RNAmE : NGS pipelines made easy

Patrick Schorderet

Patrick.schorderet@molbio.mgh.harvard.edu

Jan 2015

1. Introduction

RNAmE is a package of two applications. 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). RNApip, 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 RNAmE is to provide smear plots, differentially regulated genes, count tables and RPKM tables. RNAmE has been developed as a downstream analysis tool for NGS data that has been analyzed using the RNApip package. RNApip is a package that manages the upstream analysis of RNAseq studies, including mapping and filtering of reads. RNApip is an open source pipeline. More information on RNApip can be found on GitHub.

RNAmE has been developed by and for biologist as well as more advanced bioinformaticians to ensure user-friendliness, management of complicated experimental setups and reproducibility in the big data era. RNAmE can be run with no programming experience. As such, RNAmE is an executable app (apple script) using the MacOS GUI. In brief, this allows users to run the app and get their analysis done in no time. As an example, analyzing an entire 200mio read Highseq run over all known genes of the mouse genome can be done in less than 5 minutes.

RNAmE 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. RNApip accompanies users from RNApip projects to pdf files/excel spreadsheets in two double clicks. Although RNAmE has been developed to run from the app, advanced users can run it from command lines.

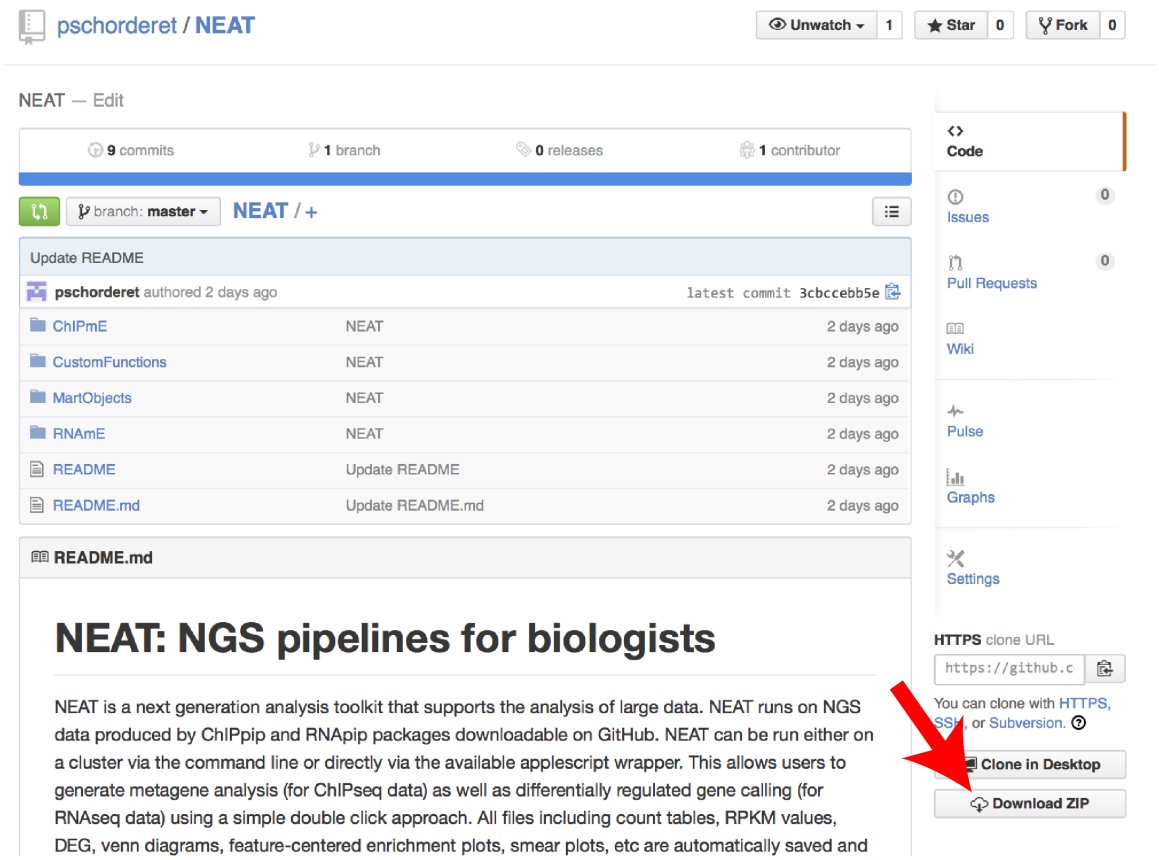
Before analyzing your own data, we suggest you follow this tutorial, which will help you better understand the logic of RNAmE. You will be able to follow the analysis of a small test dataset (provided and analyzed as part of the RNApip package) using your own computer. We hereby expect that users have followed the RNApip tutorial prior to this one. Running the entire workflow, including RNApip and RNAmE, will ensure all dependencies are correctly installed before submitting larger, memory-savvy analysis.

RNAmE 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, RNAmE is an 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. RNAmE

Install RNAmE

First you will need to install RNAmE. To this end, download NEAT 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 RNAmE

Pleas make sure all R packages required for RNAmE 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 RNAmE.

