GSEA analysis

[针对非模式生物][https://mp.weixin.qq.com/s/awieqE2LAk7YKZNkNQr9VQ]

- ORA:Over-Representation Analysis(ORA)

超几何分布(Classic Fisher Test)：例如在芯片中共有10000个基因，其中通路S含有200个基因(占芯片基因的2%)；同时有50个基因位于"distiguished"列表中(挑选出来的显著差异基因)，实际上通路S在本次实验中共有6个基因位于distiguished列表中。此时理论情况下50*2%=1个基因位于通路S，那么现有6个基因位于S。使用超几何分布求得该事件的p值(也就是随机情况下有6个或者6个以上的distingushed基因位于S的概率) p=0.00045

Fisher精确检验：

```
> d <- matrix(c(6,194,44,9756),nrow=2,dimnames=list(c("DE","Not DE"),c("In GS","Not in GS")))
> d
       In GS Not in GS
DE         6        44
Not DE   194      9756
> fisher.test(d,alternative = 'greater')

        Fisher's Exact Test for Count Data

data:  d
p-value = 0.0004534
alternative hypothesis: true odds ratio is greater than 1
95 percent confidence interval:
 2.814658      Inf
sample estimates:
odds ratio
   6.85379
```

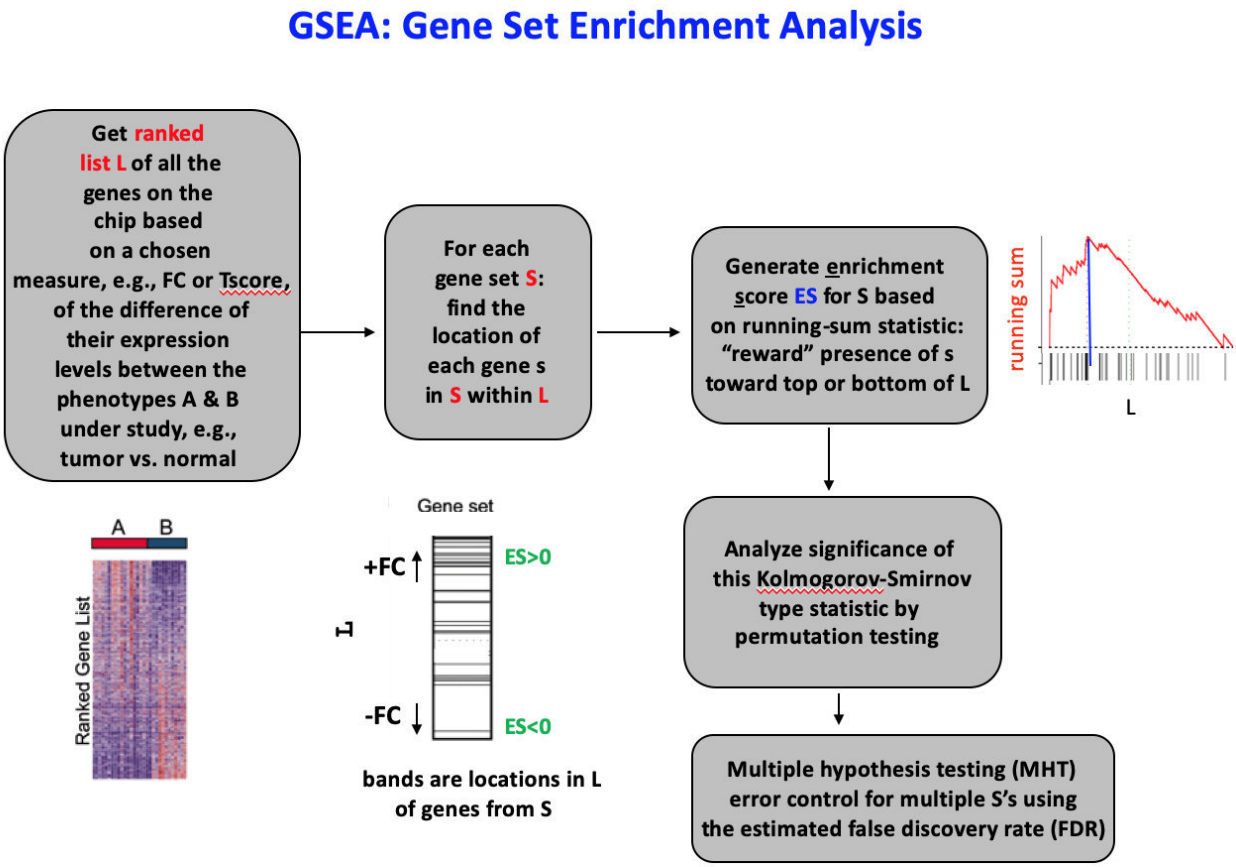
超几何分布：

```
> phyper(5,200,10000-200,50,lower.tail=F)
[1] 0.000453366
```

- GSEA: Gene Set Enrichment Analysis

Fisher and ks are just two ways of answering the same question: are the most significant genes enriched for any particular GO term annotations? Fisher's exact test compares the expected number of significant genes at random to the observed number of significant genes to arrive at a probability. The KS test compares the distribution of gene p-values expected at random to the observed distribution of the gene p-values to arrive at a probability. KS is theoretically the better choice because it does not require an arbitrary p-value threshold.

该分析解决了ORA分析过程中仅考虑差异大的基因，而忽略了差异较小但是一致性表达的一组相关基因的问题。该分析将所有基因均用于GSEA分析，GSEA整合gene set中每个基因的统计值，检测提前定义的gene set中所有基因发生小且一致性改变的情况。因为，可能出现许多表型差异与一组变化较小但是变化一致的基因所联系起来的情况。



根据表型对基因排秩序，针对一套给定prior的gene set S(eg. genes sharing the same GO category)，GSEA分析目的在于判断S中的基因是否随机分布在排完秩序后的gene list(L)中，或主要分布在list的top或bottom。

富集值(ES)代表了一组基因S(存在于同一GO类别)位于经过排秩序后gene list(L)的top或bottom的程度(也就是在top或bottom过表达程度)。通过统计L中出现S的情况，计算得到ES值，最终通过统计换算计算相对于null distribution的ES的p-value。

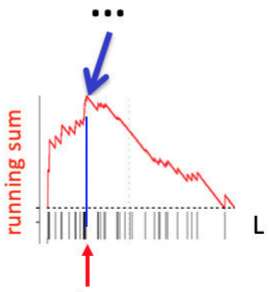
Enrichment Score (ES) Calculation

Start with ranked list (L) of genes that are in (Hit) or not in (Miss) a gene set (S),
using fold change (FC) as example metric

Ranked List (L)	FC		Contribution to running sum for ES	Hits +|FC| / Σ	Misses -1/(N-N_H)	Running sum for ES
———	15	Hit	+0.15	+0.15		0.15
———	12	Hit	+0.12	+0.12		0.27
———	10	Miss	-0.001		-0.001	0.269
———	9	Hit	+0.09	+0.09		0.359
———	8	Hit	+0.08	+0.08		0.439
———	6	Miss	-0.001		-0.001	0.438
...	...	...				...

