

ROntoTools: The R Onto-Tools suite

Calin Voichita, Sahar Ansari and Sorin Draghici
Department of Computer Science, Wayne State University, Detroit MI 48201

October 29, 2019

Abstract

This package is indented to be the R implementation of the web-based data mining and analysis suite of tools called Onto-Tools [10, 6, 5, 7, 8, 5, 12, 9, 4, 8, 9, 2, 9, 13, 3, 11]. Among these, Onto-Express (OE) was the first publicly available tool for the GO profiling of high throughput data and Pathway-Express (PE) the first tool to perform analysis of signaling pathways using important biological factors like all the interactions between the genes, the type of interaction between them and the position and magnitude of expression change for all the differentially expressed genes. We currently have over 10,000 registered users from 53 countries. Approximately, 5,000 of these are regular users (more than 10 data sets processed). This R package will provide these users with access to the direct functionalities of the online version, to new analysis methods and also expose the tools to a larger audience. As part of the first version, the pathway analysis tool Pathway-Express is made available.

1 Pathway-Express

Pathway-Express (`pe`) is a tool for the analysis of signaling pathways. Besides the original implementation [3, 14], this tool implements a number of improvements proposed in [15] that include the incorporation of gene significance and the elimination of the need to select differentially expressed genes. Pathway-Express uses two sources of data: one is the experiment data and the other is the database of pathways.

1.1 Pathway database

Pathway-Express is a general tool that accepts any set of signaling pathways defined using the standard implementation provided in the *graph* package. The only requirement is that each pathway, defined as an object of type *graph*, has a weight defined for each edge, representing the efficiency of the propagation between the two genes, and a weight for each node, that will capture the type of gene or the significance of the measured expression change. This package provides tools to access the KEGG database for signaling pathways and also tools to set these weights.

For example, to download and parse the signaling pathways available in KEGG use:

```
> require(graph)
> require(ROntoTools)
> kpg <- keggPathwayGraphs("hsa", verbose = FALSE)
```

The above code will load the available cached data for human (i.e., KEGG id *hsa*). To update the cache and download the latest KEGG pathways available use the `updateCache` parameter:

```
> kpg <- keggPathwayGraphs("hsa", updateCache = TRUE, verbose = TRUE)
```

This command is time consuming and depends on the available bandwidth.

The `kpg` is a list of *graph* objects:

```
> head(names(kpg))
```

```
[1] "path:hsa03008" "path:hsa03013" "path:hsa03015" "path:hsa03018"  
[5] "path:hsa03320" "path:hsa03460"
```

To inspect one of the pathway graphs, only the ID is required. Here is an example for the Cell Cycle:

```
> kpg[["path:hsa04110"]]
```

A graphNEL graph with directed edges

Number of Nodes = 124

Number of Edges = 632

```
> head(nodes(kpg[["path:hsa04110"]]))
```

```
[1] "hsa:1029" "hsa:51343" "hsa:4171" "hsa:4172" "hsa:4173" "hsa:4174"
```

```
> head(edges(kpg[["path:hsa04110"]]))
```

```
$`hsa:1029`
```

```
[1] "hsa:4193" "hsa:1019" "hsa:1021" "hsa:595" "hsa:894" "hsa:896"
```

```
$`hsa:51343`
```

```
[1] "hsa:983" "hsa:85417" "hsa:891" "hsa:9133"
```

```
$`hsa:4171`
```

```
character(0)
```

```
$`hsa:4172`
```

```
character(0)
```

```
$`hsa:4173`
```

```
character(0)
```

```
$`hsa:4174`
```

```
character(0)
```

In addition the parser extracted the type of interaction for each gene-gene interaction in an attribute called `subtype`:

```
> head(edgeData(kpg[["path:hsa04110"]], attr = "subtype"))
```

```
$`hsa:1029|hsa:4193`
```

```
[1] "inhibition"
```

```
$`hsa:1029|hsa:1019`
```

```
[1] "inhibition"
```

```
$`hsa:1029|hsa:1021`
```

```
[1] "inhibition"
```

```
$`hsa:1029|hsa:595`
```

```
[1] "inhibition"
```

```
$`hsa:1029|hsa:894`
```

```
[1] "inhibition"
```

```
$`hsa:1029|hsa:896`
```

```
[1] "inhibition"
```

