

04_scale-pb-expt

Sean Maden

2022-10-13

This vignette times the lpb function by sampling cells from sef, where sef represents data from a real (non-simulated) snRNAseq experiment.

time run for n samples

Define a helper function to time the run.

```
time_lpb_run <- function(nj, nk = 4, ng = 100,
                        datv.str = c("c(sample(10, ", "*",
                                     ", replace = T) + 1)")){
  # time_lpb_run
  #
  # example:
  # time_run(4)
  #
  #
  t1 <- Sys.time()
  datv = eval(parse(text = paste0(datv.str[1], nk, datv.str[2], nj, datv.str[3])))
  pb.expt <- get_pb_experiment(lz = lexpt, sef = sef, datv = datv,
                             scale.range = 100:10000, ctvarname = "celltype.treg",
                             plot.results = FALSE, method.str = "nnls", seed.num = 2)
  tdif <- Sys.time()-t1
  dfi <- data.frame(nj = nj, time = tdif)
  return(dfi)
}
```

```
time_lpb_run(4)
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: MatrixGenerics
```

```
## Loading required package: matrixStats
```

```
##
```

```
## Attaching package: 'MatrixGenerics'
```

```

## The following objects are masked from 'package:matrixStats':
##
##   colAlls, colAnyNAs, colAnys, colAvgPerRowSet, colCollapse,
##   colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##   colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##   colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##   colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##   colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##   colWeightedMeans, colWeightedMedians, colWeightedSds,
##   colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgPerColSet,
##   rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##   rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##   rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##   rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##   rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##   rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##   rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##   IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##   anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##   dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##   grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##   order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##   rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##   union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##   expand.grid, I, unname

## Loading required package: IRanges

```

```

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:Biobase':
##
##     combine

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union

```

```

## The following objects are masked from 'package:BiocGenerics':
##
##   combine, intersect, setdiff, union

## The following object is masked from 'package:matrixStats':
##
##   count

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

## identified 4 cell types...

## identified 100 features...

## getting pb data for 4 samples...

## Loading required package: reshape2

## z.final

## running nnls...

## Loading required package: nnls

##      nj           time
## 1  4 5.040065 secs

```

Define a series of samples nj, and get the times for a series of nj values.

```

njv <- seq(10)
tv <- do.call(rbind,
              lapply(njv,
                     function(nj){
                       time_lpb_run(nj)}))

```

```

## identified 4 cell types...

## identified 100 features...

## getting pb data for 1 samples...

## z.final

```

```
## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 2 samples...

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 3 samples...

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 4 samples...

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 5 samples...

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 6 samples...
```

```

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 7 samples...

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 8 samples...

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 9 samples...

## z.final

## running nnls...

## identified 4 cell types...

## identified 100 features...

## getting pb data for 10 samples...

## z.final

## running nnls...

```

```
tv
```

```

##      nj      time
## 1    1 0.305974 secs
## 2    2 0.698355 secs
## 3    3 0.663413 secs
## 4    4 0.823215 secs
## 5    5 1.013983 secs
## 6    6 1.171285 secs
## 7    7 1.387465 secs
## 8    8 1.563008 secs
## 9    9 1.775939 secs
## 10 10 1.951194 secs

```

do log scale test

```
njv <- c(1, 10, 100, 1000)
tv <- do.call(rbind,
              lapply(njv,
                     function(nj){
                       time_lpb_run(nj)}))
```

```
## identified 4 cell types...
```

```
## identified 100 features...
```

```
## getting pb data for 1 samples...
```

```
## z.final
```

```
## running nnls...
```

```
## identified 4 cell types...
```

```
## identified 100 features...
```

```
## getting pb data for 10 samples...
```

```
## z.final
```

```
## running nnls...
```

```
## identified 4 cell types...
```

```
## identified 100 features...
```

```
## getting pb data for 100 samples...
```

```
## z.final
```

```
## running nnls...
```

```
## identified 4 cell types...
```

```
## identified 100 features...
```

```
## getting pb data for 1000 samples...
```

```
## z.final
```

```
## running nnls...
```

tv

##		nj		time
## 1	1	0.2300589	secs	
## 2	10	2.1628721	secs	
## 3	100	20.0919838	secs	
## 4	1000	198.3909280	secs	