Hits: Genes ∈ S +|FC| / Σ
Misses: Genes ∉ S -1/(N-N_H)

Σ = sum of fold changes for genes in gene set (S) (e.g., 100)
N = no. of genes in the array (e.g., 1020)
N_H = no. of genes in the gene set (S) (e.g., 20)



ES(S) ≡ value of maximum deviation from 0 of the running sum

- GSEA分析构建prerank gene list

~~[使用topGO针对DESeq2分析后结果做GSEA][https://www.biostars.org/p/279097/]，KS可尝试！！！~~

```
x <- read.table("DE_genes.txt",sep = "\t",header = T)
head(x)
x$fcsign <- sign(x$log2.fold_change.)
x$logP=-log10(x$p_value)
x$metric= x$logP/x$fcsign
y<-x[,c("Gene", "metric")]
head(y)
write.table(y,file="DE_genes.rnk",quote=F,sep="\t",row.names=F)
```

[tt为exactTest检出的针对表型差异的所有基因]
[https://github.com/BaderLab/Cytoscape_workflows/blob/master/EnrichmentMapPipeline/supplemental_protocol1_rnaseq.Rmd]

https://github.com/BaderLab/Cytoscape_workflows

https://baderlab.github.io/Cytoscape_workflows/EnrichmentMapPipeline/Protocol2_createEM.html

```
### Create GSEA input list
8b. Create a two-column rank (.RNK) file of all gene IDs and corresponding scores to for GSEA pre-ranked analysis. To run GSEA
```{r}
#calculate ranks
ranks_RNAseq = sign(tt$table$logFC) * -log10(tt$table$PValue)
```

[针对GSEA软件][https://www.biostars.org/p/266073/]：
```

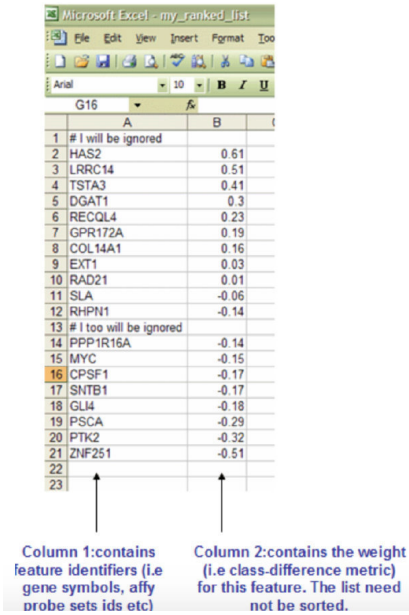
1. get a table with the list of genes on the row and log2FoldChange, p-value, and adj p-values on the column
2. order the gene list by a metric -log10(p-value)*sign(logFC) and create rank file (.rnk) in R
3. load this file to GSEA software and run GSEApreranked after choosing required and basic fields (making sure enrichment statistic is "classic")

RNA: Ranked list file format(*.rnk)

R语言构建rnk文件

RNK: Ranked list file format (*.rnk)

The RNK file contains a single, rank ordered gene list (*not* gene set) in a simple newline-delimited text format. It is used when you have a pre-ordered ranked list that you want to analyze with GSEA. For instance, you might have used your favorite tTest-like statistic to produce a ranked ordered gene list from your dataset which you now want to test for enrichment. Order of lines does not matter. It is important, however, that the second column will have numeric values - they will be used to rank order genes by GSEA.

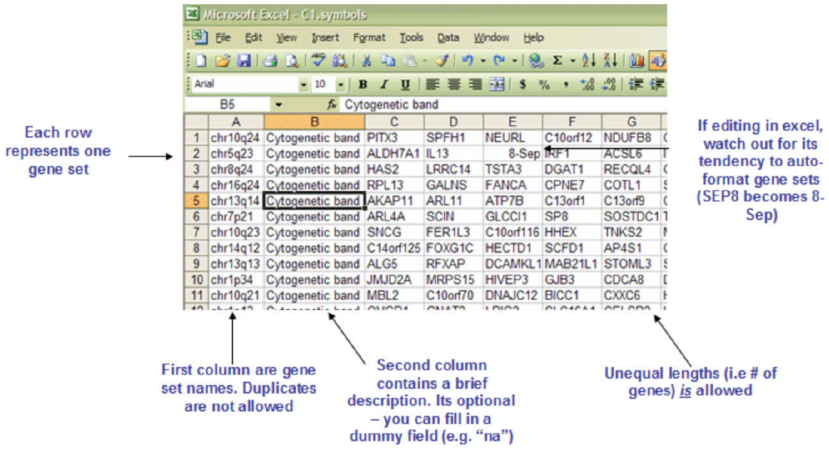



```
> head(gene_rnk_gsea)
    gene      score
1 B7H1K5 -2.2588818
2 Q6RYW5  0.0646151
3 B7I7S1 -3.1845306
4 B2I1Z2 -2.6235386
5 B7GW05 -3.4405334
6 B7GW27 -3.4051282
```

linux针对GO和gene关系构建gmt文件

GMT: Gene Matrix Transposed file format (*.gmt)

The GMT file format is a tab delimited file format that describes gene sets. In the GMT format, each row represents a gene set; in the GMX format, each column represents a gene set. The GMT file format is organized as follows:

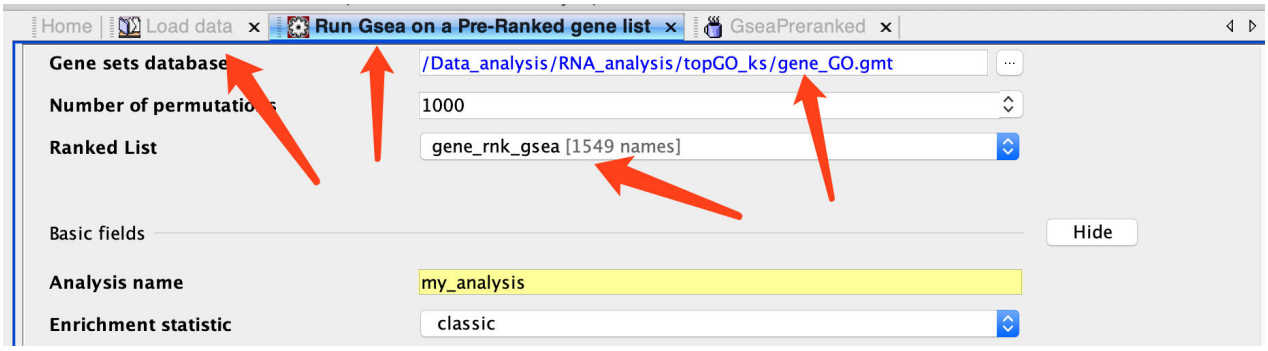


GMT format is convenient to store large databases of gene sets. For a handful of sets (<256) the gmx format offers greater excel-editability

Each gene set is described by a name, a description, and the genes in the gene set. GSEA uses the description field to determine what hyperlink to provide in the report for the gene set description: if the description is "na", GSEA provides a link to the named gene set in MSigDB; if the description is a URL, GSEA provides a link to that URL.

```
 1 GO:1905887      na      A6TEB8
 2 GO:0043186      na      Q0V8H6
 3 GO:0003910      na      P50844
 4 GO:0009992      na      P77338
 5 GO:0003911      na      A6TC47  A6TFP6  B0VU02  B7GZ71
 6 GO:0002134      na      Q88QT2
 7 GO:0047070      na      B5XQJ0
 8 GO:0003916      na      P13658
 9 GO:0003917      na      P14294  P45771  P0A2I2  Q9HZJ5
10 GO:0003918      na      P41513  P0A2I6  P0AFI3  O51859  P37411  Q9HUJ8  Q9HUK1  P0AES7
11 GO:0003919      na      P0AG42
```

GSEA analysis



[结果解释][http://www.gsea-msigdb.org/gsea/doc/GSEAUserGuideFrame.html?_Interpreting_GSEA_Results]

- ES: Enrichment Scores

Gene Set Enrichement Analysis的主要结果是enrichment score(ES)，该值反映了一个gene set在排秩序后的gene list中位于top或bottom中过表达的程度。GSEA沿着排序后的gene list移动，遇到位于gene set中的gene时增加running-sum，否则减少。因此，running-sum增加的级别和基因与表型相关性有关。ES为在gene list中获得和zero最大的偏差值。正数值表示该gene set在排序后的top富集，负数值表示该gene set在排序后的bottom的富集。

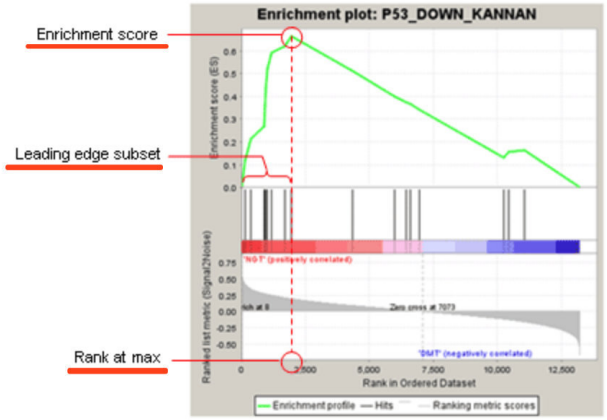


Fig 1: Enrichment plot: P53_DOWN_KANNAN
Profile of the Running ES Score & Positions of GeneSet Members on the Rank Ordered List

