VSN-Pipelines-examples

Aug 25, 2021

Contents

This tutorial shows the steps to analyze a typical scRNA-seq dataset with a single sample. We will use PBMC data available from the 10x Genomics support website. The same dataset was used in the DSL1 version of this pipeline, described in the SCENIC protocol tutorial (here).

CHAPTER 1

Prepare 10x input data

The “Feature / cell matrix (filtered)” data was downloaded from 10x Genomics, here.

```
wget http://cf.10xgenomics.com/samples/cell-exp/3.0.0/pbmc_10k_v3/pbmc_10k_v3_
↪filtered_feature_bc_matrix.tar.gz
```

When using 10x data as an input, the pipeline assumes the files are in the typical Cell Ranger directory structure. This is not the case when downloading the processed counts from the 10x website, so we will put them into the proper format:

```
mkdir -p pbmc10k/outs/
tar xvf pbmc_10k_v3_filtered_feature_bc_matrix.tar.gz -C pbmc10k/outs/
```

which results in:

```
$ tree pbmc10k
pbmc10k
└── outs
    └── filtered_feature_bc_matrix
        ├── barcodes.tsv.gz
        ├── features.tsv.gz
        └── matrix.mtx.gz

2 directories, 3 files
```

So, in the nextflow config file, generated in the following step, the tenx input channel should point to the `outs` folder. For example:

```
params.data.tenx.cellranger_mex = '/home/cflerin/analysis/pbmc10k/dsl2_0.19.0/pbmc10k/
↪outs'
```

CHAPTER 2

Setup the VSN-pipelines project

2.1 Update the repository

Pull/update the vsn-pipelines repository cached by nextflow. Here, we use the `-r` flag to specify the pipeline version to use:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
```

2.2 Build the config file

We use a combination of profiles to build the config file:

- `tenx`: defines the input data type
- `single_sample_scenic`: loads the basic parameters to run the single_sample and scenic workflows
- `scenic_use_cistarget_motifs` and `scenic_use_cistarget_tracks`: includes parameters to specify the location of the cistarget database files
- `hg38`: specifies the genome. Other options are: `hg19`, `dm6`, `mm10`.
- `singularity` (or `docker`): specifies container system to use to run the processes

```
nextflow config vib-singlecell-nf/vsn-pipelines \
    -profile tenx,single_sample_scenic,scenic_use_cistarget_motifs,scenic_use_
↪cistarget_tracks,hg38,singularity \
    > pbmc10k.vsn-pipelines.complete.config
```

Important variables to check in the config:

- `singularity.runOptions` (or `docker.runOptions`): making sure the correct volume mounts are specified (requires the user home folder (included by default in Singularity), and the location of the data).
- `params.global.project_name` (optional): will control the naming of the output files.

- `params.sc.scope.tree.level_${X}` (optional): controls the labeling of the loom file when uploaded to the SCope viewer.
- `params.sc.scanpy.filter`: filtering settings for the Scanpy steps.
- `params.sc.scanpy.feature_selection`: controls how highly variable genes are selected.
- `params.sc.scanpy.clustering`: controls cluster settings. In the example here, we select two clustering resolutions by using `resolutions = [0.4,0.8]`.

Specifying compute resource usage in the config:

- The global executor (`process.executor`) is set to `local` by default. It can be changed to `qsub`, etc. to run specific processes as jobs. The executor parameter can be added to specific labels to run only these processes as jobs. Typically the GRN step should be submitted as a job (`compute_resources__scenic_grn`).
- The number of cpus and memory usage can be adjusted for each label.

The complete config file used here is available at: pbmc10k/pbmc10k.vsn-pipelines.complete.config.

CHAPTER 3

Run the VSN-pipelines project

3.1 First pass

While the overall goal is to run the “best practices steps” and SCENIC together, we can first skip running SCENIC, and focus on getting the filtering and preprocessing steps correct. Then, we can move on to run the resource-intensive SCENIC steps. Even though we created a profile with single_sample and scenic options together, we can run just the single_sample workflow first:

```
nextflow -C pbmc10k.vsn-pipelines.complete.config \
    run vib-singlecell-nf/vsn-pipelines \
    -entry single_sample
```

Now, the QC reports can be inspected (see `out/notebooks/intermediate/pbmc10k.SC_QC_filtering_report.html`, either the original ipynb, or the converted html file). The cell and gene filters can be updated by editing the config file. For example, the relevant filters used here are:

```
params {
    sc {
        scanpy {
            filter = {
                cellFilterMinNGenes = 200
                cellFilterMaxNGenes = 4000
                cellFilterMaxPercentMito = 0.15
                geneFilterMinNCells = 3
            }
        }
    }
}
```

Re-run the pipeline as many times as needed (with `resume` to skip alread-completed steps) to select the proper filters:

```
nextflow -C pbmc10k.vsn-pipelines.complete.config \
    run vib-singlecell-nf/vsn-pipelines \
    -entry single_sample
```

3.2 Second pass

Once the cell and gene filters look ok, we can re-start the pipeline with the full SCENIC steps enabled. This will re-run any steps in which the parameters have changed (e.g. the filtering and downstream steps), while skipping the initial conversion, etc. when the `-resume` option is used:

```
nextflow -C pbmc10k.vsn-pipelines.complete.config \
    run vib-singlecell-nf/vsn-pipelines \
    -entry single_sample_scenic \
    -resume
```

CHAPTER 4

Results

Once the pipeline is complete (approximately 2 hours on a HPC system using 15 processes for the SCENIC GRN step), the output will be the following files (display truncated):

```
$ tree out
out/
├── data
│   ├── intermediate
│   │   └── [...]
│   └── pbmc10k.PBMC10k_DSL2.single_sample.output.h5ad
├── loom
│   ├── pbmc10k.SCENIC_SCope_output.loom
│   └── pbmc10k.SCope_output.loom
├── nextflow_reports
│   ├── execution_report.html
│   ├── execution_timeline.html
│   ├── execution_trace.txt
│   └── pipeline_dag.dot
├── notebooks
│   ├── intermediate
│   ├── pbmc10k.merged_report.html
│   ├── pbmc10k.merged_report.ipynb
│   ├── pbmc10k.merged_report.louvain_0.4.html
│   ├── pbmc10k.merged_report.louvain_0.4.ipynb
│   ├── pbmc10k.merged_report.louvain_0.8.html
│   └── pbmc10k.merged_report.louvain_0.8.ipynb
└── scenic
    └── pbmc10k
        ├── arboreto_with_multiprocessing
        │   ├── pbmc10k__adj.tsv
        │   └── pbmc10k.filtered.loom
        ├── aucell
        │   ├── pbmc10k__auc_mtf.loom
        │   ├── pbmc10k__auc_trk.loom
        │   └── pbmc10k.filtered.loom
        │
```

(continues on next page)

(continued from previous page)

```
        ├── cistarget
        │   ├── pbmc10k.filtered.loom
        │   ├── pbmc10k__reg_mtf.csv
        │   └── pbmc10k__reg_trk.csv
        ├── notebooks
        │   ├── SCENIC_report.html
        │   └── SCENIC_report.ipynb
        ├── SCENIC_output.loom
        └── SCENIC_SCope_output.loom
```

The final SCENIC output is packaged into a loom file, which includes the results of the parallel expression analysis (based on highly variable genes). This can be found at `out/loom/pbmc10k.SCENIC_SCope_output.loom`, and is ready to be uploaded to a SCope session. The output loom file from this analysis can be found on the SCENIC protocol SCope session.

Also included is `out/data/pbmc10k.PBMC10k_DSL2.single_sample.output.h5ad`, an anndata file generated by the Scanpy section of the pipeline, including the results of the expression analysis (but not results from SCENIC).

CHAPTER 5

10k PBMCs with multiple SCENIC iterations

This tutorial follows closely the following 10x PBMC case study:

- Analysis of 10k PBMCs from a single healthy human donor

However, here we focus on setting up the workflow for running SCENIC multiple times and automatically integrating the results. Please refer to the above case study for the explanation of the remainder of the setup steps.

5.1 Setup the VSN-pipelines project

5.1.1 Update the repository

Pull/update the vsn-pipelines repository cached by nextflow. Here, we use the `-r` flag to specify the pipeline version to use:

```
nextflow pull vib-singlecell-nf/vsn-pipelines -r v0.25.0
```

5.1.2 Build the config file

We use a combination of profiles to build the config file:

- `tenx`: defines the input data type
- `single_sample_scenic`: loads the basic parameters to run the single_sample and scenic workflows
- `scenic_multiruns`: loads the multiruns parameters
- `scenic_use_cistarget_motifs` and `scenic_use_cistarget_tracks`: includes parameters to specify the location of the cistarget database files
- `hg38`: specifies the genome. Other options are: `hg19`, `dm6`, `mm10`.
- `singularity` (or `docker`): specifies container system to use to run the processes

```
nextflow config vib-singlecell-nf/vsn-pipelines \
    -profile tenx,single_sample_scenic,scenic_multiruns,scenic_use_cistarget_motifs,
↪scenic_use_cistarget_tracks,hg38,singularity \
    > pbmc10k.vsn-pipelines.complete.config
```

Important **SCENIC multi-runs parameters** to check in the config:

- `params.sc.scenic.numRuns`: the number of SCENIC iterations to run. Using `10` will give a good balance with computation time, while `100` will provide an exhaustive evaluation of the regulon and target gene characteristics.
- `params.sc.scenic.aucell.min_genes_regulon`: The threshold used for filtering the regulons based on the number of target genes. Regulons are keps only if they contain this number of genes or greater. This parameter should be considered in proportion to the number of runs (default: 5).
- `params.sc.scenic.aucell.min_regulon_gene_occurrence`: The threshold used for filtering the genes bases on their occurrence. Target genes that occur this many times or less across all of the runs are discarded. This parameter should be considered in proportion to the number of runs (default: 5).

Specifying compute resource usage in the config:

- The global executor (`process.executor`) is set to `local` by default. It can be changed to `qsub`, etc. to run specific processes as jobs. The executor parameter can be added to specific labels to run only these processes as jobs. Typically the GRN step should be submitted as a job (`compute_resources__scenic_grn`), especially for the multi-runs approach.
- The number of cpus and memory usage can be adjusted for each label.

The complete config file used here is available at: pbmc10k/pbmc10k.vsn-pipelines.scenic-multiruns.complete.config.

5.2 Run the VSN-pipelines project

5.2.1 Testing the settings (optional)

With the SCENIC multi-runs approach, it is highly recommended to test the approach with `params.sc.scenic.numRuns` set to a small number (e.g. `2`).

It may also be useful to use a small test dataset for this purpose:

```
wget https://raw.githubusercontent.com/aertslab/SCENICprotocol/master/example/sample_
↪data_small.tar.gz
tar xzvf sample_data_small.tar.gz
```

This can be loaded in the config by temporarily replacing the input data with:

```
data {
    tenx {
        cellranger_mex = 'sample_data/outs'
    }
}
```

As this is a small dataset and the `numRuns` parameter is set to a small number, also change these settings:

```
scanpy {
        filter {
            cellFilterMinNGenes = 1
```

(continues on next page)

(continued from previous page)

```
        }
        neighborhood_graph {
            nPcs = 2
        }
        dim_reduction {
            pca {
                nComps = 2
            }
        }
    }
    scenic {
        aucell {
            min_genes_regulon = 0
            min_regulon_gene_occurrence = 0
        }
    }
```

5.2.2 First pass

As in the original PBMC10k tutorial, we can first run without the SCENIC steps to get the filtering and other parameters set correctly. Re-run the pipeline as many times as needed (with `resume` to skip alread-completed steps) to select the proper filters:

```
nextflow -C pbmc10k.vsn-pipelines.complete.config \
    run vib-singlecell-nf/vsn-pipelines \
    -entry single_sample \
    -r v0.25.0 -resume
```

5.2.3 Second pass

With the parameters set, the full multi-runs workflow can be run:

```
nextflow -C pbmc10k.vsn-pipelines.complete.config \
    run vib-singlecell-nf/vsn-pipelines \
    -entry single_sample_scenic \
    -r v0.25.0 -resume
```

This can take a long time to run, depending on the number of iterations used.

5.3 Results

Once the pipeline is complete, the output will be the following files (display truncated):

