GeneOverlap: An R package to test and visualize gene overlaps

Li Shen
Contact: li.shen@mssm.edu
or shenli.sam@gmail.com
Icahn School of Medicine at Mount Sinai
New York, New York
http://shenlab-sinai.github.io/shenlab-sinai/

November 20, 2013

Contents

1 Data preparation

The *GeneOverlap* package is composed of two classes: *GeneOverlap* and *GeneOverlapMatrix*. The *GeneOverlap* class serves as building blocks to the *GeneOverlapMatrix* class. First, let's load the package

```
> library(GeneOverlap)
```

To use the *GeneOverlap* class, create two character vectors that represent gene names. An easy way to do this is to use `read.table("filename.txt")` to read them from text files.

As a convenience, a couple of gene lists have been compiled into the `GeneOverlap` data. Use

```
> data(GeneOverlap)
```

to load them. Now, let's see what are contained in the data: there are three objects. The `hESC.ChIPSeq.list` and `hESC.RNASeq.list` objects contain gene lists from ChIP-seq and RNA-seq experiments. The `gs.RNASeq` variable contains the number of genes in the genomic background. Refer to Section 4 for details about how they were created. Let's see how many genes are there in the gene lists.

```
> sapply(hESC.ChIPSeq.list, length)
```

```
 H3K4me3  H3K9me3 H3K27me3 H3K36me3
   13302      293     3815     4485
```

```
> sapply(hESC.RNASeq.list, length)
```

```
  Exp High Exp Medium    Exp Low
      6444       5951       7647
```

```
> gs.RNASeq
```

```
[1] 20042
```

In *GeneOverlap*, we refer to a collection of gene lists as a gene set that is represented as a named list. Here we can see that the ChIP-seq gene set contains four gene lists of different histone marks: H3K4me3, H3K9me3, H3K27me3 and H3K36me3; the RNA-seq gene set contains three gene lists of different expression levels: High, Medium and Low. Two histone marks are associated with gene activation: H3K4me3 and H3K36me3 while the other two are associated with gene repression: H3K9me3 and H3K27me3.

2 Testing overlap between two gene lists

We want to explore whether the activation mark - H3K4me3 is associated with genes that are highly expressed. First, let's construct a `GeneOverlap` object

```
> go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K4me3,
+                          hESC.RNASeq.list$"Exp High",
+                          genome.size=gs.RNASeq)
> go.obj
```

```
GeneOverlap object:
listA size=13302
listB size=6444
Intersection size=5833
Overlap testing has not been performed yet.
```

As we can see, the *GeneOverlap* constructor has already done some basic statistics for us, such as the number of intersections. To test the statistical significance of association, we do

```
> go.obj <- testGeneOverlap(go.obj)
> go.obj
```

```
GeneOverlap object:
listA size=13302
listB size=6444
Intersection size=5833
Overlapping p-value=0e+00
Jaccard Index=0.4
```

The P-value is zero, which means the overlap is highly significant. To show some more details, use the `print` function

```
> print(go.obj)
```

```
Detailed information about this GeneOverlap object:
listA size=13302, e.g. DPM1 SCYL3 C1orf112 FGR FUCA2 GCLC
listB size=6444, e.g. ISG15 NOC2L KLHL17 AGRN B3GALT6 SDF4
Intersection size=5833, e.g. DPM1 C1orf112 FUCA2 SEMA3F ANKIB1 KRIT1
Union size=13913, e.g. DPM1 SCYL3 C1orf112 FGR FUCA2 GCLC
Genome size=20042
# Contingency Table:
     notA  inA
notB 6129 7469
inB   611 5833
Overlapping p-value=0e+00
Odds ratio=7.8
Overlap tested using Fisher's exact test (alternative=greater)
Jaccard Index=0.4
```

Further, we want to see if H3K4me3 is associated with genes that are lowly expressed. We do

```
> go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K4me3,
+                          hESC.RNASeq.list$"Exp Low",
+                          genome.size=gs.RNASeq)
> go.obj <- testGeneOverlap(go.obj)
> print(go.obj)

Detailed information about this GeneOverlap object:
listA size=13302, e.g. DPM1 SCYL3 C1orf112 FGR FUCA2 GCLC
listB size=7647, e.g. OR4F5 FAM138A OR4F29 PLEKHN1 C1orf170 RNF223
Intersection size=2531, e.g. FGR CFTR HS3ST1 WNT16 CYP26B1 CD38
Union size=18418, e.g. DPM1 SCYL3 C1orf112 FGR FUCA2 GCLC
Genome size=20042
# Contingency Table:
     notA   inA
notB 1624 10771
inB  5116  2531
Overlapping p-value=1
Odds ratio=0.1
Overlap tested using Fisher's exact test (alternative=greater)
Jaccard Index=0.1
```

In contrast to the highly expressed genes, the P-value is now 1 with odds ratio 0.1.

Once a GeneOverlap object is created, several accessors can be used to extract its slots. For example

```
> head(getIntersection(go.obj))

[1] "FGR"     "CFTR"    "HS3ST1"  "WNT16"   "CYP26B1" "CD38"

> getOddsRatio(go.obj)

[1] 0.07461166

> getContbl(go.obj)

     notA   inA
notB 1624 10771
inB  5116  2531

> getGenomeSize(go.obj)

[1] 20042
```

It is also possible to change slots "listA", "listB" and "genome.size" after object creation. For example

```
> setListA(go.obj) <- hESC.ChIPSeq.list$H3K27me3
> setListB(go.obj) <- hESC.RNASeq.list$"Exp Medium"
> go.obj

GeneOverlap object:
listA size=3815
listB size=5951
Intersection size=1529
Overlap testing has not been performed yet.
```

After any of the above slots is changed, the object is put into untested status. So we need to re-test it

```
> go.obj <- testGeneOverlap(go.obj)
> go.obj

GeneOverlap object:
listA size=3815
listB size=5951
Intersection size=1529
Overlapping p-value=6.6e-53
Jaccard Index=0.2
```

We can also change the genome size to see how the p-value changes with it

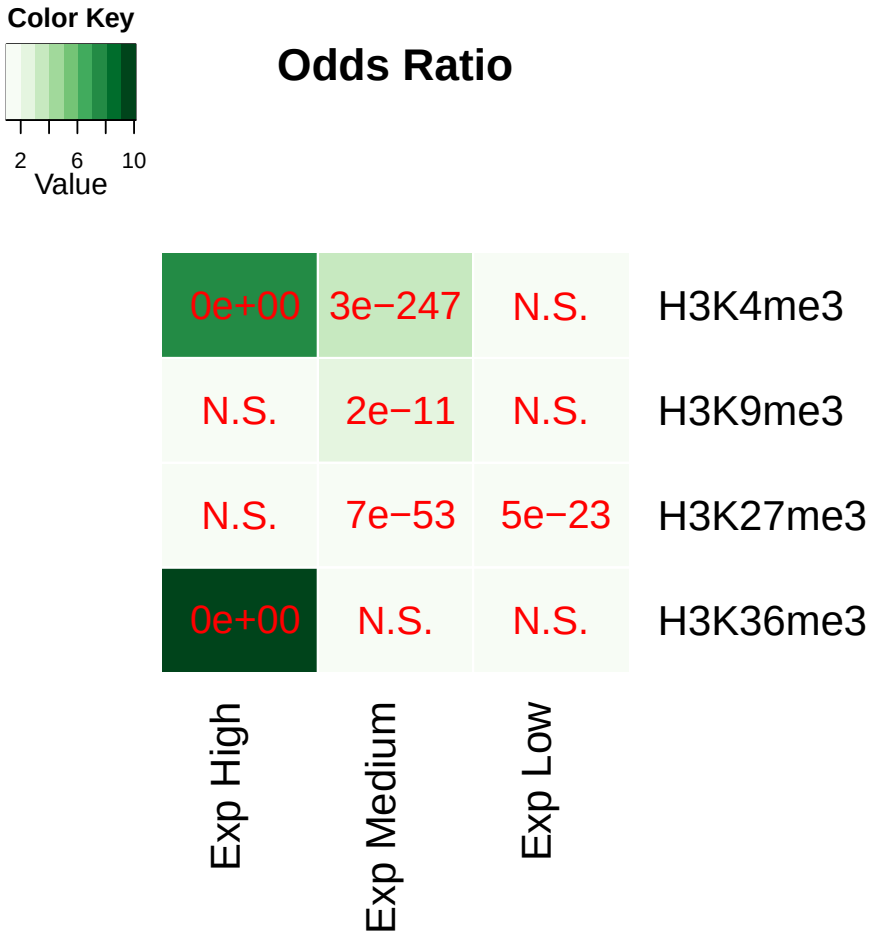
```
> v.gs <- c(12e3, 14e3, 16e3, 18e3, 20e3)
> setNames(sapply(v.gs, function(g) {
+     setGenomeSize(go.obj) <- g
+     go.obj <- testGeneOverlap(go.obj)
+     getPval(go.obj)
+ }), v.gs)

       12000        14000        16000        18000        20000
1.000000e+00 9.998297e-01 1.381083e-05 6.169626e-25 2.769996e-52
```

3 Visualizing all pairwise overlaps

When two gene sets each with one or more lists need to be compared, it would be rather inefficient to compare them manually. A matrix can be constructed, where the rows represent the lists from one set while the columns represent the lists from the other set. Each element of the matrix then is a GeneOverlap object. To visualize this overlapping information altogether, a heatmap can be used. To illustrate, we want to compare all the gene lists from ChIP-seq with that from RNA-seq

```
> gom.obj <- newGOM(hESC.ChIPSeq.list, hESC.RNASeq.list,
+                   gs.RNASeq)
> drawHeatmap(gom.obj)
```



That is neat. The `newGOM` constructor creates a new *GeneOverlapMatrix* object using two named lists. The colorkey represents the odds ratios and the significant p-values are superimposed on the grids.

To retrieve information from a *GeneOverlapMatrix* object, two important accessors are called `getMatrix` and `getNestedList`. The `getMatrix` accessor gets information such as p-values as a matrix, for example

```
> getMatrix(gom.obj, name="pval")
```

```
         Exp High    Exp Medium      Exp Low
H3K4me3  0.000000 3.367287e-247 1.000000e+00
H3K9me3  0.999903  1.868791e-11 9.994000e-01
H3K27me3 1.000000  6.584807e-53 4.901011e-23
H3K36me3 0.000000  1.000000e+00 1.000000e+00
```

or the odds ratios

```
> getMatrix(gom.obj, "odds.ratio")
```

```
           Exp High Exp Medium    Exp Low
H3K4me3   7.8338619  3.3375718 0.07461166
H3K9me3   0.6095471  2.2254522 0.66971459
H3K27me3  0.3045991  1.7855908 1.43235067
H3K36me3 10.1100840  0.7691473 0.02243317
```

The `getNestedList` accessor can get gene lists for each comparison as a nested list: the outer list represents the columns and the inner list represents the rows

```
> inter.nl <- getNestedList(gom.obj, name="intersection")
> str(inter.nl)

List of 3
 $ Exp High  :List of 4
  ..$ H3K4me3 : chr [1:5833] "DPM1" "C1orf112" "FUCA2" "SEMA3F" ...
  ..$ H3K9me3 : chr [1:66] "ZNF195" "SEC63" "RNASET2" "TSSC1" ...
  ..$ H3K27me3: chr [1:563] "C1orf112" "SLC25A13" "CCDC109B" "ITGA3" ...
  ..$ H3K36me3: chr [1:3245] "DPM1" "C1orf112" "NFYA" "SEMA3F" ...
 $ Exp Medium:List of 4
  ..$ H3K4me3 : chr [1:4938] "SCYL3" "GCLC" "C1orf201" "NIPAL3" ...
  ..$ H3K9me3 : chr [1:141] "ZNF263" "ZNF200" "ERP44" "PRKCH" ...
  ..$ H3K27me3: chr [1:1529] "TMEM176A" "SLC7A2" "SARM1" "PLXND1" ...
  ..$ H3K36me3: chr [1:1147] "SCYL3" "GCLC" "CYP51A1" "ALS2" ...
 $ Exp Low   :List of 4
  ..$ H3K4me3 : chr [1:2531] "FGR" "CFTR" "HS3ST1" "WNT16" ...
  ..$ H3K9me3 : chr [1:86] "DLEC1" "ZPBP" "TRHDE" "NTN4" ...
  ..$ H3K27me3: chr [1:1723] "FGR" "HS3ST1" "WNT16" "CYP26B1" ...
  ..$ H3K36me3: chr [1:93] "CBLN4" "ZNF285" "PKD2L2" "C20orf26" ...
```

Another important accessor is the method `"["` that allows one to retrieve *GeneOverlap* objects in a matrix-like fashion. For example,

```
> go.k4.high <- gom.obj[1, 1]
> go.k4.high

GeneOverlap object:
listA size=13302
listB size=6444
Intersection size=5833
Overlapping p-value=0e+00
Jaccard Index=0.4
```

