Gene Set Analysis

Guillermo Ayala

5/11/23

Table of contents

Abstract

Different procedures for gene set analysis are proposed in this vignette.

Packages

```
pacman::p_load(Biobase,SummarizedExperiment,tami,ggplot2)
```

Data

We load the data set from `tamidata`. We are going to analyze two different data sets. The first data set has been obtained using microarrays. The second is a RNA-seq data set.

gse21942

First an experiment with microarrays applied to samples of homo sapiens.

```
data(gse21942,package="tamidata")
```

The phenotipic variable classifying the samples is `FactorValue..DISEASE.STATE.`.

```
pData(gse21942)[,"FactorValue..DISEASE.STATE."]
```

```
 [1] multiple sclerosis multiple sclerosis multiple sclerosis multiple sclerosis
 [5] multiple sclerosis multiple sclerosis multiple sclerosis multiple sclerosis
 [9] multiple sclerosis multiple sclerosis multiple sclerosis multiple sclerosis
[13] multiple sclerosis multiple sclerosis healthy            healthy           
[17] healthy            healthy            healthy            healthy           
[21] healthy            healthy            healthy            healthy           
[25] healthy            healthy            healthy            healthy           
[29] healthy           
Levels: healthy multiple sclerosis
```

PRJNA297664

Second, an experiment with RNA-seq data of Saccharomyces cerevisiae.

```
data(PRJNA297664,package="tamidata")
```

The phenotipic variable is `treatment`.

```
colData(PRJNA297664)[,"treatment"]
```

```
[1] Wild           Wild           SEC66 deletion Wild           SEC66 deletion
[6] SEC66 deletion
Levels: Wild SEC66 deletion
```

Differential expression analysis for `gse21942`

t-tests for `gse21942`

We perform a differential expression analysis using t-tests for `gse21942`.

```
gse21942_rowt = dema(x=gse21942,y="FactorValue..DISEASE.STATE.",
                test = rowt,correction = "BH",
                fdr = .0001,foutput = "gse21942")
```

```
Warning: replacing previous import 'utils::findMatches' by
'S4Vectors::findMatches' when loading 'AnnotationDbi'
```

A `data.frame` with the results is obtained with

```
gse21942_rowt_df = tidy(gse21942_rowt)
```

The same information is saved in a html file using

```
glimpse(gse21942_rowt)
```

The same information can be studied using an html report with

```
browseURL(glimpse(gse21942_rowt))
```

Note that only the genes such that the adjusted p-value is lesser than the corresponding `fdr` are contained in the `data.frame` or the html report provided by `tidy` or `glimpse` applied over the object returned by `dema`. How many genes (features) are considered in our study?

```
nrow(gse21942)
```

```
Features
   54675
```

How many genes are returned by `dema`?

```
nrow(gse21942_rowt_df)
```

```
[1] 1044
```

Let us see a summary of the adjusted p-value for these **significant** genes.

```
summary(gse21942_rowt_df[,"adjp"])
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
3.637e-08 8.896e-06 2.598e-05 3.394e-05 5.609e-05 9.999e-05 
```

In particular, the maximum adjusted p-value is lesser than `fdr = .0001`.

Moderated t-tests for gse21942

The test used is now `rowtmod`. It is just the method proposed in tha package **Limma**. We perform a differential expression analysis using a moderated t-tests for `gse21942`.

```
gse21942_rowtmod = dema(x=gse21942,y="FactorValue..DISEASE.STATE.",
                test = rowtmod,correction = "BH",
                fdr = .0001,foutput = "gse21942")
```

A `data.frame` with the results is obtained with

```
gse21942_rowtmod_df = tidy(gse21942_rowtmod)
```

The same information can be studied using an html report with

```
browseURL(glimpse(gse21942_rowtmod))
```

Marginal differential expression analysis for `PRJNA297664`

`edgeR` method for `PRJNA297664`

First the edgeR method with a common dispersion is used.

```
PRJNA297664_erc=dema(x=PRJNA297664,y="treatment",test=edgercommon,
                       correction = "BH",fdr= 0.01,foutput = "output")
```

Second, the edgeR method with a tagwise dispersion is used.

```
PRJNA29766_ert = dema(x=PRJNA297664,y="treatment",
                      test = edgertagwise,correction = "BH",
                      fdr= 0.01,foutput = "output")
```

Over-representation analysis for `gse21942`

Introduction: exploring a gene set collection

We pretend to perform an over-representation analysis using an unilateral (or one-tail) Fisher test. The minimun gene set size should be censored to a given minimum value, for instance, 10 genes per set and the maximum size will be 100 genes per set.

```
Loading required namespace: GO.db
```

First we need a `list` of gene sets. Son los grupos Gene Ontology para humanos.

```
load(paste0(dirTamiData,"hsa_go.rda"))
```

We have load the gene set collection `hsa_go`. It is convenient to know what of data we have?

```
class(hsa_go)
```

```
[1] "list"
```

We can use generic functions for `list`. In particular, it can be know the number of gene sets with

```
length(hsa_go)
```

```
[1] 22934
```

