Chapter 2
Computational Tools for Comparing Gene Coexpression Networks

Vinícius Carvalho Jardim, Camila Castro Moreno, and André Fujita

Abstract The comparison of biological networks is a crucial step to better understanding the underlying mechanisms involved in specific experimental conditions, such as those of health and disease or high and low concentrations of an environmental element. To this end, several tools have been developed to compare whether network structures are "equal" (in some sense) across conditions. Some examples of computational methods include DCGL, EBcoexpress, DiffCorr, CoDiNA, DiffCoEx, coXpress, DINGO, DECODE, dCoxS, GSCA, GSNCA, CoGA, GANOVA, and BioNetStat. We will briefly describe these algorithms and their advantages and disadvantages.

Keywords Network science · Differential network analysis · Coexpression network · Systems biology · Network theory

2.1 Introduction

To understand complex systems, we need to consider the interactions between their elements. A graph is a useful tool for studying these systems due to the plasticity of network models for interpreting biological problems. In a biological context, network vertices can represent system elements such as proteins, metabolites, genes, among other examples. In coexpression networks, vertices represent genes, while edges represent coexpression between gene pairs.

V. C. Jardim · C. C. Moreno
Interdepartmental Bioinformatics Program, Institute of Mathematics and Statistics - University of São Paulo, São Paulo, Brazil
e-mail: vinicius.jardim.carvalho@usp.br

C. C. Moreno
e-mail: camila.moreno@usp.br

A. Fujita (✉)
Department of Computer Science, Institute of Mathematics and Statistics - University of São Paulo, São Paulo, Brazil
e-mail: andrefujita@usp.br

F. A. B. da Silva et al. (eds.), *Networks in Systems Biology*, Computational Biology 32,
https://doi.org/10.1007/978-3-030-51862-2_2

We define coexpression as the statistical dependence (correlation) between the expression values of two genes. Correlation measures how coordinated the variation of expression values of two genes are in same condition samples (obtained from microarray or RNA-seq analysis). In this chapter, we use the terms *Conditions* or *experimental conditions* as synonyms of experimental treatments, such as of high, mean, and low temperatures or clinical status, such as healthy versus cancer tissues. Usually, correlation allows us to infer whether two genes belong to the same metabolic pathway or biological process. However, it does not imply that one variable influences another. Therefore, the edges that represent the correlations have no direction, constructing undirected networks.

Changes in correlations (edges) between conditions are of interest to many studies. In some cases, the aim is to verify whether the environment or genome variations affect the relationship between genes. Considering that each network represents an experimental condition, to achieve this goal, we need effective means to compare these networks. The scientific community has developed several strategies to accomplish this task, with approaches ranging from verifying the edge's existence in differing conditions to network model comparisons.

The most used correlation measure in coexpression network studies is the Pearson correlation. However, the non-parametric Spearman correlation is also frequently used since it does not demand the assumption of normality and is not limited to only detecting linear correlations. Other strategies use mutual entropy and Bayesian inference to define coexpression between genes [1].

Beyond the choice of correlation methods, it is also vital to select the threshold for a given correlation to become an edge. In this sense, we commonly use two main kinds of techniques. The most used is the hard threshold. It works as a cut-off value to remove correlations that are below a defined value (correlation threshold) or with a predetermined level of significance (p-value threshold). Another strategy is the soft threshold proposed in WCGA paper [2]. The soft threshold ponders (or rescales) the correlation values according to a power value β. This threshold technique works by powering the correlation to a β value: the higher values increase and the lower ones decrease, therefore highlighting the most relevant correlations. At the soft threshold, the network remains complete without edge removal. Once parameters for constructing networks are defined, those such as coexpression criteria and threshold technique, we can compare the resulting networks in many ways.

2.2 Network Comparison Methods

Many studies apply network analysis to compare different experimental conditions. One way is to quantify and compare the structural features of networks such as presence or absence of edges or the number of connections of a vertex [3, 4]. Other strategies look for edges that are exclusive of a condition [5] or identify a differential network resulting from the combination of differential expression analysis (DE) and differential coexpression (DC) [6]. Despite these methods being useful, they do not take into account the intrinsic fluctuations of data to compare the networks, leading to erroneous conclusions about differences between experimental conditions.

