Downstream analysis for transcriptomic data

What to do (or not) with gene lists

A/Prof Mik Black
Department of Biochemistry
University of Otago

BioHPC group: 4 April 2018
https://github.com/mikblack/BioHPC_geneLists

Gene lists

- There are **LOTS** of situations where we might want to use a list of genes to shed some biological light on our problem
- Two typical situations:
  - List of genes (e.g., differentially expressed between two conditions) that we want to gain some more insight (e.g., by including some additional biological annotation)
  - List of gene representing a specific biological function (e.g., tissue-specific expression) that we would like to obtain an expression summary for.

Common types of annotation

- Pathways
  - Reactome
  - KEGG
  - WikiPathways
  - Biocarta
- Ontologies
  - Gene Ontology
  - Human phenotypes
  - Tissue-specific expression

Situation 1

- Over-representation analysis
  - Enrichr
  - GATHER
  - GeneSetDB
- Enrichment analysis
  - GSEA
  - Generic methods (limma): `roast`, `camera`, `wilcoxGST`

Situation 2: Metagenes!

- Dimension reduction to produce summary of multi-gene data
  - Centroids
  - Principal Components Analysis (PCA)
  - Singular Value Decomposition (SVD)
  - Mutlidimensional Scaling (MDS)
- Relationship to clinical/phenotypic variables

Example data set

- Breast cancer data: Miller et al. (2005): `breastCancerUPP` package
  - 251 tumour samples on HGU133A/B Affymetrix microarrays
  - 44,928 probe sets
  - 10 clinical variables
- Use these data to illustrate both situations described above (over-representation/enrichment and metagenes).

Setup

```
## Load packages
library(breastCancerUPP)
library(ggplot2)
library(survival)
library(limma)
library(reactome.db)
library(ReactomePA)
library(survminer)
library(gplots)
library(dplyr)
library(WGCNA)
library(gridExtra)


## Helper function for ggplot colours
gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
}
```

Load breast cancer expression data

```
data(upp)
expDat = exprs(upp)
upp
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 44928 features, 251 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: UPP_103B41 UPP_104B91 ... UPP_9B52 (251 total)
##   varLabels: samplename dataset ... e.os (21 total)
##   varMetadata: labelDescription
## featureData
##   featureNames: 1007_s_at 1053_at ... AFFX-TrpnX-M_at_2 (44928
##     total)
##   fvarLabels: probe Gene.title ... GO.Component.1 (22 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
##   pubMedIds: 16141321
## Annotation: hgu133ab
```

Extract gene symbols

```
symbols = fData(upp)$Gene.symbol
names(symbols) = fData(upp) %>%  rownames()
head(symbols)
```

```
## 1007_s_at   1053_at    117_at    121_at 1255_g_at   1294_at
##    "DDR1"    "RFC2"   "HSPA6"    "PAX8"  "GUCA1A"    "UBA7"
```

```
symbols %>%  is.na() %>% sum()
```

```
## [1] 8734
```

```
drop_na = symbols %>%  is.na() %>% which()
head(drop_na, 4)
```

```
##   201265_at   202225_at   202603_at 203326_x_at
##         793        1753        2131        2853
```

Collapse probe-sets to symbols

- `collapseRows` function in `WGCNA` package was used to generate per-gene data (e.g., single value when gene represented by multiple probe sets)
- Default is to select the probe set with teh highest mean expression value.

```
uppExpr = WGCNA::collapseRows(expDat[-drop_na,],
                  rowGroup = symbols[-drop_na],
                  rowID = names(symbols[-drop_na]))$datETcollapsed
saveRDS(uppExpr, file='uppsala-brca-perGene.rds')
```

Examine collapsed data

```
uppExpr = readRDS('uppsala-brca-perGene.rds')
dim(uppExpr)
```

```
## [1] 18821   251
```

```
head(uppExpr[,1:5])
```

```
##        UPP_103B41 UPP_104B91 UPP_112B55 UPP_114B68 UPP_130B92
## A1BG        6.750      7.356      7.631      7.189      7.333
## A1CF        6.866      6.711      7.036      7.060      6.857
## A2BP1       7.187      7.232      6.627      6.559      7.268
## A2LD1       7.245      6.994      6.751      6.484      6.918
## A2M         9.540      9.378      9.885      8.914      9.425
## A4GALT      5.673      5.520      5.015      5.622      5.536
```

Extract clinical data

```
uppClin <- pData(upp)
dim(uppClin)
```

```
## [1] 251  21
```

```
names(uppClin)
```

```
##  [1] "samplename"    "dataset"       "series"        "id"
##  [5] "filename"      "size"          "age"           "er"
##  [9] "grade"         "pgr"           "her2"          "brca.mutation"
## [13] "e.dmfs"        "t.dmfs"        "node"          "t.rfs"
## [17] "e.rfs"         "treatment"     "tissue"        "t.os"
## [21] "e.os"
```

Extract relevant variables

```
uppClin %>% is.na() %>%  colSums()
```

```
##    samplename       dataset        series            id      filename
##             0             0             0             0           251
##          size           age            er         grade           pgr
##             0             0             4             2             0
##          her2 brca.mutation       e.dmfs       t.dmfs          node
##           251           251           251           251             9
##         t.rfs         e.rfs     treatment        tissue          t.os
##            17            15            29             0           251
##          e.os
##           251
```

```
uppClin = uppClin[,c("size", "age", "er", "grade", "pgr", "node",
                     "t.rfs", "e.rfs", "treatment")]
```

Tidy up Grade and RFS time

```
## Convert RFS time from days to years
uppClin$t.rfs = uppClin$t.rfs/365

## Convert grade from numeric (1,2,3) to G1, G2, G3.
uppClin$grade = paste0("G", uppClin$grade)
uppClin$grade[uppClin$grade=="GNA"] = NA

## Table of Grade vs ER status
table(uppClin$er, uppClin$grade, useNA='always', dnn=c("ER", "Grade"))
```

```
##       Grade
## ER      G1  G2  G3 <NA>
##   0      2  11  21    0
##   1     62 116  33    2
##   <NA>   3   1   0    0
```

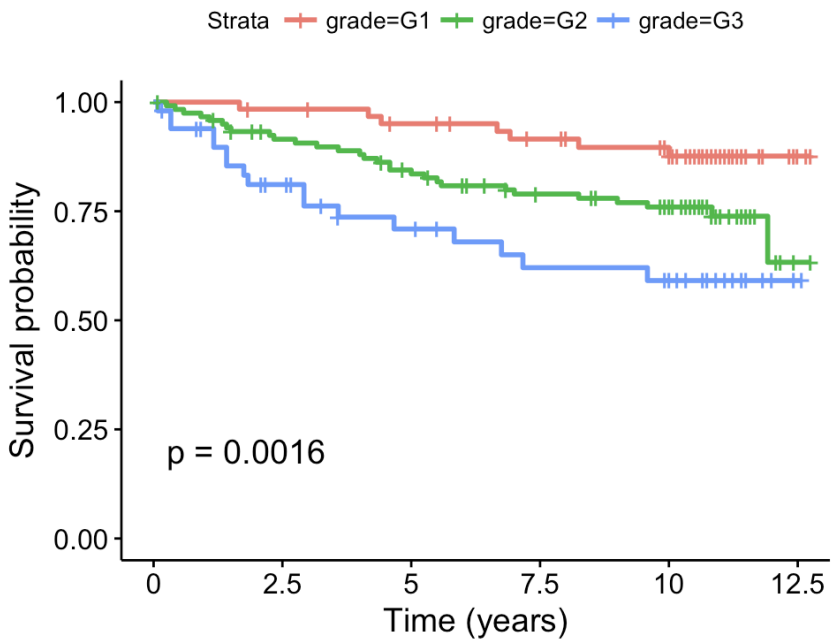
Clinical data

```
View(uppClin)
```

	size	age	er	grade	pgr	node	t.rfs	e.rfs	treatment	t.rf
UPP_103B41	2.200	52	0	G3	0	1	*NA*	*NA*	2	*NA*
UPP_104B91	2.400	80	1	G3	1	*NA*	3.58356164	0	2	3.58356164
UPP_112B55	2.400	61	1	G2	1	1	1.50136986	1	2	1.50136986
UPP_114B68	1.200	67	1	G1	1	1	11.16712329	0	2	11.16712329
UPP_130B92	1.800	73	1	G2	1	1	5.58356164	1	2	5.58356164
UPP_131B79	2.600	59	1	G3	1	1	5.83287671	1	2	5.83287671
UPP_135B40	2.600	64	1	G1	1	0	11.50136986	0	2	11.50136986
UPP_138B34	2.300	65	1	G1	1	1	11.50136986	0	2	11.50136986
UPP_139B03	2.800	84	1	G3	0	0	0.33424658	1	2	0.33424658
UPP_147B19	2.400	71	0	G3	0	1	5.08219178	0	2	5.08219178
UPP_148B49	2.600	71	1	*NA*	1	1	2.91780822	1	2	2.91780822
UPP_14B98	2.000	61	1	G2	1	1	10.83287671	0	2	10.83287671
UPP_150B81	3.500	55	1	G2	1	1	11.41643836	0	2	11.41643836
UPP_154B42	0.900	52	1	G2	1	1	4.58356164	1	2	4.58356164
UPP_159B47	1.400	57	1	G2	1	1	11.41643836	0	2	11.41643836
UPP_15C94	5.000	83	1	G2	1	0	4.41643836	0	2	4.41643836

