Using splineTimeR package

Agata Michna and Herbert Braselmann

June 8, 2026

Research Unit Radiation Cytogenetics, Group Integrative Biology
Helmholtz Zentrum München

braselm@helmholtz-muenchen.de

Contents

1 Overview

This package provides functions for differential gene expression analysis of time-course gene expression data, pathway enrichment analysis and time dependent gene regulatory network reconstruction. The functions in this package gather and simplify the usage of other Bioconductor packages and R-functions.

A natural cubic spline regression model for an experimental two-way designs with one treatment factor and time as continuous variable is fitted to the experimental time-course data. Two treatment groups (e.g. control and treated) have to be provided. Time dependent expression data must be available for each group. Parameters of fitted spline regression models are used to define differentially expressed over time genes. Those genes may further be used for pathway enrichment analysis and/or the reconstruction of time dependent gene regulatory association networks. The workflow implemented in this package is explained and discussed in detail in Michna et al. (2016). Expression microarray data generated in our laboratory are are available from the ArrayExpress database (accession number E-MTAB-4829).

2 Data preparation

2.1 Simulation of data

Two simulated gene expression data sets, generated with the open-source simulator GeneNetWeaver (GNW), are included in this package and serve as test data to demonstrate the usage of the functions (Schaffter et al., 2011; Marbach et al., 2009). The *in silico* expression data were simulated based on the network structure of a 2000-gene sub-network from the Reactome functional interaction network (Reactome project, `http://www.reactome.org/`). The sub-network was converted to a dynamical network model without autoregulatory interactions (self-loops). The data were simulated based on the `"ODEs"` model. Two types of time-series experiments were chosen: `"Time Series as in DREAM4"` and `"Multifactorial"`. Gene expression data were simulated for 48 time points after perturbations. For more details see GNW User Manual (Schaffter et al., 2010, `http://gnw.sourceforge.net`).

The names of `"Time Series as in DREAM4"` and `"Multifactorial"` simulation experiments were changed to `"T1"` and `"T2"`, respectively. From the generated data sets, eight time points are provided (1, 4, 8, 16, 24, 32, 40 and 48). The numbers correspond to the same time units after perturbation (e.g. minutes, hours, days, ect.). Replicates for both time-course experiments were generated by the addition of the normally distributed random errors with a standard deviation of 0.05 to the expression values for each time point. Subsequently, the entire dataset was normalized between 0 and 1.

2.2 Required data design

Gene expression data must be provided as a Biobase object of class `ExpressionSet`. `ExpressionSet` objects are capable of storing log-ratios or log-values of expressions from multiple microarrays as well as feature data and phenotypic data of the samples. Functions included in this R-package require phenotypic data in the `ExpressionSet` with three columns named `SampleName`, `Time` and `Treatment`. Two types of treatments (e.g. control and treated) have to be specified. This defines groups to compare. Optionally `Replicate` or any other column can be added. If the `Replicate` column is not present, all samples are considered as single replicates.

Replicates are not required. The time points for compared treatment groups should be identical. Visualisation of results is demonstrated using simulated data (`TCsimData`).

```
> library(splineTimeR)
> data(TCsimData)
> head(pData(TCsimData),8)
```

```
  SampleName Time Treatment Replicate
1     T1_1_A    1        T1         A
2     T1_4_A    4        T1         A
3     T1_8_A    8        T1         A
4    T1_16_A   16        T1         A
5    T1_24_A   24        T1         A
6    T1_32_A   32        T1         A
7    T1_40_A   40        T1         A
8    T1_48_A   48        T1         A
```

3 Time-course data differential expression analysis

3.1 Model and model parameters

To simulate a non-linear behaviour of gene expression over time a temporal trend using a natural cubic spline regression is fitted to the measured data points. The function `splineDiffExprs` compares the time dependent behaviour of genes in two different groups.

The mathematical model of natural cubic spline regression is defined by:

$$\begin{aligned} y = y(t,x) = b_0 + b_1 B_1(t-t_0) + b_2 B_2(t-t_0) + ... + b_m B_m(t-t_0) + \\ + x(d_0 + d_1 B_1(t-t_0) + d_2 B_2(t-t_0) + ... + d_m B_m(t-t_0)) \end{aligned} \quad (1)$$

where b_1, b_2, ..., b_m are the spline coefficients in the reference group (e.g. control group) and d_1, d_2, ..., d_m are differential spline coefficients for the compared group (e.g. treated group). $B_1(t-t_0)$, $B_2(t-t_0)$, ..., $B_m(t-t_0)$ are spline base functions and t_0 is the time of the first measurement. For $x = 0$, $y = y_{reference}$ and for $x = 1$, $y = y_{compared}$.

