SCALE vignette

Yuchao Jiang
yuchaoj@upenn.edu

September 13, 2016

This is a demo for using the SCALE package in R. SCALE is a statistical framework for single cell allelic expression analysis. SCALE estimates kinetic parameters that characterize the transcriptional bursting process at the allelic level, while accounting for technical bias and other complicating factors such as cell size. SCALE detects genes with significantly different bursting kinetics between the two alleles, as well as genes where the two alleles exhibit dependence in their bursting processes.

SCALE's **webpage** is here. A **demo code** can be found here. Online **Q&A forum** for SCALE is available here. If you've any questions regarding the software, you can also email us at SCALE_scRNAseq@googlegroups.com.

1. Installation

R package SCALE is availble from GitHub ((https://github.com/yuchaojiang/SCALE):

```
> install.packages("devtools")
> library(devtools)
> install_github("yuchaojiang/SCALE/package")
```

2. SCALE workflow

2.1 Data input

The input to SCALE includes allele-specific read counts at heterozygous loci from single-cell RNA sequencing. The cells should be of the same cell types from the same tissue (i.e., they are homogeneous). Cell-wise quality control procedures based on sequencing depths, mean and standard deviation of allelic ratios are recommended. To control for technical variability, SCALE uses spike-ins. The spike-in input should be a matrix, where the rows correspond to spike-ins, the first column stores the true number of molecules, the second column stores the lengths of the spike-in molecules, and the third column and on store the observed read counts in each cell.

Below is a single-cell RNA sequencing dataset of 122 mouse blastocyst cells from Deng et al. (Science 2014), followed by step-by-step analysis breakdowns.

```
> library(SCALE)
> data(mouse.blastocyst)
> alleleA = mouse.blastocyst$alleleA # Read counts for A allele
> alleleB = mouse.blastocyst$alleleB # Read counts for B allele
> spikein_input = mouse.blastocyst$spikein_input # Spike-in input
> genename = rownames(alleleA)
```

```
> sampname = colnames(alleleA)
> head(colnames(alleleA))

[1] "GSM1112611" "GSM1112612" "GSM1112613" "GSM1112614" "GSM1112615"
[6] "GSM1112616"

> head(rownames(alleleA))

[1] "Hvcn1"  "Gbp7"   "Arrdc1" "Ercc5"  "Mrpl15" "Dclk1"

> rownames(spikein_input)

[1] "RNA_SPIKE_1"  "RNA_SPIKE_2"  "RNA_SPIKE_3"  "RNA_SPIKE_4"  "RNA_SPIKE_5A"
[6] "RNA_SPIKE_6"  "RNA_SPIKE_7"  "RNA_SPIKE_8"

> head(colnames(spikein_input))

[1] "spikein_mol"    "spikein_length" "GSM1112664"     "GSM1112665"
[5] "GSM1112666"     "GSM1112667"
```

2.2 Gene classification

SCALE adopts a Bayes framework that categorizes each gene into being silent, monoallelically expressed, and biallelically expressed (including biallelically bursty). Proportions of cells expressing A and B alleles and gene categories are returned. Results from the first 10 genes are shown below. For genome-wide results, parallel computing on HPC is recommenede.

```
> gene.class.obj = gene_classify(alleleA=alleleA[1:10,], alleleB=alleleB[1:10,])

Gene 1 : Hvcn1 , Biallelic.bursty         A prop 0.231 B prop 0.264
Gene 2 : Gbp7 , Silent         A prop 0 B prop 0
Gene 3 : Arrdc1 , Biallelic.bursty         A prop 0.23 B prop 0.197
Gene 4 : Ercc5 , Biallelic.bursty         A prop 0.358 B prop 0.183
Gene 5 : Mrpl15 , Biallelic.bursty         A prop 0.875 B prop 0.925
Gene 6 : Dclk1 , Silent         A prop 0 B prop 0
Gene 7 : Tssc4 , Biallelic.bursty         A prop 0.254 B prop 0.213
Gene 8 : Gm101 , Silent         A prop 0 B prop 0
Gene 9 : Pum2 , Biallelic.bursty         A prop 0.15 B prop 0.142
Gene 10 : Erv3 , Silent         A prop 0 B prop 0

> A.prop = gene.class.obj$A.prop # Proportion of cells expressing A allele
> B.prop = gene.class.obj$B.prop # Proportion of cells expressing B allele
> gene.category = gene.class.obj$gene.category # Gene category
> results.list = gene.class.obj$results.list # Posterior assignments of cells
```