```
$ tree out
out/
├── data
│   ├── intermediate
│   │   └── [...]
│   └── pbmc10k.PBMC10k_DSL2.single_sample.output.h5ad
├── loom
│   ├── pbmc10k.SCENIC_SCope_output.loom
```

(continues on next page)

(continued from previous page)

```
│   └── pbmc10k.SCope_output.loom
├── nextflow_reports
│   ├── execution_report.html
│   ├── execution_timeline.html
│   ├── execution_trace.txt
│   └── pipeline_dag.dot
├── notebooks
│   ├── intermediate
│   ├── pbmc10k.merged_report.html
│   ├── pbmc10k.merged_report.ipynb
│   ├── pbmc10k.merged_report.louvain_0.4.html
│   ├── pbmc10k.merged_report.louvain_0.4.ipynb
│   ├── pbmc10k.merged_report.louvain_0.8.html
│   └── pbmc10k.merged_report.louvain_0.8.ipynb
└── scenic
    └── pbmc10k
        ├── arboreto_with_multiprocessing
        │   ├── run_1
        │   │   └── sample_data__run_1__adj.tsv
        │   └── run_2
        │       └── sample_data__run_2__adj.tsv
        ├── aucell
        │   ├── run_1
        │   │   └── sample_data__run_1__auc_mtf.loom
        │   └── run_2
        │       └── sample_data__run_2__auc_mtf.loom
        ├── cistarget
        │   ├── run_1
        │   │   └── sample_data__run_1__reg_mtf.csv.gz
        │   └── run_2
        │       └── sample_data__run_2__reg_mtf.csv.gz
        ├── multi_runs_aucell
        │   └── multi_runs_regulons_auc_mtf.tsv
        ├── multi_runs_cistarget
        │   ├── multi_runs_features_mtf.csv.gz
        │   └── multi_runs_regulons_mtf.pkl.gz
        ├── multi_runs_looms
        │   └── multi_runs_regulons_auc_mtf.loom
        ├── multi_runs_regulons_mtf
        │   ├── BRF1(+).tsv
        │   ├── FOS(+).tsv
        │   └── regulons.tsv
        ├── notebooks
        │   ├── SCENIC_report.html
        │   └── SCENIC_report.ipynb
        └── SCENIC_SCope_output.loom
```

The final output is similar to that of a SCENIC pipeline with a single iteration, with the exception of some additional files being stored in `out/scenic/<sampleId>`:

- GRN, cisTarget, and AUCell outputs for each run/iteration in their respective directories.
- In the `out/scenic/<sampleId>/multi_runs_regulons_mtf/` directory (and optionally `multi_runs_regulons_trk` if track databases were used):
  - `regulons.tsv`: Contains a list of all regulons found, along with their occurrence (count) across all SCENIC iterations.

  - A file for each regulon (e.g. `BRF1(+).tsv`). Each file contains two columns: 1) the target gene name, and 2) the number of times that gene occurred across all SCENIC iterations.

The final SCENIC output is packaged into a loom file, which includes the results of the multi-runs analysis. This can be found at `out/scenic/<sampleId>/SCENIC_SCope_output.loom`, and is ready to be uploaded to a SCope session.

CHAPTER 6

Hung, R.-J. et al., 2019 - A cell atlas of the adult Drosophila midgut

Some links related to the case study:

- Paper: https://www.pnas.org/content/117/3/1514.abstract
- GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120537

6.1 Analysis of 10x Genomics Samples

6.1.1 BBKNN and SCENIC

The following command was used to generate the config:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
nextflow config vib-singlecell-nf/vsn-pipelines -profile \
  sra,cellranger,pcacv,bbknn,dm6,scenic,scenic_use_cistarget_motifs,scenic_use_
↪cistarget_tracks,singularity \
  > nextflow.config
```

The generated config is available at the `vsn-pipelines-examples` GitHub repository: hungr_2019/10x_bbknn_scenic.config. You should provide the lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry sra_cellranger_bbknn_scenic
```

The resulting loom file is available at hungr_2019_bbknn_scenic.loom, and is ready to be explored in SCope.

6.1.2 Harmony and SCENIC (append mode)

```
nextflow config \
    vib-singlecell-nf/vsn-pipelines \
    -profile tenx,pcacv,harmony,scenic_append_only,singularity \
    > nextflow.config
```

The generated config is available at the `vsn-pipelines` GitHub repository: hungr_2019/10x_harmony_scenic_append_only.config. You should update The lines commented with "TO EDIT" with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry harmony_scenic -resume
```

The resulting loom file is available at hungr_2019/harmony_scenic_append_only and is ready to be explored in SCope.

