

AUT University

**Evaluation and Improvement of Current
Computational Tools for Metabolomics Data
Analysis**

Author: SiMing Li

Primary Supervisor: Andrea C. Alfaro

Secondary Supervisor: Nikola Kasabov

A thesis submitted in fulfilment of the requirements

for the degree Master of Philosophy

in the

Faculty of Health & Environmental Sciences

Aquaculture Biotechnology Research Group

February, 2017

Abstract

Metabolomics is the latest addition to the 'Omics' approaches, which includes genomics, transcriptomics, proteomics, and metabolomics among others. These approaches promise to provide a greater insight into biological systems than has been possible with traditional hypothesis-driven methods. Metabolomics can be used to examine and analyse the organism's intermediary biochemical products or metabolites that are indicative of all life functioning. Successful application of metabolomics approaches have been shown in fields such as drug discovery, disease diagnostics, environmental sciences, forensics, agriculture and aquaculture. In the past decade, metabolomics has been expanding rapidly due to improved analytical platforms, statistical analyses, and enhanced computational capabilities. Capitalizing on these advancements, researchers have unravelled a wealth of knowledge. However, analysis of metabolomics data is still complex and challenging for new researchers trying to apply this approach to other fields, such as environmental science and aquaculture. Indeed, metabolomics data analysis relies heavily on computational tools to interpret the large multivariate and multidimensional datasets generated by high throughput platforms, such as mass spectroscopy and nuclear magnetic resonance. The inherently large size and complexity of these data sets often require advance bioinformatics and computational analyses that are not usually at the reach of researchers in applied biological fields. Thus, the aim of this thesis was to identify the bioinformatics and statistical analysis needs of metabolomics research, evaluate various bioinformatics tools already available, identify the most effective and applicable methods for metabolomics data analysis, and develop an easy-to-use computational platform to conduct statistical analyses and graphic

representation of biological data. A comprehensive review of the literature, software and databases available for metabolomics bioinformatics was performed, and each major data analysis package or platform was evaluated for its effectiveness, efficiency, user-friendly capability, and functionality. From this initial investigation, MetaCore™, Metaboanalyst, InCroMAP, 3Omics and Specmine were identified as the having the greatest application for metabolomics research: data pre-processing, statistical analysis and biological interpretation. From these, MetaboAnalyst 3.0 was found to be the most comprehensive. Specifically, this software tool has advantages in its functionality, user friendly interface and accessibility. However, it was determined that several features were missing in this tool that could enhance applicability for metabolomics researchers. Thus, the software package MetaboAnalyst 3.0 was used as a base to construct a stand-alone statistical analysis application with enhanced compatibility, functionality and visualization. The new stand-alone statistical analysis application has the ability to perform all the original MetaboAnalyst 3.0 statistical analysis with additional analysis, distributed online or offline with easy installation and initialisation.

Table of Contents

Abstract	i
Attestation to Authorship	v
Acknowledgements	vi
Chapter 1	1
1. Introduction/Literature Review	1
1.1. 'Omics' Technologies	1
1.2. Metabolomics	5
1.2.1. <i>Metabolomics strategies</i>	8
1.2.1.1. Experimental Design	8
1.2.1.2. Analytical Measurements	11
1.3. From Biological Data To Answers	14
1.3.1. <i>What is Bioinformatics</i>	14
1.3.2. <i>Bioinformatics in Metabolomics</i>	16
1.3.2.1. Primary Bioinformatics	16
1.3.2.2. Secondary Bioinformatics	20
1.3.2.2.1. <i>Statistical Analysis</i>	20
1.3.2.2.1.1 Univariate Analysis	21
1.3.2.2.1.2 Multivariate Analysis	23
1.3.2.2.2. Biomarker Discovery	28
1.3.2.2.3. Pathway and Network Analysis of Metabolomics Data	30
1.4. Metabolomics Data Analysis Softwares	33
1.5. Conclusions	38
Chapter 2	40
2.1. Aim and Objectives	40
2.1.1. Aim	40
2.1.2. Research Objectives	40
2.2. Methodology	42
2.2.1. MetaboAnalyst 3.0	42
2.2.2. MetaboAnalyst 3.0 Coding Structure and Functionality.....	45
2.2.3. Improvements to MetaboAnalyst 3.0	49
2.2.3.1. <i>Interval Plot</i>	49
2.2.3.2. <i>Interactive 3D Score Plot for PCA and PLSDA</i>	52

2.2.3.3. <i>PCA and PLSDA Means plot</i>	55
2.2.3.4. <i>HeatMap Colour Contrasts</i>	57
2.2.3.5 <i>Partial Least Square Regression (PLSR) Analysis</i>	58
2.2.4. Construction of an R Shiny Application For The Statistical Module of MetaboAnalyst 3.0	61
2.2.5. R Shiny	62
2.2.5.1. <i>Selection of R Shiny</i>	62
2.2.5.2. <i>R Shiny Application System Architecture</i>	63
2.2.6. Coding Structure of The New R Shiny Application	66
2.2.6.1. <i>Original Scripts used from MetaboAnalyst 3.0</i>	66
2.2.6.2. <i>User Interface Coding Structure</i>	66
2.2.6.2.1. User Interface Functions	67
2.2.6.3. <i>Server Component Coding Structure</i>	72
2.2.6.3.1. Server Component Functions	72
Chapter 3	76
3.1. Application Description/Results	76
3.1.1. Installation and Initialization	76
3.1.2. Description	79
3.1.3. Results	91
Chapter 4	92
4.1. Discussion.....	92
4.1. Conclusions	95
References	96

Attestation to Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: _____  _____

Date: _____ 12/6/2017 _____

Acknowledgements

Firstly, I would like to express my sincerest gratitude to my primary supervisor Prof. Andrea C. Alfaro for the continuous support of my M.Phil study and related research, for her encouragement, motivation, and patience. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my M.Phil study.

My sincere thanks also go to my secondary supervisor Prof. Nicola Kasabov and the rest of KEDRI for providing me the opportunity to gain more knowledge and widen my research from various perspectives.

Besides my supervisors, I would also like to thank the rest of my fellow research students from the Aquaculture Biotechnology Research Group, Dr. Tim Young, Thao Van Nguyen, Roffi Grandiosa, and Bill Subir Singh, for their insightful comments and assistance, but also for their constant friendly support and encouragement.

Last but not the least, I would like to thank my family: my parents for supporting me spiritually throughout writing this thesis and my life in general.

Chapter 1

Introduction/Literature Review

1.1. 'Omics' Technologies

The discovery of Deoxyribonucleic Acid (DNA) in the 1944 propelled biological sciences into a whole new realm of scientific interrogation and investigation. The knowledge gained from these endeavours in the last decade has provided more comprehensive understanding of biological systems (from genotype to phenotype) at an exponential rate. This is achieved by utilizing high throughput sequencing analytical platforms, such as mass spectrometry and nuclear magnetic resonance, coupled with bioinformatics data analysis. This revolutionary approach to study biology is referred to as 'Omics' approaches, consisting of genomics, transcriptomics, proteomics, and metabolomics, among others. Specifically, genomics studies the structure, function and expression of all the genes (genome) in an organism, while transcriptomics studies the mRNA (transcriptome) within a cell or organism. Proteomics studies the proteins (proteome), including their structure and function, within a cell/system/organism, and metabolomics studies the molecules that are intermediary or end products of metabolic reactions known as metabolites (metabolome) (Horgan & Kenny, 2011). Collectively these research fields are referred to as integrative systems biology, which is based on the idea that proteins, via mRNA, and then metabolites are synthesized in a hierarchical manner when genes are activated (Alfaro & Young, 2016) (Figure 1).

Traditional biology adopts a more targeted hypothesis driven scientific approach, wherein a clearly articulated scientific question/hypothesis is proposed.

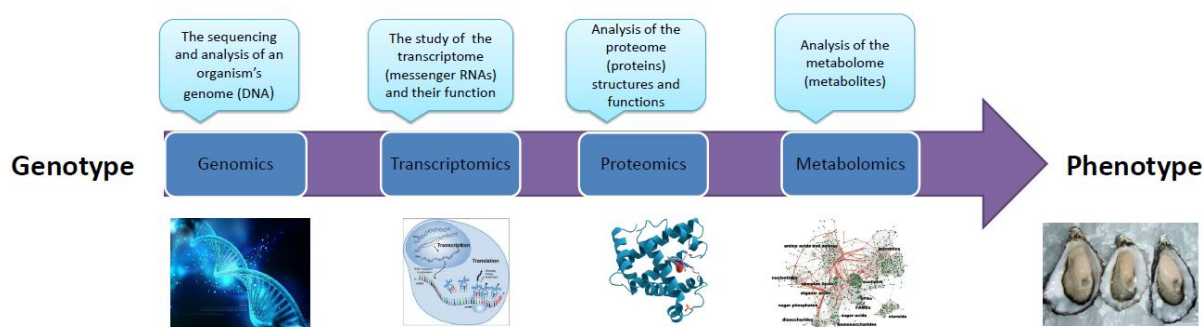


Figure 1

Diagram of the 'Omics' cascade defining genomics, transcriptomics, proteomics and metabolomics, and depicting their position along the genotype to phenotype continuum.

Subsequently experiments are carried out to obtain data in order to test the study hypothesis (Ozdemir et al., 2009). However, 'Omics' approaches allows for untargeted scientific studies. This is enabled by the rapid emergence of advanced analytical platforms, statistical methods, and computational tools. An untargeted scientific approach allows for a global analysis of an organism's genome, transcriptome, proteome and metabolome with the ultimate aim of providing us with a more comprehensive picture of the biological context. The exploratory nature of untargeted approaches has the potential to generate novel hypotheses instead of simply validating a pre-identified hypothesis. At the same time, 'Omics' provides the opportunity for unexpected information to be revealed, leading to high innovation and discovery in a very efficient manner (Young & Alfaro, 2016a).

Recently, the application of multiple 'Omics' strategies, applied simultaneously, has been adopted more frequently with great success (Horgan & Kenny, 2011).

Indeed, integrative 'Omics' have been used in many major research fields, such as pharmaceutical drug discovery (Yan et al., 2015), microbiology (Zhang, Li, & Nie, 2010), medical science (Vlaanderen et al., 2010), and environmental (Ge et al., 2013). For example, genomics and metabolomics have been used to reveal phenotype of silent mutations (Raamsdonk et al., 2001). Integration of metabolomics and proteomics has been applied in plant physiology (Weckwerth, 2008), and together transcriptomics and metabolomics have been employed to aid biomarker discovery in type 2 diabetes (Connor, Hansen, Corner, Smith, & Ryan, 2010). There are also many large scale comprehensive studies that integrate multiple 'Omics' approaches. For example, a study conducted by the medical field by Romero et al. (2006) attempted to understand the preterm parturition syndrome by using integrative 'Omics'. The study provided insightful findings in predisposing factors for preterm birth using genomics; changes in mRNA in reproductive tissues associated with preterm labour and preterm prelabour rupture of membranes using transcriptomics; identify differently expressed proteins in amniotic fluid of women with preterm labour using proteomics; and identify the metabolic footprints of women with preterm labour likely to deliver preterm and those who will deliver at term using metabolomics.

The biggest challenges with 'Omics' technologies come from data analysis. Progress in high throughput analytical platforms coupled with an expanding diversity of experimental techniques has consequently allowed for an exponential growth in biological data acquisition (Berger, Peng, & Singh, 2013). These datasets are often very large, complex, and multivariate, therefore requiring advanced statistical and computational analyses. In addition, integration of large heterogeneous datasets collected from multiple 'Omics' studies is a major challenge and fast becoming the main developmental point of

integrative systems biology in the immediate future (Gomez-Cabrero et al., 2014). Therefore, as the 'Omics' fields grow in scope and complexity, so does the need for development of more sophisticated data analyses and bioinformatics methods/tools (Boccard & Rudaz, 2014). Metabolomics is the latest addition to the 'Omics' group that can significantly benefit from these developments.

1.2. Metabolomics

Despite metabolomics being the newest member to the 'Omics' family, the study of metabolites dates back to ancient China (1500B.C-2000B.C) and ancient Egypt where urine sweetness was selected as an indicator to test for a disease known now days as diabetes. However, it was not until late 1960s, through the invention of powerful analytical platforms, such as nuclear magnetic resonance spectroscopy (NMR) and Mass spectroscopy (MS) that biologist truly began scientific studies on metabolites (Greef, Wietmarschen, Ommen, & Verheij, 2013).

Metabolites are small molecules (< 1500 Da) that are intermediary or end products of metabolic reactions (Wishart et al., 2007), and the comprehensive study of metabolites is known as metabolomics. This relatively new field has received considerable attention in the past decade and is considered as one of the most powerful 'Omics'. Metabolites, such as peptides, organic acids, lipids, sugars, and amino acids are involved in an organism's metabolism. They are responsible for many cell functions, such as energy transfer, signalling and regulation. Therefore, by profiling metabolites, we can capture a physiological snapshot of the metabolic state of an organism at a given time. This allows us to identify and understand the physiological differences between cells, tissues, organs or organisms that have been exposed to different conditions, such as environmental stress (e.g. poor water quality and pathogenic infections). In addition, we can also identify metabolite features that act as biomarkers from exposure to these stress conditions, and understand the role they play in a particular metabolic pathway (Alfaro & Young, 2016).

From a practical perspective, a metabolomics experiment is significantly easier

to conduct compared to genomics, transcriptomics and proteomics. To begin with, less time is involved in sample collection, preparation and analytical analysis, which makes metabolomics to be very cost effective. Secondly, traditional 'Omics' studies are considered to be invasive. Experiments are performed directly on a biological sample's tissues and vital body fluids, which often require killing the organism. The advantage of metabolomics is that it can be performed using non-invasive biofluids, such as plasma (Sato et al., 2012), urine (Sumner, Burgess, Snyder, Popp, & Fennell, 2010) or faeces (Ponnusamy, Choi, Kim, Lee, & Lee, 2011). Therefore, sample destruction is minimized and multiple analyses can be conducted on the same live organism, if required. This is extremely advantageous for designing a study with limited biological materials and/or performing multiple 'Omics' approaches with the aim of data integration (Alfaro & Young, 2016). Finally, a smaller number/type/class of endogenous metabolites relative to genes, mRNA and proteins (20000 to 25000 genes, 250000 to 1 million proteins, 1027 identified metabolites) allows metabolomics sampling to be applied on large sample sets at the same time, thus generating less complex data and less intensive data processing.

From a biological perspective, measuring the metabolome provides a dynamic and sensitive indicator of phenotypic changes in the organism, and the interaction between the genes, proteins, and metabolites. Being able to observe a coherent phenotypic and environmental relationship through metabolites provides information of a direct response to environmental factors without prior genome knowledge, and consequently opens up more opportunities to study species we have less knowledge about (non-model organisms). Therefore, metabolomics has significant potential applicability in primary industries, such as aquaculture. Another distinction of the metabolome is that the majority of

metabolite structures are conserved across species as opposed to genes. This feature provides metabolomics researchers with a natural way of standardizing their metabolite samples across species, hence enabling them to reapply tools and methods from multiple experiments without having to account for sample differentiation.

Currently, applications of metabolomics have assisted clinical research through drug discovery (Robertson & Frevert, 2013), toxicology (Robertson, Watkins, & Reily, 2011), and development of diagnostic tools (Nagana Gowda et al., 2008). Metabolomics has also been applied to research on diseases, such as cancer (Cambiaghi, Ferrario, & Masseroli, 2017), diabetes (Zhang, Qiu, Xu, Sun, & Wang, 2014), and cardiovascular diseases (Friedrich, 2012). Agriculture is another area that has benefited from the application of metabolomics (Yang et al., 2014). Applications of metabolomics to examine various environmental stressors on organisms have also gained significant popularity (Lankadurai, Nagato, & Simpson, 2013). Finally, food science and nutritional research utilize metabolomics to examine food components (Jacobs, Gaudier, Duynhoven, & Vaughan, 2009), food quality (Castro-Puyana & Herrero, 2013) and identify biomarkers for dietary intake (O’Gorman, Gibbons, & Brennan, 2013). It is evident that the applications of metabolomics are vast, and as our knowledge and experiences in metabolomics increase, we will be able to find more novel and innovative ways to apply it. For example, recently, metabolomics has been applied in aquaculture to investigate hatchery production (Young & Alfaro, 2014).

1.2.1. Metabolomics strategies

There are generally six steps involved in a metabolomics study: (i) experimental design, (ii) sample collection and preparation, (iii) analytical measurement and data acquisition, (iv) primary bioinformatics (data integrity checking and metabolite identifications), secondary bioinformatics involving (v) statistical analyses and (vi) biological interpretation and/or biomarker validation (Alfaro & Young, 2016) (Figure 2).

1.2.1.1. Experimental design

The fundamental experimental idea of metabolomics is measuring the effect of a treatment or treatments, such as exposure to different altered conditions (e.g., temperature, pH, oxygen, and pathogen levels) on a group of biological samples. Despite the simplicity in the question asked, noise and bias factors can cause variation in the metabolite profile. Therefore, it is crucial to follow good standard experimental design practices in order to efficiently and accurately extract the information that is most relevant to answer the question of the study, (Hendriks et al., 2011).

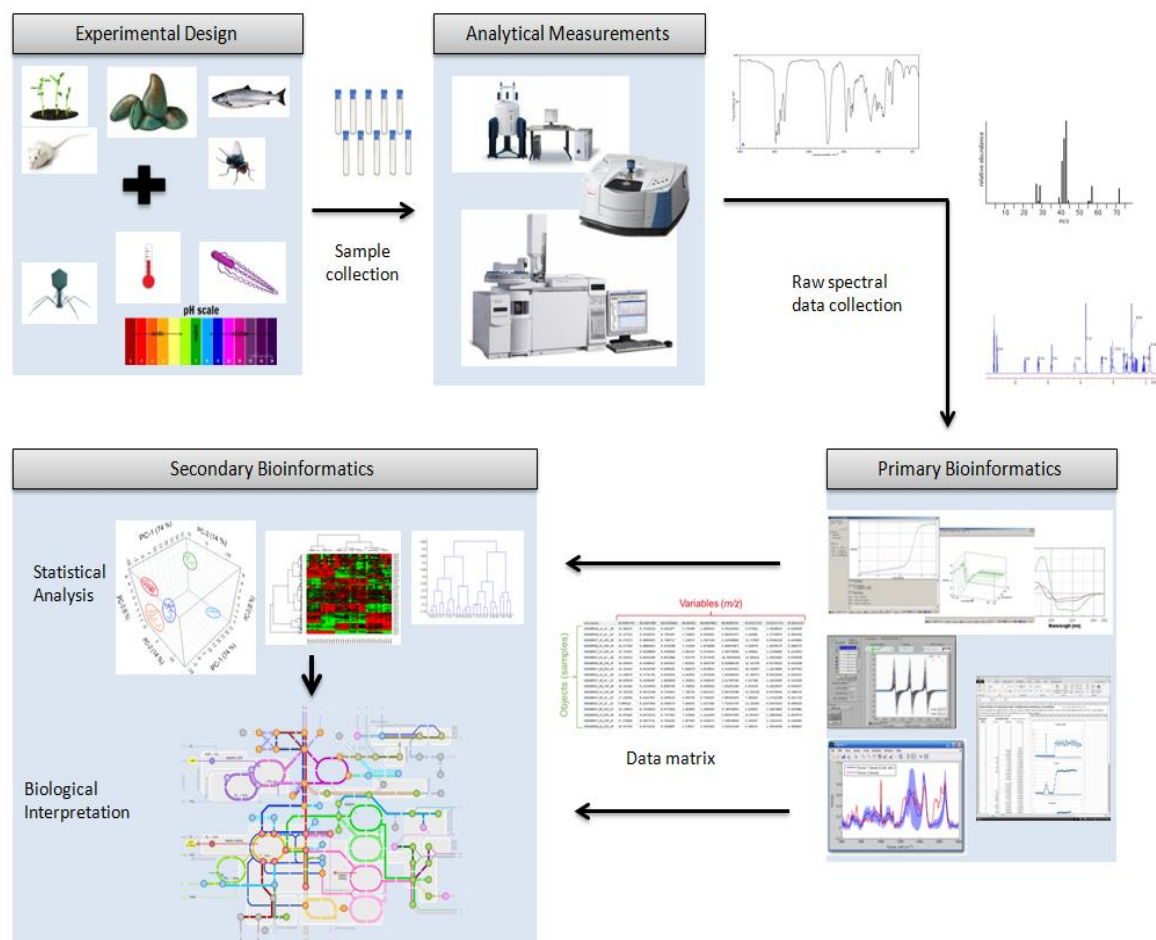


Figure 2. General Metabolomics workflow. Biological animals are treated with various conditions in the experimental design step. Samples are collected and analysed using high through-put platforms. The resulting raw spectral data are then processed using a set of primary data pre-processing software tools to generate a data matrix. The data matrix can then be used for subsequent secondary bioinformatics involving statistical analysis and biological interpretation.

Targeted Versus Untargeted Metabolomics Experimental Design

The first thing to consider when designing a metabolomics experiment is to determine the goals and hypotheses for the experiment. Depending on this, researchers may choose to adopt one of the two approaches of metabolomics: targeted or untargeted (or both for a more comprehensive analysis of the metabolome). Generally speaking, targeted metabolomics is applied when the aim of a study is to accurately determine the relative abundances and concentrations (in nM, or mg/mL) of a specific set of known metabolites (Roberts, Souza, Gerszten, & Clish, 2012). This approach allows us to directly address the hypothesis of a particular biological question. However the targeted metabolite's structure must be known and to be available in purified form (Griffiths et al., 2010; Shulaev, 2006). Hence, this method cannot be used for novel metabolite discovery where large quantities of unidentified metabolites are present in one sample. In this case, an untargeted approach is much more ideal.

Untargeted metabolomics involves the global quantification of metabolites, it captures a snapshot of the metabolism for the cells or tissues in question, providing greater insight on its biological context (Vinayavekhin & Saghatelian, 2001). With no prior knowledge or insight into what we want to find, untargeted metabolomics studies usually references/search metabolite data bases to identify significances, patterns and key distinctions in the data. It is important to note that, despite the existence of many metabolomics spectra databases, such as SMPDB (Frolkis et al., 2009; Jewison et al., 2014), KEGG (Ogata et al., 1999), MetaCyc (Caspi et al., 2010) and HumanCyc (Trupp et al., 2010), untargeted metabolomics faces a major challenge in metabolite identification.

This challenge is mostly due to instrument limitations, such as dependencies on the analytical coverage, and possible bias towards detection of the most abundant molecules. In addition, the same molecule can have different fragmentation patterns depending on specific instruments, which provide complications for metabolite data (e.g., spectral data) matching.

Sample Collection

Sample collection and preparation are outside the scope of this study. However, Álvarez-Sánchez, Priego-Capote, & Luque de Castro, (2010) provides information regarding metabolomics sample selection and reviews some practical aspects, which require consideration prior to sample preparation. For a more comprehensive information on platform specific sample preparation techniques for general biofluids and animal tissues, see Beckonert et al. (2007), Nováková & Vlčková (2009), Liebeke & Bundy (2012), Römisch-Margl et al. (2012), Vuckovic (2012) and Mushtaq, Choi, Verpoorte, & Wilson (2014).

1.2.1.2. Analytical Measurements

Once the experimental design is sound, the next step in the metabolomics workflow is selecting the correct analytical platforms to collect raw spectral data. Each platform has a distinct set of protocols applied to varying categories of samples that yield different types of spectral outputs. For example, to obtain broad metabolite coverage, including low abundance compounds, some procedures may require a tissue sample of only 2 mg wet weight, whereas others may require >100 mg (Young & Alfaro, 2016b). Types of analytical platforms currently used for metabolomics include, but not limited to, mass

spectroscopy (MS), Nuclear magnetic resonance spectroscopy (NMR) and Infra-red (IR). MS can be further enhanced by coupling with gas chromatography and/or liquid chromatography, called gas chromatography mass spectroscopy (GC-MS) and liquid chromatography mass spectroscopy (LC-MS) respectively.

NMR and MS are among the most emergent platforms in metabolomics, enabling the shortest route towards metabolite identifications and quantification. Despite each platform being very sophisticated in its own right, they do have limitations. For example, NMR is previously considered as the gold standard metabolomics largely owing to its non-destructive and non-invasive characteristics, low costs and high reproducibility. However, it suffers from low sensitivity and hinders structural identification (Emwas, 2015). MS on the other hand offers high selectivity and sensitivity, but is limited by time consuming complex sample preparation that can potentially result in metabolite loss if not carefully carried out (Lei, Huhman, & Sumner, 2011; Zhang, Sun, Wang, Han, & Wang, 2012). Therefore, it is recommended to employ a combination of different analytical platforms to gain a more informative and refined understanding of the metabolome in question. However, realistically, the choice of platform most often comes down to the analytical platforms availability in academic and commercial facilities and technical expertise (Young & Alfaro, 2016b).

Both NMR and MS produce spectral data. However, there are distinct differences between the two outputs. NMR exploits the spin properties of the atomic nucleus to measures the resonance emitted from the said nuclei. This

resonant frequency value is referred to as chemical shift (ppm). Displayed on the spectral graphs as peaks, the chemical shift can help determine the physical and chemical properties of atoms and/or molecules in which they are contained. Common standard data formats for NMR include International Union of Pure and Applied Chemistry (IUPAC) and American Standard Code for Information Interchange (ASCII).

MS capitalises on the mass to charge ratio value (m/z) of individual molecules in order to record their properties. In the case of GC-MS and LC-MS the retention time is instead measured (time taken for a solute to pass through a chromatography column of a MS instrument). The high selectivity and sensitivity properties of MS generate large amounts of data that require medium to high end computers for data storage and processing. Over the years, different manufacturers of mass spectrometers have developed various proprietary data formats for handling such data. However, this makes it difficult for academic scientists to directly manipulate the data for analysis. Many standard open formats based on the eXtensible Markup Language (XML) have been developed for MS data to address this problem. The formats include mzXML (Pedrioli et al., 2004) and mzML (Martens et al., 2011). A range of converters also exist to convert the instrument format to the standard format, and they include, but are not limited to: Hermes

(<http://www.openmath.org/meetings/bremen2003/hermes.htm>), msConvert (Holman, Tabb, & Mallick, 2014) and ReAdw (http://tools.proteomecenter.org/wiki/index.php?title=Software:ReAdW#Current_Version).

1.3. Bioinformatics: From Biological Data to Answers

Obtaining the raw data is only the first step in the metabolomics workflow. Once the data are gathered, the following analyses require extensive informatics. Indeed, to extract meaningful biological information from the thousands of metabolites quantified by modern analytic platforms presents a challenge to researchers new to metabolomics or primarily from a biological background. Like other types of 'Omics' studies, metabolomics deals with large amounts of data that are multivariate in nature, and often necessitate advanced data pre-processing, data preparation, statistical analysis, functional interpretation and in some cases integration. Additionally, with more and more data being shared, efficient ways of data retrieval, storage and matching from online data bases is also an area that demands specialized technicians and researchers. Therefore, successful modern biological studies now integrate biological knowledge, computer science, and statistical science, mathematics, and information technologies in order to enable a better understanding of the biological system. The integration of these different disciplines is achieved through bioinformatics.

1.3.1. What is Bioinformatics?

Bioinformatics is the application of computer science and information technologies to the processing and analysis of biological data. It assists biologists in three ways: data organisation, data analysis and data interpretation (Luscombe, Greenbaum, & Gerstein, 2001). Data organisation or data management aims to gather data from different sources into a databank that allows researchers to access data as well as adding new data. Additionally, standardized data formats are available to facilitate efficient computer recognition and analysis (e.g. XML [Bray, Paoli, Sperberg-McQueen, Maler, &

Yergeau, 1998]). Many biological data banks are available and they enable fast data searches, retrieval and submission. For example, EMBL (Kanz et al., 2005), Uniprot (Magrane & Consortium, 2011), and KEGG Pathway Database (Ogata et al., 1999).

Data analysis is perhaps the most challenging aspect of bioinformatics, since tools and resources need to be developed tailored towards explaining various unique forms of data. An in-depth understanding of biological science, computer science and statistical science is required to create data analysis pipelines, develop algorithms, and apply statistical methods. Numerous developed data analysis softwares have become well establish in the past ten years. To list a few: BioPerl is a comprehensive library of Perl modules available for managing and manipulating life science information (Stajich et al., 2002); Bioconductor is a collaborative creation of extensible software for computational biology and bioinformatics (Gentleman et al., 2004); BioJava is an open-source project that provides a framework for processing of biological data (Holland et al., 2008); and Galaxy is a comprehensive approach for supporting accessible, reproducible, and transparent computational research in life sciences (Goecks, Nekrutenko, Taylor, & Team, 2010).

Generating and analyzing data is only the beginning of bioinformatics and understanding and interpreting the data is the final, and perhaps the most important step. Regarding this area of bioinformatics, rapid developments have enabled global analyses of all available biological data with the aim of uncovering common principles that apply across many systems and highlight novel

features (Luscombe et al., 2001). Examples of biological data interpretation tools include KEGG (Kanehisa, Goto, Sato, Furumichi, & Tanabe, 2012), g:Profiler (Reimand et al., 2016), MESA (Xia & Wishart, 2010), and EMBL-EBI (Goujon et al., 2010).

1.3.2. Bioinformatics in Metabolomics

From the above it is evident that bioinformatics are essential in data analysis and database functionalities. Bioinformatics has also become one of the most important areas of research in metabolomics by providing tools that enables researchers to uncover information in data that was not previously achievable. These tools include a combination/workflow of computational and statistical procedures, such as identification, feature redundancy reduction, candidate biomarker selection, automation, speeding up, and pipelining workflow, deconvoluting features, and pathway mapping (Johnson, Ivanisevic, Benton, & Siuzdak, 2015). In general, these methods can be split into two major categories, primary bioinformatics and secondary bioinformatics. Primary bioinformatics processing involves analysis of raw data generated from the analytical platform and transformation. The resulting data can then be analysed through secondary bioinformatics, which involves a combination of statistical procedures and biological interpretations (Young & Alfaro, 2016b). The following section of this review explains the entire bioinformatics workflow involved in metabolomics and highlights specific techniques and challenges.

1.3.2.1. Primary Bioinformatics

Primary bioinformatics, also known as spectral processing, is concerned with processing of metabolomics raw data. This involves applying a wide range of

efficient, searching, matching and sorting algorithms in combination with heavy computational and mathematical calculations to process the raw spectral data. This can be extremely taxing on modern computers (even high end machines) if the dataset is extremely large (e.g., genetic data). As a result, many studies have attempted to produce faster, more accurate and potent computational methods of analysing raw spectral data. Despite many strides being made, there is still an urgent need of bioinformaticians in this area of research (Hendriks et al., 2011; Shulaev, 2006). Specifically, data generated from the NMR or MS platforms require intensive computational analyses, including baseline corrections, smoothing, and denoising to reduce baseline distortions and differences between samples generated by experimental and instrumental variation (Xi & Rocke, 2008; Zhang, Chen, & Liang, 2010). In addition, identification and quantification of features also known as feature identification is applied to the processed spectral data. This step involves two types of strategies that attempt to detect peaks in the spectral graphs: peak-based method and binning-based approach. Alonso, Marsal, and Julià (2015) reviewed, in great details, the feature identification techniques and strategies, their advantages and disadvantages and their applications for various outputs from analytical platforms.

Many challenges and bottlenecks still need to be overcome in spectral processing, such as overlapping of non-equivalent signals in the spectrum. This problem is present in both NMR and MS spectral data. In the case of NMR spectra, overlapping occurs when non equivalent protons align the same way against or with the applied magnetic field (from the analytical instrument) despite being chemically non-equivalent. Consequently, this results in line

broadening of the NMR spectrum (Ernst, Richard, Bodenhausen, & Wokaun, 1987; szántay, 2007). Overlapping in MS, spectra are present due to the polarity of specific molecules that enables them to pass through the column of the analytical machine at similar times despite having different masses. This causes a large number of the compounds to coelute and not completely resolve chromatographically, hence resulting in overlapping of the spectral graph (Lu, Liang, Dunn, Shen, & Kell, 2008). Overlapping makes it extremely difficult for computation algorithms to successfully identify and differentiate the structures of chemical compounds. To deal with overlapping, many softwares incorporate a mathematical technique called deconvolution (e.g., AMDIS [Vey & Voigt, 2007] , or instrument-specific software such as LECO ChromaTOF [<http://www.leco.com/products/separation-science/software-accessories/chromatof-software>]). These softwares implement many simulation based deconvolution methods using algorithms based on Bayesian and Monte Carlo, or novel algorithms aimed at predicting overlapping (Hao et al., 2014; Hefke, Schmucki, & Güntert, 2013).

In some metabolomics experiments, multiple spectra are generated from a certain sampling techniques. Therefore, it is important to match the peaks representing the same analytes for comparative analysis. However, in NMR-based studies the positions of the peaks can be affected by various chemical environmental factors, causing shifts in the spectral along the pm axis (Weljie, Newton, Mercier, Carlson, & Slupsky, 2006; Wishart, 2008). In MS-based studies, changes in stationary phase of the chromatographic column can create shifts in the spectra along the retention time axis (Burton et al., 2008; Koek, Jellema, van der Greef, Tas, & Hankemeier, 2011). These unwanted variations caused by peak shifts misrepresent and influence the quality of a study.

Therefore, spectral alignment algorithms are applied to correct this problem. They are classified into two categories: (i) spectral alignment methods where the spectral data is aligned before peak detection and (ii) peak-based alignment methods, where spectral peaks are aligned across samples once they have been detected using their coordinates (ppm in NMR, and m/z and retention time in LC/GC-MS) (Alonso et al., 2015). The specifics of these algorithms are outside the scope of this study and will not be reviewed further, but they include a range of algorithms and alignment strategies (Kazmi, Ghosh, Shin, Hill, & Grant, 2006, Nordstrom, O'Maille, Qin, & Siuzdak, 2006, Pluskal, Castillo, Villar-Briones, & Oresic, (2010) and Staab, J. M.; O'Connell, T. M.; Gomez, 2010). In addition, see He & Wang, (2010) and Vu & Laukens, (2013) for comparisons between examples of pre-existing and new alignment algorithms.

Once the processing of raw metabolomics data is complete, the data are converted into a data matrix, also known as a feature quantification matrix (FQM), which can be processed by secondary bioinformatics. This matrix is usually two dimensional and compares samples against the identified metabolite features (usually measured as concentrations). Continuous developments of software tools have helped researchers tremendously in pipelining and semi to fully-automate workflows, with the ultimate aim of simplifying the primary bioinformatics process. Many of these tools are integrated within the analytical platform itself, which will automatically perform a set of basic data pre-processing functions. Many others are free comprehensive online software, such as XCMS (Gowda et al., 2014; Smith, Want, Maille, Abagyan, & Siuzdak, 2006; Tautenhahn, Patti, Rinehart, & Siuzdak, 2012), Metaboanalyst 3.0 (Xia, Sinelnikov, Han, & Wishart, 2015b), CAMERA (Kuhl,

Tautenhahn, Böttcher, Larson, & Neumann, 2012), MetAlign (Lommen & Kools, 2012), and MZmine2 (Pluskal et al., 2010).

1.3.2.2. Secondary Bioinformatics

Metabolomics data provide countless opportunities to interpret metabolic mechanisms by analyzing hundreds to thousands of quantified metabolites. This is accomplished by statistical analysis methods and biological interpretation tools. Combined, the processes can be considered as secondary bioinformatics. These steps are less computational compared to primary bioinformatics and focus more on utilizing a combination of statistical and biological knowledge to identify potential biomarkers and their statistical significance, understand their presence and role in the biological system, and to extrapolate.

1.3.2.2.1. Statistical Analysis

Data qualities and the multivariate nature of metabolomics data must be addressed through statistical analysis in order to draw the correct conclusions (Worley & Powers, 2015). This is achieved through three goals aimed at data exploration, classification, and prediction. Each of these steps includes a set of univariate and multivariate methods. Data explorations attempts to identify trends in the data using methods, such as principle component (PCA) and cluster analysis. Classification methods, such as analysis of variance (ANOVA), partial least square discriminant analysis (PLS-DA), Orthogonal partial least square discriminant analysis (OPLS-DA), random forest, and support vector machine (SVM) aim to find differences and similarities among various groups in

the study. Finally, the relationship and predictability of the variables of interest can be determined through correlation analysis and partial least square regression (PLSR). Statistical softwares, such as R, MATLAB (matrix laboratory) and SAS (Statistical Analysis System) are comprehensive analytical environments that are very suitable to analyze metabolomics data, compared to more common statistical analysis software, such as SPSS and Excel. The former not only holds more dynamic functionalities, but also provides a coding environment that enables developments of fast, innovative, and powerful algorithms and functions through programming to significantly increase data analysis capabilities.

1.3.2.2.1. Univariate Analysis

Univariate methods involve statistical analyses of one particular feature from the data independently at a given time. These methods are relatively simple to understand and easy to interpret. To begin with, T-test and Analysis of variance (ANOVA) are among the most common univariate methods used in metabolomics studies.

The T-test, also known as Student's t-test, aims to determine whether two population means are different. In metabolomics studies, this test shows the difference in mean of an identified feature between two groups (e.g., controls vs. samples). ANOVA is similar to the t-test except in that ANOVA is applied when there are more than two groups. These tests are also split between parametric and nonparametric variants depending on the underlying statistical assumptions and consequently different types of analytical approach. When the assumed underlying distribution of the data is normal, then the normal parametric t-tests

and ANOVA is adopted. However, when dealing with unequal variances and/or non-normally distributed data, non-parametric methods, such as spearman Mann-Whitney test, Kruskal-Wallis test, Wilcoxon test and Friedman's test are preferred (Whitley & Ball, 2002). For visualization of results from t-test and ANOVA, scatter-plots are often used to display the p-value of all metabolites. This provides a convenient way to identify all the metabolites that are significantly different between samples or groups. Volcano plots are another type of scatter-plot commonly used in metabolomics studies (Garcia, García-Villalba, Garrido, Gil, & Tomás-Barberán, 2016; Perl et al., 2015; Young, Alfaro, & Villas-Bôas, 2015). Adopted from visualizing gene and protein expression data (Li, 2012), volcano plots display metabolite fold changes against the p-value, enabling quick visual identification of those data-points (e.g., metabolites) that display large magnitude changes which are also statistically significant.

Despite simplicity in its application and understanding, univariate analysis in metabolomic fails to account for potential confounding factors in the multivariate data (e.g., gender, diet, or body size). These confounding factors can introduce undesired variations that can only be exposed through multivariate statistics. Failing to accurately assess the effect of the underlying trend caused by these variations can potentially increase the possibility of obtaining false positive or negative results. In addition, there are also intricate variations between the metabolites that cannot be detected through univariate analysis. These variations can be highly important on a systems level due to the orchestrated flux of metabolites within common biochemical networks (Young & Alfaro, 2016b). To examine the presence and effects of these variations, multivariate analysis is preferred.

1.3.2.2.1.2 Multivariate Analysis

Compared to univariate methods, multivariate analysis is performed on all the metabolite features to identify relationships and patterns among them. Multivariate analyses can be divided into two sub-categories: supervised and unsupervised methods. The general idea behind supervised methods is to unravel inherent relationships in the data (e.g., distinct metabolite profiles that are strongly associated with a specific predefined response structure (Bartel, Krumsiek, & Theis, 2013)). This pre-existing relationships or associations can be modelled through constructing regression (prediction) and classification models. In unsupervised analysis methods, we attempt to analyse the dataset with little or no idea as to what the result would be. In other words, there are no inherent relationships in the data. Hence, the aim here is to model the underlying structure or distribution in the data to discover patterns and variations that can help us explain certain phenotypic observations.

Unsupervised methods

Principle component analysis (PCA) is the most widely applied unsupervised method in metabolomics studies. It is an excellent tool for detecting the largest variance between the samples and patterns between the variables. It is based on the fundamental concept of transformation where a set of possibly correlated metabolic features are transformed (i.e., orthogonal transformation) into a model consisting of a set of linearly uncorrelated variables called principle components. This model attempts to account for the maximum variance in the data by the first component, while the subsequent components explain progressively lesser amounts of variance. This allows us to identify the most significant variations in the data. The components are also independent of each other, therefore minimizing the covariance between them. The result of PCA

consists of a set of loading vectors and score vectors. The loading vectors can be plotted to summarize the variables as a mean to interpret patterns in the data. The score vectors describe the projection of each sample onto the new subspace. By plotting a 2D scatterplot of the score vectors of the first two components (since they have maximum variance) one can visually identify the global relationship among the observations (samples) (Figure.3). In some cases, 3 principle components are used to plot a 3D scatterplot in order to better visualize the separation between the samples in three dimensional spaces (Figure 3). Now days it is almost mandatory for a metabolomics statistical analysis tool to able to perform PCA and implement methods for 2D or 3D PCA visualization. In addition to statistical analysis, PCA can also be used for data quality assessment, such as outlier identifications, and biases (Alonso-herranz, Barbas, & Grace, 2015).

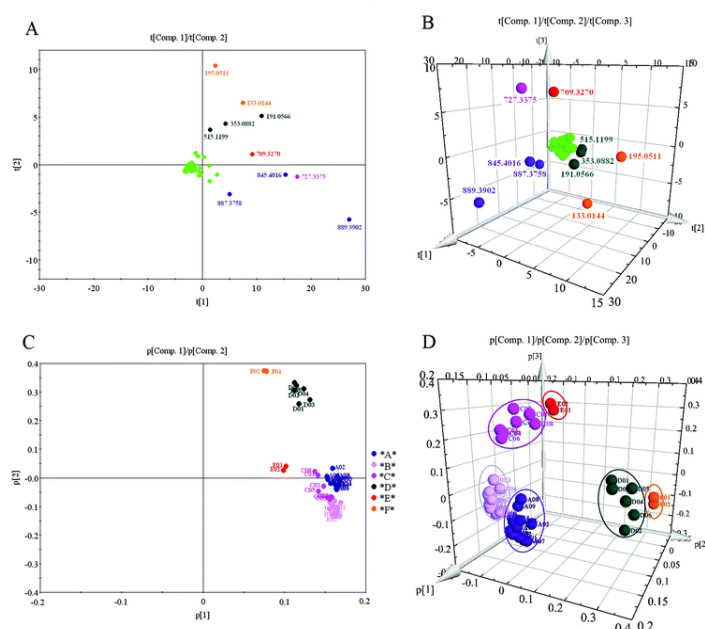


Figure 3. An example 2D PCA score plot (left) and 3D PCA plot (right).

