ChIPmE : NGS pipelines made easy

Patrick Schorderet

Patrick.schorderet@molbio.mgh.harvard.edu

Jan 2015

1. Introduction

ChIPmE is an R script wrapped into an applescript executable (double-click). Its main goal is to accompany users during the analysis of next generation sequencing (NGS) data as part of the NEAT toolkit (NGS easy analysis toolkit). ChIPpip, in conjuncture with the NEAT package, provides an easy, rapid and reproducible exploratory data analysis (EDA) tool that allows users to assess their data in as few as a couple hours (based on a 200mio read Highseq run). The primary goal of ChIPmE is to provide metagene analysis, peak overlaps and countables. ChIPmE has been developed as a downstream analysis tool for NGS data that has been analyzed using the ChIPpip package. ChIPpip is a package that manages the upstream analysis of metagene studies, including mapping of reads, filtering and peak calling. ChIPpip is an open source pipeline. More information on ChIPpip can be found on GitHub.

ChIPmE has been developed by and for biologist and can be run with no programming experience. As such, ChIPmE is an executable applescript using the MacOS GUI. In brief, this allows users to double click on the ChIPmE icon and get their analysis done in no time. As an example, analyzing an entire 200mio read Highseq run over all transcriptional start site of the mouse genome can be done in less than 5 minutes.

As such, ChIPmE manages many of the repetitive, error-prone tasks required for NGS data analysis. It is versatile and easily configurable to meet each user's preferences. ChIPpip accompanies users from ChIPpip projects to .pdf files in two double clicks.

A central feature of ChIPmE is its ability to perform repetitive tasks on complex sample setups. ChIPmE can easily be implemented in any institution with limited to no programming experience. Another advantage is that ChIPmE can be run either on a computer cluster via the command line or on a locally computer, where no internet connection is required.

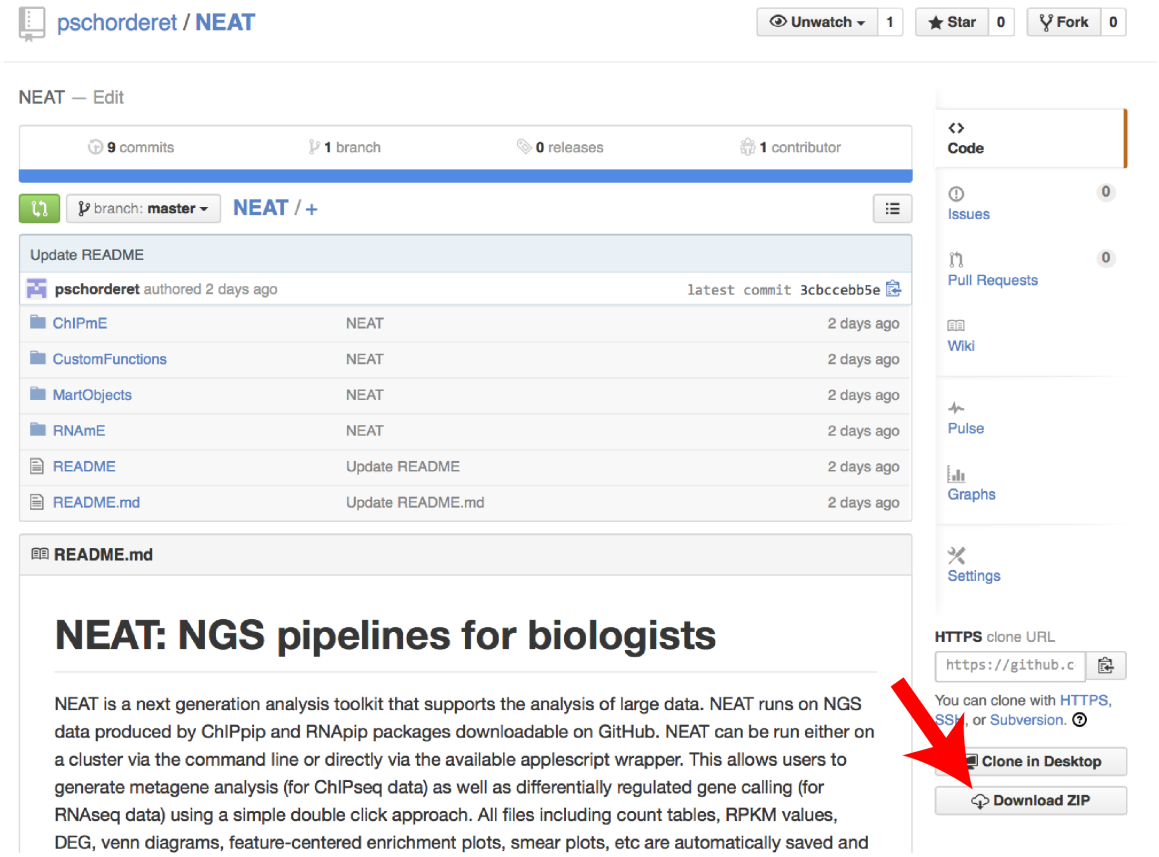
ChIPmE has been developed by and for wet-lab scientists as well as bioinformaticiens to ensure user-friendliness, management of complicated experimental setups and reproducibility in the big data era. To start using ChIPmE, please read the README file. In addition, before analyzing your own data, we suggest you follow this tutorial, which will help you better understand the logic of ChIPmE. You will be able to follow the analysis of a small test dataset (provided and analyzed as part of the ChIPpip package) using your own computer. We hereby expect that users have followed the ChIPpip tutorial prior to this one. Running the entire workflow, including ChIPpip and ChIPmE, will ensure all dependencies are correctly installed before submitting larger, memory-savvy analysis.