Survival plot (RFS by Grade)

```
ggsurvplot(survfit(Surv(t.rfs, e.rfs) ~ grade, data=uppClin),
           pval = TRUE, risk.table = FALSE) + xlab("Time (years)")
```



Differential expression (G1 vs G3)

```
## Only keep samples which do not have NA values for grade
keep = which(!is.na(uppClin$grade))

## Create design marix for linear model analysis
design = model.matrix(~uppClin$grade)
head(design)
```

```
##   (Intercept) uppClin$gradeG2 uppClin$gradeG3
## 1           1               0               1
## 2           1               0               1
## 3           1               1               0
## 4           1               0               0
## 5           1               1               0
## 6           1               0               1
```

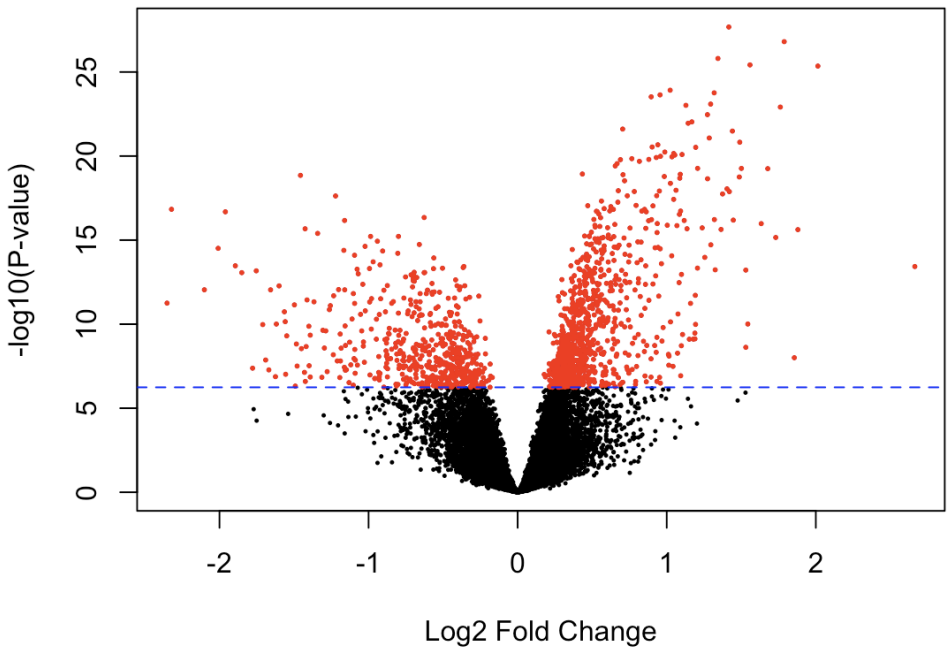
Differential expression (G1 vs G3)

```
## Fit linear model
fit = lmFit(uppExpr[, keep], design)
fit = eBayes(fit)

## Extract results
g3_vs_g1 = topTable(fit, coef=3, n=nrow(uppExpr),
                    adjust.method="holm")
## Identify significant genes
sig = g3_vs_g1$adj.P.Val < 0.01

## Number of significant genes
sum(sig)
```

```
## [1] 1472
```

Volcano plot

```
volcanoplot(fit, coef=3)
points(g3_vs_g1$logFC[which(sig)], -log10(g3_vs_g1$P.Value[which(sig)]),
       cex=0.4, col='red', pch=16)
abline(h = min(-log10(g3_vs_g1$P.Value[which(sig)])), lty=2, col='blue')
```



Add fold-change threshold

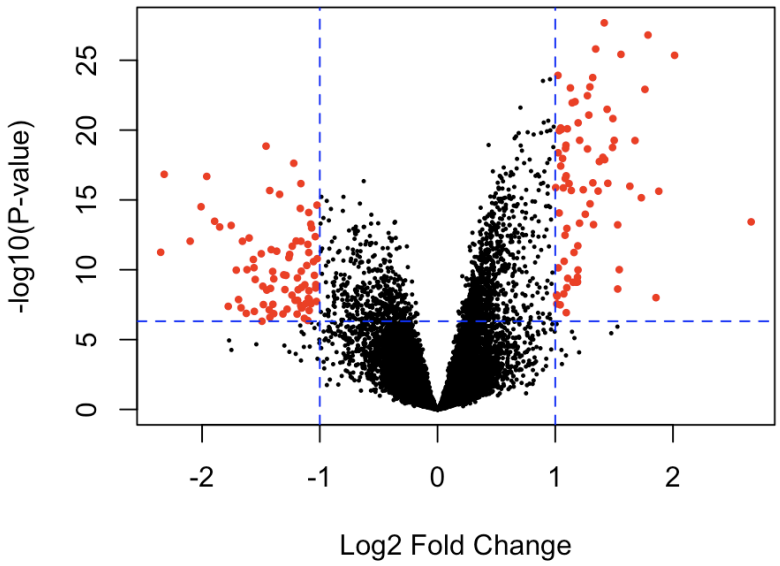
- Can reduce the number of genes in our list by imposing a fold-change threshold
- Here I've required genes to exhibit a fold-change of at least 2 (i.e., 1 on the log_2 scale) between the groups.

```
sigFC = (g3_vs_g1$adj.P.Val < 0.01)  & (abs(g3_vs_g1$logFC) > 1)
sum(sigFC)
```

```
## [1] 160
```

Volcano plot with FC threshold

```
volcanoplot(fit, coef=3)
points(g3_vs_g1$logFC[which(sigFC)],
       -log10(g3_vs_g1$P.Value[which(sigFC)]),
       cex=0.6, col='red', pch=16)
abline(h = min(-log10(g3_vs_g1$P.Value[which(sigFC)])), lty=2, col='blue')
abline(v = c(-1,1), lty=2, col='blue')
```



Significant genes

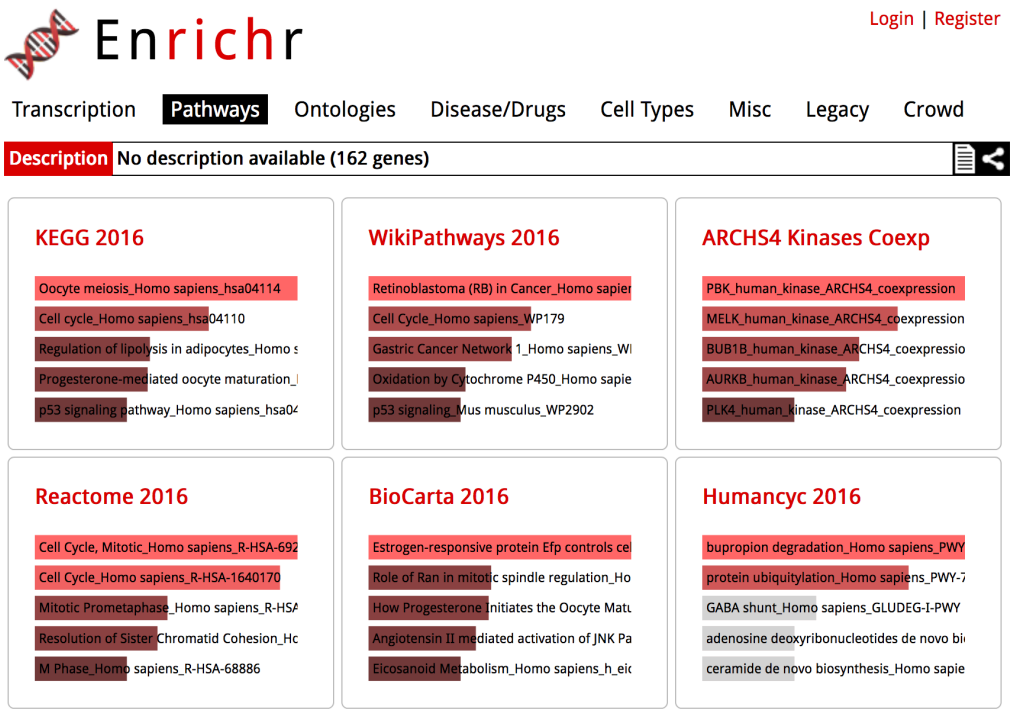
```
sigGenes = rownames(g3_vs_g1)[which(sigFC)]
sigGenes %>%  head(.,30)
```

```
##  [1] "CENPW"           "NCAPH"           "TTK"
##  [4] "ANLN"            "CDCA8"           "TRIP13"
##  [7] "TPX2"            "MELK"            "RRM2"
## [10] "MYBL2"           "MAD2L1"          "CCNE2"
## [13] "CCNB2"           "SLC7A5"          "HJURP"
## [16] "FOXM1"           "CEP55"           "DLGAP5"
## [19] "UBE2S"           "CENPN"           "AURKA"
## [22] "CDC20"           "SPC24"           "CENPE"
## [25] "PRC1"            "FOS"             "ASPM"
## [28] "FAM72A///FAM72B" "CDT1"            "BUB1"
```

```
strsplit(sigGenes, "///") %>%  unlist() %>%  unique() %>% length()
```

```
## [1] 162
```

Over-representation analysis: Enrichr



http://amp.pharm.mssm.edu/Enrichr/

Over-representation analysis: Enrichr

Reactome 2016 Bar Graph **Table** Clustergram

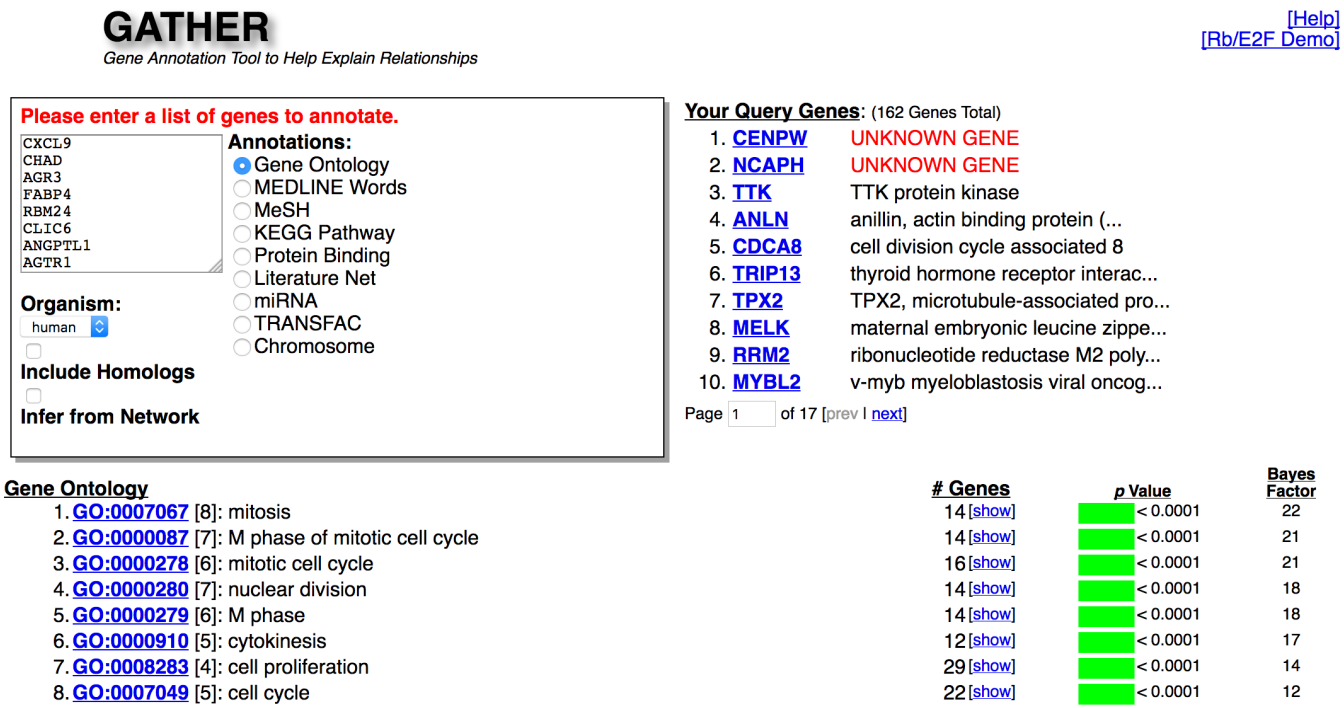
Hover each row to see the overlapping genes.

10 entries per page Search:

Index	Name	P-value	Adjusted p-value	Z-score	Combined score
1	Cell Cycle, Mitotic_Homo sapiens_R-HSA-69278	6.831e-21	2.418e-18	-2.48	115.02
2	Cell Cycle_Homo sapiens_R-HSA-1640170	3.213e-20	5.688e-18	-2.45	110.06
3	Mitotic Prometaphase_Homo sapiens_R-HSA-68877	1.466e-17	1.730e-15	-2.03	78.78
4	Resolution of Sister Chromatid Cohesion_Homo sapiens_R-HSA-2500257	1.024e-16	9.061e-15	-2.06	76.01
5	M Phase_Homo sapiens_R-HSA-68886	6.506e-13	3.838e-11	-2.41	67.49
6	Mitotic Metaphase and Anaphase_Homo sapiens_R-HSA-2555396	1.210e-11	6.120e-10	-2.30	57.80
7	RHO GTPases Activate Formins_Homo sapiens_R-HSA-5663220	4.606e-13	3.261e-11	-1.99	56.65
8	Separation of Sister Chromatids_Homo sapiens_R-HSA-2467813	5.824e-11	2.577e-9	-2.26	53.26
9	Mitotic Anaphase_Homo sapiens_R-HSA-68882	1.406e-10	5.529e-9	-2.28	51.71
10	RHO GTPase Effectors_Homo sapiens_R-HSA-195258	2.906e-10	1.029e-8	-2.19	48.14

http://amp.pharm.mssm.edu/Enrichr/

Over-representation analysis: GATHER



http://changlab.uth.tmc.edu/gather/gather.py

Over-representation: GeneSetDB

1. Gene List

paste gene list

```
KCNK15
PDZK1
BMPR1B
SEMA3B
PIP
CXCL9
CHAD
AGR3
FABP4
RBM24
CLIC6
ANGPTL1
AGTR1
```

Or

upload gene list file

Browse... No file selected.

Enrichment analysis used: 14974

2. Input ID type

Symbol Please specify the identifier for the gene list.

3. Choose DB

All
SubClass Pathway
SubClass Disease/Phenotype
SubClass Drug/Chemical
SubClass Gene Regulation

4. FDR

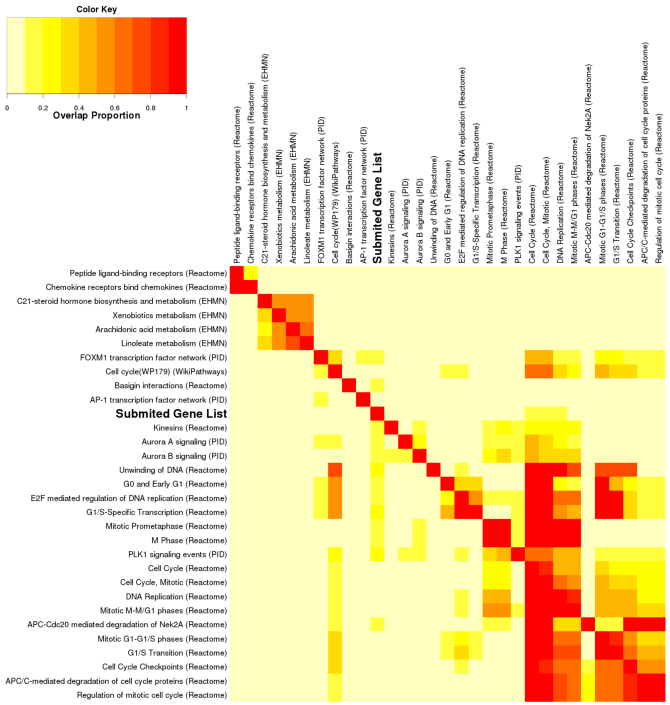
0.1

5. Submit

After submit is pressed it can take a little while until the page refreshes.

EA Submit Reset

Sample data
Enrichment Analysis tutorial



http://genesetdb.auckland.ac.nz

Over-representation: GeneSetDB

GeneSetDB Home About Source DB Download Help

Download

162 symbol input ids were converted into 161 unique gene ids.
38 entries for fdr cutoff 0.1 estimated.

Sub Class	Gene Set Name	Source DB	Gene #	Gene # with Anno	Gene # without Anno	p-value	FDR
Pathway	Cell Cycle, Mitotic	Reactome	330	26	304	4.7E-16	2.2E-13
Pathway	Cell Cycle	Reactome	409	27	382	9.4E-15	2.2E-12
Pathway	M Phase	Reactome	96	14	82	1.3E-12	2.1E-10
Pathway	DNA Replication	Reactome	200	18	182	3.0E-12	3.6E-10
Pathway	Mitotic Prometaphase	Reactome	92	13	79	1.4E-11	1.2E-9
Pathway	PLK1 signaling events	PID	42	10	32	1.5E-11	1.2E-9
Pathway	Mitotic M-M/G1 phases	Reactome	178	16	162	5.7E-11	3.8E-9
Pathway	FOXM1 transcription factor network	PID	40	7	33	2.0E-7	1.2E-5
Pathway	G1/S-Specific Transcription	Reactome	17	5	12	7.5E-7	3.9E-5
Pathway	Aurora B signaling	PID	39	6	33	3.4E-6	1.6E-4
Pathway	Aurora A signaling	PID	31	5	26	1.8E-5	7.8E-4
Pathway	E2F mediated regulation of DNA replication	Reactome	33	5	28	2.5E-5	9.9E-4
Pathway	AP-1 transcription factor network	PID	69	6	63	9.6E-5	3.4E-3
Pathway	Mitotic G1-G1/S phases	Reactome	135	8	127	1.0E-4	3.4E-3
Pathway	Cell cycle(WP179)	WikiPathways	103	7	96	1.2E-4	3.8E-3
Pathway	G0 and Early G1	Reactome	25	4	21	1.4E-4	4.0E-3
Pathway	Kinesins	Reactome	27	4	23	1.9E-4	5.0E-3

http://genesetdb.auckland.ac.nz

Over-representation analysis in R

```
reactome()
```

```
## Quality control information for reactome:
##
##
## This package has the following mappings:
##
## reactomeEXTID2PATHID has 69713 mapped keys (of 69713 keys)
## reactomeGO2REACTOMEID has 1880 mapped keys (of 1880 keys)
## reactomePATHID2EXTID has 22001 mapped keys (of 22001 keys)
## reactomePATHID2NAME has 22823 mapped keys (of 22823 keys)
## reactomePATHNAME2ID has 22795 mapped keys (of 22795 keys)
## reactomeREACTOMEID2GO has 11671 mapped keys (of 11671 keys)
##
##
## Additional Information about this package:
##
## DB schema: REACTOME_DB
## DB schema version: 62
```

