Technological Educational Institute of Crete

School of Engineering

Department of Informatics Engineering

Gene expression and gene regulatory network analysis with statistical methods and machine learning algorithms

Student

Fotini Droumalia

Supervisor

Lefteris Koumakis

Heraklion Month, 2022

Abstract

Determining the best approach and data type for Pathway Analysis is a significant difficulty for the field of diagnostic medicine. The findings of recent studies indicate their preference for Machine Learning algorithms and the utilization of continuous gene expression values rather than binary values. However, due to the limitations of Machine Learning and non-binary values, there were efforts to produce new hybrid techniques that exploit the benefits of statistical methods and discrete values. The purpose of this study was to identify the most effective approach for Pathway Analysis utilizing currently available tools and to compare the results to previous research. This was accomplished by implementing the scoring methodology for several Pathway Analysis tools and employing a Decision Tree algorithm to assess the outcomes. The tools selected for implementation were TAPPA, PRS, TEAK, DEAP, GraphiteWeb, MinePath and HiPathia, among which HiPathia achieved the highest rate of accuracy and the shortest execution time; overall, Machine Learning-based techniques outperformed those based on statistics. The outcomes obtained are consistent with prior literature, which have shown that non-binary data hold more information and that Machine Learning methods offer new opportunities for use in health. Unfortunately, a major issue that prevented our research from extracting significant sub-pathways is probably related to the fact that there are numerous genomic platforms available; as a result genes cannot be recognized across different datasets.

Table of Contents

List of Tables

List of Figures

List of Images

Introduction

Bioinformatics is a rapidly developing science, which integrates biology, informatics, and mathematics to manage huge genomic data and comprehend complex biological mechanisms. The Human Genome Project, which mapped the human genome and announced the full sequence in 2022 after almost 30 years of study, was one of the field's greatest achievements [1]. Its development allows researchers to better understand how viruses and diseases function, create effective therapeutics, find cancer-related mutations, and make more precise predictions [2].

Utilizing the genomic sequence demands methods and tools that can manage this enormous amount of data. In order to obtain a complete and more accurate picture of biological activities, the main strategy adopted nowadays is Pathway Analysis, which exploits both gene expression values and pathway network topology. Even though it's a promising technique, there are challenges and constraints that must be resolved. More particularly, the microarray data's noise makes the procedure more complicated, requiring its conversion to binary form, but doing so can result in the loss of key data and, consequently, inaccurate results [3]. On the other hand, analysis methods that use them produce fast, real-time results that are non-time and CPU consuming. The preference for discrete expression data over continuous values is supported by various technologies, including MinePath [4].

In order to improve the existing technologies and develop new pathway analysis methods, this study makes an attempt to assess and compare the existing pathway analysis tools with the aim of verifying that non-binary gene expression data provide better results and identifying any non-binary pathway analysis algorithms that may be applied to real-time predictions.

Molecular biology

The area of molecular biology is concerned with the molecular foundations of biological activity [5]. The study of biological molecules and their interactions is only one aspect of molecular biology; it also encompasses a variety of methods designed since the field's creation that have helped researchers understand how molecules function [6]. Even though every living thing has a variety of molecules, most molecular biologists concentrate on genes and proteins. Within living cells, proteins carry out a large variety of tasks, and genes store the data needed to produce additional proteins [5].

Deoxyribonucleic acid, also known as DNA, is the molecule that contains the genetic material necessary for an organism's formation and functionality [7] and is structured as a double helix, which is made up of two connected strings that spiral around one another to form a twisted ladder. The procedure for producing functional cells in accordance with DNA instructions is known as the central dogma of molecular biology.

DNA genotyping has made it possible to diagnose and potentially treat many disorders that are driven by genetic mutations, microbes, and viruses. The analysis of disease pathways, the evaluation of a person's genetic predisposition to particular diseases, the diagnosis of genetic abnormalities, the development of new medications, and the recognition of pathogens have all been significantly improved by an understanding of the structure and function of DNA [8].

Gene and gene expression

A gene is the fundamental structural and operative unit of inheritance and is composed of DNA [9]. Because of its significance in the production of proteins, which are essential for the formation, operation, and regulation of the body's tissues and organs [10], many studies are being conducted to decipher the human genome’s sequence and define the genes that it includes. The most well-known of these is the Human Genome Project, which was successful in sequencing the whole genome in 2022.

Protein synthesis is regulated by genes to produce the desired results [11] and this is achieved by a process known as gene expression. Gene expression directs when and where RNA molecules and proteins are produced [12]. When the protein that a gene encodes is made by a cell, it is said to be "expressed". Genes can be classified as active or inactive, but to fully utilize the information that is currently accessible, new methods emerge that take into consideration the level of expression.

Gene Regulatory Networks

Gene regulatory networks, or GRNs, are collections of genes that interact with each other. Graphs with nodes and edges are commonly used to describe gene regulatory networks. In a directed graph, every gene is denoted by a node and a pathway's edge represents a connection or other type of interaction between the nodes [13]. Early attempts perceive the networks as a collection of genes without considering their interactions with other genes [4], but doing so ignores information that might be crucial.

These networks can be represented by mathematical and computational models that explain the logic behind the regulatory activities among genes when a certain cell program is activated [14]. GRNs can also be used to predict changes in gene expression under various conditions, comprehend the information flow in biological systems, find circuits that can be employed for a particular operation, and understand the exchange of information in biological systems [15].

Pathway Analysis

Pathway Analysis (PA), also called functional enrichment analysis, is becoming more and more important in Omics research, which concerns biological branches that end with the suffix –omics [16]. Pathway Analysis methodologies combine knowledge from gene expression analysis and molecular pathway networks to discover strongly impacted pathways in a particular condition and better understand the biological significance of differentially expressed genes and proteins.

The term Pathway refers to the graphical representation of molecular interaction, reaction, and relationship networks. The graph consists of nodes, which correspond to genes, proteins, and/or molecules, and directed edges, which represent relationships and interactions between the nodes. The states of each gene are either on or off, indicating whether the gene is expressed or not expressed respectively. The types of interactions between the nodes vary. Activation, inhibition, and catalysis are some examples of different sorts of interactions between nodes. To determine a score for each pathway, the experimental data and the graph are analyzed. The score measures how much a pathway diverged between the two phenotypes.

Pathway analysis techniques discover the pathways that are strongly impacted in a specific circumstance by combining available pathway databases with gene expression data [17]. KEGG, Reactome, and BioCarta are some of the sources that include thorough information about pathways. KEGG is a repository of hand-drawn pathway maps that provide information about genomes, biological pathways, diseases, pharmaceuticals, and chemical compounds [18], [19], while BioCarta's interactive online services give the means to see how genes communicate in dynamic graphical models [20]. Reactome database’s mission is to provide user-friendly bioinformatics tools for visualizing, interpreting, and analyzing pathway data in support of fundamental and medical trials, genomic analysis, modeling, systems biology, and education [21].

We can distinguish two main approaches to pathway analysis: the first one takes into consideration only the expression levels of the genes of a pathway, while the second one takes advantage of the pathway topology as well, known as topology-based [22]. Since topology-based approaches use significantly more data than non-topology-based methods do, it makes sense that they should be more efficient and reliable. However, many studies come to the conclusion that both strategies are flawed [23].

Pathway Analysis Tools

TAPPA

TAPPA is a java-based tool that was introduced in [24] and uses pathway topological measurements to identify phenotype-associated genomic circuits. This is accomplished by calculating a Pathway Connectivity Index (PCI) for each pathway and then assessing its relationship to phenotypic variance.

$$PCI = \sum_{i=1}^{N} \sum_{j=1}^{N} sgn\left(x_{is} + x_{js}\right) * |x_{is}|^{0.5} * \alpha_{ij} * \left|x_{js}\right|^{0.5}$$

In the formula above, x_{is} stands for the standardized log expression measurement for gene i and sample s. N represents the total of genes, and $sgn\left(x_{is} + x_{js}\right)$ denotes the overall expression status (up- or down-regulation) of the gene pair. Given that a pathway can be visualized as a graph $\mathrm{G(V, E)}$, the adjacency matrix is defined as $A = (\alpha_{ij})$, such that $\alpha_{ij} = 1$ if $\mathrm{i} = \mathrm{j}$ or $(g_i, g_j) \in E$, otherwise $\alpha_{ij} = 0$.

Subsequently, each PCI value is divided by the number of genes in the respective pathway in order to normalize the PCIs.

The TAPPA tool was developed using JAVA and can handle both binary and numerical attributes. In the case of the binary traits, the Mann–Whitney test is used to assess the significance of the relationship between network PCI and phenotype, while the Spearman correlation is used for continuous attributes. In addition, a permutation test is used to assess the false discovery rate (FDR).

Subsequently, using different zoom ratios, the pathway is visualized and the genes that are strongly related to the phenotype are identified. Eventually, the relationship between the phenotypes of the sub-modules in a pathway is studied and determines the biological significance of the genes involved.

SPIA

Signaling Pathway Impact Analysis, also known as SPIA, uses data collected from the classical enrichment analysis and combines them with data that evaluate the perturbation of a certain pathway under a specific circumstance. The following methodology is described in [25].

Impact Analysis considers the overrepresentation of DE genes in a particular pathway and the anomalous alteration of that pathway, as determined by propagating observed expression changes across the pathway topology. These features correspond to two separate probabilities, P_{NDE} and P_{PERT}.

The $P_{NDE} = P(X \geq N_{de}|H_0)$ probability represents the importance of a pathway P_i, based on an overrepresentation study of the number of DE genes (N_{DE}) found on the pathway.

P_{PERT} probability results from the degree of perturbation in each pathway, which is calculated using the following gene perturbation function:

$$PF(g_i) = \Delta E(g_i) + \sum_{j=1}^{n} \beta_{ij} \cdot \frac{PF(g_j)}{N_{ds}(g_j)}$$

The above formula takes into consideration the type of relationship between two genes and is represented by the term β_{ij}. If the value of β equals to +1, then the type of interaction is activation, while -1 corresponds to inhibition and repression.

The net perturbation accumulation at the gene level, Acc_g, is then calculated as the difference between a gene's perturbation factor PF and its signed adjusted expression change or observed log fold-change, when comparing two circumstances.

$$Acc(g_i) = PF(g_i) - \Delta E(g_i)$$

Subsequently, the total net accumulated perturbation of a pathway is calculated as the sum of the net perturbation accumulation of each gene.

$$t_A = \sum_i Acc(g_i)$$

The possibility of seeing a total accumulated perturbation of the pathway, T_A, greater than t_A, is represented by the P_{PERT} probability:

$$P_{PERT} = P(T_A \geq t_A|H_0)$$

Eventually, P_{NDE} and P_{PERT} are integrated into a global probability value, P_G. Through this probability value the pathways are ranked, and the hypothesis is tested to see if the pathway is significantly disrupted in the study condition.

$$P_G = c_i - c_i \cdot ln\,(c_i)\,,\; c_i = P_{NDE}(i) \cdot P_{PERT}(i)$$

P_G can also set the level of type I error. It is recommended to use the common FDR approach to keep the false discovery rate (FDR) of the pathway analysis at 5%.

TopologyGSA

This method evaluates the differential expression of a pathway using graphical models as demonstrated in [26]. Then it illustrates the components of the pathway that are implicated in the deregulation. Below is a detailed description of the technique.

In this project, KEGG maps are employed, since they provide a good ratio between map accuracy and simplicity. Initially, the paths obtained from the KEGG repository are transformed into a graphical model. This is accomplished by using the following basic steps: i) simple directed edges include inhibition, phosphorylation (+p), and dephosphorylation (-p); ii) BioCarta network provides extensive information that can be used to direct undirected edges and iii) when it comes to complexes (nodes consisted of several gene products), the first principal component is defined as the complex's expression. The data of the same pathway are represented in distinct experimental states as implementations of undirected graphical Gaussian models with the same undirected graph *G*. For instance, in the case of two scenarios, we employ the Gaussian models

$$M_1(G) = \{Y \sim N_p(\mu_1, \Sigma_1), \Sigma_1^{-1} \in S^+(G)\},$$

$$M_2(G) = \{Y \sim N_p(\mu_2, \Sigma_2), \Sigma_2^{-1} \in S^+(G)\}.$$

In this case, p refers to the number of genes (nodes of the graph), while $S^+(G)$ represents the array of symmetric positive definite matrices with null components indicating the missing connections of *G*.

