# ATLAS Documentation

*Release 2.0*

**Joe Brown**

**Feb 06, 2019**

# Setup

# CHAPTER 1

# Quick Start

Three commands to start analysing your metagenome data:

```
conda install -c bioconda -c conda-forge metagenome-atlas
atlas init --db-dir databases path/to/fastq/files
atlas run
```

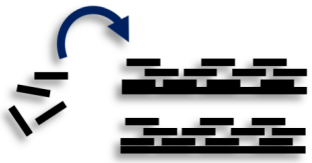
All databases and dependencies are installed on the fly in the directory `--db-dir`. You want to run this three commands on the example_data on the GitHub repo. If you have more time, then we recommend you to configure atlas according to your needs:

- check the samples.tsv
- edit the *config.yaml*
- run atlas on a *cluster system*

## 1 Quality Control

- PCR duplicates removal
- Quality trimming
- Host removal
- Common contaminant removal

➢ **QC reads**



## 2 Assembly

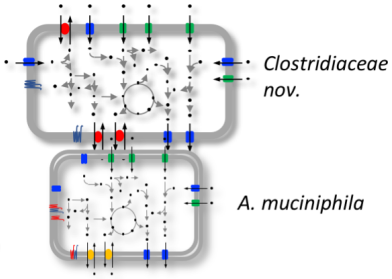
- Error correction
- Paired-end merging
- Assembly (metaSpades/megahit)
- Post-filtering

➢ **High-quality Scaffolds**

## 3 Genomic Binning

- Binning (metabat, maxbin2)
- Quality Assessment (checkM)
- Bin refining (DAS Tool)
- Dereplication (dRep)
- Quantification
- Robust taxonomic classification (CAT)

➢ **Genomes**
➢ **Abundances**




## 4 Annotation

- Gene prediction (prodigal)
- Cluster redundant genes (linclust/ cd-hit)
- Annotation (eggNOG)

➢ **Comparable gene catalog**

CHAPTER 2

# Documentation

## 2.1 Install

### 2.1.1 1a. Create conda environment

You need to install anaconda or miniconda. We recommend you to create a conda environment:

```
conda create -n atlasenv
conda activate atlasenv
```

Then install metagenome-atlas:

```
conda install -y -c bioconda -c conda-forge metagenome-atlas
```

### 2.1.2 1b. Install the development version from GitHub

Atlas is still under active development, therefore you may want to install the up to date atlas from GitHub.

Create an conda environment with all primary dependencies. All further dependencies are installed on the fly:

```
conda create -n atlasenv -c bioconda -c conda-forge python>=3.6 snakemake pandas␣
↪bbmap=37.78 click=7 ruamel.yaml biopython
```

Load the environment:

```
source activate atlasenv
```

**copy code from GitHub and install::** git clone https://github.com/metagenome-atlas/atlas.git cd atlas pip install –editable .

**Now you should be able to run atlas::** atlas init –db-dir databases path/to/fastq/files atlas run

### 2.1.3 2. Download all databases first

May be you want to make sure that all databases are downloaded correctly. Simply run:

```
atlas download --db-dir path/to/databases
```

To reassure you, most of the databases are md5 checked. The downloads use approximately 30 GB of disk space.

## 2.2 Usage

Now let's apply atlas on your data.

### 2.2.1 atlas init

```
Usage: atlas init [OPTIONS] PATH_TO_FASTQ

  Write the file CONFIG and complete the sample names and paths for all
  FASTQ files in PATH.

  PATH is traversed recursively and adds any file with '.fastq' or '.fq' in
  the file name with the file name minus extension as the sample ID.

Options:
  -d, --db-dir PATH               location to store databases (need ~50GB)
                                  [default: /Users/silas/Documents/Debug_atlas
                                  /databases]
  -w, --working-dir PATH          location to run atlas
  --assembler [megahit|spades]    assembler  [default: megahit]
  --data-type [metagenome|metatranscriptome]
                                  sample data type  [default: metagenome]
  --threads INTEGER               number of threads to use per multi-threaded
                                  job
  -h, --help                      Show this message and exit.
```

This command creates a `samples.tsv` and a `config.yaml` in the working directory.

Have a look at them with a normal text editor and check if the samples names are inferred correctly. Samples should be alphanumeric names and cam be dash delimited. Underscores should be fine too.

See the `example sample table`

The `BinGroup` parameter is used during the genomic binning. In short: all samples in which you expect the same strain to be found should belong to the same group, e.g. all metagenome samples from mice in the same cage.

You should also check the `config.yaml` file, especially:

- You may want to change the resources configuration, depending on the *system* you run atlas on.
- You may want to add ad *host genomes* to be removed.

Details about the parameters can be found in the section *Configure Atlas*

### 2.2.2 atlas run

```
Usage: atlas run [OPTIONS] [[qc|assembly|genomes|genecatalog|None|all]]
                 [SNAKEMAKE_ARGS]...

  Runs the ATLAS pipline

  By default all steps are executed but a sub-workflow can be specified.
  Needs a config-file and expects to find a sample table in the working-
  directory. Both can be generated with 'atlas init'

  Most snakemake arguments can be appended to the command for more info see
  'snakemake --help'

  For more details, see: https://metagenome-atlas.readthedocs.io

Options:
  -w, --working-dir PATH  location to run atlas.
  -c, --config-file PATH  config-file generated with 'atlas init'
  -j, --jobs INTEGER      use at most this many jobs in parallel (see cluster
                          submission for mor details).  [default: 8]
  --no-conda              do not use conda environments. good luck!  [default:
                          False]
  -n, --dryrun            Test execution.  [default: False]
  -h, --help              Show this message and exit.
```

`atlas run` need to know the working directory with a `samples.tsv` inside it.

Take note of the `--dryrun` parameter, see the section snakemake for other handy snakemake arguments.

If you want to run atlas on a cluster system you want to read the section *Execution of Atlas*.

## 2.3 Execution of Atlas

otherwise use multiple cores. The number of threads used **for each step** can be configured in the config file:

```
threads: 8
assembly_threads: 8
```

When you execute Atlas it checks how many cores are available. If you have less core available than specified in the config file. The jobs are downscaled. If you have more Atlas tries to start multiple jobs, to optimally use the cores on you machine. If you don't want that Atlas uses all the available cores then you can specify this with the `--jobs` command line argument.

### 2.3.1 Cluster execution

We recommend you to execute Atlas on a cluster system. Thank to the underlying, Snakemake workflow, Atlas can be executed on virtually all cluster systems. Each job get submitted to your cluster system with the memory (in GB) and threads specified in the config file. The `--jobs` command line argument now defines *how many jobs you want to run simultaneously on the cluster system.* The default is still the number of cores on your computer but you can set it higher `--jobs 99`.

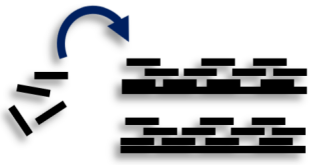
Have a look at the snakemake doc how to execute snakemake pipelines on clusters. If you have a common cluster system (Slurm, LSF, toque . . . ) you can use our snakemake profile. Note, in atlas memory is specified in GB.

## 2.4 The Atlas pipeline

### 1 Quality Control

- PCR duplicates removal
- Quality trimming
- Host removal
- Common contaminant removal

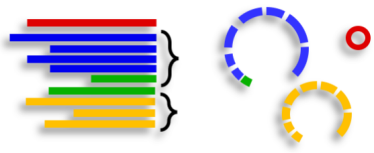
➢ **QC reads**



### 2 Assembly

- Error correction
- Paired-end merging
- Assembly (metaSpades/megahit)
- Post-filtering

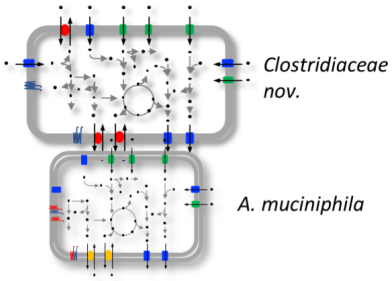
➢ **High-quality Scaffolds**



### 3 Genomic Binning

- Binning (metabat, maxbin2)
- Quality Assessment (checkM)
- Bin refining (DAS Tool)
- Dereplication (dRep)
- Quantification
- Robust taxonomic classification (CAT)

➢ **Genomes**
➢ **Abundances**




### 4 Annotation

- Gene prediction (prodigal)
- Cluster redundant genes (linclust/ cd-hit)
- Annotation (eggNOG)

➢ **Comparable gene catalog**

## 2.5 Expected output

### 2.5.1 Quality control

```
atlas run qc
#or
atlas run all
```

Runs quality control of single or paired end reads and summarizes the main QC stats in reports/QC_report.html.

Per sample it generates:

- {sample}/sequence_quality_control/{sample}_QC_{fraction}.fastq.gz
- Various quality stats in {sample}/sequence_quality_control/read_stats

#### Fractions:

When the input was paired end, we will put out three the reads in three fractions R1,R2 and se The se are the paired end reads which lost their mate during the filtering. The se are seamlessly integrated in the next steps.

### 2.5.2 Assembly

```
atlas run assembly
#or
atlas run all
```

Besides the reports/assembly_report.html this rule outputs the following files per sample:

- {sample}/{sample}_contigs.fasta
- {sample}/sequence_alignment/{sample}.bam
- {sample}/assembly/contig_stats/postfilter_coverage_stats.txt
- {sample}/assembly/contig_stats/prefilter_contig_stats.txt
- {sample}/assembly/contig_stats/final_contig_stats.txt

### 2.5.3 Genomes

```
atlas run genomes
#or
atlas run all
```

#### Binning

When you use different binners (e.g. metabat, maxbin) and a binner-reconciliator (e.g. DAS Tool), then Atlas will produce for each binner and sample:

- {sample}/binning/{binner}/cluster_attribution.tsv

which shows the attribution of contigs to bins. For the final_binner it produces the

- reports/bin_report_{binner}.html

See an example

As a summary of the quality of all bins. These bins are then De-replicated using DeRep. The Metagenome assembled genomes are then renamed, but we keep mapping files.

- genomes/Dereplication
- genomes/clustering/contig2genome.tsv
- genomes/clustering/allbins2genome.tsv

#### The main output files

- genomes/genomes
- genomes/annotations/genes
- genomes/checkm/completeness.tsv
- genomes/taxonomy/taxonomy_names.tsv
- genomes/counts/median_coverage_genomes.tsv
- genomes/counts/raw_counts_genomes.tsv

### 2.5.4 Gene Catalog

```
atlas run all
# or
atlas run genecatalog
```

The gene catalog takes all genes predicted from the genomes and clusters them according to the configuration. This rule produces the following output file for the whole dataset.

- Genecatalog/gene_catalog.fna
- Genecatalog/gene_catalog.faa
- Genecatalog/annotations/eggNog.tsv

## 2.6 Configure Atlas

### 2.6.1 Example config file

```
######################################################################
######################################################################
####                  ________   _                        _____      ####
####          /\      |__   __| | |           /\         / ____|     ####
####         /  \        | |    | |          /  \       | (___       ####
####        / /\ \       | |    | |         / /\ \       \___ \      ####
####       / ____ \      | |    | |____    / ____ \      ____) |     ####
####      /_/    \_\     |_|    |______|  /_/    \_\    |_____/      ####
####                                                               ####
######################################################################
#  For more details about the config values see:
#  https://metagenome-atlas.readthedocs.io
######################################################################

########################
# Execution parameters
########################
# max cores per process
threads: 32
# Memory for most jobs especially from BBtools, which are memory demanding
java_mem: 32
# can be a subset of threads or altered if rule run_spades or run_megahit are being␣
↪defined differently in your cluster configuration
```

(continues on next page)

(continued from previous page)

```
assembly_threads: 8
# in GB
assembly_memory: 250
# Local directory for temp files, useful for cluster execution without shared file␣
↪system
tmpdir: /tmp
# directory where databases are downloaded with 'atlas download'
database_dir: databases



########################
# Quality control
########################
data_type: metagenome # metagenome or metatranscriptome
# remove (PCR)-duplicated reads using clumpify
deduplicate: true
duplicates_only_optical: false
duplicates_allow_substitutions: 2
# used to trim adapters from reads and read ends
preprocess_adapters: /path/to/databases/adapters.fa
preprocess_minimum_base_quality: 10
preprocess_minimum_passing_read_length: 51
# 0.05 requires at least 5 percent of each nucleotide per sequence
preprocess_minimum_base_frequency: 0.05
preprocess_adapter_min_k: 8
preprocess_allowable_kmer_mismatches: 1
preprocess_reference_kmer_match_length: 27
# error correction where PE reads overlap
error_correction_overlapping_pairs: true
#contamination references can be added such that -- key: /path/to/fasta
contaminant_references:
  PhiX: /path/to/databases/phiX174_virus.fa
# We won't allow large indels
contaminant_max_indel: 20
contaminant_min_ratio: 0.65
contaminant_kmer_length: 13
contaminant_minimum_hits: 1
contaminant_ambiguous: best


########################
# Pre-assembly-processing
########################

normalize_reads_before_assembly : false
# target kmer depth
normalization_target_depth: 10000
normalization_kmer_length: 21
normalization_minimum_kmers: 3

error_correction_before_assembly : true

# join R1 and R2 at overlap; unjoined reads are still utilized
merge_pairs_before_assembly : true
# extend reads while merging to this many nucleotides
```

(continues on next page)

(continued from previous page)

```
merging_extend2: 40
# Iterations are performed until extend2 x iterations
merging_flags: ecct iterations=5
merging_k: 62


########################
# Assembly
########################
# megahit OR spades
assembler: megahit

# Megahit
#-----------
# 2 is for metagenomes, 3 for genomes with 30x coverage
megahit_min_count: 2
megahit_k_min: 21
megahit_k_max: 121
megahit_k_step: 20
megahit_merge_level: 20,0.98
megahit_prune_level: 2
megahit_low_local_ratio: 0.2
# ['default','meta-large','meta-sensitive']
megahit_preset: default

# Spades
#------------
spades_skip_BayesHammer: False
spades_use_scaffolds: true # otherwise use contigs
#Comma-separated list of k-mer sizes to be used (all values must be odd, less than␣
↪128 and listed in ascending order).
spades_k: auto
spades_preset: meta    # meta, ,normal, rna  single end libraries doesn't work for␣
↪metaspades
spades_extra: ""

# Filtering
#------------
prefilter_minimum_contig_length: 200
# filter out assembled noise
# this is more important for assemblys from megahit
filter_contigs: true
# trim contig tips
contig_trim_bp: 0
# require contigs to have read support
minimum_average_coverage: 1
minimum_percent_covered_bases: 20
minimum_mapped_reads: 0
# after filtering
minimum_contig_length: 300


########################
# Quantification
########################

# Mapping reads to contigs
```

(continues on next page)

(continued from previous page)

```
#--------------------------
contig_min_id: 0.76
contig_map_paired_only: true
contig_max_distance_between_pairs: 1000
maximum_counted_map_sites: 10

########################
# Binning
########################

final_binner: DASTool             # [DASTool or one of the binner, e.g. maxbin]

binner:                           # If DASTool is used as final_binner, use␣
↪predictions of this binners
  - metabat
  - concoct
  - maxbin


metabat:
  sensitivity: sensitive
  min_contig_length: 1500 # metabat needs >1500

concoct:
  Nexpected_clusters: 200           # important parameter
  read_length: 100                  # change this parameter !
  Niterations: 500
  min_contig_length: 1000

maxbin:
  max_iteration: 50
  prob_threshold: 0.9
  min_contig_length: 1000

DASTool:
  search_engine: 'diamond'
  score_threshold: 0.5              #Score threshold until selection algorithm will␣
↪keep selecting bins [0..1].

genome_dereplication:
  ANI: 0.99
  overlap: 0.6
  opt_parameters: ""
  filter:
      noFilter: false
      length: 5000
      completeness: 75
      contamination: 15
  score:
      completeness: 1
      contamination: 5
      N50: 0.5
      length: 0

########################
# taxonomy
########################
```

(continues on next page)

(continued from previous page)

```
# Diamond needs up to 100 GB of memory for building the taxonomy database
# If you prefer you can dowload an existing CAT database, see docs.
diamond_mem: 100
diamond_threads: 12
# number of top hits considered for taxonomic annotation
cat_range: 5
# fraction of support needed for classification, <0.5 can give rise to duble␣
↪classification.
cat_fraction: 0.3


########################
# Gene catalog
########################
genecatalog:
  source: genomes  # which predicted proteins should be used for the gene catalog
  clustermethod: linclust # cd-hit-est or cluster or linclust see mmseqs for more␣
↪details
  minlength_nt: 100
  minid: 0.95
  coverage: 0.9
  extra: ""
  SubsetSize: 500000
```

### 2.6.2 Remove reads from Host

One of the most important steps in the Quality control is to remove host genome. You can add any number of genomes to be removed.

We recommend you to use genomes where repetitive sequences are masked. See here for more details human genome.