We can access to each element in the `list` using the position

```
hsa_go[1]
```

```
$`GO:0000002`
 [1] "5428"   "6742"   "11232"  "55186"  "56652"  "84275"  "92667"  "201973"
 [9] "1763"   "7157"   "9093"   "10891"  "80119"  "83667"  "201163" "142"   
[17] "1890"   "2021"   "3980"   "4358"   "4976"   "6240"   "7156"   "10000" 
[25] "50484"  "64863"  "219736" "4205"   "9361"   "291"   
```

or the name

```
hsa_go["GO:0000002"]
```

```
$`GO:0000002`
 [1] "5428"   "6742"   "11232"  "55186"  "56652"  "84275"  "92667"  "201973"
 [9] "1763"   "7157"   "9093"   "10891"  "80119"  "83667"  "201163" "142"   
[17] "1890"   "2021"   "3980"   "4358"   "4976"   "6240"   "7156"   "10000" 
[25] "50484"  "64863"  "219736" "4205"   "9361"   "291"   
```

If we want the elements then

```
hsa_go[[1]]
```

```
 [1] "5428"   "6742"   "11232"  "55186"  "56652"  "84275"  "92667"  "201973"
 [9] "1763"   "7157"   "9093"   "10891"  "80119"  "83667"  "201163" "142"   
[17] "1890"   "2021"   "3980"   "4358"   "4976"   "6240"   "7156"   "10000" 
[25] "50484"  "64863"  "219736" "4205"   "9361"   "291"   
```

or

```
hsa_go[["GO:0000002"]]
```

```
 [1] "5428"   "6742"   "11232"  "55186"  "56652"  "84275"  "92667"  "201973"
 [9] "1763"   "7157"   "9093"   "10891"  "80119"  "83667"  "201163" "142"   
[17] "1890"   "2021"   "3980"   "4358"   "4976"   "6240"   "7156"   "10000" 
[25] "50484"  "64863"  "219736" "4205"   "9361"   "291"   
```