Some statistical techniques give reliability to significant differences in network comparison [7]. For instance, data permutation and resampling techniques allow the association of a p-value (probability of significance) to each pair of genes [7] or to compare entire networks [8]. Many tools approach the need to compare networks between different conditions. Here, we present and discuss some of these differential network analysis tools. All the tools presented in this chapter are summarized in Fig. 2.1.

2.2.1 *Edge Comparison*

Some tools test whether edges (or a group of them) are statistically different between two or more conditions. They usually return, as a result, a list of differentially coexpressed edges. Sometimes, these methods also return a list of enriched vertices belonging to these edges.

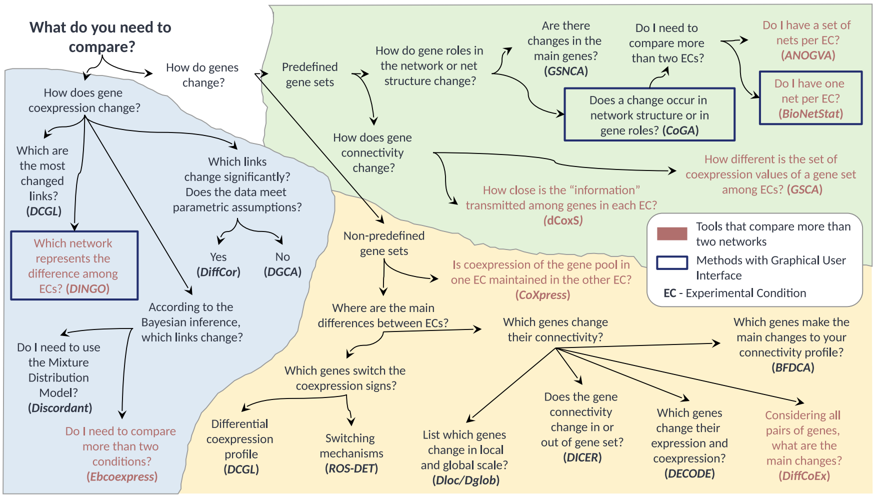
What do you need to compare?
How does gene coexpression change?
How do genes change?
Which are the most changed links? (DCGL)
Which network represents the difference among ECs? (DINGO)
Which links change significantly? Does the data meet parametric assumptions?
Yes (DiffCor)
No (DGCA)
According to the Bayesian inference, which links change?
Do I need to use the Mixture Distribution Model? (Discordant)
Do I need to compare more than two conditions? (Ebcoexpress)
Predefined gene sets
How do gene roles in the network or net structure change?
Are there changes in the main genes? (GSNCA)
Do I need to compare more than two ECs?
Do I have a set of nets per EC? (ANOGVA)
Do I have one net per EC? (BioNetStat)
Does a change occur in network structure or in gene roles? (CoGA)
How does gene connectivity change?
How different is the set of coexpression values of a gene set among ECs? (GSCA)
How close is the "information" transmitted among genes in each EC? (dCoxS)
Non-predefined gene sets
Is coexpression of the gene pool in one EC maintained in the other EC? (CoXpress)
🟥 Tools that compare more than two networks
□ Methods with Graphical User Interface
EC - Experimental Condition
Where are the main differences between ECs?
Which genes change their connectivity?
Which genes make the main changes to your connectivity profile? (BFDCA)
Which genes switch the coexpression signs?
Differential coexpression profile (DCGL)
Switching mechanisms (ROS-DET)
List which genes change in local and global scale? (Dloc/Dglob)
Does the gene connectivity change in or out of gene set? (DICER)
Which genes change their expression and coexpression? (DECODE)
Considering all pairs of genes, what are the main changes? (DiffCoEx)

Fig. 2.1 Decision tree of differential coexpression analysis tools. Each tool answers a specific question about the data set. Then, the user has to follow the questions to choose the ideal method. The methods that have a Graphical User Interface (GUI) to perform the analyses are outlined with a dark blue square. The red color indicates that the method compares more than two networks. The blue background color indicates the edge comparison tools; the yellow background indicates the untargeted vertex comparison tools and the green background indicates the targeted vertex comparison tools

2.2.1.1 Diffcorr

To compare two biological conditions, Fukushima (2013) checks whether each edge occurs under both conditions [9]. **Diffcorr** applies a direct and straightforward strategy to define differentially coexpressed edges based on correlations transformed by the Fisher Z-scores method [10]. An advantage of Diffcorr is that it is possible to test each correlation, allowing the user to examine in detail the changes between conditions. One disadvantage is the high number of tests performed, incurring the problem of multiple tests. This method is implemented in the R language and is available in the SourceForge platform (https://sourceforge.net/projects/diffcorr/).