Using this attribute the function `setEdgeWeights` sets the same weight for all the interactions of the same type:

```
> kpg <- setEdgeWeights(kpg, edgeTypeAttr = "subtype",  
+   edgeWeightByType = list(activation = 1, inhibition = -1,  
+   expression = 1, repression = -1),  
+   defaultWeight = 0)
```

At this point, `kpg` contains a list of graphs with weighted edges:

```
> head(edgeData(kpg[["path:hsa04110"]], attr = "weight"))
```

```
$`hsa:1029|hsa:4193`
```

```
[1] -1
```

```
$`hsa:1029|hsa:1019`
```

```
[1] -1
```

```
$`hsa:1029|hsa:1021`
```

```
[1] -1
```

```
$`hsa:1029|hsa:595`
```

```
[1] -1
```

```
$`hsa:1029|hsa:894`
```

```
[1] -1
```

```
$`hsa:1029|hsa:896`
```

```
[1] -1
```

To retrieve the title of the pathways and not just their ids the function `keggPathwayNames` can be used:

```
> kpn <- keggPathwayNames("hsa")
```

```
> head(kpn)
```

path:hsa03008	path:hsa03013
"Ribosome biogenesis in eukaryotes"	"RNA transport"
path:hsa03015	path:hsa03018
"mRNA surveillance pathway"	"RNA degradation"
path:hsa03320	path:hsa03460
"PPAR signaling pathway"	"Fanconi anemia pathway"

1.2 Experiment data

As an example, we provided a pre-processed data set from ArrayExpress (E-GEOD-21942) that studies the expression change in peripheral blood mononuclear cells (PBMC) between 12 MS patients and 15 controls. The data was preprocessed using the *limma* package. Only probe sets with a gene associated to them have been kept and for each gene only the most significant probe set has been selected (the table is already ordered by p-value):

```
> load(system.file("extdata/E-GEOD-21942.topTable.RData", package = "ROntoTools"))
> head(top)
```

	logFC	P.Value	adj.P.Val	entrez
200946_x_at	-1.0175141	5.833411e-13	4.172652e-09	hsa:2746
228697_at	-3.6479368	7.985427e-13	4.172652e-09	hsa:135114
210254_at	3.2807123	3.086572e-12	9.677020e-09	hsa:932
234726_s_at	-0.9792301	7.368175e-12	1.760593e-08	hsa:64418
215905_s_at	-1.7733135	7.861797e-12	1.760593e-08	hsa:9410
235542_at	-0.9447467	1.617944e-11	2.536288e-08	hsa:200424

Select differentially expressed genes at 1% and save their fold change in a vector *fc* and their p-values in a vector *pv*:

```
> fc <- top$logFC[top$adj.P.Val <= .01]
> names(fc) <- top$entrez[top$adj.P.Val <= .01]
> pv <- top$P.Value[top$adj.P.Val <= .01]
> names(pv) <- top$entrez[top$adj.P.Val <= .01]
> head(fc)
```

hsa:2746	hsa:135114	hsa:932	hsa:64418	hsa:9410	hsa:200424
-1.0175141	-3.6479368	3.2807123	-0.9792301	-1.7733135	-0.9447467

```
> head(pv)
```

hsa:2746	hsa:135114	hsa:932	hsa:64418	hsa:9410	hsa:200424
5.833411e-13	7.985427e-13	3.086572e-12	7.368175e-12	7.861797e-12	1.617944e-11

Alternatively, an analysis with all genes can be performed:

```
> fcAll <- top$logFC
> names(fcAll) <- top$entrez
> pvAll <- top$P.Value
> names(pvAll) <- top$entrez
```

The reference contains all the genes measured in the analysis:

```
> ref <- top$entrez
> head(ref)

[1] "hsa:2746"    "hsa:135114" "hsa:932"     "hsa:64418"  "hsa:9410"
[6] "hsa:200424"
```

1.3 Setting the node weights

The node weights are used to encode for the significance of each gene, the term described as α in [15]. The two alternative formulas to incorporate the gene significance:

$$\alpha = 1 - p/p_{thr} \text{ and } \alpha = -\log(p/p_{thr}) \quad (1)$$

are implemented as two function `alpha1MR` and `alphaMLG`.

To set the node weights the function `setNodeWeights` is used:

```
> kpg <- setNodeWeights(kpg, weights = alphaMLG(pv), defaultWeight = 1)
> head(nodeWeights(kpg[["path:hsa04110"]]))

hsa:1029 hsa:51343 hsa:4171 hsa:4172 hsa:4173 hsa:4174
1.0000000 1.0000000 0.8120949 1.0000000 1.0000000 1.0000000
```

1.4 Pathway analysis and results summary

Up to this point all the pieces need for the analysis have been assembled:

- the pathway database with the experiment specific gene significance - `kpg`
- the experiment data - `fc` and `ref`

To perform the analysis the function `pe` is used (increase the parameter `nboot` to obtain more accurate results):

```
> peRes <- pe(x = fc, graphs = kpg, ref = ref, nboot = 200, verbose = FALSE)
```

The result object can be summarized in a table format with the desired columns using the function `Summary`:

```
> head(Summary(peRes))
```

	totalAcc	totalPert	totalAccNorm	totalPertNorm	pPert
path:hsa05010	17.90716	121.13696	0.5550563	2.818100	0.014925373
path:hsa05110	22.83759	87.30055	4.8217858	5.261800	0.004975124
path:hsa04145	0.00000	102.93799	NA	5.863065	0.004975124
path:hsa05152	140.10374	233.91461	5.7932852	6.612930	0.004975124
path:hsa04722	56.17539	117.15557	1.9576817	3.326614	0.009950249
path:hsa05016	4.18750	123.28851	-0.1644465	3.608937	0.009950249
	pAcc	pORA	pComb	pPert.fdr	pAcc.fdr
path:hsa05010	0.557213930	1.360242e-05	3.331570e-06	0.02607445	0.71057556
path:hsa05110	0.004975124	1.085083e-04	8.330837e-06	0.01414496	0.03841902
path:hsa04145	NA	2.424942e-04	1.764759e-05	0.01414496	NA
path:hsa05152	0.004975124	8.354186e-04	5.565668e-05	0.01414496	0.03841902

```

path:hsa04722 0.049751244 4.644830e-04 6.139839e-05 0.02090994 0.12573496
path:hsa05016 0.830845771 6.852905e-04 8.793420e-05 0.02090994 0.89742129
                pORA.fdr    pComb.fdr
path:hsa05010 0.001999556 0.0004830776
path:hsa05110 0.007975357 0.0006039857
path:hsa04145 0.011882215 0.0008529668
path:hsa05152 0.017204791 0.0017805532
path:hsa04722 0.016789618 0.0017805532
path:hsa05016 0.016789618 0.0021250764

```

```

> head(Summary(peRes, pathNames = kpn, totalAcc = FALSE, totalPert = FALSE,
+             pAcc = FALSE, pORA = FALSE, comb.pv = NULL, order.by = "pPert"))

```

```

                pathNames      pPert  pPert.fdr
path:hsa03013          RNA transport 0.004975124 0.01414496
path:hsa03460          Fanconi anemia pathway 0.004975124 0.01414496
path:hsa04020          Calcium signaling pathway 0.004975124 0.01414496
path:hsa04060 Cytokine-cytokine receptor interaction 0.004975124 0.01414496
path:hsa04062          Chemokine signaling pathway 0.004975124 0.01414496
path:hsa04064          NF-kappa B signaling pathway 0.004975124 0.01414496

```

1.5 Graphical representation of results

To visualize the summary of the Pathway-Express results use the function `plot` (see Fig. 1):

```

> plot(peRes)

> plot(peRes, c("pAcc", "pORA"), comb.pv.func = compute.normalInv, threshold = .01)

```

Pathway level statistics can also be displayed one at a time using the function `plot` (see Fig. 2):

```

> plot(peRes@pathways[["path:hsa05216"]], type = "two.way")

> plot(peRes@pathways[["path:hsa05216"]], type = "boot")

```

To visualize the propagation across the pathway, two functions - `peNodeRenderInfo` and `peEdgeRenderInfo` - are provided to extract the required information from a `pePathway` object:

```

> p <- peRes@pathways[["path:hsa05216"]]
> g <- layoutGraph(p@map, layoutType = "dot")
> graphRenderInfo(g) <- list(fixedsize = FALSE)
> edgeRenderInfo(g) <- peEdgeRenderInfo(p)
> nodeRenderInfo(g) <- peNodeRenderInfo(p)
> renderGraph(g)

```

This is the *Thyroid cancer* signaling pathway and is shown in Fig. 3. Another example is the *T cell receptor signaling pathway* and is presented in Fig. 4.