The estimated covariance matrices are calculated by using a technique known as the Iterative Proportional Scaling Technique (IPS) for graph analysis, which ensures that the estimated matrices are positive definite and that their inverse has null elements correlating to the graph's missing edges. The sample covariance matrices can be generated, either from the chip covariance matrices by removing elements referring to all the pairs of genes in the pathway, or by determining the expression levels of the genes on the pathway and measuring the sample covariances.

Then, to compare the gene sets among two experimental conditions, the null hypothesis testing method is employed. The strength of the connections that define a pathway can vary under different situations, causing changes in the pathway's expression. The equivalence of two means is the corresponding hypothesis. The evaluation is determined by whether the models' covariances, which are often unknown, are homogeneous. As a result, the choice of the homogeneity hypothesis has an impact on the analysis of the means.

Eventually, the strength of the gene relationships in two experimental conditions is put to the test to see if they are equal. This is easily accomplished in the context of graphical Gaussian models by comparing the two concentration matrices (opposite of the covariance matrices), which include all the details about the underlying structure. As a result, the focus is on putting the hypothesis $\Sigma_1^{-1} = \Sigma_2^{-1}$ to the test.

The methods for comparing covariance matrices are then applied to the specific instance of graphical Gaussian models in the following methodology. Assume that $\gamma_1 = (\gamma_1^j)$, $j = 1, \dots, n_1$ observations from $N_p(0, \Sigma_1)$, and $\gamma_2 = (\gamma_2^j)$, $j = 1, \dots, n_2$ observations from $N_p(0, \Sigma_2)$, with $\Sigma_1^{-1} = K_1 \in S^+(G)$ and $\Sigma_2^{-1} = K_2 \in S^+(G)$ without losing generality. The hypothesis to be tested is $H_0: K_1 = K_2$ against $H_1: K_1 \neq K_2$. When the value of W_i is determined using the function $W_i = \sum_{j=1}^{n_i} (\gamma_i^j)(\gamma_i^j)^T$, $i = 1,2$, the likelihood function, $L(K_1, K_2)$, yields as follows:

$$L(K_1, K_2) = \prod_{i=1}^{2} (2\pi)^{-\frac{n_i p}{2}} (det\, det\, K_i\,)^{\frac{n_i}{2}} e^{-\frac{1}{2} tr(K_i W_i)}$$

Starting with the pooled covariance matrix $S = (n_1 + n_2 - 2)^{-1} \cdot \{(n_1 - 1) \cdot S_1 + (n_2 - 1) \cdot S_2\}$ and the null hypothesis, the technique calculates the estimate, $\hat{\Sigma}$, of the common covariance matrix. On the contrary, under the alternative hypothesis, the sample covariance matrices, $S_1 = (n_1 - 1)^{-1} \cdot W_1$ and $S_2 = (n_2 - 1)^{-1} \cdot W_2$, are used, so that the values of $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ are calculated.

After making several assumptions, the likelihood ratio test, Λ, is computed using the following formula:

$$\Lambda = \frac{L_{H_0}(\hat{K}_1, \hat{K}_2)}{L_{H_1}(\hat{K}_1, \hat{K}_2)} = \frac{L_{H_0}(\hat{K})}{L_{H_1}(\hat{K})}$$

It is also true that by letting $W = W_1 + W_2$ and taking advantage of the fact that $tr(\widehat{K}_i W_i) = n_i tr(\widehat{K}_i \widehat{K}_i^{-1}) = n_i p$ and $tr(\widehat{K}W) = (n_1 + n_2) tr(\widehat{K}\widehat{K}^{-1}) = (n_1 + n_2)p$, the following two formulas emerge.

$$\Lambda = \prod_{i=1}^{2} \left(\frac{det\, \widehat{K}}{det\, \widehat{K}_i} \right)^{\frac{n_i}{2}}$$

$$-2 \log \Lambda \;= \sum_{i=1}^{2} n_i \log \left(\frac{det\, \widehat{K}_i}{det\, \widehat{K}} \right)$$

If indeed the null hypothesis is false, the graphical methodology enables us to evaluate the causes of differences between the two concentration matrices. In particular, if the graph is decomposable, it is feasible to break it down into its maximal complete subgraphs (cliques) and perform the preceding test for each clique. Following the standard procedures, the equivalence of the covariance matrices on cliques can be evaluated. However, if the graph is not divisible, more edges can be introduced to create a new triangulated and thus decomposable graph. The graph's cliques can then be used to conduct the test.

Eventually, the pathway's differential expression is examined. The differential expression of the pathway, if the null hypothesis is not rejected, is evaluated by hypothesis

$$H_0: \mu_1 = \mu_2 \text{ subject to } \Sigma_1 = \Sigma_2.$$

Exact approaches, such as multivariate analysis of variance, can be used to carry out this test. If the null hypothesis of homogeneity is rejected, the hypothesis that must be tested is

$$H_0: \mu_1 = \mu_2 \text{ subject to } \Sigma_1 \neq \Sigma_2.$$

In a two-sample scenario with unequal covariance matrices, this is the standard test for equality of means, also known as the Behrens-Fisher problem.

PARADIGM

PARADIGM stands for Pathway Recognition Algorithm using Data Integration on Genomic Models and aims to infer the activity of genetic pathways from integrated patient data. The methodology is outlined in [27] and is summarized below.

The implementation starts by creating a separate probabilistic model for each pathway. A factor graph was created using a pathway diagram that included both concealed and observable states. Also, we employ variables to characterize the states of entities in a cell to illustrate a biological pathway with a factor graph.

The factor graph uses a random variable $X = x_1, x_2, \dots, x_n$ for each entity and a set of m non-negative functions, or factors, to restrict the entities' ability to take on biologically meaningful values as functions of one another to represent the status of a cell. A probability distribution over a subset of entities $X_j \subset X$ is defined by the j-th factor ϕ_j. The joint probability distribution over all the entities is encoded in the whole graph of entities and factors as follows:

$$P(X) = \frac{1}{Z}\prod_{j=1}^{m} \phi_j(X_j)$$

Each entity can be active, nominal, or deactivated in relation to a control level, and these states are encoded as 1, 0 or -1 correspondingly.

To make factor building easier, we turn the pathway into a directed graph, with each edge annotated with a positive or negative influence. In the directed graph, every interaction in the pathway is turned to a single edge. Then, a list of factors is created to define the factor graph using this directed graph. Eventually, we complete the integration of pathway and multi-dimensional functional genomics data by adding observation variables and factors to the factor graph.

Subsequently, we want to know if a particular hidden entity x_i is likely to be in state α based on patient data.

$$P(x_i = \alpha | \Phi) = \frac{1}{Z}\prod_{j=1}^{m} \sum_{S \sqsubset_{A_i(\alpha)} X_j} \phi_j(S)$$

Similarly, the likelihood that x_i is in state α, based on all the patient's observations, is:

$$P(x_i = \alpha, D | \Phi) = \frac{1}{Z}\prod_{j=1}^{m} \sum_{S \sqsubset_{A_i(\alpha) \cup D} X_j} \Phi_j(S)$$

The expectation-maximization (EM) technique is used to estimate the parameters of the observation factors. A factor graph for each patient is generated, the patient's data are applied, and EM runs until the likelihood changes by less than 0.1 percent for each pathway. The factors learned from each pathway were averaged, and the final posterior estimates for each variable were calculated using these parameters.

Afterwards, a matrix of Integrated Pathway Activities (IPA) is generated for each variable with an ‘active’ molecular type after inference. A log-likelihood ratio that describes the level to which a patient's data boosts our opinion that entity *i*'s activity is up or down is calculated based on the following formula.

$$L(i,\alpha) = log\left(\frac{P(D, x_i = \alpha|\Phi)}{P(D, x_i \neq \alpha|\Phi)}\right) - log\left(\frac{P(x_i = \alpha|\Phi)}{P(x_i \neq \alpha|\Phi)}\right) = log\left(\frac{P(D, x_i = \alpha, \Phi)}{P(D, x_i \neq \alpha, \Phi)}\right)$$

Based on the log-likelihood ratio, a single IPA for gene *i* is computed as follows:

$$IPA(i) = \begin{cases} L(i,1)\,, & L(i,1) > L(i,-1) and\ L(i,1) > L(i,0) \\ -L(i,-1)\,, & L(i,-1) > L(i,1)\ and\ L(i,-1) > L(i,0) \\ 0\,, & otherwise \end{cases}$$

The IPA score is a signed equivalent of the log-likelihood ratio, L. It is set to L if the gene is more likely to be activated. If the gene is more likely to be inhibited, the IPA is set to -L; otherwise, it is set to 0.

Two alternative permutations of the data are used to measure the significance of IPA scores. A permuted data sample is constructed for the 'within' permutation by selecting a random tuple of data first from a random real sample, and then from a random gene within the same network, until tuples have been selected for each gene in the pathway, while the technique for the 'any' permutation is much like the 'within' permutation method, except the random gene selection stage could pick a gene from anywhere in the genome. In both cases, 1000 permuted samples are constructed, and perturbation scores are calculated for each permuted sample. To evaluate the significance of real samples, the distribution of perturbation scores from permuted samples is employed as a null distribution.

GGEA

Gene Graph Enrichment Analysis (GGEA), as stated in [28], is a method that uses previous information acquired from directed gene regulation networks to find consistently and coherently enriched gene sets.

The method on which GGEA is based consists of three critical stages. Initially, to create an induced subnetwork, the gene set is first mapped into the fundamental regulatory network. This is the part of the network that is impacted, which is made up of edges that involve members of the gene set. Next, each edge of the induced network is evaluated for consistency with the expression data, i.e. the signals of two interaction partners' expression changes are compared to the regulatory type (activation/inhibition) of the link connecting both genes. Finally, using a permutation process, the edge consistencies are aggregated over the induced network, normalized, and significance evaluated.

Consistency is calculated using the following formula:

$$C(t) = cons(de_O, f_t(de_i))$$

The raw GGEA consistency score *S* is induced by summing the consistency of all gene regulatory network (GRN) transitions and then normalized by the number of transitions, to compensate for the GRN’s size.

$$S \coloneqq \sum_{t \epsilon T_u} C(t)$$

$$\bar{S} \coloneqq \frac{S}{|T_u|}$$

Eventually, for each gene set we estimate the consistency P-value and rank the gene sets based on the adjusted P-values. Gene sets that are significantly and persistently enriched fall below the predefined significance level.

HotNet

HotNet is another approach for detecting significantly altered subnetworks in a large gene interaction network, that was initially designed for cancer mutation data [29]. The method for identifying cancer pathways that have been significantly mutated is presented below and is based on [30].

First, the model that will be used for the interpretation of the data is defined. Graph $G = (V, E)$ is used to model the interaction network, while $T \subseteq V$ corresponds to a subgroup of the genes that were tested. Each g gene is classified as either mutant or normal. The notation M_i is used to describe a subset of mutant genes in group T, while S_j denotes the samples under which the gene $g_j \in T$ has been altered, and m indicates the total number of mutant genes detected across all samples. A linked subgraph of G is defined as the resultant pathway.

After that, the influence graph, that encodes the knowledge in the interaction network, is constructed. The importance of a subnetwork is determined by (i) the number of samples with mutations in the subnetwork's genes, and (ii) the linkages among genes in the subnetwork in the context of the overall network's topology. On the interaction network, a diffusion process is employed to create a strict level of influence across all network nodes. The procedure outlined by Qi et al. (2008) is used to calculate the effect of node *s* on all other nodes in the network, and therefore, the influence graph $G_I = (T, E_I)$ with the collection of nodes belonging to the subset of tested genes is obtained. The weight of each edge $w(g_j, g_k) = [i(g_k, g_j), i(g_j, g_k)]$ is also considered.

Then, to discover altered paths, a combinatorial model is developed. First, collections of nodes in the influence graph G_I *that* are (1) related by high-influence links and (2) relate to mutated genes in many samples, are selected. A threshold δ is determined and, by deleting all edges with $w(g_i, g_j) < \delta$ and all nodes belonging to genes in the sample data with no modifications, a reduced impact graph $G_I(\delta)$ of G_I is constructed. Consequently, the size of the identified related subgraphs is determined by a threshold δ, which is entirely reliant on the null hypothesis. The connected maximum coverage problem, which is an NP-hard problem, is analogous to discovering the linked subgraph of *k* genes that is mutated in the maximum number of samples. To make the algorithm run properly, a modified version of the combinatorial algorithm shown in Figure 1 is used, in which for each pair of nodes (u, v), all the shortest paths between *u* and *v* are evaluated, and the one that optimizes $\frac{|P_v(u)|}{|l_v(u)|}$ is preserved.