2.2.1.2 DCGL

Liu et al. [11] proposed a method to verify which edges are distinct between two conditions after checking the vertices associated with them. **DCGL** performs the Differential Coexpression Enrichment (DCe) analysis, which applies the limit fold change (LFC) model over each pair of edges in both conditions. This method returns a list of differential coexpression links (DCLs). Moreover, based on DCLs and a binomial model, DCGL selects a set of differentially coexpressed genes (DCGs). According to Yu et al. [12], DCGL considers two important issues to compare networks: the gene neighbor information and the quantitative coexpression change information. However, the comparisons are limited to two networks only. We present DCGL methods based on vertices in Sect. 2.2.2.2. This tool is implemented in R code and is available in cran DCGL (https://cran.r-project.org/package=DCGL).

2.2.1.3 Ebcoexpress

Based on Bayesian statistics, **Ebcoexpress** [13] infers whether edges are significantly changed. This tool uses empirical Bayesian inference and a nested expectation-maximization (NEM) algorithm to estimate the posterior probability of differential correlation between gene pairs. The advantage of EBcoexpress is that it compares more than two conditions and to provide a false discovery rate (FDR) controlled list of significant DC gene pairs minimizing the loss of power. Regarding the algorithm's run-time, there is a restriction on the number of genes that can be analyzed. The authors recommend 10,000 genes as a limit. Besides, it must be verified if genes have high correlations among them and remove these highly correlated genes pairs to avoid false-positive detections. EBcoexpress is implemented in R code and is available in the Bioconductor repository (https://bioconductor.org/packages/EBcoexpress/).

2.2.1.4 Discordant

Similar to Ebcoexpress (Sect. 2.2.1.3), the **Discordant** tool [14] also uses empirical Bayesian inference and the expectation-maximization technique to estimate the posterior probability and identify differential coexpression of gene pairs. According to Siska et al. [14], Discordant fits a mixture distribution model based on Z-scores of correlations. This technique allows Discordant to detect more types of differential coexpression scenarios than EBcoexpress. It also outperforms the Ebcoexpress method in computational time and accuracy [15]. To reduce the computational time, it assumes that the expression levels of gene pairs are independent and bivariate distributed. However, this assumption is not biologically probable. Discordant is implemented in R and is available in the Bioconductor repository (http://bioconductor.org/packages/discordant/).

2.2.1.5 DGCA

DGCA [16] identifies sets of genes as differentially correlated. It classifies differential correlation into nine possible scenarios. As does Diffcorr, DGCA also applies the transformed correlations by the Fisher Z-scores method [10]. However, DGCA differs from the existing differential correlation approaches since it calculates the FDR of differential correlation using nonparametric sample permutation and calculates the average difference in correlation between one gene and a gene set across two conditions. The permutation tests also minimize parametric assumptions. One disadvantage of the DGCA methodology is that it can compare only two experimental conditions at a time. DGCA is implemented in R code and is available in the CRAN repository (https://cran.r-project.org/package=DGCA).

2.2.1.6 DINGO

The last edge comparison tool detects differentially coexpressed edges by the use of a Gaussian Graphical Model (GGM). **DINGO** [17] calculates a global component (graph), composed of common edges between two conditions. Based on this global component, the algorithm determines specific local components for each condition. It attributes a score to each edge, determining how altered they are between conditions. Then, DINGO selects the edges that have significantly altered scores. Finally, the tool returns a single network that has significantly altered edges between the studied conditions. DINGO estimates conditional dependencies for each group. The newer version of DINGO, iDINGO, is an R package and is available in the CRAN repository (https://cran.r-project.org/web/packages/iDINGO/).

2.2.2 Untargeted Vertex Comparison

Instead of comparing edges among conditions, other tools look for subsets of genes that are differentially coexpressed between different conditions. It is possible to classify vertex comparison methods into two subgroups according to the applied strategy: untargeted and targeted (adapted from [18]). Untargeted approaches search for non-predefined gene sets. This strategy is based on grouping genes into modules according to their coexpression status under the compared conditions. We present targeted approaches in Sect. 2.2.3.

