Mouse gastrulation: scNMT-seq

Data prepared by Al Abadi, Kim-Anh Lê Cao

October 16, 2019

Contents

1 Background

Background and easy data link is provided in the Readme_data file.

2 Data

2.1 Load libraries

```
# packages to load
library(mixOmics)
library(kableExtra)
library(MultiAssayExperiment)

# to increase memory size if needed
file.create(".Renviron")
cat("R_MAX_VSIZE=100Gb", file = ".Renviron")
```

2.2 Load the data

The data are stored as a `MultiAssayExperiment` object (see cheatsheet provided in the shared folder `Guidelines`, and this link) to handle and sample match multi-omics data more efficiently.

All annotations used M. *musculus* GRCm38 mouse genome.

```
# MAE experiment:
download.file(url = "https://cloudstor.aarnet.edu.au/plus/s/kzvLWezvcL5HFXJ/download",
              destfile = 'rdsfile.rds')

gastru.mae <- readRDS(file ='rdsfile.rds')
gastru.mae  # list of objects available
```

```
## A MultiAssayExperiment object of 13 listed
##  experiments with user-defined names and respective classes.
##  Containing an ExperimentList class object of length 13:
##  [1] rna: matrix with 18345 rows and 826 columns
##  [2] met_genebody: matrix with 15837 rows and 826 columns
##  [3] met_promoter: matrix with 12092 rows and 826 columns
##  [4] met_cgi: matrix with 5536 rows and 826 columns
##  [5] met_p300: matrix with 101 rows and 826 columns
##  [6] met_CTCF: matrix with 175 rows and 826 columns
##  [7] met_DHS: matrix with 66 rows and 826 columns
##  [8] acc_genebody: matrix with 17139 rows and 826 columns
##  [9] acc_promoter: matrix with 16518 rows and 826 columns
##  [10] acc_cgi: matrix with 4459 rows and 826 columns
##  [11] acc_p300: matrix with 138 rows and 826 columns
##  [12] acc_CTCF: matrix with 898 rows and 826 columns
##  [13] acc_DHS: matrix with 290 rows and 826 columns
## Features:
##  experiments() - obtain the ExperimentList instance
##  colData() - the primary/phenotype DataFrame
##  sampleMap() - the sample availability DataFrame
##  `$`, `[`, `[[` - extract colData columns, subset, or experiment
##  *Format() - convert into a long or wide DataFrame
##  assays() - convert ExperimentList to a SimpleList of matrices
```

```
# the meta data
#colData(gastru.mae)

# meta data for 'rna' data set
# note: a few NA's
knitr::kable(summary(as.factor(colData(gastru.mae[,,'rna'])$lineage10x_2),
                     caption = 'Lineage information per cell, note that there are a few NA values'),
             format = 'markdown')
```

	x
Ectoderm	43
Endoderm	81
Epiblast	334
ExE_ectoderm	8
Mesoderm	169
Primitive_endoderm	43
Primitive_Streak	76

	x
Visceral_endoderm	69
NA's	3

```
knitr::kable(summary(as.factor(colData(gastru.mae[,,'rna'])$stage),
              caption = 'Gastrulation stage information per cell'),
             format = 'markdown')
```

	x
E4.5	104
E5.5	108
E6.5	271
E7.5	343

```
knitr::kable(summary(as.factor(colData(gastru.mae[,,'rna'])$stage_lineage),
              caption = 'Gastrulation stage x lineage information per cell'),
             format = 'markdown')
```

	x
E4.5_Epiblast	60
E4.5_NA	1
E4.5_Primitive_endoderm	43
E5.5_Epiblast	84
E5.5_Visceral_endoderm	24
E6.5_Epiblast	146
E6.5_ExE_ectoderm	8
E6.5_Mesoderm	28
E6.5_NA	1
E6.5_Primitive_Streak	43
E6.5_Visceral_endoderm	45
E7.5_Ectoderm	43
E7.5_Endoderm	81
E7.5_Epiblast	44
E7.5_Mesoderm	141
E7.5_NA	1
E7.5_Primitive_Streak	33

3 Description of the different data sets

3.1 Transriptome data

The transcriptomics data were normalised using `scran`. Other methods can be used if you wish by retrieving the raw data from the `./data` folder from the easy access link). We advise gene filtering by the most variable genes (we provide the code in this .Rmd file).