### 2.6.3 Detailed configuration

#### Quality control of reads

#### Adapter Trimming

FASTA file paths for adapter sequences to be trimmed from the sequence ends.

We provide the adapter reference FASTA included in *bbmap* for various

```
preprocess_adapters: /database_dir/adapters.fa
```

#### Quality Trimming

Trim regions with an average quality below this threshold. Higher is more stringent.

```
preprocess_minimum_base_quality: 10
```

### Adapter Trimming at Read Tips

Allow shorter kmer matches down to *mink* at the read ends. 0 disables.

```
preprocess_adapter_min_k: 8
```

### Allowable Mismatches in Adapter Hits

Maximum number of substitutions between the target adapter kmer and the query sequence kmer. Lower is more stringent.

```
preprocess_allowable_kmer_mismatches: 1
```

### Contaminant Kmer Length

Kmer length used for finding contaminants. Contaminant matches shorter than this length will not be found.

```
preprocess_reference_kmer_match_length: 27
```

### Read Length Threshold

This is applied after quality and adapter trimming have been applied to the sequence.

```
preprocess_minimum_passing_read_length: 51
```

### Sequence Complexity Filter

Require this fraction of each nucleotide per sequence to eliminate low complexity reads.

```
preprocess_minimum_base_frequency: 0.05
```

### Contamination Parameters

Contamination reference sequences in the form of nucleotide FASTA files can be provided and filtered from the reads using the following parameters.

If 'rRNA' is defined, it will be added back to metagenomes but not to metatranscriptomes. Additional references can be added arbitrarily, such as::

```
contaminant_references:
    rRNA: /database_dir/silva_rfam_all_rRNAs.fa
    phiX: /database_dir/phiX174_virus.fa
```

Don't look for indels longer than this:

```
contaminant_max_indel: 20
```

Fraction of max alignment score required to keep a site:

```
contaminant_min_ratio: 0.65
```

mapping kmer length; range 8-15; longer is faster but uses more memory; shorter is more sensitive:

```
contaminant_kmer_length: 12
```

Minimum number of seed hits required for candidate sites:

```
contaminant_minimum_hits: 1
```

Set behavior on ambiguously-mapped reads (with multiple top-scoring mapping locations):

- best (use the first best site)
- toss (consider unmapped, retain in reads for assembly)
- random (select one top-scoring site randomly)
- all (retain all top-scoring sites)

```
contaminant_ambiguous: best
```

For human reads we recommend to download the genome [here](), where contaminants and low complexity regions were masked.

### Pre-Assambly-processing

#### Normalization Parameters

To improve assembly time and often assemblies themselves, coverage is normalized across kmers to a target depth and can be set using:

```
# kmer length over which we calculated coverage
normalization_kmer_length: 21
# the normalized target coverage across kmers
normalization_target_depth: 100
# reads must have at least this many kmers over min depth to be retained
normalization_minimum_kmers: 8
```

#### Error Correction

Optionally perform error correction using `tadpole.sh` from BBTools:

```
perform_error_correction: true
```

### Assembly Parameters

#### Assembler

Currently, the supported assemblers are 'spades' and 'megahit' with the default setting of:

```
assembler: megahit
```

Both assemblers have settings that can be altered in the configuration:

```
# minimum multiplicity for filtering (k_min+1)-mers
megahit_min_count: 2
# minimum kmer size (<= 255), must be odd number
megahit_k_min: 21
# maximum kmer size (<= 255), must be odd number
megahit_k_max: 121
# increment of kmer size of each iteration (<= 28), must be even number
megahit_k_step: 20
# merge complex bubbles of length <= l*kmer_size and similarity >= s
megahit_merge_level: 20,0.98
# strength of low depth pruning (0-3)
megahit_prune_level: 2
# ratio threshold to define low local coverage contigs
megahit_low_local_ratio: 0.2
# minimum length of contigs (after contig trimming)
minimum_contig_length: 200
# comma-separated list of k-mer sizes (must be odd and less than 128)
spades_k: auto
```

### Contig Filtering

After assembly, contigs can be filtered based on several metrics:

```
# Discard contigs with lower average coverage.
minimum_average_coverage: 5
# Discard contigs with a lower percent covered bases.
minimum_percent_covered_bases: 40
# Discard contigs with fewer mapped reads.
minimum_mapped_reads: 0
# Trim the first and last X bases of each sequence.
contig_trim_bp: 0
```