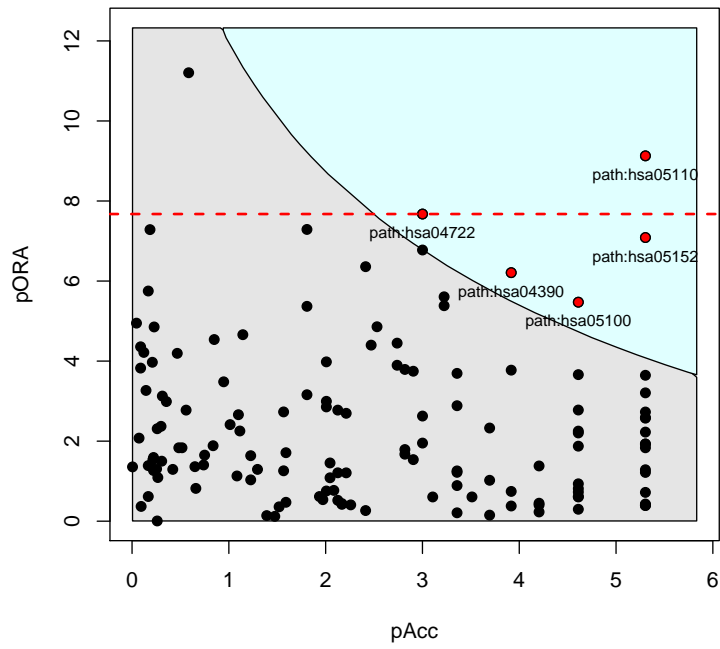
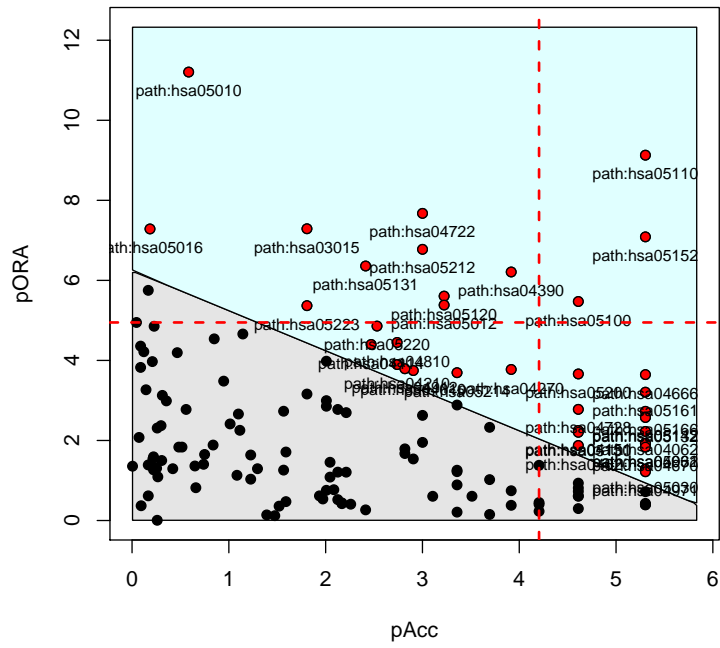