2.2.2.1 coXpress

The first untargeted method presented here is **coXpress** [19]. This methodology detects a gene set that is highly correlated in one condition and tests whether the other condition maintains the same genes in the strongly connected group. Based on hierarchical clustering, it groups the vertices in one condition and calculates a t-statistic for this group. A gene set is differentially coexpressed between two conditions when t is statistically significant in one condition, but not in the other. It also detects which gene pairs changed their correlation among networks and can compare more than two experimental conditions. The t-statistic allows coXpress to state whether the formation of a group is a random process. However, it considers that each gene belongs to only one group, as opposed to what actually occurs in biological systems, where genes generally participate in more than one process. The R package for this method is available at the coXpress website (http://coxpress.sourceforge.net/).

2.2.2.2 DCGL

The method proposed by Liu et al. [11] identifies differentially coexpressed genes using the differential coexpression profile (DCp) method [12]. Unlike the DCe method presented in Sect. 2.2.1.2, DCp is based on the coexpression profile of each vertex with all other vertices. It measures whether the average coexpression of a gene with its neighbors changes between conditions. Besides DCp and DCe, the authors implemented three other methods of gene connectivity measures to perform differential coexpression analysis: log-ratio of connections (LRC), average specific connection (ASC), and weighted gene coexpression network analysis (WGCNA). According to the authors, DCp and DCe detect if the coexpression of genes pairs changes from a positive to a negative sign, while other methods are focused on gene connectivity. All five comparison algorithms mentioned are limited to comparing two networks. This method is implemented as an R package available in the CRAN repository (http://cran.r-project.org/web/packages/DCGL/index.html).

2.2.2.3 DiffCoEx

Tesson et al. [18] proposed a method based on the dissimilarity matrix between two correlation matrices of each experimental condition. **DiffCoEx** performs the Topological Overlap Method (TOM) [2] on the dissimilarity matrix resulting in a list of altered genes. Then, it enriches this list according to biological pathways. Furthermore, DiffCoEx does not need to detect a coexpressed module in one condition to verify if this module is coherent in another; instead, it determines the differential coexpression based only on the dissimilarity matrix. Aside from this, DiffCoEx also has a similar algorithm (based on the dissimilarity matrix) that allows comparing more than two networks. This method is implemented in R and is available on the DiffCoEx website (https://rdrr.io/github/ddeweerd/MODifieRDev/man/diffcoex.html).

2.2.2.4 DICER

Amar et al. [20] state that the main differences among biological conditions occur more frequently between modules of genes than within them. Thus, **DICER** [20] classifies a set of genes as differentially coexpressed (DC) if its set of altered correlations fits in at least one of the following two scenarios: the DC cluster and Meta-module. The DC cluster is a gene set in which genes correlations are statistically different between experimental conditions. Meta-modules are the pairs of gene sets that are highly correlated within the gene sets and have high dissimilarity between them comparing two experimental conditions. The differentiation of these two scenarios allows the user to know which kinds of differences (DC cluster or Meta-module) the system has between conditions. DICER is implemented in Java code and is freely available for download at the DICER website (http://acgt.cs.tau.ac.il/dicer/).

2.2.2.5 DCloc and DCglob

Bockymayr et al. [21] developed two untargeted algorithms, **DCloc** and **DCglob**, that identify differential correlation patterns by comparing the local or global structure of correlation networks. The construction of networks from correlation structures requires fixing a correlation threshold. Instead of a single cutoff, the algorithms systematically investigate a series of correlation thresholds and permit the detection of different kinds of correlation changes at the same level of significance: great changes of a few genes and moderate changes of many genes. Using random subsampling and cross-validation methods, DCloc and DCglob identify accurate lists of differentially correlated genes. The codes to run each function are in R code and are available in additional files of the article (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3848818/).

