
scgpm-seqresults-dnanexus **Documentation**

Release 0.4.0

Nathaniel Watson

Jul 31, 2019

Contents

1	Scripts	3
2	Client API Modules	11
3	Indices and tables	17
	Python Module Index	19
	Index	21

This project is read from [GitHub](#)

1.1 accept_pending_transfers.py

Given all pending DNAnexus project transfers for the user, allows the user to accept transfers under a specified billing account, but not just any project. Only projects that have the specified queue (set in the queue property of the project) will be transferred to the user.

```
usage: accept_pending_transfers.py [-h] --access-level
                                {VIEW,UPLOAD,CONTRIBUTE,ADMINISTER} -q
                                QUEUE -o ORG
                                [--share-with-org {VIEW,UPLOAD,CONTRIBUTE,
                                ↪ADMINISTER}]
```

1.1.1 Named Arguments

--access-level	<p>Possible choices: VIEW, UPLOAD, CONTRIBUTE, ADMINISTER</p> <p>Permissions level the new member should have on transferred projects.</p>
-q, --queue	<p>The value of the queue property on a DNAnexus project. Only projects that are pending transfer that have this value for the queue property will be transferred to the specified org.</p>
-o, --org	<p>The name of the DNAnexus org under which to accept the project transfers for projects that have their queue property set to the value of the 'queue' argument.</p>
--share-with-org	<p>Possible choices: VIEW, UPLOAD, CONTRIBUTE, ADMINISTER</p> <p>Add this argument if you'd like to share the transferred projects with the org so that all users of the org will have access to the project. The value you supply should be the access level that members of the org will have.</p>

1.2 add_props_to_fq_files.py

Adds properties to FASTQ files in DNAnexus. Make sure you are logged into DNAnexus with the appropriate account before using this script in order to ensure access to the relevant projects. Accepts a tab-delimited input file containing properties to add to barcoded FASTQ files, and there must be a header line containing the two fields 'dx_project_id' and 'barcode'. The remaining columns will be treated as property names. It is assumed that the directory containing the FASTQ files in each project is named /stage0_bcl2fastq/fastqs. Note that there are some standard property fields in use. For example, in the CIRM stem-cell project, the following property names are in use:

- 1) lab_patient_id (i.e. some identifier for a give patient/sample)
- 2) lab_patient_group (i.e. A categorization for the patient)
- 3) lab_patient_condition (i.e. a treatment condition, perhaps a medicine and dosage)

The properties that represent FASTQ metadata for some lab protocol/setup should start with 'lab' followed by an underscore.

```
usage: add_props_to_fq_files.py [-h] -i INFILE
```

1.2.1 Named Arguments

-i, --infile Tab-delimited input file.

1.3 add_props_to_project.py

Given a DNAnexus project ID, or a file containing multiple DNAnexus project IDs (one per line), adds one or more properties to each project. Properties are specified as positional key=value arguments.

```
usage: add_props_to_project.py [-h]
                             (--dx-project-id DX_PROJECT_ID | --infile INFILE)
                             props [props ...]
```

1.3.1 Positional Arguments

props **One or more key=value positional arguments, each** representing a property key and value pair.

1.3.2 Named Arguments

--dx-project-id The DNAnexus project ID to add properties too.

--infile File of DNAnexus project IDs, one per line. Empty lines and lines starting with '#' will be skipped.

1.4 add_r1-r2_fastq_paths.py

Retrieves the FASTQ file names in DNAnexus for the specified sequenced libraries. The tab-delimited input file may be provided in one of two formats.

Format 1:

- 1) DNAnexus project name
- 2) barcode

Format 2

- 1) uhts run name
- 2) lane
- 3) barcode,

The format in use is determined by the number of header fields present in the header line, which must appear as the very first line in the input file and begin with a '#'.

The output file is identical to the input file, with the exception of two new columns at the start of the file being the FASTQ file name on the DNAnexus platform, and the read number. Thus, the output columns are:

- 1) FASTQ file name
- 2) Read number (1 for forward reads, 2 for reverse reads)

followed by the input file columns. Note that at present, one of three warnings may be output to stdout. The possible warnings are triggered whenever

- A DNAnexus project isn't found based on the provided criteria.
- A DNAnexus project was found, but there were not any FASTQ files found within having the specified barcodes.
- A DNAnexus project was found, but only a forward reads or reverse reads FASTQ file was found, not both.

The last warning thus implies that the script assumes all reads are paired-end, which is true.

```
usage: add_r1-r2_fastq_paths.py [-h] -i INFILE -o OUTFILE
```

1.4.1 Named Arguments

-i, --infile

Tab-delimited input file in one of two formats. In each format, the first line must be a header line starting with a '#'. Empty lines and lines beginning with '#' are ignored. The first format contains only two columns with the 1st containing the DNAnexus project name, and the second the barcode. The second format contains the three columns uhts_run name, lane, and barcode. The number of columns present in the header line determines the format - two fields for the first format, and three fields for the latter. A field-header line starting with '#' is required as the first line.

-o, --outfile

1.5 add_standard_props_to_projects.py

To fill in.

```
usage: add_standard_props_to_projects.py [-h] -b BILLING_ACCOUNT
```

1.5.1 Named Arguments

-b, --billing-account The name of the DNAnexus billing account that the project belongs to. This will only be used to restrict the search of projects.

1.6 batch_download_fastqs.py

Calls download_fastqs.py in batch, provided an input file specifying the FASTQs to download. This script passes a log file name to download_fastqs.py for error logging, i.e. if a DNAnexus project isn't found then it will be logged. The log file is named after this script and contains the number of seconds since the epoch to help generate a unique name.

```
usage: batch_download_fastqs.py [-h] -i INFILE -d FILE_DOWNLOAD_DIR
                                [--not-found-error]
```

1.6.1 Named Arguments

-i, --infile Tab-delimited input file in one of two formats. Empty lines and lines beginning with a '#' will be skipped. The first line must be a header line. The first format is used if you don't know the DNAnexus project. Format 1 has the following fields:

1. uhts run name,
2. sequencing lane,
3. library name,
4. barcode.

Format 2 has the following fields:

1. dnanexus_project_name,
2. barcode

The script will act on format 1 parsing rules if 4 Fields are detected in the header line, and those of the second format if two fields are detected in the header line. Any other number of fields found in the header line will result in an error.

A note on format 1, you don't have to include values for each field. For unknown values, just leave it blank. These values are stored as properties on a DNAnexus project, and the search for a DNAnexus project will be successful if you supply enough property information to uniquely identify a project.

-d, --file-download-dir Local directory in which to download the FASTQ files.

--not-found-error Presence of this options means to raise an Exception if a project can't be found on DNAnexus with the provided input.

Default: False

1.7 download_fastqs.py

To fill in.

```
usage: download_fastqs.py [-h] --errlog ERRLOG [-l LIBRARY_NAME]
                        [--uhts-run-name UHTS_RUN_NAME]
                        [--dx-project-name DX_PROJECT_NAME] [--lane LANE]
                        [-b BARCODE [BARCODE ...]] -d FILE_DOWNLOAD_DIR
                        [--not-found-error]
```

1.7.1 Named Arguments

--errlog	Log file name to write errors to (i.e. When a DNAnexus project isn't found). Will be opened in append mode.
-l, --library-name	The library name of the sample that was sequenced. This is name of the library that was submitted to SCGPM for sequencing. This is added as a property to all sequencing result projects through the 'library_name' project property.
--uhts-run-name	The name of the sequencing run in UHTS. This is added as a property to all projects in DNAnexus through the 'seq_run_name' project property. Use this option in combination with --library-name and --lane to further restrict the search space, which is useful especially since multiple DNAnexus projects can have the same library_name property value (i.e. if resequencing the same library).
--dx-project-name	The name of the sequencing run project in DNAnexus.
--lane	The lane number of the flowcell on which the library was sequenced. This is added as a property to all projects in DNAnexus through the 'seq_lane_index' property. Use this in conjunction with --library-name and --uhts-run-name to further restrict the project search space.
-b, --barcode	The barcode of the sequenced sample. If specified, then only FASTQs with these barcodes will be downloaded. Otherwise, all FASTQs will be downloaded.
-d, --file-download-dir	Local directory in which to download the FASTQ files.
--not-found-error	Presence of this options means to raise an Exception if a project can't be found on DNAnexus with the provided input.

