## Contents

1. **Below is the list of pipelines available in snakePipes**  
   3

2. **Quick start**  
   5

3. **Running your own analysis**  
   7

4. **Citation**  
   9

5. **Help and Support**  
   11

6. **Contents:**
   6.1 Setting up snakePipes ......................................................... 13
   6.2 Running snakePipes .......................................................... 19
   6.3 Advanced usage of snakePipes ............................................. 22
   6.4 createIndices ....................................................................... 27
   6.5 DNA-mapping ....................................................................... 31
   6.6 ChIP-seq ............................................................................. 37
   6.7 ATAC-seq ............................................................................ 45
   6.8 HiC ...................................................................................... 52
   6.9 preprocessing ....................................................................... 59
   6.10 mRNA-seq ........................................................................... 64
   6.11 noncoding-RNA-seq ............................................................ 74
   6.12 scRNA-seq .......................................................................... 81
   6.13 WGBS ............................................................................... 89
   6.14 snakePipes News ................................................................. 98

7. **Indices and tables** ................................................................. 105
snakePipes are pipelines built using snakemake and python for the analysis of epigenomic datasets.
Below is the list of pipelines available in snakePipes

<table>
<thead>
<tr>
<th>Pipeline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createIndices</td>
<td>Create indices for an organism for further use within snakePipes</td>
</tr>
<tr>
<td>DNA-mapping</td>
<td>Basic DNA mapping using bowtie2, filter mapped files, QC and create coverage plots</td>
</tr>
<tr>
<td>ChIP-seq</td>
<td>Use the DNA mapping output and run ChIP/Input normalization and peak calling</td>
</tr>
<tr>
<td>ATAC-seq</td>
<td>Use the DNA mapping output and detect open chromatin regions for ATAC-seq data</td>
</tr>
<tr>
<td>HiC</td>
<td>Hi-C analysis workflow, from mapping to TAD calling</td>
</tr>
<tr>
<td>noncoding-RNA-seq</td>
<td>noncoding-RNA-Seq workflow : From mapping to differential expression of genes and repeat elements using DESeq2</td>
</tr>
<tr>
<td>mRNA-seq</td>
<td>RNA-Seq workflow : From mapping to differential expression using DESeq2</td>
</tr>
<tr>
<td>scRNA-seq</td>
<td>Single-cell RNA-Seq (CEL-Seq2) workflow : From mapping to differential expression</td>
</tr>
<tr>
<td>WGBS</td>
<td>Whole-genome Bisulfite-Seq analysis workflow, from mapping to DMR calling and differential methylation analysis</td>
</tr>
<tr>
<td>preprocessing</td>
<td>Merging technical replicates (e.g., across lanes), removing optical duplicates, running FastQC</td>
</tr>
</tbody>
</table>
Chapter 1. Below is the list of pipelines available in snakePipes
CHAPTER 2

Quick start

- Assuming you have python3 with conda, install snakePipes with:

  `conda create -n snakePipes -c mpi-ie -c conda-forge -c bioconda snakePipes==2.4.1`

- You can update snakePipes to the latest version available on conda with:

  `conda update -n snakePipes -c mpi-ie -c conda-forge -c bioconda --prune snakePipes`

snakePipes is going to move to mamba in the future.
- Download genome fasta and annotations for your organism, and build indexes. Check in `createIndices`
- Configure snakePipes with paths to organism and cluster configs on your system using snakePipes config. For detailed information, run:

  `snakePipes config --help`

**Note:** If you have a copy of a `shared/defaults.yaml` with the necessary paths configured (i.e. from a previous installation), you can pass it to snakePipes config with `--oldConfig` and `--configMode recycle` instead of providing all the paths manually again. Config keys have to match for this to work. In the same way, you can pass your external organism yaml folder with `--organismsDir` or cluster config with `--clusterConfig`.

- Download example fastq files for the human genome [here](#)
- Execute the DNA-mapping pipeline using the example `command.sh` in the test data directory.
CHAPTER 3

Running your own analysis

For a detail introduction to setting up snakePipes from scratch, please visit *Setting up snakePipes*

For each organism of interest, snakePipes requires fasta files, genome indexes and annotation files. Paths to these files are specified in the organism/<name>.yaml files. After installation, the location of these files could be revealed by the following command:

```
snakePipes info
```

You could either modify the existing files (add your own paths), or add a new file there. See more detail in *Running snakePipes*

snakePipes could either be executed locally, or on any snakemake-supported cluster infrastructure. See details for setting up the cluster command in *Running snakePipes*
If you adopt/run snakePipes for your analysis, cite it as follows:


This tool suite is developed by the Bioinformatics Unit at the Max Planck Institute for Immunobiology and Epigenetics, Freiburg.
For query/questions regarding snakePipes, please write on biostars with the tag `#snakePipes`
For feature requests or bug reports, please open an issue on our GitHub Repository.
6.1 Setting up snakePipes

Unlike many other pipelines, setting up snakePipes is easy! All you need is a linux/OSX system with python3-conda installation.

6.1.1 Installing conda with python3

Follow the instructions here to install either miniconda or anaconda. A minimal version (miniconda) is enough for snakePipes. Get the miniconda installer here.

After installation, check your python path and version:

```bash
$ which python
$ /your_path/miniconda3/bin/python

$ python --version # anything above 3.5 is ok!
$ Python 3.6.5 :: Anaconda, Inc.

$ conda --version # only for sanity check
$ conda 4.5.8
```

Next, install snakePipes.

6.1.2 Installing snakePipes

The easiest way to install snakePipes is via our conda channel. The following command install snakePipes and also creates a conda virtual environment named snakePipes, which you can then activate via conda activate snakePipes. Specifying snakePipes version avoids issues with conda’s environment solver.

```
conda create -n snakePipes -c mpi-ie -c conda-forge -c bioconda snakePipes==2.4.1
```
This way, the software used within snakePipes do not conflict with the software pre-installed on your terminal or in your python environment.

**Note:** This might take a few minutes depending on the access to conda channels.

snakePipes is going to move to mamba in the future.

**Development installation**

If you wish to modify snakePipes you can install it via pip from within a conda environment, using our GitHub repository.

```
conda create -n snakepipes python=3.7 snakemake pandas graphviz fuzzywuzzy
conda activate snakepipes
pip install git+https://github.com/maxplanck-ie/snakepipes@develop
```

Instead of providing the URL to pip, you can also clone our GitHub repository on your computer, and modify the code before running snakePipes. Please see *Advanced usage of snakePipes* for more information on how to modify and extend snakePipes workflows.

**Testing whether the installation went fine**

After installation, you can activate the snakePipes environment via

```
conda activate snakePipes
```

In case you installed conda using the latest version of conda installers (eg. miniconda 4.5.*, or later), the *conda* command might not be available inside an environment. To enable this, export the path to conda/bin in your $PATH (or append the path manually in your *bashrc*)

```
export PATH="/path/to/miniconda3/bin:$PATH"
```

Snakemake and pandas are installed along with snakePipes as requirements. Ensure you have them working by testing these commands:

```
snakemake --help
snakePipes --help
```

### 6.1.3 Modify global options

It is often useful to store organism YAML files and the cluster configuration file outside of snakePipes, so that these can be used across snakePipes versions without needing to make copies. Since snakePipes 1.3.0, this can be done by modifying the *defaults.yaml* file, the location of which is given by `snakePipes info`. Instead of manually modifying this file, you may also use `snakePipes config`.

To see the location of the various YAML files so you can manually inspect them, you can use:

```
snakePipes info
```

This would show the locations of:

- *defaults.yaml* Defines default tool and file paths. See *Create the conda environments*
• **cluster.yaml** Defines execution command for the cluster. See *Configure your cluster*

• **organisms/<organism>.yaml** : Defines genome indices and annotations for various organisms. See *Configure the organisms*

• Workflow-specific defaults : Defines default options for our command line wrappers. See *Configure default options for workflows*

It is a good idea to keep a copy of your defaults.yaml, cluster.yaml and the whole organism folder in a dedicated location e.g. some folder outside the snakePipes installation folder named "snakePipes_configs". You can configure snakePipes to use these files after a fresh installation or update with snakePipes config --organismsDir my_organisms_dir --clusterConfig my_cluster_config . This will also work if you add --configMode recycle.

### 6.1.4 Create the conda environments

All the tools required for running various pipelines are installed via various conda repositories (mainly bioconda). The following commands installs the tools and creates the respective conda environments.

```bash
snakePipes createEnvs
```

**Note:** Creating the environments might take 1 hour. But it only has to be done once.

**Note:** snakePipes createEnvs will also set the `snakemakeOptions:` line in the global snakePipes defaults.yaml files. If you have already modified this then use the --keepCondaDir option.

**Warning:** If you installed with pip you must use the --develop option.

The place where the conda envs are created (and therefore the tools are installed) is defined in snakePipes/defaults.yaml file on our GitHub repository. You can modify it to suite your needs.

Here are the content of defaults.yaml:

```yaml
snakemakeOptions: '--use-conda --conda-prefix /data/general/scratch/conda_envs'
```

**Note:** Whenever you change the `snakemakeOptions:` line in defaults.yaml, you should run snakePipes createEnvs to ensure that the conda environments are then created.

Running snakePipes createEnvs is not strictly required, but facilitates multiple users using the same snakePipes installation.

### 6.1.5 Configure the organisms

For each organism of your choice, create a file called `<organism>.yaml` in the folder specified by `organismsDir` in defaults.yaml and fill the paths to the required files next to the corresponding yaml entry. For common organisms, the required files are downloaded and the yaml entries can be created automatically via the workflow createIndices.

The yaml files look like this after the setup (an example from drosophila genome dm3):
# Integer, size of genome in base-pairs
genome_size: 142573017

# path to genome.fasta for mapping
genomefasta: "/data/repository/organisms/dm3_ensembl/genome fasta/genome.fa"

# path to genome.fasta.fai (fasta index) for mapping
genome_index: "/data/repository/organisms/dm3_ensembl/genome fasta/genome.fa.fai"

# OPTIONAL. Needed for QC bias estimation by deepTools
genome_2bit: "/data/repository/organisms/dm3_ensembl/genome fasta/genome fasta/ genome.2bit"

# Needed for DNA-mapping workflow
bowtie2_index: "/data/repository/organisms/dm3_ensembl/BowtieIndex/ genome"

# index of the genome.fasta using HISAT2, needed for RNA-seq workflow
hisat2_index: "/data/repository/organisms/dm3_ensembl/HISAT2Index/ genome"

# needed by HISAT2 for RNA-seq workflow
known_splicsites: "/data/repository/organisms/dm3_ensembl/ensembl/release-78/HISAT2/ splice sites.txt"

bwa_index: "/data/repository/organisms/dm3_ensembl/BWAindex/ genome"

# index of the genome.fasta using STAR, needed for RNA-seq workflow
star_index: "/data/repository/organisms/dm3_ensembl/STARIndex/

# Needed for QC and annotation in DNA-mapping/RNA-Seq workflows
genes bed: "/data/repository/organisms/dm3_ensembl/Ensembl/release-78/genes.bed"

# Needed for QC and annotation in DNA-mapping/RNA-Seq workflows
genes gtf: "/data/repository/organisms/dm3_ensembl/Ensembl/release-78/genes.gtf"

# OPTIONAL. For QC and filtering of regions in multiple workflows.
blacklist bed:

# STRING. Name of the chromosomes to ignore for calculation of normalization factors for coverage files
ignoreForNormalization: "U Uextra X XHet YHet dmel mitochondrion genome"

---

**Warning:** Do not edit the yaml keywords corresponding to each required entry.

---

**Note:** Some fields are optional and can be left empty. For example, if a blacklist file is not available for your organism of interest, leave `blacklist bed:` empty. Files for either STAR or HISAT2 could be skipped for RNA-seq if the respective aligner is not used. We nevertheless recommended providing all the files, to allow more flexible analysis.

---

After setting up the yamls, we can execute a snakePipes workflow on the organism of choice by referring to the **organism** as `dm3`, where the keyword `dm3` matches the name of the yaml file (`dm3.yaml`).

**Note:** The name of the yaml file (except the `.yaml` suffix) is used as keyword to refer to the organism while running the workflows.

### 6.1.6 Download premade indices

For the sake of convenience, we provide premade indices for the following organisms:

- Human (GRCh38, Gencode release 29)
- Mouse (GRCm38/mm10, Gencode release m19)
- Mouse (GRCm37/mm9, Gencode release 1)
- Fruit fly (dm6, Ensembl release 94)
To use these, simply download and extract them. You will then need to modify the provided YAML file to indicate exactly where the indices are located (i.e., replace /data/processing/ryan with whatever is appropriate).

## 6.1.7 Configure your cluster

The `cluster.yaml` file contains both the default memory requirements as well as two options passed to snakemake that control how jobs are submitted to the cluster and files are retrieved:

```yaml
snakemake_latency_wait: 300
snakemake_cluster_cmd: module load slurm; SlurmEasy --mem-per-cpu {cluster.memory} --threads {threads} --log {snakePipes_cluster_logDir} --name {rule}.snakemake
snakePipes_cluster_logDir: cluster_logs
__default__:  
  memory: 8G
snp_split:  
  memory: 10G
```

The location of this file must be specified by the `clusterConfig` value in `defaults.yaml`.

You can change the default per-core memory allocation if needed here. Importantly, the `snakemake_cluster_cmd` option must be changed to match your needs (see table below). Whatever command you specify must include a `{cluster.memory}` option and a `{threads}` option. You can specify other required options here as well. The `snakemake_latency_wait` value defines how long snakemake should wait for files to appear before throwing an error. The default of 300 seconds is typically reasonable when a file system such as NFS is in use. Please also note that there are additional memory settings for each workflow in `snakePipes/workflows/[workflow]/cluster.yaml` that you might need to adjust.

`snakePipes_cluster_logDir:` can be used like a wildcard in `snakemake_cluster_cmd` to specify the directory for the stdout and stderr files from a job that is running on the cluster. This is given separate to make sure the directory exists before execution. A relative path is treated relative to the output directory of the workflow. If you want, you can also give an absolute log directory starting with `/`.

<table>
<thead>
<tr>
<th>Scheduler/Queuing</th>
<th>snakemake_cluster_cmd example</th>
</tr>
</thead>
</table>
| **slurm**         | snakemake_cluster_cmd: module load slurm; 
|                   | → sbatch --ntasks-per-node=1 
|                   | → -c {threads} -J {rule}.snakemake 
|                   | → --mem-per-cpu={cluster.memory} 
|                   | → -p MYQUEUE -o {snakePipes_cluster_logDir}/[rule].%j.out 
|                   | → -e {snakePipes_cluster_logDir}/[rule].%j.err 
|                   | → snakePipes_cluster_logDir: cluster_logs |
| **PBS/Torque**    | snakemake_cluster_cmd: qsub -N {rule}. 
|                   | → -q MYQUEUE -l pmem={cluster.memory} 
|                   | → -l walltime=20:00:00 -l nodes=1:ppn= 
|                   | → {cluster.threads} 
|                   | → -o {snakePipes_cluster_logDir}/[rule]. 
|                   | → -e {snakePipes_cluster_logDir}/[rule]. 
|                   | → -o $PBS_JOBID.out 
|                   | → -e $PBS_JOBID.err 
|                   | → snakePipes_cluster_logDir: cluster_logs |
| **SGE**           | Please send us a working example! |

### 6.1. Setting up snakePipes
6.1.8 Configure default options for workflows

The default options for all command-line arguments as well as for the cluster (memory) are stored in the workflow-specific folders. If you have cloned the repository locally, these files are located under `snakePipes/workflows/<workflow_name>` folder. You can modify the values in these yamls to suite your needs. Most of the default values could also be replaced from the command line wrappers while executing a workflow.

Below are some of the workflow defaults from the DNA-mapping pipeline. Empty sections means no default is set:

```yaml
## key for the genome name (eg. dm3)
genome:
## FASTQ file extension (default: ".fastq.gz")
ext: '.fastq.gz'
## paired-end read name extension (default: ['_R1', '_R2'])
reads: ['_R1', '_R2']
## mapping mode
mode: mapping
aligner: Bowtie2
## Number of reads to downsample from each FASTQ file
downsample:
## Options for trimming
trim: False
trimmer: cutadapt
trimmerOptions:
## Bin size of output files in bigWig format
bwBinSize: 25
## Run FASTQC read quality control
fastqc: false
## Run computeGCBias quality control
GCBias: false
## Retain only de-duplicated reads/read pairs
dedup: false
## Retain only reads with at least the given mapping quality
mapq: 0
```

6.1.9 Test data

Test data for the various workflows is available at the following locations:

- DNA mapping
- ChIP-seq
- ATAC-seq
- mRNA-seq
- noncoding-RNA-seq
- HiC
- WGBS
- scRNA-seq

code @ github.
6.2 Running snakePipes

Pipelines under snakePipes are designed in a way such that all workflows are configured and ran in a similar way.

6.2.1 An example with ChIP-seq data

A typical ChIP-seq analysis of human samples starts from paired-end FASTQ files in the directory `input-dir`:

```bash
$ ls /path/to/input-dir/
my_H3K27ac_sample_R1.fastq.gz my_H3K27me3_sample_R1.fastq.gz my_Input_sample_R1.fastq.gz
my_H3K27ac_sample_R2.fastq.gz my_H3K27me3_sample_R2.fastq.gz my_Input_sample_R2.fastq.gz
```

The ChIP-seq workflow requires the files to be processed via the DNA-mapping workflow first. We therefore run the DNA-mapping workflow:

```bash
$ DNA-mapping -i /path/to/input-dir -o /path/to/output-dir --mapq 5 -j 10 --dedup --hs37d5
```

- `--mapq 5` would filter mapped reads for a minimum mapping quality of 5. This would keep only primary alignments from bowtie2, sufficient for downstream analysis.
- `--dedup` would remove PCR duplicates (reads with matching 5' position in the genome), a typical step in ChIP-Seq analysis.
- `-j 10` defines 10 jobs to be run in parallel on the cluster (see below).
- `--hs37d5` is the name of the genome (keyword for the yaml). The yaml file corresponding to this genome should exist as `snakePipes/shared/organisms/hs37d5.yaml` (see Setting up snakePipes for details).

All individual jobs of the workflow will be submitted to the Grid engine using the command specified under `shared/cluster.yaml`. The parameter `-j` defines the number of jobs to be run in parallel, while the number of threads per job is hard-coded in the workflows.

To run the workflow locally, use the parameter `--local` for local mode and the parameter `-j 10` to specify the maximal number of used CPU threads (here: 10).

For single-end FASTQ files, the workflow would automatically recognize them if the file name has no suffix (e.g., "sample1.fastq" instead of "sample1_R1.fastq"). However, mixing of single and paired-end files in the same folder is not supported currently.

Once the DNA-mapping run is finished successfully. We can run the ChIP-seq analysis in the same directory:

```bash
$ ChIP-seq -d /path/to/dna-mapping-output/ hs37d5 chip-samples.yaml
```

- `-d` specifies the directory where the output of DNA-mapping workflow lies. The ChIP-seq workflow would also write it's output there.
- `hs37d5` is the name of the genome (keyword for the yaml).
- `chip-samples.yaml` is a yaml file that defines for each ChIP sample, the corresponding control (input) sample and the type of mark (broad/sharp). See ChIP-seq for more details on how to setup this yaml file.

The ChIP-seq workflow would follow up from the DNA-mapping outputs and perform peak calling, create ChIP-input normalized coverage files and also perform differential (control-test) analysis if a sample information file is provided (see below).
6.2.2 The sample sheet

Most of the workflows allow users to perform grouped operations as an option, for example differential expression analysis in mRNA-seq workflow, differential binding analysis in ChiP-Seq workflow, differential open-chromatin analysis in ATAC-seq workflow or merging of groups in Hi-C workflow. For all this analysis, snakePipes needs a `sampleSheet.tsv` file (file name is not important, but it has to be tab-separated) that contains sample grouping information. In most cases users would want to group samples by replicates. The format of the file is as follows:

```
name  condition
sample1  control
sample1  control
sample2  test
sample2  test
```

The name section refers to sample names (without the read suffix), while the condition section refers to sample group (control/test, male/female, normal/diseased etc.)

