Using the Stroma4 package

Sadiq MI Saleh, Michael T Hallett

November 9, 2016

1 Introduction

This package assigns the stroma and Lehmann properties to a set of samples. While they were identified in TNBC stromal samples, these properties are applicable outside of this context.

2 Analysis of sample dataset

To apply the stromal properties, simply use pass an ‘ExpressionSet‘ to the function ”assign.properties”

First we will load the Stroma4 package.

```
> source("https://bioconductor.org/biocLite.R")
> #biocLite("STROMA4")
> library("STROMA4")
```

Next we load a test dataset to serve as an example. We use the gene expression dataset published by Schmidt et al. [2008] provided in the breastCancerMAINZ package.

```
> library(breastCancerMAINZ)
> data(mainz, package='breastCancerMAINZ')
```

3 types of analyses can then be run:

2.1 Assignment of stromal properties alone

```
> just.stromal.properties <- assign.properties(ESet=mainz, geneID.column="Gene.symbol",
+                            genelists="Stroma4", n=10, mc.cores=1)

--Assigning properties to expression data--
--There are duplicated genes. Using most variable to collapse--
---- 206 out of 297 total genes matching for D.stroma.property ----
---- 104 out of 567 total genes matching for B.stroma.property ----
---- 315 out of 458 total genes matching for T.stroma.property ----
---- 20 out of 27 total genes matching for E.stroma.property ----
```

2.2 Assignment of Lehmann properties alone

```
> just.lehmann.properties <- assign.properties(ESet=mainz, geneID.column="Gene.symbol",
+                            genelists="TNBCType", n=10, mc.cores=1)

--Assigning properties to expression data--
--There are duplicated genes. Using most variable to collapse--
---- 1064 out of 1105 total genes matching for MSL.property ----
```

```
---- 671 out of 710 total genes matching for M.property ----
---- 1146 out of 1205 total genes matching for LAR.property ----
---- 670 out of 705 total genes matching for IM.property ----
---- 449 out of 473 total genes matching for BL1.property ----
---- 269 out of 291 total genes matching for BL2.property ----
```

2.3 Assignment of both stromal and Lehmann properties simultaneously

```
> all.properties <- assign.properties(ESet=mainz, geneID.column="Gene.symbol",
+                 genelists=c("Stroma4", "TNBCType"), n=10, mc.cores=1)
```

```
--Assigning properties to expression data--
--There are duplicated genes. Using most variable to collapse--
---- 206 out of 297 total genes matching for D.stroma.property ----
---- 1064 out of 1105 total genes matching for MSL.property ----
---- 671 out of 710 total genes matching for M.property ----
---- 104 out of 567 total genes matching for B.stroma.property ----
---- 1146 out of 1205 total genes matching for LAR.property ----
---- 670 out of 705 total genes matching for IM.property ----
---- 315 out of 458 total genes matching for T.stroma.property ----
---- 449 out of 473 total genes matching for BL1.property ----
---- 20 out of 27 total genes matching for E.stroma.property ----
---- 269 out of 291 total genes matching for BL2.property ----
```

2.4 Expected result from assinging properties

The function returns a list of assignments for each property. Assignments are from 1-3, with 1 being low, 2 being intermediate, and 3 being high assignment respectively.

```
> print(lapply(all.properties, table))
```

```
$D.stroma.property

 1  2  3
82 24 94

$MSL.property

 1  2  3
89 18 93

$M.property

  1   2   3
 69  31 100

$B.stroma.property

 1  2  3
60 68 72

$LAR.property

  1   2   3
 52  40 108

$IM.property

 1  2  3
96 30 74

$T.stroma.property

 1  2  3
87 34 79

$BL1.property

 1  2  3
73 60 67

$E.stroma.property

  1   2   3
 17 135  48

$BL2.property

 1  2  3
53 79 68
```

3 Notes:

3.1 Confidence of intermediate region

By increasing the number or random samples (n), the confidence in the intermediate region is increased at the expense of increased computing time

3.2 Using multiple cores

2. Using multiple cores (mc.cores) on machines with multiple cores available can reduce runtime. The function detectCores() can be used to determine how many cores you have available.