BNrich: a novel pathway enrichment analysis based on BN

Samaneh Maleknia

Department of Bioinformatics, Institute of Biochemistry and Biophysics, Tehran University

August 20, 2019

Abstract

This package has developed a tool for performing a novel pathway enrichment analysis based on Bayesian network (BNrich) to investigate the topology features of the pathways. This algorithm as a biologically intuitive, method, analyzes of most structural data acquired from signaling pathways such as causal relationships between genes using the property of Bayesian networks and also infer finalized networks more conveniently by simplifying networks in the early stages and using Least Absolute Shrinkage Selector Operator (LASSO). impacted pathways are ultimately prioritized the by Fisher's Exact Test on significant parameters. Here, we provide an instance code that applies BNrich in all of the fields described above.

Contents

1 Introduction

This document offers an introductory overview of how to use the package. The BNrich tool uses Bayesian Network (BN) in a new topology-based pathway analysis (TPA) method. The BN has been demonstrated as a beneficial technique for integrating and modeling biological data into causal relationships (1–5). The proposed method utilizes BN to model variations in downstream components (children) as a consequence of the change in upstream components (parents). For this purpose, The method employs 187 KEGG human non-metabolic pathways (6–8) which their cycles were eliminated manually by a biological intuitive, as BN structures and gene expression data to estimate its parameters (9,10). The cycles of inferred networks were eliminated on the basis of biologically intuitive rules instead of using computing algorithms (11). The inferred networks are simplified in two steps; unifying genes and LASSO. Similarly, the originally continuous gene expression data is used to BN parameters learning, rather than discretized data (9). The algorithm estimates regression coefficients by continuous data based on the parameter learning techniques in the BN (12,13). The final impacted pathways are gained by Fisher's exact test. This method can represent effective genes and biological relations in impacted pathways based on a significant level.

2 Unifying nods as the first step of simplification

2.1 prepare essential data

The data should be as two data frames in states disease and (healthy) control. The row names of any data frame are KEGG geneID and the number of subjects in any of them should not be less than 20, otherwise the user may encounters error in LASSO step. Initially, we load all the necessary files and data example.

At first, we can load all the 187 preprocessed KEGG pathways which their cycles were removed, the data frame includes information about the pathways and vector of pathway ID.

```
> mapkG <- BNrich:::mapkG
> PathName_final <- BNrich:::PathName_final
> pathway.id <- BNrich:::pathway.id
```

The example data extracted from a part of GSE47756 dataset, the gene expression data from colorectal cancer study (14).

```
> example <- c("GSE93601H "," GSE93601D")
> data(list= example,package = "BNrich")
> dataH <- GSE93601H
> dataD <- GSE93601D
> head(dataH)
           H1       H2       H3       H4       H5       H6       H7       H8
 hsa:1     3.37954  3.3469   3.78383  3.35186  3.2091   3.40245  4.06329  3.43424
 hsa:100   3.1147   3.15981  3.37842  2.69868  3.43759  3.38588  2.95406  3.09631
 hsa:10000 3.21876  2.93611  2.62708  3.13507  2.62864  2.61367  2.7336   2.70867
 hsa:1001  3.4549   3.18683  3.34896  3.36903  3.49353  3.35175  3.27893  3.63678
 hsa:10010 2.17522  2.59843  2.56868  2.95009  2.52181  2.24635  2.05092  2.10438
 hsa:10013 2.992    2.94325  3.22677  2.87371  3.063    2.97679  3.07247  3.08168

> head(dataD)
           D1       D2       D3       D4       D5       D6       D7       D8
 hsa:1     3.29082  3.15924  3.45716  3.15391  3.29514  3.36502  3.63823  3.22192
 hsa:100   3.069    2.97546  2.99117  2.88929  3.00292  2.94948  2.93906  3.36357
```

```
hsa:10000   2.68424   3.24284   3.57435   2.46992   4.57649   3.87179   2.94405   3.54207
hsa:1001    3.27815   2.91081   3.53487   2.95122   2.67742   2.72358   3.10172   3.07123
hsa:10010   2.68051   3.22719   3.58798   2.61269   3.72397   3.29004   2.5843    2.95756
hsa:10013   3.05107   2.86273   3.06863   3.05318   3.04536   2.92021   3.12596   3.0468
```