CHAPTER 7

Davie, K., Janssens, J., Koldere, D. et al., 2018 - A Single-Cell Transcriptome Atlas of the Aging Drosophila Brain.

Some links related to the case study:

- Paper: https://www.ncbi.nlm.nih.gov/pubmed/29909982
- GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE107451

7.1 Analysis of 10x Genomics Samples

7.1.1 BBKNN and SCENIC

The following command was used to generate the config:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
nextflow config vib-singlecell-nf/vsn-pipelines -profile \
  sra,cellranger,pcacv,bbknn,dm6,scenic,scenic_use_cistarget_motifs,scenic_use_
↪cistarget_tracks,singularity \
  > nextflow.config
```

The generated config is available at the `vsn-pipelines-examples` GitHub repository: daviek_2018/10x_bbknn_scenic.config. You should provide the lines commented with "TO EDIT" with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry sra_cellranger_bbknn_scenic
```

The resulting loom file is available at daviek_2018_bbknn_scenic.loom, and is ready to be explored in SCope.

7.1.2 Harmony and SCENIC (append mode)

```
nextflow config \
    vib-singlecell-nf/vsn-pipelines \
    -profile tenx,pcacv,harmony,scenic_append_only,singularity \
    > nextflow.config
```

The generated config is available at the `vsn-pipelines` GitHub repository: daviek_2018/10x_harmony_scenic_append_only.config. You should update The lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry harmony_scenic -resume
```

The resulting loom file is available at daviek_2018/harmony_scenic_append_only and is ready to be explored in SCope.

CHAPTER 8

Bageritz, J., Willnow, P., Valentini, E. et al., 2019 - Gene expression atlas of a developing tissue by single cell expression correlation analysis.

Some links related to the case study:

- Paper: https://www.nature.com/articles/s41592-019-0492-x
- GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE127832

8.1 Analysis of 10x Genomics Samples

8.1.1 BBKNN and SCENIC

The following command was used to generate the config:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
nextflow config vib-singlecell-nf/vsn-pipelines -profile \
  sra,cellranger,pcacv,bbknn,dm6,scenic,scenic_use_cistarget_motifs,scenic_use_
↪cistarget_tracks,singularity \
  > nextflow.config
```

The generated config is available at the `vsn-pipelines-examples` GitHub repository: bageritzj_2019/10x_bbknn_scenic.config. You should provide the lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry sra_cellranger_bbknn_scenic
```

The resulting loom file is available at bageritzj_2019_bbknn_scenic.loom, and is ready to be explored in SCope.

8.1.2 Harmony and SCENIC (append mode)

```
nextflow config \
    vib-singlecell-nf/vsn-pipelines \
    -profile tenx,pcacv,harmony,scenic_append_only,singularity \
    > nextflow.config
```

The generated config is available at the `vsn-pipelines` GitHub repository: bageritzj_2019/10x_harmony_scenic_append_only.config. You should update The lines commented with "TO EDIT" with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry harmony_scenic -resume
```

The resulting loom file is available at bageritzj_2019/harmony_scenic_append_only and is ready to be explored in SCope.

CHAPTER 9

Kurmangaliyev et al., 2019 - Modular transcriptional programs separately define axon and dendrite connectivity

Some links related to the case study:

- Paper: https://elifesciences.org/articles/50822
- GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE126139

9.1 Analysis of 10x Genomics Samples

9.1.1 BBKNN and SCENIC

The following command was used to generate the config:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
nextflow config vib-singlecell-nf/vsn-pipelines \
    -profile sra,cellranger,pcacv,bbknn,dm6,scenic,scenic_use_cistarget_motifs,scenic_
↪use_cistarget_tracks,singularity \
  > nextflow.config
```

The generated config is available at the `vsn-pipelines-examples` GitHub repository: kurmangaliyevyz_2019/10x_bbknn_scenic.config. You should provide the lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry sra_cellranger_bbknn_scenic
```

The resulting loom file is available at kurmangaliyevyz_2019_bbknn_scenic.loom, and is ready to be explored in SCope.

