Metamarker : Differential Correlation Network Methodology and Software for Metabolomic Data Analysis

© Arshad Arafat

Supervisors: Dr.Ting Hu & Dr. Yuanzhu Chen

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Science

Department Of Computer Science

Memorial University Of Newfoundland

Canada

March, 2021

Abstract

Biomarkers are the substances with quantitative properties present within organisms indicating disease progression. Metabolomics is a newer approach towards understanding the human body following the footsteps of other "omics" techniques (genomics, proteomics, transcriptomics). Metabolomics refers to the scientific study of low molecular intracellular elements called metabolites. With the advancement of technology, it is now easier to extract different sets of metabolites from various forms of biological samples such as cells, tissues, bio-fluids, etc.

Metabolomic data analysis is a complex workflow. It requires sophisticated data processing and statistical analysis. Various tools have been developed, such as data cleaning and preprocessing tools, modeling tools, validation/result visualizations, and many more. Most of these software tools are developed for comprehensive studies rather than precisely focusing on metabolomic biomarker discovery. As a result, their capacity, in most cases, is limited. The modeling techniques commonly used in these tools are also not adequate. Many of these software tools provide basic analysis methods rather than more advanced machine learning techniques. The high throughput metabolomic datasets require compound analysis techniques. This thesis designed and developed a software tool that encompasses the general metabolomic biomarker research workflow. Our software platform is equipped

with many basic to advanced analysis techniques, interactive visualizations, delicate result analysis, and comparison modules (The first version release can be found at, http://18.189.6.35:8000/). Our software is designed so that users do not have to switch in between different tools during the study since the platform provides necessary features that are commonly used throughout the workflow. Some of the software's significant features are outlier handling of the uploaded datasets, analyzing the dataset with principal component analysis or partial least square discriminant analysis, and comparing different models. The software makes the study process fast and convenient. We employed a differential correlation network analysis model for the biomarker discovery studies, which is advantageous in finding key metabolites that influence diseases through interaction.

Keywords: Metabolomics, Machine Learning, Data Visualization, Differential Correlation Network, Biomarker Discovery

Acknowledgement

I wish to thank my supervisor Dr. Ting Hu who has always been very generous with her expertise and precious time. Thank you for your countless hours of reflecting, reading, encouraging and above all for your patience throughout the entire process. Thank you Dr. Yuanzhu Chen who has always advised me and served me as a co supervisor.

I would like to acknowledge and thank my university and the school of graduate studies for allowing me to conduct my research and providing any assistance requested. I am grateful to the members and staffs in the department of computer science for their continued support.

Contents

List of Figures

List of Tables

Chapter 1 : Introduction

Systems biology, through its augmentation, has provided us diverse perspectives to understand the living organisms [38]. The discovery of DNA, Genes, and the advancement of sequencing techniques widened our capacity. In today's post-genomics era, researchers are seeking additional details and exposure to the biological system. This desire led to the development of proteomics and transcriptomics. Finally the same quest led the development of metabolomics [27, 32, 60, 61, 24].

The medical field has been remarkably benefited from the breakthrough in metabolomics. Applications of metabolomics are notable in major medical science areas such as disease/drug progression or exploration of bio-markers [40, 56]. Useful sets of biomarkers are pivotal for early disease detection and treatments [45, 65]. So far, metabolomics has been applied to identify biomarkers for complex health conditions like cancers, arthritis, psychological disorders, and many more [58]. Historically, visible bodily traits have been used as disease biomarkers. With the advent of technology and medical equipment, biomarker research progressed into the molecular horizon [62]. In this thesis, our boundary of focus will be on metabolomic biomarker analysis. To be more specific, we will be building a software platform for metabolomic biomarker analysis. Biomarker analysis is a complex process. The domain knowledge is not enough to headway through the field. Instead, data-driven

statistical modeling techniques and experiments are required [76, 44]. At present, we have an abundance of collected metabolomic data. More computing and statistical modeling techniques are needed at this point to be efficient with processing these data. Lately, computer scientists and statisticians have also started contributing to metabolomics by designing new tools and techniques [64].

Researchers use different tools for different tasks throughout biomarker discovery studies [64]. Apart from the promising possibility, we lack one single tool solely dedicated towards metabolomic biomarker research even today [50, 49]. Assessing multiple modern software tools/programs, we have identified the following set of hindrances. 1) not all the tools are easy to access. Most of them are standalone, dependent on specific environments/operating systems. 2) these tools require substantial computational resources and may not perform well with less powerful computers. 3) usability has been a big issue. Some tools are daunting and ambiguous to use as they require a lot of configurations to set up. 4) Not all the tools have varied and advanced visualizations or reports. Most of them come with unclear plots which are barely interactive. 5) Majority of these tools are available with classic trivial modeling techniques, whereas today, we need more advanced, powerful, and efficient methods to look into the dataset.

This thesis is an attempt to enrich the field. We have developed a web server that removes a great deal of burden from the potential users (The first version release is hosted temporarily at http://18.189.6.35:8000/). From data pre-processing, data analysis to result validation/report generation, most unwanted redundancy can be avoided with our new software tool. The tool is equipped with efficient and advanced algorithms. The visualization techniques implemented in the software provide more insights. We have also focused on a model that works projecting the differential correlation of the metabolites into a network where topological analysis is performed to find central/key metabolites responsible for the disease condition.

Metabolomic biomarker research requires portable, resourceful, easy to use services. Therefore we have designed and developed this web-based system that can be accessed independently to high-performing computing resources. With easy navigation, modular services, descriptive dialogues, state-of-the-art interactive visualizations, anyone can use it without any detailed knowledge of the domain. The platform allows users to store and save their data/results and access them with secure and private protocols. Mighty and out-of-the-box data analysis models that we have implemented serve the users efficiently. The differential correlation analysis model filters the significant metabolites by converting them into a network and running topological centrality analysis.

In Chapter 2, we have a thorough review of the literature. In Chapter 3, we discuss different analysis models, methods, techniques, etc. Chapter 4 demonstrates our system's overall design, software implementation, technical details, features, etc. Then in Chapter 5, we see the results, reporting, and validation processes. In chapter 6, we make a substantial discussion of our overall thesis project, findings, and limitations. Finally, we conclude the thesis report in chapter 7.

Chapter 2 : Background

The term Metabolomics has been coined recently. Nonetheless, the practice of the concept is age-old [27, 51]. Metabolites refer to the smallest viable molecules (mostly weighing 1500 Dalton's or even lesser) present inside the cells resulting from different metabolic processes [13]. Studying and profiling them signifies the cell's state, resistance concerning disease, or progress respecting drugs. Recent advancements in technologies, especially in chromatography techniques, led us to identify most of them present in subject samples such as urine, saliva, tissue, blood, and many more [58]. Some details about metabolites, study procedures, data generations, etc. remain greatly apprehended in the conscious works by Fell [19]. Hiroaki Kitano [39] quoted that the goal of computational system biology can be divided into two subgroups. The first one is the data mining or knowledge discovery from experimental datasets, and the other one is considered the simulation attempts to predict the dynamics of the systems. Cuperlovic [12] in their review article described how metabolomics is also designed to achieve these two similar goals.

The earliest reference to the idea could be traced back to 2000 BC - 1500 BC. The Greeks, the Egyptians, and Chinese physicians had known about the concepts [58]. They had learned that urine and its taste or color could be traced to diseases. They used biosensor ants to examine urine samples for diabetes.

Egyptians also knew about urine and its traceable properties. They identified frequent urination and Polyuria to be correlated. Arabs had learned that urine changes its color and smell during various illnesses. During eighteen and nineteenth centuries, metabolomics received modern attention. Mass spectronomy was developed to profile body fluids [14]. Researchers started studying enzymatic reactions caused by metabolic processes. Twenty-first century endeavor can be considered when Horning [28] adopted metabolic analysis technique in his works in 1971. In the year 1998 Oliver and his colleagues first used the term metabolomics keeping its association with its predecessor's genomics, proteomics and transcriptomics [52]. Soon after that, researchers used the terms metabolomics, metabonomics, metabolomic-profiling and many more in other works [10].

Biomarkers, the medical term shortly used for biological markers, refer to the symptoms with quantitative and predictive properties [45]. They can be characterized and evaluated with disease progression, biological, genomic, metabolic processes, and pharmacological response. For centuries researchers, epidemiologists and physicians have been looking for biomarkers in a variety of health conditions. Last few decades, biomarker research programs boomed due to economic, epidemiological, and technical reasons [44]. Metabolites are downstream representations from the genome, transcriptome, and proteome [27]. Analyzing metabolites can reveal significant traits in the upper

tiers. Many diseases tend to be showing changes in metabolic pathways long before they show potential phenotypic symptoms. Therefore metabolomic bio-markers have potential in early disease prediction [61].

Before the in silico modeling and applications, biological samples are collected in the form of biofluids. Samples need to be analyzed thoroughly with sample separation techniques. Existing metabolomic studies mostly use nuclear magnetic resonance (NMR), spectroscopy[16, 33, 43], mass spectrometry (MS) and multivariate chemometrics [34]. NMR is popular because it is reproducible. Though the NMR generated datasets require data preprocessing, it is well suited in the clinical setup. Spectrometric techniques such as mass spectrometry (MS) or gas spectrometry (GS) are also common analytic techniques that separate the molecules using motion. Mass spectrometry has the advantage of a parallel application, e.g. gas chromatography with mass spectrometry or liquid chromatography with mass spectrometry [15]. The parallel application results in added performance.

Metabolites are the results of chemical reactions. They are susceptible to perturbations making the analysis harder. It is common to see a combination of multiple techniques for added accuracy [15]. Enriched computational power and abundance of high throughput datasets enabled us to apply statistical techniques varying from univariate statistical testing to multivariate re-

gression methods [42]. Existing metabolomic data analysis techniques can be categorized into three different groups. First, data overview tools e.g. principal component analysis (PCA) or clustering methods. The second category includes the linear and nonlinear classification models. Some linear techniques can be partial least square (PLS) or orthogonal partial least square (OPLS). Examples of nonlinear techniques are neural networks or support vector machines. The last category is validation techniques.

In general, researchers studying or modeling metabolomic datasets never followed any particular guidelines. Everyone tried modeling the data set to the computational model they were interested in. Charles E. Determan, in his work [36] reviewed all the machine learning models and statistical analysis techniques that are mostly used on metabolomic data sets. He presented a guideline relating to research goals and models (Figure 1).

For the last couple of decades, exponential growth can be seen in metabolomics. Some good platforms, software, and tools are now crafted, focusing on different metabolomics areas and goals. Some of them are open-sourced tools or packages, and some others are commercial. Spicer et al., in their paper [64] reviewed some major tools used in metabolomics. They categorized the tools based on functionalities like preprocessing, annotation, and many more. Popular Softwares like XCMS, cRMN, EigenMS, Metabomxtr,

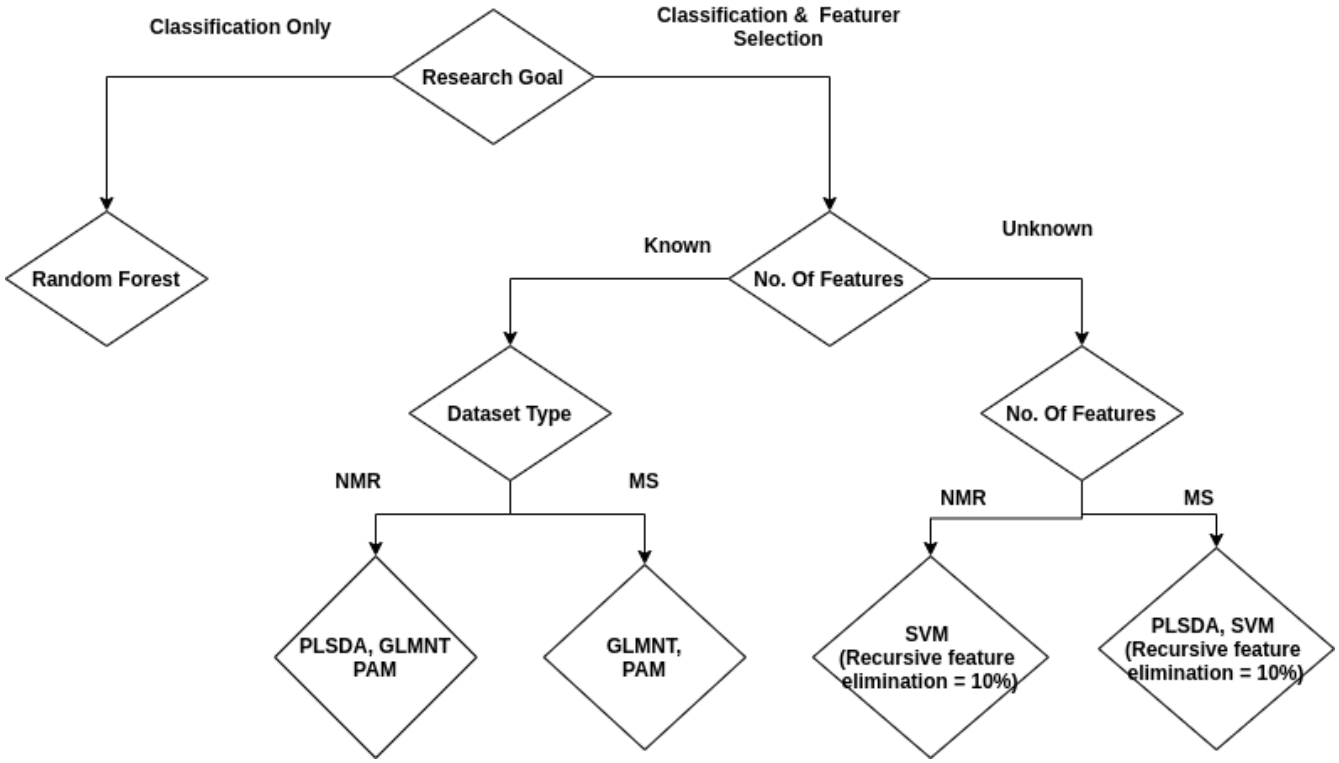


Figure 1: Determan's general guidelines of algorithms to be applied based on study purposes.If goal is classification only then popular choice is applying random forest algorithm. Otherwise algorithm is chosen based feature properties and dataset type.

MetabR etc. are used to preprocess LCMS data. Ionwinze, MetabolAnalyze, metabolomics, MetaboLyzer, muma etc are the tools to analyze preprocessed MS data [3, 4, 8, 64, 67]. From the usability perspective, we have found three different tools that serve researchers in different processes. 1) command-line interface, 2) graphical user interface, and 3) script packages or library. Table 1 highlights some major software tools used in the studies for similar tasks.

	MetaboAnalyst	MetaboNexus	Meta X
Release Year	2009	2014	2015
Programmed with	R, Java	R	R,Java
Platform Independent	Yes	No(Windows only)	Yes
Outlier Handling	No	No	Yes
Normalization Techniques	quantile normalization Or Internal Standard	quantile normalization Or Internal Standard	Sum, PQN, VSN, QC-RSC, Normalization ComBat, SVR, quantiles
Modeling Algorithms used	Internal standard	PCA,Clustering,PLS-DA, ROC-analysis	PCA,Clustering, PLS-DA, ROC-analysis
Report Geneeration	Yes	No	Yes

Table 1: Existing tools/softwares comparison

In biomarker discovery studies, model fitness can not be deliberately displayed by only accuracy or precision rate. More sophisticated measurement of proof is needed. Generally, biomarker discovery studies go through three different phases of validations [6]. The first phase is known as the discovery phase, where a set of signature biomarkers is generated from the training set, which is tested on the test set. Next to the discovery phase comes the pre-validation phase or cross-validation phase. Both the discovery phase and cross-validation phase together provide a better estimate and confidence over the bio-markers. The final step of validation is considered the significant confirmation in biomarker studies. This stage is clinically validated. In this phase, biomarkers are monitored and studied on healthy and affected subjects for a final assessment of the resulted bio markers' success or failure. In Figure 2, we have summarized a metabolomic biomarker study's general workflow. It can be seen from the figure that a study starts clinically with data generation. Then they are analyzed with computational and statistical modelings, finally, the generated results are verified through clinical trials.

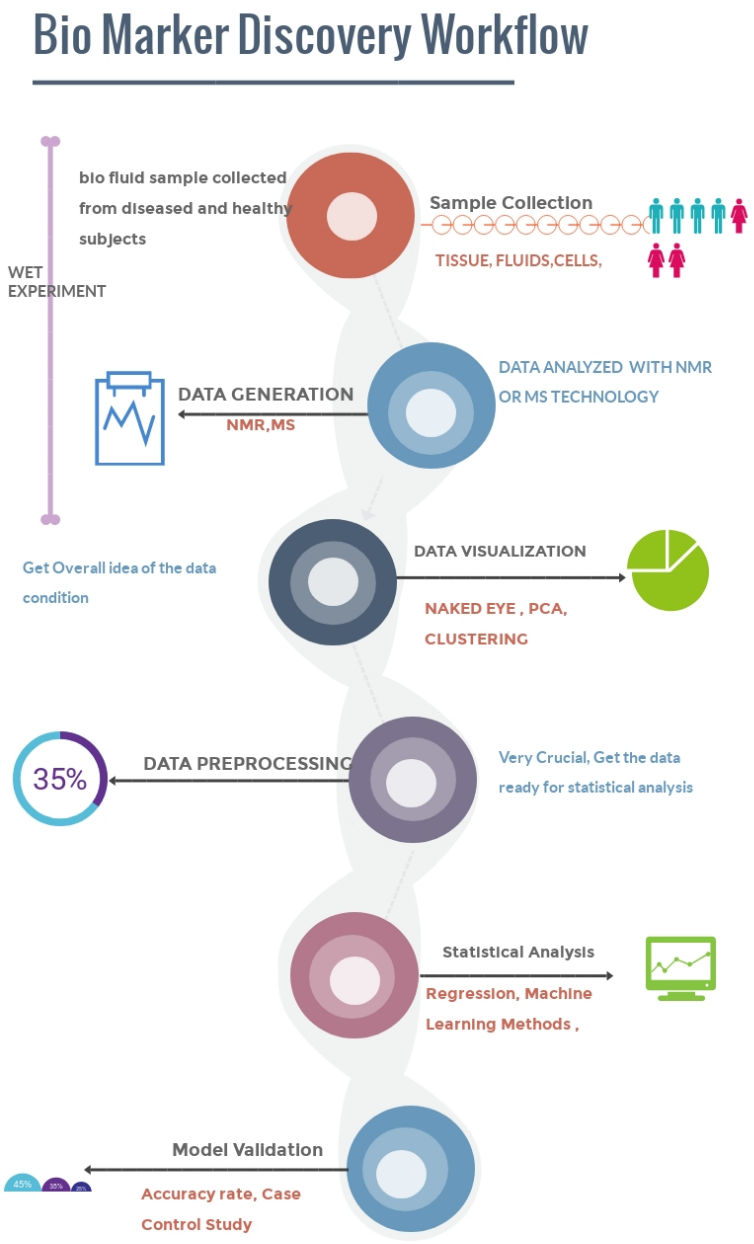


Figure 2: Metabolomics bio-marker discovery workflow. Initially samples are clinically assessed and dataset is generated. Dataset then gets curated and analysed. Finally data mining results in biomarker set which is validated.

Chapter 3 : Methodology

A good quality dataset is rudimentary to find significant biomarkers. In general, before applying statistical modeling, datasets are preprocessed with different techniques. Various preprocessing methods frequently applied on metabolomic datasets are reviewed in the article written by Jun Yang et al. [74] . Also, Table 2 lists some efficient preprocessing techniques that we have adopted in our software.

