GRAPH Interaction from pathway Topological Environment

Gabriele Sales (gabriele.sales@unipd.it)
Enrica Calura (enrica.calura@gmail.com)
Chiara Romualdi (chiara.romualdi@unipd.it)

October 13, 2014

Contents

1 Introduction

graphite (GRAPH Interaction from pathway Topological Environment) an R package built to i) provide networks from seven databases (KEGG, [1]; Biocarta, www.biocarta.com; Reactome, [2]; NCI/Nature Pathway Interaction Database, [3]; SPIKE, [10]; HumanCyc, [11]; Panther, [12]); ii) discriminate between different types of gene groups; iii) propagate pathway connections through chemical compounds; iv) allow the selection of edge attributes and the conversion of nodes identifiers to EntrezGene IDs and HUGO Symbols; finally, v) to run *SPIA*, *DEGraph*, *CliPPER* and *topologyGSA* analyses directly on *graphite* networks.

2 Pathway topology conversion to gene network

In order to gather curated information about human pathways, we have collected data from the four public databases that have emerged as reference points for the systems biology community. The KEGG database has been in development by Kanehisa Laboratories since 1995, and is now a prominent reference knowledge base for integration and interpretation

of large-scale molecular data sets generated by genome sequencing and other high-throughput experimental technologies. KEGG is the only pathway database not in biopax format, they use the KGML format. Reactome, backed by the EBI, is one of the most complete repository; it is frequently updated and provides a semantically rich description of each pathway [2]. KEGG Pathways (KGML format) provides maps for both signaling and metabolic pathways [1]. BioCarta (www.biocarta.com) is a developer, supplier and distributor company of reagents and assays for biopharmaceutical and academic research. Through an "open source" approach, this community-fed forum constantly integrates emerging proteomic information from the scientific community. It also catalogs and summarizes important resources providing information for over 120,000 genes.BioCarta pathway data in biopax format are available through NCI website. NCI (NCI/Nature Pathway Interaction Database [3]) is a highly-structured, curated collection of information about known biomolecular interactions and key cellular processes assembled into signaling pathways. This was a collaborative project between the NCI and Nature Publishing Group (NPG). Panther [12] data are a comprehensive, curated database of pathways, protein families, trees, subfamilies and functions available at http://pantherdb.org backed by the University of Southern California. HumanCyc is part of the BioCyc database collection of pathways [11]. Finally, SPIKE (www.cs.tau.ac.il/ spike/) a database for human curated signaling pathways backed by Tel Aviv University [10].

2.1 Pathway definition

KEGG database provides separate `xml` files, one for each pathway, thus, a pathway is defined by all the reactions defined within each file. For the other databases, we identify a pathway every time in the BioPax format is defined the `xml` tag `pathway`.

2.2 Nodes with multiple elements

Within a pathway nodes often correspond to multiple gene products. These can be divided into protein complexes (proteins linked by protein-protein interactions) and the groups containing alternative members (genes generally with similar biochemical functions). Thus, when considering signal propagation these groups should be considered differently; the first (hereafter group AND) should be expanded into a clique (all proteins connected to the others), while the second (hereafter group OR) should be expanded without connection among the contained elements.

In the KGML format there are two ways of defining nodes with multiple elements: protein complexes (group AND defined by entry `type="group"`) and group with alternative members (group OR defined by entry `type="gene"`). In the BioPax format only one type of group is allowed: protein complexes (group AND) with the `owl` tag `complex`. However, it often happened that `protein` tag contains multiple `xref` pointing to alternative elements of the process (group OR).

2.3 Compound mediated interactions

Compound mediated interactions are interactions for which a compound acts as a bridge between two elements. Since chemical compounds are generally not measured with high-throughput technology, they should be removed from the network. However, the trivial elimination of the compounds, without signal propagation, will strongly bias the topology interrupting the signals that pass through them. If element A is linked to compound c and compound c is linked to element B, thus elements A should be linked to elements B.

Within the KGML format there are two different ways of describing a compound mediated interaction: i) direct interaction `type="PPrel"` (A interacts whit B through compound c) and ii) indirect one `type="PCrel"` (A interacts whit compound c and c interacts whit B). In the BioPax `owl` language, one the other hand, only a indirect way of defining compound mediated signals is available.