Step 1: Download an RNApip project

To transfer an RNApip project from a remote server to a local computer, double click the RNAmE *Transfer* app.

This app is can be found in the NEAT directory *~/Desktop/NEAT/RNAmE/*. Users will be asked to locate the NEAT directory and where they want to save their RNApip project. In this example, the NEAT directory is on our desktop and we will save our RNApip project on the desktop as well.

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

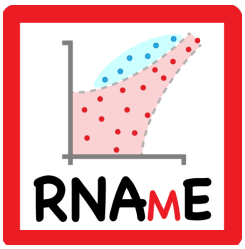
Finally, RNAmE will prompt you to enter your SSH information. Once again, following the RNApip tutorial, we will enter: *username@server-address.edu*.

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 server). 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.

Step 2: Run an RNAmE analysis

Once the project is downloaded, users can run the proper RNAmE analysis. To this end, double click the RNAmE *Analyse* app.



Users will be asked to locate the RNApip folder (the one you just downloaded). In our case, the RNApip project was downloaded to the desktop.

Users will then have to locate the NEAT folder (also on desktop in this example). RNAmE will use the databases specified in the Targets.txt file. In our example, we are using the TxDb.Mmusculus.UCSC.mm9.knownGene database. This comprises all known transcripts from UCSC and is a good starting point for initial EDA. Please note here that care should be brought to match the database objects with the reference genome initially used. In our case, our data was mapped to the mouse mm9 genome, hence the TxDb.Mmusculus.UCSC.mm9.knownGene. Several parameters can be set before running the analysis including *topGenes* and *toHighlight*, which correspond to the number of top DEG users want ot highlight in the tables and on the graphs. 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 initially 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.

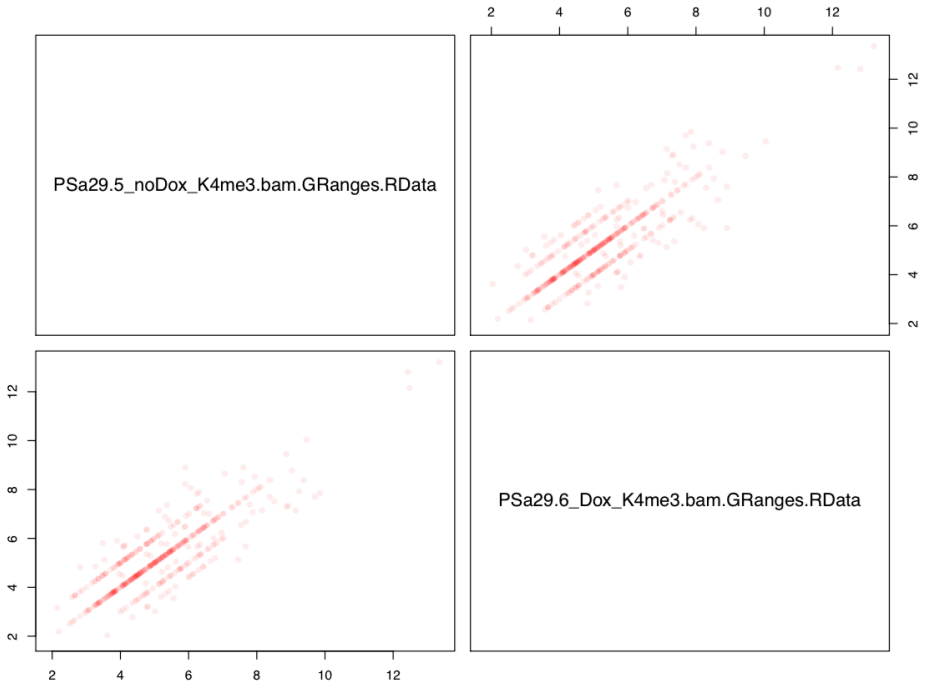
Count tables and RPKM tables

RNAmE will generate count tables that should be self-explanatory. In brief, rows correspond to genes and columns correspond to samples.

Differentially expressed gene plots

RNAmE generates pdf plots that are saved in */EXAMPLE/plots/*. General smear plots are created as a quick EDA between samples and replicates. In additional to smear plots, RNAmE generates all possible combinations of QQplots, with highlighted up and downregulated genes. More information on these genes can be found in the corresponding count tables. Finally, RNAmE generates gene lists meant to be fed to Gorilla, a well-established GO term EDA software. Venn diagrams are produced if selected. Users should note that computational time increases exponentially with the number of datasets as all combinations of Venn plots are created.

The smear plots below are an example of the results obtained using the test data. Obviously, the limited number of reads in the test data makes this plot relatively uninformative, but analyzing your own data will give much better resolution.



4. Advanced settings

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 two orders 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)

R packages (with dependencies) required to run ChIPmE:

- Rsamtools
- GenomicRanges
- GenomicAlignments
- caTools
- VennDiagram

sessionInfo()

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.