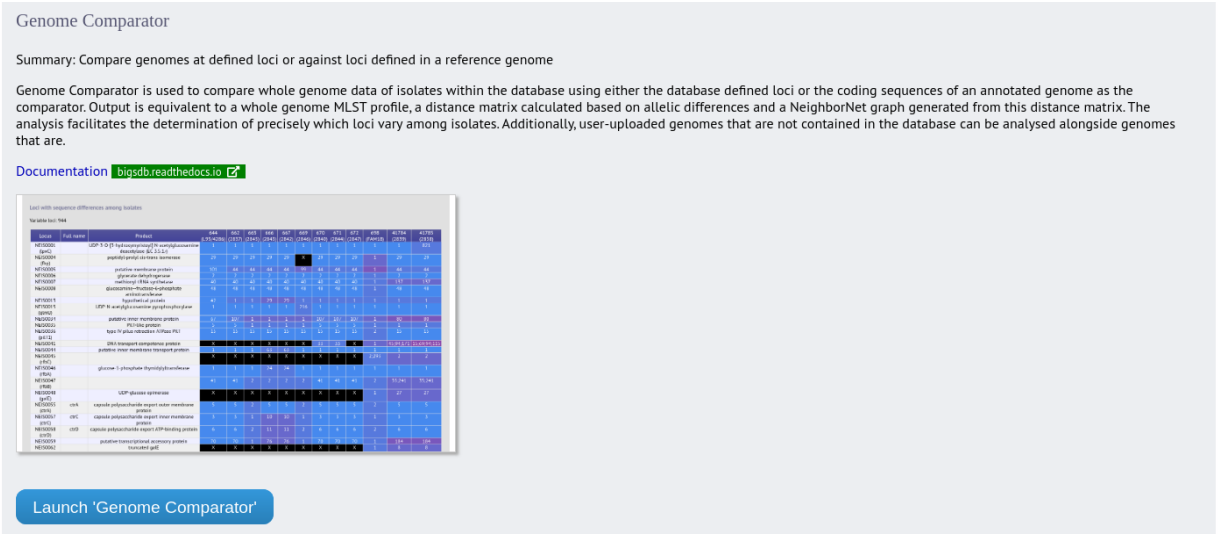
Genome Comparator is an optional plugin that can be enabled for specific databases. It is used to compare whole genome data of isolates within the database using either the database defined loci or the coding sequences of an annotated genome as the comparator.

Output is equivalent to a whole genome MLST profile, a distance matrix calculated based on allelic differences and a NeighborNet graph generated from this distance matrix.

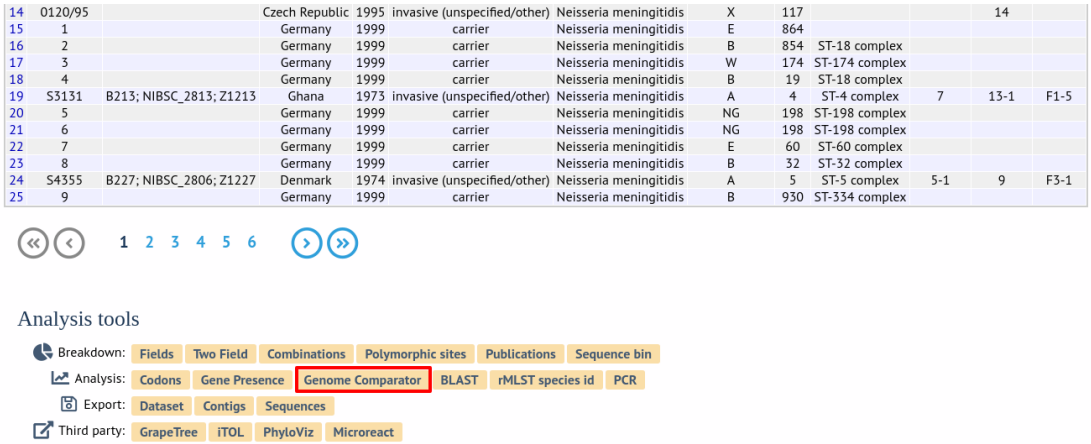
Genome Comparator can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Analysis' category, follow the link to 'Genome Comparator', then click 'Launch Genome Comparator'.

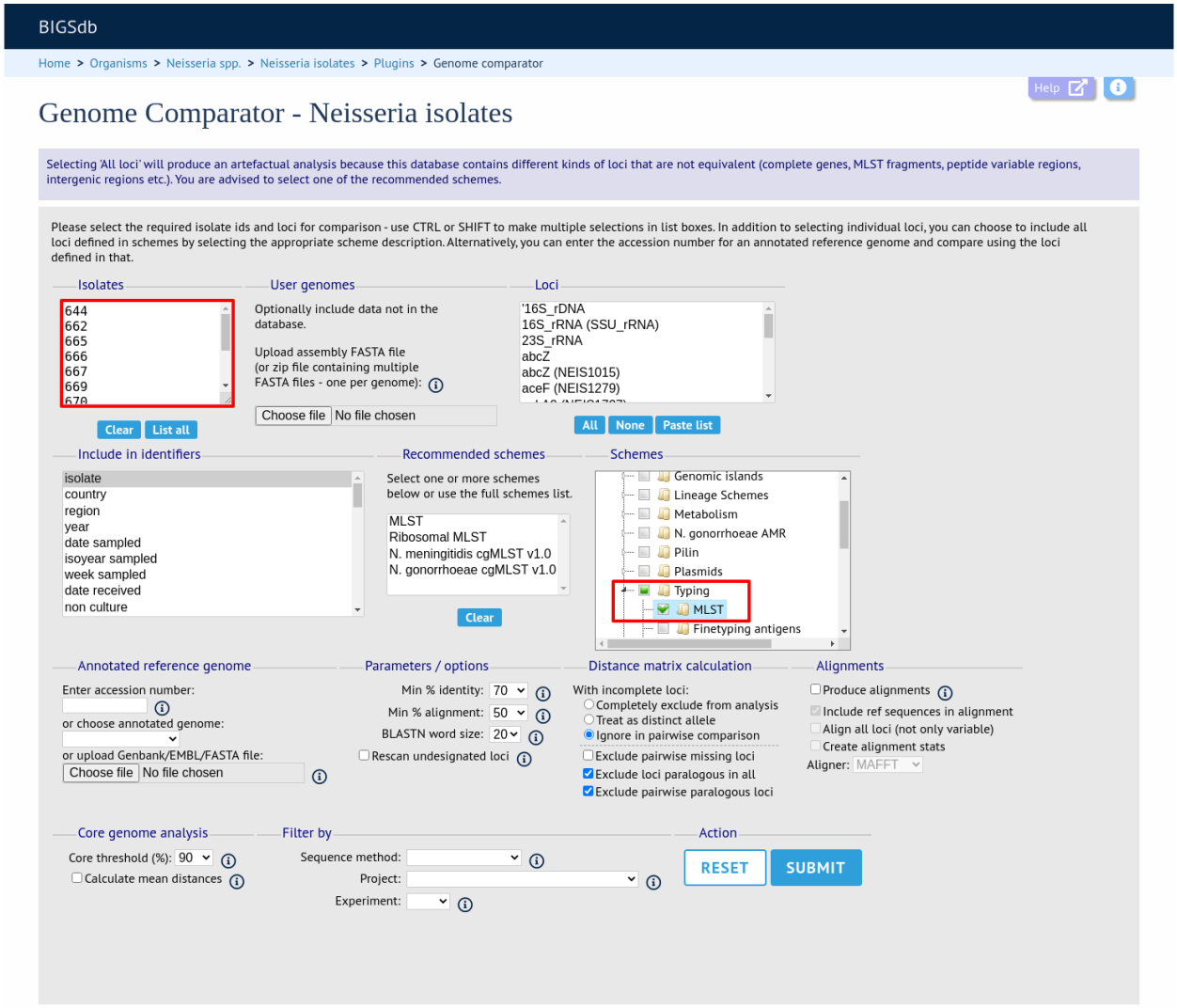


Alternatively, it can be accessed following a query by clicking the 'Genome Comparator' button at the bottom of the results table. Isolates with sequence data returned in the query will be automatically selected within the Genome Comparator interface.



14.6.1 Analysis using defined loci

Select the isolate genomes that you wish to analyse. These will either be in a dropdown list or, if there are too many in the database, a text input where a list can be entered. You can also upload your own genomes for analysis - these should be either a single file in FASTA format (if you have just one genome), or a zip file containing multiple FASTA files. Select either the loci from the list or a set of schemes (either from the schemes box or from a list of recommended schemes if these have been set up). Press submit.



The job will be submitted to the job queue and will start running shortly.

There will be a series of tables displaying variable loci, colour-coded to indicate allelic differences. Finally, there will be links to a distance matrix which can be loaded in to SplitsTree for further analysis and to a NeighborNet chart showing relatedness of isolates. Due to processing constraints on the web server, this NeighborNet is only calculated if 200 or fewer genomes are selected for analysis, but this can be generated in the stand-alone version of SplitsTree using the distance matrix if required.

BIGSdb

Home > Organisms > Neisseria spp. > Neisseria isolates > Job status viewer

Job status viewer

Status

Job id: BIGSdb_024566_1596186459_13471
Submit time: 2020-07-31 09:07:39
Status: finished
Start time: 2020-07-31 09:07:42
Progress: 100%
Stop time: 2020-07-31 09:07:46
Total time: 4 seconds

Output

Analysis against defined loci

All loci

Allele numbers are used where these have been defined, otherwise sequences will be marked as 'New#1, 'New#2' etc. Missing alleles are marked as 'X'. Incomplete alleles (located at end of contig) are marked as 'I'.

Locus	Full name	Product	644 (L93/4286)	662 (2837)	665 (2845)	666 (2843)	667 (2842)	669 (2846)	670 (2840)	671 (2844)	672 (2847)	698 (FAM18)	41784 (2839)	41785 (2838)
abcZ			2	2	2	2	2	7	2	2	2	2	2	2
adk			3	3	3	3	3	3	3	3	3	3	3	3
aroE			4	19	4	4	4	4	19	19	19	4	4	4
fumC			3	3	24	23	23	3	3	3	3	3	3	3
gdh			8	8	8	8	8	8	8	8	8	8	8	8
pdhC			4	4	4	6	6	4	4	4	4	4	4	4
pgm			6	6	6	6	6	6	6	6	6	6	6	6

Loci with sequence differences among isolates

Variable loci: 4

Locus	Full name	Product	644 (L93/4286)	662 (2837)	665 (2845)	666 (2843)	667 (2842)	669 (2846)	670 (2840)	671 (2844)	672 (2847)	698 (FAM18)	41784 (2839)	41785 (2838)
abcZ			2	2	2	2	2	7	2	2	2	2	2	2
aroE			4	19	4	4	4	4	19	19	19	4	4	4
fumC			3	3	24	23	23	3	3	3	3	3	3	3
pdhC			4	4	4	6	6	4	4	4	4	4	4	4

Exactly matching loci

These loci are identical in all isolates.

Matches: 3

Locus	Full name	Product	644 (L93/4286)	662 (2837)	665 (2845)	666 (2843)	667 (2842)	669 (2846)	670 (2840)	671 (2844)	672 (2847)	698 (FAM18)	41784 (2839)	41785 (2838)
adk			3	3	3	3	3	3	3	3	3	3	3	3
gdh			8	8	8	8	8	8	8	8	8	8	8	8
pgm			6	6	6	6	6	6	6	6	6	6	6	6

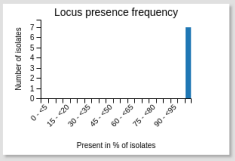
Unique strains

Unique strains: 5

Strain 1	Strain 2	Strain 3	Strain 4	Strain 5
662 (2837)	644 (L93/4286)	666 (2843)	669 (2846)	665 (2845)
670 (2840)	698 (FAM18)	667 (2842)		
671 (2844)	41784 (2839)			
672 (2847)	41785 (2838)			

Charts

Click to enlarge.

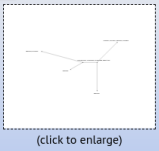


Files

Text output file

Excel format output

Splits graph (Neighbour-net; SVG format)



(click to enlarge)

Distance matrix (Nexus format) - Suitable for loading in to SplitsTree. Distances between taxa are calculated as the number of loci with different allele sequences

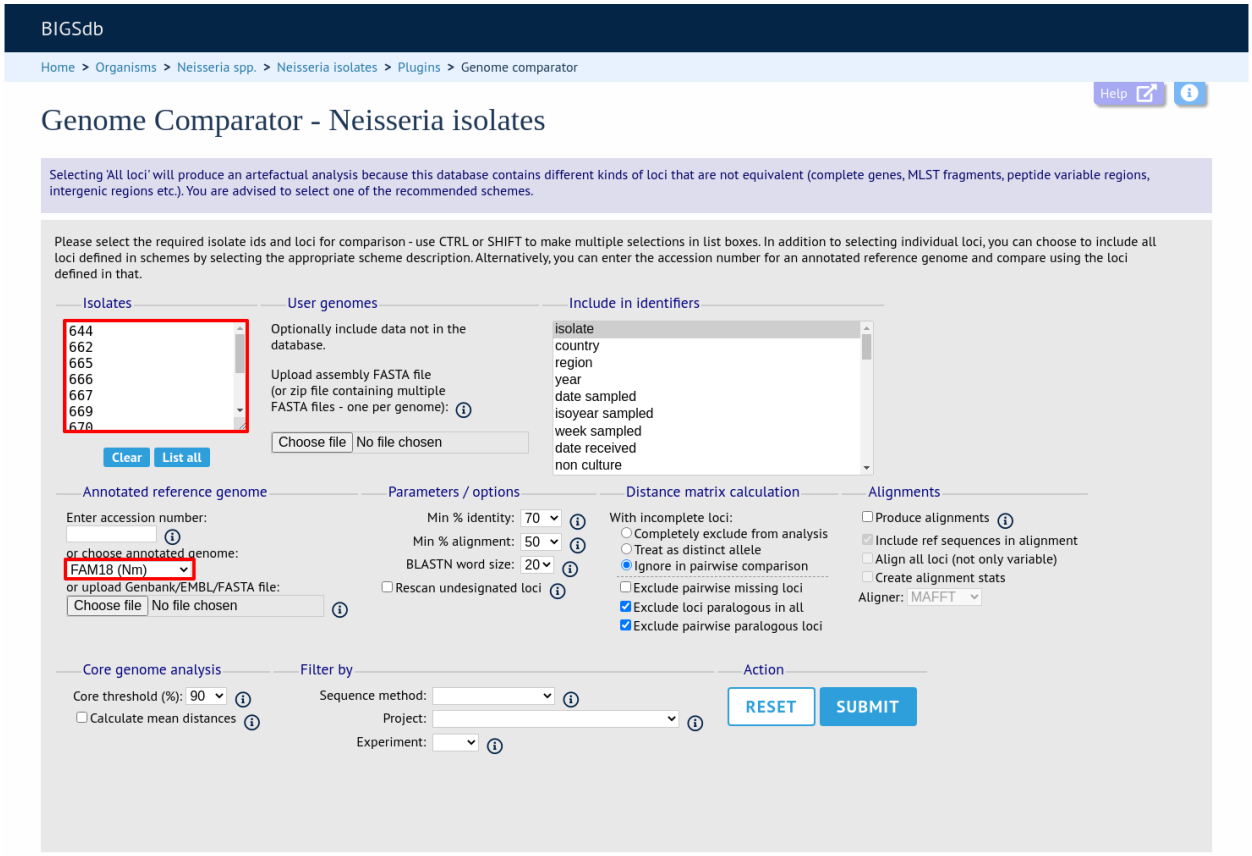
Locus presence frequency (text)

Tar file containing all output files

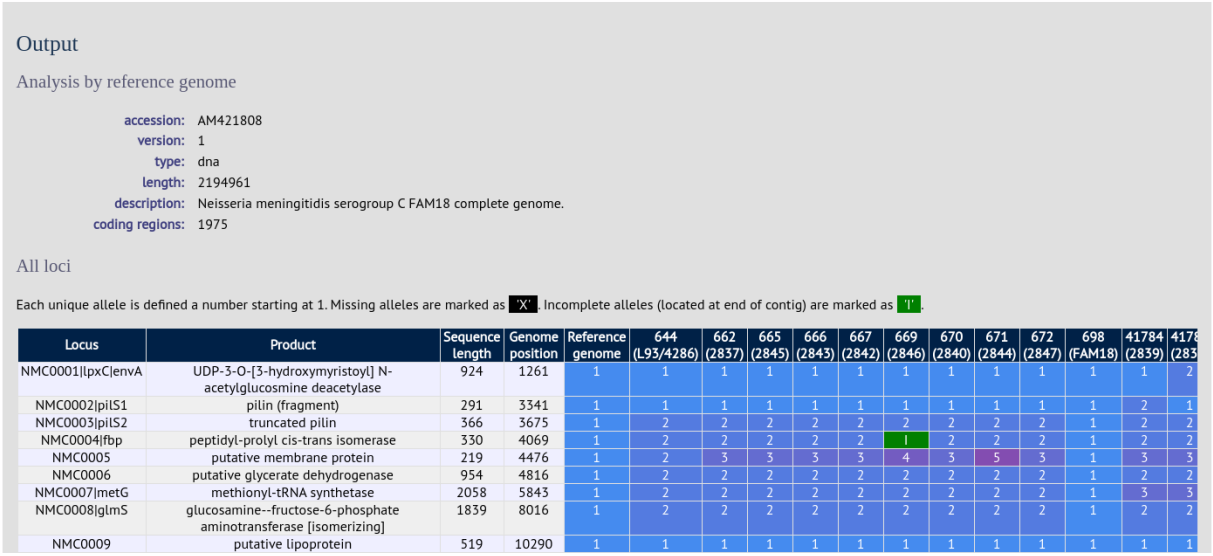
Please note that job results will remain on the server for 7 days.

14.6.2 Analysis using annotated reference genome

Select the isolate genomes that you wish to analyse and then either enter a Genbank accession number for the reference genome, or select from the list of reference genomes (this list will only be present if the administrator has *set it up*). Selecting reference genomes will hide the locus and scheme selection forms.



Output is similar to when comparing against defined loci, but this time every coding sequence in the annotated reference will be BLASTed against the selected genomes. Because allele designations are not defined, the allele found in the reference genome is designated allele 1, the next different sequence is allele 2 etc.



Output

Analysis by reference genome

accession: AM421808
version: 1
type: dna
length: 2194961
description: Neisseria meningitidis serogroup C FAM18 complete genome.
coding regions: 1975

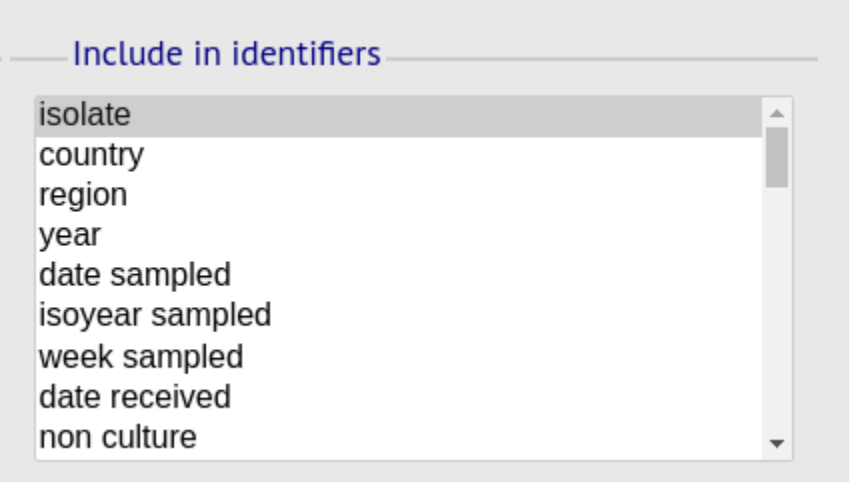
All loci

Each unique allele is defined a number starting at 1. Missing alleles are marked as 'X'. Incomplete alleles (located at end of contig) are marked as 'I'.

Locus	Product	Sequence length	Genome position	Reference genome	644 (L93/4286)	662 (2837)	665 (2845)	666 (2843)	667 (2842)	669 (2846)	670 (2840)	671 (2844)	672 (2847)	698 (FAM18)	41784 (2839)	417 (283
NMC0001|lpxC|envA	UDP-3-O-[3-hydroxymyristoyl] N-acetylglucosmine deacetylase	924	1261	1	1	1	1	1	1	1	1	1	1	1	1	2
NMC0002|pilS1	pilin (fragment)	291	3341	1	1	1	1	1	1	1	1	1	1	1	2	1
NMC0003|pilS2	truncated pilin	366	3675	1	2	2	2	2	2	2	2	2	2	1	2	2
NMC0004|fbp	peptidyl-prolyl cis-trans isomerase	330	4069	1	2	2	2	2	2	1	2	2	2	1	2	2
NMC0005	putative membrane protein	219	4476	1	2	3	3	3	3	4	3	5	3	1	3	3
NMC0006	putative glycerate dehydrogenase	954	4816	1	2	2	2	2	2	2	2	2	2	1	2	2
NMC0007|metG	methionyl-tRNA synthetase	2058	5843	1	2	2	2	2	2	2	2	2	2	1	3	3
NMC0008|glmS	glucosamine--fructose-6-phosphate aminotransferase [isomerizing]	1839	8016	1	2	2	2	2	2	2	2	2	2	1	2	2
NMC0009	putative lipoprotein	519	10290	1	1	1	1	1	1	1	1	1	1	1	1	1

14.6.3 Include in identifiers fieldset

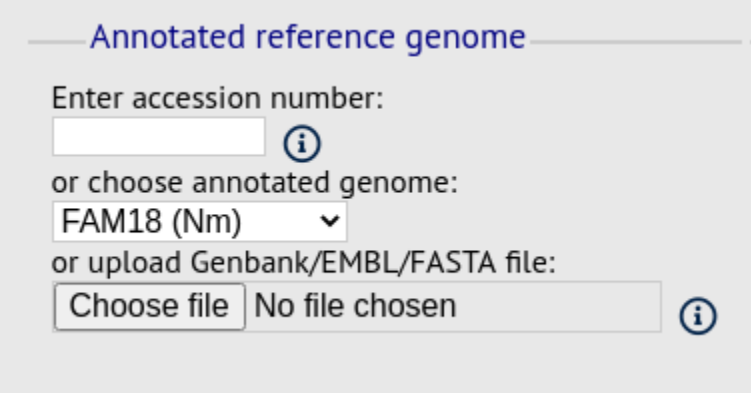
This selection box allows you to choose which isolate provenance fields will be included in the results table. This does not affect the output of the alignments as taxa names are limited in length by the alignment programs.



Multiple values can be selected by clicking while holding down Ctrl.

14.6.4 Reference genome fieldset

This section allows you to choose a reference genome to use as the source of comparator sequences.

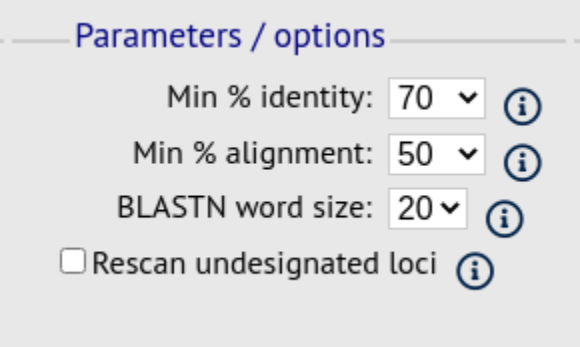


There are three possibilities here:

1. Enter accession number - Enter a Genbank accession number of an annotated reference and Genome Comparator will automatically retrieve this from Genbank.
2. Select from list - The administrator may have selected some genomes to offer for comparison. If these are present, simply select from the list.
3. Upload genome - Click 'Browse' and upload your own reference. This can either be in Genbank, EMBL or FASTA format. Ensure that the filename ends in the appropriate file extension (.gb, .embl, .fas) so that it is recognized.

14.6.5 Parameters/options fieldset

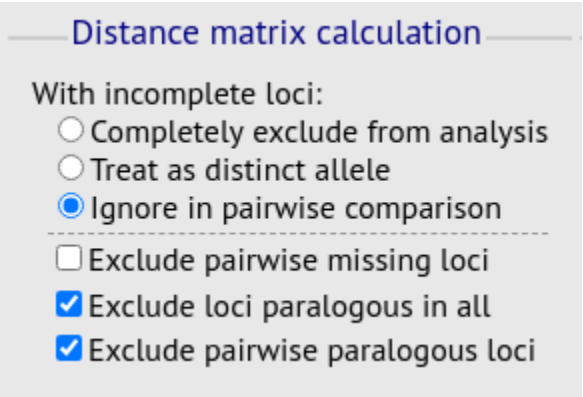
This section allows you to modify BLAST parameters. This affects sensitivity and speed.



- Min % identity - This sets the threshold identity that a matching sequence has to be in order to be considered (default: 70%). Only the best match is used.
- Min % alignment - This sets the percentage of the length of reference allele sequence that the alignment has to cover in order to be considered (default: 50%).
- BLASTN word size - This is the length of the initial identical match that BLAST requires before extending a match (default: 20). Increasing this value improves speed at the expense of sensitivity. The default value gives good results in most cases. The default setting used to be 15 but the new default of 20 is almost as good (there was 1 difference among 2000 loci in a test run) but the analysis runs twice as fast.

14.6.6 Distance matrix calculation fieldset

This section provides options for the treatment of incomplete and paralogous loci when generating the distance matrix.



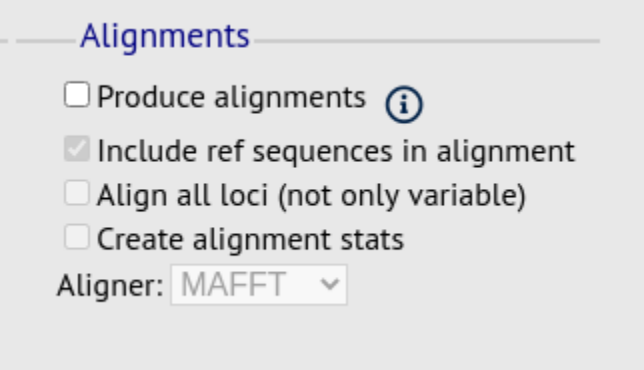
For incomplete loci, i.e. those that continue beyond the end of a contig so are incomplete you can:

- Completely exclude from analysis - Any locus that is incomplete in at least one isolate will be removed from the analysis completely. Using this option means that if there is one bad genome with a lot of incomplete sequences in your analysis, a large proportion of the loci may not be used to calculate distances.
- Treat as a distinct allele - This treats all incomplete sequences as a specific allele 'I'. This varies from any other allele, but all incomplete sequences will be treated as though they were identical.
- Ignore in pairwise comparison (default) - This is probably the best option. In this case, incomplete alleles are only excluded from the analysis when comparing the particular isolate that has it. Other isolates with different alleles will be properly included. The effect of this option will be to shorten the distances of isolates with poorly sequenced genomes with the others.

Paralogous loci, i.e. those with multiple good matches, can be excluded from the analysis (default). This is the safest option since there is no guarantee that differences seen between isolates at paralogous loci are real if the alternative matches are equally good. NB: Loci are also only classed as paralogous when the alternative matches identify different sequences, otherwise multiple contigs of the same sequence region would result in false positives.

14.6.7 Alignments fieldset

This section enables you to choose to produce alignments of the sequences identified.

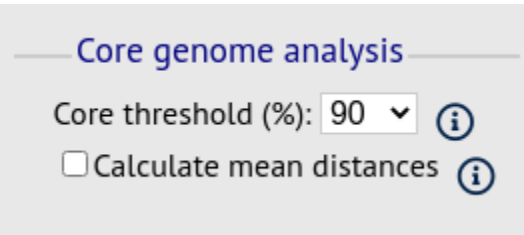


Available options are:

- Produce alignments - Selecting this will produce the alignment files, as well as XMFA and FASTA outputs of aligned sequences. This will result in the analysis taking longer to run.
- Include ref sequences in alignment - When doing analysis using an annotated reference, selecting this will include the reference sequence in the alignment files.
- Align all loci - By default, only loci that vary among the isolates are aligned. You may however wish to align all if you would like the resultant XMFA and FASTA files to include all coding sequences.
- Aligner - There are currently two choices of alignment algorithm (provided they have both been installed)
  - MAFFT (default) - This is the preferred option as it is significantly quicker than MUSCLE, uses less memory, and produces comparable results.
  - MUSCLE - This was originally the only choice. It is still included to enable previous analyses to be re-run and compared but it is recommended that MAFFT is used otherwise.

14.6.8 Core genome analysis fieldset

This section enables you to modify the inclusion threshold used to calculate whether or not a locus is part of the core genome (of the dataset).

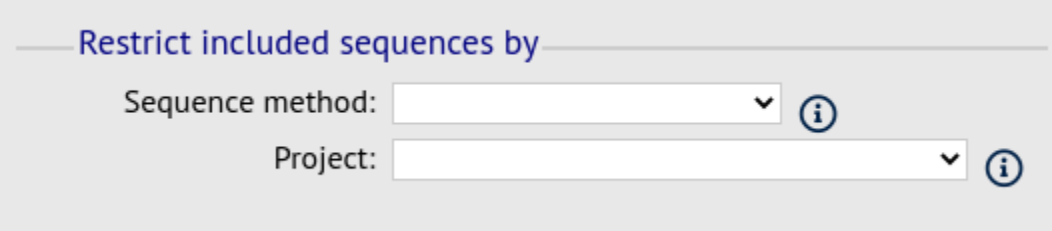


The default setting of 90% means that a locus is counted as core if it appears within 90% or more of the genomes in the dataset.

There is also an option to calculate the mean distance among sequences of the loci. Selecting this will also select the option to produce alignments.

14.6.9 Filter fieldset

This section allows you to further filter your collection of isolates and the contigs to include.



Available options are:

- Sequence method - Choose to only analyse contigs that have been generated using a particular method. This depends on the method being set when the contigs were uploaded.
- Project - Only include isolates belonging to the chosen project. This enables you to select all isolates and filter to a project.

14.6.10 Understanding the output

Distance matrix

The distance matrix is simply a count of the number of loci that differ between each pair of isolates. It is generated in NEXUS format which can be used as the input file for SplitsTree. This can be used to generate NeighborNet, Split decomposition graphs and trees offline. If 200 isolates or fewer are included in the analysis, a Neighbor network is automatically generated from this distance matrix.

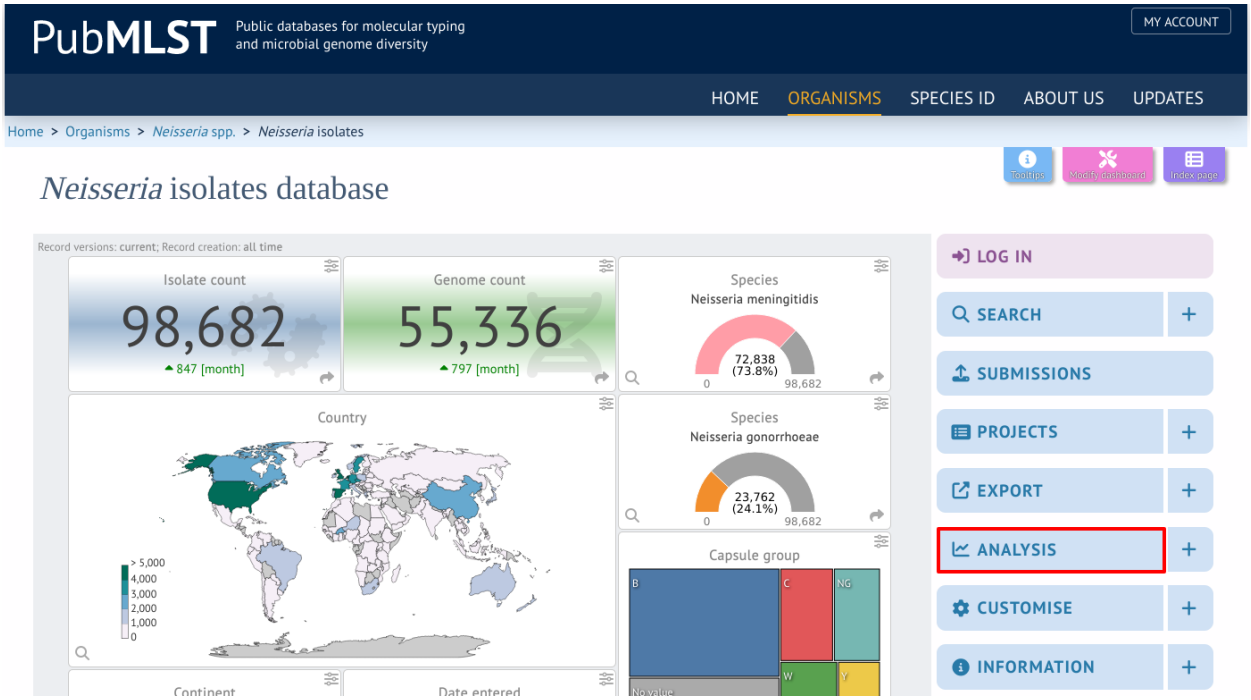
Unique strains

The table of unique strains is a list of isolates that are identical at every locus. Every isolate is likely to be classed as unique if a whole genome analysis is performed, but with a constrained set of loci, such as those for MLST, this will group isolates that are indistinguishable at that level of resolution. nce matrix calculated based on allelic differences and a NeighborNet graph generated from this distance matrix.

14.7 GrapeTree

GrapeTree is a tool for generating and visualising minimum spanning trees. It has been developed to handle large datasets (in the region of 1000s of genomes) and works with 1000s of loci as used in cgMLST. It uses an improved minimum spanning algorithm that is better able to handle missing data than alternative algorithms and is able to produce publication quality outputs. Datasets can include metadata which allows nodes in the resultant tree to be coloured interactively.

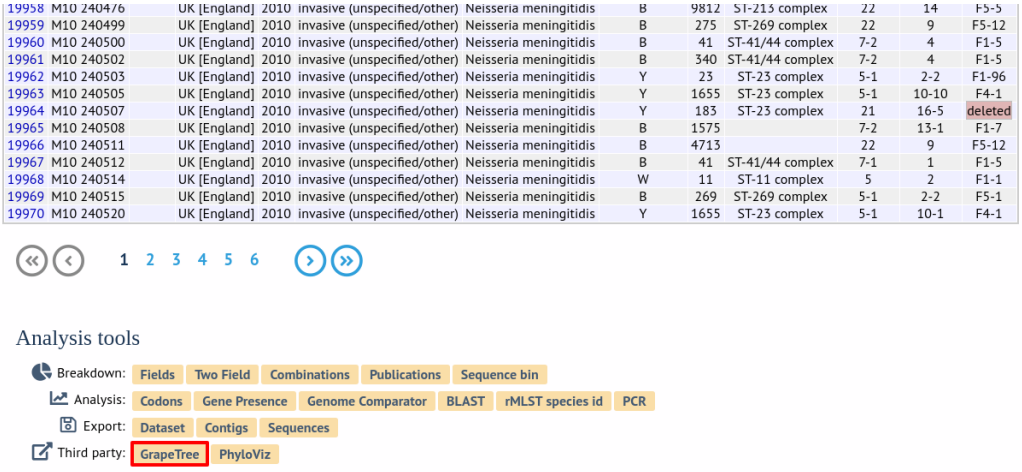
GrapeTree can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Third party' category, follow the link to GrapeTree, then click 'Launch GrapeTree'.

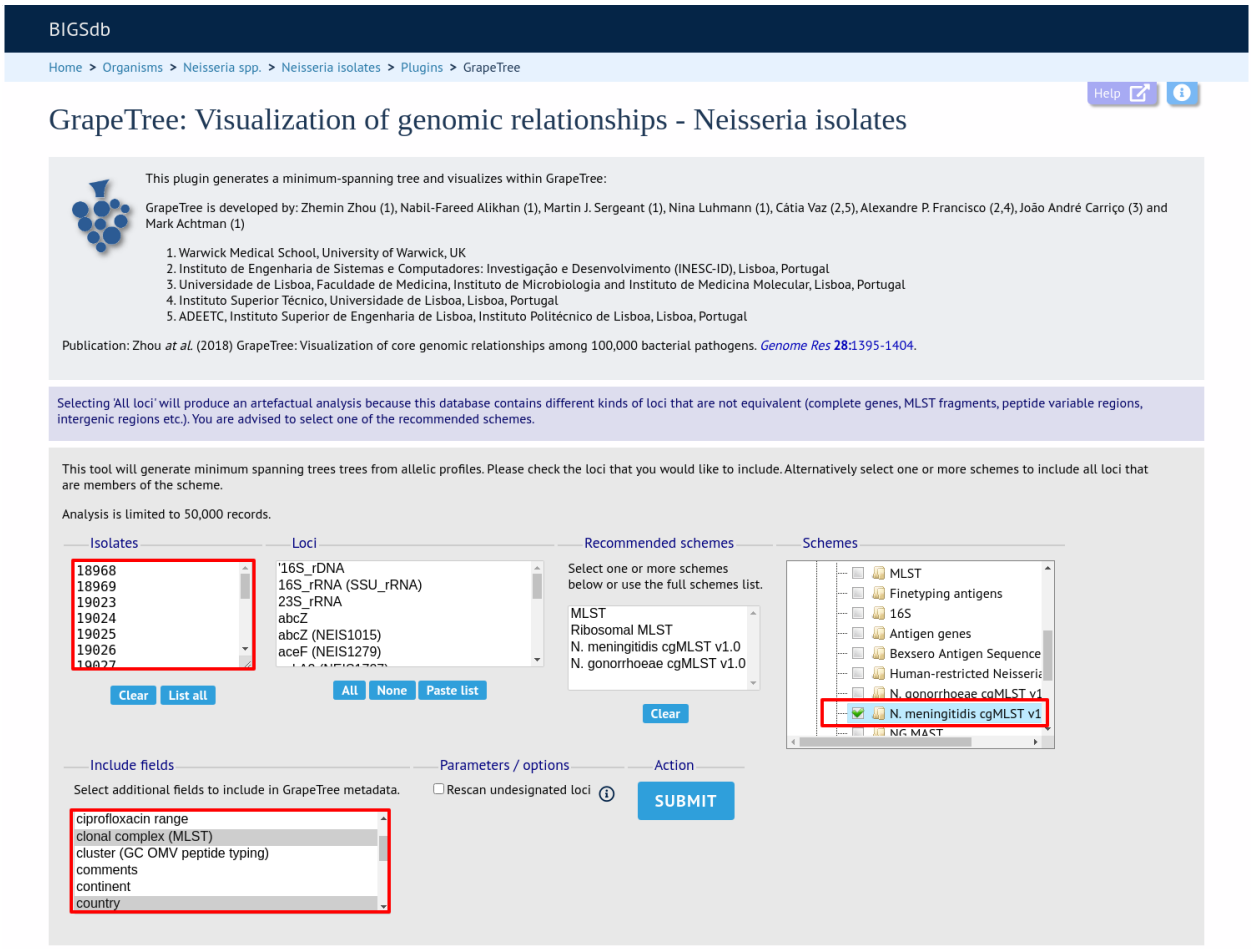
GrapeTree can be accessed from the contents page by clicking the 'GrapeTree' link.

Alternatively, it can be accessed following a query by clicking the 'GrapeTree' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the GrapeTree interface.

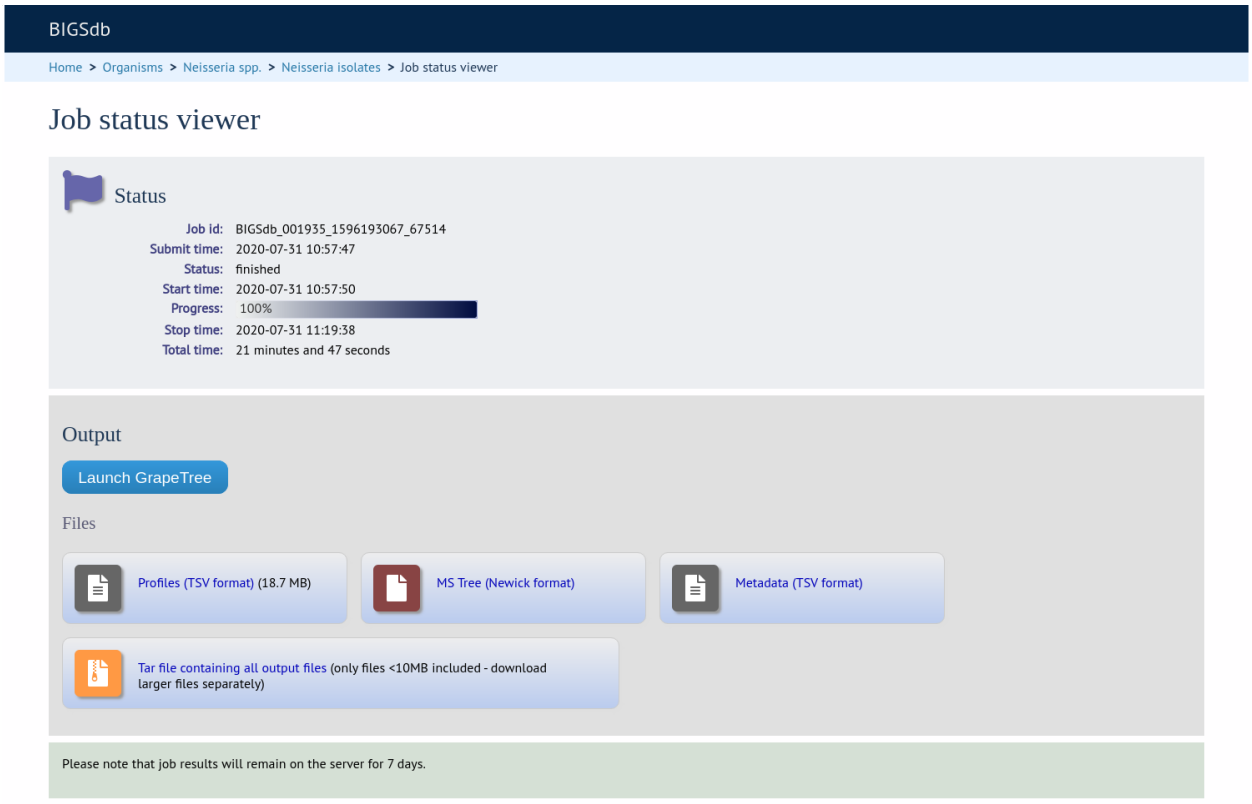


Select the isolates to include. The tree can be generated from allelic profiles of any selection of loci, or more conveniently, you can select a scheme in the scheme selector, or choose from recommended schemes if these have been set, to include all loci belonging to that scheme.

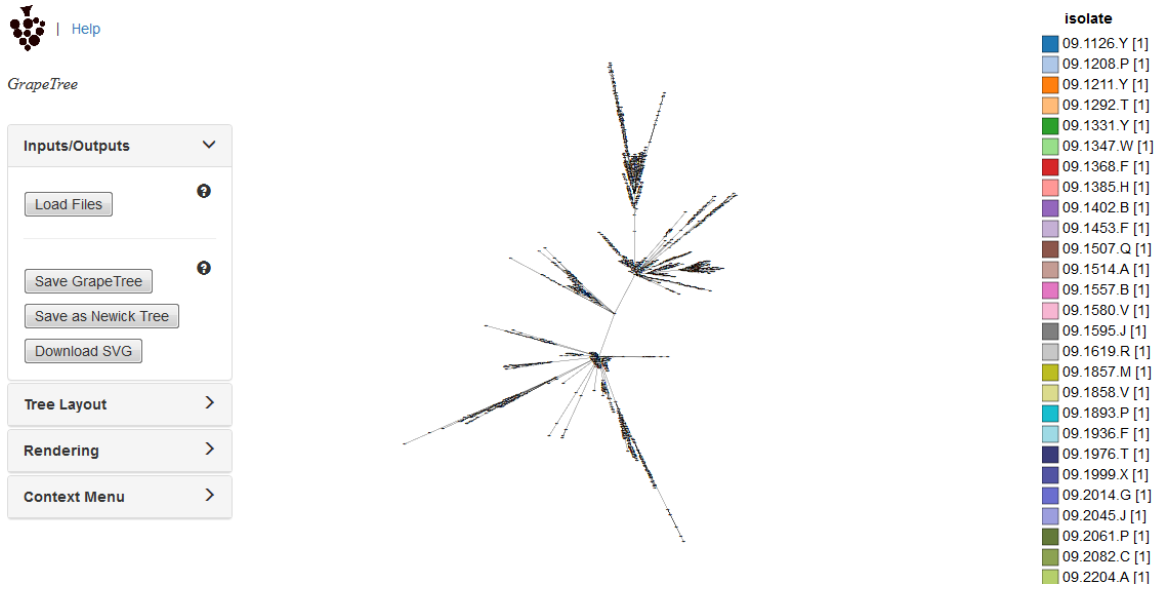
Additional fields can be selected to be included as metadata for use in colouring nodes - select any fields you wish to include. Multiple selections can be made by holding down shift or ctrl while selecting. Click 'Submit' to start the analysis.



The job will be sent to the job queue. When it has finished, click the button marked 'Launch GrapeTree'.



The generated tree will be rendered in the GrapeTree application page.

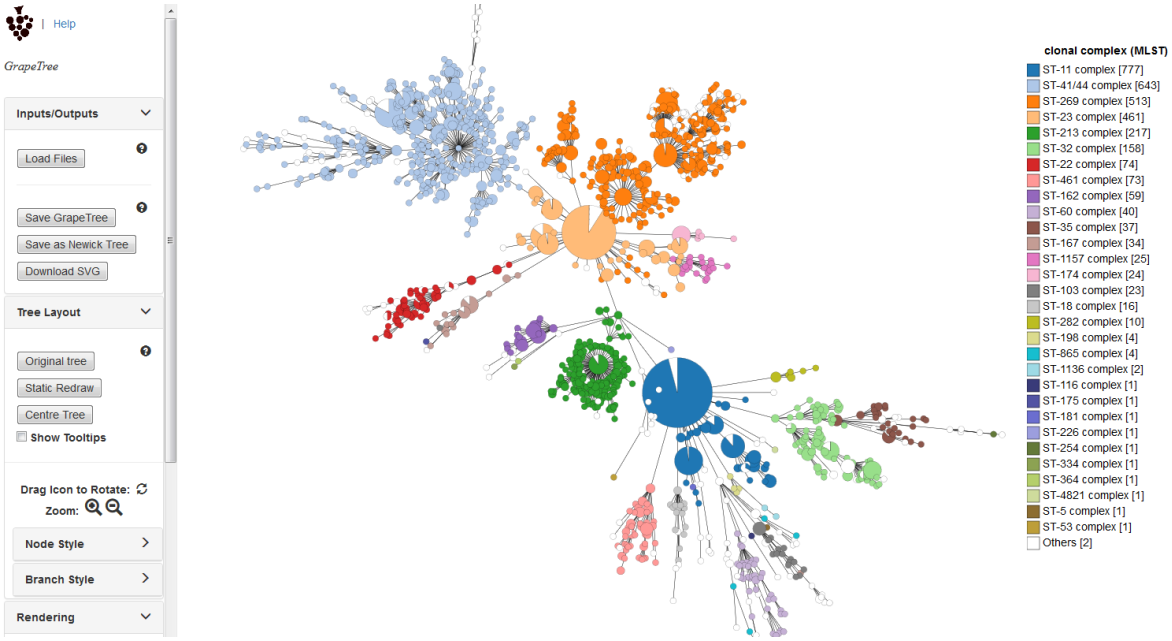


The image can be manipulated in various ways. These include modifying the tree layout, customising node labels and size, modifying branch lengths and collapsing branches. The image can be saved in SVG format which can be further edited in image publishing software such as Inkscape.

As an example, the default cgMLST tree (above) has been modified (below) as follows:

- Nodes coloured by clonal complex
- Labels removed

- Branches collapsed where <=100 loci different
- Node size set to 200%
- Kurtosis (node size relative to number of isolates) set to 75%
- Dynamic rendering allowed to run to fan out nodes



Full details can be found in the GrapeTree manual.

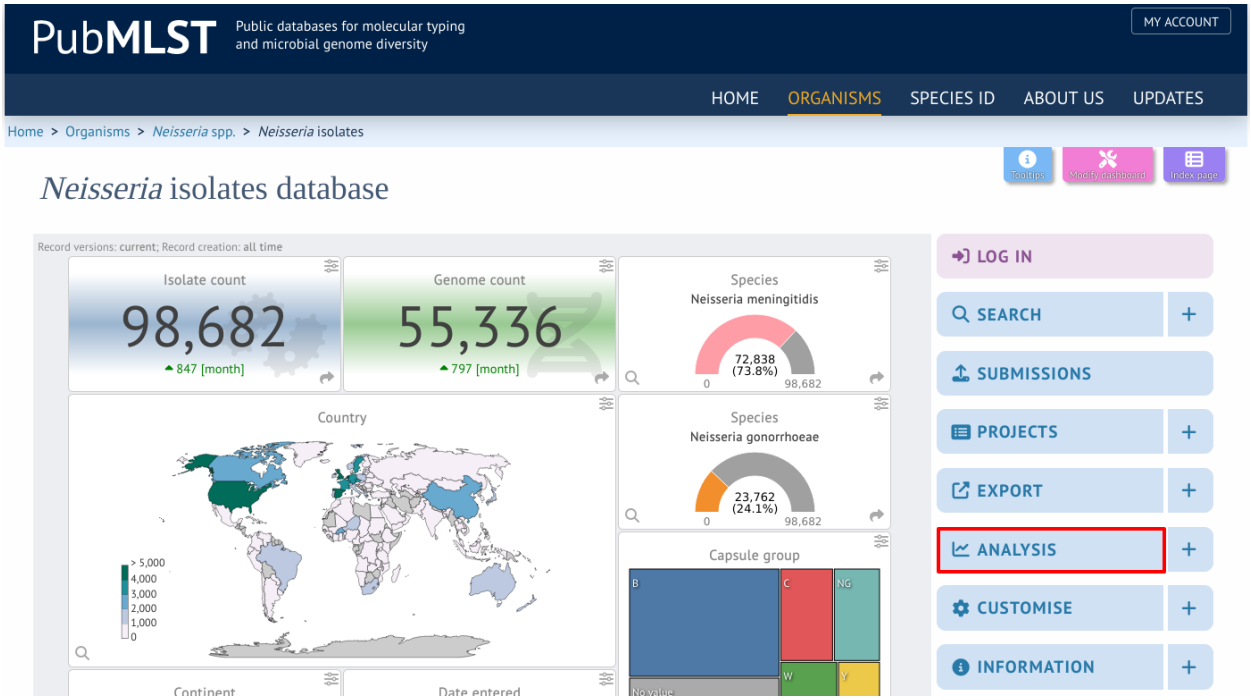
Note: GrapeTree has been described in the following publication:

Z Zhou, NF Alikhan, MJ Sergeant, N Luhmann, C Vaz, AP Francisco, JA Carrico, M Achtman (2018) GrapeTree: Visualization of core genomic relationships among 100,000 bacterial pathogens. Genome Res 28:1395-1404.

14.8 In silico PCR

This is a tool that can be used to simulate PCR reactions run against genomes stored in the database. This is useful for designing and testing primers. The plugin uses the exonerate ipcress program to perform its simulation.

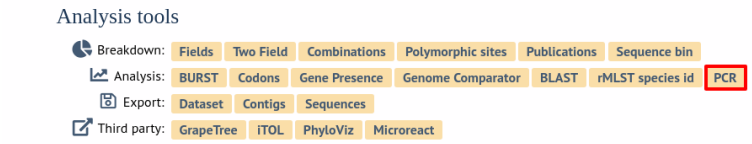
The tool can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Analysis' category, follow the link to BLAST, then click 'Launch PCR'.

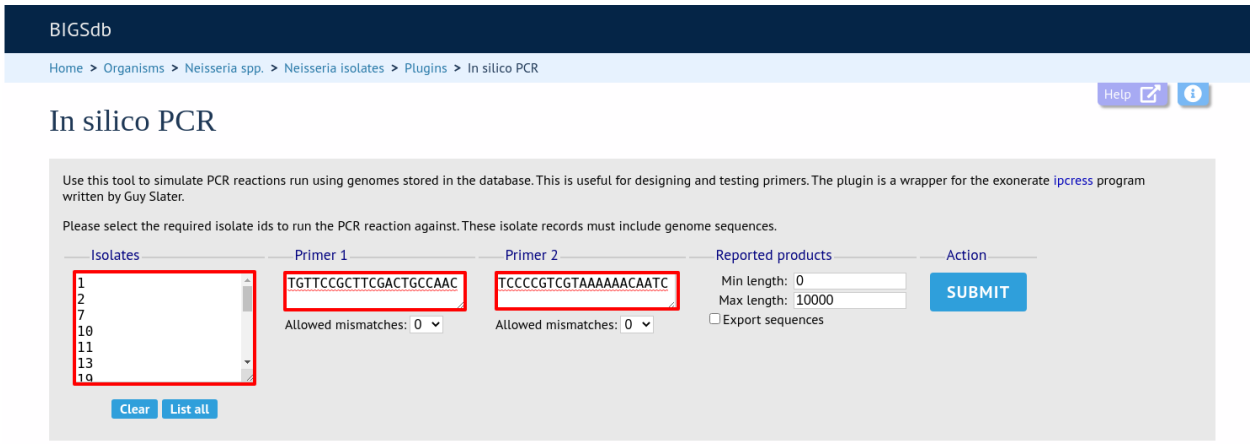
Alternatively, it can be accessed following a query by clicking the 'PCR' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the analysis interface.

10	6748	B73; NIBSC_2784; Z1073	Canada	1971	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	18-1	3	F5-1
11	129E	B92; NIBSC_2828; Z1092	Germany	1964	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F3-6
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
30	14	BennettTree10; alpha14	Germany	1999	carrier	Neisseria meningitidis	cnl	53	ST-53 complex	7	30-3	F5-5
31	10	B269; NIBSC_2825; Z1269	Burkina Faso	1963	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
34	20	B275; NIBSC_2767; Z1275	Niger	1963	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-7
35	26	B278; NIBSC_2764; Z1278	Niger	1963	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13	F1-5
52	243	B362; NIBSC_2779; Z1362	Cameroon	1966		Neisseria meningitidis	A	4	ST-4 complex	7	13	F1-5
61	393	B392; NIBSC_2823; Z1392	Greece	1968	carrier	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
64	254	B439; NIBSC_2812; Z1439	Djibouti	1966	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F1-7
67	S5611	B466; NIBSC_2765; Z1466	Australia	1977	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
82	11-004	B503; NIBSC_2826; Z1503	China	1984	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-8
84	IAL2229	B506; NIBSC_2816; Z1506	Brazil	1976		Neisseria meningitidis	A	5	ST-5 complex	20	9	F2-1
90	CN100	B534; NIBSC_2768; Z1534	UK [England]	1941	invasive (unspecified/other)	Neisseria meningitidis	A	21		5-2	10	F3-9

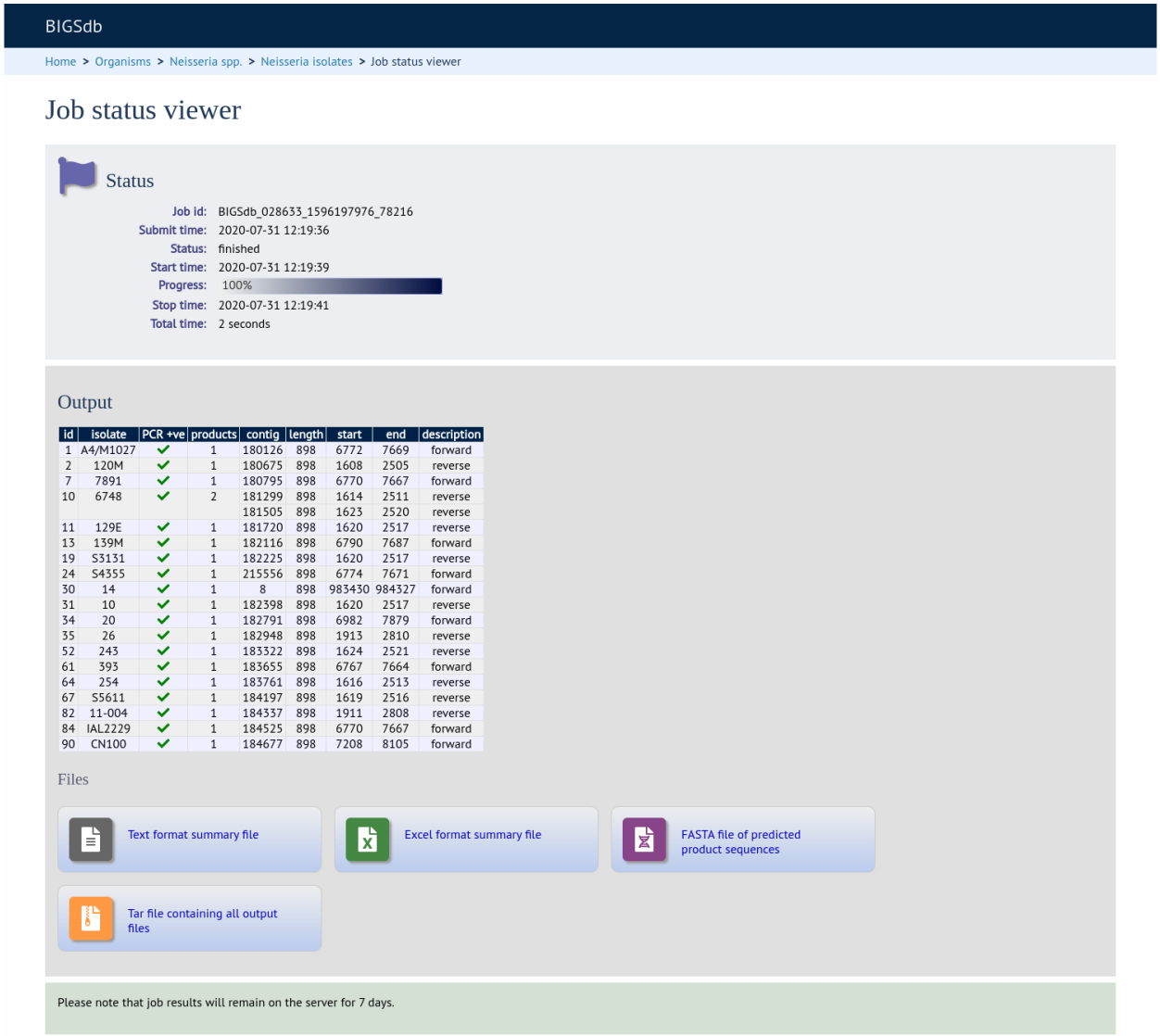


Select the isolates to include. These will be pre-populated if you arrive here following a search.

