MARBLES Vignette

Biqing Zhu

2021-06-30

Contents

Installing MARBLES from GitHub

```
library(MARBLES)
```

Main Function

The main function is

```
?run_MRF()
```

It requires three input values: (1) paraMRF: Starting value of the model parameter Φ; (2) expr: A pseudo-bulk expression list for a gene. Each item is a vector of the pseudo-bulk expression across all individuals for a cell type in a condition. The list names should start with ‘Cond1_’ or ‘Cond2_’, and followed by the name of the cell type; (3) c_c: A binary cell type relationship network matrix. 1 means connected, and 0 means not connected; and (4) x_init: Initial DE status. Should be a binary vector of length n_k. 1 means DE, and 0 means EE. The usage of this function will be illustrated in the following real data application.

Example

We obtained the mouse cortex snRNA-seq dataset from the R package *muscData* (Crowell et al. 2020). The data has already been pre-processed as well as annotated, and contains four control and four lipopolysaccharide (LPS)-treated mice.

```
library(muscData)
library(muscat)

mouse <- Crowell19_4vs4()
mouse <- prepSCE(mouse,
                kid = "cluster_id", # subpopulation assignments
                gid = "group_id",  # group IDs (ctrl/stim)
                sid = "sample_id",   # sample IDs (ctrl/stim.1234)
                drop = TRUE)  # drop all other colData columns
```

Since we wanted to focus on neurons and glial cells, we only selected astrocytes, microglia, oligodendrocyte progenitor cells (OPC), oligodendrocytes, excitatory neurons, and inhibitory neurons for the downstream analyses, and only kept the first 1,000 genes for demonstration purposes. And we applied the function `aggregateData` in the R package *muscat* (Crowell et al. 2020) to get cell-type-specific pseudobulk data.

```
pb <- aggregateData(mouse[1:1000, !mouse$cluster_id %in% c('Endothelial', "CPE cells")])
```

Then, we created a cell type relationship network based on domain knowledge to represent the similarity and cell lineage.

```
## Create connection matrix (OPC connected to neurons)
celltypes <- c('Astrocytes', 'Excit. Neuron', 'Inhib. Neuron',
               'Microglia', 'Oligodendrocytes', 'OPC')
c_c <- matrix(1, nrow = length(celltypes), ncol = length(celltypes))
rownames(c_c) <- colnames(c_c) <- celltypes
conn <- c(0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1)
c_c[lower.tri(c_c)] <- conn
c_c <- t(c_c)
c_c[lower.tri(c_c)] <- conn
c_c
#>                  Astrocytes Excit. Neuron Inhib. Neuron Microglia
#> Astrocytes                1             0             0         1
#> Excit. Neuron             0             1             1         0
#> Inhib. Neuron             0             1             1         0
#> Microglia                 1             0             0         1
#> Oligodendrocytes          0             0             0         0
#> OPC                       1             1             1         1
#>                  Oligodendrocytes OPC
#> Astrocytes                      0   1
#> Excit. Neuron                   0   1
#> Inhib. Neuron                   0   1
#> Microglia                       0   1
#> Oligodendrocytes                1   1
#> OPC                             1   1
```

edgeR (Robinson, McCarthy, and Smyth 2010) was first applied to the pseudo-bulk data to get the DE states using `pbDS`

```
res_edger <- pbDS(pb, method = "edgeR", verbose = FALSE)
tbl_edger <- res_edger$table[[1]]
names(tbl_edger)
#> [1] "Astrocytes"       "Microglia"        "Oligodendrocytes" "OPC"
#> [5] "Excit. Neuron"    "Inhib. Neuron"
# view results for 1st cluster
k1 <- tbl_edger[[1]]
head(format(k1[, -ncol(k1)], digits = 2))
#>                         gene cluster_id   logFC logCPM       F   p_val
#> 1   ENSMUSG00000051951.Xkr4 Astrocytes -1.0444   10.3 5.0e+00 4.0e-02
#> 2 ENSMUSG00000033845.Mrpl15 Astrocytes -0.5690    9.6 3.1e+00 9.9e-02
#> 3  ENSMUSG00000033813.Tcea1 Astrocytes  0.1379    9.6 2.3e-01 6.4e-01
#> 4  ENSMUSG00000002459.Rgs20 Astrocytes  0.6762   13.4 5.1e+00 3.9e-02
#> 5 ENSMUSG00000033793.Atp6v1h Astrocytes  0.3710   10.3 2.2e+00 1.6e-01
#> 6  ENSMUSG00000025907.Rb1cc1 Astrocytes -0.1732   10.8 3.8e-01 5.5e-01
#>   p_adj.loc p_adj.glb
#> 1   3.0e-01   1.0e+00
#> 2   4.7e-01   1.0e+00
#> 3   9.1e-01   1.0e+00
#> 4   3.0e-01   1.0e+00
#> 5   5.6e-01   1.0e+00
#> 6   8.6e-01   1.0e+00
```

Next, genes with FDR corrected p value >= 0.05, and abs(logFC) <=1 were filtered out.

```
results_fil <- lapply(tbl_edger, function(u) {
      u <- dplyr::filter(u, p_adj.loc < 0.05 & abs(logFC) > 1)
      dplyr::arrange(u, p_adj.loc)
})
results_gene <- lapply(results_fil, function(u) u$gene)
pseudo_bulk_lst <- pb@assays@data@listData
```