Additional unsupervised methods used in metabolomics include hierarchical clustering analysis (HCA) and self-organizing maps (SOMs). These methods are very useful in indentifying non-linear trends in the data that are not usually exposed by PCA. Partial clustering SOMs, and specialized versions of it, allow us to visualize patterns and cluster significant features in the metabolomics profile data, as well as prioritize correlated features (Goodwin et al., 2014; He, Johnston, Zeitlinger, City, & City, 2015; Jae et al., 2007; Lloyd, Wongravee, Silwood, Grootveld, & Brereton, 2009). HCA is another powerful clustering and visualization tool that uses predefined distance measures to cluster samples based on the intrinsic similarities/dissimilarities in their measurements, irrespective of sample groupings. The results of HCA can usually be displayed by a dendrogram and heatmap (Figure 4).

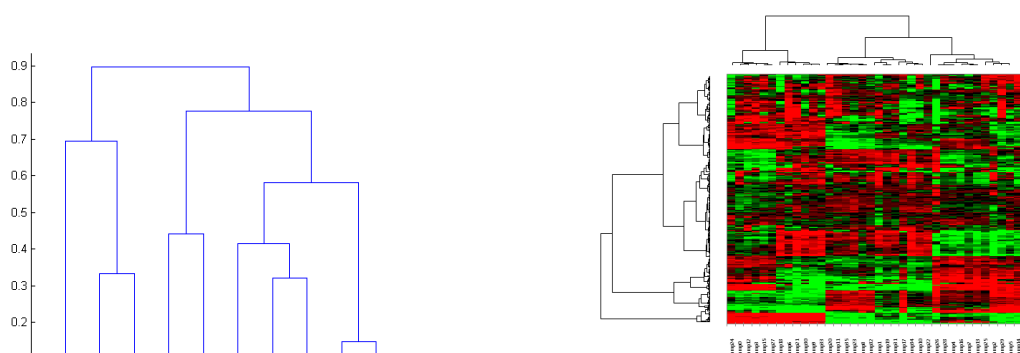


Figure 4. An example dendrogram (left) and heatmap (right)

Supervised Methods

Supervised methods aim to recognize variations/metabolic patterns that correlate with a defined classification of data points or a phenotypic variable of interest. These methods are often used to construct useful scientific

classification and prediction models. For example, we can model the variables that contribute the most the difference between samples from a control and diseased group (Jones, 2014). The most commonly applied supervised methods in metabolomics is partial least square (PLS; Abdi, 2007). There are two variants of PLS: Partial least square regression (PLSR) and partial least square discrimination analysis (PLS-DA). PLSR utilizes regression modelling to identify the relationship between a set of predicted variables and a set of observable variables (quantitative variables). PLS-DA extends PLSR to categorical variables (binary variable of interest), hence, they act as classifiers of the variables (Alonso et al., 2015).

Instead of accounting for the maximum variance in the dataset like PCA, PLS components attempt to explain the covariance between the features of interest of the metabolomics data. Consequently, this may result in metabolic features that are uncorrelated with the variable of interest, thus influencing results. To compensate for this problem, Trygg & Wold, (2002) published a method called orthogonal project to latent squares (OPLS). OPLS models separate the data variance into two components that are orthogonal to each other: the first component is correlated with the variable of interest and a second uncorrelated component. Similarly, classification models involving categorical/discrete data utilize the discriminant analysis variant OPLS-DA. Compared to PLSR, OPLS improves diagnostics, as well as producing more easily interpreted visualizations. However, compared to PLS models, OPLS only improve the interpretability, not the predictability (Trygg & Wold, 2002). In terms of interpretability, supervised methods PLS and OPLS are both better than the unsupervised PCA with regard separation power. However, PLS and OPLS can aggressively over-fit the model to the data, therefore model validations are often

a necessity (Worley & Powers, 2015). Therefore the predictability of PLS and OPLS models have shown no distinct advantage over each another despite many publications comparing the two (Tapp & Kemsley, 2009).

An area of great development in supervised metabolomics statistical analysis is creating more methods that utilize effective machine learning algorithms to build regression and classification models such as PLS. Support vector machines (SVMs) is an example of such methods commonly applied in metabolomics (Guan et al., 2009; Heinemann, Mazurie, Tokmina-Lukaszewska, Beilman, & Bothner, 2014; Lin et al., 2012) and has been argued to outperformed PLS-DA (Mahadevan, Shah, Marrie, & Slupsky, 2008). Typically, SVM performs classification tasks by constructing hyperplanes in a multidimensional space that separates two categories or classes. The construction of an SVM prediction model relies on the algorithm to “train” the model by initially assigning examples to one category or the other. The model is then able to represent the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples can then be mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. The separation can be linear, but also non-linear.

While there are many discussed supervised and unsupervised multivariate techniques to analyse metabolomics data, it is important to note that there are other procedures available which may be better suited for the analysis of specific data sets in different situations. These include: multivariate analysis of variance, linear discriminant analysis, k-nearest neighbour, k-means, random forests, and soft independent modelling of class analogies, among others. For

further information on these alternative data analysis approaches, see Steuer, Morgenthal, Weckwerth, & Selbig (2007), Liland (2011), Bartel, Krumsiek, & Theis, (2013) and Xi, Gu, Baniasadi, & Raftery (2014). Finally, research in multivariate analysis procedures will always be of crucial importance in metabolomics statistics. As mentioned previously, the complexity, sheer volume and multivariate nature of the data will only increase as metabolomics is continuously being applied in new areas in the future. This will no doubt introduce even more complicated variable relationships, underlying variances and confounding effects into the data. Therefore, bioinformatics stands as a key tool in the development of better computational and statistical tools to improve data analysis and visualization.

1.3.2.2.2. Biomarker Discovery

Statistical techniques are also necessary in metabolite biomarker discovery. Candidate biomarkers can be discovered by fitting supervised analysis methods, which in turn can be applied as an early clinical diagnostics tool (Patel & Ahmed, 2015; Wang, Zhang, & Sun, 2013), or employed to understand the mechanisms in disease, such as diabetes (Zhang, Sun, & Wang, 2013) and pathogenesis (Jung et al., 2013). However, the validity of the fitted classification and prediction model must be verified through a set of analytical steps involving model performance assessment and model validation before it can be applied in a practical setting.

A model's performance is assessed through several measurements reviewed by Alonso et al. (2015): predictive accuracy (percentage of correctly classified subjects), sensitivity (percentage of true positives that are correctly classified), and specificity (percentages of true negatives that are correctly classified).

However, these measurements are dependent on a couple of factors, such as population size and balance, the outcome prevalence and predetermined optimal decision boundary (critical biomarker concentration/score) (Xia, Broadhurst, Wilson, & Wishart, 2013). Receiver operating curve (ROC) is a type statistical graphical analysis for model performance assessment that eliminates some of the biases introduced as a result from the above mentioned dependencies/limitations. This is currently the most used method not just in metabolomics, but also in other 'Omics' approaches. A ROC curve is plotted from the true positive rate (sensitivity) against the false positive rate (specificity) at different threshold. The area under the curve (AU-ROC) and shape of the curve measures the performance of the classification model's performance. Xia et al. (2013) and Zhang et al. (2013) provided detailed information regarding the functionality and applications of ROC in metabolomics.

The results from fitted classification models on metabolomics datasets are not always accurate. Therefore, there are model validation methods to identify possible over fitting and/or instabilities (sensitive to chance/correlations) in the model (Rubingh et al., 2006). Since metabolomics deals with multivariate data, statistic resampling procedures like cross-validation, permutation, and jack-knifing are necessary. The specifics for each of these methods, such as their functionalities, performance and applications are reviewed by Rubingh et al., (2006) and Xia et al., (2013).

1.3.2.2.3. Pathway and Network Analysis of Metabolomics Data

The final component of secondary bioinformatics is to understand and interpret the results of the statistical analyses and infer the underlying biological (metabolic) mechanisms of the organism in question under given conditions. Pathway and network analyses are the two primary approaches applied in metabolomics in this regard.

In the past decade, our understanding of metabolic pathways and metabolite relationships have enabled constructions of large and comprehensive metabolite databases, such as KEGG (Okuda et al., 2008), WikiPathways (Kelder et al., 2012), MetaCyc (Caspi et al., 2008), METLIN (Smith et al., 2005). Pathway-based analysis utilizes prior information gathered in these databases through computational procedures to discover and isolate predefined metabolic pathways or biological networks that are altered in a coordinated manner in a metabolomics experiment.

There are many potent pathway analysis bioinformatics tools currently available that implement modern algorithms. PAPI is an algorithm developed and implemented in R that compares metabolic pathway activities from metabolite profiles (Aggio, Ruggiero, & Villas-bôas, 2010). Another analysis method developed by Xia & Wishart, (2010) is the metabolite set enrichment analysis (MSEA), which help researchers identifies classes of specific metabolites that are over-represented in a large set metabolites, and may have an association with various phenotypes. In addition, a web based tool: MetPA developed by Xia, Wishart, & Valencia, (2011), combines the results of MSEA analysis with pathway topological measurements to increase interpretability of the results. The results are displayed by a Google-map style network visualization system

that supports various interactive data exploration functions, as well as many additional statistical functions. Other programs, such as Paintomics (García-alcalde et al., 2011), Vanted (Lukas, Unker, & Chreiber, 2006), and Cytoscape (Smoot, Ono, Ruscheinski, Wang, & Ideker, 2011), offer alternative pathway visualization tools to MetPA. In addition, Impala (Kamburov, Cavill, Ebbels, Herwig, & Keun, 2011) and MetScape2 (Karnovsky et al., 2012) also implement different specific MSEA methods. Finally, Metaboanalyst 3.0 (Xia, Sinelnikov, Han, & Wishart, 2015a) is a powerful program that includes a highly comprehensive pathways analysis package, and incorporates a wide range of MSEA methods, as well as topological visualization tools.

Indifferent to pathway analysis, correlation-based network analysis uses the correlation pattern identified in the metabolomics data to construct metabolite pathway networks. Within the metabolomics data, correlations may exist between metabolites within a common pathway. These correlations may be caused by global perturbations, specific perturbations, or the intrinsic variability of metabolomics data (Alonso et al., 2015). As a result, metabolites that do not show significant differences among observed phenotypes or between the control and treatment may still be correlated with other metabolites. These correlation patterns are very useful in providing information regarding the underlying metabolic networks associated with a specific biological process. Currently the best way to display the results of network analysis is by constructing a correlation-based network tree (Figure 5). The nodes of the tree represent individual metabolites while the lines that connect a pair or multiple nodes describe the degree of mathematical relationship between them (e.g., solid line for positive correlation, dotted lines for negative correlation and line width to describe quantitative relationship). There are a number of software

packages available that performs correlation network analyses, such as DPCLus (Tsuji, Kurokawa, & Asahi, 2006), COVAIN (Sun & Weckwerth, 2013), 3Omics (Kuo et al., 2013) and MetaMapR (Grapov, Wanichthanarak, & Fiehn, 2015). These software packages implements various correlation identification techniques (e.g., partial correlation) and enhanced network visualization algorithms to construct more accurate networks and pathway trees and better define clusters of metabolite module.

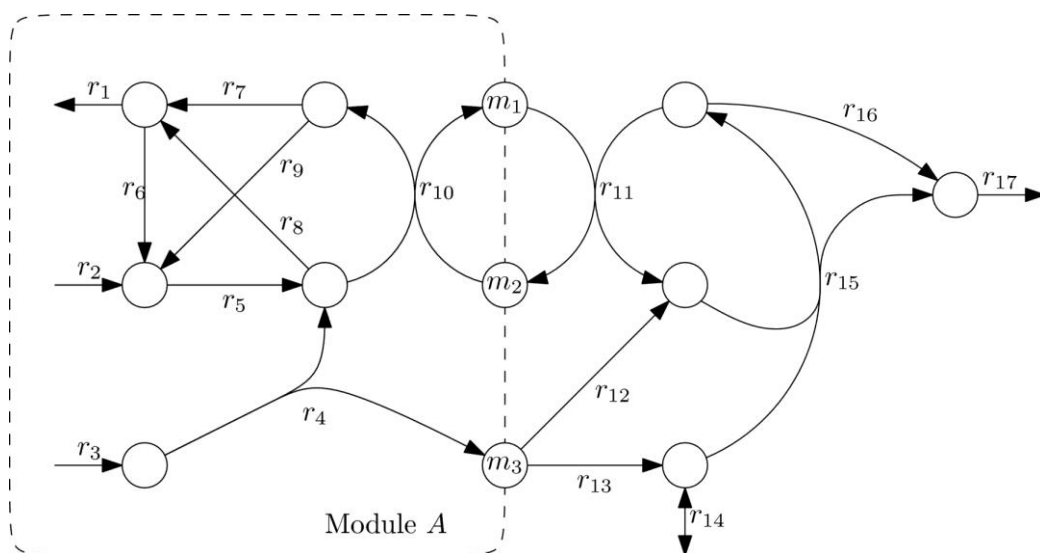


Figure 5. An example metabolite correlation network represented using a network tree (Reimers, 2015).

1.4. Metabolomics Data Analysis Software

From the previous sections it is evident that metabolomics and 'Omics' in general is no longer a subject based purely on biological knowledge. Specific knowledge and understanding is required in the informatics area involving statistics, computer science, and mathematics. The combination of interdisciplinary subjects can be overwhelming for biological researchers new to the 'Omics' fields. However, in the past decade, this gap of knowledge has been bridged to some extent by softwares and tools developed by bioinformaticians. Despite this fact, researchers may find it difficult to select the appropriate package for the data analysis of their experiment, and in some cases, multiple packages may be required. In addition, understanding the functions in each of the tools and their usage can be time consuming to learn and grasp. Therefore, it is essential to develop powerful metabolomics software tools that incorporate most of the functions/procedures described in the previous sections: (i) raw spectral data processing; (ii) statistical analysis to identify significant metabolite biomarkers; (iii) searching the metabolite databases for metabolite identification; and (iv) analysis and visualization of molecular interaction networks. In addition, these tools need to be simple, easy to use and understand. Currently, metabolomics data analysis tools range from commercial software to free to use online applications. Some also come as plugins to powerful statistical platforms, such as R and SAS. Table 1 summarises five existing and new metabolomics tools: MetaCore™ (Ekins, Nikolsky, Bugrim, Kirillov, & Nikolskaya, 2006); InCroMAP (Wrzodek, Eichner, Büchel, & Zell, 2013); MetaboAnalyst ; 3Omics (Kuo et al., 2013); and Specmine (Costa, Maraschin, & Rocha, 2015). In addition, their functionalities are compared to the basis of some of the procedures required in metabolomics data analysis. From

Table 1, we can see that each tool has distinct advantages and areas where they are lacking functionality. From the five compared programs, MetaCore™ has better integrated databases of molecular information compared to the others. MetaCore™ holds 1.3 million molecular interactions, and is being continuously updated to ensure reliability and comprehensiveness. In addition, it also implements network algorithms that analyse high-throughput data and provide interactive and informative network maps. This makes MetaCore™ the ideal tool for drug discovery, biomarker identification and clinical applications (Hohman et al., 2009; Oh et al., 2011; Ummanni et al., 2011). However, the lack of data pre-processing and statistical analysis components can complicate the data analysis procedure by forcing researchers to find other tools or implement their own methods. In addition, MetaCore™ requires a purchased license which can present a problem for researchers with limited funding.

The easy-to-use InCroMAP complements multiple analytical disciplines, therefore making it very suitable to the evaluation of systems biology (detailed experiences in bioinformatics are not necessary to use this program). InCroMAP is based on the Java programming language and provides enrichment analysis plus pathway-based visualizations for genomic, transcriptomics, proteomics, and metabolomics data (Eichner et al., 2014). Even so, InCroMAP lacks data pre-processing and statistical analysis components and presents the same problems in MetaCore™. 3Omics is similar to InCroMAP in terms of functionality, and it is a very useful tool for researchers interested in integrated visualization and one-click comparative analysis of multiple 'Omics' data in a simple and rapid way (Cambiaghi et al., 2017). Once, again like InCroMAP, 3Omics does not support data prep-processing and only incorporates a small number of statistical methods, and only supports human data evaluation.

Specmine is a newly developed package that focuses on data pre-processing and statistical analysis of metabolomics data. This package has been recently published in R and it implements a wide range of data structure manipulation functions (e.g., sub-setting and merging), spectral processing algorithms (e.g., Savitzky-Golay, baseline correction and shift corrections etc.), and statistical functions (e.g., PCA, PLSR, PLS-DA). This package is initialised in R, by importing the package into the R environment and executed in a function by function manner. Despite the comprehensive statistical and data pre-processing functionalities it brings, it lacks the network and pathway analysis that MetaCore™, InCroMAP and 3Omics possess. In addition, Specmine provides neither tutorials nor pipelines to follow. Therefore, to use the package one must have a clear understanding of each of the function's specifics and application. In addition, knowledge in R language is required to use Specmine without running into difficulties. This, once again, can be a challenge for researcher with minimal background in these computational areas.

The last tool on the market is MetaboAnalyst, which is the most comprehensive tool compared to the rest. MetaboAnalyst was developed by Xia, Psychogios, Young, & Wishart, (2009), and the package has undergone two iterations from MetaboAnalyst 2.0 (Xia, Mandal, Sinelnikov, Broadhurst, & Wishart, 2012), to the current version MetaboAnalyst 3.0 (Xia et al., 2015a). It is an integrated multifunctional free web-based tool, offering a wide range of methods that combine data pre-processing, statistical analysis, and biological interpretation. At the same time, it also provides excellent visualization and interpretation tools. MetaboAnalyst 3.0 is continuously being updated to incorporate new data analysis techniques for metabolomics (e.g. recent addition of OPLS-DA and sPLS-DA analysis). Currently, it offers eight modules: (i) statistical analysis; (ii)

Enrichment analysis; (iii) Pathway analysis; (iv) time-series/two factor design; (v) power analysis; (vi) biomarker analysis; (vii) integrated pathway analysis; and (v) other utilities. In addition, MetaboAnalyst also provide numerous tutorials and protocol papers on their website. Due to its comprehensiveness, MetaboAnalyst has experienced a 50 times growth in user traffic since its first launch in 2009, with more than 50000 jobs processed each month in MetaboAnalyst 3.0 (Xia et al., 2015a). However, no program is perfect, and the same can be seen from MetaboAnalyst. From a new metabolomics researcher's perspective, it is extremely well polished and easy to use, but it is rather static and forces users to reside with using predefined options and visualizations. Therefore, perhaps more user friendly visualization options and dynamic functionality plus more statistical analyses can further enhance MetaboAnalyst 3.0 in its current state. This is very achievable as MetaboAnalyst 3.0 is open source, and thus, presents a great opportunity for new bioinformaticians to further improve this tool, build upon the developers' foundations and further facilitates collaborative research and future development.

Table 1. Main features of the selected tools for metabolomics data analysis modified from Cambiaghi et al., (2016)

Tool	MetaCore™	Metaboanalyst	InCroMAP	3Omics	Specmine
Year	2004	2009	2011	2013	2015
Institution	GeneGo	University of Alberta, McGill University, Montreal	Center For Bioinformatics of the University of Tübingen	Molecular Design & Metabolomics Laboratory, University of Taiwan	Plant Morphogenesis & Biochemistry Laboratory, Federal University of Santa Catarina
Implementation License	Web-Based + stand-alone Commercial	Web-based GPL (GNU General Public License)	Stand-alone LGPL (GNU lesser General Public License)	Web-based	R-based GPL (GNU General Public License)
Type of knowledge	Proprietary	Public	Public	Public	Public
Input data	Gene, protein or metabolite lists imported as tab-delimited text (TXT), comma-separated values (CSV) or Excel files; gene lists from microarray analysis software (Affymetrix, Agilent)	Tab-delimited text (TXT) or comma-separated values (CSV) for concentrations, spectral bins or peak intensity data, zipped files (ZIP) of NMR or MS peak lists or of MS spectra (in NetCDF, mzXML or mzDATA format).	Tab-delimited text (TXT) or comma-separated values (CSV) of heterogeneous types of processed 'omics' data.	Comma-separated values (CSV) of processed transcriptomic, proteomic or metabolomic data.	Tab-delimited text (TXT) or comma-separated values (CSV); JCAMP Chemical Spectroscopic Data Exchange (JDX) spectra files; MS spectra (in NetCDF, mzXML or mzDATA format).
Data preparation					
Data integrity checking		✓			✓
Data normalization		✓			✓
Compound name identification	✓	✓			✓
Statistical analysis					
Univariate analysis		✓			✓
Multivariate analysis		✓			✓
Clustering		✓			✓
Classification		✓			✓
Data interpretation and integration					
Functional interpretation	✓	✓	✓	✓	
Metabolite set enrichment analysis	✓	✓	✓	✓	
Metabolic pathway analysis	✓	✓	✓	✓	
Metabolite mapping	✓		✓	✓	
Hyperlinks to external database	✓	✓	✓	✓	
Data Integration	✓	✓	✓	✓	
Output data	Networks can be exported in two formats: Netshot and Network; images as PNG files.	PDF reports containing plots, graphs and tables with all the results. Images are available as TIF or PNG files.	Tabular format (e.g. CSV) for enrichment analysis results and JPG files for pathway-based visualization	PNG, SVG or SIF formats for images.	Tabular formats (e.g. CSV) for most file output formats and R output formats for images (e.g. PNG, JPG, TIFF, BMP etc.)

1.5. Conclusions

Bioinformatics is a new discipline that arose in the past two decades to address the need to manage and interpret data generated by genomics research. This discipline represents the convergence of genomics, biotechnology and information technology, and encompasses analysis and interpretation of data, modelling of biological phenomena, and development of algorithms and statistics. As researches in transcriptomics and proteomics began to increase, bioinformatics became undeniably essential for the successful excursion of all 'Omics' approaches, especially with rapid increase in data generated by newer high-throughput technologies, innovative experimental designs, and additional fields of study to the 'Omic's banner. In spite of the large number of readily accessible tools, a big bioinformatics challenge still lies ahead in integrating multiple 'Omics' analysis to thoroughly and comprehensively evaluate experimental data in order to gain a deeper understanding of biological processes. Bioinformaticians are also consistently developing new innovative tools and methods to unravel hidden information in biological data. Indeed, intricate variations, relationships, associations, patterns and confounding effects exist within biological data that have long been overlooked. These information can potentially be very critical in answering complex biological questions, ergo applications and developments of bioinformatics is vital future biological studies.

Metabolomics is the newest member of the 'Omics' family that requires significant bioinformatics. Metabolomics data necessitates complex and multi-stepped data analysis procedures to reveal the inherent information within. Many innovative methods in these procedures are still being researched, such as more intelligent machine learning algorithms for analysing time series data,

feature detection and identification and data interpretation. The continuous bioinformatics advancements in metabolomics can significantly increase its potential to be innovatively applied as a mainstream industry tool. However, identification of unknowns, development of standardized data repositories that can be queried like the genomics resource GenBank, and integration of metabolomics with other systems-wide data are areas of metabolomics that still face many challenges and can only be solved through the collective efforts of the bioinformatics community.

Applications of bioinformatic tools currently bottleneck the biological community in metabolomics research. Indeed, this is not an easy task, considering that bioinformatics was not popularized as a disciplinary field until the last decade. Therefore, applications of bioinformatics tools can be challenging for new and traditional biologist alike. Development of easy-to-use tools and pipelines that can be accessed by researchers without in-depth knowledge in mathematics, computer science and statistics can significantly benefit the biological community. Although, many powerful tools like MetaboAnalyst have indeed eased the bioinformatics knowledge required for researchers, there are still limitations in these tools. The ideal tool for metabolomics data analysis needs to be (i) comprehensive of all components of data analysis; (ii) user friendly with minimal computational jargon, flexible and diverse options; and (iii) well pipelined with a workflow complemented by tutorials and protocol. Towards this goal, further developments and improvements of computational, visualization, and statistical techniques in metabolomics bioinformatics tools are essential and will be the focus of many bioinformaticians worldwide.

Chapter 2

Development of Metabolomics Statistical Analysis Application

2.1 Aim and objectives

2.1.1 Aim

This thesis aims to develop a user friendly metabolomics statistical analysis package that incorporates a wide range of modern data analysis methods and visualization options for inexperienced and new researchers studying in the field of metabolomics. Due to the complex nature of metabolomics data, the current selection of software and packages are either too intricate to use (require extensive knowledge in statistical and computer sciences) or limited in the number of functions. The application developed in this thesis will provide a more effective way to integrate numerous metabolomics statistical analysis methods and personalized visualization options within a highly dynamic environment that affords great flexibility for metabolomics researchers.

2.1.2 Research objectives

1. To examine one of the most popular free to use metabolomics data analysis tools (MetaboAnalyst 3.0). The source code provided by the developer was taken apart to understand the program's coding structure and functionality.
2. To identify current features and functions that are either lacking or could be potentially improved in MetaboAnalyst 3.0.

3. To implement new features, functions and improved visualizations of plots, graphs and tables using R as the primary coding language.
4. To construct a standalone dynamic GUI metabolomics statistical analysis tool that not only retains all the original statistical functions in MetaboAnalyst 3.0, but also implements the additional features mentioned in objective 3.

2.2 Methodology

2.2.1 MetaboAnalyst 3.0

MetaboAnalyst 3.0 (Xia et al., 2015) was used as the foundation of this study. Metaboanalyst is a free to use web-based metabolomics data analysis tool (see section 1.4). Version 1.0 was developed in 2009. Since then, MetaboAnalyst has gone through 2 iterations to the current version of 3.0. Indeed, the first released version of Metaboanalyst (1.0) included just two functions in data processing and statistical analysis. The second version released in 2012 (2.0) implemented two additional functions in metabolomics functional analysis and data interpretation.

Continuous updates to functions, additional new features, upgrading underlying design and framework and server hardware eventually saw a drastic rise in the tool's popularity among metabolomics researchers. The number of data analysis jobs submitted to the server has grown from ~ 800/month (in 2010) to ~3200/month (in 2013) to ~40 000/month (in 2014) (Xia et al., 2015). Due to its accessibility and comprehensiveness, the developers eventually released a 3.0 version with further enhancements to the 2.0 version in 2015. Some of these latest enhancements include: (1) re-implemented web framework; (2) consolidated interface with substantially improved graphical outputs; (3) updates to MetaboAnalyst's compound library and metabolic pathways library based on the latest versions of HMDB, SMPDB and KEGG; (4) new biomarker analysis module; (5) A new module to support sample size estimation and power analysis; and finally (6) a module for integrated pathway analysis for combining results from transcriptomic and metabolomics studies. The current

version of MetaboAnalyst (3.0) is relatively easy to use and covers a good number of steps required in a typical metabolomics data analysis pipeline.

Coded in R and Java, Metaboanalyst offers a wide range of data process, statistical and interpretation functions. A flow chart describing the overall design, structure and functional modules for MetaboAnalyst 3.0 is given in Figure 6.

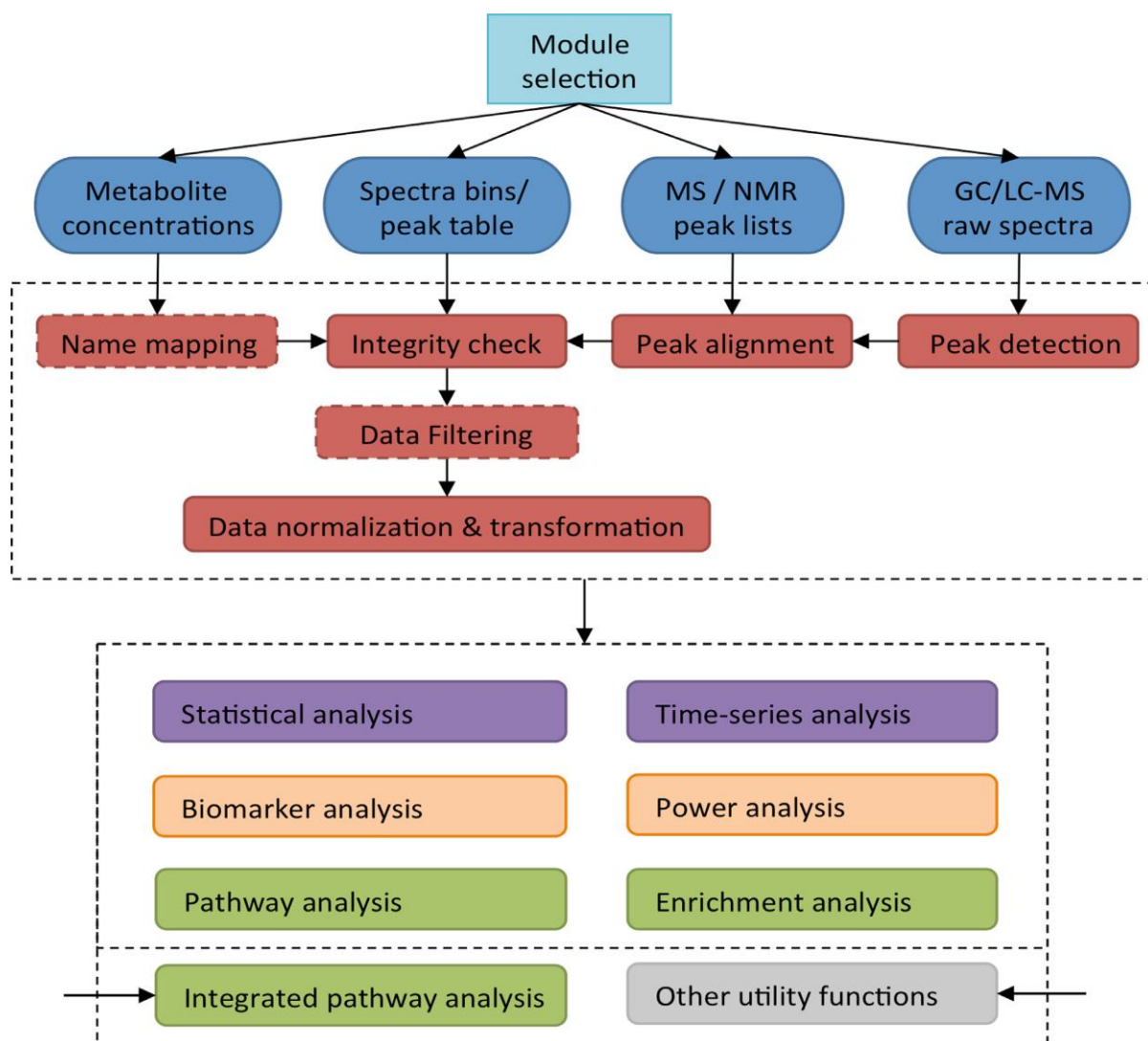


Figure 6. A MetaboAnalyst 3.0 Flowchart taken from Xia et al., (2015). This figure illustrates the general logic and data processing pipeline behind MetaboAnalyst. Different functions will be applied to process different types of data into matrices. The red boxes with dashed boundaries indicate functions that are only triggered in certain data analysis scenarios. After data integrity checks and normalization steps have been completed, downstream statistical analyses (purple box), functional analyses (green box) or advanced analyses for translational studies (orange box) can be applied. Note that different inputs are required for integrated pathway analysis and for invoking some of the general utility functions

2.2.2 MetaboAnalyst 3.0 coding structure and functionality

Metaboanalyst is constructed from two programming languages: Java and R. Java codes are implemented to handle the web interface design and communications between the server and user. However, all the actual functionalities working behind the screen of the user are performed through scripts written in R and executed by the R engine from within the MetaboAnalyst server. These R scripts consist of codes written by the developers and pre-existing R codes from various R packages developed through the collective efforts of R community.

Data uploaded to MetaboAnalyst must meet numerous format requirements for R to recognise and process. These requirements include but not limited to: numbers of replicates per sample, character formats for sample and feature labels, and data value specifics etc. (refer to <http://www.metaboanalyst.ca/faces/home.xhtml> for a complete list of the requirements in data formats). The data uploaded by the user will then be received by the MetaboAnalyst 3.0 server, prompting R to execute various scripts and the functions.

The coding structure and functionalities of the MetaboAnalyst 3.0 statistical module was closely examined in this thesis. The statistical package initialises the R engine and executes the R functions when the user uploads a dataset. The first function executed is “`InitDataObjects()`” (or “`Read.PeakList()`” and “`Read.MSspec()`” depending on the types of data uploaded). This function constructs an empty R data matrix object. In addition, a number of empty

variables are assigned to this data matrix such as: data type, data formats, data values, sample names, and sample numbers etc.

```
InitDataObjects <- function(dataType, analType, paired=F) {  
  dataSet <- list();  
  dataSet$type <- dataType;  
  dataSet$design.type <- "regular";  
  dataSet$cls.type <- "disc";  
  dataSet$format <- "rowu";  
  dataSet$paired <- paired;  
  analSet <- list();  
  analSet$type <- analType;  
  imgSet <- list();  
  ...  
}
```

After the data empty matrix data object has been successfully been initialised, the user data can then be taken apart based on its properties and assigned to the empty data matrix so that the empty variables previously created will now be occupied and describes every property of the users data. This data matrix now acts as the foundation upon which other functions (data analysis) are able to calculate and manipulate it.

Before data analysis can begin, a number of processing steps are applied. These steps involve functions that deal with the data's integrity, such as missing values. For example: "ReplaceMin()" replaces zero/missing values by half of the minimum positive values; "RemoveMissingPercent()" remove variable with over certain percentage values that are missing; various functions that checks the data integrity for peak lists and mass spectras. Lastly, the "SanityCheckData()" function polishes up the data matrix, checks to see if the data matrix meets all the requirements and confirms the user of all the properties of their data. The data matrix can now proceed to the statistical analysis step.

Data analyses (statistical) that are applied on the data matrix simply consist of various metabolomics mathematical, statistical calculations and manipulations on the various properties of the data matrix. For example, a typical step in metabolomics data analysis is data normalization. In MetaboAnalyst's server side this is performed by the R function "`Normalization()`".

```
Normalization<-function(rowNorm,      transNorm,      scaleNorm,  
ref=NULL, ratio=FALSE, ratioNum=20)
```

This function consists of six parameters (`rowNorm`, `transNorm`, `scaleNorm`, `ref`, `ratio`, and `ratioNum`) that will prompt the function itself to perform specific normalization methods on the original data values of the data matrix. These parameters are decided by selecting various options through the web interface from the user side. Once the function completes the statistical calculation, the graphical function "`PlotNormSum()`" will be executed to generate a side by side plot showing the distribution of the original data versus post normalization (Figure 7).

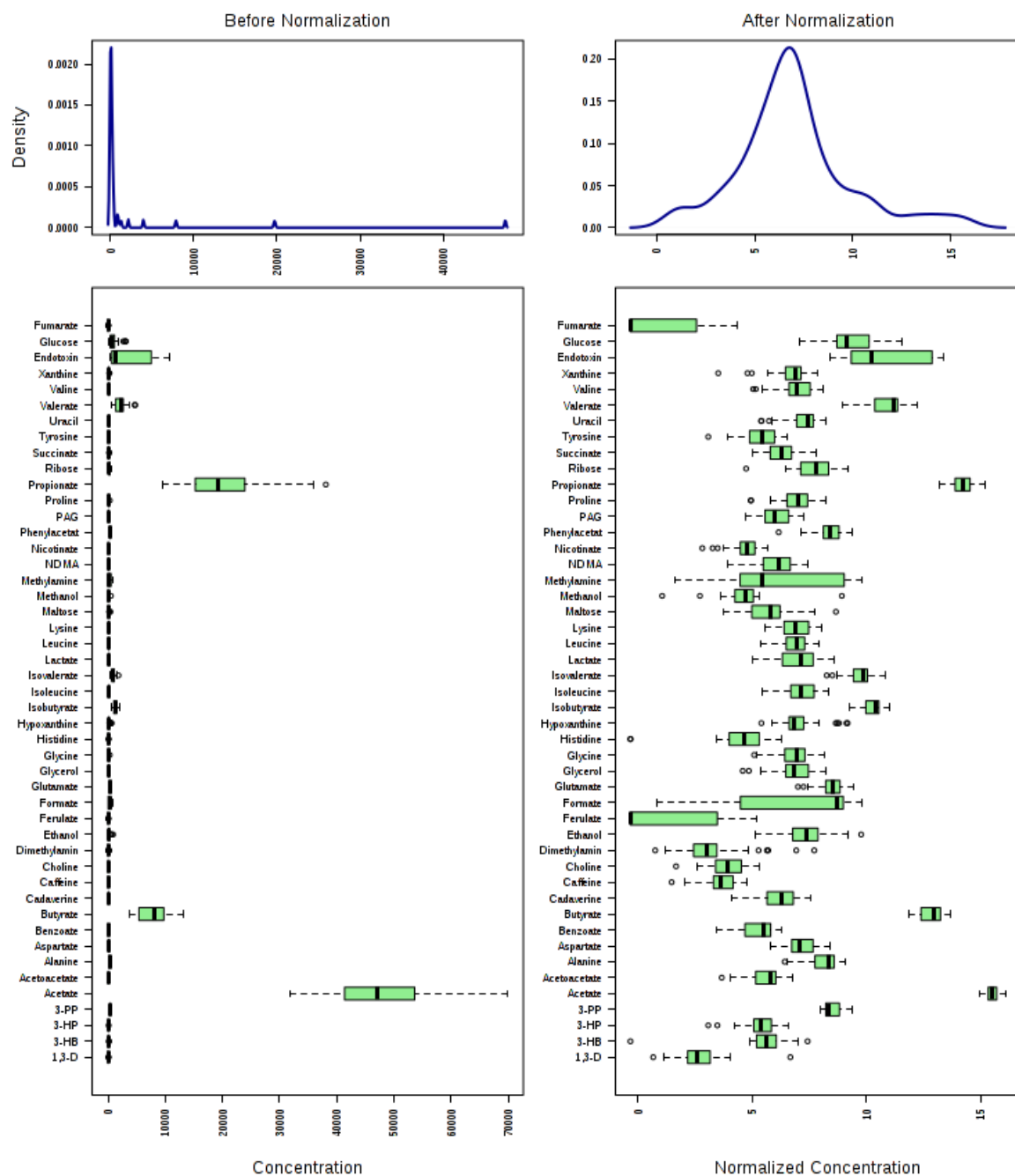


Figure 7. A plot comparing the distribution of the uploaded user data before and after normalization. The plot was constructed from the R function “`PlotNormSum()`” that is executed on the server side.

2.2.3 Improvements to MetaboAnalyst 3.0

After closer examinations of MetaboAnalyst's functionality and coding structure, it was apparent that MetaboAnalyst lacks certain features and functions that would be useful to metabolomics researchers. Therefore, numerous functions were created to complement the disadvantages of Metaboanalyst 3.0. The scripts for these additional functions were written in R studio using R Version 3.2.5 and are provided in a USB complemented with the thesis.

2.2.3.1 Interval Plot

The original MetaboAnalyst contains a function that compares the statistics of an individual compound between the different groups. The result of this comparison is displayed using a box and whiskers plot (Figure 8). However, this does not meet publication standards in some cases as box plot is not a good way to represent results when there are less than 5 groups of samples within the data. In the case of a data set having less than 5 samples groups an interval plot is preferred.

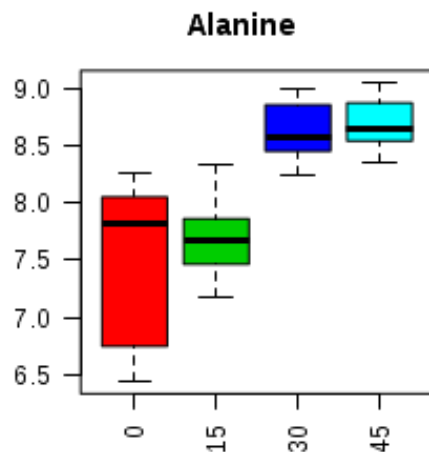


Figure 8. A box plot from *MetaboAnalyst* constructed using an example dataset provided by *MetaboAnalyst 3.0* (*cow_diet.csv*). The plot displays the normalised values of Alanine between 4 separate groups (0, 14, 30, and 40).

A new function, “`IntervalPlot()`” was implemented in R that utilises the “`ggplot2`” package to create interval plots. The plot will display the mean, upper confidence interval (Mean + the standard error) and the lower confidence interval (mean – the standard error) (Figure 9). The option to switch the statistics to standard deviation can also be selected. In addition, an option to change the colours of the plot was also implemented. The algorithm used to create this plot is as follows

```
plot=ggplot(dfp, aes(x=dfp$groups, y=dfp$means,
group=dfp$groups, color=dfp$groups))+
  theme_bw()+
  theme(panel.grid.major = element_blank(),
panel.grid.minor=element_blank(),panel.background=element_b
lank(),
axis.line = element_line(size = 0.3,colour =
"black"),
```

```
axis.text=element_text(size=12,colour="black"),axis.title=element_text(size=14,face="bold"))+
  geom_errorbar(aes(ymin=SE.dn, ymax=SE.up), width=.2)
+
  geom_point(size=3.5) + scale_colour_manual(name =
"Groups",values=colors) +
  xlab(" ") + ylab(" ") + ggtitle(cmpName) + list()
```

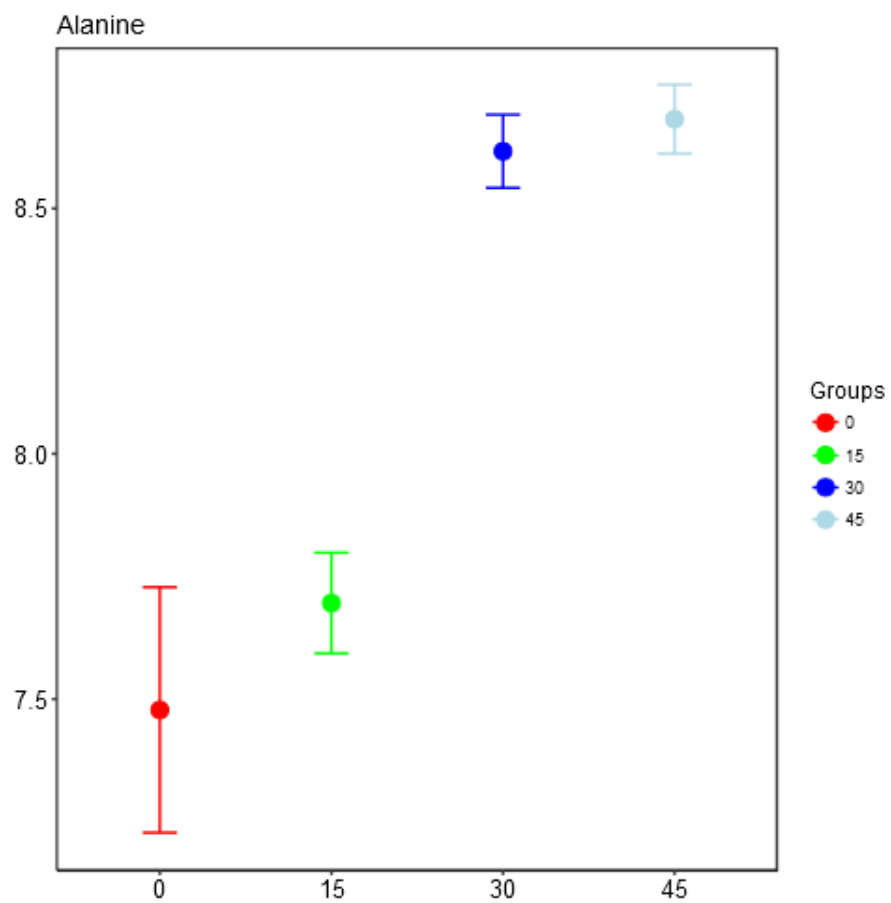


Figure 9. Interval plot created from the “ggplot2” package in R displaying normalised values of Alanine between 4 separate groups (0, 14, 30, and 40). The dataset used is the cow_diet.csv.

2.2.3.2 Interactive 3D score plot for PCA and PLS-DA

MetaboAnalyst 3.0 has a function that outputs an interactive 3D score plot for PCA and PLSDA analysis as seen in Figure 10. This is accomplished through the R function “`PlotPCA3DScore()`” which creates an .json file that in turn is used to construct the interactive plot. JSON (JavaScript Object Notation) is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML. The JSON format is used in MetaboAnalyst because R cannot directly produce a web-based interactive plot through its engine. Therefore, JSON acts as an intermediary format of conveying information between user and server to construct the plot. Although the 3D interactive score plot from MetaboAnalyst 3.0 displays the scores and colour codes the different groups, it is relatively static and limited in its features. There are no further ways to customize the plot to alter colour, point size, nor adding 95% confidence ellipses, etc.

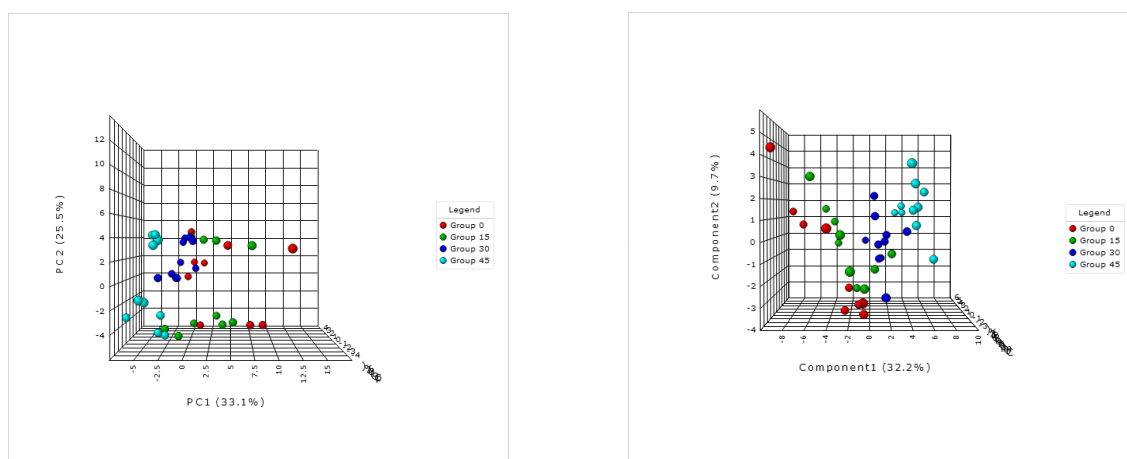
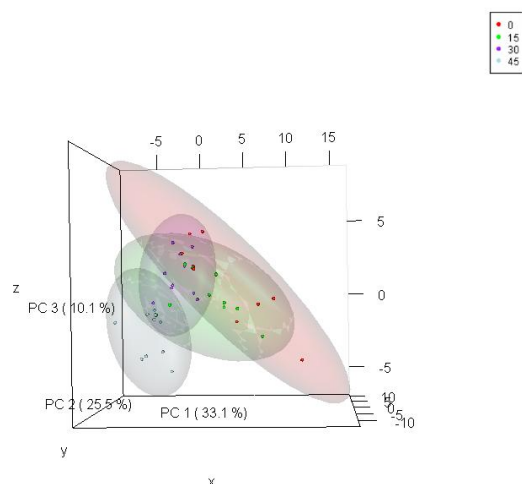


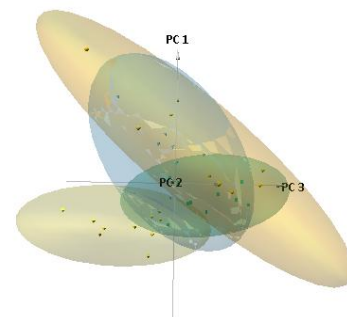
Figure 10. An interactive 3D PCA score plot (left) and an interactive 3D PLSDA score plot (right) constructed in MetaboAnalyst 3.0 (dataset used is *cow_diet.csv*).

A new and improved 3D score plot was implemented in R to mend the disadvantages in the original plot from MetaboAnalyst 3.0. The new functions `"Graphs3DPCA()"` and `"Graphs3DPLSDA()"` utilises the R packages `"plot3d"` and `"R Shiny"` to create interactive 3D scatter plots that enables the user to select multiple graphics options for enhanced plot visualization. These options are: point size, ellipses, ellipses transparency, title input, colours, and an option to add background grid (Figure 11). In addition, an alternate 3D scatter plot option was also implemented using the package `"pca3d"`. This package comes with even further options for viewing the scores of PCA and PLSDA analysis in an interactive 3D environment. These options include: data scaling, data centering, show scale, show labels, show pane, show shadow, add ellipses, and show group labels (Figure 11). Finally, both versions of the 3D plots also have an option that enables user to capture a snapshot of the graph.

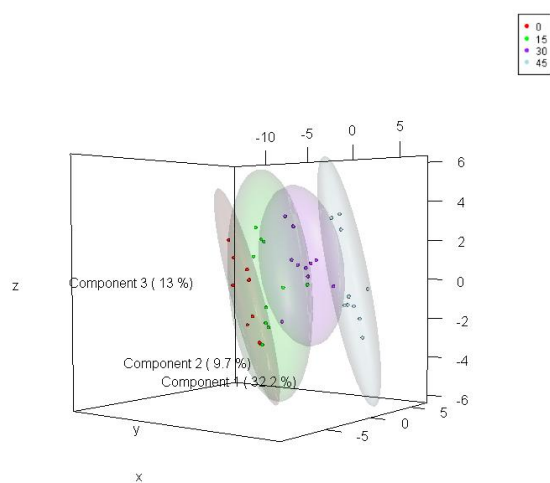
a)



b)



c)



d)

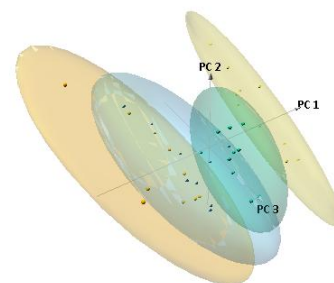


Figure 11. a) The new interactive 3D PCA score plot constructed using “plot3d” package. b) The new interactive 3D PLSDA score plot constructed using “pca3d” package. c) The new interactive 3D PCA score plot constructed using “plot3d” package. d) The new interactive 3D PLSDA score plot constructed using “pca3d” package. All of the plots used the same dataset (cow_diet.csv).

2.2.3.3 PCA and PLS-DA means plot

PCA and PLS-DA score plots are great ways of showing and accounting for the highly multivariate, noisy, collinear and possibly incomplete data in metabolomics. However, in some cases, a trajectory analysis of PCA and PLS-DA can further highlight and reflect the differences between the model's groups. Plotting the trajectory analysis can be very beneficial for visualizing data pattern/trend taken over different time intervals or other qualitative factors. The function to create a trajectory plots is currently nonexistent in MetaboAnalyst 3.0.

An R function was implemented to provide a way of visualising the trajectory of PCA and PLS-DA analysis. This function essentially plots the means of the different groups from a 2D PCA and PLS-DA score plot. The means are calculated by selecting two significant components from the results of a PCA or PLS-DA analysis. The newly implemented R function `PlotTraPCA()` and `PlotTraPLSDA()` computes the standard error (positive and negative) of the means for each of the two significant component then plots the results as a 2D graph (Figure 12). Both the PCA and PLS-DA means plot utilised the R package `ggplot2` and `R Shiny`. The plot allows the user to select specific significant components for x-axis and y-axis for plotting. It also allows the user to customize the title, point size, range of the scales, colours, width of the error bars, and the size (width and height) of the plot itself. Lines connecting the data points of the means can then be manually drawn on the resulting PCA and PLS-DA means plot to visualise the trajectory.

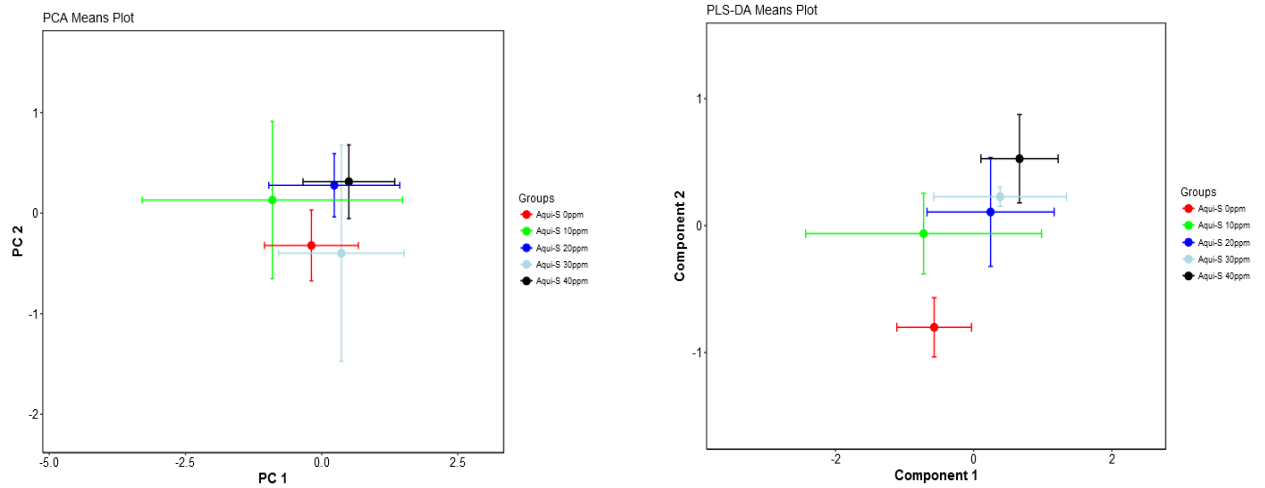


Figure 12. A dataset was obtained to demonstrate the PCA and PLS-DA means plot. This dataset measured the concentrations of various blood metabolites of salmon fish after injecting them with different concentrations of anaesthetics. A PCA means plot of the PCA analysis on the dataset (left), and a PLSDA means plot of the PLSDA analysis on the same dataset (right) are shown above. From the above means plots, we can clearly see a linearly trend/pattern in the PLS-DA means plot of as concentration of anaesthetics increases from 0 to 40.

2.2.3.4 Heatmap Colour Contrasts

MetaboAnalyst comes with 4 different colour contrasts for plotting HeatMaps. 3 more colour contrasts options were added to increase visualization and appeal (red/white/blue, red/white/green/, white/navy/blue) (Figure 13).

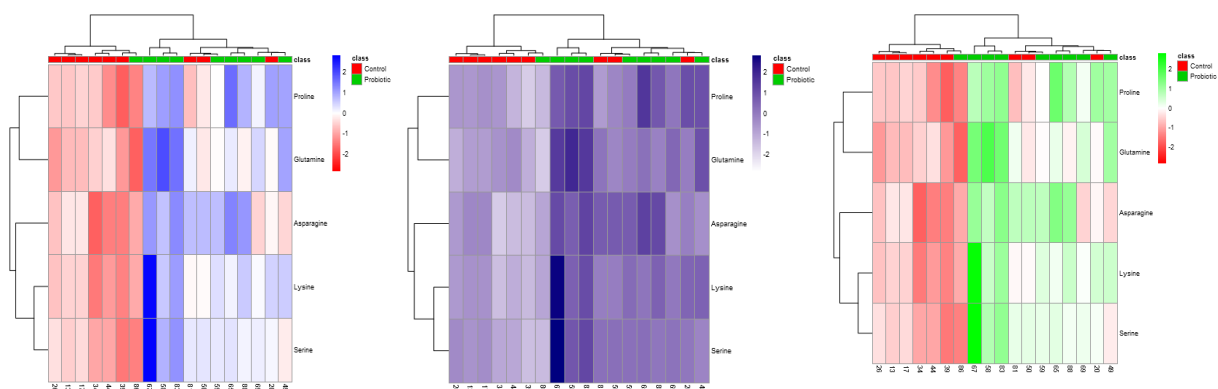


Figure 13. New implemented colour contrasts for Heatmaps. From left to right: red/white/blue, red/white/green and white/navy/blue.

2.2.3.5 Partial Least Square Regression (PLSR) Analysis

MetaboAnalyst 3.0 can perform partial least square regression (PLSR) analysis. This is accomplished by the function “`PLSR.Anal()`” which utilises the R package “`pls`” to perform the partial least square calculations with the “`oscorespls`” method. However, MetaboAnalyst only offers PLSR visualization tools that are compatible with categorical data. In other words, MetaboAnalyst can only perform the partial least square discrimination analysis (PLS-DA) variant of the PLSR to discriminate the difference between the groups and visualize their separation (refer to section 1.3.2.2.1) . In many cases, studies produce continuous data that relies on the normal form of PLSR analysis and its associated visualization options to construct a regression model that can in turn be used for prediction and validation.

Three new functions were implemented to unlock the original limitations present from the MetaboAnalyst’s PLSR function to allow continuous data compatibilities:

1)

`PlsRegPlot(comp.num)`

This function allows the original “`PLSR.Anal()`” function in MetaboAnalyst to fit a PLSR model on a set of continuous data and displays the results through a prediction plot (Figure 8)

2)

`plsRegPlotCV(comp.num)`

This function performs a cross validation model check using the “leave one out cross validation” method (LOOCV) and displays the results through a prediction plot (Figure 8).

3)

`predOvrlyPlt(comp.num)`

The difference between a normal PLSR prediction plot and a LOOCV prediction plot is that each point on the LOOCV prediction plot is estimated based on the results of PLSR analysis. Therefore, the better the results of PLSR analysis are the closer the LOOCV prediction plot will resemble the original PLSR prediction plot. The “predOvrlyPlt(comp.num)” function was implemented to enable a way of visualising and assessing how well the PLSR regression model fitted the data and whether there is any potential overfitting. The predOvrlyPlt(comp.num) achieves this by overlaying the original prediction plot on top of the prediction plot constructed by LOOCV (Figure 14).

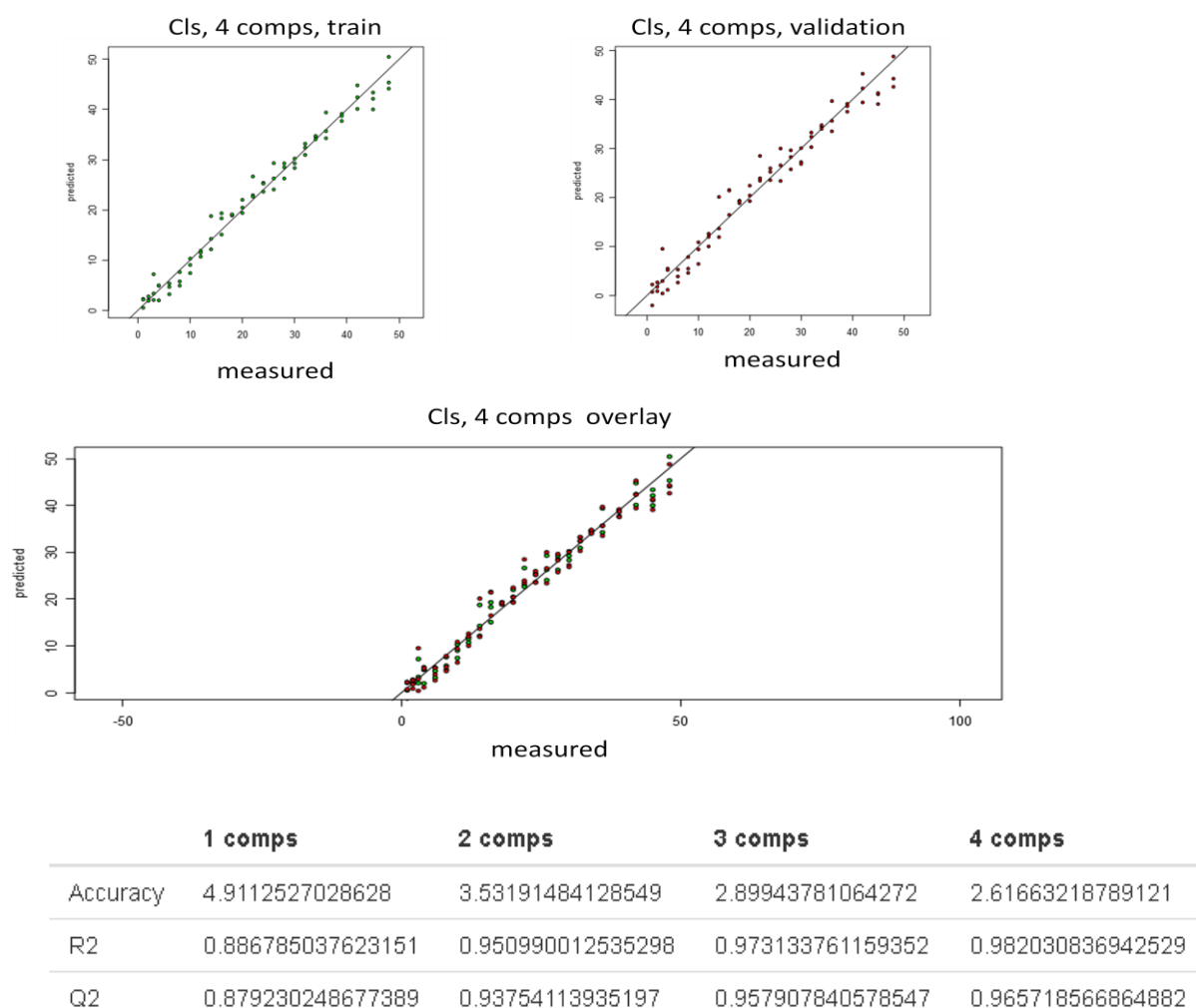


Figure 14. A continuous metabolite profile dataset obtained from a study that applied metabolomics to examine developmental stages of zebra fish embryos (Hayashi, Akiyama, Tamaru, Takeda, & Fujiwara, 2009). This dataset was used as an example to demonstrate the implemented PLSR analysis. Top left: A prediction plot constructed from a PLSR analysis on the dataset after fitting 4 components. Top right: A prediction plot showing displaying the results of the LOOCV validation of the original fitted PLSR model. Bottom: The comparison plot that combines both the original and validation prediction plots. This plot showed that the original prediction model matched very well with the validation model. The accuracy, R2, and Q2 were also provided in the data table shown in this figure.

2.2.4 Construction of an R shiny Application for the statistical analysis module of MetaboAnalyst 3.0

In order to further enhance MetaboAnalyst 3.0, additional options need to be implemented to grant the user more ways to customize the output of their graphs and plot. In addition, the program needs to be more dynamic in terms of reacting to user inputs. Currently MetaboAnalyst 3.0 offers very limited options of directly changing properties of a specific plot such as: plot size (height and width), titles, labels and legends etc. In fact, MetaboAnalyst 3.0 only includes 3 user options for each individual plot: format, resolution and size (default, half page, and full page). Despite having an image option function implemented under the processing step that allows the user to change colour for the different groups from the data and shapes the points used for plots, these options cannot be uniquely applied to individual plots. In other words, once it is set at the starts of the statistical analysis pipeline all following plots generated will apply the chosen setting. The user would have to navigate back and forth between mid analysis and start of the analysis in order to customize the colour and point shape for each individual plot.

To accomplish the above stated improvements the statistical analysis module of the MetaboAnalyst 3.0 was selected and re-implemented using the R Shiny package. The new standalone package can potentially be distributed online and offline. It integrated the original list of statistical analysis functions from MetaboAnalysis 3.0 with the additional and extra features implemented previously (Section 2.2.3). Most importantly, the new package is significantly more dynamic and includes many changeable options for visualising, customising and personalising outputs.

2.2.5 R Shiny

2.2.5.1 Selection of R Shiny

MetabAnalyst 3.0 utilises the R package Rserve (Urbanek, 2003). This package is a TCP/IP server which allows other programs to use facilities of R from various languages without the need to initialize R or link against R library. Every connection has a separate workspace and working directory. User interface side implementations are available for popular languages such as C/C++, PHP and Java. Rserve supports remote connection, authentication and file transfer as seen in MetaboAnalyst 3.0.

R Shiny is a simpler, efficient and more convenient way for R users to turn their analyses into an interactive web applications or an offline R application that anyone can use. These applications let the user specify input parameters using friendly controls like sliders, drop-downs, and text fields; and they can easily incorporate any number of outputs like plots, tables, and summaries. The biggest advantage of using R Shiny is that neither HTML nor JavaScript knowledge are required to code working applications. There are also many ways of sharing the created R shiny application. Its source code can be uploaded online as a GitHub gist, R package, or zip file and ran locally from the user given that they have R installed. A working R Shiny can also be upload to a server such as the application sharing service provided by R Shiny itself. There is also potential to turn an R Shiny application to windows executable. More information regarding R Shiny can be found at <http://shiny.rstudio.com/>.

2.2.5.2 R Shiny Application System Architecture

A working R Shiny application is constructed from two separate parts; an interface component and a server component. The user interface handles the user inputs and output displays. A set of pre defined R Shiny functions assist in setting up the interface elements that the user can interact with (Figure 15). In R Shiny these elements are referred to as controllers and widgets. The server component is where the actual computation takes place. The interactivities of both of these components are controlled by reaction expressions: codes defined within the server component of the application. The building block of Shiny package is based on this form reactive programming. Since the major task of a statistical application is to acquire inputs and produce outputs, the whole R Shiny programming language is designed so that a change in any input whether it is input data or method parameters from the user interface will change the end result. This process is done by immediately alerting the R server component of any input changes made in the user interface by the user. This in turn activates the reactive expressions on the server side which then signal various methods to compute, or recalculate. The results are then reflected into the form of texts, tables or figures and then updated in the interface (Figure 16). This is the biggest advantage of R shiny application's architecture; it relies on reacting to user inputs and enables the R Shiny procedure to provide different outputs without the need to refresh the web page or user interface.

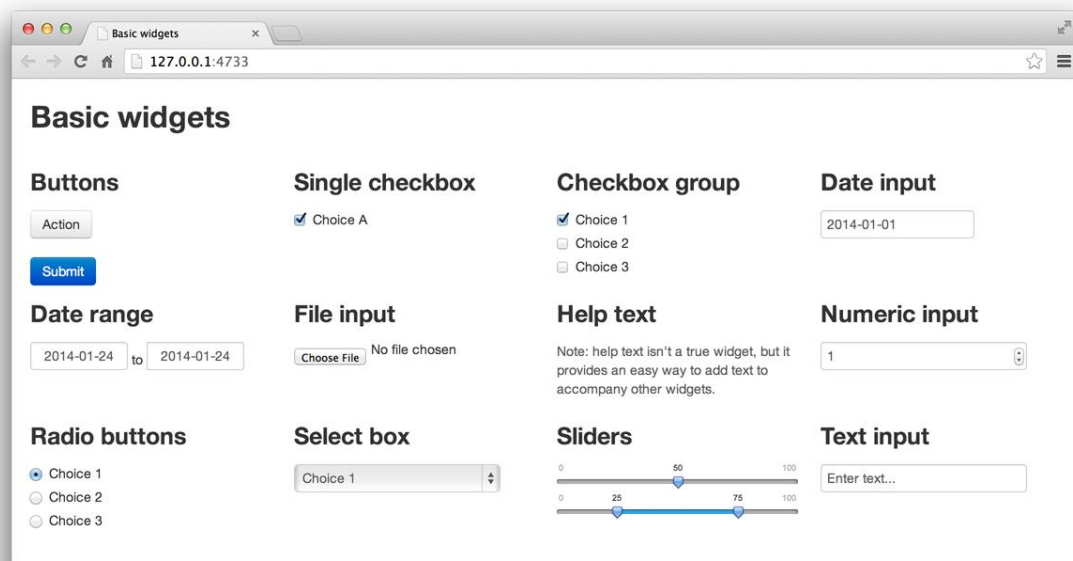


Figure 15. Basic pre-built R Shiny widgets.

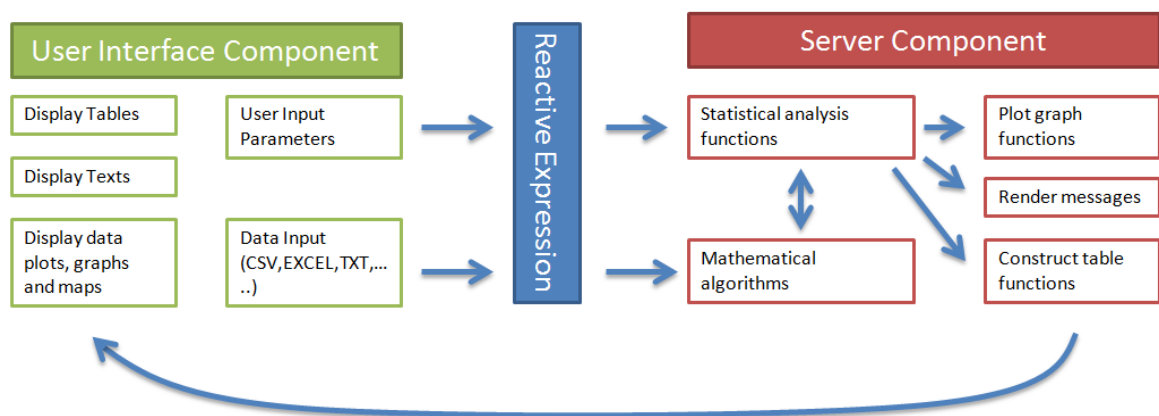


Figure 16. An R Shiny application's system architecture.

For online usage of R Shiny applications, the communication between the user and server is done with the fast websockets package. Websockets is a computer communications protocol, providing full-duplex communication channels over a single TCP (Transmission Control Protocol) connection. Websockets are important in situations where there are constant back and forth dialogue or data exchange between the user interface component and the

server component. This protocol operates separately and only handshake between the client and server is done over the HTTP (Hypertext Transfer Protocol). The duplex connection is open all the time and therefore the authentication is not needed when exchange is done. The websocket is currently supported by many modern browsers.

If the R Shiny application is operated offline, an R Shiny function will be executed by the R engine to create an object that combine both the server and interface component. In this case the object created will be the application and will run on a local server created by the local computer. Offline R Shiny applications will have a much faster performance compared to an online version as it is not affected by the speed of the internet provider.

2.2.6 Coding Structure of the New R Shiny Application

2.2.6.1 Original Scripts used from *MetaboAnalyst 3.0*

The original source code of MetaboAnalyst 3.0 was downloaded from <http://www.metaboanalyst.ca/faces/home.xhtml>. The R scripts: chemometrics.R, classification.R, clustering.R, correlations.R, datautils.R, misceutils.R, normalization.R, processing.R, sigfeatures.R, and univartes.R were extracted to be integrated with the new R Shiny application as they directly contribute to the statistical analysis module of Metaboanalyst 3.0. The functions from these scripts were placed in the server component of the R shiny Application.

2.2.6.2 User Interface Coding Structure

Codes used to construct the user interface component of the application consists of a list of R Shiny functions that defines and sets up the format and position of widgets, controllers, buttons, graphs and tables. A list of the interface components are described in greater detail in the following section (The full codes for the user interface component are provided in the USB complemented with the thesis).

It is important to note that the interface functions simply create elements in the user interface. The actual functionalities of the widgets, buttons and control when the user interacts with them must be defined through functions in the server component. In the case of the graphs, plots and tables, interface functions produce an empty space for them with defined size and position. The actual results needed for and construction of these graphs, plots and tables are also obtained through functions in the server component.

2.2.6.2.1 User Interface Functions

1)

```
navbarPage(title, ..., id = NULL, selected = NULL,  
position = c("static-top", "fixed-top", "fixed-bottom"),  
header = NULL, footer = NULL, inverse = FALSE,  
collapsible = FALSE, collapsable, fluid = TRUE,  
responsive = NULL, theme = NULL, windowTitle = title)
```

```
navbarMenu(title, ..., id = NULL, selected = NULL,  
position = c("static-top", "fixed-top", "fixed-bottom"),  
header = NULL, footer = NULL, inverse = FALSE,  
collapsible = FALSE, collapsable, fluid = TRUE,  
responsive = NULL, theme = NULL, windowTitle = title)
```

```
sidebarPanel(..., width = 4)
```

The “`navbarPage()`” functions were implemented to create a page with a top level navigation bar that can be used to toggle a set of `tabPanel` elements. The “`navbarMenu()`” functions were also implemented to create an embedded menu within the navbar that in turns includes additional `tabPanels`. In addition numerous “`sidebarPanel()`” functions were included to create panels that group a list of options together. Parameters for these two functions define the tiles, position and customised theme for the navbar page and menu.

2)

```
tabPanel(title, ..., value = title, icon = NULL)
```

This function was implemented multiple times to create tab elements. Tab elements are useful in this application for dividing the interface into multiple independently viewable sections. The “`title`” parameter variable displays the title for the tab.

3)

```
radioButtons(inputId, label, choices, selected = NULL, inline = FALSE,  
width = NULL)
```

This function was implemented multiple times to create a set of radio buttons in the application interface that allowed the user to select options from a list. The parameter “label”, “choices”, “selected” helps set the title, selection choices, and the default selection respectively.

4)

```
selectInput(inputId, label, choices, selected = NULL,  
multiple = FALSE, selectize = TRUE, width = NULL,  
size = NULL)
```

This function was implemented multiple times to create selection lists that can be used to select a single or multiple items from a list of values. The parameter variables define the same properties as in the “radioButtons” function.

5)

```
fileInput(inputId, label, multiple = FALSE, accept = NULL,  
width = NULL)
```

This function was responsible for creating file upload control in the R Shiny application. This control allows the user to upload one or more files. Whenever a file upload completes, an input variable is set to a dataframe. This dataframe contains one row for each selected file, and the following columns:

name

The filename provided by the web browser. This is not the path to read to get at the actual data that was uploaded.

size

The size of the uploaded data, in bytes.

type

The MIME type reported by the browser (for example, text/plain), or empty string if the browser didn't know.

datapath

The path to a temp file that contains the data that was uploaded. This file may be deleted if the user performs another upload operation.

6)

`helpText(...)`

```
textOutput(outputId, container = if (inline) span else div,
inline = FALSE)
```

These functions were implemented multiple times to create help texts or areas where texts are to be display in the application interface. These texts were useful in providing additional explanations or contexts for various analyses.

7)

```
checkboxInput(inputId, label, value = FALSE, width = NULL)
```

This function was implemented multiple times create checkboxes that can be used to specify logical values. The “value” variable represents whether the checkbox has been clicked (TRUE) or left empty (FALSE).

8)

```
numericInput(inputId, label, value, min = NA, max = NA, step = NA,
width = NULL)
```

This function was implemented multiple times to create an input controls for entry of numeric values. The “value” parameter defines the initial value. The “min” and “max” parameter defines the minimum and maximum allowed value. The “step” parameter sets the intervals to use when stepping between min and max.

9)

```
actionButton(inputId, label, icon = NULL, width = NULL, ...)
```

This function was implemented multiple times to create clickable buttons in the user interface that performs certain functions when interacted with by the user.

10)

```
plotOutput(outputId, width = "100%", height = "400px",
click = NULL, dblclick = NULL, hover = NULL,
hoverDelay = NULL, hoverDelayType = NULL, brush = NULL,
clickId = NULL, hoverId = NULL, inline = FALSE)
```

This function was implemented multiple times to handle plot rendering within the user interface page. The “width” and “height” variables defined by a valid CSS unit (for example “100%”, “400p”) will set the size of the plot. For some plots the variable “click” was assigned by an object created through the function “clickOpts”. This will prompt the plot to send the mouse coordinates to the server whenever it is clicked, and the value will be accessible via the code “input\$plot_click”. The value will be a named list with x and y elements indicating the mouse position. Other variables: “dblclick”, “hover”, “hoverDelay”, “hoverDelayType”, “brush”, “clickId”, “hoverId” and “inline” are not applied in this application.

11)

```
tableOutput(outputId)
dataTableOutput(outputId)
```

These two functions were implemented multiple times to define table elements or data table elements within the user interface.

12)

```
uiOutput(...)
renderUI({...})
```

These are dynamic UI functions that were implemented many times in the application. The previous list of functions creates a set of controls that affect a

fixed set of outputs. However, R Shiny also has the ability to generate dynamic UIs. This is done by creating an empty dynamic object with the `“uiOutput()”` function in the user interface component. The `“renderUI({...})”` is then called in the server component to render certain widgets, controllers, graphs or tables only when conditions defined within the `“renderUI({...})”` function are met.

2.2.6.3 Server Component Coding Structure

Codes written in the server component of the application consist of a combination of render functions and reactive expressions. The render functions are involved with construction and display of the plots, graphs, and tables that were declared through codes written in the user interface component (see previous section). The reactive expression functions constantly monitors and picks up changes in conditions, such as data properties and user inputs made through the user interface and relay the information to the render functions. In order to obtain the results required for certain plots and tables, the render functions will also call various original MetaboAnalyst 3.0 statistical functions integrated in the server component in a cascading fashion. A list of the server components functions used in the creation of this application is described in the following section. The full code of the sever component is included in the USB complemented with the thesis.

2.2.6.3.1 Server component Functions

1)
`reactive(x, env = parent.frame(), quoted = FALSE,
label = NULL,
domain = getDefaultReactiveDomain(), ..stacktraceon = TRUE)`

This is a reactive expression that was implemented many times in this application. These functions are expressions that can read reactive values and call other reactive expressions. Whenever a reactive value change, any reactive expressions that depended on it are marked as "invalidated" and will automatically re-execute if necessary. If a reactive expression is marked as invalidated, any other reactive expressions that recently called it are also marked as invalidated. In this way, invalidations ripple through the expressions that depend on each other.

2)

```
observeEvent(eventExpr, handlerExpr,  
event.env = parent.frame(), event.quoted = FALSE,  
handler.env = parent.frame(), handler.quoted = FALSE,  
label = NULL, suspended = FALSE, priority = 0,  
domain = getDefaultReactiveDomain(), autoDestroy = TRUE,  
ignoreNULL = TRUE)
```

```
eventReactive(eventExpr, valueExpr,  
event.env = parent.frame(), event.quoted = FALSE,  
value.env = parent.frame(), value.quoted = FALSE,  
label = NULL, domain = getDefaultReactiveDomain(),  
ignoreNULL = TRUE)
```

Shiny's reactive programming framework is primarily designed for calculated values and side-effect-causing actions that respond to *any* of their inputs changing. In other words any changes in the user interface input will immediately alert the “`reactive()`” function to re-execute certain methods. That is often what is desired in Shiny apps. However, for many scenarios in this application it is preferred to wait for one specific action to be taken from the user, like clicking an `actionButton`, before calculating an expression or taking another action. This allows the user to input/change multiple inputs before one action takes place. A reactive value or expression that is used to trigger other calculations in this way is called an *event*.

These situations demanded a more imperative, “event handling” style of programming that is possible--but not particularly intuitive--using the reactive programming primitives `observe` and `isolate`. The “`observeEvent()`” and “`eventReactive()`” functions provided straightforward APIs for event handling that wrap `observe` and `isolate`.

“`observeEvent()`” was applied for situations that required *performing an action* in response to an event. The parameter variable “`eventExpr`” defines an

event for the application to respond to, and the second parameter variable “handlerExpr” defines a function that was called whenever the event occurs.

The “eventReactive()” function creates a *calculated value* that only updates in response to an event. This is just like a normal reactive expression (“reactive()”) except it ignores all the usual invalidations that come from its reactive dependencies; it only invalidates in response to the given event.

3)

```
renderText(expr, env = parent.frame(), quoted = FALSE,  
outputArgs = list())
```

This function was implemented to complement the “helpText(...)” functions declared in the interface components. They renders text by generating an HTML element that contains the text.

4)

```
renderPlot(expr, width = "auto", height = "auto",  
res = 72, ..., env = parent.frame(), quoted = FALSE,  
execOnResize = FALSE, outputArgs = list())
```

The render plot functions were implemented to render a reactive plot that is suitable for assigning to an output slot created by the “plotOutput()” declared in the interface component. In most cases the metabolomics statistical analysis functions extracted from MetaboAnalyst 3.0 will be called from within the “renderPlot” function.

5)

```
renderTable(expr, striped = FALSE, hover = FALSE,  
bordered = FALSE, spacing = c("s", "xs", "m", "l"),  
width = "auto", align = NULL, rownames = FALSE,  
colnames = TRUE, digits = NULL, na = "NA", ...,  
env = parent.frame(), quoted = FALSE, outputArgs = list())
```

```
renderDataTable(expr, options = NULL, searchDelay = 500,
callback = "function(oTable) {}", escape = TRUE,
env = parent.frame(), quoted = FALSE, outputArgs = list())
```

These two functions were implemented to create reactive tables that were suitable for assigning to an output element declared in the user interface component. “renderTable()” uses a standard HTML table, while “renderDataTable()” uses the DataTables Javascript library to create an interactive table with more features such as sorting, searching, and paging.

6)

```
nearPoints(df, coordinfo, xvar = NULL, yvar = NULL,
panelvar1 = NULL, panelvar2 = NULL, threshold = 5,
maxpoints = NULL, addDist = FALSE, allRows = FALSE)
```

This function enabled graphs and plots constructed by “renderPlot()” to be interactive. In the original MetaboAnalyst the points plotted in many graphs can be clicked, which brings up a boxplot of the clicked point. This feature is significantly enhanced using the R shiny function “nearPoints()”.

This function performs by creating an invisible rectangle around every point on an output graph. Any mouse click mouse event (click, hover, or double-click) within the rectangle near a specific point will return the exact coordinate of that point. This in turn will enable more functions such as plotting additional information regarding that specific point.

Chapter 3

3.1 Application Description/Results

3.1.1 Installation and Initialisation

A testing version of the developed R Shiny application titled “Metabolomics Statistical Analysis R Shiny App 0.1” is available for Windows (R version 3.2.5). The application depends on numerous R packages and packages from Bioconductor (<https://www.bioconductor.org/>). These packages must be installed when using the application for the first time. Subsequent usages of the application require loading the packages each time.

Installation

The user needs to have a working version of R installed on their computer preferably version 3.2.5.

The application comes in the forms of 2 R scripts: “packagesUtils.R” and “Metabolomics Statistics Analysis App 0.1”.

To install the application:

- 1) Open the “packagesUtils.R” script in R.
- 2) Highlight all the codes before the “`library(shiny)`” line in the “packagesUtils.R” script then right-click and choose “run line or selected”.
- 3) A window will appear prompting the user to select a CRAN mirror. Select the appropriate mirror and click OK.

- 4) The required packages and their dependencies will now be downloaded and installed onto the user's computer. If R prompts the user to a personal library to place the installed packages, choose yes.
- 5) After installation, a message will appear in the R console:

```
Old packages: 'rgl', 'mgcv', 'nlme'  
Update all/some/none? [a/s/n]: n
```

Type “a” in the R console and press enter.

- 6) The application has now finished installing all the required packages onto computer

Initialisation

To use the application:

- 1) Open the “packagesUtils” in R. Scroll down to the bottom of the script, highlight the lines “library(shiny)” and click “run line or selected”. This will initialise R Shiny.
- 2) The next step involves starting the application itself. The function “runApp(“ ”)” in the “PackageUtil.R” will look for where the script “Metabolomics Statistics Analysis App 0.1” is located and initialise the application. Find the directory path of the “Metabolomics Statistics Analysis App 0.1.R” script on your computer and enter it between the

“ “ of the “runApp(“ ”) function. Then Replace every “\” (back dash) with “/” (forward dash).

For example:

```
runApp("C:/Users/Desktop/Metabolomics      Statistics  
Analysis App 0.1.R")
```

- 3) Highlight the “runApp (“...”)” line, right-click and select “run line or selected”. The Application will now start.

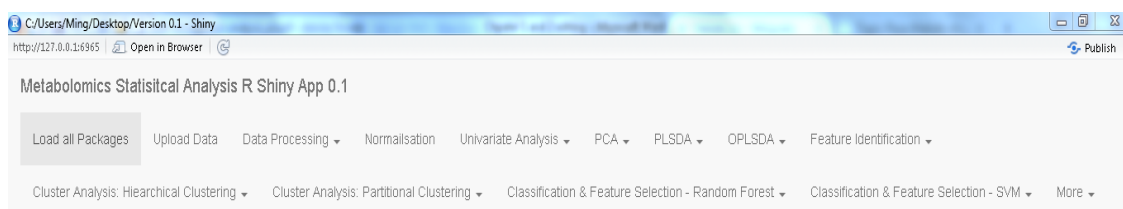
The above 3 steps are required to use the start the application every time.

3.2 Description

A metabolomics dataset was selected to perform a series of statistical analysis using the developed R Shiny application: Metabolomics Statistics Analysis App 0.1 (testing version). The experiment that yielded this data involved treating 9 samples of juvenile (20-30mm in maximum shell length) New Zealand abalone (*Haliotis iris*) with a multi-strain conglomerate of 2- and 3- probiotic bacterial strains that were supplemented into a commercial abalone feed over a 4 months trial period to compare its growth rate against 9 control samples. The 2-probiotic conglomerate consisted of *Exiguobacterium* JHEb1 and *Vibrio* JH1, and the 3-probiotic conglomerate consisted of *Exiguobacterium* JHEb1, *Vibrio* JH1 and *Enterococcus* JHLDc (Hadi, Gutierrez, Alfaro, & Roberts, 2014). The raw data was generated following the same metabolic profiling (analytical platform procedures) and data pre processing procedures (primary bioinformatics) as in Young, Alfaro, & Villas- Bôas, (2016). The final post processed data is a CSV formatted data file titled "Ablone_GC-MS Result(PeakHeight)_QC.csv". A selected list of statistical analysis will be performed on this dataset using Metabolomics Statistics Analysis App 0.1 to demonstrate its improved functionalities and new features.

Load Packages

Highlighting and run the lines “library(shiny)” and “runApp("..")” from “PackageUtil.R” script in R will initialise the application (section 3.1.1). After initialisation an interface window will appear.



In this window the user must load all packages required by the application by clicking “Load All Packages”. The process is completed when the following are displayed in window.

All packages must be initialised before uploading the data.

Load All Packages

After clicking on the Load All Packages button. Please wait until the application displays below "TRUE" for all the packages

row.names(m)

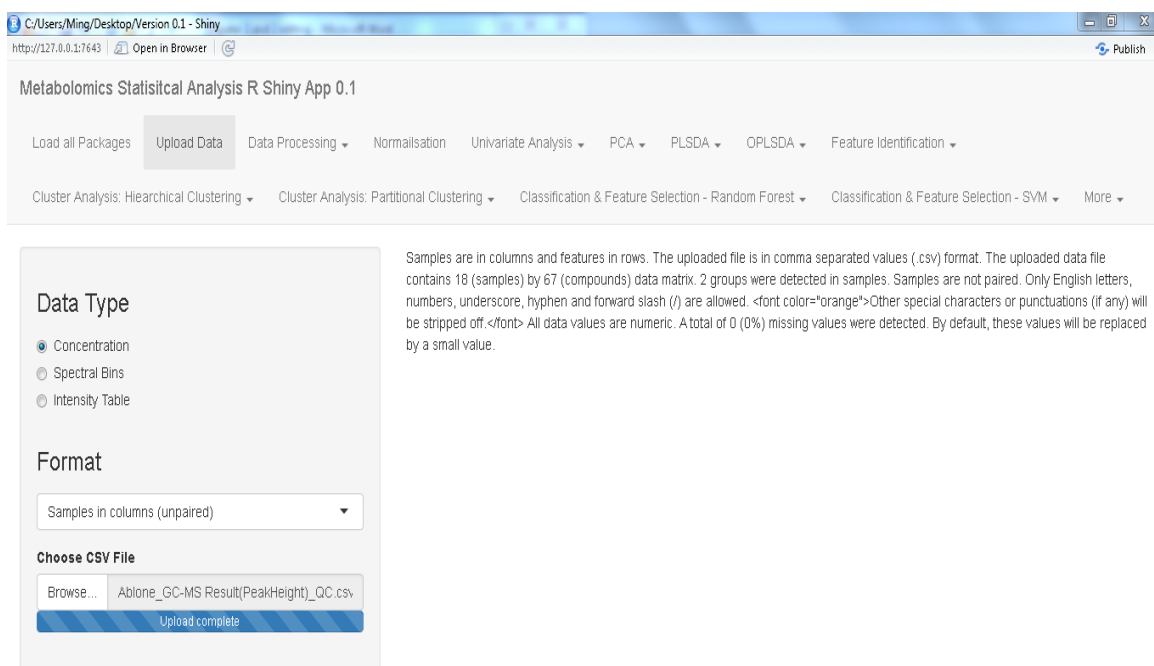
xtable	TRUE
ggplot2	TRUE
shiny	TRUE
rgl	TRUE
pca3d	TRUE
ellipse	TRUE
scatterplot3d	TRUE
pls	TRUE
caret	TRUE
lattice	TRUE
Cairo	TRUE
randomForest	TRUE
e1071	TRUE
gplots	TRUE
som	TRUE
RColorBrewer	TRUE
genefilter	TRUE
pheatmap	TRUE

The package loading process can be viewed by opening up the R console. Any potential error messages will also be displayed there.

Data Upload

The data can then be uploaded from the “Upload Data” tab in the navigation menu located at the top of the application interface window. The user must select the data format and data types before clicking “browse” (refer to <http://www.metaboanalyst.ca/faces/ModuleView.xhtml>) for data types and format requirements). The format of the dataset “Ablone_GC-MS Result (PeakHeight)_QC.csv” has the following properties:

- 1) Samples are in columns and features in rows
- 2) The uploaded file is in comma separated values (.csv) format
- 3) The uploaded data file contains 18 (samples) by 67 (compounds) data matrix. 2 groups were detected in samples
- 4) Samples are not paired
- 5) A total of 0 (0%) missing values were detected



Metabolomics Statistical Analysis R Shiny App 0.1

Load all Packages Upload Data Data Processing Normalisation Univariate Analysis PCA PLSDA OPLSDA Feature Identification

Cluster Analysis: Hierarchical Clustering Cluster Analysis: Partitional Clustering Classification & Feature Selection - Random Forest Classification & Feature Selection - SVM More

Data Type

☒ Concentration
☐ Spectral Bins
☐ Intensity Table

Format

Samples in columns (unpaired)

Choose CSV File

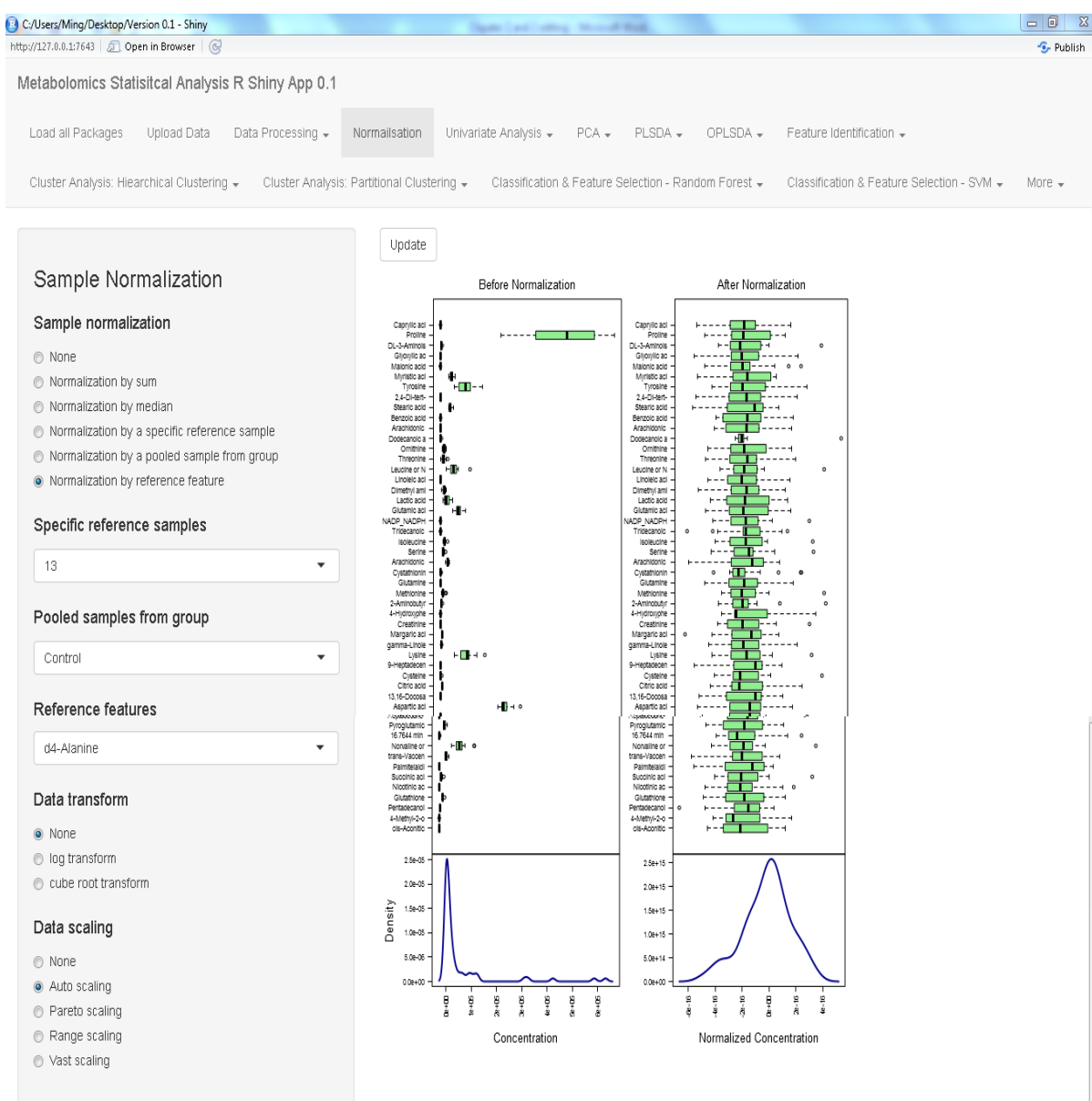
Browse... Ablone_GC-MS Result(PeakHeight)_QC.csv

Upload complete

Samples are in columns and features in rows. The uploaded file is in comma separated values (.csv) format. The uploaded data file contains 18 (samples) by 67 (compounds) data matrix. 2 groups were detected in samples. Samples are not paired. Only English letters, numbers, underscore, hyphen and forward slash (/) are allowed. Other special characters or punctuations (if any) will be stripped off. All data values are numeric. A total of 0 (0%) missing values were detected. By default, these values will be replaced by a small value.

Normalisation

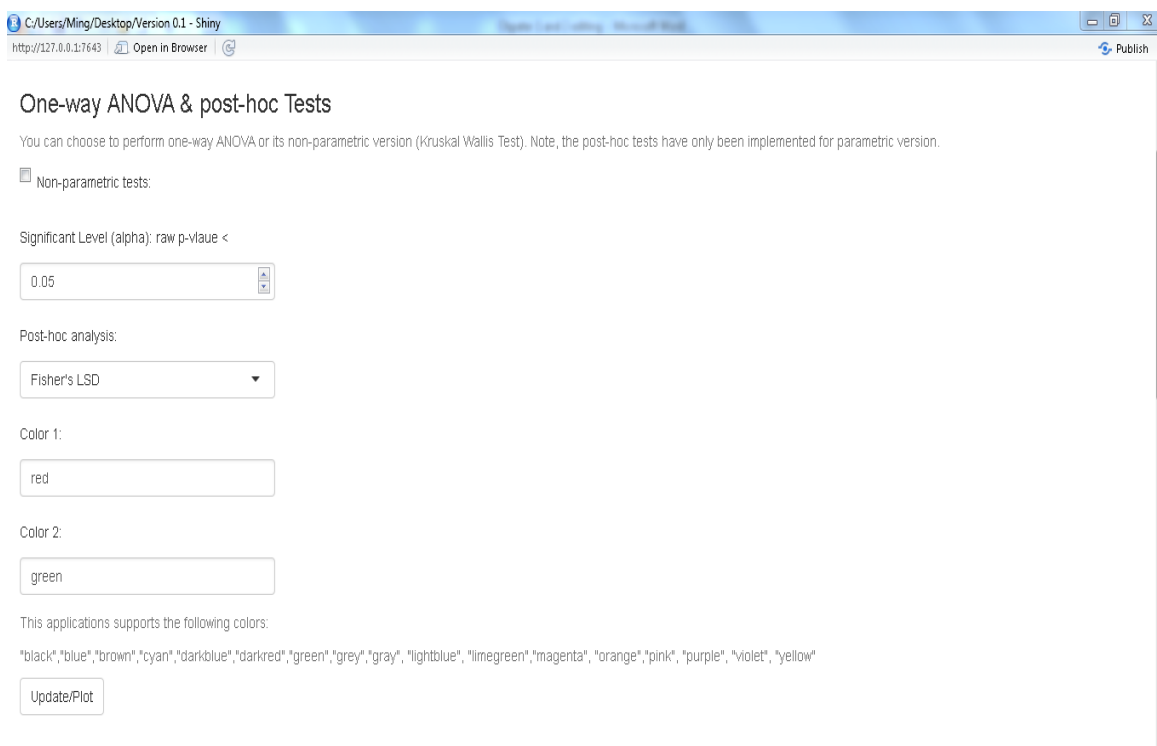
Since a total of 0 (0%) missing values were detected the Data Processing step can be skipped. The next step is data normalisation. This step must be completed in order to proceed with statistical analysis. If the user does not want to normalise their data this step must still be completed by selecting the option “none” for all the normalisation methods then clicking the “update” button to proceed.



If results from the normalization do not satisfy the user, simply reselect the methods then click the “update” button to re-normalise the data.

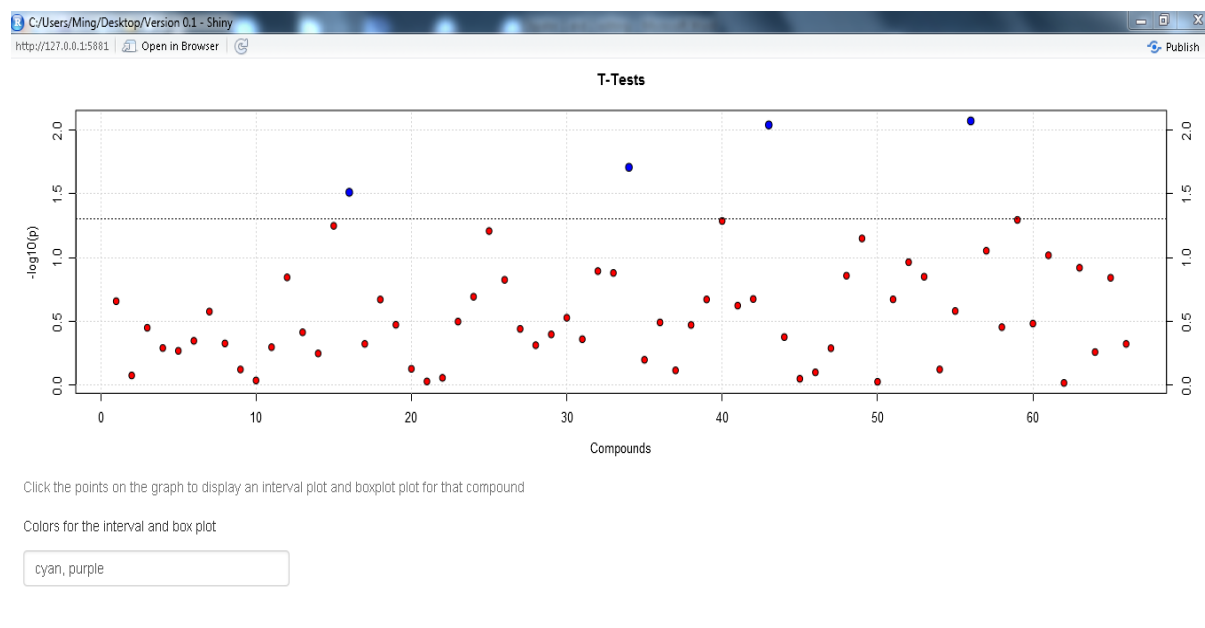
T-Test

The user can now begin statistical analysis. T-test can be selected from the tab navigation bar. (ANOVA cannot be applied on this data set since there are only two groups). The t-test analysis comes with 5 user input perimeters, two of which dynamically adjust the colour setting for the T-test plot.

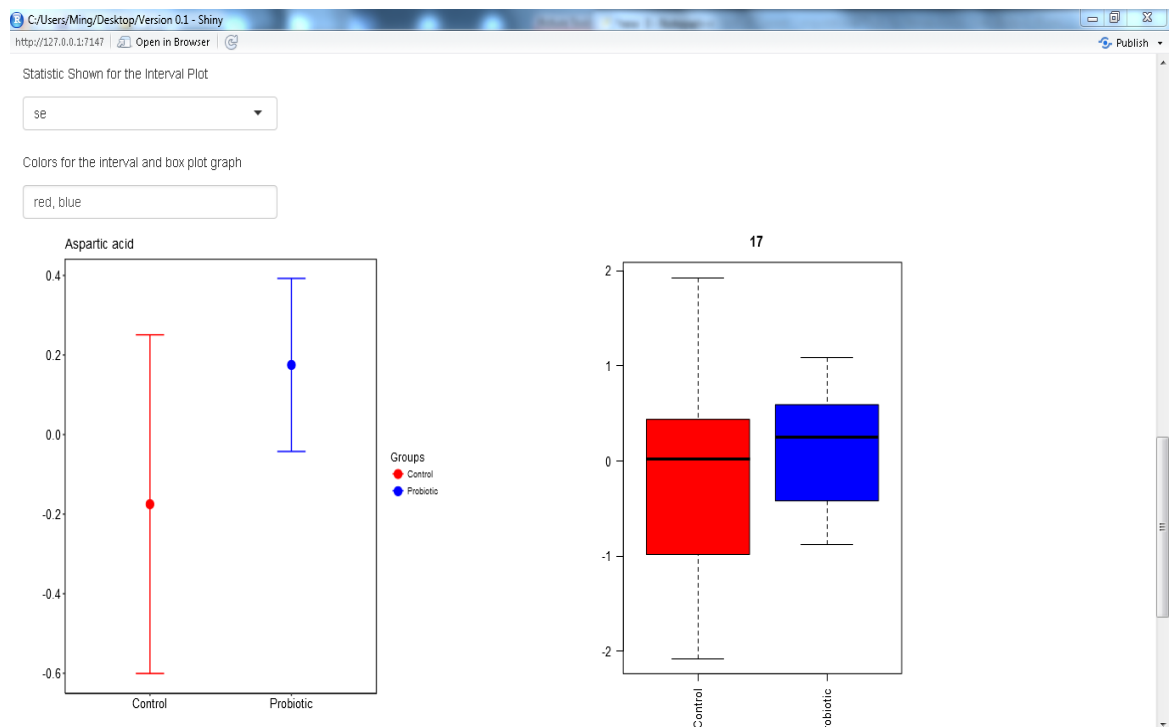


The screenshot shows a web browser window titled "C:/Users/Ming/Desktop/Version 0.1 - Shiny". The address bar shows "http://127.0.0.1:7643". The page title is "One-way ANOVA & post-hoc Tests". Below the title, there is a note: "You can choose to perform one-way ANOVA or its non-parametric version (Kruskal Wallis Test). Note, the post-hoc tests have only been implemented for parametric version." There is a checkbox labeled "Non-parametric tests:". Below this, there is a label "Significant Level (alpha): raw p-value <" followed by a text input field containing "0.05". Below that, there is a label "Post-hoc analysis:" followed by a dropdown menu showing "Fisher's LSD". Below that, there is a label "Color 1:" followed by a text input field containing "red". Below that, there is a label "Color 2:" followed by a text input field containing "green". Below these fields, there is a note: "This applications supports the following colors:" followed by a list of colors in quotes: "black", "blue", "brown", "cyan", "darkblue", "darkred", "green", "grey", "gray", "lightblue", "limegreen", "magenta", "orange", "pink", "purple", "violet", "yellow". At the bottom, there is a button labeled "Update/Plot".

Currently, Metabolomics Statistics Analysis App 0.1 supports 17 unique colours (black, blue, brown, cyan, darkblue, darkred, green, grey, gray, lightblue, limegreen, magenta, orange, pink, purple, violet, yellow). To change the colour, the user must enter the name of the colour from the colour list into the text input field then click the update/plot button. The T-test will generate the following scatter plot.



Each point on the T-test scatter plot represents a compound. The points are clickable and will display the interval plot (see section 2.2.3.1) alongside the original MetaboAnalyst boxplot for that compound. In addition, the option to change the colours for these two plots is also provided. To change the colours of the interval and boxplots, input the colours in a sequence format into the text input field. The number of colours the users input must match the number of groups in the data. The colour change will happen automatically and the updated graph will be displayed as soon as the user finishes the input. Below is a screen shot of an Interval and boxplot produced for the compound Aspartic acid.



A data table containing all the significant features (its t-stat, p-value, $-\log_{10}(p)$ and FDR) will be displayed at the bottom of the T-test tab. This data table has sorting and searching capabilities when the dataset are large.