Default: False

1.8 download_multiple_projects.py

Downloads all projects from DNAnexus that have the given value for the 'seq_run_name' property set. A billing account can be specified in order to limit the project search space to only those belonging to that account.

```
usage: download_multiple_projects.py [-h] --seq-run-name SEQ_RUN_NAME
                                     --download-dir DOWNLOAD_DIR
                                     [-b BILLING_ACCOUNT_ID]
                                     [-s SKIP_PROJECTS [SKIP_PROJECTS ...]]
```

1.8.1 Named Arguments

--seq-run-name	The name of the sequencing run, as set by the 'seq_run_name' property of a DNAnexus project.
--download-dir	Local directory in which to download the DNAnexus project.
-b, --billing-account-id	A DNAnexus user account or org to restrict the project search in DNAnexus.
-s, --skip-projects	One ore more DNAnexus project IDs (space delimited) to skip downloading. Useful if you need to download multiple projects that have the same value for the 'seq_run_name' property, but not all if, for example some were already downloaded.

1.9 download_project.py

Downloads a SCGPM sequencing results project from DNAnexus. Currently, there is one DNAnexus project for each lane of Illumina sequencing. A folder will be created by the name of the DNAnexus project within the specified output folder. The project folder will contain a FASTQ subfolder, a FASTQC subfolder, and several files including 1) the sequencing QC report, 2) the list of barcodes in the sequencing lane in barcodes.json, run information in run_details.json, and a meta-data tarball by the name of \${sequencing_run_name}.metadata.tar. This last file contains important XML files output by the sequencer, such as the runParameters.XML and RunInfo.xml.

```
usage: download_project.py [-h] --download-dir DOWNLOAD_DIR
                          (--dx-project-id DX_PROJECT_ID | --dx-project-list DX_
↳PROJECT_LIST)
```

1.9.1 Named Arguments

--download-dir	Local directory in which to download the DNAnexus project.
--dx-project-id	The DNAnexus ID of the project to download.
--dx-project-list	File with DNAnexus project IDs, one per line, for batch project download. Empty lines and lines starting with a '#' will be ignored.

1.10 list_projects.py

Writes a tab-delimited file containg information about all projects in DNAnexus that are billed to the specified org. The field names are:

Name, ID, library_name, seq_run_name, lab, paired_end, sequencer_type, seq_instrument, seq_lane_index, queue, experiment_type

```
usage: list_projects.py [-h] -o OUTFILE --org ORG
```

1.10.1 Named Arguments

-o, --outfile	The output file.
--org	The DNAnexus org in which the projects belong.

1.11 scgpm_clean_raw_data.py

This script calls the DNAnexus app I built called SCGPM Clean Raw Data at https://platform.dnanexus.com/app/scgpm_clean_raw_data to unwanted files (that drive up the storage costs) from the raw_data folder of a DNAnexus project containing sequencing results from the SCGPM sequencing workflow. Most of the files in the raw_data folder are removed. Moreover, the lane tarball is removed; the XML files RunInfo.xml and runParameters.xml are extracted from Interop.tar and then the tarball is removed; finally, metadata.tar is removed. The extracted XML files are uploaded back to the raw_data folder.

Queries DNAnexus for all projects billed to the specified org and that were created within the last -d days.

You must have the environment variable DX_SECURITY_CONTEXT set (described at <http://autodoc.dnanexus.com/bindings/python/current/dxpy.html?highlight=token>) in order to authenticate with DNAnexus.

```
usage: scgpm_clean_raw_data.py [-h] [-d DAYS_AGO] -o ORG
```

1.11.1 Named Arguments

- | | |
|-----------------------|--|
| -d, --days-ago | The number of days ago to query for new projects that are billed to the org specified by -o. |
| | Default: 30 |
| -o, --org | Limits the project search to only those that belong to the specified DNAnexus org. Should begin with 'org-'. |

2.1 scgpm_seqresults_dnanexus

Utilities for working with the SCGPM Sequencing Center application logic on DNAnexus.

`scgpm_seqresults_dnanexus.LOG_DIR = 'Logs_scgpm_seqresults_dnanexus'`

The directory that contains log files.

`scgpm_seqresults_dnanexus.debug_logger = <Logger scgpm_seqresults_dnanexus (DEBUG)>`

A logging.Logger instance that logs all messages sent to it to STDOUT, as well as a log file in the folder specified by LOG_DIR. In addition, another file handler is created that accepts messages at the level logging.ERROR, and lives in the same folder.