图中第一部分展示了沿着排序后gene list移动时，ES的变化；图中第二部分展示了gene set中的gene在排序后的gene list中位置，gene set 的 leading edge subset为对ES贡献最大的gene set部分；图中第三部分展示了沿着排序后gene list移动时ranking metric 值变化情况，正数值表示和表型相关，负数值表示与表型无关或负相关。

- Normalized Enrichement Score(NES)

NES值为解释gene set富集结果的主要统计值。通过标准化富集值，GSEA标准化gene set大小和gene sets与表达数据之间的差异，因此标准化后的富集值能用于在不同的gene set间比较:

$$NES = \frac{\text{actual ES}}{mean(\text{ESs against all permutations of the dataset})}$$

permutation: [/ˈpɜːmjʊˈteɪʃən/] (一组事物可能的一种)序列, 排列; 排列中的任一组数字或文字 NES是基于针对所有数据的排列的gene set富集值, 因此, 改变排列方式, 排列数目或表达数据大小都会影响NES值。考虑两种分析: 分析表达数据，GSEA生成了ranked list且分析ranked list；使用GSEAPreranked分析由第一种分析生成的ranked list。若使用相同的参数设置，得到富集值是一致的。然而，NES会反映不同数据用于排序的差异(the expression dataset versus the ranked list of genes)

	Expression Dataset		Ranked List	
Gene Set Name	**ES**	**NES**	**ES**	**NES**
BRENTANI_DNA_MET_AND_MOD	0.1233649	0.37071982	0.1233649	0.42405358
BRCA_BRCA1_NEG	0.13040805	0.6497973	0.13040808	0.6497975
PENG_RAPAMYCIN_DOWN	0.14286387	0.84542555	0.14286387	0.76681024
BASSO_REGULATORY_HUBS_SET	0.14299561	0.6870111	0.14299563	0.69177157
VENTRICLES_UP	0.14565612	0.7033464	0.14565612	0.6915998
ALANINE_AND_ASPARTATE_METABOLISM	0.14693332	0.422703	0.14693332	0.36949828
BRCA1_OVEREXP_DN	0.15077576	0.7929205	0.15077576	0.68026066

Analysis parameters: P53_hgu95av2.gct, P53.cls#MUT_versus_WT, c2.may_2006.symbols.gmt, permutation type = gene_set, seed for permuation = 149, number of permutations = 10

- False Discovery Rate(FDR)

FDR为针对gene set在给定NES时为假阳性的可能性。例如, FDR为25%表明75%的结果是有效的。**GSEA报告最高的同时含小于25%的ES值的gene set为感兴趣结果，同时针对给gene set进行下一步分析，但是针对所有gene set都提供分析结果。**(However, if you have a small number of samples and use gene_set permutation (rather than phenotype permutation) for your analysis, you are using a less stringent assessment of significance and would then want to use a more stringent FDR cutoff, such as 5%.)

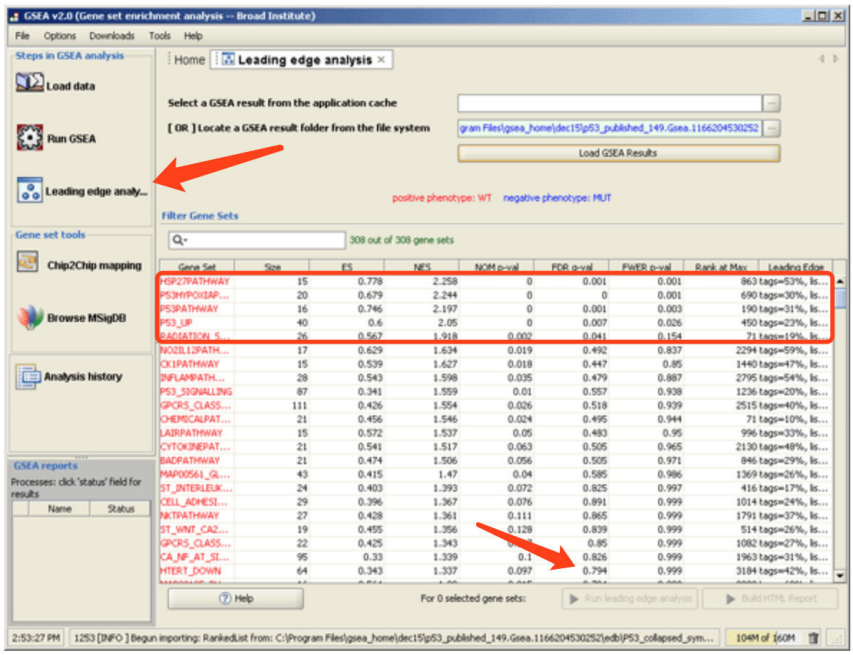
- Nominal P Value

名义上的p值评估了单个gene set富集值的显著程度。FDR值根据gene set大小和多重假设检验进行调整，而p值没有。**当一个top gene set拥有一个小的p值和高的FDR值时，则表示根据empirical null分布，该gene set和其他gene sets相比并没有那么显著。这可能因为，没有足够的样本, biological signal 微弱，或该gene sets并不能很好的解释生物问题。另一方面，FDR是基于所有gene sets的两分布，假如许多gene sets中的一个富集，同时该gene set可能拥有很高的FDR值。最终，拥有很好名义p值和低的FDR值的top gene set一般表示阴性结果: 该gene set本身不够显著同时其他gene sets 显著性更差。**(In the "Interpreting GSEA Results" section, under "Nominal P Value", the last paragraph states: In the GSEA report, a p value of zero (0.0) indicates an actual p value of less than 1/number-of-permutations. For example, if the analysis performed 100 permutations, a reported p value of 0.0 indicates an actual p value of less than 0.01. For a more accurate p value, increase the number of permutations performed by the analysis. Typically, you will want to perform 1000 permutations (phenotype or gene_set). (If you attempt to perform significantly more than 1000 permutations, GSEA may run out of memory.) [参考文章][http://www.gsea-msigdb.org/gsea/doc/subramanian_tamayo_gsea_pnas.pdf]

- No enriched gene sets of significance may indicate that, in fact, no gene sets are enriched. It may also be that you are analyzing too few samples, the biological signal in question is subtle, or the gene sets that you are analyzing do not represent the biology in question very well. You may still want to look at the top ranked gene sets, keeping in mind that these results provide weak evidence for potentially interesting hypotheses. You might also want to consider analyzing other gene sets or, if possible, additional samples.
- Too many enriched gene sets of significance may indicate that, in fact, many gene sets are enriched between phenotypes. Perhaps the gene sets represent the same biological signal. You can check for this by looking for overlap in the leading-edge subsets within the gene sets (see Running a Leading Edge Analysis). Or, you might be seeing significant differences between the phenotypes due to technical artifacts, such as samples being run in different labs, by different operators, or against different arrays. As with too few enriched gene sets, you may still want to look at the top ranked gene sets, keeping in mind that these results provide potentially biased evidence for interesting hypotheses. You might also want to consider analyzing other gene sets or, if possible, additional samples.

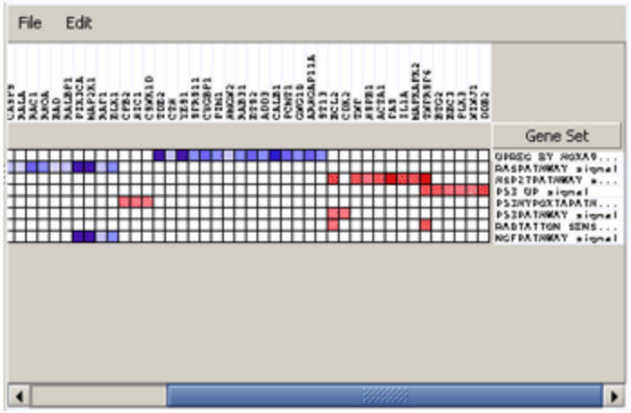
Running a Leading Edge Analysis

Leading-edge subset为位于ES值前的gene集合，leading-edge subset可解释为解释gene set富集信号的核心genes。运行完gene set enrichment analysis后，使用leading edge analysis来检测位于富集的gene sets的leading-edge subset中的genes。位于多个leading-edge subset中的gene更可能是感兴趣的gene。可选择正调控或负调控的多个gene sets(GO items)进行分析:



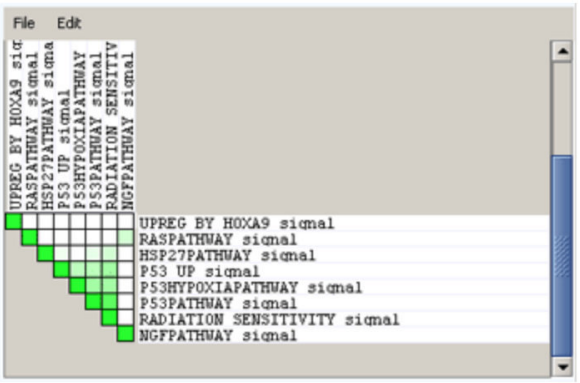
[结果解释][http://www.gsea-msigdb.org/gsea/doc/GSEAUserGuideFrame.html?_Interpreting_GSEA_Results]

- Heat Map



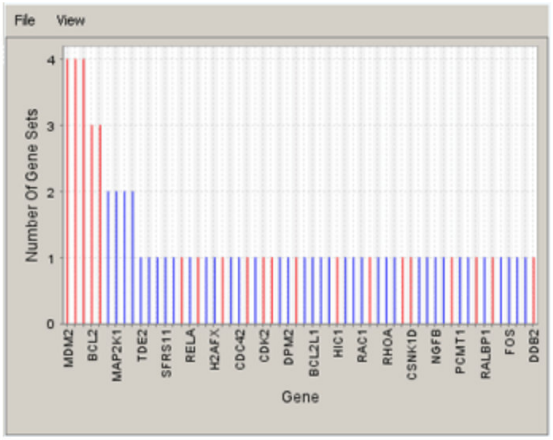
热图展示了位于leadng-edge subsets中的genes(clustered)。改图使用颜色(red, pink, light blue, dark blue)表示表达值(expression values)的范围(high, moderate, low, lowest)。

- Set-to-Set



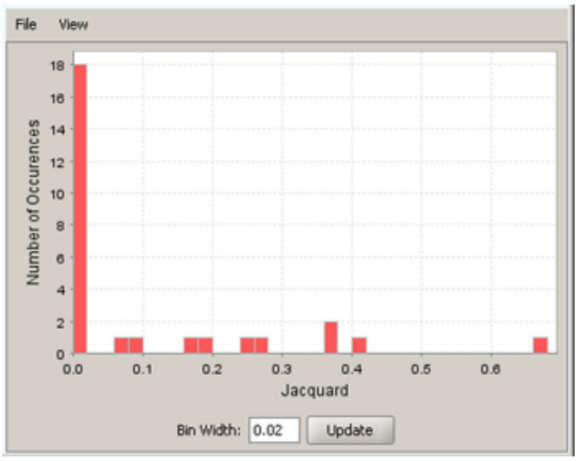
改图使用颜色强度来表示subsets间的重叠程度。颜色越深，重叠程度越强。

- Gene in Subsets



改图展示了每个基因以及它所出现的subsets的数目

- Histogram



Jacquard is the intersection divided by the union for a pair of leading edge subsets。Number of Occurences is the number of leading edge subset pairs in a particular bin。略!

[fgsea][https://github.com/ctlab/fgsea]

1. 构建genes列表

基因排列: `-sign(avg_fold)*log10(pvalue)`

```
> head(gene_ranks)
IX87_RS00010 IX87_RS00035 IX87_RS00040 IX87_RS00130 IX87_RS00135 IX87_RS00195
   0.19042445   0.36984408   0.08686197   0.08686197   0.39015684  -0.25840699
```

2. 构建注释文件

```
$`map00230_Purine metabolism`
 [1] "IX87_RS09205" "IX87_RS12380" "IX87_RS14260" "IX87_RS14265" "IX87_RS08640" "IX87_RS18600"
 [7] "IX87_RS13855" "IX87_RS09315" "IX87_RS04625" "IX87_RS06755" "IX87_RS06340" "IX87_RS09200"
[13] "IX87_RS16620" "IX87_RS15485" "IX87_RS08600" "IX87_RS05020"

$`map02010_ABC transporters`
 [1] "IX87_RS03490" "IX87_RS03485" "IX87_RS03495" "IX87_RS19330" "IX87_RS19325" "IX87_RS17540"
 [7] "IX87_RS08665" "IX87_RS08675" "IX87_RS08670" "IX87_RS08680" "IX87_RS09125" "IX87_RS09115"
[13] "IX87_RS09120" "IX87_RS09105" "IX87_RS11980" "IX87_RS11975" "IX87_RS11970" "IX87_RS02710"
```

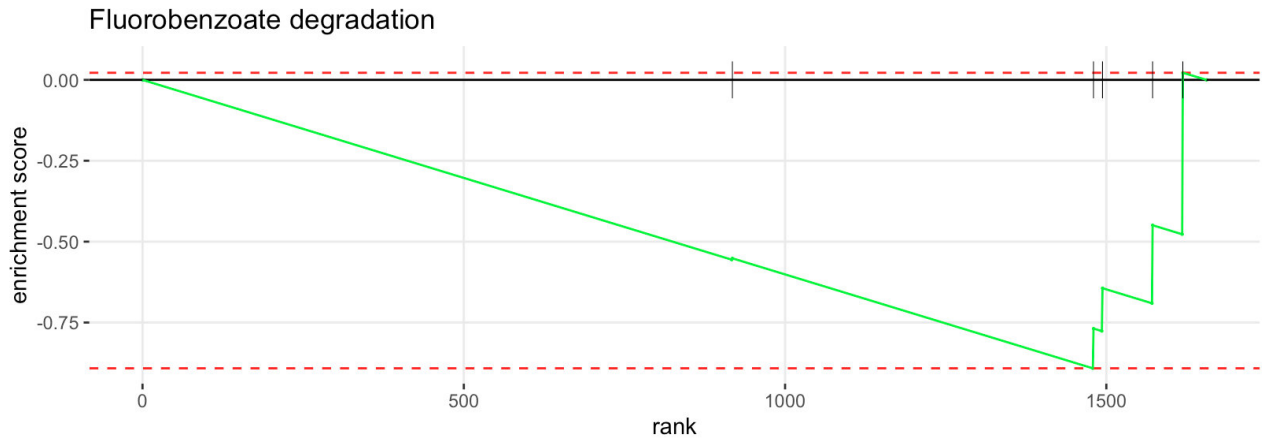
3. 分析

`fgseaMultilevelRes <- fgseaMultilevel(pathways=Ann_list, stats = gene_ranks, minSize=5, maxSize=500)`

`head(fgseaMultilevelRes[order(pval),])`

```
> head(fgseaMultilevelRes[order(pval), ])
                                       pathway       pval      padj   log2err         ES
1:                map00364_Fluorobenzoate degradation 0.03969754 0.4961240 0.3153248 -0.8914761
2: map00280_Valine, leucine and isoleucine degradation 0.04160000 0.4961240 0.2820134 -0.7494071
3:                      map00340_Histidine metabolism 0.04651163 0.4961240 0.3077500  0.8288385
4:                map03420_Nucleotide excision repair 0.09936575 0.6190955 0.2065879  0.7942063
5:                       map01100_Metabolic pathways 0.10727497 0.6190955 0.1482615 -0.5137191
6:                       map00680_Methane metabolism 0.15856237 0.6190955 0.1608014  0.7542499
        NES size                                                                leadingEdge
1: -1.492795    5                                IX87_RS04450,IX87_RS04460,IX87_RS04465,IX87_RS04455
2: -1.502855   15 IX87_RS03335,IX87_RS03990,IX87_RS03330,IX87_RS03645,IX87_RS19200,IX87_RS19205,...
3:  1.454050    5                                IX87_RS13745,IX87_RS13765,IX87_RS13760,IX87_RS13750
4:  1.393294    5                                             IX87_RS15790,IX87_RS08975,IX87_RS16450
5: -1.292169   66 IX87_RS19255,IX87_RS18140,IX87_RS19200,IX87_RS04995,IX87_RS00860,IX87_RS19205,...
6:  1.323197    5                                                          IX87_RS13990,IX87_RS13685
```

```
plotEnrichment(pathway=Ann_list[["map00364_Fluorobenzoate degradation"]],
stats=gene_ranks, gseaParam = 1) + labs(title="Fluorobenzoate degradation")
```




```
topPathwaysUp <- fgseaMultilevelRes[ES > 0][head(order(pval), n =10), pathway]

topPathwaysDown <- fgseaMultilevelRes[ES < 0][head(order(pval), n=10), pathway]

topPathways <- c(topPathwaysUp, rev(topPathwaysDown))

plotGseaTable(Ann_list[topPathways], gene_ranks, fgseaMultilevelRes,
gseaParam=0.5)
```

