KEGGprofile: Application Examples

Shilin Zhao

April 5, 2015

Abstract

Abstract: In this vignette, we demonstrate the application of KEGGprofile as an annotation and visualization tool in analysis of multi-types and multi-groups high-throughput expression data. Superior to existing approaches, KEGGprofile combined the KEGG pathway map with expression profiles of genes in that pathway and facilitated more detailed analysis about the specific function changes inner pathway or temporal correlations in different genes and samples. Here we introduce the data preparation and functions used for pathway gene expression profile visualization.

1 Introduction

KEGG is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, from genomic and molecular-level information (http://www.kegg.jp/kegg/). The KEGG pathway database is composed by a lot of pathway maps focused on different biological functions, including metabolism, signal transduction, cellular process, and disease. It is now a prominent reference knowledge base for integration and interpretation of large-scale molecular data sets generated by high-throughput experimental technologies.

There are plenty of tools developed for KEGG pathway mapping or function annotation. But most of them are limited in finding significant enriched pathways for selected genes. To further analysis the function changes inner pathway, some tools were developed to map selected genes in pathway map, such as Color Pathway in KEGG mapper tools[1]. When direct comparing of two different samples, such as disease and normal persons, the gene expression changes in each KEGG pathway could be visualized, which would be helpful in understanding the function changes between samples.

With the development of high-throughput experimental technologies, the systematic analysis for complicated biological questions, such as drug stimulation, disease progression and cell differentiation, often contains multi-types or multi-groups of data, including the expression in transcriptome and proteome, in disease and normal samples, and in different time points. However, none of the currently available software for KEGG pathway mapping could be used for visualization of the expression profiles in such complicated data analysis. The only solution is to generate multiple pathway maps, each map for each time point, or illustrate the expression profile manually, which is inconvenient in visualization and ambiguous in function analysis.

To address this problem, we developed an R package KEGGprofile, which provided an easy and automatic pipeline for analyze and visualization for multi-types and multi-groups expression data. With this package, the expression profile of genes and the annotation in KEGG pathway maps could be integrated together. Then the researcher could directly focus on the function changes inner pathway or expression correlation between different types of data. This would be a valuable tool for systematic profiling or time series data analysis.

2 Example usage

2.1 Data preparation

The NCBI gene IDs (such as 67040, 93683) is used in KEGG database to represent genes in the pathway. We need to transform the identifiers in our expression data into NCBI gene IDs. After the transformation, KEGGprofile could be generally applicable for genomics, transcriptomics and proteomics data. A previously published data of proteome and phosphoproteome analysis in different cell phase was taken as an example[2].

We have prepared an example data in the data directory. Then it can be import into R environment with:

```
library(KEGGprofile)

## Loading required package:  DBI
##
## KEGG.db contains mappings based on older data because the original
##  resource was removed from the the public domain before the most
##  recent update was produced.  This package should now be
##  considered deprecated and future versions of Bioconductor may
##  not have it available.  Users who want more current data are
##  encouraged to look at the KEGGREST or reactome.db packages

data(pro_pho_expr)
data(pho_sites_count)
ls()

## [1] "pho_sites_count" "pro_pho_expr"

colnames(pro_pho_expr)

##  [1] "Proteome.G1.phase"        "Proteome.G1.S"
##  [3] "Proteome.Early.S"         "Proteome.Late.S"
##  [5] "Proteome.G2.phase"        "Proteome.Mitosis"
##  [7] "Phosphoproteome.G1.phase" "Phosphoproteome.G1.S"
##  [9] "Phosphoproteome.Early.S"  "Phosphoproteome.Late.S"
## [11] "Phosphoproteome.G2.phase" "Phosphoproteome.Mitosis"

pro_pho_expr[1:3, 1:4]

##           Proteome.G1.phase Proteome.G1.S Proteome.Early.S Proteome.Late.S
## 100                   -0.14          0.12            -0.08           -0.18
## 100008586             -0.14         -1.47            -0.60           -0.12
## 10001                 -0.71          1.17             0.92            0.82
```

The pro_pho_expr is a data.frame with expression profiles. The column 1-6 are proteome data and column 7-12 are phosphoproteome data. The 6 time points are G1, G1/S, Early S, Late S, G2, Mitosis. For the phosphorylation sites mapping to the same gene, the one with largest variation in 6 time points are kept. The pho_sites_count is a data.frame with number of phosphorylation sites quantified for each gene.