Since signal propagation reconstruction is crucial for topological gene set analysis we decide to include additional rules for the propagation reconstructing a connection between two genes connected through a series of chemical compounds. Not all compound are considered for the propagation because some of them, such as Hydrogen, H2O, ATP, ADP etc., are highly frequent in map descriptions and the signal propagation through them would lead to degenerate too long chains of compounds. The compounds not considered for propagation are not characteristic of a specific reaction but secondary substrates/products widely shared among different processes.

2.4 Relation attributes

graphite allows the user to see the single/multiple relation types that characterized an edge. The type of edges have been conserved as much as possible to those annotated in the data formats. Some new types have been introduced due to topological conversion needs.

2.5 Pathway functions recovery

A pathway network can be retrieved using the name of the pathway:

```
> names(biocarta)[1:10]
```

```
 [1] "acetylation and deacetylation of rela in nucleus"
 [2] "actions of nitric oxide in the heart"
 [3] "activation of camp-dependent protein kinase pka"
 [4] "activation of csk by camp-dependent protein kinase inhibits signaling through the t cell receptor"
 [5] "activation of pkc through g-protein coupled receptors"
 [6] "adp-ribosylation factor"
 [7] "agrin in postsynaptic differentiation"
 [8] "ahr signal transduction pathway"
 [9] "akap95 role in mitosis and chromosome dynamics"
[10] "akt signaling pathway"
```

```
> p <- biocarta[["acetylation and deacetylation of rela in nucleus"]]
> p
```

```
"acetylation and deacetylation of rela in nucleus" pathway from biocarta
Number of nodes     = 5
Number of edges     = 12
Type of identifiers = native
Retrieved on        = 2014-10-01
```

or its position in the list of pathways:

```
> p <- biocarta[[1]]
> p@title
```

```
[1] "acetylation and deacetylation of rela in nucleus"
```

In the network, nodes represent genes and edges functional relationships among them. Nodes can have heterogeneous IDs (see section 4 for more details):

```
> nodes(p)
```

```
[1] "EntrezGene:1387" "EntrezGene:2033" "EntrezGene:4792" "EntrezGene:5970"
[5] "EntrezGene:8841"
```

Edges can be characterized by multiple functional relationships:

```
> edges(p)
```

```
               src            dest  direction                        type
1  EntrezGene:1387 EntrezGene:2033 undirected                     binding
2  EntrezGene:1387 EntrezGene:4792   directed     control(Out(ACTIVATION))
3  EntrezGene:1387 EntrezGene:5970   directed process(BiochemicalReaction)
4  EntrezGene:1387 EntrezGene:8841   directed      control(In(ACTIVATION))
5  EntrezGene:2033 EntrezGene:4792   directed     control(Out(ACTIVATION))
6  EntrezGene:2033 EntrezGene:5970   directed process(BiochemicalReaction)
7  EntrezGene:2033 EntrezGene:8841   directed      control(In(ACTIVATION))
8  EntrezGene:4792 EntrezGene:5970 undirected                     binding
```

```
9  EntrezGene:5970 EntrezGene:1387 undirected                    binding
10 EntrezGene:5970 EntrezGene:2033 undirected                    binding
11 EntrezGene:5970 EntrezGene:4792   directed     control(Out(ACTIVATION))
12 EntrezGene:8841 EntrezGene:5970   directed     control(Out(ACTIVATION))
```

This same steps can be used to access the Reactome, KEGG, SPIKE and NCI databases (through the `reactome`, `kegg`, `spike` and `nci` lists, respectively).

3 Graph

The function `pathwayGraph` builds a *graphNEL* object from a pathway `p`:

```
> g <- pathwayGraph(p)
> g
```

```
A graphNEL graph with directed edges
Number of Nodes = 5
Number of Edges = 13
```

```
> edgeData(g)[1]
```

```
$`EntrezGene:1387|EntrezGene:2033`
$`EntrezGene:1387|EntrezGene:2033`$weight
[1] 1

$`EntrezGene:1387|EntrezGene:2033`$edgeType
[1] "binding"
```