Pathway	Gene ranks	NES	pval	padj
map00340_Histidine metabolism		1.45	4.7e-02	5.0e-01
map03420_Nucleotide excision repair		1.39	9.9e-02	6.2e-01
map00680_Methane metabolism		1.32	1.6e-01	6.2e-01
map00983_Drug metabolism - other enzymes		1.23	2.5e-01	6.2e-01
map00240_Pyrimidine metabolism		1.22	2.5e-01	6.2e-01
map00010_Glycolysis / Gluconeogenesis		1.18	3.1e-01	6.2e-01
map00620_Pyruvate metabolism		1.06	3.9e-01	6.2e-01
map01212_Fatty acid metabolism		0.94	5.2e-01	7.2e-01
map00760_Nicotinate and nicotinamide metabolism		0.87	6.2e-01	7.9e-01
map00660_C5-Branched dibasic acid metabolism		0.83	7.0e-01	7.9e-01
map00920_Sulfur metabolism		-1.15	3.4e-01	6.2e-01
map03430_Mismatch repair		-1.16	3.2e-01	6.2e-01
map00330_Arginine and proline metabolism		-1.19	2.7e-01	6.2e-01
map00450_Selenocompound metabolism		-1.24	2.5e-01	6.2e-01
map02010_ABC transporters		-1.21	2.3e-01	6.2e-01
map00550_Peptidoglycan biosynthesis		-1.33	1.7e-01	6.2e-01
map00627_Aminobenzoate degradation		-1.34	1.6e-01	6.2e-01
map01100_Metabolic pathways		-1.29	1.1e-01	6.2e-01
ap00280_Valine, leucine and isoleucine degradation		-1.50	4.2e-02	5.0e-01
map00364_Fluorobenzoate degradation		-1.49	4.0e-02	5.0e-01

0 400 800 1200 1600

collapsedPathways:

```
collapsedPathways <- collapsePathways(fgseaRes[order(pval)][padj < 0.01],
                                      examplePathways, exampleRanks)
```

Fwrite:

```
fwrite(fgseaRes, file="fgseaRes.txt", sep="\t", sep2=c("", " ", ""))
```

[pathview] [http://www.bioconductor.org/packages/release/bioc/html/pathview.html] : pathway based data integration and visualization

Pathview是基于数据整合和可视化的通路工具组合. 根据用户提供的基因和化合物数据比对到指定的通路上. Pathview自该动下载通路图像数据, 解析数据文件, 比对用户提供的数据. 同时提供KEGG原始的图形和Graphviz图形(keggview.native and kegg.graph are the two viewer functions, and pathview is the main function proving a unified interface to downloader, parser, mapper and viewer functions).

Parameter

```
pathview(gene.data = NULL, cpd.data = NULL, pathway.id,
species = "hsa", kegg.dir = ".", cpd.idtype = "kegg", gene.idtype =
"entrez", gene.annotpkg = NULL, min.nnodes = 3, kegg.native = TRUE,
map.null = TRUE, expand.node = FALSE, split.group = FALSE, map.symbol =
TRUE, map.cpdname = TRUE, node.sum = "sum", discrete=list(gene=FALSE,
cpd=FALSE), limit = list(gene = 1, cpd = 1), bins = list(gene = 10, cpd
= 10), both.dirs = list(gene = T, cpd = T), trans.fun = list(gene =
NULL, cpd = NULL), low = list(gene = "green", cpd = "blue"), mid =
list(gene = "gray", cpd = "gray"), high = list(gene = "red", cpd =
"yellow"), na.col = "transparent", ...)
```

`gene.data`: 单个样本向量或多个样本矩阵. 向量需要为数字, 同时gene IDs为其名称(基因IDs也可以是字符). gene IDs为遗传概念, 包含多重唯一比对到KEGG gene IDs的gene/transcript/protein名称类型.

`gene.idtype`: gene.data数据ID类型

`cpd.data`: 同`gene.data`, 但是名称为可比对KEGG compound IDs的IDs名称

`cpm.idtype`: cpd.data数据ID类型

`pathway.id`: KEGG pathway ID.

`species`: 物种. kegg code/common name

`kegg.dir`: KEGG pathway data file(.xml)和image file(.png)路径

`kegg.native`: 是否提供原始KEGG图像(`.png`)/ graphvie为`.pdf`

`expand.node`: 是否将多重基因nodes扩展为单个基因node, 默认FLASE

`split.group`: 是否将多重node groups分成单个的node(each split member nodes inherits all edges from the node group).

`map.symbol`: 是否比对gene IDs到symbols来表示节点(This option is only effective for kegg.native=FALSE or same.layer=FALSE when kegg.native=TRUE)

`map.cpdname`: 是否比对compoud IDs到正式名称来表示compound节点标签, 或使用来自KGML文件(kEGG compound accessions)的图像名称(This option is only effective for kegg.native=FALSE)

`node.sum`: 当多重genes或compouds比对到一个节点时, 用于计算node summary的方法: 'sum', 'mean', 'median', 'max', max.abs', 'random'. 默认为node.sum='sum'

`sign.pos`: 控制通路签名的位置: bottomleft/bottomright/topleft/topright

`key.pos`: 控制颜色图例位置: bottomleft/bottomright/topleft/topright

...

Value

`kegg.names`: 比对到节点的标准KEGG IDs/Names. 为Entez Gene ID或KEGG Compound Accessions

`labels`: 节点标签

`all.mapped`: 所有比对到该节点的所有分子(gene or compound) IDs

`type`: 节点类型, 当前4种: 'gene', 'enzyme', 'compound', 'ortholog'

`x`: 原始KEGG通路图的x坐标

`y`: 原始KEGG通路图的y坐标

`width`: 原始KEGG通路图的node宽度

`height`: 原始KEGG通路图的node高度

`other columns`: 比对的gene/compound数据的列, 和对应样本的pseudo-color codes

Tutorial

`pathview`自动下载通路图像数据(pathway graph data), 解析数据文件, 在通路(pathway)上比对用户的数据, 同时提供包含比对了数据后到通路图像.

`pathview`可分为4个功能模块: the Downloader, Parser, Mapper, Viewer. 最重要的是, `pathview`比对并提交用户的数据到相关的通路图像上.

当前`pathview`仅能使用KEGG的通路数据.

`pathview`对于数据整合提供了强有力支持. 1) 包含可比对到通路的, 所有必要的生物数据类型; 2) 支持超过10中基因或蛋白ID类型, 20种化合物或代谢物ID类型; 3) 包含约4800个物种和KEGG orthology的通路; 4) 支持多种数据属性和格式, 例如, 连续/离散数据, 矩阵/向量, 单个/多个样本等.

查看pathview所支持的KEGG物种和其默认gene IDs

```
data(korg)
```

```
head(korg)
```

```
     ktax.id  tax.id  kegg.code scientific.name
[1,] "T01001" "9606"  "hsa"     "Homo sapiens"
[2,] "T01005" "9598"  "ptr"     "Pan troglodytes"
[3,] "T02283" "9597"  "pps"     "Pan paniscus"
[4,] "T02442" "9595"  "ggo"     "Gorilla gorilla gorilla"
[5,] "T01416" "9601"  "pon"     "Pongo abelii"
[6,] "T03265" "61853" "nle"     "Nomascus leucogenys"
     common.name                    entrez.gnodes kegg.geneid ncbi.geneid
[1,] "human"                        "1"           "374659"    "374659"
[2,] "chimpanzee"                   "1"           "474020"    "474020"
[3,] "bonobo"                       "1"           "100989900" "100989900"
[4,] "western lowland gorilla"      "1"           "101125212" "101125212"
[5,] "Sumatran orangutan"           "1"           "100172878" "100172878"
```

Bioconductor支持的注释物种包

```
data(bods)
```

```
bods
```

```
      package             species               kegg code id.type
 [1,] "org.Ag.eg.db"      "Anopheles"           "aga"     "eg"
 [2,] "org.At.tair.db"    "Arabidopsis"         "ath"     "tair"
 [3,] "org.Bt.eg.db"      "Bovine"              "bta"     "eg"
 [4,] "org.Ce.eg.db"      "Worm"                "cel"     "eg"
 [5,] "org.Cf.eg.db"      "Canine"              "cfa"     "eg"
 [6,] "org.Dm.eg.db"      "Fly"                 "dme"     "eg"
 [7,] "org.Dr.eg.db"      "Zebrafish"           "dre"     "eg"
 [8,] "org.EcK12.eg.db"   "E coli strain K12"   "eco"     "eg"
 [9,] "org.EcSakai.eg.db" "E coli strain Sakai" "ecs"     "eg"
[10,] "org.Gg.eg.db"      "Chicken"             "gga"     "eg"
```

```
BiocManager::install("pathview")
```

```
BiocManager::install(c("Rgraphviz", "png", "KEGGgraph", "org.Hs.eg.db"))
```

加载查看帮助:

```
library(pathview)
```

```
library(help=pathview)
```

一般可视化操作:

```
filename <- system.file("extdata/gse16873.demo", package="pathview")
```

```
gse16873 <- read.delim(filename,row.names=1)
```

```
gse16873.d <- gse16873[,2*(1:6)] - gse16873[,2*(1:6)-1]
```

```
data(demo.paths)
```

```
i <- 1
```

```
pv.out <- pathview(gene.data = gse16873.d[,1], pathway.id =
demo.paths$sel.paths[i], species = "hsa", out.suffix = "gse16873", kegg.native
=TRUE)
```

```
Warning: None of the genes or compounds mapped to the pathway!
Argument gene.idtype or cpd.idtype may be wrong.
'select()' returned 1:1 mapping between keys and columns
Info: Working in directory /Data_analysis/RNA_analysis/ab_cs_11_hq_1
Info: Writing image file hsa04110.gse16873.png
```

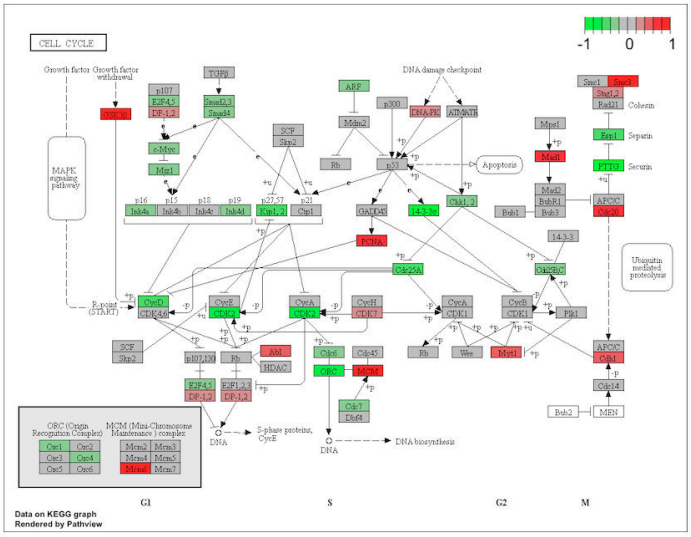
根据返回结果重构`gene.data`文件, 对应提供向量名称, gene IDs:

```
tmp <- head(gse16873.d[,1],92)
```

```
names(tmp) <- pv.out$plot.data.gene$kegg.names
```
#(gene.idtype="ENTREZID")或者

```
#names(tmp) <- pv.out$plot.data.gene$labels
```
#(gene.idtype="SYMBOL")

```
pv.out <- pathview(gene.data = tmp, gene.idtype="ENTREZID", pathway.id =
demo.path$sel.paths[i], species="hsa",out.suffix="gse16873",kegg.native=TRUE)
```

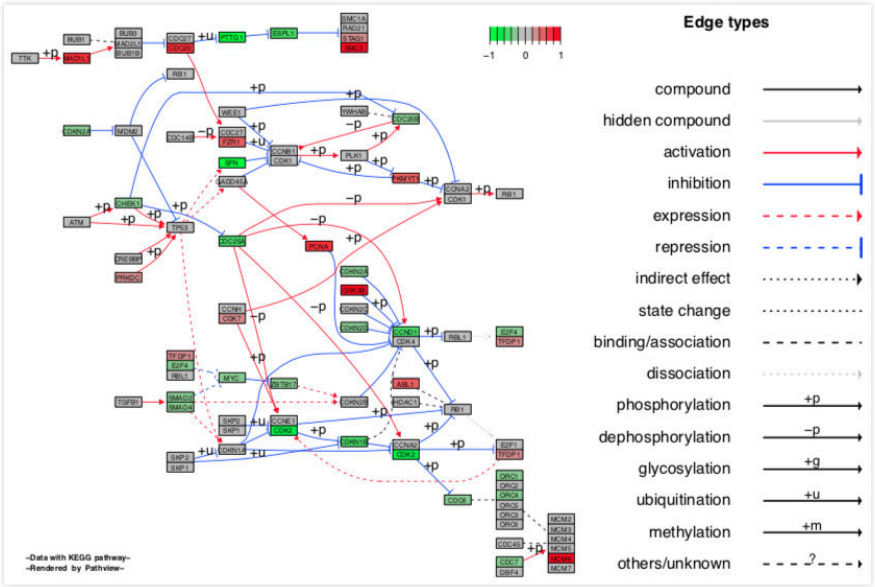



```
str(pv.out)
```

```
List of 2
 $ plot.data.gene:'data.frame': 92 obs. of  10 variables:
  ..$ kegg.names: chr [1:92] "1029" "51343" "4171" "4998" ...
  ..$ labels    : chr [1:92] "CDKN2A" "FZR1" "MCM2" "ORC1" ...
  ..$ all.mapped: chr [1:92] "1029" "51343" "4171,4172,4173,4174,4175,4176" "4998,4999,500
0,5001,23594,23595" ...
  ..$ type      : chr [1:92] "gene" "gene" "gene" "gene" ...
  ..$ x         : num [1:92] 532 919 553 494 919 919 188 432 123 77 ...
  ..$ y         : num [1:92] 124 536 556 556 297 519 519 191 704 687 ...
  ..$ width     : num [1:92] 46 46 46 46 46 46 46 46 46 46 ...
  ..$ height    : num [1:92] 17 17 17 17 17 17 17 17 17 17 ...
  ..$ mol.data  : num [1:92] -0.3076 0.4159 0.8594 -0.6382 -0.0449 ...
  ..$ mol.col   : Factor w/ 9 levels "#00FF00","#30EF30",..: 4 7 9 2 5 5 5 5 5 5 ...
 $ plot.data.cpd : NULL
```

或者使用Graphviz显示de novo pathway graph, 图像拥有相同的nodes和edges, 但是使用不用的形式显示:

```
pv.out <- pathview(gene.data="tmp", pathway.id=demo.paths$sel.paths[i],
species="hsa", out.suffix="gse16873", kegg.native=F, sign.pos="bottomleft")
```



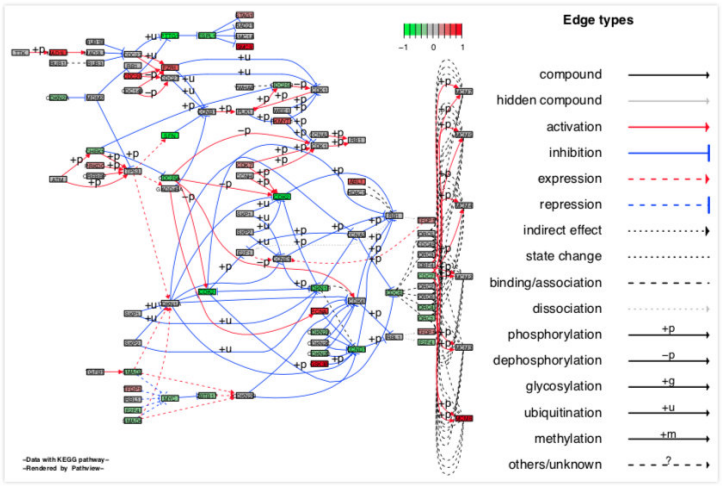
可以通过设置参数`same.layer=FASLE`将图和图例显示在不同的页面, 略!

在原始的KEGG view, 一个基因node可能代表具有相似或重复功能角色的基因/蛋白. 将其当成一个node来对待是为了更清晰地显示在图中.