Here the NCBI gene IDs should be row.names of all the data.frame. If your expression data is not in NCBI gene IDs, you need to first convert it. We provided a function called 'convertId' to do it.

```
example(convertId)

##
## cnvrtI> temp<-cbind(rnorm(10),rnorm(10))
##
## cnvrtI> row.names(temp)<-c("Q04837","P0C0L4","P0C0L5","O75379","Q13068","A2MYD1","P607
##
## cnvrtI> colnames(temp)<-c("Exp1","Exp2")
##
## cnvrtI> convertId(temp,filters="uniprot_swissprot",keepMultipleId=TRUE)
##                 Exp1        Exp2
## 2543      -0.2320496  0.08995889
## 645073    -0.2320496  0.08995889
## 729396    -0.2320496  0.08995889
## 729408    -0.2320496  0.08995889
## 729422    -0.2320496  0.08995889
## 729431    -0.2320496  0.08995889
## 729442    -0.2320496  0.08995889
## 100008586 -0.2320496  0.08995889
## 100132399 -0.2320496  0.08995889
## 3106      -1.3349646  0.92910287
## 60         0.9640561 -0.50455889
## 6742      -0.2662850  1.08714346
## 720       -1.5178994 -0.79049591
## 721       -0.1400703 -0.78840709
## 100293534 -0.1400703 -0.78840709
## 8674      -1.3451836  0.22946111
## A2MYD1     1.4686199 -1.25141675
##
## cnvrtI> ## Not run:
## cnvrtI> ##D temp<-cbind(rnorm(5000),rnorm(5000),rnorm(5000),rnorm(5000),rnorm(5000),rn
## cnvrtI> ##D row.names(temp)<-1000:5999
## cnvrtI> ##D colnames(temp)<-c("Control1","Control2","Control3","Treatment1","Treatment
## cnvrtI> ##D convertId(temp,filters="entrezgene",attributes =c("entrezgene","uniprot_sw
## cnvrtI> ## End(Not run)
## cnvrtI>
## cnvrtI>
## cnvrtI>
```

Besides, The package requires original KEGG pathway maps as backgrounds and KGML (KEGG XML) files to extract the gene locations in the pathway maps. These files can be downloaded from KEGG website (http://www.kegg.jp/kegg/) and we also provide a function called 'download_KEGGfile' to do so. Now download the pathway map and KGML file for human pathway '04110' to the work directory:

```
download_KEGGfile(pathway_id = "04110", species = "hsa")

## [1] "Downloading files: 1/1"
```

Here the pathway_id could be set as 'all', and then the entire pathway ids for human would be extracted from the KEGG.db package and the related files would be downloaded.

2.2 Find enriched pathways

The function 'find_enriched_pathways' could be used to find enriched pathways for interested genes. The interested genes could be selected in several methods, such as genes response to specific stimulation, or genes with negative correlation between disease and normal samples. And the result of statistic tests could also be used. Then the selected genes would be annotated with KEGG pathway database and hypergeometric tests were used to estimate the significance of enrichment. Besides, a criterion for number of annotated genes in the pathway could also be used for pathway selection.

There is a very important parameter 'download_latest' in 'find_enriched_pathway' function. As the KEGG.db package was only updated until 2012, we can download the lateset genes and pathways links from KEGG database when 'download_latest' was set as TRUE. It is very important when the users were interested in some non model organisms which were imported into KEGG after 2012.