The first elements of the `list`.

```
head(hsa_go)
```

```
$`GO:0000002`
 [1] "5428"   "6742"   "11232"  "55186"  "56652"  "84275"  "92667"  "201973"
 [9] "1763"   "7157"   "9093"   "10891"  "80119"  "83667"  "201163" "142"   
[17] "1890"   "2021"   "3980"   "4358"   "4976"   "6240"   "7156"   "10000" 
[25] "50484"  "64863"  "219736" "4205"   "9361"   "291"   

$`GO:0000003`
   [1] "49"         "167"        "190"        "268"        "301"        "367"       
   [7] "638"        "699"        "701"        "993"        "994"        "995"       
  [13] "1060"       "1654"       "1761"       "2072"       "2177"       "2348"      
  [19] "2350"       "2352"       "2488"       "2492"       "2515"       "2528"      
  [25] "2622"       "3010"       "3024"       "3248"       "3267"       "3364"      
  [31] "3619"       "3973"       "4342"       "4361"       "4438"       "4439"      
  [37] "5021"       "5048"       "5139"       "5617"       "5620"       "5660"      
  [43] "5819"       "5888"       "5889"       "5892"       "6117"       "6406"      
  [49] "6407"       "6847"       "6865"       "6869"       "6870"       "6954"      
  [55] "7141"       "7142"       "7153"       "7155"       "7222"       "7225"      
  [61] "7272"       "7283"       "7432"       "7783"       "7784"       "8438"      
  [67] "8468"       "8521"       "8653"       "8747"       "8748"       "8749"      
  [73] "9232"       "9319"       "9576"       "9700"       "9918"       "9985"      
  [79] "10018"      "10111"      "10370"      "10388"      "10426"      "10655"     
  [85] "10744"      "10844"      "10942"      "11022"      "11055"      "11057"     
  [91] "11085"      "11086"      "11105"      "11144"      "22862"      "22917"     
  [97] "23291"      "23310"      "23424"      "23626"      "25788"      "25858"     
 [103] "26108"      "26255"      "26271"      "26528"      "26998"      "27175"     
 [109] "27229"      "27443"      "29781"      "29893"      "43847"      "50511"     
 [115] "51298"      "51314"      "54514"      "54558"      "54586"      "54742"     
 [121] "54763"      "54970"      "55521"      "56154"      "56155"      "56158"     
 [127] "56159"      "56165"      "56907"      "56979"      "57113"      "57151"     
```

```
[133] "57587"     "57820"     "57829"     "58524"     "58531"     "63948"
[139] "63950"     "63951"     "64100"     "64753"     "79084"     "79703"
[145] "79846"     "79925"     "80010"     "80198"     "80217"     "81626"
[151] "81833"     "83449"     "83540"     "83639"     "83893"     "83990"
[157] "84057"     "84071"     "84223"     "84225"     "84229"     "84464"
[163] "84501"     "84690"     "84944"     "85376"     "85378"     "85438"
[169] "89765"     "89869"     "90780"     "91646"     "114791"    "117144"
[175] "117155"    "119710"    "124626"    "124783"    "124817"    "124912"
[181] "126549"    "128153"    "128497"    "130951"    "131375"    "132141"
[187] "135458"    "135927"    "136242"    "139212"    "140894"    "145645"
[193] "146378"    "146849"    "146956"    "147650"    "147872"    "150221"
[199] "150365"    "151246"    "152015"    "154313"    "157695"    "157855"
[205] "158401"    "163589"    "164045"    "168391"    "170370"    "171169"
[211] "171482"    "171483"    "171484"    "197342"    "201254"    "203102"
[217] "221400"    "221711"    "246777"    "254528"    "255101"    "255220"
[223] "257044"    "257062"    "283129"    "283417"    "283847"    "284067"
[229] "284071"    "284359"    "284680"    "285498"    "285588"    "286151"
[235] "286207"    "286826"    "317761"    "339168"    "339345"    "339834"
[241] "340069"    "340719"    "342977"    "346673"    "347732"    "349152"
[247] "374768"    "375337"    "378708"    "378807"    "387885"    "388649"
[253] "389320"    "389852"    "390243"    "400629"    "440804"    "494551"
[259] "644186"    "728637"    "729201"    "768239"    "100125288" "100130988"
[265] "100131137" "100507650" "100996631" "101928601" "107983988" "93426"
[271] "91"        "397"       "480"       "796"       "928"       "1235"
[277] "1392"      "1393"      "2516"      "2661"      "2693"      "3965"
[283] "4838"      "4880"      "5047"      "5864"      "6469"      "6662"
[289] "6736"      "7490"      "9126"      "10497"     "10635"     "22999"
[295] "51738"     "57647"     "64750"     "66037"     "84868"     "130399"
[301] "255061"    "284382"    "373861"    "406949"    "2"         "18"
[307] "51"        "90"        "92"        "100"       "113"       "117"
[313] "133"       "150"       "151"       "174"       "181"       "182"
[319] "183"       "207"       "249"       "269"       "285"       "330"
[325] "338"       "361"       "374"       "383"       "409"       "412"
[331] "472"       "493"       "538"       "546"       "551"       "552"
[337] "558"       "572"       "578"       "581"       "582"       "595"
[343] "596"       "598"       "599"       "604"       "639"       "646"
[349] "652"       "653"       "654"       "655"       "657"       "659"
[355] "666"       "668"       "675"       "682"       "694"       "718"
[361] "734"       "771"       "780"       "811"       "824"       "835"
[367] "836"       "867"       "875"       "881"       "898"       "952"
[373] "989"       "991"       "1027"      "1028"      "1047"      "1069"
[379] "1164"      "1268"      "1271"      "1364"      "1390"      "1399"
[385] "1459"      "1495"      "1499"      "1508"      "1539"      "1543"
```

```
[391] "1545"      "1588"      "1602"      "1617"      "1618"      "1620"
[397] "1621"      "1636"      "1642"      "1718"      "1730"      "1731"
[403] "1738"      "1739"      "1785"      "1788"      "1812"      "1816"
[409] "1822"      "1828"      "1869"      "1906"      "1909"      "1910"
[415] "1912"      "2026"      "2056"      "2057"      "2067"      "2099"
[421] "2120"      "2175"      "2176"      "2182"      "2185"      "2189"
[427] "2192"      "2241"      "2253"      "2254"      "2255"      "2288"
[433] "2296"      "2353"      "2354"      "2475"      "2560"      "2583"
[439] "2591"      "2593"      "2620"      "2623"      "2660"      "2662"
[445] "2674"      "2683"      "2695"      "2706"      "2707"      "2709"
[451] "2735"      "2741"      "2743"      "2796"      "2827"      "2908"