4 Identifiers

Gene annotations databases are widely used as public repositories of biological information. Our current knowledge on biological elements is spread out over a number of databases (such as: Entrez Gene , RefSeq, backed by the NCBI http://www.ncbi.nlm.nih.gov/, UniProt, ENSEMBL backed by the EBI http://www.ebi.ac.uk/ to name just a few), specialised on a subset of specific biological entities (for instance, UniProt focuses on proteins while Entrez Gene focuses on genes). Key identifiers (IDs) in the internal structure of each such database uniquely represent biological entities, thus biological entities can be identified by heterogeneous IDs according to the selected database they refer to. Due to their different origins and specificity, switching from an ID to another is possible but not trivial: there could be either no correspondence between them or many-to-many relations. For detailed information about IDs, their structures and differences please consult those resources. For our purposes, we have chosen EntrezGene IDs and Gene Symbols because of their widespread use and simplicity. The function `converterIdentifiers` allows the user to map such variety of IDs to a single type. This mapping process, however, may lead to the loss of some nodes (not all identifiers may be recognized) and has an impact on the topology of the network (one ID may correspond to multiple IDs in another annotation or vice versa).

```
> pEntrez <- convertIdentifiers(p, "entrez")
> pEntrez
```

```
"acetylation and deacetylation of rela in nucleus" pathway from biocarta
Number of nodes     = 5
Number of edges     = 12
Type of identifiers = Entrez Gene
Retrieved on        = 2014-10-01
```

```
> nodes(pEntrez)
```

```
[1] "1387" "2033" "4792" "5970" "8841"
```

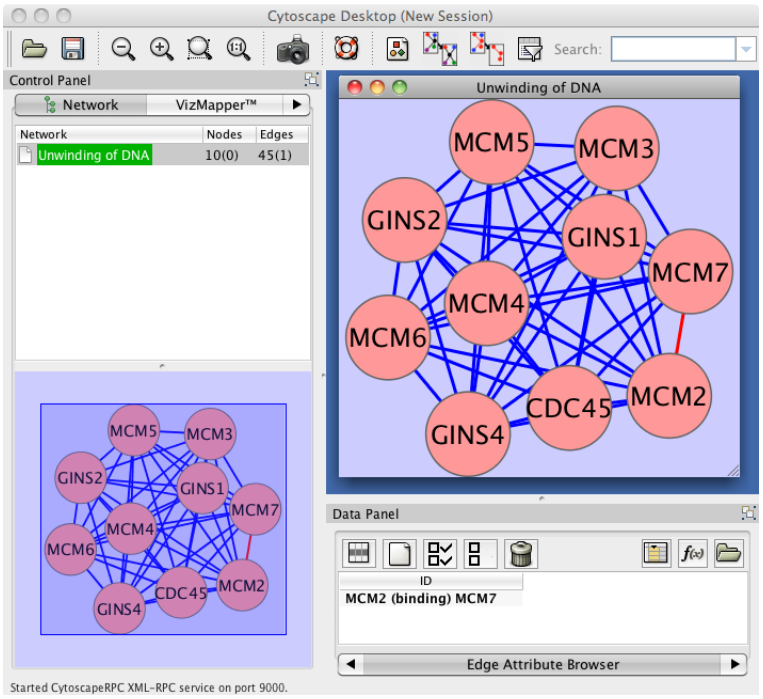
```
> pSymbol <- convertIdentifiers(p, "symbol")
> nodes(pSymbol)
```

```
[1] "CREBBP" "EP300"  "NFKBIA" "RELA"   "HDAC3"
```

5 Cytoscape Plot

Several pathways have a huge number of nodes and edges, thus there is the need of an efficient system of visualization. To this end *graphite* uses the *Rcytoscape* package to export the network to Cytoscape. Cytoscape is a Java based software specifically built to manage biological network complexity and for this reason it is widely used by the biological community. Unfortunately, since *Rcytoscape* use CytoscapeRCP plugin, which is not available for the new Cytoscape 3.0 version, the use of `cytoscapePlot` is currently limited to Cytoscape 2.8 and lower versions.

```
> cytoscapePlot(convertIdentifiers(reactome$`Unwinding of DNA`, "symbol"))
```



6 Topological pathway analysis

graphite gives access to three types of topological pathway analyses recently proposed. For more details on the results obtained by these analyses see the corresponding R packages.