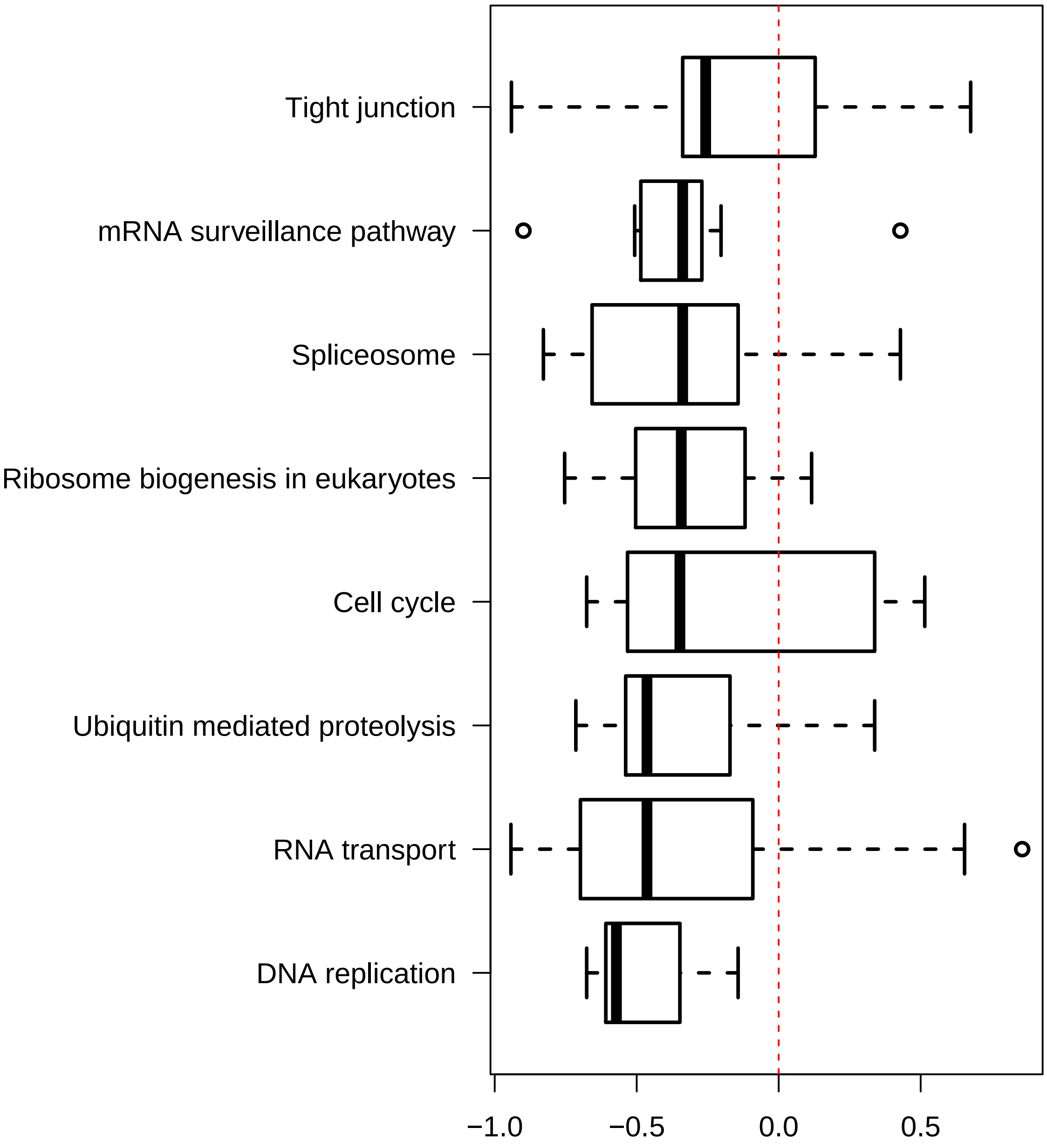
Here we used the proteins highly phosphorylated as candidates for annotation. The number of phosphorylation sites quantified larger than 10 was set as a criterion.

```
genes <- row.names(pho_sites_count)[which(pho_sites_count >= 10)]
pho_KEGGresult <- find_enriched_pathway(genes, species = "hsa")
pho_KEGGresult[[1]][, c(1, 5)]

##                            Pathway_Name       pvalue
## 04530                    Tight junction 1.966358e-04
## 03013                     RNA transport 1.154993e-14
## 03008 Ribosome biogenesis in eukaryotes 1.368032e-03
## 04120    Ubiquitin mediated proteolysis 1.096403e-03
## 04110                        Cell cycle 1.374848e-04
## 03040                       Spliceosome 1.565396e-16
## 03015          mRNA surveillance pathway 1.600093e-03
## 03030                   DNA replication 3.968022e-05
```

Then we compared the correlations between proteins and phospholations for these enriched in highly phosphorylated proteins pathways.

```
plot_pathway_cor(gene_expr = pro_pho_expr, kegg_enriched_pathway = pho_KEGGresult)
```



```
##                    DNA replication                       RNA transport
##                         0.08767730                          0.07181928
##     Ubiquitin mediated proteolysis                          Cell cycle
##                         0.29992078                          0.72446442
## Ribosome biogenesis in eukaryotes                          Spliceosome
##                         0.43376060                          0.05778624
##          mRNA surveillance pathway                      Tight junction
##                         0.37690666                          0.80654151
```

As the example data here was from an research in different cell phase, the Cell cycle pathway (pathway id 04110) was further visualized.