[457] "2956"      "2959"      "3014"      "3037"      "3066"      "3073"
[463] "3074"      "3148"      "3169"      "3171"      "3172"      "3205"
[469] "3206"      "3207"      "3209"      "3235"      "3236"      "3239"
[475] "3291"      "3295"      "3298"      "3301"      "3305"      "3306"
[481] "3309"      "3371"      "3400"      "3417"      "3480"      "3482"
[487] "3485"      "3488"      "3490"      "3516"      "3549"      "3552"
[493] "3553"      "3566"      "3623"      "3625"      "3633"      "3640"
[499] "3643"      "3645"      "3673"      "3675"      "3678"      "3688"
[505] "3690"      "3691"      "3726"      "3753"      "3791"      "3814"
[511] "3815"      "3856"      "3857"      "3880"      "3948"      "3952"
[517] "3955"      "3976"      "3985"      "4033"      "4086"      "4089"
[523] "4090"      "4142"      "4179"      "4188"      "4201"      "4216"
[529] "4221"      "4240"      "4254"      "4292"      "4313"      "4318"
[535] "4323"      "4327"      "4436"      "4487"      "4488"      "4521"
[541] "4627"      "4678"      "4683"      "4693"      "4735"      "4751"
[547] "4808"      "4809"      "4824"      "4846"      "4851"      "4861"
[553] "4882"      "4889"      "4914"      "4920"      "4926"      "4948"
[559] "4956"      "4957"      "4986"      "4987"      "4988"      "4991"
[565] "5000"      "5010"      "5016"      "5017"      "5020"      "5023"
[571] "5028"      "5050"      "5066"      "5079"      "5087"      "5111"
[577] "5154"      "5156"      "5159"      "5176"      "5224"      "5228"
[583] "5232"      "5238"      "5241"      "5266"      "5268"      "5270"
[589] "5324"      "5327"      "5347"      "5360"      "5367"      "5368"
[595] "5469"      "5515"      "5518"      "5535"      "5591"      "5608"
[601] "5618"      "5619"      "5719"      "5724"      "5727"      "5729"
[607] "5730"      "5743"      "5781"      "5806"      "5887"      "5901"
[613] "5914"      "5916"      "5925"      "5926"      "5932"      "5972"
[619] "5997"      "6045"      "6194"      "6196"      "6198"      "6305"
[625] "6382"      "6414"      "6422"      "6423"      "6461"      "6477"
[631] "6498"      "6513"      "6522"      "6532"      "6573"      "6595"
[637] "6615"      "6647"      "6654"      "6670"      "6674"      "6676"
[643] "6677"      "6690"      "6714"      "6715"      "6716"      "6751"
```

```
[649] "6752"  "6753"  "6768"  "6777"  "6781"  "6788"
[655] "6789"  "6790"  "6794"  "6795"  "6812"  "6833"
[661] "6874"  "6875"  "6895"  "6950"  "7004"  "7013"
[667] "7016"  "7026"  "7043"  "7046"  "7048"  "7054"
[673] "7056"  "7067"  "7068"  "7073"  "7079"  "7080"
[679] "7098"  "7103"  "7110"  "7130"  "7182"  "7203"
[685] "7226"  "7257"  "7258"  "7301"  "7314"  "7320"
[691] "7337"  "7345"  "7349"  "7351"  "7372"  "7421"
[697] "7425"  "7458"  "7473"  "7474"  "7476"  "7484"
[703] "7528"  "7536"  "7704"  "7707"  "7855"  "8061"
[709] "8086"  "8204"  "8322"  "8398"  "8399"  "8434"
[715] "8528"  "8531"  "8546"  "8549"  "8614"  "8626"
[721] "8633"  "8654"  "8743"  "8751"  "8820"  "8852"
[727] "8879"  "8894"  "8924"  "8932"  "9104"  "9133"
[733] "9134"  "9148"  "9156"  "9184"  "9191"  "9241"
[739] "9271"  "9289"  "9389"  "9403"  "9420"  "9444"
[745] "9468"  "9509"  "9510"  "9514"  "9519"  "9612"
[751] "9667"  "9724"  "9825"  "9897"  "10049" "10051"
[757] "10096" "10097" "10116" "10134" "10155" "10179"
[763] "10184" "10403" "10461" "10468" "10481" "10491"
[769] "10524" "10549" "10560" "10574" "10575" "10576"
[775] "10592" "10653" "10657" "10661" "10694" "10735"
[781] "10765" "10818" "10881" "10919" "10927" "10935"
[787] "10959" "10991" "11189" "11218" "11251" "11315"
[793] "11331" "22836" "22933" "22948" "22994" "23064"
[799] "23157" "23205" "23230" "23304" "23345" "23353"
[805] "23394" "23397" "23399" "23411" "23414" "23492"
[811] "23542" "23598" "23609" "23633" "23641" "23705"
[817] "23762" "24145" "24149" "25777" "25809" "25831"
[823] "25911" "25932" "25945" "25976" "26003" "26064"
[829] "26165" "26206" "26471" "27030" "27125" "27127"
[835] "27136" "27285" "27306" "29844" "29924" "29974"
[841] "29988" "49855" "50487" "50846" "51087" "51247"
[847] "51343" "51361" "51460" "51465" "51542" "51665"
[853] "51742" "51807" "53340" "54106" "54361" "54407"
[859] "54457" "54466" "54585" "54851" "54852" "54888"
[865] "54993" "55064" "55120" "55124" "55231" "55329"
[871] "55342" "55366" "55585" "55636" "55706" "55723"
[877] "55811" "55815" "55818" "55840" "55870" "55905"
[883] "56163" "56729" "56848" "56956" "57054" "57055"
[889] "57095" "57097" "57135" "57178" "57599" "57728"
[895] "57731" "57828" "59272" "59338" "59343" "59350"
[901] "63894" "63946" "63978" "64224" "64321" "64395"
```

```
 [907] "64396"     "64478"     "64591"     "64645"     "64847"     "64848"
 [913] "65010"     "65110"     "79582"     "79727"     "79747"     "79820"
 [919] "79893"     "79969"     "79977"     "79989"     "80000"     "80025"
 [925] "81539"     "81616"     "81623"     "81671"     "81892"     "81930"
 [931] "83700"     "83890"     "83943"     "84056"     "84132"     "84152"
 [937] "84159"     "84168"     "84172"     "84215"     "84221"     "84687"
 [943] "84688"     "84694"     "84812"     "84930"     "85315"     "85413"
 [949] "85417"     "89766"     "89887"     "91746"     "91978"     "93649"
 [955] "113746"    "114112"    "115948"    "116369"    "117154"    "120892"
 [961] "121355"    "122042"    "122258"    "122664"    "124404"    "125972"
 [967] "128637"    "130497"    "132243"    "132612"    "133558"    "135138"
 [973] "135935"    "139886"    "143471"    "143689"    "144195"    "144535"
 [979] "146310"    "146852"    "147912"    "148229"    "148327"    "149685"
 [985] "150159"    "151056"    "151195"    "151449"    "152006"    "152586"
 [991] "157506"    "157777"    "158062"    "161829"    "164091"    "164395"
 [997] "164684"    "166378"    "169981"    "170690"    "192670"    "199720"
[1003] "200232"    "200373"    "202051"    "203523"    "204474"    "219670"
[1009] "219793"    "219938"    "221656"    "221823"    "222698"    "225689"
[1015] "259266"    "261734"    "283471"    "283629"    "283677"    "284338"
[1021] "285335"    "286128"    "286234"    "317719"    "338879"    "339906"
[1027] "340784"    "344758"    "353189"    "373863"    "387712"    "388799"