2.3 Technical variability

A hierarchical model based on TASC (Toolkit for Analysis of Single Cell data) is fit to the spike-in data. Parameters $\{\alpha, \beta, \kappa, \tau\}$ associated with dropouts, amplification and sequencing bias are returned. A pdf plot is generated by default.

```
> abkt = tech_bias(spikein_input = spikein_input, alleleA = alleleA,
+                  alleleB = alleleB, pdf = TRUE)
```

2.3 Allele-specific bursting kinetics

The two alleles of a gene have two Poisson-Beta distributions with respective parameters. These two Poisson-Beta distributions share the same cell-size factor. Cell-size factor can be estimated by the expression level of *GAPDH* or by the ratio of total number of endogenous RNA reads over the total number of spike-in reads. A Poisson hierarchical model is used to account for technical variability that is introduced by sequencing and library prep. Histogram repiling method is used to adjust for technical variability (bandwidth is optimized based on correlations of the inferred kinetic parameters between the two alleles). Moment estimator is used to estimate bursting kinetics. A plot (pdf format) is generated by default as is shown in Figure 1.

```
> cellsize = rep(1, ncol(alleleA))  # cell size input
> allelic.kinetics.obj = allelic_kinetics(alleleA = alleleA[1:1000,],
+                                         alleleB = alleleB[1:1000,],
+                                         abkt = abkt,
+                                         gene.category = gene.category[1:1000],
+                                         cellsize = cellsize, pdf = TRUE)

Bandwidth  1 :        % non-neg estimates 0.872         corr. freq 0.898         corr. size 0.78
Bandwidth  2 :        % non-neg estimates 0.883         corr. freq 0.897         corr. size 0.791
Bandwidth  3 :        % non-neg estimates 0.88          corr. freq 0.898         corr. size 0.79
Bandwidth  4 :        % non-neg estimates 0.88          corr. freq 0.894         corr. size 0.774
Bandwidth  5 :        % non-neg estimates 0.88          corr. freq 0.892         corr. size 0.766
Bandwidth  6 :        % non-neg estimates 0.883         corr. freq 0.898         corr. size 0.774
Bandwidth  7 :        % non-neg estimates 0.875         corr. freq 0.877         corr. size 0.723
Bandwidth  8 :        % non-neg estimates 0.872         corr. freq 0.884         corr. size 0.725
Bandwidth  9 :        % non-neg estimates 0.875         corr. freq 0.884         corr. size 0.722
Bandwidth  10 :        % non-neg estimates 0.875         corr. freq 0.868         corr. size 0.664
```

```
> bandwidth = allelic.kinetics.obj$bandwidth
> konA = allelic.kinetics.obj$konA; konB = allelic.kinetics.obj$konB
> koffA = allelic.kinetics.obj$koffA; koffB = allelic.kinetics.obj$koffB
> sA = allelic.kinetics.obj$sA; sB = allelic.kinetics.obj$sB
> sizeA = sA/koffA; sizeB = sB/koffB
```