ChIPmE runs through the MacOS automator software. This software should be installed by default on most modern Apple computers. Please ensure this is installed on your computer in the *applications* directory. In addition, ChIPmE is a R script that requires a few widely recognized R packages. For details on these, please refer to the *Version information and required packages* section below.

2. ChIPmE

Install ChIPmE

First you will need to install ChIPmE. To this end, download NEAT (next generation sequencing easy analysis toolkit) from GitHub.



Save this directory on your local computer. For this tutorial, we will suppose the NEAT folder was saved on the user's desktop (*~/Desktop*).

Before running ChIPmE

Pleas make sure all R packages required for ChIPmE are installed on your computer (refer to the R manual). Refer to the Version information and required packages section below for more information.

Finally, make sure your *Terminal* software is closed before launching ChIPmE.

Download a ChIPpip project

To transfer a ChIPpip project, double click the *ChIPmE_1_Download* icon found in the NEAT directory *~/Desktop/NEAT/ChIPmE/*. Users will be asked to locate the NEAT directory and where they want to save their ChIPpip project. In this example, the NEAT directory is on our desktop and we will save our ChIPpip project on the desktop as well.

The next step is to provide the path to the ChIPpip directory on the remote server. In the ChIPpip tutorial, we had saved our project in the ChIPpip directory (*/HOME/ChIPpip/EXAMPLE*), so this is the path we will enter.

Finally, ChIPmE will prompt you to enter your SSH information. Once again, following the ChIPpip tutorial, we will enter: *username@server-address.edu*.

Starting the download will launch the *Terminal* to open and start running R. It will require users to enter their password several times (for each call to the remote servers). Please follow this process as failing to enter your password will break the connection after some time.

Downloading an entire project should not take more than a few minutes.

Run a ChIPmE analysis

Once the project is downloaded, users can run the proper ChIPmE analysis. To this end, double click the *ChIPmE_2_Analyse* icon. Users will be asked to locate the ChIPpip folder. In our case, the ChIPpip project was downloaded to the desktop.

Users will then have to locate the NEAT folder (also on desktop in this example). Finally, users will have to choose a mart object. These are .bed files of regions you are interested in aligning your data to. Example of mart objects are transcriptional start sites (TSS), transcripts, enhancers, etc. Some mart objects are provided as part of the NEAT package in the MartObject folder. For this example, we will choose to align our data over TSS. The provided mart object (*mm9_TSS_10kb.bed*) is comprises of 10kb around all TSS of the mouse genome. Please note here that care should be brought to match the mart objects with the reference genome initially used. In our case, our data was mapped to the mouse mm9 genome, hence the *mm9_TSS_10kb.bed*. Several parameters can be set before running the analysis including *binNumber*, *strand*, *runmeank*, *Venn* and *normInp*. Values are set as a reference, but we suggest users experiment to find the best values for their own need.

Running the analysis using the test data should take less than a minute.

