Comparison of E(z) mutant brain expression profiles and age-correlated probesets indicates a “younger” brain

Jason Kennerdell

9/25/2017

Input the data and metadata:

```
inpath <- "~/Desktop/brain"
outpath <- "~/Desktop/brain/_8C_orderedList"
setwd(inpath)
files <- list.files()
htseq_files <- files[grepl("^JKL.*txt$", files)]
sampleNames <- read.csv("EzSampleNames.csv")
sampleTable <- data.frame(fileName = htseq_files,
                          stringsAsFactors=FALSE)
sampleTable$Library <- gsub("-counts.txt", "", sampleTable$fileName)
sampleTable$Library <- gsub("b", "", sampleTable$Library)
sampleTable$seq.batch <- ifelse(grepl("b", sampleTable$fileName), "B", "A")
sampleTable$seq.batch <- paste(sampleTable$Library, sampleTable$seq.batch)
sampleTable <- merge(sampleTable[,c(1,3)], sampleNames, by = "seq.batch")
sampleTable$genotype <- gsub("-.*$", "", sampleTable$Sample)
sampleTable$genotype <- factor(sampleTable$genotype, levels = c("w", "Ez"))
sampleTable$Temp <- gsub("[A-Z, a-z, -]", "", sampleTable$Sample)
sampleTable$Temp <- factor(sampleTable$Temp, levels = c("25", "29"))
sampleTable$rep <- gsub("^.*-", "", sampleTable$Sample)
sampleTable$condition <- paste(sampleTable$genotype, sampleTable$Temp, sep = "-")
sampleTable
```

```
##    seq.batch          fileName    Sample batch Library genotype Temp rep
## 1    JKL10 A  JKL10-counts.txt   w-25-II     A   JKL10        w   25  II
## 2    JKL10 B JKL10b-counts.txt   w-25-II     B   JKL10        w   25  II
## 3    JKL11 A  JKL11-counts.txt  w-25-III     A   JKL11        w   25 III
## 4    JKL11 B JKL11b-counts.txt  w-25-III     B   JKL11        w   25 III
## 5    JKL12 A  JKL12-counts.txt Ez-25-III     A   JKL12       Ez   25 III
## 6    JKL12 B JKL12b-counts.txt Ez-25-III     B   JKL12       Ez   25 III
## 7    JKL13 A  JKL13-counts.txt    w-29-I     A   JKL13        w   29   I
## 8    JKL13 B JKL13b-counts.txt    w-29-I     B   JKL13        w   29   I
## 9    JKL14 A  JKL14-counts.txt   Ez-29-I     A   JKL14       Ez   29   I
## 10   JKL14 B JKL14b-counts.txt   Ez-29-I     B   JKL14       Ez   29   I
## 11   JKL15 A  JKL15-counts.txt   w-29-II     A   JKL15        w   29  II
## 12   JKL15 B JKL15b-counts.txt   w-29-II     B   JKL15        w   29  II
## 13   JKL16 A  JKL16-counts.txt  Ez-29-II     A   JKL16       Ez   29  II
## 14   JKL16 B JKL16b-counts.txt  Ez-29-II     B   JKL16       Ez   29  II
## 15   JKL17 A  JKL17-counts.txt  w-29-III     A   JKL17        w   29 III
## 16   JKL17 B JKL17b-counts.txt  w-29-III     B   JKL17        w   29 III
## 17   JKL18 A  JKL18-counts.txt Ez-29-III     A   JKL18       Ez   29 III
## 18   JKL18 B JKL18b-counts.txt Ez-29-III     B   JKL18       Ez   29 III
## 19   JKL19 A  JKL19-counts.txt   w-29-IV     A   JKL19        w   29  IV
## 20   JKL19 B JKL19b-counts.txt   w-29-IV     B   JKL19        w   29  IV
## 21   JKL20 A  JKL20-counts.txt  Ez-29-IV     A   JKL20       Ez   29  IV
## 22   JKL20 B JKL20b-counts.txt  Ez-29-IV     B   JKL20       Ez   29  IV
## 23   JKL21 A  JKL21-counts.txt    w-29-V     A   JKL21        w   29   V
## 24   JKL21 B JKL21b-counts.txt    w-29-V     B   JKL21        w   29   V
## 25   JKL22 A  JKL22-counts.txt   Ez-29-V     A   JKL22       Ez   29   V
## 26   JKL22 B JKL22b-counts.txt   Ez-29-V     B   JKL22       Ez   29   V
## 27    JKL7 A   JKL7-counts.txt    w-25-I     A    JKL7        w   25   I
## 28    JKL7 B  JKL7b-counts.txt    w-25-I     B    JKL7        w   25   I
## 29    JKL8 A   JKL8-counts.txt   Ez-25-I     A    JKL8       Ez   25   I
## 30    JKL8 B  JKL8b-counts.txt   Ez-25-I     B    JKL8       Ez   25   I
## 31    JKL9 A   JKL9-counts.txt  Ez-25-II     A    JKL9       Ez   25  II
## 32    JKL9 B  JKL9b-counts.txt  Ez-25-II     B    JKL9       Ez   25  II
##    condition
## 1        w-25
## 2        w-25
## 3        w-25
## 4        w-25
## 5       Ez-25
## 6       Ez-25
## 7        w-29
## 8        w-29
## 9       Ez-29
## 10      Ez-29
## 11       w-29
## 12       w-29
## 13      Ez-29
## 14      Ez-29
## 15       w-29
## 16       w-29
## 17      Ez-29
## 18      Ez-29
## 19       w-29
## 20       w-29
## 21      Ez-29
## 22      Ez-29
## 23       w-29
## 24       w-29
## 25      Ez-29
## 26      Ez-29
## 27       w-25
## 28       w-25
## 29      Ez-25
## 30      Ez-25
## 31      Ez-25
## 32      Ez-25
```

