BACTpipe Documentation

Release 3.1.0

Abinhav Sharma, Emilio Rudbeck, Joseph Kirangwa, Sandra Alva

May 10, 2021

Contents

BACTpipe is a complete pipeline for assembly of whole genome sequenced bacterial isolates.

BACTpipe can use a known reference genome to order and annotate assembled contigs, as well as assemble genomes of unknown species and automatically determine the best reference genome to use for contig ordering and annotation.

CHAPTER 1

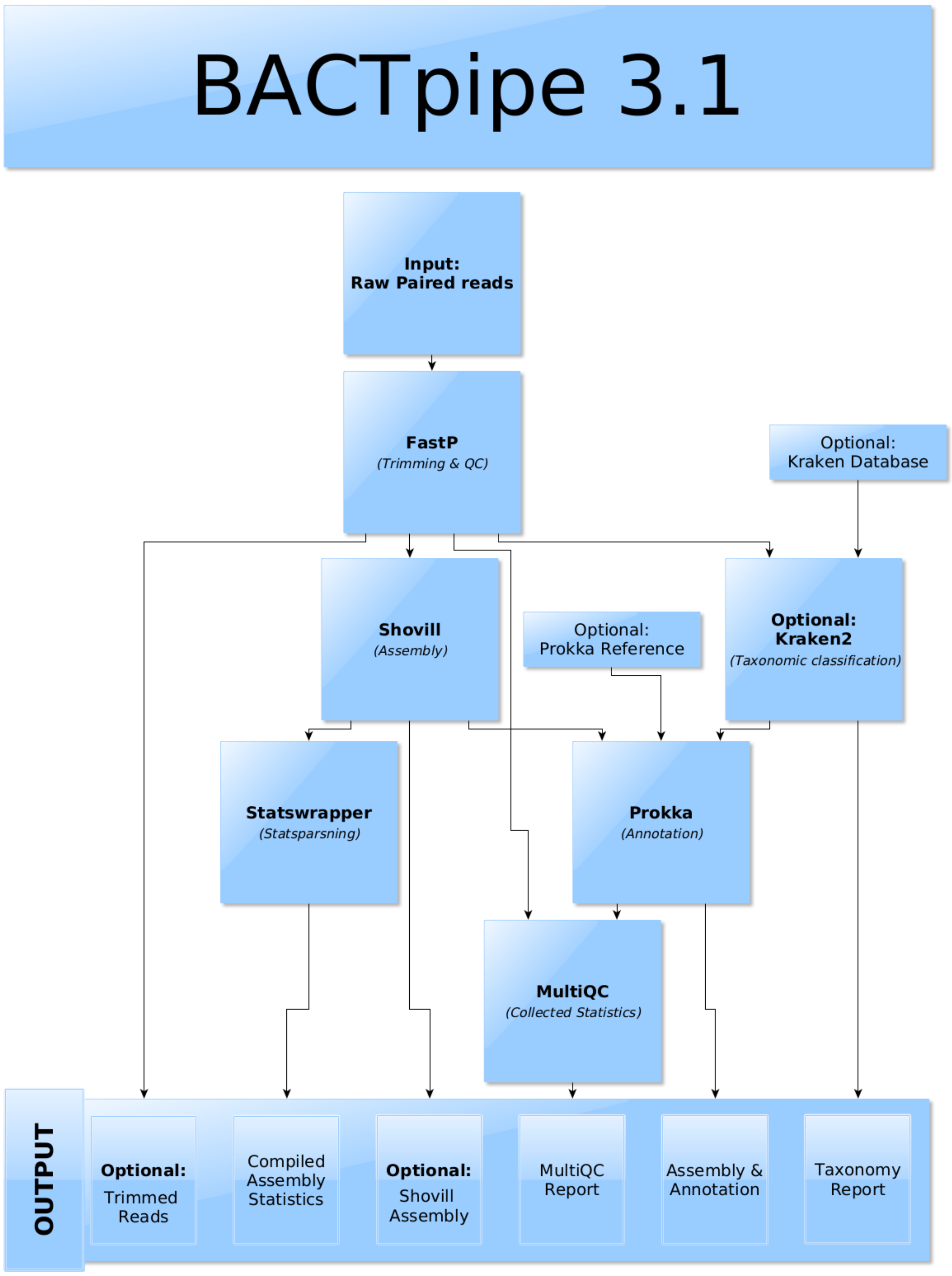
BACTpipe introduction and overview

BACTpipe uses whole genome shotgun sequenced, paired end reads, to assemble and annotate single bacterial genomes.