Reactome data

```
## Pathways and Entrez IDs (first 2 pathways, first 6 genes from each)
lapply( as.list(reactomePATHID2EXTID)[1:2], head )
```

```
## $`R-HSA-109582`
## [1] "1"     "10000" "10000" "10019" "10112" "10125"
##
## $`R-HSA-114608`
## [1] "1"     "10184" "10257" "10447" "10487" "10490"
```

```
## Pathway names (first 2)
as.list(reactomePATHID2NAME)[1:2]
```

```
## $`R-ATH-73843`
## [1] "1-diphosphate: 5-Phosphoribose"
##
## $`R-ATH-1369062`
## [1] "Arabidopsis thaliana: ABC transporters in lipid homeostasis"
```

ReactomePA

```
## Convert gene list symbols to Entrez gene IDs
sigEntrez = fData(upp)$EntrezGene.ID[ match(sigGenes,
                            fData(upp)$Gene.symbol) ] %>%
            as.vector()
head(sigEntrez)
```

```
## [1] "387103" "23397"  "7272"   "54443"  "55143"  "9319"
```

```
## Perform pathway analysis via ReactomePA
rPAoverrep <- enrichPathway(gene=sigEntrez, organism = "human",
                            pvalueCutoff=0.05, readable=T)
```

ReactomePA results

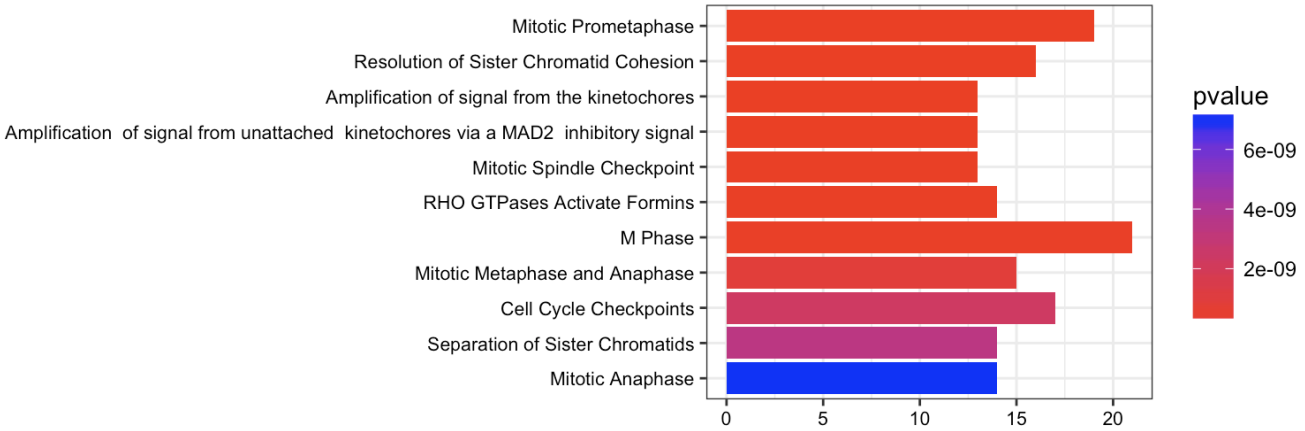
```
rPAoverrep %>%  as.data.frame() %>%  View()
```

	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue	geneID	Count
R-HSA-68877	R-HSA-68877	Mitotic Prometaphase	19/101	200/10513	3.574491e-14	9.425177e-12	7.803221e-12	NCAPH/CDCA8/MAD2L1/CCNB2/CENPN/CDC20/SPC2...	19
R-HSA-2500257	R-HSA-2500257	Resolution of Sister Chromatid Cohesion	16/101	127/10513	5.295043e-14	9.425177e-12	7.803221e-12	CDCA8/MAD2L1/CCNB2/CENPN/CDC20/SPC24/CENP...	16
R-HSA-141424	R-HSA-141424	Amplification of signal from the kinetochores	13/101	96/10513	5.631343e-12	5.011895e-10	4.149410e-10	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/BUB1/...	13
R-HSA-141444	R-HSA-141444	Amplification of signal from unattached kinetochores v...	13/101	96/10513	5.631343e-12	5.011895e-10	4.149410e-10	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/BUB1/...	13
R-HSA-69618	R-HSA-69618	Mitotic Spindle Checkpoint	13/101	112/10513	4.181019e-11	2.976886e-09	2.464601e-09	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/BUB1/...	13
R-HSA-5663220	R-HSA-5663220	RHO GTPases Activate Formins	14/101	141/10513	6.017159e-11	3.570181e-09	2.955797e-09	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/BUB1/...	14
R-HSA-68886	R-HSA-68886	M Phase	21/101	395/10513	1.132125e-10	5.757665e-09	4.766843e-09	NCAPH/CDCA8/MAD2L1/CCNB2/CENPN/CDC20/SPC2...	21
R-HSA-2555396	R-HSA-2555396	Mitotic Metaphase and Anaphase	15/101	203/10513	7.967853e-10	3.545695e-08	2.935525e-08	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/BUB1/...	15
R-HSA-69620	R-HSA-69620	Cell Cycle Checkpoints	17/101	293/10513	2.253131e-09	8.912385e-08	7.378674e-08	CDCA8/MAD2L1/CCNE2/CCNB2/CENPN/CDC20/SPC2...	17
R-HSA-2467813	R-HSA-2467813	Separation of Sister Chromatids	14/101	191/10513	3.408672e-09	1.213487e-07	1.004661e-07	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/BUB1/...	14
R-HSA-68882	R-HSA-68882	Mitotic Anaphase	14/101	202/10513	7.037382e-09	2.277553e-07	1.885614e-07	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/BUB1/...	14
R-HSA-195258	R-HSA-195258	RHO GTPase Effectors	16/101	318/10513	5.257507e-08	1.559727e-06	1.291317e-06	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/PRC1/...	16
R-HSA-194315	R-HSA-194315	Signaling by Rho GTPases	17/101	446/10513	1.024841e-06	2.806487e-05	2.323526e-05	CDCA8/MAD2L1/CENPN/CDC20/SPC24/CENPE/PRC1/...	17
R-HSA-539107	R-HSA-539107	Activation of E2F1 target genes at G1/S	5/101	28/10513	6.109139e-06	1.449902e-04	1.200392e-04	RRM2/CDT1/CDK1/FBXO5/CDC45	5
R-HSA-69205	R-HSA-69205	G1/S-Specific Transcription	5/101	28/10513	6.109139e-06	1.449902e-04	1.200392e-04	RRM2/CDT1/CDK1/FBXO5/CDC45	5
R-HSA-174143	R-HSA-174143	APC/C-mediated degradation of cell cycle proteins	7/101	86/10513	1.769268e-05	3.705055e-04	3.067462e-04	MAD2L1/AURKA/CDC20/CDK1/AURKB/NEK2/FBXO5	7
R-HSA-453276	R-HSA-453276	Regulation of mitotic cell cycle	7/101	86/10513	1.769268e-05	3.705055e-04	3.067462e-04	MAD2L1/AURKA/CDC20/CDK1/AURKB/NEK2/FBXO5	7
R-HSA-4615885	R-HSA-4615885	SUMOylation of DNA replication proteins	5/101	44/10513	5.976921e-05	1.182102e-03	9.786772e-04	CDCA8/AURKA/AURKB/TOP2A/NUP93	5
R-HSA-453279	R-HSA-453279	Mitotic G1-G1/S phases	8/101	147/10513	8.147982e-05	1.526675e-03	1.263953e-03	RRM2/MYBL2/CCNE2/CDT1/CDK1/TOP2A/FBXO5/CD...	8
R-HSA-1538133	R-HSA-1538133	G0 and Early G1	4/101	27/10513	1.188395e-04	2.115344e-03	1.751319e-03	MYBL2/CCNE2/CDK1/TOP2A	4
R-HSA-2514853	R-HSA-2514853	Condensation of Prometaphase Chromosomes	3/101	11/10513	1.342875e-04	2.276493e-03	1.884737e-03	NCAPH/CCNB2/CDK1	3
R-HSA-176974	R-HSA-176974	Unwinding of DNA	3/101	12/10513	1.778018e-04	2.877157e-03	2.382034e-03	GINS1/GINS2/CDC45	3
R-HSA-156711	R-HSA-156711	Polo-like kinase mediated events	3/101	16/10513	4.401101e-04	6.812139e-03	5.639855e-03	MYBL2/CCNB2/FOXM1	3
R-HSA-375276	R-HSA-375276	Peptide ligand-binding receptors	8/101	190/10513	4.731074e-04	7.017760e-03	5.810091e-03	CX3CR1/CXCL10/NMU/CXCL11/ACKR1/NPY1R/CXCL9...	8
R-HSA-69275	R-HSA-69275	G2/M Transition	8/101	196/10513	5.814574e-04	8.279953e-03	6.855076e-03	TPX2/MYBL2/CCNB2/FOXM1/AURKA/CDK1/NEK2/HA...	8
R-HSA-453274	R-HSA-453274	Mitotic G2-G2/M phases	8/101	198/10513	6.217429e-04	8.513095e-03	7.048097e-03	TPX2/MYBL2/CCNB2/FOXM1/AURKA/CDK1/NEK2/HA...	8

ReactomePA visualisations

- NB: `barplot.enrichResult` is from the `DOSE` pacakge.