2.2 Unify data, the first step of simplification

Initially, we need to unify gene products based on 187 imported signaling pathways (mapkG list) in two states disease (dataD) and control (dataH). This is the first simplification step, unifying nodes in signaling pathways with genes those exist in gene expression data.

```
> unify_results <- unify_path(dataH, dataD, mapkG ,pathway.id)
```

This function performs the following processes:

- Split datasets into KEGG pathways
- Delete all gene expression data are not in pathways
- Removes all gene products in pathways are not in dataset platforms
- Remove any pathways with the number of edges is less than 5

The *unify_path* function returns a list contain *data_h,data_d,mapkG1* and *pathway.id1*. data_h and data_d are lists contain data frames related to control and disease objects unified for any signaling pathways. The mapkG1 is a list contains unified signaling pathways and *pathway.id1*is new pathway ID vector based on remained pathways.
In the example dataset, the number of edges in the one pathway becomes less than 5 and are removed:

```
> length(mapkG)
[1] 187

> mapkG1 <- unify_results$mapkG1
> length(mapkG1)
[1] 186
```

As well, the number of edges reduces in the remaining pathways. In first pathway *hsa:01521* the number of edges from 230 reduces to 204:

```
> pathway.id[1]
[1] "hsa:01521"

> mapkG[[1]]
A graphNEL graph with directed edges
Number of Nodes = 79
Number of Edges = 230

> pathway.id1 <- unify_results$pathway.id1
> pathway.id1[1]
```

```
[1] "hsa:01521"

> mapkG1 <- unify_results$mapkG1[[1]]
> mapkG1[[1]]
A graphNEL graph with directed edges
Number of Nodes = 71
Number of Edges = 204
```

3 BN: construct structures and estimate parameters

3.1 construct BN structures

Now we can construct BN structures based on unified signaling pathways and consequently need the results of *unify_path* function.

```
> BN <- BN_struct(unify_results$mapkG1)
```

The *BN_struct* function returns a list contains BNs structures reconstructed from all *mapkG1.*

3.2 The LASSO regression, the second step of simplification

Given that the data used is continuous, each node is modeled as a regression line from its parents (12,15). Thus, on some of these regression lines, the number of these independent variables is high, so in order to avoid the collinearity problem, we need to use the Lasso regression (16,17).

We perform this function for any node with more than one parent, in all BNs achieved by *BN_struct* function, based on control and disease data obtained by *unify_results* function.

```
> data_h <- unify_results$data_h
> data_d <- unify_results$data_d
> LASSO_results <- LASSO_BN(BN,data_h,data_d)
```

The *LASSO_BN* function returns a list contains two lists *BN_H* and *BN_D* are simplified BNs structures based on LASSO regression related to healthy and disease objects. This function lead to reduce number of edges too:

```
> nrow(arcs(BN[[1]]))
[1] 204
> nrow(arcs(LASSO_results$BN_H[[1]]))
[1] 116
> nrow(arcs(LASSO_results$BN_D[[1]]))
[1] 116
```

3.3 Estimate the BN parameters

Now we can estimate (learn) parameters for any BNs based on healthy and disease data lists.

```
> BN_H <- LASSO_results$BN_H
> BN_D <- LASSO_results$BN_D
> esti_results <- esti_par(BN_H,BN_D,data_h,data_d)
```

The *esti_par* function returns a list contains four lists. The *BN_h*, *BN_d*, are lists of BNs which their parameters learned by control and disease objects data. The *coef_h* and *coef_d* are lists of parameters of *BN_h* and *BN_d*.
As you can see in below, node *hsa:1978* in the first BN has one parent. The coefficient in control (healthy) data is **0.6958609** and in disease data is **1.1870730**.

```
> esti_results$BN_h[[1]]$` hsa:1978`
Parameters of node hsa:1978 (Gaussian distribution)
Conditional density: hsa:1978 | hsa:2475
Coefficients:
(Intercept)     hsa:2475
  2.8841264    0.6958609
Standard deviation of the residuals: 0.3489612
```

```
> esti_results$BN_d[[1]]$`hsa:1978`
Parameters of node hsa:1978 (Gaussian distribution)
Conditional density: hsa:1978 | hsa:2475
Coefficients:
(Intercept)     hsa:2475
  0.9046357   1.1870730
Standard deviation of the residuals: 0.2713789
```