[1033] "389730"    "389761"    "389762"    "389763"    "431707"    "440699"
[1039] "440822"    "441452"    "449520"    "474343"    "619556"    "642623"
[1045] "642636"    "644150"    "644890"    "645832"    "645961"    "646480"
[1051] "647060"    "727830"    "727905"    "728132"    "728137"    "728395"
[1057] "728403"    "729967"    "100130958" "100289087" "102724560" "1396"
[1063] "1594"      "2626"      "2627"      "3714"      "5310"      "6092"
[1069] "6299"      "6586"      "7022"      "7482"      "8433"      "8510"
[1075] "8644"      "8909"      "9353"      "10046"     "10361"     "10732"
[1081] "22803"     "22873"     "26292"     "29127"     "54456"     "56977"
[1087] "84073"     "116832"    "132625"    "138474"    "140732"    "140801"
[1093] "3624"      "4762"      "5810"      "6627"      "140947"    "497189"
[1099] "226"       "708"       "1672"      "1767"      "2013"      "2295"
[1105] "2529"      "2697"      "3479"      "6658"      "6691"      "6926"
[1111] "7076"      "7417"      "7422"      "8701"      "8890"      "8892"
[1117] "8893"      "8912"      "9083"      "9898"      "10265"     "10699"
[1123] "10734"     "10916"     "23639"     "25790"     "25836"     "25981"
[1129] "27019"     "27161"     "51673"     "51759"     "55036"     "55743"
[1135] "55779"     "57119"     "57122"     "57697"     "60675"     "64220"
[1141] "79816"     "80314"     "84074"     "84660"     "84733"     "85360"
[1147] "89876"     "122402"    "131118"    "144132"    "144406"    "146845"
[1153] "148281"    "150921"    "151648"    "154197"    "159686"    "199223"
[1159] "219990"    "253943"    "286464"    "338323"    "339829"    "346288"
```

```
[1165] "347688"    "377630"    "378948"    "401024"    "402573"    "406950"
[1171] "442867"    "442868"    "100506013" "109"       "142"       "351"
[1177] "406"       "583"       "585"       "658"       "676"       "1051"
[1183] "1080"      "1435"      "1525"      "1811"      "1843"      "1958"
[1189] "2054"      "2069"      "2263"      "2625"      "2649"      "2678"
[1195] "2712"      "2801"      "2879"      "3622"      "3953"      "3975"
[1201] "4036"      "4184"      "4192"      "4603"      "4753"      "4867"
[1207] "4878"      "5049"      "5104"      "5125"      "5127"      "5350"
[1213] "5414"      "5566"      "5764"      "5798"      "5872"      "5950"
[1219] "5990"      "6098"      "6652"      "6665"      "6774"      "6901"
[1225] "6943"      "7042"      "7584"      "7589"      "7917"      "8085"
[1231] "8195"      "8243"      "8320"      "8372"      "8382"      "9025"
[1237] "9421"      "9425"      "9575"      "9633"      "9665"      "9698"
[1243] "9702"      "9940"      "10038"     "10409"     "10420"     "10519"
[1249] "10733"     "11020"     "11063"     "11077"     "22887"     "23139"
[1255] "23198"     "23213"     "23236"     "23315"     "23318"     "23617"
[1261] "23627"     "26038"     "26091"     "26140"     "26330"     "27120"
[1267] "28981"     "29118"     "29947"     "30812"     "51441"     "51547"
[1273] "51574"     "51668"     "51804"     "54760"     "54890"     "54937"
[1279] "55063"     "55726"     "55810"     "56262"     "56339"     "56603"
[1285] "56776"     "57721"     "58494"     "63979"     "64147"     "64207"
[1291] "64518"     "78995"     "79173"     "79645"     "79670"     "79733"
[1297] "81492"     "81629"     "83447"     "83853"     "83942"     "83983"
[1303] "84072"     "84236"     "84515"     "84519"     "84678"     "84691"
[1309] "90410"     "90853"     "91603"     "94107"     "126206"    "127579"
[1315] "133308"    "136332"    "136991"    "143678"    "144455"    "147700"
[1321] "149095"    "150280"    "151254"    "153218"    "157680"    "160762"
[1327] "161142"    "161514"    "161931"    "162540"    "162979"    "164714"
[1333] "170506"    "200162"    "200558"    "201164"    "202500"    "203074"
[1339] "221481"    "245711"    "254394"    "255626"    "256006"    "256710"
[1345] "326340"    "338773"    "341277"    "341567"    "346653"    "375189"
[1351] "375341"    "388336"    "388553"    "391714"    "399949"    "402381"
[1357] "431705"    "441161"    "642658"    "643376"    "646799"    "730249"
[1363] "101927581" "56"        "247"       "259"       "283"       "996"
[1369] "1081"      "1394"      "1538"      "2302"      "2692"      "4117"
[1375] "5069"      "5670"      "5675"      "5885"      "6013"      "7356"
[1381] "7455"      "8605"      "8697"      "8881"      "9130"      "10393"
[1387] "10658"     "10983"     "23742"     "23780"     "25847"     "25906"
[1393] "26256"     "29882"     "29945"     "51433"     "51434"     "51529"
[1399] "55339"     "56648"     "56853"     "57082"     "57446"     "64682"
[1405] "79400"     "80237"     "84203"     "84750"     "93185"     "113451"
[1411] "116138"    "117285"    "119504"    "219771"    "246184"    "256126"
[1417] "344018"    "406991"    "441531"    "728695"    "100137049" "116"
```

```
[1423] "841"       "983"       "1017"      "1082"      "1307"      "1586"
[1429] "1833"      "2491"      "2834"      "3293"      "3593"      "3620"
[1435] "3972"      "4012"      "4486"      "4543"      "5073"      "5467"
[1441] "5568"      "5571"      "5669"      "5671"      "5672"      "5673"
[1447] "5676"      "5678"      "5680"      "5737"      "5744"      "5858"
[1453] "5890"      "5987"      "6019"      "6046"      "6159"      "6696"
[1459] "6732"      "6863"      "6866"      "7005"      "7156"      "7216"
[1465] "7516"      "7812"      "7932"      "7993"      "8031"      "8239"
[1471] "8287"      "8607"      "8900"      "8999"      "9082"      "9085"
[1477] "9125"      "9183"      "9210"      "9426"      "9463"      "10007"
[1483] "10017"     "10149"     "10343"     "10406"     "10407"     "10522"
[1489] "10566"     "10609"     "10766"     "10863"     "10876"     "23764"
[1495] "26476"     "26609"     "26664"     "30014"     "51207"     "51686"
[1501] "53405"     "80705"     "94027"     "203611"    "253175"    "728712"

$`GO:0000009`
[1] "79087" "55650"

$`GO:0000010`
[1] "23590" "57107"

$`GO:0000012`
 [1] "54840"     "55775"     "1161"      "2074"      "3981"      "7141"
 [7] "7515"      "100133315" "142"       "23411"     "200558"    "7014"

$`GO:0000014`
 [1] "2021"  "2072"  "4361"  "6419"  "9941"  "2067"  "10111" "64421" "7515"
[10] "5932"
```