Set up the statistical model

```
design <- formula(~ Temp + genotype)
```

DESEq2 Statistics

```
dds <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable, directory = inpath, design = design)
# Combine the technical replicates (different runs) by adding the count
# totals for each gene across the two runs:
dds <- collapseReplicates(dds, groupby=dds$Library, run = dds$batch)
dds <- DESeq(dds, betaPrior=T)
# What does the data look like?
head(assay(dds)) # This is the sum of the two runs HTSeq-count output!
```

```
##             JKL10 JKL11 JKL12 JKL13 JKL14 JKL15 JKL16 JKL17 JKL18 JKL19
## FBgn0000003     0     0     0     0     0     0     0     0     0     0
## FBgn0000008  1444  1874  1687  1305  1453  1725  1529  1856  1500  1558
## FBgn0000014     0     0     0     0     0     0     1     0     1     0
## FBgn0000015     1     6     1     3     0     1     1     1     2     0
## FBgn0000017  9186 10798  9189  6877  8808  9121  9834 11313  9272  8564
## FBgn0000018   262   306   286   336   288   317   272   346   274   285
##             JKL20 JKL21 JKL22  JKL7  JKL8  JKL9
## FBgn0000003     1     0     0     0     0     1
## FBgn0000008  1353  1589  1367  1980  1861  1884
## FBgn0000014     1     1     1     0     3     0
## FBgn0000015     0     1     0     0     0     2
## FBgn0000017  8697  8196  8612 11453 11360 11139
## FBgn0000018   285   238   284   286   333   333
```

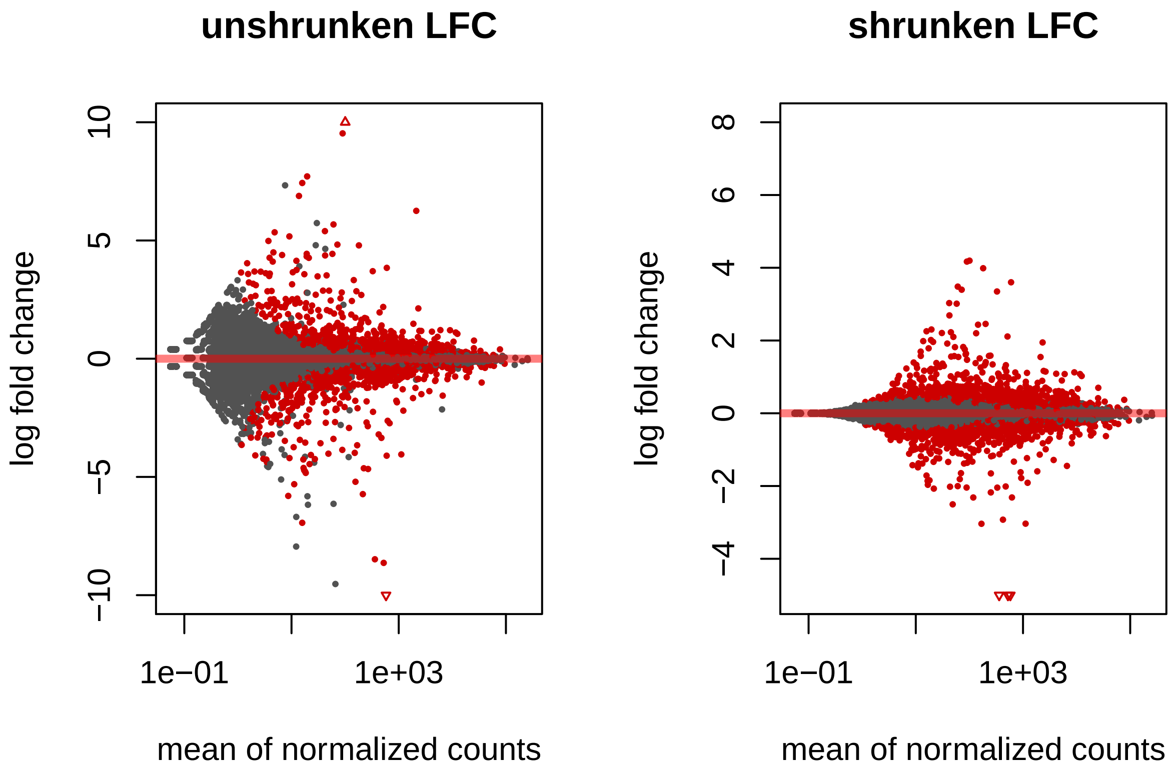
```
# What are the columns?
colData(dds)
```

```
## DataFrame with 16 rows and 9 columns
##          Sample    batch  Library genotype     Temp         rep
##        <factor> <factor> <factor> <factor> <factor> <character>
## JKL10   w-25-II        A    JKL10        w       25          II
## JKL11  w-25-III        A    JKL11        w       25         III
## JKL12 Ez-25-III        A    JKL12       Ez       25         III
## JKL13    w-29-I        A    JKL13        w       29           I
## JKL14   Ez-29-I        A    JKL14       Ez       29           I
## ...         ...      ...      ...      ...      ...         ...
## JKL21    w-29-V        A    JKL21        w       29           V
## JKL22   Ez-29-V        A    JKL22       Ez       29           V
## JKL7     w-25-I        A     JKL7        w       25           I
## JKL8    Ez-25-I        A     JKL8       Ez       25           I
## JKL9   Ez-25-II        A     JKL9       Ez       25          II
##         condition runsCollapsed sizeFactor
##       <character>   <character>  <numeric>
## JKL10        w-25           A,B  0.9090574
## JKL11        w-25           A,B  1.1403424
## JKL12       Ez-25           A,B  1.0027251
## JKL13        w-29           A,B  1.0220507
## JKL14       Ez-29           A,B  0.9570594
## ...           ...           ...        ...
## JKL21        w-29           A,B  0.8750556
## JKL22       Ez-29           A,B  0.9618390
## JKL7         w-25           A,B  1.1341278
## JKL8        Ez-25           A,B  1.1093046
## JKL9         Ez-25           A,B   1.1303920
```

```
results <- results(dds, alpha=0.05)
results$ensembl <- rownames(results)
```