Figure 1. Pseudocode for the combinatorial model's algorithm

Combinatorial Algorithm

```
Input: Influence graph $G_I$ and parameters $\delta$ and k
Output: Connected subgraph C of $G_I(\delta)$ with k vertices
    Construct $G_I(\delta)$ by removing from $G_I$ all edges with weight < $\delta$;
    $C \leftarrow \emptyset$;
    for each node $v \in V$ do
        $C_v \leftarrow \{v\}$;
        for each $u \in V \backslash \{v\}$ do $p_v(u) \leftarrow$ shortest path from $v$ to u in $G_I(\delta)$;
        while $|C_v| < k$ do
        $//\boldsymbol{\ell_v(u) = set\ of\ nodes\ in\ p_v(u);\ P_v(u) = elements\ of\ I\ covered\ by\ \ell_v(u);\ P_{C_v} = elements\ covered\ by\ C_v;\ P_C = elements\ covered\ by\ C}$
        $\boldsymbol{u \leftarrow arg\ max_{u \in V \setminus C_v : |l_v(u) \cup C_v| \leq k} \left\{ \frac{|P_v(u) \setminus P_{C_v}|}{|l_v(u) \setminus C_v|} \right\}}$;
        $\boldsymbol{C_v \leftarrow \ell_v(u) \cup C_v}$;
    if $|\boldsymbol{P_{C_v}}| > |\boldsymbol{P_C}|$ then $\boldsymbol{C \leftarrow C_v}$
return $C$;
```

Subsequently, to detect mutated subnetworks, a computationally efficient enhanced influence model is generated. The Enhanced Influence Model is rooted in the idea of increasing the influence measure among genes by the number of mutations found in each of these genes, and then breaking the resulting enhanced influence graph into linked components.

H refers to the enhanced influence graph. All genes g_j having at least one mutation in the data make up the set V_H of *H*'s vertices, while the improved influence

$$h(g_j, g_k) = w(g_j, g_k) \times \{|S_j|, |S_k|\}$$

determines the weight of the edge (g_j, g_k). Then, to produce a graph *H(δ)*, whose linked components represent the significant subnetworks, any edges with a weight less than a threshold δ are eliminated.

Eventually, a statistical analysis is performed to determine the network's significance. There are two null hypothesis distributions considered:

i. H_0^{sample} in which $m = \sum_i |M_i|$ mutations are randomly distributed throughout the nodes correlating to the $|T|$ tested genes

ii. H_0^{gene} which is obtained by permuting the identities of the network's evaluated genes, using a random permutation σ

A two-stage multi-hypothesis test is executed and the Family Wise Error Rate (FWER), that is the probability of making at least one Type I error in any of the tests, is used as the rigorous indicator of its significance level. The False Discovery Rate is a less conventional alternative to minimizing errors in multiple testing (FDR). It is denoted as $FDR = E[V/R]$, where V represents the number of Type I errors and R represents the total amount of null hypotheses excluded. The two-stage test identifies several subnetworks in the data as statistically significant with low FDR values.

Using a similar approach to that used in the Combinatorial model, it is shown in this study how the number of hypotheses can be limited to merely $K = |T|$ hypotheses. The first stage of evaluating each hypothesis with confidence level α/K determines the smallest size s, such that the null hypothesis that the number of linked components of size $\geq s$ detected in the graph H(d), r_s, may be rejected with confidence level α. The test also includes a second criterion that ensures that the FDR is kept within a certain range.

A Monte-Carlo simulation ("permutation test") or analytical bounds can be used to calculate the null hypothesis distributions. Two properties of the Monte-Carlo simulation approach considerably minimize the cost of the estimations. The Monte Carlo simulation must be done on the graph G_I. The p-value of the distribution of the number of connected linked subgraphs/components of a particular size is used in the statistical test. As a result, it is essential to determine p-values that are a magnitude larger for this test, using vastly fewer simulation rounds.

Using analytical bounds, the null hypothesis can be approximated for a greater number of tested genes. For any node g_i in G_I, the maximum δ is set such that the weight of less than $\alpha M/|T|$ connected edges gratify $s_{max} w(g_i, g_j) \geq \delta$, for any given $\alpha < 1$.

PRS

Pathway Regulation Score, or else PRS, is a method which distinguishes between essential processes in real-world biological datasets. The procedure that follows is a simplified version of the method as described in [31].

The data were first pre-processed using the Robust Multiarray Average (RMA) approach, and the DEG lists were produced using simple fold change and p-value calculations. The pathways retrieved by the KEGG database were represented in the form of a graph. Due to redundancies in KEGG pathways, fold-change values for a node may be assigned to a route several times, resulting in a skewed PRS calculation. Consequently, a new structure emerged, in which duplicated genes were unified into a single term with a unique ID.

In order to implement the PRS algorithm, the pathways were represented as networks, so that each pathway is characterized by a unique identity, definition name, and its corresponding nodes. Particularly, a pathway's nodes are described by the following attributes. *Node_genes* correspond to a distinct function that maps to one or more transcripts and *Node_value* (*NV*) represents a value based on expression data. *Node_weight* (*NW*) concerns only the significant nodes and indicates their structural strength. The *Node_Score* (*NS*) is calculated by combining the *NV* and *NW* values.

$$NS = \begin{cases} NV * NW & if\ NV > 1 \\ 0 & if\ NV \leq 1 \end{cases}$$

Subsequently, using the following formula the PRS is determined:

$$PRS(p_i) = \sum_{j=1}^{n_i} NS_j$$

Prior to rating the paths, a normalization step is performed to account for two crucial features.

i. Pathway size: the bias caused by pathway size was reduced as seen in the following equation.

$$PRS(p_i) = \left(\sum_{j=1}^{n_i} NS_j\right) * \left(\frac{NDEGs_i}{NEGs_i}\right)$$

ii. Pathway-specific PRS score null distributions that contribute to statistical bias: a nonparametric permutation approach is employed to determine the null distributions of raw PRS values acquired for each pathway.

$$nPRS_i = \frac{PRS_i - mean(pPRS_i)}{STD(pPRS_i)}$$

$$npPRS_{ij} = \frac{pPRS_{ij} - mean(pPRS_i)}{STD(pPRS_i)}$$

For pathway ranking, the normalized raw scores ($nPRS_i$) were used.

To determine the significance, the PRS values were recalculated using the equation used to reduce the bias caused by the pathway size, after the fold-change values for the full gene list were permuted and mapped back onto pathways. To construct a null distribution of each raw score, this procedure was repeated 1000 times. Then, the normalized scores were compared, and the p-values were determined as shown below:

$$P(nPRS_i) = \frac{\sum_{j=1}^{n} I\left(npPRS_{ij} \geq nPRS_i\right)}{n}$$

Finally, a multiple test adjustment was implemented, and the FDR modified the P$_{\text{final}}$ to account for type I errors.

DEGraph

DEGraph is yet another pathway analysis tool that uses modern hypothesis testing approaches to predict whether a specific gene network is differentially expressed between two scenarios and is very useful in cancer research [32]. The step-by-step methodology is defined in [33].

First, a lower-dimension basis is constructed, after which the multivariate test of means is used. The testing question of whether two sets of random vectors of gene expression measures are expected to have emerged from equal-mean distributions, can be directly formulated, and solved using multivariate statistics.

A network of p genes is depicted as graph $G = (V, E)$, having $|V| = p$ nodes and edge set E, while δ refers to the mean shift, to wit, the vector of differences between the p genes' mean expression values among the two study populations. Afterwards, a lower-dimensional $k \ll p$ space is constructed, retaining most of the low-energy functions $E_G(\delta)$. To accomplish this, the function with the least potential energy is identified, followed by the function with the lowest possible energy in the orthogonal space of the last one, and so on, up to the kth function with the minimum energy in the orthogonal subspace of the first $k-1$ functions.

$$u_i = \begin{cases} \arg\min\limits_{f \in \mathbb{R}p} E_{\mathcal{G}}(f) \\ such\ that\ \ u_i \perp u_j, j < i \end{cases}, i \leq k$$

The following energy function states that if the variation in mean expression of any gene among the two populations is equivalent to the (signed) average of the difference between the mean expression for the genes that either activate or inhibit it, an expression shift will have limited power:

$$E_{\mathcal{G}}(\delta) = \sum_{i:d_i^- \neq 0}^{p} \left(\delta_i - \frac{1}{d_i^-} \sum_{(j,i) \epsilon \mathcal{E}} a_{ji} \delta_j \right)^2$$

The number of directed edges leading from any node to u_i is denoted by d_i^-.

Then, to achieve orthonormal functions with low intensity, the first few eigenvectors of M_G are employed to construct a lower dimension space.

Following that, a graph-structured two-sample test statistic is demonstrated. Hotelling's T^2-test, a classic location shift test, is a consistently most powerful invariant against global-shift alternatives for multivariate normal distributions. The statistical test $T^2 = \frac{n_1 n_2}{n_1+n_2}(\bar{x}_1 - \bar{x}_2)^\intercal \hat{\Sigma}^{-1}(\bar{x}_1 - \bar{x}_2)$ is predicated on the sample mean shift's squared *Mahalanobis norm*. In this work, T^2-statistics follow the nominal *F*-distribution, while Hotelling's test in the new area limited to its first k components is said to generate greater power than testing in the complete new space.

Subsequently, a systematic way for identifying nonhomogeneous subgraphs, or subgraphs of a large graph with a significant shift in means, is to examine each one individually. Due to the huge combinatorial issue posed by large sizes of graphs, it's critical to rapidly discover groups of subgraphs that all fit the null hypothesis of equal means. This is achieved by using a threshold on the value of the test statistic for every subgraph containing a particular network. The corresponding algorithm is described below.

Figure 2. Nonhomogeneous subgraph discovery algorithm

Nonhomogeneous subgraph discovery algorithm

```
Input: $G$, $X_1$, $X_2$, $\alpha$, q
Output: selectedSubgraphs
  selectedSubgraphs = ∅;
  previousSubgraphs = nodes ($\boldsymbol{G}$);
  prunedSubgraphs = ∅;
  for each $s$ ∈ {1 ...$q$-1} do
      checkedSubgraphs = ∅;
      for each previousSubgraph do
          for each subgraph ∈ subgraphBoundary(previousSubgraph) do
              if subgraph has been checked or has a pruned subgraph then next;
              if $s$ < $q$-1 then
                  if $\boldsymbol{upperBound(subgraph, G, X_1, X_2, q) < T^2_{\alpha,k}}$ then
                      add subgraph to prunedSubgraphs;
                  else
```

```
                    add subgraph to currentSubgraphs;
                **end**
            **else**
                **for** each q-subgraph ∈ subgraphBoundary(subgraph) **do**
                    **if** q-subgraph has been checked or has a pruned subgraph
                      **then** next
                    **else**
                        **if** $\tilde{T}_k^2(q - subgraph, X_1, X_2) > T_{\alpha,k}^2$ **then**
                            add q-subgraph to selectedSubgraphs
                        **end**
                        add q-subgraph to checkedSubgraphs
                    **end**
                **end**
            **end**
            add subgraph to checkedSubgraphs
        **end**
    **end**
    set previousSubgraphs to currentSubgraphs
**end**
```

In the case of "limited" graphs over a certain level of connectivity and q large enough, $v(g', q - s)$, the (q - s)-neighborhood of g, increases at the initial stage of the above exact process, while the number of tests being conducted may not reduce significantly considering the number of feasible tests. As a result, a faster, approximation algorithm is introduced. The main idea is to find subgraphs with sample mean shifts in the first k components of a new space, where the Euclidean norm $\|\hat{\hat{\delta}}_{[k]}(g)\| = \|U_{[k]}^T\left(\underline{x}_1(g) - \underline{x}_2(g)\right)\|$ is greater than a specified threshold. The output of substituting the *upper-Bound* with the following inequality in the Nonhomogeneous subgraph discovery algorithm produces an upper bound on $\tilde{T}_k^2(g)$.

$$\begin{aligned}\|U_{[k]}^{T}\left(\bar{x}_1(g)-\bar{x}_2(g)\right)\|^{2} \leq \|U^{T}\left(\bar{x}_1(g)-\bar{x}_2(g)\right)\|^{2} &= \|\bar{x}_1(g)-\bar{x}_2(g)\|^{2} \\ &\leq \|\bar{x}_1(g')-\bar{x}_2(g')\|^{2} \\ &+ \max_{v_1,\ldots,v_{q-s}\in\nu(g',q-s)} \|\bar{x}_1\left(u_1,\ldots,u_{q-s}\in\nu(g',q-s)\right) \\ &- \bar{x}_2\left(u_1,\ldots,u_{q-s}\in\nu(g',q-s)\right)\|^{2}\end{aligned}$$

This specifies a technique for detecting all subgraphs whose sample mean shift's Euclidean norm exceeds a certain threshold. Employing the T^2-test on these preselected subgraphs can also predict the group of subgraphs produced by the Nonhomogeneous subgraph discovery procedure.