6.2.3 Using BAM input

In many workflows it is possible to directly use BAM files as input by specifying `--fromBAM`. Note that you must then specify whether you have paired-end (the default) or single-end data. This is typically done with the `--singleEnd` option.

6.2.4 Changing read extensions or mate designators

The default file names produced by Illumina sequencers are of the form `<sample>_R1.fastq.gz` and `<sample_R2.fastq.gz`. However, sometimes public datasets will instead have a `.fq.gz` suffix or use `_1` and `_2` as mate designators. To enable this, the `--ext` option can be used to change `.fastq.gz` default suffix to `.fq.gz` and `--reads` to `_1 _2`.

6.2.5 Common considerations for all workflows

All of the snakePipes workflows that begin with a FASTQ file, perform the same pre-processing steps.

- **Linking/downsampling the FASTQ file**: The FASTQ rule in the workflows links the input FASTQ file into the FASTQ folder in the output directory. If downsampling is specified, the FASTQ folder would contain the downsampled FASTQ file.

**Note**: The DNA-mapping and RNA-mapping pipelines can take either single, or paired-end FASTQ files. For paired-end data, the reads R1 and R2 are expected to have the suffix `_R1` and `_R2` respectively, which can be modified in the `defaults.yaml` file using the `reads` key, to your needs. For example, if downloaded from NCBI would normally have the extension `.1.fastq.gz` and `.2.fastq.gz`. Also, please check the `ext` key in the configuration file if you wish to modify the read extension (default is `.fastq.gz`).

- **Quality/adaptor trimming** (optional): If `--trim` is selected, the `trimming` rule would run the selected program (either Trimgalore, or Cutadapt) on the files in the FASTQ folder, and would produce another folder with name `FASTQ_<program>`, where `<program>` is either Cutadapt or Trimgalore.

- **FastQC** (optional): If `--fastqc` is specified, the `FASTQ` rule would run FastQC on the input files and store the output under `FastQC` folder. If trimming is specified, FastQC is always produced on trimmed files, and stored under `FastQC_trimmed` folder.
• **--snakemakeOptions**: All wrappers contain a `--snakemakeOptions` parameter, which is quite useful as it can be used to pass on any arguments directly to snakemake. One use case is to perform a dry run, i.e. to check which programs would be executed and which outputs would be created by the workflow, without actually running it. This can be executed via `--snakemakeOptions="-np"`. This would also print the commands to be used during the run.

• **–DAG**: All workflows can produce a directed acyclic graph of themselves, using the `--DAG` option in the wrappers. This could be useful in reporting/presenting the results.

• **–keepTemp**: This option control temporary/intermediate files are to be kept after the workflow is finished. Normally the temporary files are removed after analysis.

• **–bwBinSize**: This option is available for most workflows, and refers to the bin size used to create the coverage files. BigWig files are created by most workflows in order to allow downstream analysis and visualization of outputs. This argument controls the size of the bins in which the genome is divided for creating this file. The default is sufficient for most analysis.

• **Temporary directory/files**: Some tools need additional space during runtime (eg. `samtools sort -T [DIR] ...`). SnakePipes uses the core tool `mktemp` to create temporary directories in some rules. On Linux-based systems the global env variable `$TMPDIR` is honored. On Mac OS and if `$TMPDIR` is empty, we fallback to `/tmp/` as the parent temporary directory. For performance reasons, it is recommended that the `$TMPDIR` points to a local drive (and not eg. an NFS share). Please make sure there is enough space!

### Logging of outputs

snakePipes produces logs at three different levels.

• **<workflow>.log**: This file would be generated on the working directory, and contains everything printed on the screen via snakemake and python wrappers.

• **<workflow>_organism.yaml**: This file is a copy of the YAML file specifying where all of the genomic indices, annotations, and other files are located.

• **cluster_logs**: In case snakePipes is setup with a cluster, the folder `cluster_logs` would contain the output and error messages from the cluster scheduler.

• **<output>/logs**: Each output folder from snakePipes workflows contain their own log (.err and .out) file under `/logs/` folder. This contains the messages directly from the executed tools.

**Note**: For most cases where a tool fails, these files contain useful debugging information. However sometimes, the error can’t be captured in these files and therefore ends up in the `cluster_logs` folder.

### Quality-Checks

All workflows under snakePipes employ various quality-checks (QC) to inform users of the data quality.

• **MultiQC**: All workflows in snakePipes output a `MultiQC` folder, which summerizes the QC metrics obtained from various tools in the workflow via `MultiQC`, in an interactive HTML report. This output is quite useful to compare samples and get an overview of the data quality from all samples.

• **deepTools**: `deepTools` are a popular set of tools that perform QC, normalization and visualization of NGS data. In snakePipes, most workflows (except HiC and scRNAseq) contain outputs from various deepTools modules on the samples. The coverage files (bigWigs), are also generated by deepTools (bamCoverage and bamCompare modules). Therefore, it’s useful to look at the deepTools documentation before inspecting these results.
Note: We strongly encourage users to understand these quality matrices and inspect the results from QC, before making biological conclusions or proceeding to downstream analysis.

code @ github.

6.3 Advanced usage of snakePipes

snakePipes is designed in a modular way, such that it’s easy to modify or extend it. Advanced users or developers can either use the underlying snakemake and Rscripts directly, or extend/add modules to the existing workflows.

6.3.1 Understanding snakePipes implementation

The implementation of snakePipes modules has been described in our preprint. Please clone our github repository locally to understand this organisation. Since snakePipes is dependent on snakemake and conda, we recommend being familiar with them first, by reading the documentation. Also, we utilize bioconda as a source of our biology-related tools implemented in snakePipes.

- Getting started with snakemake
- Getting started with conda
- How conda is used with snakemake
- What is bioconda

Once you are familiar with snakemake and conda/bioconda, we can look at how snakePipes workflows are implemented.

snakePipes folders

All files needed to be modified in order to extend/modify a workflow, are available under the snakePipes directory (snakepipes/snakePipes). Here is the structure of this directory:

```
  /snakePipes
  ├── common_functions.py
  │    └── __init__.py
  ├── parserCommon.py
  │    └── shared
  │      ├── cluster.yaml
  │      └── defaults.yaml
  │          └── organisms
  │             └── rscripts
  │                 └── rules
  │                     └── tools
  └── workflows
      ├── ATAC-seq
      ├── ChIP-seq
      └── createIndices
          ├── DNA-mapping
          └── HiC
              └── mRNA-seq
                  └── noncoding-RNA-seq
```

(continues on next page)
• **common_functions.py** contains functions that directly operate on the variables received via various wrappers.

• **parserCommon.py** contains common command-line arguments for the wrappers.

• **shared**: This folder contains some important files.
  - **cluster.yaml**: defines the command for the execution of rules on a cluster or cloud, this command is passed on to the call to snakemake via the wrappers.
  - **defaults.yaml**: defined the default options for snakemake and also defines the temporary directory to store intermediate files.
  - **organisms**: This folder contains yaml files corresponding to each organism (see Setting up snakePipes for details)
  - **rscripts**: Contains the R wrappers that are invoked via the rules. You would find the Rscripts for DESeq, CSAW and other R packages here.
  - **rules**: These are the snakemake rules which are invoked during execution of a workflow. Depending upon the global parameters passed on from the wrappers, a rule may/may not be included in a workflow (controlled by various if conditionals).

  **Warning**: Some rules are shared via multiple workflows, therefore be sure to check each Snakemake file for each workflow to see which rules you need to modify.

  - **tools**: This folder contains online tools which can not be distributed via bioconda, and therefore are included with snakePipes package itself.

• **workflows**: This folder contains files which are specific to each workflow. Under each folder (named by the workflow), you would find a common set of files.
  - **<workflow_name>.py**: The command line python wrappers that are visible to users after installation.
  - **Snakefile**: This is the file that collects outputs from various rules, therefore contains the rule all for each workflow. This Snakefile also controls which rules from shared/rules folder are included in the final workflow, depending on the global parameters passed on from the wrappers.
  - **internals.snakefile**: Contains some python functions which are specific to the workflow, and therefore can’t be included under common_functions.py, these functions are imported in the Snakefile
  - **cluster.yaml and defaults.yaml**: contains workflow-specific options for the cluster, and for the wrappers. Modify them to suite your needs.

### 6.3.2 Calling snakemake directly using the snakefiles

It’s possible to directly run snakemake using the Snakefile provided in each workflow, therefore surpassing the command-line wrappers. In order to do that, you can begin with a copy of `<workflow_name>.defaults.yaml` file that you will find in your output folder after running the workflow with --snakemakeOptions ‘--dryrun ’ and add or adjust further options in that file. This file will contained a merged dictionary from the workflow defaults as well as from the global (cross-workflows) defaults.

Finally, provide an adjusted config via --configfile parameter to snakemake!

example call:
6.3.3 Executing the Rscript wrappers outside snakePipes

It’s also possible to use one of our Rscript wrappers present under the shared/rscripts folder. In order to do that, check how the parameters are supplied to the wrappers in the corresponding rule.

For example, in order to execute the DESeq2 wrapper, we can look at how it’s done via the DESeq2 rule under shared/rules/DESeq2.Snakefile

example call:

```bash
cd DeSeq2_test &&
Rscript /path/to/shared/rscripts/DESeq2.R \$
{input.sample_info} \$
{input.counts_table} \$
{params.fdr} \$
{input.symbol_file} \$
{params.importfunc} \$
{params.allele_info} \$
{params.tx2gene_file} \$
{params.rmdTemplate}
```

Replace each variable by the corresponding required file. The required files are indicated in the DESeq2 rule.

6.3.4 Updating/adding new tools to the workflows

Several yaml files provided under the folder shared/rules/envs are used to define the tools which are executed via each workflow. Here is an example from the HiC conda env:

```yaml
name: hic_conda_env_1.0
channels:
- conda-forge
- anaconda
- bioconda
dependencies:
- hicexplorer = 2.1.4
- bwa = 0.7.17
- samtools = 1.8
- python-dateutil = 2.7.3
```

This file can be pointed out to the conda directive of any rule, under shared/rules. Example below

```python
rule get_restrictionSite:
    input:
        genome_fasta
    output:
        enzyme + ".bed"
    params:
        res_seq = get_restriction_seq(enzyme)
    conda: CONDA_HIC_ENV
```

(continues on next page)
Where CONDA_HIC_ENV points to the location of the above yaml file. Under snakePipes all such global variables are defined under common_functions.py

Therefore in order to change or upgrade a tool version, all you need to do is to edit the dependencies key in the yaml file to point to the new/modified tool version!

6.3.5 Modifying or adding new rules to the workflows

Modifying or adding new rules to snakePipes workflows is relatively easy. Considering you want to add a new Rscript that performs a downstream analysis on the DESeq2 output in mRNA-seq workflow. These would be the steps needed:

- Test the Rscript on command line first, then move it in the shared/rscripts folder.
- Add a rule that called the Rscript and put it under shared/rules folder.
- Add the corresponding rule all that defines the expected output into workflows/mRNA-seq/Snakefile
- Now, for easy and reproducible execution of the rule, add a conda directive and point it to the relevant conda env under shared/rules/envs. Since your rule might need a new R package, search whether it’s available in one of the conda channels and add the package name (as indicated in the conda channel) and version under the dependencies key.
- Finally, modify the command line wrapper (workflows/mRNA-seq/mRNA-seq) to make this new feature available to the users!

6.3.6 Using AWS or other cloud platforms

There is nothing particularly special about performing computations on AWS or other cloud platforms. Below are a few recommendations, using AWS as an example:

1. Use a small compute node for initial installation. On AWS a t2.small node is sufficient for general installation since conda will need 1-2GB RAM for dependency resolution during setup.
2. If you can need to create custom indices, then you will need a node with at least 80GB RAM and 10 cores.
3. Ensure that you install snakePipes on a separate EBS (or equivalent) storage block. We found that a 200GB /data partition was most convenient. This absolutely must not be the / partition, as mounting such a persistent image on other instances will result in paths being changed, which result in needing to modify large numbers of files.
4. It’s usually sufficient to use a single large (e.g., m5.24xlarge) compute node, with 100+ cores and a few hundred GB RAM. This allows one to use the --local option and not have to deal with the hassle of setting up a proper cluster on AWS. Make sure the then set -j to the number of available cores on the node, so snakePipes can make the most efficient use of the resources (and minimize your bill).

Below is an example of running the mRNA-seq pipeline on AWS using the resources outlined above. Note that it’s best to store your input/output data on a separate storage block, since its lifetime is likely to be shorter than that of the indices.
# Using a t2.small

```bash
sudo mkdir /data
mount /dev/sdf1 /data # /dev/sdf1 is a persistent storage block!
sudo chown ec2-user /data
cd /data

# get datasets
mkdir indices
wget https://zenodo.org/record/1475957/files/GRCm38_gencode_snakePipes.tgz?download=1
mv GRC* indices/GRCm38.tgz
cd indices
tar xf GRCm38.tgz
rm GRCm38.tgz
cd ..
mkdir data
wget some_data_url
mv snakePipes_files.tar data/
cd data
tar xf snakePipes_files.tar
rm snakePipes_files.tar
cd ..

# Edit the yaml file under indices to point to /data/indices

# Get conda
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh -b -p conda
export PATH=/data/conda/bin:$PATH
conda config --set always_yes yes --set changeps1 no
conda update -q conda
conda create -n snakePipes -c mpi-ie -c conda-forge -c bioconda snakePipes
conda activate snakePipes
rm Miniconda3-latest-Linux-x86_64.sh

# setup snakePipes
snakePipes createEnvs --only CONDA_SHARED_ENV CONDA_RNASEQ_ENV

# Update defaults.yaml to use /data/tmp for temporary space

Then a larger instance can be spun up and the *mRNA-seq* pipeline run as normal.

```bash
mkdir /data
mount /dev/sdf1 /data
chown ec2-user /data
export PATH=/data/snakePipes/bin:$PATH
conda activate snakePipes
mRNA-seq -m alignment -i /data/data -o /data/output --local -j 192 /data/indices/__GRCm28.yaml
```

## 6.3.7 Receiving emails upon pipeline completion

SnakePipes can send an email to the user once a pipeline is complete if users specify `--emailAddress`. In order for this to work, the following values need to be set in `defaults.yaml`:

- **smtpServer** The address of the outgoing SMTP server
- **smtpPort** The port on the SMTP server to use (0 means to use the standard port)
onlySSL  Set this to "True" if your SMTP server requires a full SSL connection from the beginning.

emailSender  The name of the "user" that sends emails (e.g., snakepipes@your-domain.com)

There are two additional parameters that can be set: smtpUsername and smtpPassword. These are relevant to SMTP servers that require authentication to send emails. On shared systems, it's important to ensure that other users cannot read your defaults.yaml file if it includes your password!

code @ github

6.4 createIndices

6.4.1 What it does

This is a special pipeline in that it creates index files required by various tools within snakePipes. This workflow takes as input a fasta file (or URL) and GTF file (or URL) as well as various optional files and generates both indices and the organism yaml file used by snakePipes.

6.4.2 Input requirements

The pipeline has two required inputs: a fasta file or URL and a GTF file or URL. These may both be gzipped. Optionally, you may specify a blacklist file (such as that provided by ENCODE), an effective genome size, and a file listing chromosomes to be ignored during normalization steps.

Note: If you specify a blacklist file, please ensure that regions within it do NOT overlap. Overlapping regions in this file will cause incorrect results in some tools. Further, it is best to flank blacklisted regions by at least 50 bases, as otherwise many reads originating within these regions may be nonetheless included.

Configuration file

There is a configuration file in snakePipes/workflows/createIndices/defaults.yaml:

```
pipeline: createIndices
outdir:
configFile:
clusterConfigFile:
local: false
maxJobs: 5
verbose: False
## Genome name used in snakePipes (no spaces!)
genome:
## Tools to create indices for. "all" for all of them
tools: all
```

(continues on next page)
## URLs or paths for fasta and GTF files

<table>
<thead>
<tr>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>genomeURL</td>
<td>Genome FASTA file URL</td>
</tr>
<tr>
<td>gtfURL</td>
<td>GTF file URL</td>
</tr>
</tbody>
</table>

## The effective genome size

- effectiveGenomeSize: 0

## Regions to blacklist in the ChIP-seq and related workflows

- blacklist: A list of regions to be excluded from analysis.

## Regions to ignore during normalization (e.g., with bamCompare)

- ignoreForNorm: A list of regions to ignore during normalization.

These values are most conveniently set on the command line.

### 6.4.3 Hybrid genome

To create a hybrid fasta, specify the host genome with `--genomeURL` and the spikein genome with `--spikeinGenomeURL`. On top of `--gtfURL` and `--blacklist`, you may optionally provide `--spikeinGtfURL` and `--spikeinBlacklist`. Default extension added to spikein chromosomes is `_spikein` and can be changes with `--spikeinExt`.

### 6.4.4 Output structure

The following structure will be created in the designated `outdir`:

```
./
  annotation
  |   blacklist.bed
  |   genes.bed
  |   genes.gtf
  |   genes.slop.gtf
  BowtieIndex
  BWAIndex
  BWAmethIndex
  create Indices.cluster_config.yaml
  create Indices.config.yaml
  create Indices_run-1.log
  genome fasta
  |   effectiveSize
  |   genome.2bit
  |   genome.fa
  |   genome.fa.fai
  HISAT2Index
  STARIndex
```

These files are used internally within snakePipes and don’t require further inspection. The `create Indices_run-1.log` file contains a full log and will include the URLs or file paths that you specified. Whether the `annotation/blacklist.bed` file exists is dependent upon whether you specified one. The `genome fasta/effectiveSize` fill will have the effective genome size (if you didn’t specify it, the number of non-N bases in the genome will be used).

In addition to these, an organism yaml file will be created. Its location can be found with `snakePipes info`.

**Note:** The astute observer will note that no Salmon index is created. This is intentional and done to facilitate users changing which transcripts should be included on the fly.
6.4.5 Command line options

Create indices for use by snakePipes. A YAML file will be created by default in the default location where snakePipes looks for organism YAML files.