Name	Method Details
Missing Value correction	This method is by default applied to prto all preprocessing tasks. It ensures that if the dataset consists any feature - variable with more than 80% missing or null value. then that feature variable is discarded.
Mean Centering and Variance Scaling	This method is very popular among data scientist. It ensures proper variance among the feature variables. This method scales the value by substracting the mean value of the feature from every value. So basically $X_{new} = X_{old} - X^{mean}$ where X denotes the feature variable.
Univariate Scaling	This scaling method generates new feature values by dividing the old feature value with its standard deviation. Thus $X_{new} = X_{old} - S^{old}$ where X denotes the feature and S denotes the standard deviation of Xth variable.
Pareto Scaling	Pareto scaling is little improvement of Univariate scaling. Here in stead of standard deviation we use square root of standard deviation. Thus $X_{new} = X_{old}/S^{old}$ where X denotes the feature and S denotes the standard deviation of Xth variable.
Ln Transformation	Ln transformation scales the feature value to its ln scale so $X_{new} = lnX_{old}$
Vast Scaling	Designed by Keaun et all in 2003. It works similar like Univariate scaling. It multiplies the old feature by the ratio of mean and standard deviation. So basically $X_{new} = (X_{mean}/S_{old}) * X_{old}$
Range Scaling	It substracts the feature mean from the feature and then divides the feature with variance. Thus $X_{new} = (X_{old} - X_{mean})/X_{variance}$
Level Scaling	Just like Range scaling however in stead of variance it uses mean as denominator also. So $X_{new} = (X_{old} - X_{mean})/X_{mean}$

Table 2: List of available data preprocessing techniques

Mostly, the analysis techniques used in metabolomics are adopted from other Omics study fields. Especially from Transcriptomics. They vary from univariate to multivariate analysis techniques. Some of the classic methods used

in earlier days involved the parameter by parameter studies of the metabolites such as T-test or analysis of variance (ANOVA) [41]. Multivariate techniques include some observatory, supervised or unsupervised methodologies like support vector machine, K-Means clustering, principal component analysis (PCA), partial least squares projection to latent structures (PLS), Orthogonal partial least squares projection to latent structures (OPLS) and many more [41, 71].

3.1 Principal component analysis (PCA)

Principal component analysis (PCA) [69] is the most widely applied multivariate analysis technique used in Metabolomics. PCA uses the concept of lower-dimensional principal components explaining the variance of its higher dimensional original dataset. PCA makes a linear transformation to the dataset capturing maximum variance and minimum dimensions. PCA converts the data matrix into two factorized matrices. The first one is a score matrix that contains the new positioning of the data points. The second one is a loading matrix that has weights for the original variables. If we have a data matrix X with $x_1, x_2, x_3, ...x_n$ set of vectors. Where x_i denotes the i^{th} observation/metabolite in the dataset. Then PCA seeks a transformed matrix X_a where,

$$X_a = \sum_{i=1}^{n} a_i x_i \tag{1}$$

Here $a = a_1, a_2, a_3....a_n$ is a vector of constants. The transformed matrix and its covariance can also be generated by $a'Sa$ where S is the covariance matrix. On the other hand, another matrix factorization technique named singular value decomposition (SVD) can also generate principal components. SVD represents dataset $X = UEW^T$ where E is a rectangular diagonal matrix, U is an n-by-n matrix, W is a p-by-p matrix. (p is the number of instances in the dataset).

PCA is effective both as data reduction and visualization model [46]. It is also highly regarded as a preprocessing step before applying clustering algorithms [17]. Generally, PCA is used as an introductory analysis technique to explore the dataset. It is considered as an exploratory multivariate data analysis model by many researchers [18, 54]. Many researchers consider PCA as a starting point to form the overview of a hypothesis. Some notable PCA applications in metabolomics are urine or serum metabolites studies in kidney cancer or Parkinson's disease, gut microbiome studies, cancer-related pathways analysis studies, and many others.

3.2 Partial least squares discriminant analysis (PLS-DA)

PCA performs well on observatory and unbiased dimension reduction problems. However, at times its capacity can be limited. PCA depends on the

data points' variations both within the similar and between the other group or classes. Hence supervised form of analysis technique PLS became popular as in near to hand modeling technique [68, 70] . PLS serves multiple purposes, such as data observation, classification, prediction, regression, visualizations, etc. PLS's underlying principle is to transform or project dataset into lower-dimensional latent spaces preserving the dataset's overall covariance structure.

Datasets using PLS can be represented by two metrices(X and Y). X being an $M * N$ size matrix of dependent data points (M is the number of variables while N is the number of data points). Y is a $P * N$ sized matrix denoting the group of the data points (P is the number of dependent variables while N is the number of data points). PLS projects this data matrix into a k dimensional matrix T. T is basically the rotated matrix of X and Y with $t_1, t_2, t_3...t_k$ data points. T is represented by the following equation -

$$T = XW \tag{2}$$

where W is a matrix representing weights of the X variables. Once T is generated, it is then possible to design a prediction or classification model Y^*. It can be done by multiplying T with another weight matrix C'. C' is known as Y weight matrix describing Y matrix on the rotated latent space.

so basically -

$$Y^* = TC' \tag{3}$$

Since T is represented by XW so we can reform it as -

$$Y^* = XWC' \tag{4}$$

The formula above can be again converted into a linear regression model by simply replacing WC' with B_{PLS}. So finally -

$$Y^* = TB_{PLS} \tag{5}$$

where $B_0, B_1, B_2.....B_M$ represents linear coefficients for X. at any given time new prediction y^* can be done by -

$$y^* = B_0 + B_1x_1 + B_2x_2 + B_3x_3 + ... + B_Mx_M \tag{6}$$

PLS can also serve to visualize the datasets. Once X is projected on latent space we can then visualize T on a scatter plot to understand closely the cluster of the variables. We can achieve that goal by looking into weight vectors of the variables represented by W. The score matrix T also allows us to compare scores among the projections. Such as if t_1 and t_2 are two projections in T then they are represented as following equation and can be

portrayed on any graphs to compare them.

$$t_1 = w_{0,1} + w_{1,1}\, x_1 + w_{2,1} x_2 + ... + w_{m,1} x_M \tag{7}$$

$$t_2 = w_{0,2} + w_{1,2}\, x_1 + w_{2,2}\, x_2 + ... + w_{m,2} x_M \tag{8}$$

Apart from classification, regression, visualization, dimension reduction, PLS also serves another popular way. PLS uses the concept of variable influence on projection, directly measuring the contribution of data features. We can also achieve the same goal by looking into the regression coefficient from previous equations as well. However, VIP scores are explicitly designed in PLS to verify the variable contribution. VIP score of variables can be calculated by the following equation -

$$VIP = \sqrt{M * (\sum_{i=1}^{k} W_i^2 * SSY_i)/SSY_{cum})} \tag{9}$$

Here w_i^2 is the sum of the square of the corresponding Y variable. SSY_i represents the normalized sum of squares of Y on latent space, and SSY_{cum} represents the cumulative sum of squares of Y on latent space

3.3 Differential correlation network analysis

PCA, PLS, and other individual testing techniques are commonly adopted in biological studies. Enormous research and application have embraced these methods to extrapolate the interrelated elements associated with disease conditions. There have been recent developments in tools and techniques studying biological systems. Some current perspective includes the studies of pairwise interaction of the elements, e.g: protein-protein, gene-gene or metabolite-metabolite. Another recently developed significant study model in biomarker research is the study of differential correlations among the factors [22, 31]. Instead of looking for the absolute property behind any biological condition, the fundamental idea is that it is easier to map them since the metabolites tend to show deferentially correlated characteristics.

Network mapping is essential when it comes to studying differential correlation or interaction among the metabolites. Metabolites are results of cellular processes, and they can be represented in the form of a network that we call the pathway. Thus mapping differentially correlated pairwise interactions among the metabolites, we highlight the significant metabolites and figure out the cellular processes producing them.

Differential correlation studies of the metabolites, mapping them into networks and analyzing the topological properties can be a robust biomarker

detection method. A similar study exhibited tremendous outcome finding responsible metabolite bio-markers in Osteoarthritis study affecting knee joint [29]. It can be a potential model studying metabolites for other bio-marker studies as well. Thus we have incorporated the differential correlation analysis model in our thesis and broadened its capacity. The differential correlation methodology and topological analysis is the major concentration of this thesis. We have extended our analysis model by giving it a platform combining easy-to-use and configurable panels, handy network mapping, advanced 2d, and 3d interactive visualization and multiple topological analysis techniques, automatic and manual filtering techniques, and many more.

This model works in two phases. First phase includes the study of case control population data. The initial dataset is divided into two sets: the case set and the control set. Pairwise correlation using Pearson correlation coefficient r in both case and control is then measured. Corresponding correlation r_{case} and $r_{control}$ are used to compute the change of correlation difference r_{diff} among the two set of data points. If i and j represents two metabolites and $r_{case}(i,j)$, $r_{control}(i,j)$ are the corresponding pair wise Pearson correlation coefficient then $r_{diff}(i,j)$ is calculated using the normalized difference of Fisher's z transformation among $r_{case}(i,j)$ and $r_{control}(i,j)$,

$$r_{diff}(i,j) = (\sqrt{(n_{case} - 3)/2} * z_{case}(i,j) - (\sqrt{(n_{control} - 3)/2} * z_{control}(i,j)) \tag{10}$$

$$z_{case}(i,j) = 1/2ln((1 + r_{case}(i,j))(1 - r_{case}(i,j))) \tag{11}$$

$$z_{control}(i,j) = 1/2ln((1 + r_{control}(i,j))(1 - r_{control}(i,j))) \tag{12}$$

n_{case} and $n_{control}$ here defines the number of data points among the case and control population. z_{case} and $z_{control}$ on the other hand defines Fisher's z -transformation of correlation coefficient r and can be computed by equation 11 and 12. This model of analysis involves an extraordinary approach to eradicate the data bias to get better results. This model incorporated a 1000-fold permutation test. On each permutation random data points among the case and control are shuffled. Then r_{diff} is calculated again, and this process is continued 1000 times. The entire iteration ensures a null distribution among the data points.

The second phase of the analysis model incorporates a network mapping

of the metabolite pairs. The concept of p-value to signify significant pairwise differential correlation among the metabolites is embraced. p-value can be represented as a notifier of the metabolite pair importance. Usually, a p-value cut-off is set to ignore the metabolite pairs that resemble low priority. This is done to make the network more robust and easy to visualize. This also handles the computational burden when the topological analysis is done on the network represented by the metabolites. The differential correlation model incorporates the concept of positive and negatively correlated metabolites denoting the metabolite concentrations are significantly correlated on cases than control and vice versa. In the network graph resulted from the correlation matrix, the vertices denote the metabolites, and the edges among them represent the p-value signifying their correlation. Significant metabolites are likely to be more central and connected to most other metabolites. That is why once the network is constructed, a topological analysis is done, including centrality probing among the metabolites.

In graph theory, researches are conducted on graphs studying the centrality of the vertices [53]. Centrality works as an indicator identifying how closely connected a vertex is in the network. In different studies, this technique determines the important vertices [21]. Graph represented by significance matrix of the metabolites also shows similar traits. Centrality analysis on this network reveals the important vertices which indicate the responsible

metabolites behind the disease condition [41]. In our system, we have incorporated four different centrality measurement techniques.

The first topological centrality analysis technique we have implemented in our system is degree centrality. This is the simplest yet most popular and effective centrality measurement technique in graph theory. This technique defines the number of connections the vertices have. The idea is that the more connection a vertex has, the more important it plays in the network. If we have a graph G:=(V, E) where V is the set of vertices and E is the set of edges, then i'th vertex V_i's degree centrality can be measured by $C_D(V_i) = deg(V_i)$.

The next adopted centrality analysis technique in our system is closeness centrality. Closeness centrality indicates how connected a vertex is with the rest of the network vertices. If we have a graph G:=(V,E) then closeness centrality of the ith vertex V_i would be $C_c(V_i) = 1/(\sum_{j=1}^{n} d(V_i to V_j))$. So checking the shortest path from a vertex to every other vertex in the network and then calculating the average gives us the closeness centrality of a vertex.

The third kind of centrality measurement technique that we have implemented in our system is the betweenness centrality which indicates how often a vertex comes up in the shortest path between two different vertex pairs in

the network. It is a complex yet potent indicator of graph vertices. If we have a graph $G := (V, E)$, then we can calculate the betweenness centrality of a vertex V_i such that -

$$C_b(V_i) = \sum (SP(V_s \rightarrow V_t \; through \; V_i)/(SP(V_s \rightarrow V_t), where[V_s \neq V_t \neq V_i] \tag{13}$$

Finally, the last among the centrality measurement techniques that we have implemented in our system is the page rank centrality analysis. The founders of google initially coined this technique. Page rank algorithm indicates a vertex's importance based on how important the neighbouring vertices -

$$C_{pr}(V_i) = (1 - d) + d(C_{pr}(V_j)/C(V_j) + d(C_{pr}(V_k)/C(V_k) + ... + d(C_{pr}(V_n)/C(V_n) \tag{14}$$

$j, k....n$ represents the neighboring node of i. d is a damping factor coefficient set between 0 and 1. Historically it is set to 0.85. $C(v)$ is the total number of outgoing edges from v. The formula was initially proposed for directed graphs; however it works great in undirected graphs by replacing each undirected edge with two separate directed edges. Overall steps of the analysis model are represented on the following page -

Algorithm 1 Differential correlation analysis

1: Read and split them into case and control dataset matrix naming $case_{matrix}$ and $control_{matrix}$
2: generate correlation matrix r_{case} and $r_{control}$ using Pearson correlation Coefficient
3: Generate a single diagonal matrix r_{diff} from r_{case} and $r_{control}$ using the above stated equations
4: Initiate a similar sized matrix $compute_{sig}$ to keep track of the p-value significance of the pairwise correlations
5: Initiate a similar sized matrix $compute_{sig}$ to keep track of the p-value significance of the pairwise correlations
6: **for** $number0fpermutation \leftarrow 1$ to 1000 **do**
7: generate a new case and control matrix with intial values naming $case - copy_{matrix}$ and $control - copy_{matrix}$
8: Select an integer n_{swap} where $1 \leq n_{swap}$
9: **for** $j \leftarrow 1$ to n_{swap} **do**
10: randomly genrate two integer i and j where $i \leq size(ncase_{matrix})$ and $i \leq size(control_{matrix})$
11: Swap the data point from ith position in $case - copy_{matrix}$ and jth position in $control - copy_{matrix}$
12: generate new correlation matriices like step 2 but this time from $case - copy_{matrix}$ and $control - copy_{matrix}$
13: Similarly generate a single diagonal matrix $permute - r_{diff}$ from the metrices generated in step 12
14: Compare intial r_{diff} from step 3 with $permute - r_{diff}$ from step 14.
15: **if** Corresponding pair in $r_{diff} < permute - r_{diff}$ **then**
16: Increment corresponding pair in the significance matrix $compute_{sig}$
17: Once $compute_{sig}$ is generated completely with all the p-vaalues of the corresponding metabolite pairs,discard the pairs not matching the initail p-value cut off margin generally $p - value < 0.05$
18: Render the matrix into a network and apply topological centrality analysis to find out the most central metabolite vertices.

Chapter 4 : Software implementation

4.1 Technical specification

Highlighting the software's domain and functionalities, we have named the project **Metamarker**. Inside the server platform, the software serves in the unit form that we call a project. Inside a project, users can create multiple jobs or tasks. Tasks can be of two kinds: 1) applying some normalization or prepossessing techniques, 2) some machine learning algorithms. The four significant modules that we have in the system are as follows :

1. Authentication and privacy module
2. Preprocessing and outlier handling module
3. Data processing and result generation module
4. Result analysis, comparison and reporting module

We also have a resource allocation module that ensures proper allocation of the computing resources and works in the server's background. The authentication module encompasses the user details and generates a unique reference key. Preprocessing and outlier handling module comes with data preprocessing algorithms to create normalized, cleaner, and efficient datasets. The data processing module comes with three different statistical data analysis techniques. Finally, the result analysis module provides an easy-to-use interface

where users can compare different modeling results and examine the differences. It also offers downloadable resources for users to serve other purposes. A general outline of its modular and parallel design is illustrated in Figure 3. It shows the primary module's activities and how they are mapped.

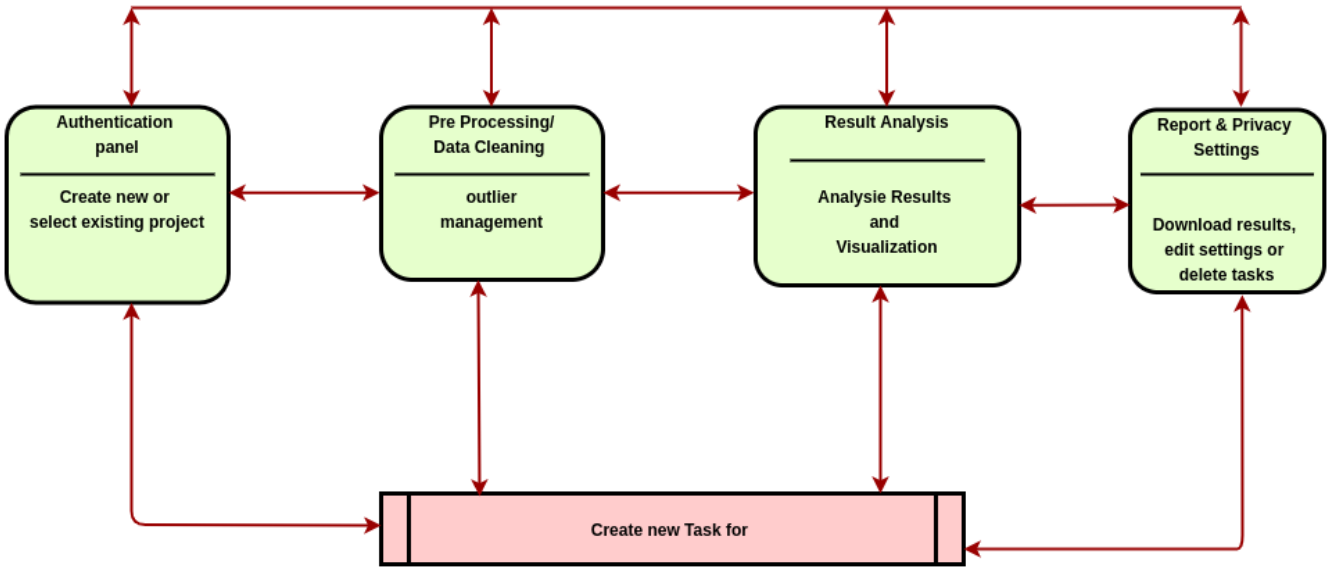


Figure 3: Metamarker design workflow. Major modules work independently. At any given time user can switch between the modules. The figure describes the pathways: which modules are accessible from other modules.

4.2 System environment

The project is developed with Python Django framework and various machine-learning libraries like Pandas, Numpy, Scickit, etc. We have used Github

(https://github.com/arafatarshad/metamarker) to manage the build process. The beta version is hosted with Amazon AWS web services with a dynamic IP address (http://18.189.6.35:8000/) and soon to be available for public access. Table 3 contains a more technical specification of the software.

We have designed the database using the Mysql workbench. The schema diagram is illustrated in Figure 4. The project has 24 tables, as shown. The bottom colored layer depicts the tables responsible for web sessions, authentications, migrations, background task processing/tracking, and other backend significant data storage. The background_task and completed_background tasks tables from this layer are essential since these two tables work as the queue for processing job requests. The system looks into these tables every three seconds to see if any new job is in the queue or not. Once the processing job gets applied, the request status gets updated and shifted to the other table. The second layer on the top left position represents the database tables designed to keep track of the project and associated processing jobs. The job table inside the left green layer holds all the job references. The tables inside the top right layer save each processing models' parameters and results in the system.