9.1.2 Harmony and SCENIC (append mode)

```
nextflow config \
    vib-singlecell-nf/vsn-pipelines \
    -profile tenx,pcacv,harmony,scenic_append_only,singularity \
    > nextflow.config
```

The generated config is available at the `vsn-pipelines` GitHub repository: kurmangaliyevyz_2019/10x_harmony_scenic_append_only.config. You should update The lines commented with "TO EDIT" with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry harmony_scenic -resume
```

The resulting loom file is available at kurmangaliyevyz_2019/harmony_scenic_append_only and is ready to be explored in SCope.

CHAPTER 10

Guo, X. et al., 2019 - The Cellular Diversity and Transcription Factor Code of Drosophila Enteroendocrine Cells.

Some links related to the case study:

- Paper: https://www.ncbi.nlm.nih.gov/pubmed/31851941
- GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132274

10.1 Analysis of 10x Genomics Samples

10.1.1 BBKNN and SCENIC

The following command was used to generate the config:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
nextflow config vib-singlecell-nf/vsn-pipelines -profile \
  sra,cellranger,pcacv,bbknn,dm6,scenic,scenic_use_cistarget_motifs,scenic_use_
↪cistarget_tracks,singularity \
  > nextflow.config
```

The generated config is available at the `vsn-pipelines-examples` GitHub repository: guox_2019/10x_bbknn_scenic.config. You should provide the lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry sra_cellranger_bbknn_scenic
```

The resulting loom file is available at guox_2019_bbknn_scenic.loom, and is ready to be explored in SCope.

10.1.2 Harmony and SCENIC (append mode)

```
nextflow config \
    vib-singlecell-nf/vsn-pipelines \
    -profile tenx,pcacv,harmony,scenic_append_only,singularity \
    > nextflow.config
```

The generated config is available at the `vsn-pipelines` GitHub repository: guox_2019/10x_harmony_scenic_append_only.config. You should update The lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry harmony_scenic -resume
```

The resulting loom file is available at guox_2019/harmony_scenic_append_only and is ready to be explored in SCope.

CHAPTER 11

Ji, T., et al., 2019 - Dynamic MAPK signaling activity underlies a transition from growth arrest to proliferation in Drosophila scribble mutant tumors.

Some links related to the case study:

- Paper: https://www.ncbi.nlm.nih.gov/pubmed/31371383
- GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE130566

11.1 Analysis of 10x Genomics Samples

11.1.1 BBKNN and SCENIC

The following command was used to generate the config:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
nextflow config vib-singlecell-nf/vsn-pipelines -profile \
  sra,cellranger,pcacv,bbknn,dm6,scenic,scenic_use_cistarget_motifs,scenic_use_
↪cistarget_tracks,singularity \
  > nextflow.config
```

The generated config is available at the `vsn-pipelines-examples` GitHub repository: jit_2019/10x_bbknn_scenic.config. You should provide the lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry sra_cellranger_bbknn_scenic
```

The resulting loom file is available at jit_2019_bbknn_scenic.loom, and is ready to be explored in SCope.

11.1.2 Harmony and SCENIC (append mode)

```
nextflow config \
    vib-singlecell-nf/vsn-pipelines \
    -profile tenx,pcacv,harmony,scenic_append_only,singularity \
    > nextflow.config
```

The generated config is available at the `vsn-pipelines` GitHub repository: jit_2019/10x_harmony_scenic_append_only.config. You should update The lines commented with "TO EDIT" with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry harmony_scenic -resume
```

The resulting loom file is available at jit_2019/harmony_scenic_append_only and is ready to be explored in SCope.

CHAPTER 12

Brunet Avalos, C. et al., 2019 - Single cell transcriptome atlas of the Drosophila larval brain

Some links related to the case study:

- Paper: https://elifesciences.org/articles/50354
- GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE134722

12.1 Analysis of 10x Genomics Samples

12.1.1 BBKNN and SCENIC

The following command was used to generate the config:

```
nextflow pull vib-singlecell-nf/vsn-pipelines
nextflow config vib-singlecell-nf/vsn-pipelines -profile \
  sra,cellranger,pcacv,bbknn,dm6,scenic,scenic_use_cistarget_motifs,scenic_use_
↪cistarget_tracks,singularity \
  > nextflow.config
```

The generated config is available at the `vsn-pipelines-examples` GitHub repository: brunetavalosc_2019/10x_bbknn_scenic.config. You should provide the lines commented with “TO EDIT” with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry sra_cellranger_bbknn_scenic
```

The resulting loom file is available at brunetavalosc_2019_bbknn_scenic.loom, and is ready to be explored in SCope.