4 Testing the equality BNs parameters

4.1 Variance of BNs parameters

We require the variance of the BNs parameters to perform the T-test between the corresponding parameters.

```
> BN_h <- esti_results$BN_h
> BN_d <- esti_results$BN_d
> coef_h <- esti_results$coef_h
> coef_d <- esti_results$coef_d
> var_mat_results<- var_mat (data_h,coef_h,BN_h,data_d,coef_d,BN_d)
```

The *var_mat* function returns a list contains two lists *var_mat_Bh* and *var_mat_Bd* which are the variance-covariance matrixes for any parameters of *BN_h* and *BN_d*. The variance-covariance matrixes for *hsa:1978* in first BN in two states control and disease is as follow:

```
> (var_mat_results$var_mat_Bh[[1]])[5]
[[1]]
        [,1]                    [,2]
```

```
[1,] 10.177073            -3.630152
[2,] -3.630152            1.296990

> (var_mat_results$var_mat_Bd[[1]])[5]
[[1]]
       [,1]               [,2]
[1,]  3.549338            -1.0392040
[2,] -1.039204            0.3053785
```

4.2 Testing the equality BNs parameters

T-test runs between any corresponding parameters between each pair of learned BNs (*BN_h* and *BN_d*) in disease and control states. Assumptions are unequal sample sizes and unequal variances for all samples.

```
> var_mat_Bh <- var_mat_results $var_mat_Bh
> var_mat_Bd <- var_mat_results $var_mat_Bd
> Ttest_results <- parm_Ttest(data_h,coef_h,BN_h,data_d,coef_d,BN_d,var_mat_Bh, var_mat_Bd)
> head(Ttest_results)
```

From	**To**	**pathway.number**	**pathwayID**	**Pval**	**coefficient in disease**	**coefficient in control**	**fdr**
intercept	hsa:2065	1	hsa:01521	0.605294	4.893503	5.535163	6.72E-01
hsa:7039	hsa:2065	1	hsa:01521	2.04E-05	1.072296	-0.21107	6.95E-05
hsa:1950	hsa:2065	1	hsa:01521	0.154223	0.125977	-0.21675	2.11E-01
hsa:4233	hsa:2065	1	hsa:01521	0.083296	-0.63254	-0.33154	1.23E-01
hsa:3084	hsa:2065	1	hsa:01521	0.135981	-0.55586	-0.18792	1.89E-01
hsa:9542	hsa:2065	1	hsa:01521	0.373051	-0.39859	-0.11334	4.49E-01

This function returns a data frame contains T-test results for all parameters in all final BNs. The row that is *intercept* in "*From*" variable, shows significance level for gene product that is shown in "*To*" variable. The rest of the data frame rows shows significance level for any edge of networks.

5 Identification of enriched pathways

In the last step we can determine enriched pathways by own threshold on p-value or fdr. Hence we run the *Fisher's exact test* for any final pathways. As stated above, the Ttest_results is a data frame contains T-test results for all parameters in final BNs achieved by parm_Ttest function and fdr.value A numeric threshold to determine significant parameters (default is 0.05).

```
> BNrich_results <- BNrich(Ttest_results,fdr.value = 0.05,pathway.id1,PathName_final)
> head(BNrich_results)
```

pathwayID	**p.value**	**fdr**	**pathway.number**	**Name**
hsa:05016	2.66E-17	2.47E-15	123	Huntington disease
hsa:05202	1.64E-17	2.47E-15	156	Transcriptional misregulation in cancer

hsa:05012	2.92E-16	1.81E-14	121	Parkinson disease
hsa:05010	1.55E-11	7.19E-10	120	Alzheimer disease
hsa:04144	3.25E-08	1.21E-06	22	Endocytosis
hsa:04714	2.99E-07	9.26E-06	72	Thermogenesis

The *BNrich* function returns a data frame contains the *Fisher's exact test* results for any final pathways.