Prepare MA plots:

```
# For Maximum likelihood estimates:
resultsMLE <- results(dds, addMLE=TRUE, alpha = 0.05)
par(mfrow=c(1,2))
plotMA(resultsMLE, MLE=TRUE, alpha = 0.05, main="unshrunken LFC", ylim=c(-10,10))
plotMA(results, alpha = 0.05, main="shrunken LFC", ylim=c(-5,8))
```



Add Annotation

```
# Add usefull gene names:
library(biomaRt)

# Add FBgn Names
mart = useMart("ENSEMBL_MART_ENSEMBL", host="aug2017.archive.ensembl.org")
#listDatasets(mart)
mart = useMart("ENSEMBL_MART_ENSEMBL", host="aug2017.archive.ensembl.org",
               dataset = "dmelanogaster_gene_ensembl")
genemap <- getBM(attributes = c("ensembl_gene_id", "entrezgene", "external_gene_name", "flybasecgid_gen
idx <- match(results$ensembl, genemap$ensembl_gene_id)
results$entrez <- genemap$entrezgene[idx]
```

```
results$geneSymbol <- genemap$external_gene_name[idx]
results$cg <- genemap$flybasecgid_gene[idx]
```

Prepare Ranked lists of genes for OrderedList analysis

```
# Using age correlated genes by p value and beta coefficient:
correl_lst <- read.csv("~/Desktop/brain/GSE25007/ranked_aging_correlated_genes_pvalue_only.csv")
# How many genes are there in this list?
dim(correl_lst)[1]
```

```
## [1] 11651
```

```
results_lst <- as.data.frame(results)
results_lst <- results_lst[!is.na(results_lst$pvalue),]
results_lst$product <- -log10(results_lst$pvalue)*results_lst$log2FoldChange
results_lst <- results_lst[order(results_lst$product),]
# How many genes are there in this list?
dim(results_lst)[1]
```

```
## [1] 15167
```

```
# Select only members of lists that are present in the opposite list
results_lst <- results_lst[results_lst$ensembl %in% correl_lst$ensembl,]
head(results_lst)
```

```
##              baseMean log2FoldChange      lfcSE      stat        pvalue
## FBgn0015037  422.7991      -2.925188 0.08788261 -33.28517 6.328883e-243
## FBgn0039321  926.5579      -1.780766 0.05706890 -31.20380 9.462272e-214
## FBgn0033395 1117.2706      -3.035148 0.13523096 -22.44418 1.458786e-111
## FBgn0031489  621.9415      -2.314285 0.10254531 -22.56841 8.856526e-113
## FBgn0003227  331.5243      -2.043967 0.09020525 -22.65906 1.135676e-113
## FBgn0036790 1848.6685      -1.596376 0.06316438 -25.27335 6.272728e-141
##                      padj     ensembl entrez geneSymbol      cg   product
## FBgn0015037 8.401241e-240 FBgn0015037  45524     Cyp4p1 CG10842 -708.4766
## FBgn0039321 1.130458e-210 FBgn0039321  43061    CG10550 CG10550 -379.3460
## FBgn0033395 7.577443e-109 FBgn0033395  35946     Cyp4p2  CG1944 -336.4036
## FBgn0031489 4.809496e-110 FBgn0031489  33510    CG17224 CG17224 -259.3220
## FBgn0003227 6.783959e-111 FBgn0003227  49241        rec CG31293 -230.8553
## FBgn0036790 4.683768e-138 FBgn0036790  40020    AstC-R1  CG7285 -223.8159
```

```
tail(results_lst)
```

```
##              baseMean log2FoldChange      lfcSE     stat        pvalue
## FBgn0052365 200.54575       2.456391 0.08867186 27.70203 6.599880e-169
## FBgn0029114 513.57093       2.110276 0.05834028 36.17186 1.686935e-286
## FBgn0054031 100.54137       4.194846 0.14435159 29.05992 1.152960e-185
## FBgn0052437  89.80844       4.169186 0.14265941 29.22476 9.401423e-188
## FBgn0029924 327.41872       3.347829 0.09073609 36.89633 5.291929e-298
## FBgn0035999 599.31421       3.600854 0.07547901 47.70670  0.000000e+00
##                      padj     ensembl  entrez geneSymbol      cg  product
## FBgn0052365 6.065289e-166 FBgn0052365   38876    CG32365 CG32365 413.1169
## FBgn0029114 3.358969e-283 FBgn0029114   44497      Tollo  CG6890 603.0597
## FBgn0054031 1.147868e-182 FBgn0054031 3885665    CG34031 CG34031 775.7872
## FBgn0052437 1.021080e-184 FBgn0052437   40366    CG32437 CG32437 779.7496
## FBgn0029924 1.264454e-294 FBgn0029924   31641     CG4586  CG4586 995.2304
## FBgn0035999  0.000000e+00 FBgn0035999   39097     CG3552  CG3552      Inf
```

```
# How many genes are there in this list?
dim(results_lst)[1]
```

```
## [1] 11096
```

```
correl_lst <- correl_lst[correl_lst$ensembl %in% results_lst$ensembl,]
head(correl_lst)
```

```
##      X     ensembl       beta      p.value      prod
## 1 3946 FBgn0031775 0.09066047 9.907021e-08 0.6349911
## 2 8941 FBgn0040735 0.02357414 5.324881e-07 0.1478968
## 3 2651 FBgn0029507 0.08291106 5.737657e-07 0.5174699
## 4 8939 FBgn0040733 0.02273366 4.631188e-06 0.1212683
## 5 3366 FBgn0030808 0.02275069 6.806960e-06 0.1175539
## 6 3046 FBgn0030310 0.04808252 7.391507e-06 0.2467242
```

```
tail(correl_lst)
```

```
##           X     ensembl         beta      p.value        prod
## 11646  6124 FBgn0035649 -0.018821410 4.377408e-05 -0.08203844
## 11647   692 FBgn0004242 -0.005928273 3.225493e-05 -0.02662627
## 11648 10122 FBgn0053202 -0.015904473 2.073152e-05 -0.07448651
## 11649  5636 FBgn0034731 -0.015560635 1.480967e-05 -0.07514938
## 11650  8719 FBgn0039937 -0.030295639 1.348605e-05 -0.14754326
## 11651  8700 FBgn0039900 -0.018599584 4.715846e-06 -0.09906957
```

```
# How many genes are there in this list?
dim(correl_lst)[1]
```

```
## [1] 11096
```