2.4 Hypothesis testing

Nonparametric hypothesis test and chi-square test are carried out to test whether the two alleles of a gene share the same bursting kinetics and whether they burst independently. For test of same burst size and burst frequency between the two alleles, there are two 'modes': the *raw* mode bootstrap-samples from the raw observed allelic read counts; the *corrected* mode bootstrap-samples from the adjusted allelic read counts. Both modes give very similar results while the latter runs faster.

```
> # Nonparametric test on whether the two alleles share the same burst frequency and burst size.
> diff.allelic.obj = diff_allelic_bursting(alleleA = alleleA,
+                                          alleleB = alleleB,
+                                          cellsize = cellsize,
+                                          gene.category = gene.category,
+                                          abkt = abkt,
+                                          allelic.kinetics.obj = allelic.kinetics.obj,
+                                          mode = 'corrected')
> pval.kon = diff.allelic.obj$pval.kon; pval.size = diff.allelic.obj$pval.size
```

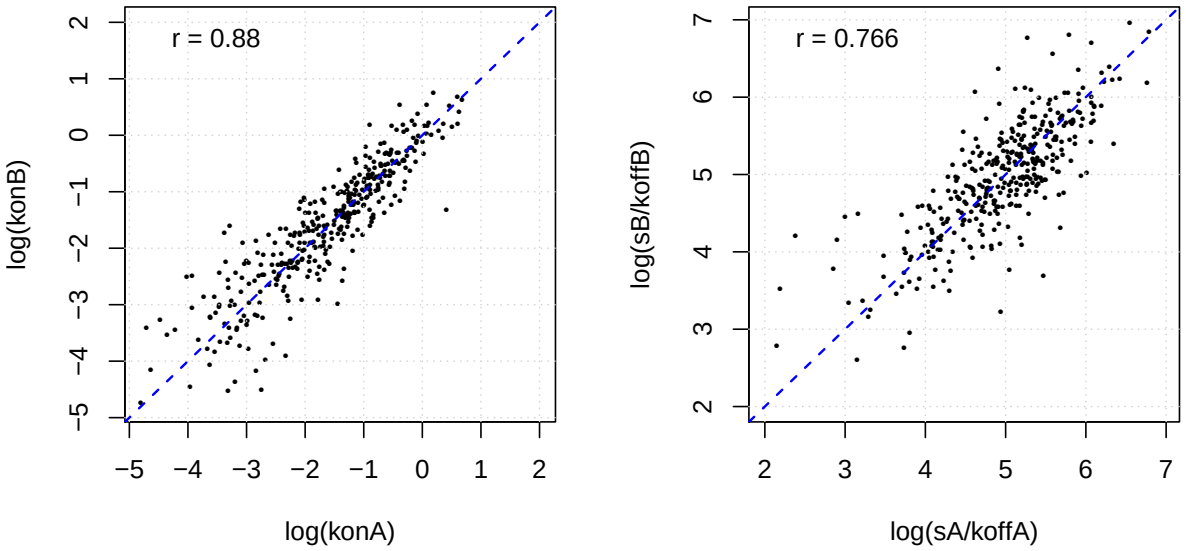

Figure 1: Allelic bursting kinetics (burst frequency and bursty size). Only first 1000 genes are computed.

```
> # Chi-square test on whether the two alleles fire independently.
> non.ind.obj = non_ind_bursting(alleleA = alleleA, alleleB = alleleB,
+                                gene.category = gene.category,
+                                results.list = results.list)
> pval.ind = non.ind.obj$pval.ind; non.ind.type = non.ind.obj$non.ind.type
```

2.5 Plot and output

For each gene, a plot (pdf format) can be generated with inferred parameters as well as summary statistics, as is shown in Figure 2.

```
> i=which(genename=='Btf3l4')
> allelic_plot(alleleA = alleleA, alleleB = alleleB,
+              gene.class.obj = gene.class.obj,
+              allelic.kinetics.obj = allelic.kinetics.obj,
+              diff.allelic.obj = diff.allelic.obj,
+              non.ind.obj = non.ind.obj, i= i)
```

The final output of `SCALE` is a tab delimited text file. The columns include: `genename` (gene name), `gene.category` (gene category), `konA` (burst frequency A), `konB` (burst frequency B), `pval.kon` (p-value of shared burst frequency), `sizeA` (burst size A), `sizeB` (burst size B), `pval.size` (p-value of shared burst size), `A_cell`, `B_cell`, `AB_cell`, `Off_cell` (number of cells with posterior assignment of A, B, AB, and Off), `A_prop` (proportion of cells expressing A allele), `B_prop` (proportion of cells expressing B allele), `p.ind` (p-value of burst independence), and `non.ind.type` (direction of non-independent bursting: 'C' is for coordinated bursting; 'R' for repulsed bursting).