12.1.2 Harmony and SCENIC (append mode)

```
nextflow config \
    vib-singlecell-nf/vsn-pipelines \
    -profile tenx,pcacv,harmony,scenic_append_only,singularity \
    > nextflow.config
```

The generated config is available at the `vsn-pipelines` GitHub repository: brunetavalosc_2019/10x_harmony_scenic_append_only.config. You should update The lines commented with "TO EDIT" with the correct information.

To start the pipeline, run the following command:

```
nextflow \
    -C nextflow.config \
    run vib-singlecell-nf/vsn-pipelines \
        -entry harmony_scenic -resume
```

The resulting loom file is available at brunetavalosc_2019/harmony_scenic_append_only and is ready to be explored in SCope.

CHAPTER 13

VSN-Pipelines-examples

13.1 Tutorials

13.1.1 Single-sample workflows

- Analysis of 10k PBMCs from a single healthy human donor (10x Genomics): Includes standard "best practices" analysis alongside SCENIC gene regulatory network inference.

SCENIC multi-runs

- Analysis of 10k PBMCs with multiple SCENIC iterations (10x Genomics): Includes standard "best practices" analysis alongside SCENIC gene regulatory network inference.

13.1.2 Sample aggregation workflows

- Analysis of multiple samples in the adult Drosophila midgut (Hung, R.-J. et al., 2019):
  - Obtain expression counts, combine samples, apply BBKNN batch correction, then run SCENIC gene regulatory inference.
  - Obtain expression counts, combine samples, apply Harmony batch correction, then append SCENIC results from the BBKNN run.
- Analysis of multiple samples in the adult Drosophila ageing brain (Davie, K., Janssens, J., Koldere, D. et al., 2018):
  - Obtain expression counts, combine samples, apply BBKNN batch correction, then run SCENIC gene regulatory inference.
  - Obtain expression counts, combine samples, apply Harmony batch correction, then append SCENIC results from the BBKNN run.

- Analysis of multiple samples in the third instar female larvae Drosophila wing disc (Bageritz, J., Willnow, P., Valentini, E. et al., 2019):
  - Obtain expression counts, combine samples, apply BBKNN batch correction, then run SCENIC gene regulatory inference.
  - Obtain expression counts, combine samples, apply Harmony batch correction, then append SCENIC results from the BBKNN run.
- Analysis of multiple samples in the Drosophila brain: (Kurmangaliyev et al., 2019):
  - Obtain expression counts, combine samples, apply BBKNN batch correction, then run SCENIC gene regulatory inference.
  - Obtain expression counts, combine samples, apply Harmony batch correction, then append SCENIC results from the BBKNN run.
- Analysis of multiple samples in the third instar female larvae Drosophila Enteroendocrine Cells (Guo, X. et al., 2019):
  - Obtain expression counts, combine samples, apply BBKNN batch correction, then run SCENIC gene regulatory inference.
  - Obtain expression counts, combine samples, apply Harmony batch correction, then append SCENIC results from the BBKNN run.
- Analysis of multiple samples in the Drosophila scribble mutant tumors (Ji, T., et al., 2019):
  - Obtain expression counts, combine samples, apply BBKNN batch correction, then run SCENIC gene regulatory inference.
  - Obtain expression counts, combine samples, apply Harmony batch correction, then append SCENIC results from the BBKNN run.
- Analysis of multiple samples in the Drosophila larval brain (Brunet Avalos, C. et al., 2019):
  - Obtain expression counts, combine samples, apply BBKNN batch correction, then run SCENIC gene regulatory inference.
  - Obtain expression counts, combine samples, apply Harmony batch correction, then append SCENIC results from the BBKNN run.