3. Outputs

Logs

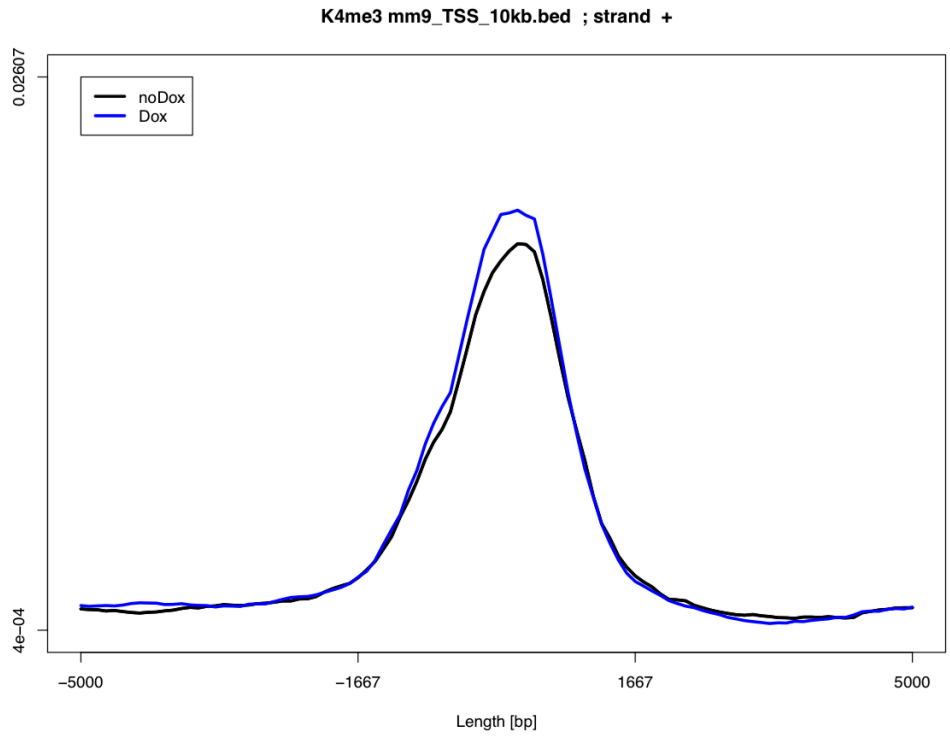
Each time ChIPmE is run, a log file is created and named using the date and time. This file is save in the *./EXAMPLE/logs/* directory. We strongly encourage users to look at these files, as any error that might have occurred will be saved there. Usually, if no error is prompted from the terminal, ChIPmE has terminated correctly. Also, please note that if there are unrecognized chromosome names such as random chromosomes, warning messages will appear. In the test data, there are 50 or more warnings. These can usually be disregarded.

Count tables

ChIPmE will generate count tables that should be self-explanatory. In brief, rows correspond to a mart object line and columns correspond to bins. Please note here that bins will be of same length for centered mart objects (for example TSSs) and will have names corresponding to base pairs, but will be of varied length for non-centered objects (for example transcripts), hence the columns will be named arbitrarily V1, V2, …, VN. Users should not worry about this to interpret graphs as it is accounted for by normalizing the values per bin by the bin length.

Metagene plots

ChIPmE generates pdf plots that are saved in */EXAMPLE/plots/*. Below is the plot generated for K4me3 over TSSs.



4. Advanced settings

Custom mart objects

Custom mart objects can easily be created with your favorites genes. The files are simple .bed files that can either be manually or automatically created using various online tools or can be directly downloaded from genome browsers such as USCS or Ensembl. Store custom mart objects in the *MartObject* folder.

Bam files and GRanges

Once Granges objects have been created, bam files are no longer required locally. Users are thus free to delete these files as they are often one order of magnitude larger than GRanges objects (respectively several Gb vs tens of Mb). We do suggest that users backup their bam files on the remote server.

5. Version information and required packages

```
R version 3.1.2 (2014-10-31)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
 [1] grid      parallel  stats4    stats     graphics  grDevices utils
 [8] datasets  methods   base

other attached packages:
 [1] VennDiagram_1.6.9       caTools_1.17.1          GenomicAlignments_1.2.1
 [4] Rsamtools_1.18.2        Biostrings_2.34.1       XVector_0.6.0
 [7] GenomicRanges_1.18.3    GenomeInfoDb_1.2.3      IRanges_2.0.1
[10] S4Vectors_0.4.0         BiocGenerics_0.12.1

loaded via a namespace (and not attached):
 [1] base64enc_0.1-2    BatchJobs_1.5      BBmisc_1.8         BiocParallel_1.0.0
 [5] bitops_1.0-6       brew_1.0-6         checkmate_1.5.0    codetools_0.2-9
 [9] DBI_0.3.1          digest_0.6.4       fail_1.2           foreach_1.4.2
[13] iterators_1.0.7    RSQLite_1.0.0      sendmailR_1.2-1    stringr_0.6.2
[17] tools_3.1.2        zlibbioc_1.12.0
```

6. Funding

This pipeline was developed with funding from the Swiss National Science Foundation.