Table 3: Technical specification

Programming & ScriptingLanguage	Html: 58%, Javascript : 26.7%, Python:5.5%,CSS: 9.8%
Database and Database Server	MySQL, MySQL Server
Version Control and Repo	Git, GitHub
Backend Library	Django==2.1.7, django-background-tasks==1.2.5, django-compat==1.0.15 , djangorestframework==3.11.0, Flask==1.1.2, Flask-Compress==1.5.0
Data mining libraries	matplotlib==3.2.1, numpy==1.18.3 pandas==1.0.3, Pillow==7.1.2 plotly==4.6.0, PyMySQL==0.9.3 pyparsing==2.4.7, python-dateutil==2.8.1 pytz==2019.3, reportlab==3.5.42 retrying==1.3.3, scikit-learn==0.22.2.post1 scipy==1.4.1, six==1.14.0 sklearn==0.0, Werkzeug==1.0.1
Data Visualization Library	dash-core-components==1.9.1 dash-html-components==1.0.3 dash-renderer==1.4.0 dash-table==4.6.2 dash-table-experiments==0.6.0 D3 js V4
Network ing and Visualization Library	cytoscape js 3.16 3d-force-graph - 1.66.7
Preprocessing and Outlier Handling	Missing value Handling Mean Removal and Variance Saclling Univariate ScalingPareto ScalingLog Scaling Vast Scaling Xvast Scaling Range Scaling Level Scaling
Data Processing Algorithms	Principal Component Analysis Partial Least Sqaure (PLS -DA) Differentail Correlation Analysis and Network modeling
reporting	Table , chart, datasets, files
Mail Notification Server	smtp.gmail.com
Background Processing	Yes
Concurrent Task Processing	Yes

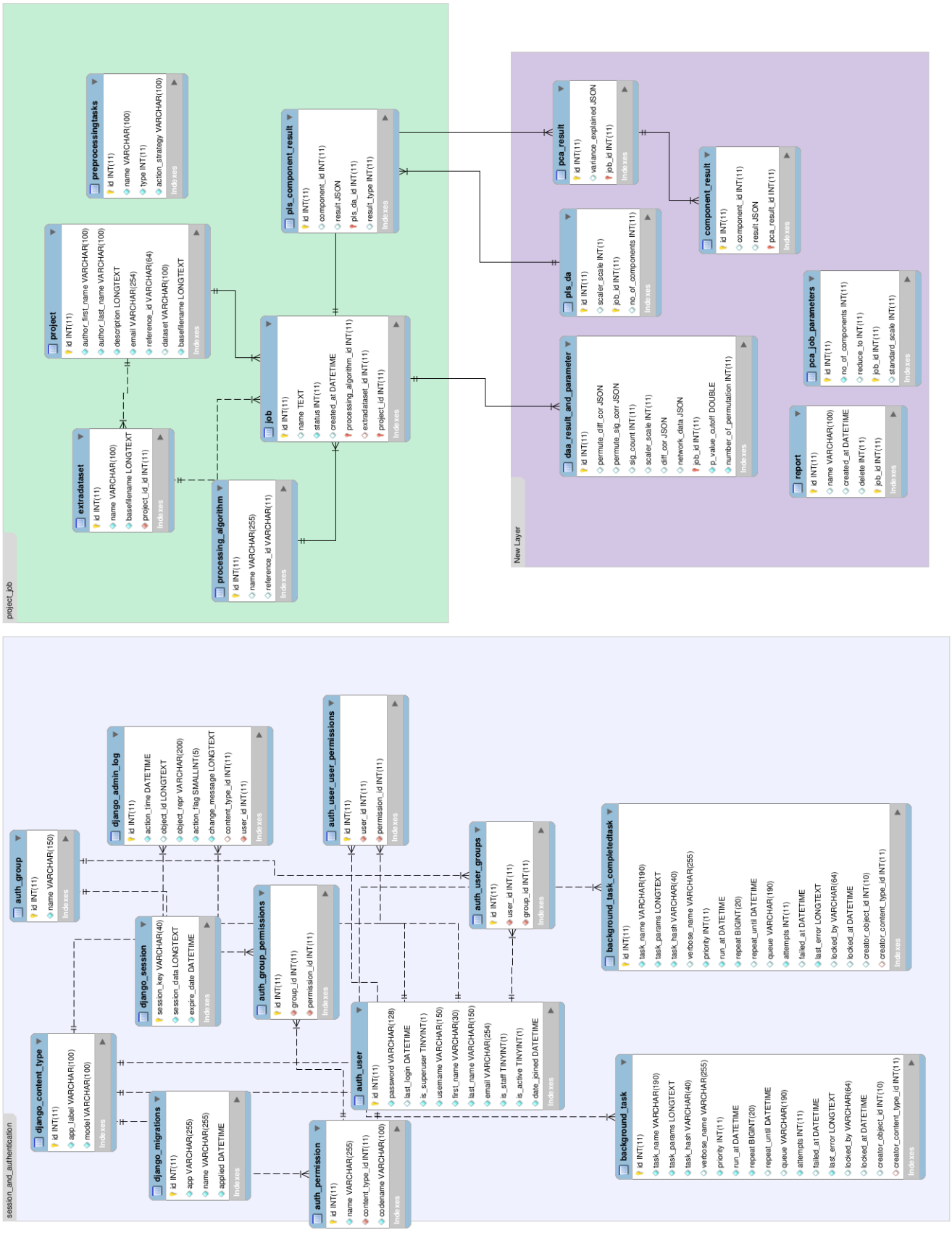


Figure 4: Database schema design

4.2.1 Project, task and data flow

Metamarker is manageable, uncomplicated, and efficient. To focus on the rapid purpose serving capability, we have considered minimizing overhead on futile tasks such as account registration or sign up, etc. Throughout the entire project, the user interface has been kept user-friendly, attractive, and resourceful.

The software serves users in the unit form of a project where users can create multiple projects with their email addresses. The system generates a unique reference key identifying the projects. Users can access the project anytime throughout the project life cycle using the reference key. Inside a project, users can apply preprocessing and outlier handling techniques to the datasets and generate new datasets. The system handles these kinds of functionalities in the form of instant tasks. On the other hand, the system serves users to apply statistical data modeling techniques as scheduled jobs since they generally require longer processing time. So, preprocessing of the datasets and processing of the datasets are handled differently in our system. Inside a project, users can create multiple processing jobs with preferred existing data set and preferred statistical modeling algorithms. Figure 5 Illustrates how user projects and processing tasks are related.

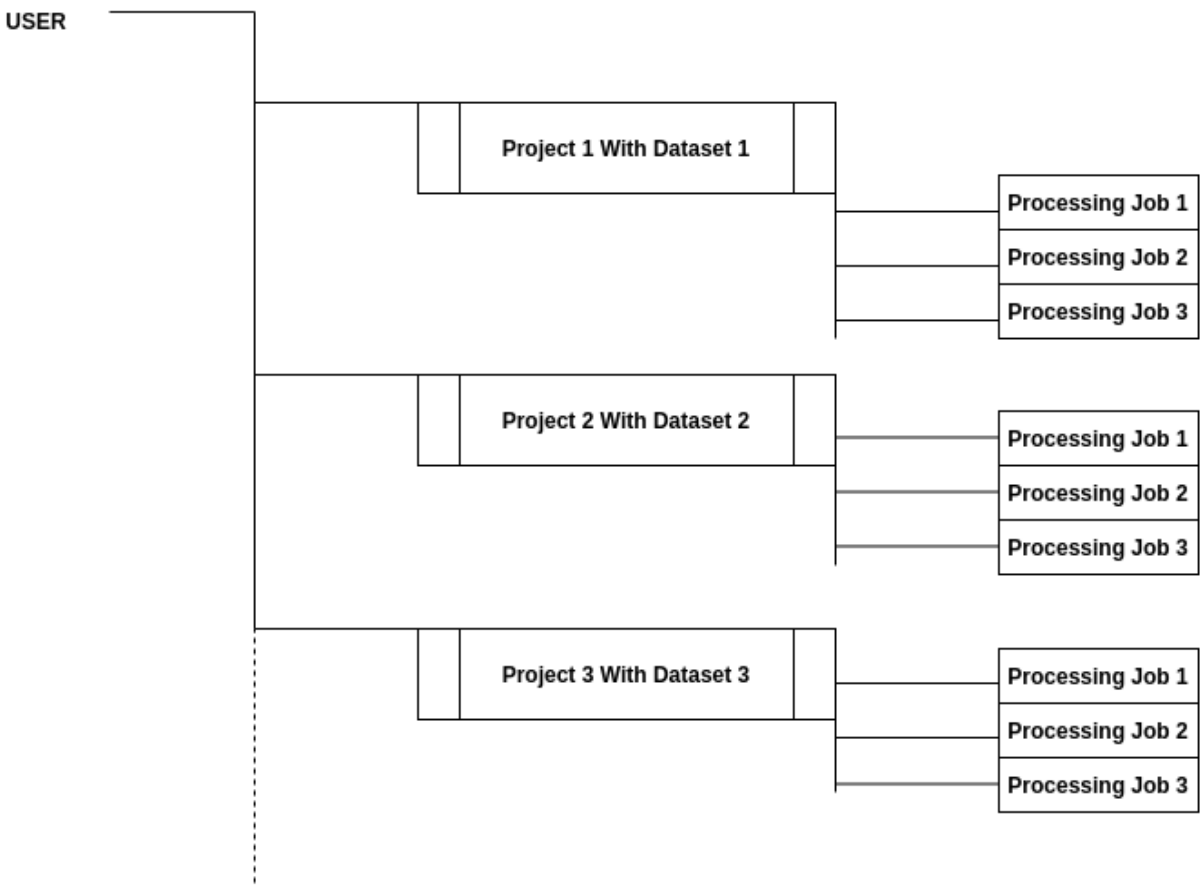


Figure 5: Projcet and processing task relation grid. All the processing jobs are part of a project. Users can create multiple projects and access them independently.

4.2.2 Job scheduling and task manager

High throughput data processing requires heavy computation. We have tested our system with a dataset of 152 metabolites and 380 instances of data points. We came to notice that preprocessing tasks were not taking

much time. On the other hand, processing the dataset with PCA or PLS and differential correlation analysis consumed time ranging from 10 seconds to 10 minutes to generate results. We have tested the system with the same data set in three different environments and noticed a similar outcome. The settings used during these testings are as follows. 1) Macintosh computer with 2.7 GHz intel core i5 processor with 8 gigabytes of RAM and one terabyte of the storage drive. 2) Linux operating system on 2.5 GHz quad-core Intel i5 processor and 16 gigabytes of ram with one terabyte of storage. 3) Amazon AWS, an instance with Linux based operating system, and an amazon real-time SD database with one terabyte of storage. More about the testing process is discussed in the discussion section. We have designed our system to be portable and released it with open-source licensing through the Github repository. So the users can also install and host their private portal with our program. To ensure that the project works on a different platform, we have implemented a job scheduler for the processing tasks.

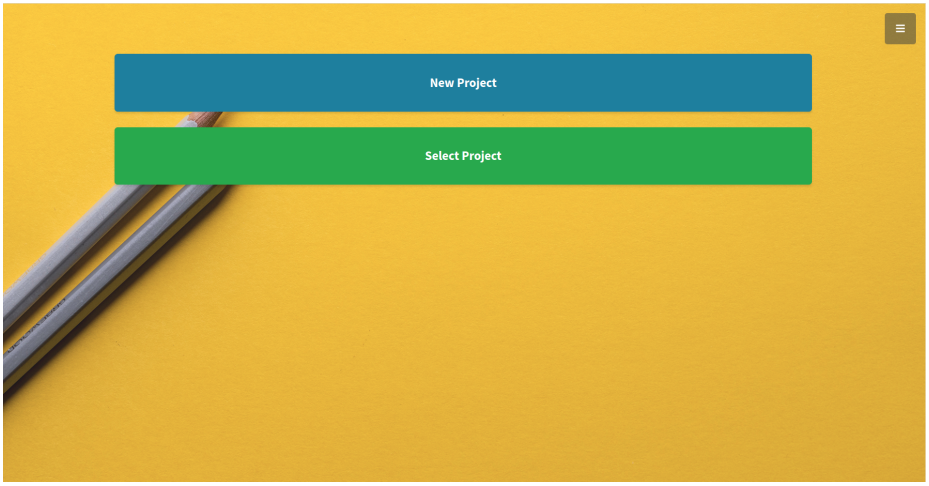
The job scheduling method runs in our server's background to ensure that the server host is putting all its resources on one processing task at a time. Once a user creates a processing job, the request gets stored in a queue. The server processes one request applying the designated processing methods. Once the job is executed, the server moves to the next job request waiting in the queue. At any given time, the user can see the job requests that have

been created under a project through the task manager panel in our system. The task manager is designed like a table with six columns. A glimpse of the task manager interface of the software is presented in Figure 6. The rows represent the jobs. The first column denotes the name of the job that the user has put during the job creation. The second column denotes the dataset name. The third column denotes the date when the job was created. The fourth column defines the status of the job. Jobs can be under pending, running, and finished status. The fifth column denotes the model which is used for that processing job. Finally the last column is what provides the user manage the jobs. User can either delete the jobs or once the system completes job, user can access the results by clicking on the show results button.

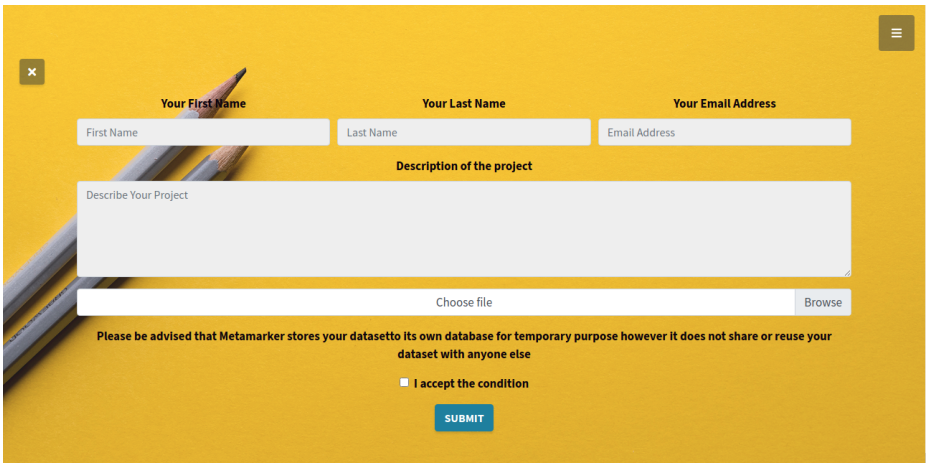
Figure 6: Task manager table : list of existing processing job

4.3 Software interfaces

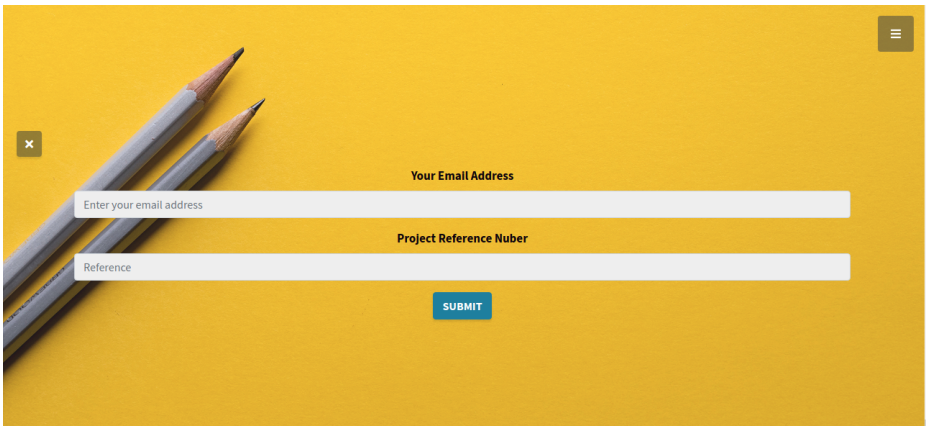
We have designed the software with minimal registration/sign-up process for rapid transaction towards the data processing. Upon arrival to the landing page user will be asked to select an option whether to create a new project or select an existing one from the user's project list (shown in Figure 7(a)). The project life cycle starts with a dataset uploaded through the project creation form which is shown in the figure 7(b). The form has only five input fields to fill up, which takes less than a minute to complete. During project creation, the system generates a 128-bit unique reference key. This key is crucial for the user to preserve for follow-up access to the project. Our system shares the reference key only once throughout the project life cycle that is during project creation. The key is forwarded to the user via the email channel that the user signed up with. Select a project button as shown in Figure 7(a), which allows the user to access an existing project using the reference key of the project (Figure 7(c)). The authentication grid takes only two inputs. The first one is the user email, and the second one is the project reference key. Once the submit button is clicked, the system verifies the key against the email address and loads the corresponding project.



(a) Landing Page Create or Select Project



(b) Create New Project page

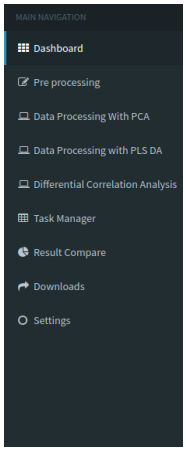


(c) Select existing project page

Figure 7: Project landing page, create, select ,authenticate

Successfully authenticated users are initially redirected to the home page/ dashboard of the software. The admin panel's major components are shown in Figure 8. The admin panel is developed on an open-source admin panel interface named Admin LTE [11]. The fundamental structure of the web pages throughout the admin panel has been kept similar and consistent.

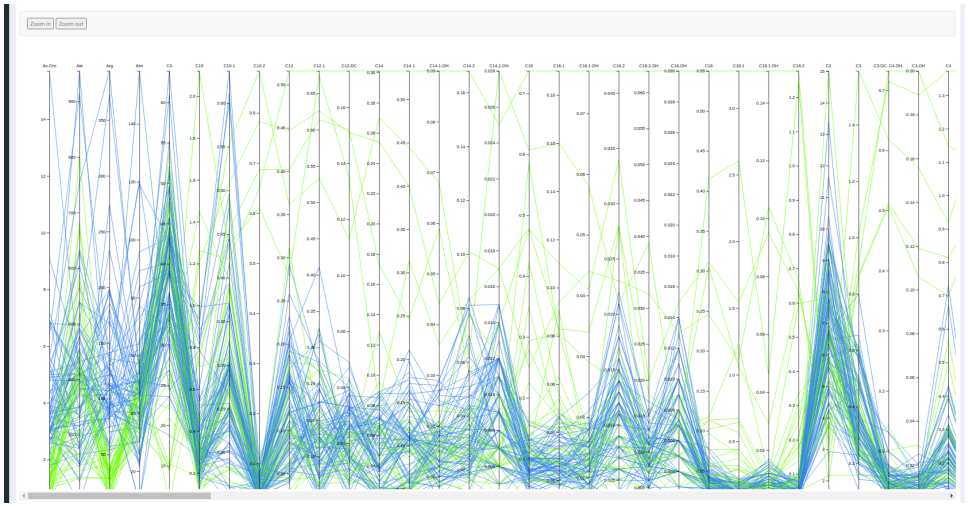
The navigation bar (Figure 8(b)) provides access to all the modular features. The homepage offers a basic overview of the initially uploaded dataset. It comes with two tables and a line chart. The first table in Figure 8(c), provides a glimpse of the dataset. Right next to that comes a line chart. It represents all the data instances in the data set regarding the feature name or dataset column. Again, The table shown in Figure 8(d) provides an overall description of the initial dataset. This second table (Figure 8(d)) has four columns the first column is the serial number, and the second column lists all the feature names in the dataset. The third column indicates the type of the data instances, and the last column shows the number of outlier/missing values associated with that feature.