2.3 Visualization of expression profile on KEGG maps

In each KEGG pathway map, genes are represented by a polygon and biological relations between genes such as activation or phosphorylation are represented by lines. The function 'plot_pathway' could be used to integrate the expression profiles in the pathway map instead of the original gene polygon. There are two visualization methods to represent gene expression profiles: "background" and "lines". The first one is applicable for analysis with only one sample or one type of data, which divides the gene polygon into several sub-polygons to represent different time points. And each sub-polygon has a specific background color to represent expression changes in that time point.

We used the phosphoproteome changes in 6 time points as a example. Firstly a function 'col_by_value' was used to transform the expression difference between samples into specific color. After that, we can use "plot_profile" function to visualiz the gene expression profile in the KEGG pathway. A pathway map named 'hsa04110_profile_bg.png' would be generated at the working directory.

```
## the phosphoproteome data
pho_expr <- pro_pho_expr[, 7:12]
temp <- apply(pho_expr, 1, function(x) length(which(is.na(x))))
pho_expr <- pho_expr[which(temp == 0), ]
## transform the expression difference into specific color
col <- col_by_value(pho_expr, col = colorRampPalette(c("green", "black", "red"))(1024),
    range = c(-6, 6))
## visualization by method 'bg'
temp <- plot_pathway(pho_expr, type = "bg", bg_col = col, text_col = "white",
    magnify = 1.2, species = "hsa", database_dir = system.file("extdata", package = "KEG
    pathway_id = "04110")

## [1] "Downloading files: 1/1"
## [1] "The figure was generated in hsa04110_profile_bg.png"
```

The second method plots lines with different colors in the gene polygon to represent different samples or different types of data. The dynamic changes of lines are determined by the profiles of genes in different time points. The background colors could also be added to the pathway map to provide more biological information, such as p values and subcellular locations.

The proteome and phosphoproteome changes were used as example for method 'lines'. Firstly the function 'col_by_value' was used to transform the number of phosphorylation sites quantified for each gene into specific color as the background for each gene polygon. Then the "plot_profile" function was performed and a pathway map named 'hsa04110_profile_lines.png' would be generated at the working directory.

```
## transform the number of phosphorylation sites into specific color
col <- col_by_value(pho_sites_count, col = colorRampPalette(c("white", "khaki2"))(4),
    breaks = c(0, 1, 4, 10, Inf))  ## visualization by method 'lines'
temp <- plot_pathway(pro_pho_expr, type = "lines", bg_col = col, line_col = c("brown1",
    "seagreen3"), groups = c(rep("Proteome", 6), rep("Phosphoproteome", 6)),
    magnify = 1.2, species = "hsa", database_dir = system.file("extdata", package = "KEG
    pathway_id = "04110", max_dist = 5)
```

```
## [1] "The figure was generated in hsa04110_profile_lines.png"
```

In this section, we just used the background colors of gene polygon to represent the number of phosphorylation sites. In fact, the colors for gene name (text_col) and gene polygon border (border_col) could also be determined by function ’col_by_value’ and represent some other important biological information, such as subcellular locations, correlation between samples. Here we just demonstrated the application of gene expression data. In fact Compound data was also supported by KEGGprofile. You can see the examples in ’plot_pathway’ function for more details.

3 More details

To make the visualization process more easier, the function ’plot_pathway’ is in fact a wrapper function for download_KEGGfile, parse_XMLfile and plot_profile functions. Firstly, the existence of KEGG pathway map files (.xml and .png) would be checked in the database_dir. If not, the download_KEGGfile function would be used to download the files. Then the function parse_XMLfile would be used to parse xml file to get a matrix containing the genes in this pathway, and their names, locations etc. At last, the function ’plot_profile’ would be used to generate the pathway map.

References

[1] Kanehisa M. Goto S. Sato Y. Furumichi M. Tanabe M. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res*, 40(Database issue):D109–14, 2012.

[2] Olsen J. V. Vermeulen M. Santamaria A. Kumar C. Miller M. L. Jensen L. J. Gnad F. Cox J. Jensen T. S. Nigg E. A. Brunak S. Mann M. Quantitative phosphoproteomics reveals widespread full phosphorylation site occupancy during mitosis. *Sci Signal*, 3(104):ra3, 2010.