3.2 Epigenome data

The methylome (`met`) and the accessibility data (`acc`) can be summarised over overlapping genic regions (such as gene bodies and gene promoters) and CG islands, as well as other genomic regions of epigenetic interest. In particular, and in addition to the datasets for gene-body and promoter methylome, we have chosen the following 3 regions which were highlighted in the original publication as epigenetically heterogeneous during development:

- P300: enhancer sites where p300 TFs bind
- CTCF binding sites: The primary role of CTCF transcription factors is thought to be in regulating the 3D structure of chromatin.
- DHS: In these specific regions of the genome, chromatin has lost its condensed structure, exposing the DNA and making it accessible.

Thus, we can investigate various subsets of DNA methylation (5 data sets) and DNA accessibility (5 data sets).

The features in epigenetic data (`acc` and `met`) are ranked according to their level of estimated biological variation in the assay matrices, after accounting for the uncertainty in observations due to the number of total CpGs observed in the region.

3.2.1 Pre-processing of epigenome data

From the data provided by Argelaguet *et al.* (in `./data` folder), we re-calculated the `met` and `acc` rates with a beta(1,1) prior so that the Standard Errors (SE) of estimates can be used as weights as explained below.

We required a minimum of 3 calls in the region followed by a stringent cell detection filtering of ~ 60% (500 cells out of 826 matching ones) for acc data and ~50% (400/826 cells) for met data which on average had less coverage. These selected thresholds can be changed. In the easy data we provide, features were sorted according to the lower bound of their estimated variance 95% confidence interval. We used the custom-made function `calc_site_stats` which can be found in the `.Rmd` version of this guide with its documentation and also in the `./src/utils/utils.R` file. This function also calculates a weight for each estimation based on the SE estimate (see function description) which can be used in various analyses.

The processed `.rds` data files can be found in the `./output` folder.

3.2.2 Caution about epigenome data

- As highlighted above, the data provided are ordered based on increasing biological variation (the top features are of most interest)
- All epigenome data include a **large amount** of missing values (a proportion of 35-40%!). You may choose to either estimate missing values, or choose / develop a method that can handle missing values.
- You will need to compromise between estimate uncertainty and the amount of data missing: if you put more stringent requirements on minimum number of calls at a given locus, more data will be discarded, resulting in matrices with more `NA`s.
- We do not recommend you run PCA on the epigenome data as it does not account for the uncertainties we measure on the methylation and accessibility rates. Try to account for it using the given SE in `.output//met_dt_list.rds` and `.output/met_dt_list.rds`, as detailed in the ReadMe file. To visualise population structure, an MDS using weighted Euclidean distances might be better suited.
- Methylome and chromatin accessibility data are far less predictive of cellular behaviour than transcriptome. Hence, these data may not be able to highlight phenotype heterogeneity, or identify outliers.

- For markers, is the discrimination power / effect size correlated with site CpG coverage (proportion of CpGs covered)?
- Differentially Methylated Regions are biased towards CpG rich regions.

Possible Quick fixes for some issues highlighted above: * Filter more stringently based on cell coverage so matrices contain less NAs, but also less features (interpretation should be given with care). * Replace missing values by the mean value across all cells (or use other imputation approach).