A major issue with DE genes is the classification of non-significant differences as significant. The approaches presented by Lönnstedt and Speed (2002) can be used to solve such a problem.

Finally, the problem of multiple testing is raised due to the huge number of subgraphs assessed for homogeneity. This issue can be resolved by employing a permutation technique, which minimizes the amount of false positive subgraphs. Initially, the $n_1 + n_2$ observations' class/population labels are permuted, and then the nonhomogeneous subgraph discovery process is implemented to the permuted data to give a specific amount of false positive subgraphs. This technique is repeated several times to provide an approximation of the distribution of Type I error rates.

The procedures' performance is first assessed on synthetic data, and subsequently on breast cancer microarray data examined using KEGG pathways.

TEAK

Based on [34], Topology Enrichment Analysis frameworK (TEAK) was developed to discover active subpathways that underpin biological processes making use of the KEGG pathway database. Nodes represent gene products and/or complexes of gene products, while edges denote relationships between proteins or enzymes and are exploited to generate a set of unweighted adjacency matrices, which illustrate the KEGG pathways.

The subpathways extracted by the previous procedure can be either linear or nonlinear. Subpathways that consist of root to leaf linear paths are identified as linear. On the contrary, nonlinear subpathways are distinguished by feed-forward loops that are adjacent and overlap.

To evaluate the linear and nonlinear subpathways, TEAK initially fits a context specific Gaussian Bayesian network for each subpathway using the Bayes Net Toolbox. A Gaussian Bayesian network is a Bayesian network that is a probabilistic graphical model, with all its nodes being linear Gaussians. Specifically, the Conditional Probability Distribution of Y for a continuous node Y with m continuous parents X_1, ..., X_m is:

$$p(Y|x_1, \dots, x_m) = N(\beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m;\ \sigma^2)$$

Subsequently, for subpathways consisting of context specific data, TEAK uses the Bayesian Information Criterion (BIC) provided in the Bayes Net Toolbox and scores each Bayesian network.

$$Score_{BIC} = \log P\left(D|\hat{\theta}\right)\ - 0.5d \log N\ .$$

Finally, since BIC is capable of breaking, which means that each node's score is computed separately and then added together to get the final score, each sub-pathway's value is normalized by the amount of nodes in order to make the scores equivalent.

PATHiWAYS

PATHiWAYS is another tool for pathway analysis. More specifically, it is a web server that can interpret the consequences of multiple changes in gene expression levels when it comes to signaling pathways. The approach, as detailed in [35], is explained below.

It is based on a probabilistic model of the pathway, in which the probabilities of signal transmission are calculated. Gene expression values represent the gene activity and, therefore, the presence or absence of a protein. The 90th percentile of the distribution of the probe activation probabilities is used in order to reduce the number of false positives caused by faulty probe observations.

After calculating the individual probability of each node, depending on the number of proteins they are composed of, a simple product of probabilities is computed to estimate the probability of signal transmission through the pathway.

Eventually, the final probabilities are compared and detect which stimulus–response circuits had their probabilities of signal transmission significantly altered.

DEAP

DEAP, which is short for Differential Expression Analysis for Pathways, is a pathway analysis method for identifying relevant regulatory patterns from differential expression data that takes advantage of information about biological pathways. Unlike previous methods of analysis, DEAP takes advantage of existing knowledge about pathway structure and recognizes the path that is the most differentially expressed. This technique calculates the scores of each subpathway with the use of the DEAP algorithm, which is determined in [36].

Initially, to estimate the null distribution of the test statistics and compute the p-values, a random rotation technique was employed. Rotation testing asserts that pathway and set data come from independent random samples of a multivariate normal distribution with mean zero under the null hypothesis.

Subsequently, the DEAP algorithm is applied. The algorithm handles expression data that are composed using the following multivariate normal distribution:

$$E = d(\mu + g) + e \,,$$

where d signifies if a gene is ‘on’ or ‘off’, μ represents the ‘pathway effect’ and g and e are generated using a normal distribution with both means equal to 0 and variances ${\sigma^2}_g$ and 1 respectively.

The procedure is based on the following discrete steps. At first, the expression data are overlaid onto the network and each path from the graph is separately examined. Afterward, a recursive function is implemented and estimates the minimum and maximum differential expression for each pathway considering the type of relationship between nodes by adding or subtracting all downstream nodes.

The preceding method is summarized by the following formulas: $Score = \sum_{z \in reactants} E(z) + T(edge) * max_{recursive}$ and $Score = \sum_{z \in reactants} E(z) + T(edge) * min_{recursive}$

The edge type is represented by $T(edge)$ in the formula above and can either be 1 for activation or -1 for inhibition.

The path with the maximal absolute differential expression is detected by using and comparing the absolute value of the differential expression calculated for each pathway on the previous step.

Subsequently, the data were rotated n times to simulate a null distribution of the test statistic, s*, and the DEAP score was recalculated for every rotation sample. The random rotation approach helps resolve difficulties, such as not directly comparable DEAP scores for different paths, due to variances in size and structure among pathways, and determines the statistical significance.

Finally, the *p*-value is determined as the proportion of simulated DEAP scores, whose value is greater than or equal to the observed DEAP score, divided by the number of scores that are at least as extreme as the observed DEAP score:

$$p = \frac{\#(s_i \geq s\,*)}{n}$$

GraphiteWeb

Another option is GraphiteWeb, which is an innovative web tool for network analyses and visualization for gene expression data from both microarray and RNA-seq studies. Given [37], it integrates topological and multivariate pathway studies with an efficient model of interactive network representations for simple comprehension of the results and uses a variety of multivariate gene set techniques. In addition, it uses multivariate gene set analysis including conventional hypergeometric enrichment, global test, GSEA, SPIA, and CliPPER, as well as the KEGG and Reactome pathway databases. In this paper, we will focus solely on the Enrichment Analysis (competitive and non-topological) approach.

Enrichment analysis uses the Fisher Exact test to estimate the odds of finding a certain number of genes in a specific pathway among the DEGs, denoted as $n_{G,deg}$. Within a set of N_{deg} genes, the likelihood P of seeing at least $n_{G,deg}$ genes is calculated by

$$P(N_{G,deg} \geq n_{G,deg}) = \sum_{i=n_{G,deg}}^{N_{deg}} \frac{\binom{N_G}{i}\binom{N-N_G}{N_{deg}-i}}{\binom{N}{N_{deg}}},$$

where N is the actual population of genes tested, G denotes the pathway, and N_i and n_i measure the frequency of genes within every table cell.

Subsequently, using the Benjamini and Hochberg technique, *P*s are normalized.

The statistical methodology typically employed to identify DEGs in RNA-seq count data is built on the negative binomial distribution. Given the strong relationship between read count and gene length, the read count specifies the test's power in this scenario. The P-value correction for gene length is an option in graphite web for adjusting for this bias.

PATHOME

PATHOME stands for Pathway and Transcriptome Information and is a computational approach for detecting differentially expressed subpathways. Its methodology from [38] is based on gene expression profiles of two control groups and relevant biochemical pathways.

At first, PATHOME divides the pathways into subpathways and then uses statistical tests to assess the significance of differential expression profiles alongside the pathway. The type of interaction is also considered.

The decomposition of the main pathway into linear paths is achieved using a depth-first search algorithm. Due to the huge number of possible paths resulting from the previous step, a selection step is used prior to the statistical significance test step to avoid such difficulties.

In order to select which segment of the subpathway will be statistically reviewed in the test step, the following rule is applied:

$$I^k = arg\min_{m}\left\{-\sum_{i=1}^{m} I\left(sgn\left(r_{i,i+1}^{k} \times e_{i,i+1}\right) = 1\right) + \sum_{i=1}^{m} R\left(sgn\left(r_{i,i+1}^{k} \times e_{i,i+1}\right)\right)\right\}$$
$$+ 1, m\epsilon\{1,\dots,p-1\}, R(x) = \begin{Bmatrix} 0, if\ x \in \{1\} \\ \infty, otherwise \end{Bmatrix}$$

A subpathway is chosen and continuous to the test step if the association rule between the expression correlation and the edge information for the neighboring items along the path is agreed upon by both experimental groups, and both consecutive segments include at least four components.

The final step determines which subpathway has a statistically significant difference in correlation between two subsequent segments for the two studies. The significance is examined under the null hypothesis, in which the alternative hypothesis represents the case in which the global mean of the correlations between the two groups are different.

Finally, to determine significance, we employed the z-test statistic, considered multiple comparisons, and the FDR was set at 0.05.

SubSPIA

The SPIA approach, mentioned above, was paired with a current subpathway analysis method to create the sub-SPIA method, which was used to find cancer-related pathways. [39] provides the exact technique. To avoid problems resulting from the k-clique structure, used to define subpathways in the original subpathway analysis, the sub-SPIA method uses the minimal-spanning-tree structure.

A minimal-spanning tree is a tree-like subgraph, in which all nodes are connected, without forming a cycle. Because of the sparse connections between genes and the indirect connectivity of DEGs, this technique outperforms the k-clique notion.

The implementation of sub-SPIA was done using the R programming language. The following steps outline the main idea behind this method. Initially, we reassemble the gene network based on the signaling pathway. Then, in the gene network that has been created, the DEGs are mapped, and, finally, the subpathways are identified and their statistical and perturbation significance is evaluated. The Kruskal algorithm is used to create the minimal-spanning tree and then remove any non-signature nodes remaining in the leaves of the MST.

To determine the statistical significance of each subpathway, the hypergeometric test and anomalous perturbation are employed. As in the SPIA tool, the present method contains two probabilities, P_{NDE} and P_{PERT}. The *p*-value can then be used to determine the pathway's enrichment significance using the following equation:

$$p = 1 - \sum_{x=0}^{r-1} \frac{\left(\frac{t}{x}\right)\left(\frac{m-t}{n-x}\right)}{\left(\frac{m}{n}\right)}$$

In the equation above, *m* represents the total number of genes in the genome, while t is the number of genes involved in the studied pathway. The variable t denotes the number of genes provided for study, of which r are participating in the same pathway.

The P_{PERT} is calculated the same way as shown in the SPIA method and is combined with P_{NDE} in order to form a new global probability, P_G.

MinePath

Another tool for pathway analysis is MinePath, which combines knowledge from gene expression profiles and molecular pathways. MinePath recognizes functionally differential sub-paths among different phenotype classes. Focusing on [4], bellow is a basic outline of the methodology.

MinePath's general technique consists of five modular components that must be implemented. Initially, the gene expression data must be discretized so that the domain dimensionality is reduced. The supervised Entropy-based global discretization approach was implemented to convert gene expression values into binary equivalents that are either high (expressed) or low (non-expressed). In addition, because of the differences in the nomenclature of pathways and gene expression data, MinePath examines each gene's various probesets and indicates a combined expression value by employing a logic OR to the probeset values.

Then, following a depth-first search technique, each pathway is broken into all its sub-paths. The sub-paths that emerged are compared to the binary gene expression sample profiles provided as input. A collection of binary (Boolean) operations and a number of semantics that interpret the precise molecular nature of the underlying gene interactions are used to determine the functional capacity of a sub-path in a sample. MinePath handles two types of single gene regulatory relationships: activation/expression and inhibition, which are described by the AND and XOR Boolean operators, respectively. In the event of more complex sub-paths, MinePath uses an AND operator to merge the binary values of the path's final relation and the binary value of the sub-component path's investigated so far.

Subsequently, the most discriminant sub-paths are identified using a multi-parametric sub-path selection technique, which is implemented by the employment of feature selection and classification techniques. MinePath includes three independent filters to analyze the phenotype differential power of sub-paths and identify the most discriminant among them: coverage, p-value, and polarity, each with its own customizable threshold. Those sub-paths that pass all the filters are chosen and maintained as the most discriminant.

Finally, MinePath evaluates the relevance of the pathways and ranks them according to their p-value, which is calculated based on the following formula:

$$p - value_p = \frac{\left(\frac{(a_p - 1) + b_p}{a_p - 1}\right)\left(\frac{c_p + b_p}{c_p}\right)}{\left(\frac{n}{(a_p - 1) + c}\right)}$$

MinePath's final output is a p-value ranked list of pathways from which the user can choose one to visualize and study.

HiPathia

HiPathia is a method that uses transcriptome data to calculate signal transduction along signaling pathways. [40] and [41] explain the technique.

To model the various cell functions in detail, each pathway is first broken down into circuits. The algorithm used by the HiPathia method models signal propagation by considering the level of activity of the proteins that make up the circuit. The simultaneous presence of the chain of proteins that connect the receptor to the effector, as well as the absence of inhibitor proteins that could compromise the signal's transduction along the circuit, in order to be active and thus transduce the signal to eventually trigger a function, is necessary for a circuit. The signal generated by the input node is communicated along the pathway as in the direction of the interactions and the output is collected by an output node, which activates a cell function. The signal is transmitted along the path according to the following recursive formula:

$$S_n = \upsilon_n \cdot \left(\prod_{S_a \epsilon A} (1 - S_a) \right) \cdot \prod_{S_i \epsilon I} (1 - S_i)$$

S_n and υ_n represent the signal intensity for the current node *n* and its normalized gene expression value respectively. *A* describes all the activation signals (S_a) that are collected at the current node *n* from activation edges, and *I* describes the corresponding inhibitory signals (S_i).

Finally, based on each circuit's structure, the last effector node will either detect variations in the nodes' activity or not.

Table 1. Pathway's scoring formula of Pathways Analysis tools

Method	*Date*	*Formula*
TAPPA	2007	$PCI = \sum_{i=1}^{N}\sum_{j=1}^{N} sgn\left(x_{is} + x_{js}\right) * \lvert x_{is}\rvert^{0.5} * \alpha_{ij} * \lvert x_{js}\rvert^{0.5}$
SPIA	2008	$P_G = c_i - c_i \cdot ln\,(c_i)\,,\ c_i = P_{NDE}(i) \cdot P_{PERT}(i)$
TopologyGSA	2010	$\Lambda = \frac{L_{H_0}\left(\widehat{K}_1, \widehat{K}_2\right)}{L_{H_1}\left(\widehat{K}_1, \widehat{K}_2\right)} = \frac{L_{H_0}\left(\widehat{K}\right)}{L_{H_1}\left(\widehat{K}\right)}$
PARADIGM	2010	$IPA(i) = \begin{cases} L(i,1)\,, & L(i,1) > L(i,-1) and\ L(i,1) > L(\\ -L(i,-1)\,, & L(i,-1) > L(i,1)\ and\ L(i,-1) > \\ 0\,, & otherwise \end{cases}$
GGEA	2011	$S \coloneqq \sum_{t \epsilon T_u} C(t)\,, C(t) = cons(de_o, f_t(de_i))$
HotNet	2011	$h\left(g_j, g_k\right) = w\left(g_j, g_k\right) \times \left\{\lvert S_j\rvert, \lvert S_k\rvert\right\}$
PRS	2012	$PRS(p_i) = \sum_{j=1}^{n_i} NS_j\,, NS = \begin{cases} NV * NW & if\ NV > 1 \\ 0 & if\ NV \leq 1 \end{cases}$
DEGraph	2012	$E_{\mathcal{G}}(\delta) = \sum_{i:d_i^- \neq 0}^{p} \left(\delta_i - \frac{1}{d_i^-} \sum_{(j,i)\epsilon\mathcal{E}} a_{ji}\delta_j\right)^2$
TEAK	2012	$Score_{BIC} = \log P\left(D \mid \hat{\theta}\right) - 0.5 d \log N$
PATHiWAYS	2013	Probabilistic model
DEAP	2013	$Score = \sum_{z \in reactants} E(z) + T(edge) * max_{recursive}$ and $Score = \sum_{z \in reactants} E(z) + T(edge) * min_{recursive}$
GraphiteWeb	2013	$P\left(N_{G,deg} \geq n_{G,deg}\right) = \sum_{i=n_{G,deg}}^{N_{deg}} \frac{\binom{N_G}{i}\binom{N - N_G}{N_{deg} - i}}{\binom{N}{N_{deg}}}$

PATHOME	2014	$I^k = arg\min_{m}\left\{-\sum_{i=1}^{m} I\left(sgn\left(r_{i,i+1}^{k} \times e_{i,i+1}\right) = 1\right) + \sum_{i=1}^{m} R\left(sgn\left(r_{i,i+1}^{k} \times e_{i,i+1}\right)\right)\right\} + 1, m\epsilon\{1, \dots, p-1\}, R(x) = \left\{\begin{matrix} 0, if\ x \in \{1\} \\ \infty, otherwise \end{matrix}\right\}$
SubSPIA	2015	$P_G = c_i - c_i \cdot ln\ (c_i)\ , c_i = P_{NDE}(i) \cdot P_{PERT}(i)$
MinePath	2015	Logical operators
HiPathia	2017	$S_n = v_n \cdot \left(\prod_{S_a \epsilon A}(1 - S_a)\right) \cdot \prod_{S_i \epsilon I}(1 - S_i)$

Materials and Methods

Datasets and processing

Two datasets are used in this study. The GSE2034 gene expression dataset was obtained from the Gene Expression Omnibus data repository available at https://ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=gse2034 and contains 286 breast cancer samples, of which 209 are ER-positive and 77 are ER-negative. The second dataset is a collection of 47 sub-paths in total from the KEGG database (https://www.genome.jp/kegg-bin/show_organism?menu_type=pathway_maps&org=hsa) of which 15 relate to cellular activities, 24 to signal propagation and 8 to cancer in general. The relation between two nodes is either activation or inhibition.

Due to the mapping of a gene to more than one Entrez identifiers, each Entrez identifier is translated to its associated gene. The expression value that results after the KEGG-IDs are combined into one gene is calculated as their average expression value. If a gene or gene ID involved in one of the investigated sub-paths does not match the dataset GSE2034, a new 'noProbe' gene is created with values equivalent to the mean of each corresponding sample.

The dataset of the sub-paths is handled in such a way that the nodes and edges for each sub-path can be distinguished and interpreted easier. More than one gene can be found in each node. In this case, the average value is assigned.

Moreover, the data from the two data sets is integrated to create new collective data structures. As a result, values such as expression values, p-values, and fold changes are directly linked to sub-path nodes, simplifying the analysis process.

Computing the score

The methods implemented in the present paper are TAPPA, SPIA, PRS, TEAK, DEAP, GraphiteWeb, MinePath and HiPathia (see Appendix for more details). The decision was made based on the degree of simplicity with which each method could be performed. The score of each sub-path was determined using the methodology described in each tool's respective papers. Each approach offers details regarding the data normalization procedure, although the FDR correction method was employed for the DEAP and PRS tools. Due to their computational complexity, FDR correction was used in place of DEAP's random rotation approach and PRS' second normalization step's non-parametric permutation method.

Despite the fact that all techniques take pathway topology into consideration, we may generally identify two approaches to the problem. Tools TAPPA, GraphiteWeb, TEAK and PRS are based on probability theories and interpret pathway topology as the influence of nodes upon one another. Tools HiPathia, SPIA, DEAP and MinePath, on the other hand, focus on the relation type among genes and how it influences the output.

After computing the score of each sample for each unique sub-path according to the approach of the corresponding tool, a two-dimensional matrix emerged for every one of the tools, with rows representing samples and columns representing sub-paths.

Subpathway ranking

After acquiring the scores of each sub-path for each tool independently, machine learning algorithms were used to train and evaluate the outcomes of each approach in order to perform the comparison among them.

Machine Learning is a subset of artificial intelligence that utilizes data and algorithms to model the learning process of a human while continually improving its accuracy [42]. Given that labeled data were available for this study and that the samples were to be divided into classes (ERpos, ERneg) based on the scores of the sub-paths, supervised learning techniques were considered. Thus, the Decision Trees algorithm was used to classify the findings and evaluate each approach since it also has the capability of recognizing the most important features; as a result, in addition to providing information about the accuracy of the results, it also provides insight into which sub-paths are significant.

The basis of the Decision Trees algorithm is the continuous division of the data by a given criterion. Two things -decision nodes and leaves- can explain the structure of the tree. The options or results are the leaves and the data is divided at the decision nodes [43].

Initially, since machine learning algorithms only comprehend numbers, the datasets representing the study’s results of each approach were prepared and the samples' categorical classes were transformed into numerical variables.

Then, the modified datasets were divided, at a ratio of 70% to 30%, into training and testing sets. Each Machine Learning model was trained with the training sets that emerged from the previous step and evaluated using the corresponding testing sets. The outcome is how accurate the models are for the data acquired from each approach and a list of the most significant sub-paths utilized as features for each approach's classification model predictions.

The scikit-learn package (https://scikit-learn.org/stable/) which offers pre-built machine learning functions for the Python programming language, was used to implement the aforementioned procedure.

Results

Predictive performance/ Data validation

The findings of each methodology employed in this study are provided in this section and are presented in Table 2 and Table 3.

Accuracy, which is the proportion of correctly predicted data points among all the data points [44], is the parameter used to assess the performance of each approach. The tool or tools with the highest accuracy are highlighted. Overall, the accuracy of every technique is at least satisfactory, and for tools with an accuracy rate greater than 75%, it can be described as very good.

It is important to note that, when the findings were corrected using the Benjamini and Hochberg approach in the case of GraphiteWeb, the classification model's accuracy declined. One interpretation of such a reduction is improved capacity for adaption to new, previously unobserved data.

Table 2. Predictive performance of each tool

Tool	*Accuracy*
TAPPA	65,12%
PRS	83,72%
TEAK	67,44%
DEAP	82,56%
GraphiteWeb	75,58%
MinePath	68,6%
HiPathia	**84,88%**

However, despite the relatively good accuracy results, no common sub-paths could be identified across the tools in terms of the feature importance of each methodology. Consequently, no further information could be derived about the pathways because our study was unable to provide us with the most discriminant sub-paths.

The execution time of the methods is an additional criterion for evaluation based on which we may determine which algorithms can be used for real-time and consequently online analysis. The total time required to execute the code on the computer's central processing unit (CPU) is referred to as the algorithm's execution time. Table 3 displays the execution time in seconds for each method’s scoring algorithm.

In general, we observe both algorithms that have a relatively short execution time and algorithms that require even days to complete. Nevertheless, most tools complete the analysis in less than a day, while there are methods that take even less than an hour to complete, which makes them ideal for online analysis.

Table 3. Execution Time of each tool

Tool	*Execution Time (seconds)*
TAPPA	3185.17
PRS	8754.69
TEAK	28918.27
DEAP	24294.49
GraphiteWeb	366047.77
MinePath	3099.49
HiPathia	**2905.48**

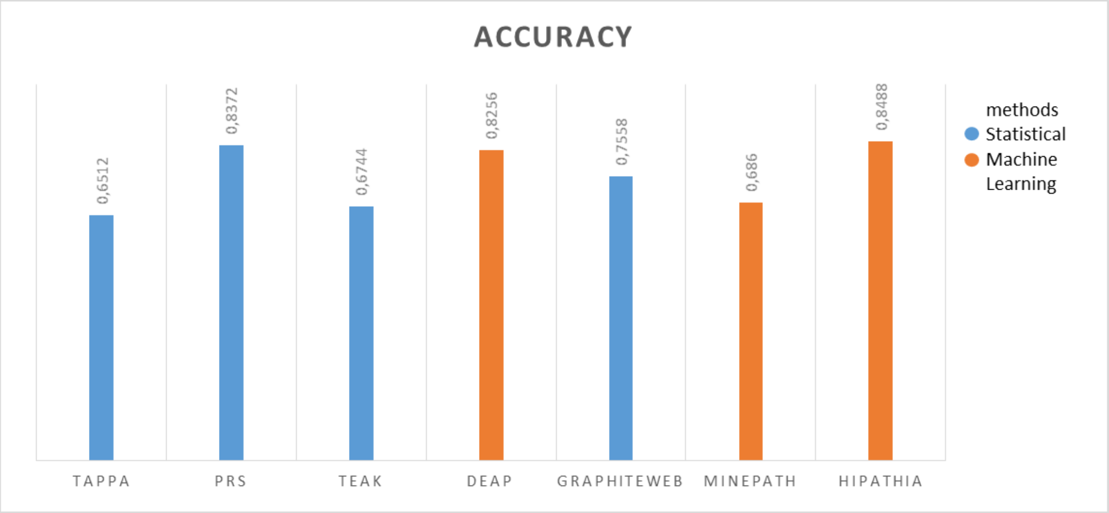
Tools comparison

This study's primary goal was to compare several pathway activity analysis tools that use statistical and machine learning techniques. The comparison considers both each method individually and the two fundamental approaches collectively (see Section “Computing the score”).

Primarily, as demonstrated in Table 2, HiPathia has the best predictive performance with a value of 84,88%, while the worst performance was TAPPA’s with 65,12% accuracy. The performance of the remaining tools range from 65% to 84% and exceeds 75% for more than half of them.

Overall, the tools with accuracy greater than 80% are PRS, DEAP and HiPathia; PRS is based on statistical methods, while DEAP and HiPathia are predicated on Machine Learning algorithms. It is obvious that, in this study, tools based on Machine Learning algorithms outperform those based on statistical methods on average, when the main approach of each method is taken into account.

Figure 3. Predictive Performance of each tool based on the method

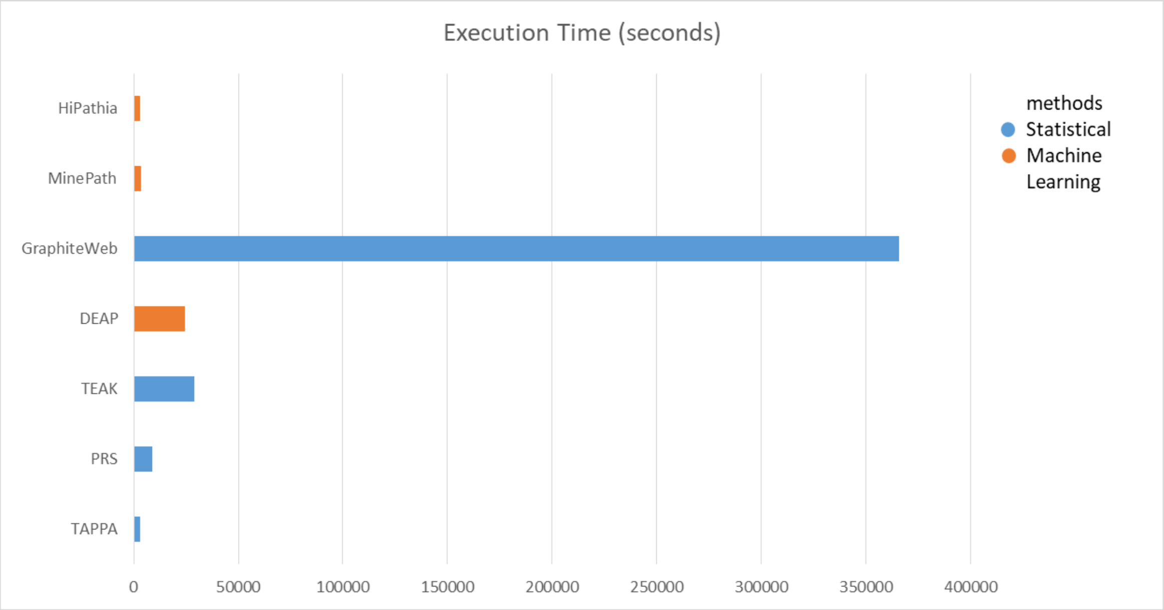


Regarding the execution time of the algorithms, the approach that seems to be the best solution for real-time analysis is HiPathia with a runtime of 2905,48 seconds corresponding to less than an hour, followed by MinePath and TAPPA. On the contrary, a tool such as GraphiteWeb could not be utilized for real-time analysis because the computation of the results takes around 4 days, which makes it impractical.

Most methodologies exhibit relatively low execution time as shown in Table 3, but there is always potential for improvement. The majority of the methods complete the analysis in less than a day, while tools TAPPA, MinePath and HiPathia are suitable for online analysis since the procedure takes less than an hour.

Machine learning algorithms also seem to have an advantage in terms of the execution time over statistical methods, since they are particularly useful in situations like these, when there is a massive amount of data to handle, in order for the model to become smarter and more reliable [45]. Additionally, it is known that statistical algorithms generally struggle to analyze very big datasets and demand smaller datasets with fewer features or data [46].

Figure 4. Tools' execution time based on the method



When both assessment criteria are taken into account, HiPathia's approach is identified as the best option, because it offers great accuracy rate and low execution time. This conclusion is consistent with existing research, which claims that machine learning algorithms are more efficient and accurate when evaluating huge amounts of data [47].

Additionally, there are variations when it comes to the types of values that each tool's scoring technique handles. While some methods employ non-binary data, others use binary values for gene expression. Given this, it is clear that techniques that utilize continuous values are preferred over those that use discrete valued data. This is due to the possibility that, even though categorizing genes as active or inactive is more practical and effective, important information may be missed.

However, the tools used in this paper are not sufficient to demonstrate that Machine Learning techniques are superior to Statistics, because there are only 7 tools in total and their distribution between the two main approaches is not proportional, which is extremely likely to influence the results. In addition, it cannot help with sub-pathway targeting for medical treatment, since no significant sub-pathways were discovered.

Discussion

In this work, we implemented the scoring techniques for a number of Pathway Analysis tools that either rely on Machine Learning algorithms or statistical methods to identify similarities and differences among them. The findings of this research support the preference of Machine Learning over statistical methods, as well as the usage of non-binary gene expression data, but given the limitations observed the conclusions should be regarded cautiously.

Machine Learning is widely used in the field of Pathway Analysis due to the continuously growing amount of information in biology and the capability of Machine Learning algorithms to manage vast and complicated datasets [48]. Additionally, given that personalized therapies are the future of medicine, Machine Learning provides the tools needed to accomplish this purpose, because of its capacity to comprehend biological processes [49]. All of this, along with the exponential increase in computing power that is now available, has allowed Machine Learning to be considered as a promising technology for Pathway Analysis [50]. However, the application of Machine Learning in healthcare has some drawbacks that must be resolved, such as poor data quality and ethical dilemmas [51]. [52]

Multiple studies have been conducted to determine the type of gene expression data—binary and non-binary—that yields the most beneficial outcomes, while it should be mentioned that the majority of the methods examined in this study dealt with continuous data. Genomic data cannot simply be classified as on or off, because they are continuous, therefore they must be translated into discrete values. By translating the data it is likely to result in the loss of potentially important information, which is a very significant drawback [53], regardless of the fact that binary data require less processing time. As a result, current research attempts to combine them in an effort to optimize their benefits and minimize their disadvantages. Therefore, as we also discovered from the literature review of the tools analyzed in this paper [4], [24]–[41], there is a preference for non-binary valued gene expression data, or at least a tendency of transitioning from binary to continuous values. [54]–[56]

The abundance of genomic platforms available is a major issue that has been identified in numerous bioinformatics studies, while several studies have been conducted with the goal of comparing and determining the best platform or attempting to merge different databases [57]. Given that most datasets are obtained using various genomic platforms and processing techniques [58], it is highly likely that this limitation contributed to this study's inability to discover common significant sub-paths between two or more tools. This is due to variations in the platform design, synthesis and probe annotation [59], which lead to different outcomes. [60], [61]

Other scientific studies and experiments can be conducted taking into account both the positive and negative findings of this research. One approach could be applying additional tools in this study’s scripts to ensure that the findings are unbiased and, given the availability of databases, using datasets from various genomic platforms to overcome the problem of correlating genes across multiple databases and thus the inability of identifying common significant sub-paths. Another approach could be concentrating on non-binary gene expression data and employing mostly Machine Learning to analyze gene expression and gene regulatory networks in future works. As a consequence, by utilizing the capabilities of Machine Learning technology and enhancing existing pathway analysis techniques, the rates of accurate medical diagnoses could increase.

Conclusions

Pathway analysis is a promising technique for more precise medical diagnosis and determining the genetic basis of disorders. It makes use of current knowledge to better comprehend biological processes by combining both gene expression data and gene topology. However, a number of pathway analysis tools have been developed, each of which employs its own algorithm for pathway scoring that is either based on statistical techniques or machine learning algorithms. These scoring methodologies constituted the subject of this comparison study.

Overall, the outcomes indicate that Machine Learning-based methodologies perform better and faster than those based on statistical methods. When looking at each tool separately (see Table 2), HiPathia had the highest accuracy rate and the shortest execution time, while TAPPA performed the worst regarding the accuracy of the results and GraphiteWeb required the highest execution time. Furthermore, a significant proportion of the tools analyzed handle non-binary gene expression values, indicating their preference over binary data. Unfortunately, the research only analyzed a limited number of tools, and no significant common sub-paths between the tools were found, making it difficult to confidently accept the findings. However, future study may focus on the application of more datasets. More specifically, for an experiment utilizing four datasets, it is anticipated that it will take around 20 days to complete.

Based on the results, we can recommend Machine Learning algorithms as a better technique for pathway analysis, but it would be preferable to conduct a new study that incorporates proportionately additional tools from both approaches as well as apply this paper’s scripts to gene expression data from several genomic platforms. Also, choosing continuous data over discrete values is suggested. As a result, we will be able to identify the optimal method for future studies with greater confidence.

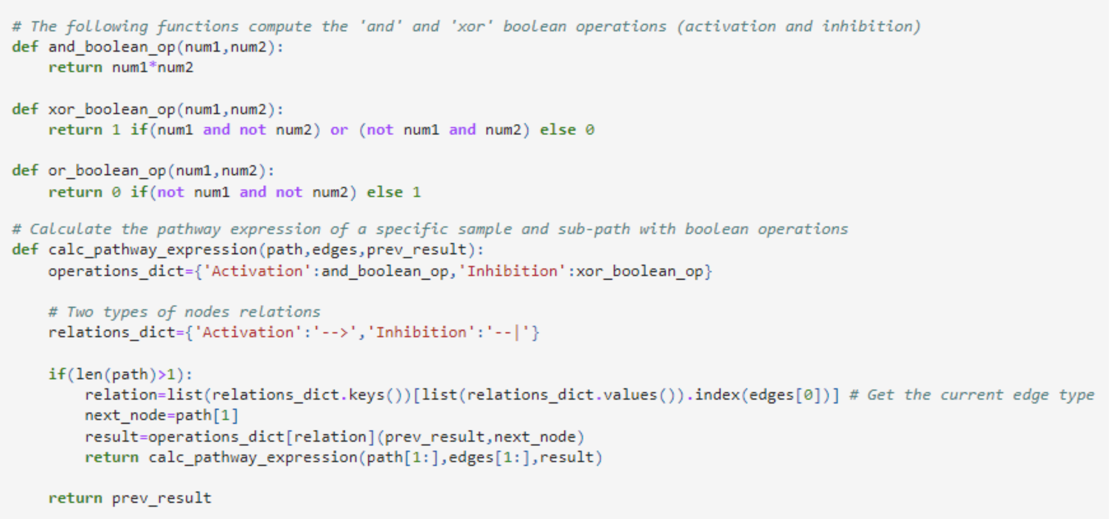
It is important to utilize every knowledge and technology provided to us, so that better and more accurate results are obtained every time. Consequently, as we get better at making sense of the data that is already available, more precise technologies will become available. These technologies will offer more precise medical diagnoses and, possibly, cures for diseases that we still don't fully understand.

Appendix

In this section we provide snips of code we developed to compute the pathway scores for a better understanding of each approach implemented.