```bash
usage: createIndices -o OUTDIR [-h] [-v] [-c CONFIGFILE]
                [---clusterConfigFile CLUSTERCONFIGFILE] [-j INT]
                [---local] [---keepTemp]
                [---smakemakeOptions SNAKEMAKEOPTIONS] [---DAG] [---version]
                [---emailAddress EMAILADDRESS] [---smtpServer SMTPSERVER]
                [---smtpPort SMTPPORT] [---onlySSL]
                [---emailSender EMAILSENDER] [---smtpUsername SMTPUSERNAME]
                [---smtpPassword SMTPPASSWORD] --genomeURL GENOMEURL
                [---gtfURL GTFURL] [---spikeinGenomeURL SPIKEINGENOMEURL]
                [---spikeinGtfURL SPIKEINGTFURL] [---spikeinExt SPIKEINEXT]
                [---tools {all,bowtie2,hisat2,bwa,bwameth,star,none} [all,
                --bowtie2,hisat2,bwa,bwameth,star,none] ...]
                [---effectiveGenomeSize EFFECTIVEGENOMESIZE]
                [---spikeinBlacklist SPIKEINBLACKLIST]
                [---blacklist BLACKLIST]
                [---ignoreForNormalization IGNOREFORNORMALIZATION]
                [---rmskURL RMSKURL] [---userYAML]
```

**Positional Arguments**

**GENOME**

The name to save this genome as. No spaces or special characters! Specifying an organism that already exists will cause the old information to be overwritten. See also the --userYAML option.

**Required Arguments**

- **-o, --output-dir**
  output directory

- **--genomeURL**
  URL or local path to where the genome fasta file is located. The file may optionally be gzipped.

- **--gtfURL**
  URL or local path to where the genome annotation in GTF format is located. GFF is NOT supported. The file may optionally be gzipped. If this file is not specified, then RNA-seq related tools will NOT be usable.

**General Arguments**

- **-v, --verbose**
  verbose output (default: 'False')

- **-c, --configFile**
  configuration file: config.yaml (default: 'None')

- **--clusterConfigFile**
  configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')
-j, --jobs maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')

--local run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')

--keepTemp Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.

--snakemakeOptions Snakemake options to be passed directly to snakemake, e.g. use --snakemakeOptions='--dryrun --rerun-incomplete --unlock --forceall'. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ['--use-conda'])

--DAG If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.

--version show program's version number and exit

Email Arguments

--emailAddress If specified, send an email upon completion to the given email address

--smtpServer If specified, the email server to use.

--smtpPort The port on the SMTP server to connect to. A value of 0 specifies the default port.

--onlySSL The SMTP server requires an SSL connection from the beginning.

--emailSender The address of the email sender. If not specified, it will be the address indicated by --emailAddress

--smtpUsername If your SMTP server requires authentication, this is the username to use.

--smtpPassword If your SMTP server requires authentication, this is the password to use.

Options

--spikeinGenomeURL URL or local path to where the spikein genome fasta file is located. The file may optionally be gzipped.

--spikeinGtfURL URL or local path to where the spikein genome annotation in GTF format is located. GFF is NOT supported. The file may optionally be gzipped.

--spikeinExt Extention of spikein chromosome names in the hybrid genome. (default: "None")

--tools Possible choices: all, bowtie2, hisat2, bwa, bwameth, star, none

Only produce indices for the following tools (by default, all indices will be created). The default is 'all'. 'none' will create everything except aligner indices.

--effectiveGenomeSize The effective genome size. If you don't specify a value then the number of non-N bases will be used.

--spikeinBlacklist An optional URL or local path to a file to use to blacklist spikein organism regions (such as that provided by the ENCODE consortium).
--blacklist An optional URL or local path to a file to use to blacklist regions (such as that provided by the ENCODE consortium).

--ignoreForNormalization An optional file list, with one entry per line, the chromosomes to ignore during normalization. These are typically sex chromosomes, mitochondrial DNA, and unplaced contigs.

--rmskURL URL or local path to where the repeat masker output file is located. This is only required if you plan to run the non-coding RNA-seq workflow.

--userYAML By default, this workflow creates an organism YAML file where snakePipes will look for it by default. If this isn’t desired (e.g., you don’t want the organism to be selectable by default or you don’t have write permissions to the snakePipes installation) you can specify this option and the YAML file will instead be created in the location specified by the -o option.

code @ github.

6.5 DNA-mapping

6.5.1 What it does

This is the primary DNA-mapping pipeline. It can be used both alone or upstream of the ATAC-seq and ChIP-seq pipelines. This has a wide array of options, including trimming and various QC steps (e.g., marking duplicates and plotting coverage and PCAs). In addition, basic coverage tracks are created to facilitate viewing the data in IGV.

6.5.2 Input requirements

The only requirement is a directory of gzipped fastq files. Files could be single or paired end, and the read extensions could be modified using the keys in the defaults.yaml file below.

Configuration file

There is a configuration file in snakePipes/workflows/DNA-mapping/defaults.yaml:
```yaml
## General/Snakemake parameters, only used/set by wrapper or in Snakemake cmdl, but not in Snakefile
pipeline: dna-mapping
outdir:
configFile:
clusterConfigFile:
local: False
maxJobs: 5
## directory with fastq files
indir:
## preconfigured target genomes (mm9,mm10,mm3,...) , see /path/to/snakemake_workflows/shared/organisms/
## Value can be also path to your own genome config file!
genome:
## FASTQ file extension (default: ".fastq.gz")
ext: ".fastq.gz"
## paired-end read name extension (default: ['_R1', '_R2'])
reads: ['_R1', '_R2']
## mapping mode
mode: mapping
aligner: Bowtie2
## Number of reads to downsample from each FASTQ file
downsample:
## Options for trimming
trim: False
trimmer: cutadapt
trimmerOptions:
## Bin size of output files in bigWig format
bwBinSize: 25
## Run FASTQC read quality control
fastqc: false
## Run computeGCBias quality control
GCBias: false
## Retain only de-duplicated reads/read pairs
dedup: false
## Retain only reads with at least the given mapping quality
mapq: 0
## Retain only reads mapping in proper pairs
properPairs: false
## Mate orientation in paired-end experiments for Bowtie2 mapping
mateOrientation: --fr
## other Bowtie2 stuff
insertSizeMax: 1000
alignerOpts:
plotFormat: png
UMIBarcode: False
bcPattern: NNNNCCCCCCCC #default: 4 base umi barcode, 8 base cell barcode (eg. RELACS_barcode)
UMIDedup: False
UMIDedupSep: "_"
UMIDedupOpts:
## Median/mean fragment length, only relevant for single-end data (default: 200)
fragmentLength: 200
qualimap: false
verbose: false
```

Many of these options can be more conveniently set on the command-line (e.g., `--qualimap` sets `qualimap`:
true). However, you may need to change the reads: setting if your paired-end files are not denoted by `sample_R1.fastq.gz` and `sample_R2.fastq.gz`, but rather `sample_1.fastq.gz` and `sample_2.fastq.gz`.

### 6.5.3 Understanding the outputs

The DNA mapping pipeline will generate output of the following structure:

```
<table>
<thead>
<tr>
<th></th>
<th>bamCoverage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bowtie2</td>
</tr>
<tr>
<td>deepTools_qc</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bamPEFragmentSize</td>
</tr>
<tr>
<td></td>
<td>estimateReadFiltering</td>
</tr>
<tr>
<td></td>
<td>multiBamSummary</td>
</tr>
<tr>
<td></td>
<td>plotCorrelation</td>
</tr>
<tr>
<td></td>
<td>plotCoverage</td>
</tr>
<tr>
<td></td>
<td>plotPCA</td>
</tr>
<tr>
<td>FASTQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FastQC</td>
</tr>
<tr>
<td></td>
<td>filtered_bam</td>
</tr>
<tr>
<td></td>
<td>multiQC</td>
</tr>
<tr>
<td></td>
<td>multiqc_data</td>
</tr>
<tr>
<td>Sambamba</td>
<td></td>
</tr>
</tbody>
</table>
```

In addition to the FASTQ module results (see *Running snakePipes*), the workflow produces the following outputs:

- **Bowtie2**: Contains the BAM files after mapping with Bowtie2 and indexed by Samtools.
- **filtered_bam**: Contains the BAM files filtered by the provided criteria, such as mapping quality (`--mapq`) or PCR duplicates (`--dedup`). This file is used for most downstream analysis in the DNA-mapping and ChIP-seq/ATAC-seq pipeline.
- **bamCoverage**: Contains the coverage files (bigWig format) produced from the BAM files by deepTools bamCoverage. The files are either raw, or 1x normalized (by sequencing depth). They are useful for plotting and inspecting the data in IGV.
- **deepTools_qc**: Contains various QC files and plots produced by deepTools on the filtered BAM files. These are very useful for evaluation of data quality. The folders are named after the tools. Please look at the deepTools documentation on how to interpret the outputs from each tool.
- **Sambamba**: Contains the alignment metrics evaluated on the BAM files by Sambamba.

A number of other directories may optionally be present if you specified read trimming, using Qualimap, or a variety of other options. These are typically self-explanatory.

A fair number of useful QC plots are or can be generated by the pipeline. These include correlation and PCA plots as well as the output from MultiQC.
6.5.4 Command line options

MPI-IE workflow for DNA mapping

**usage example:** DNA-mapping -i input-dir -o output-dir mm10

```bash
usage: DNA-mapping -i INDIR -o OUTDIR [-h] [-v] [-ext EXT]
        [-reads READS READS] [-c CONFIGFILE]
        [--clusterConfigFile CLUSTERCONFIGFILE] [-j INT] [--local]
        [--keepTemp] [--snakemakeOptions SNAKEMAKEOPTIONS] [--DAG]
        [--version] [--emailAddress EMAILADDRESS]
        [--smtpServer SMTPSERVER] [--smtpPort SMTPPORT] [--onlySSL]
        [--emailSender EMAILSENDER] [--smtpUsername SMTPUSERNAME]
        [--smtpPassword SMTPPASSWORD] [--VCFfile VCFFILE]
        [--strains STRAINS] [--SNPfile SNPFILE]
        [--NMaskedIndex NMASKEDINDEX] [-m MODE] [--downsample INT]
        [--trim] [--trimmer {cutadapt,trimgalore,fastp}]
        [--trimmerOptions TRIMMEROPTIONS] [--fastqc] [--bcExtract]
        [--bcPattern BCPATTERN] [--UMIDedup]
```

(continues on next page)
Positional Arguments

**GENOME**

Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_genencodeM13, mm9, GRCz10, hs37d5, dm6

Required Arguments

- **-i, --input-dir**
  input directory containing the FASTQ files, either paired-end OR single-end data
- **-o, --output-dir**
  output directory

General Arguments

- **-v, --verbose**
  verbose output (default: 'False')
- **--ext**
  Suffix used by input fastq files (default: "'.fastq.gz'").
- **--reads**
  Suffix used to denote reads 1 and 2 for paired-end data. This should typically be either '_1' '_2' or '_R1' '_R2' (default: ['_R1', '_R2']). Note that you should NOT separate the values by a comma (use a space) or enclose them in brackets.
- **-c, --configFile**
  configuration file: config.yaml (default: 'None')
- **--clusterConfigFile**
  configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')
- **-j, --jobs**
  maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')
- **--local**
  run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')
- **--keepTemp**
  Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.
- **--snakemakeOptions**
  Snakemake options to be passed directly to snakemake, e.g. `use --snakemakeOptions='--dryrun --rerun-incomplete --unlock --forceall'`. **WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN!** (default: [''--use-conda''])
- **--DAG**
  If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.

6.5. DNA-mapping
--version
show program’s version number and exit

Email Arguments

--emailAddress
If specified, send an email upon completion to the given email address

--smtpServer
If specified, the email server to use.

--smtpPort
The port on the SMTP server to connect to. A value of 0 specifies the default port.

--onlySSL
The SMTP server requires an SSL connection from the beginning.

--emailSender
The address of the email sender. If not specified, it will be the address indicated by --emailAddress

--smtpUsername
If your SMTP server requires authentication, this is the username to use.

--smtpPassword
If your SMTP server requires authentication, this is the password to use.

Allele-specific mapping arguments

--VCFfile
VCF file to create N-masked genomes (default: ‘None’)

--strains
Name or ID of SNP strains separated by comma (default: ‘None’)

--SNPfile
File containing SNP locations (default: ‘None’)

--NMaskedIndex
N-masked index of the reference genome (default: ‘None’)

Options

-m, --mode
workflow running modes (available: ‘mapping,allelic-mapping’) (default: ‘”mapping”’)

--downsample
Downsample the given number of reads randomly from of each FASTQ file (default: ‘False’)

--trim
Activate fastq read trimming. If activated, Illumina adaptors are trimmed by default. Additional parameters can be specified under --trimmerOptions. (default: ‘False’)

--trimmer
Possible choices: cutadapt, trimgalore, fastp

Trimming program to use: Cutadapt, TrimGalore, or fastp. Note that if you change this you may need to change --trimmerOptions to match! (default: ‘”cutadapt”’)

--trimmerOptions
Additional option string for trimming program of choice. (default: ‘”’)

--fastqc
Run FastQC read quality control (default: ‘False’)

--bcExtract
To extract umi barcode from fastq file via UMI-tools and add it to the read name (default: ‘False’)

--bcPattern
The pattern to be considered for the barcode. ‘N’ = UMI position (required) ‘C’ = barcode position (optional) (default: ‘”NNNCCCCCCCC”’)

--UMIDedup
Deduplicate bam file based on UMIs via umi_tools dedup that are present in the read name. (default: ‘False’)

36 Chapter 6. Contents:
6.6 ChIP-seq

6.6.1 What it does

The ChIP-seq pipeline takes one or more BAM files and attempts to find peaks. If multiple samples and a sample sheet are provided, then CSAW is additionally used to call differential peaks. Both sharp and broad peak calling are supported.
In addition to peaks, bigWig tracks are also generated.

### 6.6.2 Input requirements

The DNA mapping pipeline generates output that is fully compatible with the ChIP-seq pipeline input requirements! When running the ChIP-seq pipeline, please specify the output directory of DNA-mapping pipeline as the working directory (`-w`).

If you need to provide file NOT generated by the DNA-mapping pipeline, then you must provide a directory with the following structure:

```
.  
  deepTools_qc
    bamPEFragmentSize
      fragmentSize.metric.tsv
      fragmentSizes.png
    filtered_bam
      sample1.filtered.bam
      sample1.filtered.bam.bai
      sample2.filtered.bam
      sample2.filtered.bam.bai
    Sambamba
      flagstat_report_all.tsv
      sample1.markdup.txt
      sample2.markdup.txt
    sampleSheet.yaml
```

- **deepTools_qc** contains the output of `bamPEFragmentSize` from deepTools, run on all the BAM files.
- **Sambamba** directory contains the output of `flagstat` command from sambamba (the `.markdup.txt` files) and a single file summarizing that with columns `sample` (sample name, such as `sample1`), `total` (total reads), `dup` (number of duplicate reads), and `mapped` (number of mapped reads).
- **filtered_bam** directory contains the input BAM files (either filtered or unfiltered, however you prefer).
- **sampleSheet.tsv** (OPTIONAL) is only needed to test for differential binding.
Sample configuration

The ChIP-seq sample configuration yaml file describes what type of peak calling to perform on each sample and which sample to use as the input control

```
chip_dict:
  SRR6761497:
    control: SRR6761502
    broad: True
  SRR6761498:
    control: SRR6761502
    broad: True
  SRR6761495:
    control: SRR6761502
    broad: False
  SRR6761499:
    control: SRR6761502
    broad: False
```

As you can see above, the same control can be used for multiple samples.

**Note:** Set the flag broad to True for broad marks, such as H3K27me and H3K9me3

Spikein Normalization

If chromatin from an external organism was spikein in, it is possible to obtain spikein-derived scaling factors for the ChIP (and input) samples with the flag --useSpikeInForNorm. This requires providing a hybrid bam file, with reads aligned to a hybrid genome of host and spikein chromosomes. Spikein chromosome extention can be specified with --spikeinExt. Scale factors can be obtained either from whole spikein genome in the ChIP samples, from windows centered on TSS in the spikein genome in the ChIP samples, or from whole spikein genome in the input samples. The default scale factors from whole spikein genome in the ChIP samples can be changed to something else with --getSizeFactorsFrom.

DESeq2-style scaling factors produced with deepTools multiBamSummary will then be used to create bam coverage tracks and passed to CSAW as size Factors if sample sheet is provided.

A hybrid genome can be obtained with createIndices workflow and can be passed to the DNA-mapping workflow without any particular arguments.

Differential Binding analysis

If you wish to perform differential binding analysis between two group of samples, for example wild-type vs Knockouts, via snakePipes. You would require a sample-sheet and the --sampleSheet option. Sample sheet may contain only a subset of samples used in the previous steps e.g. for peak calling. In addition, input samples are filtered out prior to the analysis using the sample configuration yaml (see above).

The sample sheet is a tab-separated file with two columns, named name and condition. An example is below:

```
name   condition
sample1 wild-type
sample2 wild-type
SRR7013047 wild-type
SRR7013048 mutant
```

(continues on next page)
For comparison between two conditions, the name you assign to "condition" is not relevant, but rather the order is. The group mentioned first (in the above case "wild-type") would be used as a "control" and the group mentioned later would be used as "test".

The differential binding module utilizes the R package CSAW to detect significantly different peaks between two conditions. The analysis is performed on a "union" of peaks from all samples mentioned in the sample sheet. This merged set of regions are provided as an output inside the CSAW folder as the file 'DiffBinding_allregions.bed'. All differentially bound regions are available in 'CSAW/DiffBinding_significant.bed'. Two thresholds are applied to produce Filtered.results.bed: FDR (default 0.05) as well as absolute log fold change (1). These can be specified either in the defaults.yaml dictionary or via commandline parameters '-FDR' and '-LFC'. Additionally, filtered results are split into up to 3 bed files, representing direction change (UP, DOWN, or MIXED).

If the user provides additional columns between 'name' and 'condition' in the sample sheet, the variables stored there will be used as blocking factors in the order they appear in the sample sheet. Condition will be the final column and it will be used for any statistical inference.

**Note:** In order to include or exclude peaks from selected samples in the union of peaks used in the differential binding analysis, the user may provide an additional column named 'UseRegions' and set it to True or False, accordingly. This column must supersede the 'condition' column in the column order.

Merged regions from filtered results with any direction change are further used to produce deepTools heatmaps, using log2 ratio of chip signal to input or depth-normalized coverage. For this purpose, the regions are rescaled to 1kb, and extended by 0.2kb on each side.

An html report summarizing the differential binding analysis is produced in the same folder.

Filtered results are also annotated with the distance to the closest gene using bedtools closest and written as '.txt' files to the AnnotatedResults_* folder.

**Configuration file**

There is a configuration file in `snakePipes/workflows/ChIP-seq/defaults.yaml`:

```yaml
pipeline: chip-seq
configFile:
clusterConfigFile:
local: false
maxJobs: 5

## workingdir need to be required DNA-mapping output dir, 'outdir' is set to workingdir internally
workingdir:

## preconfigured target genomes (mm9,mm10,dm3,...) , see /path/to/snakemake_workflows/shared/organisms/
genome:

## Which peak caller should be used?
peakCaller: 'MACS2'

## paired end data?
pairedEnd: true

## Bin size of output files in bigWig format
bwBinSize: 25
```

(continues on next page)
## Median/mean fragment length, only relevant for single-end data (default: 200)

```bash
fragmentLength: 200
verbose: false
```

# sampleInfo_DB

```bash
sample_info:
```

# windowSize

```bash
windowSize: 150
```

```bash
plot_format: png
```

## #dummy string to skip filtering annotation

```bash
filter_annotation:
```

## #parameters to filter DB regions on

```bash
fdr: 0.05
absBestLFC: 1
```

The only parameters that are useful to change are `bwBinSize`, `fragmentLength`, and `windowSize`. Note however that those can be more conveniently changed on the command line.

### 6.6.3 Understanding the outputs

The ChIP-seq pipeline will generate additional output as follows:

```
depthTools_ChIP
  bamCompare
    sample1.filtered.log2ratio.over_SRR6761502.bw
    sample1.filtered.subtract.SRR6761502.bw
    sample2.filtered.log2ratio.over_SRR6761502.bw
    sample2.filtered.subtract.SRR6761502.bw
  plotFingerprint
    plotFingerprint.metrics.txt
    plotFingerprint.png

histoneHMM
  sample2.filtered.histoneHMM-em-posterior.txt.gz
  sample2.filtered.histoneHMM-regions.gff.gz
  sample2.filtered.histoneHMM-regions.gff.gz.tbi
  sample2.filtered.histoneHMM.txt.gz
  sample2.filtered.histoneHMM-zinba-emfit.pdf
  sample2.filtered.histoneHMM-zinba-params-em.RData
  sample2.filtered.histoneHMM-zinba-params-em.txt