References:

1. Yu J, Smith VA, Wang PP, Hartemink AJ, Jarvis ED. Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics [Internet]. 2004 Dec 12 [cited 2019 Jun 10];20(18):3594–603. Available from: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bth448
2. Gendelman R, Xing H, Mirzoeva OK, Sarde P, Curtis C, Feiler HS, et al. Bayesian network inference modeling identifies TRIB1 as a novel regulator of cell-cycle progression and survival in cancer cells. Cancer Res. 2017;77(7):1575–85.
3. Luo Y, El Naqa I, McShan DL, Ray D, Lohse I, Matuszak MM, et al. Unraveling biophysical interactions of radiation pneumonitis in non-small-cell lung cancer via Bayesian network analysis. Radiother Oncol [Internet]. 2017 Apr 1 [cited 2019 Jun 10];123(1):85–92. Available from: https://www.sciencedirect.com/science/article/pii/S0167814017300634
4. Agrahari R, Foroushani A, Docking TR, Chang L, Duns G, Hudoba M, et al. Applications of Bayesian network models in predicting types of hematological malignancies. Sci Rep [Internet]. 2018 Dec 3 [cited 2019 Jun 10];8(1):6951. Available from: http://www.nature.com/articles/s41598-018-24758-5
5. Ramos J, Das J, Felty Q, Yoo C, Poppiti R, Murrell D, et al. NRF1 motif sequence-enriched genes involved in ER/PR −ve HER2 +ve breast cancer signaling pathways. Breast Cancer Res Treat [Internet]. 2018 Nov 20 [cited 2019 Jun 10];172(2):469–85. Available from: http://link.springer.com/10.1007/s10549-018-4905-9
6. Zhi-wei J, Zhen-lei Y, Cai-xiu Z, Li-ying W, Jun L, Hong-li W, et al. Comparison of the Network Structural Characteristics of Calcium Signaling Pathway in Cerebral Ischemia after Intervention by Different Components of Chinese Medicine. J Tradit Chinese Med. 2011;31(3):251–5.
7. Lou S, Ren L, Xiao J, Ding Q, Zhang W. Expression profiling based graph-clustering approach to determine renal carcinoma related pathway in response to kidney cancer. Eur Rev Med Pharmacol Sci. 2012;16(6):775–80.
8. Fu C, Deng S, Jin G, Wang X, Yu ZG. Bayesian network model for identification of pathways by integrating protein interaction with genetic interaction data. BMC Syst Biol. 2017;11.
9. Isci S, Ozturk C, Jones J, Otu HH. Pathway analysis of high-throughput biological data within a Bayesian network framework. 2011;27(12):1667–74.
10. Korucuoglu M, Isci S, Ozgur A, Otu HH. Bayesian pathway analysis of cancer microarray data. PLoS One. 2014;9(7):1–8.
11. Spirtes P, Richardson T. Directed Cyclic Graphical Representations of Feedback Models. Proc Elev Conf Uncertain Artif Intell. 1995;1–37.
12. Neapolitan RE. Learning Bayesian networks. first. Chicago: Pearson Prentice Hall; 2004. 291–425 p.
13. Scutari M. Learning Bayesian Networks with the bnlearn R Package. J Stat Softw. 2010;35(3):1–22.
14. Hamm A, Prenen H, Van Delm W, Di Matteo M, Wenes M, Delamarre E, et al. Tumour-educated circulating monocytes are powerful candidate biomarkers for diagnosis and disease follow-up of colorectal cancer. Gut. 2016;65(6):990–1000.
15. Nagarajan R, Scutari M, Lèbre S. Bayesian Networks in R [Internet]. New York, NY: Springer New York; 2013 [cited 2018 Apr 17]. Available from: http://link.springer.com/10.1007/978-1-4614-6446-4
16. Tibshirani R. The lasso method for variable selection in the cox model. Stat Med. 1997;16(4):385–95.
17. Bühlmann P, Geer S van de. Statistics for high-dimensional data: Methods, Theory and Applications [Internet]. Springer Series in Statistics. 2011. 7–34 p. Available from: http://www.springer.com/statistics/statistical+theory+and+methods/book/978-3-642-20191-2