6.1 SPIA

The analysis with *SPIA* requires the conversion of the networks in a suitable format. This conversion is performed by the function `prepareSPIA` that must be executed before the analysis command `runSPIA`. The *SPIA* data will be saved in the current working directory; every time you change it, you should also re-run `prepareSPIA`. Edges not included in SPIA have been coerced into the admitted SPIA types. Compound mediated interactions annotated in graphite with "indirect" type are mapped into the SPIA edge type "indirect effect" by default set to zero. To use the signal propagated through compounds type 1 in "indirect effect".

```
> library(SPIA)
> data(colorectalcancer)
> library(hgu133plus2.db)
> x <- hgu133plus2ENTREZID
> top$ENTREZ <- unlist(as.list(x[top$ID]))
```

```
> top <- top[!is.na(top$ENTREZ), ]
> top <- top[!duplicated(top$ENTREZ), ]
> tg1 <- top[top$adj.P.Val < 0.05, ]
> DE_Colorectal = tg1$logFC
> names(DE_Colorectal) <- as.vector(tg1$ENTREZ)
> ALL_Colorectal <- top$ENTREZ
> prepareSPIA(biocarta[1:2], "biocartaEx")
> res <- runSPIA(de=DE_Colorectal, all=ALL_Colorectal, "biocartaEx")
```

```
Done pathway 1 : acetylation and deacetylation ..
Done pathway 2 : actions of nitric oxide in the..
```

```
> res
```

```
                                          Name pSize NDE       pNDE         tA pPERT
1 acetylation and deacetylation of rela in nucleus     5   3 0.06089421 -0.2752649 0.921
         pG     pGFdr    pGFWER    Status
1 0.2176554 0.2176554 0.2176554 Inhibited
```

For more details see the *SPIA* package [5, 6, 4].

6.2 DEGraph

DEGraph implements recent hypothesis testing methods which directly assess whether a particular gene network is differentially expressed between two conditions.

In *graphite*, *DEGraph* has two dedicated functions. The first is `runDEGraph`, which performs the analysis on a single pathway.

```
> library(DEGraph)
> data("Loi2008_DEGraphVignette")
> p <- convertIdentifiers(biocarta[["actions of nitric oxide in the heart"]], "entrez")
> res <- runDEGraph(p, exprLoi2008, classLoi2008)
> res$`1`
```

```
$p.value
                     T2 T2 (1 Fourier components)
               0.6106982                 0.3221514

$graph
A graphNEL graph with directed edges
Number of Nodes = 2
Number of Edges = 1

$k
[1] 1
```

The second function is `runDEGraphMulti`, which easily performs the analysis on the entire pathway database having as result a list of two elements: a list with the results of the pathway analyses and the list of generated errors.

For more details see the *DEGraph* package [7].

6.3 topologyGSA

topologyGSA uses graphical models to test the pathway components and to highlight those involved in its deregulation.

In *graphite*, *topologyGSA* has two dedicated functions. The first is `runTopologyGSA`, which performs the analysis on a single pathway.

```
> library(topologyGSA)
> data(examples)
> p <- convertIdentifiers(kegg[["Fc epsilon RI signaling pathway"]], "symbol")
> runTopologyGSA(p, "var", y1, y2, 0.05)

        Pathway Variance Test

data: exp1, exp2 and g

lambda = 23.34925, df = 10, p-value = 0.009528322, equal variances: TRUE
```

The second function is `runTopologyGSAMulti`, which easily performs the analysis on the entire pathway database having as result a list of two elements: a list with the results of the pathway analyses and the list of generated errors.

For more details see the *topologyGSA* package [8].

6.4 clipper

clipper is a package for topological gene set analysis. It implements a two-step empirical approach based on the exploitation of graph decomposition into a junction tree to reconstruct the most relevant signal path. In the first step clipper selects significant pathways according to statistical tests on the means and the concentration matrices of the graphs derived from pathway topologies. Then, it "clips" the whole pathway identifying the signal paths having the greatest association with a specific phenotype.

In *graphite*, *clipper* has two dedicated functions. The first is `runClipper`, which performs the analysis on a single pathway.

```
> library(clipper)
> library(ALL)
> path <- convertIdentifiers(kegg$'Chronic myeloid leukemia', "entrez")
> genes <- nodes(path)
> data(ALL)
> all <- as.matrix(exprs(ALL[1:length(genes),1:20]))
> classes <- c(rep(1,10), rep(2,10))
> rownames(all) <- genes
> clipped <- runClipper(path, all, classes, "mean", pathThr=0.1)
> clipped[,1:5]

      startIdx endIdx maxIdx lenght         maxScore
1;9          1      9      3      6 3.43427045434975
26;27       26     27      1      2 1.80239023880678
```