BACTpipe's analysis flow starts with pre-processing of paired end reads in fastq format using `fastp`, followed by taxonomic classification and gram stain identification by `Kraken2`. This step also identifies if the sample is potentially contaminated, i.e. contains more than one species. Then the sample is *de-novo* assembled using `Shovill`. The draft genome fasta file headers are renamed to get unique genome-specific headers.

Finally, genome annotation is performed using `prokka` with genus, species, and gram stain information (if possible to uniquely identify in the Kraken2 step) added. Lastly, basic statistics about the assembly and annotation are collected into a HTML report using `MultiQC`.

BACTpipe is implemented in Nextflow and an overview of the workflow can be seen below with the different output files at the bottom.


BACTpipe 3.1
Input:
Raw Paired reads
FastP
(Trimming & QC)
Optional:
Kraken Database
Shovill
(Assembly)
Optional:
Prokka Reference
Optional:
Kraken2
(Taxonomic classification)
Statswrapper
(Statsparsning)
Prokka
(Annotation)
MultiQC
(Collected Statistics)
OUTPUT
Optional:
Trimmed Reads
Compiled Assembly Statistics
Optional:
Shovill Assembly
MultiQC Report
Assembly & Annotation
Taxonomy Report

CHAPTER 2

Installation

BACTpipe pipeline is written in Nextflow and runs on Linux and Mac OSX systems.

2.1 Dependencies

In order to run BACTpipe, you need to have the following programs installed:

- Java v8+ for Nextflow
- Nextflow for workflow management, more specifically v21.04.0
- Conda for installation of workflow tools
- (Optional) A Kraken2 database if you want to classify taxonomy and gram stain to potentially improve genome annotations in the Prokka step.

2.2 Install BACTpipe

After installing all the beforementioned dependencies, BACTpipe will automatically install the software needed for the process using conda when running the pipeline.

2.3 Quick guide on how to install most tools

1. Java v8+: https://anaconda.org/cyclus/java-jdk
2. Nextflow: https://www.nextflow.io/docs/latest/getstarted.html#installation
3. Conda: https://docs.conda.io/projects/conda/en/latest/user-guide/install/
4. Kraken2 database: http://ccb.jhu.edu/software/kraken2/index.shtml?t=downloads

Tip: The repository contains a convenience script to easily download a Kraken2 DB: `resources/download_kraken2_db.sh`.

CHAPTER 3

Running BACTpipe

After installing all the required dependencies and downloading the required mash sketches of refseq genomes, it is very easy to run BACTpipe. There are several ways to run BACTpipe, but we'll start with the easiest:

```
$ nextflow run ctmrbio/BACTpipe --reads 'path/to/reads/*_R{1,2}.fastq.gz'
```

This will instruct Nextflow to go to the `ctmrbio` Github organization to download and run the `BACTpipe` workflow. The argument `--reads` is used to tell the workflow which input files you want to run BACTpipe on. Note that the path to the reads must be enclosed in single quotes (`'`) to prevent the shell from automatically expanding asterisks (`*`) and curly braces (`{}`). In the above example, the part of the filename matched by the asterisk will be used as the sample name in BACTpipe, and `{1,2}` refers to the pair of FASTQ files for paired-end data. Input data should be in FASTQ format, and can be either plain FASTQ, or compressed with gzip or bzip2 (with `.gz` or `.bz2` file suffixes).

Note: When you run BACTpipe for the first time using a command like the one shown above, Nextflow downloads the current version of the Github repo to your computer. If BACTpipe is updated after your first run, the subsequent runs will still use the old version that you have downloaded. To get the newest version, tell Nextflow to update your local copy: `nextflow pull ctmrbio/BACTpipe`.

When BACTpipe is run like this, it by default assumes you want to run everything locally, on the current machine. Note that BACTpipe is capable of running on a wide range of computers, ranging from laptops to powerful multicore machines to high-performance computing (HPC) clusters.