在使用Graphviz查看时, 可将node groups分为individual detached nodes; 或者将multiple-gene nodes扩展到individual genes, 这两种方式都会继承unsplit group/unexpanded nodes的edges信息.

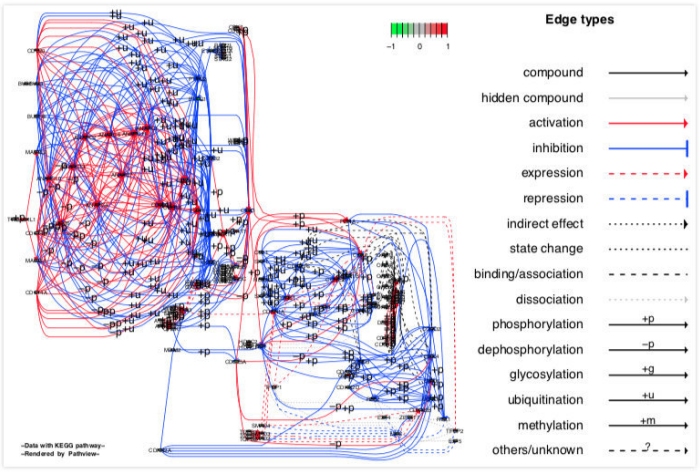
默认将涉及到相同反应的基因自动聚集到一起, split.group=FALSE

```
pv.out <- pathview(gene.data=tmp, pathway.id=demo.paths$sel.paths[i], species="hsa", out.suffix="ges16873.split", kegg.native=F, sign.pos="bottomleft", split.group=TURE)
```



扩展multiple-gene nodes到individual genes

```
pv.out <- pathview(gene.data=tmp, pathway.id=demo.paths$sel.paths[i], species="hsa", out.suffix="gse16873.split.expanded", kegg.native=FALSE, sign.pos="bottomleft", split.group=TRUE, expand.nodes=TREU)
```




```
str(pv.out)
```

```
List of 2
 $ plot.data.gene:'data.frame': 124 obs. of  10 variables:
  ..$ kegg.names: chr [1:124] "1029" "51343" "4171" "4172" ...
  ..$ labels    : chr [1:124] "CDKN2A" "FZR1" "MCM2" "MCM3" ...
  ..$ all.mapped: chr [1:124] "1029" "51343" "4171" "4172" ...
  ..$ type      : chr [1:124] "gene" "gene" "gene" "gene" ...
  ..$ x         : num [1:124] 532 919 553 553 553 553 553 553 494 494 ...
  ..$ y         : num [1:124] 124 536 556 556 556 556 556 556 556 556 ...
  ..$ width     : num [1:124] 46 46 46 46 46 46 46 46 46 46 ...
  ..$ height    : num [1:124] 17 17 17 17 17 17 17 17 17 17 ...
  ..$ mol.data  : num [1:124] -0.3076 0.4159 0.1985 -0.0119 -0.177 ...
  ..$ mol.col   : Factor w/ 10 levels "#00FF00","#30EF30",..: 4 7 5 5 5 5 9 5 5 5 ...
 $ plot.data.cpd : NULL
```

pathview拥有强大的数据整合能力, 可用于整合, 分析, 可视化多种生物数据: 基因表达, 蛋白表达, 遗传相关性, 代谢, 基因组数据, 文献和其他可以比对到通路的数据类型

化合物和基因数据

查看代谢通路, 因此除了gene nodes外, 还有compound nodes. 因此在代谢通路中整合gene和compound数据.

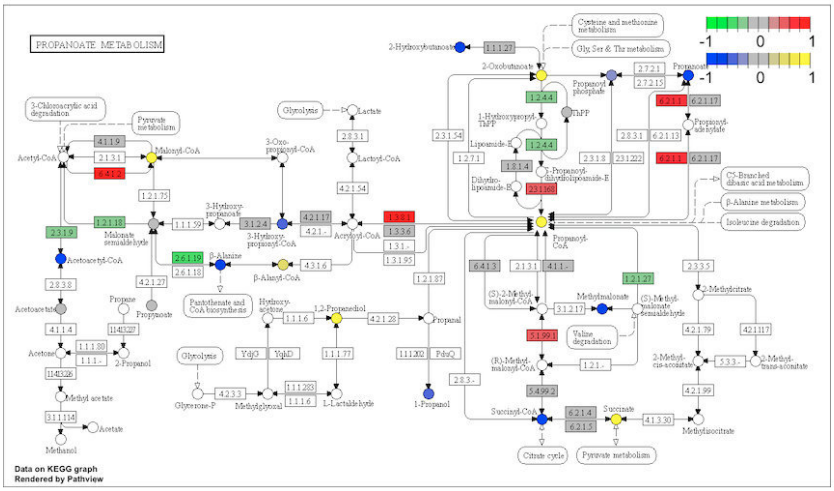
```
sim.mol.data(mol.type = c("gene", "gene.ko", "cpd")[1], id.type = NULL,
species="hsa", discrete = FALSE, nmol = 1000, nexp = 1, rand.seed=100)
```

`sim.cpd.data <- sim.mol.data(mol.type="cpd",nmol=3000)` #模拟cpd数据

```
tmp <- sample(tmp,25)
```

```
names(tmp) <- pv.out$plot.data.gene$kegg.names
```

```
pv.out <- pathview(gene.data=tmp, cpd.data=sim.cpd.data,
pathway.id=demo.paths$sel.paths[3], speceis="hsa", out.suffix="gse16873.cpd",
keys.align="y", kegg.native=TREU, key.pos="topright")
```

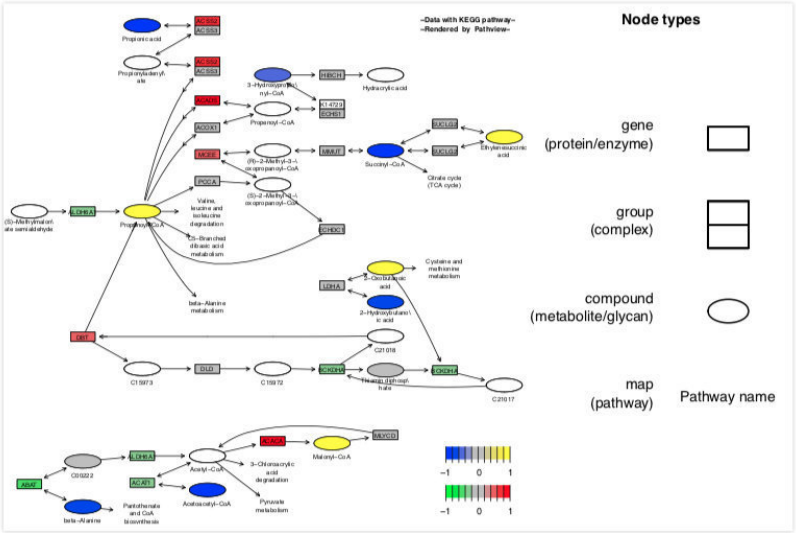



```
str(pv.out)
```

```
List of 2
 $ plot.data.gene:'data.frame': 25 obs. of  10 variables:
  ..$ kegg.names: chr [1:25] "4329" "31" "23417" "18" ...
  ..$ labels    : chr [1:25] "ALDH6A1" "ACACA" "MLYCD" "ABAT" ...
  ..$ all.mapped: chr [1:25] "4329" "31" "23417" "18" ...
  ..$ type      : chr [1:25] "gene" "gene" "gene" "gene" ...
  ..$ x         : num [1:25] 159 159 159 276 377 ...
  ..$ y         : num [1:25] 325 252 204 378 327 409 494 390 390 229 ...
  ..$ width     : num [1:25] 46 46 46 46 46 46 46 46 46 46 ...
  ..$ height    : num [1:25] 17 17 17 17 17 17 17 17 17 17 ...
  ..$ mol.data  : num [1:25] -0.2786 1.0392 -0.0897 -0.438 -0.1608 ...
  ..$ mol.col   : Factor w/ 6 levels "#5FDF5F","#8FCE8F",..: 2 6 3 1 3 2 4 3 3 3 ...
 $ plot.data.cpd :'data.frame': 48 obs. of  10 variables:
  ..$ kegg.names: chr [1:48] "C00222" "C00804" "C01013" "C00099" ...
  ..$ labels    : chr [1:48] "C00222" "C00804" "C01013" "C00099" ...
  ..$ all.mapped: chr [1:48] "C00222" "C00804" "" "C00099" ...
  ..$ type      : chr [1:48] "compound" "compound" "compound" "compound" ...
  ..$ x         : num [1:48] 225 222 324 325 222 ...
  ..$ y         : num [1:48] 327 449 327 388 228 105 105 105 157 228 ...
  ..$ width     : num [1:48] 8 8 8 8 8 8 8 8 8 8 ...
  ..$ height    : num [1:48] 8 8 8 8 8 8 8 8 8 8 ...
  ..$ mol.data  : num [1:48] 0.14 0.143 NA -0.638 1.053 ...
  ..$ mol.col   : Factor w/ 8 levels "#0000FF","#3030EF",..: 5 5 8 2 7 7 4 1 8 8 ...
```