2.2.2.6 BFDCA

BFDCA (Bayes Factor Approach for Differential Coexpression Analysis) [22] aims to detect gene sets that possess different distributions of gene coexpression profiles between two different conditions. It first estimates the differential coexpression of gene pairs based on Bayes factors. Then, it infers DC modules with higher Bayes factor edges and selects significant DC gene pairs based on the vertex and edge importance. BFDCA provides a relatively small number of gene pairs, which can lead to a high-accuracy classifier [22]. One limitation of BFDCA is the small sample problem: the method needs enough samples to estimate the hyperparameters. BFDCA is implemented in a comprehensive R package freely available for download at this website (http://dx.doi.org/10.17632/jdz4vtvnm3.1).

2.2.2.7 ROS-DET

Kayano et al. [23] consider that other approaches have problems under three real cases regarding experimental data: 1) when there are outliers, 2) when there are expression values with a tiny range and 3) when there is a small number of samples. The authors proposed the **ROS-DET** (RObust Switching mechanisms DETector), a detector of switching mechanisms. This switch is the alteration of the correlation signal between two conditions. The ROS-DET overcomes these three problems while keeping the computational complexity of current approaches. ROS-DET is implemented in shell script and is available on this website (https://www.bic.kyoto-u.ac.jp/pathway/kayano/ros-det.htm).

2.2.2.8 DECODE

Lui et al. [24] proposed to combine differential expression (DE) and coexpression (DC) analysis. **DECODE** identifies characteristics not detected through DC or DE approaches alone. This tool combines both strategies and performs a Z-test to select significant differences in coexpression between two conditions [24]. The main advantage of DECODE is that it combines both strategies, which allows it to detect differential coexpression scenarios not detected by other methods. DECODE is implemented in R code and is available in the CRAN repository (http://cran.r-project.org/web/packages/decode/index.html.).

2.2.3 Targeted Vertex Comparison

Following the classification cited in Sect. 2.2.2 (adapted from [18]), targeted approaches compare predefined genes modules according to previous knowledge about the studied system.

2.2.3.1 dCoxS

dCoxS is a targeted strategy because it compares and tests whether predefined gene sets are differentially coexpressed between experimental conditions [25]. It verifies if the Interaction Score (IS) between two gene groups changes among conditions. dCoxS calculates the relative entropy among genes to build the coexpression network. Considering that adjacency matrices represent the networks, the distance between them is measured by the correlation method. dCoxS is unique in that it applies entropy as coexpression measure and correlation as a distance measure between two adjacency matrices. It is implemented in R and is available on this website (http://www.snubi.org/publication/dCoxS/index.html.).

2.2.3.2 GSCA

Choi et al. [26] proposed a method to perform a network comparison based on the distance between adjacency matrices. **GSCA** constructs the adjacency matrices by calculating the correlation measure and compares them using Euclidean distances. If the distance is statistically significant, tested by the permutation samples technique, the two networks are classified as differentially coexpressed. Besides this, the GSCA method has a generalization for more than two conditions: for this, GSCA calculates the average of pairwise distances between correlation matrices. As GSCA does not correct the comparisons for multiple tests, comparing many experimental conditions could lead to false-positive results. The GSCA package is implemented in R and is available on the GSCA website (https://www.biostat.wisc.edu/ kendzior/GSCA/).

2.2.3.3 GSNCA

The comparison of network structures is a strategy employed by this and the following three tools. Rahmatallah et al. [27] state that the significant changes in a system occur at the most critical vertices of the network. Based on this statement, the **GSNCA** tool tests whether there are differences between the vertex's weight vectors given by the eigenvector centrality. Centrality is the weight of a vertex according to its position in a network. The eigenvector centrality determines a vertex centrality by the centralities of its neighbors pondered by the strength of the connections [28]. In other words, the method tests whether the most critical vertices of the network (higher eigenvector centralities) change between conditions. However, this method only compares two experimental conditions. The GSNCA implementation in R is available upon request from the authors.

2.2.3.4 CoGA

Santos et al. (2015) [29] proposed a statistical method to compare the graph spectrum of correlation networks. The spectrum of a graph is the probability distribution of eigenvalues of the adjacency matrix, which represents the network, also called spectral distribution. Based on this measure, **CoGA** (Coexpression Graph Analyzer) tests the equality between the spectral distributions of two networks. CoGA also compares two networks by other structural measures, such as spectral entropy (the entropy of spectral distribution), centralities, clustering coefficient, and distribution of vertex degrees. However, as GSNCA, this method is also restricted to the comparison of only two conditions. CoGA was implemented in R code. It is available on the CoGA website (https://www.ime.usp.br/ suzana/coga/) and can be used with graphical interface features to perform the analysis easily.