Genrich
  sample2.narrowPeak

MACS2
  sample1.filtered.BAM_peaks.narrowPeak
  sample1.filtered.BAM_peaks.qc.txt
  sample1.filtered.BAM_peaks.xls
  sample1.filtered.BAMPE_peaks.narrowPeak
  sample1.filtered.BAMPE_peaks.xls
  sample1.filtered.BAMPE_summits.bed
  sample1.filtered.BAM_summits.bed
  sample2.filtered.BAM_peaks.broadPeak
  sample2.filtered.BAM_peaks.gappedPeak
  sample2.filtered.BAM_peaks.qc.txt
  sample2.filtered.BAM_peaks.xls
  sample2.filtered.BAMPE_peaks.broadPeak
  sample2.filtered.BAMPE_peaks.gappedPeak
  sample2.filtered.BAMPE_peaks.xls
```
Following up on the DNA-mapping module results (see *DNA-mapping*), the workflow produces the following output directories:

- **deepTools_ChIP**: Contains output from two of the deepTools modules. The `bamCompare` output contains the input-normalized coverage files for the samples, which is very useful for downstream analysis, such as visualization in IGV and plotting the heatmaps. The `plotFingerPrint` output is a useful QC plot to assess signal enrichment in the ChIP samples.

- **Genrich**: This folder contains the output of Genrich. This will only exist if you specified `--peakCaller Genrich` and you have samples with non-broad peaks. The output is in narrowPeak format, like that from MACS2.

- **MACS2**: This folder contains the output of MACS2 on the ChIP samples, MACS2 would perform either a narrow or broad peak calling on the samples, as indicated by the ChIP sample configuration file (see *Configuration file*). The outputs files would contain the respective tags (narrowPeak or broadPeak). This folder will only exist if you have non-broad marks and use MACS2 for peak calling.

- **histoneHMM**: This folder contains the output of histoneHMM. This folder will only exist if you have broad marks.

- **CSA\_W\_sampleSheet**: This folder is created optionally, if you provide a sample sheet for differential binding analysis. (see *Differential Binding analysis*).

- **AnnotatedResults\_sampleSheet**: This folder is created optionally, if you provide a sample sheet for differential binding analysis. (see *Differential Binding analysis*). Differentially bound regions annotated with distance to nearest gene are stored here.

**Note:** Although in case of broad marks, we also perform the MACS2 broadpeak analysis (output available as `MACS2/<sample>.filtered.BAM_peaks.broadPeak`), we would recommend using the histoneHMM outputs in these cases, since histoneHMM produces better results than MACS2 for broad peaks.

**Note:** The _sampleSheet suffix for the CSAW\_sampleSheet is drawn from the name of the sample sheet you use. So if you instead named the sample sheet `mySampleSheet.txt` then the folder would be named CSAW\_mySampleSheet. This facilitates using multiple sample sheets.

**Note:** At the moment Genrich is NOT jointly calling peaks within a group since it’s not aware of which samples contain which antibody. It is utilizing the input control if one exists.

### 6.6.4 Command line options

MPI-IE workflow for ChIP-seq analysis

**Usage example:** ChIP-seq -d working-dir mm10 samples.yaml
Positional Arguments

**GENOME**
Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_gencodeM13, mm9, GRCz10, hs37d5, dm6

**SAMPLESCONFIG**
configuration file (eg. `example.chip_samples.yaml`) with sample annotation

Required Arguments

--- **-d, --working-dir**
working directory is output directory and must contain DNA-mapping pipeline output files

General Arguments

--- **-v, --verbose**
verbose output (default: 'False')

--- **-c, --configFile**
configuration file: config.yaml (default: 'None')

--- **--clusterConfigFile**
configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')

--- **-j, --jobs**
maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')

--- **--local**
run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')

--- **--keepTemp**
Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.

--- **--snakemakeOptions**
Snakemake options to be passed directly to snakemake, e.g. use –snakemakeOptions='–dryrun –rerun-incomplete –unlock –forceall’. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ‘[‘–use-conda’]’)

--- **--DAG**
If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.

--- **--version**
show program’s version number and exit

6.6. ChIP-seq
Email Arguments

--emailAddress
If specified, send an email upon completion to the given email address

--smtpServer
If specified, the email server to use.

--smtpPort
The port on the SMTP server to connect to. A value of 0 specifies the default port.

--onlySSL
The SMTP server requires an SSL connection from the beginning.

--emailSender
The address of the email sender. If not specified, it will be the address indicated by --emailAddress

--smtpUsername
If your SMTP server requires authentication, this is the username to use.

--smtpPassword
If your SMTP server requires authentication, this is the password to use.

Options

--peakCaller
Possible choices: MACS2, Genrich
The peak caller to use. The default is "MACS2" and this is only applicable for sharper peaks (broad peaks will always use histoneHMM).

--singleEnd
Input data is single-end, not paired-end

--useSpikeInForNorm
Use the spikeIn chromosomes of the hybrid genome for normalization.

--getSizeFactorsFrom
Possible choices: genome, TSS, input
Which part of the spikein genome to use to calculate sizeFactors from.

--spikeinExt
Extension of spikein chromosome names in the hybrid genome. Ignored if useSpikeInForNorm is False (default: ""spikein"").

--bigWigType
Type of bigWig file to create. Options are: 'subtract' (control-subtracted ChIP coverage), 'log2ratio' (for log2 ratio of ChIP over control) or 'both' (create both set of bed files). Note that the allele-specific mode currently only produces 'log2ratio' bigwigs. (default: "both")

--fragmentLength
Fragment length in sequencing. Used only if --singleEnd (default: '200')

--bwBinSize
bin size of output files in bigWig format (default: '25')

--qval
qvalue threshold for MACS2 (default: '0.001')

--sampleSheet
Information on samples (If differential binding analysis required); see 'https://github.com/maxplanck-ie/snakepipes/tree/master/docs/content/sampleSheet.example.tsv' for example. IMPORTANT: The first entry defines which groups of samples are control. By this, the order of comparison and likewise the sign of values can be changed! Also, the condition control should only be used for input samples (control peaks are not evaluated for differential binding) (default: ")

--windowSize
Window size to counts reads in (If differential binding analysis required); Default size is suitable for most transcription factors and sharp histone marks. Small window sizes (~20bp) should be used for very narrow transcription factor peaks, while large window sizes (~500 bp) should be used for broad marks (eg. H3K27me3) (default: '150')
--predictChIPDict Use existing bam files to predict a CHiP-seq sample configuration file. Write it to the workingdir. If no value is given, samples that contain 'input' are used as ChIP input/ctrl. Provide a custom pattern like ‘input,H3$,.H4$’ to change that!

--fromBAM Input folder with bam files. If provided, the analysis will start from this point. If bam files contain single ends, please specify --singleEnd additionally. (default: 'False')

--bamExt Extention of provided bam files, will be substracted from basenames to obtain sample names. (default: ""filtered.bam")

--plotFormat Possible choices: png, pdf, None
Format of the output plots from deepTools. Select 'none' for no plots (default: '"png")

--mfold Select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. Fold-enrichment in regions must be lower than upper limit, and higher than the lower limit. Use as ":-m 10 30". This setting is only used while building the shifting model. Tweaking it is not recommended. DEFAULT:5 50 (default: ""0 50")

--FDR FDR threshold to apply for filtering DB regions (default: '0.05')

--LFC Log fold change threshold to apply for filtering DB regions (default: '1')

code @ github.

6.7 ATAC-seq

6.7.1 What it does

The ATAC-seq pipeline takes one or more BAM files and attempts to find accessible regions. If multiple samples and a sample sheet are provided, then CSAW is additionally used to find differentially accessible regions. Prior to finding open/accessible regions, the BAM files are filtered to include only properly paired reads with appropriate fragment sizes (<150 bases by default). These filtered fragments are then used for the remainder of the pipeline.
6.7.2 Input requirements

The DNA mapping pipeline generates output that is fully compatible with the ATAC-seq pipeline input requirements! When running the ATAC-seq pipeline, please specify the output directory of DNA-mapping pipeline as the working directory (-d).

- **filtered_bam** directory contains the input BAM files (either filtered or unfiltered, however you prefer).
- **sampleSheet.tsv** (OPTIONAL) is only needed to test for differential binding.

Differential open chromatin analysis

Similar to differential binding analysis with the ChIP-Seq data. We can perform the differential open chromatin analysis, using the --sampleSheet option of the ATAC-seq workflow. This requires a sample sheet, which is

---

**Note:** The CSAW step will be skipped if there is no sample_info tsv file (see Running snakePipes).
identical to that required by the ChIP-seq and RNA-seq workflows (see ChIP-seq for details).

An example is below:

<table>
<thead>
<tr>
<th>name</th>
<th>condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample1</td>
<td>eworo</td>
</tr>
<tr>
<td>sample2</td>
<td>eworo</td>
</tr>
<tr>
<td>SRR7013047</td>
<td>eworo</td>
</tr>
<tr>
<td>SRR7013048</td>
<td>OreR</td>
</tr>
<tr>
<td>SRR7013049</td>
<td>OreR</td>
</tr>
<tr>
<td>SRR7013050</td>
<td>OreR</td>
</tr>
</tbody>
</table>

Note: This sample sheet has the same requirements as the sample sheet in the ChIP-seq workflow, and also uses the same tool (CSAW) with a narrow default window size.

For comparison between two conditions, the name you assign to "condition" is not relevant, but rather the order is. The group mentioned first (in the above case "wild-type") would be used as a "control" and the group mentioned later would be used as "test".

If the user provides additional columns between 'name' and 'condition' in the sample sheet, the variables stored there will be used as blocking factors in the order they appear in the sample sheet. Condition will be the final column and it will be used for any statistical inference.

The differential binding module utilizes the R package CSAW to detect significantly different peaks between two conditions. The analysis is performed on a union of peaks from all samples mentioned in the sample sheet. This merged set of regions are provided as an output inside the CSAW_MACS2_sampleSheet folder as the file 'Diff-Binding_allregions.bed'. All differentially bound regions are available in 'CSAW/DiffBinding_significant.bed'. Two thresholds are applied to produce Filtered.results.bed: FDR (default 0.05) as well as absolute log fold change (1). These can be specified either in the defaults.yaml dictionary or via commandline parameters ‘-FDR’ and ‘-LFC’. Additionally, filtered results are split into up to 3 bed files, representing direction change (UP, DOWN, or MIXED).

Note: In order to include or exclude peaks from selected samples in the union of peaks used in the differential binding analysis, the user may provide an additional column named 'UseRegions' and set it to True or False, accordingly. This column must supersede the 'condition' column in the column order.

Merged regions from filtered results with any direction change are further used to produce deepTools heatmaps, using depth-normalized coverage. For this purpose, the regions are rescaled to 1kb, and extended by 0.2kb on each side.

An html report summarizing the differential binding analysis is produced in the same folder.

Filtered results are also annotated with the distance to the closest gene using bedtools closest and written as '.txt' files to the AnnotatedResults_* folder.

Configuration file

There is a configuration file in snakePipes/workflows/ATACseq/defaults.yaml:

```yaml
## General/Snakemake parameters, only used/set by wrapper or in Snakemake cmdl, but not in Snakefile
pipeline: ATAC-seq
configFile: clusterConfigFile:
local: false
```

(continues on next page)
maxJobs: 5
## workingdir need to be required DNA-mapping output dir, 'outdir' is set to
→ workingdir internally
workingdir:
## preconfigured target genomes (mm9, mm10, dm3,...), see /path/to/snakemake_workflows/
→ shared/organisms/
## Value can be also path to your own genome config file!
genome:
## The maximum fragment size to retain. This should typically be the size of a
→ nucleosome
maxFragmentSize: 150
minFragmentSize: 0
verbose: false
## which peak caller to use
peakCaller: 'MACS2'
# sampleSheet_DB
sampleSheet:
# windowSize
windowSize: 20
fragmentCountThreshold: 1
### Flag to control the pipeline entry point
bamExt: '.filtered.bam'
fromBAM:
## Bin size of output files in bigWig format
bwBinSize: 25
pairedEnd: True
plotFormat: png
## Median/mean fragment length, only relevant for single-end data (default: 200)
fragmentLength:
trim:
fastqc:
qval: 0.001
##dummy string to skip filtering annotation
filter_annotation:
##parameters to filter DB regions on
fdr: 0.05
absBestLFC: 1

Useful parameters are maxFragmentSize, minFragmentSize and windowSize, also available from commandline.

- **windowSize**: is the size of windows to test differential binding using CSAW. The default small window size is sufficient for most analysis, since an ATAC-seq peak is sharp.
- **fragmentCountThreshold**: refers to the minimum number of counts a chromosome must have to be included in the MACS2 analysis. It is introduced to avoid errors in the peak calling step and should only be changed if MACS2 fails.
- **Qval**: a value provided to MACS2 that affects the number and width of the resulting peaks.

### 6.7.3 Understanding the outputs

Assuming a sample sheet is used, the following will be **added** to the working directory:

```bash
| CSAW_MACS2_sampleSheet
```

(continues on next page)
Currently the ATAC-seq workflow performs detection of open chromatin regions via MACS2 (or HMMRATAC or Genrich, if specified with `--peakCaller`), and if a sample sheet is provided, the detection of differential open chromatin sites via CSAW. There are additionally log files in most of the directories. The various outputs are documented in the CSAW and MACS2 documentation. For more information on the contents of the `CSAW_MACS2_sampleSheet` folder, see section [Differential open chromatin analysis](#).

- **MACS2 / HMMRATAC / Genrich**: Contains peaks found by the peak caller. The most useful files end in `.narrowPeak` or `.gappedPeak` and are appropriate for visualization in IGV.
- **MACS2_QC**: contains a number of QC metrics that we find useful, namely:
  - the number of peaks
  - fraction of reads in peaks (FRiP)
  - percentage of the genome covered by peaks.
- **deepTools_ATAC**: contains the output of `plotFingerPrint`, which is a useful QC plot to assess signal enrichment between the ATAC-seq samples.
Note: The _sampleSheet suffix for the CSAW_MACS2_sampleSheet is drawn from the name of the sample sheet you use. So if you instead named the sample sheet mySampleSheet.txt then the folder would be named CSAW_mySampleSheet. This facilitates using multiple sample sheets. Similarly, _MACS2 portion will be different if you use HMMRATAC or Genrich for peak calling.

Note: The output from Genrich will be peaks called per-group if you specify a sample sheet. This is because Genrich is capable of directly using replicates during peak calling.

6.7.4 Where to find final bam files and biwigs

Bam files with the extention filtered.bam are only filtered for PCR duplicates. The final bam files filtered additionally for fragment size and used as direct input to MACS2 are found in the short_bams folder with the exention .short.cleaned.bam. Bigwig files calculated from these bam files are found under deepTools_ATAC/bamCompare with the extension .filtered.bw.

6.7.5 Command line options

MPI-IE workflow for ATAC-seq Analysis

usage example: ATAC-seq -d working-dir mm10

usage: ATAC-seq -d WORKINGDIR [-h] [-v] [-c CONFIGFILE]
    [--clusterConfigFile CLUSTERCONFIGFILE] [-j INT] [--local]
    [--keepTemp] [--snakemakeOptions SNAKEMAKEOPTIONS] [--DAG]
    [--version] [--emailAddress EMAILADDRESS]
    [--smtpServer SMTPSERVER] [--smtpPort SMTPPORT] [--onlySSL]
    [--emailSender EMAILSENDER] [--smtpUsername SMTPUSERNAME]
    [--smtpPassword SMTPPASSWORD]
    [--peakCaller {MACS2,HMMRATAC,Genrich}]
    [--maxFragmentSize MAXFRAGMENTSIZE]
    [--minFragmentSize MINFRAGMENTSIZE] [--qval INT]
    [--sampleSheet SAMPLESHEET] [--fromBAM FROMBAM]
    [--bamExt BAMEXT] [--FDR FDR] [--LFC ABSBESTLFC]

Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_gencodeM13, mm9, GRCz10, hs37d5, dm6

Required Arguments

-d, --working-dir working directory is output directory and must contain DNA-mapping pipeline output files
General Arguments

- **-v, --verbose**
  verbose output (default: 'False')

- **-c, --configFile**
  configuration file: config.yaml (default: 'None')

- **--clusterConfigFile**
  configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')

- **-j, --jobs**
  maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')

- **--local**
  run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')

- **--keepTemp**
  Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.

- **--snakemakeOptions**
  Snakemake options to be passed directly to snakemake, e.g. use –snakemakeOptions=‘–dryrun –rerun-incomplete –unlock –forceall’. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ['–use-conda'])

- **--DAG**
  If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.

- **--version**
  show program’s version number and exit

Email Arguments

- **--emailAddress**
  If specified, send an email upon completion to the given email address

- **--smtpServer**
  If specified, the email server to use.

- **--smtpPort**
  The port on the SMTP server to connect to. A value of 0 specifies the default port.

- **--onlySSL**
  The SMTP server requires an SSL connection from the beginning.

- **--emailSender**
  The address of the email sender. If not specified, it will be the address indicated by --emailAddress

- **--smtpUsername**
  If your SMTP server requires authentication, this is the username to use.

- **--smtpPassword**
  If your SMTP server requires authentication, this is the password to use.

Options

- **--peakCaller**
  Possible choices: MACS2, HMMRATAc, Genrich
  The peak caller to use. The default is "MACS2"

- **--maxFragmentSize**
  Maximum size of (typically nucleosomal) fragments for inclusion in the analysis (default: '150')

- **--minFragmentSize**
  Minimum size of (typically nucleosomal) fragments for inclusion in the analysis (default: '0')
--qval qvalue threshold for MACS2 (default: '0.001')

--sampleSheet Invoke differential accessibility analysis by providing information on samples; see 'https://github.com/maxplanck-ie/snakepipes/tree/master/docs/content/sampleSheet.example.tsv' for example. IMPORTANT: The first entry defines which group of samples are control. With this, the order of comparison and likewise the sign of values can be changed! Also, the condition control should not be used (reserved to mark input samples in the ChIP-Seq workflow (default: 'None').

--fromBAM Input folder with bam files. If provided, the analysis will start from this point. (default: 'False')

--bamExt Extension of provided bam files, will be subtracted from basenames to obtain sample names. (default: "filtered.bam")

--FDR FDR threshold to apply for filtering DB regions (default: '0.05')

--LFC Log fold change threshold to apply for filtering DB regions (default: '1')

code @ github.

6.8 HiC

6.8.1 What it does

The snakePipes HiC workflow allows users to process their HiC data from raw fastq files to corrected HiC matrices and TADs. The workflow utilized mapping by BWA, followed by analysis using HiCExplorer. The workflow follows the example workflow described in the documentation of HiCExplorer, which explains each step in detail and would be useful for new users to have a look at. The output matrices are produced in the .hdf5 format.
6.8.2 Input requirements and outputs

This pipeline requires paired-end reads fastq files as input in order to build a contact matrix and to call TADs. Prior to building the matrix, the pipeline maps reads against a user-specified reference genome. The output of mapping step is then used for building the contact matrix.

6.8.3 Workflow configuration file

Default parameters from the provided config file can be altered by user. Below is the config file description for the HiC workflow:

```plaintext
# This file is the default configuration of the HiC workflow!

# In order to adjust some parameters, please either use the wrapper script
# (eg. /path/to/snakemake_workflows/workflows/HiC/HiC)
# or save a copy of this file, modify necessary parameters and then provide
# this file to the wrapper or snakemake via '--configFile' option
# (see below how to call the snakefile directly)
#
# Own parameters will be loaded during snakefile execution as well and hence
# can be used in new/extended snakemake rules!