(a) Navigation bar

Ac.Orn	Ala	Arg	Asn	C0	C10	C10.1	C10.2	C12	C12.1	C12.DC	C14	C14.1	C14.1.O	C14.2	C14.2.O	C16	C16.1	C16.1.O	C16.2
1.6	612	34.9	46.2	38.8	0.512	0.373	0.113	0.126	0.247	0.031	0.09	0.076	0.025	0.042	0.01	0.544	0.102	0.023	0.016
0.8	420	26.52	48	48.2	0.266	0.3879	0.092	0.061	0.213	0.031	0.025	0.051	0.0110	0.038	0.0110	0.166	0.063	0.015	0.011
2.1	367.8	61.9	59.7	45.53	0.257	0.5379	0.045	0.1119	0.294	0.03	0.043	0.138	0.0260	0.0279	0.0130	0.1180	0.0289	0.019	0.008
0.3	433.9	3.8	60.4	40.095	0.261	0.2960	0.064	0.096	0.179	0.0370	0.0540	0.1040	0.017	0.0289	0.006	0.159	0.045	0.0139	0.006
1.4	305.6	33.4	37.4	40.029	0.262	0.293	0.035	0.095	0.154	0.0279	0.05	0.0969	0.016	0.0260	0.006	0.1080	0.0270	0.01	0.006
1	342.4	25.9	28.3	13.804	2.168	0.43	0.895	1.754	0.7609	0.12	0.602	0.4070	0.085	0.1009	0.0260	1.4140	0.078	0.034	0.023
1.7	801.7	35.7	49.4	27.521	0.223	0.35	0.047	0.081	0.201	0.0370	0.038	0.12	0.0130	0.031	0.0069	0.09	0.0270	0.012	0.009
0.7	640.7	7.3	39	33.13	0.153	0.2460	0.0440	0.0520	0.128	0.04	0.046	0.085	0.0130	0.0110	0.004	0.141	0.0279	0.0110	0.004
1.6	590.2	31	37.6	33.533	0.207	0.304	0.038	0.084	0.1639	0.045	0.0409	0.12	0.0130	0.0270	0.006	0.109	0.033	0.012	0.006
1.5	353.1	34.3	42.8	14.537	0.758	0.4379	0.612	0.2789	0.675	0.135	0.081	0.314	0.0360	0.083	0.017	0.25	0.094	0.0440	0.023
2.6	337.5	63.8	47.7	32.396	0.21	0.266	0.04	0.077	0.172	0.0409	0.04	0.113	0.015	0.019	0.008	0.069	0.023	0.008	0.006
0.7	454.3	16.7	54.6	42.515	0.228	0.273	0.0559	0.08	0.131	0.03	0.048	0.1040	0.012	0.0260	0.006	0.122	0.033	0.0139	0.009
1.4	553.4	25.9	50.8	57.078	0.3720	0.2319	0.06	0.136	0.131	0.047	0.081	0.102	0.015	0.0270	0.0069	0.161	0.0440	0.021	0.012
1.8	357.8	44	48.2	53.016	0.354	0.198	0.048	0.1119	0.114	0.049	0.0579	0.077	0.0090	0.025	0.0069	0.128	0.034	0.0130	0.008
1.2	466.1	24.4	38	47.99	0.298	0.2269	0.042	0.12	0.141	0.046	0.077	0.067	0.019	0.0220	0.006	0.177	0.035	0.017	0.006
1.3	388	33.4	48.3	51.4	0.2860	0.268	0.0559	0.067	0.188	0.032	0.033	0.0590	0.017	0.0409	0.0110	0.138	0.0540	0.008	0.013
2.4	413.7	62.3	42.2	29.274	0.187	0.222	0.025	0.092	0.154	0.0360	0.055	0.115	0.017	0.0090	0.004	0.0540	0.01	0.015	0.003
1	381.6	8.9	42.1	42.928	0.379	0.35	0.07	0.131	0.21	0.04	0.094	0.129	0.0260	0.048	0.01	0.278	0.057	0.0220	0.011
1.2	670.4	20.9	57.9	48.288	0.335	0.257	0.1180	0.1230	0.1730	0.045	0.095	0.139	0.023	0.0409	0.0069	0.585	0.081	0.0289	0.013
1	622.5	16.8	49.7	51.663	0.29	0.267	0.0819	0.15	0.151	0.05	0.083	0.154	0.0180	0.023	0.005	0.2269	0.043	0.017	0.01
2.4	264.7	70.5	36.2	47.215	0.354	0.244	0.061	0.149	0.163	0.04	0.08	0.151	0.017	0.033	0.0069	0.1159	0.0440	0.019	0.008
1.3	241.3	25.6	35.1	27.148	0.3289	0.233	0.0360	0.136	0.1180	0.04	0.061	0.071	0.01	0.025	0.005	0.051	0.019	0.01	0.006
1.2	267	18.7	45.4	41.876	0.298	0.168	0.04	0.12	0.1040	0.038	0.063	0.102	0.017	0.021	0.0090	0.145	0.031	0.0130	0.006
1.6	254	50.4	43.2	37.2	0.133	0.257	0.0540	0.0409	0.168	0.033	0.0260	0.046	0.0110	0.019	0.008	0.0890	0.025	0.008	0.006
1.1	391.1	3.7	43	40.343	0.366	0.341	0.0969	0.142	0.159	0.042	0.079	0.156	0.019	0.035	0.0090	0.21	0.065	0.02	0.01
1.4	333.2	25.5	48.8	38.156	0.342	0.335	0.1119	0.1369	0.2080	0.043	0.06	0.083	0.02	0.025	0.005	0.086	0.0279	0.015	0.006

(b) Data overview table



(c) Line chart representing data with respect to features

Observations about the dataset

No	Name Of Column	Value Type	Missing Value
1	Ac.Orn	float64	3.0848 %
2	Ala	float64	5.9126 %
3	Arg	float64	1.5424 %
4	Asn	float64	1.5424 %
5	C0	float64	0 %
6	C10	float64	0 %
7	C10.1	float64	0 %
8	C10.2	float64	0 %
9	C12	float64	0 %
10	C12.1	float64	0 %
11	C12.DC	float64	0 %
12	C14	float64	0 %
13	C14.1	float64	7.7121 %
14	C14.1.OH	float64	0.5141 %
15	C14.2	float64	0 %
16	C14.2.OH	float64	2.5707 %
17	C16	float64	0 %
18	C16.1	float64	0 %
19	C16.1.OH	float64	1.0283 %
20	C16.2	float64	3.3419 %
21	C16.2.OH	float64	2.8278 %
22	C16.OH	float64	5.3985 %
23	C18	float64	0 %
24	C18.1	float64	0 %
25	C18.1.OH	float64	2.3136 %
26	C18.2	float64	0 %
27	C2	float64	0 %

(d) Dataset detail table

Figure 8: Project homepage, main navigation, and associated data overview

All the modular features are arranged through the left navigation bar in the software. The first tab in the navigation bar provides access to the preprocessing panel of the software. Once users arrive at the data preprocessing and outlier handling page (Figure 9), the system will allow the user to select the existing dataset under the project. Every time a preprocessing task is performed, a new dataset is created and stored inside the project. Users can choose one or multiple methods from the checklist. Finally, when confirmed, the system will apply the techniques and generate the new dataset.

4.3.1 Data processing with principal component analysis

The navigation menu has an option labeled **Data Processing with PCA** which links the webpage where users can apply PCA to their preferred datasets. The system performs PCA tasks in the form of a scheduled processing job. Once the user creates a job with the desired configuration, the job is submitted to the processing queue. When the job is executed, user can find the results through the task manager.

The web page interface for PCA has been implemented with minimal configuration, Keeping consistency with simplicity(Figure 10). PCA serves users both exploratory and dimension reduction purposes. When the focus is an

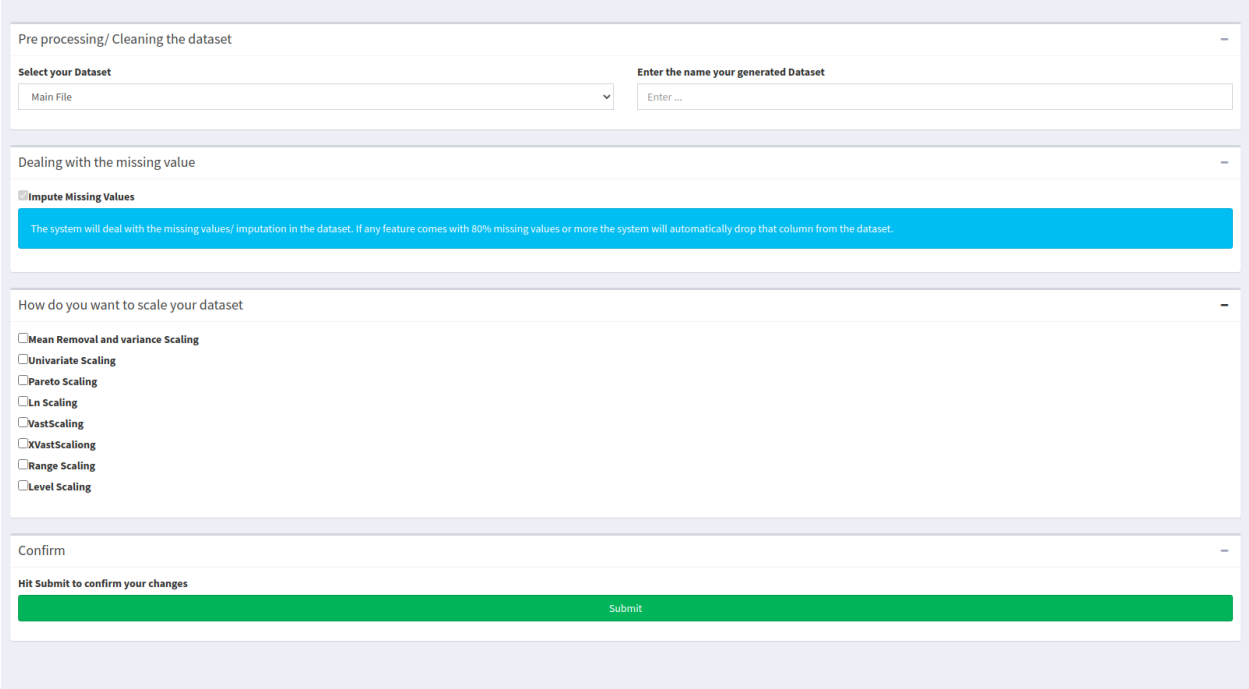


Figure 9: Data preprocessing panel. Users can select the dataset and name of the resulting dataset from the input boxes. Users can choose one or multiple preprocessing techniques from the checkbox. Finally the system applies the tecniques and saves the resulting dataset with user given name.

observation of the dataset, users can select the number of components generated by the system. When the primary intention is to reduce the number of dimensions and inspect significant metabolites, users can configure the other dropdown. The system allows users to reduce the variables to as low as 10% of the total number of variables present in the dataset. For example, if a dataset comes with 100 variables, the system will lower it and present the first ten variables showing the dataset's highest variance. Components in PCA's are generated in an ordered form. The first component holds the most

variance, and the second component is lesser than that, and so on. From this panel, users can generate a minimum of two to a maximum of N number of components. N being the size of the dataset variables.

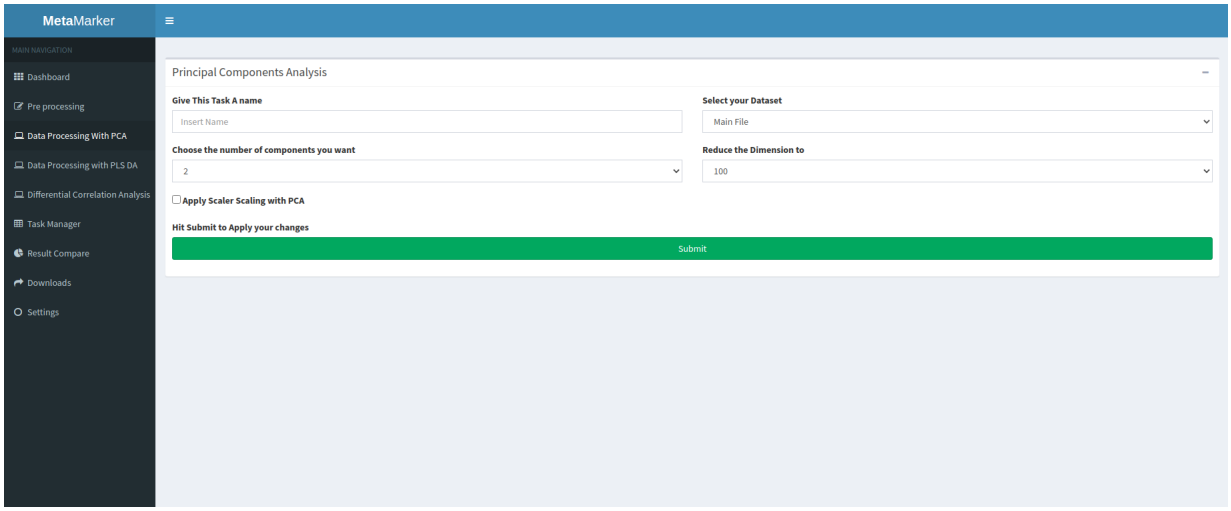


Figure 10: Data processing with PCA

4.3.2 Data processing with PLS-DA

The navigation menu labeled **Data Processing with PLS** links the web page for the PLS application interface (Figure 11). The system applies PLS in the form of a processing job. From the interface, users can select a dataset, put a name and create the job. Once the server executes the job, users get notified to access the result. Our system adopted the idea of a component-based PLS application. Users can select multiple components. On each component,

a different latent space is found with different loadings and weight matrices. The first component captures the most significant latent space and associated results. The second one captures comparatively lesser and so on.

PLS depends on the data group variables or class variables. The program must know which column represents the group. In general, PLS is designed to work on multiple dependent variables. It can have multiple Y variables; however, to keep it simple, we have designed our PLS model to support only one dependent Y variable, which has to be represented in the last column of the dataset.

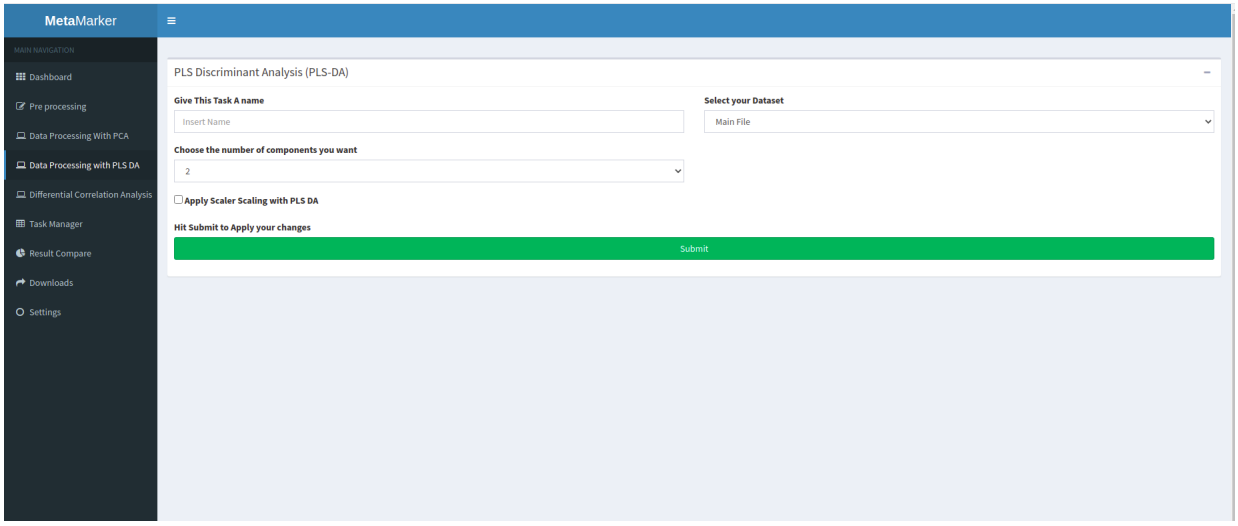


Figure 11: Data processing with PLS-DA

4.3.3 Differential correlation analysis in data processing

Our quest for an out of box analysis techniques led us to look over enormous new methods. We understand that the differential correlation study model

could be a compelling alternative viewpoint for metabolite interactions. The model has proven its constructive results finding the key metabolites responsible for Osteoarthritis among the adults [29]. As mentioned before, this model has two distinct phases or stages. The first stage is where differential correlation among the metabolite pairs is analyzed, significance p-value among the pairs is studied. In the second stage, a network is modeled. Topological analysis is conducted to find out the central-most significant metabolites. Similarly, in our software, we have divided the model to work on the datasets in two different stages.

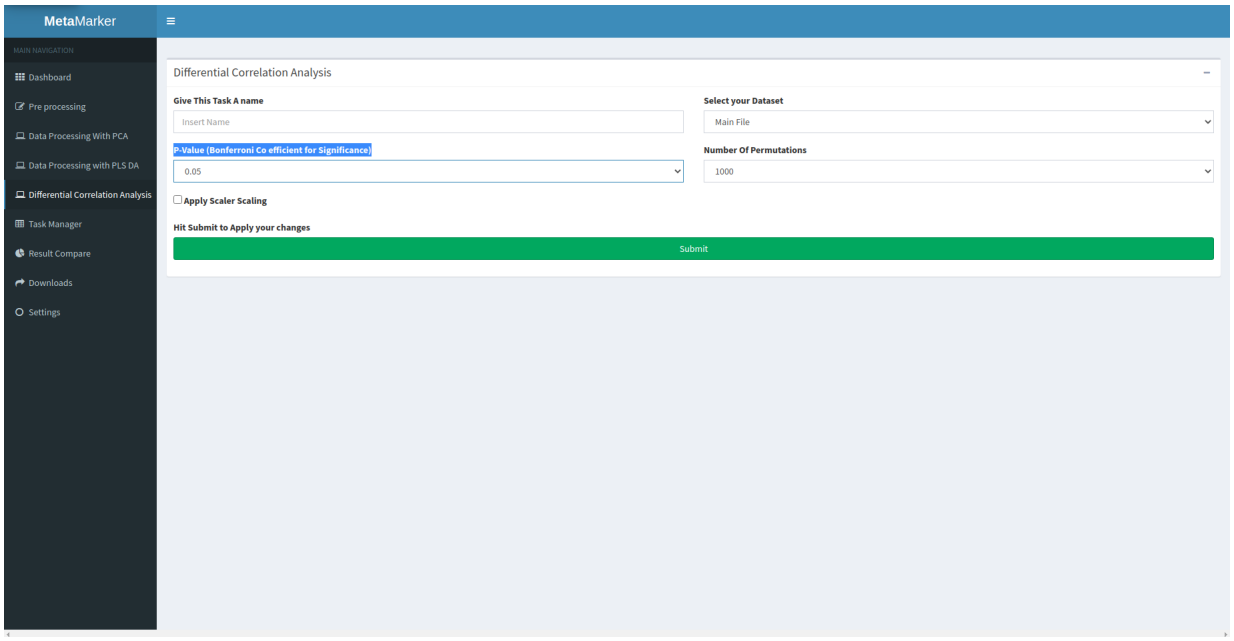


Figure 12: Data processing with differential correlation analysis

In the first step, we generate the significance matrix denoting pair wise interaction of the metabolites. Like any other processing job in the software, this

task is also carried out in the form of a scheduled processing job. Users can create a new job from the **Data processing with differential correlation analysis** panel in our software (Figure 12). During the job creation, the user can select the number of permutation starting from 100 to 1000. User can also configure the p-value cutoff anything in between from 0.001 to 0.05.

Chapter 5 : Results, reports and result comparison module

So far, we have discussed the analytical features and different models of the system. We have also discussed the design workflow, including the analytical features that can be accessed through our system. We have discussed other pages like the preprocessing and processing module. We have also looked into the system environment, database, and many more. Now we are moving more into our outputs, deliverable and observational side of the software. This thesis has been aimed to provide a rigid shape to metabolomic biomarker research. Simultaneously, we were also focusing primarily on providing a platform for the differential correlation analysis represented by network and topological sorting. PCA and PLS has been used in numerous studies, and they already have the proper attention from researchers. The differential correlation analysis model [29] on the other hand, is new to the domain. Thus this method was our center of attention. To make the analysis model familiar, accessible, and effective, we have added enormous interactive visualizations and sorting mechanisms. This chapter will first discuss the results and illustrative visualizations we have implemented for differential correlation analysis and then move forward with other results from PCA and PLS.

Moving through this chapter to keep consistency, we will demonstrate our system with the same metabolic dataset. The data set we have used in our test purpose is collected and initiated in 2011 for an Osteoarthritis Study among the Knee OA patients undergoing knee replacement surgeries in Newfoundland [75, 77]. The data set was generated recruiting patients at St Claire's Mercy hospital and Health Science's Center General Hospital in St John's over a couple of years. The primary motivation for us to use this dataset is because it is a very well-maintained and clean dataset.

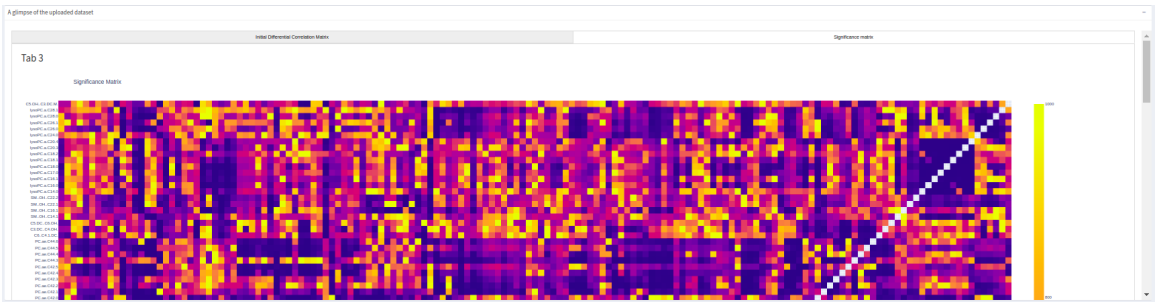
5.1 Differential correlation analysis: results and visualizations

We have already discussed that the model has two major distinctive stages. In the first stage, the pairwise correlations of the variables are analyzed, and a diagonal matrix representing their significance is generated. The second stage involves converting the significance matrix into a network and sorting the vertices based on topological properties. The differential correlation network and associated results are implemented over two different web pages in our system. Network rendering, visualization, and run-time processing require heavy computations. Various front-end scripting libraries are involved in the pages. Combining them in one page would not be efficient, and users might experience lagging or delay in page loading. That is why multiple pages were designed to reduce the run time resource and computation requirements.

The first result page covers the entire prospect of our analysis method. On top of the web page, we have generated two heat maps. Both of the heat maps are rendered in the backend server using python's plotting library named Plotly and Dash[2]. The heat maps display the correlation matrix and the significance matrix. The heat maps are interactive with scroll-able and zoom-able properties. Hovering over the map elements, users can get the corresponding metabolite names and their Pearson correlation coefficient. In the significance matrix, the actual significance coefficient value is generally a tiny fractional number. To illustrate that visually we have multiplied the value by 1000 to make it more visual and illustrative. An example can be seen in Figure 13, where we have generated the heat maps using our Osteoarthritis dataset. The rows and columns of the heat maps represent the metabolites from our dataset while the cell represents the values.

First result page also contains the network graph representing the significance matrix (Figure 14). The network is generated on a JavaScript canvas pane. We have used front end java script library named cytoscape.js [20] to render the graph. The vertices in the network represent the metabolites and the edges represent the correlation among them. The network is highly user interactive. User can use their mouse to drag and resize the pane as well as the graph separately. Users can rearrange the vertices. It also has the

(a) Correlation heatmap (partial view)



(b) Significance Heatmap (partial view)

Figure 13: Heatmaps placed in the first result page. (a) represents a screenshot of the correlaton matrix generated by differential correlation analysis. (b) represents the screenshot of the significance matrix which is resulted as final outcome of first stage in differential corrleation analysis. The X and Y axis in both these plots represents metabolites.

analysis and different modeling functionalities that user can select from the drop downs placed on top of the pane.

The first drop-down option from the menu bar is for layouts. Users can model the graph with four different layout models. The next drop-down is for the topological analysis. This drop-down provides the centrality measurement techniques which we have described in chapter 3. The third drop-down

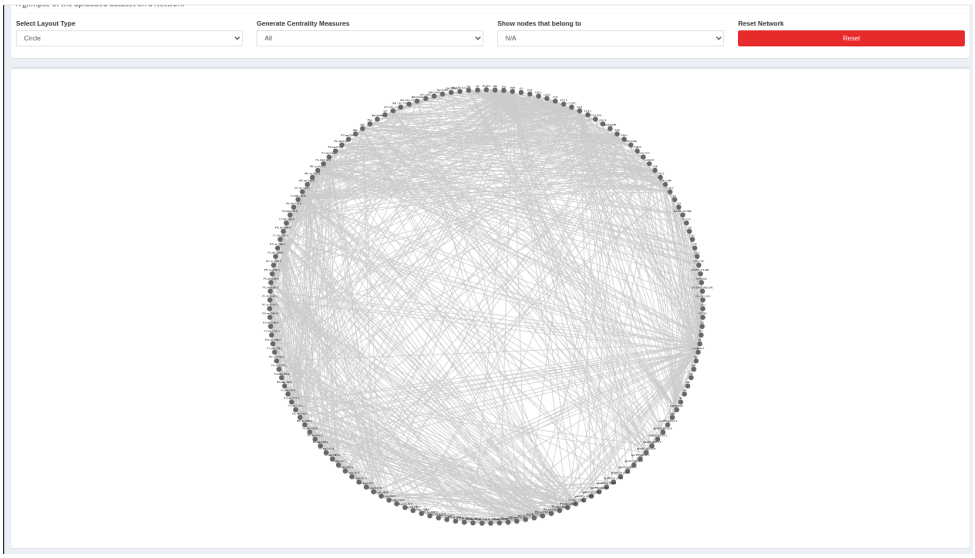
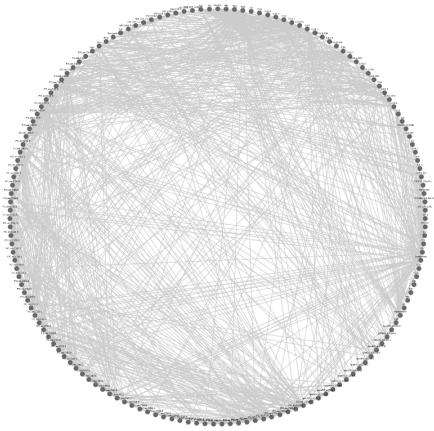
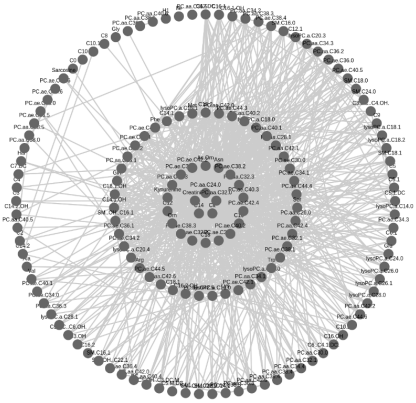


Figure 14: Differential correlation network analysis: default view with circular layout. The top of the pane contains three dropdown options with more functionalities applicable to the network. Inside the network the vertices represents the metabolites. The edges represents the interactions having p-value greater than cutoff value.

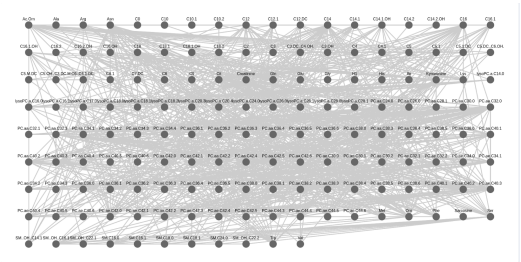
in the menu bar allows users to filter the graph for better visualizations. Finally, we have a button in the menu bar that resets the map. By default, the network is rendered with a circular layout. While circular layout places the vertices on the circumference, the concentric layout uses multiple layers, one over the other. The grid layout represents the vertices in the shape of a rectangular grid. Finally, we have the fourth layout model named breadth-first approach. The concept is similar to concentric, yet the vertices in this layout options are layered in a parallel manner. A glimpse of all these layouts applied on our test dataset can be seen in Figure 15.



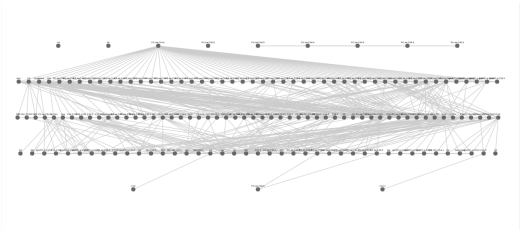
(a) Circular layout of the graph



(b) Concentric layout of the graph



(c) Grid layout of the graph



(d) Breadth-First layout of the graph

Figure 15: Differential correlation analysis: layout rendering options

Centrality analysis on the network reveals important vertices indicating the responsible metabolites behind the disease condition. As mentioned before, the second drop-down from our menu bar provides users the four centrality measurement techniques applicable to the network. Once selected a method, the system generates a table listing all the metabolites and their respective centrality values in the form of a pop-up modal table. Such an example is shown in Figure 16 using our test dataset. Here we have generated a table of page rank centrality from the significance matrix of our dataset. One great additional feature of this table is the sorting/filtering ability. Users can sort the list of metabolites from the table and select their preferred number of metabolites for further validation studies.

The third dropdown in the menu bar allows the user to filter the network for better analysis. Once the user selects from the list options, the system runs a centrality analysis on the network and removes the vertices that do not fit the selected criteria. The dropdown selection options are sorted percentage-wise. For example, suppose a user wants to sort the network according to degree centrality, keep the top 50% and remove the rest. In that case, all the user has to do is select the top 50% central vertices option from the dropdown, and the system re-rends the graph accordingly, running all the steps in the background. Finally, the last option in the menu bar is a button that resets the network any time throughout the session. Instead of reloading the entire

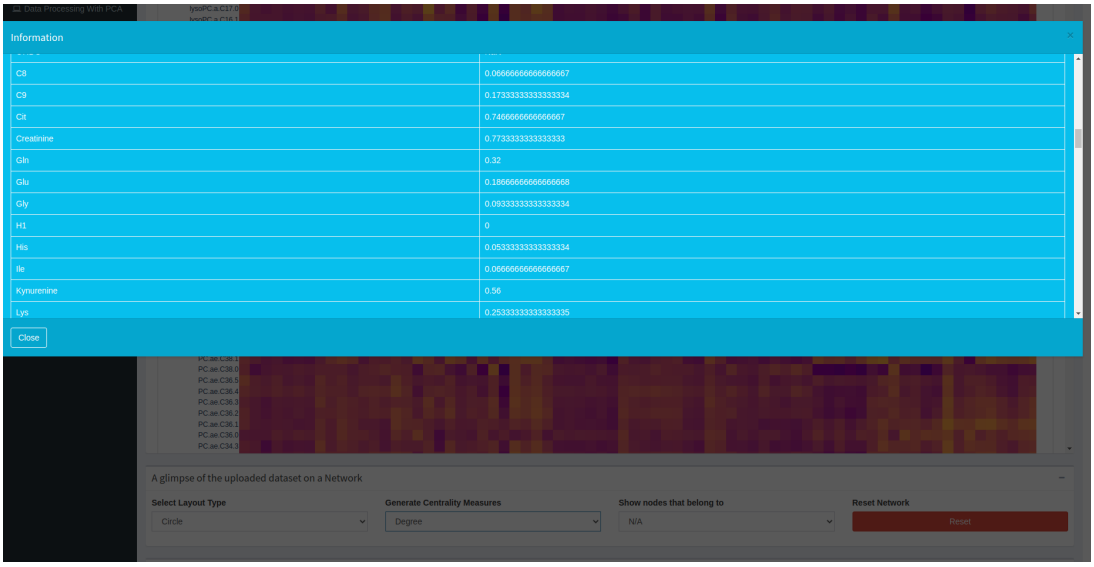


Figure 16: Modal table with centrality analysis. The modal appears once user applies the centrality measurement method from the second dropdown menu.Left column lists the name of metabolits and right column lists their value. The table can be sorted or exported into csv or json file.

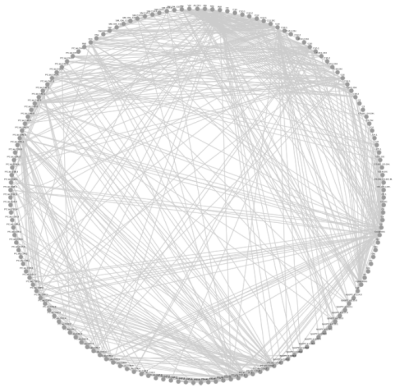
page, users can refresh the network and save loading time through the Reset button.

Second web page is entirely focused on the correlation network and its different run time states. Upon arrival to this web page, users will see a similar graph to the previous page. While the last pane offered filtering by the vertices, this new pane allows users to filter the network based on edge properties. The input form allows the user to filter the graph with p-values ranging between 0.001 to 0.05. Edge filtering based on p-values provides more visibility and perspective when the network is very dense. To show how it

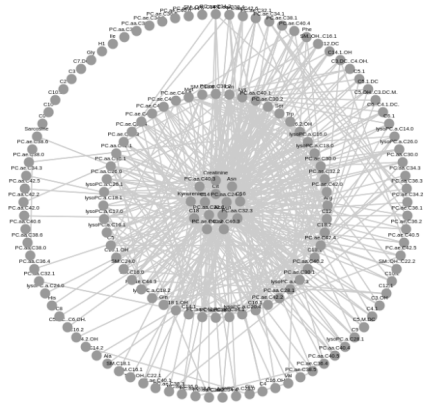
makes the difference, in Figure 17 we have regenerated the same graph presented in Figure 15. This time we have filtered the edges with 0.001 p-value limit. In contrast, the previous one had a p-value limit of 0.01. We can see this time the graph seems cleaner, less dense, and more visible.

The graph on this second web page has more interactive capabilities than the one on the previous page. The edges and vertices both respond to mouse left clicks and right clicks differently. Also, along with the network pane, we have added two more boxes on the bottom. The first box among them provides clicked object information and the second box offers some applicable functionalities on the related object Such an example can be seen in Figure 18 based on our data set. We have designed the interaction with four different approaches described below.

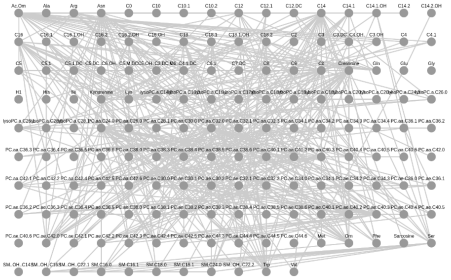
Browsing and navigating through a graph could be very hectic sometimes. Especially with sparse networks where edges overlap each other and vertices are very hard to notice. Also, sometimes, a user might need to explore the network figuring out preferred interactions randomly. To make this process smooth and easy, we have incorporated a selection feature. At any given time, users can select a vertex by left mouse click on the vertex. This will do three things. First, the vertex will be highlighted (animated) as a selected item. The clicked object information table will generate the vertex's informa-



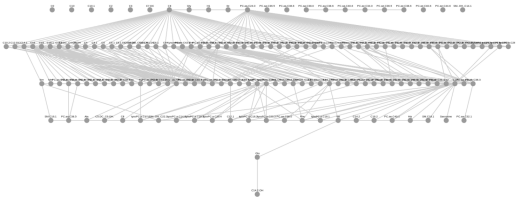
(a) Circular layout of the graph



(b) Concentric Layout of the graph



(c) Grid Layout of the graph



(d) Breadth-First layout of the graph

Figure 17: Network layouts with edge filtering applied

tion, including the vertex's name and a table listing the neighboring nodes and the edges' p-values. On the other hand, more options for the clicked element table will provide options to change the element color for helpful browsing or delete the vertex to make the network density low. Using the

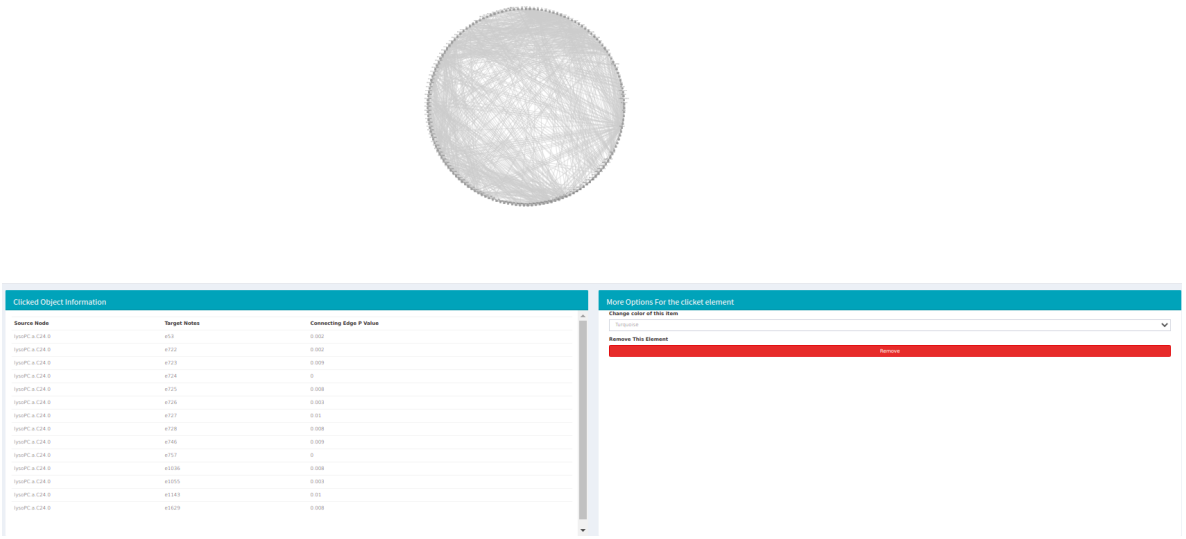
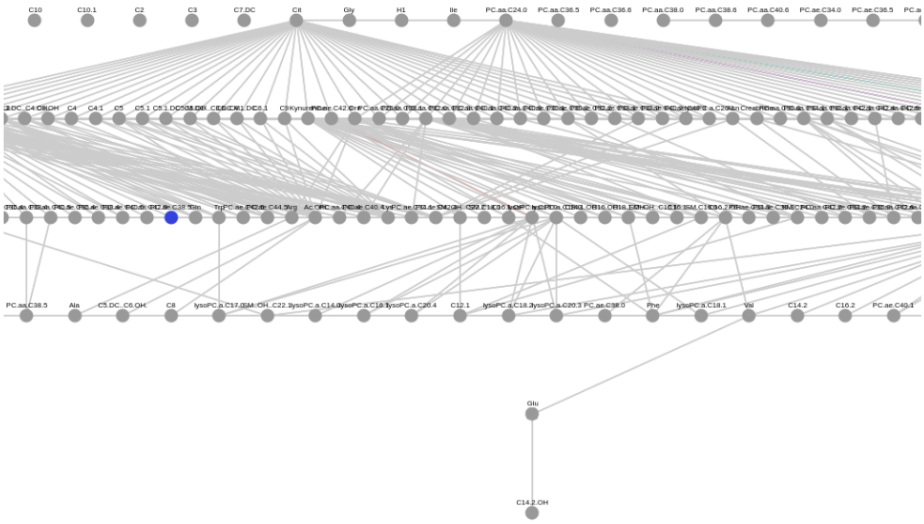


Figure 18: Extended network analysis options in second result page. Apart from the network the UI provides two tables left one named clicked object information table right one named more options for the clicked element

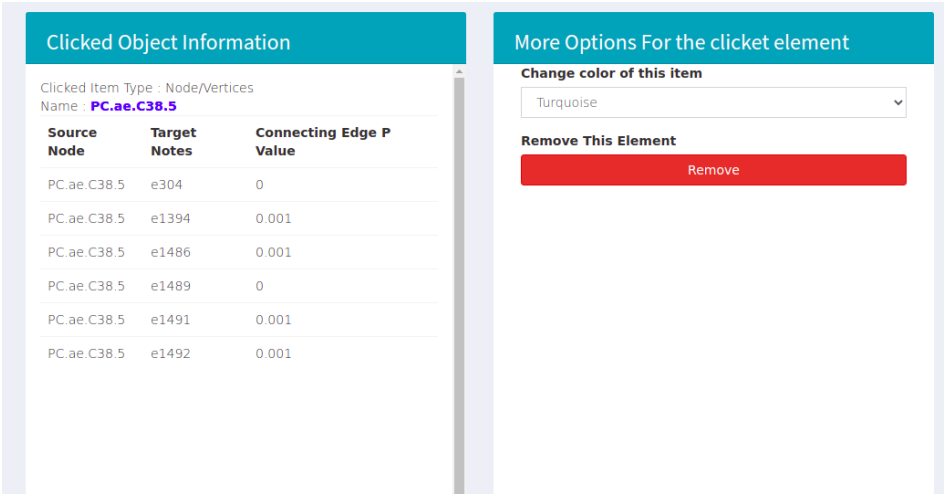
Osteoarthritis dataset, We have demonstrated this interaction in Figure 19.

Another interaction feature provides an ability to find out similarly valued edges in the network. Upon right-clicking on edges, the system animates and highlights all other edges having the same p-value. This interaction also dynamically renders the clicked object information table and more options for the clicked element table. Since the clicked element is an edge, the first table generates the source vertex and the target vertex connected by the edge and shows their p-value. The second table provides an edge coloring or deleting option. In Figure 20 we have presented a glimpse of the right-click action on edges. The network edges also support left mouse click events that animate the source and destination vertices of the edge in the network.

The graph that we have implemented is very interactive and informative.

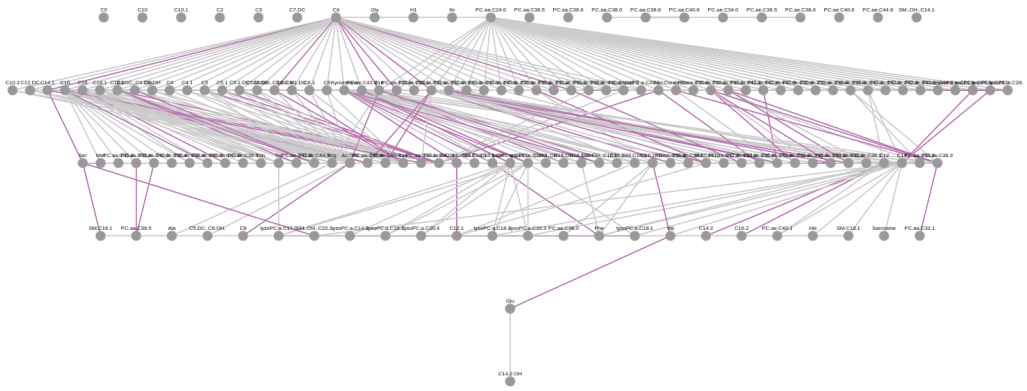


(a) This pictures demonstrates the left mouse click action on the network. Upon mouse click on vertex PC.ae.C38.5 highlighted it with color blue

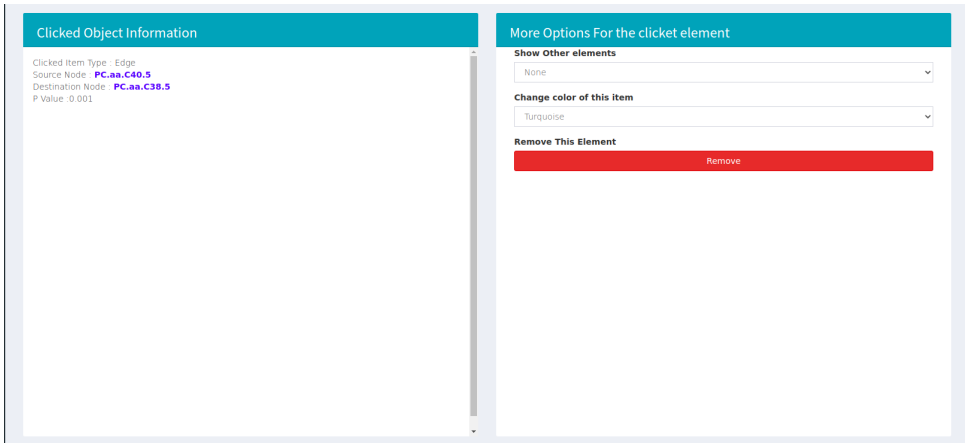


(b) and generated two boxes in (b). Left one in (b) includes a table listing all the neighbours of the vertex.Right one provides a option to manually color or delete the vertex.