2.2.3.5 ANOGVA

To solve the problem of being limited to comparing only two experimental conditions, Fujita et al. [30] generalized CoGA statistics (Sect. 2.2.3.4). The **ANOGVA** (ANalysis Of Graph VAriability) compares graph populations through spectral distributions using the Kullback-Leibler divergence, much like an ANOVA method for graphs. This tool is useful for comparing two or more sets of networks, such as functional brain networks, where each sample has one network and, consequently, each experimental condition has many networks. However, this method does not compare experimental conditions that have only one coexpression network each. ANOGVA is implemented in R code and is available in the package statGraph (http://www.ime.usp.br/ fujita/software.html).

2.2.3.6 BioNetStat

Finally, **BioNetStat** [31] generalizes the graph spectrum comparison performed by CoGA (Sect. 2.2.3.4), without the necessity of graph populations, as ANOGVA. Also, it compares networks by spectral entropy, vertex centralities, clustering coefficient, and degree distribution. BioNetStat also performs statistical tests for each vertex (centralities and clustering coefficient), highlighting which vertices differ among networks. As we have other methods that compare the spectral distribution of networks (CoGA Sect. 2.2.3.4 and ANOGVA Sect. 2.2.3.5), BioNetStat has a restriction for the number of genes. A high number of genes—over 5,000 —slows the algorithm since it has to find the eigenvalues of the adjacency matrix, which is time-consuming for larger data sets. BioNetStat is an R package and is available in the Bioconductor repository (https://bioconductor.org/packages/BioNetStat/). It is also possible to perform this analysis behind a Graphical User Interface.

2.3 Conclusion

As was shown, there are many methods based on a wide range of strategies to compare coexpression networks. Beyond just coexpression, it is also possible to determine the correlation between metabolites and protein concentrations, applying all methodologies mentioned above to metabolites or protein networks. Unfortunately, most of these methods are only readily usable for those who have some prior knowledge in programming, specifically in the R language. For this reason, researchers and developers should provide a graphical user interface for their methods, both improving the reach of these tools and increasing the number of data analysis tools available for nonprogramming scientists.

Acknowledgements This work was partially supported by FAPESP (2018/21934-5, 2019/03615-2), CAPES (Finance Code 001), CNPq (303855/2019-3), Alexander von Humboldt Foundation, Newton Fund, and The Academy of Medical Sciences.

References

1. Santos SDS, Takahashi DY, Nakata A, Fujita A (2013) Brief Bioinform **15**(6):906. https://doi.org/10.1093/bib/bbt051
2. Langfelder P, Horvath S (2008) BMC Bioinform 9:559. https://doi.org/10.1186/1471-2105-9-559
3. Caldana C, Degenkolbe T, Cuadros-Inostroza A, Klie S, Sulpice R, Leisse A, Steinhauser D, Fernie AR, Willmitzer L, Hannah MA (2011) Plant J 67(5):869. https://doi.org/10.1111/j.1365-313X.2011.04640.x
4. Weston DJ, Karve AA, Gunter LE, Jawdy SS, Yang X, Allen SM, Wullschleger SD (2011) Plant, Cell Environ **34**(9):1488. https://doi.org/10.1111/j.1365-3040.2011.02347.x
5. Zhang H, Yin T (2016) Tree Genet Genomes 12(3):61. https://doi.org/10.1007/s11295-016-1016-9
6. Sun SY, Liu ZP, Zeng T, Wang Y, Chen L (2013) Scientific reports 3:2268. https://doi.org/10.1038/srep02268, http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3721080&tool=pmcentrez&rendertype=abstract
7. Walley AJ, Jacobson P, Falchi M, Bottolo L, Andersson JC, Petretto E (2012) Int J 36(1):137. https://doi.org/10.1038/ijo.2011.22.Differential
8. Hochberg U, Degu A, Toubiana D, Gendler T, Nikoloski Z, Rachmilevitch S, Fait A (2013) BMC Plant Biol 13(1):184. https://doi.org/10.1186/1471-2229-13-184, http://dx.plos.org/10.1371/journal.pone.0115581://www.biomedcentral.com/1471-2229/13/184
9. Fukushima A (2013) Gene 518(1):209. https://doi.org/10.1016/j.gene.2012.11.028
10. Fisher RA (1915) Biometrika 10(4):507. www.jstor.org/stable/2331838
11. Liu BH, Yu H, Tu K, Li C, Li YX, Li YY (2010) Bioinformatics 26(20):2637. https://doi.org/10.1093/bioinformatics/btq471
12. Yu H, Liu BH, Ye ZQ, Li C, Li YX, Li YY (2011) BMC Bioinform 12(1):315. https://doi.org/10.1186/1471-2105-12-315, http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3199761&tool=pmcentrez&rendertype=abstract
13. Dawson JA, Ye S, Kendziorski C (2012) Bioinformatics 28(14):1939. https://doi.org/10.1093/bioinformatics/bts268
14. Siska C, Bowler R, Kechris K (2015) Bioinformatics 32(5):690. https://doi.org/10.1093/bioinformatics/btv633