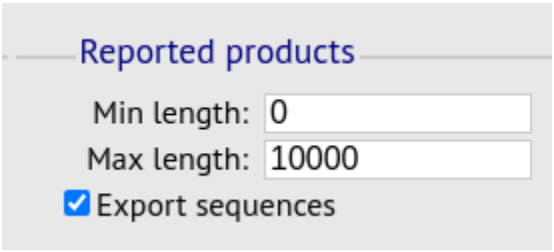
Enter your forward and reverse primer sequences in the appropriate boxes. These may contain wobble bases if necessary. You can also specify how many mismatches are allowed for each primer. Finally, you can restrict the reported length to only those products that fall between a minimum and maximum length range.



Click 'Submit'. The job will be sent to the job queue. The output will be a table of predicted products, showing the number of products and their positions within a contig. A summary of this table is also availabe to download in tab-delimited text of Excel formats.



It is also possible to export the predicted product sequence. You can do this by selecting the 'Export sequences' checkbox on the options form.

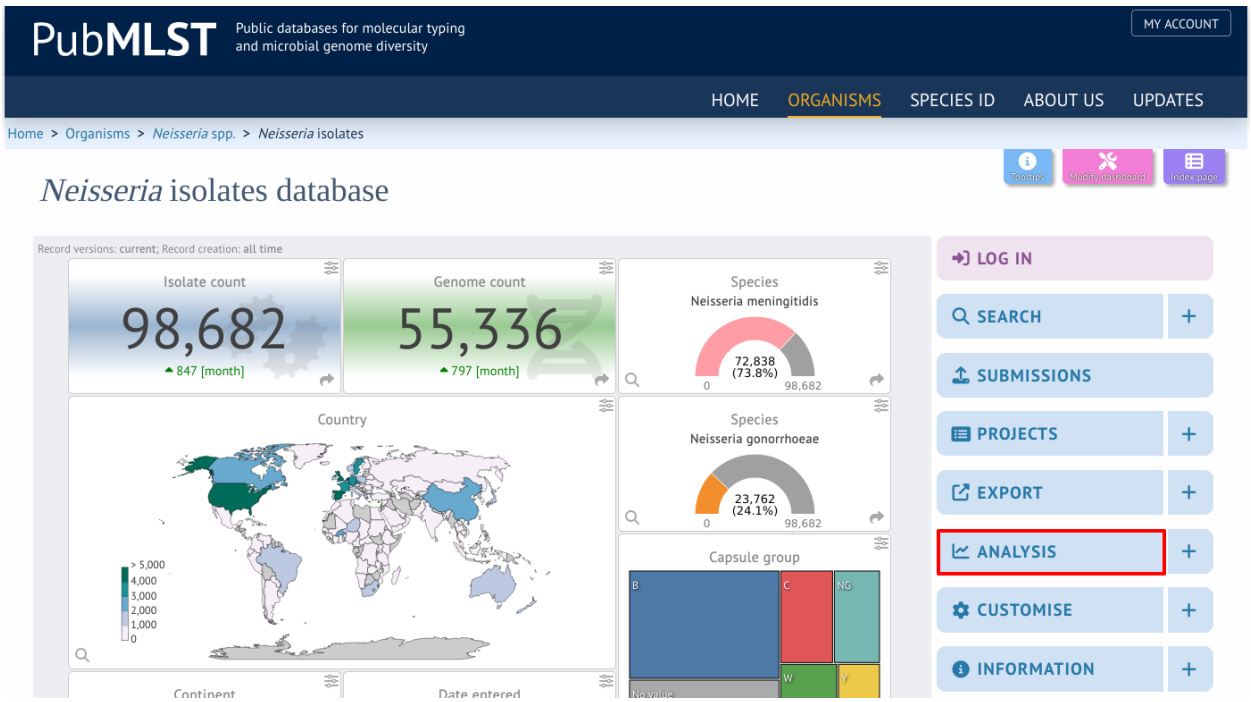


Note: The exported sequences will include the primer regions. It is important to note that, unlike a real PCR reaction, these sequences represent the sequence within this region in the genome. In a real PCR reaction, the primers are themseleves incorporated in to the product, so even if there was a mismatch in the primer region, the product sequence would include the primer sequence.

14.9 Interactive Tree of Life (iTOL)

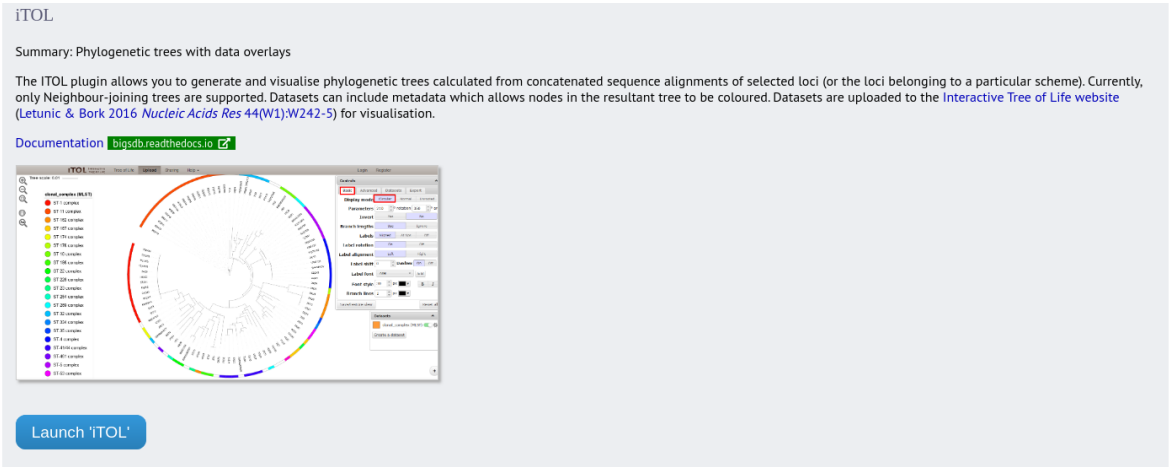
The ITOL plugin allows you to generate and visualise phylogenetic trees calculated from concatenated sequence alignments of selected loci (or the loci belonging to a particular scheme). Currently, only Neighbour-joining trees are supported. Datasets can include metadata which allows nodes in the resultant tree to be coloured.

ITOL can be accessed by selecting the 'Analysis' section on the main contents page.

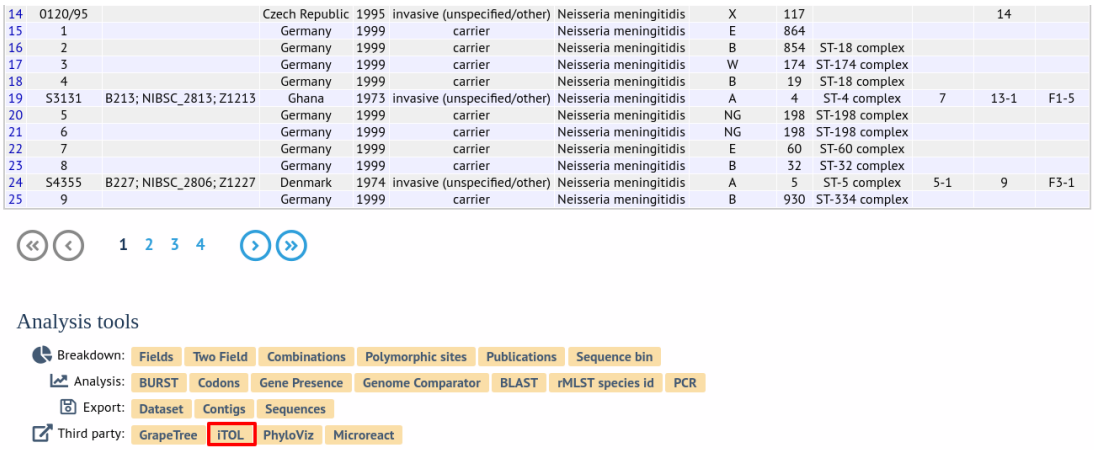


Jump to the 'Third party' category, follow the link to iTOL, then click 'Launch iTOL'.

ITOL can be accessed from the contents page by clicking the 'iTOL' link.



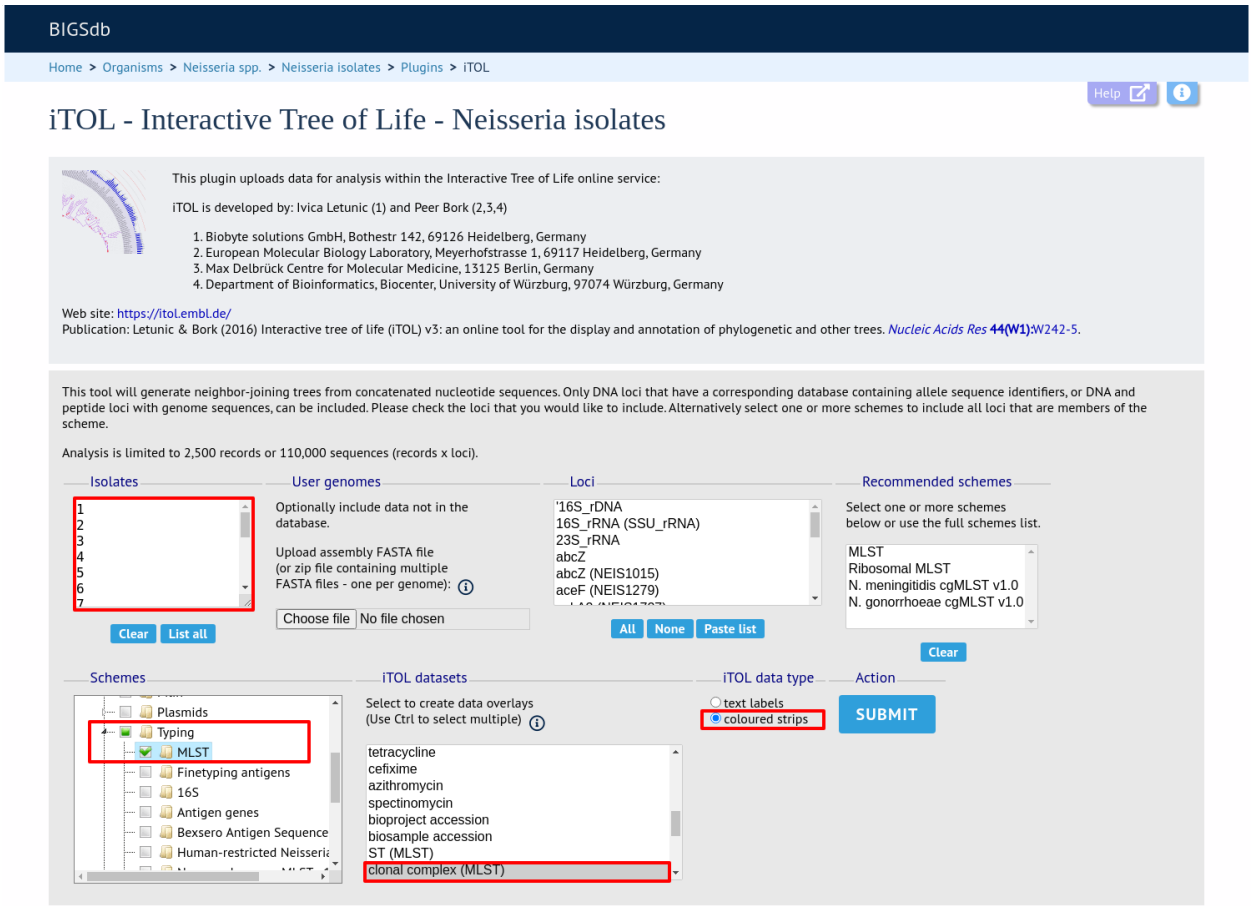
Alternatively, it can be accessed following a query by clicking the 'iTOL' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the iTOL interface.



Select the isolates to include. The tree can be generated from concatenated sequences of any selection of loci, or more conveniently, you can select a scheme in the scheme selector, or in the list of recommended schemes if these have been set up, to include all loci belonging to that scheme.

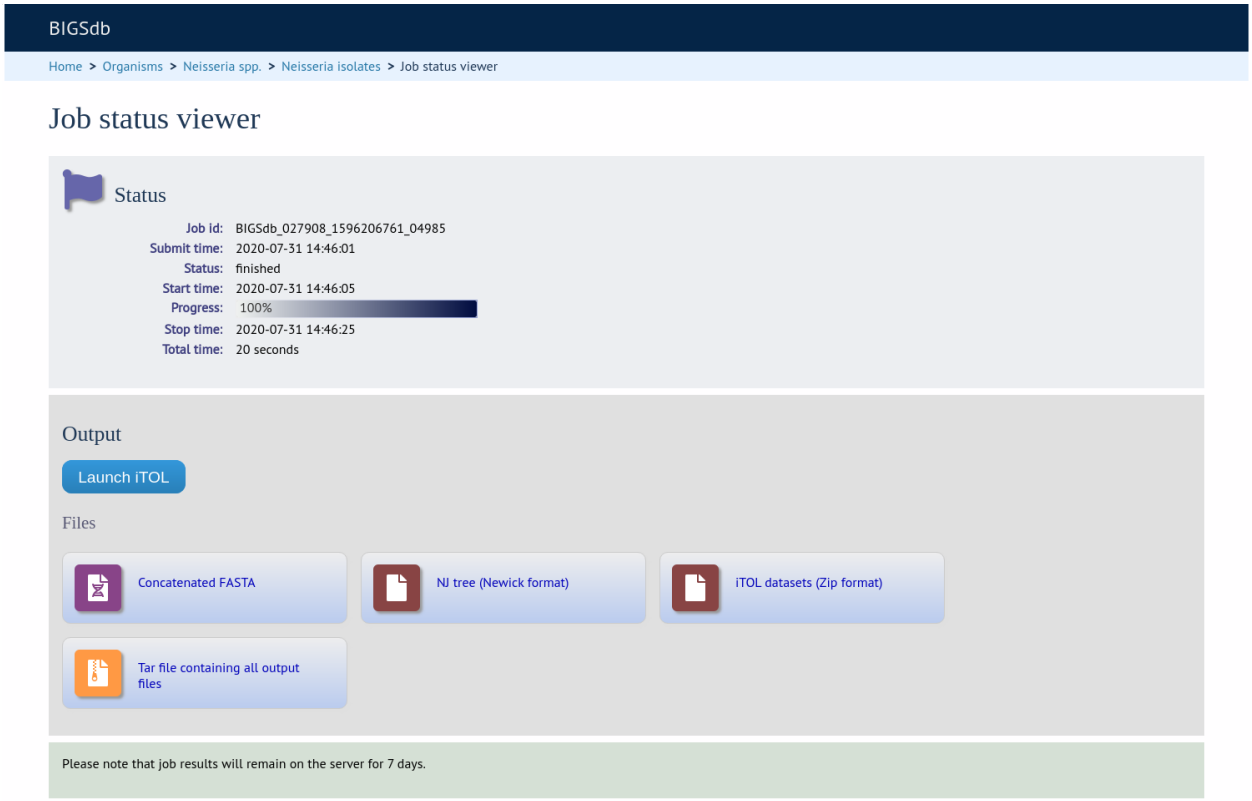
Additional fields can be selected to be included as metadata for use in colouring nodes - select any fields you wish to include in the 'iTOL datasets' list. Multiple selections can be made by holding down Shift or Ctrl while selecting. You can also choose how nodes are labeled by metadata - either by colouring the labels or using coloured strips.

Click 'Submit' to start the analysis.

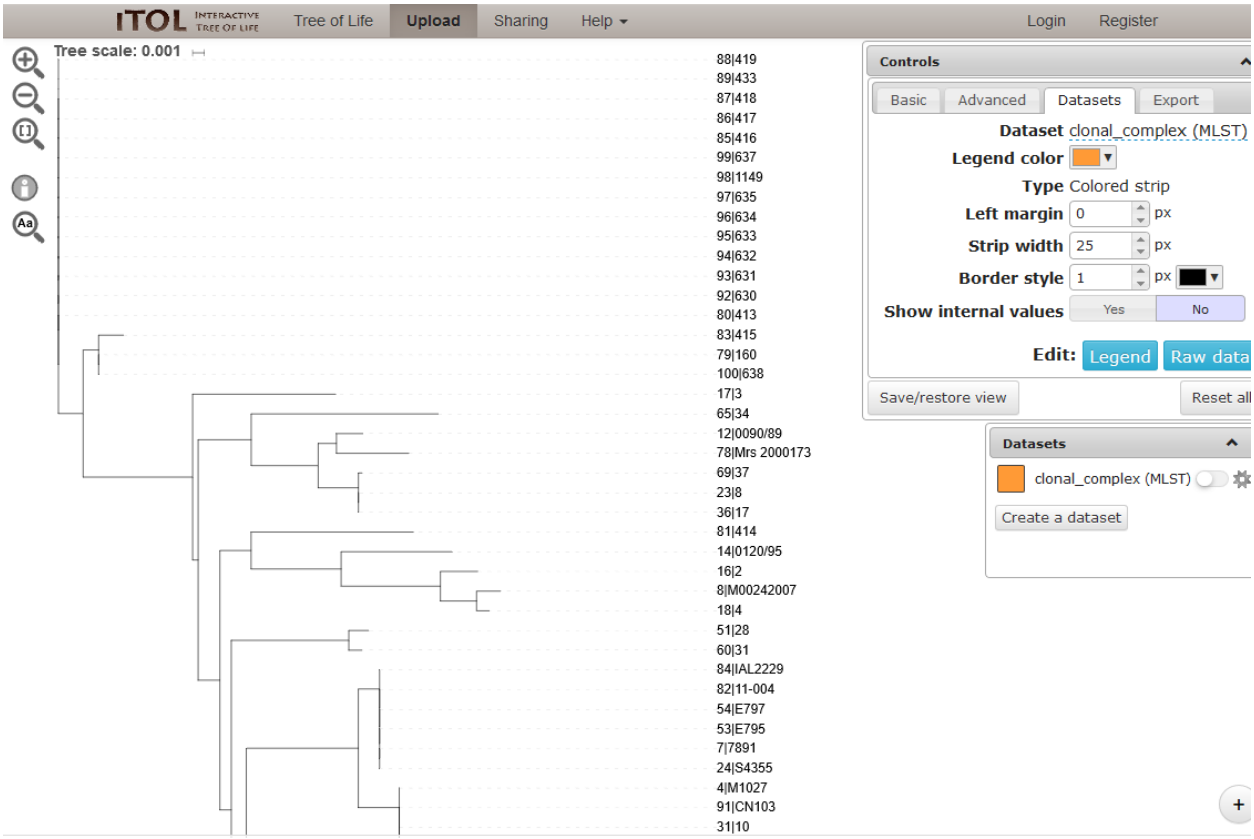


The job will be sent to the job queue. When it has finished, the generated tree and associated metadata will be uploaded

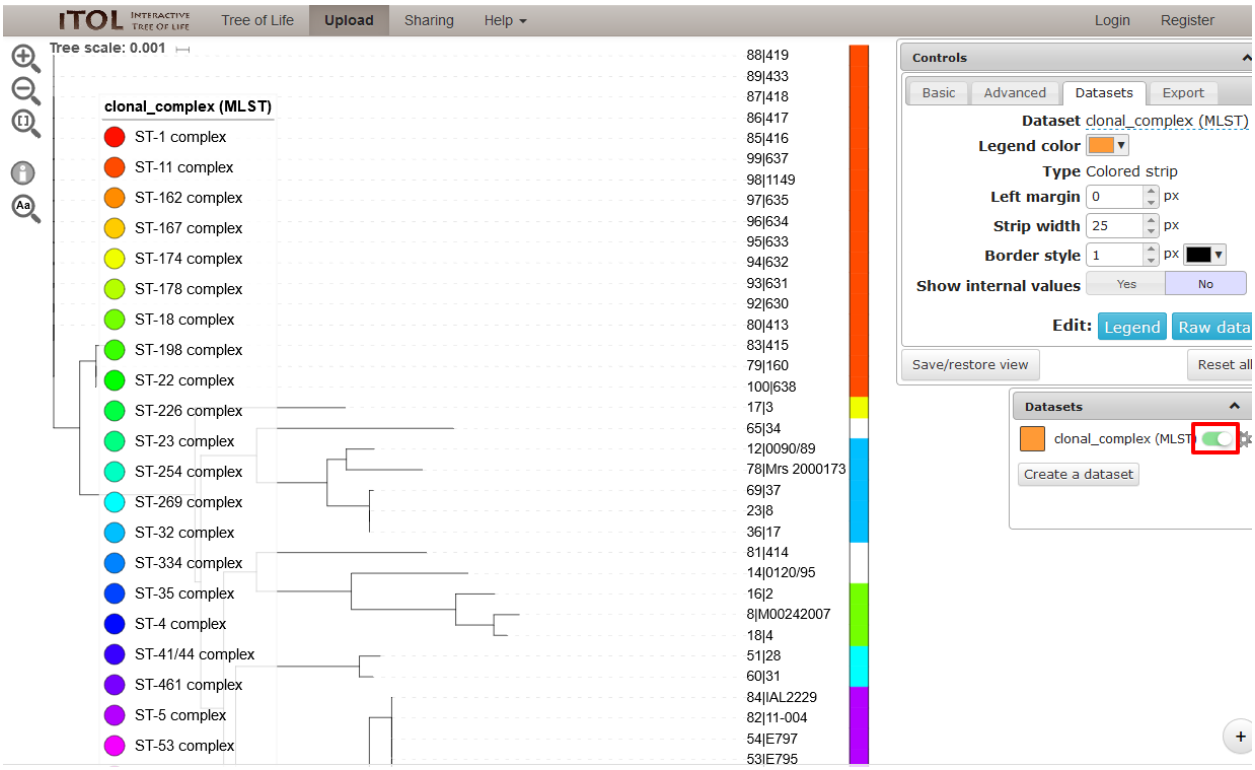
to the Interactive Tree of Life website (https://itol.embl.de/). Click the button marked 'Launch iTOL'.



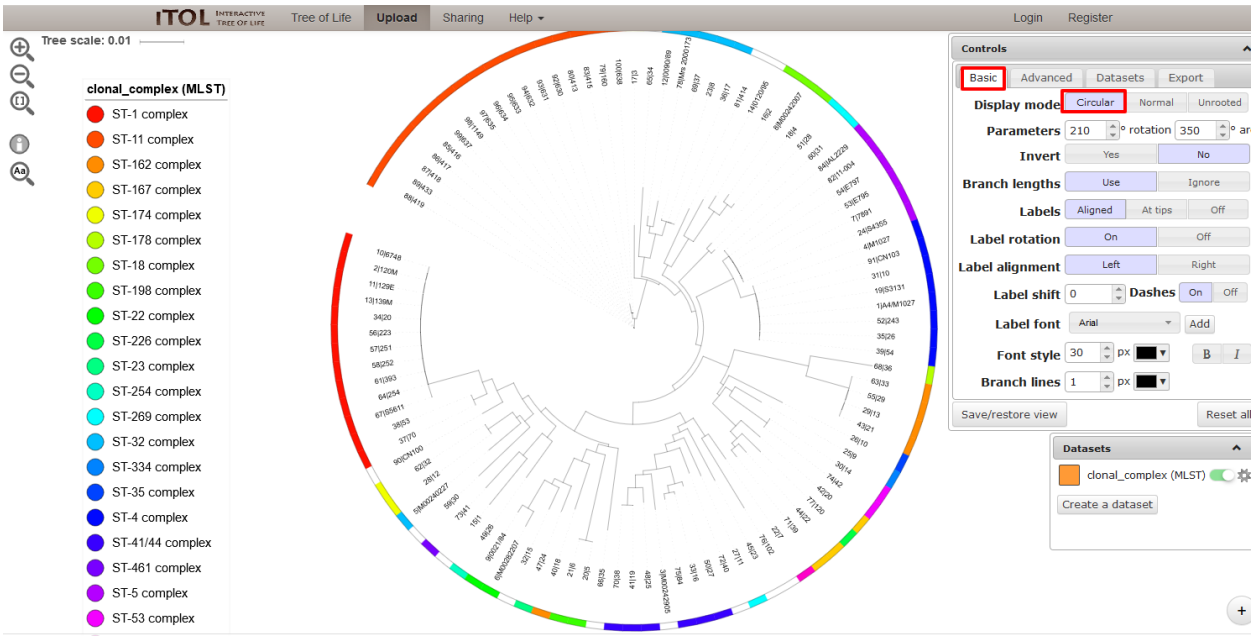
Your browser will open the iTOL website with your tree.



You can manipulate the tree in the browser, and display metadata by selecting the appropriate toggle.



The tree layout can be changed by clicking the 'Basic tab' and, for example, selecting a circular display mode.



See the detailed documentation on the iTOL website for more information about manipulating and exporting trees.

14.10 Kleborate

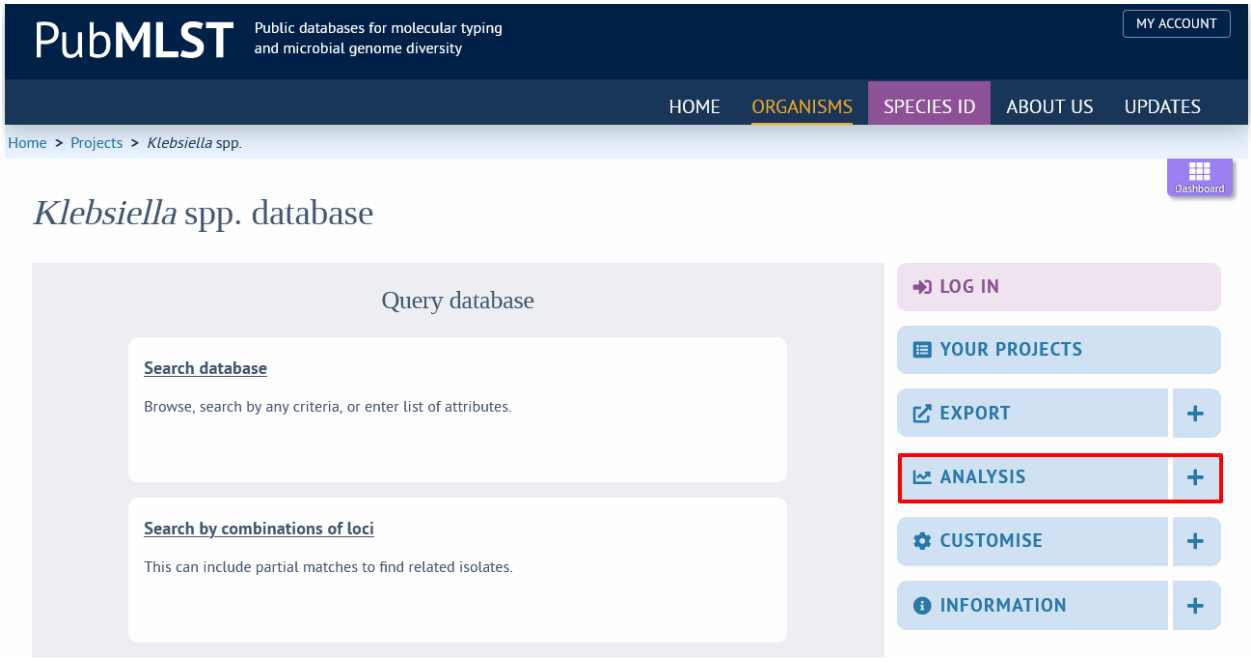
Kleborate is a tool that can be used to screen assemblies of *Klebsiella pneumoniae* and the *Klebsiella pneumoniae* species complex (KpSC) for:

- MLST sequence type
- species (e.g. *K. pneumoniae*, *K. quasipneumoniae*, *K. variicola*, etc.)
- ICE *Kp* associated virulence loci: yersiniabactin (*ybt*), colibactin (*clb*), salmochelin (*iro*), hypermucoidy (*rmpA*)
- virulence plasmid associated loci: salmochelin (*iro*), aerobactin (*iuc*), hypermucoidy (*rmpA*, *rmpA2*)
- antimicrobial resistance determinants: acquired genes, SNPs, gene truncations and intrinsic -lactamases
- K (capsule) and O antigen (LPS) serotype prediction, via *wzi* alleles and Kaptive.

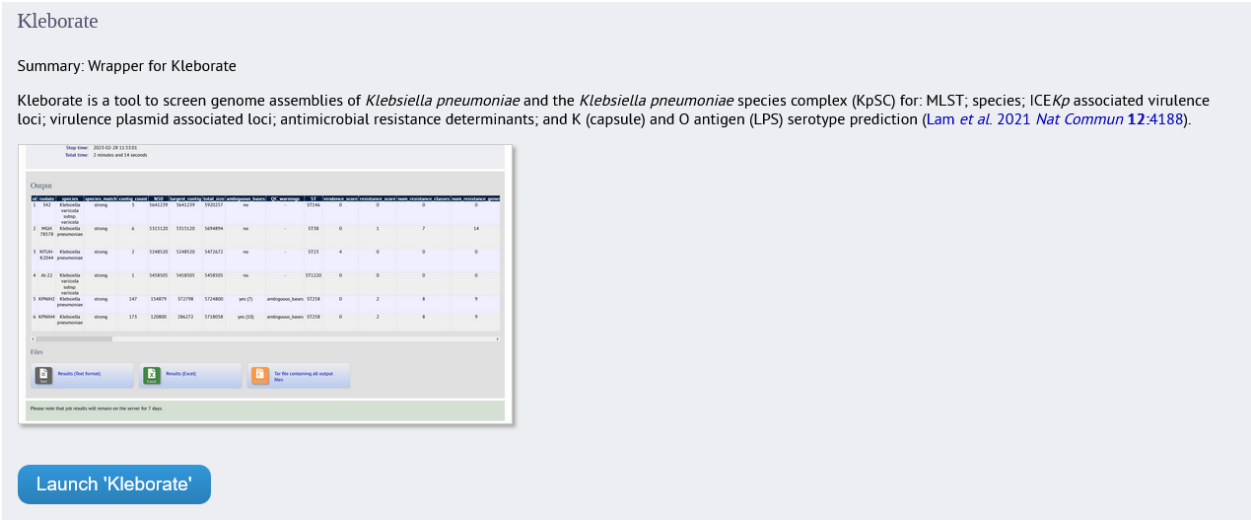
Kleborate and Kaptive are described in Lam *et al.* 2021 *Nat Commun* **12:** 4188 (PubMed: 34234121) and Lam *et al.* 2022 *Microb Genom* **8:** 000800 (PubMed: 35311639) respectively.

Kleborate will usually only be available on databases hosting suitable data i.e. *Klebsiella* isolates.

The function can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Third party' category, follow the link to Kleborate, then click 'Launch Kleborate'.



Alternatively, it can be accessed following a query by clicking the 'Kleborate' button in the 'Third party' list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.

17	KPNIH18	GCA_000281475.1	USA	2011		Klebsiella pneumoniae		31218	Klebsiella	Klebsiella pneumoniae			258
19	KPNIH20	GCA_000281375.1	USA	2011		Klebsiella pneumoniae		18941	Klebsiella	Klebsiella pneumoniae			258
20	KPNIH21	GCA_000281495.1	USA	2011		Klebsiella pneumoniae		31218	Klebsiella	Klebsiella pneumoniae			258
21	KPNIH22	GCA_000281515.1	USA	2011		Klebsiella pneumoniae		31218	Klebsiella	Klebsiella pneumoniae			258
22	KPNIH23	GCA_000281635.1	USA	2011		Klebsiella pneumoniae		31218	Klebsiella	Klebsiella pneumoniae			258
23	KCTC 2190	GCA_000215745.1				Klebsiella aerogenes	1	44287	Klebsiella	Klebsiella aerogenes			
24	EA1509E	GCA_000334515.1				Klebsiella aerogenes	1	65928	Klebsiella	Klebsiella aerogenes			
25	KCTC 1686	GCA_000240325.1				Klebsiella michiganensis	1	18893	Klebsiella	Klebsiella michiganensis			
26	B6	GCA_000367425.1				Raoultella ornithinolytica	1	39828	Raoultella	Raoultella ornithinolytica			

1 2 3 4 5 6

Analysis tools

Breakdown: Fields | Two Field | Combinations | Polymorphic sites | Publications | Sequence bin

Analysis: BURST | Codons | Gene Presence | Genome Comparator | BLAST | rMLST species id | PCR

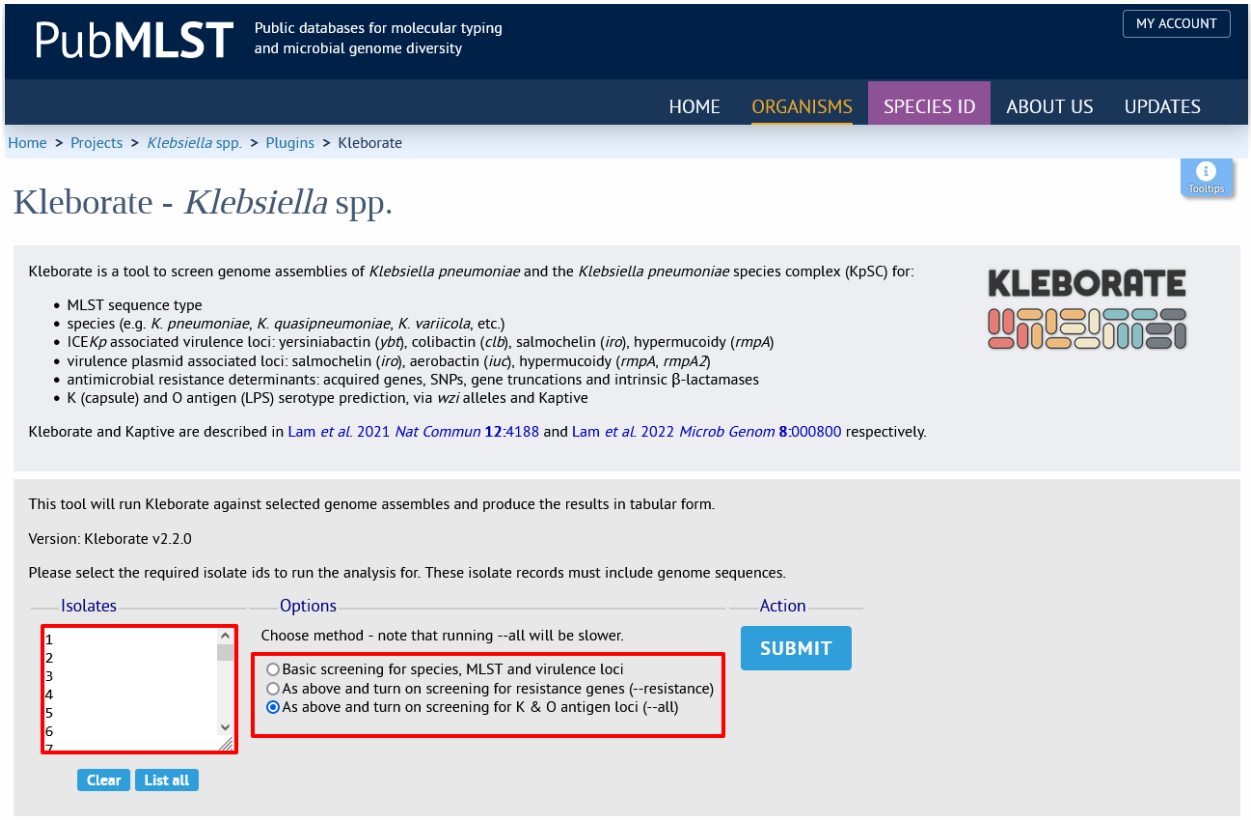
Export: Dataset | Contigs | Sequences

Third party: GrapeTree | iTOL | Kleborate | Microreact | PhyloViz

Select the isolate records to analyse - these will be pre-selected if you accessed the plugin following a query. You can then choose the analysis to run:

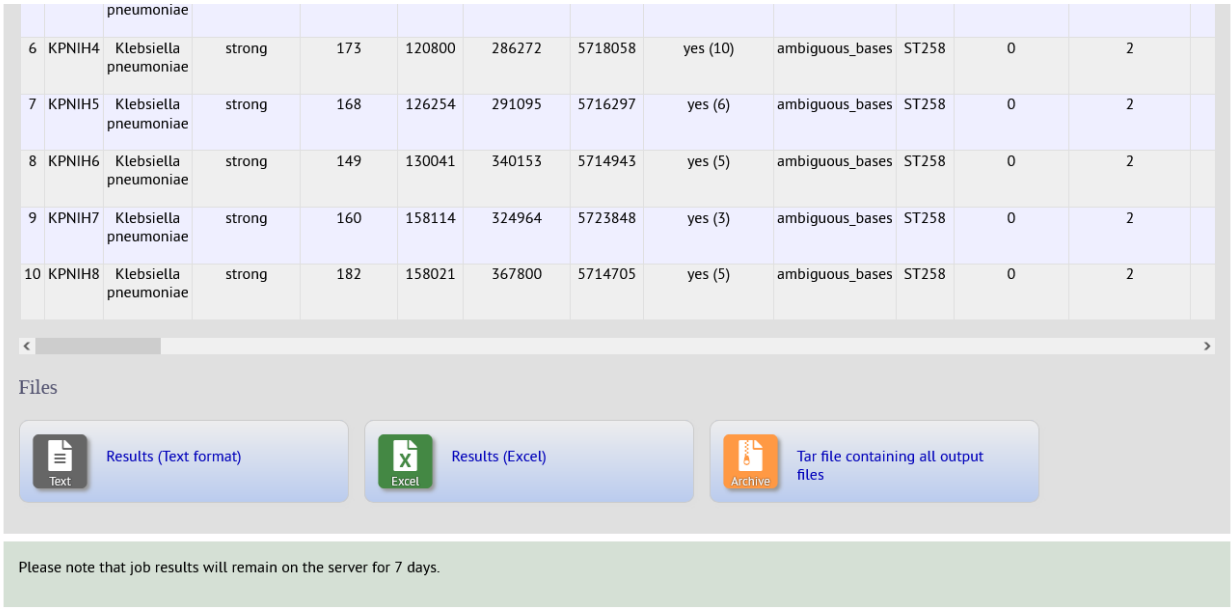
- Basic screening for species, MLST and virulence loci
- As above and turn on screening for resistance genes (–resistance)
- As above and turn on screening for K & O antigen loci (–all)

Click submit.



The analysis will be submitted to the job queue.

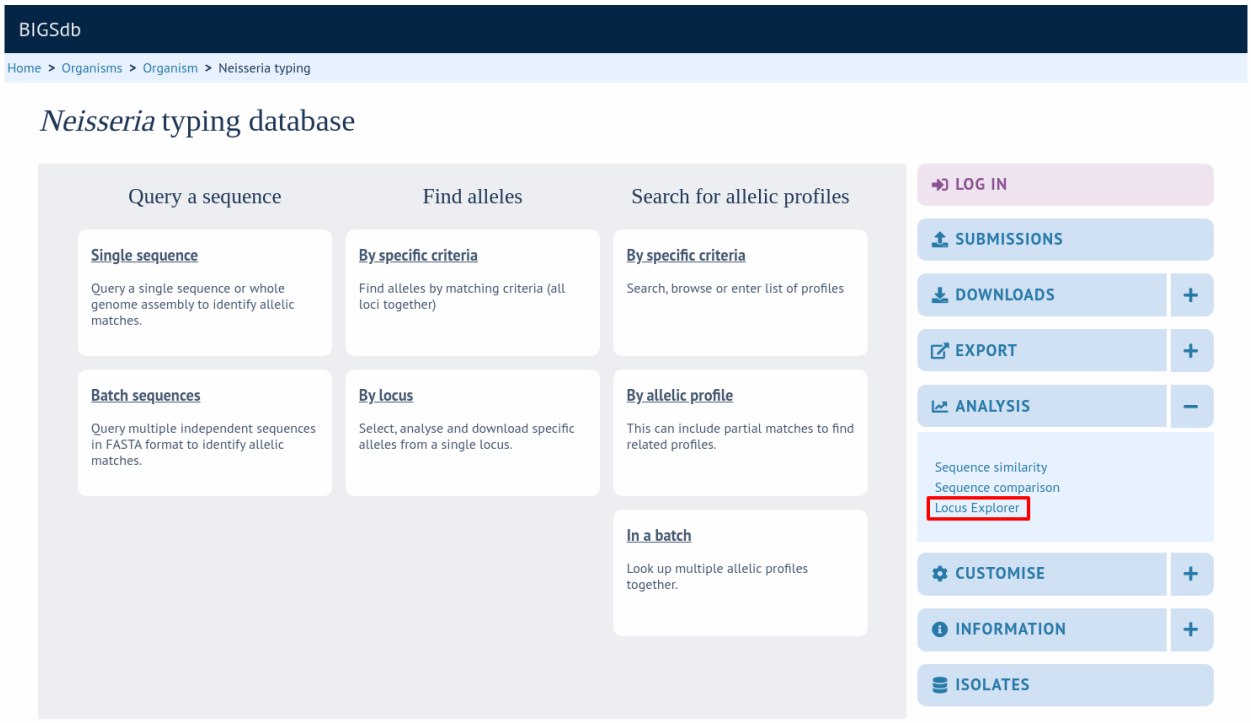
When the job has completed you will see a table of results that is also available for download in tab-delimited text and Excel formats.



14.11 Locus explorer

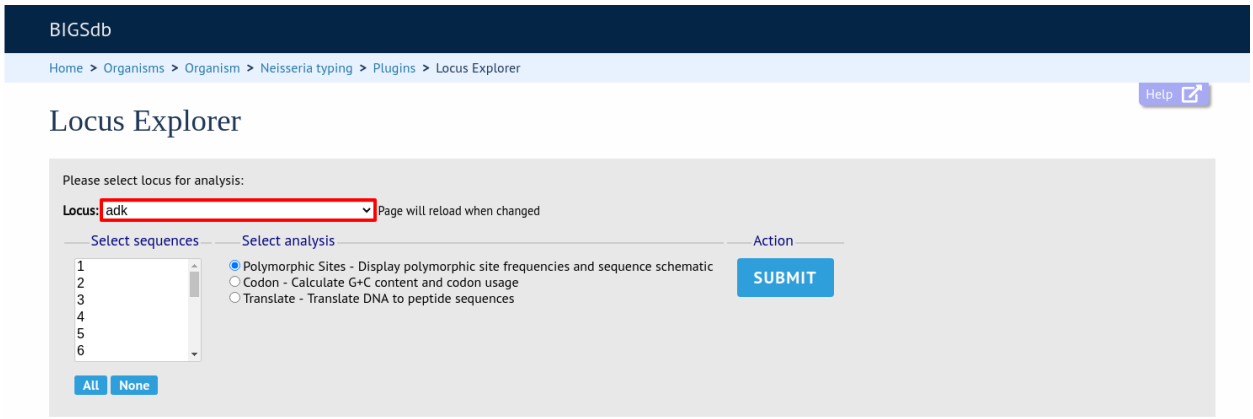
The locus explorer is a sequence definition database plugin. It can create schematics showing the polymorphic sites within a locus, calculate the GC content and generate aligned translated sequences.

Expand the 'Analysis' section on the main contents page of a sequence definition database and click 'Locus Explorer'.

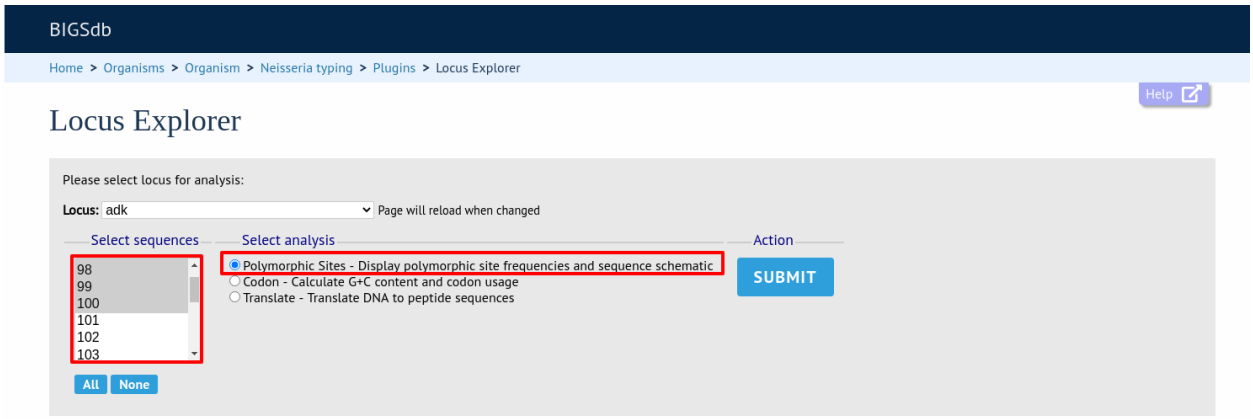


14.11.1 Polymorphic site analysis

Select the locus you would like to analyse in the Locus dropdown box. The page will reload.

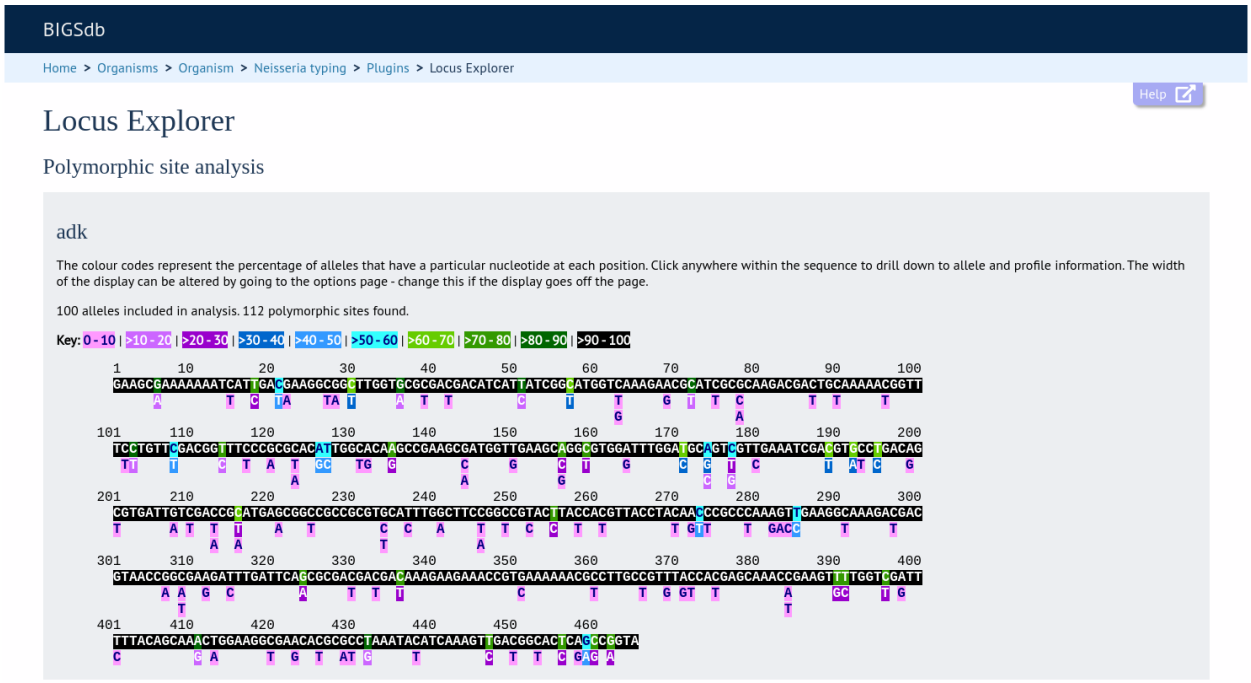


Select the alleles that you would like to include in the analysis. Variable length loci are limited to 2000 sequences or fewer since these need to be aligned. Select 'Polymorphic Sites' in the Analysis selection and click 'Submit'.

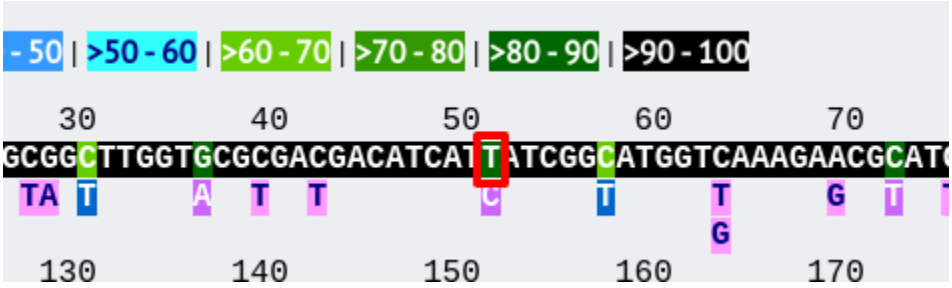


If an alignment is necessary, the job will be submitted to the job queue and the analysis performed. If no alignment is necessary, then the analysis is shown immediately.

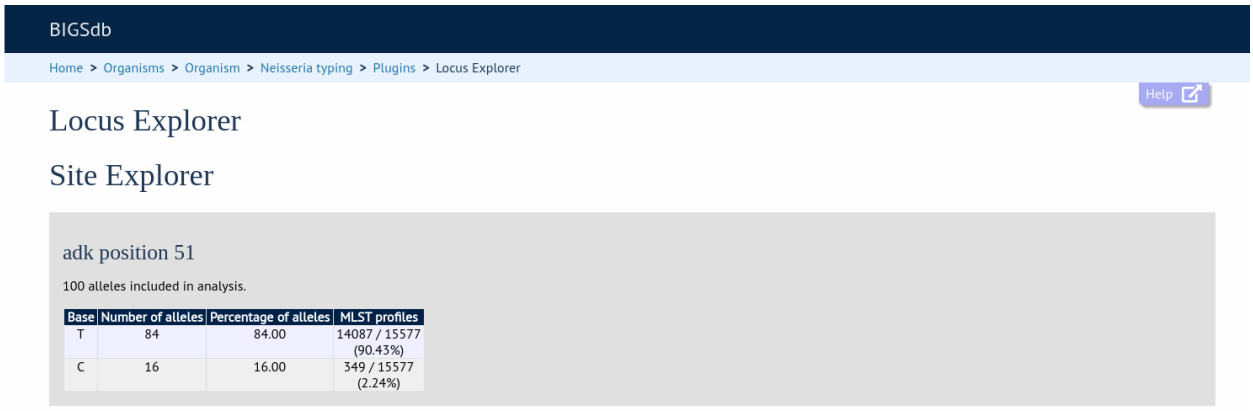
The first part of the page shows the schematic.



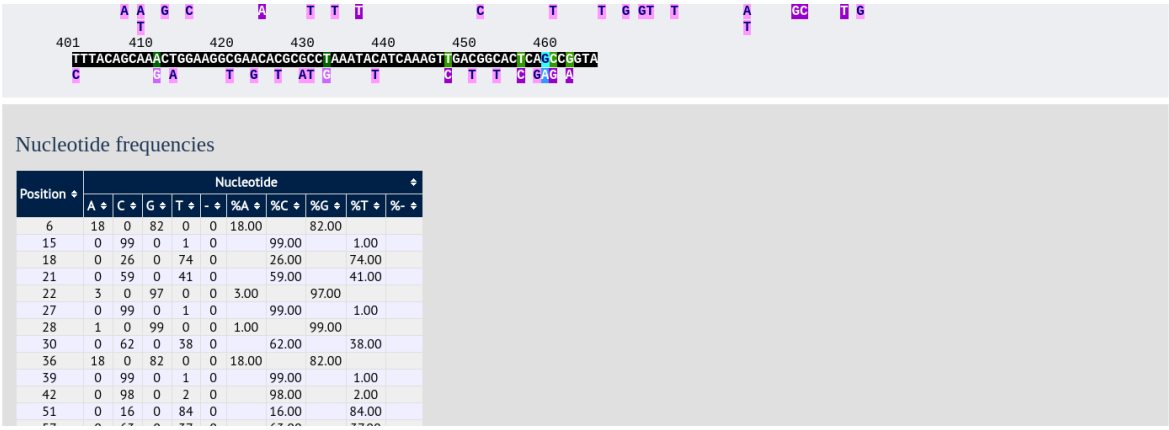
Clicking any of the sequence bases will calculate the exact frequencies of the different nucleotides at that position.



Along with the nucleotide frequencies, it will also show the percentage of allelic profiles containing each nucleotide at that position if the locus is part of a scheme such as MLST.



The second part of the page shows a table listing nucleotide frequencies at each of the variable positions.

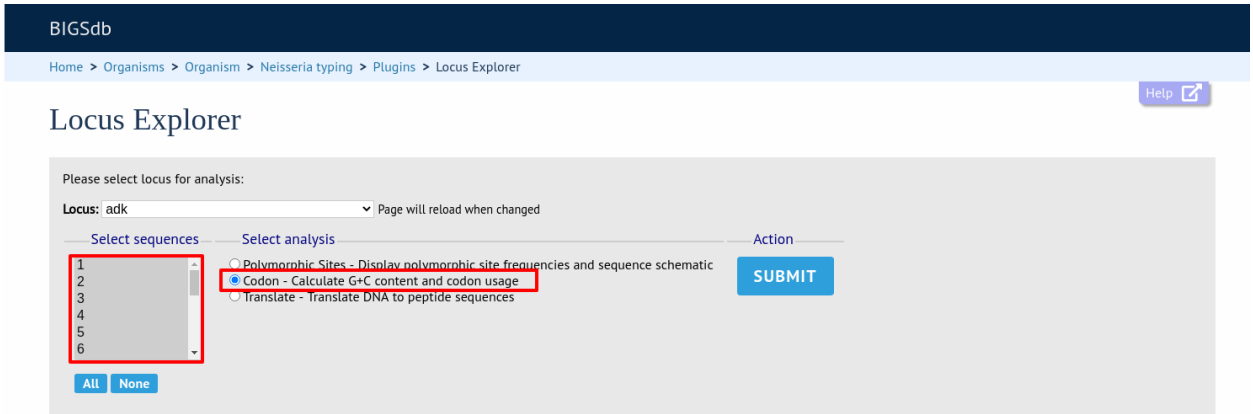


See also:

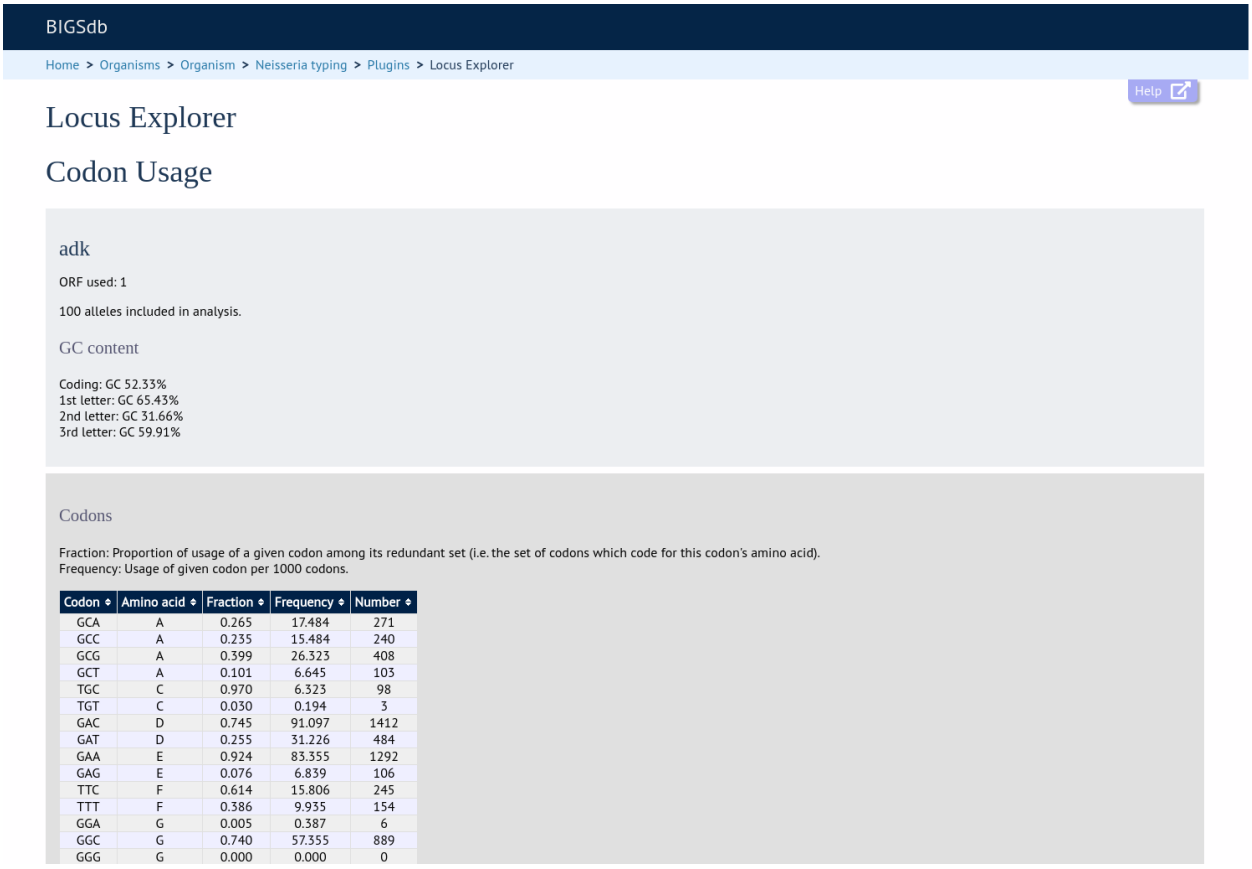
- *Investigating allele differences*.
- *Polymorphism analysis following isolate query*.