使用Graphviz查看, `cpd.lab.offset` 指定高过默认compoud labels的程度

```
pv.out <- pathview(gene.data=tmp, cpd.data=sim.cpd.data,
pathway.id=demo.paths$sel.paths[3], speceis="hsa", out.suffix="gse16873.cpd",
keys.align="y", kegg.native=FALSE, key.pos="topright", cpd.lab.offset="-1")
```



多重条件或样本

```
sed.seed(10)
```

```
sim.cpd.data2 <- matrix(sample(sim.cpd.data, 18000, replace=T), ncol=6)
```

```
rownames(sim.cpd.data2) <- names(sim.cpd.data)
```

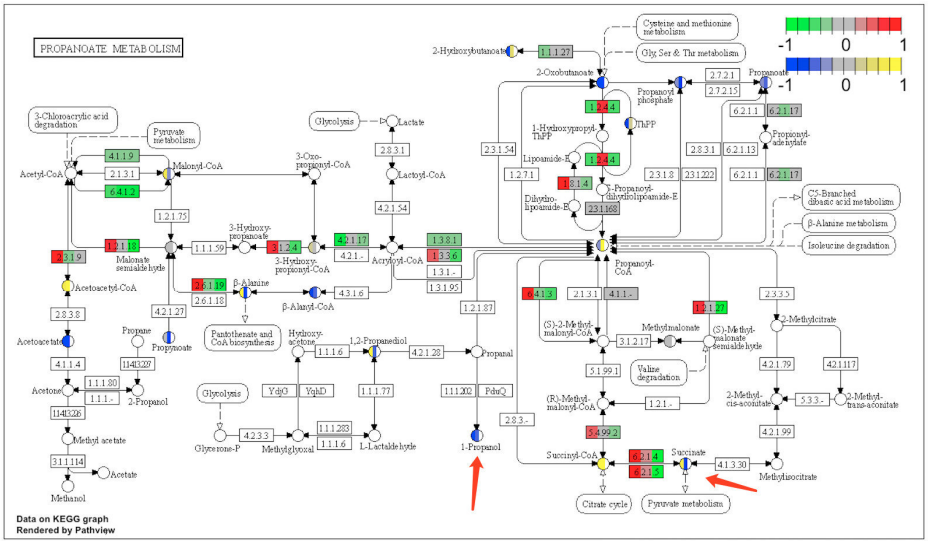
```
colnames(sim.cpd.data2) <- paste("exp", 1:6, sep="")
```

```
head(sim.cpd.data2, 3)
```

```
             exp1       exp2        exp3        exp4        exp5       exp6
C02787  0.62355826 -0.1108793  1.06939782 -0.95954034 1.653444849  1.360614
C08521 -1.23737070  0.4676360 -2.06425336 -0.65938385 0.004274093  0.512765
C01043 -0.01768295  0.5472769 -0.59238800 -0.11908824 0.950917578 -1.130288
C11496 -0.41272992 -0.4345066 -1.00197437 -2.07348891 0.499060037  1.093979
C07111 -0.30936407  0.6082035 -0.80780378  0.00939061 0.324284961  1.270100
C00031  0.28776830 -0.2776492  0.08823917 -0.61412075 2.003281574 -1.302442
>
```

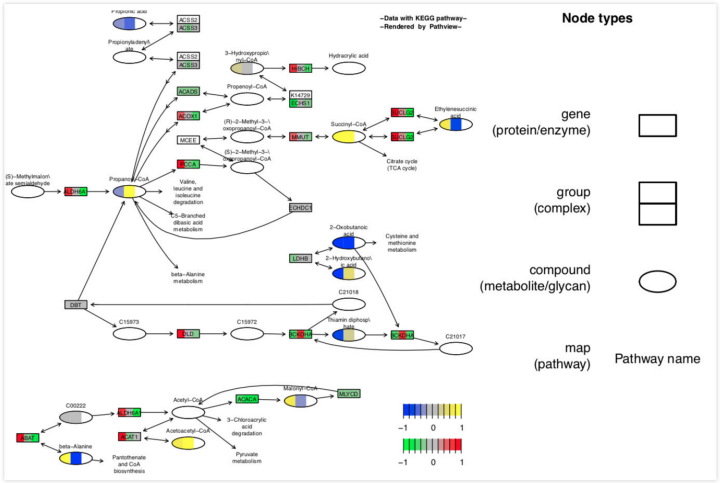
KEGG view with data match(建议使用`data.match=T`, 可使samples和compounds对应)

```
pv.out <- pathview(gene.data=gse16873.d[,1:3], cpd.data = sim.cpd.data2[,1:2],
pathway.id=demo.paths$sel.pathes[3], species="hsa", out.suffix="gse16873.cpd.3-
2s.match", keys.align="y", kegg.native=T, match.data=T, multi.state=T,
same.layer=T)
```



Graphviz view

```
pv.out <- pathview(gene.data = gse16873.d[,1:3], cpd.data=sim.cpd.data2[,1:2],
pathway.id=demo.paths$sel.paths[3], species="hsa",out.suffix="gse16873.cpd.3-2s",
keys.align="y", kegg.native=F, match.data=T, multi.state=T, same.layer=T,
key.pos="bottomright", sign.pos="topright")
```



离散型数据 例如, 基于一些统计检验(p-value, fold change, etc) 选择一组显著性genes或compounds. 输入数据可以命名为两个水平的向量, 1和0(显著或不显著), 或者是一小组显著性gene/compound名称

`discrete`: 指定两逻辑参数,gene/cpd, 该参数指定gene/cpd是否为离散值; 默认`discrete=list(gene=FALSE, cpd=FALSE)`

`limit`: 指定两数字单元, gene/cpd, 该参数指定gene.data/cpd.data转换为颜色时的限制数值. 长度为1时, 表示离散值/正值方向的数据, 或针对2个方向数据的绝对限制; 长度为2表示两个方向的数据. 默认为`limit=list(gene=1,cpd=1)`

`bins`: 两个整数单元的列表(gene/cpd), 该参数指定gene.data/cpd.data转换为颜色的水平. 默认为`bins=list(gene=10,cpd=10)`

```
require(org.Hs.eg.db)
```

```
gse16873.t <- apply(gse16873.d, 1, function(x)t.test(x,alternative="two sided")$p.value)
```

```
sel.genes <- names(gse16873.t)[gse16873.t < 0.1]
```

```
sel.cpds <- names(sim.cpd.data)[abs(sim.cpd.data) > 0.5]
```

```
> pv.out <- pathview(gene.data = sel.genes, cpd.data = sel.cpds,
+     pathway.id = demo.paths$sel.paths[i], species = "hsa", out.suffix = "sel.genes.sel.cpd",
+     keys.align = "y", kegg.native = T, key.pos = demo.paths$kpos1[i],
+     limit = list(gene = 5, cpd = 2), bins = list(gene = 5, cpd = 2),
+     na.col = "gray", discrete = list(gene = T, cpd = T))
> pv.out <- pathview(gene.data = sel.genes, cpd.data = sim.cpd.data,
+     pathway.id = demo.paths$sel.paths[i], species = "hsa", out.suffix = "sel.genes.cpd",
+     keys.align = "y", kegg.native = T, key.pos = demo.paths$kpos1[i],
+     limit = list(gene = 5, cpd = 1), bins = list(gene = 5, cpd = 10),
+     na.col = "gray", discrete = list(gene = T, cpd = F))
```