Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Calculating Module Enrichment and Visualizing Data on Large-scale Molecular Maps with the R packages ACSNMineR and RNaviCell

Paul Deveau
Institut Curie
PSL Research University
INSERM U900
Mines-ParisTech

Eric Bonnet
Institut Curie
PSL Research University
INSERM U900
Mines-ParisTech

Abstract

The abstract of the article.

Keywords: keywords, comma-separated, not capitalized, Java.

1. Introduction

Biological pathways and networks comprise sets of interactions or functional relationships, occurring at the molecular level in living cells (?). A large body of knowledge on cellular biochemistry is organized in publicly available repositories such as the KEGG database (?), Reactome (?) and MINT (?). All these biological databases facilitate a large spectrum of analyses, improving our understanding of cellular systems. For instance, it is a very common practice to cross the output of high-throughput experiments, such as mRNA or protein expression levels, with curated biological pathways in order to visualize changes, analyze their impact on a network and formulate new hypotheses about biological processes. Many biologists and computational biologists establish list of genes of interest (e.g. a list of genes that are differentially expressed between two conditions, such as normal vs disease) and then try to see if known biological pathways are enriched with this list of genes.

We have recently released the Atlas of Cancer Signalling Network (ACSN), a web-based database which describes signaling and regulatory molecular processes that occur in a healthy mammalian cell but that are frequently deregulated during cancerogenesis (?). The ACSN atlas aims to be a comprehensive description of cancer-related mechanisms retrieved from the most recent literature. The web interface for ACSN is using the NaviCell technology, a software framework dedicated to web-based visualization and navigation for biological pathway maps (?). This environment is providing an easy navigation of maps through the use of the Google Maps JavaScript library, a community interface with a web blog system, and a comprehensive module for visualization and analysis of high-throughput data (?).

In this article, we describe two software packages related to ACSN analysis and data visualization for the popular R statistical enrironment (??). The package **ACSNMineR** is designed for the calculation of gene enrichment and depletion in ACSN maps, while **RNaviCell** is dedicated to data visualization on ACSN maps. Both packages are available on the Comprehensive R Archive Network (`https://cran.r-project.org/web/packages/ACSNMineR/` and `https://cran.r-project.org/web/packages/RNaviCell/`), and on the GitHub repository (`https://github.com/sysbio-curie/ACSNMineR` and `https://github.com/sysbio-curie/RNaviCell`). For the remainder of this article, we describe the organization of each package and illustrate their capacities with several concrete examples demonstrating their capabilities.

2. Packages organization

2.1. ACSNMineR

Currently, ACSN maps covers signaling pathways involved in DNA repair, cell cycle, cell survival, cell death, epithelial-to-mesenchymal transition (EMT) and cell motility. Each of these large-scale molecular map is decomposed in a number of functional modules. The maps themselves are merged into a global ACSN map. Thus the information included in ACSN is organized in three hierarchical levels: a global map, five individual maps, and a total of 55 functional modules. Each ACSN map covers hundreds of molecular players, biochemical reactions and causal relationships between the molecular players and cellular phenotypes. ACSN represents a large-scale biochemical reaction network of 4,826 reactions involving 2,371 proteins, and is continuously updated and expanded. We have included the three hierarchical levels in the **ACSNMineR** package, in order to be able to calculate enrichments at all three levels. The calculations are made by counting the number of occurences of gene symbols (HUGO gene names) from a given list of genes of interest in all ACSN maps and modules. Table 1 is detailling the number of gene symbols contained in all the ACSN maps.

The statistical significance of the counts in the modules is assessed by using either the Fisher exact test (??) or the hypergeometric test, which are equivalent for this purpose ?.

The current ACSN maps are included in the **ACSNMineR** package, as a list of character matrices.

```
> length(ACSN_maps)
[1] 6
> names(ACSN_maps)
[1] "Apoptosis"    "CellCycle"    "DNA_repair"   "EMT_motility" "ACSN_master"
[6] "Survival"
```

For each matrix, rows represent a module, with the name of the module in the first column,

Table 1: ACSN maps included in the **ACSNMineR** package. Map: map name, Total: total number of gene symbols (HUGO), Nb mod.: number of modules, Min: mimimum number of gene symbols in the modules, Max: maximum number of gene symbols in the modules, Mean: average number of gene sybols per module. N.B.: one gene symbol may be present in several modules of the map.

Map	Total	Nb mod.	Min	Max	Mean
ACSN global	2137	55	2	625	85
Survival	1065	46	1	434	54
Apoptosis	668	44	1	382	33
EMT & Cell motility	620	41	1	615	43
DNA repair	337	48	1	69	19
Cell cycle	119	45	1	27	8

followed by a description of the module (optional), and then followed by all the gene symbols of the module. The maps will be updated according to every ACSN major release.