Figure 1: Two-way plot of Pathway-Express result

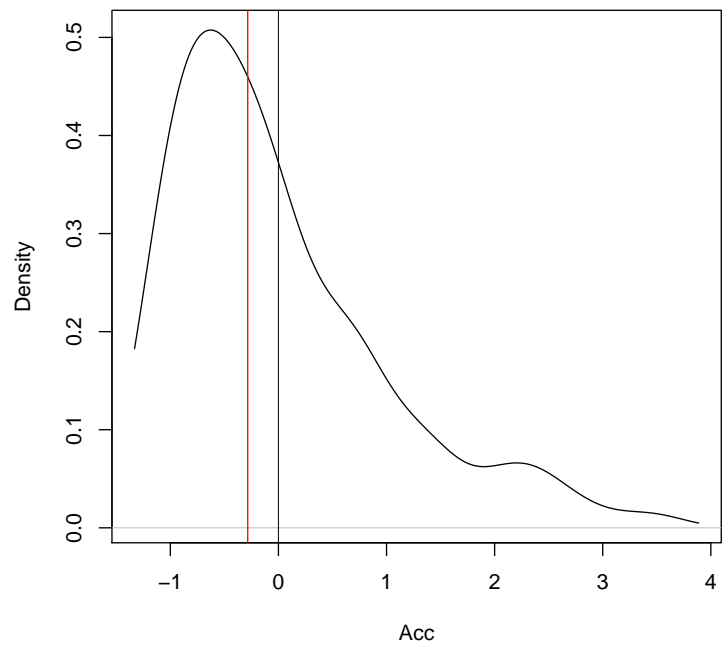
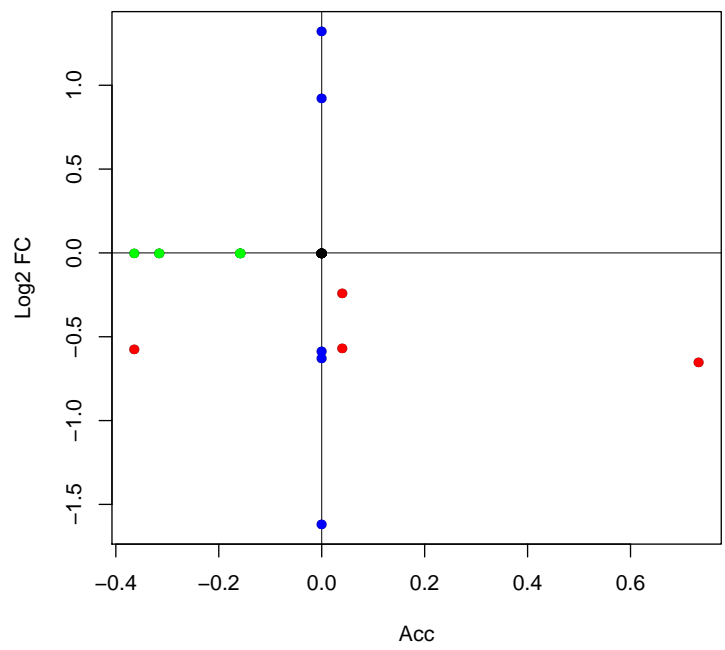


Figure 2: Pathway level statistics: perturbation accumulation versus the measured expression change (above) and the bootstrap simulations of the perturbation accumulation (below).

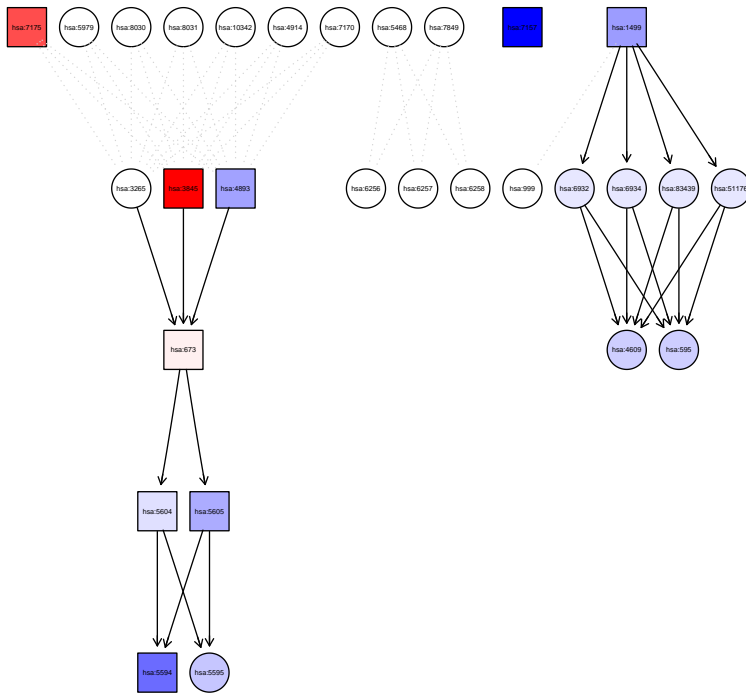


Figure 3: Perturbation propagation on the *Thyroid cancer signaling pathway*.

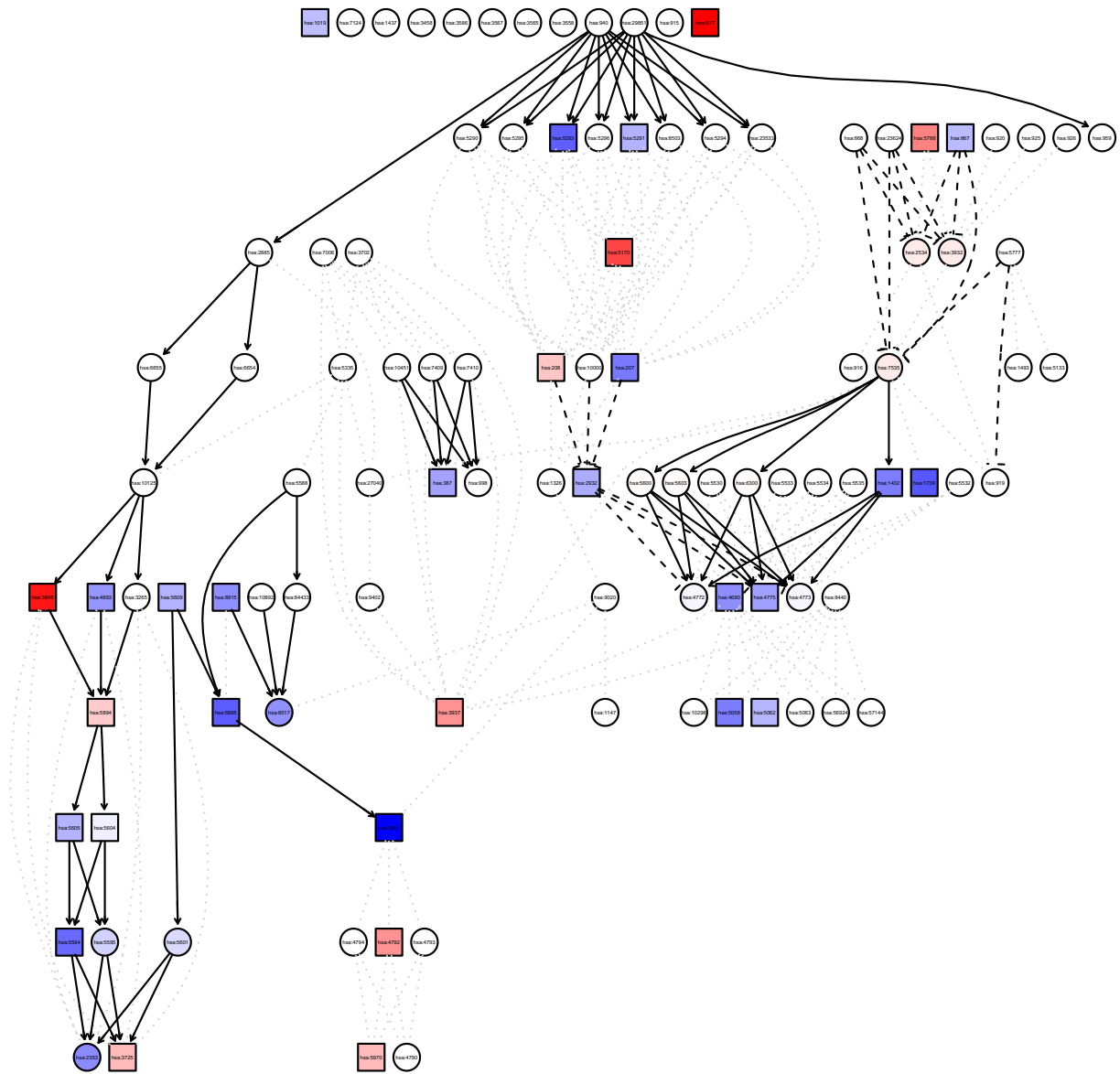


Figure 4: Perturbation propagation on the *T cell receptor signaling pathway*.

2 Primary dis-regulation

Primary dis-regulation analysis (pDis) is a tool for the analysis of signaling pathways. This is the original implementation of the algorithm introduced in [1]. This method takes into consideration the primary dis-regulation of a given gene itself and the effects of signaling coming from upstream. Similar to Pathway Express, primary dis-regulation uses two sources of data: one is the experiment data and the other is the database of pathways.

The pathway database can be obtained from KEGG as explained in section Pathway database.

For example, to download and parse the signaling pathways available in KEGG use:

```
> require(graph)
> require(ROntoTools)
> kpg <- keggPathwayGraphs("hsa", verbose = FALSE)
```

The above code will load the available cached data for human (i.e., KEGG id *hsa*). To update the cache and download the latest KEGG pathways available use the `updateCache` parameter:

```
> kpg <- keggPathwayGraphs("hsa", updateCache = TRUE, verbose = TRUE)
```

This command is time consuming and depends on the available bandwidth.