## General/Snakemake parameters, only used/set by wrapper or in Snakemake cmdl, but
## not in Snakefile
pipeline: hic
outdir: 
configFile: 
clusterConfigFile: 
#set to true if running locally
local: False
#number of threads
maxJobs: 5

## directory with fastq files
indir: 

## preconfigured target genomes (mm9,mm10,dm3,...), see /path/to/snakemake_workflows/
##_shared/organisms/
## Value can be also path to your own genome config file!
genome: 

## FASTQ file extension (default: ".fastq.gz")
ext: ".fastq.gz"

## paired-end read name extension (default: ['_R1', '_R2'])
reads: ['_R1', '_R2']

## aligner
aligner: BWA

## Reducer number of reads to downsample from each FASTQ file
downsample: 

## Options for trimming and fastqc
trim: False
trimmer: cutadapt
trimmerOptions: 
fastqc: false
verbose: False

## is the Matrix RF resolution?
RFResolution: false

## which restriction enzyme was used
enzyme: HindIII
```

(continues on next page)
6.8.4 Structure of output directory

In addition to the FASTQ module results (see Running snakePipes), the workflow produces the following outputs:

|--BWA
|--FASTQ
|--HiC_matrices

(continues on next page)
| |--logs | |--QCplots | --HiC_matrices_corrected | |--logs | --TADs | |--logs |

- **BWA** folder contains the mapping results in BAM format. The files were obtained after running **BWA** on each of the paired-end reads individually.

- **HiC_matrices** folder accommodates the contact matrices generated by **hicBuildMatrix**. In case of merging samples or merging bins the initial matrix is saved in this folder along with the merged ones.
  
  - **QCplot** includes the QC measurements for each sample along with a diagnostic plot which illustrates a distribution of counts per bin. This information can be used to set a cutoff to prune (correct) the contact matrix.

**Note:** The cutoff value is computed by the pipeline and by default will be applied to build a corrected matrix. Generated matrices by the pipeline can further be used for downstream analysis such as detecting A/B compartments and they can also be visualized using **hicPlotMatrix**.

- **HiC_matrices_corrected** folder is in fact containing the corrected matrix which has been generated via **hicCorrectMatrix** after pruning as has been mentioned above.

- **TADs** folder includes the output of calling TADs using **hicFindTADs**. The output contains TAD boundaries, TAD domains and TAD scores. These along with the matrices can be visualized together as several tracks using **pyGenomeTracks** or can be interactively browsed via **hicBrowser**. Check figure below as an example.

**6.8.5 Command line options**

MPI-IE workflow for Hi-C analysis

**usage example:** HiC -i input-dir -o output-dir mm10

Positional Arguments

**GENOME**

Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_gencodeM13, mm9, GRCz10, hs37d5, dm6

Required Arguments

- **-i, --input-dir**
  input directory containing the FASTQ files, either paired-end OR single-end data

- **-o, --output-dir**
  output directory

General Arguments

- **-v, --verbose**
  verbose output (default: 'False')

- **--ext**
  Suffix used by input fastq files (default: ‘”.fastq.gz”‘).

- **--reads**
  Suffix used to denote reads 1 and 2 for paired-end data. This should typically be either ‘_1’ ‘_2’ or ‘_R1’ ‘_R2’ (default: ‘[’_R1’, ‘_R2’]’). Note that you should NOT separate the values by a comma (use a space) or enclose them in brackets.

- **-c, --configFile**
  configuration file: config.yaml (default: ‘None’)

- **--clusterConfigFile**
  configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: ‘None’)

- **-j, --jobs**
  maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: ‘5’)

- **--local**
  run workflow locally; default: jobs are submitted to Slurm queue (default: ‘False’)

- **--keepTemp**
  Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.

- **--snakemakeOptions**
  Snakemake options to be passed directly to snakemake, e.g. use –snakemakeOptions='–dryrun –rerun-incomplete –unlock –forceall'. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ‘[’–use-conda’]’)

- **--DAG**
  If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.

- **--version**
  show program’s version number and exit
Email Arguments

--emailAddress
If specified, send an email upon completion to the given email address

--smtpServer
If specified, the email server to use.

--smtpPort
The port on the SMTP server to connect to. A value of 0 specifies the default port.

--onlySSL
The SMTP server requires an SSL connection from the beginning.

--emailSender
The address of the email sender. If not specified, it will be the address indicated by --emailAddress

--smtpUsername
If your SMTP server requires authentication, this is the username to use.

--smtpPassword
If your SMTP server requires authentication, this is the password to use.

Options

--downsample
Downsample the given number of reads randomly from of each FASTQ file (default: 'False')

--trim
Activate fastq read trimming. If activated, Illumina adaptors are trimmed by default. Additional parameters can be specified under --trimmerOptions. (default: 'False')

--trimmer
Possible choices: cutadapt, trimgalore, fastp

--trimmerOptions
Trimming program to use: Cutadapt, TrimGalore, or fastp. Note that if you change this you may need to change --trimmerOptions to match! (default: '"cutadapt"')

--fastqc
Run FastQC read quality control (default: 'False')

--bcExtract
To extract umi barcode from fastq file via UMI-tools and add it to the read name (default: 'False')

--bcPattern
The pattern to be considered for the barcode. 'N' = UMI position (required) 'C' = barcode position (optional) (default: '"NNNNCCCCCCCCC"')

--UMIDedup
Deduplicate bam file based on UMIs via umi_tools dedup that are present in the read name. (default: 'False')

--UMIDedupSep
umi separation character that will be passed to umi_tools.(default: '"_"')

--UMIDedupOpts
Additional options that will be passed to umi_tools.(default: '"')

--RFResolution
Create Hi-C matrices at the restriction fragment resolution. Using RFResolution would override the --binSize argument. (default: 'False')

--enzyme
Possible choices: DpnII, HindIII

Which enzyme was used to create Hi-C library (default: '"HindIII"')

--binSize
Create Hi-C matrices at the given binSize. This option is mutually exclusive with the --RFResolution option (default: '10000')

--restrictRegion
Restrict building of HiC Matrix to given region [Chr:Start-End]. Only one chromosome can also be specified (default: 'None')
--mergeSamples  Merge the HiC matrices and create a new matrix. If this option is specified together with --sampleInfo (see below), the samples would be merged based on the defined groups. (default: 'False')

--nBinsToMerge  If > 0, create a lower resolution HiC matrix for each sample by merging the given number of bins. (default: '0')

--findTADParams  parameters for HiCFindTADs. (default: "--thresholdComparisons 0.01")

--noTAD  Stop the pipeline before TAD calling. (default: 'False')

--noCorrect  Stop the pipeline before ICE-correction (i.e. run only upto building the matrix). (default: 'False')

--distVsCount  Produce a plot of the ICE-corrected HiC counts as a function of distance. This plot could be used for QC as well as comparison between samples for biological effects. The plot is create using the tool 'hicDistVsCount'. (default: 'False')

--distVsCountParams  parameters to run hicDistVsCount. (default: 'None')

--sampleSheet  A .tsv file containing sample names and associated groups. If provided, the file would be used to identify groups to merge the samples. An example can be found at 'docs/content/sampleSheet.example.tsv' (default: None)

--correctionMethod  Method to be used to balance the hic matrix. Available options are KR and ICE. (default: '"KR"')

code @ github.

6.9 preprocessing

6.9.1 What it does

The preprocessing pipeline handles a few tasks that are commonly done by some, but not all, sequencing providers:

- Merging samples across lanes (or technical replicates)
- Removal of apparent optical duplicates
- Reformating fastq files to extract UMIs
- Running FastQC
6.9.2 Input requirements

The minimal requirement is a directory of fastq files. If files should be merged (e.g., the sequencing provider did not merge samples across lanes) then a sample sheet should also be provided of the following form:

```
sample1_S1_L001_R1_001.fastq.gz _R1 sample1
sample1_S1_L001_R2_001.fastq.gz _R2 sample1
sample1_S1_L002_R1_001.fastq.gz _R1 sample1
sample1_S1_L002_R2_001.fastq.gz _R2 sample1
sample1_S1_L003_R1_001.fastq.gz _R1 sample1
sample1_S1_L003_R2_001.fastq.gz _R2 sample1
```

The first column contains file names, the second the associated read 1/2 designator (this should match the --reads option), and finally the desired sample name.

Care should be given when setting --optDedupDist, as values of 0 (the default) disable removal of optical duplicates. The appropriate value to use is sequencer-dependent.

6.9.3 Understanding the outputs

The preprocessing pipeline can generate the following files and directories (depending on the options given):

```
  deduplicatedFASTQ
    sample1.metrics
    sample1_R1.fastq.gz
    sample1_R1_optical_duplicates.fastq.gz
    sample1_R2.fastq.gz
    sample1_R2_optical_duplicates.fastq.gz
    optical_dedup_mqc.json
  FASTQ
  FastQC
  mergedFASTQ
  multiQC
  originalFASTQ
```

As shown above, the pipeline produces the following directories:

- **mergedFASTQ**: If a sample sheet is given, this file contains the merged fastq files.
- **deduplicatedFASTQ**: The results of optical duplicate removal (or symlinks to mergedFASTQ). The "_optical_duplicates" files contain the reads marked by clumpify as being likely optical duplicates. The associated "_metrics" file contains two values: number of optical duplicates and then the total reads. The optical_dedup_mqc.json file merges the various sample metrics files for downstream use by MultiQC.
- **originalFASTQ**: This folder exists from compatibility with other pipelines and will contain either symlinks to the original fastq files or, if a sample sheet is specified, those in deduplicatedFASTQ.
- **FASTQ**: Fastq files produced by UMI processing (or symlinks to originalFASTQ).
- **FastQC**: If the --fastqc parameter was given, the output of FastQC.
- **multiQC**: If either FastQC was run or optical duplicates were removed, an interactive web report will be created using MultiQC.

6.9.4 Command line options

MPI-IE workflow for preprocessing
usage example: preprocessing -i input-dir -o output-dir --optDedupDist 2500 --clumpifyOptions

usage: Preprocessing -i INDIR -o OUTDIR [-h] [-v] [--ext EXT]
        [--SMTPserver SMTPSERVER] [-c SMTPPORT] [-onlySSL]
        [-emailSender EMAILSENDERS] [-c SMTPUsername SMTPUSERNAME]
        [-c SMTPPassword SMTPPASSWORD] [-c fastqc] [-c bcExtract]
        [-bcPattern BCPATTERN] [-c optDedupDist OPTDEDUPDIST]
        [-c clumpifyOptions CLUMPIFYOPTIONS]
        [-c clumpifyMemory CLUMPIFYMEMORY]
        [-c sampleSheet SAMPLESHEET] 

Required Arguments

-i, --input-dir  input directory containing the FASTQ files, either paired-end OR single-end data
-o, --output-dir output directory

General Arguments

-v, --verbose   verbose output (default: 'False')
--ext          Suffix used by input fastq files (default: '.fastq.gz').
--reads        Suffix used to denote reads 1 and 2 for paired-end data. This should typically be either '_1', '_2' or '_R1', '_R2' (default: ['_R1', '_R2']). Note that you should NOT separate the values by a comma (use a space) or enclose them in brackets.
-c, --configFile configuration file: config.yaml (default: 'None')
--clusterConfigFile configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')
-j, --jobs      maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')
--local         run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')
--keepTemp      Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.
--snakemakeOptions Snakemake options to be passed directly to snakemake, e.g. use --snakemakeOptions='--dryrun --rerun-incomplete --unlock --forceall'. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ['--use-conda'])

-DAG           If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.
--version      show program’s version number and exit
### Email Arguments

- **--emailAddress**
  If specified, send an email upon completion to the given email address

- **--smtpServer**
  If specified, the email server to use.

- **--smtpPort**
  The port on the SMTP server to connect to. A value of 0 specifies the default port.

- **--onlySSL**
  The SMTP server requires an SSL connection from the beginning.

- **--emailSender**
  The address of the email sender. If not specified, it will be the address indicated by `--emailAddress`

- **--smtpUsername**
  If your SMTP server requires authentication, this is the username to use.

- **--smtpPassword**
  If your SMTP server requires authentication, this is the password to use.

### Options

- **--fastqc**
  Run FastQC read quality control (default: 'False')

- **--bcExtract**
  To extract umi barcode from fastq file via UMI-tools and add it to the read name (default: 'False')

- **--bcPattern**
  The pattern to be considered for the barcode. 'N' = UMI position (required) 'C' = barcode position (optional) (default: '"

- **--optDedupDist**
  The maximum distance between clusters to mark one as an optical duplicate of the other. Setting this to 0 will disable optical deduplication, which is only needed on patterned flow cells (NextSeq, NovaSeq, HiSeq 3000/4000, etc.). Common values are: 2500 (HiSeq 3000/4000), 40 (NextSeq) or 12000 (NovaSeq). (default: '0')

- **--clumpifyOptions**
  Options passed to clumpify, which should generally NOT be changed. The exception to this is with NextSeq runs, where 'spany=t adjacent=t' should be ADDED. (default: '"dupesub=0 qin=33 markduplicates=t optical=t"

- **--clumpifyMemory**
  This controls how much memory clumpify is instructed to use, in GB. This may need to be increased if samples are particularly large or there are MANY optical duplicates. This is passed to clumpify (e.g., as -Xmx30G). You may additionally need to instruct your scheduler about the per-core memory usage (e.g., in cluster.yaml). (default: '"30G"'

- **--sampleSheet**
  Information on samples (required for merging across lanes); see 'docs/content/preprocessing_sampleSheet.example.tsv' for an example. The first set of columns should hold the current sample names (excluding things like _R1.fastq.gz or _1.fastq.gz) while the second holds the name that the final sample should have (again, excluding things like _R1.fastq.gz or _1.fastq.gz). (default: 'None')

- **--automaticIlluminaMerging**
  Instead of manually specifying a sample sheet that can be used to merge samples, this option assumes that fastq files in the input directory follow the default output of bcl2fastq from Illumina and should be merged across lanes. An example of this would be that files named Sample1_S1_L001_R1_001.fastq.gz and Sample1_S1_L002_R1_001.fastq.gz would be merged to Sample1_R1.fastq.gz. This option CAN NOT be combined with --sampleSheet. It is assumed that standard R1/R2 read designators are being used and that all files end in .fastq.gz
6.10 mRNA-seq

6.10.1 What it does

The snakePipes mRNA-seq workflow allows users to process their single or paired-end mRNA-seq fastq files up to the point of gene/transcript-counts and differential expression. It also allows full allele-specific mRNA-seq analysis (up to allele-specific differential expression) using the *allelic-mapping* mode.

6.10.2 Input requirements

The only requirement is a directory of gzipped fastq files. Files could be single or paired end, and the read extensions could be modified using the keys in the *defaults.yaml* file below.

Configuration file

There is a configuration file in `snakePipes/workflows/mRNA-seq/defaults.yaml`:

```yaml
## General/Snakemake parameters, only used/set by wrapper or in Snakemake cmdl, but not in Snakefile
pipeline: rna-seq
outdir:
configFile:
clusterConfigFile:
local: False
maxJobs: 5
## directory with fastq files
indir:
## preconfigured target genomes (mm9, mm10, dm3,...), see /path/to/snakemake_workflows/shared/organisms/
genome:
## FASTQ file extension (default: ".fastq.gz")
ext: ".fastq.gz"
```

(continues on next page)
Apart from the common workflow options (see Running snakePipes), the following parameters are useful to consider:

- **aligner**: You can choose either STAR or HISAT2. While HISAT2 requires less memory than STAR, we keep STAR as the default aligner due to its superior accuracy (see this paper). Make sure that `--alignerOptions` matches this.

- **alignerOptions**: Options to pass on to your chosen aligner. Note that library type and junction definitions doesn’t have to be passed to the aligners as options, as they are handle either automatically, or via other parameters.

- **featureCountsOptions**: Options to pass to featureCounts (in case the alignment or allelic-mapping mode is used). Note that the paired-end information is automatically passed to featurecounts via the workflow, and the summerization is always performed at gene level, since the workflow assumes that featurecounts output is being used for gene-level differential expression analysis. If you wish to perform a transcript-level DE analysis, please choose the mode alignment-free.

- **filterGTF**: Options you can pass on to filter the original GTF file. This is useful in case you want to filter certain kind of transcripts (such as pseudogenes) before running the counts/DE analysis.
• **libraryType**: The default library-type is suitable for most RNAseq protocols (using Illumina Tru-Seq). Change this option in case you have a different strandedness.

• **salmonIndexOptions**: In the alignment-free mode (see below), this option allows you to change the type of index created by salmon. New users can leave it to default.

• **dnaContam**: Enable this to test for possible DNA contamination in your mRNA-seq samples. DNA contamination is quantified as the fraction of reads falling into intronic and intergenic regions, compared to those falling into exons. Enabling this option would produce a directory called GenomicContamination with .tsv files containing this information.

• **plotFormat**: You can switch the type of plot produced by all deeptools modules using this option. Possible choices: png, pdf, svg, eps, plotly

• **SNPFile**: For the allelic-mapping mode. The SNPFile is the file produced by SNPsplit after creating a dual-hybrid genome. The file has the suffix .vcf.

• **NMaskedIndex**: For the allelic-mapping mode. The NMaskedIndex refers to the basename of the index file created using STAR, on the SNPsplit output.

**Note**: SNPFile and NMaskedIndex file could be specified in case you already have this and would like to re-run the analysis on new data. In case you are running the allele-specific analysis for the first time, you would need a VCF file and the name of the two strains. In this case the SNPFile as well as the NMaskedIndex files would be automatically created by the workflow using SNPsplit.

### 6.10.3 Differential expression

Like the other workflows, differential expression can be performed using the --sampleSheet option and supplying a sample sheet like that below:

<table>
<thead>
<tr>
<th>name</th>
<th>condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample1</td>
<td>eworo</td>
</tr>
<tr>
<td>sample2</td>
<td>eworo</td>
</tr>
<tr>
<td>SRR7013047</td>
<td>eworo</td>
</tr>
<tr>
<td>SRR7013048</td>
<td>OreR</td>
</tr>
<tr>
<td>SRR7013049</td>
<td>OreR</td>
</tr>
<tr>
<td>SRR7013050</td>
<td>OreR</td>
</tr>
</tbody>
</table>

**Note**: The first entry defines which group of samples are control. This way, the order of comparison and likewise the sign of values can be changed. The DE analysis might fail if your sample names begin with a number. So watch out for that!

### 6.10.4 Differential Splicing

In addition to differential expression, differential splicing analysis can be performed by using --rMats option in addition to supplying a sample sheet. This will invoke the rMats turbo on the samples.

**Complex designs with blocking factors**

If the user provides additional columns between 'name' and 'condition' in the sample sheet, the variables stored there will be used as blocking factors in the order they appear in the sample sheet. Eg. if the first line of your sample sheet
looks like 'name batch condition', this will translate into a formula \( \text{batch} + \text{condition} \). 'condition' has to be the final column and it will be used for any statistical inference.

**Multiple pairwise comparisons**

The user may specify multiple groups of independent comparisons by providing a 'group' column after the 'condition' column. This will cause the sample sheet to be split into the groups defined in this column, and a corresponding number of independent pairwise comparisons will be run, one for each split sheet, and placed in separate output folders named accordingly. This will be applied to DESeq2, sleuth, and rMats pairwise comparisons as requested by the user. Specifying a value of 'All' in the 'group' column will cause that sample group to be used in all pairwise comparisons, e.g. if the same set of controls should be used for several different treatment groups.

An example sample sheet with the group information provided looks like this:

```
name  condition  group   sample1  Control  All  sample2  Control  All  sample3  Treatment  Group1 sample4  Treatment  Group1 sample5  Treatment  Group2 sample6  Treatment  Group2
```

### 6.10.5 Analysis modes

Following analysis (modes) are possible using the mRNA-seq workflow:

"alignment"

In this mode, the pipeline uses one of the selected aligners to create BAM files, followed by gene-level quantification using `featureCounts`. Gene-level differential expression analysis is then performed using `DESeq2`.

"allelic-mapping"

`allelic-mapping` mode follows a similar process as the "mapping" mode, however the alignment performed on an allele-masked genome, followed by allele-specific splitting of mapped files. Gene-level quantification is performed for each allele using `featureCounts`. Allele-specific, gene-level differential expression analysis is then performed using `DESeq2`.

**Note:** `allelic-mapping` mode is mutually exclusive with `mapping` mode

"alignment-free"

In this mode, the pipeline uses `salmon` to perform transcript-level expression quantification. This mode performs both transcript-level differential expression (using `Sleuth`), and gene-level differential expression (using `wasabi`, followed by `DESeq2`).

**Note:** The salmon index is recreated each time in alignment-free mode. This is done to facilitate changing how the GTF file is filtered, which necessitates reindexing.
"deepTools_qc"

The pipeline provides multiple quality controls through deepTools, which can be triggered using the `deepTools_qc` mode. It’s a very useful add-on with any of the other modes.

**Note:** Since most deeptools functions require an aligned (BAM) file, the deepTools_qc mode will additionally perform the alignment of the fastq files. However this would not interfere with operations of the other modes.

### 6.10.6 Understanding the outputs

Assuming the pipeline was run with `--mode 'alignment-free,alignment,deepTools_qc'` on a set of FASTQ files, the structure of the output directory would look like this (files are shown only for one sample)