OrderedList analysis

Test for overlap of Positively age correlated genes and down regulated in E(z) genes, one sided:

```
library(OrderedList)
```

```
## Loading required package: twilight
```

```
## Loading required package: splines
```

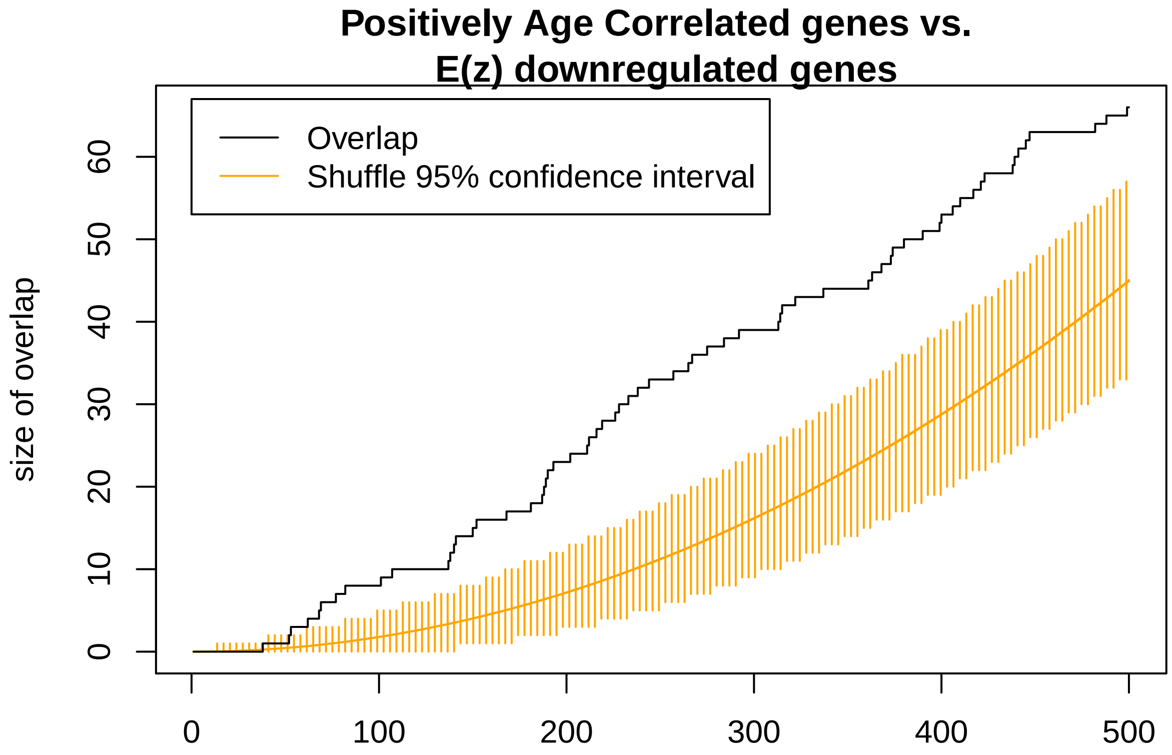
```
# Test for overlap of age correlated genes and down regulated in E(z) genes, one sided:
myComp <- compareLists(ID.List1=results_lst$ensembl,
                       ID.List2=correl_lst$ensembl,
                       mapping=NULL, two.sided = F)
```

```
##   Simulating random scores...
##   0%......:........:........:........:......100%
##   --------------------------------------------------
```