3.1 Changing settings on the command line

When running BACTpipe, you may want to modify parameters to customize it for your purpose. It is possible to modify several settings for how BACTpipe operates using configuration parameters. All changes can be added as command-line arguments when running BACTpipe, e.g.:

```
$ nextflow run ctmrbio/BACTpipe --shovill_kmers 21,33,55,77 --reads 'path/to/reads/*_
↪{1,2}.fastq.gz'
```

The `--shovill_kmers` flag will modify the kmer lengths that shovill will use in its SPAdes assembly. The following parameters can be easily configured from the command line:

```
Parameter name              Default setting         Description
reads                       [empty]                 Input fastq files, required!
output_dir                  BACTpipe_results        Name of outuput directory
keep_trimmed_fastq          [FALSE]                 Output trimmed fastq files from␣
↪fastp into output_dir
keep_shovill_output         [FALSE]                 Output shovill output directory into␣
↪output_dir
kraken2_db                  [empty]                 Path to Kraken2 database to use for␣
↪taxonomic classification
kraken2_confidence          0.5                     Kraken2 confidence parameter, refer␣
↪to `kraken2`_ documentation for details
kraken2_min_proportion      1.00                    Minimum proportion of reads on␣
↪sample level to classify sample as containing species
shovill_depth               100                     See the `shovill`_ documentation for␣
↪details
shovill_kmers               31,33,55,77,99,127
shovill_minlen              500
prokka_evalue               1e-09                   See the `prokka`_ documentation for␣
↪details
prokka_kingdom              Bacteria
prokka_reference            [not used]
prokka_signal_peptides      false
```

To modify any parameter, just add `--<parameter_name> <new_setting>` on the command line when running BACTpipe, e.g. `--shovill_depth 75` to set Shovill's depth parameter to 75 instead of 100. Refer to `params.config` in the `conf` directory of the BACTpipe repository for a complete up-to-date listing of all available parameters.

3.1.1 Change many settings at once

If you want to change many different settings at the same time when running BACTpipe, it can quickly result in very long command lines. A way to make it easier to change several parameters at once is to create a custom configuration file in YAML or JSON format that you give to BACTpipe using `-params-file`.

The parameter settings you define in your custom configuration file will override the default settings. Custom configuration files can be written in either YAML or JSON format. The simplest format for the custom parameters file is probably YAML, and is the recommended choice. Here is an example YAML configuration file that modifies some shovill parameters and leaves all other settings to their default values:

```
shovill_depth: "100"
shovill_kmers: "31,33,55,77,99,111,127"
shovill_minlen: "400"
```

If you save the above into a plain text file called `custom_bactpipe_config.yaml` you can provide it when running BACTpipe using the `-params-file` command line argument:

```
$ nextflow run ctmrbio/BACTpipe -params-file path/to/your/custom/params.yaml --reads
↪'path/to/reads/*_{1,2}.fastq.gz'
```

There is also another way to modify parameters that uses Nextflow's own configuration format. This can be useful if you want to modify *a lot* of settings at once, since it is possible to download a copy of the default configuration settings file directly from Github, params.config, and make any changes you want directly in your custom version of `params.config`. The file contains some comments explaining how the different variables work, to help out when modifying the settings. To run BACTpipe with a custom configuration in the Nextflow format, you use `-c` on the command line:

```
$ nextflow run ctmrbio/BACTpipe -c path/to/custom_params.config --reads 'path/to/
↪reads/*_{1,2}.fastq.gz'
```

3.1.2 Note:

There are two different type of commandline arguments when running workflows using Nextflow: 1) arguments using double dashes (i.e. `--reads`) and 2) arguments using a single dash (i.e. `-params-file`). Arguments using double dashes are sent to BACTpipe for evaluation, and are typically configuration variables that are defined inside BACTpipe. Arguments using a single dash are not visible to BACTpipe but are instead used by Nextflow itself, and typically alters how Nextflow executes BACTpipe.

3.2 Profiles

A convenient way to modify the way BACTpipe is run in your environment is to load a profile. BACTpipe comes with a few pre-installed profiles:

- `standard` – For local use on e.g. a laptop or Linux server. This is the default profile used if no profile is explicitly specified.
- `rackham` – For use on the UPPMAX's Rackham HPC system.
- `ctmr_nas` – For local execution on CTMR's old analysis server.
- `ctmr_gandalf` – For use on CTMR's Gandalf Slurm HPC system.
- `docker` – For use with docker containers.

Cluster profiles

Note that when running profiles that uses a cluster scheduler, for example like Slurm that is used on UPPMAX systems in the `rackham` profile, you also need to provide what Slurm account/project BACTpipe should use when submitting jobs. This can be done with `--project account_name` on the command line, or by adding it to a custom configuration file (see previous section).

To run BACTpipe with a specific profile, use the `-profile <profilename>` argument (note the single dash before `profile`) when running, e.g.:

```
$ nextflow run ctmrbio/BACTpipe -profile rackham --project SNIC001 --reads '/proj/
↪projectname/reads/*_{1,2}.fastq.gz'
```

This will run BACTpipe using the `rackham` profile with the project set to `SNIC001`, which automatically configures settings so BACTpipe can find all the required software and databases in the UPPMAX HPC cluster environment. Running BACTpipe without a `-profile` argument will default to running the `standard` profile directly on the node you are logged in to (avoid doing that on shared HPC systems).