2.2 dnanexus_utils

exception `scgpm_seqresults_dnanexus.dnanexus_utils.DxMissingAlignmentSummaryMetrics`
Bases: Exception

Raised by `DxSeqResults.get_alignment_summary_metrics()` when it can't locate a Picard alignment summary metrics file for a given barcoded sample of FASTQ sequencing results.

exception `scgpm_seqresults_dnanexus.dnanexus_utils.DxMissingLibraryNameProperty`
Bases: Exception

Raised when creating a new `DxSeqResults` instance for a DNAnexus project that doesn't have the `library_name` project property present.

exception `scgpm_seqresults_dnanexus.dnanexus_utils.DxProjectMissingQueueProperty`
Bases: Exception

exception `scgpm_seqresults_dnanexus.dnanexus_utils.DxMultipleProjectsWithSameLibraryName`
Bases: Exception

exception `scgpm_seqresults_dnanexus.dnanexus_utils.FastqNotFound`
Bases: Exception

exception `scgpm_seqresults_dnanexus.dnanexus_utils.DnanexusBarcodeNotFound`
 Bases: `Exception`

`scgpm_seqresults_dnanexus.dnanexus_utils.select_newest_project(dx_project_ids)`
 Given a list of DNAnexus project IDs, returns the one that is newest as determined by creation date.

Parameters `dx_project_ids` – list of DNAnexus project IDs.

Returns *str*.

`scgpm_seqresults_dnanexus.dnanexus_utils.accept_project_transfers(access_level, queue, org, share_with_org=None)`

Parameters

- **access_level** – *str*. Permissions level the new member should have on transferred projects. Should be one of ["VIEW","UPLOAD","CONTRIBUTE","ADMINISTER"]. See <https://wiki.dnanexus.com/API-Specification-v1.0.0/Project-Permissions-and-Sharing> for more details on access levels.
- **queue** – *str*. The value of the queue property on a DNAnexus project. Only projects that are pending transfer that have this value for the queue property will be transferred to the specified org.
- **org** – *str*. The name of the DNAnexus org under which to accept the project transfers for projects that have their queue property set to the value of the 'queue' argument.
- **share_with_org** – *str*. Set this argument if you'd like to share the transferred projects with the org so that all users of the org will have access to the project. The value you supply should be the access level that members of the org will have.

Returns The projects that were transferred to the specified billing account. Keys are the project IDs, and values are the project names.

Return type *dict*

`scgpm_seqresults_dnanexus.dnanexus_utils.find_org_projects_by_name_glob(org, glob)`

Parameters `glob` – *str*.

Ex:

Find the project(s) with SREQ-163 at the end of the project's name:

`find_org_projects_by_name_glob(org="org-someorg", glob="*SREQ-163")`

`scgpm_seqresults_dnanexus.dnanexus_utils.share_with_org(project_ids, org, access_level, suppress_email_notification=False)`

Shares one or more DNAnexus projects with an organization. It appears that DNAnexus requires for the user that wants to share the org to first have ADMINISTER access on the project. Only then could he share the project with the org.

Parameters

- **project_ids** – *list*. One or more DNAnexus project identifiers, where each project ID is in the form "project-FXq6B809p5jKzp2vJkjkKvg3".
- **org** – *str*. The name of the DNAnexus org with which to share the projects.
- **access_level** – The permission level to give to members of the org - one of ["VIEW","UPLOAD","CONTRIBUTE","ADMINISTER"].

- **suppress_email_notification** – *bool*. True means to allow the DNAnexus platform to send an email notification for each shared project.

```
class scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults(dx_project_id=False,
                                                            dx_project_name=False,
                                                            uhts_run_name=False,
                                                            sequencing_lane=False, library_name=False,
                                                            billing_account_id=None, latest_project=False)
```

Bases: `object`

Finds the DNAnexus sequencing results project that was uploaded by GSSC. The project can be precisely retrieved if the project ID is specified (via the `dx_project_id` argument). Otherwise, you can supply the `dx_project_name` argument if you know the name, or use the `library_name` argument if you know the name of the library that was submitted to GSSC. All sequencing result projects uploaded to DNAnexus by GSSC contain a property named 'library_name', and projects will be searched on this property for a matching library name when the `library_name` argument is specified. If both the `library_name` and the `dx_project_name` arguments are specified, only the latter is used in finding a project match. The `billing_account` argument can optionally be specified to restrict all project searches to only those that are billed to that particular billing account (unless `dx_project_id` is specified in which case the DNAnexus project is directly retrieved).