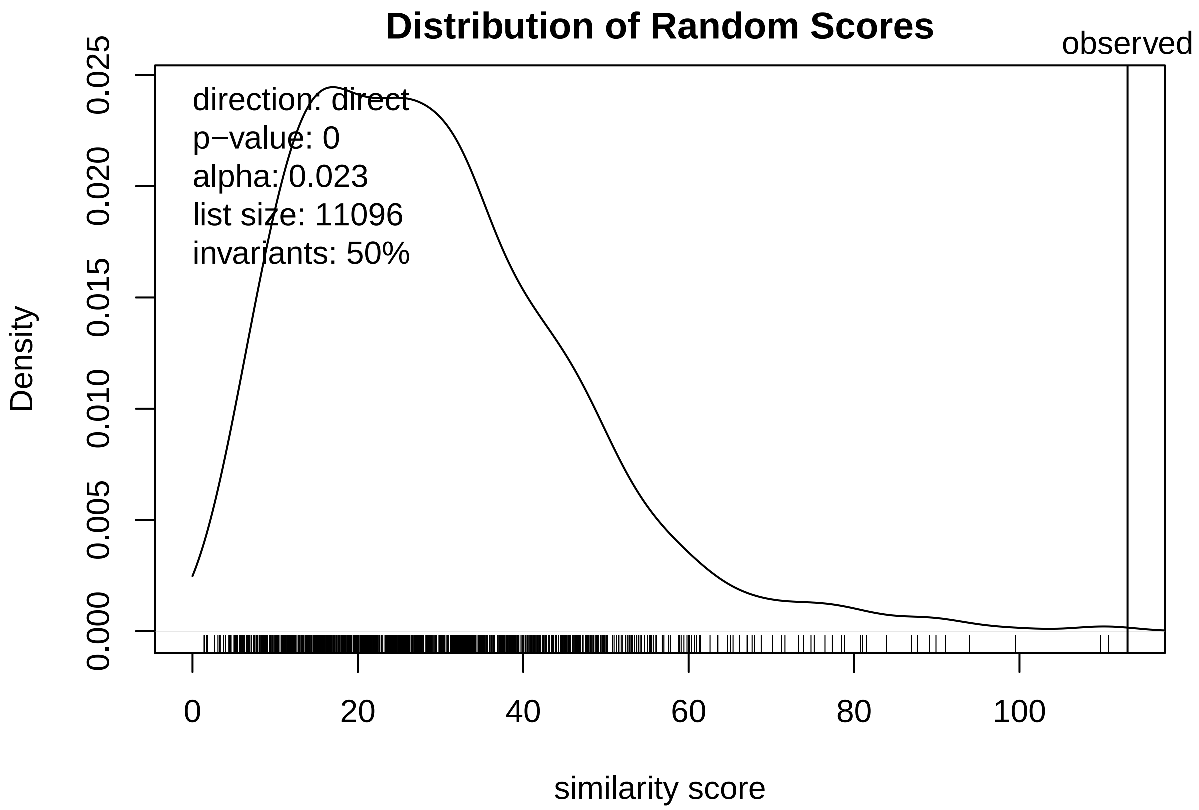
```
myComp
```

```
## List comparison
##   Assessing similarity of     : top ranks
##   Length of lists              : 11096
##   Quantile of invariant genes : 0.5
##   Number of random samples     : 1000
## --------------------------------------
##        Genes        Scores p.values   Rev.Scores Rev.p.values
## 0.115    100     0.1750498    0.175 1.000000e-05        0.837
## 0.077    150     1.5421404    0.117 6.755207e-03        0.860
## 0.058    200     5.4372546    0.080 7.045099e-02        0.874
## 0.038    300    23.9592517    0.017 8.581160e-01        0.886
## 0.029    400    59.2457191    0.004 3.450717e+00        0.910
## 0.023    500   113.0714173    0.000 8.752541e+00        0.925
## 0.015    750   336.3657095    0.000 3.921233e+01        0.971
## 0.012   1000   694.8006617    0.000 1.066578e+02        0.988
## 0.008   1500  1845.9043047    0.000 4.269078e+02        0.997
## 0.006   2000  3635.2856954    0.000 1.118397e+03        1.000
## 0.005   2500  6147.6877328    0.000 2.310002e+03        1.000
```

```
myOverlap <- getOverlap(myComp, max.rank=500, percent=.95)
#par(mfrow=c(1,2))
plot(myOverlap, no.title=TRUE, no.legend=TRUE,
     main = "Positively Age Correlated genes vs. \n E(z) downregulated genes")
legend(0,67,legend=c("Overlap", "Shuffle 95% confidence interval"),
       col=c("black", "orange"), lty=c(1,1))
```



```
plot(myOverlap, "scores")
```



```
myOverlap
```

```
## List comparison
##   Assessing similarity of                : top ranks
##   Length of lists                        : 11096
##   Number of random samples               : 1000
## -----------------------------------------------------------
##   Lists are more alike in direct order
##   Chosen regularization parameter        : alpha = 0.023 ( 500 genes)
##   Weighted overlap score                 : 113.0714
##   Significance of similarity             : p-value = 0
##   Score percentage for common entries    : 95
##   Entries contributing score percentage  : 33
```

```
# Make a publication plot:
##########################################################################
jpeg(file=paste(outpath, "Fig_8c.jpg", sep="/"),
     quality=100,
     res=300,
     width=1440,
     height=960)
par(cex.axis=1.25, cex.lab=2, mar=c(4,5,4,2), col.lab="white")
plot(myOverlap, no.title=TRUE, no.legend=TRUE, cex.lab=2,
     #xlab = NA,#Overlapping gene count",
     #ylab = NA,#Rank of genes in the two lists",
     main = "", las=1)
legend(0,67,legend=c("Overlap", "Shuffle 95% confidence interval"),
       col=c("black", "orange"), lty=c(1,1), cex=.75)
dev.off()
```

```
## pdf
##   2
```

```
###########################################################################
```

Test for overlap of age correlated genes and down regulated in E(z) genes, two sided:

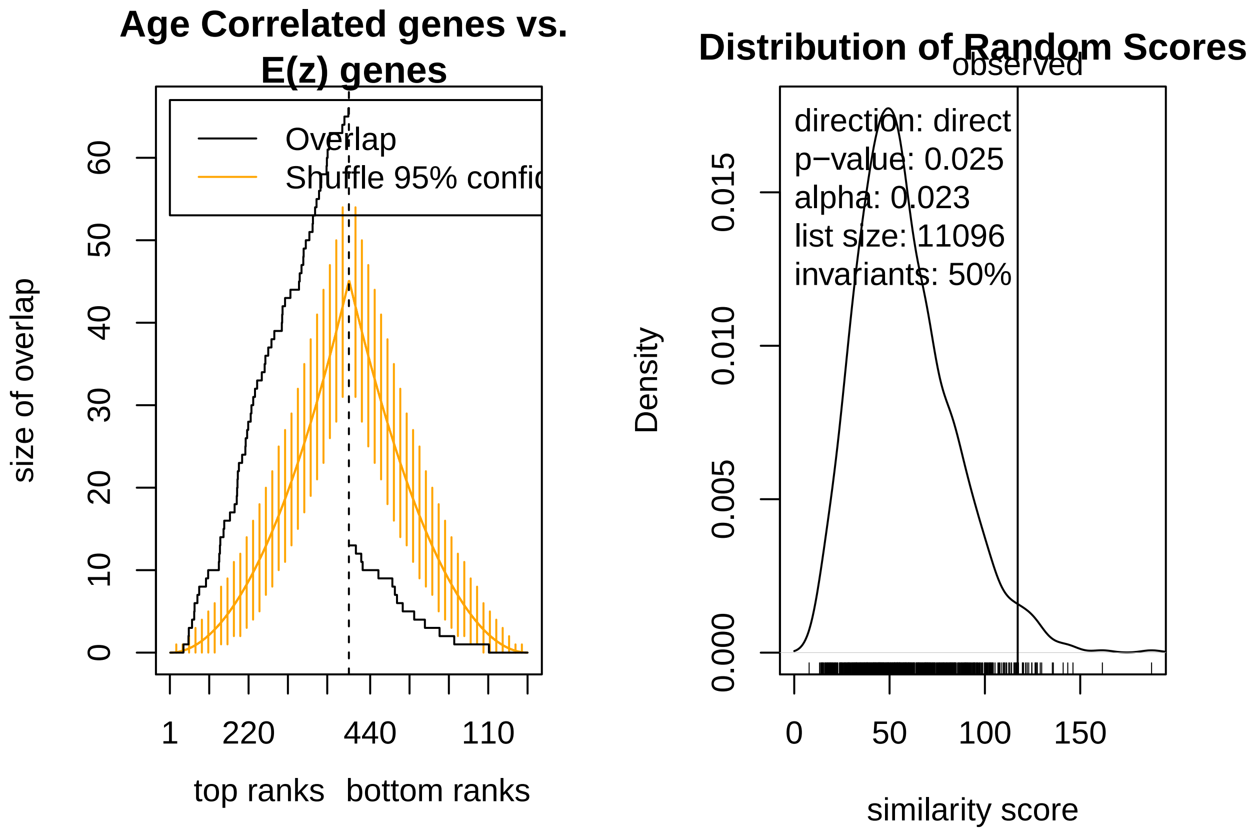
```
myComp2 <- compareLists(ID.List1=results_lst$ensembl,
                        ID.List2=correl_lst$ensembl,
                        mapping=NULL, two.sided = T)
```

```
##   Simulating random scores...
##   0%......:.........:.........:.........:......100%
##   --------------------------------------------------
```

```
myComp2
```

```
## List comparison
##   Assessing similarity of     : top and bottom ranks
##   Length of lists             : 11096
##   Quantile of invariant genes : 0.5
##   Number of random samples    : 1000
## ---------------------------------------
##       Genes       Scores p.values   Rev.Scores Rev.p.values
## 0.115   100    0.1750498    0.365 5.793481e-05        0.962
## 0.077   150    1.5449312    0.316 1.541563e-02        0.952
## 0.058   200    5.4688754    0.239 1.561663e-01        0.953
## 0.038   300   24.3598726    0.107 1.889960e+00        0.958
## 0.029   400   60.8640411    0.048 7.645187e+00        0.963
## 0.023   500  117.2033568    0.025 1.948915e+01        0.969
## 0.015   750  355.2388347    0.006 8.626226e+01        0.995
## 0.012  1000  746.3071778    0.005 2.257384e+02        1.000
## 0.008  1500 2048.4685764    0.020 8.266810e+02        1.000
## 0.006  2000 4165.2537622    0.146 2.026616e+03        1.000
## 0.005  2500 7269.9349495    0.549 4.011702e+03        1.000
```

```
myOverlap2 <- getOverlap(myComp2, max.rank=500, percent=.95)
par(mfrow=c(1,2))
plot(myOverlap2, no.title=TRUE, no.legend=TRUE,
     main = "Age Correlated genes vs. \n E(z) genes")
legend(0,67,legend=c("Overlap", "Shuffle 95% confidence interval"),
       col=c("black", "orange"), lty=c(1,1))
plot(myOverlap2, "scores")
```



```
myOverlap2
```

```
## List comparison
##   Assessing similarity of              : top and bottom ranks
##   Length of lists                      : 11096
##   Number of random samples             : 1000
## ----------------------------------------------------------
##   Lists are more alike in direct order
##   Chosen regularization parameter      : alpha = 0.023 ( 500 genes)
##   Weighted overlap score               : 117.2034
##   Significance of similarity           : p-value = 0.025
##   Score percentage for common entries  : 95
##   Entries contributing score percentage : 36
```

Test for overlap of Negatively age correlated genes and up regulated in E(z) genes, one sided:

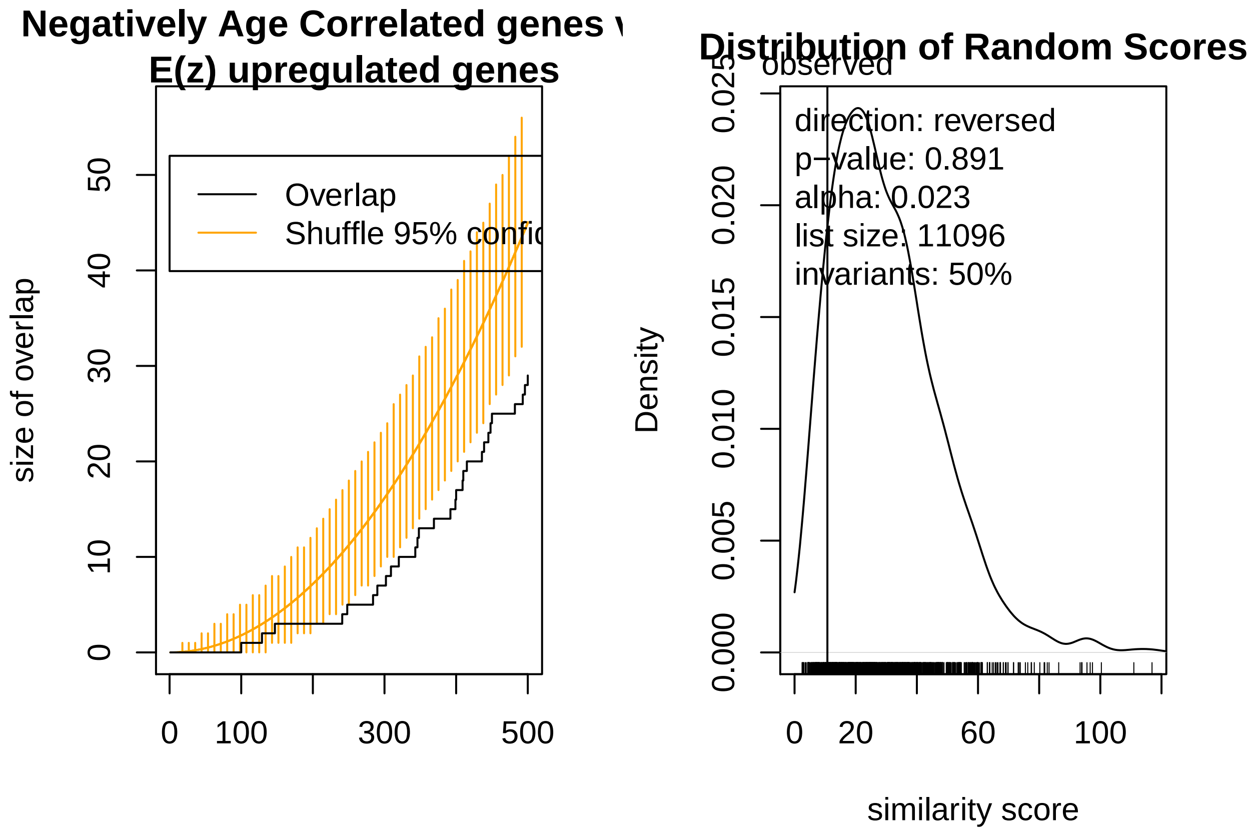
```
# Test for overlap of age correlated genes and down regulated in E(z) genes, one sided:
myComp3 <- compareLists(ID.List1=rev(results_lst$ensembl),
                        ID.List2=rev(correl_lst$ensembl),
                        mapping=NULL, two.sided = F)
```

```
##   Simulating random scores...
##   0%......:.........:.........:.........:......100%
##   --------------------------------------------------
```

```
myComp3
```

```
## List comparison
##   Assessing similarity of      : top ranks
##   Length of lists              : 11096
##   Quantile of invariant genes : 0.5
##   Number of random samples     : 1000
## --------------------------------------
##       Genes       Scores p.values   Rev.Scores Rev.p.values
## 0.115   100 0.000000e+00    0.846 4.793481e-05        0.821
## 0.077   150 2.790725e-03    0.914 8.660425e-03        0.838
## 0.058   200 3.162078e-02    0.930 8.571532e-02        0.846
## 0.038   300 4.006210e-01    0.949 1.031844e+00        0.857
## 0.029   400 1.618322e+00    0.972 4.194470e+00        0.870
## 0.023   500 4.131939e+00    0.991 1.073661e+01        0.891
## 0.015   750 1.887313e+01    1.000 4.704993e+01        0.926
## 0.012  1000 5.150652e+01    1.000 1.190806e+02        0.974
## 0.008  1500 2.025643e+02    1.000 3.997732e+02        0.999
## 0.006  2000 5.299681e+02    1.000 9.082185e+02        1.000
## 0.005  2500 1.122247e+03    1.000 1.701700e+03        1.000
```

```
myOverlap3 <- getOverlap(myComp3, max.rank=500, percent=.95)
par(mfrow=c(1,2))
plot(myOverlap3, no.title=TRUE, no.legend=TRUE,
     main = "Negatively Age Correlated genes vs. \n E(z) upregulated genes")
legend(0,52,legend=c("Overlap", "Shuffle 95% confidence interval"),
       col=c("black", "orange"), lty=c(1,1))
plot(myOverlap3, "scores")
```



```
myOverlap3
```

```
## List comparison
##   Assessing similarity of                : top ranks
##   Length of lists                        : 11096
##   Number of random samples               : 1000
## -------------------------------------------------------------
##   Lists are more alike in reversed order
##   Chosen regularization parameter        : alpha = 0.023 ( 500 genes)
##   Weighted overlap score                 : 10.73661
##   Significance of similarity             : p-value = 0.891
##   Score percentage for common entries    : 95
##   Entries contributing score percentage : 8
```

```
# Make a publication plot:
jpeg(file=paste(outpath, "Fig_S5f.jpg", sep="/"), width=360, height=240)
par(cex.axis=1.25, cex.lab=2, mar=c(4,5,4,2), col.lab="white")
plot(myOverlap3, no.title=TRUE, no.legend=TRUE, cex.lab=2,
     main = "", las=1)
legend(0,52,legend=c("Overlap", "Shuffle 95% confidence interval"),
       col=c("black", "orange"), lty=c(1,1), cex=.75)
dev.off()
```

```
## pdf
##   2
```

Test for overlap of age correlated genes and up regulated in E(z) genes, two sided:

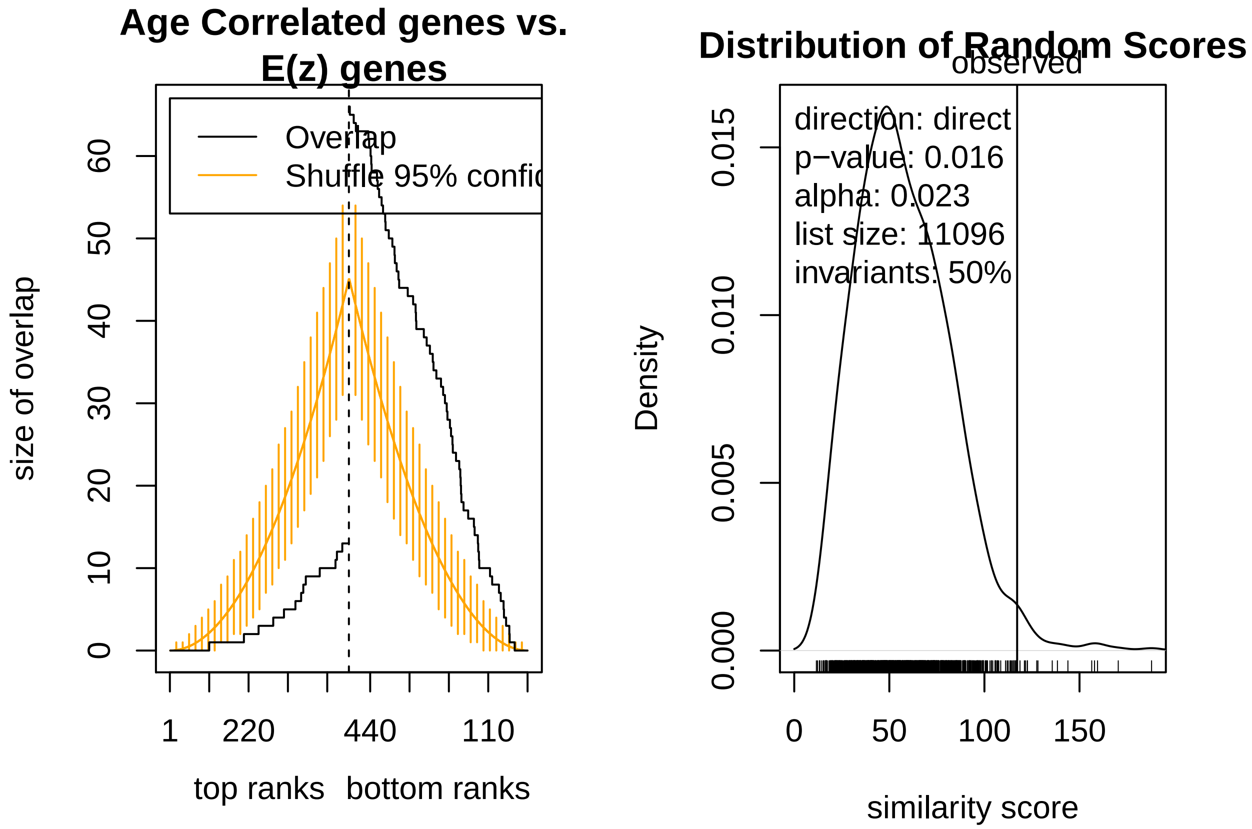
```
# Test for overlap of age correlated genes and down regulated in E(z) genes, one sided:
myComp4 <- compareLists(ID.List1=rev(results_lst$ensembl),
                        ID.List2=rev(correl_lst$ensembl),
                        mapping=NULL, two.sided = T)
```

```
##   Simulating random scores...
##   0%.......:........:........:........:......100%
##   -----------------------------------------------
```

```
myComp4
```

```
## List comparison
##   Assessing similarity of     : top and bottom ranks
##   Length of lists             : 11096
##   Quantile of invariant genes : 0.5
##   Number of random samples    : 1000
## --------------------------------------
##       Genes       Scores p.values   Rev.Scores Rev.p.values
## 0.115   100    0.1750498    0.361 5.793481e-05        0.962
## 0.077   150    1.5449312    0.305 1.541563e-02        0.954
## 0.058   200    5.4688754    0.240 1.561663e-01        0.948
## 0.038   300   24.3598726    0.099 1.889960e+00        0.950
## 0.029   400   60.8640411    0.037 7.645187e+00        0.959
## 0.023   500  117.2033568    0.016 1.948915e+01        0.975
## 0.015   750  355.2388347    0.006 8.626226e+01        0.993
## 0.012  1000  746.3071778    0.004 2.257384e+02        0.999
## 0.008  1500 2048.4685764    0.009 8.266810e+02        1.000
## 0.006  2000 4165.2537622    0.162 2.026616e+03        1.000
## 0.005  2500 7269.9349495    0.547 4.011702e+03        1.000
```

```
myOverlap4 <- getOverlap(myComp4, max.rank=500, percent=.95)
par(mfrow=c(1,2))
plot(myOverlap4, no.title=TRUE, no.legend=TRUE,
     main = "Age Correlated genes vs. \n E(z) genes")
legend(0,67,legend=c("Overlap", "Shuffle 95% confidence interval"),
       col=c("black", "orange"), lty=c(1,1))
plot(myOverlap4, "scores")
```



```
myOverlap4
```

```
## List comparison
##   Assessing similarity of              : top and bottom ranks
##   Length of lists                      : 11096
##   Number of random samples             : 1000
## ----------------------------------------------------------
##   Lists are more alike in direct order
##   Chosen regularization parameter      : alpha = 0.023 ( 500 genes)
##   Weighted overlap score               : 117.2034
##   Significance of similarity           : p-value = 0.016
##   Score percentage for common entries   : 95
##   Entries contributing score percentage : 36
```

Prepare a ranked list for publication: (Supplemental Data 2)

```
ranked_list <- correl_lst[,c(2,4)]
rownames(ranked_list) <- 1:dim(ranked_list)[1]
write.csv(file=file.path(outpath, "rankedAgingGenes.csv"), ranked_list)
```

Save image

```
save.image(file=file.path(outpath, "OrderedList.RData"))
```