In order to input the data into our model, it's structure was rearranged such that each element of the list represents a gene, and within each element, each item is a vector of the pseudo-bulk expression across all individuals for a cell type in a condition.

```
pseudo_bulk_lst_new <- list()
for(gene in rownames(pseudo_bulk_lst[[1]])) {
    pseudo_bulk_lst_new[[gene]] <- list()
    for(celltype in celltypes) {
        pseudo_bulk_lst_new[[gene]][[paste0('Cond1_', celltype)]] <-
          unlist(pseudo_bulk_lst[[celltype]][gene, 1:4])
        pseudo_bulk_lst_new[[gene]][[paste0('Cond2_', celltype)]] <-
          unlist(pseudo_bulk_lst[[celltype]][gene, 5:8])
    }
}
str(pseudo_bulk_lst_new[[1]])
#> List of 12
#>  $ Cond1_Astrocytes      : Named num [1:4] 13 15 28 61
#>   ..- attr(*, "names")= chr [1:4] "LC016" "LC019" "LC022" "LC025"
#>  $ Cond2_Astrocytes      : Named num [1:4] 12 8 18 30
#>   ..- attr(*, "names")= chr [1:4] "LC017" "LC020" "LC023" "LC026"
#>  $ Cond1_Excit. Neuron   : Named num [1:4] 5295 3707 4645 4232
#>   ..- attr(*, "names")= chr [1:4] "LC016" "LC019" "LC022" "LC025"
#>  $ Cond2_Excit. Neuron   : Named num [1:4] 3092 2426 3468 5693
#>   ..- attr(*, "names")= chr [1:4] "LC017" "LC020" "LC023" "LC026"
#>  $ Cond1_Inhib. Neuron   : Named num [1:4] 1412 1020 1846 1893
#>   ..- attr(*, "names")= chr [1:4] "LC016" "LC019" "LC022" "LC025"
#>  $ Cond2_Inhib. Neuron   : Named num [1:4] 903 850 1258 2055
#>   ..- attr(*, "names")= chr [1:4] "LC017" "LC020" "LC023" "LC026"
#>  $ Cond1_Microglia       : Named num [1:4] 14 9 18 70
#>   ..- attr(*, "names")= chr [1:4] "LC016" "LC019" "LC022" "LC025"
#>  $ Cond2_Microglia       : Named num [1:4] 36 21 21 37
#>   ..- attr(*, "names")= chr [1:4] "LC017" "LC020" "LC023" "LC026"
#>  $ Cond1_Oligodendrocytes: Named num [1:4] 58 53 125 134
#>   ..- attr(*, "names")= chr [1:4] "LC016" "LC019" "LC022" "LC025"
#>  $ Cond2_Oligodendrocytes: Named num [1:4] 40 43 43 62
#>   ..- attr(*, "names")= chr [1:4] "LC017" "LC020" "LC023" "LC026"
#>  $ Cond1_OPC             : Named num [1:4] 35 92 55 42
#>   ..- attr(*, "names")= chr [1:4] "LC016" "LC019" "LC022" "LC025"
#>  $ Cond2_OPC             : Named num [1:4] 20 35 50 91
#>   ..- attr(*, "names")= chr [1:4] "LC017" "LC020" "LC023" "LC026"
```

Finally, we ran our method with edgeR initialization.

```
x_mat_edger <- matrix(0, length(pseudo_bulk_lst_new), length(celltypes))
rownames(x_mat_edger) <- names(pseudo_bulk_lst_new)
colnames(x_mat_edger) <- celltypes
for (celltype in celltypes) {
   x_mat_edger[results_fil[[celltype]]$gene, celltype] <- 1
}
x_mat_edger <- lapply(seq_len(nrow(x_mat_edger)), function(i) x_mat_edger[i,])
results_edger <- mapply(run_MRF, expr = pseudo_bulk_lst_new, x_init = x_mat_edger,
                        MoreArgs = list(paraMRF = c(0, 2), c_c = c_c), SIMPLIFY = FALSE)
```

The Φ parameter for the 10th gene can be retrieved by

```
results_edger[[10]]$phi_mat[,ncol(results_edger[[10]]$phi_mat)]
#>    gamma     beta
#> 21.39938 43.02440
```

And the DE states is

```
results_edger[[10]]$x_mat[,ncol(results_edger[[10]]$x_mat)]
#>       Astrocytes    Excit. Neuron    Inhib. Neuron        Microglia
#>                0                1                1                0
#> Oligodendrocytes              OPC
#>                0                0
```

References

Crowell, Helena L, Charlotte Soneson, Pierre-Luc Germain, Daniela Calini, Ludovic Collin, Catarina Raposo, Dheeraj Malhotra, and Mark D Robinson. 2020. "Muscat Detects Subpopulation-Specific State Transitions from Multi-Sample Multi-Condition Single-Cell Transcriptomics Data." *Nature Communications* 11 (1): 1–12.

Robinson, Mark D, Davis J McCarthy, and Gordon K Smyth. 2010. "EdgeR: A Bioconductor Package for Differential Expression Analysis of Digital Gene Expression Data." *Bioinformatics* 26 (1): 139–40.