

BIOL 343

Applied Bioinformatics I

Building pipelines

Dr. Nic Wheeler

Learning Objectives

You will be able to:

- 1.

Managing computational pipelines

Problems when using many tools:

1. Poor file management

Input/output files end up in unpredictable locations

2. Uninformative file names

Names tend to get loooooong with suffixes like `_new_reallynew_1.txt`

3. Difficult to checkpoint

*Re-running one step sometimes requires re-running **all** the steps*

4. Difficult to take notes

When testing new tools, it's difficult to recall what worked and didn't work (and why)

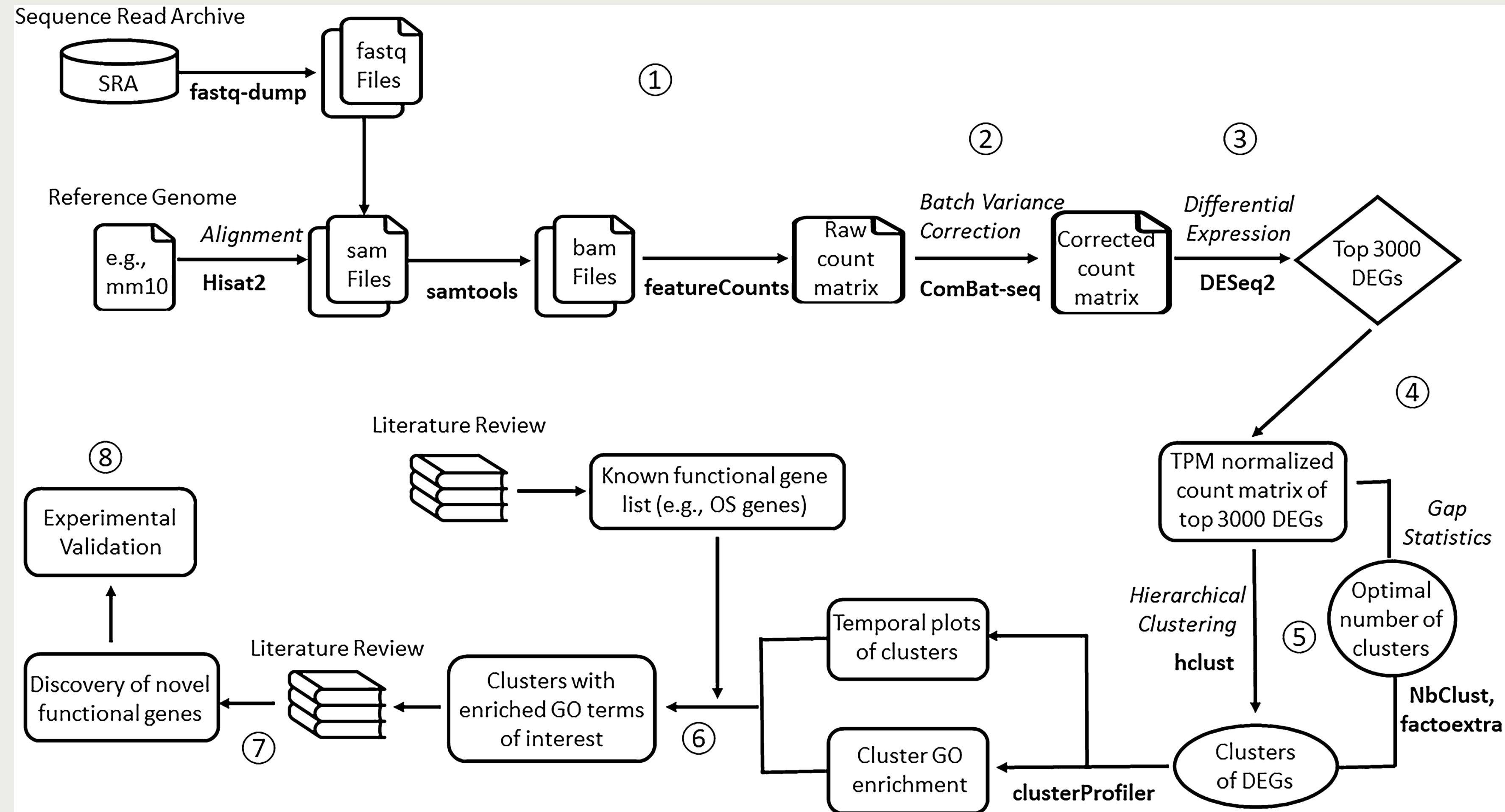


Workflow (pipeline) scripts

Managing computational pipelines

Options for managing a pipeline like this:

- Run one-by-one with CLI
- Run with Jupyter notebook
- Write a script



Managing computational pipelines

One-by-one

Good for alpha tests

Difficult to reproduce

Many opportunities for error

Jupyter notebook

Good when visualization/
manual inspection is necessary (beta tests)

Single-use

Doesn't scale

Scripting

Good when publishing/
distributing

Generic for multiple uses

Scales well

Managing computational pipelines

Scripting options

Shell script:

- Ex: https://github.com/zamanianlab/Filarid_IsoSeq/blob/master/main.sh
- Key features:
 - Can manage directories with variables
 - Checkpoint with comments (kind of)
 - Allows use of all command line tools, even Bash commands
 - No special language knowledge required
 - Could be made generic (but this one isn't)

Managing computational pipelines

Scripting options



Nextflow script:

- Ex: https://github.com/zamanianlab/Core_RNAseq-nf/blob/master/WB-pe.nf
- Key features:
 - Built-in directory and file management
 - Built-in checkpointing
 - Generic (with some restrictions)
 - Allows use of all command line tools, even Bash commands
 - Requires some Groovy knowledge
 - Large user base, lots of support

Managing computational pipelines

Scripting options



Snakelike script:

- Key features:
 - Built-in directory and file management
 - Built-in checkpointing
 - Generic (with some restrictions)
 - Allows use of all command line tools, even Bash commands
 - Uses Python!
 - Uses conda!
 - Integrates with Slurm!

Scripting pipelines in BIOL 343

- Each student will script their own pipelines
- Can choose which type you want to use
- Can use all available documentation/support online
- Can use your own conda envs

Live code activity...

1. Complete the snakemake tutorial on BOSE
 - a. Start a VS Code session
 - b. Install the snakemake extension
 - c. Start at Step 2 (above link) to setup the env, then go here and edit your snakefile in a VS Code session