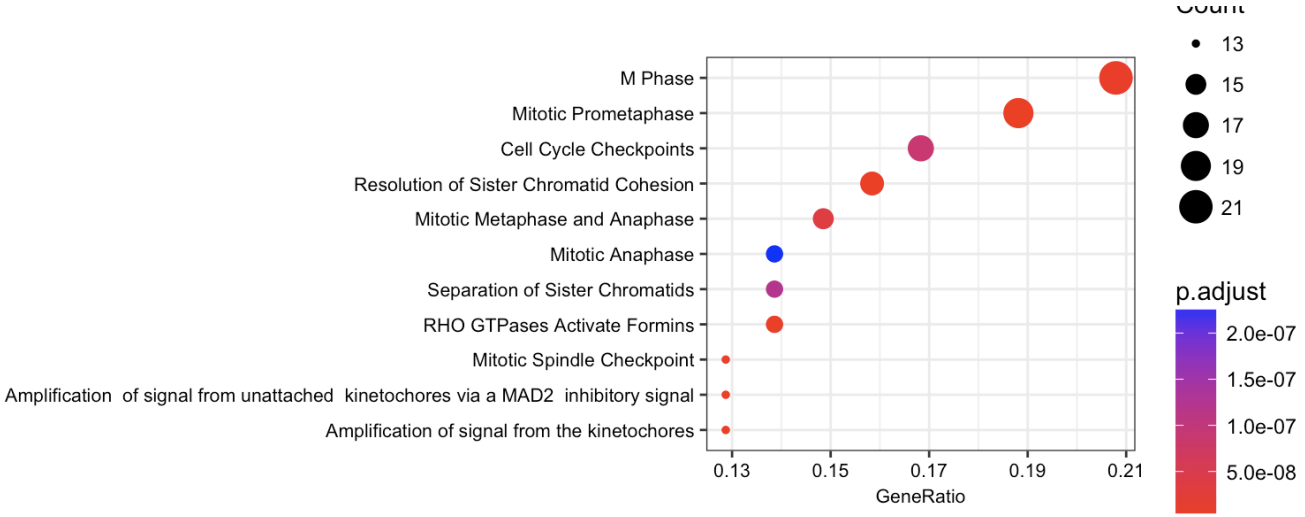
```
barplot(rPAoverrep, showCategory=11, font.size=8)
```



ReactomePA visualisations

- NB: `dotplot` is from the `DOSE` pacakge.

```
dotplot(rPAoverrep, showCategory=11, font.size=8)
```



ReactomePA: GSEA

```
tstats = g3_vs_g1$t
names(tstats) = fData(upp)$EntrezGene.ID[ match(rownames(g3_vs_g1),
                            fData(upp)$Gene.symbol) ] %>%
                as.vector()
tstats = sort(tstats, decreasing=TRUE)
head(tstats)
```

```
##   387103    23397     7272    54443    55143     9319
## 12.57173 12.31195 12.01225 11.89672 11.87608 11.44096
```

```
rPAgsea = gsePathway(tstats, nPerm=10000,
                     minGSSize=50, pvalueCutoff=0.2,
                     pAdjustMethod="BH", verbose=FALSE)
```

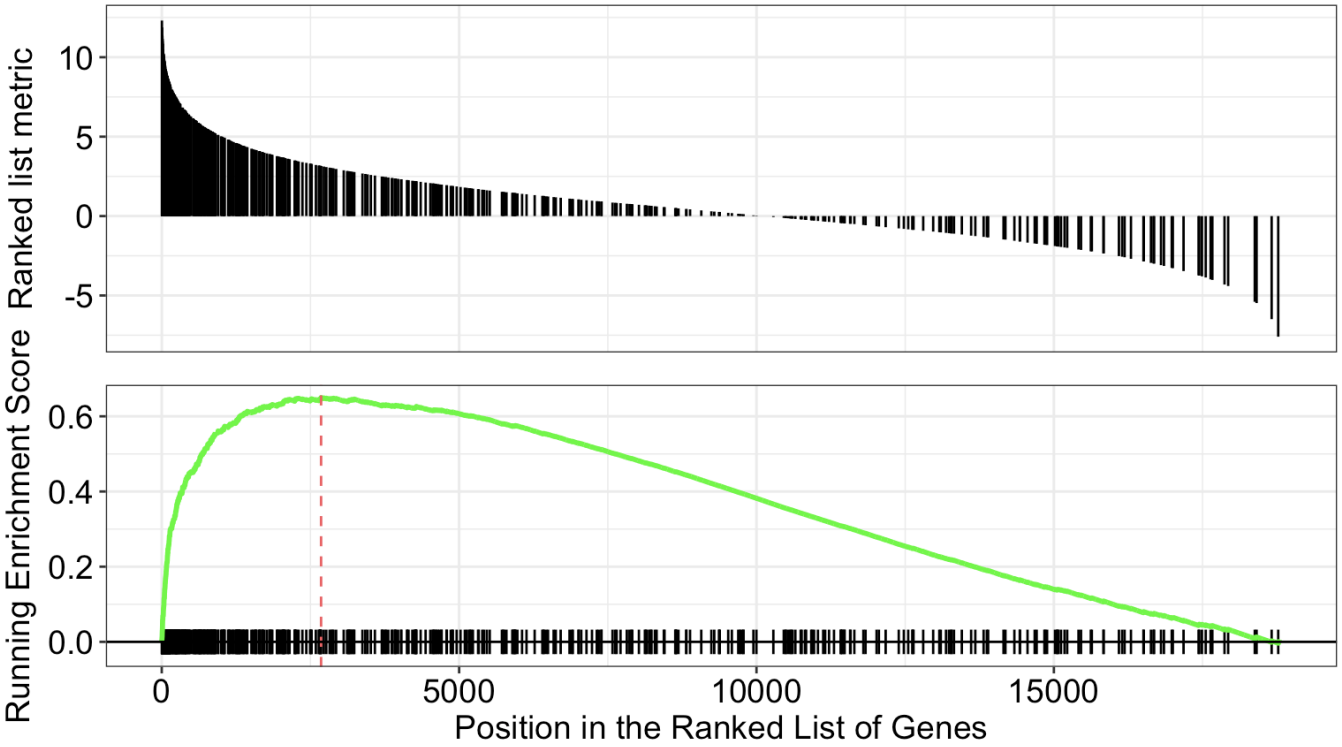
ReactomePA: GSEA results

```
rPAgsea %>%  as.data.frame() %>%  View()
```

	ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalues	rank
R-HSA-69278	R-HSA-69278	Cell Cycle, Mitotic	492	0.6513111	3.085692	0.0001108279	0.0003379949	0.0001103625	2679
R-HSA-8953897	R-HSA-8953897	Cellular responses to external stimuli	461	0.4463163	2.105840	0.0001118443	0.0003379949	0.0001103625	4365
R-HSA-6798695	R-HSA-6798695	Neutrophil degranulation	457	0.3800468	1.791979	0.0001119821	0.0003379949	0.0001103625	4261
R-HSA-194315	R-HSA-194315	Signaling by Rho GTPases	399	0.4259700	1.989862	0.0001142988	0.0003379949	0.0001103625	3457
R-HSA-2262752	R-HSA-2262752	Cellular responses to stress	387	0.4661951	2.174195	0.0001147052	0.0003379949	0.0001103625	4274
R-HSA-5663205	R-HSA-5663205	Infectious disease	361	0.5438124	2.523521	0.0001157675	0.0003379949	0.0001103625	4496
R-HSA-3700989	R-HSA-3700989	Transcriptional Regulation by TP53	353	0.4647744	2.152749	0.0001160497	0.0003379949	0.0001103625	3801
R-HSA-983169	R-HSA-983169	Class I MHC mediated antigen processing & presentation	355	0.4184910	1.939257	0.0001160766	0.0003379949	0.0001103625	4675
R-HSA-68886	R-HSA-68886	M Phase	352	0.6354256	2.942449	0.0001161305	0.0003379949	0.0001103625	2679
R-HSA-71291	R-HSA-71291	Metabolism of amino acids and derivatives	342	0.4021524	1.859350	0.0001166317	0.0003379949	0.0001103625	3615
R-HSA-983168	R-HSA-983168	Antigen processing: Ubiquitination & Proteasome degr...	295	0.4013299	1.827817	0.0001191469	0.0003379949	0.0001103625	4669
R-HSA-195721	R-HSA-195721	Signaling by WNT	288	0.3811752	1.732904	0.0001193175	0.0003379949	0.0001103625	2828
R-HSA-73894	R-HSA-73894	DNA Repair	287	0.5130989	2.331089	0.0001195457	0.0003379949	0.0001103625	3750
R-HSA-72766	R-HSA-72766	Translation	282	0.5791830	2.626401	0.0001200336	0.0003379949	0.0001103625	4086
R-HSA-195258	R-HSA-195258	RHO GTPase Effectors	276	0.5443395	2.463262	0.0001201779	0.0003379949	0.0001103625	3457
R-HSA-69620	R-HSA-69620	Cell Cycle Checkpoints	266	0.6898064	3.111213	0.0001206418	0.0003379949	0.0001103625	2148
R-HSA-5688426	R-HSA-5688426	Deubiquitination	254	0.4429510	1.988909	0.0001218027	0.0003379949	0.0001103625	4053
R-HSA-72203	R-HSA-72203	Processing of Capped Intron-Containing Pre-mRNA	238	0.5279674	2.354442	0.0001230618	0.0003379949	0.0001103625	3622
R-HSA-162906	R-HSA-162906	HIV Infection	222	0.5907054	2.618951	0.0001235483	0.0003379949	0.0001103625	4049
R-HSA-8951664	R-HSA-8951664	Neddylation	223	0.4428202	1.963243	0.0001236858	0.0003379949	0.0001103625	2592
R-HSA-376176	R-HSA-376176	Signaling by ROBO receptors	205	0.4732645	2.077522	0.0001256281	0.0003379949	0.0001103625	4426
R-HSA-8878171	R-HSA-8878171	Transcriptional regulation by RUNX1	202	0.4516929	1.979601	0.0001258020	0.0003379949	0.0001103625	2069
R-HSA-201681	R-HSA-201681	TCF dependent signaling in response to WNT	195	0.4379162	1.911297	0.0001265342	0.0003379949	0.0001103625	2069
R-HSA-453274	R-HSA-453274	Mitotic G2-G2/M phases	190	0.6115142	2.663345	0.0001265983	0.0003379949	0.0001103625	2365
R-HSA-2555396	R-HSA-2555396	Mitotic Metaphase and Anaphase	193	0.7034368	3.068885	0.0001266143	0.0003379949	0.0001103625	2274
R-HSA-72312	R-HSA-72312	rRNA processing	194	0.5619611	2.451336	0.0001266785	0.0003379949	0.0001103625	3580
R-HSA-68877	R-HSA-68877	Mitotic Prometaphase	192	0.6416826	2.797701	0.0001267106	0.0003379949	0.0001103625	2519
R-HSA-68882	R-HSA-68882	Mitotic Anaphase	192	0.7014150	3.058130	0.0001267106	0.0003379949	0.0001103625	2274

ReactomePA: GSEA plot

```
gseaplot(rPAgsea, geneSetID = "R-HSA-69278")
```



Heatmaps

- Heatmaps are a great way to visualise the expression patterns of gene sets across samples.
  - Incredibly overused (especially by me), and rather easy to abuse.
  - Popular to use in conjunction with hierarchical clustering of rows (genes) and columns (samples).
- Clusters often extracted for comparison with clinical/phenotypic variables.
- Let's have a look at the expression of the genes in our list.

Heatmap setup

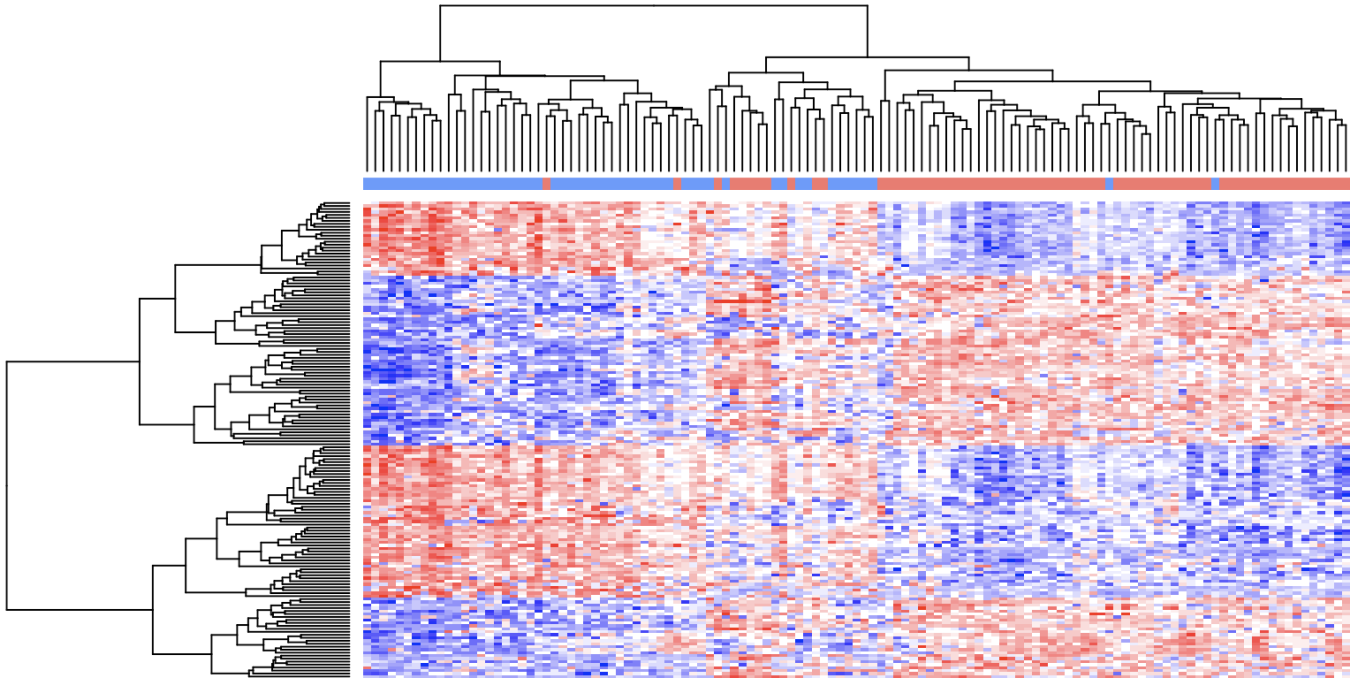
```
## Extract expression data for gens of interest
uppExprSig = uppExpr[match(sigGenes, rownames(uppExpr)), ]
dim(uppExprSig)
```

```
## [1] 160 251
```

```
## Use ggplot colours for Grades
gradeCols = gg_color_hue(3)[as.numeric(as.factor(uppClin$grade))]

## Identify which samples are Grade 1 or 3
g1_g3 = uppClin$grade%in%c("G1", "G3") %>%  which()
```

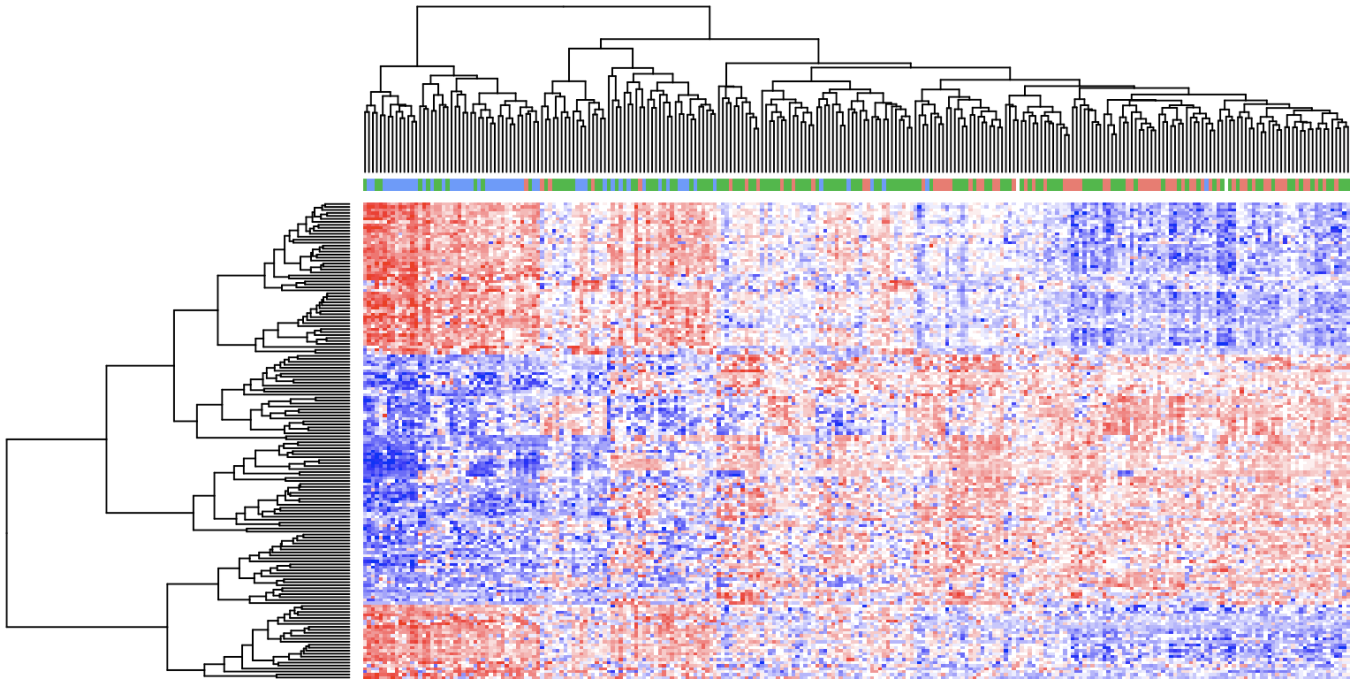
Heatmap: G1 & G3

```
heatmap.2(uppExprSig[,g1_g3], col=bluered(30), trace='none',
          scale='row', ColSideColors = gradeCols[g1_g3], key=FALSE,
          breaks=seq(-3,3,l=31), labCol="", labRow="", mar=c(0,0))
```