Parameters

- – **str. The ID of the DNAnexus project** (`dx_project_id`) – will be performed as it will be directly retrieved.
- – **str. Name of a DNAnexus project containing sequencing results that were** (`dx_project_name`) – uploaded by GSSC.
- – **str. Name of the sequencing run in UHTS. This is added as a property to** (`uhts_run_name`) – all projects in DNAnexus through the 'seq_run_name' property.
- – **int. Lane number of the flowcell on which the library was sequenced.** (`sequencing_lane`) – This is in a property named `seq_lane_index` on all GSSC projects in DNAnexus.
- – **str. Library name of the sample that was sequenced. This is the name of** (`library_name`) – the library that was submitted to GSSC for sequencing, and is added as a property to all GSSC DNAnexus projects via the 'library_name' property.
- – **str. Name of the DNAnexus billing account that the project belongs to.** (`billing_account_id`) – This will only be used to restrict the search of projects that the user can see to only those billed by the specified account.
- – **bool. True indicates that if multiple projects are found given the search** (`latest_project`) – criteria, the most recently created project will be returned.

FQEXT = `'.fastq.gz'`

The extension used for FASTQ files.

get_run_details_json()

Retrieves the JSON object for the stats in the file named `run_details.json` in the project specified by `self.dx_project_id`.

Returns JSON object of the run details.

get_alignment_summary_metrics (*barcode*)

Parses the metrics in a `${barcode}alignment_summary_metrics` file in the DNAnexus project (usually in the qc folder). This contains metrics produced by Picard Tools's `CollectAlignmentSummaryMetrics` program.

get_barcode_stats (*barcode*)

Loads the JSON in a `${barcode}_stats.json` file in the DNAnexus project (usually in the qc folder).

get_sample_stats_json (*barcode=None*)

Deprecated since version 0.1.0: GSSC has removed the `sample_stats.json` file since the entire folder it was in has been removed. Use `get_barcode_stats()` instead.

Retrieves the JSON object for the stats in the file named `sample_stats.json` in the project specified by `self.dx_project_id`. This file is located in the DNAnexus folder `staged_qc_report`.

Parameters **barcode** – *str*. The barcode for the sample. Currently, the `sample_stats.json` file is of the following form when there isn't a genome mapping:

```
[{"Sample name": "AGTTCC"}, {"Sample name": "CAGATC"}, {"Sample name": "GC-CAAT"}, ... ]
```

When there is a mapping, each dictionary has many more keys in addition to the "Sample name" one.

Returns *list* of dicts if `barcode=None`, otherwise a dict for the given barcode.

download_metadata_tar (*download_dir*)

Downloads the `${run_name}.metadata.tar` file from the DNAnexus sequencing results project.

Parameters **download_dir** – *str* - The local directory path to download the QC report to.

Returns The filepath to the downloaded metadata tarball.

Return type *str*

download_run_details_json (*download_dir*)

Downloads the `run_details.json` and the `barcodes.json` from the DNAnexus sequencing results project.

Parameters **download_dir** – *str* - The local directory path to download the QC report to.

Returns *str*. The filepath to the downloaded `run_details.json` file.

download_barcodes_json (*download_dir*)

Downloads the `run_details.json` and the `barcodes.json` from the DNAnexus sequencing results project.

Parameters **download_dir** – *str* - The local directory path to download the QC report to.

Returns *str*. The filepath to the downloaded `barcodes.json` file.

download_samplesheet (*download_dir*)

Downloads the `SampleSheet` used in demultiplexing from the DNAnexus sequencing results project.

Parameters **download_dir** – *str* - The local directory path to download the QC report to.

Returns *str*. The filepath to the downloaded QC report.

download_qc_report (*download_dir*)

Downloads the QC report from the DNAnexus sequencing results project.

Parameters **download_dir** – *str* - The local directory path to download the QC report to.

Returns *str*. The filepath to the downloaded QC report.

download_fastqc_reports (*download_dir*)

Downloads the QC report from the DNAnexus sequencing results project.

Parameters `download_dir` – *str* - The local directory path to download the QC report to.