The second function is *runClipperMulti*, which easily performs the analysis on the entire pathway database having as result a list of two elements: a list with the results of the pathway analyses and the list of generated errors.

```
> library(clipper)
> library(ALL)
> paths <- lapply(kegg[1:5], function(x) convertIdentifiers(x, "entrez"))
> genes <- unlist(lapply(paths, nodes))
> data(ALL)
> all <- as.matrix(exprs(ALL[1:length(genes),1:20]))
> classes <- c(rep(1,10), rep(2,10))
> rownames(all) <- genes
> clipped <- runClipperMulti(paths, all, classes, "mean", pathThr=0.1)
```

```
> resClip <- do.call(rbind,clipped$results)
> resClip[,1:5]
```

	startIdx	endIdx	maxIdx	lenght	maxScore
Acute myeloid leukemia.1;13	1	13	1	8	0.489002875678518
Acute myeloid leukemia.1;15	1	15	1	8	0.381466628156972
Acute myeloid leukemia.1;29	1	29	1	15	0.21459172165788
Adherens junction.1;11	1	11	2	5	2.98607089916969
Adherens junction.1;12	1	12	1	4	0.748933068388498
Adipocytokine signaling pathway	1	15	1	3	1.70109091010236
Adrenergic signaling in cardiomyocytes	1	6	2	5	2.71346704237864

For more details see the *clipper* package [9].

References

[1] Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 1999 Jan 1;27(1):29-34.

[2] Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, de Bono B, Garapati P, Hemish J, Hermjakob H, Jassal B, Kanapin A, Lewis S, Mahajan S, May B, Schmidt E, Vastrik I, Wu G, Birney E, Stein L, D'Eustachio P. Reactome knowledgebase of human biological pathways and processes. Nucleic Acids Res. 2009 Jan;37(Database issue):D619-22. Epub 2008 Nov 3.

[3] Schaefer CF, Anthony K, Krupa S, Buchoff J, Day M, Hannay T, Buetow KH. PID: the Pathway Interaction Database. Nucleic Acids Res. 2009 Jan;37(Database issue):D674-9. Epub 2008 Oct 2.

[4] Draghici, S., Khatri, P., Tarca, A.L., Amin, K., Done, A., Voichita, C., Georgescu, C., Romero, R. A systems biology approach for pathway level analysis. Genome Research, 17, 2007.

[5] Tarca AL, Draghici S, Khatri P, Hassan SS, Mittal P, Kim JS, Kim CJ, Kusanovic JP, Romero R. A novel signaling pathway impact analysis. Bioinformatics. 2009 Jan 1;25(1):75-82.

[6] Adi L. Tarca, Sorin Draghici, Purvesh Khatri, et. al. A Signaling Pathway Impact Analysis for Microarray Experiments. Bioinformatics, 2009, 25(1):75-82.

[7] L. Jacob, P. Neuvial, and S. Dudoit. Gains in power from structured two-sample tests of means on graphs. Technical Report arXiv:q-bio/1009.5173v1, arXiv, 2010.

[8] Massa MS, Chiogna M, Romualdi C. Gene set analysis exploiting the topology of a pathway. BMC System Biol. 2010 Sep 1;4:121.

[9] Martini P, Sales G, Massa MS, Chiogna M, Romualdi C. Along signal paths: an empirical gene set approach exploiting pathway topology. Nucleic Acids Res. 2013 Jan 7;41(1):e19. doi: 10.1093/nar/gks866. Epub 2012 Sep 21.

[10] Arnon Paz, Zippora Brownstein, Yaara Ber, Shani Bialik, Eyal David, Dorit Sagir, Igor Ulitsky, Ran Elkon, Adi Kimchi, Karen B. Avraham, Yosef Shiloh and Ron Shamir. SPIKE: a database of highly curated human signaling pathways. Nucleic Acids Research, 2011, Vol. 39, Database issue.

[11] Caspi R, Altman T, Dale JM, Dreher K, Fulcher CA, Gilham F, Kaipa P, Karthikeyan AS, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Paley S, Popescu L, Pujar A, Shearer AG, Zhang P, Karp PD. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Research 38:D473-9 2010.

[12] PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. Huaiyu Mi, Anushya Muruganujan and Paul D. Thomas Nucl. Acids Res. (2012) doi: 10.1093/nar/gks1118