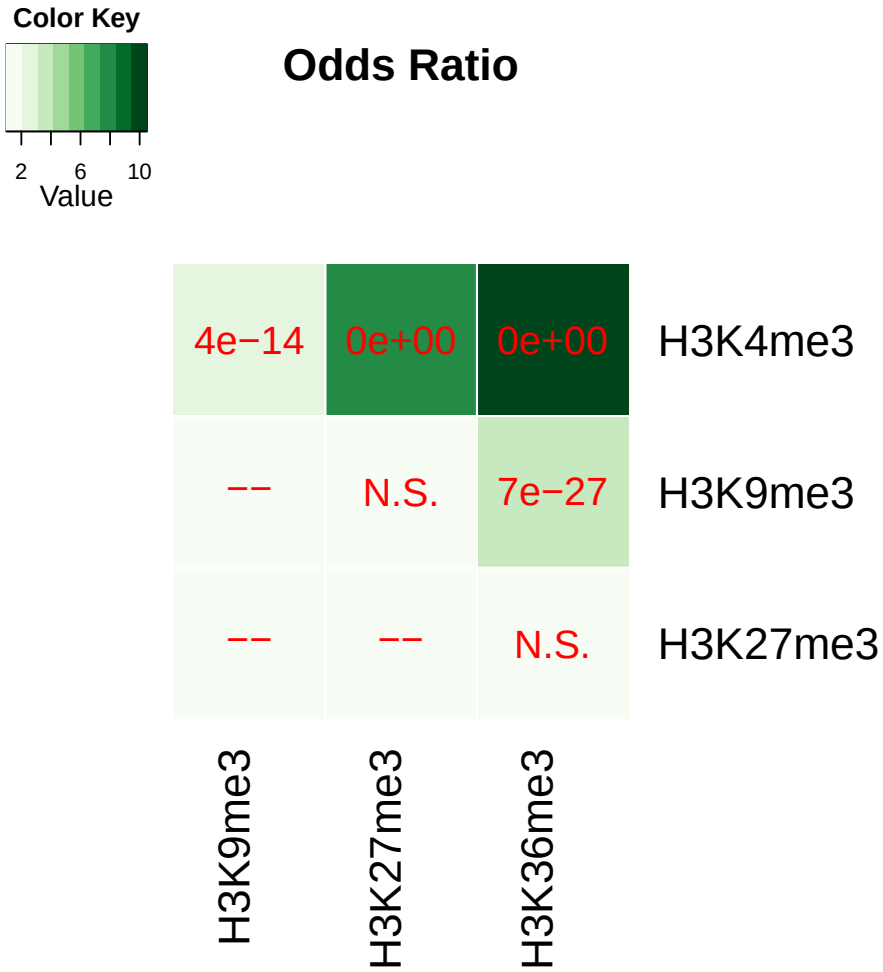
gets the *GeneOverlap* object that represents the comparison between H3K4me3 and highly expressed genes. It is also possible to get *GeneOverlap* objects using labels, such as

```
> gom.obj["H3K9me3", "Exp Medium"]

GeneOverlap object:
listA size=293
listB size=5951
Intersection size=141
Overlapping p-value=1.9e-11
Jaccard Index=0.0
```

GeneOverlapMatrix can also perform self-comparison on one gene set. For example, if we want to know how the ChIP-seq gene lists associate with each other, we can do

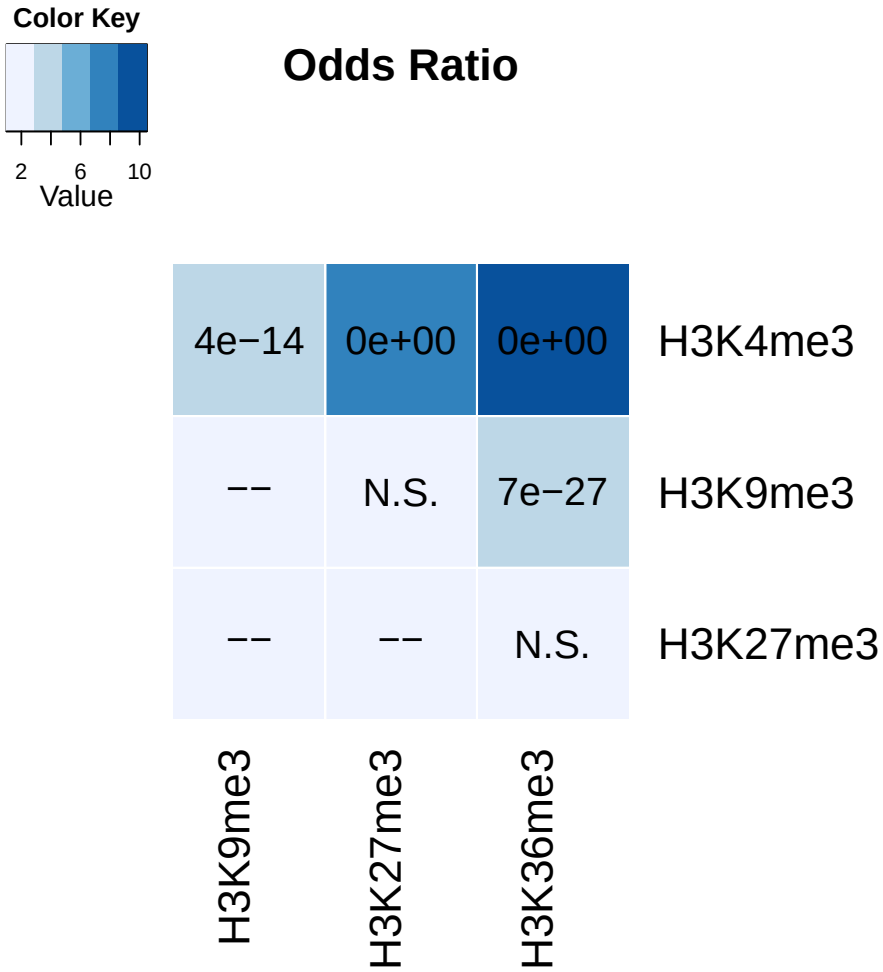
```
> gom.self <- newGOM(hESC.ChIPSeq.list,
+                    genome.size=gs.RNASeq)
> drawHeatmap(gom.self)
```



Only the upper triangular matrix is used.

It is also possible to change the number of colors and the colors used for the heatmap. For example

```
> drawHeatmap(gom.self, ncolused=5, grid.col="Blues", note.col="black")
```



4 Data source and processing

The experimental data used here were downloaded from the ENCODE [ENCODE Consortium,] project's website. Both ChIP-seq and RNA-seq samples were derived from the human embryonic stem cells (hESC). The raw read files were aligned to the reference genome using Bowtie [Langmead et al., 2009] and Tophat [Trapnell et al., 2009].

Cufflinks [Trapnell et al., 2010] was used to estimate the gene expression levels from the RNA-seq data. The entries with duplicated gene names were removed all together to avoid ambiguity. Only protein coding genes were retained for further analysis. The genes were further filtered by FPKM status and only the genes with "OK" status were kept. This left us with 20,042 coding genes whose FPKM values were reliabled estimated. The genes were then separated into three groups: high (FPKM>10), medium (FPKM>1 and <=10) and low (FPKM<=1).

For ChIP-seq, one replicate from IP treatment and one replicate from input control were used. Peak calling was performed using MACS2 (v2.0.10) with default parameter values. The peak lists then went through region annotation using the diffReps package [Shen et al., 2013]. After that, the genes that had peaks on genebody and promoter regions were extracted. The genes were further filtered using the RNA-seq gene list obtained above.

5 SessionInfo

```
> sessionInfo()
```

```
R version 3.0.2 (2013-09-25)
Platform: x86_64-pc-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] GeneOverlap_0.99.1

loaded via a namespace (and not attached):
[1] BiocStyle_1.0.0    bitops_1.0-6       caTools_1.16       gdata_2.13.2
[5] gplots_2.12.1      gtools_3.1.0       KernSmooth_2.23-10 RColorBrewer_1.0-5
[9] tools_3.0.2
```

References

[ENCODE Consortium,] ENCODE Consortium. The encyclopedia of dna elements (http://encodeproject.org/encode/).

[Langmead et al., 2009] Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, 10(3):R25.

[Shen et al., 2013] Shen, L., Shao, N.-Y., Liu, X., Maze, I., Feng, J., and Nestler, E. J. (2013). diffreps: Detecting differential chromatin modification sites from chip-seq data with biological replicates. *PLoS ONE*, 8(6):e65598.

[Trapnell et al., 2009] Trapnell, C., Pachter, L., and Salzberg, S. L. (2009). Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111.

[Trapnell et al., 2010] Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (2010). Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotech*, 28(5):511–515. 10.1038/nbt.1621.