Dependent on the data type and the number of measured time points, user has to define the number of degrees of freedom `df`. Choosing effective degrees of freedom in range 3-5 is reasonable.

3.2 Statistical methods and description of output table

Time dependent differential expression of a gene is determined by the application of empirical Bayes moderate F-statistics on the differences of coefficient values of the fitted natural cubic spline regression models for the same gene in the two compared treatment groups (Smyth, 2004). In other words, comparing the coefficient values of the fitted splines in both groups allows the detection of differences in the shape of the curves, which represent the gene expressions changes over time.

The `reference` parameter specify which treatent group should be considered as reference. The `intercept` defines, which coefficients are considered for the statistical analysis. Two options are possible. If `intercept=TRUE`, the F-test includes all parameters, i.e shape parameters and intercept at the first given time point. If `intercept=FALSE`, only shape parameters are considered.

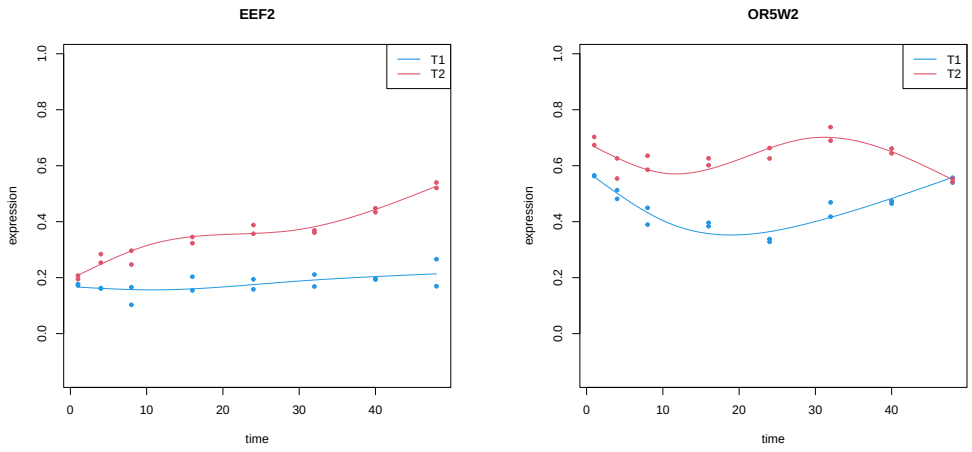
The `cutoff.adj.pVal` input parameter defines the Benjamini-Hochberg adjusted p-value cut-off for genes to be considered as differentially expressed. The `splineDiffExprs` function generates a output table, where the first columns contain all feature data of the `ExpressionSet` object (`fData(eSetObject)`), if any feature data were defined. Otherwise, only one column `row_IDs`, containing the row names is created. The b_0, b_1,..., b_m coefficients correspond to the reference model parameters. The d_0, d_1,..., d_m coefficients represent the differences between the reference model parameters and the model parameters in the compared group. `AveExprs` refers to the average log2-expression for a probe (representing a gene) over all arrays. The `F` column contains moderate F-statistics, `P.Value` raw p-value and `adj.P.Value` Benjamini-Hochberg adjusted p-value.

```
> diffExprs <- splineDiffExprs(eSetObject = TCsimData, df = 3,
+                              cutoff.adj.pVal = 0.01, reference = "T1",
+                              intercept = TRUE)

-------------------------------------------------
Differential analysis done for df = 3 and adj.P.Val <= 0.01
Number of differentially expressed genes:  952

> head(diffExprs, 3)

      row_IDs       b_0         b_1         b_2         b_3        d_0
BRAF     BRAF 0.2248186 -0.06672491 -0.11466685 -0.01113329 0.04537218
EEF2     EEF2 0.1662325  0.03093095  0.02415684  0.05805703 0.04218390
OR5W2   OR5W2 0.5600514 -0.11141556 -0.26764248  0.13582290 0.10798569
            d_1        d_2        d_3   AveExpr         F      P.Value
BRAF  0.09884069 0.32667028  0.1362235 0.2616200 100.47655 8.552316e-19
EEF2  0.03907564 0.36776057  0.1663112 0.2632523  75.34076 8.442937e-17
OR5W2 0.29355341 0.08948516 -0.1577280 0.5454758  71.98754 1.723753e-16
         adj.P.Val
BRAF  1.710463e-15
EEF2  8.442937e-14
OR5W2 1.149169e-13
```