Figure 19: Left mouse button interactions and panel information about the clicked items



(a) Selection of one edge and animating the similar or lesser p-valued edge in the network



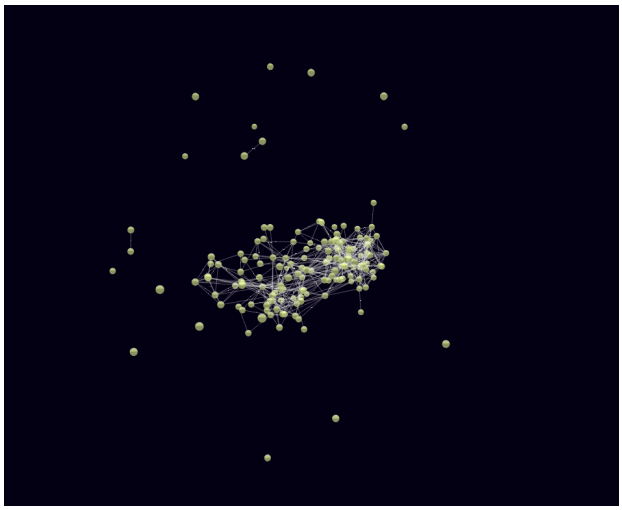
(b) Boxes that are generated dynamically with more information and added function

Figure 20: Right mouse click interaction on the network. which highlights similar p-valued edges and generates two tables. First table provides general information second one provides interactive options.

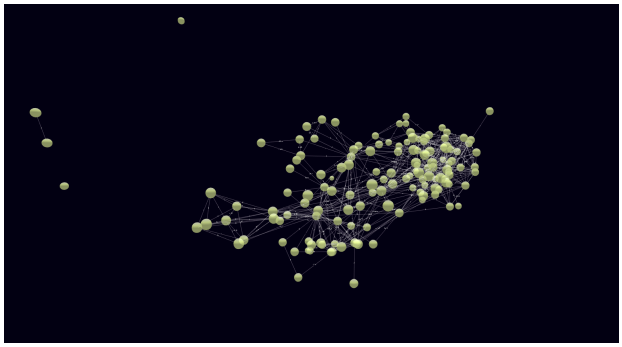
Additionally, the system allows the graph presented in the network pane to be converted into a three dimensional animated diagram for more visual ca-

pacity and intuition (see Figure 21). It is done in real-time by simply clicking on the blue button on top in the menu bar labeled "launch 3d network". The 3d graph appears in a new pop-up modal with a black background and an animation of rendering the map over few seconds. It starts with a small compact network and then expands to the entire dimension of the panel. It holds all the vertices and edges the 2d graph in the network pane holds since it is rendered in real-time. If a user deletes a vertex or edge from the 2d graph, that won't be reflected in the 3d graph.

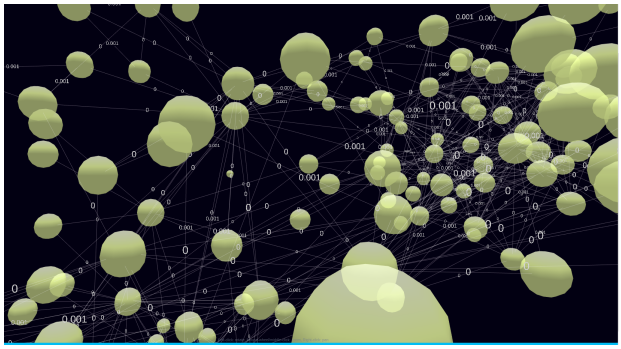
While the 2d graph, by default, shows the vertex name in the graph 3d model but is designed to show the edges' p-values. And user can get the name of the vertices by hovering the mouse pointer over the vertex. The 3d model, unlike the previous 2d model, supports a middle mouse button for zoom in and zoom out interaction.



(a) 3d modeling of the differential correlation network from zoom 0



(b) 3d modeling of the differential correlation network from zoom 30

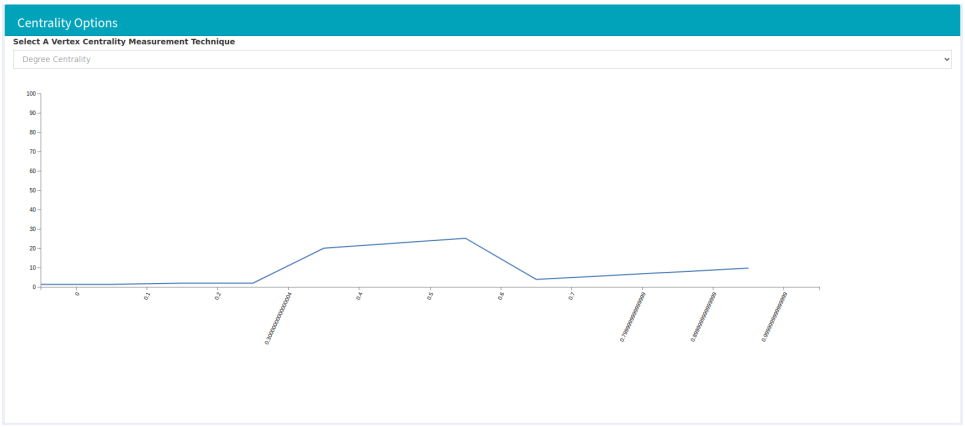


(c) 3d modeling of the differential correlation network from zoom 70

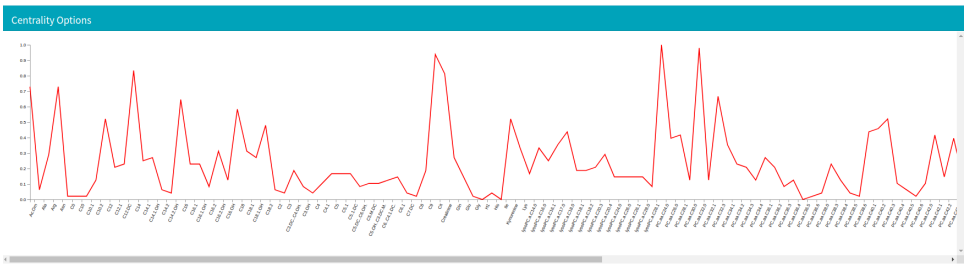
Figure 21: 3d modeling of the real time 2d network from the network pane.

Network centrality indicators are very crucial for our model. While the previous web page was designed to represent the centrality in tabular form here, we have brought it to a new look and feel. We have graphically illustrated the centralities of the vertices. On the second result page, we have two plots associated with centrality. The first plot shows the overall centrality distribution throughout the network in real-time. And the second plot, followed by the first one, shows the metabolites' actual centrality in a chart.

Both the plots are generated in real-time. Every time users make changes to the network. The charts will be different. Just like the previous one, we have incorporated all four of the centrality indicators adopted in our system. We have a similar dropdown from where users can select and observe the result. While the first plot shows the overall network centrality concentration and gives the user idea of what to expect in the graph. The second plot shows vertex-wise centrality indicators. Since it is represented graphically, it is easier to compare or see the differences among them. Figure 22 shows a representation of the two plots with degree centrality. We can see the distribution of the degree variance throughout the samples from these plots for our test dataset.



(a) Degree Centrality Distribution



(b) Degree Centrality's of the Vertices

Figure 22: two line charts providing real time information of the centrality distribution and actual real time centrality value

The final piece of information associated with the second result web page is the general network information tab attached at the very bottom of the page. It provides specific size information of the network in real-time. It is beneficial when users are dealing with an extensive network and looking to know the size. It gives the number of vertices and edges in the network in real-time. Finally, it also provides a list of all the orphan vertices in the network (a vertex with no edges connecting them).

5.2 Partial least square analysis result

In the previous chapter, we have discussed PLS in detail. Also, we have discussed how we have implemented PLS in our system. Now it's time we describe once the system processes the dataset with PLS what happens next. As we recall from the previous chapter PLS converts the dataset X and Y into a latent space. X is converted into a x score; similarly, Y is transformed into a y score. x score and y score represent the data points into projected space. Two vectors x loading and y loading are also generated by PLS. While x score and y score represent the data, x loading and y loading represent the variables in X and Y. One popular way to look into the significance or correlation of the variables is to look into the $x$loading and y loading vectors. Another perspective of looking into the variable correlations in PLS is to look into the third set of vectors X and Y weights. If we remember the equation $Y = XW$, W is what X stands for, and From $Y* = TC'$, C' is what the y wights stand for.

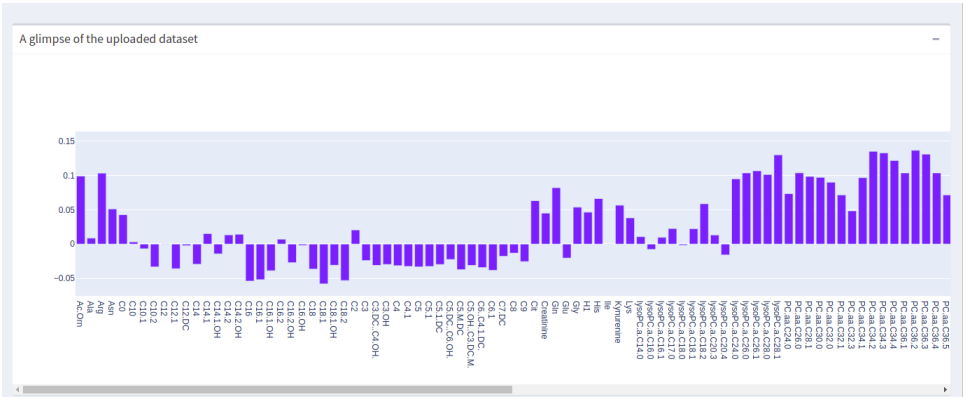
PLS results have few sets of parameters. Loading vectors (x loading and y loading) , score matrix(x score and y score), weight vector (x weight and y weight). Y weights or y loadings are relevant when there are multiple variables in Y. In our case, we will always have only one dependent Y variable. Also, we are mostly concerned about the correlation and covariance among the X variables. That is why we have chosen to ignore the Y loading and y

weights. We have focused on X loading and X weights instead.

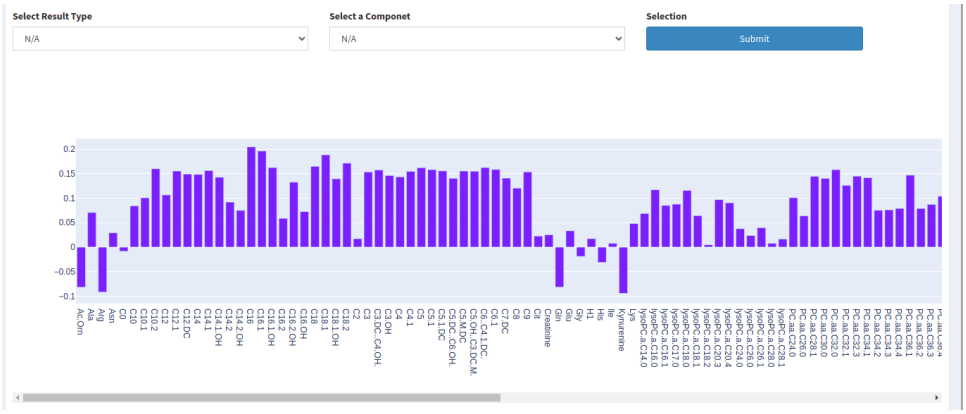
X and Y scores represent the actual data points in projected space, which could be analyzed if we were interested in the classification or prediction of Y variables. Since our primary concern is to find the relationship among the X variables only, we have chosen to ignore the score matrices. Instead, we have decided to focus on something else. In the previous chapter, we have mentioned that PLS directly ranks the X variables known as VIP values for the variables. In our system, we have demonstrated the PLS-generated VIP values. X loading and X weight.

After PLS is applied, the task manager's job status gets updated and the user gets notified by email. Like the previous differential correlation analysis job result, the user can access the result through the task manager table. Inside the PLS result page, users will see a line chart representing the variables' VIP scores. In the previous chapter, we have mentioned that PLS is applied as in component basis. Each component comes with its own set of loadings, scores, and weights. The first component captures the best-fitted latent space. Then comes the second one and then the third one. This is why we have added a menu bar that allows users to select the associated x loading and weight vector represented in the graph from each component. A demo presentation of the graphs are shown in Figure 23. We have ran

the PLS algorithm on the Osteoarthritis dataset and and generated this bar plot shown in the picture. The top bar plot represents the VIP score of the metabolites. In the bottom one the component wise loading or weight of the metabolites are generated.



(a) PLS VIP score



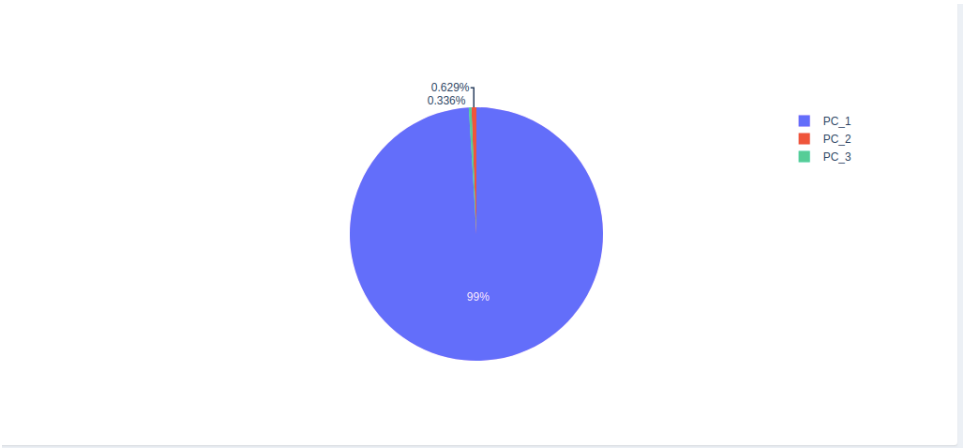
(b) PLS loading and weight chart

Figure 23: In both the bar chart the row represents the metabolites and the column represents values. (a) shows vip scrore and (b) has a dropdown option that lets user to toggle between loading and weight values.

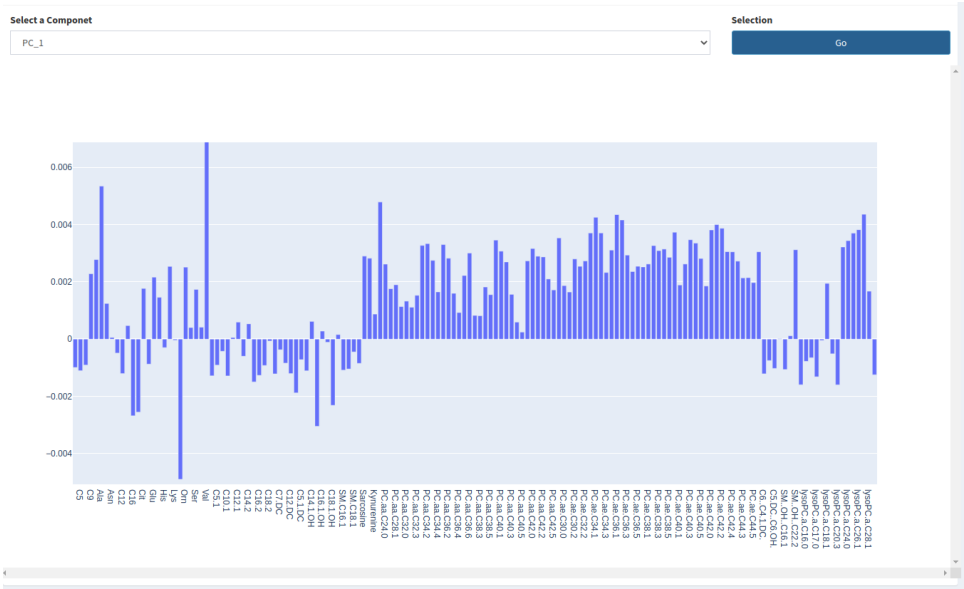
5.3 Principal component analysis result

In chapter 4, we have discussed PCA in detail, including we have implemented PCA in our system. Now we will discuss how we have assembled the results of the model, like the previous two models. Users can access the results of the PCA model through the task manager. PCA works with components, and the prior components carry more variance than the latter. The first thing that we have shown in our PCA result section is the variance covered by each component. We have implemented a pie chart that describes the variance overview of the components. Next to that pie chart, we have added a menu for the user to analyze component-wise results . Upon selecting a component, a bar chart is generated showing the relevant variables and how they contribute.

An example of the PCA model application on our dataset is shown in Figure 24. As we have run the model with three components, the pie chart shows how each part captures the variances. It can be seen the first component holds the majority variance. Later in the image, we can also see the significant metabolites and how they contribute to the variance. One additional feature that we have implemented here in the bar chart is filtering the variables that do not contribute. The bar chart filters out variables (metabolites) that are not significant in component results and do not show them in the plot.



(a) Pie Chart for PCA Component variance



(b) Bar chart for PCA component wise result

Figure 24: Principal component analysis results. (a) represents a pie chart representing the number of component and corresponding variance captured by them, (b) shows the actual value of the metabolites how they vary in selected component.

5.4 Result comparison module

Another remarkable feature that we have implemented is the result comparison module. It allows comparison of different model results. This feature provides additional validity and insights into the findings. Every job can be projected to a single platform through our system and compared to each other. The comparison can be made on a job to job basis as well as model to model basis. Not just with similar models but also within the same job with different parameters or different modeling jobs. For example: If we have a processing job done through our system with a PLS algorithm, we can use the result comparison module to compare the loading matrix with the same job's VIP score or weight matrix. We can also compare the result of that job with other jobs in the project, such as a job processed by PCA or differential correlation analysis.