To retrieve the title of the pathways and not just their ids the function `keggPathwayNames` can be used:

```
> kpn <- keggPathwayNames("hsa")
> head(kpn)
```

```
          path:hsa03008          path:hsa03013
"Ribosome biogenesis in eukaryotes"      "RNA transport"
          path:hsa03015          path:hsa03018
"mRNA surveillance pathway"              "RNA degradation"
          path:hsa03320          path:hsa03460
"PPAR signaling pathway"                 "Fanconi anemia pathway"
```

As an example, a publicly available data is provided in the package. For more information please refer to Experimental data section.

```
> load(system.file("extdata/E-GEOD-21942.topTable.RData", package = "ROntoTools"))
> head(top)
```

```
          logFC      P.Value    adj.P.Val    entrez
200946_x_at -1.0175141 5.833411e-13 4.172652e-09 hsa:2746
228697_at   -3.6479368 7.985427e-13 4.172652e-09 hsa:135114
210254_at    3.2807123 3.086572e-12 9.677020e-09  hsa:932
234726_s_at -0.9792301 7.368175e-12 1.760593e-08 hsa:64418
215905_s_at -1.7733135 7.861797e-12 1.760593e-08 hsa:9410
235542_at   -0.9447467 1.617944e-11 2.536288e-08 hsa:200424
```

Select differentially expressed genes at 1% and save their fold change in a vector *fc* and their p-values in a vector *pv*:

```

> fc <- top$logFC[top$adj.P.Val <= .01]
> names(fc) <- top$entrez[top$adj.P.Val <= .01]
> pv <- top$P.Value[top$adj.P.Val <= .01]
> names(pv) <- top$entrez[top$adj.P.Val <= .01]
> head(fc)

    hsa:2746 hsa:135114    hsa:932 hsa:64418    hsa:9410 hsa:200424
-1.0175141 -3.6479368  3.2807123 -0.9792301 -1.7733135 -0.9447467

> head(pv)

    hsa:2746    hsa:135114    hsa:932    hsa:64418    hsa:9410    hsa:200424
5.833411e-13 7.985427e-13 3.086572e-12 7.368175e-12 7.861797e-12 1.617944e-11

```

Alternatively, an analysis with all genes can be performed:

```

> fcAll <- top$logFC
> names(fcAll) <- top$entrez
> pvAll <- top$P.Value
> names(pvAll) <- top$entrez

```

The reference contains all the genes measured in the analysis:

```

> ref <- top$entrez
> head(ref)

[1] "hsa:2746"    "hsa:135114" "hsa:932"     "hsa:64418"  "hsa:9410"
[6] "hsa:200424"

```

2.1 Pathway analysis and results summary

Here are the input needed to run a sample test:

- the pathway database with the experiment specific gene significance - `kpg`
- the experiment data - `fc` and `ref`

To perform the analysis the function `pDis` is used (increase the parameter `nboot` to obtain more accurate results):

```

> pDisRes <- pDis(x = fc, graphs = kpg, ref = ref, nboot = 200, verbose = FALSE)

```

The result object can be summarized in a table format with the desired columns using the function `Summary`:

```

> head(Summary(pDisRes))

      totalpDis totalpDisNorm      ppDis      pORA      pComb
path:hsa05010  38.59681    -1.4298642 0.134328358 1.360242e-05 2.596938e-05
path:hsa05110  20.46240      1.5327388 0.129353234 1.085083e-04 1.708715e-04
path:hsa04390  36.64334      2.4790906 0.009950249 2.008410e-03 2.362243e-04
path:hsa04145  37.06796      0.7098611 0.417910448 2.424942e-04 1.033375e-03
path:hsa03015  23.11144     -1.3701226 0.164179104 6.821488e-04 1.130816e-03

```

```

path:hsa04722  32.98817      -0.8724431  0.378109453  4.644830e-04  1.694286e-03
                ppDis.fdr      pORA.fdr      pComb.fdr
path:hsa05010  0.5807726  0.001999556  0.003817499
path:hsa05110  0.5762099  0.007975357  0.011574993
path:hsa04390  0.2437811  0.026839656  0.011574993
path:hsa04145  0.7754001  0.011882215  0.033245989
path:hsa03015  0.6375813  0.016789618  0.033245989
path:hsa04722  0.7754001  0.016789618  0.041510008

```

```

> head(Summary(pDisRes, pathNames = kpn, totalpDis = FALSE,
+             pORA = FALSE, comb.pv = NULL, order.by = "ppDis"))