Heatmap: all grades

```
heatmap.2(uppExprSig, col=bluered(30), trace='none',
          scale='row', ColSideColors = gradeCols, key=FALSE,
          breaks=seq(-3,3,l=31), labCol="", labRow="", mar=c(0,0))
```



Cluster groups

```
clusterGroups = uppExprSig %>%  t() %>% dist() %>%
  hclust() %>% cutree(., 3)

table(clusterGroups)
```

```
## clusterGroups
##   1   2   3
##  45 161  45
```

```
table(clusterGroups, uppClin$grade)
```

```
##
## clusterGroups G1 G2 G3
##             1  1 11 33
##             2 62 91  6
##             3  4 26 15
```

Gene expression summaries

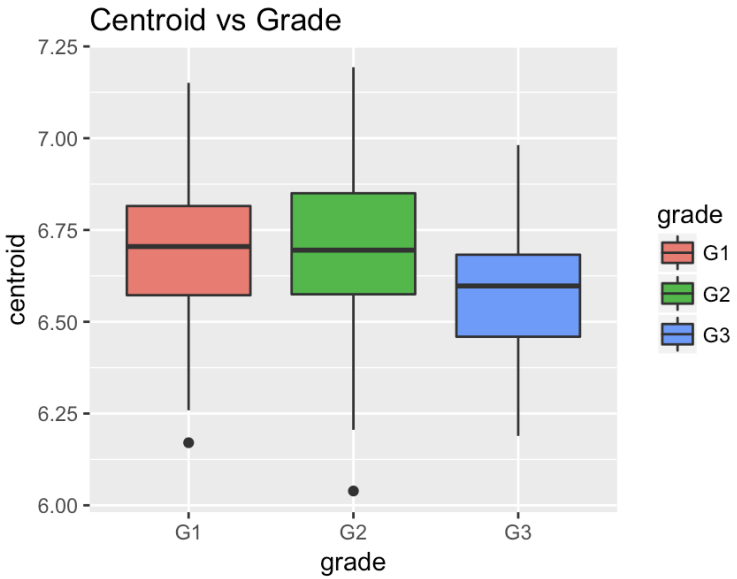
- The clusters appear to relate (to some extent) to the tumour grades.
- What if we want a continuous-valued variable to represent gene expression for each sample?
  - Can sometimes use the *centroid* (mean of genes in set per sample - column means here)
  - This works well when all the genes have a similar pattern…

```
centroid = colMeans(uppExprSig)
head(centroid)
```

```
## UPP_103B41 UPP_104B91 UPP_112B55 UPP_114B68 UPP_130B92 UPP_131B79
##   6.291062   6.888775   6.802450   6.856950   6.799506   6.822463
```

Centroid boxplot

```
data.frame(centroid = centroid, grade=as.factor(uppClin$grade)) %>%
     na.omit() %>% ggplot(., aes(x=grade, y=centroid, fill=grade)) +
     geom_boxplot() + ggtitle("Centroid vs Grade")
```



Here it doesn't provide any distinction between grades.

Metagenes

- Dimension reduction technqiues are commonly used to generate metagenes:
  - SVD/PCA/MDS
  - these are all basically the same: https://github.com/mikblack/msg-pca-20171114
- Other methods also exist:
  - Non-negative matrix factorisation (NMF)
  - Generally gives similar results to methods above

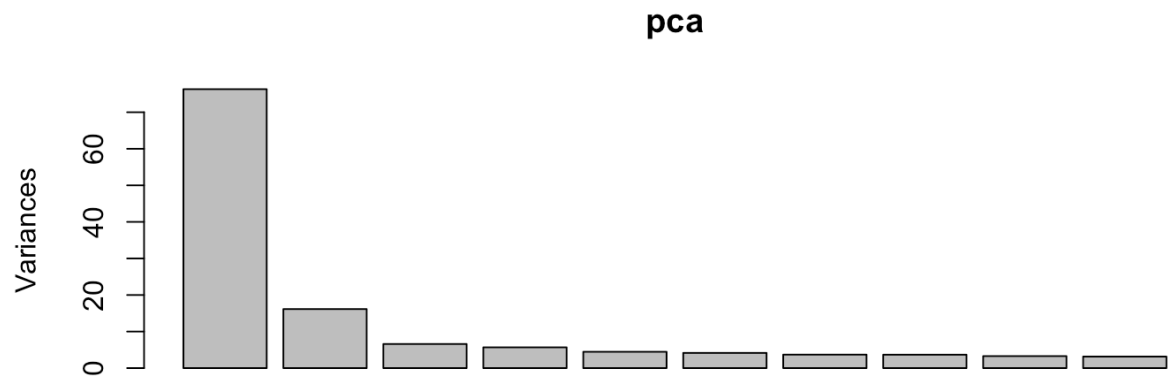
Metagene creation

```
pca = uppExprSig %>% scale(., scale=FALSE) %>%  t() %>%  prcomp()
metagene = pca$x[,1:3]
head(metagene)
```

```
##                   PC1       PC2        PC3
## UPP_103B41 -18.934243 -1.767337 -1.8475819
## UPP_104B91  -3.307331  4.807649  3.7349320
## UPP_112B55   2.520988  6.358609 -4.6019407
## UPP_114B68   1.544139  2.879148  1.9585313
## UPP_130B92  11.289497  6.677049  2.5467931
## UPP_131B79  -4.422589  3.259645 -0.6350316
```

Percentage variance explained

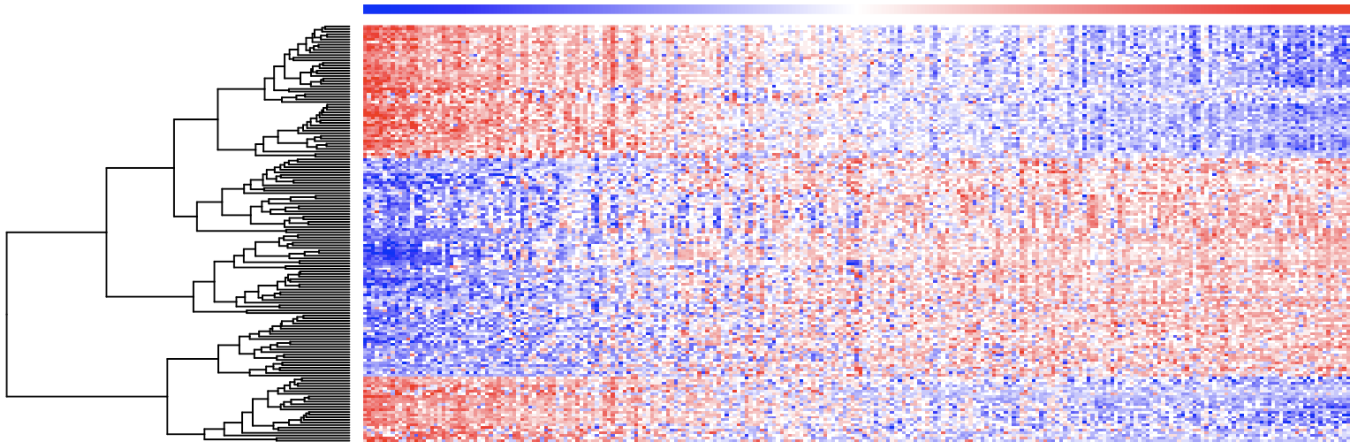
```
screeplot(pca)
```



Most of the variance is explained by the first eigenvector / PC.