```plaintext
- Annotation
  - filter_command.txt
  - genes.annotated.bed
  - genes.filtered.bed
  - genes.filtered.fa
  - genes.filtered.gtf
  - genes.filtered.symbol
  - genes.filtered.t2g
- bamCoverage
  - logs
  - sample1.coverage.bw
  - sample1.RPKM.bw
  - sample1.uniqueMappings.fwd.bw
  - sample1.uniqueMappings.rev.bw
- cluster_logs
- deepTools_qc
  - bamPEFragmentSize
    - fragmentSize.metric.tsv
    - fragmentSizes.png
  - estimateReadFiltering
    - sample1_filtering_estimation.txt
  - logs
- multiBigwigSummary
- plotCorrelation
  - correlation.pearson.bed_coverage.heatmap.png
  - correlation.pearson.bed_coverage.tsv
  - correlation.spearman.bed_coverage.heatmap.png
  - correlation.spearman.bed_coverage.tsv
- plotEnrichment
  - plotEnrichment.png
  - plotEnrichment.tsv
- plotPCA
  - PCA.bed_coverage.png
  - PCA.bed_coverage.tsv
- DESeq2_Salmon_sampleSheet
  - DESeq2_Salmon.err
  - DESeq2_Salmon.out
  - citations.bib
  - DESeq2_report_files
  - DESeq2_report.html
  - DESeq2_report.Rmd
```

(continues on next page)
6.10. mRNA-seq
Note: The _sampleSheet suffix for the DESeq2_sampleSheet and sleuth_Salmon_sampleSheet is
drawn from the name of the sample sheet you use. So if you instead named the sample sheet mySampleSheet.txt
then the folders would be named DESeq2_mySampleSheet and sleuth_Salmon_mySampleSheet. This
facilitates using multiple sample sheets.

Apart from the common module outputs (see Running snakePipes), the workflow would produce the following folders:

- **Annotation**: This folder would contain the GTF and BED files used for analysis. In case the file has been
  filtered using the --filterGTGTFF option (see Configuration file), this would contain the filtered files.

- **STAR/HISAT2**: (not produced in mode alignment-free) This would contain the output of RNA-alignment by
  STAR or HISAT2 (indexed BAM files).

- **featureCounts**: (not produced in mode alignment-free) This would contain the gene-level counts (output of
  featureCounts), on the filtered GTF files, that can be used for differential expression analysis.

- **bamCoverage**: (not produced in mode alignment-free) This would contain the bigWigs produced by deepTools
  bamCoverage. Files with suffix .coverage.bw are raw coverage files, while the files with suffix RPKM.bw
  are RPKM-normalized coverage files.

- **deepTools_QC**: (produced in the mode deepTools_QC) This contains the quality checks specific for mRNA-seq,
  performed via deepTools. The output folders are names after various deepTools functions and the outputs are ex-
  plained under deepTools documentation. In short, they show the insert size distribution (bamPEFragmentSize),
  mapping statistics (estimateReadFiltering), sample-to-sample correlations and PCA (multiBigwigSummary,
  plotCorrelation, plotPCA), and read enrichment on various genic features (plotEnrichment).

- **DESeq2_[sampleSheet]/DESeq2_Salmon_[sampleSheet]**: (produced in the modes alignment or alignment-
  free, only if a sample-sheet is provided.) The folder contains the HTML result report DESeq2_report.html, the annotated output file from DESeq2 (DEseq_basic_DEresults.tsv) and normalized counts for all samples, produced via DEseq2 (DEseq_basic_counts_DESeq2.normalized.tsv) as well as an Rdata file (DEseq_basic_DESeq.Rdata) with the R objects dds <- DESeq2::DESeq(dds) and
  ddr <- DDESeq2::results(dds, alpha = fdr). DESeq2_[sampleSheet] uses gene counts from
  featureCounts/counts.tsv, whereas DESeq2_Salmon_[sampleSheet] uses transcript counts from
  Salmon/counts.tsv that are merged via tximport in R.

- **Salmon**: (produced in mode alignment-free) This folder contains transcript-level (counts.tsv) and gene-
  level (counts.genes.tsv) counts estimated by the tool Salmon.

- **sleuth_Salmon_[sampleSheet]** (produced in mode alignment-free, only if a sample-sheet is provided) This
  folder contains a transcript-level differential expression output produced using the tool Sleuth.

### 6.10.7 Command line options

MPI-IE workflow for RNA mapping and analysis

**usage example:** RNA-seq -i input-dir -o output-dir mm10

```
usage: mRNA-seq -i INDIR -o OUTDIR [-h] [-v] [--ext EXT] [--reads READS READS]
        [-c CONFIGFILE] [--clusterConfigFile CLUSTERCONFIGFILE]
        [-j INT] [--local] [--keepTemp]
        [--snakemakeOptions SNAKEMAKEOPTIONS] [--DAG] [--version]
```
Positional Arguments

**GENOME**

Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_gencodeM13, mm9, GRCz10, hs37d5, dm6

Required Arguments

- **-i, --input-dir**
  input directory containing the FASTQ files, either paired-end OR single-end data

- **-o, --output-dir**
  output directory

General Arguments

- **-v, --verbose**
  verbose output (default: 'False')

- **--ext**
  Suffix used by input fastq files (default: "'.fastq.gz'`).

- **--reads**
  Suffix used to denote reads 1 and 2 for paired-end data. This should typically be either '_1' ',_2' or '_R1' ',_R2' (default: '['_R1', '_R2']'). Note that you should NOT separate the values by a comma (use a space) or enclose them in brackets.

- **-c, --configFile**
  configuration file: config.yaml (default: 'None')

- **--clusterConfigFile**
  configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')

- **-j, --jobs**
  maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')

- **--local**
  run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')

- **--keepTemp**
  Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.
--snakemakeOptions Snakemake options to be passed directly to snakemake, e.g. use –snakemakeOptions='–dryrun –rerun-incomplete –unlock –forceall'. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ['–use-conda'])

--DAG If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.

--version show program’s version number and exit

Email Arguments

--emailAddress If specified, send an email upon completion to the given email address
--smtpServer If specified, the email server to use.
--smtpPort The port on the SMTP server to connect to. A value of 0 specifies the default port.
--onlySSL The SMTP server requires an SSL connection from the beginning.
--emailSender The address of the email sender. If not specified, it will be the address indicated by –emailAddress
--smtpUsername If your SMTP server requires authentication, this is the username to use.
--smtpPassword If your SMTP server requires authentication, this is the password to use.

Allele-specific mapping arguments

--VCFfile VCF file to create N-masked genomes (default: 'None')
--strains Name or ID of SNP strains separated by comma (default: 'None')
--SNPfile File containing SNP locations (default: 'None')
--NMaskedIndex N-masked index of the reference genome (default: 'None')

Options

-m, --mode workflow running modes (available: ‘alignment-free, alignment, allelic-mapping, deepTools_qc’) (default: ‘alignment,deepTools_qc’)
--downsample Downsample the given number of reads randomly from of each FASTQ file (default: ‘False’)
--trim Activate fastq read trimming. If activated, Illumina adaptors are trimmed by default. Additional parameters can be specified under –trimmerOptions. (default: ‘False’)
--trimmer Possible choices: cutadapt, trimgalore, fastp
Trimming program to use: Cutadapt, TrimGalore, or fastp. Note that if you change this you may need to change –trimmerOptions to match! (default: ‘cutadapt’)
--trimmerOptions Additional option string for trimming program of choice. (default: ‘None’)
--fastqc Run FastQC read quality control (default: ‘False’)
To extract umi barcode from fastq file via UMI-tools and add it to the read name (default: 'False')

The pattern to be considered for the barcode. 'N' = UMI position (required) 'C' = barcode position (optional) (default: "NNNCCCCCCCCC")

Deduplicate bam file based on UMIs via umi_tools dedup that are present in the read name. (default: 'False')

umi separation character that will be passed to umi_tools.(default: "\"\"\")

Additional options that will be passed to umi_tools.(default: ")

Bin size of output files in bigWig format (default: '25')

Possible choices: png, pdf, None

Format of the output plots from deepTools. Select 'none' for no plots (default: "png")

user provided library type strand specificity. featureCounts style: 0, 1, 2 (Illumina TruSeq); default: '2')

Possible choices: STAR, HISAT2

Program used for mapping: STAR or HISAT2 (default: "STAR"). If you change this, please change --alignerOptions to match.

STAR or hisat2 option string, e.g.: '--twopassMode Basic' (default: 'None')

Salmon index options, e.g. '--type puff' (default: "--type puff -k 31")

featureCounts option string. The options '-p -B' are always used for paired-end data (default: "-C -Q 10 --primary")

filter annotation GTF by grep for use with Salmon, e.g. use --filterGTF='-v pseudogene'; default: 'None')

Information on samples (required for DE analysis); see 'https://github.com/maxplanck-ie/snakepipes/tree/master/docs/content/sampleSheet.example.tsv' for example. The column names in the tsv files are 'name' and 'condition'. The first entry defines which group of samples are control. This way, the order of comparison and likewise the sign of values can be changed. The DE analysis might fail if your sample names begin with a number. So watch out for that! (default: 'None')

Returns a plot which presents the proportion of the intergenic reads (default: 'False')

Input folder with bam files. If provided, the analysis will start from this point. If bam files contain single ends, please specify --singleEnd additionally.

Extention of provided bam files, will be substracted from basenames to obtain sample names. (default: ".bam")

input data is single-end, not paired-end. This is only used if --fromBAM is specified.

Run differential splicing analysis using rMats-turbo. Note that this flag requires --sampleSheet to be specified.
6.11 noncoding-RNA-seq

6.11.1 What it does

The snakePipes noncoding-RNA-seq workflow allows users to process their single or paired-end ribosomal-depleted RNA-seq fastq files up to the point of gene/transcript/repeat-element counts and differential expression. Repeat elements are quantified and tested for differential expression at the name, family and class level. Since changes in repeat element expression tend to be unidirectional, size factors from gene expression are used when normalizing repeat element expression.

Note that in addition to the normal GTF file, this pipeline requires a repeat masker output file, which can be downloaded from UCSC or other sites. The chromosome names here must match that used in the other files.

6.11.2 Input requirements

The only requirement is a directory of gzipped fastq files. Files could be single or paired end, and the read extensions could be modified using the keys in the defaults.yaml file below.

Configuration file

There is a configuration file in snakePipes/workflows/noncoding-RNA-seq/defaults.yaml:

```yaml
## General/Snakemake parameters, only used/set by wrapper or in Snakemake cmdl, but not in Snakefile
pipeline: noncoding-rna-seq
outdir:
configFile:
clusterConfigFile:
local: False
maxJobs: 5
```

(continues on next page)
Apart from the common workflow options (see Running snakePipes), the following parameters are useful to consider:

- **aligner**: The only choice at the moment is STAR.

- **alignerOptions**: Options to pass on to your chosen aligner. Note that library type and junction definitions don’t have to be passed to the aligners as options, as they are handled either automatically, or via other parameters.

- **plotFormat**: You can switch the type of plot produced by all deeptools modules using this option. Possible choices: png, pdf, svg, eps, plotly

### 6.11.3 Differential expression

Like the other workflows, differential expression can be performed using the **--sampleSheet** option and supplying a sample sheet like that below:
Note: The first entry defines which group of samples are control. This way, the order of comparison and likewise the sign of values can be changed. The DE analysis might fail if your sample names begin with a number. So watch out for that!

Complex designs with blocking factors

If the user provides additional columns between 'name' and 'condition' in the sample sheet, the variables stored there will be used as blocking factors in the order they appear in the sample sheet. Eg. if the first line of your sample sheet looks like 'name batch condition', this will translate into a formula batch + condition. 'condition' has to be the final column and it will be used for any statistical inference.

Multiple pairwise comparisons

The user may specify multiple groups of independent comparisons by providing a 'group' column after the 'condition' column. This will cause the sample sheet to be split into the groups defined in this column, and a corresponding number of independent pairwise comparisons will be run, one for each split sheet, and placed in separate output folders named accordingly. This will be applied to DESeq2 pairwise comparison. Specifying a value of 'All' in the 'group' column will cause that sample group to be used in all pairwise comparisons, e.g. if the same set of controls should be used for several different treatment groups.

An example sample sheet with the group information provided looks like this:

```
name condition group sample1 Control All sample2 Control All sample3 Treatment Group1 sample4 Treatment Group1 sample5 Treatment Group2 sample6 Treatment Group2
```

6.11.4 Analysis modes

Following analysis (modes) are possible using the noncoding-RNA-seq workflow:

"alignment"

In this mode, the pipeline uses STAR to create BAM files and TEtranscripts to quantify genes and repeat elements. Differential expression of genes and repeat elements is then performed with DESeq2.

"deepTools_qc"

The pipeline provides multiple quality controls through deepTools, which can be triggered using the deepTools_qc mode. It’s a very useful add-on with any of the other modes.
Note: Since most deeptools functions require an aligned (BAM) file, the deepTools_qc mode will additionally perform the alignment of the fastq files. However this would not interfere with operations of the other modes.

6.11.5 Understanding the outputs

Assuming the pipeline was run with --mode 'alignment,deepTools_qc' on a set of FASTQ files, the structure of the output directory would look like this (files are shown only for one sample):

```
- bamCoverage
  - sample1.coverage.bw
  - sample1.RPKM.bw
  - sample1.uniqueMappings.fwd.bw
  - sample1.uniqueMappings.rev.bw
- cluster_logs
- deepTools_qc
  - bamPEFragmentSize
    - fragmentSize.metric.tsv
    - fragmentSizes.png
  - estimateReadFiltering
    - sample1_filtering_estimation.txt
- logs
  - bamPEFragmentSize.err
  - bamPEFragmentSize.out
  - multiBigwigSummary.err
  - multiBigwigSummary
    - coverage.bed.npz
  - plotCorrelation
    - correlation.pearson.bed_coverage.heatmap.png
    - correlation.pearson.bed_coverage.tsv
    - correlation.spearman.bed_coverage.heatmap.png
    - correlation.spearman.bed_coverage.tsv
  - plotEnrichment
  - plotEnrichment.png
  - plotEnrichment.tsv
  - plotPCA
  - PCA.bed_coverage.png
  - PCA.bed_coverage.tsv
- DESeq2_sampleSheet
  - DESeq2_report_genes.html
  - DESeq2_report_repeat_class.html
  - DESeq2_report_repeat_family.html
  - DESeq2_report_repeat_name.html
  - DESeq2.session_info.txt
  - genes_counts_DESeq2.normalized.tsv
  - genes_DEresults.tsv
  - genes_DESeq.Rdata
  - repeat_class_counts_DESeq2.normalized.tsv
  - repeat_class_DEresults.tsv
  - repeat_class_DESeq.Rdata
  - repeat_family_counts_DESeq2.normalized.tsv
  - repeat_family_DEresults.tsv
  - repeat_family_DESeq.Rdata
  - repeat_name_counts_DESeq2.normalized.tsv
```

(continues on next page)
Note: The _sampleSheet suffix for the DESeq2_sampleSheet is drawn from the name of the sample sheet you use. So if you instead named the sample sheet mySampleSheet.txt then the folder would be named DESeq2_mySampleSheet. This facilitates using multiple sample sheets.

Apart from the common module outputs (see Running snakePipes), the workflow would produce the following folders:

- **bamCoverage**: This would contain the bigWigs produced by deepTools bamCoverage. Files with suffix .coverage.bw are raw coverage files, while the files with suffix RPKM.bw are RPKM-normalized coverage files.

- **deepTools_QC**: (produced in the mode deepTools_QC) This contains the quality checks specific for mRNA-seq, performed via deepTools. The output folders are names after various deepTools functions and the outputs are explained under deepTools documentation. In short, they show the insert size distribution (bamPEFragmentSize), mapping statistics (estimateReadFiltering), sample-to-sample correlations and PCA (multiBigwigSummary, plotCorrelation, plotPCA), and read enrichment on various genic features (plotEnrichment).

- **DESeq2_[sampleSheet]**: (produced only if a sample-sheet is provided.) The folder contains the HTML result reports DESeq2_report_genes.html, DESeq2_report_repeat_name.html, DESeq2_report_repeat_class.html and DESeq2_report_repeat_family.html as well as the annotated output file from DESeq2 (genes_DEresults.tsv, etc.) and normalized counts for all samples, produced via DEseq2 (genes_counts_DESeq2.normalized.tsv, etc.) as well as an Rdata file (genes_DESeq.Rdata, etc.) with the R objects dds <- DESeq2::DESeq(dds) and ddr <- DESeq2::results(dd, alpha = fdr).

- **filtered_bam**: This contains sorted and indexed BAM files that have been filtered to remove secondary alignments. This are used by deepTools and are appropriate for use in IGV.

- **multiQC**: This folder contains the report produced by MultiQC and summarizes alignment metrics from STAR and possibly various deepTools outputs.

- **STAR**: This would contain the output logs of RNA-alignment by STAR. The actual BAM files are removed at the end of the pipeline as they’re not compatible with typical visualization programs.

- **TEcount**: (produced in the alignment mode) This contains the counts files and logs from the TEcount program in the TEtranscripts package. These are used by DESeq2 for differential expression.

### 6.11.6 Command line options

**MPI-IE workflow for noncoding RNA mapping and analysis**

**usage example**: noncoding-RNA-seq -i input-dir -o output-dir mm10

**usage**:

```
```

(continues on next page)
Positional Arguments

**GENOME**

Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_gencodeM13, mm9, GRCz10, hs37d5, dm6

Required Arguments

- `-i`, `--input-dir`
  input directory containing the FASTQ files, either paired-end OR single-end data

- `-o`, `--output-dir`
  output directory

General Arguments

- `--verbose`
  verbose output (default: 'False')

- `--ext`
  Suffix used by input fastq files (default: '`.fastq.gz`).

- `--reads`
  Suffix used to denote reads 1 and 2 for paired-end data. This should typically be either `'_1'`, `'_2'` or `'_R1'`, `'_R2'` (default: `[''_R1', '_R2'']`). Note that you should NOT separate the values by a comma (use a space) or enclose them in brackets.

- `-c`, `--configFile`
  configuration file: config.yaml (default: 'None')

- `--clusterConfigFile`
  configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')

- `-j`, `--jobs`
  maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')

- `--local`
  run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')

- `--keepTemp`
  Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.

---

**noncoding-RNA-seq**

6.11. noncoding-RNA-seq
--snakemakeOptions Snakemake options to be passed directly to snakemake, e.g. use --snakemakeOptions='--dryrun --rerun-incomplete --unlock --forceall'. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ['--use-conda'])

--DAG If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.

--version show program's version number and exit

Email Arguments

--emailAddress If specified, send an email upon completion to the given email address
--smtpServer If specified, the email server to use.
--smtpPort The port on the SMTP server to connect to. A value of 0 specifies the default port.
--onlySSL The SMTP server requires an SSL connection from the beginning.
--emailSender The address of the email sender. If not specified, it will be the address indicated by --emailAddress
--smtpUsername If your SMTP server requires authentication, this is the username to use.
--smtpPassword If your SMTP server requires authentication, this is the password to use.

Options

-m, --mode workflow running modes (available: 'alignment, deepTools_qc') (default: '"alignment,deepTools_qc"')

--downsample Downsample the given number of reads randomly from of each FASTQ file (default: 'False')

--trim Activate fastq read trimming. If activated, Illumina adaptors are trimmed by default. Additional parameters can be specified under --trimmerOptions. (default: 'False')

--trimmer Possible choices: cutadapt, trimgalore, fastp
Trimming program to use: Cutadapt, TrimGalore, or fastp. Note that if you change this you may need to change --trimmerOptions to match! (default: '"fastp"')

--trimmerOptions Additional option string for trimming program of choice. (default: 'None')

--fastqc Run FastQC read quality control (default: 'False')

--bcExtract To extract umi barcode from fastq file via UMI-tools and add it to the read name (default: 'False')

--bcPattern The pattern to be considered for the barcode. 'N' = UMI position (required) 'C' = barcode position (optional) (default: '"NNNCCCCCCCCC"')

--UMIDedup Deduplicate bam file based on UMIs via umi_tools dedup that are present in the read name. (default: 'False')

--UMIDedupSep umi separation character that will be passed to umi_tools.(default: '"_"')
6.12 scRNA-seq

6.12.1 What it does

The scRNA-seq pipeline is intended to process UMI-based data, expecting the cell barcode and UMI in Read1, and the cDNA sequence in Read2. The workflow has predefined settings for CelSeq2 and 10x data, but can be extended to custom protocols.

There are currently two analysis modes available:  - "STARsolo" which uses STAR solo for mapping and quantitation.  - "Alevin" based on Salmon for generating the count matrix.

Note: Mode "Gruen" has been deprecated.

The general procedure for mode "STARsolo" involves:

1. moving cell barcodes and UMIs from read 1 into the CB and UMI tags of read 2 during mapping (STARsolo),
2. quantification of genic read counts at the single cell level (STARsolo),
3. quantification of reads supporting spliced and unspliced transcripts in each cell (velocyto) - unless this has been disabled with –skipVelocyto
4. generation of seurat objects for genic counts.

UMIs in the read headers are used to avoid counting PCR duplicates. A number of bigWig and QC plots (e.g., from plotEnrichment) are generated as well.

Mode "Alevin" involves:

1. Generation of a salmon index used for mapping.
4. General QC of the Alevin run using the AlevinQC R package.
5. Quantification of "spliced" and "unspliced" read counts in each cell with Alevin - unless this has been disabled with --skipVelocyto. This analysis is derived from the code underlying Soneson et al. 2020, bioRxiv https://doi.org/10.1101/2020.03.13.990069.

6.12.2 Mode STARsolo

With current settings, this mode should work with any UMI-based protocol that stores UMI and CB in read 1, each in one chunk. The mode comes with four presets that can be passed to the ‘--myKit’ argument: CellSeq192, CellSeq384, 10xV2, 10xV3. Choosing a preset will select a corresponding barcode whitelist file as well as cell barcode and umi length and positions to be used. Choosing the Custom preset allows the user to run the workflow providing own barcode whitelist and CB/UMI positions and lengths. CellSeq384 is the current default preset.

In this mode, STARsolo is used to map, UMI-deduplicate and count reads. Importantly, read 1 is expected to carry the UMI and the cell barcode, while read 2 is expected to carry the cDNA sequence. Default positions of UMI and CB in read 1 are specified, as well as their respective lengths. If your setup is different from the available presets, change it via the --STARsoloCoords commandline argument or in the defaults.yaml dictionary, in addition to providing --myKit Custom argument.

In the STARsolo folder, bam files are stored, along with 10x-format count matrices and log files summarizing barcode detection and UMI-deduplication. Bam files have the UB and CB tags set.

Deeptools QC is run on these bam files.

Before running velocyto, bam files from STARsolo are filtered to remove unmapped reads as well as reads with an empty CB tag and then cell-sorted by the CB tag. In the VelocytoCounts folder, loom files with counts of spliced, unspliced and ambiguous reads are stored. A merged loom file containing counts for all samples together can be found
in the VelocytoCounts_merged folder. As Velocyto tends to consume a lot of memory and result in long runtimes with cell numbers in ~10^5, it can be disabled with --skipVelocyto.

### 6.12.3 Input requirements

The primary input requirement is a directory of paired-end fastq files. For the STAR solo mode, a barcode whitelist is required, as well as specification of UMI and CB positions and length, if different from default or available presets.

#### Barcode whitelist

Required for the STARsolo mode. The expected format is a one-column txt file with barcodes the user wishes to retain. Default is a whitelist file for CellSeq2 384 barcodes, provided with the pipeline. If 'myKit' is changed to another available preset, the corresponding barcode whitelist provided with the pipeline will be used.

#### Configuration file

The default configuration file is listed below and can be found in `snakePipes/workflows/scRNAseq/defaults.yaml`:

```yaml
pipeline: scrna-seq
outdir: configFile: clusterConfigFile: local: False
maxJobs: 5
## directory with fastq files
indir: ## preconfigured target genomes (mm9,mm10,dm3,...), see /path/to/snakemake_workflows/ →shared/organisms/
## Value can be also path to your own genome config file!
genome: ## FASTQ file extension (default: ".fastq.gz")
ext: '.fastq.gz'
## paired-end read name extension (default: ["_R1", "_R2"])
reads: ["_R1","_R2"]
## Analysis mode
mode: STARsolo
## Number of reads to downsample from each FASTQ file
downsample:
## Options for trimming
trim: False
trimmer: cutadapt
trimmerOptions: -a A{'30'}
## --twopassMode Basic is not compatible with --outStd in all STAR versions
alignerOptions: ""
## further options
filterGTF: "-v -P 'decay|pseudogene' "
cellBarcodeFile: cellBarcodePattern: "NNNNNNXXXXXX"
splitLib: False
cellNames:
## mode STARsolo options
myKit: CellSeq384
BCwhiteList:

(continues on next page)
STARsoloCoords: ["1", "7", "8", "7"]
skipVelocyto: False

## mode Alevin options
alevinLibraryType: "ISR"
prepProtocol: "celseq2"
salmonIndexOptions: --type puff -k 31
expectCells:
readLengthFrQ: 0.2

# generic options
libraryType: 1
bwBinSize: 10
verbose: False
plotFormat: pdf
dnaContam: False

## Parameters for the statistical analysis
cellFilterMetric: gene_universe
# Option to skip RaceID to save time
skipRaceID: False
# umi_tools options:
UMIBarcode: False
bcPattern: NNNNCCCCCCCCC #default: 4 base umi barcode, 9 base cell barcode (e.g. RELACS barcode)
UMIDedup: False
UMIDedupSep: "_"
UMIDedupOpts: --paired

Pseudogene filter

As default, transcripts or genes that contain that are related to biotypes like 'pseudogene' or 'decay' are filtered out before tag counting (sec --filterGTF default). Here we assume you provide eg. a gencode or ensemble annotation file (via genes_gtf in the organism configuration yaml) that contains this information.

Library Type

The CEL-seq2 protocol produces reads where read 2 maps in sense direction (libraryType: 1).

Fraction of read length required to overlap the intron

In mode Alevin, the fraction of read length required to overlap the intron in order to be counted as "unspliced" is set to 0.2 (i.e. 20%) by default. This corresponds to 10nt in a 50nt-long read, or to 20nt in a 100nt-long read. The user is encouraged to modify this value as deemed appropriate via the --readLengthFrQ commandline argument. In practice, this variable affects the length of the exon sequence flank added to the intron sequence to generate reference sequences for Salmon Alevin. Exon sequence flank length is set to one minus 'readLengthFrQ' of read length.

6.12.4 Output structure

The following will be produced in the output directory when the workflow is run in mode STARsolo:

```
analysis/
  scRNAseq_run-1.log
  multiQC
```

(continues on next page)
The following output structure will be produced when running in Alevin mode:

- The **Salmon** directory contains the generated genome index.
- The **Alevin** directory contains the matrix files (both bootstrapped and raw) per sample in subdirectories.
- The **multiQC** directory contains an additional alevinQC html file generated per sample.
- The **AlevinForVelocity** directory contains the matrix files with "spliced" and "unspliced" reads per cell in subdirectories.
- The **SingleCellExperiment** directory contains the RDS files with "SingleCellExperiment" class R objects, storing spliced/unspliced counts per cell in corresponding assays.
6.12.5 Understanding the outputs: mode STARsolo

- **Main result:** output folders with 10x-format count matrices can be found in sample subfolders under STARsolo. The output consists of three files: barcodes.tsv, features.tsv, matrix.mtx. Their gzipped versions are stored in the same folder. Seurat objects from merged samples are available in the Seurat folder.

- Corresponding annotation files are: Annotation/genes.filtered.bed and Annotation/genes.filtered.gtf, respectively.

- The folders QC_report, FASTQC, deeptools_qc and multiQC contain various QC tables and plots.

- **STARsolo** directory contain the output from genomic alignments.

6.12.6 Understanding the outputs: mode Alevin

- **Main result:** output folders containing the raw and bootstrapped count matrices are found under the sample subfolders under Alevin. The sample specific Alevin folders contain the matrices, as well as column data (barcodes) and row data (genes). Alevin spliced/unspliced counts for RNA velocity are stored as alevin matrices in the sample subfolders under AlevinForVelocity and as "SingleCellExperiment" class R objects under SingleCellExperiment.

- Corresponding annotation files are: Annotation/genes.filtered.bed and Annotation/genes.filtered.gtf, respectively.

- The QC plots (both from multiQC and AlevinQC) are available in the multiQC folder.

6.12.7 Command line options

MPI-IE workflow for scRNA-seq (CEL-Seq2 and related protocols)

**usage example:** scRNAseq -i input-dir -o output-dir mm10

```bash
usage: scRNAseq -i INDIR -o OUTDIR [-h] [-v] [--ext EXT] [--reads READS READS]
    [-c CONFIGFILE] [--clusterConfigFile CLUSTERCONFIGFILE]
    [-j INT] [--local] [--keepTemp]
    [--snakemakeOptions SNAKEMAKEOPTIONS] [--DAG] [--version]
    [--emailAddress EMAILADDRESS] [--smtpServer SMTPSERVER]
    [--smtpPort SMTPPORT] [--onlySSL] [--emailSender EMAILSENDER]
    [--smtpUsername SMTPUSERNAME] [--smtpPassword SMTPPASSWORD]
    [--mode STR] [--downsample INT] [--trim]
    [--trimmerOptions STR] [--alignerOptions ALIGNEROPTIONS]
    [--cellBarcodeFile STR] [--cellBarcodePattern STR]
    [--splitLib] [--filterGTF FILTERGTF] [--cellNames STR]
    [--BCwhiteList STR] [--STARsoloCoords STARSOLOCOORDS]
    [--bwBinSize BWBINSIZE] [--plotFormat STR]
    [--cellFilterMetric STR] [--skipRaceID] [--myKit STR]
    [--skipVelocyto]
    [--prepProtocol {dropseq,chromiumV3,chromium,gemcode,citeseq,celseq,
        celseq2,quartzseq2}]
    [--salmonIndexOptions SALMONINDEXOPTIONS]
    [--alevinLibraryType {ISR,ISF,MSF,MSR,OSR,OSF}]
    [--expectCells EXPECTCELLS] [--readLengthFrx READLENGTHFRX]
    GENOME
```
Positional Arguments

**GENOME**
Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_gencodeM13, mm9, GRCz10, hs37d5, dm6

Required Arguments

- **-i, --input-dir**
in input directory containing the FASTQ files, either paired-end OR single-end data
- **-o, --output-dir**
output directory

General Arguments

- **-v, --verbose**
verbose output (default: 'False')
- **--ext**
Suffix used by input fastq files (default: '".fastq.gz"').
- **--reads**
Suffix used to denote reads 1 and 2 for paired-end data. This should typically be either '_1' '_2' or '_R1' '_R2' (default: ['_R1', '_R2']). Note that you should NOT separate the values by a comma (use a space) or enclose them in brackets.
- **-c, --configFile**
configuration file: config.yaml (default: 'None')
- **--clusterConfigFile**
configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')
- **-j, --jobs**
maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '5')
- **--local**
run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')
- **--keepTemp**
Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.
- **--snakemakeOptions**
Snakemake options to be passed directly to snakemake, e.g. use --snakemakeOptions='--dryrun --rerun-incomplete --unlock --forceall'. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: ['|--use-conda']')
- **--DAG**
If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.
- **--version**
show program's version number and exit

Email Arguments

- **--emailAddress**
If specified, send an email upon completion to the given email address
- **--smtpServer**
If specified, the email server to use.
- **--smtpPort**
The port on the SMTP server to connect to. A value of 0 specifies the default port.
- **--onlySSL**
The SMTP server requires an SSL connection from the beginning.
--emailSender The address of the email sender. If not specified, it will be the address indicated by --emailAddress

--smtpUsername If your SMTP server requires authentication, this is the username to use.

--smtpPassword If your SMTP server requires authentication, this is the password to use.

Options

--mode Possible choices: Gruen, STARsolo, Alevin
Analysis mode. Possible settings are 'Gruen, STARsolo and Alevin' Default: "STARsolo"

--downsample Downsample the given number of reads randomly from of each FASTQ file

--trim Activate trimming with Cutadapt. Default: no Trimming!

--trimmerOptions Options passed to the selected trimmer, e.g. use --trimmerOptions='-a A[30]' for polyA trimming with Cutadapt (default: "'-a A['30']'")

--alignerOptions STAR option string, e.g.: '–twopassMode Basic' (default: "’–outBAMsortingBinsN 30 –twopassMode Basic’")

--cellBarcodeFile 2-column file with cell-index (1st col) and barcode (2nd col). Default/None will use internal CelSeq2@MPI-IE file with 192 barcodes. (default: 'None')

--cellBarcodePattern Defines the cell barcode and UMI order and length at the 5' end of R1 (Cel-seq protocol). 'N' defines UMI/random positions, X defines fixed positions; (default "’NNNNNNXXXXXX’")

--splitLib Set this option if only 96 out of 192 barcodes were used per sample.

--filterGTF filter annotation GTF by grep for feature counting, e.g. use --filterGTF='-v pseudogene'; (default: "’-v -P ‘decay/pseudogene’ ‘’")

--cellNames either tab-sep. file with cell name ranges or directory with *.tsv files that contain cell names and plate/library information for all fastq files! (default: 'None')

--BCwhiteList Path to a one-column txt file with barcode whitelist. Required for the STARsolo mode,optional for Alevin mode. (default: 'None')

--STARsoloCoords Comma-separated list of values: UMI start position, UMI length, CB start position, CB length. Required for the STARsolo mode (default: '[1', '7', '8', '7']')

--bwBinSize Bin size of output files in bigWig format (default: '10')

--plotFormat Possible choices: png, pdf, None
Format of the output plots from deeptools. Select 'none' for no plot (default: "’png’")

--cellFilterMetric Possible choices: gene_universe, medGPC
The metric to maximise when selecting a cell filtering threshold (default: "’gene_universe’")

--skipRaceID Skip RaceID analysis.

--myKit Possible choices: 10Xv2, 10Xv3, CellSeq192, CellSeq384, Custom
Library preparation kit and version to use preset barcode whitelist and CB/UMI positions for (default: "’CellSeq384’")
--skipVelocyto Skip bam filtering and generating RNA velocity counts by velocyto to save time and memory usage. (default: 'False')

--prepProtocol Possible choices: dropseq, chromiumV3, chromium, gemcode, citeseq, celseq, celseq2, quartzseq2

Alevin mode. Specify the library prep method. (default: '"celseq2"')

--salmonIndexOptions Alevin mode. Salmon index options e.g. '--type puff' (default: '"--type puff -k 31"')

--alevinLibraryType Possible choices: ISR, ISF, MSF, MSR, OSR, OSF

Alevin mode. Library orientation type. (default: '"ISR"')

--expectCells Alevin mode. Optional to fill in if you know how many cells are expected. (default: 'None')

--readLengthFrx Fraction of read length required to align to the intronic sequence (default: '0.2')

code @ github.

6.13 WGBS

6.13.1 What it does

Optionally trimmed reads are mapped to the reference genome using a bisulfite-specific aligner (bwa-meth). Quality metrics are collected and synthesized in a QC report, including bisulfite conversion rate, mapping rate, coverage metrics, and methylation bias.

There are two flags that allow skipping certain QC metric calculation, i.e. --skipDOC and --GCbias. These deactivate or activate, respectively, the depth of coverage (DOC) calculations or GC bias calculation done by deepTools. If you run the workflow with --fromBAM, you can also choose to skip the re-calculation of most QC metrics with --skipBamQC.

Methylation ratios are extracted (via MethylDackel) for CpG positions in the reference genome with a minimum coverage specified by --minCoverage and low SNP allelic frequency (<0.25 illegitimate bases). If a sample sheet is provided, Metilene, DMRseq and/or DSS (as specified by --DMRprograms) will be used to find differentially methylated regions (DMRs). Filtering criterion can be changed both for the CpGs used to find DMRs as well as what are considered as significant DMRs.
6.13.2 Input requirements

This pipeline requires fastq files and a genome alias, for which bwa-meth index has been built. Optional inputs include a sample sheet with grouping information to use in differential methylation analysis and a blacklist bed file with genomic positions corresponding to known snps to mask single CpG methylation values.

It is possible to use pipeline-compatible bam files as input. For that, the user has to use the --fromBAM flag and provide the bam file extension if not matched by the default.

6.13.3 Workflow configuration file

```
## General/Snakemake parameters, only used/set by wrapper or in Snakemake cmdl, but not in Snakefile
outdir: 
configFile: 
clusterConfigFile: 
local: False
maxJobs: 12
## directory with fastq or bam files
indir: 
## Genome information
genome: 
###SNP black list (bed file)
blacklist: 
###sample Sheet
sampleSheet: 
###inclusion bounds for methylation extraction
noAutoMethylationBias: False 
## FASTQ file extension (default: ".fastq.gz")
ext: '.fastq.gz'
## paired-end read name extension (default: ['_R1', '_R2'])
read: [_R1, _R2]
## Number of reads to downsample from each FASTQ file
downsamp: 
## Options for trimming
trim: False 
trimmer: 'fastp'
trimmerOptions: '-q 5 -l 30 -M 5'
## Bin size of output files in bigWig format
bwBinSize: 25
## Run FASTQC read quality control
fastqc: false 
verbose: False
plotFormat: 'png'
## Flag to control the pipeline entry point
fromBAM: False
bamExt: '.bam'
pairedEnd: True
##Flags to control skipping of certain QC calculations
skipQC: False
GCbias: 
##Thresholds for filtering of statistical comparisons (DMRs and DMLs)
DMRprograms: 'metilene, dmrseq'
maxDist: 300
minCpGs: 10
minCoverage: 5
```
FDR: 0.1
minMethDiff: 0.1
###MethylDackel options
MethylDackelOptions: '--mergeContext --maxVariantFrac 0.25 --minDepth 4'

##umi_tools
UMIBarcode: False
bcPattern: NNNNCCCCCCCCC #default: 4 base umi barcode, 9 base cell barcode (eg. RELACS barcode)
UMIDedup: False
UMIDedupSep: "_"
UMIDedupOpts: --paired
aligner: bwameth

6.13.4 Understanding the outputs

The WGBS pipeline invoked fastq files and a sample sheet and the --trim and --fastqc options will generate the following output:

<table>
<thead>
<tr>
<th>output_dir</th>
</tr>
</thead>
<tbody>
<tr>
<td>bwameth</td>
</tr>
<tr>
<td>cluster_logs</td>
</tr>
<tr>
<td>dmrseq_sampleSheet_minCoverage5</td>
</tr>
<tr>
<td>FASTQ</td>
</tr>
<tr>
<td>FastQC_trimmed</td>
</tr>
<tr>
<td>FASTQ_fastp</td>
</tr>
<tr>
<td>MethylDackel</td>
</tr>
<tr>
<td>metilene_sampleSheet_minCoverage5</td>
</tr>
<tr>
<td>multiQC</td>
</tr>
<tr>
<td>originalFASTQ</td>
</tr>
<tr>
<td>QC_metrics</td>
</tr>
</tbody>
</table>

The workflow produces the following outputs:

- **FASTQ_downsampled**: contains read files downsampling to 5mln reads. These are used to calculate conversion rate which would otherwise take a very long time.

- **bwameth**: contains bam files obtained through read alignment with bwa-meth and the PCR duplicate removal with sambamba, as well as matching bam index files.

- **dmrseq_sampleSheet_minCoverage<X>**: DMRs (DMRs.txt) and a report (Stats_report.html) from DMRseq as well as a saved R session (Session.RData) using the requested minimum coverage. If you rerun the pipeline with a different minimum coverage specified then a new directory will be created.

- **DSS_sampleSheet_minCoverage<X>**: As with DMRseq above.

- **FastQC_trimmed**: FastQC output on the trimmed reads.

- **FASTQ_fastp**: The trimmed reads and QC metrics from FastP.

- **MethylDackel**: BigWig coverage and methylation files as well as the bedGraph files produced by MethylDackel.