Using `lapply` and `sapply`

If we want to know the number of genes for each gene set.

What is `ngs`?

```r
class(ngs)
```

Using `unlist` we have a vector with the lengths of the different gene sets.

Probably a better choice to calculate the lengths would be

```r
ngs = sapply(hsa_go,length)
```

Now ngs is

```
class(ngs)
```

```
[1] "integer"
```

A summary of the lengths is

```
summary(ngs)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    1.0     2.0     6.0    90.3    24.0 19518.0 
```

We can filter those gene sets with less than 10 genes per set,

```
hsa_go1 = hsa_go[ngs>=10]
```

or those gene sets with a number of genes greater or equal to 10 and lesser or equal to 10 with

```
hsa_go1 = hsa_go[ngs>=10 & ngs<=10]
```

ora for gse21942

An over-representation analysis of the data set gse21942 will be done using the gene set list hsa_go previously loaded. No userGeneSet is provided and this significant gene set is defined as those genes with an adjusted p-value lesser that fdr. The fdr value, test and correction have been given in dema.

```
gse21942_ora = overRepresentation(gse21942_rowt,minsize=10,maxsize=100,
                            correction = "BH",
                            GeneSetList = hsa_go,
                            foutput="gse21942_ora")
```

A data.frame with the results is obtained with

```
gse21942_ora_df = tidy(gse21942_ora)
```

The report in a html file is obtained as usual as

```
glimpse(gse21942_ora)
```

It can be opened simply with

```
browseURL(glimpse(gse21942_ora))
```

`ora` for PRJNA297664

We are going to do an over-representation analysis of the dataset PRJNA297664. The significant gene set is calculated as the set of genes with an adjusted p-value lesser or equal to `fdr`.

First we need a gene set list.

```
PRJNA297664_ora = overRepresentation(PRJNA297664_erc,
                                    minsize=10,maxsize=100,
                                    correction = "BH",
                                    GeneSetList = sce_go,
                                    foutput="gse21942_ora")
```

The report in a html file is obtained as usual as

```
glimpse(PRJNA297664_ora)
```

It can be opened simply with

```
browseURL(glimpse(PRJNA297664_ora))
```

A `data.frame` with the results is given by

We can provide the significant gene set. Let us evaluate this significant gene set and will give it `overRepresentation`. We use the slot `GeneStat` of the `DifferentialExpressionOutput` object.

```
GeneStat(PRJNA297664_erc)
```