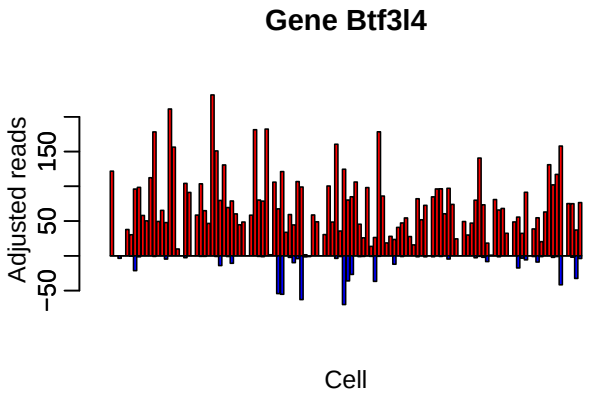

Figure 2: SCALE plot output for gene *Btf3l4*.

```
> SCALE.output=output_table(alleleA=alleleA, alleleB=alleleB,
+                           gene.class.obj = gene.class.obj,
+                           allelic.kinetics.obj = allelic.kinetics.obj,
+                           diff.allelic.obj = diff.allelic.obj,
+                           non.ind.obj = non.ind.obj)
> head(SCALE.output)

     genename gene.category      konA     konB     pval.kon  sizeA    sizeB
[1,] "Hvcn1"  "Biallelic.bursty" "0.08"   "0.0908" "0.6983"  "232.91" "263.37"
[2,] "Gbp7"   "Silent"           "-"      "-"      "-"       "-"      "-"
[3,] "Arrdc1" "Biallelic.bursty" "0.0825" "0.073"  "0.7166"  "199.31" "144.44"
[4,] "Ercc5"  "Biallelic.bursty" "0.0997" "0.0198" "0.05322" "322.53" "968.07"
[5,] "Mrpl15" "Biallelic.bursty" "1.2421" "1.3933" "0.72107" "150.31" "162.2"
[6,] "Dclk1"  "Silent"           "-"      "-"      "-"       "-"      "-"
     pval.size A_cell B_cell AB_cell Off_cell A.prop  B.prop  pval.ind
[1,] "0.7402"  "15"   "19"   "13"    "74"     "0.231" "0.264" "0.00624"
[2,] "-"       "0"    "0"    "0"     "122"    "0"     "0"     "-"
[3,] "0.50921" "18"   "14"   "10"    "80"     "0.23"  "0.197" "0.015"
[4,] "0.14033" "30"   "9"    "13"    "68"     "0.358" "0.183" "0.01181"
[5,] "0.86669" "5"    "11"   "100"   "4"      "0.875" "0.925" "0.00259"
[6,] "-"       "0"    "0"    "0"     "122"    "0"     "0"     "-"
     non.ind.type
[1,] "C"
[2,] "-"
[3,] "C"
[4,] "C"
[5,] "C"
[6,] "-"

> write.table(SCALE.output, file = 'SCALE.output.txt', col.names = T,
+             row.names = F, quote = F, sep = '\t')
```

3. Citation

Modeling allele-specific gene expression by single-cell RNA sequencing, Yuchao Jiang, Nancy R zhang, Mingyao Li, *submitted*, 2016.

4. Session information:

Output of sessionInfo on the system on which this document was compiled:

- R version 3.2.3 (2015-12-10), `x86_64-apple-darwin13.4.0`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: SCALE 1.0.0, rje 1.9
- Loaded via a namespace (and not attached): tools 3.2.3