The main function of the **ACSNMineR** package is the `enrichment` function, which is calculating over-representation or depletion of genes in the ACSN maps and modules. We have included a small list of 12 Cell Cycle related genes in the package, named `genes_test` that can be used to test the main enrichment function and to get familiar with its different options.

```
> genes_test
 [1] "ATM"     "ATR"     "CHEK2"   "CREBBP"  "TFDP1"   "E2F1"    "EP300"
 [8] "HDAC1"   "KAT2B"   "GTF2H1"  "GTF2H2"  "GTF2H2B"
```

The example shown below is the simplest command that can be done to test a gene list for over-representation on the six included ACSN maps. With the list of 12 genes mentionned above and a default p-value cutoff of 0.05, we have a set of 36 maps or modules that are significantly enriched. The results are structured as a data frame with nine columns displaying the module name, the module size, the number of genes from the list in the module, the names of the genes that are present in the module, the size of the reference universe, the number of genes from the list that are present in the universe, the raw p-value, the p-value corrected for multiple testing and the type of test performed. The module field in the results data frame indicate the map name and the module name separated by a column character. If a complete map is significantly enriched or depleted, then only the map name is shown, without any module or column character. For instance, the first line of the results object below concern the E2F1 module of the Apoptosis map.

```
> library(ACSNMineR)
> results <- enrichment(genes_test)
> dim(results)
[1] 36  9
> results[1,]
            module module_size nb_genes_in_module      genes_in_module
V12 Apoptosis:E2F1           5                  4 ATM E2F1 EP300 KAT2B
    universe_size nb_genes_in_universe      p.value p.value.corrected    test
V12          2133                   12 1.043478e-08       2.81739e-07 greater
```

The `enrichment` function can take up to eight arguments: the gene list (as a character vector), the list of maps that will be used to calculate enrichment or depletion, the type of statistical test (either the Fisher exact test or the hypergeometric test), the module minimal size for which the calculations will be done, the universe, the p-value threshold and the alternative (”greater” for calculating over-representation, ”less” for depletion and ”both” for both tests).

Only the gene list is mandatory to call the `enrichment` function, all the other arguments have default values.

[explain default values]
The `maps` argument can either be a dataframe imported from a gmt file with the `format_from_gmt` function or a list of dataframes generated by the same procedure. By default, the function uses the ACSN maps previously described.

The correction for multiple testing by default is set to false discovery rate (fdr), which is equivalent to Benjamini & Hochberg correction, but can be changed to any of the usual corrections such as Bonferroni, Holm, Hochberg, Holm, or Benjamini & Yekutieli **?**, or even disabled .

The minimal module size represents the smallest size value of a module that will be used to compute enrichment or depletion. This is meant to remove results of low significance for module of small size.

The universe in which the computation is made by default is defined by the maps. Which means that all genes that were given as input and that are not present on the maps will be discarded. To keep all genes, the user can set the universe to `HUGO`,referring to the database.

The threshold is the maximal value of the corrected p-value (unless user chose not to correct for multiple testing) that will be displayed in the result table.

[give examples with maps, universe, correction multitest]

[explain multiple samples] It may be of interest to compare enrichment of pathwways in different cohorts or experiments. For example, enrichment of highly expressed pathways can reveal differences between two cancer types or two cell lines. To ease such comparisons, **ACSNMineR** provides a `multisample_enrichment` function. It relies on the `enrichment` function but takes a list of character vector genes. The name of each element of the list will be assumed to be the name of the sample for further analysis. Most of the arguments given to `multisample_enrichment` are the same as the ones passed to `enrichment`. However, the `cohort_threshold` is designed to filter out modules which would not pass the significance threshold in all samples.

[explain graphs generation] Finally,

2.2. RNaviCell

The NaviCell Web Service provides a server mode, which allows automating visualization tasks and retrieving data from molecular maps via RESTful (standard http/https) calls. Bindings to different programming languages are provided in order to facilitate the development of data visualization workflows and third-party applications (**?**). RNaviCell is the R binding to the NaviCell Web Service. It is implemented as a standard R package, using the R object-oriented framework known as Reference Classes (**?**). Most of the work done by the user using graphical point-and-click operations on the NaviCell web interface are encoded as functions in the library encapsulating http calls to the server with appropriate parameters and data. Calls to the NaviCell server are performed using the library RCurl (**?**), while data encoding/decoding in JSON format is performed with the RJSONIO library (**?**).

3. Examples

Affiliation:

Eric Bonnet
Computational Systems Biology of Cancer
Institut Curie
26, rue d'Ulm
75248 Paris, France
E-mail: eric.bonnet@curie.fr

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd