RNA-seq Analysis 4:

Introduction to Gene Set and Pathway Analysis

Pietà Schofield

Contents

Functional Annotation

As with most things there are multiple ways and multiple packages available in Bioconductor for doing functional annotation analysis of set of genes identified by an NGS experiment. Many of these are listed under the various BiocViews for example:

- Pathways BiocView, and again the choice of which to choose is a mix of personal preference and specific application.
- Gene Set Enrichment BiocView
- Network Enrichment BiocView
- Systems Biology BiocView

This final workshop introduces a couple of Bioconductor packages that can be used for visualisation and exploration of functional annotations of sets of enriched genes. These are

- STRINGdb package (Franceschini, Szklarczyk, Frankild, et al., 2013)
- pathview package (Luo and Brouwer, 2013)

Protein Interactions

In particular these packages give access to the STRING Known and predicted protein interaction database, Gene Ontology Database and the KEGG Kyoto Encyclopedia of Genes and Genomes Database,

```
require(STRINGdb)

species <- get_STRING_species(version="10", species_name=NULL)
species[which(grepl("^Homo sapiens",species$official_name)),]
```

```
    species_id official_name compact_name   kingdom type
103       9606  Homo sapiens Homo sapiens eukaryota core
```

```
# create a new STRING_db object
string_db <- STRINGdb$new(version="10",species=9606)
```

I constant issue with bioinformatics is the issue of different databases and annotations having different identifiers for the same thing. This is no different with STRINGdb. So the first task is to map the gene identifiers in the data set to genes that STRINGdb knows about

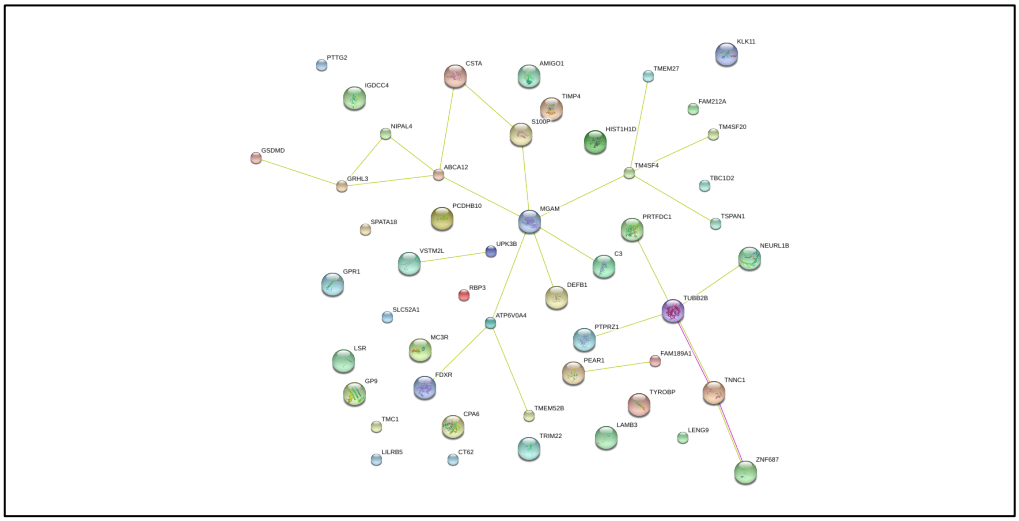
pvalue	logFC	gene
0.0001018	3.333461	VSTM2L
0.0001392	3.822383	TBC1D2
0.0001720	3.306055	LENG9
0.0001739	3.024605	TMEM27
0.0001990	3.854414	LOC100506014
0.0002393	3.082052	TSPAN1

The gene identifier column is called gene. Using the `string_db$map()` function to map the gene ids

```
# map to STRING ids
example1_mapped = string_db$map( diff_exp_example1[1:1000,], "gene", removeUnmappedRows = TRUE )
```

```
Warning:  we couldn't map to STRING 21% of your identifiers
```

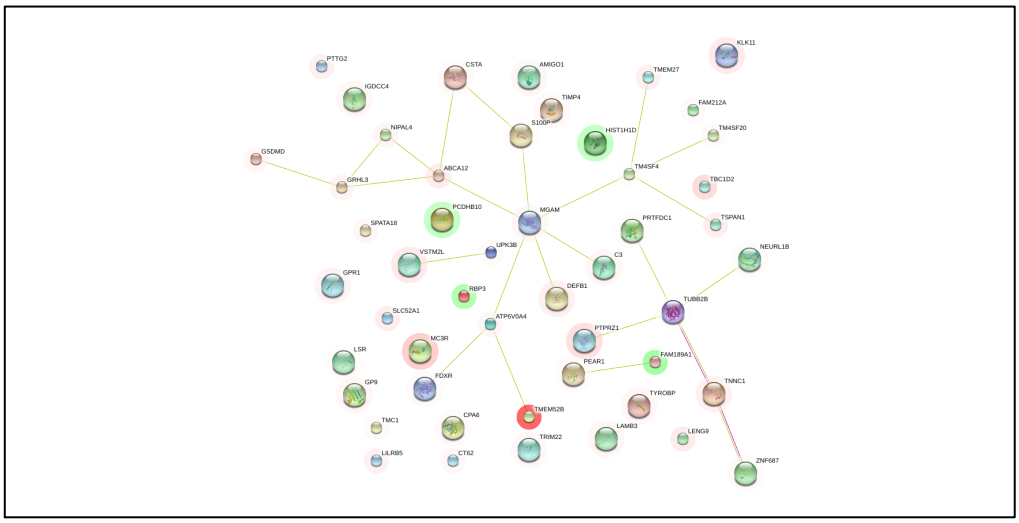
```
# get the STRING_id for the top 50 genes
hits = example1_mapped$STRING_id[1:50]

# plot the STRING network png
string_db$plot_network( hits )
```



It is possible to add halos to the plot that signify the level of differential expression.

```
# filter by p-value and add a color column
# (i.e. green down-regulated gened and red for up-regulated genes)
example1_mapped_pval05 = string_db$add_diff_exp_color( subset(example1_mapped, pvalue<0.05),
                                                       logFcColStr="logFC" )

# post payload information to the STRING server
payload_id = string_db$post_payload( example1_mapped_pval05$STRING_id,
                                     colors=example1_mapped_pval05$color )

# display a STRING network png with the "halo"
string_db$plot_network( hits, payload_id=payload_id )
```

proteins: 50
interactions: 24
expected interactions: 16 (p−value: 0.0561922764527946)



http://string−db.org/10/p/1095184754

Geno Ontology

STRINGdb also provides a simple interface to the Gene Ontolgy database and can perform GO term enrichment analysis. The `string_db$get_enrichment()` function returns an ordered list of GO terms that are over represented in the gene hits list provided. It has parameters that provide for the selection of specific GO categories, Biological Process, Molecular Function and Celular Compartment. It also has options for specifying multiple test correction methods and the evidence type for the GO annotations.

```
hits = example1_mapped$STRING_id
######## compute enrichment in GO annotations for Biological Process ########
enrichmentGO = string_db$get_enrichment( hits, category = "Process", methodMT = "fdr", iea = TRUE )
head(enrichmentGO, n=7)
```

```
     term_id proteins hits       pvalue   pvalue_fdr
1 GO:0006952     1286   90 1.311617e-13 6.651208e-10
2 GO:0043207      627   49 2.157102e-09 3.646222e-06
3 GO:0051707      627   49 2.157102e-09 3.646222e-06
```

```
4 GO:0009607      654   49 8.516805e-09 9.732744e-06
5 GO:0016477      698   51 9.596474e-09 9.732744e-06
6 GO:1903034      351   33 1.281763e-08 1.083303e-05
7 GO:0006955     1203   73 1.782235e-08 1.291102e-05
                     term_description
1                     defense response
2 response to external biotic stimulus
3           response to other organism
4          response to biotic stimulus
5                        cell migration
6   regulation of response to wounding
7                       immune response
```

KEGG Pathways

The `string_db$get_enrichment()` can also be used to get KEGG Pathway enrichment values

```
enrichmentKEGG = string_db$get_enrichment( hits, category = "KEGG", methodMT = "fdr", iea = TRUE )
head(enrichmentKEGG, n=7)
```

```
  term_id proteins hits       pvalue   pvalue_fdr
1   01100     1161   54 1.387908e-16 3.317100e-14
2   04610       68   14 1.755247e-13 2.097520e-11
3   04115       66   13 2.392312e-12 1.905876e-10
4   04060      260   17 5.497861e-08 3.284972e-06
5   00590       61    9 8.638487e-08 4.129197e-06
6   05202      164   12 1.594285e-06 6.350569e-05
7   05133       69    8 3.013613e-06 1.022468e-04
                        term_description
1                     Metabolic pathways
2     Complement and coagulation cascades
3                   p53 signaling pathway
4  Cytokine-cytokine receptor interaction
5             Arachidonic acid metabolism
6 Transcriptional misregulation in cancer
7                              Pertussis
```

Visualising Pathways

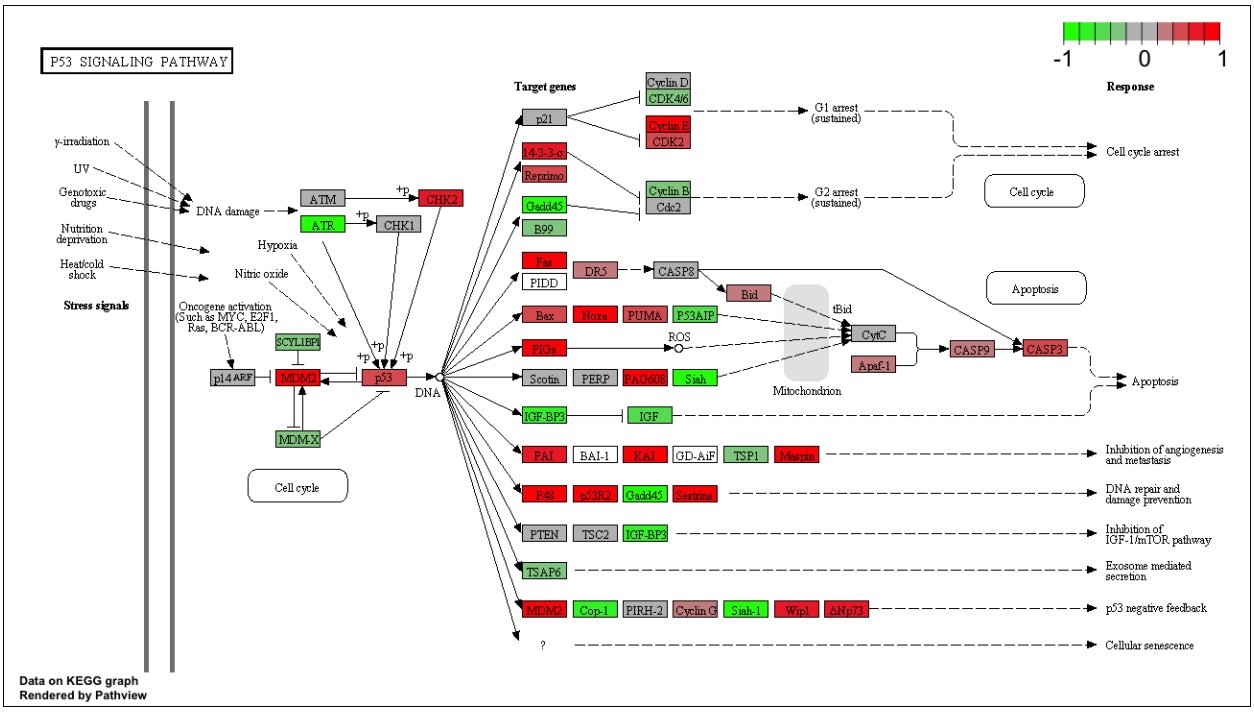
The package pathview can be used to visualise KEGG Pathways and will annotate the pathway diagram with fold change or significance colouring. So for example the enriched pathway for p53 in the above example can be visualise as follows.

```
# Load the pathview package
require(pathview)

# create a named vector of fold changes from the expression data using the gene Ids as names
gene.data <- as.numeric(diff_exp_example1$logFC)
names(gene.data) <- diff_exp_example1$gene

# lets look at one of the top path in the significant KEGG pathway list above for example
# the p53 pathway which has KEGG id 04115, homo sapiens has the KEGG id hsa
```

```
# the gene ids that have been used in this case are the gene symbols

pathID <- "04115"
pview <- pathview(gene.data=gene.data,
                  gene.idtype="SYMBOL",
                  pathway.id=pathID,
                  species="hsa",
                  out.suffix="kegg",
                  kegg.native=T,
                  same.layer=T)
```

```
[1] "Note: 1673 of 20861 unique input IDs unmapped."
```

The `pathview()` function creates a PNG graphics file in the local directory.



Exercise 4

Using the expression data in the RData file /usr/local/share/BS32010/pschofield/Rdata/dge_scerevisiae.RData

- use STRINGdb to visualise the relationshops of the top 25 differentially expressed genes.

- find the top 6 KEGG pathways
- use pathview to visualise the differential expression of the genes in the glycine, serine and threonine metabolism pathway.

Bibliography

[1] A. Franceschini, D. Szklarczyk, S. Frankild, et al. “STRING v9.1: protein-protein interaction networks, with

increased coverage and integration." In: Nucleic acids research 41 (Database issue Jan. 2013), pp. D808–815. ISSN: 1362-4962 0305-1048. DOI: 10.1093/nar/gks1094. pmid: pmid.

[2] W. Luo and C. Brouwer. "Pathview: an R/Bioconductor package for pathway-based data integration and visualization". In: Bioinformatics (Jun. 04, 2013). DOI: 10.1093/bioinformatics/btt285. URL: http://bioinformatics.oxfordjournals.org/content/early/2013/06/11/bioinformatics.btt285.abstract.

Session Info

```
R version 3.2.3 (2015-12-10)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.11.3 (El Capitan)

locale:
[1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8

attached base packages:
 [1] parallel  stats4    tcltk     stats     graphics  grDevices utils
 [8] datasets  methods   base

other attached packages:
 [1] pathview_1.10.1      org.Hs.eg.db_3.2.3   AnnotationDbi_1.32.3
 [4] IRanges_2.4.8        S4Vectors_0.8.11     Biobase_2.30.0
 [7] BiocGenerics_0.16.1  STRINGdb_1.10.0      hash_2.2.6
[10] gplots_2.17.0        RColorBrewer_1.1-2   plotrix_3.6-1
[13] RCurl_1.95-4.7       bitops_1.0-6         igraph_1.0.1
[16] plyr_1.8.3           sqldf_0.4-10         RSQLite_1.0.0
[19] DBI_0.3.1            gsubfn_0.6-6         proto_0.3-10
[22] png_0.1-7            knitr_1.12.3         RefManageR_0.10.6
[25] pietalib_0.1

loaded via a namespace (and not attached):
 [1] KEGGREST_1.10.1      gtools_3.5.0         htmltools_0.3
 [4] yaml_2.1.13          chron_2.3-47         XML_3.98-1.3
 [7] Rgraphviz_2.14.0     stringr_1.0.0        zlibbioc_1.16.0
[10] Biostrings_2.38.4    caTools_1.17.1       evaluate_0.8
[13] GenomeInfoDb_1.6.3   highr_0.5.1          Rcpp_0.12.3
[16] KernSmooth_2.23-15   formatR_1.2.1        gdata_2.17.0
[19] graph_1.48.0         XVector_0.10.0       digest_0.6.9
[22] stringi_1.0-1        RJSONIO_1.3-0        GenomicRanges_1.22.4
[25] grid_3.2.3           bibtex_0.4.0         tools_3.2.3
[28] magrittr_1.5         KEGGgraph_1.28.0     lubridate_1.5.0
[31] rmarkdown_0.9.5      httr_1.1.0           R6_2.1.2
```