15. Singh AJ, Ramsey SA, Filtz TM, Kioussi C (2018) Cell Mol Life Sci 75(6):1013. https://doi.org/10.1007/s00018-017-2679-6
16. McKenzie AT, Katsyv I, Song WM, Wang M, Zhang B (2016) BMC Syst Biol 10(1):106. https://doi.org/10.1186/s12918-016-0349-1, http://www.ncbi.nlm.nih.gov/pubmed/27846853%5Cnwww.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5111277
17. Ha MJ, Baladandayuthapani V, Do KA (2015) Bioinformatics 31(21):3413. https://doi.org/10.1093/bioinformatics/btv406
18. Tesson BM, Breitling R, Jansen RC (2010) BMC Bioinform 11(1):497. https://doi.org/10.1186/1471-2105-11-497, http://bmcbioinformatics.biomedcentral.com/articles/
19. Watson M (2006) BMC Bioinform 7:509. https://doi.org/10.1186/1471-2105-7-509, http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&id=17116249&retmode=ref&cmd=prlinks%5Cnpapers3://publication/
20. Amar D, Safer H, Shamir R (2013) PLoS Comput Biol 9:3. https://doi.org/10.1371/journal.pcbi.1002955
21. Bockmayr M, Klauschen F, Györffy B, Denkert C, Budczies J (2013) BMC Syst Biol 7(1):78. https://doi.org/10.1186/1752-0509-7-78, http://www.ncbi.nlm.nih.gov/pubmed/23945349%5Cnwww.biomedcentral.com/1752-0509/7/78
22. Wang D, Gu J (2016) Quant Biol 4(1):58. https://doi.org/10.1007/s40484-016-0063-4
23. Kayano M, Takigawa I, Shiga M, Tsuda K, Mamitsuka H (2011) Nucl Acids Res 39(11):1. https://doi.org/10.1093/nar/gkr130
24. Lui TWH, Tsui NBY, Chan LWC, Wong CSC, Siu PMF, Yung BYM (2015) BMC Bioinform 16(1):182. https://doi.org/10.1186/s12859-015-0582-4, http://bmcbioinformatics.biomedcentral.com/articles/
25. Cho SB, Kim J, Kim JH (2009) BMC Bioinform 10(1):109. https://doi.org/10.1186/1471-2105-10-109
26. Choi Y, Kendziorski C (2009) Bioinformatics. https://doi.org/10.1093/bioinformatics/btp502
27. Rahmatallah Y, Emmert-Streib F, Glazko G (2014) Bioinformatics 30(3):360. https://doi.org/10.1093/bioinformatics/btt687
28. Bonacich P (1972) Sociol Methodol 4:176. https://www.jstor.org/stable/270732
29. Santos SDS, De Almeida Galatro TF, Watanabe RA, Oba-Shinjo SM, Marie SKN, Fujita A (2015) PLoS ONE. https://doi.org/10.1371/journal.pone.0135831
30. Fujita A, Vidal MC, Takahashi DY (2017) Front Neurosci 11. https://doi.org/10.3389/fnins.2017.00066, http://journal.frontiersin.org/article/10.3389/fnins.2017.00066/full
31. Jardim VC, Santos SSS, Fujita A, Buckeridge MS (2019) Front Genet 10(1). https://doi.org/10.3389/fgene.2019.00594