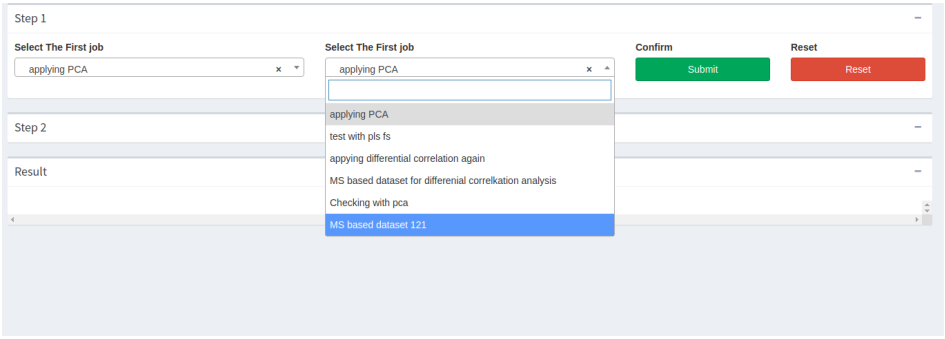
The comparison module is designed to work in three sequential steps. In the first step, the user can select two processing jobs from the drop-down list. Users can choose two jobs with the same model or different model or even two identical jobs. Table 4 demonstrates which properties of the first and second job can be compared through the platform. Once job selection is made, the second step configuration tab with additional criteria gets visible on the screen for first and second jobs. In the third step, a line chart is generated. The blue line in the diagram denotes the first job, and the red

First Job/ Second Job	Principal Component Analysis	Partial Least Square	Differenatial Correlation Analysis
Principal Component Analysis	The first job's component-wise values can be compared with the second jobs similar or different component-wise values	First Job component-wise values can be comapred with Second job's component-wise loading, weight or VIP values	First Job component-wise values can be compared with second job's centrality values
Partial Least Square	First job component-wise loading,weight or VIP valuecomparison with Second job's component-wise values	The first job's loading, weight,or VIP value comparison with second job's loading, weight or VIP value's	First Job component-wise loading, weights or VIP value can becompared to Second job centrality wise comparison
Differenatial Correlation Analysis	First Job centrality values can be compared to component-wise values from second jo	First Job's centrality values can be compared with the component-wise loading, weights, or VIP value from the second job	First Job'scentrality values can be compared to second job's centrality values

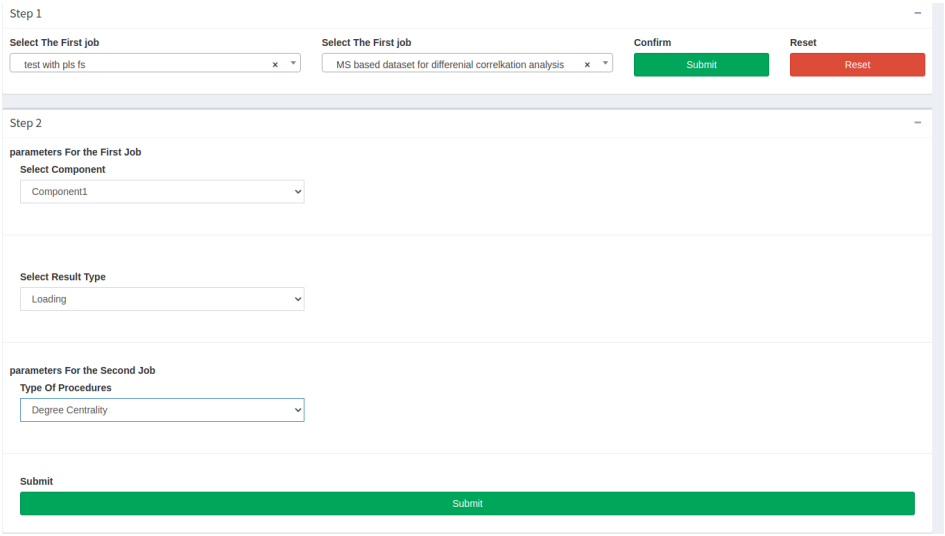
Table 4: First job vs second job result comparison critetria

one represents the second one. The chart's left side shows the scale for the first job, and the right one shows the scale for the second one. The X-axis is where the metabolites are represented, and they are the same for both. On top of the chart, there is a dropdown menu that is used for added visibility. This dropdown allows the user to flip the scale for the second job to see inverted results for more insights.

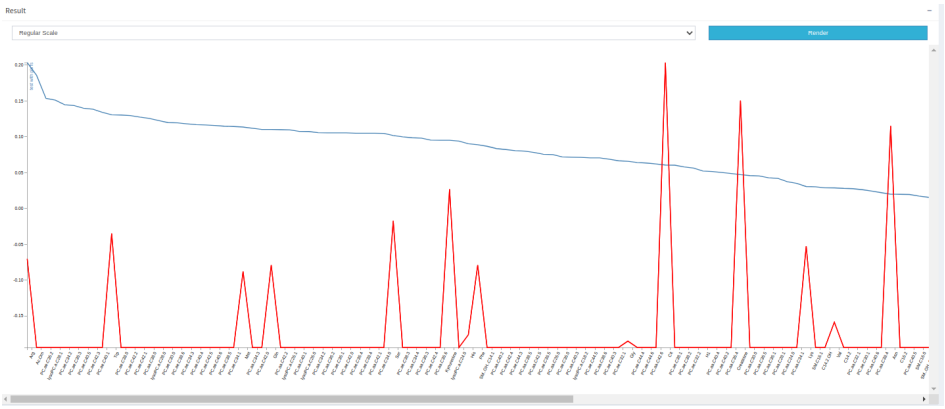
Let us describe this page feature with an example. Let's say we want to compare a result analyzed by PLS with a differential correlation analysis result. So in step one, we select PLS-based jobs from the dropdown, and in the second dropdown, we choose a differential correlation analysis-based job. Then we click submit, and a second step criterion pops up. We select the desired component and type of result for PLS, and then we choose the



(a) Step 1: select two processing job from the list for comparison



(b) Step 2: Select the parameters based on the selection from step 1



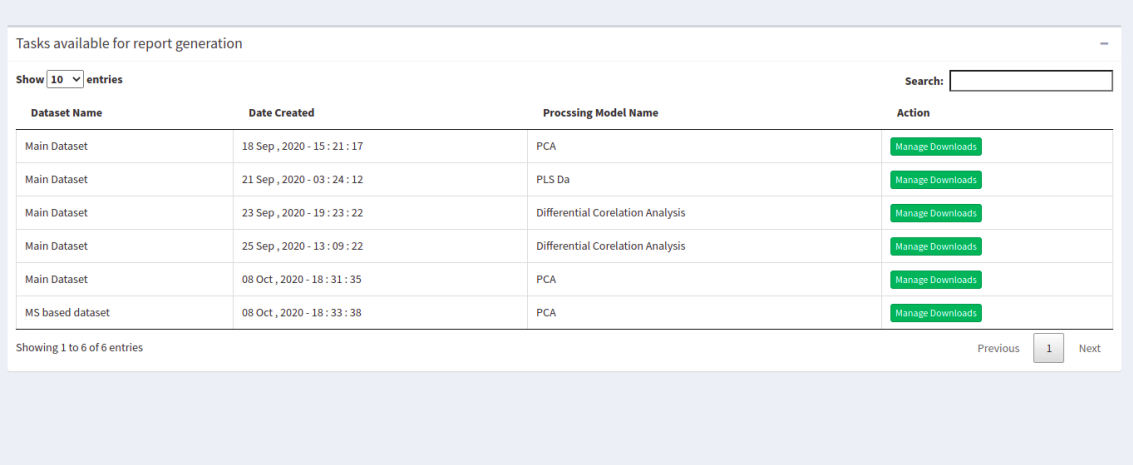
(c) Step 3: the line chart generated for both the job

Figure 25: Job comparison panel and the three steps associated to the process

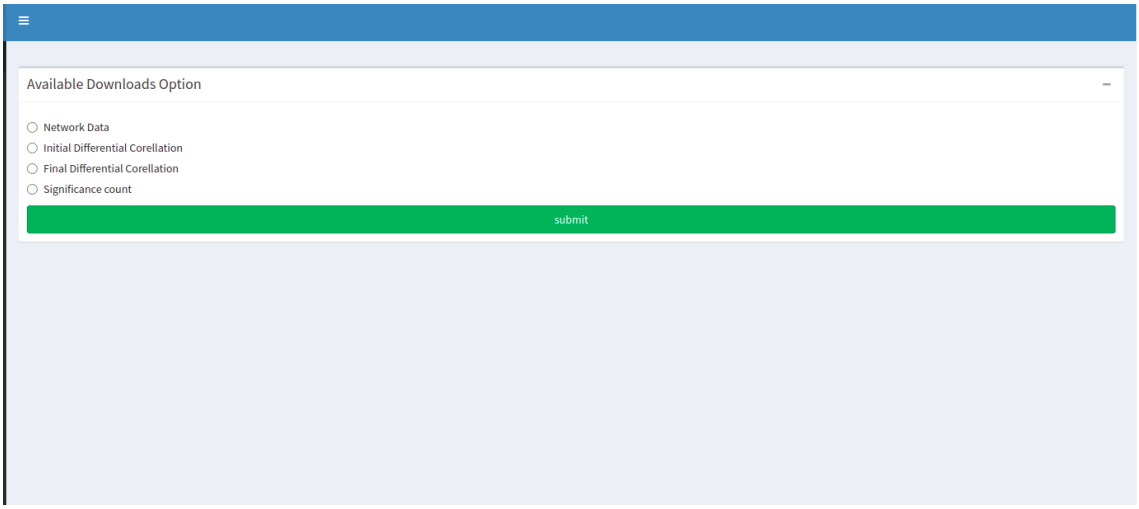
kind of centrality measurement criteria for the second job. Then once again, click submit button, and we get the line chart generated. Such an application can be seen in Figure 25 where we have used the comparison module to compare two jobs. The first one is a PLS-based job, and the second one is a differential correlation analysis job. The standard comparison criteria that match these two jobs are component-wise loadings or weights from the first job and centrality matrices from the second job. Thus, we selected our desired configurations from these two jobs and generated a plot defining their relationship or differences.

5.5 Download reports and manage project

Another vital part of the result and reports module is the download manager. Each task in our system results in deliverables stored in the database. Our system provides proper visualization and validation techniques, and it also allows users to download the deliverable for further assessment or future reference. The deliverables are sorted according to the jobs and listed under the downloads panel in a similar table to the task manager's table. Each job has different kinds of deliverables. For example, jobs that are processed with PCA lets the user download the component-wise variable results. PLS, on the other hand, enables users to download loading or weight vectors. Details list of job-wise data downloading options is presented in Table 5, and a demo representation of the download panel is shown in Figure 26.



(a) Download panel,list of processing jobs to manage downloads



(b) Download options for Differential Correlation Analysis

Figure 26: Download panel and the job wise downloadables

Type Of Processing Model	Downlodables	
Principal Component Analysis	Componenet Result	Defines the contribution of the vriables for covariance in that compnent
Partial Least Square	X weight	The model generated weights associated to the X variables
Partial Least Square	Y weight	The model generated weights for the Y variabel
Partial Least Square	VIP Score	Model ranked list of varaibles defining their significacne
Partial Least Square	X Loading	The model generated values associated with X variables for projection onto latent space
Partial Least Square	Y Loading	The model generated values associated with Y variables for projection onto latent space
Differential Correlation Analysis	Initial Correlation Matrix	the initial pair wise correlation matrix using pearson correlation analysis
Differential Correlation Analysis	Final Correlation Matrix	Corrleation matrix that is generated after applying the n number of fold permutation
Differential Correlation Analysis	Significacne Matrix	The pairwise significance matrix with the p-values used to generate the network
Differential Correlation Analysis	Network Data	Original Network data with edge, vertices, and p-value as edge weights that can be downloaded and imported to cytoscape

Table 5: Download options according to processing task

Chapter 6 : Discussion and system evaluation

We have designed this research project to eradicate the void due to the lack of data overview and processing tools in metabolomic biomarker studies. Before we started this project, we studied the literature thoroughly and understood the dire need for software solely dedicated to biomarker research. Most of the existing tools and software's are designed for multidisciplinary usage and do not explicitly encompass biomarker research capacity. In many cases, these tools are ambiguous and require technical knowledge/training. The models work as a black box, and the results or reports are hard to understand. Users need to use multiple tools throughout the study for different purposes, and a lot of time is spent managing these tasks for different phases. Similarly, most of the existing software is hard to access, standalone or requires heavy computing resources that most personal computers can not support. Our thesis has created a general outline and successfully implemented a quick and easy webserver to tackle all the problems we have just mentioned.

Our potential users are biologists, pathologists, pharmacists, data scientists, and, lastly, computer scientists. Mostly our users are individuals with heavy domain knowledge rather than having technical, statistical, or computational modeling knowledge. When it comes to getting a general overview of any dataset, form any hypothesis, or validate them, our users are mostly dependent on daunting software that works like complete black boxes. All the

modeling techniques require hundreds of parameter calibrations since they are designed for multidisciplinary usage. Our thesis has developed an entire software encompassing one task and one task only, biomarker discovery with metabolite data set. Thus we have designed our user interface and user experience with precise and minimal setup requirements. Our software is designed to be attractive. It has interactive, responsive, and modern visualization techniques that allow users to play with the system and provide more significant insights.

Our workflow is simple and effective. And the most crucial part: it bundles necessary techniques presented in one single platform. It's not like users have to use one tool for preprocessing, another tool for modeling, and the third one for result analysis. Everything has been manifested in the same project efficiently and smoothly. Our rapid accessibility protocol ensures that users do not have to waste time with redundant or unnecessary tasks like account creation or maintenance. Our unique reference key ensures that user's data and results are secured and private.

Efficiency is another issue with most other software. While other tools are designed to run on high-performing computers, we have designed our system to be equipped with low or average-performing computers. The users can avail of our service through our online portal, and our open source licensing

allows users to set up the service in user preferred live or local environment. Our singleton job scheduling technique ensures one task at a timing mechanism, which provides efficiency in low-performing servers or computers.

We have designed the system to be fast and rapid, at the same time we have not compromised with resources. With multiple preprocessing and processing techniques, we have encompassed our system to fulfill users' exploratory, dimension reduction, or advanced correlation mapping purposes. We have covered both unsupervised and supervised modeling techniques. The comparison tool allows users to play with different model outputs for more insights and validation.

Lastly, we have to mention the advanced differential correlation analysis technique, which proved to be promising in metabolomic biomarker research. While most other fundamental modeling techniques are prevalent and well adopted, this new technique needed more attention, popularity, and a rigid platform. We can undoubtedly say with our advanced graph analysis and 2D/3D interactive network visualizations, we have provided it a basis for recognition. With the model's potential and our simple, intuitive service interface, we are confident that this model will flourish and help users better understand.

6.1 Metamarker compared to other popular softwares

We have discussed our system briefly and described how it aids the void in finding biomarkers from a dataset. Now we will look into some other popular software used for similar purposes by users, and we will make a comparison of our system along with the rest.

The first popular software that we have reviewed is named MetaboAnalyst 3.0 [73]. This system was introduced in 2009 as a stand-alone software with one single module to process and apply general functional analysis on metabolomic datasets. In 2012 the second version of the software was released with only four statistical modeling techniques for functional data analysis. Even with these limited capabilities, this tool saw tremendous demand among the researchers. By 2013, the system was processing around 3200 tasks a month. in 2015, the system was redesigned and transformed into a web-based service, adding few new features due to popular demand. At that time, it was processing almost 40000 processing tasks from researchers all over the world. The software has been through various updates, and currently, it has eight different modules providing three categories of studies. The first category is exploratory studies on the dataset. The second category is functional analysis or statistical modeling of the dataset, and the third one is advanced methods for translational studies of the dataset.

MetaboAnalyst is designed to focus on comprehensive studies such as gene gene interaction network analysis or time series data analysis, pathway analysis, and more (the homepage shows the list in figure 27). Yet, these different modules cover fundamental techniques. The biomarker analysis module available in Metaboanalyst is designed mostly for classification or predictive modeling purposes. The p-value combination, vote counts or direct merging, receiver operating characteristic (ROC) curves are some of its techniques. Metamarker, on the other hand, is precisely designed for bio-marker research only and, more specifically, dimension reduction or finding significant biomarkers among the sample. It may not provide a platform for other branches of studies yet for one specific task; it is equipped with all the basic to advanced techniques. We will discuss more on the advantage and shortcomings of these two systems in Table 6.

Another popular software tool used in metabolomics study is MetaX [67], an R package designed for metabolomic dataset analysis. It was released in 2017 as a command-line tool serving comprehensive analysis like univariate and multivariate statistics, power analysis and sample size estimation, receiver operating characteristic analysis, biomarker selection, etc. It also had a web-based interface that allowed a user to apply data quality assessment and normalization method evaluation on a dataset. Some essential advantages and drawbacks of MetaX and Metamarker is shown in Table 7.

Criteria	MetaboAnalyst 3.0	Metamarker
Hosted Environment	Linux server with 16GB RAM and eight-core 2.6 GHz Processor	Linux server with 16GB RAM and quad-core 2.4 GHz Processor
Preporcissing and Outlier	Mostly focused for Raw spectral data	Varied range of preprocessing techniques availabe for different kind of data sourrce
Biomarker Analysis	provides ROC curve analysis, along with the p-value vote count. The analysis is mainly designed for new sample prediction or classification purpose	Analysis is specifically designed for biomarker identification with dimension reduction and other metabolite ranking methods. Prediction is not part of the scope, although provides processed data to be downloaded and importet for specific classification or regression purposes.
Exploration of the dataset	Provides models like PCA or PLS to explore the dataset	All three adopted models can fulfil exploratory purpose
Reports	Reports are presented as tables and 2D graphs and charts. Mostly Ven diagrams	Reports are presented in form of tables, graphs, chart, networks, csv files
Visualizations	Most of the visualizations are heatmaps generated by R package, feature details table	line charts,bar charts, pie charts, 2d Network mapping, 3d network mapping distribution plot (all interactive generated with python plotly, dash, cytoscape js, D3, and many more)
Session	User session is not saved everytime user visits the system they have to process again	Session is saved and user can revisit with reference key
Network visualization	allows for gene gene interaction network	Metabolite pair wise correlation network visualization along wtih topological analysis
Model comparison	models applied on datasets can not be compared. Allows single job processing per session only	Advanced platform to comparer tasks performd on dataset
Other branches of metabolomics	Provides varied option for other brnaches (like pathway analysis)	Limited exposure for other branches of studies apart from biomarker identification
Portability	Can be installed in local environment	Can be installed in user's local environment also

Table 6: Comparison between MetaboAnalyst and Metamarker

Criteria	Meta X	Metamarker
Hosted Environment	Initially provided web based GUI along with command line tool. Yet Web based sercvice is no longer available	Offers web-based multiple paged GUI service
Missing Value Imputation	features with 50% <missing values are imputed	features with 80% <missing values are imputed
Scaling & Transformation	Pareto scaling, vast scaling, range scaling, autoscaling and level scaling generalized logarithm (glog) and cube root transformation.	Pareto scaling, vast scaling, range scaling, level scaling, Mean Centering and Variance Scaling,Ln Transformation,
Data Quality Assesment Techniques	peak distribution and box plot, the number of missing value distribution, correlation heatmap, the metabolite m/z (or mass) distribution, the plot of m/z versus retention time, the PCA score or loading plot of all samples.	data table, missing value and data type table, correlation heatmap, centrality distribution table, Line chart, PCA score, PLS loading,Score, Weight VIP Score
Statistical Analysis	Univariate and Multivaraite, including PCA,PLS-DA, OPLS-DA, U-test,ROC curve	PCA, PLS-DA, Differential Correlation network topological analysis
Metabolite correlation network analysis	Pair wise correlation and pair wise differential correlation network.	pairwise differential correlation network
Network Visualization	2D network generated with igraph package. can be exported to file which can be added to Cytoscape and Gephi for further analysis. Network is colored according to component. doesnot provide interaction or filtering 3D network is not available	2D as well as 3D network generated with D3, Cytoscape JS. No need to explicitly export to Cytoscape for further analysis rather . It can be done directly through the system. On top of that network data is available to be downloaded for further analysis Network coloring is based on neighboring vertices. Both automated vertiex and edge-based filtering option is available. different layout options available. realtime 2D to 3D conversion available.
Network Analysis	univariate and multivariate statistical analysis methods.	Topological centrality measurement analysis methods
Model comparison	models applied on datasets can not be compared. Allows single job processing per session only	Advanced platform to comparer tasks performd on dataset
Other branches of metabolomics	Provides varied option for other brnaches (like pathway analysis)	Limited exposure for other branches of studies apart from biomarker identification
Portability	command line tool can be installed in local environment	GUI based tool can be installed in local environment
Repository and remote access	User session is not saved, Does not provide repository for user data,job and results Does not allow revisit	Offers repository to save data and results can be revisited and remote access to the repository with a secured reference key
Optimization	GUI based system is not optimised to work with low configured computers	System is optimized with singleton design pattern ensuring one task at a time

Table 7: Comparison between Metamarker and MetaX

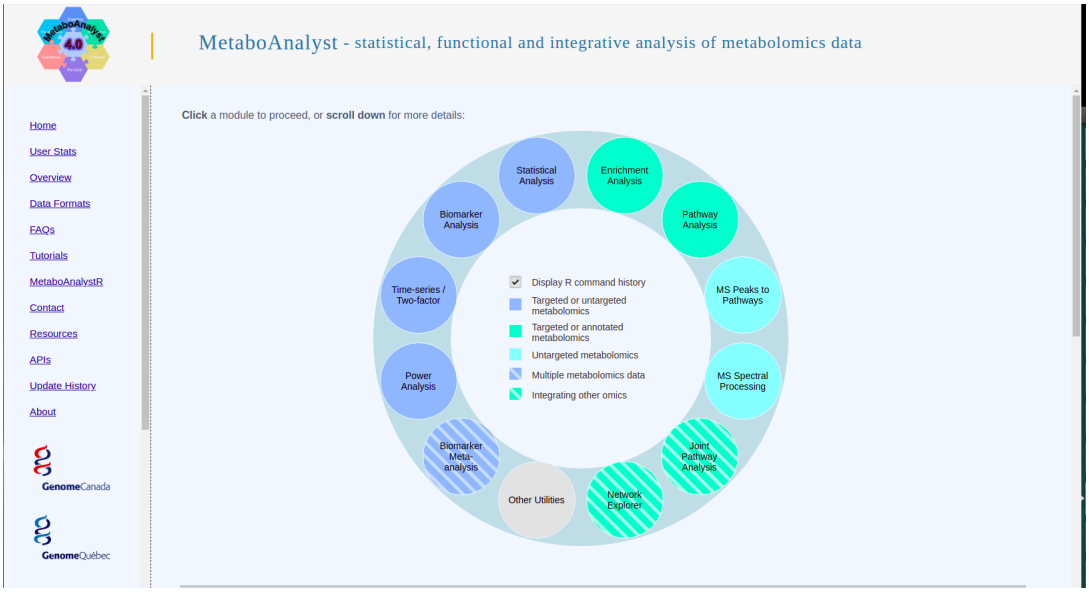


Figure 27: Landing page for MetaboAnalyst. It lists all the module/category names available to the system.