4 Visualization of time-course data with natural cubic spline regression model

The `splinePlot` function visualises the time dependent behaviour of genes in two treatment groups. The natural cubic spline regression curves fitted to discrete, time dependent expression data are plotted. One plot shows two curves - representing the reference group and the compared group, respectively. Reference group is specified by `reference` parameter. User can decide which genes to plot by setting `toPlot` input parameter, otherwise all genes included in `ExpressionSet` object are plotted. Defined genes to plot have to be included in `ExpressionSet` object. The plots are saved in a .pdf file.

```
> splinePlot(eSetObject = TCsimData, df = 3,
+            reference = "T1", toPlot = c("EEF2","OR5W2"))

Log of function splinePlot
--------------------------
Header of design matrix used for model fit:
[1] "(Intercept)" "d_T2"        "b_1"         "b_2"         "b_3"
[6] "d_T2:b_1"    "d_T2:b_2"    "d_T2:b_3"

Model coefficient names:
[1] "(Intercept)" "d_T2"        "b_1"         "b_2"         "b_3"
[6] "d_T2:b_1"    "d_T2:b_2"    "d_T2:b_3"

pdf-filename for plot: plots_df3_spline.pdf
```



5 Pathway enrichment analysis

The `pathEnrich` function performs a pathway enrichment analysis of defined genes (`geneList`). Collection of curated gene sets (`geneSets`) representing metabolic and signaling pathways has to be downloaded from a database of interest in Gene Matrix Transposed file format (*.gmt), where each gene set is described by a pathway name, a description, and the genes in the gene set. Two examples are shown bellow to demonstrate how to define `geneSets` object.

The variable `universe` represents a total number of genes that were probed in the initial experiment, e.g. the number of all genes on a microarray. If `universe` is not defined, `universe` is the number of all genes that can be mapped to any pathways in the chosen database.

The result table contains the name of the enriched pathway (`pathway`), the gene set description (`description`), the total number of genes in the pathway (`genes_in_pathway`), the number of matched genes from the given gene set (`matches`), the percentage of matched genes referred to the total number of known genes in the pathway (`%_match`), the p-value (`pValue`), Benjamini-Hochberg adjucted p-value (`adj.pValue`), and a list of genes that are part of the pathway (`overlap`). Statistical significance was determined by hypergeometric test (Fisher's exact test).

Gene names for pathway enrichment must be provided in HGNC-approved gene nomenclature, otherwise the genes are not included in the analysis. Simulated time-course expression data were derived from Reactome functional interaction network containing 6536 unique identifiers (nodes), meaning that in our experiment universe is equal to 6536.

Example 1 - gene sets from the Molecular Signatures Database (MSigDB collections), `http://www.broadinstitute.org/gsea/msigdb/collections.jsp` (Subramanian et al., 2005)

```
> ## Not run
> ## Download .gmt file 'c2.all.v5.0.symbols.gmt' (all curated gene sets,
> ## gene symbols) from the Broad,
> ## http://www.broad.mit.edu/gsea/downloads.jsp#msigdb, then
> geneSets <- getGmt("/path/to/c2.all.v5.0.symbols.gmt")
> ## load ExpressionSet object containing simulated time-course data
> data(TCsimData)
> ## check for differentially expressed genes
> diffExprs <- splineDiffExprs(eSetObject = TCsimData, df = 3,
+                              cutoff.adj.pVal = 0.01, reference = "T1")
> ## use differentially expressed genes for pathway enrichment analysis
> enrichPath <- pathEnrich(geneList = rownames(diffExprs), geneSets = geneSets,
+                          universe = 6536)
```

```
> ## End(Not run)
```