14.11.2 Codon usage

Select the alleles that you would like to include in the analysis. Again, variable length loci are limited to 200 sequences or fewer since these need to be aligned. Click 'Codon'.



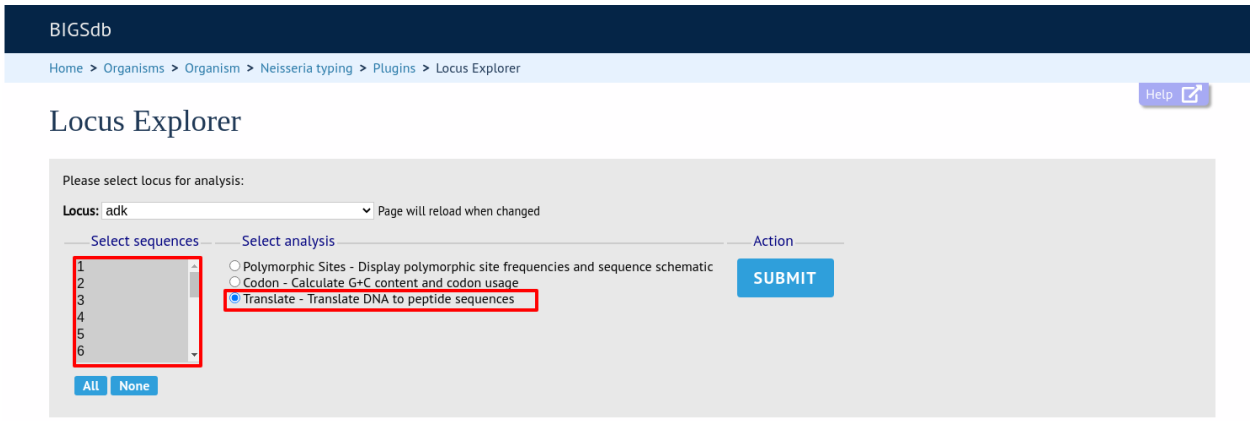
The GC content of the alleles will be determined and a table of the codon frequencies displayed.



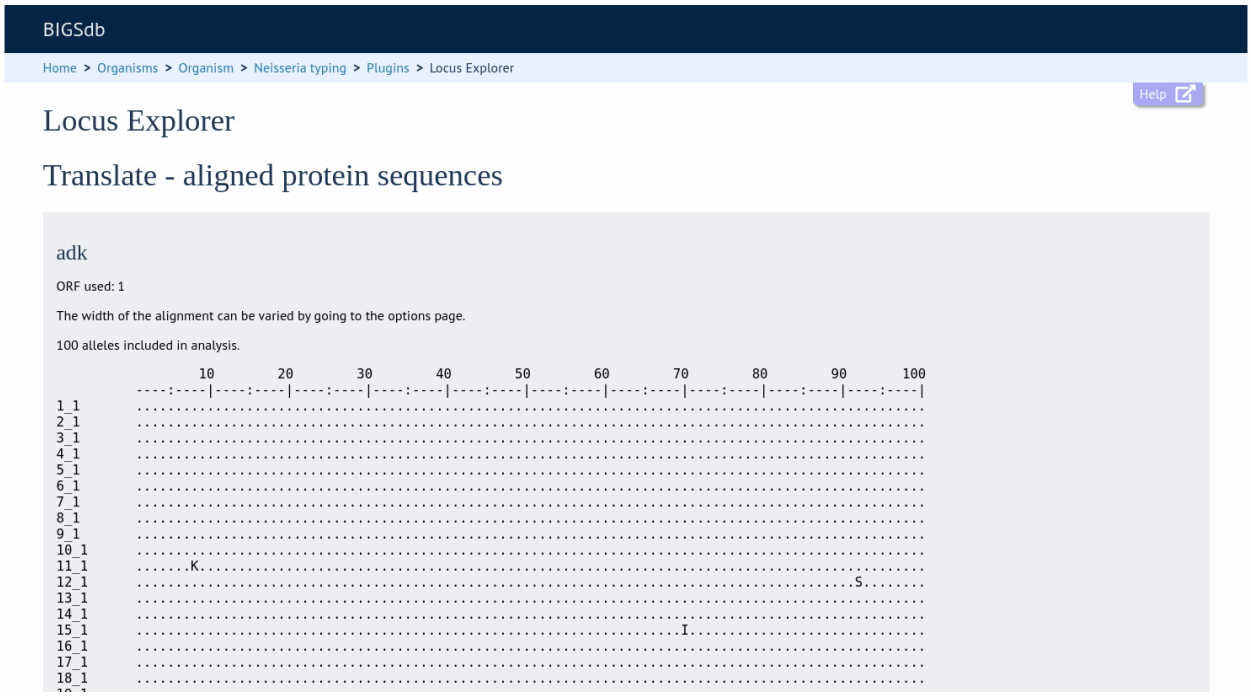
14.11.3 Aligned translations

If a DNA coding sequence locus is selected, an aligned translation can be produced.

Select the alleles that you would like to include in the analysis. Again, variable length loci are limited to 200 sequences or fewer since these need to be aligned. Click 'Translate'.



An aligned amino acid sequence will be displayed.



If there appear to be a lot of stop codons in the translation, it is possible that the orf value in the *locus definition* is not set correctly.

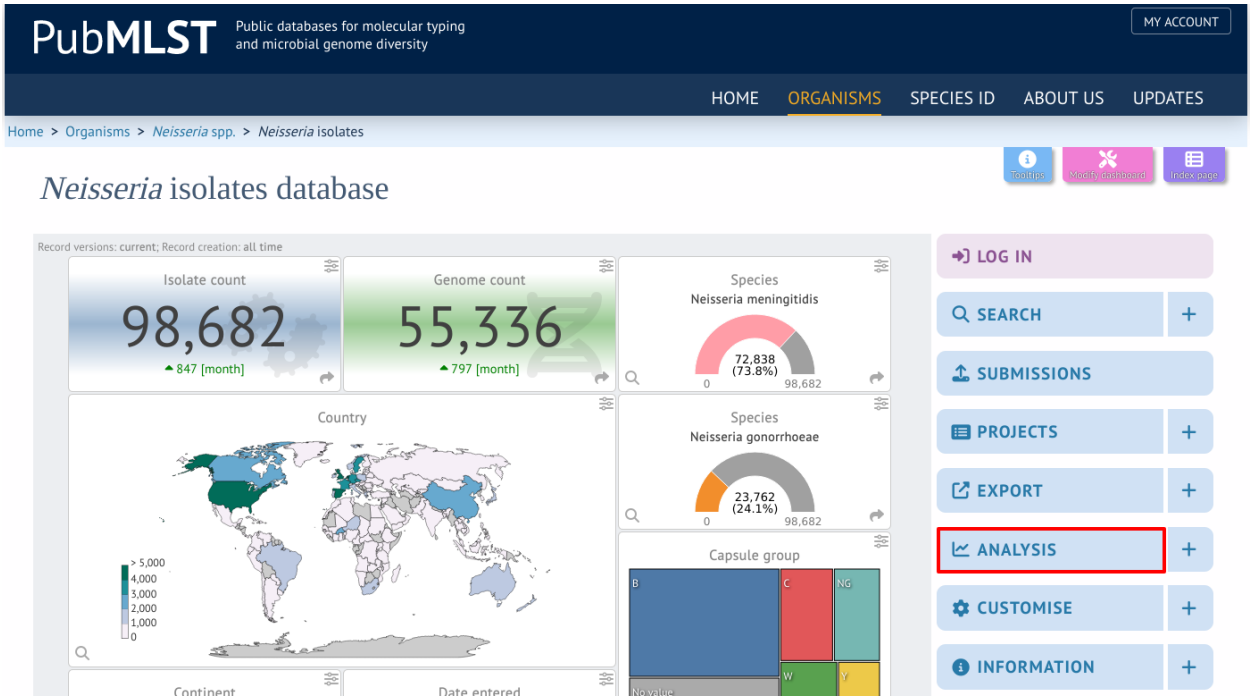
14.12 Microreact

Microreact is a tool for visualising genomic epidemiology and phylogeography. Individual nodes on a displayed tree are linked to nodes on a geographical map and/or timeline, making any spatial and temporal relationships between isolates apparent.

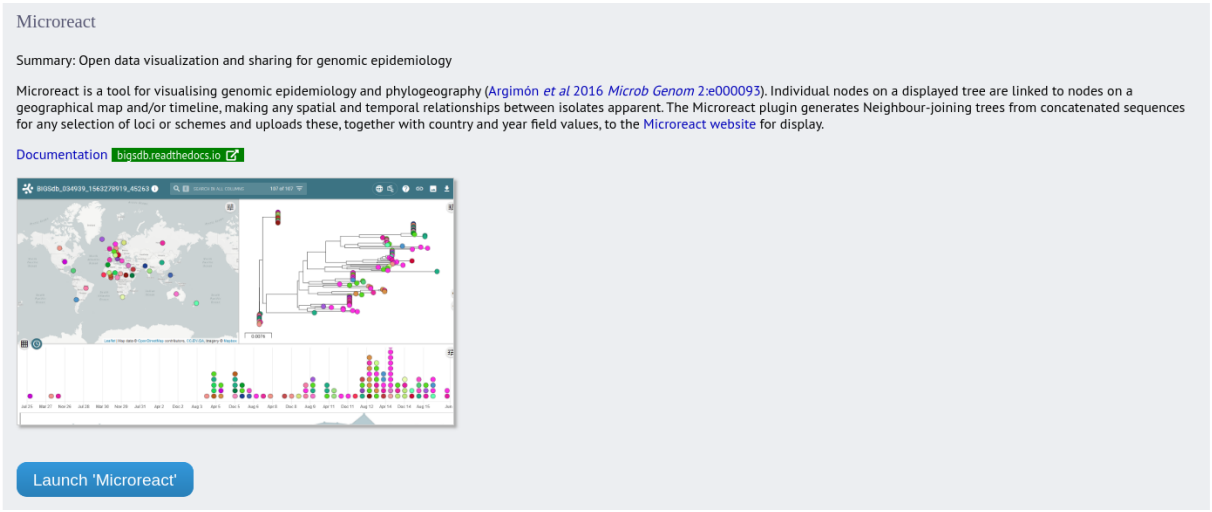
The Microreact plugin generates Neighbour-joining trees from concatenated sequences for any selection of loci or schemes and uploads these, together with country and year field values to the Microreact website for display.

Note: While Microreact itself is able to display isolates using GPS coordinates, the BIGSdb plugin is currently limited to the level of country.

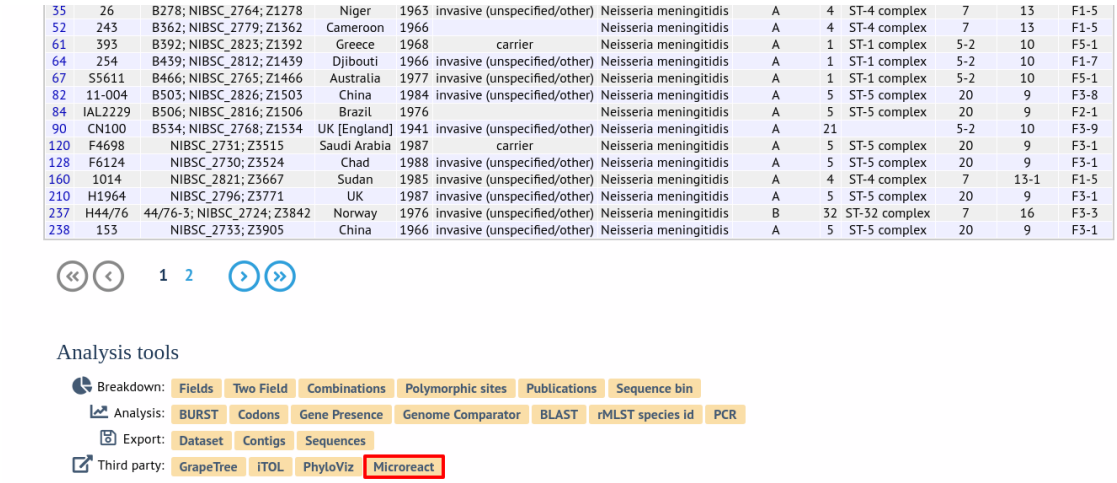
Microreact can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Third party' category, follow the link to BLAST, then click 'Launch Microreact'.

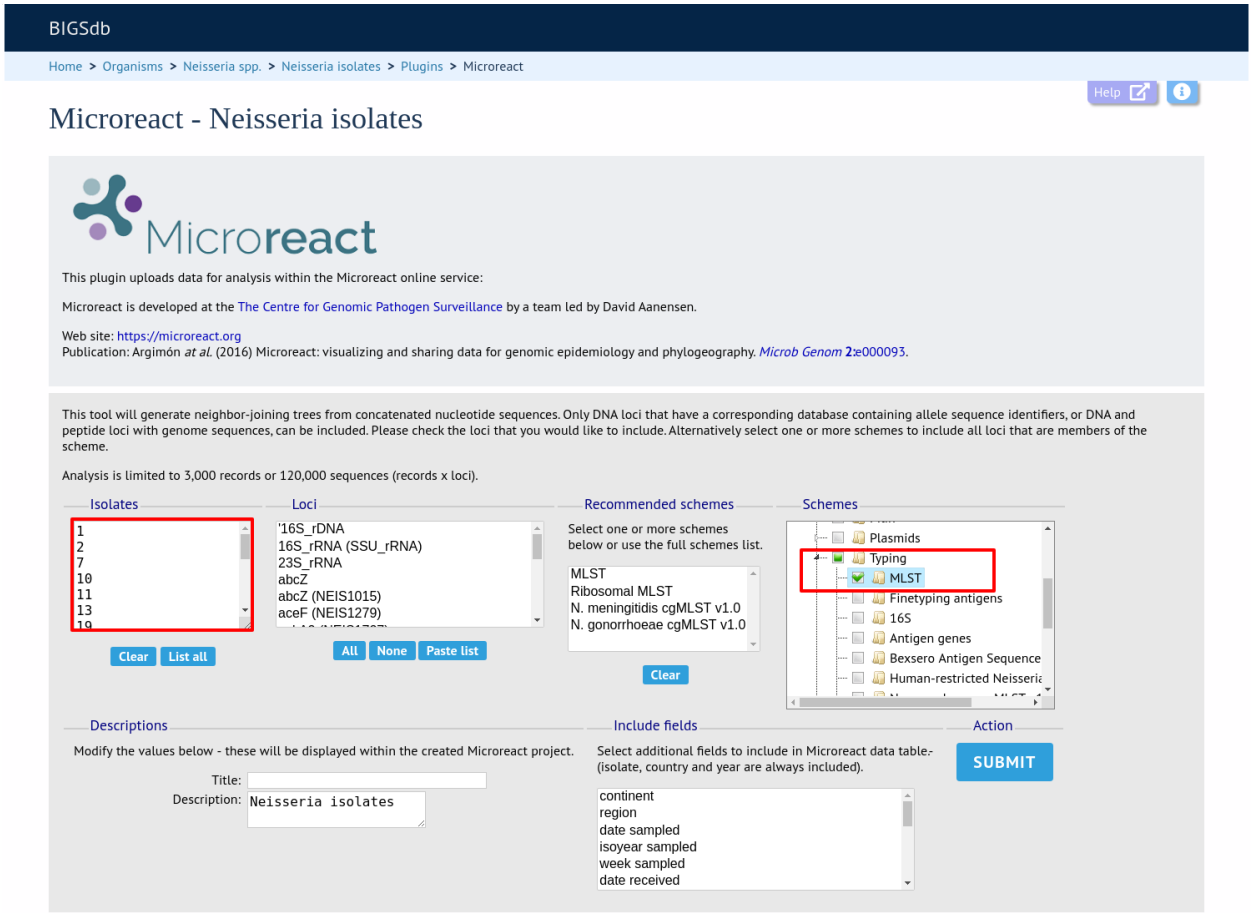


Alternatively, it can be accessed following a query by clicking the 'Microreact' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the Microreact plugin interface.

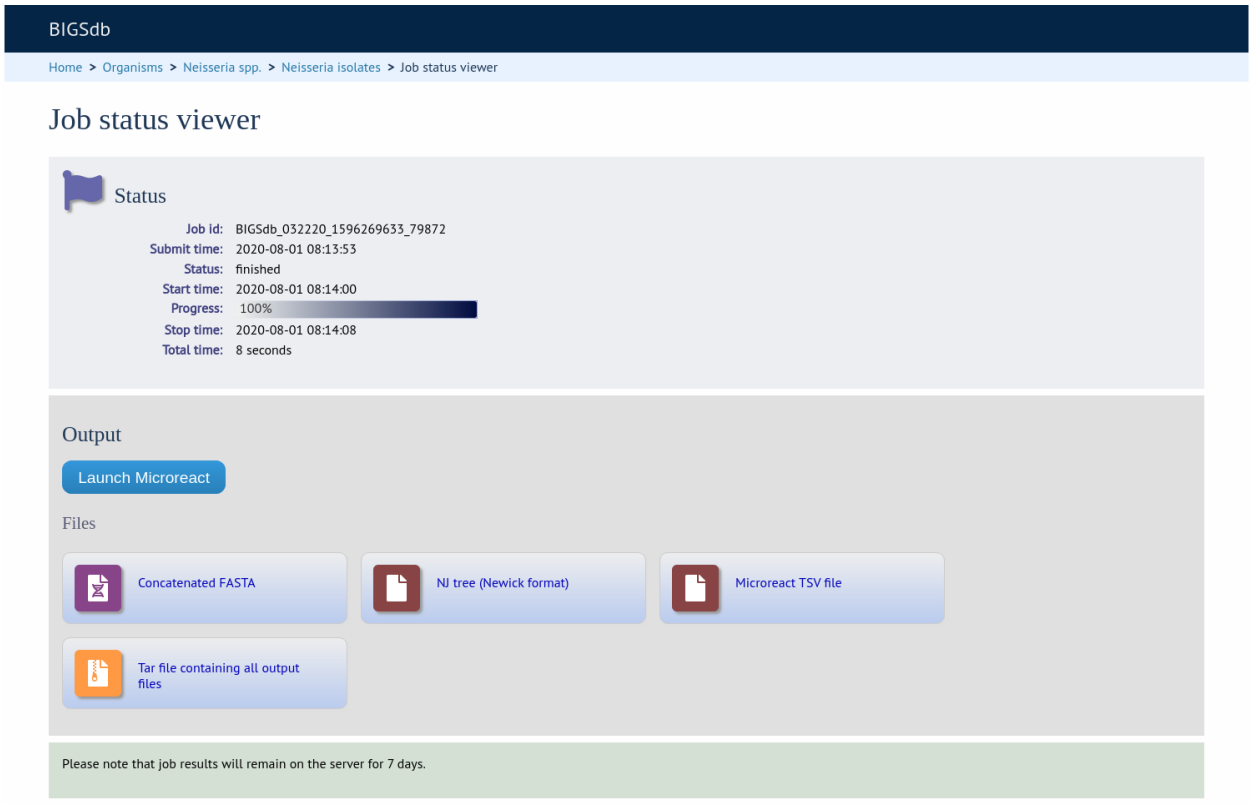


Select the isolates to include. The tree can be generated from allelic profiles of any selection of loci, or more conveniently, you can select a scheme in the scheme selector, or from a list of recommended schemes if these have been set, to include all loci belonging to that scheme.

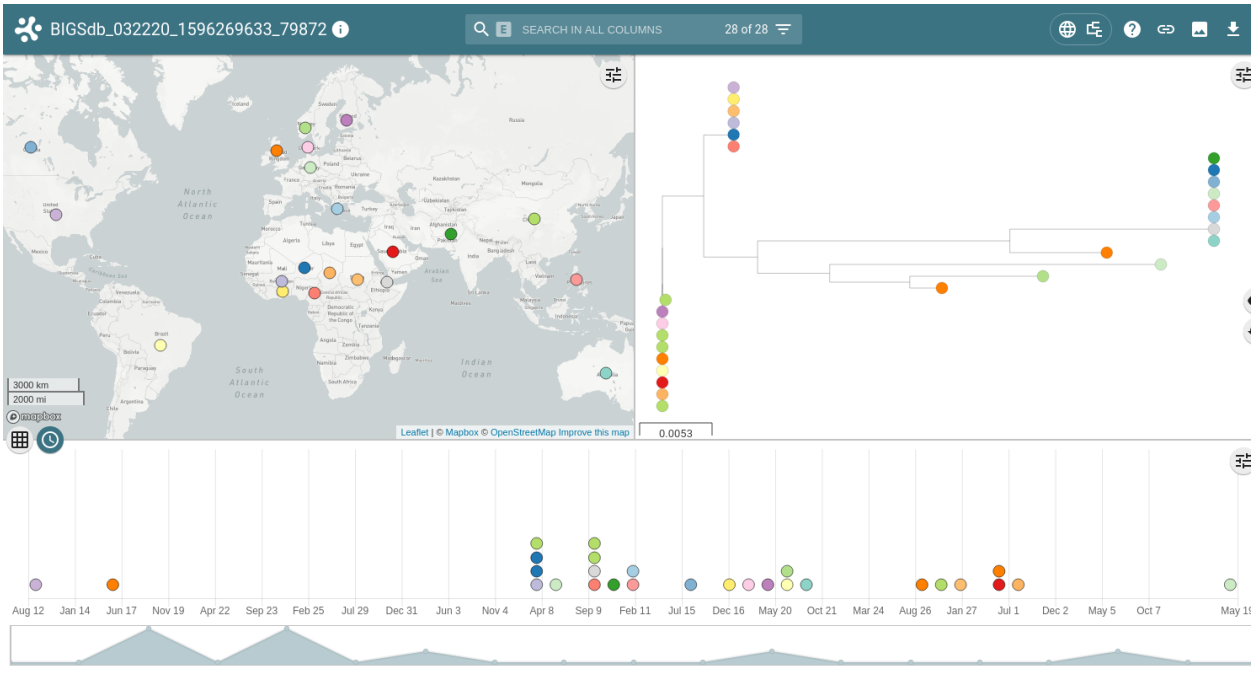
Additional fields can be selected to be included as metadata for use in colouring nodes - select any fields you wish to include. Multiple selections can be made by holding down shift or ctrl while selecting. Click 'Submit' to start the analysis.



The job will be sent to the job queue. When it has finished, click the button marked 'Launch Microreact'.



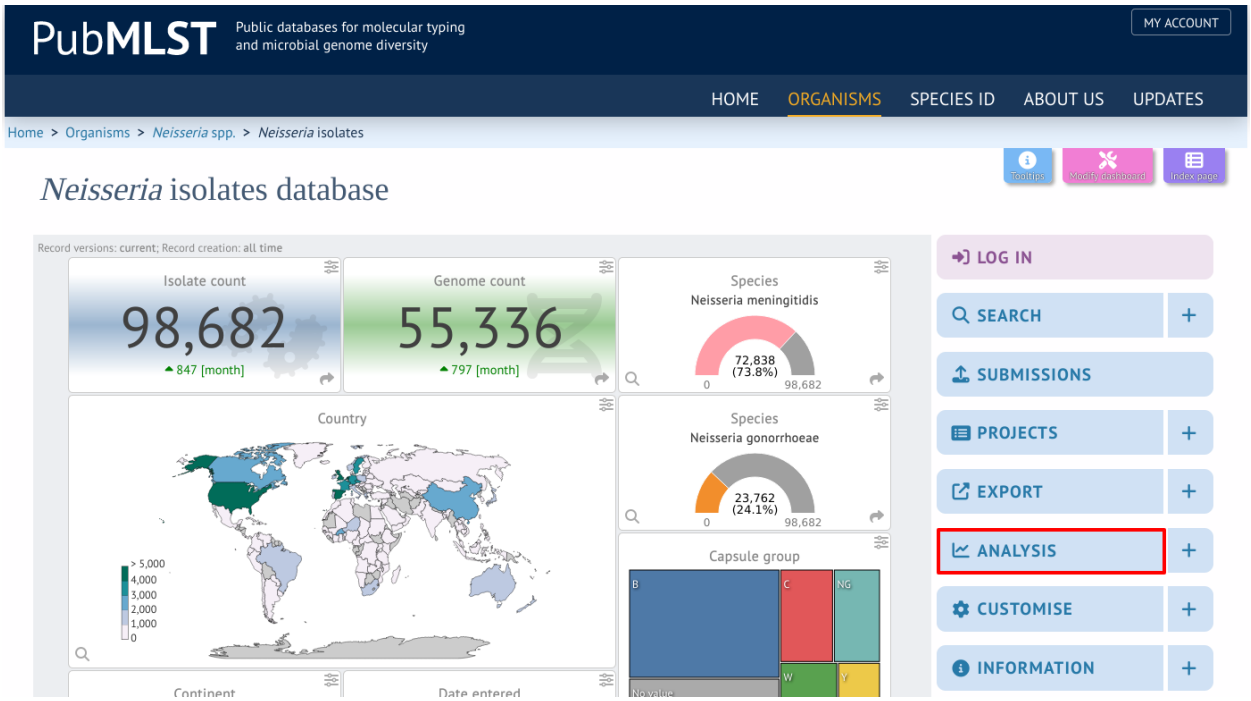
The generated tree will be uploaded to the Microreact website and displayed. Clicking any node will show its position(s) within the tree, map and timeline. A node on the map may correspond to multiple nodes in the tree or timeline.



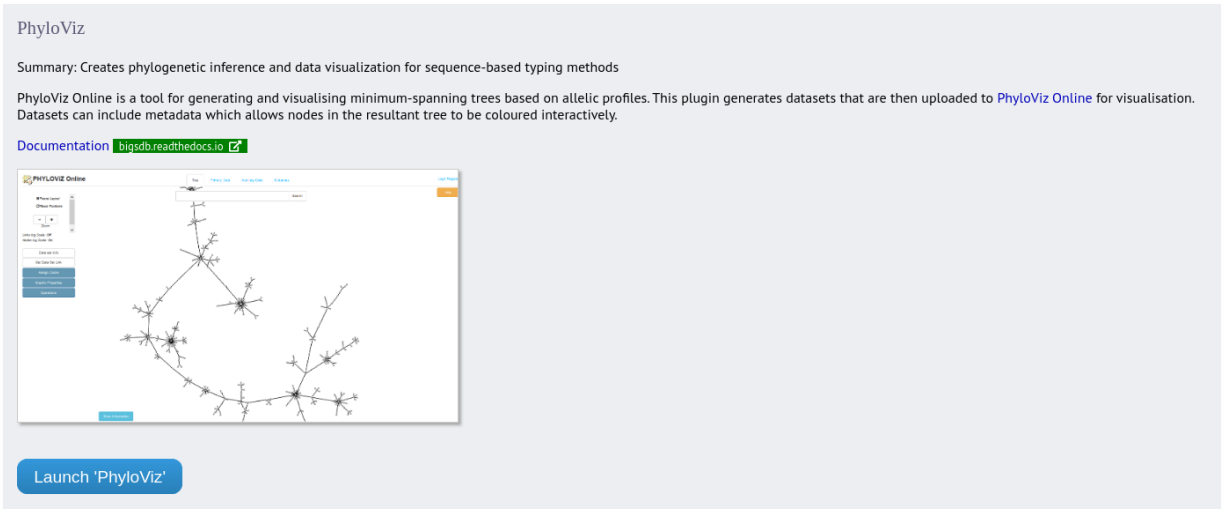
14.13 PhyloViz

PhyloViz Online is a tool for generating and visualising minimum-spanning trees. Datasets can include metadata which allows nodes in the resultant tree to be coloured interactively.

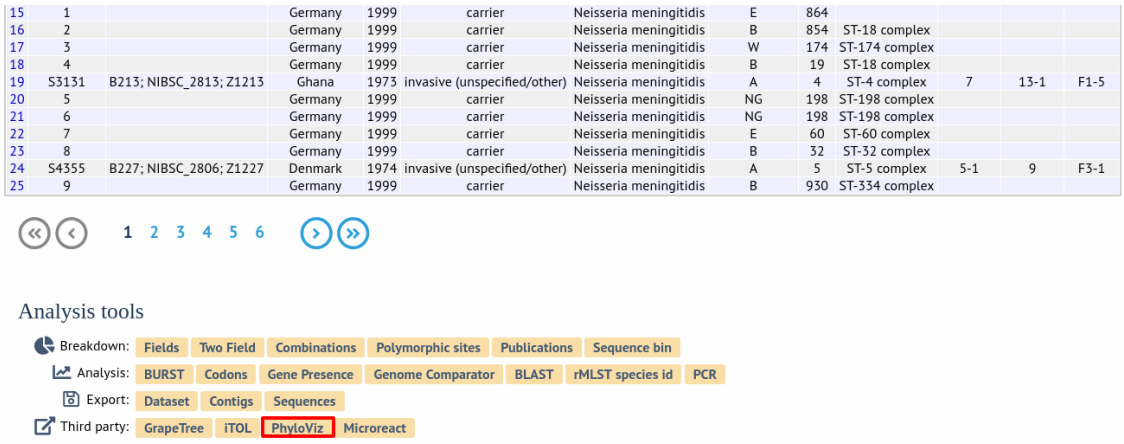
PhyloViz can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Third party' category, follow the link to PhyloViz, then click 'Launch PhyloViz'.

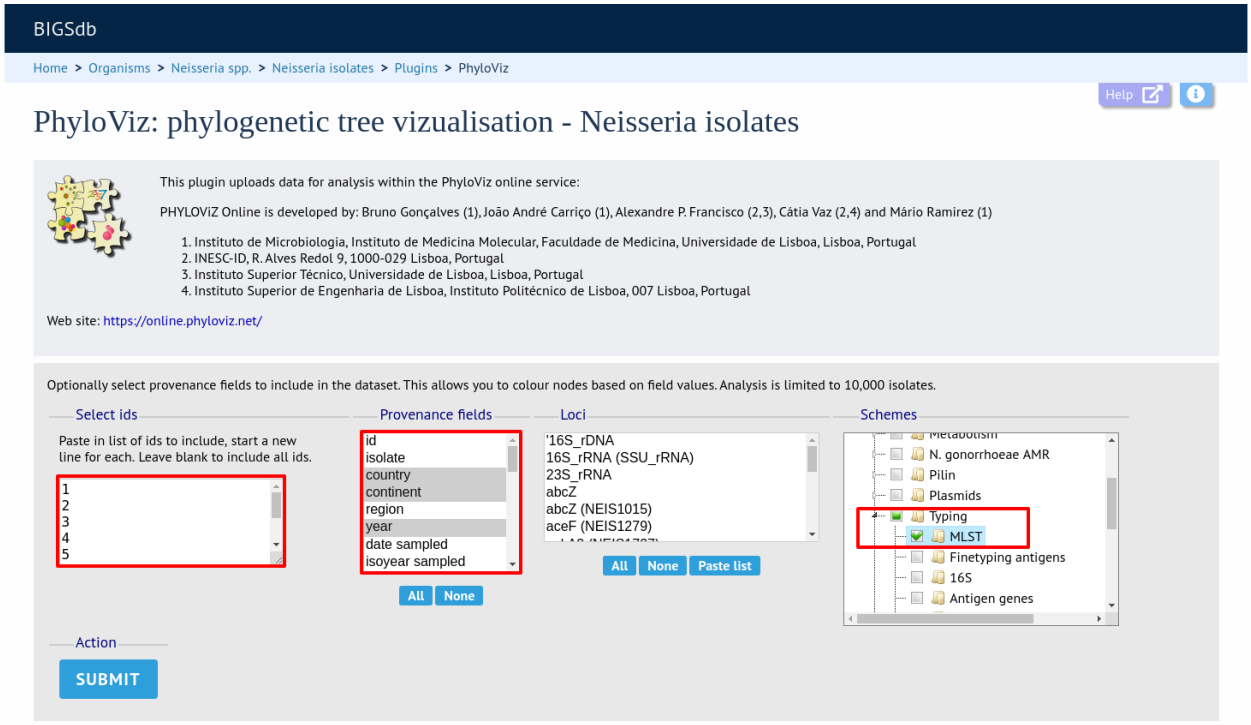


Alternatively, it can be accessed following a query by clicking the 'PhyloViz' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the PhyloViz interface.



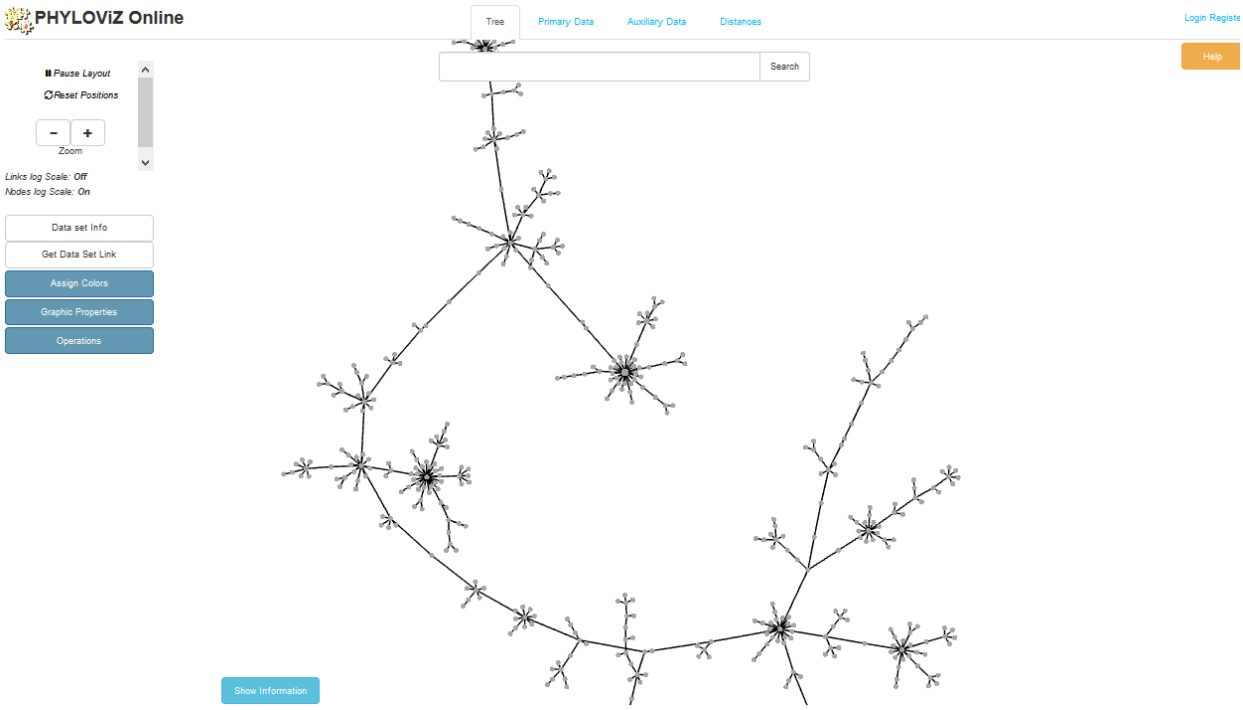
Select the isolates to include. The tree can be generated from allelic profiles of any selection of loci, or more conveniently, you can select a scheme in the scheme selector to include all loci belong to that scheme.

Provenance fields can be selected to be included as metadata for use in colouring nodes - select any fields you wish to include. Multiple selections can be made by holding down Shift or Ctrl while selecting. Click 'Submit' to start the analysis.



The necessary files will be generated immediately. When this has finished, click the button launch 'Launch PhyloViz'.

The tree will be sent to and rendered within the PhyloViz website.

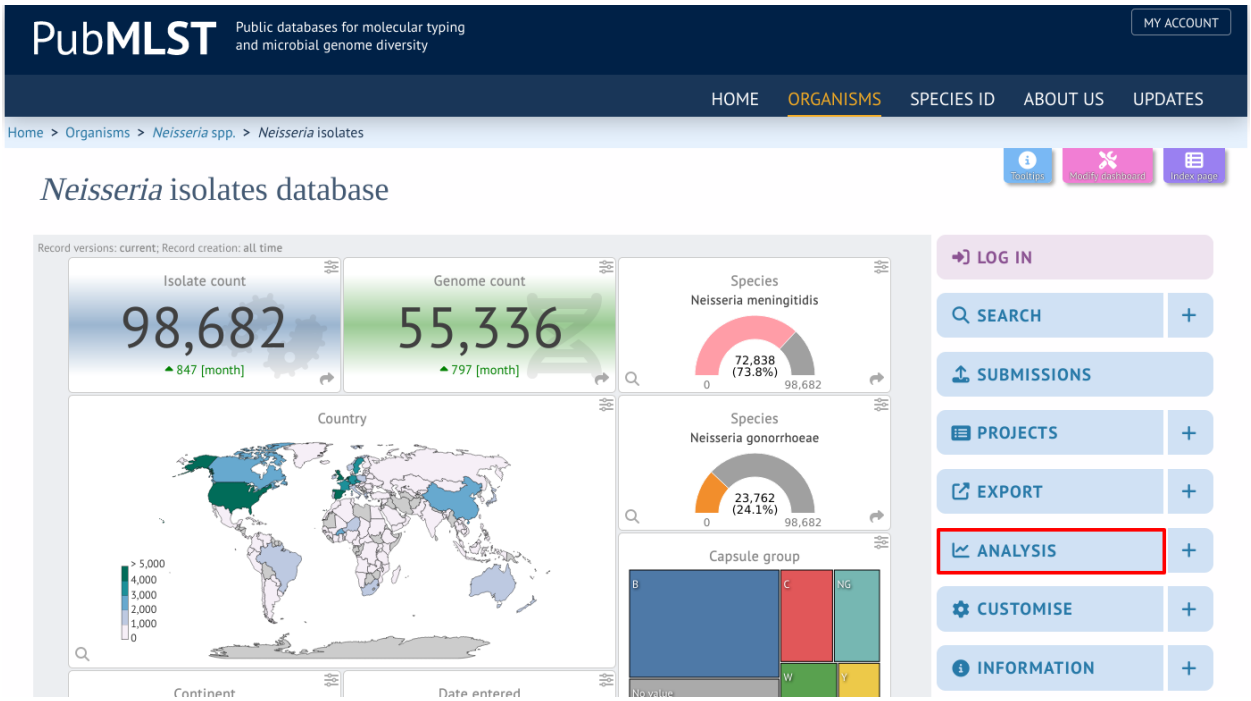


See more information about manipulating the tree on the PhyloViz website.

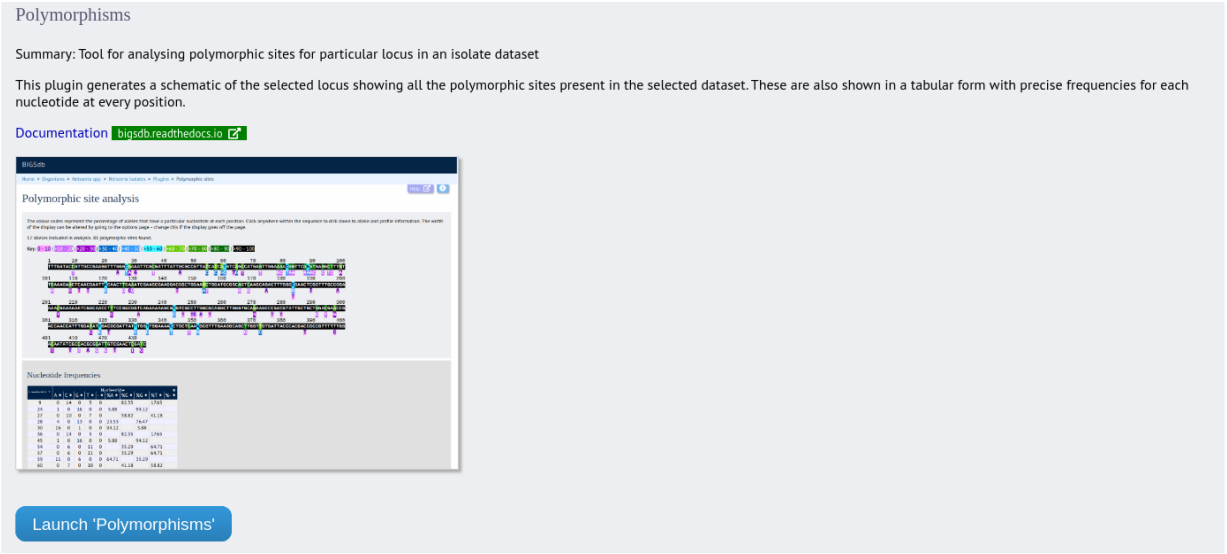
14.14 Polymorphisms

The Polymorphisms plugin generates a *Locus Explorer* polymorphic site analysis on the alleles designated in an isolate dataset following a query.

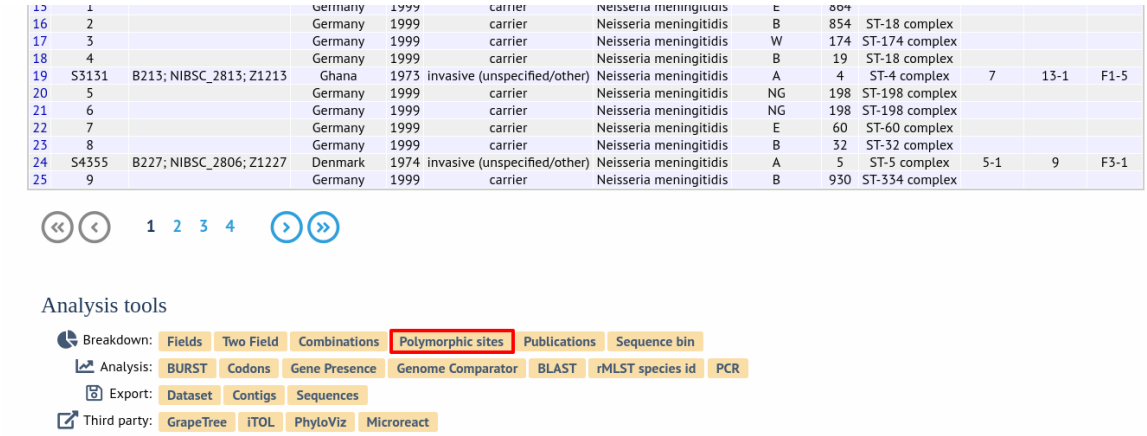
The analysis can be accessed by selecting the 'Analysis' section on the main contents page.



Jump to the 'Breakdown' category, follow the link to Polymorphisms, then click 'Launch Polymorphisms'.

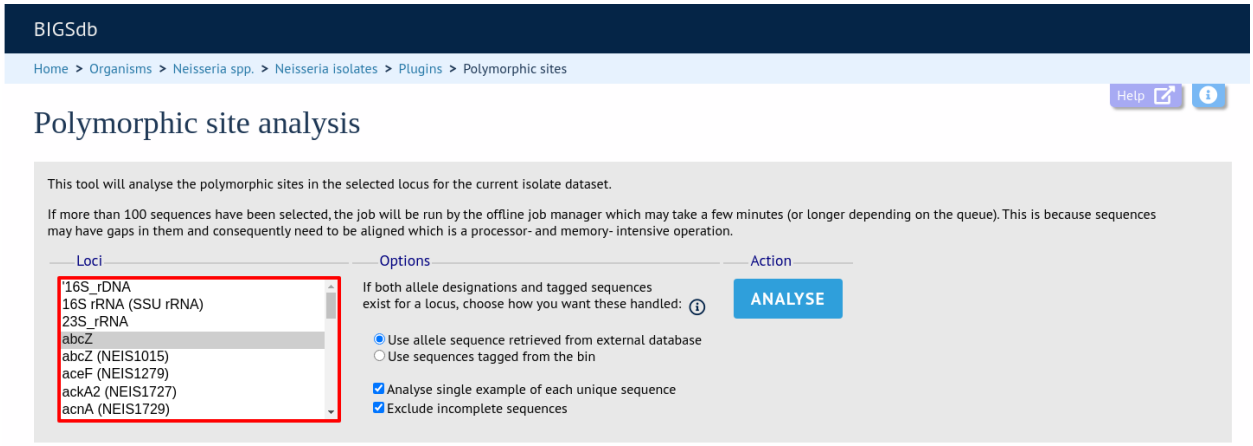


The analysis is accessed by clicking the 'Polymorphic sites' button in the Breakdown list at the bottom of a results table following a query.



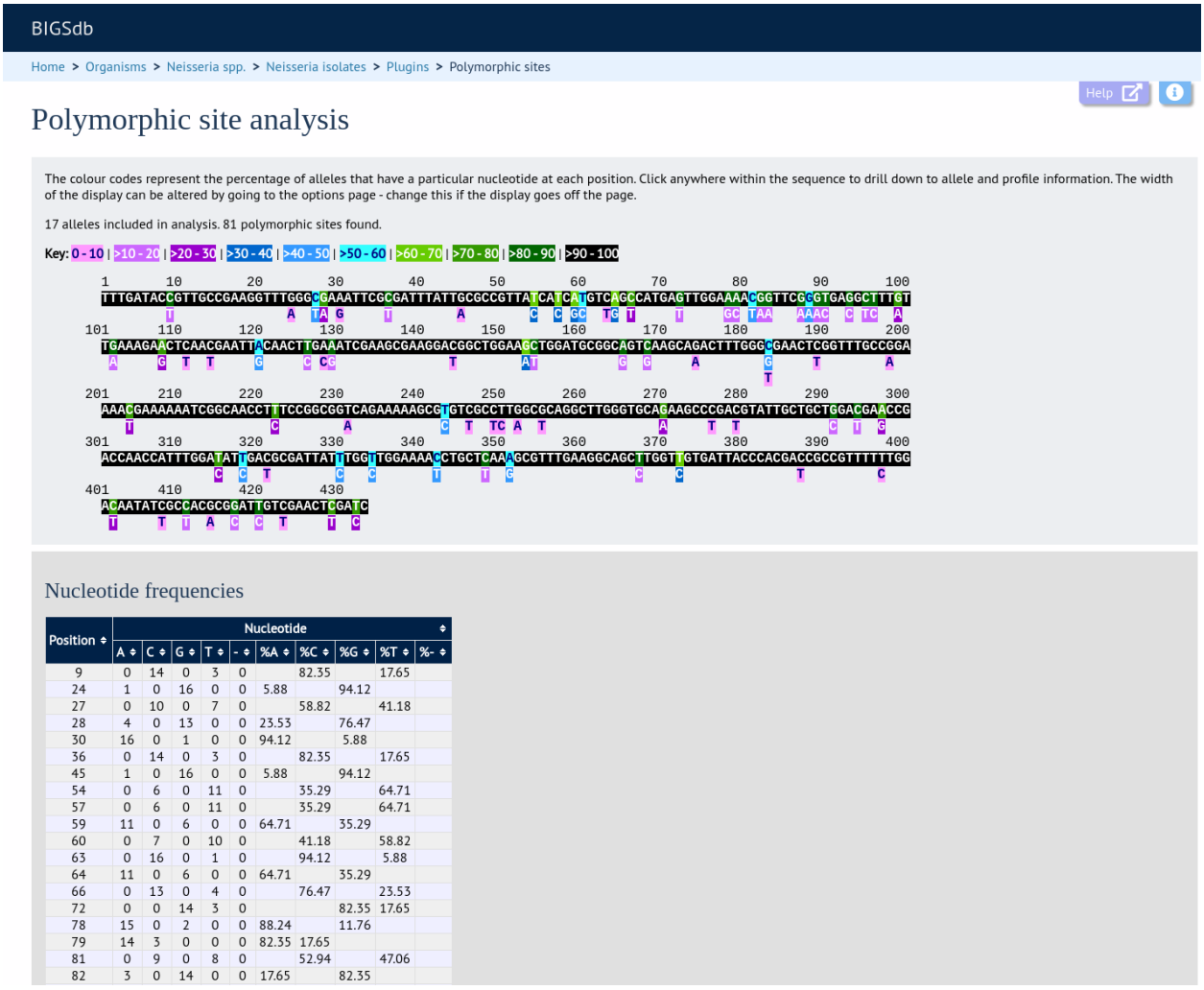
16	2		Germany	1999	carrier	Neisseria meningitidis	B	854	ST-18 complex			
17	3		Germany	1999	carrier	Neisseria meningitidis	W	174	ST-174 complex			
18	4		Germany	1999	carrier	Neisseria meningitidis	B	19	ST-18 complex			
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
20	5		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
21	6		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
22	7		Germany	1999	carrier	Neisseria meningitidis	E	60	ST-60 complex			
23	8		Germany	1999	carrier	Neisseria meningitidis	B	32	ST-32 complex			
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
25	9		Germany	1999	carrier	Neisseria meningitidis	B	930	ST-334 complex			

Select the locus that you would like to analyse from the list.



Click 'Analyse'.

A schematic of the locus is generated showing the polymorphic sites. A full description of this can be found in the *Locus Explorer polymorphic site analysis* section.

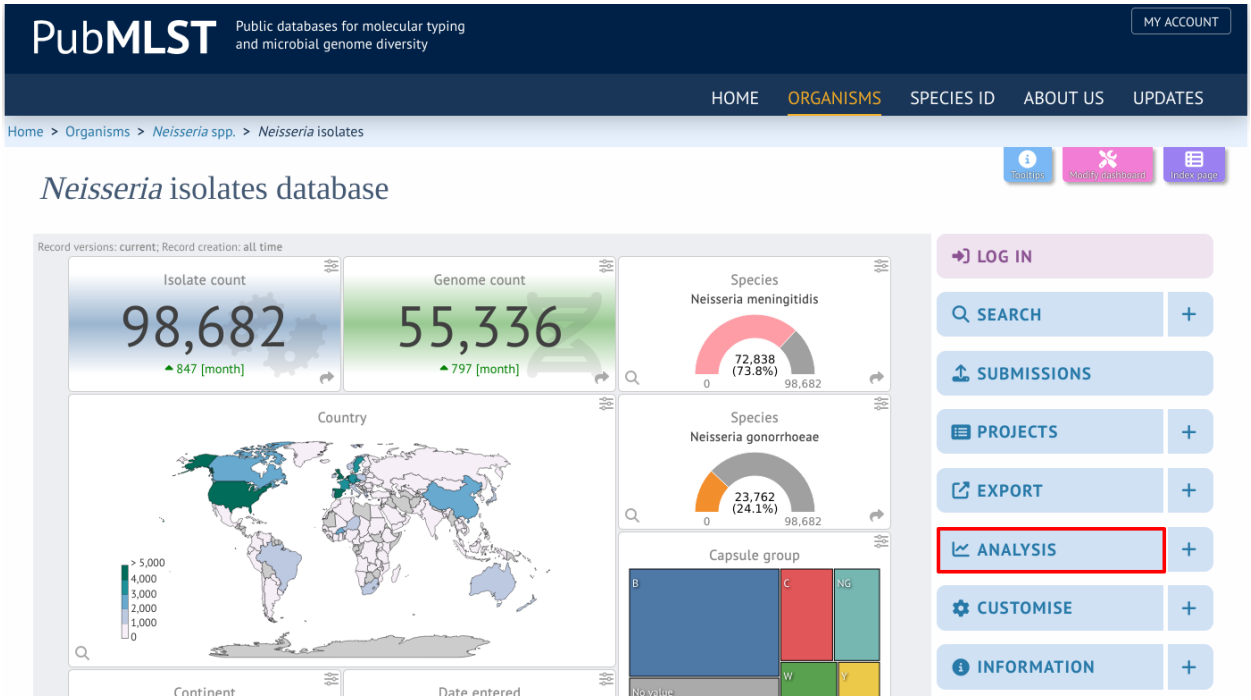


Position	Nucleotide									
	A	C	G	T	-	%A	%C	%G	%T	%-
9	0	14	0	3	0		82.35		17.65	
24	1	0	16	0	0	5.88		94.12		
27	0	10	0	7	0		58.82		41.18	
28	4	0	13	0	0	23.53		76.47		
30	16	0	1	0	0	94.12		5.88		
36	0	14	0	3	0		82.35		17.65	
45	1	0	16	0	0	5.88		94.12		
54	0	6	0	11	0		35.29		64.71	
57	0	6	0	11	0		35.29		64.71	
59	11	0	6	0	0	64.71		35.29		
60	0	7	0	10	0		41.18		58.82	
63	0	16	0	1	0		94.12		5.88	
64	11	0	6	0	0	64.71		35.29		
66	0	13	0	4	0		76.47		23.53	
72	0	0	14	3	0			82.35	17.65	
78	15	0	2	0	0	88.24		11.76		
79	14	3	0	0	0	82.35	17.65			
81	0	9	0	8	0		52.94		47.06	
82	3	0	14	0	0	17.65		82.35		

14.15 Species identification

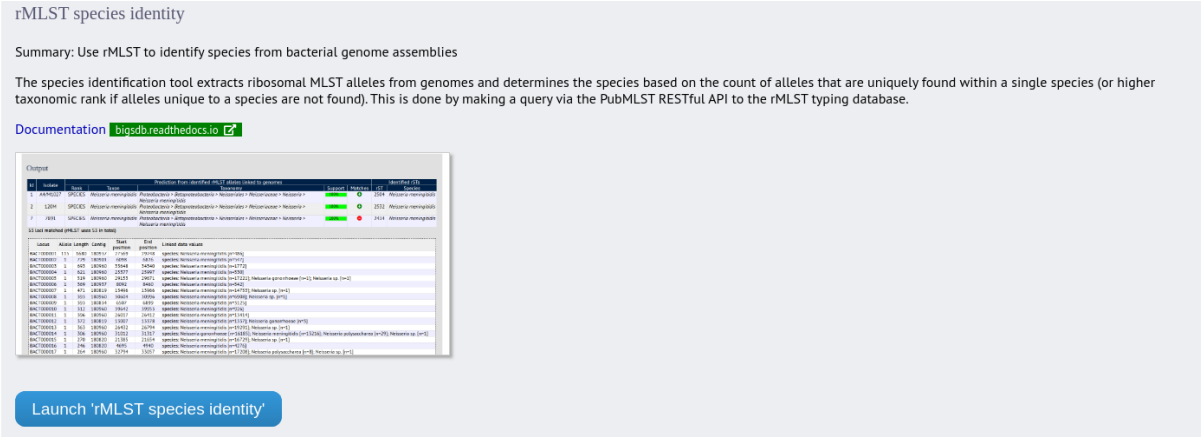
The species identification tool extracts ribosomal MLST alleles from genomes and determines the species based on the count of alleles that are uniquely found within a single species (or higher taxonomic rank if alleles unique to a species are not found). This is done by making a query to the rMLST genome database.

The tool can be accessed by selecting the 'Analysis' section on the main contents page.

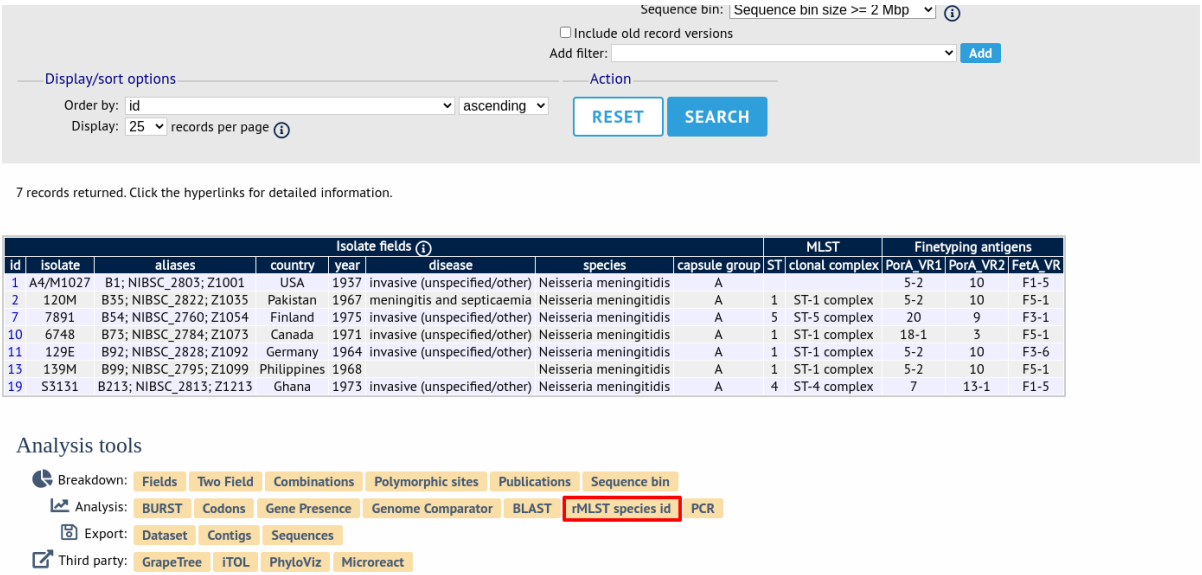


Jump to the 'Analysis' category, follow the link to rMLST species identity, then click 'Launch rMLST species identity'.