Returns *str*. The filepath to the downloaded FASTQC reports folder.

download_fastqs (*dest_dir*, *barcode*, *overwrite=False*)

Downloads all FASTQ files in the project that match the specified barcode, or if a barcode isn't given, all FASTQ files as in this case it is assumed that this is not a multiplexed experiment. Files are downloaded to the directory specified by *dest_dir*.

Parameters

- **barcode** – *str*. The barcode sequence used.
- **dest_dir** – *str*. The local directory in which the FASTQs will be downloaded.
- **overwrite** – *bool*. If True, then if the file to download already exists in *dest_dir*, the file will be downloaded again, overwriting it. If False, the file will not be downloaded again from DNAnexus.

Returns

The key is the barcode, and the value is a dict with integer keys of 1 for the forward reads file, and 2 for the reverse reads file. If not paired-end,

Return type *dict*

Raises *Exception* – The barcode is specified and less than or greater than 2 FASTQ files are found.

get_fastq_dxfile_objects (*barcode=None*)

Retrieves all the FASTQ files in project *self.dx_project_name* as DXFile objects.

Parameters **barcode** – *str*. If set, then only FASTQ file properties for FASTQ files having the specified barcode are returned.

Returns *list* of DXFile objects representing FASTQ files.

Raises *dnanexus_utils.FastqNotFound* – No FASTQ files were found.

revcomp_barcode_in_fastqfile_prop (*i7=False*, *i5=False*)

Use this method if you need to update the barcode sequence stored as the value of the barcode property of a FASTQ file on DNAnexus.

Parameters

- **i7** – *bool*. True means to reverse complement the i7 barcode.
- **i5** – *bool*. True means to reverse complement the i5 barcode.

revcomp (*seq*)

Returns The reverse complement of a DNA sequence.

Parameters **seq** – *str*.

Returns *str*.

get_fastq_files_props (*barcode=None*)

Returns the DNAnexus file properties for all FASTQ files in the project that match the specified barcode, or all FASTQ files if not barcode is specified.

Parameters **barcode** – *str*. If set, then only FASTQ file properties for FASTQ files having the specified barcode are returned.

Returns *dict*. Keys are the FASTQ file DXFile objects; values are the dict of associated properties on DNAnexus on the file. In addition to the properties on the file in DNAnexus, an additional property is added here called 'fastq_file_name'.

Raises `dnanexus_utils.FastqNotFound` exception if no FASTQ files were found.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`scgpm_seqresults_dnanexus`, [11](#)
`scgpm_seqresults_dnanexus.dnanexus_utils`,
[11](#)

A

`accept_project_transfers()` (in module `scgpm_seqresults_dnanexus.dnanexus_utils`), 12

D

`debug_logger` (in module `scgpm_seqresults_dnanexus`), 11

`DnanexusBarcodeNotFound`, 12

`download_barcodes_json()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`download_fastqc_reports()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`download_fastqs()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 15

`download_metadata_tar()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`download_qc_report()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`download_run_details_json()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`download_samplesheet()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`DxMissingAlignmentSummaryMetrics`, 11

`DxMissingLibraryNameProperty`, 11

`DxMultipleProjectsWithSameLibraryName`, 11

`DxProjectMissingQueueProperty`, 11

`DxSeqResults` (class in `scgpm_seqresults_dnanexus.dnanexus_utils`), 13

F

`FastqNotFound`, 11

`find_org_projects_by_name_glob()` (in module `scgpm_seqresults_dnanexus.dnanexus_utils`), 12

`FQEXT` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` attribute), 13

G

`get_alignment_summary_metrics()`

(`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`get_barcode_stats()`

(`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

`get_fastq_dxfile_objects()`

(`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 15

`get_fastq_files_props()`

(`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 15

`get_run_details_json()`

(`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 13

`get_sample_stats_json()`

(`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 14

L

`LOG_DIR` (in module `scgpm_seqresults_dnanexus`), 11

R

`revcomp()` (`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 15

`revcomp_barcode_in_fastqfile_prop()`

(`scgpm_seqresults_dnanexus.dnanexus_utils.DxSeqResults` (method), 15

S

`scgpm_seqresults_dnanexus` (module), 11

```
scgpm_seqresults_dnanexus.dnanexus_utils
    (module), 11
select_newest_project()      (in      module
    scgpm_seqresults_dnanexus.dnanexus_utils),
    12
share_with_org()            (in      module
    scgpm_seqresults_dnanexus.dnanexus_utils),
    12
```