Example 2 - gene sets from the Reactome Pathway Database, `http://www.reactome.org/pages/download-data/` (Croft et al., 2014)

```
> ## Not run
> ## Download and unzip .gmt.zip file 'ReactomePathways.gmt.zip'
> ## ("Reactome Pathways Gene Set" under "Specialized data formats") from
> ## the Reactome website http://www.reactome.org/pages/download-data/, then
> geneSets <- getGmt("/path/to/ReactomePathways.gmt")
> data(TCsimData)
> diffExprs <- splineDiffExprs(eSetObject = TCsimData, df = 3,
+                              cutoff.adj.pVal = 0.01, reference = "T1")
> enrichPath <- pathEnrich(geneList = rownames(diffExprs), geneSets = geneSets,
+                          universe = 6536)
> ## End(Not run)
```

6 Gene association network reconstruction of time-course data

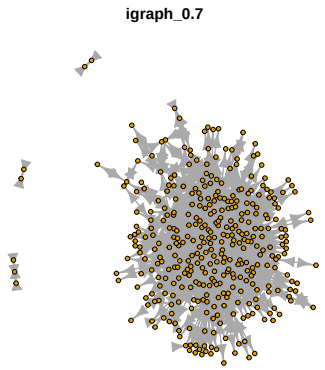
The `splineNetRecon` function reconstructs gene association networks from time-course data. Based on given `ExpressionSet` object, longitudinal data object is created. Subsequantly the function estimates edges using partial correlation method with shrinkage approach applying `ggm.estimate.pcor` and `network.test.edges` functions. As a result an object or list of object of class `igraph` is created.

Gene association network reconstruction is done for a selected type of `Treatment`. This allows to identify regulatory associations between genes under a certain condition (treatment). First, a `longitudinal` data object of the gene expression data with possible replicates is created. This object is used to estimate the partial correlation with the selected shrinkage method (`dynamic` or `static`) with the `ggm.estimate.pcor` function (for details see `ggm.estimate.pcor` function help). Finally, the the `network.test.edges` function estimates the probabilities for all possible edges and lists them in descending order (for details see `network.test.edges` help).

An object or list of objects of class `igraph` with all edges that exceeded the probability-cutoff (`cutoff.ggm`) is returned. If more than one value for `cutoff.ggm` is defined than function returns a `list` of objects of class `igraph` for each defined `cutoff.ggm` value. Otherwise a single object of class `igraph` with one selected probability is returned.

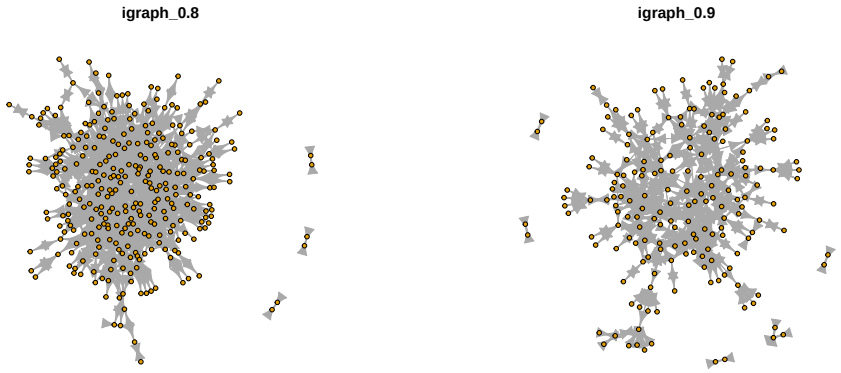
For one defined cut-off:

```
> igr <- splineNetRecon(eSetObject = TCsimData, treatmentType = "T2",
+                       probesForNR = rownames(diffExprs),
+                       cutoff.ggm = 0.7, method = "dynamic")

------------------------------------------------------------
Longitudinal object
------------------------------------------------------------
$time
[1]  1  4  8 16 24 32 40 48

$repeats
[1] 2 2 2 2 2 2 2 2

Estimating optimal shrinkage intensity lambda (correlation matrix): 0.5457

Estimate (local) false discovery rates (partial correlations):
Step 1... determine cutoff point
Step 2... estimate parameters of null distribution and eta0
Step 3... compute p-values and estimate empirical PDF/CDF
Step 4... compute q-values and local fdr


-------------- igraph_0.7 --------------
Significant edges:  838
    Corresponding to  0.19 %  of possible edges
Number of nodes:  359
> plot(igr, vertex.label = NA, vertex.size = 3, main = "igraph_0.7")
```



For more defined cut-off values:

```
> igr <- splineNetRecon(eSetObject = TCsimData, treatmentType = "T2",
+                       probesForNR = rownames(diffExprs),
+                       cutoff.ggm = c(0.8,0.9), method = "dynamic")

------------------------------------------------------------
Longitudinal object
------------------------------------------------------------
$time
[1]  1  4  8 16 24 32 40 48

$repeats
[1] 2 2 2 2 2 2 2 2

Estimating optimal shrinkage intensity lambda (correlation matrix): 0.5457

Estimate (local) false discovery rates (partial correlations):
Step 1... determine cutoff point
Step 2... estimate parameters of null distribution and eta0
Step 3... compute p-values and estimate empirical PDF/CDF
Step 4... compute q-values and local fdr


-------------- igraph_0.8 --------------
Significant edges:  587
    Corresponding to  0.13 %  of possible edges
Number of nodes:  302

-------------- igraph_0.9 --------------
Significant edges:  275
    Corresponding to  0.06 %  of possible edges
Number of nodes:  184

> plot(igr[[1]], vertex.label = NA, vertex.size = 3, main = "igraph_0.8")
> plot(igr[[2]], vertex.label = NA, vertex.size = 3, main = "igraph_0.9")
```



7 Scale-free properties of a network

Biological networks are thought to be scale-free. Scale-free networks follow a power-law distribution of the degrees of nodes in the network (Barabási and Albert, 1999; Albert, 2005). Power-law distribution is characterised by the exponent γ. The probability that a node has the degree k (k connections) is given by:

$$P(k) \sim k^{-\gamma} \tag{2}$$

or after logarithmic transformation:

$$logP(k) \sim -\gamma logk \tag{3}$$

For most biological networks the exponent γ ranges between 2 and 3 (Barabási and Oltvai, 2004). Nevertheless, subnetworks of a scale free-network are not necessarily scale-free (Stumpf et al., 2005).

The `networkProperties` function plots the degree distribution of nodes in a given network (`igraph`). For comparison, the degree exponents of the Reactome and BioGRID networks are shown (Croft et al., 2014; Stark et al., 2006). Both of these networks are derived from functional interaction pairs obtained from Reactome and BioGRID repositories. The functional interaction pairs were extracted from mentioned databases and are provided in FIs data package.

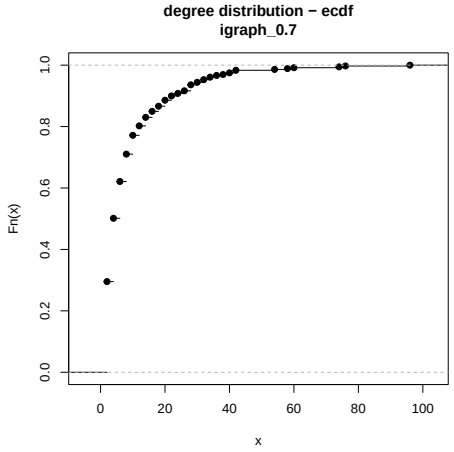
```
> library(FIs)
> data(FIs)
> names(FIs)
> head(FIs$FIs_Reactome)
> head(FIs$FIs_BioGRID)
```

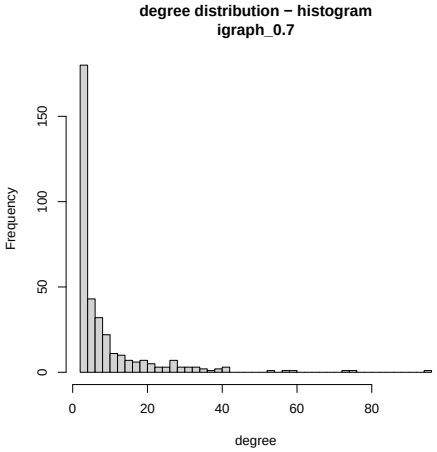
For each given `igraph` three types of plots are created: empirical cumulative distribution, degree distribution and power-law degree distribution on log-log scale with fitted trend line. Additionally, a summary table containing the number of nodes, the number of edges and the degree exponents for each given network is returned.

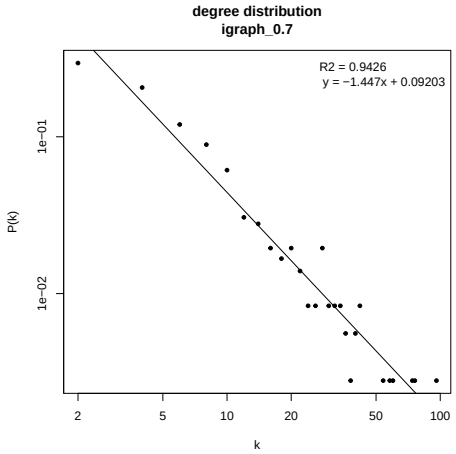
```
> igr <- splineNetRecon(eSetObject = TCsimData, treatmentType = "T2",
+                       probesForNR = rownames(diffExprs),
+                       cutoff.ggm = c(0.7,0.8,0.9), method = "dynamic")

------------------------------------------------------------------
Longitudinal object
------------------------------------------------------------------
$time
[1]  1  4  8 16 24 32 40 48

$repeats
[1] 2 2 2 2 2 2 2 2

Estimating optimal shrinkage intensity lambda (correlation matrix): 0.5457

Estimate (local) false discovery rates (partial correlations):
Step 1... determine cutoff point
Step 2... estimate parameters of null distribution and eta0
Step 3... compute p-values and estimate empirical PDF/CDF
Step 4... compute q-values and local fdr


-------------- igraph_0.7 --------------
Significant edges:  838
    Corresponding to  0.19 %  of possible edges
Number of nodes:  359

-------------- igraph_0.8 --------------
Significant edges:  587
    Corresponding to  0.13 %  of possible edges
Number of nodes:  302

-------------- igraph_0.9 --------------
Significant edges:  275
    Corresponding to  0.06 %  of possible edges
Number of nodes:  184

> scaleFreeProp <- networkProperties(igr)
> head(scaleFreeProp)
```

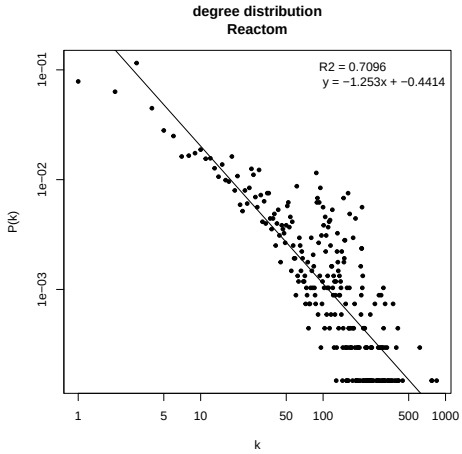
	nodes	edges	degree_exponent
igraph_0.7	359	1676	1.4466
igraph_0.8	302	1174	1.4214
igraph_0.9	184	550	1.4494
Reactome	6770	148733	1.2531
BioGRID	19127	213150	1.5292

Examplary figures ploted in .pdf files are shown below.









References

Albert, R. (2005). Scale-free networks in cell biology. *Journal of Cell Science*, 118:4947–4957.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.

Barabási, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113.

Croft, D., Mundo, A. F., Haw, R., Milacic, M., Weiser, J., Wu, G., Caudy, M., Garapati, P., Gillespie, M., Kamdar, M. R., Jassal, B., Jupe, S., Matthews, L., May, B., Palatnik, S., Rothfels, K., Shamovsky, V., Song, H., Williams, M., Birney, E., Hermjakob, H., Stein, L., and D'Eustachio, P. (2014). The reactome pathway knowledgebase. *Nucleic Acids Research*, 42(Database issue):472–477.

Marbach, D., Schaffter, T., Mattiussi, C., and Floreano, D. (2009). Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology*, 16(2):229–239.

Michna, A., Braselmann, H., Selmansberger, M., Dietz, A., Hess, J., Gomolka, M., Hornhardt, S., Blüthgen, N., Zitzelsberger, H., and Unger, K. (2016). Natural cubic spline regression modeling followed by dynamic network reconstruction for the identification of radiation-sensitivity gene association networks from time-course transcriptome data. *PLoS One*, 11(8):e0160791.

Reactome project (n.d.). Reactome functional interaction network. Retrieved September 25, 2015 from http://www.reactome.org/.

Schaffter, T., Marbach, D., and Floreano, D. (2011). GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270.

Schaffter, T., Marbach, D., and Gilles, R. (2010). *GNW User Manual*.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 3.

Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34(Database issue):535–539.

Stumpf, M. P. H., Wiuf, C., and May, R. M. (2005). Subnets of scale-free networks are not scale-free: Sampling properties of networks. *PNAS*, 102(12):4221–4224.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, 102(43):15545–15550.