The last popular model that we have looked into for motivation is designed and implemented in the form of a standalone program [30]. It was released in 2014 and can be installed and used only on the Windows platform. It is designed with a basic workflow. Upload dataset preprocessed with existing raw peak data preprocessing technique and then apply statistical modeling to identify metabolites. Though it was intended mainly for metabolite identification however it provides comprehensive study features. For biomarker discovery, it adopted PLS and Random forest to find significant metabolite variables from the dataset.

The software MetaboNexus is programmed with R as a batch file that can

be executed in the Windows operating system. The system's performance depends on the configuration of the computer. We have described the pros and cons of these two systems compared to each other in Table 8.

Criteria	Meta X	Metamarker
Hosted Environment	Locally Windows machine highly dependent on the host configuration	Hosted online and also can be hosted local machines
Pre processing	Allows preprocessing to be done on peak lsit of molecular feature	9 different kind of processing techniques available for user preference
Statistical Analysis	Univariate analysis and PCA, PLS-DA,ROC curve and Random Forest	PCA, PLS-DA, Differential Correlation network topological analysis
Reporting and Visualization	Graphs, Score plot, Box plot, Heatmap	Bar plot, tables, line chart, csv files, 2d and 3d network graphs
Network Analysis	Does not allow network analysis	Interactive network analysis with filtering, also real time Topological centrality measurement analysis methods
Model comparison	models applied on datasets can not be compared. Allows single job processing per session only	Advanced platform to comparer tasks performd on dataset
Portability	Can be installed in windows based (minimum XP) computers only	GUI based tool can be installed in local environment independent to operating system
Repository and remote access	User session is not saved however limited exposure to pro processing since the files are saved in local machine. No direct remote access to the system	Offers online repository to save data and results can be revisited and remote access to the repository with a secured reference key is available
Optimization	Not optimised to work with low configured computers	System is optimized with singleton design pattern ensuring one task at a time

Table 8: Comparison between Metamarker and MetaboNexus

6.2 Challenges, open issues and limitations

Every efficient major scale software takes a team of programmers, architects, QA testers to work few years in assembling and the pieces together. The process works iteratively. New features are added every now and then. Con-

sider the related software MetaboAnalyst. In 2009 it was implemented with only four statistical modeling techniques. And six years of effort later, it flourished with eight different modules serving three categories of studies.

One programmer programs our Metamarker project, and it took 18 months to design, develop and test the first version of the software. This is just the beginning of unlimited potential. It will undoubtedly flourish in the future with added features and updates. Some of the drawbacks, issues, and limitations are discussed in the following section.

1. Metamarker is not designed for pathway analysis or other comprehensive studies. While most other software provides a platform for comprehensive data analysis, Metamarker allows limited exposure to that. Metamarker is wholly focused on biomarker discovery purpose and designed to aid the shortcomings of the other software.

2. Metamarker is designed for rapid study, at the same time provides an online repository to the user. Thus it is designed with very narrow user registration properties. Also, it is crucially dependent on the unique reference key to access a project. And if the key is lost, the user will lose access to data. This is one of the tradeoffs we had to consider to make the design fast and simple. Once the key is lost, the only way to get the access back is to contact the database administrator, who can generate a new key or delete the project.

3. Metamarker can only work effectively with CSV file formated metabolomic dataset where the first row has to be the variables' name. The last column has to have the group variables/class variables. This requirement is widespread with other popular software and needed to ensure efficient performance.

4. While PCA and PLS are adopted for bio-marker discovery, the result and visualizations for these two models are not as varied as the differential correlation analysis. The reason is that PCA and PLS are already well established compared to the differential correlation analysis model. We explicitly wanted to provide a solid ground to this third lesser-known model. This is one reason for our first version release; we mainly focus on designing more for the differential correlation and network analysis. In future releases, we will be adding more visualizations for PCA and PLS models also.

5. The differential correlation analysis can only support binary case-control population dataset. If it has more than two group variables, then the model will not perform accordingly. The system works assuming the variable group size to keep the interface simple.

6. The system does not allow concurrent processing tasks at a time. To ensure that our software performs efficiently, we have implemented a singleton design pattern. By default, the system enables one job at

a time; however, it can be converted into a system offering parallel processing by changing few lines of configuration in the program. We prefer the one task at a time model to ensure the lower configured computer's efficiency.

7. In case of server starvation (when there is no processing job for the job scheduler), the job scheduler module sometimes goes to a standby mode. In such situations, users have to refresh the task manager page to restart the job scheduler. We have designed it this way to save server resources.

8. As of now, the system can only process numeric data input for the processing model. The system is designed to ignore alphabetical or non-numeric data. This is one of the design requirements to ensure proper assessment of the dataset. One automated functionality is present in our system to convert non-numeric ordinal values to numeric values; however, that feature is not active in the prototype. Users with technical knowledge can download the open-source program and change some program lines to activate that feature.

Chapter 7 : Conclusion

Metabolomics is growing as a popular field of study, and it will see more advancement in the future. It has many potentials to offer, and diverse applications are possible through the study of Metabolomics. While metabolomics encompasses various tasks, we have entirely focused on one aspect in this thesis. We have focused on enriching its capacity in the field of biomarker discovery.

As a newer branch of study, metabolomics lacked advanced modeling tools and software. Most of the tools focus on a wide range of applications rather than concentrating specifically on significant bio-marker discovery from the dataset. Not to mention these tools require specific technical training and knowledge about the system to work with. Apart from usability, the system's accessibility and portability were other concerns. Not all of them are suitable to perform in different computing environments, and not all are remotely accessible. Most of the software tools are also very much dependent on computation power, and in low-configured computers, they do not perform very well. Besides, the modeling techniques used in these software tools need to be more innovative.

We have encountered all these shortcomings throughout this thesis project, then designed and developed a robust modern, and efficient system called

Metamarker. Our approach is very intuitive. The interface is designed in a straightforward and contemporary fashion, which will ensure users' comfort of eyes and always grab their attention. The tedious account registration process is eradicated rather than that rapid project-based authentication protocol has been developed. Advanced reporting and interactive visualizations have been adopted to provide more insights to the users. Also, diverse analysis and data modeling techniques have been adopted in our system. We have designed a standard workflow for biomarker discovery where users can create a project, handle outliers, run analysis and observe results. We assembled all of them in one platform and provided the ground to easily compare the processing jobs for a user to have more observatory insights. Besides, we have also ensured the system performs well in lowly configured computer environments with its one task at time scheduling techniques.

We have also worked our way by providing a solid ground for a network-based analysis model: Differential correlation analysis holds impressive potential. We have just designed the bridge for users to use this promising model on their dataset. We have also extended its capacity with advanced topological analysis, various filtering options, different distribution plots, and many more.

A complete dedicated software solution was a dire need for the biomarker

discovery prospect with Metabolomic datasets. It was challenging, and time-consuming effort was needed to fill the void. Our software named Metamarker is planned and developed considering a broad aspect of the study field and user capacity. And we are confident it will play a substantial role among the user serving data cleaning, exploratory, and dimension reduction purposes.

We wish Metamarker can be a useful tool for the metabolomics research community. All the limitations that we have addressed will be the scope of future extensions and upgrades.

References

[1] 3d-force-graph. https://github.com/vasturiano/3d-force-graph/blob/master/README.md. Accessed: 2021-03-1.

[2] Dash user guide. https://dash.plotly.com/. Accessed: 2021-03-1.

[3] Metabolomics softare list. http://metabolomicssociety.org/resource/metabolomics-software. Accessed: 2018-10-24.

[4] Metabolomics softare list online. http://pmv.org.au/metabolomics/metabolomic-software/. Accessed: 2018-10-24.

[5] three.js – javascript 3d library. https://threejs.org/. Accessed: 2021-03-1.

[6] Validating biomarkers in targeted metabolomics. https://www.news-medical.net/life-sciences/Validating-Biomarkers-in-Targeted-Metabolomics.aspx. Accessed: 2018-09-31.

[7] C. W. Beecher. The human metabolome. In *Metabolic profiling: Its role in biomarker discovery and gene function analysis*, pages 311–319. Springer, 2003.

[8] C. Z. Bo Wen, Zhanlong Mei and S. Liu. metax: a flexible and comprehensive software for processing metabolomics data. *BMC Bioinformatics*, mar 2017.

[9] R. D. Boccard J1. A consensus orthogonal partial least squares discriminant analysis (opls-da) strategy for multiblock omics data fusion. jan 2013.

[10] J. G. Bundy, M. P. Davey, and M. R. Viant. Environmental metabolomics: a critical review and future perspectives. *Metabolomics*, 5(1):3–21, 2009.

[11] P. Charles. Admin lte. `https://github.com/ColorlibHQ/AdminLTE`, 2013.

[12] M. Cuperlovic-Culf. Machine learning methods for analysis of metabolic data and metabolic pathway modeling. 2018.

[13] c. David S. Wisharta, b. Metabolomics: applications to food science and nutrition research. *Trends in Food Science and Technology*, 2008.

[14] K. Dettmer, P. A. Aronov, and B. D. Hammock. Mass spectrometry-based metabolomics. (26(1)):51 – 78, 2007. Author Manuscript.

[15] H. B. Dettmer K1, Aronov PA. Mass spectrometry-based metabolomics. 2006.

[16] A. C. Dona, M. Kyriakides, F. Scott, E. A. Shephard, D. Varshavi, K. Veselkov, and J. R. Everett. A guide to the identification of metabolites in nmr-basedmetabonomics metabolomics experiments. *Computational and Structural Biochemistry Journal*, 2016.

[17] B. S. Everitt. Unresolved problems in cluster analysis. *Biometrics*, pages 169–181, 1979.

[18] B. S. Everitt and G. Dunn. Applied multivariate data analysis. Technical report, 1991.

[19] D. A. Fell. Understanding the control of metabolism. *Biochemical Education*, 25(4), 1997.

[20] M. Franz, C. T. Lopes, G. Huck, Y. Dong, O. Sumer, and G. D. Bader. Cytoscape. js: a graph theory library for visualisation and analysis. *Bioinformatics*, 32(2):309–311, 2016.

[21] L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[22] A. Fukushima. Diffcorr: an r package to analyze and visualize differential correlations in biological networks. *Gene*, 518(1):209–214, 2013.

[23] M. A. Gillette and S. A. Carr. Protein biomarker discovery and validation: the long and uncertain path to clinical utility. *Nature Biotechnology*, 24:971–983, august 2006.

[24] P. Goldsmith, H. Fenton, G. Morris-Stiff, N. Ahmad, J. Fisher, and K. R. Prasad. Metabonomics: a useful tool for the future surgeon. *Journal of Surgical Research*, 160(1):122–132, 2010.

[25] G. G. Harrigan, R. H. LaPlante, G. N. Cosma, G. Cockerell, R. Goodacre, J. F. Maddox, J. P. Luyendyk, P. E. Ganey, and R. A. Roth. Application of high-throughput fourier-transform infrared spectroscopy in toxicology studies: contribution to a study on the development of an animal model for idiosyncratic toxicity. *toxico lett*, 2004.

[26] J. Heinemann, A. Mazurie, M. Tokmina-Lukaszewska, G. J. Beilman, and B. Bothner. Application of support vector machines to metabolomics experiments with limited replicates. mar 2014.

[27] R. P. Horgan and L. C. Kenny. 'omic'technologies: genomics, transcriptomics, proteomics and metabolomics. *The Obstetrician & Gynaecologist*, 13(3):189–195, 2011.

[28] E. Horning and M. Horning. Metabolic profiles: Gas-phase methods for analysis of metabolites. 2006.

[29] T. Hu, W. Zhang, Z. Fan, G. Sun, S. Likhodi, E. Randell, and G. Zhai. Metabolomics differential correlation network analysis of osteoarthritis. *Pacific Symposium on Biocomputing*, jan 2016.

[30] S.-M. Huang, W. Toh, P. I. Benke, C. S. Tan, and C. N. Ong. Metabonexus: An interactive platform for integrated metabolomics analysis. *Metabolomics*, 10(6):1084–1093, 2014.

[31] T. Ideker and N. J. Krogan. Differential network biology. *Molecular systems biology*, 8(1):565, 2012.

[32] J. R. Idle and F. J. Gonzalez. Metabolomics. *Cell Metabolism*, 6(5):348 – 351, 2007.

[33] I. D. W. Jeremy K. Nicholson. Understanding 'global' systems biology: Metabonomics and the continuum of metabolism. *Nature Reviews Drug Discovery*, 2(668-676), 2003.

[34] J. K. N. John C. Lindon. Analytical technologies for metabonomics and metabolomics, and multi-omic information recovery. *Trends in Analytical Chemistry*, Vol. 27(,No. 3), 2008.

[35] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065).

[36] C. E. D. Jr. Optimal algorithm for metabolomics classification and feature selection varies by dataset. dec 2014.

[37] H. C. Keun, T. M. Ebbels, H. Antti, M. E. Bollard, O. Beckonert, E. Holmes, J. C. Lindon, and J. K. Nicholson. Improved analysis of multivariate data by variable stability scaling: application to nmr-based metabolic profiling. *Analytica chimica acta*, 490(1-2):265–276, 2003.

[38] H. Kitano. Systems biology: a brief overview. *science*, 295(5560):1662–1664, 2002.

[39] H. Kitano. Computational systems biology. *Nature*, 2004.

[40] A. KLUPCZY—SKA, P. DEREZI—SKI, and Z. J. Kokot. Metabolomics in medical sciences ń trends, challenges and perspectives. *Acta poloniae pharmaceutica*, page 629, 2015.

[41] D. Koschützki and F. Schreiber. Centrality analysis methods for biological networks and their application to gene regulatory networks. *Gene regulation and systems biology*, 2:GRSB–S702, 2008.

[42] S. Mahadevan, S. L. Shah, T. J. Marrie, and C. M. Slupsky. Analysis of metabolomic data using support vector machines. 2008.

[43] J. L. Markley, R. Brüschweiler, A. S. Edison, H. R. Eghbalnia, R. Powers, D. Raftery, and D. S. Wishart. The future of nmr-based metabolomics. *Computational and Structural Biochemistry Journal*, 43:34–40, feb 2017.

[44] H. Matthews, J. Hanison, and N. Nirmalan. "omics"-informed drug and biomarker discovery: opportunities, challenges and future perspectives. *Proteomes*, 4(3):28, 2016.

[45] R. Mayeux. Biomarkers: potential uses and limitations. *NeuroRx*, 1(2):182–188, 2004.

[46] J. E. McDermott, J. Wang, H. Mitchell, B.-J. Webb-Robertson, R. Hafen, J. Ramey, and K. D. Rodland. Challenges in biomarker discovery: Combining expert insights with statistical analysis of complex omics data. *Nature Biotechnology*, pages 37–51, january 2013.

[47] W. McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[48] K. M. Mendez, D. I. Broadhurst, and S. N. Reinke. Migrating from partial least squares discriminant analysis to artificial neural networks: a comparison of functionally equivalent visualisation and feature contribution tools using jupyter notebooks. *Metabolomics*, 16(2):17, 2020.

[49] B. B. Misra and S. Mohapatra. Tools and resources for metabolomics research community: A 2017–2018 update. *Electrophoresis*, 40(2):227–246, 2019.

[50] B. B. Misra and J. J. van der Hooft. Updates in metabolomics tools and resources: 2014–2015. *Electrophoresis*, 37(1):86–110, 2016.

[51] J. K. Nicholson, J. C. Lindon, and E. Holmes. 'metabonomics': understanding the metabolic responses of living systems to pathophysiological stimuli via multivariate statistical analysis of biological nmr spectroscopic data. *Xenobiotica*, 29(11):1181–1189, 1999.

[52] S. G. Oliver, M. K. Winson, D. B. Kell, and F. Baganz. Systematic functional analysis of the yeast genome. 1998.

[53] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos. Using graph theory to analyze biological networks. *BioData mining*, 4(1):1–27, 2011.

[54] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[56] L. Puchades-Carrasco and A. Pineda-Lucena. Metabolomics applications in precision medicine: an oncological perspective. *Current topics in medicinal chemistry*, 17(24):2740–2751, 2017.

[57] S. P. Putri, Y. Nakayama, F. Matsuda, T. Uchikata, S. Kobayashi, A. Matsubara, and E. Fukusaki. Current metabolomics: Practical applications. *Journal of Bioscience and Bioengineering*, 2013.

[58] J. T. Rasmus Madsen, Torbjörn Lundstedt. Chemometrics in

metabolomics—a review in human disease diagnosis. *Analytica Chimica Acta*, 2009.

[59] S. Ren, A. A. Hinzman, E. L. Kang, R. D. Szczesniak, and L. J. Lu. Computational and statistical analysis of metabolomics data. *Metabolomics*, 2015.

[60] R. Romero, J. Espinoza, F. Gotsch, J. P. Kusanovic, L. Friel, O. Erez, S. Mazaki-Tovi, N. Than, S. Hassan, and G. Tromp. The use of high-dimensional biology (genomics, transcriptomics, proteomics, and metabolomics) to understand the preterm parturition syndrome. *BJOG: An International Journal of Obstetrics & Gynaecology*, 113:118–135, 2006.

[61] C. Schmidt. Metabolomics takes its place as latest up-and-coming "omic" science. *Journal of the National Cancer Institute*, 96(10):732–734, 2004.

[62] B. R. Simoneit. A review of current applications of mass spectrometry for biomarker/molecular tracer elucidations. *Mass Spectrometry Reviews*, 24(5):719–765, 2005.

[63] A. Smolinska, A.-C. Hauschild, R. Fijten, J. Dallinga, J. Baumbach, and F. Van Schooten. Current breathomics—a review on data preprocessing techniques and machine learning in metabolomics breath analysis. *Journal of Breath Research*, april 2014.

[64] R. Spicer, R. M. Salek, P. Moreno, D. Cañueto, and C. Steinbeck. Navigating freely-available software tools for metabolomics analysis. sep 2017.

[65] K. Strimbu and J. A. Tavel. What are biomarkers? *Current Opinion in HIV and AIDS*, 5(6):463, 2010.

[66] L. E. Svante Wold, Michael Sjöström. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58:109–130, oct 2001.

[67] B. Wen, Z. Mei, C. Zeng, and S. Liu. metax: a flexible and comprehensive software for processing metabolomics data. *BMC bioinformatics*, 18(1):183, 2017.

[68] H. Wold. Path models with latent variables: The nipals approach. In *Quantitative sociology*, pages 307–357. Elsevier, 1975.

[69] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[70] S. Wold, E. Johansson, M. Cocchi, et al. Pls: partial least squares projections to latent structures. 1993.

[71] B. Worley and R. Powers*. Multivariate analysis in metabolomics. *curr metabolomics*, mar 2013.

[72] B. Worley and R. Powers. Multivariate analysis in metabolomics. *Current Metabolomics*, 1(1):92–107, 2013.

[73] J. Xia, I. V. Sinelnikov, B. Han, and D. S. Wishart. Metaboanalyst 3.0—making metabolomics more meaningful. *Nucleic acids research*, 43(W1):W251–W257, 2015.

[74] J. Yang, X. Zhao, X. Lu, X. Lin, and G. Xu. A data preprocessing strategy for metabolomics to reduce the mask effect in data analysis. *Front. Mol. Biosci.*, feb 2015.

[75] G. Zhai, E. Aref-Eshghi, P. Rahman, H. Zhang, G. Martin, A. Furey, R. C. Green, and G. Sun. Attempt to replicate the published osteoarthritis-associated genetic variants in the newfoundland & labrador population. *Journal of Orthopedics & Rheumatology*, 1(3), 2014.

[76] A. Zhang, H. Sun, G. Yan, P. Wang, and X. Wang. Metabolomics for biomarker discovery: moving to the clinic. *BioMed research international*, Volume: 2015, Year: 2015.

[77] W. Zhang, S. Likhodii, Y. Zhang, E. Aref-Eshghi, P. E. Harper, E. Randell, R. Green, G. Martin, A. Furey, G. Sun, et al. Classification of osteoarthritis phenotypes by metabolomics analysis. *BMJ open*, 4(11), 2014.