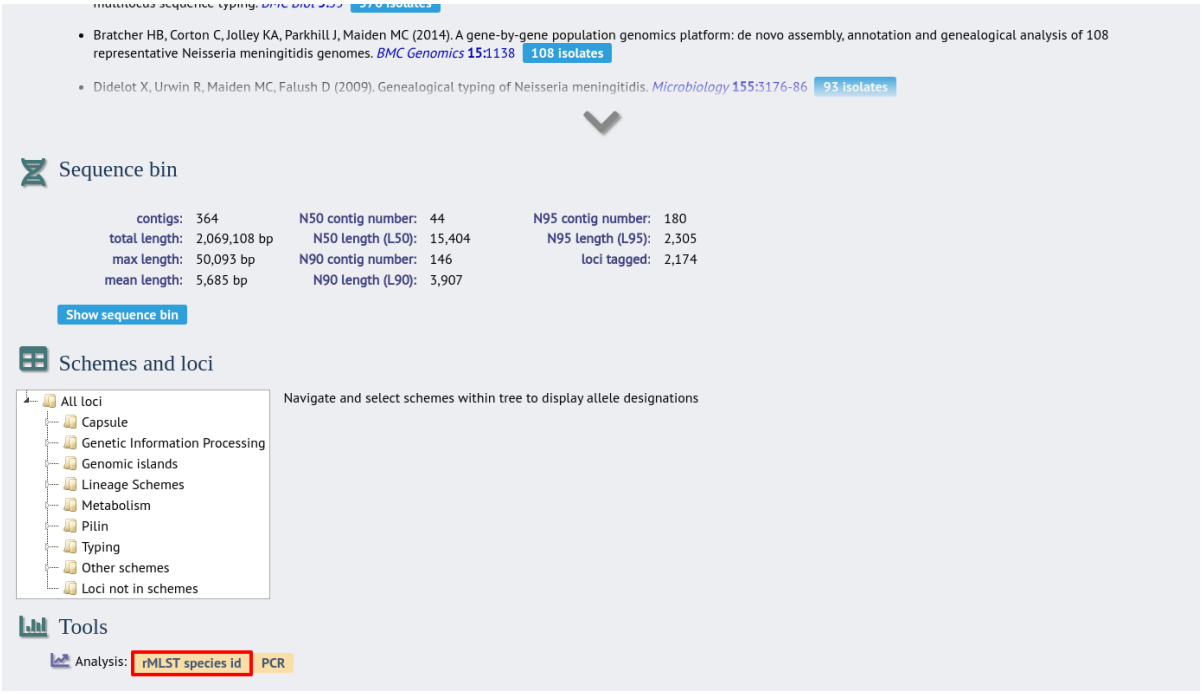
The tool can be accessed from the front page of an isolate database.



Alternatively, it can be accessed following a query by clicking the 'rMLST species id' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the species id interface (note that only isolates with a genome assembly will be able to be checked).

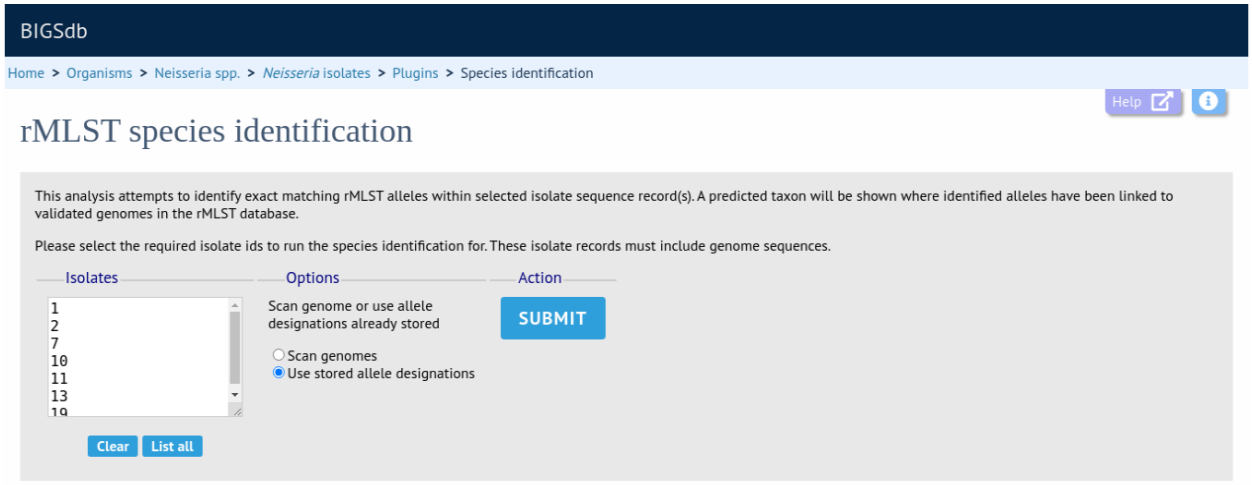


Finally, the analysis is also possible directly from an isolate record, if the isolate has a genome assembly associated with it.



The tool interface consists of a list of isolate ids to check. This will be pre-populated if accessed following a query or directly from an isolate record. If the rMLST scheme is defined on the system, you will have a choice as to whether to BLAST the genome sequences to identify the rMLST alleles, or just use the designations that are tagged in the database. The latter is much quicker but relies on the record having been scanned and annotated with the rMLST loci.

Click 'Submit'.

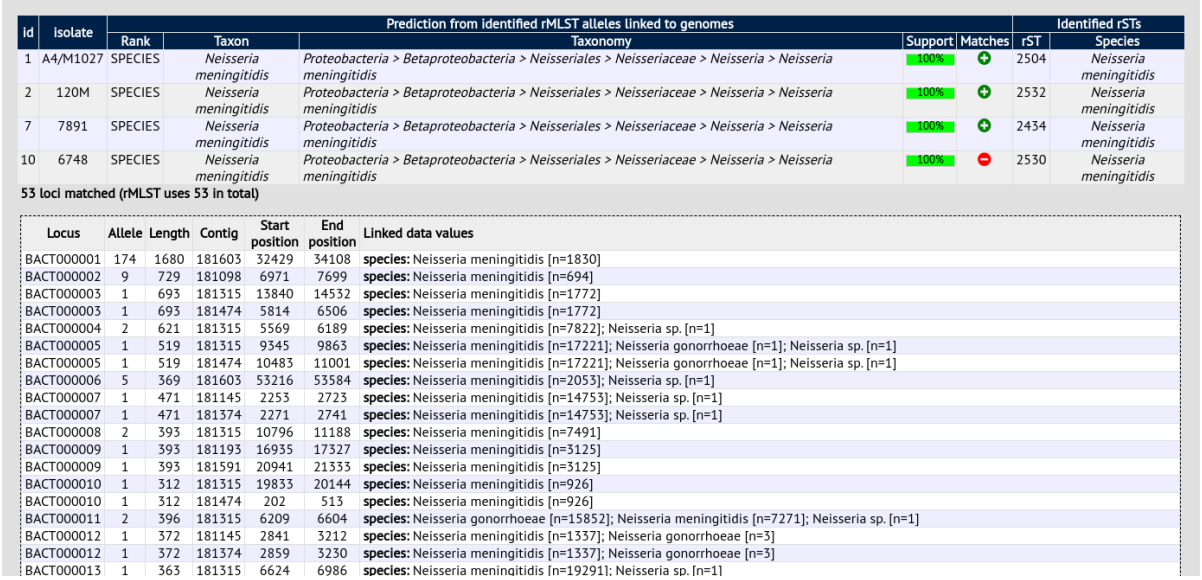


The job will be sent to the job queue.

Results will be displayed in a table as they are generated. The table will display the highest taxonomic rank that can be reliably identified, e.g. species, the taxon and its full taxonomy. An indication of the confidence for the result will also be displayed - this is based on the proportion of alleles found that are unique to a taxon.

id	isolate	Prediction from identified rMLST alleles linked to genomes					Identified rSTs	
		Rank	Taxon	Taxonomy	Support	Matches	rST	Species
1	A4/M1027	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2504	*Neisseria meningitidis*
2	120M	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2532	*Neisseria meningitidis*
7	7891	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2434	*Neisseria meningitidis*
10	6748	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2530	*Neisseria meningitidis*
11	129E	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2528	*Neisseria meningitidis*
13	139M	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2433	*Neisseria meningitidis*
19	S3131	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2285	*Neisseria meningitidis*

Clicking the '+' icon on any row will display further details about the matches.



id	isolate	Prediction from identified rMLST alleles linked to genomes					Identified rSTs	
		Rank	Taxon	Taxonomy	Support	Matches	rST	Species
1	A4/M1027	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2504	*Neisseria meningitidis*
2	120M	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2532	*Neisseria meningitidis*
7	7891	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2434	*Neisseria meningitidis*
10	6748	SPECIES	*Neisseria meningitidis*	*Proteobacteria > Betaproteobacteria > Neisseriales > Neisseriaceae > Neisseria > Neisseria meningitidis*	100%		2530	*Neisseria meningitidis*

53 loci matched (rMLST uses 53 in total)

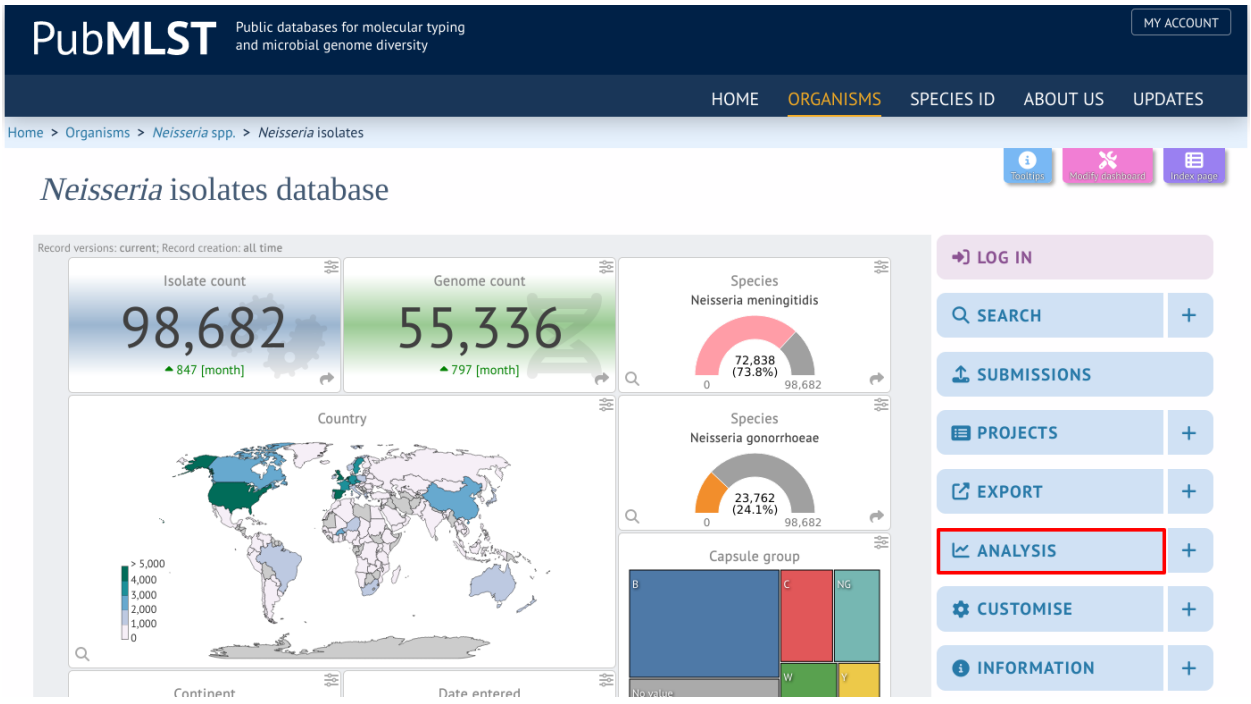
Locus	Allele	Length	Contig	Start position	End position	Linked data values
BACT000001	174	1680	181603	32429	34108	**species:** Neisseria meningitidis [n=1830]
BACT000002	9	729	181098	6971	7699	**species:** Neisseria meningitidis [n=694]
BACT000003	1	693	181315	13840	14532	**species:** Neisseria meningitidis [n=1772]
BACT000003	1	693	181474	5814	6506	**species:** Neisseria meningitidis [n=1772]
BACT000004	2	621	181315	5569	6189	**species:** Neisseria meningitidis [n=7822]; Neisseria sp. [n=1]
BACT000005	1	519	181315	9345	9863	**species:** Neisseria meningitidis [n=17221]; Neisseria gonorrhoeae [n=1]; Neisseria sp. [n=1]
BACT000005	1	519	181474	10483	11001	**species:** Neisseria meningitidis [n=17221]; Neisseria gonorrhoeae [n=1]; Neisseria sp. [n=1]
BACT000006	5	369	181603	53216	53584	**species:** Neisseria meningitidis [n=2053]; Neisseria sp. [n=1]
BACT000007	1	471	181145	2253	2723	**species:** Neisseria meningitidis [n=14753]; Neisseria sp. [n=1]
BACT000007	1	471	181374	2271	2741	**species:** Neisseria meningitidis [n=14753]; Neisseria sp. [n=1]
BACT000008	2	393	181315	10796	11188	**species:** Neisseria meningitidis [n=7491]
BACT000009	1	393	181193	16935	17327	**species:** Neisseria meningitidis [n=3125]
BACT000009	1	393	181591	20941	21333	**species:** Neisseria meningitidis [n=3125]
BACT000010	1	312	181315	19833	20144	**species:** Neisseria meningitidis [n=926]
BACT000010	1	312	181474	202	513	**species:** Neisseria meningitidis [n=926]
BACT000011	2	396	181315	6209	6604	**species:** Neisseria gonorrhoeae [n=15852]; Neisseria meningitidis [n=7271]; Neisseria sp. [n=1]
BACT000012	1	372	181145	2841	3212	**species:** Neisseria meningitidis [n=1337]; Neisseria gonorrhoeae [n=3]
BACT000012	1	372	181374	2859	3230	**species:** Neisseria meningitidis [n=1337]; Neisseria gonorrhoeae [n=3]
BACT000013	1	363	181315	6624	6986	**species:** Neisseria meningitidis [n=19291]; Neisseria sp. [n=1]

Note: Ribosomal MLST was first described in Jolley et al. 2012. Ribosomal multilocus sequence typing: universal characterization of bacteria from domain to strain. *Microbiology* 158:1005-15

14.16 ReporTree

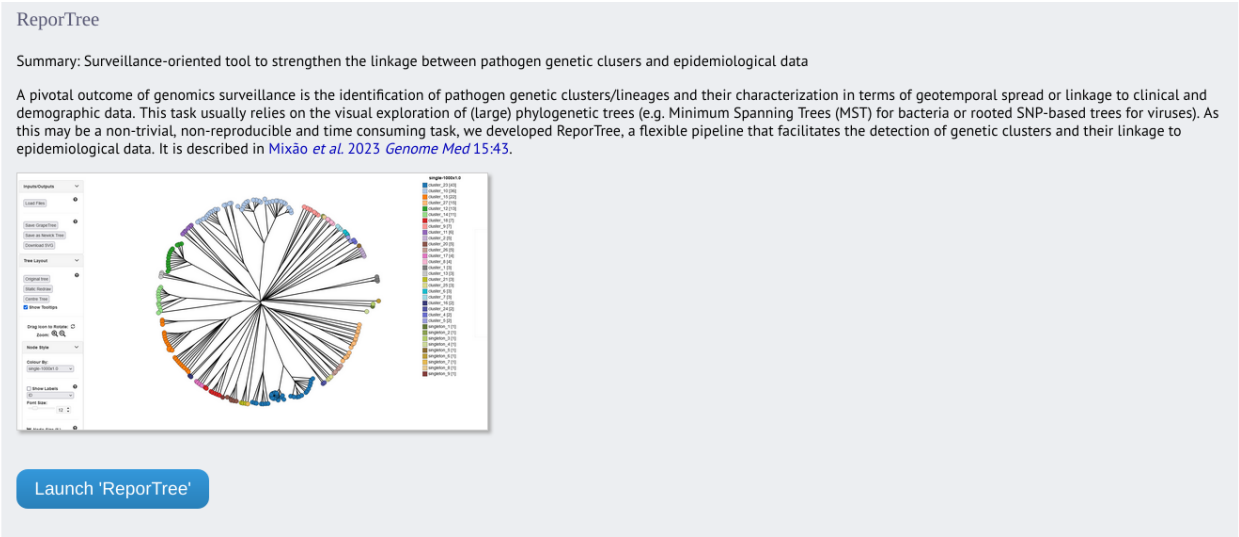
ReporTree is a pan-pathogen tool for automated and reproducible identification and characterization of genetic clusters.

ReporTree can be accessed by selecting the 'Analysis' section on the main contents page.

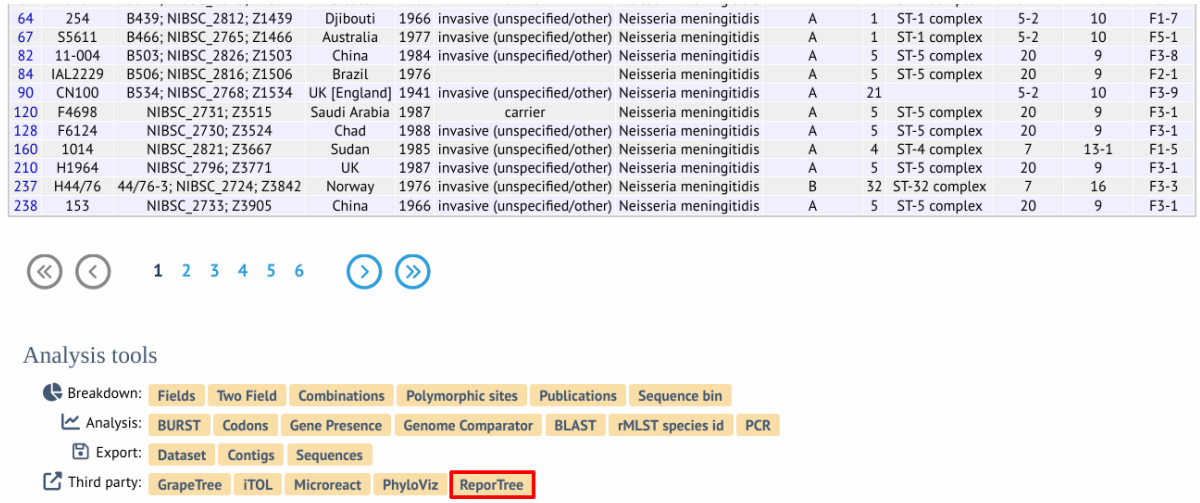


Jump to the 'Third party' category, follow the link to ReporTree, then click 'Launch ReporTree'.

ReporTree can be accessed from the contents page by clicking the 'ReporTree' link.



Alternatively, it can be accessed following a query by clicking the 'ReporTree' button at the bottom of the results table. Isolates returned from the query will be automatically selected within the GrapeTree interface.



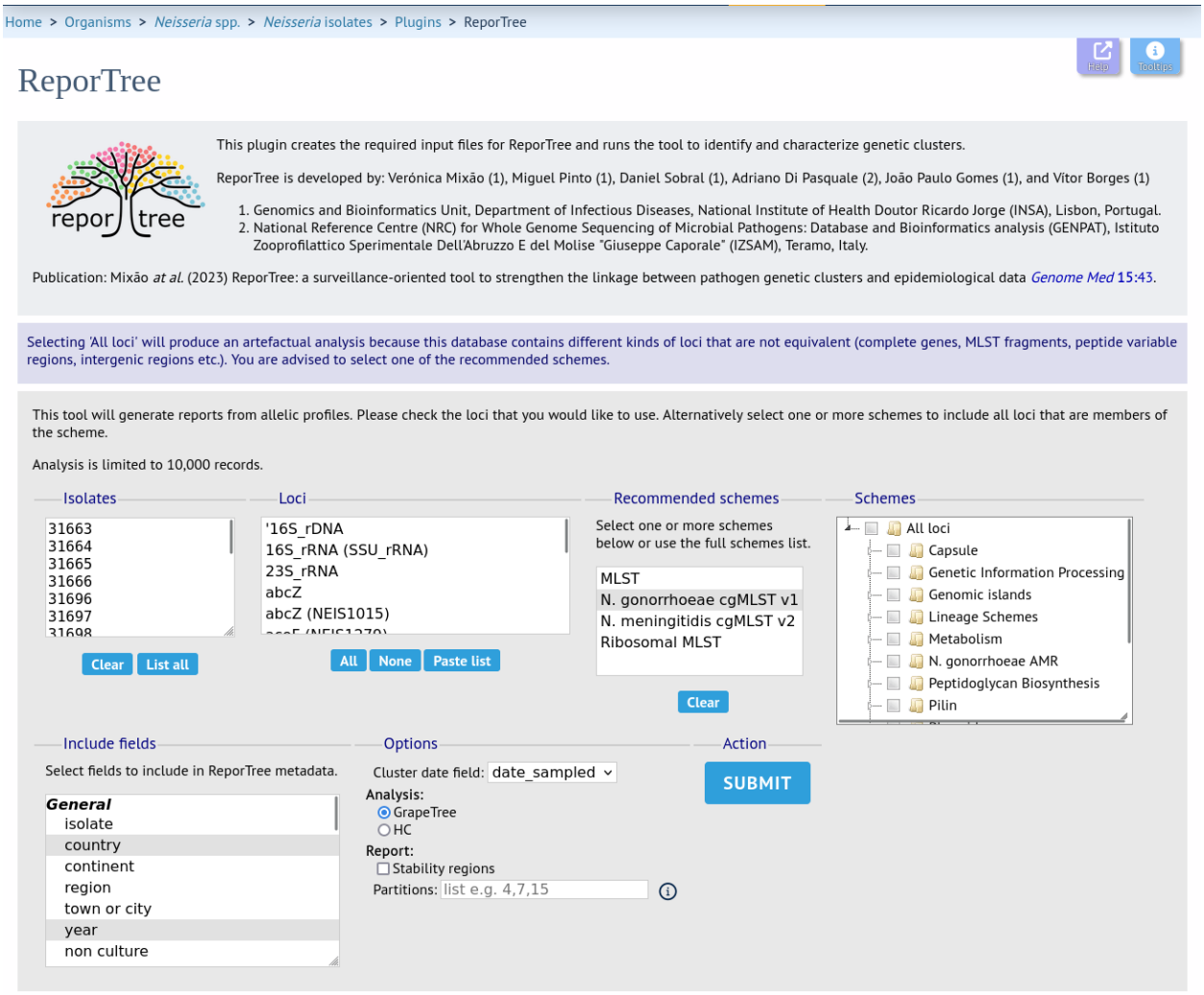
Select the isolates to include. Trees and clusters can be generated from allelic profiles of any selection of loci, or more conveniently, you can select a scheme in the scheme selector, or choose from recommended schemes if these have been set, to include all loci belonging to that scheme.

Additional fields can be selected to be included as metadata for use in colouring nodes and for inclusion in the partition summary - select any fields you wish to include. Multiple selections can be made by holding down shift or ctrl while selecting.

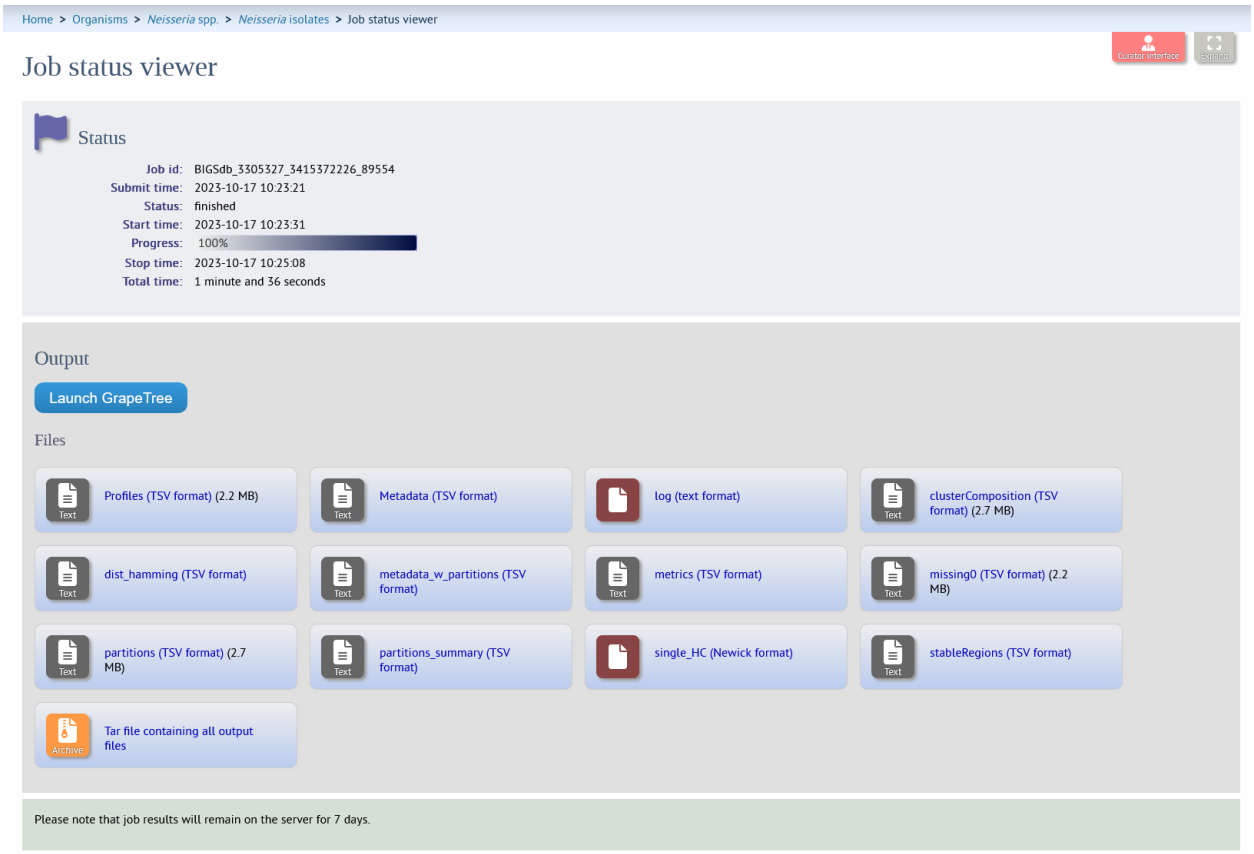
You can also select any date field which will be used in cluster definitions to indicate the first and last dates of the cluster as well as its total duration. This should ideally represent the date of sampling.

Finally, you can choose to use GrapeTree and HC methods for clustering and a comma-separated list of partitions defined by locus differences, e.g. '4,7,15'. Alternatively, you can choose to calculate and display 'stability regions' which will include the first partition of each stability region (threshold ranges in which cluster composition is similar).

Click 'Submit' to start the analysis.



The job will be sent to the job queue. When it has finished, you should see a link to load the resultant tree in GrapeTree and a list of output files.



- *See further information about using GrapeTree.*

Note: ReporTree has been described in the following publication:

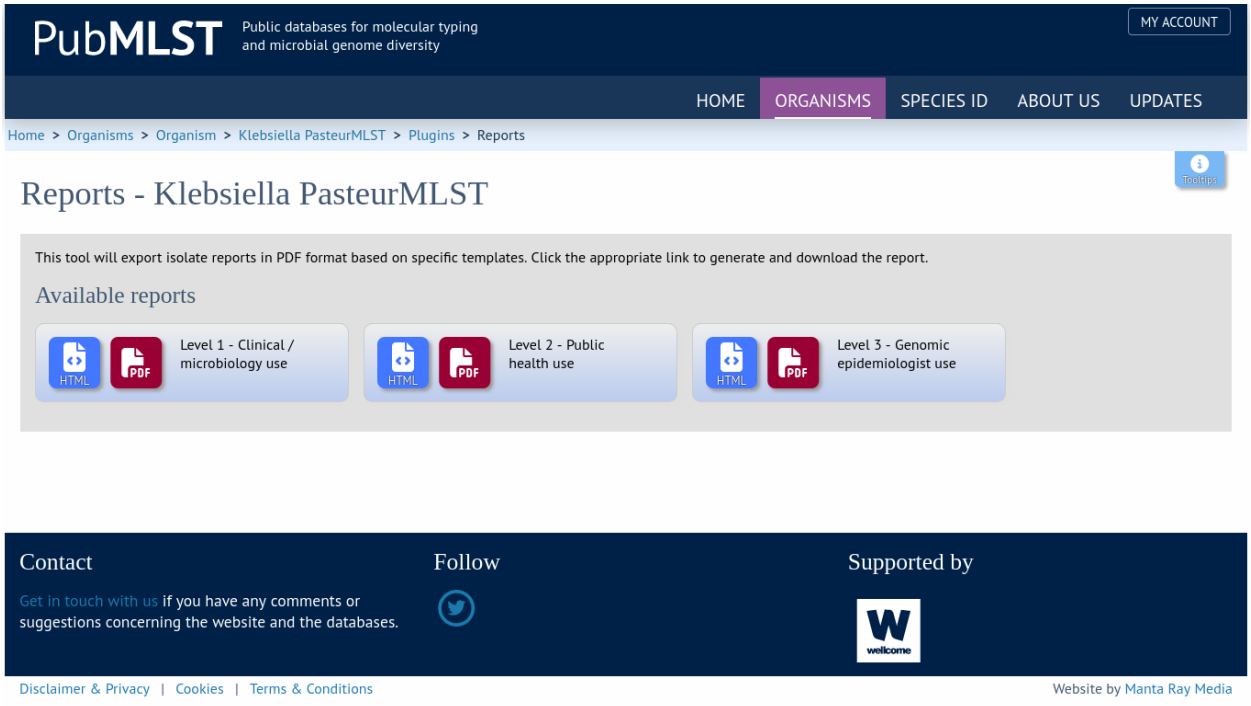
V Mixão, M Pinto, D Sobral, A Di Pasquale, J Gomes, V Borges (2023) ReporTree: a surveillance-oriented tool to strengthen the linkage between pathogen genetic clusters and epidemiological data Genome Med 15:43.

It also makes use of the following tools which should also be cited:

- cgmlst-dists: https://github.com/tseemann/cgmlst-dists for original code and https://github.com/genpat-it/cgmlst-dists for improvements regarding memory efficiency.
- If you requested a GrapeTree analysis:
  - Z Zhou, NF Alikhan, MJ Sergeant, N Luhmann, C Vaz, AP Francisco, JA Carrico, M Achtman (2018) GrapeTree: Visualization of core genomic relationships among 100,000 bacterial pathogens. Genome Res 28:1395-1404.
- If you requested 'stability regions':
  - JA Carrico, C Silva-Costa, J Melo-Cristino, FR Pinto, H de Lencastre, JS Almeida, M Ramirez (2006) Illustration of a common framework for relating multiple typing methods by application to macrolide-resistant *Streptococcus pyogenes*. J Clin Microbiol 44:2524-32.
  - DOR Barker, JA Carriço, P Kruczkiewicz, F Palma, M Rossi, EN Taboada (2018) Rapid identification of stable clusters in bacterial populations using the Adjusted Wallace Coefficient. BioRxiv DOI: https://doi.org/10.1101/299347.

14.17 Reports

The Reports plugin is a tool for generating formatted reports for specific audiences, e.g. clinical/microbiology use, public health, or genomic specialists. It uses template files that can be modified as required.

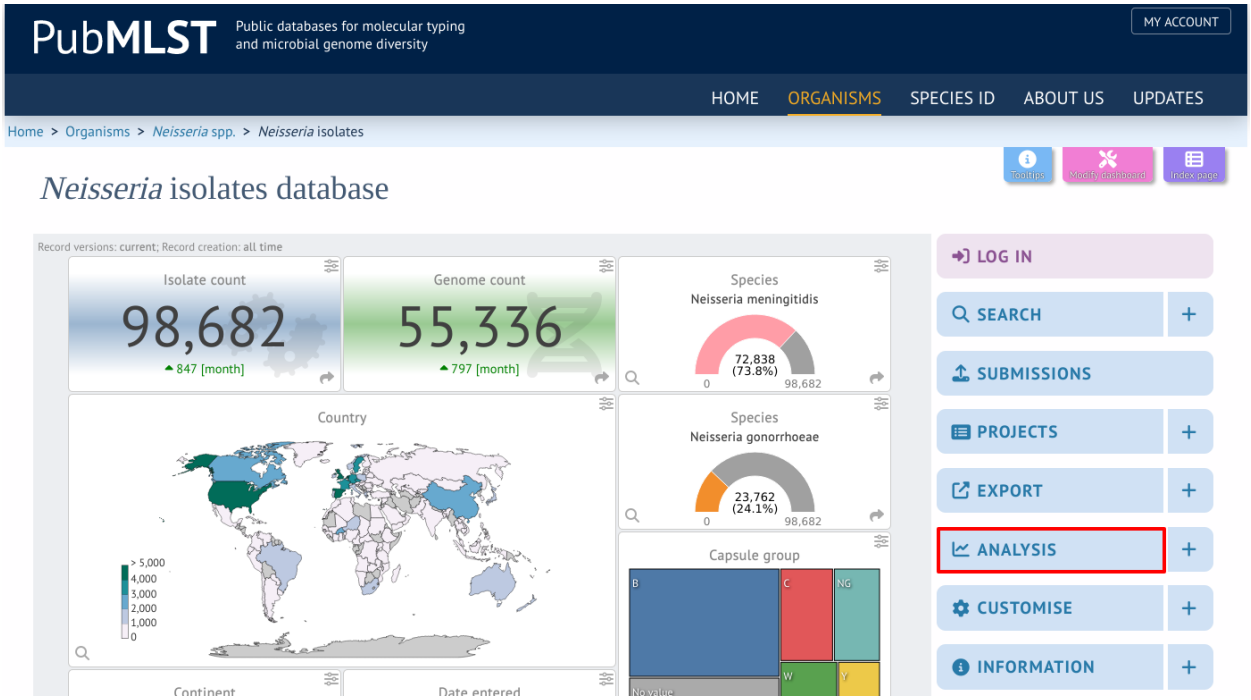


See https://github.com/kjolley/Klebsiella_reports for examples of how to configure report templates.

14.18 Sequence bin breakdown

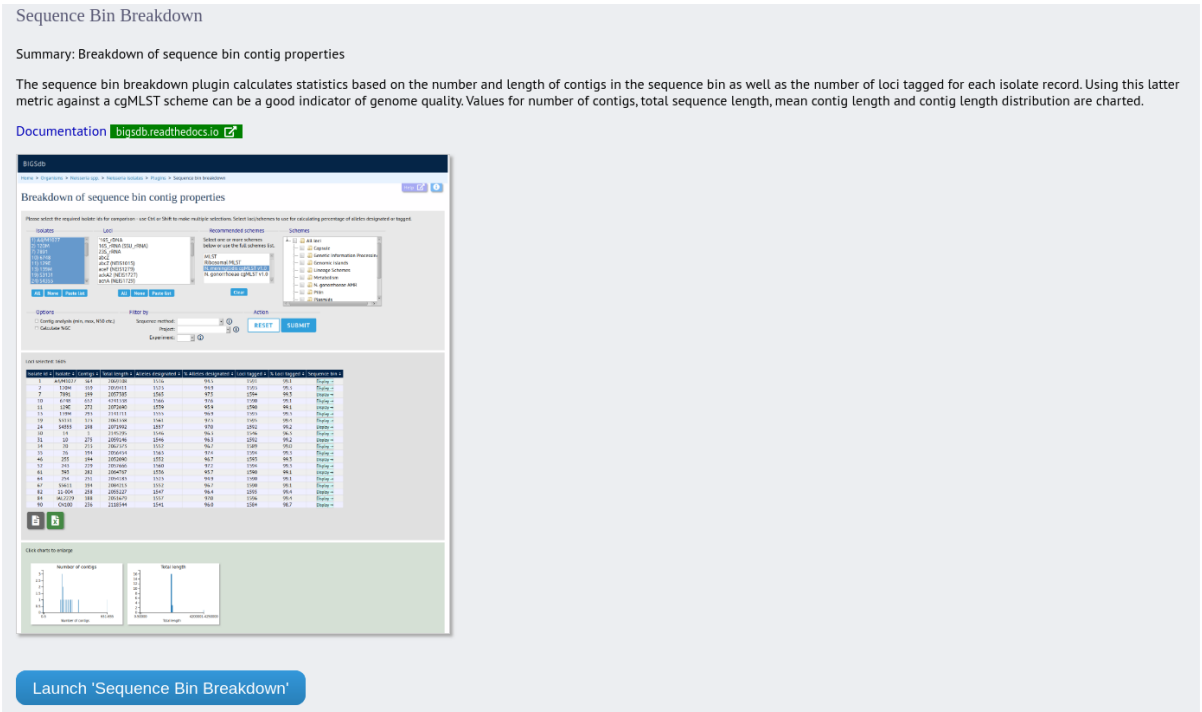
The sequence bin breakdown plugin calculates statistics based on the number and length of contigs in the sequence bin as well as the number of loci tagged for an isolate record.

The function can be accessed by selecting the 'Analysis' section on the main contents page.

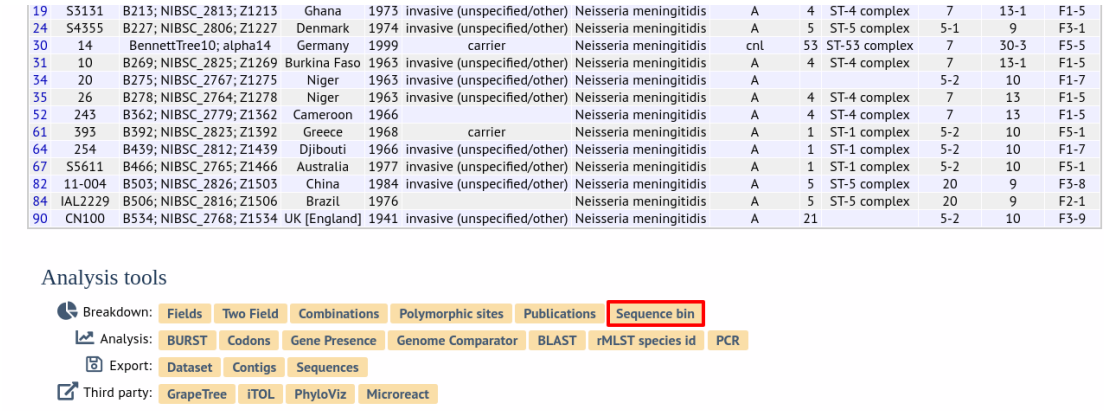


Jump to the 'Breakdown' category, follow the link to Sequence Bin Breakdown, then click 'Launch Sequence Bin Breakdown'.

The function can be selected by clicking the 'Sequence bin' link on the Breakdown section of the main contents page.



Alternatively, it can be accessed following a query by clicking the 'Sequence bin' button in the Breakdown list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.

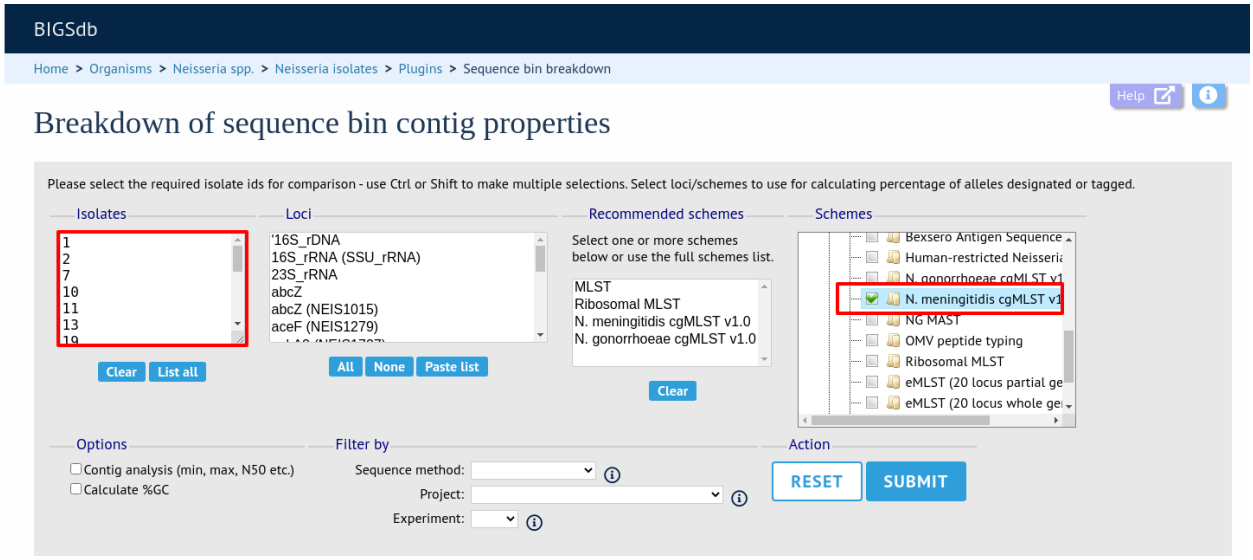


19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
30	14	BennettTree10; alpha14	Germany	1999	carrier	Neisseria meningitidis	cnl	53	ST-53 complex	7	30-3	F5-5
31	10	B269; NIBSC_2825; Z1269	Burkina Faso	1963	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
34	20	B275; NIBSC_2767; Z1275	Niger	1963	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-7
35	26	B278; NIBSC_2764; Z1278	Niger	1963	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13	F1-5
52	243	B362; NIBSC_2779; Z1362	Cameroon	1966		Neisseria meningitidis	A	4	ST-4 complex	7	13	F1-5
61	393	B392; NIBSC_2823; Z1392	Greece	1968	carrier	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
64	254	B439; NIBSC_2812; Z1439	Djibouti	1966	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F1-7
67	S5611	B466; NIBSC_2765; Z1466	Australia	1977	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
82	11-004	B503; NIBSC_2826; Z1503	China	1984	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-8
84	IAL2229	B506; NIBSC_2816; Z1506	Brazil	1976		Neisseria meningitidis	A	5	ST-5 complex	20	9	F2-1
90	CN100	B534; NIBSC_2768; Z1534	UK [England]	1941	invasive (unspecified/other)	Neisseria meningitidis	A	21		5-2	10	F3-9

Select the isolate records to analyse - these will be pre-selected if you accessed the plugin following a query. You can also select loci and/or schemes which will be used to calculate the totals and percentages of loci designated and tagged. This may be useful as a guide to assembly quality if you use a scheme of core loci where a good assembly would be expected to include all member loci. To determine the total of all loci designated or tagged, click 'All loci' in the scheme tree.

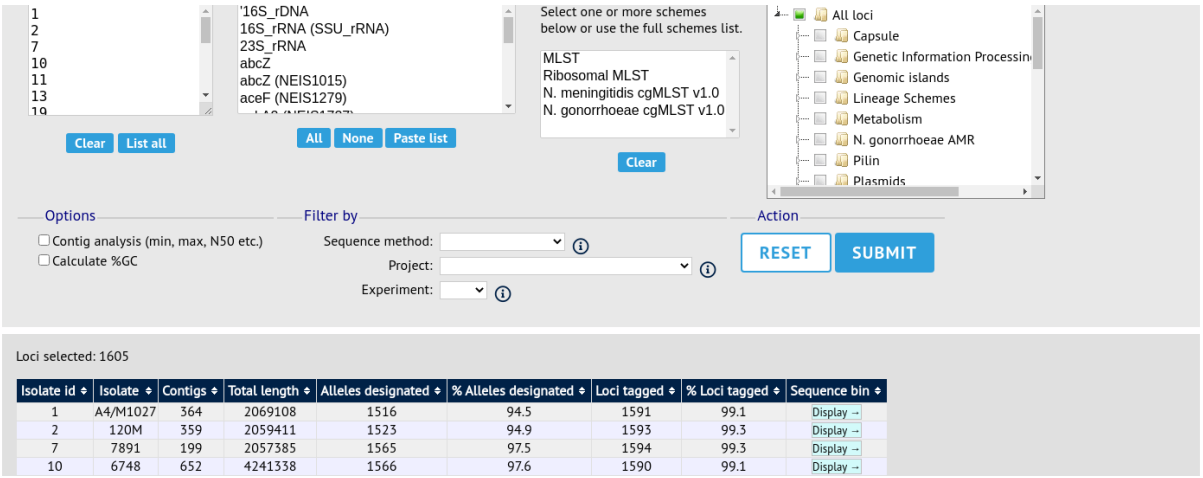
There is also an option to determine the mean G+C content and various assembly stats of the sequence bin of each isolate. Note that selecting these will make the analysis run much slower since each contig needs to be examined.

Click submit.

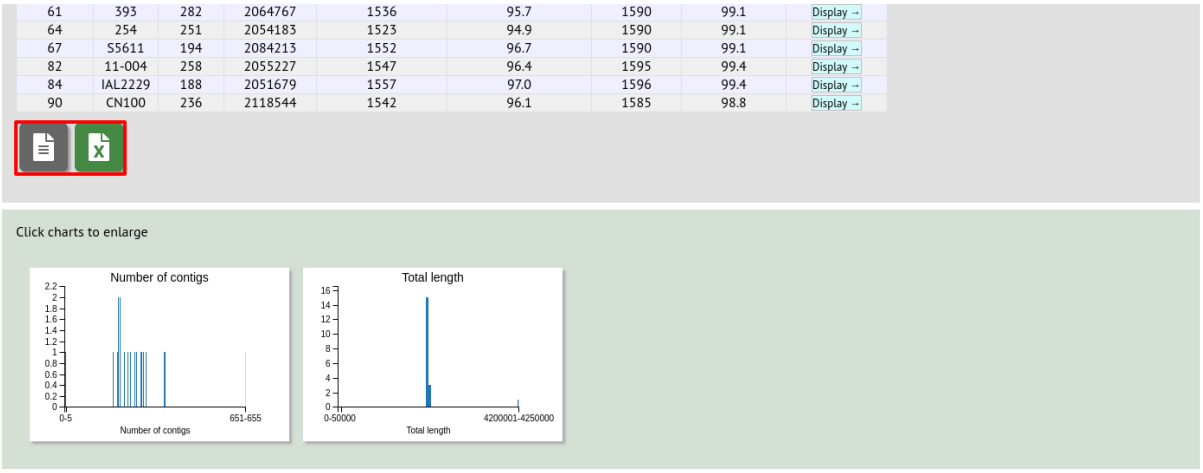


If there are fewer than 100 isolates selected, the table will be generated immediately. Otherwise it will be submitted to the job queue.

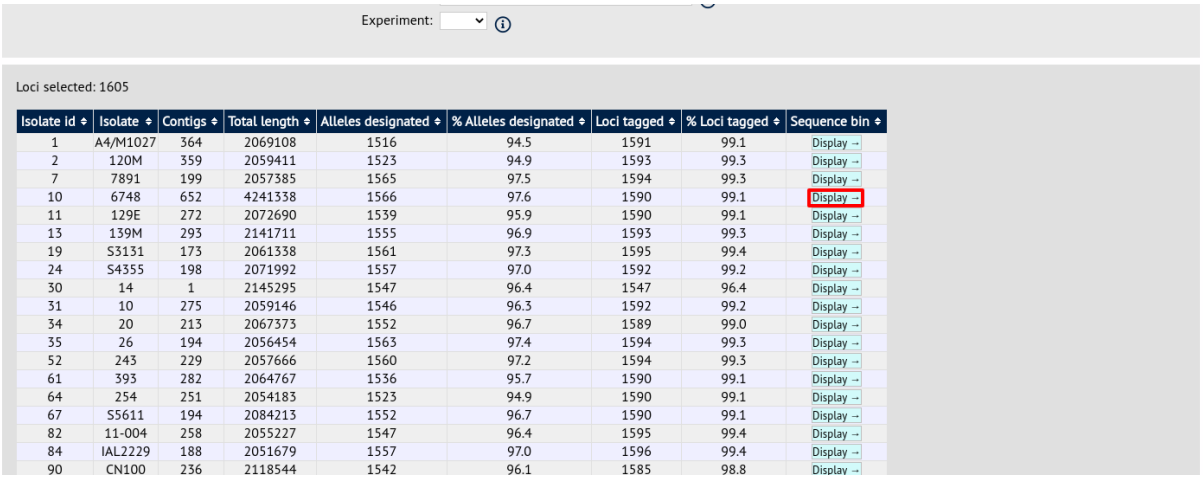
A table of sequence bin stats will be generated.



You can choose to export the data in tab-delimited text or Excel formats by clicking the appropriate link at the bottom of the table.



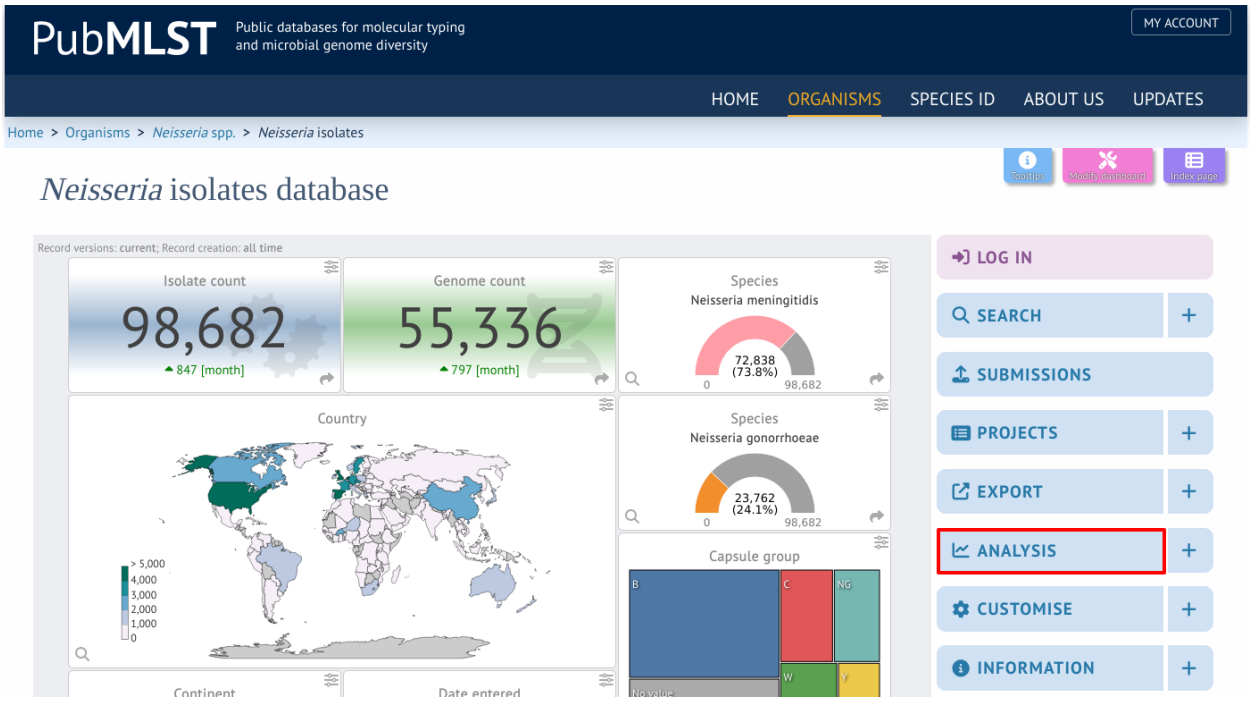
Sequence bin records can also be accessed by clicking the 'Display' button for each row of the table.



14.19 Two field breakdown

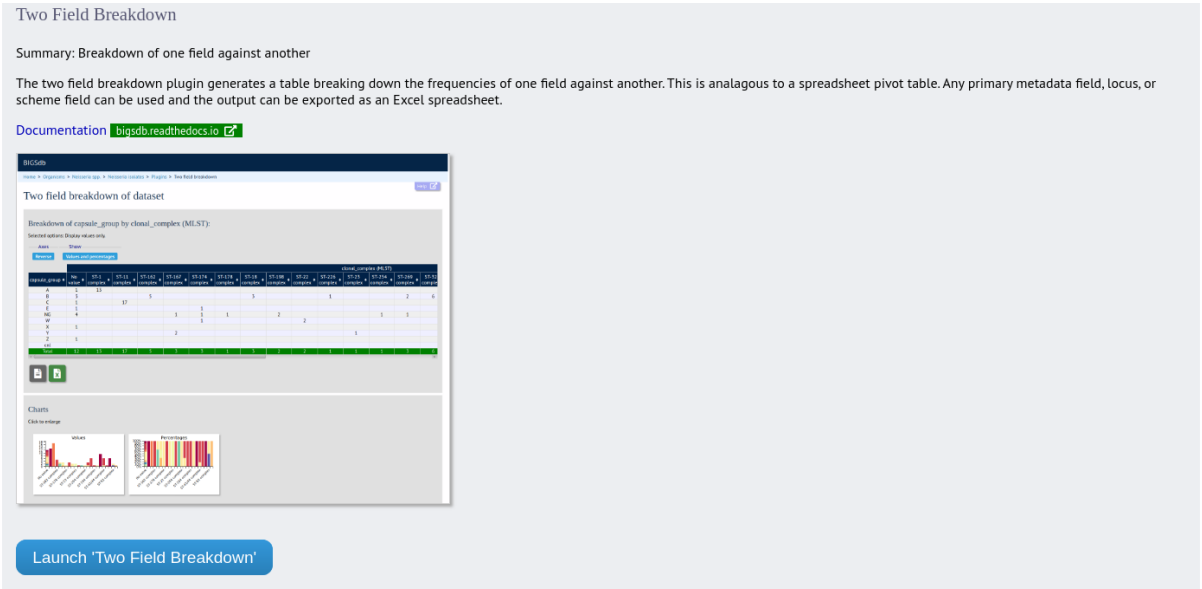
The two field breakdown plugin displays a table breaking down one field against another, e.g. breakdown of serogroup by year.

The analysis can be accessed by selecting the 'Analysis' section on the main contents page.



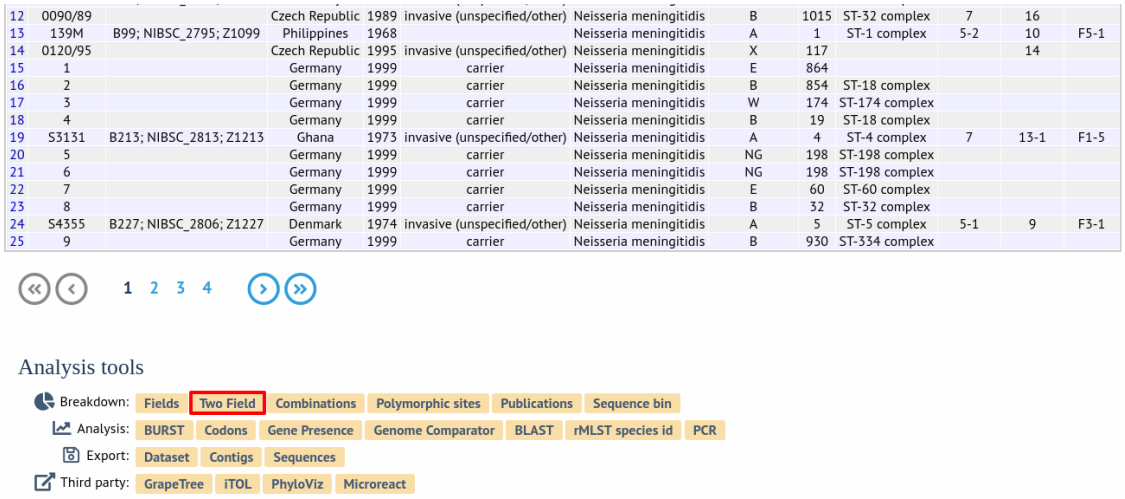
Jump to the 'Breakdown' category, follow the link to Two Field Breakdown, then click 'Launch Two Field Breakdown'.

The analysis can be selected for the whole database by clicking the 'Two field breakdown' link on the main contents page.