3.3 Custom profile

It is possible to create a custom profile to use instead of the preconfigured ones. This is useful if you want to run BACTpipe on another cluster system than UPPMAX's Rackham, or if the data you are analyzing requires you to change the pre-defined expected CPU, memory, and time requirements for processes on the cluster. The best way to start is probably to download one of the pre-existing profiles from conf directory of the BACTpipe repository.

If you are working on a Slurm-managed system, starting with either the `rackham.config` or the `ctmr_gandalf` profile would be a good choice, as both of those are Slurm-managed HPC systems. Download the configuration file from the conf directory of the BACTpipe repository and modify settings to your preference. Then, to run BACTpipe using your custom configuration file, you need to tell Nextflow to read parameters from your file instead of the default parameters:

```
$ nextflow run ctmrbio/BACTpipe -c path/to/your/custom/profile.config --reads 'path/
↪to/reads/*_{1,2}.fastq.gz'
```

The custom profile is not limited to configuring CPU, memory and time limits for the different processes. It is also possible to set parameter values inside the custom profile, i.e. to change paths to reference databases or adjust runtime parameters for the different processes. It is also possible to just use a configuration file that changes settings without modifying how the workflow is run, see *Change many settings at once*.

CHAPTER 4

Output Files

BACTpipe pipeline outputs several files in specific folders within the specified output directory (`BACTpipe_results` is the default output directory).

4.1 Output folders

- fastp
- shovill (optional)
- kraken2 (optional)
- prokka
- multiqc

4.2 Description of file outputs

The following table provides a description of the most relevant files in each folder mentioned above.

Output Folder	File output	File Description
fastp	• `*.fastp.fq.gz` (optional) • `*fastp.json`	• Trimmed reads for each sample (For R1 and R2 respectively) • Quality trimming statistics
kraken2	• `*.kreport` • `*.classification.txt`	• Kraken2 report • Genus, species, and Gram stain classification
shovill	• `*_shovill` (optional) • `*.assembly_stats.txt`	• Shovill output directory containing `*.fa`, `*.fasta`, `*.fastg`, `*.gfa`, `*.hist`, `*.log`, `*.changes`, and `*.tab` files • Summary of assembly statistics
prokka	• `*.gff` • `*.gbk` • `*.fna` • `*.faa` • `*.ffn` • `*.txt`	• Annotation in GFF3 format, containing both sequences and annotations • Standard Genbank file • Nucleotide FASTA file of the input contig sequences • Protein FASTA file of the translated CDS sequences • Nucleotide FASTA file of all the prediction transcripts (CDS, rRNA, tRNA, tmRNA, misc_RNA) • Statistics relating to the annotated features found
multiqc	• `multiqc-report.html`	• Summary report of statistics generated by prokka and fastp tools

In addition to the files listed in the table above, Nextflow also produces two report files in the main run folder after the pipeline is finished. They are called `BACTpipe_report.html` and `BACTpipe_timeline.html`. The reports shows a summary of overall execution time, resource usage, and all executed tasks and their respective run time metrics. Lastly, Nextflow produces a `work` directory containing all intermediate files and logs produced in the process. This folder can be removed once the process has completed.

CHAPTER 5

About

Authors Abhinav Sharma, Emilio Rudbeck, Joseph Kirangwa, Sandra Alvarez-Carretero, Fredrik Boulund, Kaisa Thorell

Contact kaisa.thorell@gu.se

License MIT

This is the documentation for BACTpipe version 3.1.0, last updated May 10, 2021. The documentation is available online at https://bactpipe.readthedocs.org.

BACTpipe is published as open source under the MIT license and you are welcome to look at, suggest improvements, or download and improve/contribute to the code at the project's Github repository page.

CHAPTER 6

Contributing

If you want to contribute to the development of BACTpipe, please fork the Github repository and submit a pull request.

CHAPTER 7

Citing BACTpipe

If you find BACTpipe useful and publish something using BACTpipe, please cite:

Emilio Rudbeck; Abhinav Sharma; Joseph Kirangwa; Yue Hu; Sandra Álvarez-Carretero; Fredrik Boulund; Kaisa Thorell (2017-2021).
BACTpipe: bacterial whole genome sequence assembly and annotation pipeline.
https://github.com/ctmrbio/BACTpipe
DOI: https://doi.org/10.5281/zenodo.4742358

CHAPTER 8

Indices and tables

- genindex
- modindex
- search