Heatmap ordered by Metagene 1

```
ord = order(metagene[,1])
mgCols = bluered(nrow(metagene))[rank(metagene[,1])]
heatmap.2(uppExprSig[, ord], col=bluered(30), trace='none', Colv=FALSE,
          scale='row', ColSideColors = mgCols[ord], key=FALSE,
          breaks=seq(-3,3,l=31), labCol="", labRow="", mar=c(0,0))
```

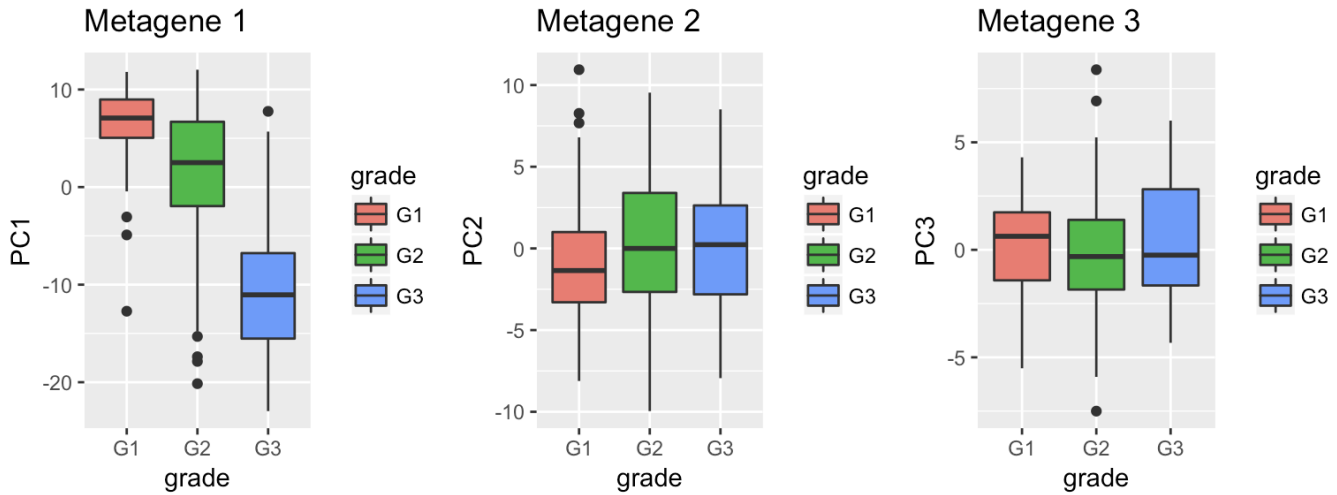


Setup for grade plots

```
p1 = data.frame(PC1 = metagene[,1], grade=as.factor(uppClin$grade)) %>%
     na.omit() %>% ggplot(., aes(x=grade, y=PC1, fill=grade)) +
     geom_boxplot() + ggtitle("Metagene 1")

p2 = data.frame(PC2 = metagene[,2], grade=as.factor(uppClin$grade)) %>%
     na.omit() %>% ggplot(., aes(x=grade, y=PC2, fill=grade)) +
     geom_boxplot() + ggtitle("Metagene 2")

p3 = data.frame(PC3 = metagene[,3], grade=as.factor(uppClin$grade)) %>%
     na.omit() %>%
     ggplot(., aes(x=grade, y=PC3, fill=grade)) +
     geom_boxplot() + ggtitle("Metagene 3")
```

Metagenes versus Grade

```
grid.arrange(p1, p2, p3, ncol=3)
```



First metagene is strongly associated with grade: provides a good continuous-valued summary of proliferation level.

Genomic grade

- In their 2005 publication, Miler et al. used "proliferation level" to split the Grade 2 tumours into two groups.
  - low proliferation Grade 2 tumours behave like Grade 1
  - high proliferation Grade 2 tumours behave like Grade 3
- This result was validated in other data sets.
- Let's try the same thing here.

Genomic Grade

```
prolif = ifelse(metagene[,1] < median(metagene[,1]), "High", "Low")
table(prolif, uppClin$grade)
```

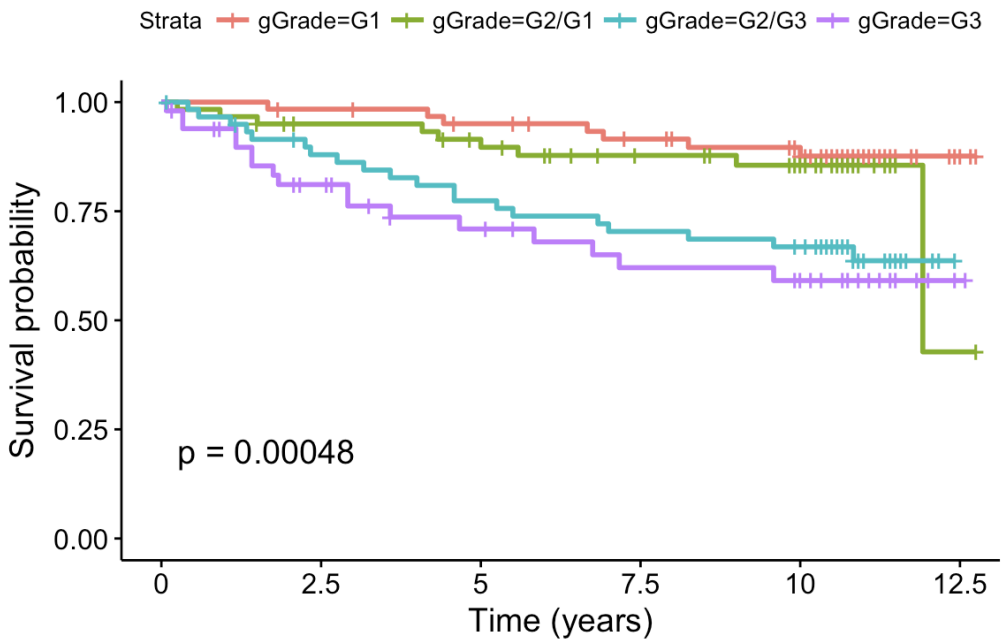
```
##
## prolif G1 G2 G3
##   High  9 63 52
##   Low  58 65  2
```

```
uppClin$gGrade = uppClin$grade
uppClin$gGrade[uppClin$grade=="G2" & prolif=="High"] = "G2/G3"
uppClin$gGrade[uppClin$grade=="G2" & prolif=="Low"]  = "G2/G1"
table(uppClin$gGrade)
```

```
##
##    G1 G2/G1 G2/G3    G3
##    67    65    63    54
```

Survival plot (RFS vs Genomic Grade)

```
ggsurvplot(survfit(Surv(t.rfs, e.rfs) ~ gGrade, data=uppClin),
           pval = TRUE, risk.table = FALSE) + xlab("Time (years)")
```



Metagenes from Reactome pathways

```
reactomePath = as.list(reactomePATHID2NAME)
reactomePath = reactomePath[grep("Homo sapiens", reactomePath)]
reactomePath[ grep('immune', reactomePath, ignore.case = TRUE)] %>%
  head(.,3)
```

```
## $`R-HSA-1280218`
## [1] "Homo sapiens: Adaptive Immune System"
##
## $`R-HSA-1280215`
## [1] "Homo sapiens: Cytokine Signaling in Immune system"
##
## $`R-HSA-5260271`
## [1] "Homo sapiens: Diseases of Immune System"
```

Metagenes from Reactome pathways

```
## Get Entrez IDs for genes in Cytokine Signalling pathway
pathRows = as.list(reactomePATHID2EXTID)["R-HSA-1280215"] %>%
  unlist() %>%  match(., fData(upp)$EntrezGene.ID) %>%
  na.omit() %>%  as.vector()
head(pathRows)
```

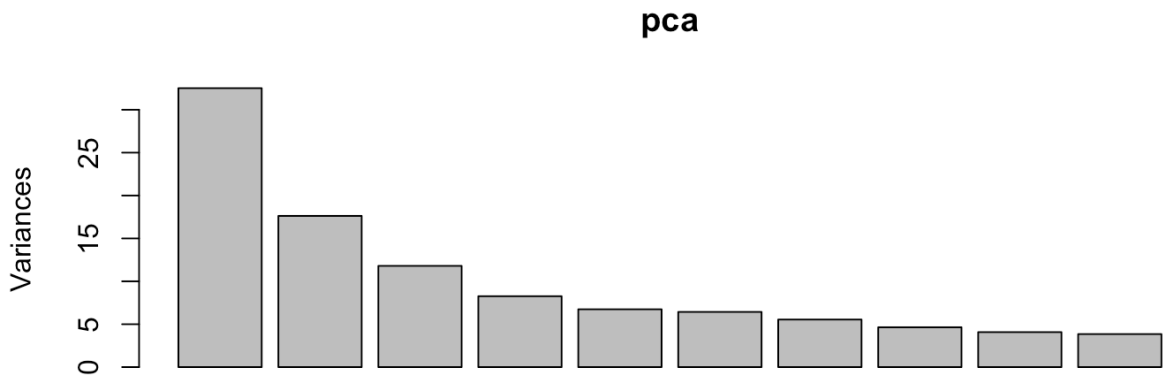
```
## [1]  2110 18687 10045  5117  7389 18788
```

```
## Extract expression data for these genes
pathExprDat = uppExpr[fData(upp)$Gene.symbol[pathRows], ]
dim(pathExprDat)
```

```
## [1] 881 251
```

Immune response metagene

```
## Use PCA to generate metagenes
pca = pathExprDat %>% scale(., scale=FALSE) %>%  t() %>%  prcomp()
metagene = pca$x[,1]
screeplot(pca)
```



IR metagene vs survival (ER neg)

```
mgHilo = ifelse(metagene < quantile(metagene,0.2), "mgLo", "mgHi")
ggsurvplot(survfit(Surv(t.rfs, e.rfs) ~ mgHilo, subset=uppClin$er==0,
           data=uppClin),pval=TRUE) + xlab("Time (years)")
```