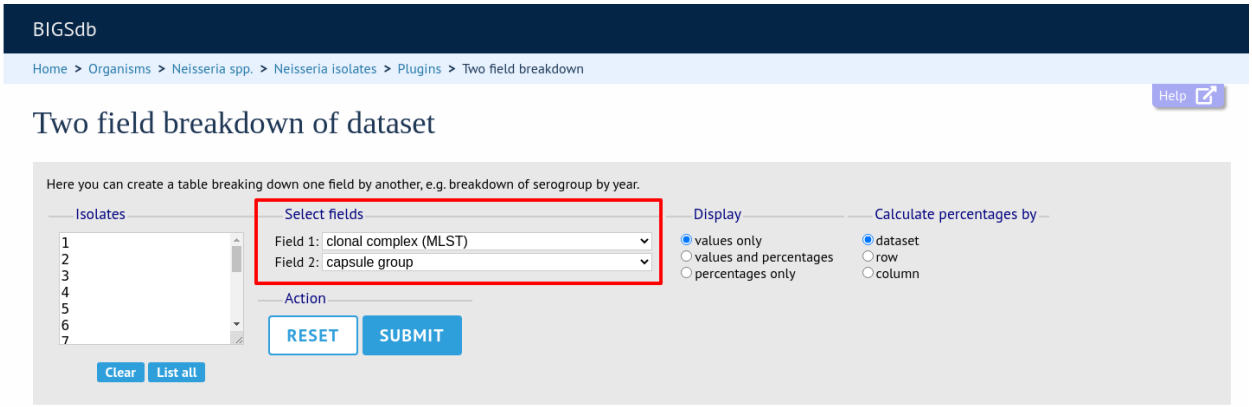
Alternatively, a two field breakdown can be displayed of the dataset returned from a query by clicking the 'Two field' button in the Breakdown list at the bottom of the results table. Please note that the list of functions here may vary

depending on the setup of the database.



12	0090/89		Czech Republic	1989	invasive (unspecified/other)	Neisseria meningitidis	B	1015	ST-32 complex	7	16	
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
14	0120/95		Czech Republic	1995	invasive (unspecified/other)	Neisseria meningitidis	X	117			14	
15	1		Germany	1999	carrier	Neisseria meningitidis	E	864				
16	2		Germany	1999	carrier	Neisseria meningitidis	B	854	ST-18 complex			
17	3		Germany	1999	carrier	Neisseria meningitidis	W	174	ST-174 complex			
18	4		Germany	1999	carrier	Neisseria meningitidis	B	19	ST-18 complex			
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
20	5		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
21	6		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
22	7		Germany	1999	carrier	Neisseria meningitidis	E	60	ST-60 complex			
23	8		Germany	1999	carrier	Neisseria meningitidis	B	32	ST-32 complex			
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
25	9		Germany	1999	carrier	Neisseria meningitidis	B	930	ST-334 complex			

Select the two fields you wish to breakdown and how you would like the values displayed (percentage/absolute values and totaling options).



Click submit. The breakdown will be displayed as a table. Bar charts will also be displayed provided the number of returned values for both fields are fewer than 30.

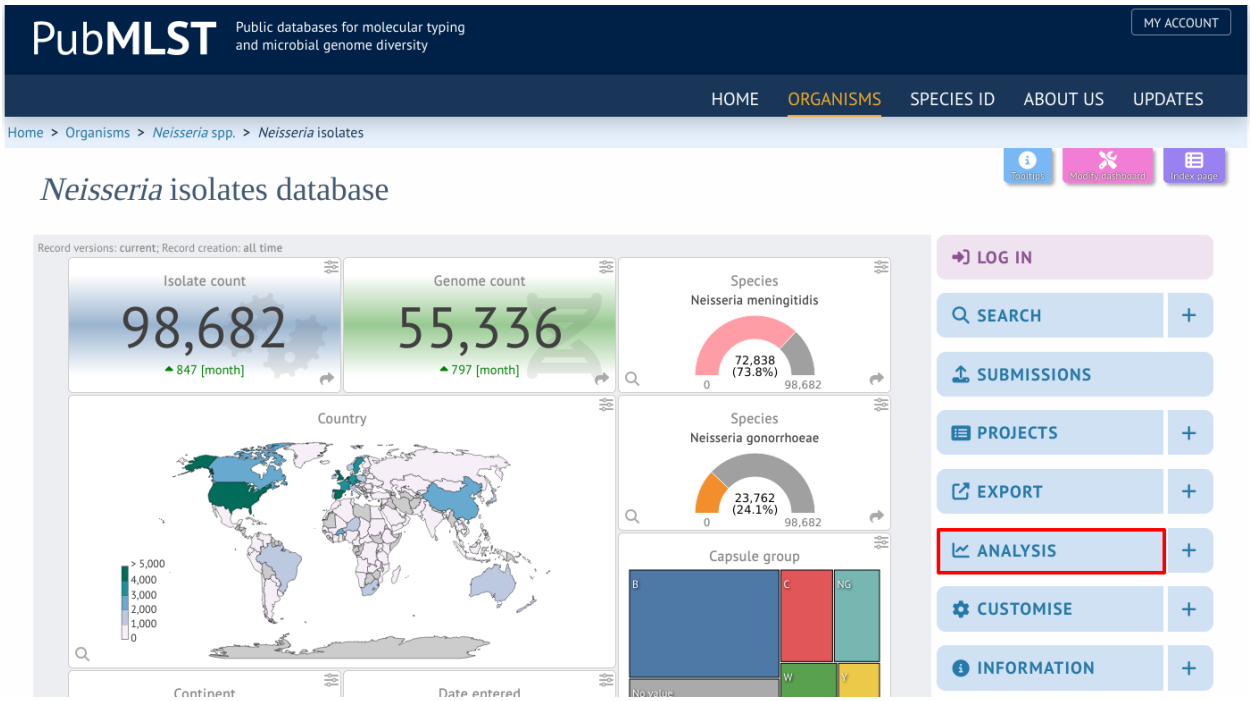
The table values can be exported in a format suitable for copying in to a spreadsheet by clicking 'Download as tab-delimited text' underneath the table.

Note: The job will be submitted to the offline job queue if the query returns 10,000 or more isolates. In this case, the buttons to reverse the axes or to change whether values or percentages are shown will not be available.

14.20 Unique combinations

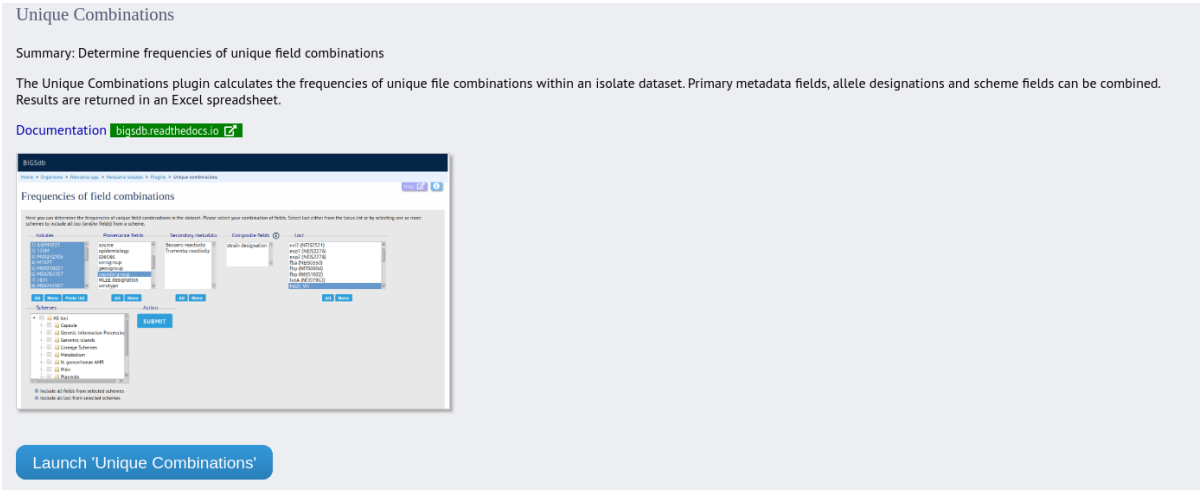
The Unique Combinations plugin calculates the frequencies of unique field combinations within an isolate dataset. Provenance fields, composite fields, allele designations and scheme fields can be combined.

The function can be accessed by selecting the 'Analysis' section on the main contents page.

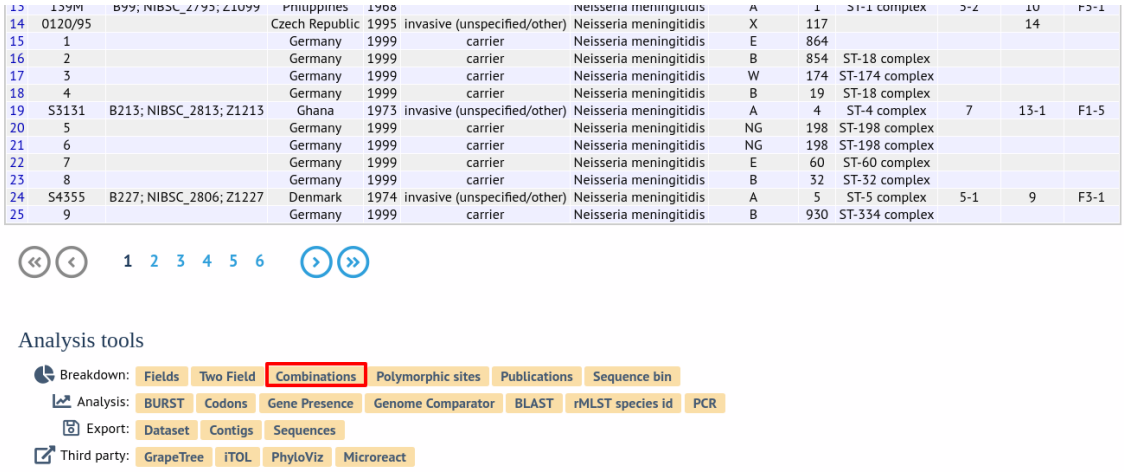


Jump to the 'Breakdown' category, follow the link to Unique Combinations, then click 'Launch Unique Combinations'.

The function can be selected by clicking the 'Unique combinations' link in the Breakdown section of the main contents page. This will run the analysis on the entire database.



Alternatively, it can be accessed following a query by clicking the 'Combinations' button in the Breakdown list at the bottom of the results table. This will run the analysis on the dataset returned from the query. Please note that the list of functions here may vary depending on the setup of the database.



13	159M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
14	0120/95		Czech Republic	1995	invasive (unspecified/other)	Neisseria meningitidis	X	117			14	
15	1		Germany	1999	carrier	Neisseria meningitidis	E	864				
16	2		Germany	1999	carrier	Neisseria meningitidis	B	854	ST-18 complex			
17	3		Germany	1999	carrier	Neisseria meningitidis	W	174	ST-174 complex			
18	4		Germany	1999	carrier	Neisseria meningitidis	B	19	ST-18 complex			
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
20	5		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
21	6		Germany	1999	carrier	Neisseria meningitidis	NG	198	ST-198 complex			
22	7		Germany	1999	carrier	Neisseria meningitidis	E	60	ST-60 complex			
23	8		Germany	1999	carrier	Neisseria meningitidis	B	32	ST-32 complex			
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
25	9		Germany	1999	carrier	Neisseria meningitidis	B	930	ST-334 complex			

1 2 3 4 5 6

Analysis tools

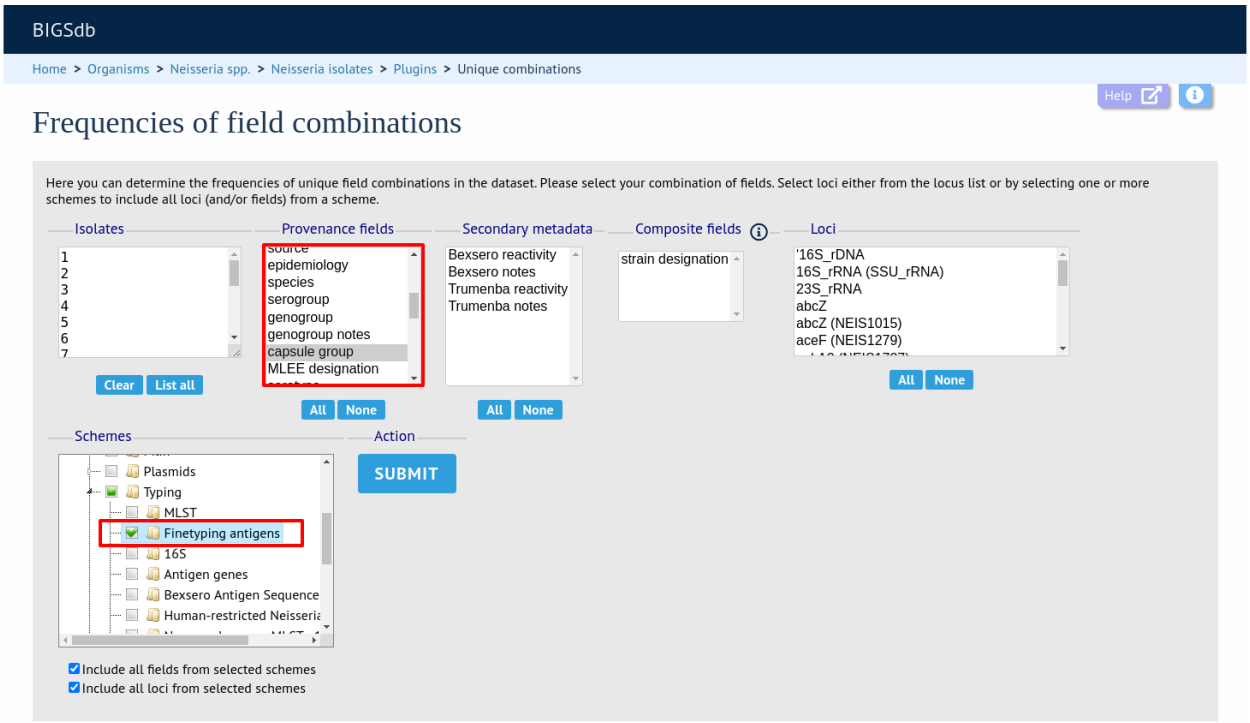
Breakdown: Fields | Two Field | Combinations | Polymorphic sites | Publications | Sequence bin

Analysis: BURST | Codons | Gene Presence | Genome Comparator | BLAST | rMLST species id | PCR

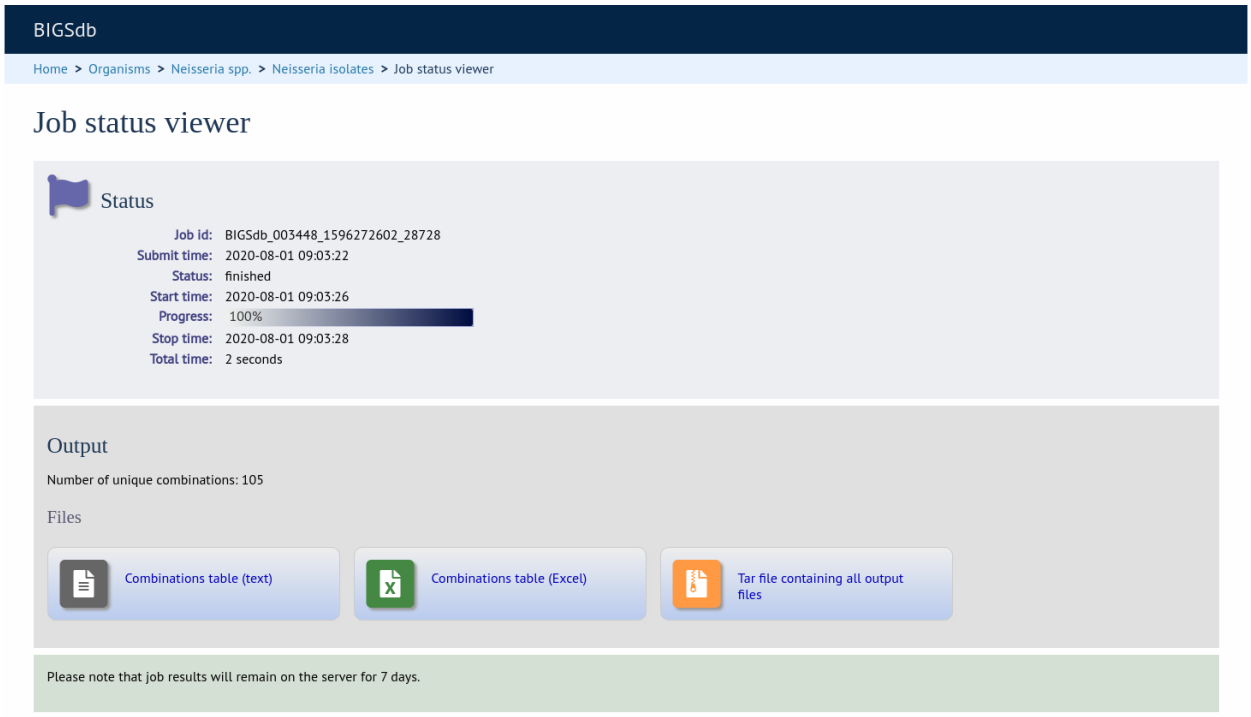
Export: Dataset | Contigs | Sequences

Third party: GrapeTree | iTOL | PhyloViz | Microreact

Select the combination of fields to analyse, e.g. serogroup and finetyping antigens.



Click submit. The job will be submitted to the job queue. Once analysis has completed, you will be able to download the results in tab-delimited text or Excel formats.


BIGSdb
Home > Organisms > Neisseria spp. > Neisseria isolates > Job status viewer
Job status viewer
Status
Job id: BIGSdb_003448_1596272602_28728
Submit time: 2020-08-01 09:03:22
Status: finished
Start time: 2020-08-01 09:03:26
Progress: 100%
Stop time: 2020-08-01 09:03:28
Total time: 2 seconds
Output
Number of unique combinations: 105
Files
Combinations table (text)
Combinations table (Excel)
Tar file containing all output files
Please note that job results will remain on the server for 7 days.

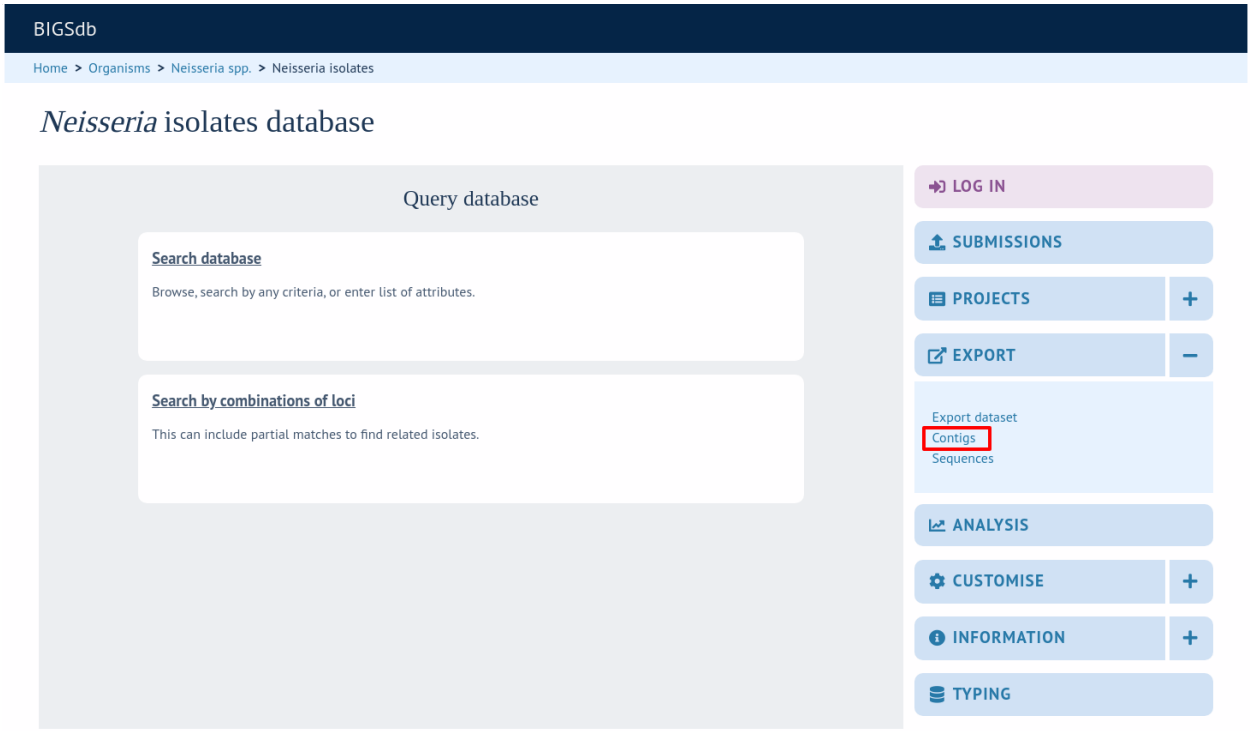
CHAPTER

FIFTEEN

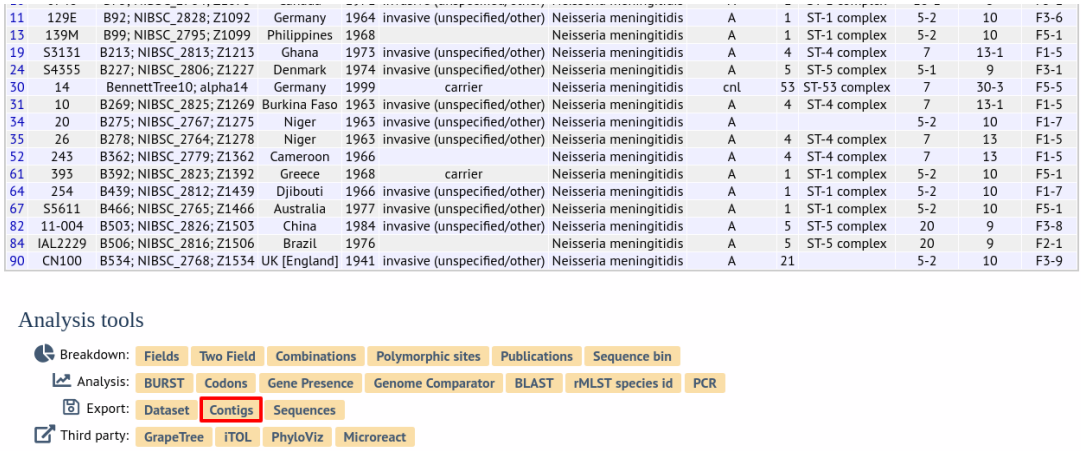
DATA EXPORT PLUGINS

15.1 Contig export

The contig export plugin can be accessed by expanding the 'Export' section and clicking the 'Contigs' link in the contents page of isolate databases.

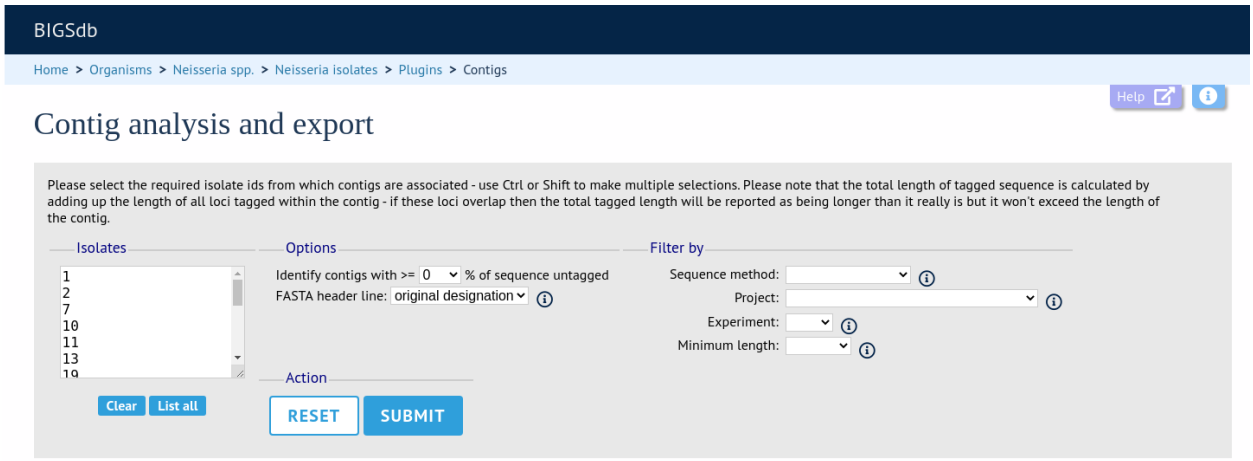


Alternatively, it can be accessed following a query by clicking the 'Contigs' button in the Export section at the bottom of the results table.



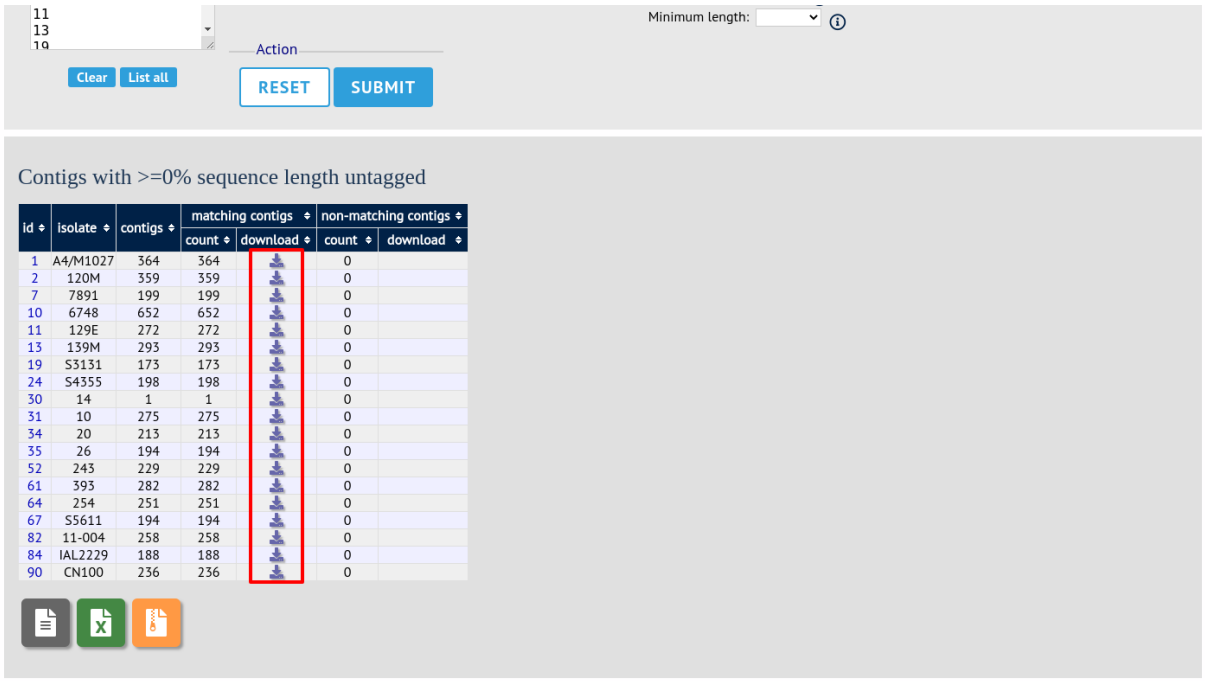
11	129E	B92; NIBSC_2828; Z1092	Germany	1964	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F3-6
13	139M	B99; NIBSC_2795; Z1099	Philippines	1968		Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
19	S3131	B213; NIBSC_2813; Z1213	Ghana	1973	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
24	S4355	B227; NIBSC_2806; Z1227	Denmark	1974	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	5-1	9	F3-1
30	14	BennettTree10; alpha14	Germany	1999	carrier	Neisseria meningitidis	cnl	53	ST-53 complex	7	30-3	F5-5
31	10	B269; NIBSC_2825; Z1269	Burkina Faso	1963	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13-1	F1-5
34	20	B275; NIBSC_2767; Z1275	Niger	1963	invasive (unspecified/other)	Neisseria meningitidis	A			5-2	10	F1-7
35	26	B278; NIBSC_2764; Z1278	Niger	1963	invasive (unspecified/other)	Neisseria meningitidis	A	4	ST-4 complex	7	13	F1-5
52	243	B362; NIBSC_2779; Z1362	Cameroon	1966		Neisseria meningitidis	A	4	ST-4 complex	7	13	F1-5
61	393	B392; NIBSC_2823; Z1392	Greece	1968	carrier	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
64	254	B439; NIBSC_2812; Z1439	Djibouti	1966	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F1-7
67	S5611	B466; NIBSC_2765; Z1466	Australia	1977	invasive (unspecified/other)	Neisseria meningitidis	A	1	ST-1 complex	5-2	10	F5-1
82	11-004	B503; NIBSC_2826; Z1503	China	1984	invasive (unspecified/other)	Neisseria meningitidis	A	5	ST-5 complex	20	9	F3-8
84	IAL2229	B506; NIBSC_2816; Z1506	Brazil	1976		Neisseria meningitidis	A	5	ST-5 complex	20	9	F2-1
90	CN100	B534; NIBSC_2768; Z1534	UK [England]	1941	invasive (unspecified/other)	Neisseria meningitidis	A	21		5-2	10	F3-9

Select the isolates for which you wish to export contig data for. In databases with a large number of isolates you will need to enter the id numbers rather than select from a list. If the export function was accessed following a query, isolates returned in the query will be pre-selected.

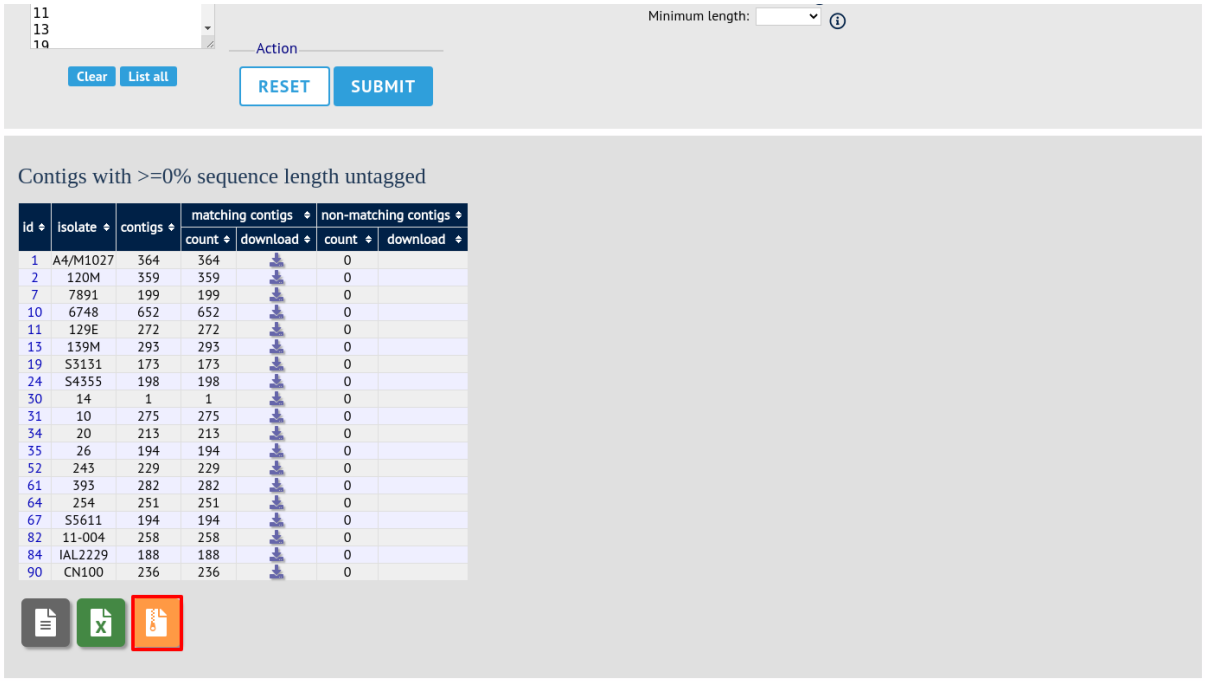


At its simplest, press submit.

A table will be produced with download links. Clicking these will produce the contigs in FASTA format.



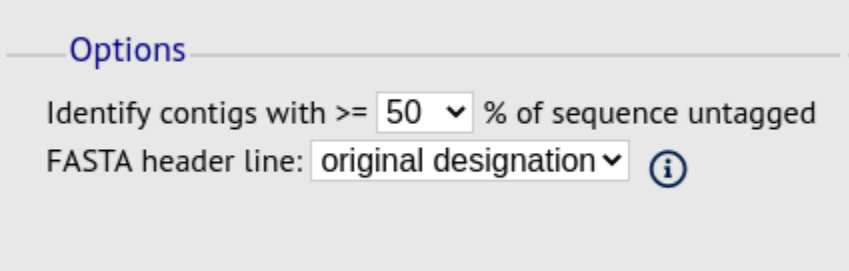
You can also download all the data in a tar file by clicking the 'Batch download' link.



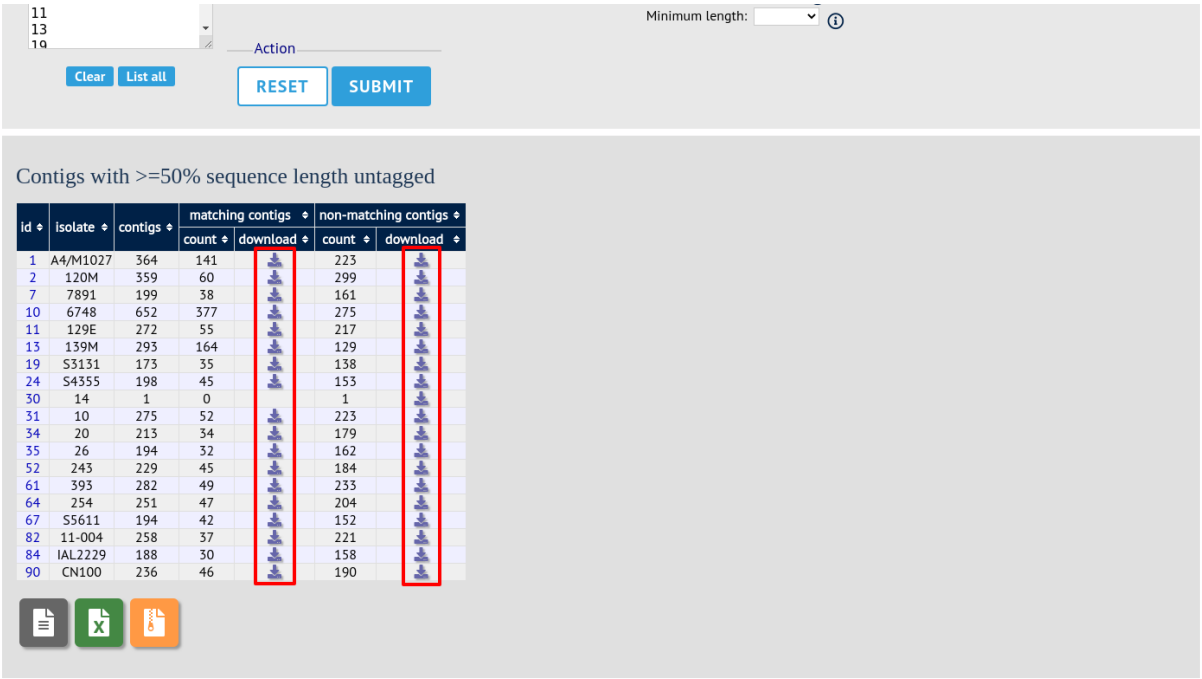
15.1.1 Filtering by tagged status of contigs

You can also export contigs based on the percentage of the sequence that has been tagged. This is useful to find sequences to target for gene discovery.

In order to export contigs where at least half the sequence has been tagged (and also the remaining contigs in a separate file), select '50' in the dropdown box for %untagged.



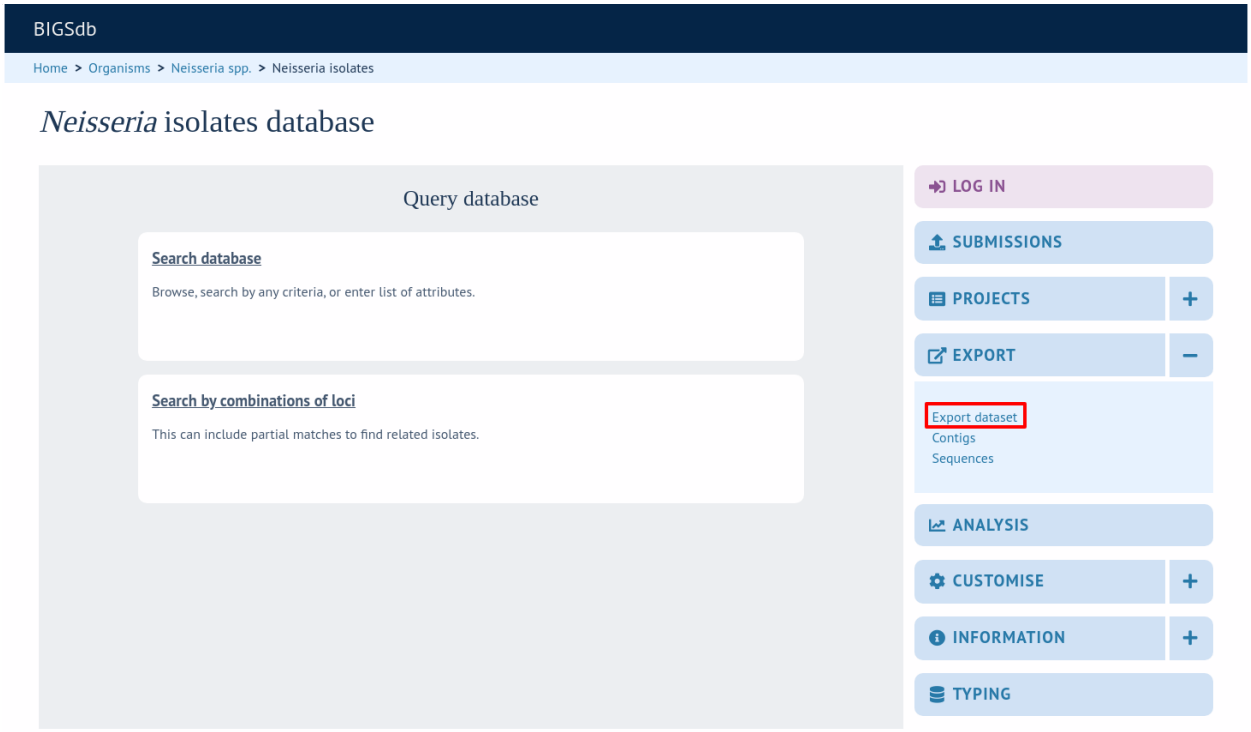
The resulting table has two download links for each isolate, one for contigs matching the condition, and one for contigs that don't match.



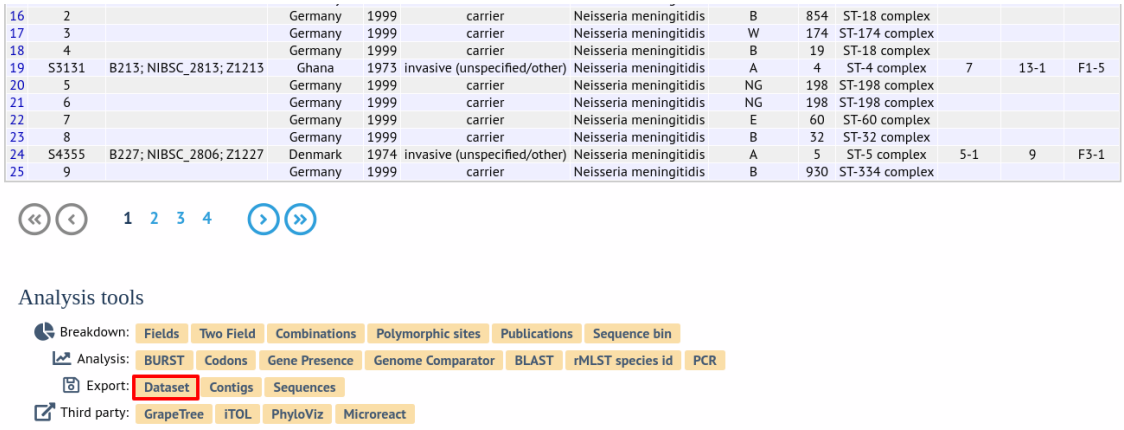
id	isolate	contigs	matching contigs		non-matching contigs	
			count	download	count	download
1	A4/M1027	364	141		223	
2	120M	359	60		299	
7	7891	199	38		161	
10	6748	652	377		275	
11	129E	272	55		217	
13	139M	293	164		129	
19	S3131	173	35		138	
24	S4355	198	45		153	
30	14	1	0		1	
31	10	275	52		223	
34	20	213	34		179	
35	26	194	32		162	
52	243	229	45		184	
61	393	282	49		233	
64	254	251	47		204	
67	S5611	194	42		152	
82	11-004	258	37		221	
84	IAL2229	188	30		158	
90	CN100	236	46		190	

15.2 Isolate record export

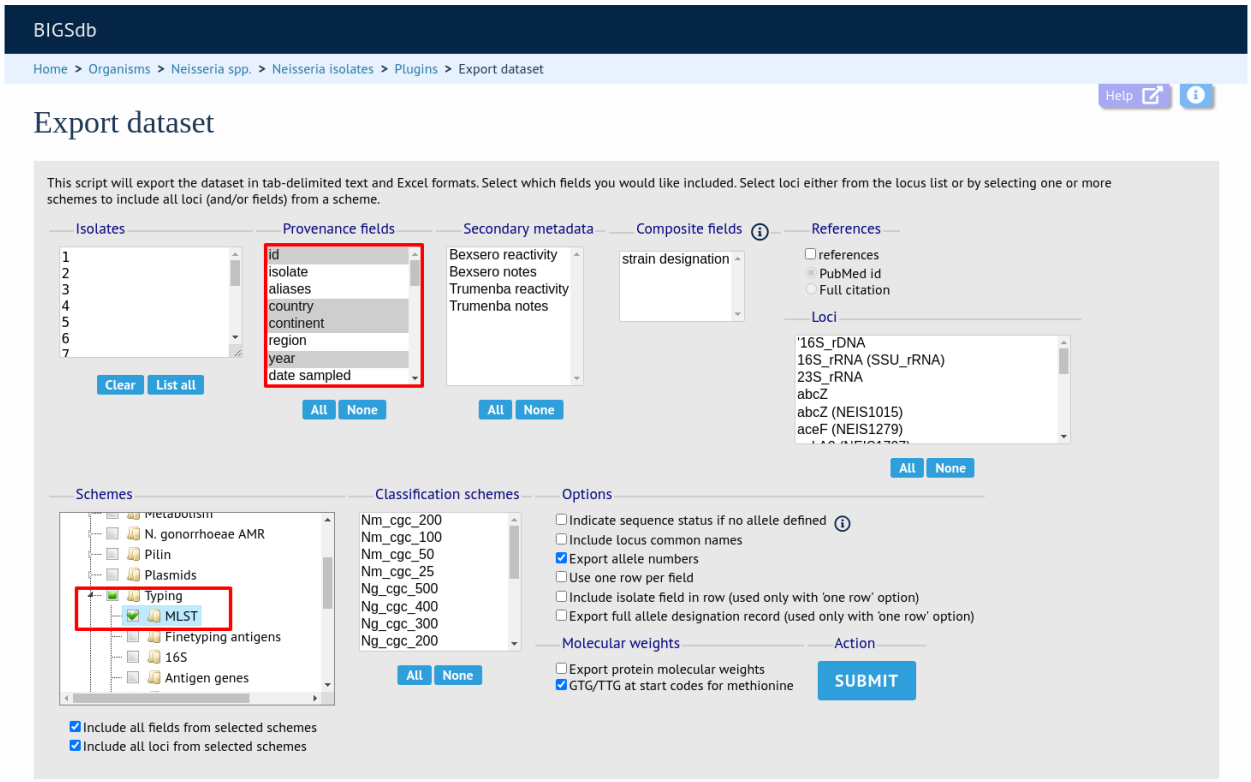
You can export the entire isolate recordset by expanding the Export section on the main contents page and clicking the 'Export dataset' link.



Alternatively, you can export the recordsets of isolates returned from a database query by clicking the 'Dataset' button in the Export list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.

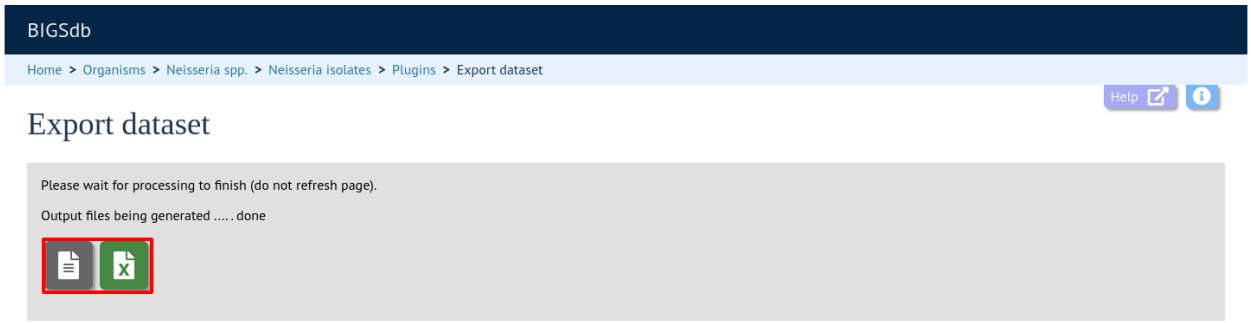


Select the isolate ids (if they have not been pre-selected from your query), isolate fields and schemes to include.



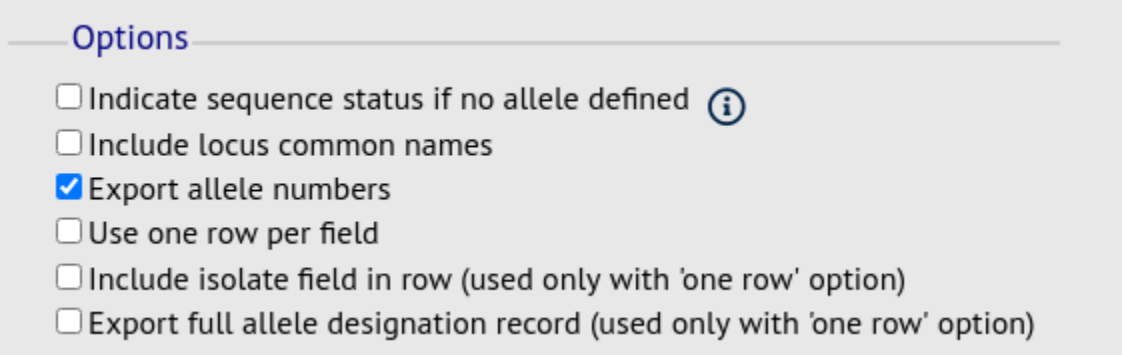
Click Submit.

You can then download the data in tab-delimited text or Excel formats.



Export jobs for larger datasets will be sent to the job queue.

15.2.1 Advanced options

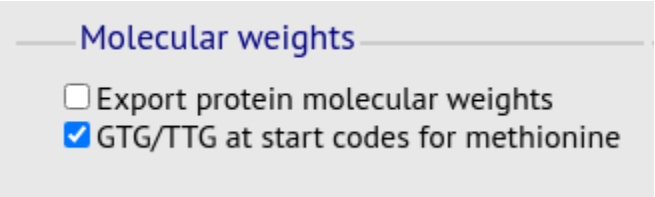


The options fieldset has the following options.

- Include locus common names - any common name for the locus is displayed in parentheses following the primary name.
- Export allele numbers - the allele designation is included for any locus included.
- Use one row per field - this is an alternative output format where instead of each locus and field having a separate column, each field is export on a separate row.
- Include isolate field in row - the name of the isolate is included as a separate column when exporting in 'one row per field' fomrmat.
- Export full allele designation record - export sender, curator and datestamp information as separate rows when exporting allele designation data.

15.2.2 Molecular weight calculation

The plugin can also calculate the predicted molecular weight of the gene product of any allele designated in the dataset.

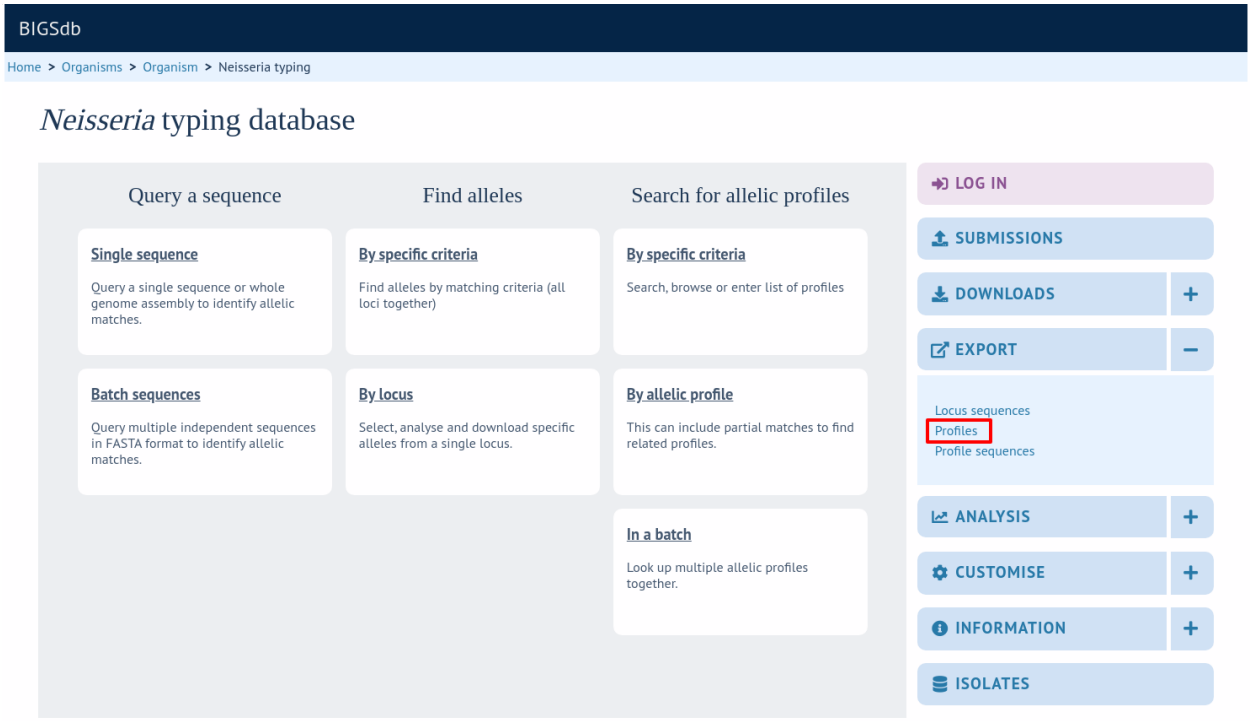


Click the 'Export protein molecular weight' checkbox. Additional columns (or rows depending on the output format) will be created to include the molecular weight data.

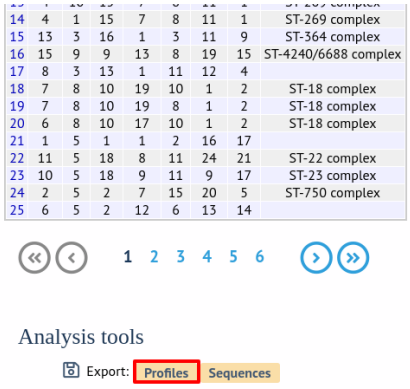
15.3 Profile export

You can export the allelic profiles for any indexed scheme (those containing a primary key field) defined in the sequence definition database.

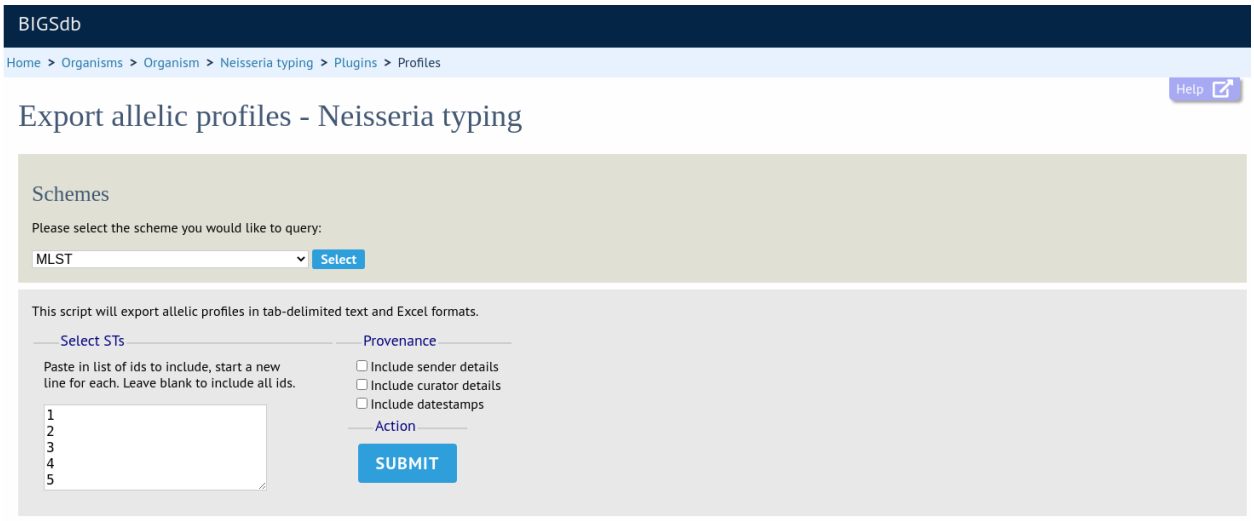
The profile export function can be accessed by expanding the 'Export' section and clicking the 'Profiles' link on the contents page.



Alternatively, you can access this function by clicking the 'Profiles' button in the Export list at the bottom of a results table. Please note that the list of functions here may vary depending on the setup of the database.

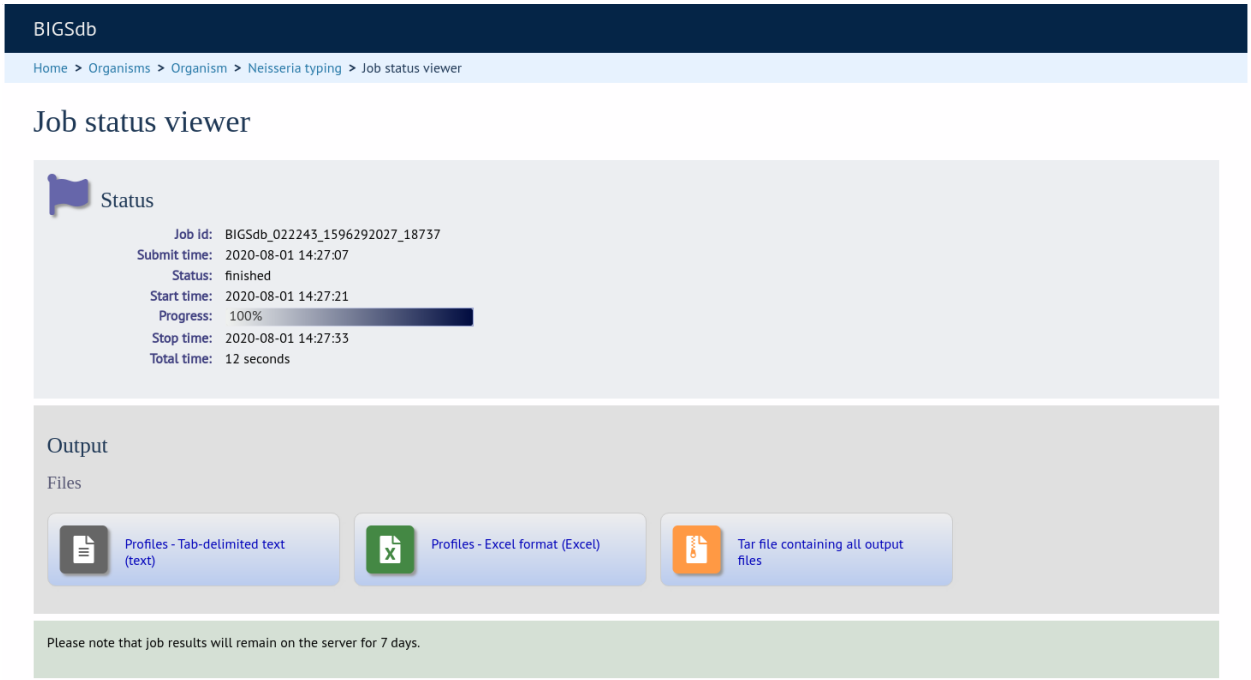


This will take you to a form with a list box in which the identifiers of the profile definitions you wish to include can be entered. Following a query, these values will be pre-entered. If the box is left empty then all profiles will be included. You can optionally include provenance information (sender, curator and datestamps) by selecting the appropriate checkboxes.



Click submit.

The export job will be submitted to the job queue.