Image 1. MinePath's scoring code

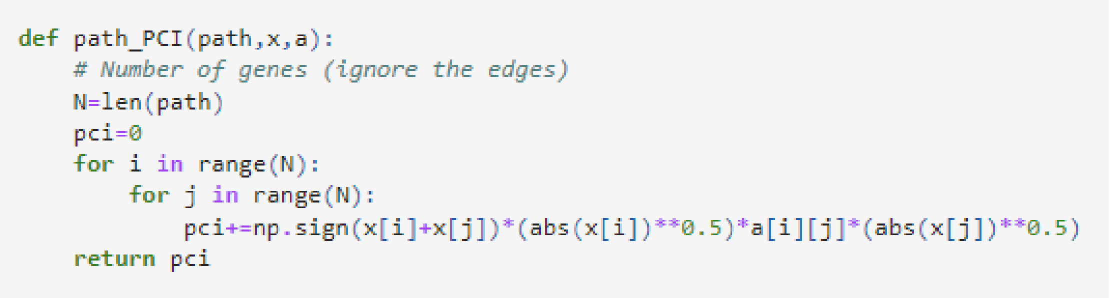
The code employed in this paper to compute the pathway score using MinePath's methodology is displayed in this image. MinePath handles binary expression values and employs Boolean operators according to the relationships among genes.



```python
# The following functions compute the 'and' and 'xor' boolean operations (activation and inhibition)
def and_boolean_op(num1,num2):
    return num1*num2

def xor_boolean_op(num1,num2):
    return 1 if(num1 and not num2) or (not num1 and num2) else 0

def or_boolean_op(num1,num2):
    return 0 if(not num1 and not num2) else 1

# Calculate the pathway expression of a specific sample and sub-path with boolean operations
def calc_pathway_expression(path,edges,prev_result):
    operations_dict={'Activation':and_boolean_op,'Inhibition':xor_boolean_op}

    # Two types of nodes relations
    relations_dict={'Activation':'-->','Inhibition':'--|'}

    if(len(path)>1):
        relation=list(relations_dict.keys())[list(relations_dict.values()).index(edges[0])] # Get the current edge type
        next_node=path[1]
        result=operations_dict[relation](prev_result,next_node)
        return calc_pathway_expression(path[1:],edges[1:],result)

    return prev_result
```

Image 2. TAPPA's scoring code

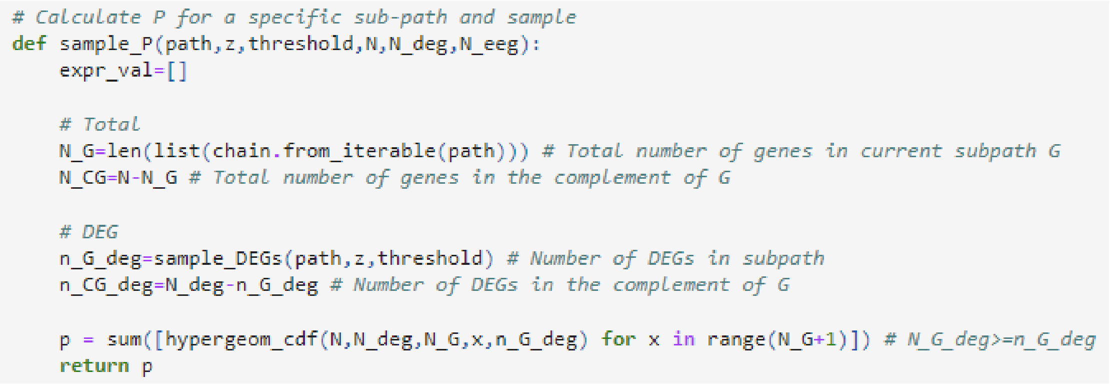
This image displays the code developed based on the description of TAPPA’s algorithm. The variable *x* denotes the standard log expression estimate of the genes for a specified sub-path and each sample, while the corresponding adjacency matrices are defined as *a*.



```python
def path_PCI(path,x,a):
    # Number of genes (ignore the edges)
    N=len(path)
    pci=0
    for i in range(N):
        for j in range(N):
            pci+=np.sign(x[i]+x[j])*(abs(x[i])**0.5)*a[i][j]*(abs(x[j])**0.5)
    return pci
```

Image 3. GraphiteWeb's scoring code

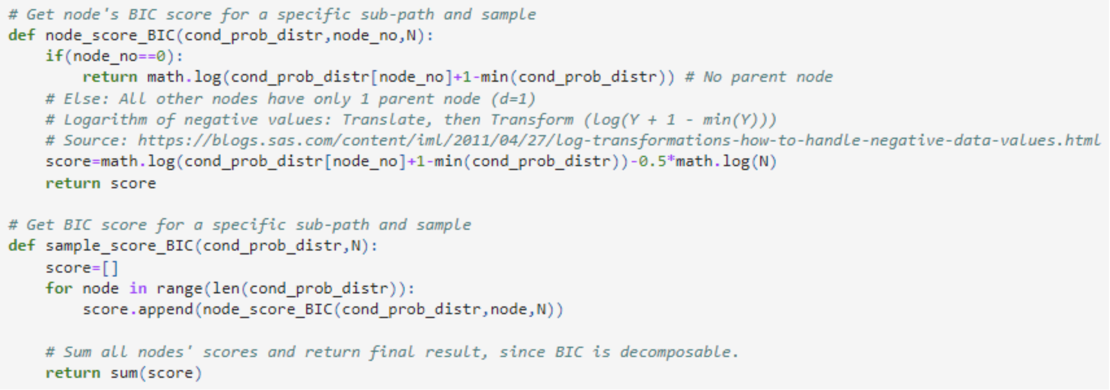
The algorithm for the sub-path score calculation according to GraphiteWeb’s approach requires the total number of genes screened *N*, the total number of differentially expressed genes *N_deg* and the total number of equally expressed genes *N_eeg*.



```python
# Calculate P for a specific sub-path and sample
def sample_P(path,z,threshold,N,N_deg,N_eeg):
    expr_val=[]

    # Total
    N_G=len(list(chain.from_iterable(path))) # Total number of genes in current subpath G
    N_CG=N-N_G # Total number of genes in the complement of G

    # DEG
    n_G_deg=sample_DEGs(path,z,threshold) # Number of DEGs in subpath
    n_CG_deg=N_deg-n_G_deg # Number of DEGs in the complement of G

    p = sum([hypergeom_cdf(N,N_deg,N_G,x,n_G_deg) for x in range(N_G+1)]) # N_G_deg>=n_G_deg
    return p
```

Image 4. TEAK's scoring code

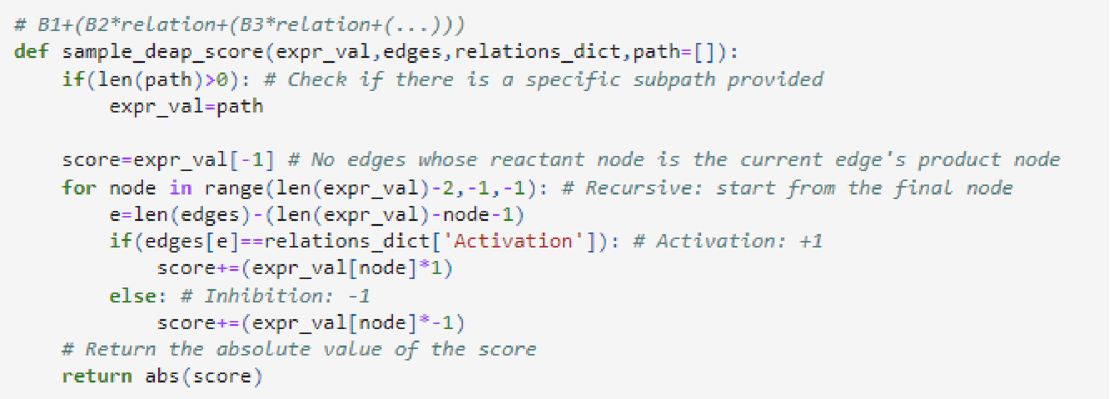
The image shows the algorithm produced for this study as explained by TEAK’s methodology, where *N* represents the number of samples in the gene expression data and *cond_prob_distr* stands for Conditional Probability Distribution.



```python
# Get node's BIC score for a specific sub-path and sample
def node_score_BIC(cond_prob_distr,node_no,N):
    if(node_no==0):
        return math.log(cond_prob_distr[node_no]+1-min(cond_prob_distr)) # No parent node
    # Else: All other nodes have only 1 parent node (d=1)
    # Logarithm of negative values: Translate, then Transform (log(Y + 1 - min(Y)))
    # Source: https://blogs.sas.com/content/iml/2011/04/27/log-transformations-how-to-handle-negative-data-values.html
    score=math.log(cond_prob_distr[node_no]+1-min(cond_prob_distr))-0.5*math.log(N)
    return score

# Get BIC score for a specific sub-path and sample
def sample_score_BIC(cond_prob_distr,N):
    score=[]
    for node in range(len(cond_prob_distr)):
        score.append(node_score_BIC(cond_prob_distr,node,N))

    # Sum all nodes' scores and return final result, since BIC is decomposable.
    return sum(score)
```

Image 5. DEAP's scoring code

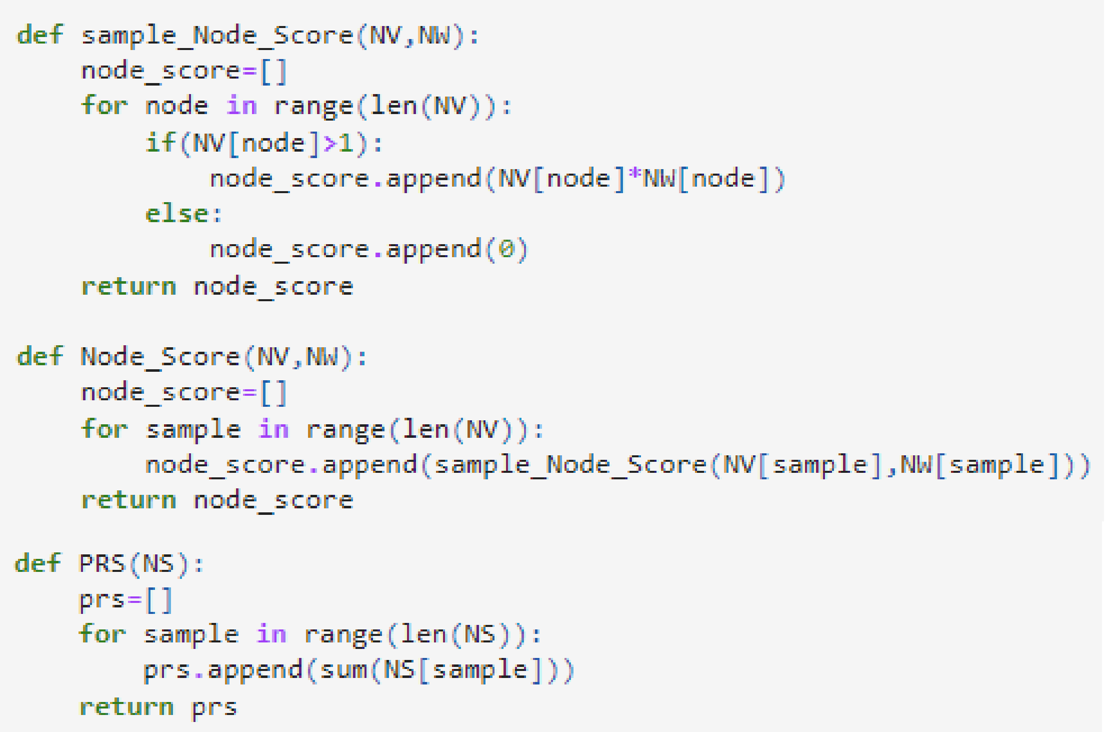
The script used to calculate the DEAP pathway score is shown below. The type of interaction between genes is taken into account via a recursive function.



```python
# B1+(B2*relation+(B3*relation+(...)))
def sample_deap_score(expr_val,edges,relations_dict,path=[]):
    if(len(path)>0): # Check if there is a specific subpath provided
        expr_val=path

    score=expr_val[-1] # No edges whose reactant node is the current edge's product node
    for node in range(len(expr_val)-2,-1,-1): # Recursive: start from the final node
        e=len(edges)-(len(expr_val)-node-1)
        if(edges[e]==relations_dict['Activation']): # Activation: +1
            score+=(expr_val[node]*1)
        else: # Inhibition: -1
            score+=(expr_val[node]*-1)
    # Return the absolute value of the score
    return abs(score)
```

Image 6. PRS's scoring code

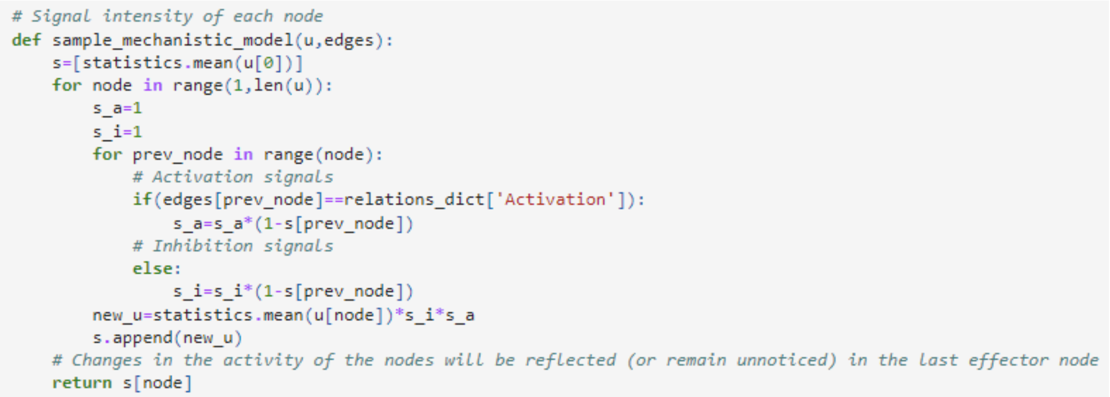
The following block of code for pathway score calculation was developed based on the guidelines provided by the associated paper of PRS. The pathway score is obtained by adding the scores of each node, which was determined using the respective node values and weights.



```python
def sample_Node_Score(NV,NW):
    node_score=[]
    for node in range(len(NV)):
        if(NV[node]>1):
            node_score.append(NV[node]*NW[node])
        else:
            node_score.append(0)
    return node_score

def Node_Score(NV,NW):
    node_score=[]
    for sample in range(len(NV)):
        node_score.append(sample_Node_Score(NV[sample],NW[sample]))
    return node_score

def PRS(NS):
    prs=[]
    for sample in range(len(NS)):
        prs.append(sum(NS[sample]))
    return prs
```

Image 7. HiPathia's scoring code

The HiPathia's pathway score computing mechanism is displayed in this image. The variable *u*, which represents the normalized gene expression values for a particular sub-path and sample, is essential for calculating the score.



```python
# Signal intensity of each node
def sample_mechanistic_model(u,edges):
    s=[statistics.mean(u[0])]
    for node in range(1,len(u)):
        s_a=1
        s_i=1
        for prev_node in range(node):
            # Activation signals
            if(edges[prev_node]==relations_dict['Activation']):
                s_a=s_a*(1-s[prev_node])
            # Inhibition signals
            else:
                s_i=s_i*(1-s[prev_node])
        new_u=statistics.mean(u[node])*s_i*s_a
        s.append(new_u)
    # Changes in the activity of the nodes will be reflected (or remain unnoticed) in the last effector node
    return s[node]
```

References

[1] “Human Genome Project,” *National Human Genome Research Institute*, Aug. 24, 2022.

[2] “Human Genome Project,” *Wikipedia*. Sep. 24, 2022.

[3] Y. Tu, G. Stolovitzky, and U. Klein, “Quantitative noise analysis for gene expression microarray experiments,” Oct. 2002.

[4] Lefteris Koumakis *et al.*, “MinePath: Mining for Phenotype Differential Subpaths in Molecular Pathways,” *Natl. Cent. Biotechnol. Inf.*, Nov. 2016.

[5] Ask A Scientist Staff, “What is Molecular Biology?,” Sep. 03, 2019. https://www.thermofisher.com/blog/ask-a-scientist/what-is-molecular-biology/

[6] “Molecular biology,” *Wikipedia*. Sep. 24, 2022.

[7] Sarah A. Bates, “Deoxyribonucleic Acid (DNA),” Sep. 2022.

[8] “DNA,” *what is biotechnology?* https://www.whatisbiotechnology.org/index.php/science/summary/dna/

[9] “What is a gene?,” *MedlinePlus*, Mar. 22, 2021.

[10] “What are proteins and what do they do?,” *MedlinePlus*, Mar. 26, 2021.

[11] The Editors of Encyclopaedia Britannica, “gene,” *Britannica*. Sep. 20, 2022.

[12] “Gene Expression,” *National Human Genome Research Institute*, Sep. 28, 2022.

[13] Lefteris Koumakis, “Computational methods for knowledge discovery from heterogeneous data sources: methodology and implementation on biological and molecular sources,” Technical University of Crete, School of Production Engineering and Management, 2014.

[14] Paolo Tieri and Filippo Castiglione, “Modeling Macrophage Differentiation and Cellular Dynamics,” *ScienceDirect*, 2021.

[15] Lesley T. MacNeil and Albertha J.M. Walhout, “Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression,” *PubMed Cent.*, May 2011.

[16] Miguel A. García-Campos, Jesús Espinal-Enríquez, and Enrique Hernández-Lemus, "Pathway Analysis: State of the Art," *Front. Physiol.*, Dec. 2015.

[17] Tuan-Minh Nguyen, Adib Shafi, Tin Nguyen, and Sorin Draghici, "Identifying significantly impacted pathways: a comprehensive review and assessment," *Genome Biol.*, Oct. 2019.

[18] "KEGG," *Wikipedia*. Feb. 11, 2022. [Online]. Available: https://el.wikipedia.org/wiki/KEGG

[19] "KEGG PATHWAY Database," *KEGG: Kyoto Encyclopedia of Genes and Genomes*. Mar. 24, 2022. [Online]. Available: https://www.genome.jp/kegg/pathway.html

[20] "BioCarta Pathways," *SciCrunch | Research Resource Resolver*. https://scicrunch.org/ADC/resolver/RRID:SCR_006917

[21] "Home - Reactome Pathway Database," *Reactome*. https://reactome.org/

[22] Ivana Ihnatova, Vlad Popovici, and Eva Budinska, "A critical comparison of topology-based pathway analysis methods," *PLOS ONE*, Jan. 2018.

[23] "A comprehensive review and assessment of pathway analysis methods," *ADVAITA*.

[24] Shouguo Gao and Xujing Wang, "TAPPA: topological analysis of pathway phenotype association," *Oxf. Acad.*, Sep. 2007.

[25] Adi Laurentiu Tarca *et al.*, "A novel signaling pathway impact analysis," *Natl. Cent. Biotechnol. Inf.*, Jan. 2009.

[26] Maria Sofia Massa, Monica Chiogna, and Chiara Romualdi, "Gene set analysis exploiting the topology of a pathway," *BMC*, Sep. 2010.

[27] Charles J Vaske *et al.*, "Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM," *Natl. Cent. Biotechnol. Inf.*, Jun. 2010.

[28] Ludwig Geistlinger, Gergely Csaba, Robert Küffner, Nicola Mulder, and Ralf Zimmer, "From sets to graphs: towards a realistic enrichment analysis of transcriptomic systems," *Natl. Cent. Biotechnol. Inf.*, Jul. 2011.

[29] "HotNet," *Raphael Lab*. http://compbio.cs.brown.edu/projects/hotnet/

[30] Fabio Vandin, Eli Upfal, and Benjamin J Raphael, “Algorithms for detecting significantly mutated pathways in cancer,” *Natl. Cent. Biotechnol. Inf.,* Mar. 2011.

[31] Maysson Al-Haj Ibrahim, Sabah Jassim, Michael Anthony Cawthorne, and Kenneth Langlands, “A Topology-Based Score for Pathway Enrichment,” *Natl. Cent. Biotechnol. Inf.,* Mar. 2012.

[32] Laurent Jacob, Pierre Neuvial, and Sandrine Dudoit, “DEGraph: differential expression testing for gene networks,” *Bioconductor,* Oct. 2014.

[33] Laurent Jacob, Pierre Neuvial, and Sandrine Dudoit, “More Power via Graph-Structured Tests for Differential Expression of Gene Networks,” *Proj. Euclid,* Jun. 2012.

[34] Thair Judeh, Cole Johnson, Anuj Kumar, and Dongxiao Zhu, “TEAK: Topology Enrichment Analysis frameworK for detecting activated biological subpathways,” *Natl. Cent. Biotechnol. Inf.,* Feb. 2013.

[35] Patricia Sebastián-León, José Carbonell, Francisco Salavert, Rubén Sanchez, Ignacio Medina, and Joaquín Dopazo, “Inferring the functional effect of gene expression changes in signaling pathways,” *Oxf. Acad.,* Jun. 2013.

[36] Winston A. Haynes, Roger Higdon, Larissa Stanberry, Dwayne Collins, and Eugene Kolker, “Differential Expression Analysis for Pathways,” *Natl. Cent. Biotechnol. Inf.,* Mar. 2013.

[37] Gabriele Sales, Enrica Calura, Paolo Martini, and Chiara Romualdi, “Graphite Web: web tool for gene set analysis exploiting pathway topology,” *Natl. Cent. Biotechnol. Inf.,* May 2013.

[38] S Nam *et al.*, “PATHOME: an algorithm for accurately detecting differentially expressed subpathways,” *Natl. Cent. Biotechnol. Inf.,* Oct. 2014.

[39] Xianbin Li, Liangzhong Shen, Xuequn Shang, and Wenbin Liu, “Subpathway Analysis based on Signaling-Pathway Impact Analysis of Signaling Pathway,” *PLOS ONE,* Jul. 2015.

[40] Kinza Rian *et al.*, “Genome-scale mechanistic modeling of signaling pathways made easy: A bioconductor/cytoscape/web server framework for the analysis of omic data,” *ScienceDirect,* May 2021.

[41] Marta R Hidalgo, Cankut Cubuk, Alicia Amadoz, Francisco Salavert, José Carbonell-Caballero, and Joaquin Dopazo, “High throughput estimation of functional cell activities reveals disease mechanisms and predicts relevant clinical outcomes,” *Natl. Cent. Biotechnol. Inf.*, Jan. 2017.

[42] IBM Cloud Education, “Machine Learning,” *IBM*, Jul. 15, 2020. https://www.ibm.com/cloud/learn/machine-learning

[43] “Decision Trees for Classification: A Machine Learning Algorithm,” *Xoriant*. https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm

[44] “Accuracy (error rate),” *DeepAI*. https://deepai.org/machine-learning-glossary-and-terms/accuracy-error-rate

[45] “MACHINE LEARNING VS. STATISTICS,” *Univ. Del.*, Jul. 2021.

[46] Michele Bennett, Karin Hayes, Ewa J. Kleczyk, and Rajesh Mehta, “Similarities and Differences between Machine Learning and Traditional Advanced Statistical Modeling in Healthcare Analytics,” Jan. 2022.

[47] Zohar Barnett-Itzhaki *et al.*, “Machine learning vs. classic statistics for the prediction of IVF outcomes,” Aug. 2020.

[48] Hayat Ali Shah, Juan Liu, Zhihui Yang, and Jing Feng, “Review of Machine Learning Methods for the Prediction and Reconstruction of Metabolic Pathways,” Jun. 2021.

[49] Kexin Huang, Cao Xiao, Lucas M. Glass, Cathy W. Critchlow, Greg Gibson, and Jimeng Sun, “Machine learning applications for therapeutic tasks with genomic data,” vol. 2, no. 10, Oct. 2021.

[50] Miroslava Cuperlovic-Culf, “Machine Learning Methods for Analysis of Metabolic Data and Metabolic Pathway Modeling,” Jan. 2018.

[51] Junjie Peng, Elizabeth C. Jury, Pierre Dönnes, and Coziana Ciurtin, “Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges,” Sep. 2021.

[52] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L. Beam, Irene Y. Chen, and Rajesh Ranganath, “A Review of Challenges and Opportunities in Machine Learning for Health,” May 2020.

[53] Koji Kyoda, Mineo Morohashi, Shuichi Onami, and Hiroaki Kitano, “A gene network inference method from continuous-value gene expression data of wild-type and mutants,” Feb. 2000.

[54] Andrew McDavid *et al.*, “Data exploration, quality control and testing in single-cell qPCR-based gene expression experiments,” Feb. 2013.

[55] Yoram Ben-Shaul, Hagai Bergman, and Hermona Soreq, “Identifying subtle interrelated changes in functional gene categories using continuous measures of gene expression,” Jan. 2005.

[56] Lorena Postiglione, Marco Santorelli, Barbara Tumaini, and Diego di Bernardo, “From a discrete to continuous actuation for improved real-time control of gene expression in mammalian cells,” 2016.

[57] Jason Rudy and Faramarz Valafar, “Empirical comparison of cross-platform normalization methods for gene expression data,” Dec. 2011.

[58] Andrey A. Shabalin, Hakon Tjelmeland, Cheng Fan, Charles M. Perou, and Andrew B. Nobel, “Merging two gene-expression studies via cross-platform normalization,” 2008.

[59] Anna V. Tinker, Alex Boussioutas, and David D.L. Bowtell, “The challenges of gene expression microarrays for the study of human cancer,” *ScienceDirect*, vol. 9, no. 5, May 2006.

[60] Tracy Tucker *et al.*, “Comparison of genome-wide array genomic hybridization platforms for the detection of copy number variants in idiopathic mental retardation”.

[61] Paul K. Tan *et al.*, “Evaluation of gene expression measurements from commercial microarray platforms”.