00 Pre-class setup

Grigorios Georgolopoulos

2024-07-01

Table of contents

1 Setup working directory **1**
1.1 Login to AUTH cluster (Remote coding only) 1
1.1.1 Install VSCode or MobaXTerm (Windows Users Only) 2
1.1.2 Connect to AUTH cluster 2
1.2 Clone the git repo with the course material 2

2 Creating a conda environment **2**

3 Data download **3**

In this chapter you will learn how to setup your environment either on the remote server or locally.

Learning objectives

1. Creating a directory
2. Cloning a git repo with class material
3. Setting up a conda environment
4. Download data

1 Setup working directory

1.1 Login to AUTH cluster (Remote coding only)

Before setting up, there are some necessary steps specific for remote coding which are specific to Windows users. If you are a Linux os MacOS user skip to section

1.1.1 Install VSCode or MobaXTerm (Windows Users Only)

In order to use SSH (remote host access) to the AUTH computer cluster you will either need to have Windows Subprocesses for Linux (WSL) installed and enabled or use and IDE such as VSCode (preferable) or MobaXTerm

1.1.2 Connect to AUTH cluster

If you are not logged in an AUTH network (e.g. working from home), make sure you have eduVPN enabled. More info here

Then open a terminal window or your IDE and type the following:

```
ssh [username]@aristotle.it.auth.gr
```

1.2 Clone the git repo with the course material

```
git clone https://github.com/ggeorgol/ATACseq_course

cd ATACseq_course
```

2 Creating a conda environment

There is an established set of tools required for analyzing high throughput sequencing data, and ATAC-seq in particular. For this reason we will create a virtual environment using the ANACONDA/miniconda (`conda` for short) package manager.

Specifically, we are going to need the following tools:

- htslib See SAMtools
- SAMtools The holy grail of HTS data processing. Your trusty hammer. An all-in-one kit for manipulating alignment files (BAM)
- picard Next to SAMtools there is Picard. A set of Java command line tools for manipulating high-throughput sequencing (HTS) data and formats.
- deepTools A suite of tools for expliring HTS data. Great for QC and visualization.
- bedTools a swiss-army knife of tools for a wide-range of genomics analysis tasks and genome arithmetic
- bedops Similar to bedTools, BEDOPS is a fast, highly scalable and easily-parallelizable genome analysis toolkit

- subread A suite of software programs for processing next-gen sequencing read data with `featureCounts` being one of the most popular read counters.

The following snippet will take a few minutes to complete

```
module load gcc miniconda3
source $CONDA_PROFILE/conda.sh

conda config --add channels defaults
conda config --add channels bioconda
conda config --add channels conda-forge
conda config --set channel_priority strict

conda create -n atac python=3.10 htslib samtools picard deeptools bedtools bedops subread
```

To activate the environment type the following:

```
conda activate atac
```

Load the R module

```
module load gcc/9.4.0-eewq4j6 r/4.2.2-2oxptjk
```

3 Data download

In this course we are going to work with ATAC-seq data generated by the ENCODE project. We will work with naïve and activated T-cells from a female adult with the following accession numbers: ENCSR977LVI, and ENCSR558ZSN. We will use the alignment (BAM) files and the already generated peaks.

If you work on the AUTH cluster, the data should be stored in your personal scartch space `$SRCATCH`. Keep in mind that data in `$SCRATCH` will be stored for 30 days only before the scratch space is cleaned up.

If you work on the cluster, type:

```
DATADIR=${SCRATCH}/ATACseq_course/data
ln -s $DATADIR data # Make a data shortcut to your working directory
```

If your work locally, type:

```
DATADIR=data
```

Continue

```
mkdir -p data/{ENCSR977LVI,ENCSR558ZSN}

# Download ENCSR558ZSN dataset
# BAM files
wget -P data/ENCSR558ZSN https://www.encodeproject.org/files/ENCFF287DFF/@@download/ENCFF287
wget -P data/ENCSR558ZSN https://www.encodeproject.org/files/ENCFF218OSF/@@download/ENCFF218

# Peaks
wget -P data/ENCSR558ZSN https://www.encodeproject.org/files/ENCFF002MKC/@@download/ENCFF002
wget -P data/ENCSR558ZSN https://www.encodeproject.org/files/ENCFF235RAD/@@download/ENCFF235

# Download ENCSR977LVI dataset
# BAM files
wget -P data/ENCSR977LVI https://www.encodeproject.org/files/ENCFF984NGC/@@download/ENCFF984
wget -P data/ENCSR977LVI https://www.encodeproject.org/files/ENCFF978AJO/@@download/ENCFF978

# Peaks
wget -P data/ENCSR977LVI https://www.encodeproject.org/files/ENCFF851MGR/@@download/ENCFF851
wget -P data/ENCSR977LVI https://www.encodeproject.org/files/ENCFF284IBU/@@download/ENCFF284
```

Lastly, we will need to setup R for the downstream analyses.

In your terminal, type the following:

```
module load gcc r/4.4.0-ervxjzd
R
```

Then, within R type the following to install the necessary packages.

```
if (!require("data.table", quietly = TRUE)) {
  install.packages("data.table")
}

if (!require("ggplot2", quietly = TRUE)) {
  install.packages("ggplot2")
}

if (!require("dplyr", quietly = TRUE)) {
  install.packages("dplyr")
}

if (!require("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

if (!require("R.utils", quietly = TRUE)) {
  install.packages("R.utils")
}

if (!require("GenomicRanges", quietly = TRUE)) {
BiocManager::install("GenomicRanges")
}

if (!require("Matrix", quietly = TRUE)) {
  install.packages("Matrix")
}
```