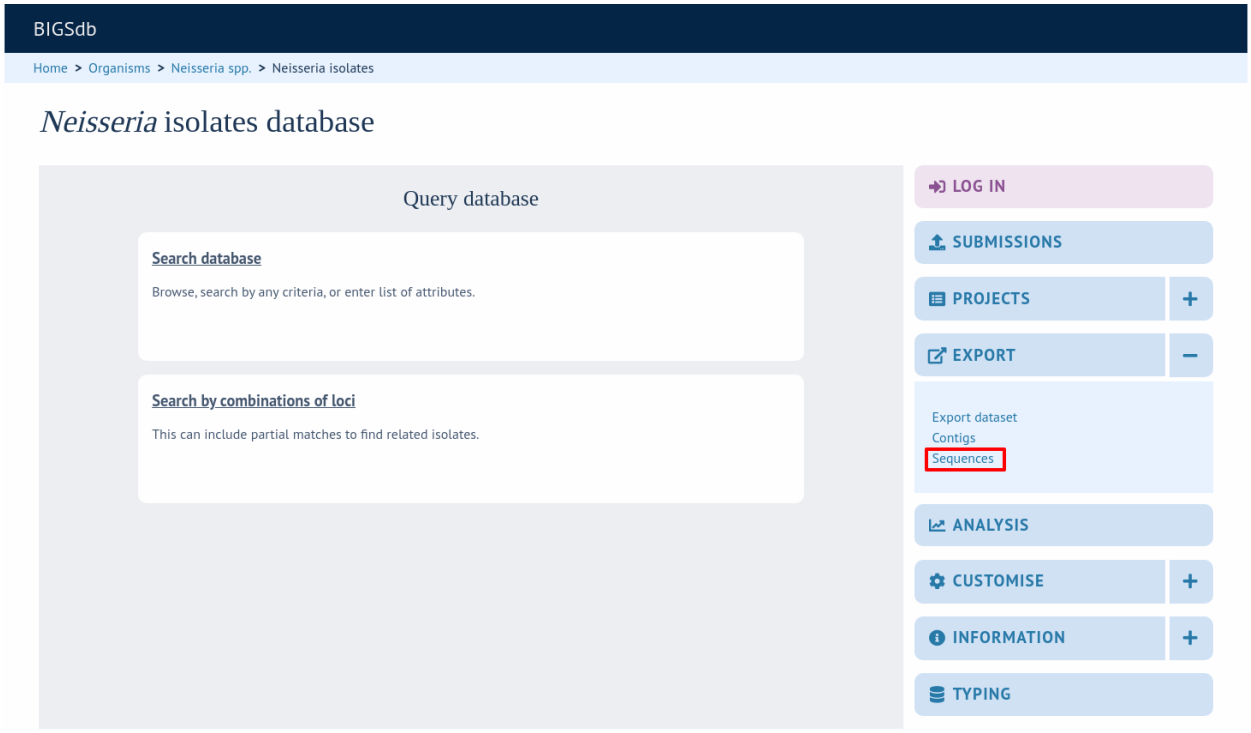


The profiles will be exported in tab-delimited text and Excel formats.

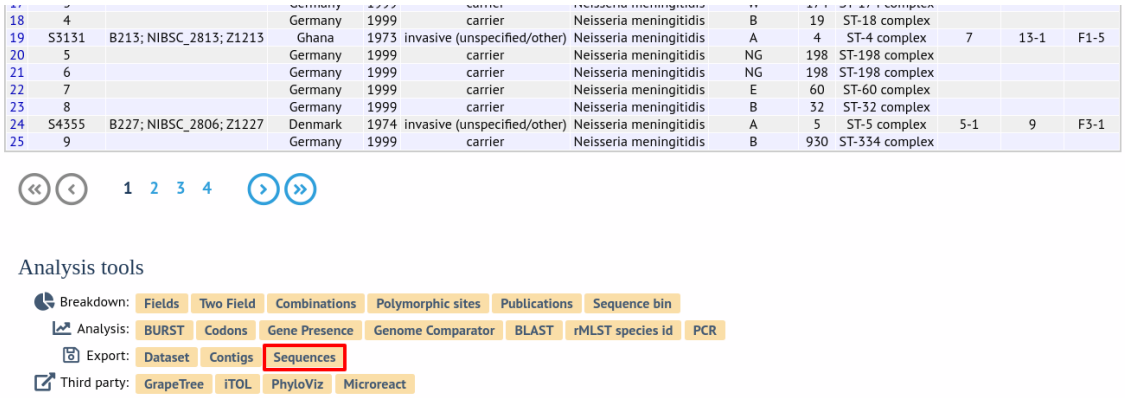
15.4 Sequence export

You can export the sequences for any set of loci designated in isolate records, or belonging to scheme profiles in the sequence definition database.

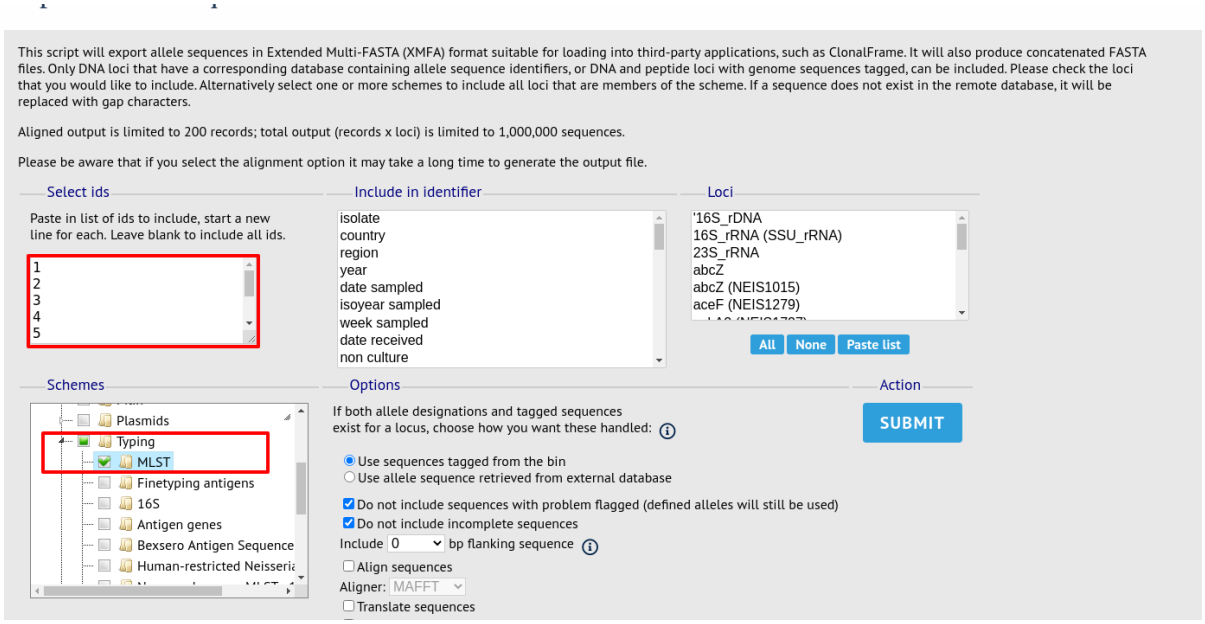
The sequence export function can be accessed by expanding the 'Export' section and clicking the 'Sequences' link on the contents page of isolate databases, or the 'Profile sequences' link on the contents page of sequence definition databases.



Alternatively, you can access this function by clicking the 'Sequences' button in the Export list at the bottom of the results table. Please note that the list of functions here may vary depending on the setup of the database.

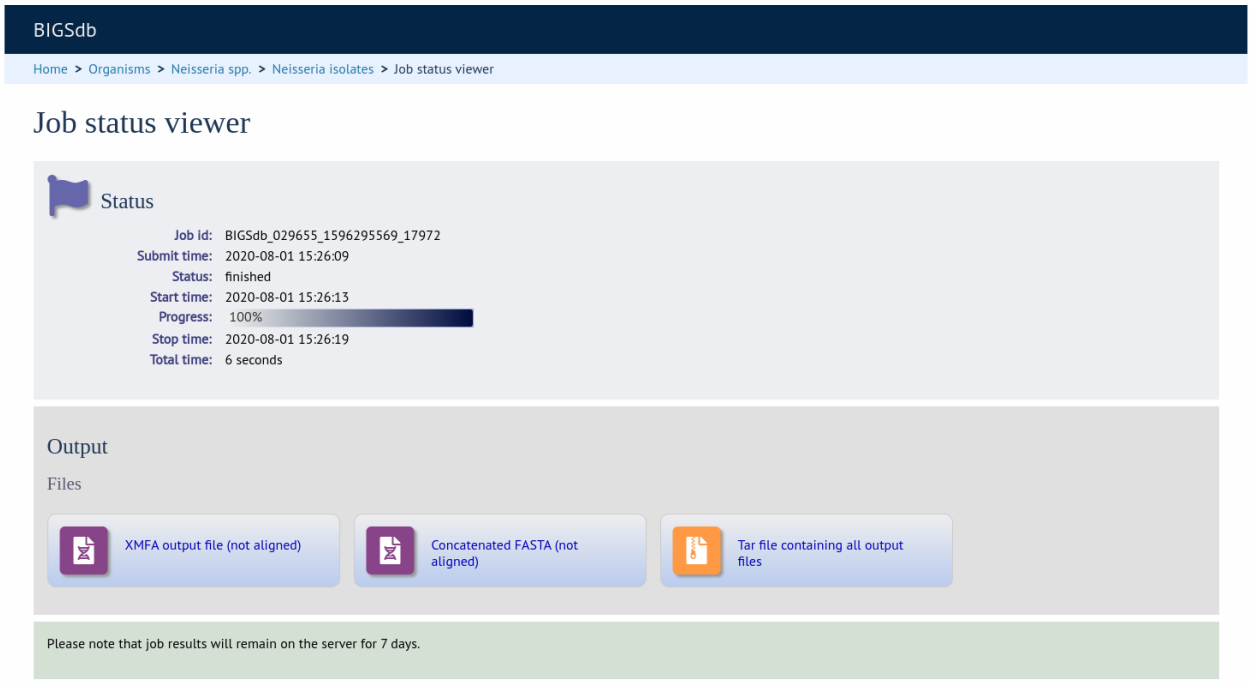


Select the isolate or profile records to analyse - these will be pre-selected if you accessed the plugin following a query. Select the loci to include either directly within the loci list and/or using the schemes tree.



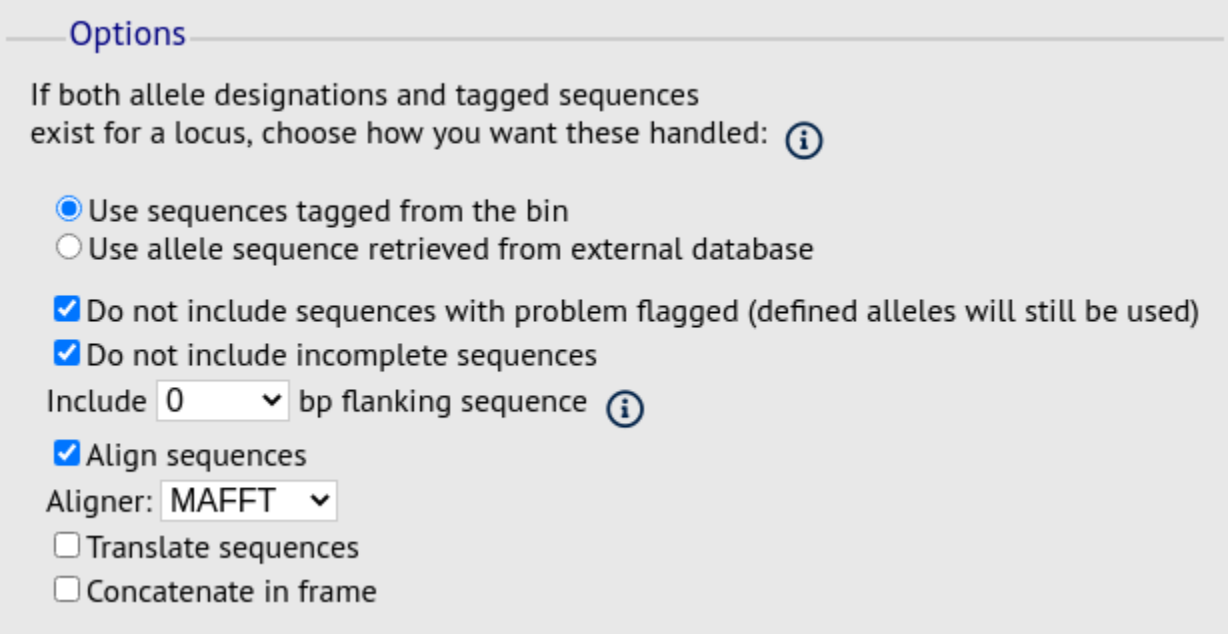
Click submit. The job will be submitted to the job queue.

Sequences will be export in XMFA and FASTA file formats.



15.4.1 Aligning sequences

By default, sequences will be exported unaligned - this is very quick since no processing is required. You can choose to align the sequences by checking the 'Align sequences' checkbox.



You can also choose to use MUSCLE or MAFFT as the aligner. MAFFT is the default choice and is usually much quicker than MUSCLE. Both produce comparable results.

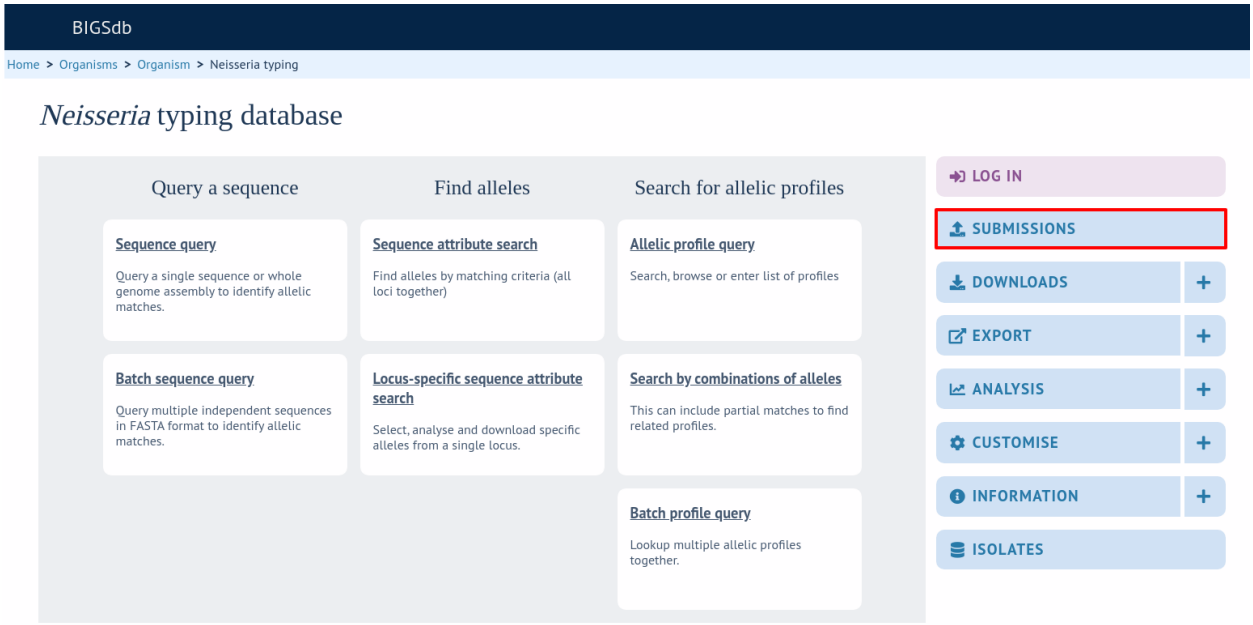
CHAPTER

SIXTEEN

SUBMITTING DATA USING THE SUBMISSION SYSTEM

The automated submission system allows users to submit data (new alleles, profiles, or isolates) to the database curators for assignment and upload to the database. The submission system is enabled on a per-database basis so will not always be available.

If the system is enabled, new submissions can be made by clicking the 'Manage submissions' link on the database front page.



16.1 Registering a user account

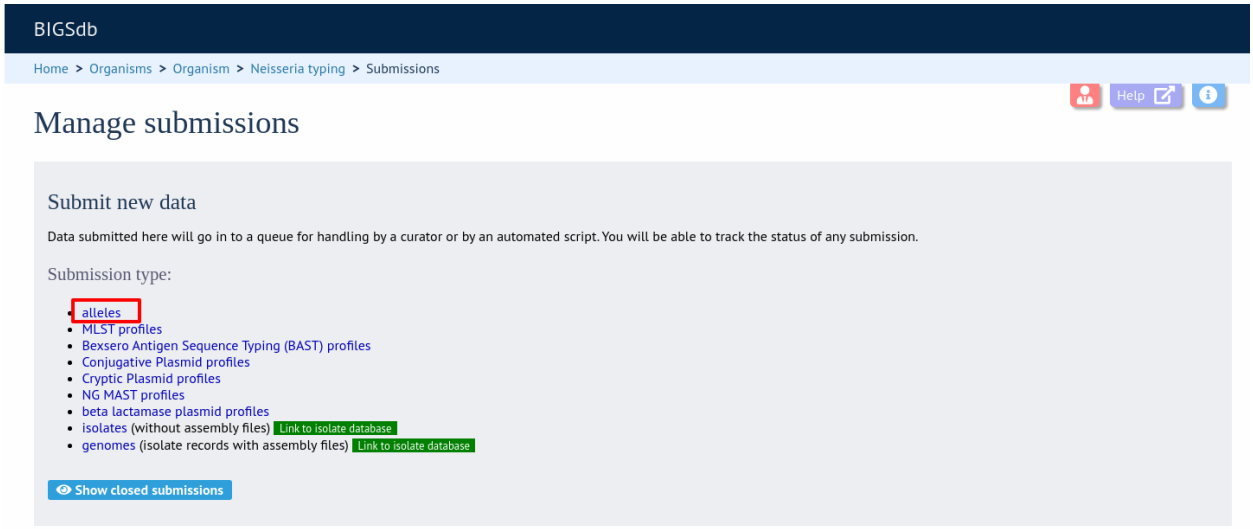
You must have an account for the appropriate database in order to use the submission system. On systems utilizing site-wide databases, such as PubMLST, this can be done automatically via the web. Other sites may require you to contact a curator to set this up.

16.2 Allele submission

New allele data can only be submitted from within the appropriate sequence definition database. Submissions consist of one or more new allele sequences for a single locus. You will need to create separate submissions for each locus - this is because different loci may be handled by different curators.

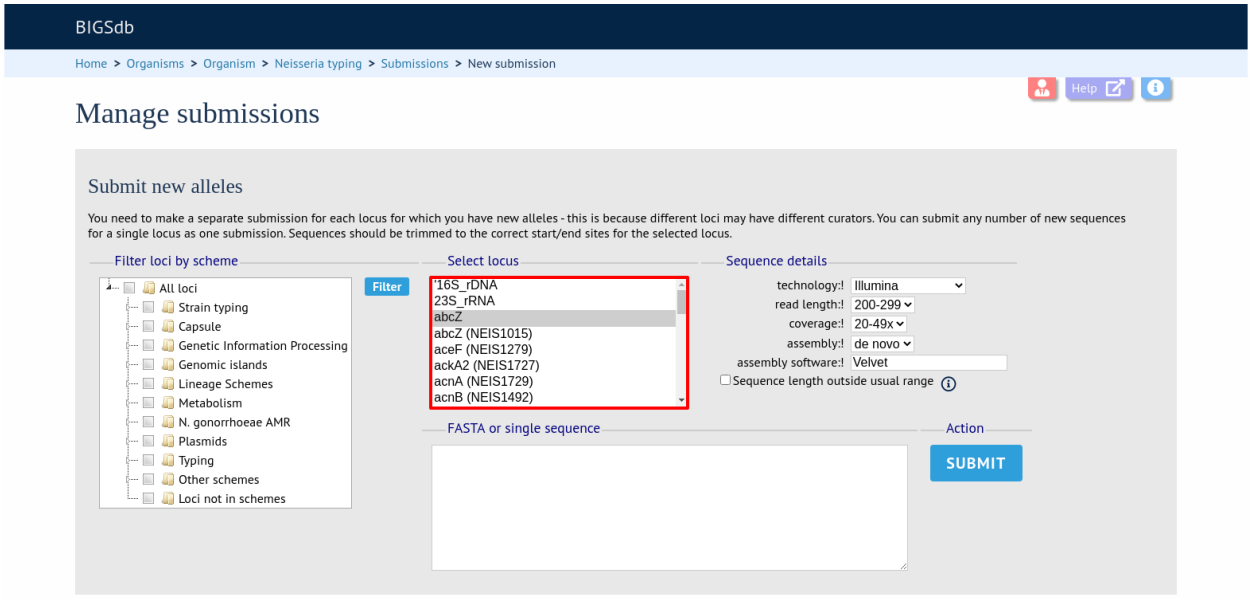
16.2.1 Start

Click the 'alleles' link under submission type on the submission management page.

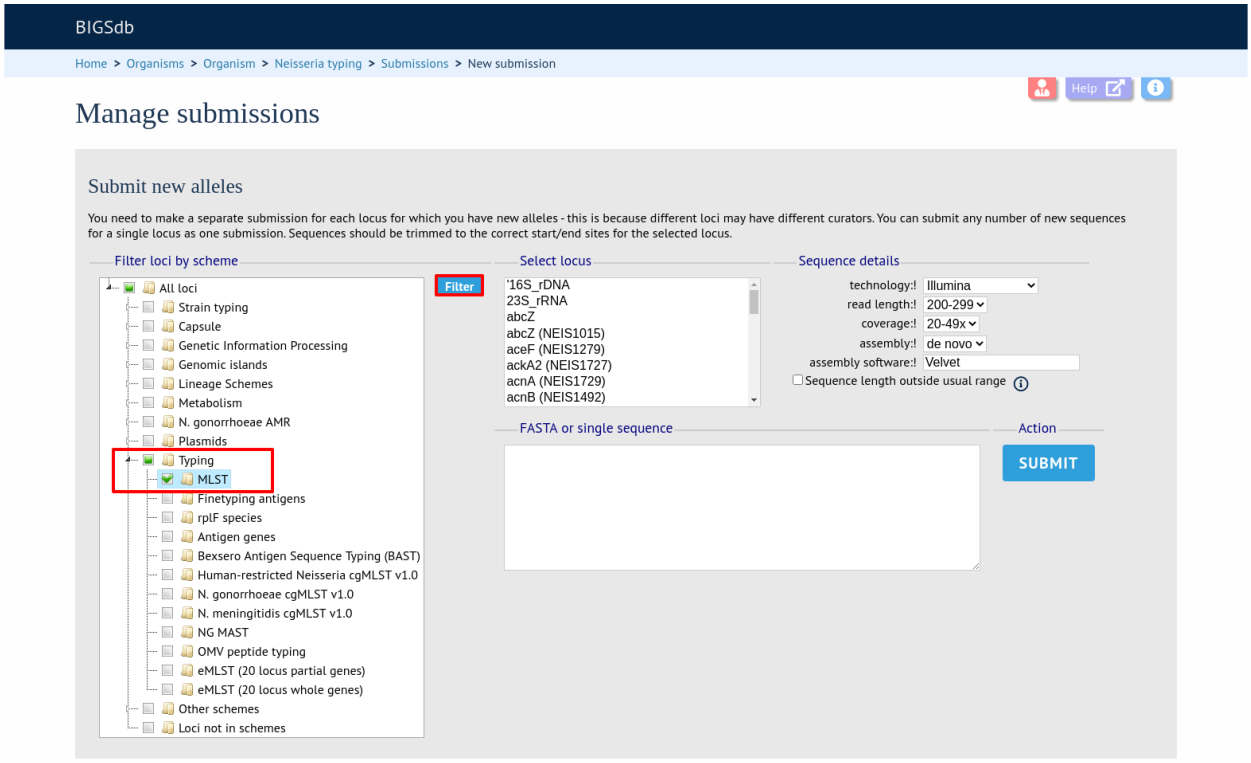


16.2.2 Select the submission locus

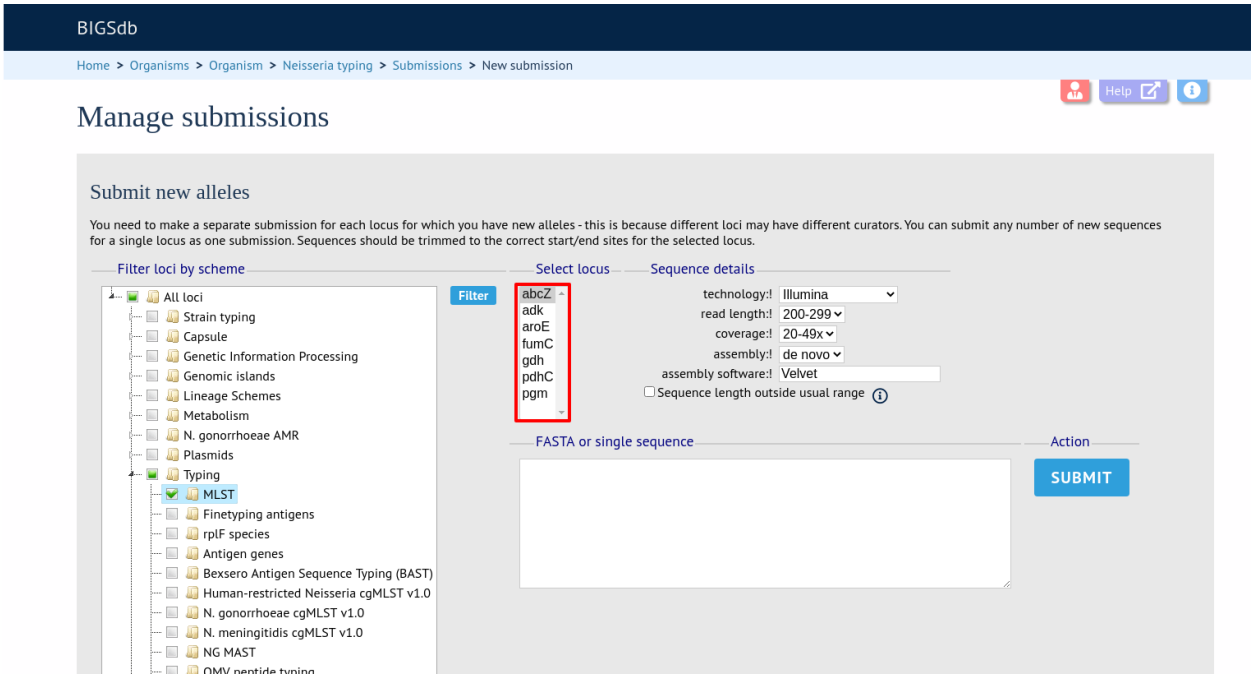
Select the locus from the locus list box:



The locus list may be very long in some databases. It may be possible to filter these to those belonging to specific schemes. If the scheme tree is shown, select the appropriate scheme, e.g. 'MLST' and click 'Filter'.



The locus list is now constrained making selection easier.



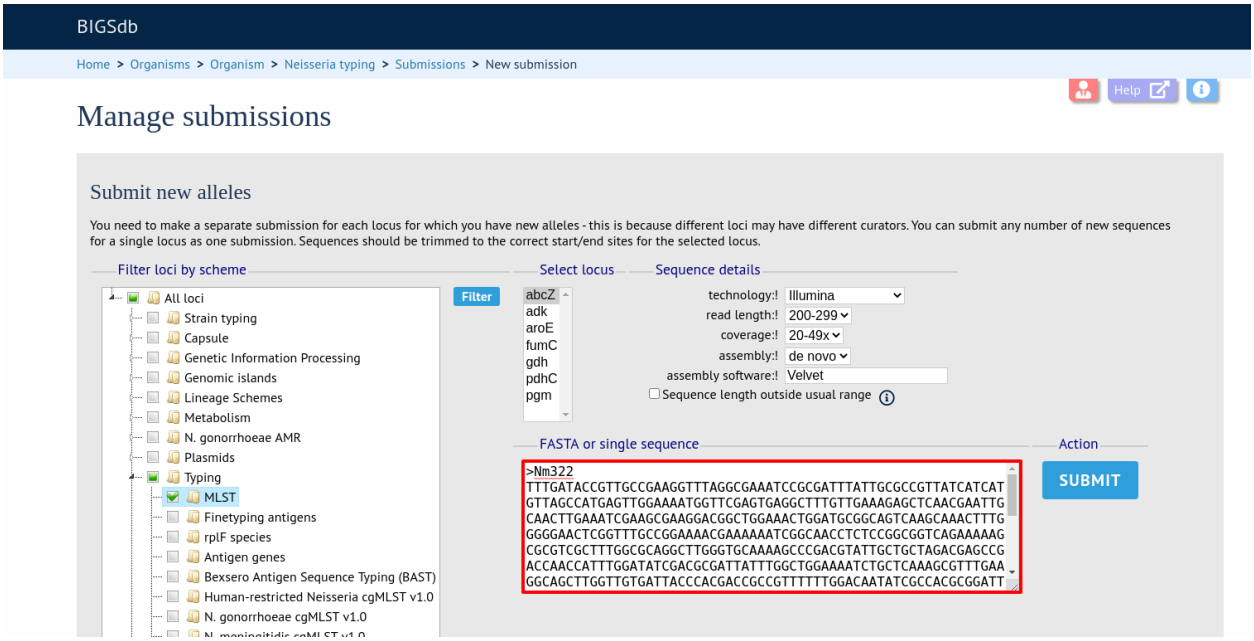
16.2.3 Enter details of sequencing method

There are a number of fields that must be filled in so that the curator knows how the sequence was obtained:

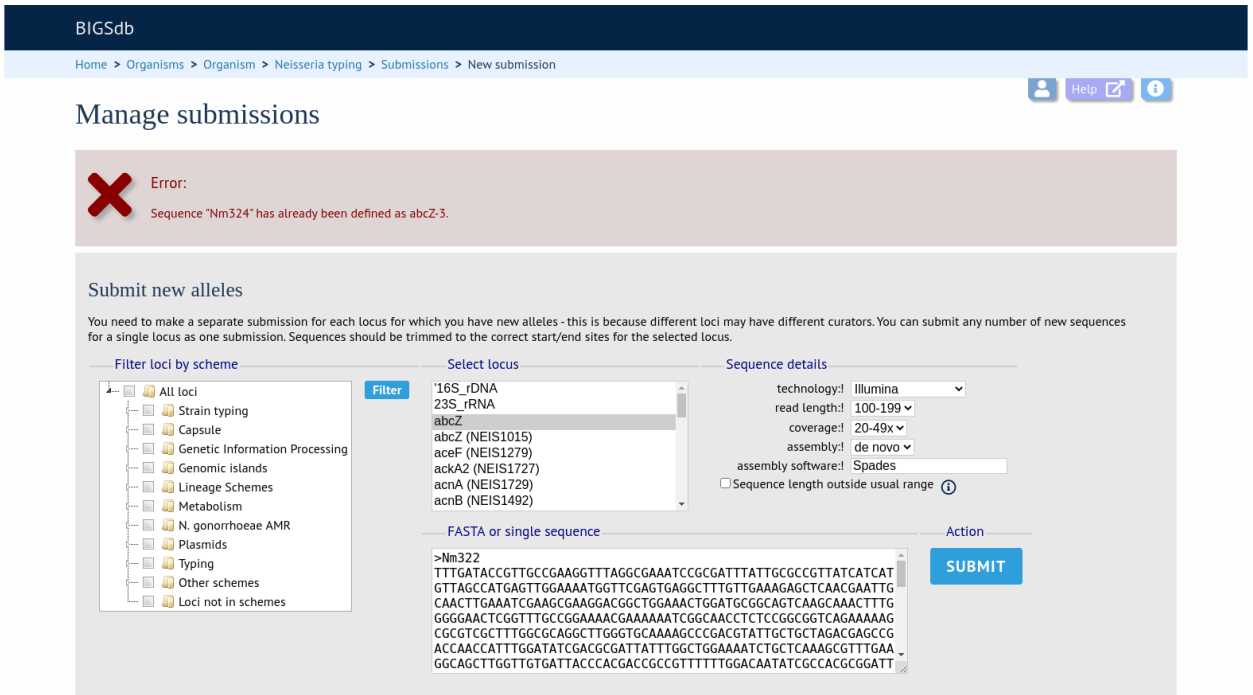
- technology - the sequencing platform used, allowed values are:
  - 454
  - Illumina
  - Ion Torrent
  - PacBio
  - Oxford Nanopore
  - Sanger
  - Solexa
  - SOLiD
  - other
  - unknown
- read length - this is the length of sequencing reads. This is a required field for Illumina data, and not relevant to Sanger sequencing. Allowed values are:
  - <100
  - 100-199
  - 200-299
  - 300-499
  - >500
- coverage - the mean number of reads covering each nucleotide position of the sequence. This is not relevant to Sanger sequencing, Allowed values are:
  - <20x
  - 20-49x
  - 50-99x
  - >100x
- assembly - the means of generating the submitted sequence from the sequencing reads. Allowed values are:
  - de novo
  - mapped
- assembly software - this is a free text field where you should enter the name of the software used to generate the submitted sequence.

16.2.4 Paste in sequence(s)

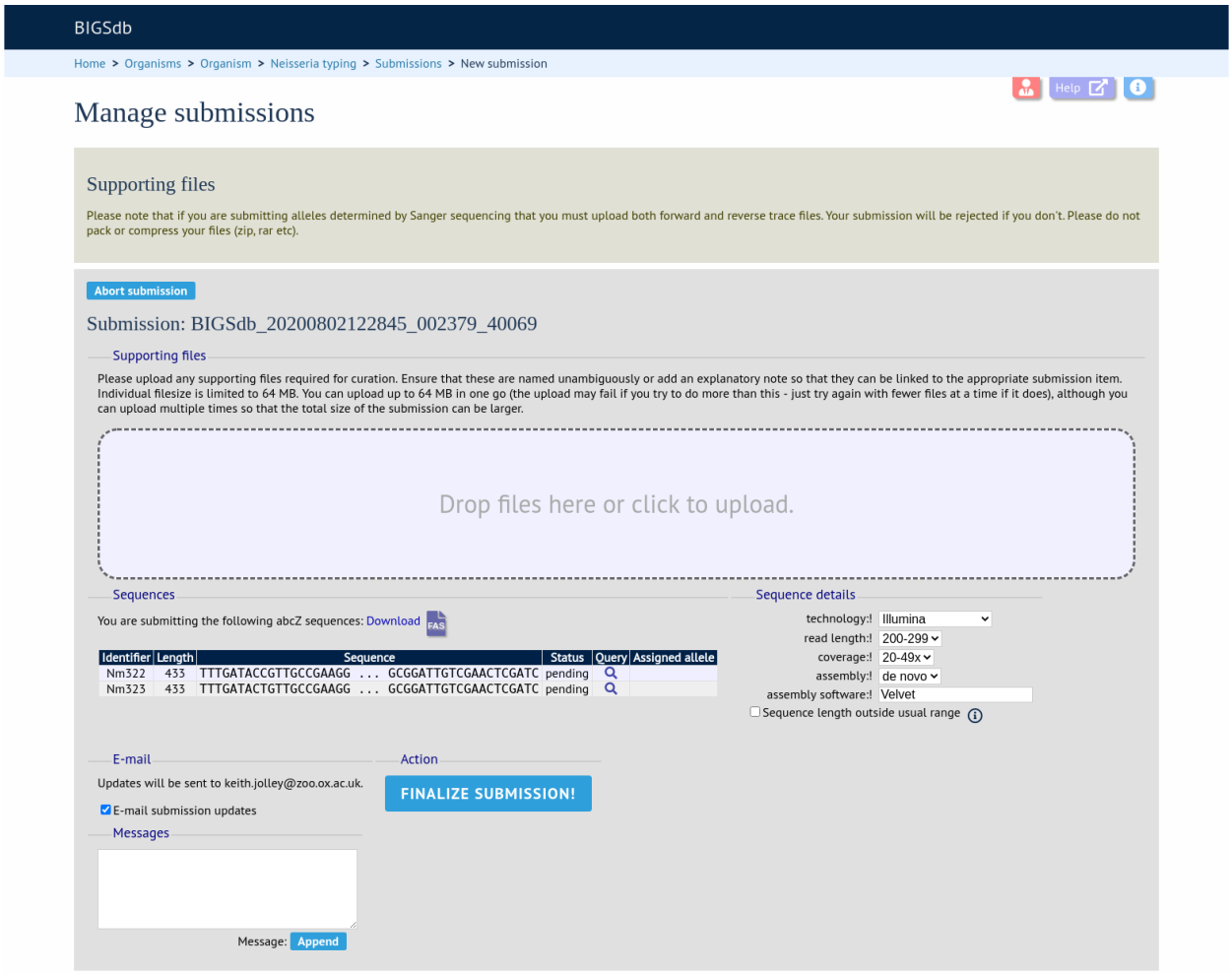
Paste in the new variant sequences to the box. This can either be a stand- alone sequence or multiple sequences in FASTA format. The sequences must be trimmed to the start and end points of the loci - check existing allele definitions if in doubt. The submission is likely to be rejected if sequences are not trimmed. Click submit.



The system will perform some basic checks on the submitted sequences. If any of the sequences have been defined previously they must be removed from the submission before you can proceed. Curators do not want to waste their time dealing with previously defined sequences.

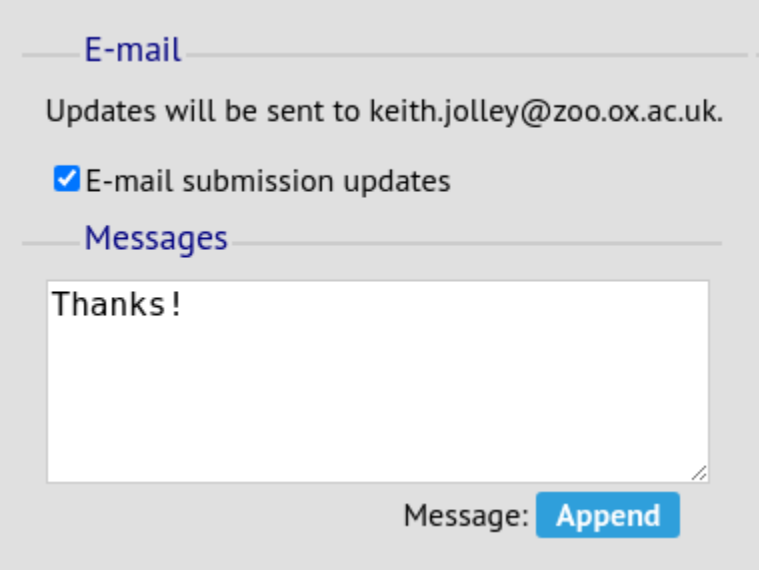


Assuming the preliminary checks have passed you will then be able to add additional information to your submission.

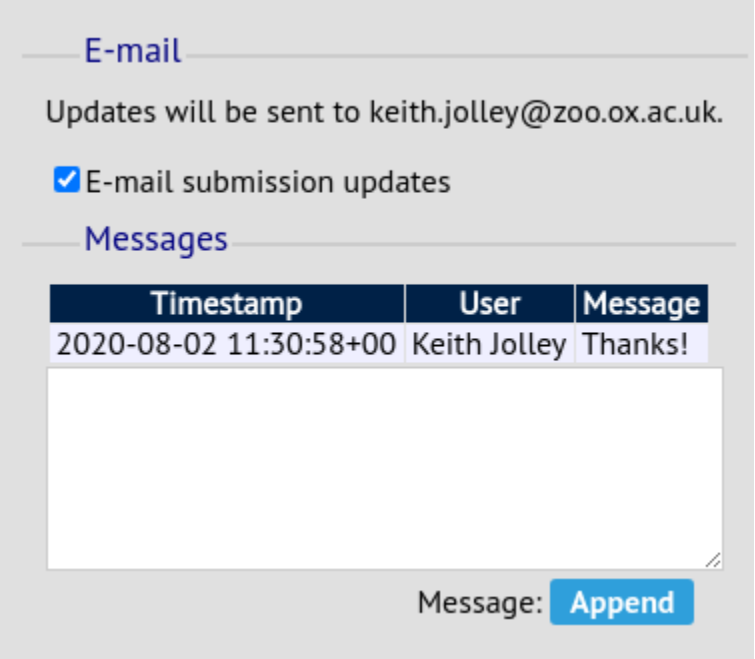


16.2.5 Add message to curator

If you wish to enter a message to the curator, enter this in the messages box and click 'Append'. This is not normally necessary for routine submissions.



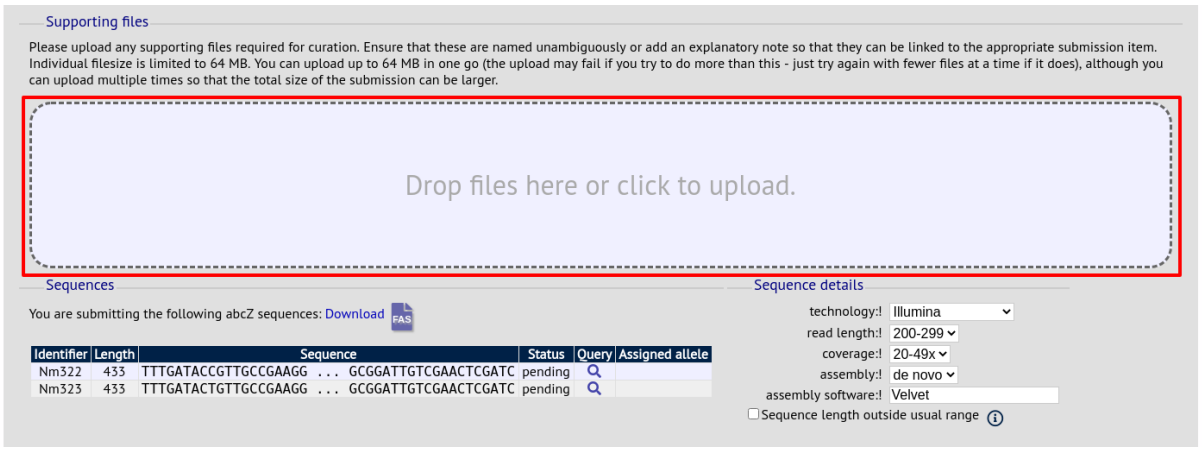
The message will be attached. A curator may respond to the message and attach their own, with the full conversation becoming part of the submission record.



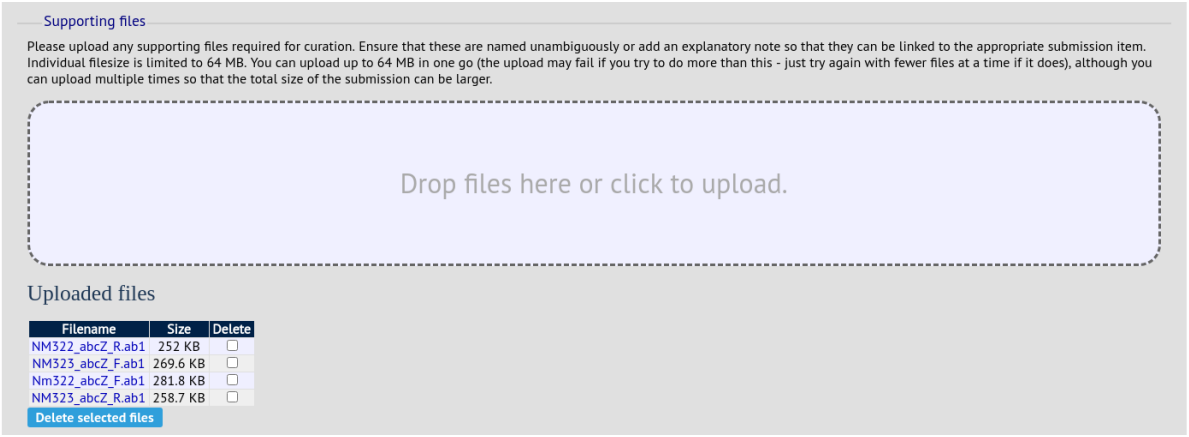
16.2.6 Add supporting files

Some submissions will require the attachment of supporting files. This will depend on the policies of the individual databases. Sequences determined by Sanger sequencing should normally have forward and reverse trace files attached.

Files can be added to the submission by dragging and dropping in to the large dotted area in the 'Supporting files' section. Alternatively, you can click this area and select files from the local file system.



The files will be uploaded and shown in a table.

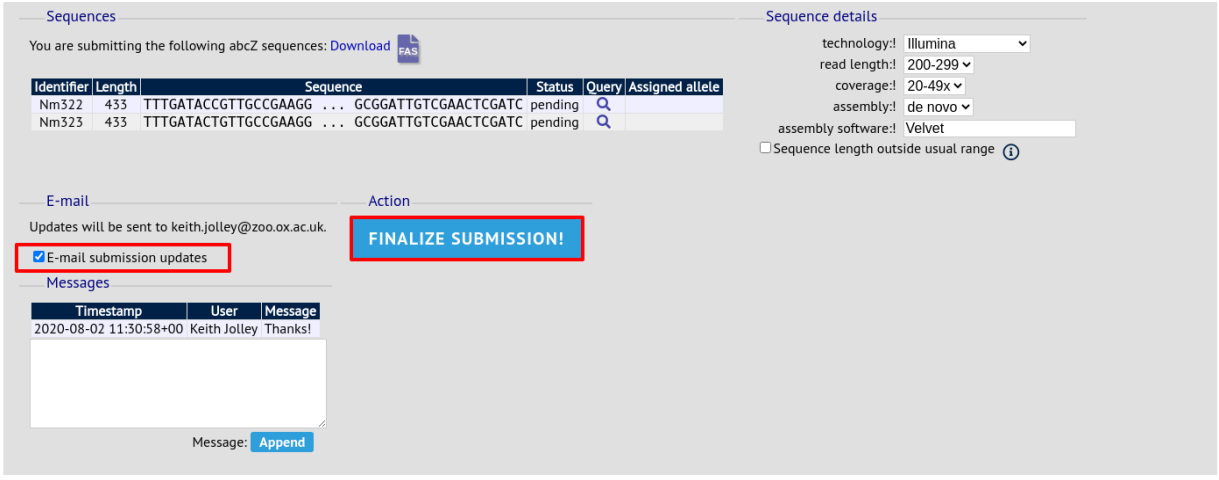


Files can be removed from the submission by checking the appropriate 'Delete' box and clicking 'Delete selected files'.

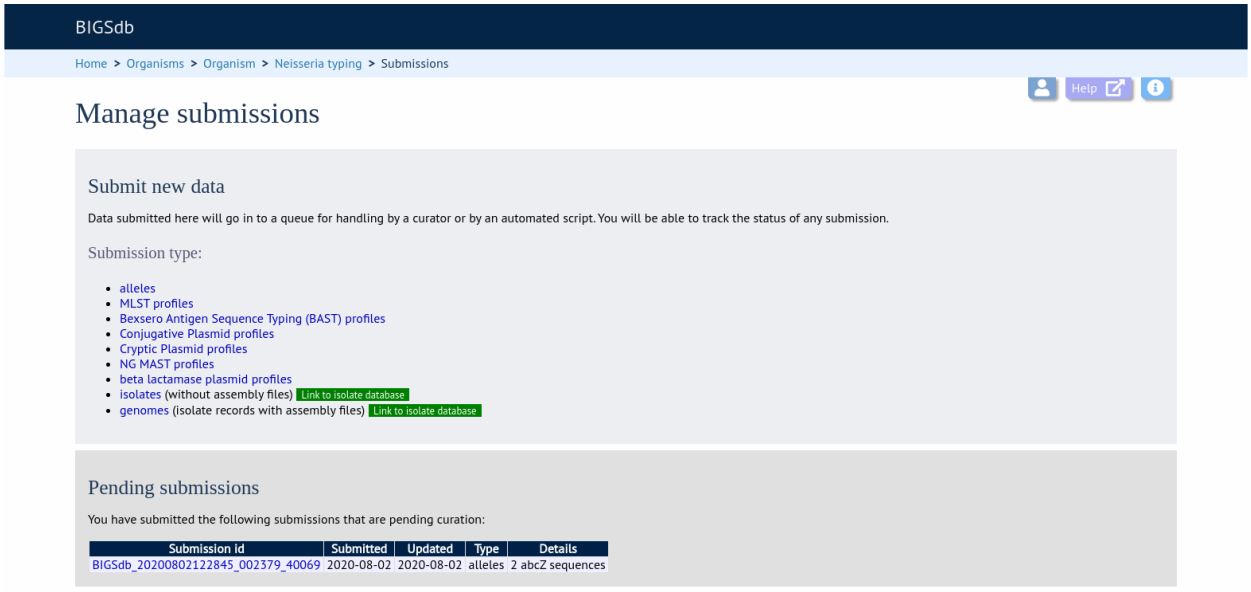
16.2.7 Finalize submission

Make sure the 'E-mail submission updates' box is checked if you wish to receive E-mail notification of the result of your submission. This setting is remembered between submissions.

Click 'Finalize submission!'.



Your submission will then be listed under 'Pending submissions' on your submission page.

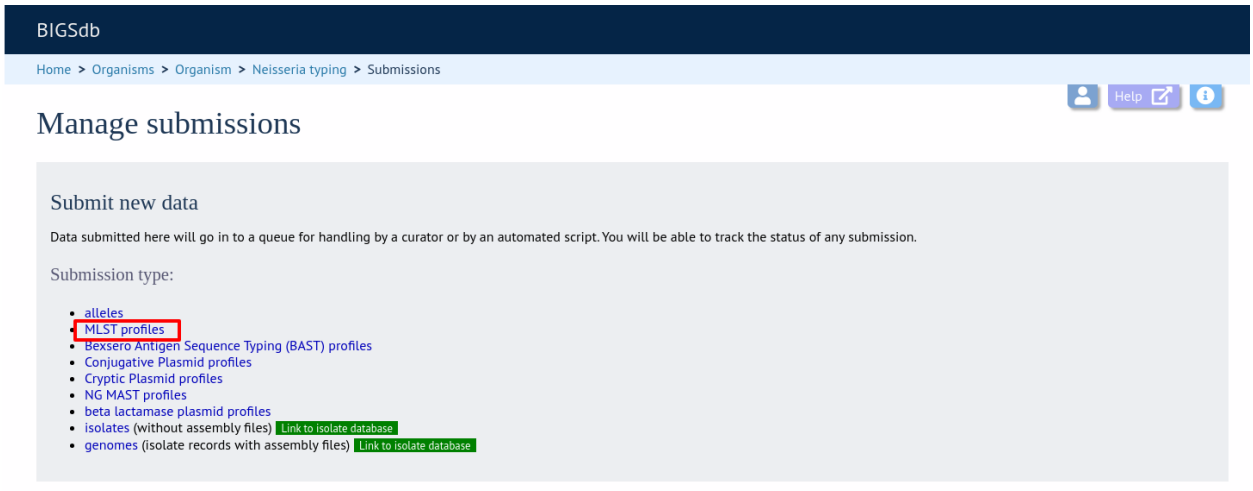


16.3 Profile submission

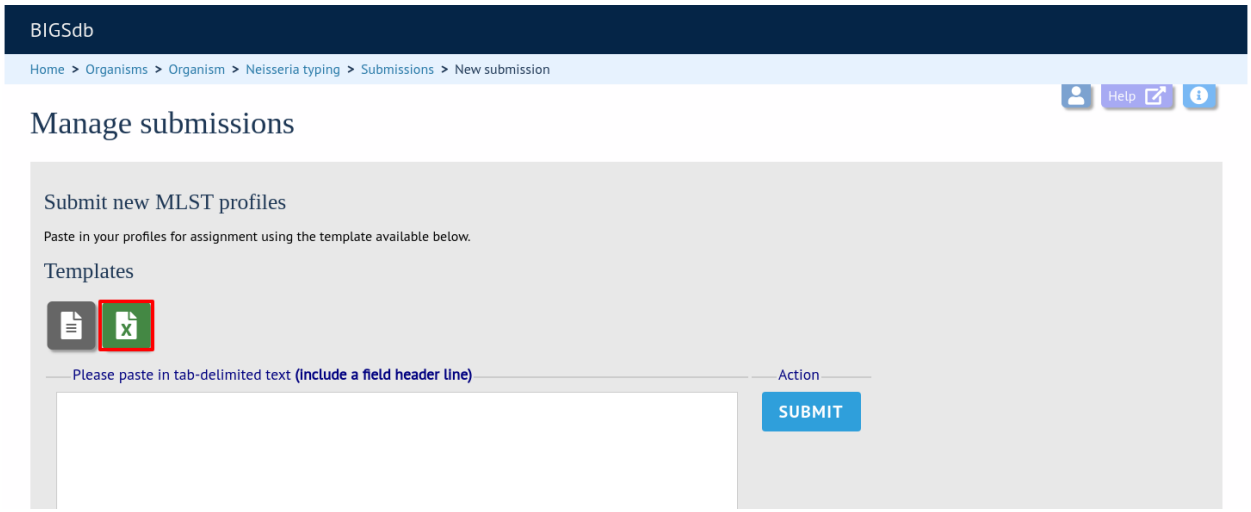
16.3.1 Start

Note: Most MLST databases on PubMLST.org require you to submit an isolate record for each new ST that you wish to be defined. In these cases, you should add the isolate name to the id field of your profile submission and make a corresponding *isolate submission* containing the allelic profile.

Click the appropriate profiles link under submission type on the submission management page.



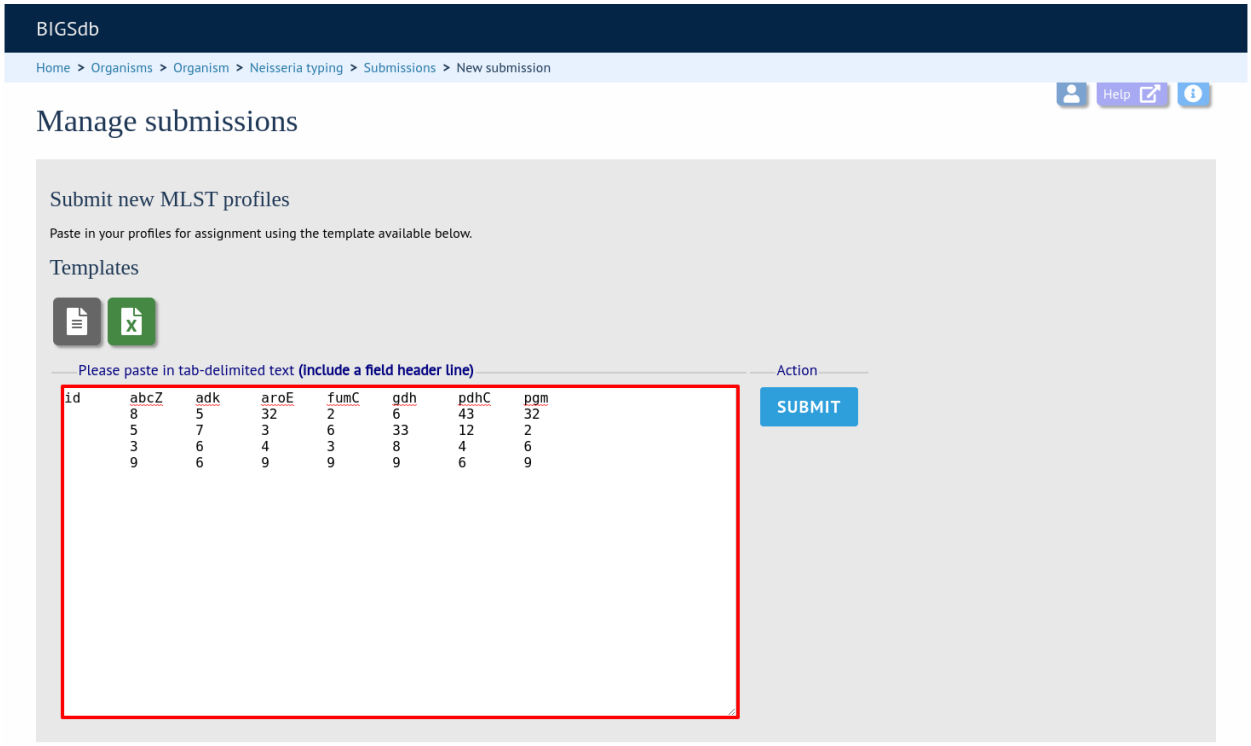
Download the Excel submission template.