The significant gene set will be composed by those genes with an adjusted p-value lesser or equal to 0.001. The row indices in the expression matrix, `assay(PRJNA297664)`, is

```
(sig0 = which(GeneStat(PRJNA297664_erc)[,"adjp"] < .0001))
```

```
  [1]   66   87  141  180  208  232  318  368  391  414  473  498  499  500  513
 [16]  543  667  706  708  722  741  744  834  871  872  882  968 1047 1070 1173
 [31] 1259 1277 1285 1328 1330 1389 1449 1509 1511 1563 1625 1641 1650 1758 1762
 [46] 1814 1840 1847 1854 1863 1869 1876 1916 1952 1963 2009 2108 2123 2150 2152
 [61] 2162 2180 2222 2257 2341 2346 2420 2437 2443 2452 2464 2487 2515 2524 2534
 [76] 2578 2579 2582 2694 2698 2699 2776 2785 2855 2920 3068 3069 3086 3155 3220
 [91] 3231 3253 3315 3348 3358 3377 3385 3400 3430 3440 3473 3520 3573 3617 3686
[106] 3702 3730 3833 3840 3852 3903 3971 4104 4119 4141 4142 4174 4193 4220 4235
[121] 4260 4261 4385 4387 4434 4450 4484 4485 4496 4570 4633 4673 4696 4713 4733
[136] 4766 4777 4808 4901 4904 4907 4908 4928 5014 5020 5022 5025 5026 5027 5038
[151] 5091 5215 5216 5246 5260 5345 5392 5432 5464 5473 5483 5494 5551 5573 5583
[166] 5607 5624 5633 5634 5638 5707 5720 5780 5821 5829 5943 5962 6011 6063 6089
[181] 6161 6185 6199 6238 6287 6307 6324 6338 6364 6393 6484 6488 6491 6556 6591
[196] 6647 6650 6651 6671 6687 6688 6695
```

Note that the gene set list is specified using `ENTREZID` identifiers. The significant gene set has to be specified using this identifier.

```
(SigGeneSet0 = GeneData(PRJNA297664_erc)[sig0,"ENTREZID"])
```

```
  [1] "851259" "851209" "851230" "851299" "852259" "852237" "852291" "852343"
  [9] "852365" "852390" "852442" "852467" "852468" "852469" "852481" "852507"
 [17] "850334" "850297" "850295" "850361" "850385" "850388" "851561" "851524"
 [25] "851522" "851512" "851426" "851323" "851371" "851646" "851733" "851751"
 [33] "851759" "851800" "851802" "851862" "851925" "851986" "851987" "852041"
 [41] NA       "852119" "852127" "856648" "856644" "856754" "856779" "856786"
 [49] "856793" "856801" "856807" "856812" "856847" "856885" "856893" NA      
 [57] "850569" "850588" NA       "850614" "852878" "852863" "852818" "852783"
 [65] "852696" "852691" "852637" "852891" "852897" "852905" "852920" "852943"
 [73] "852970" "852979" "852990" "853039" NA       "853043" "853159" "853163"
 [81] "853164" "856365" "856355" "856432" "856487" "856625" "856626" "854805"
 [89] "854740" "854676" "854666" "854647" "854853" "853428" "853418" "853395"
 [97] "853386" "853371" "853342" "853332" "853300" "853252" "853466" "853514"
[105] "853586" "853602" "853869" "853766" "853760" "853748" "853696" "853872"
[113] "850636" "850620" "850665" "850664" "850711" "850731" "850758" "850773"
[121] "850798" "850799" "850911" "850914" "850963" "850979" "851010" "851013"
[129] "851023" "851098" "851158" "855005" "854981" "854964" "854943" "854883"
[137] "854872" "854911" "855102" "855105" "855108" "855109" "855130" "855217"
[145] "855222" "855224" "855227" "855228" "855229" "855239" "855291" "855692"
[153] "855691" "855660" "855647" "855567" "855520" "855480" "855448" "855440"
[161] "855427" "855416" "855748" "855770" "855780" "855805" "854155" "854146"
[169] "854145" "854141" "854070" "854058" "854013" "854185" "854192" "854303"
```

```
[177] "854324" "854370" "854421" "854447" "854516" "854542" "854556" "856093"
[185] "856044" "856024" "856007" "855993" "855968" "855940" "855825" "855822"
[193] "855819" "856143" "856178" "856237" NA       "856241" "856263" "856279"
[201] "856280" "856287"
```

Note that some genes have no ENTREZID identifier. It does not matter. The analysis will remove these genes. Now we call overRepresentation indicating the significant gene set.

```
PRJNA297664_ora = overRepresentation(PRJNA297664_erc,minsize=10,maxsize=100,
                              correction = "BH",
                              GeneSetList = sce_go,
                              SigGeneSet = SigGeneSet0,
                              foutput="gse21942_ora")
```

Gene set analysis

Introduction

The over-representation analysis is a gene set analysis procedure. In this section we use other procedures.

gse21942

It means that we use rowtmod for the marginal differential expression analysis. The null distribution to be tested is the randomization distribution, GeneNullDistr = "randomization", the gene set null distribution is the self-contained hypothesis, GeneSetNullDistr = "self-contained", the maximum number of simulations is given by nmax. The name of the identifiers used in the gene set collection gsc is id. The function has as an argument the gene set collection gsc. The enrichment method is a function descriptive. Finally, the file foutput is used to save the html report.

```
gse21942_self_mean = GeneSetTest(x = gse21942,y="FactorValue..DISEASE.STATE.",
          test = rowtmod,association="statistic",correction="BH",
          GeneNullDistr = "randomization",
          GeneSetNullDistr = "self-contained",
          alternative="two-sided",nmax = 100,
          id = "ENTREZID",gsc=hsa_go,descriptive=mean,
          foutput = "gse21942_self_mean")
```