4 Prepare the data for analysis

4.1 RNA data filtering

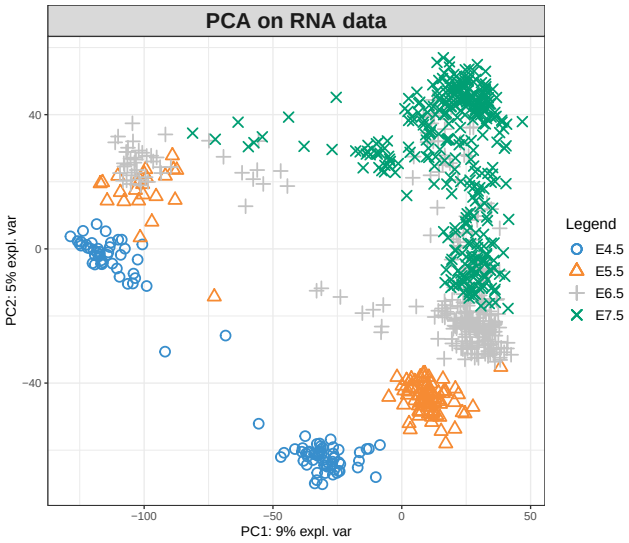
We use the following function to filter the top most variable genes from the RNA-seq data (open the .Rmd file to see the code of the function `get_hvgs_rna`)

```
# first, extract the data and filter the most variable genes
X <- get_hvgs_rna(log_counts = assay(gastru.mae, "rna"), ## a matrix of normalised counts
                    n_genes = 5000, ## No. of genes
                    use_bio=TRUE, ## choose based on biological variance or total variance?
                    do_plot = FALSE ## plot mean-variance dependency pre and post decomposition?
                   )
#dim(X)

# transpose if needed
X <- t(X)
dim(X)
```

```
## [1]  826 5000
```

Example of extracting one dataset and PCA:

```
metadata.X <- colData(gastru.mae[,,'rna'])
# assign the NA lineage to Unknown for sample plots
metadata.X$lineage10x_2[is.na(metadata.X$lineage10x_2)] = 'Unknown'
#summary(as.factor(metadata.X$lineage10x_2))

library(mixOmics)
pca.X = pca(X, ncomp = 5)
#plot(pca.X)
plotIndiv(pca.X, group = metadata.X$stage, legend = TRUE, ind.names =FALSE, title = 'PCA on RNA data')
```



4.2 DNA chromatin accessibility

Here is an example of code to extract a data set of interest and the matching meta data:

```
name_assay = 'acc_CTCF' # can be 'acc_p300', 'acc_CTCF', 'acc_cgi'
X <- assay(gastru.mae[,,name_assay])
# extract relevant meta data
metadata.X <- colData(gastru.mae[,,name_assay])
# assign the NA lineage to Unknown
metadata.X$lineage10x_2[is.na(metadata.X$lineage10x_2)] = 'Unknown'

#summary(as.factor(metadata.X$lineage10x_2))

# percentage of missing values
sum(is.na(X))/(nrow(X)*ncol(X))
```

```
## [1] 0.3651928
```

4.3 Several data sets

Here is an example to extract the list of all assays from `gastru.mae`. Note that you may not need to extract **all** data sets!

```
X <- assays(gastru.mae)
#lapply(X, dim)
metadata.X <- colData(gastru.mae)
#metadata.X$stage
```

5 Types of analysis challenges to address

- Identification of multi-omics signatures that characterise lineage, stage or both.
- Handling missing values
- Do epigenetic changes in some genomic contexts affect cell fate decision than others and how?

6 If you wish to help out

- We use a fixed threshold for filtering, while every site has potentially different number of CpGs. How can we use an adaptive filtering?
- How can we weight the data using the SE estimated (provided in `.output//met_dt_list.rds` and `.output/met_dt_list.rds`) so we can use the conventional component-based approaches directly?
- The protocol assumes every C to T conversion is the result of deamination of an unmethylated C, which ignores the C to T mutations in the genome. This introduces errors in methylation estimations. What is the most effective way to account for this in the absence of a wildtype genome?
- Any way to account for bias towards CpG rich regions in the methylation data?

7 Acknowledgements

Ricard Argelaguet (Stegle lab) for providing the data and Al Abadi (Lê Cao lab) for re-processing some part of the data. Technical questions can be directed to Al and Ricard on Slack.