```

```

                pathNames      ppDis ppDis.fdr
path:hsa04976                Bile secretion 0.004975124 0.2437811
path:hsa04390                Hippo signaling pathway 0.009950249 0.2437811
path:hsa05030                Cocaine addiction 0.009950249 0.2437811
path:hsa05142 Chagas disease (American trypanosomiasis) 0.009950249 0.2437811
path:hsa05143                African trypanosomiasis 0.009950249 0.2437811
path:hsa05031                Amphetamine addiction 0.014925373 0.2437811

```

References

- [1] S. Ansari, C. Voichița, M. Donato, R. Tagett, and S. Drăghici. A novel pathway analysis approach based on the unexplained dysregulation of genes. *Proceedings of the IEEE*, PP(99):1–14, March 2016.
- [2] V. Desai, P. Khatri, A. Done, A. Friedman, M. Tainsky, and S. Draghici. A novel bioinformatics technique for predicting condition-specific transcription factor binding sites. In *Proc. of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, San Diego, USA, November 14-15 2005.
- [3] S. Draghici, P. Khatri, A. L. Tarca, K. Amin, A. Done, C. Voichita, C. Georgescu, and R. Romero. A systems biology approach for pathway level analysis. *Genome Research*, 17(10):1537–1545, 2007.
- [4] S. Draghici, S. Sellamuthu, and P. Khatri. Babel’s tower revisited: a universal resource for cross-referencing across annotation databases. *Bioinformatics*, 22(23):2934–2939, 2006.
- [5] S. Drăghici, P. Khatri, P. Bhavsar, A. Shah, S. A. Krawetz, and M. A. Tainsky. Onto-tools, the toolkit of the modern biologist: Onto-express, onto-compare, onto-design and onto-translate. *Nucleic Acids Research*, 31(13):3775–81, July 2003.
- [6] S. Drăghici, P. Khatri, R. P. Martins, G. C. Ostermeier, and S. A. Krawetz. Global functional profiling of gene expression. *Genomics*, 81(2):98–104, February 2003.
- [7] S. Drăghici, P. Khatri, A. Shah, and M. Tainsky. Assessing the functional bias of commercial microarrays using the Onto-Compare database. *BioTechniques*, Microarrays and Cancer: Research and Applications:55–61, March 2003.

- [8] P. Khatri, P. Bhavsar, G. Bawa, and S. Drăghici. Onto-tools: an ensemble of web-accessible, ontology-based tools for the functional design and interpretation of high-throughput gene expression experiments. *Nucleic Acids Research*, 32:W449–56, Jul 2004.
- [9] P. Khatri, V. Desai, A. L. Tarca, S. Sellamuthu, D. E. Wildman, R. Romero, and S. Draghici. New Onto-Tools: Promoter-Express, nsSNPCounter, and Onto-Translate. *Nucleic Acids Research*, 34:W626–31, 2006.
- [10] P. Khatri, S. Draghici, G. C. Ostermeier, and S. A. Krawetz. Profiling gene expression using Onto-Express. *Genomics*, 79(2):266–270, February 2002.
- [11] P. Khatri, S. Draghici, A. L. Tarca, S. S. Hassan, and R. Romero. A system biology approach for the steady-state analysis of gene signaling networks. In *12th Iberoamerican Congress on Pattern Recognition*, Valparaiso, Chile, November 13-16 2007.
- [12] P. Khatri, S. Sellamuthu, P. Malhotra, K. Amin, A. Done, and S. Drăghici. Recent additions and improvements to the Onto-Tools. *Nucleic Acids Research*, 33(Web server issue), Jul 2005.
- [13] P. Khatri, C. Voichita, K. Kattan, N. Ansari, A. Khatri, C. Georgescu, A. L. Tarca, and S. Drăghici. Onto-Tools: New additions and improvements in 2006. *Nucleic Acids Research*, 37(Web Server issue), July 2007.
- [14] A. L. Tarca, S. Draghici, P. Khatri, S. S. Hassan, P. Mittal, J. sun Kim, C. J. Kim, J. P. Kusanovic, and R. Romero. A novel signaling pathway impact analysis (SPIA). *Bioinformatics*, 25(1):75–82, 2009.
- [15] C. Voichita, M. Donato, and S. Draghici. Incorporating gene significance in the impact analysis of signaling pathways. *Proceedings of the International Conference on Machine Learning Applications (ICMLA)*, Dec. 2012.