- **metilene_sampleSheet_minCoverage<X>**: contains output files from metilene in DMRs.txt. DMRs.annotated.txt is an annotated version of that, wherein DMRs are annotated with the nearest gene and the distance to it. There is additionally a QC report (Stats_report.html) that summarizes various properties of the DMRs.
- **QC_metrics**: contains output files from conversion rate, flagstat, depth of coverage, GCbias and methylation bias calculations. The QC report in pdf format collecting those metrics in tabular form is also found in this folder.

### 6.13.5 Example output plots

Using data from Habibi et al., Cell Stem Cell 2013 corresponding to mouse chr6:4000000-6000000, following plots could be obtained:
PCA of methylation values
6.13.6 Command line options

MPI-IE workflow for WGBS analysis

usage example: WGBS -i read_input_dir -o output-dir mm10

usage: WGBS -i INDIR -o OUTDIR [-h] [-v] [-ext EXT] [-reads READS READS]
        [-c CONFIGFILE] [-clusterConfigFile CLUSTERCONFIGFILE] [-j INT]
        [--local] [-keepTemp] [-snakemakeOptions SNACKEMAKEOPTIONS]
        [-DAG] [--version] [-emailAddress EMAILADDRESS]
        [-smtpServer SMTPSERVER] [-smtpPort SMTPPORT] [-onlySSL]
        [-emailSender EMAILSNDER] [-smtpUsername SMTPUSERNAME]
        [-smtpPassword SMTPPASSWORD] [-downsample INT] [-trim]
        [--trimmer {cutadapt,trimgalore,fastp}]
        [--trimmerOptions TRIMMEROPTIONS] [-fastqc] [-bcExtract]
        [--bcPattern BCPATTERN] [-UMIDedup] [-UMIDedupSep UMIDEDUPSEP]
        [-UMIDedupopts UMIDEDUPOPTS] [-bwBinSize BWBINSIZE]
        [-plotFormat STR] [-blacklist BLACKLIST]
        [-targetRegions TARGETREGIONS] [-sampleSheet SAMPLESHEET]
        [-noAutoMethylationBias] [-maxDist MAXDIST] [-minCpGs MINCPGS]
        [-minMethDiff MINMETHDIFF] [-minCoverage MINCOVERAGE]
        [-FDR FDR] [-MethylDackelOptions METHYLDACKELOPTIONS]
        [-fromBAM] [-skipBamQC] [-bamExt BAMEXT] [-singleEnd]
        [-DMRprograms DMRPROGRAMS] [-metileneOptions METILENEOPTIONS]
        [-skipDOC] [-GCbias]
        GENOME
Positional Arguments

**GENOME**
Genome acronym of the target organism. Either a yaml file or one of: mm10, SchizoSPombe_ASM294v2, hg38, dm3, mm10_gencodeM13, mm9, GRCz10, hs37d5, dm6

Required Arguments

- **-i, --input-dir**
  input directory containing the FASTQ files, either paired-end OR single-end data
- **-o, --output-dir**
  output directory

General Arguments

- **-v, --verbose**
  verbose output (default: 'False')
- **--ext**
  Suffix used by input fastq files (default: "".fastq.gz"").
- **--reads**
  Suffix used to denote reads 1 and 2 for paired-end data. This should typically be either '_1' '_2' or '_R1' '_R2' (default: [''_R1', '_R2']). Note that you should NOT separate the values by a comma (use a space) or enclose them in brackets.
- **-c, --configFile**
  configuration file: config.yaml (default: 'None')
- **--clusterConfigFile**
  configuration file for cluster usage. In absence, the default options specified in defaults.yaml and workflows/[workflow]/cluster.yaml would be selected (default: 'None')
- **-j, --jobs**
  maximum number of concurrently submitted Slurm jobs / cores if workflow is run locally (default: '12')
- **--local**
  run workflow locally; default: jobs are submitted to Slurm queue (default: 'False')
- **--keepTemp**
  Prevent snakemake from removing files marked as being temporary (typically intermediate files that are rarely needed by end users). This is mostly useful for debugging problems.
- **--snakemakeOptions**
  Snakemake options to be passed directly to snakemake, e.g. use –snakemakeOptions='--dryrun --rerun-incomplete --unlock --forceall'. WARNING! ONLY EXPERT USERS SHOULD CHANGE THIS! THE DEFAULT VALUE WILL BE APPENDED RATHER THAN OVERWRITTEN! (default: [''–use-conda''])
- **--DAG**
  If specified, a file ending in _pipeline.pdf is produced in the output directory that shows the rules used and their relationship to each other.
- **--version**
  show program’s version number and exit

Email Arguments

- **--emailAddress**
  If specified, send an email upon completion to the given email address
- **--smtpServer**
  If specified, the email server to use.
- **--smtpPort**
  The port on the SMTP server to connect to. A value of 0 specifies the default port.
- **--onlySSL**
  The SMTP server requires an SSL connection from the beginning.
--emailSender  The address of the email sender. If not specified, it will be the address indicated by --emailAddress
--smtpUsername  If your SMTP server requires authentication, this is the username to use.
--smtpPassword  If your SMTP server requires authentication, this is the password to use.

Options

--downsample  Downsample the given number of reads randomly from each FASTQ file (default: "None")
--trim  Activate fastq read trimming. If activated, Illumina adaptors are trimmed by default. Additional parameters can be specified under --trimmerOptions. (default: "True")
--trimmer  Possible choices: cutadapt, trimgalore, fastp
  Trimming program to use: Cutadapt, TrimGalore, or fastp. Note that if you change this you may need to change --trimmerOptions to match! (default: "fastp")
--trimmerOptions  Additional option string for trimming program of choice. (default: ":q 5 -l 30 -M 5")
--fastqc  Run FastQC read quality control (default: 'False')
--bcExtract  To extract umi barcode from fastq file via UMI-tools and add it to the read name (default: 'False')
--bcPattern  The pattern to be considered for the barcode. 'N' = UMI position (required) 'C' = barcode position (optional) (default: ")
--UMIDedup  Deduplicate bam file based on UMIs via umi_tools dedup that are present in the read name. (default: 'False')
--UMIDedupSep  umi separation character that will be passed to umi_tools.(default: "_")
--UMIDedupOpts  Additional options that will be passed to umi_tools.(default: ")
--bwBinSize  Bin size of output files in bigWig format (default: '25')
--plotFormat  Possible choices: png, pdf, None
  Format of the output plots from deepTools. Select 'none' for no plots (default: ""png"")
--blacklist  Bed file(s) with positions to mask for methylation calling. Useful for masking SNPs in your strain of interest. (default: 'None')
--targetRegions  Bed file(s) with regions of interest to evaluate methylation for. (default: 'None')
--sampleSheet  Perform differential methylation analysis between groups of samples by providing a text file with sample information to use for statistical analysis. (default: 'None')
--noAutoMethylationBias  If specified, MethylDackel mbias will NOT be run and the suggested parameters from it will NOT be used for methylation extraction. You can instead supply them manually in --MethylDackelOptions.
--maxDist  The maximum distance between CpGs in a DMR (for metilene, default: '300')
--minCpGs  The minimum number of CpGs in a DMR (for metilene, default: '10')
--minMethDiff  The minimum methylation change in methylation for CpG inclusion in DMR detection (for metilene, default: '0.1')

--minCoverage  The minimum coverage needed for across all samples for a CpG to be used in DMR calling and PCA. Note that you can change this value without overwritting the DMR output. Default: '5'

--FDR  FDR threshold for returned DMRs (default: '0.1')

--MethylDackelOptions  Options to pass to MethylDackel extract. You are highly advised NOT to set a minimum coverage at this step. Default: "–mergeContext –maxVariantFrac 0.25 –minDepth 4"

--fromBAM  If specified, the input is taking from BAM files containing alignments rather than fastq files. See also --bamExt.

--skipBamQC  If specified, filtering of input bam files, as well as calculation of quality metrics, will be skipped.

--bamExt  If --fromBAM is specified, this is the expected file extension. Removing it yields sample names. Default: "*.bam"

--singleEnd  If --fromBAM is specified, this indicates that the input BAM files contain paired-end data. The option is ignored unless --fromBAM is given.

--DMRprograms  If a sample sheet is provided, use the specified DMR-calling programs. Multiple programs can be comma-separated with no spaces (e.g., 'metilene, dmrseq, DSS'). The available programs are metilene, dmrseq, and DSS (note that this is very slow). Default: "metilene, dmrseq".

--metileneOptions  Options to pass to metilene. Default: 'None'

--skipDOC  Skip depth of coverage calculation with deepTools.

--GCbias  Perform GC bias calculation with deepTools.

code @ github.

### 6.14 snakePipes News

#### 6.14.1 snakePipes 2.4.1

- Fixed sampleSheet splitting for multiple pairwise comparisons when group "All" is not listed.

#### 6.14.2 snakePipes 2.4.0

- Added support for multiple pairwise comparisons for DESeq2, sleuth, and rMats in the mRNA-seq workflow, as well as for DESeq2 in the noncoding-RNA-seq workflow.
- Loompy from conda is now used in mode STARsolo in scRNA-seq workflow.
- Added bamExt to mRNA-seq and noncoding-RNA-seq commandline arguments.
- Added multi-thread support to rMats in mRNA-seq workflow.
- Fixed deepTools GC bias command with SE reads.
- Bumped HiC explorer version.
• Fixed STARsoloCoords for Custom kit.

6.14.3 snakePipes 2.3.1

• Fixed aligner options for bwa in DNA-mapping.
• Fixed allelic mode for single end reads.
• Bumped hiC explorer version in HiC.

6.14.4 snakePipes 2.3.0

• Deprecated mode Gruen in scRNAseq.
• scRNAseq mode Alevin now outputs spliced/unspliced counts for RNA velocity estimation based on Soneson et al. 2020, bioRxiv https://doi.org/10.1101/2020.03.13.990069.
• Fixed "external_gene_name" and "Status" columns in DESeq2 html report.
• Removed warning when sample names start with a number.

6.14.5 snakePipes 2.2.3

• Genrich will now run if sampleSheet without replicates is provided.
• Updated zenodo link to mouse genome GRCm38/mm10.
• Fixed start coordinates in Filtered results bed from CSAW.

6.14.6 snakePipes 2.2.2

• Fix DAG inconsistencies for ChIP-seq and ATAC-seq ran fromBAM and from -d.
• DESeq2 Rmd file is not deleted anymore in noncoding-RNAseq.
• Fixed labels in deepTools commands.
• Allele_info is now boolean.

6.14.7 snakePipes 2.2.1

• Fix a bug in DAG for ChIPseq allelic with CSAW.
• Fixed deepTools qc DAG for ChIPseq with spikein.
• Added DAG test for allelic ChIPseq.
• Fixed a bug with deepTools QC for allelic mRNAseq.

6.14.8 snakePipes 2.2.0

• Added Alevin mode in scRNA workflow
• Added a new conda environment using to call AlevinQC.
• Added filtering of empty drops with Dropletutils to scRNA-seq mode STARsolo
• Added spikein normalization to ChIPseq workflow
• Added hybrid genome creation to createIndices
• Added STARsolo report for all samples to STARsolo output folder
• FASTQ1 and FASTQ2 are not localrules anymore due to buggy logging
• Included optional differential splicing analysis using rmats within mRNA-seq workflow
• Symlinks in the output path are relative
• Increased BBmap version
• Increased STAR version to 2.7.4a in scRNAseq, noncoding-RNA-seq and mRNA-seq workflows
• Fixed snakemake version at 5.18.0 due to a bug in DAG handling
• Minor changes to shared FastQC and multiQC rule with regards to scRNA-seq workflow.
• Fixed issue with missing input for running the DNA-mapping Snakefile
• Fixed rule TrimGalore for single end reads
• deepTools heatmaps for differentially bound regions are now ordered by sample sheet condition
• Genrich is now run on namesorted bams
• Workflow help message now points to example sampleSheet on GitHub
• organismsDir can now be updated with snakePipes config mode "recycle"

Note: Please be aware that this version requires regeneration of STAR indices!

6.14.9 snakePipes 2.1.2

• small bug fix: SE mode in noncoding-RNA-seq pipeline

6.14.10 snakePipes 2.1.1

• small bug fix: a typo in atac-seq pipeline

6.14.11 snakePipes 2.1.0

• Snakemake version is bumped to 5.13.0
• Updated docs on running single snakefiles
• Added user-input target regions and freetext parameters to differential methylation analysis with metilene
• Added PCA to metilene report in WGBS
• Added Genrich support for SE data
• Edited symlinking rules to ln -s or python
• TMPDIR is now passed at rule-level to the shell
• Added logs in a couple of places
• Added –skipBamQC to WGBS to be included with –fromBAM to suppress recalculation of QC metrics on the bam file
• Added tempDir check to snakePipes info
• Added –oldConfig and –configMode options to snakePipes config that allow passing a copy of an existing pre-configured config file instead of passing the single paths. Previous mode can be used with –configMode manual (default), the new mode with –configMode recycle.
• Updated histoneHMM version to 1.8. Changed number formatting in histoneHMM output from scientific to general.
• Small fixes in DESeq2 report for noncoding-RNA-seq, WGBS reports
• Fixed –verbose in WGBS
• Fixed an important bug in differential binding analysis with CSAW (mismatch between sampleSheet rownames and countdata colnames).

6.14.12 snakePipes 2.0.2

• DAG print is now moved to _after_ workflow run execution such that any error messages from e.g. input file evaluation do not interfere with the DAG and are visible to the user.
• Fixed fastqc for –forBAM .
• Fixed DESeq2 report failure with just 1 DEG.
• Updated links to test data and commands on zenodo in the docs.
• SampleSheet check now explicitly checks for tab-delimited header.
• Fixed metilene groups, as well methylation density plots in WGBS.

6.14.13 snakePipes 2.0.1

• Fixed a bug in snakePipes config that caused the toolsVersion variable to be removed from defaults.yaml. This is likely related to issue #579.

6.14.14 snakePipes 2.0.0

• Added a noncoding-RNA-seq workflow and renamed RNA-seq to mRNA-seq for clarity. The noncoding workflow will also quantify protein coding genes, but its primary use is analyzing repeat expression.
• In order to use the noncoding-RNA-seq workflow organism YAML files must now include a rmsk_file entry.
• Fixed STAR on CIFS mounted VFAT file systems (issue #537).
• Added mode STARsolo to scRNAseq. This mode is now default.
• Added log fold change shrinkage with "apeglm" to DESeq2 basic in the mRNAseq workflow. Two versions of results tables (with and without shrinkage) are now written to the DESeq2 output folder.
• Added Genrich as peakCaller option to ChIPseq and ATACseq.
• Added HMMRATAc as peakCaller option to ATACseq.
• ATAC-seq short bam (filtered for short fragments) is now stored in a separate folder.
Note: Please be aware that this version requires regeneration of STAR indices!

6.14.15 snakePipes 1.3.2

- Fixed missing multiQC input in allelic RNAseq
- Added sample check to those workflows that were missing it.

6.14.16 snakePipes 1.3.1

- Support for snakeMake 5.7.0

6.14.17 snakePipes 1.3.0

- Overhauled WGBS pipeline
- Standardized options to be camelCase
- Further standardized options between pipelines
- UMI handling is now available in most pipelines
- The `--fromBAM` option is now available and documented
- Users can now change the read number indicator ("_R1" and "_R2" by default) as well as the fastq file extension on the command line.
- Added the preprocessing pipeline, prevented python packages in users’ home directories from inadvertently being used.
- Added a `snakePipes config` command that can be used in lieu of editing defaults.yaml

6.14.18 snakePipes published

SnakePipes was published: https://www.ncbi.nlm.nih.gov/pubmed/31134269

6.14.19 snakePipes 1.2.3

- Updated citation for snakePipes
- Fixed replicate check for samples with trailing spaces in names
- Fixed input filtering in CSAW
- Several allele-specific RNAseq fixes
- ATACseq peakQC is now run on fragment-size filtered bam
- Fixed Salmon output (Number of Reads output in "prefix_counts.tsv" files and file naming)
- Fixed CSAW QC plot error with single end reads
- Updated histone HMM environment to a working conda version
- Salmon_wasabi is now a localrule
6.14.20 snakePipes 1.2.2

- Fixed a bug in the ATAC-seq environment where GenomeInfoDbData was missing.
- Also an occasional issue with CSAW

6.14.21 snakePipes 1.2.1

- Fixed a typo in `createIndices`.
- Implemented complex experimental design in RNAseq (differential gene expression), ChIP/ATACseq (differential binding).
- Fixed an issue with ggplot2 and log transformation in RNAseq report Rmd.
- fastqc folder is created and its content will be added to multiqc only if fastqc flag is called.
- fastqc-trimmed folder is created and its content will be added to multiqc only if both fastqc and trim flags are called.

6.14.22 snakePipes 1.2.0

- A number of minor bug fixes across all of the pipelines
- Updates of all tool versions and switching to R 3.5.1
- A `snakePipes flushOrganisms` option was added to remove the default organism YAML files.
- Renamed `--notemp` to `--keepTemp`, which should be less confusing

6.14.23 snakePipes 1.1.2

- A number of minor bug fixes and enhancements in the HiC and WGBS pipelines
- The RNA-seq pipeline now uses samtools for sorting. This should avoid issues with STAR running out of memory during the output sorting step.
- Increased the memory allocation for MACS2 to 8GB and bamPEFragmentSize to 3G
- Fixed the scRNA-seq pipeline, which seems to have been broken in 1.1.1

6.14.24 snakePipes 1.1.1

- Fixed some conda environments so they could all be solved in a reasonable amount of time.
- Updated some WGBS memory limits

6.14.25 snakePipes 1.1.0

- A wide number of bug fixes to scRNA-seq and other pipelines. In particular, many memory limits were updated.
- An optional email can be sent upon pipeline completion.
- The RNA-seq pipeline can now produce a fuller report upon completion if you are performing differential expression.
- Sample merging in HiC works properly.
• GTF files are now handled more generically, which means that they no longer need to have _gencode and _ensembl in their path.

• WGBS:
  – Merging data from WGBS replicates is now an independent step so that dependent rules don’t have to wait for successful completion of single CpG stats but can go ahead instead.
  – Filtering of differential methylation test results is now subject to two user-modifiable parameters minAbs-Diff (default 0.2) and FDR (0.02) stored in defaults.yaml.
  – Methylene commandline parameters are now available in defaults.yaml. Defaults are used apart from requesting output intervals with any methylation difference (minMethDiff 0).
  – Additional diagnostic plots are generated - p value distribution before and after BH adjustment as well as a volcano plot.
  – Automatic reports are generated in every folder containing results of statistical analysis (single CpG stats, methylene DMR stats, user interval aggregate stats), as long as sample sheet is provided.
  – R sessionInfo() is now printed at the end of the statistical analysis.

• scRNAseq:
  – An extention to the pipeline now takes the processed csv file from Results folder as input and runs cell filtering with a range of total transcript thresholds using monocle and subsequently runs clustering, produces tsne visualizations, calculates top 2 and top10 markers per cluster and produces heatmap visualizations for these using monocle/seurat. If the skipRaceID flag is set to False (default), all of the above are also executed using RaceID.
  – Stats reports were implemented for RaceID and Monocle/Seurat so that folders Filtered_cells_RaceID and Filtered_cells_monocle now contain a Stats_report.html.
  – User can select a metric to maximize during cell filtering (cellFilterMetric, default: gene_universe).
  – For calculating median GPC, RaceID counts are multiplied by the TPC threshold applied (similar to ‘down-scaling’ in RaceID2).

• all sample sheets now need to have a "name" and a "condition" column, that was not consistent before

• consistent –sampleSheet [FILE] options to invoke differential analysis mode (RNA-seq, ChIP-seq, ATAC-seq), –DE/--DB were dropped

6.14.26 snakePipes 1.0.0 (king cobra) released

9.10.2018
First stable version of snakePipes has been released with various feature improvements. You can download it from GitHub

6.14.27 snakePipes preprint released

We released the preprint of snakePipes describing the implementation and usefulness of this tool in integrative epigenomics analysis. Read the preprint on bioRxiv

[code @ github]
Indices and tables

- genindex
- modindex
- search

code @ github.