Now we reproduce the previous analysis using as descriptive the function maxmean.

```
gse21942_self_maxmean =
  GeneSetTest(x = gse21942,y="FactorValue..DISEASE.STATE.",
         test = rowtmod,association="statistic",correction="BH",
         GeneNullDistr = "randomization",
         GeneSetNullDistr = "self-contained",
         alternative="two-sided",nmax = 100,
         id = "ENTREZID",gsc=hsa_go,descriptive= maxmean,
         foutput = "gse21942_self_maxmean")
```

Or the `median`.

```
gse21942_self_median =
  GeneSetTest(x = gse21942,y="FactorValue..DISEASE.STATE.",
         test = rowtmod,association="statistic",correction="BH",
         GeneNullDistr = "randomization",
         GeneSetNullDistr = "self-contained",
         alternative="two-sided",nmax = 100,
         id = "ENTREZID",gsc=hsa_go,descriptive= median,
         foutput = "gse21942_self_maxmean")
```

We can test the `competitive` hypothesis using as enrichment method the `maxmean`.

```
gse21942_comp_mean =
  GeneSetTest(x = gse21942,y="FactorValue..DISEASE.STATE.",
         test = rowtmod,association="statistic",correction="BH",
         GeneNullDistr = "randomization",
         GeneSetNullDistr = "competitive",
         alternative="two-sided",nmax = 100,
         id = "ENTREZID",gsc=hsa_go,descriptive=maxmean,
         foutput = "gse21942_comp_mean")
```

```
gse21942_comp_maxmean =
  GeneSetTest(x = gse21942,y="FactorValue..DISEASE.STATE.",
         test = rowtmod,association="statistic",correction="BH",
         GeneNullDistr = "randomization",
         GeneSetNullDistr = "competitive",
         alternative="two-sided",nmax = 100,
         id = "ENTREZID",gsc=hsa_go,descriptive=maxmean,
         foutput = "gse21942_comp_maxmean")
```

```
gse21942_comp_median =
  GeneSetTest(x = gse21942,y="FactorValue..DISEASE.STATE.",
         test = rowtmod,association="statistic",correction="BH",
         GeneNullDistr = "randomization",
         GeneSetNullDistr = "competitive",
         alternative="two-sided",nmax = 100,
         id = "ENTREZID",gsc=hsa_go,descriptive=median,
         foutput = "gse21942_comp_maxmean")
```

Now the report can be obtained as a `data.frame`

```
gse21942_self_mean_df = tidy(gse21942_self_mean)
gse21942_comp_mean_df = tidy(gse21942_comp_mean)
gse21942_self_maxmean_df = tidy(gse21942_self_maxmean)
gse21942_comp_maxmean_df = tidy(gse21942_comp_maxmean)
gse21942_self_median_df = tidy(gse21942_self_median)
gse21942_comp_median_df = tidy(gse21942_comp_median)
```

or like a html report

```
glimpse(gse21942_self_mean)
glimpse(gse21942_comp_mean)
glimpse(gse21942_self_maxmean)
glimpse(gse21942_comp_maxmean)
```

that can be opened with

```
browseURL(glimpse(gse21942_self_mean))
browseURL(glimpse(gse21942_comp_mean))
browseURL(glimpse(gse21942_self_maxmean))
browseURL(glimpse(gse21942_comp_maxmean))
```

PRJNA297664

It means that we use `rowtmod` for the marginal differential expression analysis. The null distribution to be tested is the `randomization` distribution and the enrichment measure is the mean of the phenotype-expression association that it has been chosen as the `statistic` of the hypothesis testing procedure.

```
load(paste0(dirTamiData,"sce_go.rda"))
```

```
PRJNA297664_self =
    GeneSetTest(x = PRJNA297664,y="treatment",
                test = edgercommon,association="statistic",
                correction="BH",
                GeneNullDistr = "randomization",
                GeneSetNullDistr = "self-contained",
                alternative="two-sided",nmax = 100,
                id = "ORF",gsc=sce_go,descriptive=mean,
                foutput = "PRJNA297664_self")
```

```
PRJNA297664_comp =
    GeneSetTest(x = PRJNA297664,y="treatment",
                test = edgercommon,association="statistic",
                correction="BH",
                GeneNullDistr = "randomization",
                GeneSetNullDistr = "competitive",
                alternative="two-sided",nmax = 100,
                id = "ORF",gsc=sce_go,descriptive=mean,
                foutput = "PRJNA297664_comp")
```

Now the report can be obtained as a `data.frame`

```
PRJNA297664_self_df = tidy(PRJNA297664_self)
PRJNA297664_comp_df = tidy(PRJNA297664_comp)
```

or like a html report

```
glimpse(PRJNA297664_self)
glimpse(PRJNA297664_comp)
```

that can be opened with

```
browseURL(glimpse(PRJNA297664_self))
browseURL(glimpse(PRJNA297664_comp))
```