The Livnat Lab

How Evolution Happens

The MEMDS analysis pipeline

Quick start guide

By

Assaf Malik, Evgeni Bolotin, Daniel Melamed, Yuval Nov and Adi Livnat

Department of Evolution - University of Haifa;

Corresponding author: Adi Livnat;
Email: adi.livnat@sci.haifa.ac.il

Table of contents

Introduction

The “Quick run guide” provides a brief outline of the MEMDS pipeline steps and of scripts needed to run each step. It aims to serve as a cheat sheet for users that already have some experience with the pipeline and just need a quick reminder about its structure. It encompasses “Quick start guide” chapter in the main MEMDS pipeline guide, and is provided as a separate file for user convenience.

Quick start guide

Notes before the run

1) The “**$i”** symbol after command name indicates that multiple substeps are available for a given pipeline step, with available run options listed in parentheses.
For example: *bash my_script.sh $i (1-3)* means that pipeline step run by **my_script.sh** is composed of 3 substeps and the user needs to choose which one they want to activate. To complete the pipeline step, all relevant substeps need to be run in a sequential order, save for substeps marked as “optional”.

2) When running on a cluster, for each job submitted by the script the following message would appear: “Submitted batch job #job_serial_number”. **Before moving to the next option in the script or to the next step in the pipeline**, a**lways check that all running jobs were completed successfully.** To check job status, use the following commands (for SLURM systems):

a) *squeue -u “username”| grep -c “username”* – this command displays number of jobs in queue for account defined by the “username”. Completed or canceled jobs would be removed from the queue.

b) *sacct -u “username”* – this command lists all the jobs that ran on the account defined by the “username” at current session. In the last column it indicates for each job if it is completed, canceled or still running. Ensure that all relevant jobs have ‘Completed’ status before submitting new ones!

3) **Job submission commands vary between different cluster systems**. Before the run remember to update the **“sbatch”** command in the pipeline job submission bash

(“.sh”) files with the syntax relevant to your cluster.

4) **Remember to check the “.err” and the “.out” log files** produced during job run on the cluster. Log files are generated in the same folder as the result files produced by the pipeline in each step or substep. They record warnings and error messages raised by the script or by the cluster during job run.

5) Before running the pipeline, remember to put the folder containing pipeline scripts and associated parameter files into the relevant sample directory. If multiple samples are analyzed, “scripts” folder should be created in each sample’s directory.

Pipeline wrapper script

The wrapper script provides an interactive menu allowing the users to navigate through the pipeline and choose which step or substep to execute.

To run the wrapper script, navigate from the terminal to the directory containing it and use the command: *bash MEMDS_pipeline_wrapper.sh.* By default, the wrapper script is provided with the rest of the pipeline scripts, but it can be placed in any directory. If pipeline scripts are placed on a remote machine, helper script should be placed on the same machine, and not in a local directory.

At the beginning of the run, helper script prompts to provide paths to the folders containing pipeline scripts, e.g: ***My_computer/analysis/sample/scripts***.
Multiple paths, pointing to script folders of different samples can be provided, separated by semi-colon, e.g:
My_computer/analysis/sample1/scripts;My_computer/analysis/sample2/scripts.

After the paths are entered, the script offers on-screen menu with possible run options. If a step contains several substeps, additional menu would appear asking to choose relevant substep. If multiple paths are provided, step and substep choice prompts would appear for each path provided.

Running pipeline scripts directly

1) Navigate into the scripts folder from command line (***cd /path/to/sample/scripts***) to

use the pipeline. All pipeline commands should be run from inside the scripts folder!

2) **(Optional) Merging raw data:**

a) *bash concatenate_partfiles.sh*

b) **Local run:** *bash fastq_merging/samples_table.concat.sh* **or**

c) **Cluster run:** *srun bash fastq_merging/samples_table.concat.sh*

3) **Formatting parameter data for use by the pipeline:**

a) **Single-end data:** *bash setting_1-SE.sh* **or**

b) **Paired-end data:** *bash setting_1-PE.sh*

4) **Quality control and clearing of raw data + paired-end data merging**:

a) **Single-end data:** *bash filter-SE4.sh $i* (1-3) **or**

b) **Paired-end data:** *bash filter-PE4.sh $i* (1-4)

5) **Selecting reads with correct barcodes and separating between barcodes and genomic data**:

a) *bash trim7.sh 1*

6) **Sorting reads by their origin gene:**

a) *bash sort2.sh 1*

7) **Mapping reads to the reference sequences:**

a) *bash bwa9.sh $i* (1-2)

8) **Making alignment files viewable in IGV:**

a) *bash create_dummy_genome5.sh 1*

9) **Creating mutation table that lists sequencing quality alongside each mutation:**

a) *bash sam_to_mutation-list-3.sh 1*

10) **Creating a table of mutations found in the analyzed data:**

a) *bash sam_to_mutation-table_5.2.sh 1*

11) **Detecting consensus mutations that pass a set of user defined thresholds:**

a) *bash consensus_15.1.sh $i* (1-2)