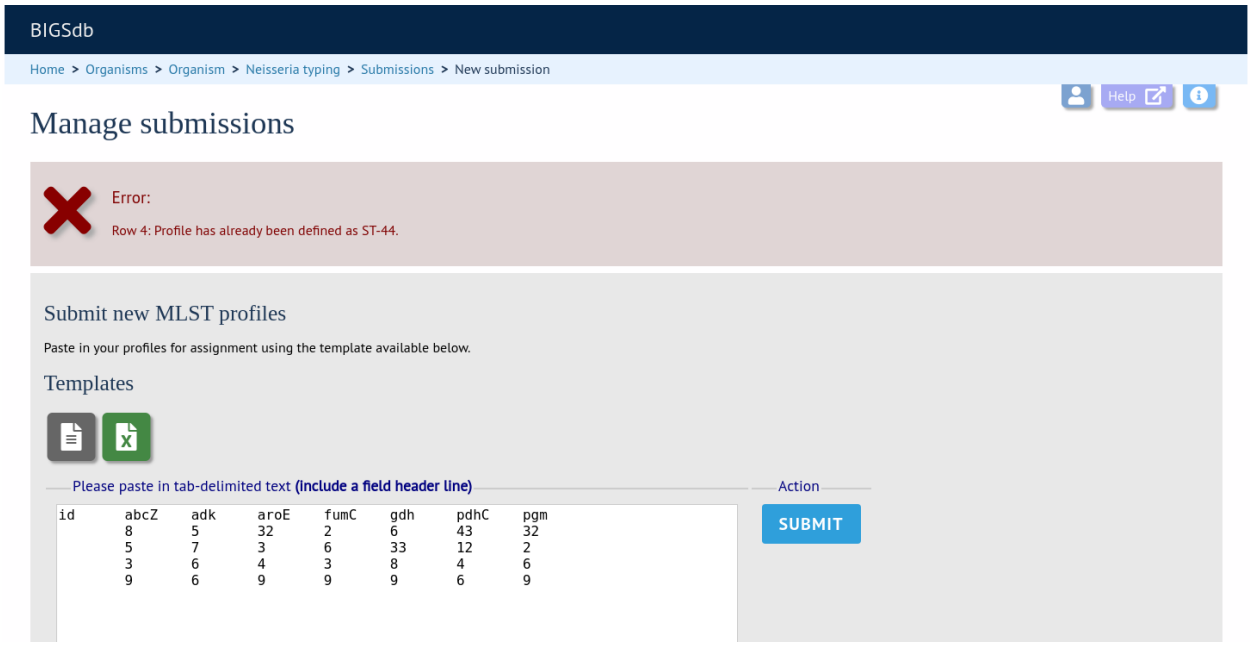
16.3.2 Paste in profile(s)

Fill in the template. The first column 'id' can be used to enter an identifier that is meaningful to you - it is used to report back the results but is not uploaded to the database. It can be left blank, or the entire column can be removed - in which case individual profiles will be identified by row number.

Copy and paste the entire contents of the submission worksheet. Click submit.



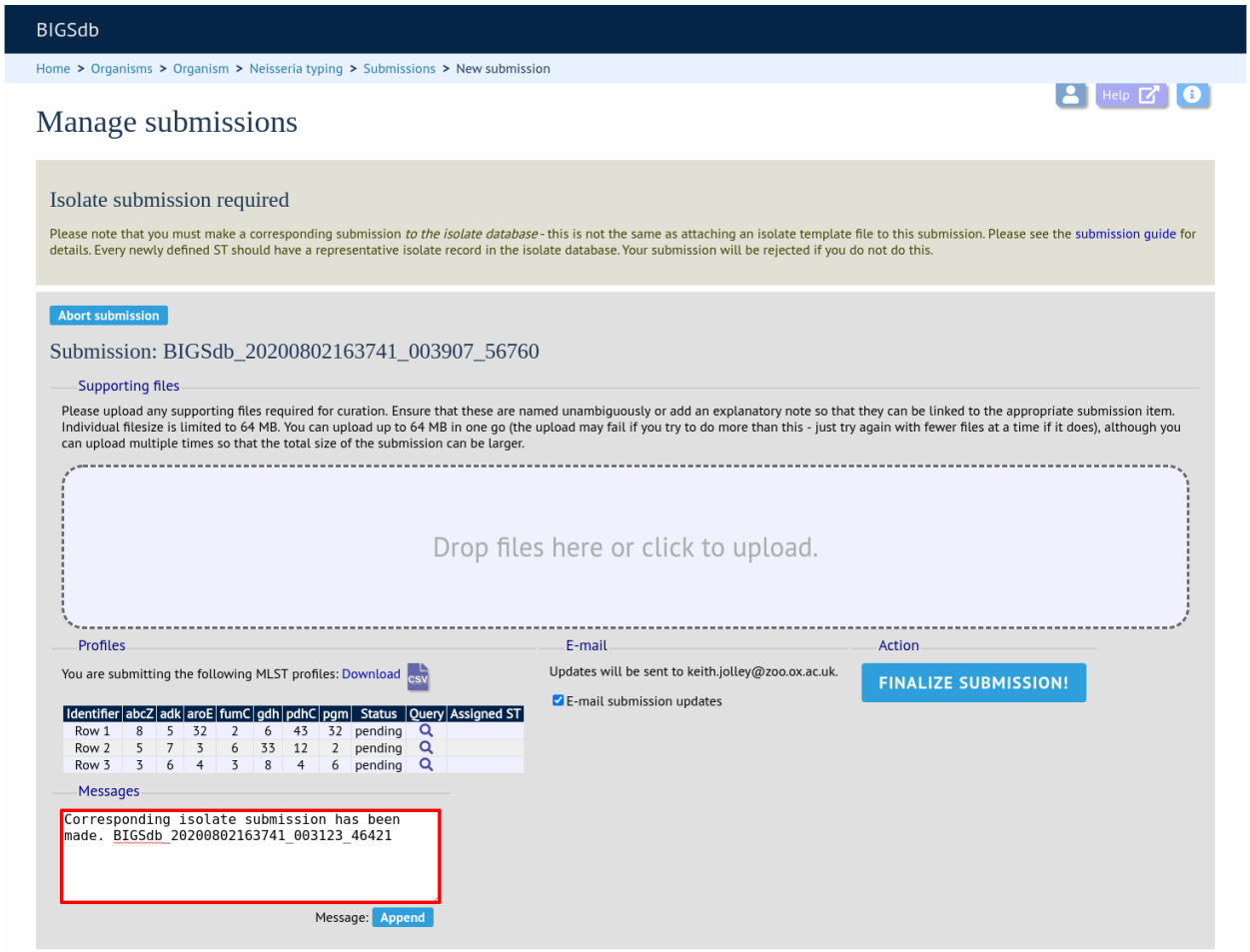
Some basic checks will be performed. These include whether the profile has already been assigned and whether each allele identifier exists. The submission cannot proceed if the checks fail.



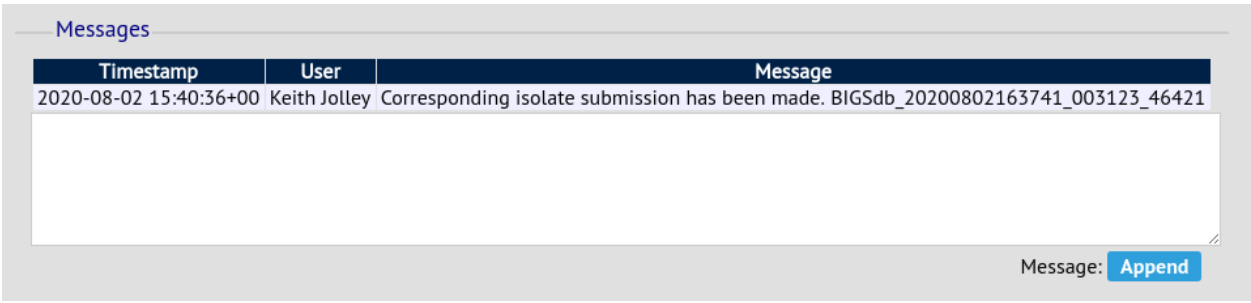
Provided the checks pass, you will then be able to add additional information to your submission. New profile submissions usually don't require supporting files directly in the submission. You generally will need to make a corresponding *submission to the isolate database* though.

16.3.3 Add message to curator

If you wish to enter a message to the curator, enter this in the messages box and click 'Append'.



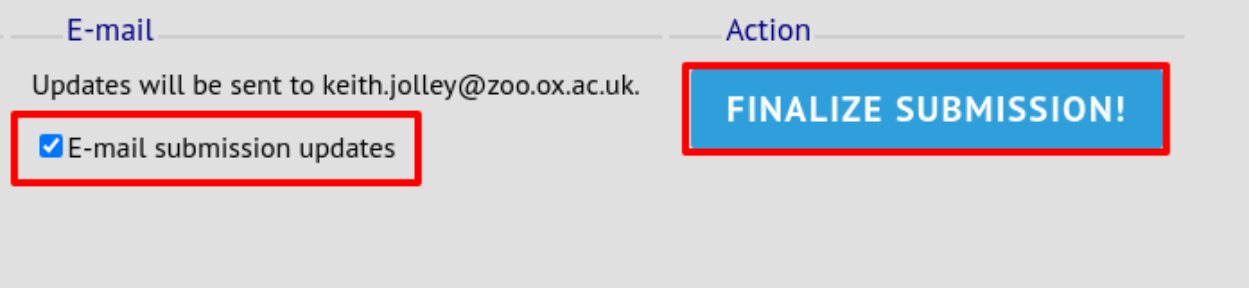
The message will be attached. A curator may respond to the message and attach their own, with the full conversation becoming part of the submission record.



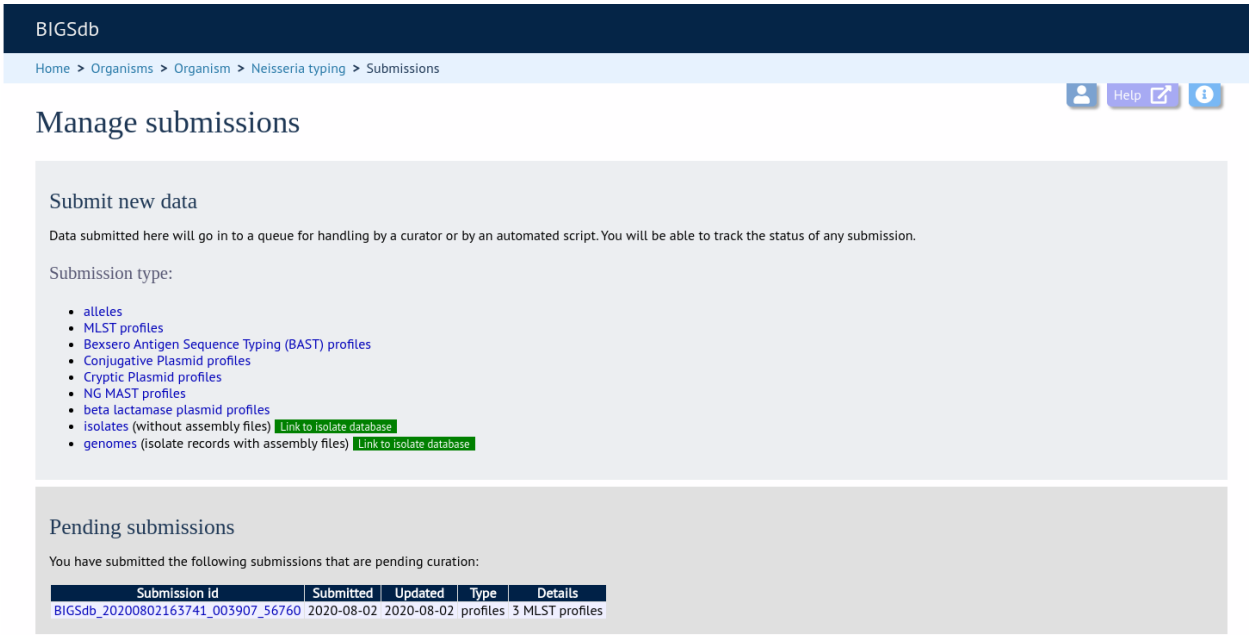
16.3.4 Finalize submission

Make sure the 'E-mail submission updates' box is checked if you wish to receive E-mail notification of the result of your submission. This setting is remembered between sessions.

Click 'Finalize submission!'.



Your submission will then be listed under 'Pending submissions' on your submission page.

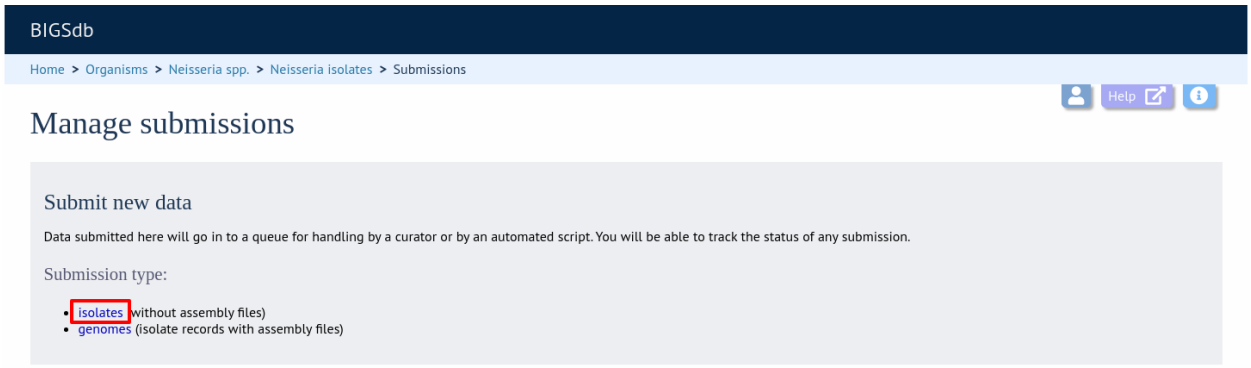


16.4 Isolate submission

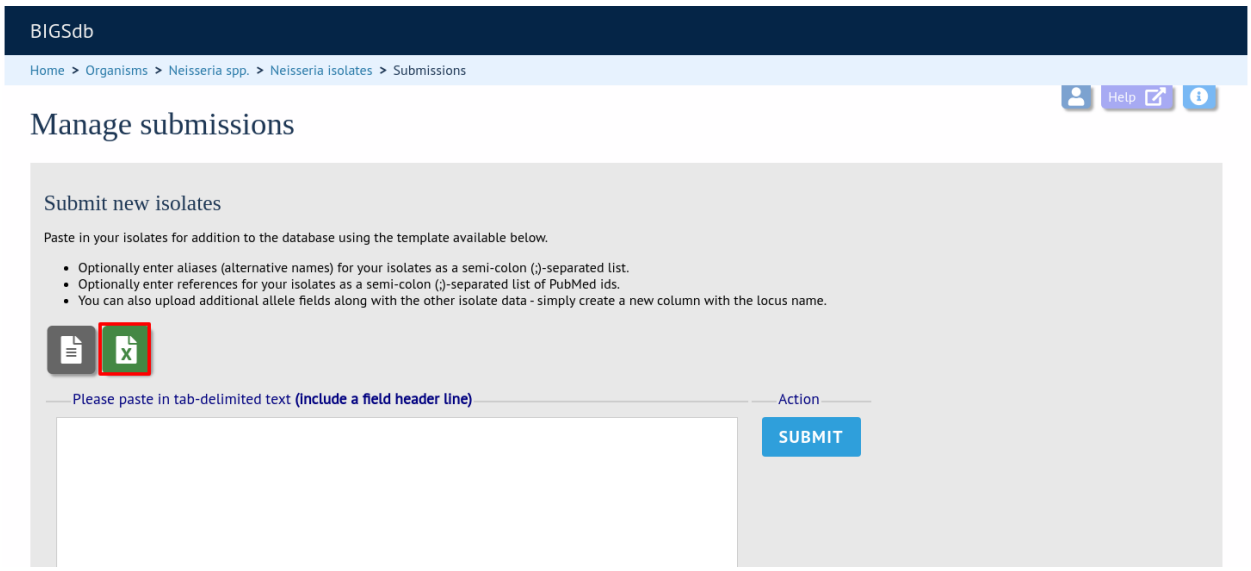
New isolate data can only be submitted from within the appropriate isolate database. You may be required to submit isolate data if you would like to get a new MLST sequence type defined, but this depends on individual database policy.

16.4.1 Start

Click the 'isolates' link under submission type on the submission management page.



Download the Excel submission template.

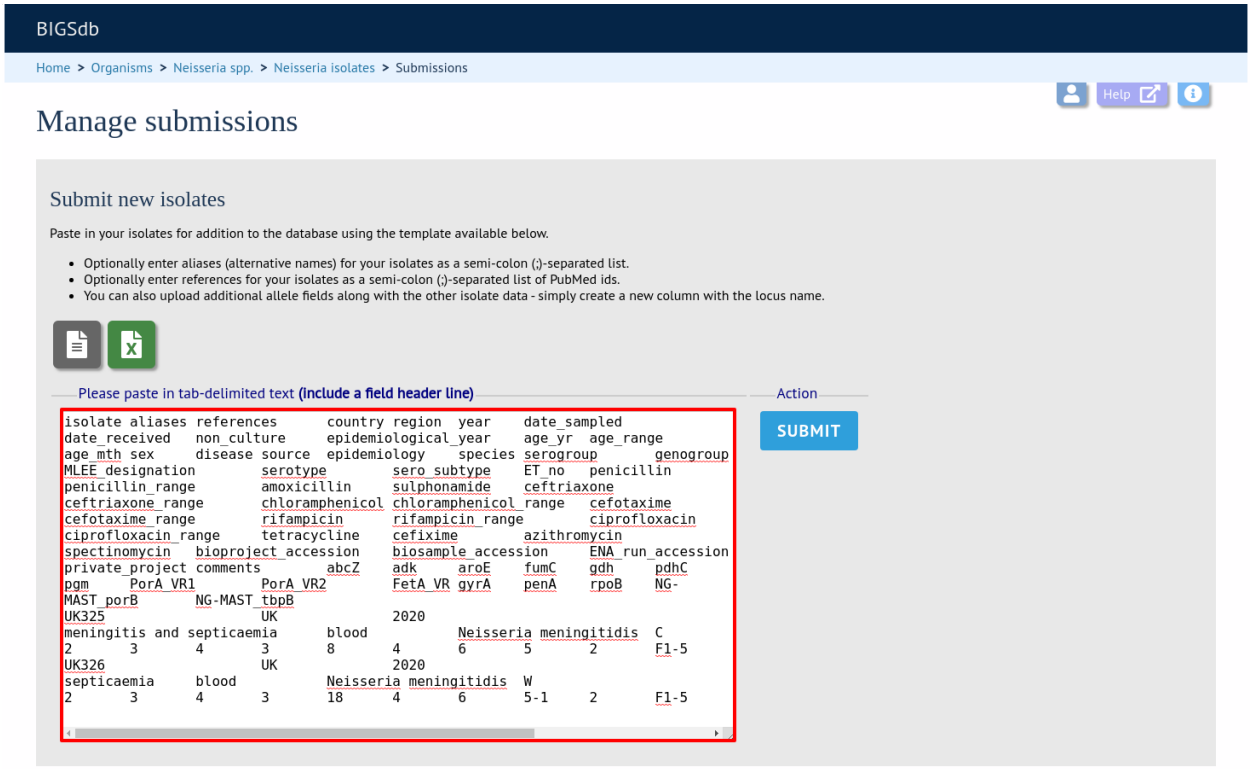


16.4.2 Paste in isolate data

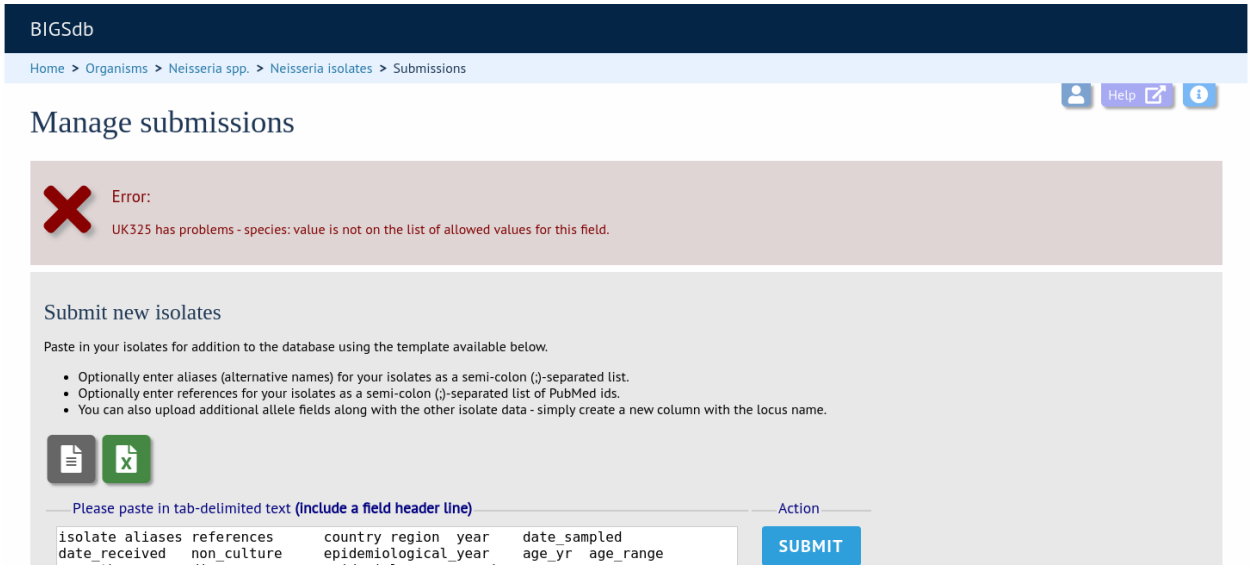
Fill in the template. Some fields are required and cannot be left blank. Check the 'Description of database fields' link on the database contents page to see a description of the fields and allowed values where these have been defined. Where allowed values have been set, the template will have dropdown boxes (although these require newer versions of Excel to work).

Some databases may have hundreds of loci defined, and most will not have a column in the template. You can add new columns for any loci that have been defined and for which you would like to include allelic information for. These locus names must be the primary locus identifier. A list of loci can be found in the 'allowed_loci' tab of the Excel submission template.

Copy and paste the entire contents of the submission worksheet. Click submit.



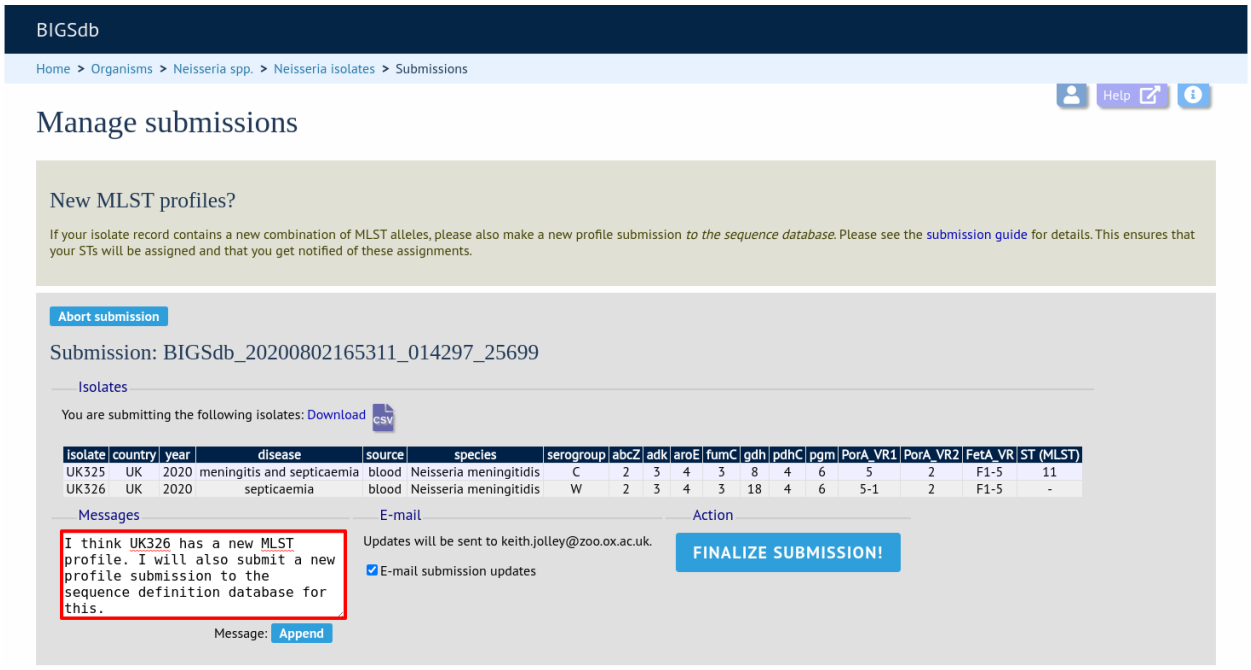
Some basic checks will be performed. These include checking all field values conform to allowed lists or data types. The submission cannot proceed if any checks fail.



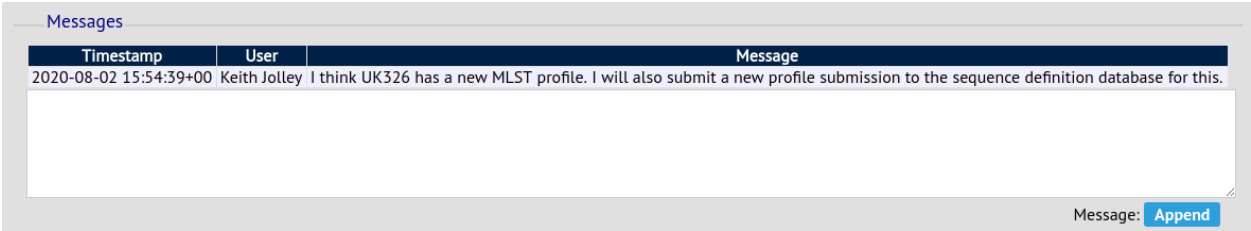
Provided the checks pass, you will then be able to add additional information to your submission.

16.4.3 Add message to curator

If you wish to enter a message to the curator, enter this in the messages box and click 'Append'.



The message will be attached. A curator may respond to the message and attach their own, with the full conversation becoming part of the submission record.

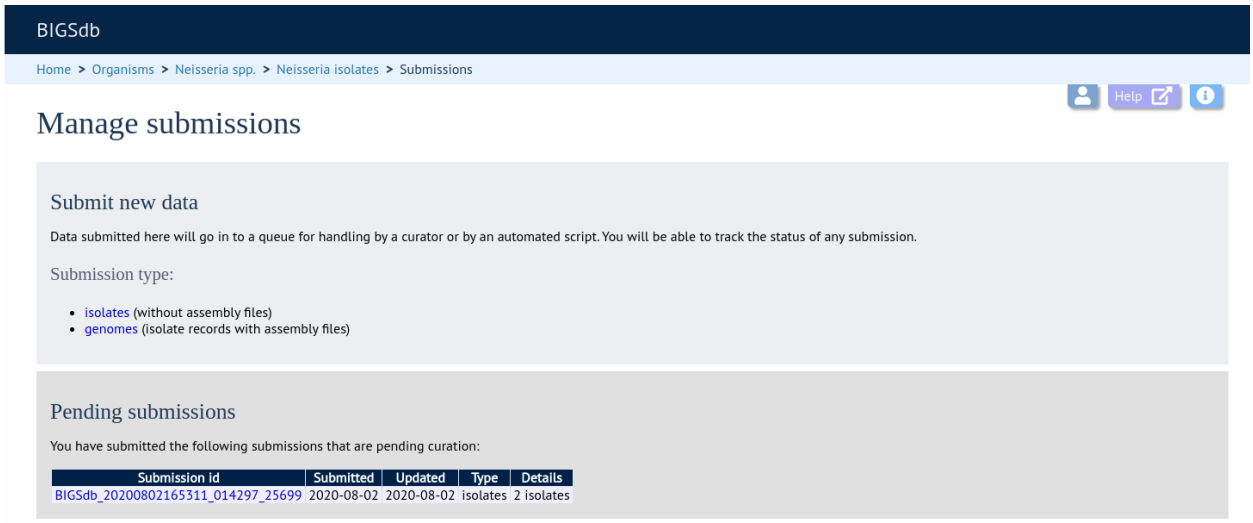


16.4.4 Finalize submission

Make sure the 'E-mail submission updates' box is checked if you wish to receive E-mail notification of the result of your submission. This setting is remembered between sessions.

Click 'Finalize submission!'.

Your submission will then be listed under 'Pending submissions' on your submission page.



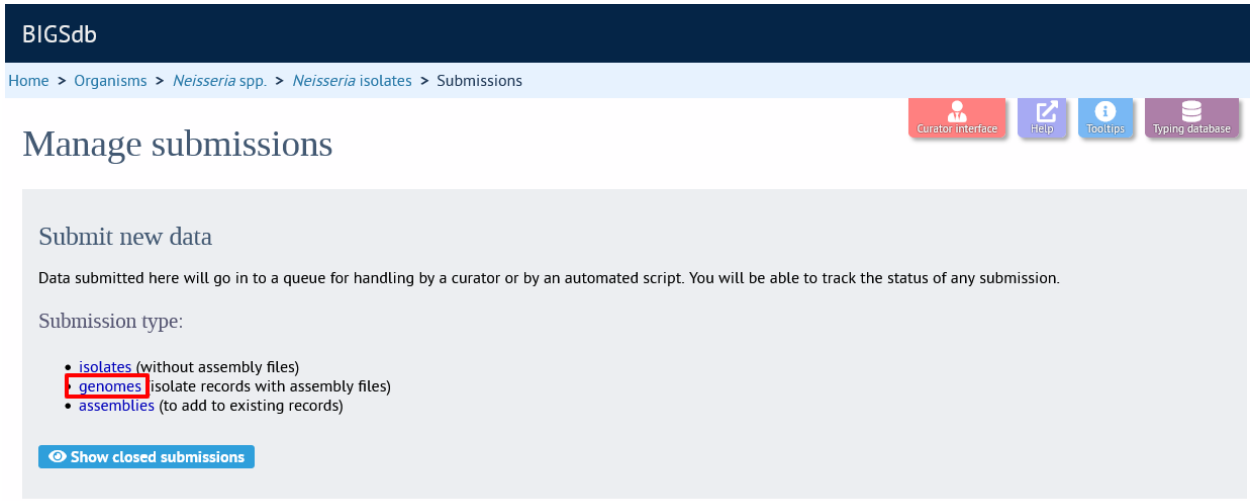
16.5 Genome submission

Submitting genomes uses the same process as standard *isolate submission*. The only difference is that there are a couple of extra required fields in the submission table:

- assembly_filename - this is the name of the FASTA file containing the assembly contigs. This must be uploaded as a supporting file - you will not be able to finalize the submission until every isolate record has a matching contig file.
- sequence_method - the sequencing technology used to generate the sequences. The allowed values are listed on the submission page.

Locus fields are not usually included in a genome submission as these can be readily extracted from the genome.

To start the submission, click the 'genomes' link under submission type on the submission management page.



Then follow the steps for *isolate submission*, uploading the contig files as supporting files. You will be able to finalize the submission only after all the assembly files have been uploaded.

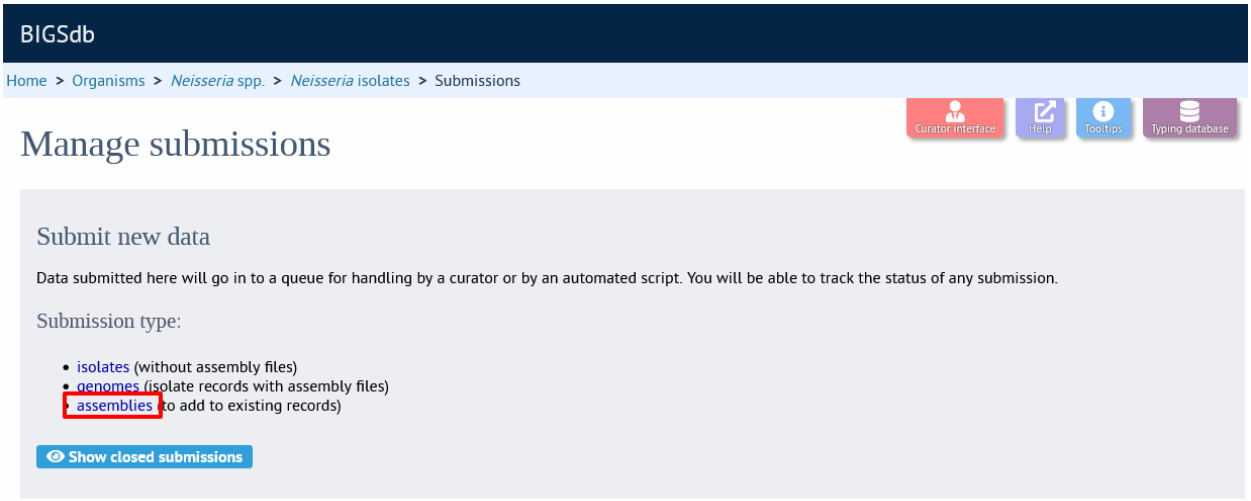
Note: When including the filename for your FASTA file containing the genome assembly, please note that Windows will, by default, hide the file extension, e.g. .fas or .fasta. Even if it is hidden in the Windows interface, the file extension is part of the filename and must be included so that the uploaded file has exactly the same name as entered in the submission template. See https://www.techadvisor.co.uk/how-to/windows/windows-10-file-extensions-3697651 to see how to display hidden file extensions in Windows 10.

16.6 Assembly submission

Genome assemblies can be submitted to add to existing isolate records. These are often old records that have been submitted with just MLST results but whole genome sequencing has been performed later.

16.6.1 Start

Click the 'assemblies' link under submission type on the submission management page.

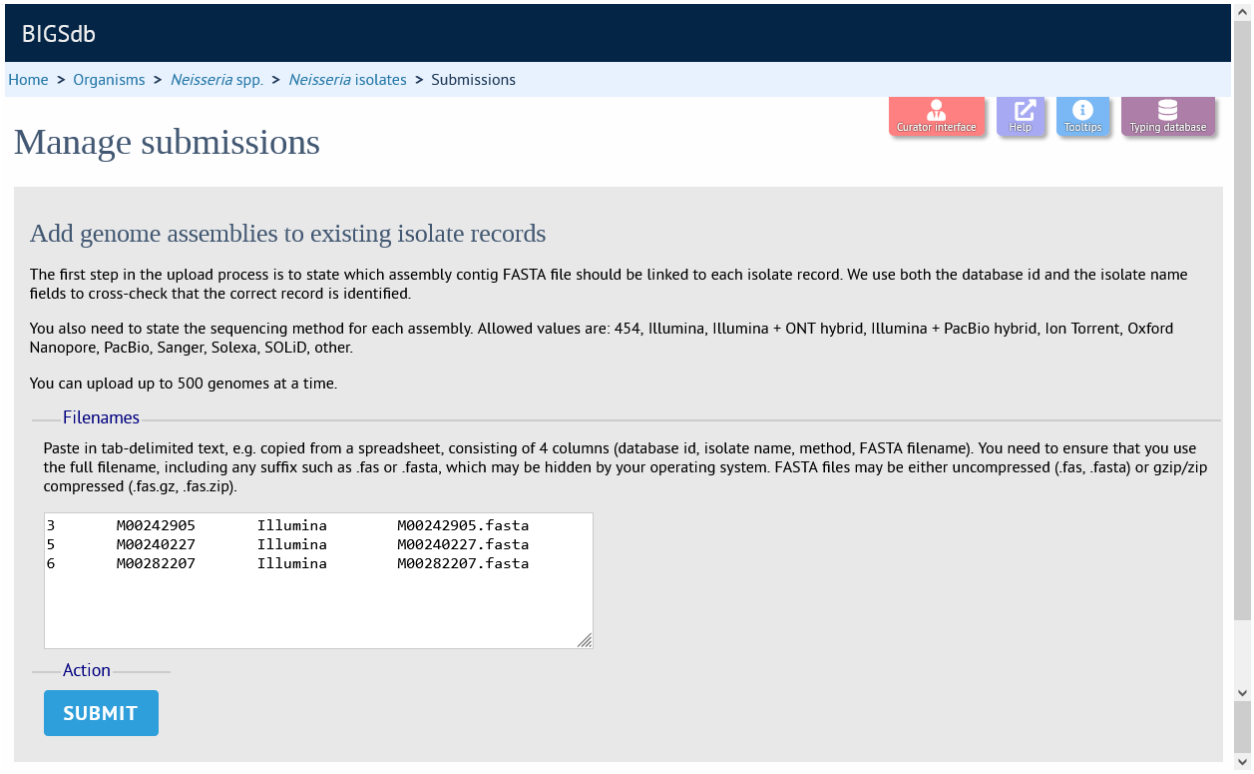


16.6.2 Link assembly files to isolate records

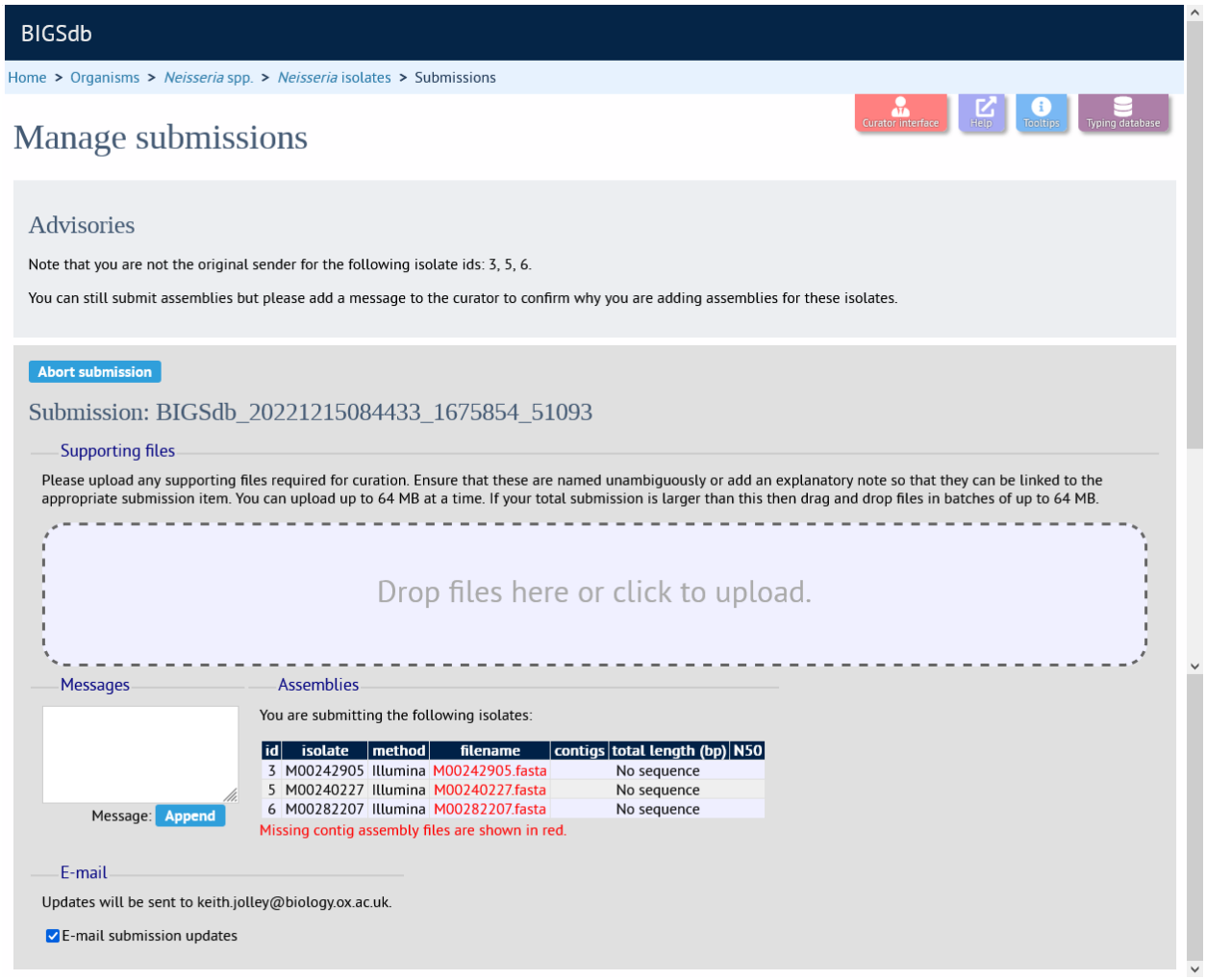
You need to tell the system which isolate record should be linked to each assembly that is being uploaded. In order to do this you should prepare a spreadsheet consisting of four columns that you then copy and paste into the web form. The columns are:

- database id number
- isolate name
- sequence method
- assembly filename

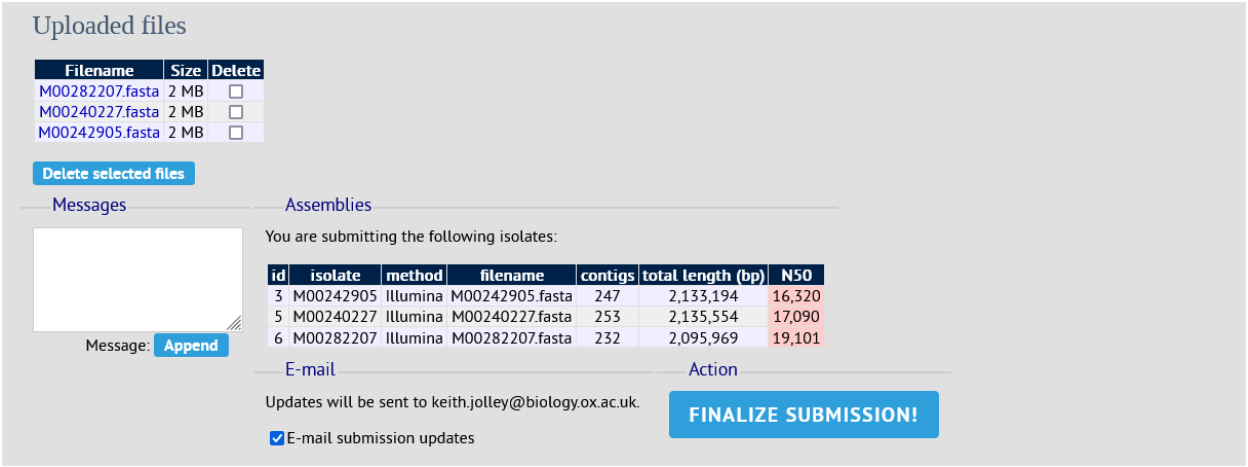
Both the database id and isolate name are used so that they can be cross-checked to ensure that the correct isolate record has been selected.



Click 'Submit' and the system will check that the id and isolate names correspond to isolate records that do not currently have assemblies. Provided these match, you will then be prompted to drag-and-drop your genome assemblies on to the web form. A check will also be performed to see if you are the original submitter of the isolate. If you are not, you can still make the submission but should add a message to the curator to confirm why you are adding assemblies for these records.



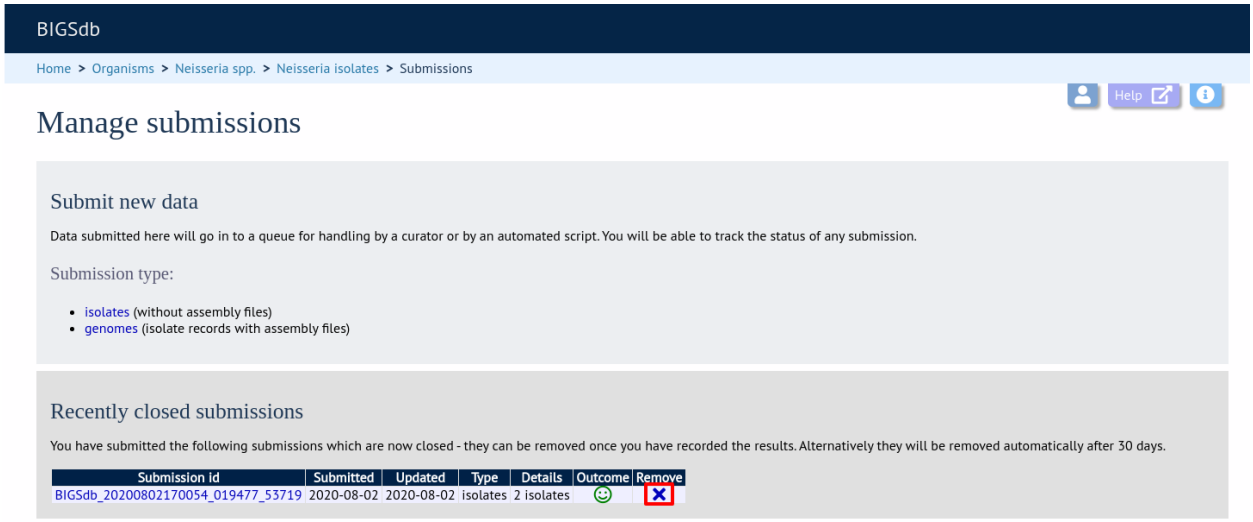
Files are uploaded as you drag-and-drop them. Basic checks wil be performed for sequence length, number of contigs, and N50 values. If values are outside the preferred range you will see a warning for a specific value shown with a pink background. If values are outside the allowed range than the validation will fail and you will need to abort the submission. In the image below, the N50 values have a warning but have not failed the validation.



Once the files have been uploaded and passed validation, add any message to the curator if necessary, e.g. if the original isolate submissions were made by someone else. Click 'Finalize submission'.

16.7 Removing submissions from your notification list

Once a submission has been closed by a curator, the results will be displayed in your 'Manage submissions' area. You can remove submissions once you have noted the result by clicking the 'Remove' link.



Alternatively, submissions will be removed automatically a specified period of time after closure. By default, this time is 90 days, but this can vary depending on the site configuration.

CHAPTER

SEVENTEEN

RESTFUL APPLICATION PROGRAMMING INTERFACE (API)

The REST API allows third-party applications to retrive data stored within BIGSdb databases or to send new submissions to database curators. To use the REST API, your application will make a HTTP request and parse the response. The response format is JSON (except for routes that request a FASTA or CSV file).

Access to protected resources, i.e. those requiring an account, can be accessed via the API using *OAuth authentication*.

17.1 Passing additional/optional parameters

If you are using a method called with GET, optional parameters can be passed as arguments to the query URL by adding a '?' followed by the first argument and its value (separated by a '='). Additional parameters are separated by a '&', e.g.

https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates?page=2&page_size=100

Methods called with POST require their arguments to be sent as JSON within the post body.

17.2 Paging using request headers

Paging of results can be selected using query parameters as described above. In this case, methods that support paging will include a paging object in the JSON response. This will contain links to the next page, last page etc.

The API also supports paging using request headers. The following request headers are supported:

- X-OFFSET
- X-PER-PAGE

e.g.

```
curl -i -H "X-PER-PAGE:10" -H "X-OFFSET:0" https://rest.pubmlst.org/db/pubmlst_neisseria_
↪isolates/isolates
```

If either of these headers are used, the paging object is no longer returned as part of the JSON response. The response will include the following headers:

- X-OFFSET
- X-PER-PAGE
- X-TOTAL-PAGES

17.3 Resources

- *GET / or /db* - List site resources
- *GET /db/{database}* - List database resources
- *GET /db/{database}/classification_schemes* - List classification schemes
- *GET /db/{database}/classification_schemes/{classification_scheme_id}* - Retrieve classification scheme information and groups
- *GET /db/{database}/classification_schemes/{classification_scheme_id}/groups* - List groups defined for a classification scheme
- *GET /db/{database}/classification_schemes/{classification_scheme_id}/groups/{group_id}* - List isolates or profiles belonging to a classification scheme group
- *GET /db/{database}/loci* - List loci
- *GET /db/{database}/loci/{locus}* - Retrieve locus record
- *GET /db/{database}/loci/{locus}/alleles* - Retrieve list of alleles defined for a locus
- *GET /db/{database}/loci/{locus}/alleles_fasta* - Download alleles in FASTA format
- *GET /db/{database}/loci/{locus}/alleles/{allele_id}* - Retrieve full allele information
- *POST /db/{database}/loci/{locus}/sequence* - Query sequence to identify allele
- *POST /db/{database}/sequence* - Query sequence to identify allele without specifying locus
- *GET /db/{database}/sequences* - Get summary of defined sequences
- *GET /db/{database}/schemes* - List schemes
- *GET /db/{database}/schemes/{scheme_id}* - Retrieve scheme information
- *GET /db/{database}/schemes/{scheme_id}/loci* - Retrieve scheme loci
- *GET /db/{database}/schemes/{scheme_id}/fields/{field}* - Retrieve information about scheme field
- *GET /db/{database}/schemes/{scheme_id}/profiles* - List allelic profiles defined for scheme
- *GET /db/{database}/schemes/{scheme_id}/profiles_csv* - Download allelic profiles in CSV (tab-delimited) format
- *GET /db/{database}/schemes/{scheme_id}/profiles/{profile_id}* - Retrieve allelic profile record
- *POST /db/{database}/schemes/{scheme_id}/sequence* - Query sequence to extract allele designations/fields for a scheme
- *POST /db/{database}/schemes/{scheme_id}/designations* - Query allelic profile to extract fields for a scheme
- *GET /db/{database}/isolates* - Retrieve list of isolate records
- *GET /db/{database}/genomes* - Retrieve list of isolate records that have genome assemblies
- *POST /db/{database}/isolates/search* - Search isolate database
- *GET /db/{database}/isolates/{isolate_id}* - Retrieve isolate record
- *GET /db/{database}/isolates/{isolate_id}/allele_designations* - Retrieve list of allele designations
- *GET /db/{database}/isolates/{isolate_id}/allele_designations/{locus}* - Retrieve full allele designation record
- *GET /db/{database}/isolates/{isolate_id}/allele_ids* - Retrieve allele identifiers

- *GET /db/{database}/isolates/{isolate_id}/schemes/{scheme_id}/allele_designations* - Retrieve scheme allele designation records
- *GET /db/{database}/isolates/{isolate_id}/schemes/{scheme_id}/allele_ids* - Retrieve list of scheme allele identifiers
- *GET /db/{database}/isolates/{isolate_id}/contigs* - Retrieve list of contigs
- *GET /db/{database}/isolates/{isolate_id}/contigs_fasta* - Download contigs in FASTA format
- *GET /db/{database}/isolates/{isolate_id}/history* - Retrieve isolate update history
- *GET /db/{database}/contigs/{contig_id}* - Retrieve contig record
- *GET /db/{database}/fields* - Retrieve list of isolate provenance field descriptions
- *GET /db/{database}/fields/{field}* - Retrieve values set for a provenance field
- *GET /db/{database}/users/{user_id}* - Retrieve user information
- *GET /db/{database}/curators* - Retrieve list of curators of the database
- *GET /db/{database}/projects* - Retrieve list of projects
- *GET /db/{database}/projects/{project_id}* - Retrieve project information
- *GET /db/{database}/projects/{project_id}/isolates* - Retrieve list of isolates belonging to a project
- *GET /db/{database}/submissions* - Retrieve list of submissions
- *POST /db/{database}/submissions* - Create new submission
- *GET /db/{database}/submissions/{submission_id}* - Retrieve submission record
- *DELETE /db/{database}/submissions/{submission_id}* - Delete submission record
- *GET /db/{database}/submissions/{submission_id}/messages* - Retrieve submission correspondence
- *POST /db/{database}/submissions/{submission_id}/messages* - Add submission correspondence
- *GET /db/{database}/submissions/{submission_id}/files* - retrieve list of supporting files uploaded for submission
- *POST /db/{database}/submissions/{submission_id}/files* - Upload submission supporting file
- *GET /db/{database}/submissions/{submission_id}/files/{filename}* - Download submission supporting file
- *DELETE /db/{database}/submissions/{submission_id}/files/{filename}* - Delete submission supporting file

17.3.1 GET / or /db - List site resources

Required route parameters: None

Optional query parameters: None

Example request URI: https://rest.pubmlst.org/

Response: List of resource groupings (ordered by name). Groups may consist of paired databases for sequence definitions and isolate data, or any set of related resources. Each group contains:

- name [string] - short name (usually a single word)
- description [string] - fuller description
- databases [array] - list of database objects, each consists of three key/value pairs:
  - name [string] - name of database config
  - description [string] - short description of resource

  - href [string] - URI to access resource

17.3.2 GET /db/{database} - List database resources

These will vary depending on whether the resource is an isolate or a sequence definition database.

Required route parameter: database [string] - Database configuration name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates

Response: Object containing a subset of the following key/value pairs:

- *fields* [string] - URI to isolate provenance field information
- *isolates* [string] - URI to isolate records
- *genomes* [string] - URI to genome records
- *schemes* [string] - URI to list of schemes
- *loci* [string] - URI to list of loci
- *projects* [string] - URI to list of projects

17.3.3 GET /db/{database}/classification_schemes - List classification schemes

Required route parameter: database [string] - Database configuration name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/classification_schemes

Response: Object containing:

- records [integer] - Number of classification schemes.
- classification_schemes [array] - List of *URIs to classificaton schemes*.

17.3.4 GET /db/{database}/classification_schemes/{classification_scheme_id} - Retrieve classification scheme information and groups

Required route parameters:

- database [string] - Database configuration name
- classification_scheme_id [integer] - Classification scheme id number

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/classification_schemes/1

Response: Object containing some or all of:

- id [integer] - Classification scheme id
- name [text] - Name of classification scheme
- description [text] - Description of classification scheme
- relative_threshold [boolean] - True if a *relative thresold* is used
- inclusion_threshold [integer] - The threshold for number of loci difference used to group

- groups [string] (sequence definition databases only) - URI to list of groups
  - id [integer] - group id
  - profiles [array] - list of *URIs to profiles* belonging to the group

17.3.5 GET /db/{database}/classification_schemes/{classification_scheme_id}/groups - List groups defined for a classification scheme

Sequence definition databases only.

Required route parameters:

- database [string] - Database configuration name
- classification_scheme_id [integer] - Classification scheme id number

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

  Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/classification_schemes/1/groups

  Response: Object containing of:
- records [integer] - Number of groups
- groups [array] - List of *URIs to classification group records*.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.6 GET /db/{database}/classification_schemes/{classification_scheme_id}/groups/{group_id - List isolates or profiles belonging to a classification scheme group

Required route parameters:

- database [string] - Database configuration name
- classification_scheme_id [integer] - Classification scheme id number
- group_id [integer] - Group id number

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/classification_schemes/4/groups/65

Response: Object containing some of:

- records [integer] - Number of isolates or profiles
- isolates (isolate database only) [array] - List of *URIs to isolate records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- profiles (sequence definition databases only) [array] - List of *URIs to profile records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.7 GET /db/{database}/loci - List loci

Required route parameter: database [string] - Database configuration name

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.
- alleles_added_after [date] - Include only loci with alleles added after (but not on) specified date (ISO 8601 format). Only recognized in sequence definition databases.
- alleles_updated_after [date] - Include only loci with alleles last modified after (but not on) specified date (ISO 8601 format). Only recognized in sequence definition databases.
- alleles_added_reldate [integer] - Include only loci with alleles added within the number of days specified. Only recognized in sequence definition databases.
- alleles_updated_reldate [integer] - Include only loci with alleles last modified within the number of days specified. Only recognized in sequence definition databases.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/loci

Response: Object containing:

- records [integer] - Number of loci
- loci [array] - List of *URIs to defined locus records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results

  - return_all - URI to page containing all results (paging disabled)

Note: See also the *scheme specific version*, allowing filtering by date of last allele update for just the loci that are members of a scheme.

17.3.8 GET /db/{database}/loci/{locus} - Retrieve locus record

Provides information about a locus, including links to allele sequences (in seqdef databases).

Required route parameters:

- database [string] - Database configuration name
- locus [string] - Locus name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/loci/abcZ

Response: Object containing a subset of the following key/value pairs:

- id [string] - locus name
- data_type [string] - 'DNA' or 'peptide'
- allele_id_format [string] - 'integer' or 'text'
- allele_id_regex [string] - regular expression constraining allele ids
- common_name [string]
- aliases [array] - list of alternative names of the locus
- length_varies [boolean]
- length [integer] - length if alleles are of a fixed length
- coding_sequence [boolean]
- orf [integer] - 1-6
- schemes [array] - list of scheme objects, each consisting of:
  - scheme [string] - URI to scheme information
  - description [string]
- min_length [integer] (seqdef databases) - minimum length for variable length loci
- max_length [integer] (seqdef databases) - maximum length for variable length loci
- alleles [string] (seqdef databases) - *URI to list of allele records*
- alleles_fasta [string] (seqdef databases) - *URI to FASTA file of all alleles of locus*
- curators [array] (seqdef databases) - list of *URIs to user records* of curators of the locus
- publications [array] (seqdef databases) - list of PubMed id numbers of papers describing the locus
- full_name [string] (seqdef databases)
- product [string] (seqdef databases)
- description [string] (seqdef databases)

- extended_attributes [array] (seqdef databases) - list of extended attribute objects. Each consists of a subset of the following fields:
  - field [string] - field name
  - value_format [string] - 'integer', 'text', or 'boolean'
  - value_regex [string] - regular expression constraining value
  - description [string] - description of field
  - length [integer] - maximum length of field
  - required [boolean]
  - allowed_values [array] - list of allowed values
- genome_position [integer] (isolate databases)

17.3.9 GET /db/{database}/loci/{locus}/alleles - Retrieve list of alleles defined for a locus

Required route parameters:

- database [string] - Database configuration name
- locus [string] - Locus name

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.
- include_records [integer] - Set to non-zero value to include array of allele records rather than links.
- extended [integer] - Set to non-zero value to include extended attributes if defined (only if include_records is selected).
- variation [integer] - Set to non-zero value to include defined single amino-acid variant (SAV) and/or single nucleotide variant (SNP) information if defined for the locus (only if include_records is selected).
- added_after [date] - Include only alleles added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Include only alleles added within the specified number of days.
- added_on [date] - Include only alleles added on specified date (ISO 8601 format).
- updated_after [date] - Include only alleles last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Include only alleles updated within the specified number of days.
- updated_on [date] - Include only alleles last modified on specified date (ISO 8601 format).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/loci/abcZ/alleles

Response: Object containing:

- records [integer] - Number of alleles.
- last_updated [date] - Latest allele addition/modification date (ISO 8601 format).
- alleles [array] - If include_records = 0 this is a list of *URIs to defined allele records*. If include_records = 1 then this is a list of allele objects (with values as used in *single allele records*). Pages are 100 records by default. Page size can be modified using the page_size parameter.

- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.10 GET /db/{database}/loci/{locus}/alleles_fasta - Download alleles in FASTA format

Required route parameters:

- database [string] - Database configuration name
- locus [string] - Locus name

Optional parameters:

- added_after [date] - Include only alleles added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Include only alleles added within the specified number of days.
- added_on date [date] - Include only alleles added on specified date (ISO 8601 format).
- updated_after [date] - Include only alleles last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Include only alleles last modified within the specified number of days.
- updated_on [date] - Include only alleles last modified on specified date (ISO 8601 format).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/loci/abcZ/alleles_fasta

Response: FASTA format file of allele sequences

17.3.11 GET /db/{database}/loci/{locus}/alleles/{allele_id} - Retrieve full allele information

Required route parameters:

- database [string] - Database configuration name
- locus [string] - Locus name
- allele_id [string] - Allele identifier

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/loci/abcZ/alleles/5

Response: Object containing the following key/value pairs:

- locus [string] - *URI to locus description*
- allele_id [string] - allele identifier
- sequence [string] - sequence
- status [string] - either 'Sanger trace checked', 'WGS: manual extract', 'WGS: automated extract', or 'unchecked'
- sender [string] - *URI to user details* of sender

- curator [string] - *URI to user details* of curator
- date_entered [string] - record creation date (ISO 8601 format)
- datestamp [string] - last updated date (ISO 8601 format)

17.3.12 POST /db/{database}/loci/{locus}/sequence - Query sequence to identify allele

Required route parameters:

- database [string] - Database configuration name
- locus [string] - Locus name

Required additional parameters (JSON-encoded in POST body):

- sequence [string] - Sequence string or base64-encoded FASTA file

Optional parameters (JSON-encoded in POST body):

- details [true/false] - Return detailed exact match parameters
- base64 [true/false] - Sequence is a base64-encoded FASTA file

Response: Object containing the following key/value pairs:

- exact_matches [array] - list of match objects, each consisting of:
  - allele_id
  - href - *URI to allele record*.

  additionally if 'details' parameter passed:
  - start - start position on query
  - end - end position on query
  - orientation - forward/reverse
  - length - length of matched allele
  - contig - contig name if FASTA file is uploaded

  If the locus is linked to field data in client isolate databases, there may also be an object called 'linked_data' containing values and frequencies of the field for the returned allele.
- best_match [object] - consisting of key/value pairs (if no exact matches)
  - allele_id
  - href - *URI to allele record*.
  - start - start position on query (predicted taking account of allele length)
  - end - end position on query (predicted taking account of allele length)
  - orientation - forward/reverse
  - length - length of matched allele
  - alignment - length of BLAST alignment
  - mismatches - number of mismatches
  - identity - %identity of match

  - gaps - number of gaps in alignment

17.3.13 POST /db/{database}/sequence - Query sequence to identify allele without specifying locus

Required route parameters:

- database [string] - Database configuration name

Required additional parameters (JSON-encoded in POST body):

- sequence [string] - Sequence string or base64-encoded FASTA file

Optional parameters (JSON-encoded in POST body):

- details [true/false] - Return detailed exact match parameters
- base64 [true/false] - Sequence is a base64-encoded FASTA file

Response:

- exact_matches [object] consisting of locus keys, each consisting of array of match objects consisting of:
  - allele_id
  - href - *URI to allele record*.

  additionally if 'details' parameter passed:
  - start - start position on query
  - end - end position on query
  - orientation - forward/reverse
  - length - length of matched allele
  - contig - contig name if FASTA file is uploaded

  If the locus is linked to field data in client isolate databases, there may also be an object called 'linked_data' containing values and frequencies of the field for the returned allele.

Note: This method only supports exact matches. If no match is indicated for a specific locus, use the *locus-specific call* to identify the closest match.

17.3.14 GET /db/{database}/sequences - Get summary of defined sequences

Required route parameter: database [string] - Database configuration name

Optional parameters:

- added_after [date] - Count only alleles added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Count only alleles added within the specified number of days.
- added_on [date] - Count only alleles added on specified date (ISO 8601 format).
- updated_after [date] - Count only alleles last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Count only alleles last modified within the specified number of days.
- updated_on [date] - Count only allele updated on specified date (ISO 8601 format).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/sequences

Response: Object containing a subset of the following key/value pairs:

- *loci* [string] - URI to list of loci
- records [integer] - Number of alleles defined
- last_updated [date] - Latest allele addition/modification date (ISO 8601 format).

17.3.15 GET /db/{database}/schemes - List schemes

Required route parameter: database [string] - Database configuration name

Optional parameters:

- with_pk [integer] - Set to non-zero value to only show indexed schemes, i.e. those with a primary key field that defines each unique combination of alleles, e.g. MLST.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/schemes

Response:

- records [integer] - Number of schemes
- schemes [array] - list of scheme objects, each containing:
  - scheme [string] - *URI to scheme information*
  - description [string]

17.3.16 GET /db/{database}/schemes/{scheme_id} - Retrieve scheme information

Includes links to allelic profiles (in seqdef databases, if appropriate). **Required route parameters:**

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id number

Optional parameters:

- added_after [date] - Count only profiles added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Count only profiles added within the specified number of days.
- added_on [date] - Count only profiles added on specified date (ISO 8601 format).
- updated_after [date] - Count only profiles last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Count only profiles last modified within the specified number of days.
- updated_on [date] - Count only profiles updated on specified date (ISO 8601 format).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/schemes/1

Response: Object containing a subset of the following key/value pairs:

- id [integer]
- description [string]
- locus_count [integer] - number of loci belonging to scheme
- loci [array] - list of *URIs to locus descriptions*
- has_primary_key_field [boolean]

- fields [array] - list of *URIs to scheme field descriptions*
- primary_key_field [string] - *URI to primary key field description*
- profiles [string] - URI to list of profile definitions (only seqdef databases)
- profiles_csv [string] - URI to tab-delimited file of all scheme profiles
- curators [array] (seqdef databases) - list of *URIs to user records* of curators of the scheme
- records [integer] - Number of profiles
- last_added [date] - Latest profile addition/modification date (ISO 8601 format).
- last_updated [date] - Latest profile addition/modification date (ISO 8601 format).

17.3.17 GET /db/{database}/schemes/{scheme_id}/loci - Retrieve scheme loci

Required route parameters:

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id number

Optional parameters:

- alleles_added_after [date] - Include only loci with alleles added after (but not on) specified date (ISO 8601 format). Only recognized in sequence definition databases.
- alleles_added_reldate [integer] - Include only loci with alleles added within the specified number of days. Only recognized in sequence definition databases.
- alleles_updated_after [date] - Include only loci with alleles last modified after specified date (ISO 8601 format). Only recognized in sequence definition databases.
- alleles_updated_reldate [integer] - Include only loci with alleles last modified within the specified number of days. Only recognized in sequence definition databases.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/schemes/1/loci

Response: Object containing:

- records [integer] - Number of loci
- loci [array] - List of *URIs to defined locus records*.

17.3.18 GET /db/{database}/schemes/{scheme_id}/fields/{field} - Retrieve information about scheme field

Required route parameters:

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id number
- field [string] - Field name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/schemes/1/fields/ST

Response: Object containing the following key/value pairs:

- field [string] - field name

- type [string] - data type of field (integer or text)
- primary_key [boolean] - true if field is the scheme primary key

17.3.19 GET /db/{database}/schemes/{scheme_id}/profiles - List allelic profiles defined for scheme

Required route parameters:

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.
- added_after [date] - Include only profiles added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Include only profiles added within the specified number of days.
- added_on [date] - Include only profiles added on specified date (ISO 8601 format).
- updated_after [date] - Include only profiles last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Include only profiles last modified within the specified number of days.
- updated_on [date] - Include only profiles last modified on specified date (ISO 8601 format).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/schemes/1/profiles

Response: Object containing:

- records [integer] - Number of profiles
- last_updated [date] - Latest profile addition/modification date (ISO 8601 format).
- profiles [array] - List of *URIs to defined profile records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

Note: This method also supports content negotiation. If the request accepts header includes TSV or CSV, then the call is redirected to */db/{database}/schemes/{scheme_id}/profiles_csv*.

17.3.20 GET /db/{database}/schemes/{scheme_id}/profiles_csv - Download allelic profiles in CSV (tab-delimited) format

Required route parameters:

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id

Optional parameters:

- added_after [date] - Include only profiles added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Include only profiles added within the specified number of days.
- added_on [date] - Include only profiles added on specified date (ISO 8601 format).
- updated_after [date] - Include only profiles last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Include only profiles last modified within the specified number of days.
- updated_on [date] - Include only profiles last modified on specified date (ISO 8601 format).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/schemes/1/profiles_csv

Response: Tab-delimited text file of allelic profiles

17.3.21 GET /db/{database}/schemes/{scheme_id}/profiles/{profile_id} - Retrieve allelic profile record

Required route parameters:

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id
- profile_id [string/integer] - Profile id

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/schemes/1/profiles/11

Response: Object containing the following key/value pairs:

- *primary_key_term* [string/integer] - The field name is the primary key, e.g. ST. The value is the primary key value (primary_id used as an argument).
- alleles [object] - *list of URIs to allele descriptions*
- *other_scheme_fields* [string/integer] - Each scheme field will have its own value if defined. The field name is the name of the field.
- sender [string] - *URI to user details* of sender
- curator [string] - *URI to user details* of curator
- date_entered [string] - record creation date (ISO 8601 format)
- datestamp [string] - last updated date (ISO 8601 format)

17.3.22 POST /db/{database}/schemes/{scheme_id}/sequence - Query sequence to extract allele designations/fields for a scheme

Required route parameters:

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id

Required additional parameters (JSON-encoded in POST body):

- sequence [string] - Sequence string or base64-encoded FASTA file

Optional parameters (JSON-encoded in POST body):

- details [true/false] - Return detailed exact match parameters
- partial_matches [true/false] - Return details of partial matches if exact match is not found
- base64 [true/false] - Sequence is a base64-encoded FASTA file

Response: Object containing the following key/value pairs:

- exact_matches [array] - list of match objects, each consisting of:
  - allele_id
  - href - *URI to allele record*.

  additionally if 'details' parameter passed:
  - start - start position on query
  - end - end position on query
  - orientation - forward/reverse
  - length - length of matched allele
  - contig - contig name if FASTA file is uploaded

  If the locus is linked to field data in client isolate databases, there may also be an object called 'linked_data' containing values and frequencies of the field for the returned allele.

Example curl call to upload a FASTA file 'contigs.fasta' and extract MLST results from Neisseria database:

```
(echo -n '{"base64":true,"sequence": "'; base64 contigs.fasta; echo '"}') |
curl -s -H "Content-Type: application/json" -X POST "https://rest.pubmlst.org/db/pubmlst_
↪neisseria_seqdef/schemes/1/sequence" -d @-
```

Note: This method only supports exact matches. If no match is indicated for a specific locus, use the *locus-specific call* to identify the closest match.

17.3.23 POST /db/{database}/schemes/{scheme_id}/designations - Query allelic profile to extract fields for a scheme

Required route parameters:

- database [string] - Database configuration name
- scheme_id [integer] - Scheme id

Required additional parameters (JSON-encoded in POST body):

- designations [object] - consisting of
  - locus objects each containing an array of alleles (see example)

Response: Object containing the following key/value pairs:

- exact_matches [object] - consisting of locus values, each consisting of an array of allele values:
  - allele_id [string]

  If the locus is linked to field data in client isolate databases, there may also be an object called 'linked_data' containing values and frequencies of the field for the returned allele.
- fields [object] - consisting of key/value pairs of scheme fields (if defined)

Example curl call to query an allelic profile and extract MLST results from Neisseria database:

```
curl -s -H "Content-Type: application/json" -X POST "https://rest.pubmlst.org/db/pubmlst_
↪neisseria_seqdef/schemes/1/designations" -d '{"designations":{"abcZ":[{"allele":"2"}],
↪"adk":[{"allele":"3"}],"aroE":[{"allele":"4"}],"fumC":[{"allele":"3"}],"gdh":[{"allele
↪":"8"}],"pdhC":[{"allele":"4"}],"pgm":[{"allele":"6"}]}}'
```

17.3.24 GET /db/{database}/isolates - Retrieve list of isolate records

Required route parameter: database [string] - Database configuration name

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.
- added_after [date] - Include only isolates added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Include only isolates added within the specified number of days.
- added_on [date] - Include only isolates added on specified date (ISO 8601 format).
- include_old_versions [integer] - Set to 1 to include old record versions (the default is to only include new versions)
- updated_after [date] - Include only isolates last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Include only isolates last modified within the specified number of days.
- updated_on [date] - Include only isolates updated on specified date (ISO 8601 format).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates

Response: Object containing:

- records [integer] - Number of isolates

- isolates [array] - List of *URIs to isolate records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.25 GET /db/{database}/isolates/{isolate_id} - Retrieve isolate record

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier

Optional parameter:

- provenance_only [integer] - Set to non-zero value to only return provenance metadata

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1

Response: Object containing some or all of the following key/value pairs:

- provenance [object] - set of key/value pairs. Keys are defined by calling the */fields route* route. The fields will vary by database but will always contain the following:
  - id [integer]
  - sender [string] - *URI to user details* of sender
  - curator [string] - *URI to user details* of curator
  - date_entered [string] - record creation date (ISO 8601 format)
  - datestamp [string] - last updated date (ISO 8601 format)
- aliases [array] - list of alternative names for isolate
- publications [array] (seqdef databases) - list of PubMed id numbers of papers that refer to the isolate
- sequence_bin [object] - consists of the following key/value pairs:
  - contigs_fasta [string] - *URI to FASTA file containing all the contigs belonging to this isolate*
  - contigs [string] - *URI to list of contig records*
  - contig_count [integer] - number of contigs
  - total_length [integer] - total length of contigs
- allele_designations [object] - consists of the following key/value pairs:
  - allele_ids - *URI to list of all allele_id values* defined for the isolate
  - designation_count - number of allele designations defined for the isolate
  - full_designations - *URI to list of full allele designation records*
- schemes [array] - list of scheme objects, each containing some of the following:
  - description [string] - description of scheme

  - loci_designated_count [integer] - number of loci within scheme that have an allele designated for this isolate.
  - allele_ids [string] - *URI to list of all allele_id values defined for this scheme* for this isolate
  - full_designations [string] - *URI to list of full allele designation records* for this isolate
  - fields [object] - consisting of key/value pairs where the key is the name of each scheme field
  - classification_schemes [object] - consisting of key/value pairs, where each key is the name of the classification scheme and the value is an object consisting of:
    * href [string] - *URI to classification scheme description*
    * groups [array] - list of group objects consisting of:
      · group [integer] - group id
      · records [integer] - number of isolates in group
      · isolates [string] - URI to classification group record containing URIs to member isolate records
- projects [array] - list of project objects, each containing the following:
  - id [string] - *URI to project information*
  - description [string] - description of project
- history [string] - *URI to isolate history record*
- new_version [string] - URI to newer version of record
- old_version [string] - URI to older version of record

17.3.26 GET /db/{database}/isolates/{isolate_id}/allele_designations - Retrieve list of allele designation records

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/allele_designations

Response: Object containing:

- records [integer] - Number of allele designations
- allele_designations [array] - List of *URIs to allele designation records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results

  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.27 GET /db/{database}/isolates/{isolate_id}/allele_designations/{locus} - Retrieve full allele designation record

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier
- locus [string] - Locus name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/allele_designations/BACT000065

Response: List of allele_designation objects (there may be multiple designations for the same locus), each containing:

- locus [string] - *URI to locus description*
- allele_id [string]
- method [string] - either 'manual' or 'automatic'
- status [string] - either 'confirmed' or 'provisional'
- comments [string]
- sender [string] - *URI to user details* of sender
- curator [string] - *URI to user details* of curator
- datestamp [string] - last updated date (ISO 8601 format)

17.3.28 GET /db/{database}/isolates/{isolate_id}/allele_ids - Retrieve allele identifiers

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/allele_ids

Response: Object containing:

- records [integer] - Number of allele id objects
- allele_ids [array] - List of allele id objects, each consisting of a key/value pair where the key is the locus name. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:

- previous - URI to previous page of results
- next - URI to next page of results
- first - URI to first page of results
- last - URI to last page of results
- return_all - URI to page containing all results (paging disabled)

17.3.29 GET /db/{database}/isolates/{isolate_id}/schemes/{scheme_id}/allele_designations - Retrieve scheme allele designation records

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier
- scheme_id [integer] - Scheme identifier

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/schemes/1/allele_designations

Response:

- records [integer] - Number of allele designation objects
- allele_designations [array] - List of *allele designation objects* for each locus in the specified scheme that has been designated.

17.3.30 GET /db/{database}/isolates/{isolate_id}/schemes/{scheme_id}/allele_ids - Retrieve list of scheme allele identifiers

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier
- scheme_id [integer] - Scheme identifier

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/schemes/1/allele_ids

Response:

- records [integer] - Number of allele id objects
- allele_ids [array] - List containing allele id objects for each locus in the specified scheme that has been designated. Each allele_id object contains a key which is the name of the locus with a value that may be either a string, integer or array of strings or integers (required where there are multiple designations for a locus). The data type depends on the allele_id_format set for the specific locus.

17.3.31 GET /db/{database}/isolates/{isolate_id}/contigs - Retrieve list of contigs

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/contigs

Response: Object containing:

- records [integer] - Number of contigs
- contigs [array] - List of *URIs to contig records* Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.32 GET /db/{database}/isolates/{isolate_id}/contigs_fasta - Download contigs in FASTA format

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier

Optional parameter:

- header [string] - either 'original_designation' or 'id' (default is 'id'). This selects whether the FASTA header lines contain the originally uploaded FASTA headers or the sequence bin id numbers.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/contigs_fasta?header=original_designation

Response: FASTA format file of isolate contig sequences

17.3.33 GET /db/{database}/isolates/{isolate_id}/history - Retrieve isolate update history

Required route parameters:

- database [string] - Database configuration name
- isolate_id [integer] - Isolate identifier

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/1/history

Response: Object containing:

- records [integer] - Number of updayes
- contigs [array] - List of update objects each consisting of the following key/value pairs:
  - curator [string] - *URI to user details* of curator
  - timestamp [string] - Time of update
  - actions [array] - List of update descriptions [strings]

17.3.34 GET /db/{database}/genomes - Retrieve list of isolate records that have genome assemblies

Required route parameter: database [string] - Database configuration name

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.
- added_after [date] - Include only isolates added after (but not on) specified date (ISO 8601 format).
- added_reldate [integer] - Include only isolates added within the specified number of days.
- added_on [date] - Include only isolates added on specified date (ISO 8601 format).
- include_old_versions [integer] - Set to 1 to include old record versions (the default is to only include new versions)
- updated_after [date] - Include only isolates last modified after (but not on) specified date (ISO 8601 format).
- updated_reldate [integer] - Include only isolates last modified within the specified number of days.
- updated_on [date] - Include only isolates updated on specified date (ISO 8601 format).
- genome_size [integer] - Filter to only include records with a sequence bin of at least the specified size (default is 500,000bp).

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/genomes

Response: Object containing:

- records [integer] - Number of isolates
- isolates [array] - List of *URIs to isolate records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results

  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.35 POST /db/{database}/isolates/search - Search isolate database

Required route parameters:

- database [string] - Database configuration name

Optional parameters (appended to URI):

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Query parameters (JSON-encoded in POST body):

You must include at least one query parameter.

Parameter names in the following forms are supported:

- field.{field} - key/value pairs for provenance fields. Supported field names can be found by calling the */fields route*. The fields will vary by database.
- locus.{locus} - key/value pairs of locus and its allele designation. Supported locus names can be found by calling the */loci route*.
- scheme.{scheme_id}.{scheme_field} - key/value pairs of scheme fields and their values. Supported field names can be determined by following routes from the */schemes route*.

Example method call using curl: The following searches for *Neisseria* ST-11 isolates from Europe in 2015 (MLST is scheme#1 in this database).

```
curl -s -H "Content-Type: application/json" -X POST \
"https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/isolates/search" \
-d '{"field.continent":"europe","field.year":2015,"scheme.1.ST":11}'
```

Response: Object containing:

- records [integer] - Number of isolates
- isolates [array] - List of *URIs to isolate records*. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.36 GET /db/{database}/contigs/{contig_id} - Retrieve contig record

Required route parameters:

- database [string] - Database configuration name
- contig_id [integer] - Contig identifier

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/contigs/180062

Response: Contig object consisting of the following key/value pairs:

- id [integer] - contig identifier
- isolate_id [integer] - isolate identifier
- sequence [string] - contig sequence
- length [integer] - length of contig sequence
- method [string] - sequencing method
- sender [string] - *URI to user details* of sender
- curator [string] - *URI to user details* of curator
- date_entered [string] - record creation date (ISO 8601 format)
- datestamp [string] - last updated date (ISO 8601 format)
- loci [array] - list of sequence tag objects consisting of:
  - locus [string] - *URI to locus description*
  - locus_name [string]
  - start [integer]
  - end [integer]
  - direction [string] - forward/reverse
  - complete [boolean] - true/false

17.3.37 GET /db/{database}/fields - Retrieve list of isolate provenance field descriptions

Required route parameters:

- database [string] - Database configuration name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/fields

Response: Array of field objects, each consisting of some or all of the following key/value pairs:

- name [string] - name of field
- type [string] - data type (int, text, date, float)
- length [integer] - maximum length of field
- required [boolean] - true if field value is required
- min [integer] - minimum value for integer values

- max [integer] - maximum value for integer values
- regex [string] - regular expression that constrains the allowed value of the field
- comments [string]
- allowed values [array] - list of allowed values for the field
- values [string] - URI to list of used field values

17.3.38 GET /db/{database}/fields/{field} - Retrieve values set for a provenance field

Required route parameters:

- database [string] - Database configuration name
- field [string] - Provenance metadata field name

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/fields/country

Response: Object containing:

- records [integer] - Number of values
- values [array] - List of values used in isolate records. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.39 GET /db/{database}/users/{user_id} - Retrieve user information

Users may be data submitters or curators.

Required route parameters:

- database [string] - Database configuration name
- user_id [integer] - User id number

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/users/2

Response: Object containing the following key/value pairs:

- id [integer] - user id number
- first_name [string]

- surname [string]
- affiliation [string] - institutional affiliation
- email [string] - E-mail address (may be hidden depending on server configuration)

17.3.40 GET /db/{database}/curators - Retrieve list of curators

Required route parameters:

- database [string] - Database configuration name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/curators

Response: Object containing:

- records [integer] - Number of curators
- curators [array] - List of *URIs to user records*.

17.3.41 GET /db/{database}/projects - Retrieve list of projects

Required route parameter: database [string] - Database configuration name

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/projects

Response:

- projects [array] - List of project objects, each containing:
  - project [string] - *URI to project information*
  - description [string]
  - isolate_count [integer] - number of isolates in project

17.3.42 GET /db/{database}/projects/{project_id} - Retrieve project information

Required route parameters:

- database [string] - Database configuration name
- project_id [integer] - Project id number

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/projects/3

Response: Object containing a subset of the following key/value pairs:

- id [integer]
- description [string]
- isolates [string] - *URI to list of URIs of member isolate records*.

17.3.43 GET /db/{database}/projects/{project_id}/isolates - Retrieve list of isolates belonging to a project

Required route parameter:

- database [string] - Database configuration name
- project_id [integer] - Project id number

Optional parameters:

- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/projects/3/isolates

Response: Object containing:

- records [integer] - Number of isolates in the project
- isolates [array] - List of URIs to isolate records. Pages are 100 records by default. Page size can be modified using the page_size parameter.
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results
  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.44 GET /db/{database}/submissions - retrieve list of submissions

Required route parameter: database [string] - Database configuration name

Optional parameters:

- type [string] - either 'alleles', 'profiles' or 'isolates'
- status [string] - either 'closed' or 'pending'
- page [integer]
- page_size [integer]
- return_all [integer] - Set to non-zero value to disable paging.

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_isolates/submissions

Response: Object containing:

- records [integer] - Number of submissions
- submissions [array] - List of *URIs to submission records*
- paging [object] - Some or all of the following:
  - previous - URI to previous page of results
  - next - URI to next page of results

  - first - URI to first page of results
  - last - URI to last page of results
  - return_all - URI to page containing all results (paging disabled)

17.3.45 POST /db/{database}/submissions - create new submission

Required route parameter: database [string] - Database configuration name

Required additional parameters (JSON-encoded in POST body):

- type [string] - either:
  - alleles (sequence definition databases only)
  - profiles (sequence definition databases only)
  - isolates (isolate databases only)
  - genomes (isolate databases only)

  The following are required with the specified database type:

  Allele submissions

- locus [string] - name of locus
- technology [string] - name of sequencing technology: either '454', 'Illumina', 'Ion Torrent', 'PacBio', 'Oxford Nanopore', 'Sanger', 'Solexa', 'SOLiD', or 'other'
- read_length [string] - read length of sequencing: either '<100', '100-199', '200-299', '300-499', '>500', or any positive integer (only required for Illumina)
- coverage [string] - mean coverage of sequencing: either '<20x', '20-49x', '50-99x', '>100x', or any positive integer (only required for Illumina)
- assembly [string] - assembly method: either 'de novo' or 'mapped'
- software [string] - name of assembly software
- sequences [string] - either single raw sequence or multiple sequences in FASTA format

  Profile submissions

- scheme_id [integer] - scheme id number
- profiles [string] - tab-delimited profile data - this should include a header line containing the name of each locus

  Isolate submissions

- isolates [string] - tab-delimited isolate data - this should include a header line containing each field or locus included

  Genome submissions

- isolates [string] - tab-delimited isolate data - this should include a header line containing each field or locus included as well as for 'assembly_filename' and 'sequence_method'. The 'sequence_method' should be either '454', 'Illumina', 'Ion Torrent', 'PacBio', 'Oxford Nanopore', 'Sanger', 'Solexa', 'SOLiD', or 'other'. Following submission, contig files should be uploaded with the same names as set for 'assembly_filename'. This can be done using the *file upload route*.

Optional parameters:

- message [string] - correspondence to the curator
- email [integer] - set to 1 to enable E-mail updates (E-mails will be sent to the registered user account address).

Response: Object containing:

- submission - *URI to submission record*

  For genome submissions, the response object will also contain:

- missing_files [array] - List of filenames that need to be uploaded to complete the submission. These filenames are defined in the 'assembly_filename' field of the isolate record upload. The files should contain the contig assemblies.
- message [string] - 'Please upload missing contig files to complete submission.'

17.3.46 GET /db/{database}/submissions/{submission_id} - Retrieve submission record

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/submissions/BIGSdb_20151013081836_14559_14740

Response: Object containing some of the following:

- id [string] - Submission id
- type [string] - Either 'alleles', 'profiles', 'isolates'
- date_submitted [string] - Submission date (ISO 8601 format)
- datestamp [string] - Last updated date (ISO 8601 format)
- submitter [string] - *URI to user details* of submitter
- curator [string] - *URI to user details* of curator
- status [string] - either 'started', 'pending', or 'closed'
- outcome [string] - either 'good' (data uploaded), 'bad' (data rejected), or 'mixed' (parts of submission accepted)
- correspondence [array] - List of correspondence objects in time order. Each contains:
  - user [string] *URI to user details* of user
  - timestamp [string]
  - message [string]

  Allele submissions

- locus [string] - name of locus
- technology [string] - name of sequencing technology: either '454', 'Illumina', 'Ion Torrent', 'PacBio', 'Oxford Nanopore', 'Sanger', 'Solexa', 'SOLiD', or 'other'
- read_length [string] - read length of sequencing: either '<100', '100-199', '200-299', '300-499', '>500', or any positive integer (only required for Illumina)
- coverage [string] - mean coverage of sequencing: either '<20x', '20-49x', '50-99x', '>100x', or any positive integer (only required for Illumina)
- assembly [string] - assembly method: either 'de novo' or 'mapped'

- software [string] - name of assembly software
- seqs [array] - List of sequence objects each containing:
  - seq_id [string] - Sequence identifier
  - assigned_id [string] - Allele identifier if uploaded to the database (otherwise undefined)
  - status [string] - Either 'pending', 'assigned', or 'rejected'
  - sequence [string]

Profile submissions

- scheme [string] - *URI to scheme information*
- profiles [array] - List of profile record objects. Each contains:
  - profile_id [string] - Record identifier
  - assigned_id [string] - Profile identifier if uploaded to the database (otherwise undefined)
  - status [string] - Either 'pending', 'assigned', or 'rejected'
  - designations [object] containing key/value pairs for each locus containing the allele identifier

Isolate submissions

- isolates [array] - List of isolate record objects. Each contains key/value pairs for included fields.

17.3.47 DELETE /db/{database}/submissions/{submission_id} - Delete submission record

You must be the owner and the record must be closed.

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/submissions/BIGSdb_20151013081836_14559_14740

Response: message [string] - 'Submission deleted.'

17.3.48 GET /db/{database}/submissions/{submission_id}/messages - Retrieve submission correspondence

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/submissions/BIGSdb_20151013081836_14559_14740/messages

Response: Array of correspondence objects in time order. Each contains:

- user [string] *URI to user details* of user

- timestamp [string]
- message [string]

17.3.49 POST /db/{database}/submissions/{submission_id}/messages - Add submission correspondence

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id

Required additional parameter (JSON-encoded in POST body):

- message [string] - Message text

Optional parameters: None

Response: message [string] - 'Message added.'

17.3.50 GET /db/{database}/submissions/{submission_id}/files - Retrieve list of supporting files uploaded for submission

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id

Optional parameters: None

Example request URI: https://rest.pubmlst.org/db/pubmlst_neisseria_seqdef/submissions/BIGSdb_20151013081836_14559_14740/files

Response: Array of URIs to files

17.3.51 POST /db/{database}/submissions/{submission_id}/files - Upload submission supporting file

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id

Required additional parameters (JSON-encoded in POST body):

- filename [string] - Name of file to store within submission
- upload [base64 encoded data] - Raw file data

Optional parameters: None

Response: message [string] - 'File uploaded.'

17.3.52 GET /db/{database}/submissions/{submission_id}/files/{filename} - Download submission supporting file

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id
- filename [string] - Name of file

Optional parameters: None

Response: File download

17.3.53 DELETE /db/{database}/submissions/{submission_id}/files/{filename} - Delete submission supporting file

Required route parameters:

- database [string] - Database configuration name
- submission_id [string] - Submission id
- filename [string] - Name of file

Optional parameters: None

Response: message [string] - 'File deleted.'

17.4 Authentication

Protected resources, i.e. those requiring a user to log in, can be accessed via the API using OAuth (1.0A) authentication (see IETF RFC5849 for details). Third-party client software has to be registered with the BIGSdb site before they can access authenticated resources. The overall three-legged flow works as follows:

1. *Developer signs up* and gets a consumer key and consumer secret specific to their application.
2. Application *gets a request token* and directs user to authorization page on BIGSdb.
3. BIGSdb *asks user for authorization* for application to access specific resource using their credentials. A verifier code is provided.
4. The application exchanges the request token and OAuth verifier code for an *access token and secret* (these do not expire but may be revoked by the user or site admin).
5. Application uses access token/secret to *request session token* (this is valid for 12 hours).
6. All calls to *access protected resources* are signed using the session token/secret and consumer key/secret.

It is recommended that application developers use an OAuth library to generate and sign requests.

Note: There are Python and Perl example scripts available at https://github.com/kjolley/BIGSdb/tree/develop/scripts/rest_examples to demonstrate and test OAuth authentication.

17.4.1 Developer sign up to get a consumer key

Application developers should apply to the site administrator of the site running BIGSdb. The administrator can *generate a key and secret* using a script - both of these will need to be used by the application to sign requests.

The client id is usually a 24 character alphanumeric string. The secret is usually a 42 character alphanumeric (including punctuation) string, e.g.

- **client_id:** efKXmqp2D0EBlMBkZaGC2lPf
- **client_secret:** F$M+fQ2AFFB2YBDfF9fpHF^qSWJdmmN%L4Fxf5Gur3

17.4.2 Getting a request token

- **Relative URL:** /db/{database}/oauth/get_request_token
- **Supported method:** GET

The application uses the consumer key to obtain a request token. The request token is a temporary token used to initiate user authorization for the application and will expire in 60 minutes. The request needs to contain the following parameters and to be signed using the consumer secret:

- oauth_consumer_key
- oauth_request_method ('GET')
- oauth_request_url (request URL)
- oauth_signature_method ('HMAC-SHA1')
- oauth_signature
- oauth_timestamp (UNIX timestamp - seconds since Jan 1 1970) - this must be within 600 seconds of the current time.
- oauth_callback ('oob' for desktop applications)
- oauth_nonce (random string)
- oauth_version ('1.0')

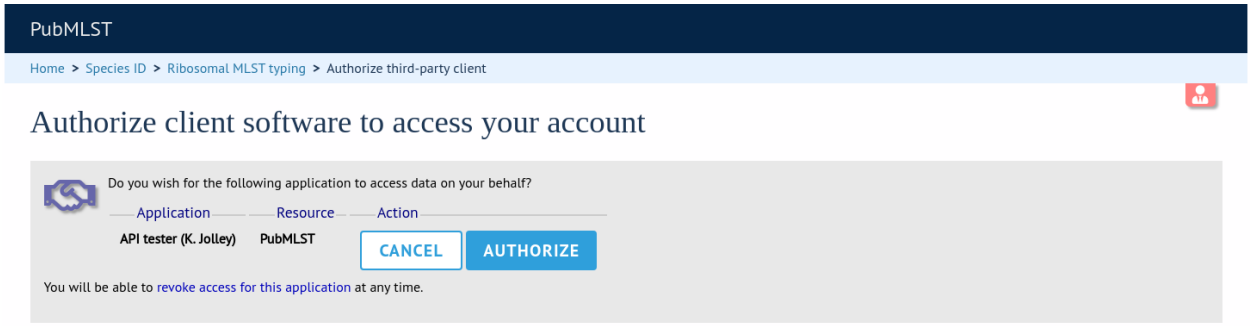
If the application has been registered and has been granted permission to access the specific resource, a JSON response will be returned containing the following parameters:

- **oauth_token**
  - This is the request token. It is usually a 32 character alphanumeric string.
  - e.g. fKFm0WNhCfbEX8zQm6qhDA8K23FOWDGE
- **oauth_token_secret**
  - This is the secret associated with the request token. It is usually a 32 character alphanumeric string.
  - e.g. aZ0fncP7i5w5jlebdK5zyQ4vrRRVcdnv
- **oauth_callback_confirmed**
  - This parameter is always set to true.

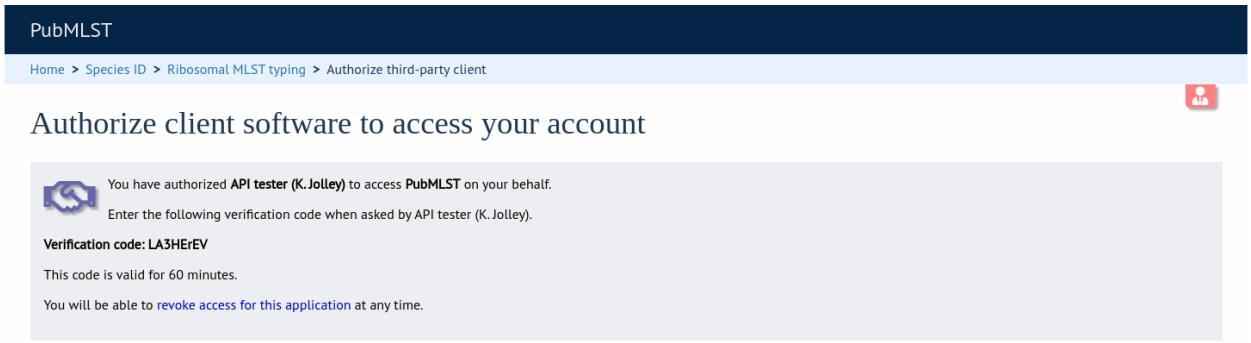
17.4.3 Getting user authorization

Once a request token has been obtained, this can be used by the end user to grant permission to access a specific resource to the application. The application should direct the user to the client authorization page (authorizeClient) specific to a database within BIGSdb, e.g. http://pubmlst.org/cgi-bin/bigsdb/bigsdb.pl?db=pubmlst_neisseria_seqdef&page=authorizeClient&oauth_token=fKFm0WNhCfbEX8zQm6qhDA8K23FOWDGE

The user will be asked if they wish to grant access to the application on their behalf:



If they authorize the access, they will be presented with a verifier code. This should be entered in to the client application which will use this together with the request token to request an access token.



The verifier code is valid for 60 minutes.

17.4.4 Getting an access token

- **Relative URL:** /db/{database}/oauth/get_access_token
- **Supported method:** GET

The application uses the request token, verifier code and its consumer key to obtain an access token. The access token does not expire but can be revoked by either the end user or the site administrator. The request needs to contain the following parameters and to be signed using the consumer secret and request token secret:

- oauth_consumer_key
- oauth_request_method ('GET')
- oauth_request_url (request URL)
- oauth_signature_method ('HMAC-SHA1')
- oauth_signature
- oauth_token (request token)

- oauth_timestamp (UNIX timestamp - seconds since Jan 1 1970) - this must be within 600 seconds of the current time.
- oauth_nonce (random string)
- oauth_version ('1.0')

If the application has been registered and has been granted permission to access the specific resource, a JSON response will be returned containing the following parameters:

- **oauth_token**
  - This is the access token. It is usually a 32 character alphanumeric string.
  - e.g. SDrC74ZVl5SYSqY8lWZqrRxnyDnNGVFO
- **oauth_token_secret**
  - This is the secret associated with the access token. It is usually a 32 character alphanumeric string.
  - e.g. tYI2SPzgiO02IRVzW4JR1ez6Vvm4gVyv

17.4.5 Getting a session token

- **Relative URL:** /db/{database}/oauth/get_session_token
- **Supported method:** GET

The application uses the access token and its consumer key to obtain a session token. The session token is valid for 12 hours before it expires. The request needs to contain the following parameters and to be signed using the consumer secret and access token secret:

- oauth_consumer_key
- oauth_request_method ('GET')
- oauth_request_url (request URL)
- oauth_signature_method ('HMAC-SHA1')
- oauth_signature
- oauth_token (access token)
- oauth_timestamp (UNIX timestamp - seconds since Jan 1 1970) - this must be within 600 seconds of the current time.
- oauth_nonce (random string)
- oauth_version ('1.0')

If the application has been registered and has been granted permission to access the specific resource, a JSON response will be returned containing the following parameters:

- **oauth_token**
  - This is the session token. It is usually a 32 character alphanumeric string.
  - e.g. H8CjIS8Ikq6hwCUqUfF1l4pTaCYl8Ljw
- **oauth_token_secret**
  - This is the secret associated with the session token. It is usually a 32 character alphanumeric string.
  - e.g. RfponbaNPO7tkZ2miHFISk0pMndePNfJ

17.4.6 Accessing protected resources

The application uses the session token and its consumer key to access a protected resource. The request needs to contain the following parameters and to be signed using the consumer secret and session token secret:

- oauth_consumer_key
- oauth_request_method ('GET')
- oauth_request_url (request URL)
- oauth_signature_method ('HMAC-SHA1')
- oauth_signature
- oauth_token (session token)
- oauth_timestamp (UNIX timestamp - seconds since Jan 1 1970) - this must be within 600 seconds of the current time.
- oauth_nonce (random string)
- oauth_version ('1.0')

CHAPTER

EIGHTEEN

FREQUENTLY ASKED QUESTIONS (FAQS)

18.1 General

1. **What is the minimum specification of hardware required to run BIGSdb?**

   The software will run on fairly modest hardware. For an installation with only local users, the following minimum is recommended:

   - 4 processor cores
   - 16 GB RAM
   - 50 GB partition for temporary files
   - 100 GB partition for databases

   As usual, the more RAM that is available the better. Ideally you would want enough RAM that the whole database(s) can reside in memory (an approximation is roughly twice the total size of your contigs), although this is not absolutely required.

   Offline jobs, such as *Genome Comparator* will use multiple cores (depending on the settings in bigsdb.conf), so if you want to run multiple jobs in parallel then you may want more cores (and memory). Tagging of new genomes using the offline *autotagger* is usually run in multi-threaded mode so the more cores available the faster this will be.

   As a comparison, the PubMLST site is run on two machines - separate web and database servers. All offline jobs and tagging of genomes is performed on the database server. These have the following specification:

   - web server: 40 cores, 128GB RAM
   - database server: 80 cores, 1TB RAM, 7TB ZFS RAID-Z2 NVMe local storage

2. **Why might icons be missing when using Internet Explorer?**

   This can occur if you have Compatibility Mode enabled. BIGSdb generates valid HTML5 and Compatibility Mode should not be used. Please ensure this is not enabled in the Internet Explorer tools section.

18.2 Installation

1. **BIGSdb is accumulating files in various temp directories - is this normal and how do I clean them out?**

   See: *Periodically delete temporary files*.

2. **BIGSdb is complaining of an invalid script path - what does this mean?**

   In your database config.xml file system tag are two attributes - script_path_includes and curate_path_includes. These contain regexes that the web url to your script (bigsdb.pl and bigscurate.pl respectively) must match. This prevents somebody from accessing a private database using an instance of bigsdb.pl that is not in a protected directory if you're using apache authentication.

   So, if you access the script from http://localhost/cgi-bin/bigsdb/bigsdb.pl then you can set script_path_includes to something like "/bigsdb/" (which is the default), or "/cgi-bin/" or just "/" if you don't care about this check.

18.3 Administration

1. **How can I make some isolates public but not others?**

   The easiest way to do this is to set up two or more separate configuration directories that refer to the database. The URLs to access these will differ by the value of the 'db' attribute, which refers to the name of the configuration directory (in /etc/bigsdb/dbases/). The database view accessed by each of these configurations can be different as can the access restrictions.

   Example:

   We have a database 'bigsdb_test' that contains data, only some of which we wish to make publicly available. The isolates to make public are all members of a project. First we can make a view of the isolates table that contains only isolates within this project.

   For isolates in project id 3, create a database view by logging in to psql as the postgresql user. We will name this view 'public'.:

```
sudo su postgres
psql bigsdb_test

CREATE VIEW public AS SELECT * FROM isolates WHERE id IN (SELECT isolate_id
  FROM project_members WHERE project_id=3);
GRANT SELECT ON public TO apache;
```

   Create a private configuration that can access everything in the database in /etc/bigsdb/dbases/test_private. This will be accessible from http://IP_ADDRESS/cgi-bin/bigsdb/bigsdb.pl?db=test_private.

   The important attributes to set in the system tag of the config.xml file in this directory are::

```
view="isolates"
read_access="authenticated_users"
```

   This means that anyone with an account can log in and view all the isolates (because the view is set to the isolates table).

   Now create a public configuration in /etc/bigsdb/dbases/test_public. This will be accessible from http://IP_ADDRESS/cgi-bin/bigsdb/bigsdb.pl?db=test_public. It is better to create a symlink to the private config.xml and then override the attributes that are different. So create a symlink to the private config file:

```
cd /etc/bigsdb/dbases/test_public
sudo ln -s ../test_private/config.xml .
```

You can now override the view and access settings. Within /etc/bigsdb/dbases/test_public, create a file called system.overrides and add the following:

```
view="public"
read_access="public"
```

See also *Restricting particular configurations to specific user accounts* and *private records*.

CHAPTER

NINETEEN

APPENDIX

19.1 Query operators

Various query forms have operators for use with field values. Available operators are:

- =
  - Exact match (case-insensitive).
- contains
  - Match to a partial string (case-insensitive), e.g. searching for clonal complex 'contains' st-11 would return all STs belonging to the ST-11 complex.
- starts with
  - Match to values that start with the search term (case-insensitive).
- ends with
  - Match to values that end with the search term (case-sensitive).
- >
  - Greater than the search term.
- >=
  - Greater than or equal the search term.
- <
  - Less than the search term.
- <=
  - Less than or equal the search term.
- NOT
  - Match to values that do not equal the search term (case-insensitive).
- NOT contain
  - Match to values that do not contain the search term (case-insensitive).

19.2 Sequence tag flags

Sequences tagged in the sequence bin can have features indicated by specific flags. The presence of these flags can be queried. These are a superset of *flags available for allele sequences*. Available flags are:

- alternative start codon
  - A start codon other than ATG, GTG, or TTG is used. This can be the case with some yeasts.
- ambiguous read
  - Genome sequence contains ambiguous nucleotides in coding sequence.
- apparent misassembly
  - Sequence has a region of very high identity to existing allele in one region but looks completely different in another.
- atypical
  - Catch-all term for a sequence that is unusual compared to other alleles of locus.
- contains IS element
  - Coding sequence is interrupted by insertion sequence.
- downstream fusion
  - No stop codon present resulting in translation continuing.
- frameshift
  - Frameshift in sequence relative to other alleles, not resulting in internal stop codon.
- indel
  - Insertion/deletion in sequence that is uncommon compared to other alleles.
- internal stop codon
  - Frameshift in sequence relative to other alleles, resulting in internal stop codon.
- no start codon
  - No apparent start codon in immediate vicinity of usual start.
- no stop codon
  - No stop codon in immediate vicinity of usual stop.
- phase variable: off
  - Coding sequence has a homopolymeric run with a frameshift resulting in a stop codon preventing complete translation.
- truncated
  - Coding sequence is unusually short resulting in a truncated protein (not the same as running off the end of a contig).
- upstream fusion
  - No apparent start codon in immediate vicinity of usual start, likely due to a gene fusion (sequence is transcribed together with upstream coding sequence).

19.3 Allele sequence flags

Sequences can be flagged with specific attributes - these are searchable when doing a sequence attribute query. These are a subset of *flags available for tagged sequences*. These are mainly for use with whole genome MLST type data. Multiple flags can be selected by Ctrl-clicking the list. Available flags are:

- alternative start codon
  - A start codon other than ATG, GTG, or TTG is used. This can be the case with some yeasts.
- atypical
  - Catch-all term for a sequence that is unusual compared to other alleles of locus.
- contains IS element
  - Coding sequence is interrupted by insertion sequence.
- downstream fusion
  - No stop codon present resulting in translation continuing.
- frameshift
  - Frameshift in sequence relative to other alleles, not resulting in internal stop codon.
- indel
  - Insertion/deletion in sequence that is uncommon compared to other alleles.
- internal stop codon
  - Frameshift in sequence relative to other alleles, resulting in internal stop codon.
- no start codon
  - No apparent start codon in immediate vicinity of usual start.
- no stop codon
  - No stop codon in immediate vicinity of usual stop.
- phase variable: off
  - Coding sequence has a homopolymeric run with a frameshift resulting in a stop codon preventing complete translation.
- truncated
  - Coding sequence is unusually short resulting in a truncated protein (not the same as running off the end of a contig).
- upstream fusion
  - No apparent start codon in immediate vicinity of usual start, likely due to a gene fusion (sequence is transcribed together with upstream coding sequence).

CHAPTER

TWENTY

DATABASE SCHEMA

- Sequence definition database
- Isolate database

INDEX

A

access
    control lists, 60
    restricting, 61
adding
    classification groups, 98
    isolates, 150
    locus, 69, 71, 72, 77, 103
    MLST scheme, 96
    schemes, 82
allele definition
    records, 227
allele designations
    count, 264
    query, 263
    status, 266
allele sequence
    identify, 233
allele sequences
    adding, 128
alleles, 3
    list query, 243
    retiring, 137
    un-retiring, 138
allelic profiles
    isolates, 275
amino acid variants
    locus, 80
annotation status
    query, 267
API authentication
    access token, 461
    accessing protected resources, 462
    consumer key, 459
    request token, 460
    session token, 462
    user authorization, 460
API resources
    add submission correspondence, 458
    create new submission, 455
    DELETE /db/{database}/submissions/{submission_id}, 457
    DELETE /db/{database}/submissions/{submission_id}/file, 459
    delete submission record, 457
    delete submission supporting file, 459
    download alleles in FASTA format, 435
    download allelic profiles in CSV (tab-delimited) format, 440
    download contigs in FASTA format, 448
    download submission supporting file, 458
    GET /, 429
    GET /db, 429
    GET /db/{database}, 430
    GET /db/{database}/classification_schemes, 430
    GET /db/{database}/classification_schemes/{classificat 430
    GET /db/{database}/classification_schemes/{classificat 431
    GET /db/{database}/classification_schemes/{classificat 431
    GET /db/{database}/contigs/{contig_id}, 450
    GET /db/{database}/curators, 453
    GET /db/{database}/fields, 451
    GET /db/{database}/fields/{field}, 452
    GET /db/{database}/genomes, 449
    GET /db/{database}/isolates, 443
    GET /db/{database}/isolates/{isolate_id}, 444
    GET /db/{database}/isolates/{isolate_id}/allele_design 445
    GET /db/{database}/isolates/{isolate_id}/allele_design 446
    GET /db/{database}/isolates/{isolate_id}/allele_ids, 446
    GET /db/{database}/isolates/{isolate_id}/contigs, 447
    GET /db/{database}/isolates/{isolate_id}/contigs_fasta 448
    GET /db/{database}/isolates/{isolate_id}/history, 448
    GET /db/{database}/isolates/{isolate_id}/schemes/{sche

447
GET /db/{database}/isolates/{isolate_id}/schemes, 447
GET /db/{database}/loci, 432
GET /db/{database}/loci/{locus}, 433
GET /db/{database}/loci/{locus}/alleles, 434
GET /db/{database}/loci/{locus}/alleles/{allele_id}, 435
GET /db/{database}/loci/{locus}/alleles_fasta, 435
GET /db/{database}/projects, 453
GET /db/{database}/projects/{project_id}, 453
GET /db/{database}/projects/{project_id}/isolates, 453
GET /db/{database}/schemes, 438
GET /db/{database}/schemes/{scheme_id}, 438
GET /db/{database}/schemes/{scheme_id}/fields/{field}, 439
GET /db/{database}/schemes/{scheme_id}/loci, 439
GET /db/{database}/schemes/{scheme_id}/profiles, 440
GET /db/{database}/schemes/{scheme_id}/profiles/{profile_id}, 441
GET /db/{database}/schemes/{scheme_id}/profiles_csv, 440
GET /db/{database}/sequences, 437
GET /db/{database}/submissions, 454
GET /db/{database}/submissions/{submission_id}, 456
GET /db/{database}/submissions/{submission_id}/files, 458
GET /db/{database}/submissions/{submission_id}/files/{filename}, 458
GET /db/{database}/submissions/{submission_id}/messages, 457
GET /db/{database}/users/{user_id}, 452
get summary of defined sequences, 437
list allelic profiles defined for scheme, 440
list classification schemes, 430
list database resources, 430
list loci, 432
list schemes, 438
list site resources, 429
POST /db/{database}/isolates/search, 450
POST /db/{database}/loci/{locus}/sequence, 436
POST /db/{database}/schemes/{scheme_id}/designations, 442
POST /db/{database}/schemes/{scheme_id}/sequence, 441
POST /db/{database}/sequences, 437
POST /db/{database}/submissions, 455
POST /db/{database}/submissions/{submission_id}/files, 458
POST /db/{database}/submissions/{submission_id}/messages, 458
query allele sequence, 436
query allele sequence without specifying locus, 437
query scheme designations, 442
query scheme sequences, 441
retrieve allele identifiers, 446
retrieve classification scheme information and groups, 430, 431
retrieve contig record, 450
retrieve full allele designation record, 446
retrieve full allele information, 435
retrieve information about scheme field, 439
retrieve isolate record, 444
retrieve isolate update history, 448
retrieve list of allele designations, 445
retrieve list of alleles defined for a locus, 434
retrieve list of contigs, 447
retrieve list of curators, 453
retrieve list of groups for a classification scheme, 431
retrieve list of isolate provenance field descriptions, 451
retrieve list of isolate records, 443
retrieve list of isolate records that have genome assemblies, 449
retrieve list of isolates belonging to a project, 453
retrieve list of projects, 453
retrieve list of scheme allele identifiers, 447
retrieve list of submissions, 454
retrieve list of supporting files uploaded for submission, 458
retrieve locus record, 433
retrieve project information, 453
retrieve scheme allele designation records, 447
retrieve scheme information, 438
retrieve scheme loci, 439
retrieve specific allelic profile record, 441
retrieve submission correspondence, 457
retrieve submission record, 456
retrieve user information, 452

retrieve values set for a provenance field, 452
search isolate database, 450
upload submission supporting file, 458
assembly stats, 211
auto allele definer, 209
stop, 213
automated assignment
scheme profiles, 97
autotagger, 205
stop, 213

B

batch profile definitions
query, 253
batch uploading contigs - multiple isolates, 168
BLAST, 309
BLAST caches
refreshing, 124
bookmarks, 278
breakdown
provenance field, 322
sequence bin, 383
two-field, 386
browse
isolates, 259
scheme profiles, 244
BURST, 315

C

caching
schemes, 64
classification groups, 4
adding, 98
client authorization
RESTful interface, 123
client databases, 91
clustering
core genome, 101
codon usage, 319
Locus Explorer, 364
composite fields, 113
config-specific
downloads, 125
configuration
export, 121
configuration settings
validation, 120
core genome
clustering, 101
count
allele designations, 264
sequence tags, 269

D

database
logging, 20
defining
exemplar alleles, 208
deleting
isolates, 155, 158
profiles, 146
downloads
config-specific, 125

E

exemplar alleles
defining, 208
export
configuration, 121
extended attributes
locus, 78
provenance fields, 116

F

filters, 274

G

gene presence;, 332
genetic cluster characterization
ReporTree, 379
Genome Comparator, 338
genome filtering, 108
in silico hybridization, 110
in silico PCR, 108
genome report
Reports, 382
geographic coordinates, 183
geographic point lookup values
populating, 215
GrapeTree
minimum-spanning trees, 346
groups
scheme, 88

H

hosts
mapping, 63

I

identify
allele sequence, 233
sequence type, 235
in silico hybridization, 110
in silico PCR, 108
in silico PCR analysis, 350
isolate

records, 223
isolate aliases, 162
isolate record
options, 288
isolates
adding, 150
allelic profiles, 275
browse, 259
deleting, 155, 158
query, 260
retiring, 159
un-retiring, 160
updating, 154, 157
uploading contigs, 165
iTOL
phylogenetic trees, 353

K

kiosk mode, 45
Kleborate, 358

L

LINcode schemes, 104
linking remote contigs, 171
list query
alleles, 243
loci, 3
modifying display, 292
locus
adding, 69, 71, 72, 77, 103
adding; copying existing record, 76
amino acid variants, 80
extended attributes, 78
SNPs, 80
Locus Explorer, 361
codon usage, 364
polymorphic sites, 362
translated sequences, 365
locus positions
setting, 112
log files
rotation, 16
logging
database, 20

M

main results table
options, 286
mapping
hosts, 63
Microreact
spatio-temporaral trees, 366
minimum-spanning trees
GrapeTree, 346
PhyloViz online, 369
MLST, 3
MLST scheme
adding, 96
mod_perl, 64
modifying display
loci, 292
schemes, 292

O

offline curation
auto allele definer, 209
autotagger, 205
options, 284
isolate record, 288
main results table, 286
provenance fields, 290
query, 291

P

partitioning
sets, 65
passwords
setting, 61
setting; first user, 61
performance
caching schemes, 64
mod_perl, 64
permissions, 57
locus curation, 59
scheme curation, 59
phylogenetic trees
iTOL, 353
PhyloViz online
minimum-spanning trees, 369
plugins
enabling, 62
polymorphic sites
Locus Explorer, 362
polymorphisms, 372
populating
geographic point lookup values, 215
profile
records, 229
profiles, 4
deleting, 146
retiring, 147
un-retiring, 148
updating, 145
projects, 176
provenance field
breakdown, 322
provenance fields
extended attributes, 116

options, 290
publications, 163, 281

Q

query
allele designations, 263
batch profile definitions, 253
isolates, 260
options, 291
scheme profiles, 246
sequence tags, 271
ST definitions from allelic profiles, 250

R

records
allele definition, 227
isolate, 223
profile, 229
sequence bin, 230
sequence tag, 228
refreshing
BLAST caches, 124
ReporTree
genetic cluster characterization, 379
Reports
genome report, 382
RESTful interface
client authorization, 123
retiring
alleles, 137
isolates, 159
profiles, 147
rMLST
species identification, 375
storing analysis results, 212
rotation
log files, 16

S

scheme
groups, 88
scheme profiles
automated assignment, 97
browse, 244
query, 246
schemes, 3
adding, 82
caching, 64
modifying display, 292
sequence bin
breakdown, 383
query, 268
records, 230
sequence similarity
determining, 256
sequence tag
records, 228
sequence tags, 4
count, 269
query, 271
sequence type
identify, 235
sets, 4
partitioning, 65
SNPs
locus, 80
sparsely-populated fields, 42
spatio-temporaral trees
Microreact, 366
species identification
rMLST, 375
ST definitions from allelic profiles
query, 250
stop
auto allele definer, 213
autotagger, 213
storing analysis results
rMLST, 212

T

translated sequences
Locus Explorer, 365
two-field
breakdown, 386

U

un-retiring
alleles, 138
isolates, 160
profiles, 148
unique combinations, 389
updates
disabling, 62
updating
isolates, 154, 157
profiles, 145
uploading contigs
isolates, 165
user groups, 57
user projects, 296
user types, 57
users
adding, 127