Show 20 entries

Search:

Name	t.stat	p.value	$-\log_{10}(p)$	FDR
Proline	-2.9997	0.0084847	2.0714	0.301
Lysine	-2.965	0.0091214	2.0399	0.301
Glutamine	-2.5935	0.019597	1.7078	0.43113
Asparagine	-2.3691	0.030753	1.5121	0.493

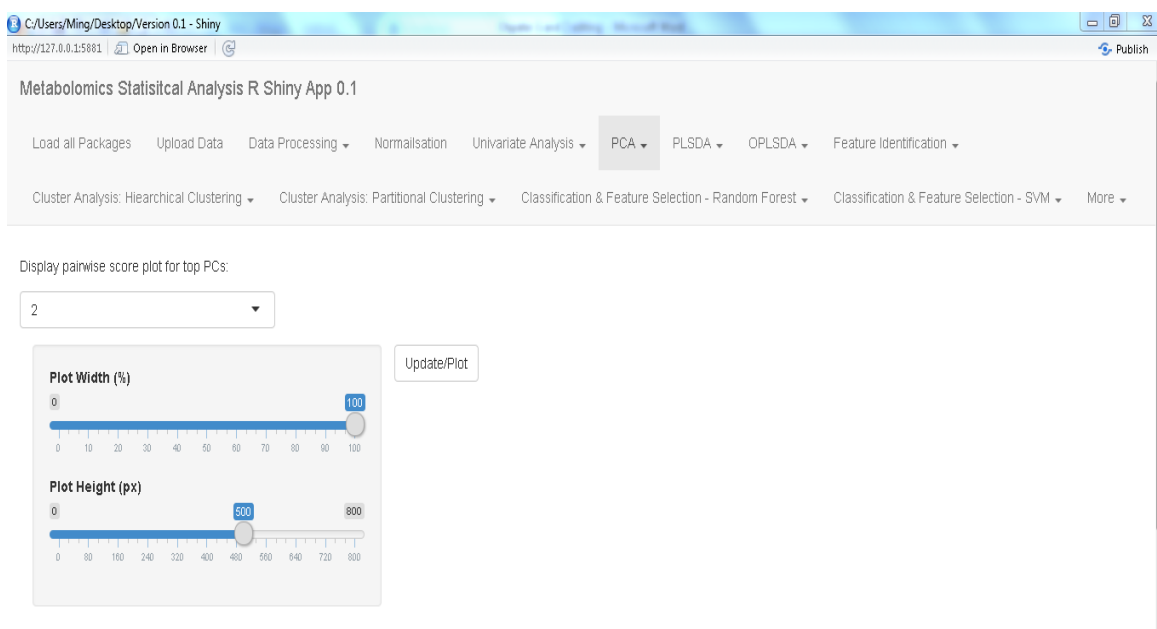
Name t.stat p.value $-\log_{10}(p)$ FDR

Showing 1 to 4 of 4 entries

Previous 1 Next

PCA and PLSDA

Many plots from the PCA and PLS-DA analysis come with dynamic sliders to adjust size (height and width) of the plot, for example:



The 3D scatter plot of the PCA and PLS-DA also contains many options that the user can change. (See section 2.2.3.2 and below). When the user changes a setting, they must click on the plot button and a new 3D plot with the altered settings will be plotted. If the user chooses to capture a snapshot, the image will be saved in the same folder the Metabolomics Statistics Analysis App 0.1.R script is located in.

3D PCA plot using raw rgl

Point Size
0.7

Transparency for ellipses
0.1

☒ Add ellipses

☐ add grid to plot

Title

Colors for PCA
brown, orange

plot pca 1 Snap Shot

Note: the SnapShot will only capture latest plotted graph.

3D pca plot using PCA3d package

☒ Data Scaling

☐ data Centering

☐ show scale

☐ show labels

☐ Show Plane

☐ show Shadow

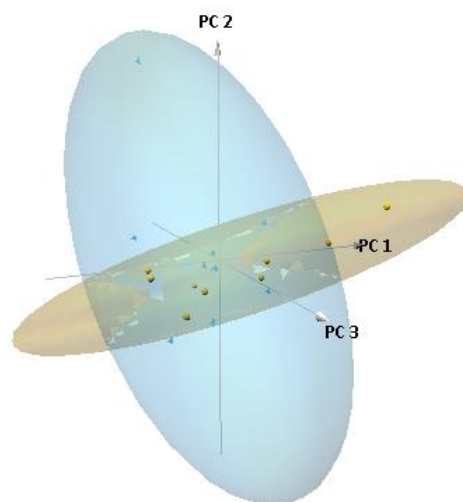
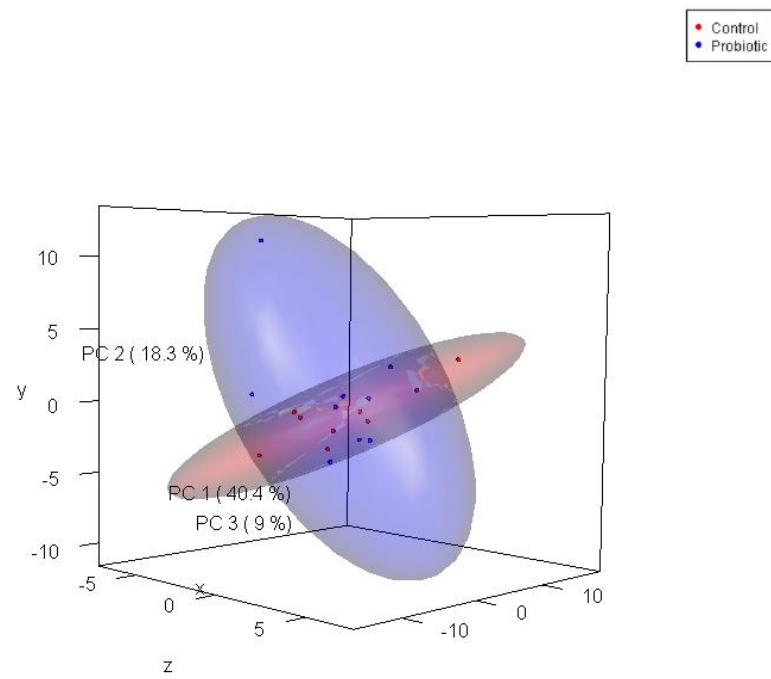
☒ Add ellipses

☐ show group labels

plot pca 2 Snap Shot

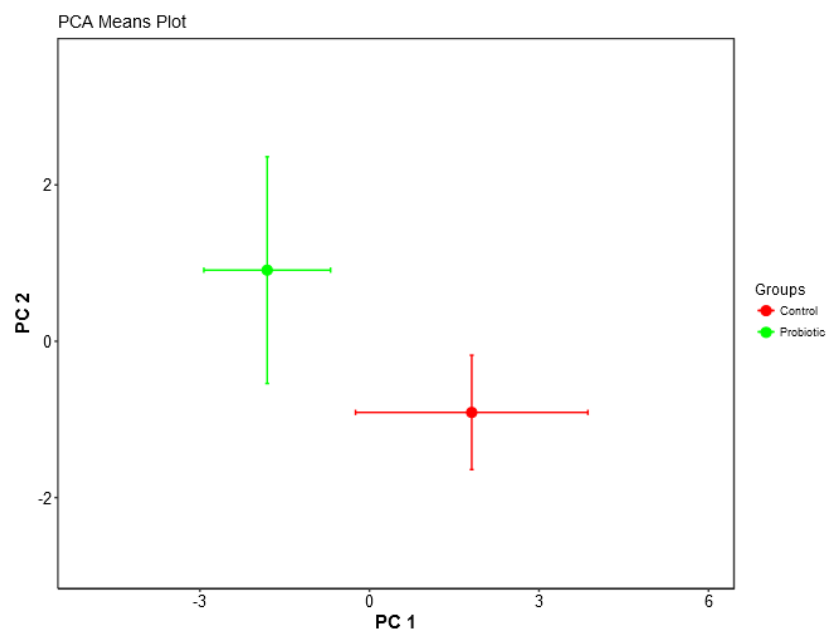
Note: the SnapShot will only capture latest plotted graph.

The 3D scatter plots below displays the results of the PCA analysis using the Ablone_GC-MS Result(PeakHeight)_QC.csv data



The user can also choose to plot a PCA means plot (See section 2.2.3.3). Dynamically adjustable options for the PCA trajectory plot include changeable title, colour options error bar width, point size and graph scale range.

The screenshot shows a Shiny web application interface. At the top, the browser address bar shows 'http://127.0.0.1:15881'. The interface includes several input fields and a 'Publish' button. The inputs are: 'Specify PC on x-axis:' with a dropdown set to '1'; 'Specify PC on y-axis:' with a dropdown set to '2'; 'Title' with a text box containing 'PCA Trajectory Plot'; 'Colors for the graph' with a text box containing 'lightblue, darkred'; 'Error Bar Width' with a text box containing '0.03'; 'Point Size' with a text box containing '2'; and 'Increase The range of the scale by (%)' with a text box containing '20'. Below these inputs is a 'Plot Width (%)' slider set to 100 and a 'Plot Height (px)' slider set to 500. An 'Update/Plot Graph' button is located to the right of the sliders.



3.3 Results

The PLSR function cannot be demonstrated using the “Ablone_GC-MS Result(PeakHeight)_QC.csv” dataset as this dataset is discrete. However section 3.2 have explained the construction of this function and provided an example dataset to demonstrate the result.

The application has been installed on several windows computers with two different versions of R, R 3.2.5 portable and R 3.3.2. The installation (and all the required packages) time ranged from 10 minutes to 30 minutes depending on the CPU of the computer itself. The application also ran very smoothly and did not encounter any errors resulting in wrong results or crashes.

Errors however did occur after publishing this application on the R Shiny server. The package “ggplot” would not be loaded by the R Shiny server. In addition, errors are encountered in regards to the 3D interactive plots, potentially due incompatibility between 3D objects directly constructed from R and HTTP.

Various dynamic functions such as reading and generating input selections depending on the dataset were all working as intended.

The Metabolomics Statistical App 0.1 is supplied in a supplementary USB complemented with this thesis. It included an R portable version 3.2.5 with all the necessary packages installed. In addition, all datasets used in this thesis were also provided in the USB

Chapter 4

Discussion and Conclusions

4.1 Discussion

This thesis reviewed a few popular and new metabolomics data analysis softwares, tools, packages and highlighted their advantages and disadvantages in the context of its comprehensiveness and user friendliness. Because metabolomic is a relatively new and unique addition to “Omics” technologies, its complex data analysis procedures involves large multivariate datasets which pose a problem to new researchers or researchers from a biological background looking to apply metabolomics. We believed that the ideal metabolomics tool needs to be comprehensive yet simple in its design. Such a feat is not easy to achieve in data analysis softwares. Indeed, increase in data analysis complexity would naturally entail increase in the intricacy of a using a software to perform those data analyses.

Amongst the popular metabolomics bioinformatic softwares and packages reviewed, MetaboAnalyst 3.0 is the most comprehensive. It is a free-to-use and easy-to-use web based application that implements multiple modules for metabolomics data analysis. MetaboAnalyst 3.0 is open sourced and the developers encourages downloading their codes to further develop R metabolomics data analysis tools. After closer examination of the functionality and coding architecture of MetaboAnalyst 3.0, we identified many of its original features can be enhanced, and potential new functions can be added. Therefore, we proposed to not only implement the new features and functions, but to also attempt to create an application using MetaboAnalyst as a

foundation that can be both comprehensive in its functionality and dynamic in its user interactivity.

This project introduced Metabolomics Statistics Analysis App 0.1. A GUI based application that can be directly executed from the R console or potentially published online. This application utilised the R Shiny package which enabled the implementation of the application itself without knowledge in JAVA and HTML. Employment of the R Shiny package basically enabled a user friendly way to directly alter and manipulate various R functions and its input parameters through a graphic user interface without the user having knowledge in the R language and programming. In other words, Metabolomics Statistics Analysis App 0.1 can integrate complex R packages and dynamically control many if not all of the function parameters from the said packages to significantly increase the application's analytical and visualization capabilities. As a result, Metabolomics Statistics Analysis App 0.1 substantially enhanced graphical visualisation of the original MetaboAnalyst 3.0 3D interactive plots and provided more visualization options for user outputs through the implementation of numerous dynamically adjustable inputs parameters. Furthermore, we implemented numerous new statistical functions that are currently not available in MetaboAnalyst 3.0 such as PLSR.

Although current Metabolomics Statistics Analysis App 0.1 have increased functionalities and enhanced visualization options, this application faces some bottleneck in its distribution as was discovered after publishing the application on the R Shiny server. Ideally, the current Metabolomics Statistics Analysis App 0.1 should be distributed online as a web based application or as a

downloadable executable. However, due to the large number of R packages this application integrated, deploying the application on the R Shiny server can potentially result in various packages not being able to load. A way around this bottleneck is to transform the R application into a windows executable that can be downloaded and installed on any windows computer and ran locally. Another method is to construct a server specifically for this application. The latter option however will require cost and maintenance fees. In addition to distribution bottlenecks, the R Shiny package has various limitations in its coding structure that limit certain interactions such as dynamically generating certain number of options depending on the dataset uploaded. Therefore, future upgrade of these applications can potentially find ways around this issue through efficient coding or coding additional scripts to improve the R Shiny package's capabilities itself.

4.2 Conclusion

Metabolomics Statistics Analysis App 0.1 is an R GUI application developed using the R Shiny Package. This application was constructed using the statistical functions from MetaboAnalyst 3.0 as a foundation to provide a tool that not only include a comprehensive list of statistical analysis, and visualization options but also easy to use and interact with.

From a new metabolomics researcher's perspective we believe that this application provided the balance between software complexity and friendliness the user seeks. Although the application is still in its early versions, the significance of being able to use R Shiny to create a GUI R application for something as complex metabolomics data analysis cannot be ignored. This application unlocked much potential in R as a functional programming language in creating various standalone statistical tools.

The final frontier of "Omics" bioinformatics is data integration. To accomplish the integration of multiple "Omics" techniques, complex programs must be created to incorporate novel ways of integrating heterogeneous and large "Omics" datasets. These datasets will also require novel methods to analyse, interpret and visualise. This conceptual challenge and also practical hurdle can perhaps be overcome with more research into R and R Shiny. The R statistical platform is already world renowned, offering more than 4000 add-on packages, a comprehensive and extensive functional programming environment and support from countless R programmers globally. Couple this with continuous research into R Shiny will no doubt unlock the potential to create innovative softwares with powerful analysis, interpretation and visualization capabilities.

References

- Abdi, H. (2007). Partial Least Square Regression. *Encyclopedia of Measurement and Statistics*, 741–744. <http://doi.org/10.4135/9781412952644>
- Aggio, R. B. M., Ruggiero, K., & Villas-bôas, S. G. (2010). Pathway Activity Profiling (PAPI): from the metabolite profile to the metabolic pathway activity, 26(23), 2969–2976. <http://doi.org/10.1093/bioinformatics/btq567>
- Alfaro, A. C., & Young, T. (2016). Showcasing metabolomic applications in aquaculture: a review. *Reviews in Aquaculture*, 1–18. <http://doi.org/10.1111/raq.12152>
- Alonso-herranz, J. G. V., Barbas, C., & Grace, E. (2015). Controlling the quality of metabolomics data : new strategies to get the best out of the QC sample. *Metabolomics*, 518–528. <http://doi.org/10.1007/s11306-014-0712-4>
- Alonso, A., Marsal, S., & Julià, A. (2015). Analytical methods in untargeted metabolomics: state of the art in 2015. *Frontiers in Bioengineering and Biotechnology*, 3(March), 23. <http://doi.org/10.3389/fbioe.2015.00023>
- Álvarez-Sánchez, B., Priego-Capote, F., & Luque de Castro, M. D. (2010). Metabolomics analysis I. Selection of biological samples and practical aspects preceding sample preparation. *TrAC - Trends in Analytical Chemistry*, 29(2), 111–119. <http://doi.org/10.1016/j.trac.2009.12.003>
- Bartel, J. A., Krumsiek, J., & Theis, F. J. (2013). Statistical methods for the analysis of high-throughput metabolomics data. *Computational and Structural Biotechnology*, 4(5), 1–9. <http://doi.org/10.5936/csbj.201301009>
- Bartel, J., Krumsiek, J., & Theis, F. J. (2013). Statistical methods for the analysis of high-throughput metabolomics data. *Computational and Structural Biotechnology Journal*, 4(January), e201301009. <http://doi.org/10.5936/csbj.201301009>
- Beckonert, O., Keun, H. C., Ebbels, T. M. D., Bundy, J. G., Holmes, E., Lindon, J. C., & Nicholson, J. K. (2007). Metabolic profiling, metabolomic and metabonomic procedures for NMR spectroscopy of urine, plasma, serum and tissue extracts. *Nature Protocols*, 2, 2692–2703. <http://doi.org/10.1038/nprot.2007.376>
- Berger, B., Peng, J., & Singh, M. (2013). Computational solutions for omics data. *Nature Publishing Group*, 14(5), 333–346. <http://doi.org/10.1038/nrg3433>
- Boccard, J., & Rudaz, S. (2014). Harnessing the complexity of metabolomic data with chemometrics, (October 2013). <http://doi.org/10.1002/cem.2567>
- Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (1998). Extensible markup language (XML). *World Wide Web Consortium Recommendation REC-Xml-19980210*. <http://www.w3.org/TR/1998/REC-Xml-19980210>, 16, 16.
- Burton, L., Ivosev, G., Tate, S., Impey, G., Wingate, J., & Bonner, R. (2008). Instrumental and experimental effects in LC-MS-based metabolomics. *Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences*, 871(2), 227–235. <http://doi.org/10.1016/j.jchromb.2008.04.044>
- Cambiaghi, A., Ferrario, M., & Masseroli, M. (2017). Analysis of metabolomic data : tools , current strategies and future challenges for omics data integration, (February 2016), 1–13. <http://doi.org/10.1093/bib/bbw031>
- Caspi, R., Altman, T., Dale, J. M., Dreher, K., Fulcher, C. A., Gilham, F., ... Karp, P. D. (2010). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway / genome databases,

- 38(October 2009), 473–479. <http://doi.org/10.1093/nar/gkp875>
- Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker, M., Latendresse, M., ... Karp, P. D. (2008). The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway / Genome Databases, 36(October 2007), 623–631. <http://doi.org/10.1093/nar/gkm900>
- Castro-Puyana, M., & Herrero, M. (2013). Metabolomics approaches based on mass spectrometry for food safety, quality and traceability. *TrAC - Trends in Analytical Chemistry*, 52, 74–87. <http://doi.org/10.1016/j.trac.2013.05.016>
- Connor, S. C., Hansen, M. K., Corner, A., Smith, R. F., & Ryan, T. E. (2010). Integration of metabolomics and transcriptomics data to aid biomarker discovery in type 2 diabetes. *Molecular bioSystems*, 6(5), 909–21. <http://doi.org/10.1039/b914182k>
- Costa, C., Maraschin, M., & Rocha, M. (2015). An R package for the integrated analysis of metabolomics and spectral data. *Computer Methods and Programs in Biomedicine*, 129, 117–124. <http://doi.org/10.1016/j.cmpb.2016.01.008>
- Eichner, J., Rosenbaum, L., Wrzodek, C., Häring, H. U., Zell, A., & Lehmann, R. (2014). Integrated enrichment analysis and pathway-centered visualization of metabolomics, proteomics, transcriptomics, and genomics data by using the InCroMAP software. *Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences*, 966, 77–82. <http://doi.org/10.1016/j.jchromb.2014.04.030>
- Ekins, S., Nikolsky, Y., Bugrim, A., Kirillov, E., & Nikolskaya, T. (2006). Pathway Mapping Tools for Analysis of High Content Data. In D. L. Taylor, J. R. Haskins, & K. A. Giuliano (Eds.), *High Content Screening: A Powerful Approach to Systems Cell Biology and Drug Discovery* (pp. 319–350). Totowa, NJ: Humana Press. <http://doi.org/10.1385/1-59745-217-3:319>
- Emwas, A.-H. M. (2015). The Strengths and Weaknesses of NMR Spectroscopy and Mass Spectrometry with Particular Focus on Metabolomics Research. In T. J. Bjerrum (Ed.), *Metabonomics: Methods and Protocols* (pp. 161–193). New York, NY: Springer New York. http://doi.org/10.1007/978-1-4939-2377-9_13
- Ernst, Richard, R., Bodenhausen, G., & Wokaun, A. (1987). *Principles of nuclear magnetic resonance in one and two dimensions Vol. 14. Magnetic Resonance Imaging*.
- Friedrich, N. (2012). Metabolomics in diabetes research. *Journal of Endocrinology*, 215(1), 29–42. <http://doi.org/10.1530/JOE-12-0120>
- Frolkis, A., Knox, C., Lim, E., Jewison, T., Law, V., Hau, D. D., ... Wishart, D. S. (2009). SMPDB: The small molecule pathway database. *Nucleic Acids Research*, 38(SUPPL.1), 480–487. <http://doi.org/10.1093/nar/gkp1002>
- García-alcalde, F., García-lópez, F., Dopazo, J., Conesa, A., Investigaciones, C. De, & Felipe, P. (2011). Paintomics : a web based tool for the joint visualization of transcriptomics and metabolomics data, 27(1), 137–139. <http://doi.org/10.1093/bioinformatics/btq594>
- Garcia, C. J., García-Villalba, R., Garrido, Y., Gil, M. I., & Tomás-Barberán, F. A. (2016). Untargeted metabolomics approach using UPLC-ESI-QTOF-MS to explore the metabolome of fresh-cut iceberg lettuce. *Metabolomics*, 12(8), 1–13. <http://doi.org/10.1007/s11306-016-1082-x>
- Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., ... Zhang, J. (2004). Bioconductor : open software development for computational biology and bioinformatics, (10).

- Goecks, J., Nekrutenko, A., Taylor, J., & Team, T. G. (2010). Galaxy : a comprehensive approach for supporting accessible , reproducible , and transparent computational research in the life sciences.
- Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merckenschlager, M., Gisel, A., ... Tegnér, J. (2014). Data integration in the era of omics: current and future challenges. *BMC Systems Biology*, 8 Suppl 2(2), I1. <http://doi.org/10.1186/1752-0509-8-S2-I1>
- Goodwin, C. R., Sherrod, S. D., Marasco, C. C., Bachmann, B. O., Schramm-sapyta, N., Wikswo, J. P., & Mclean, J. A. (2014). Phenotypic Mapping of Metabolic Profiles Using Self-Organizing Maps of High-Dimensional Mass Spectrometry Data.
- Goujon, M., McWilliam, H., Li, W., Valentin, F., Squizzato, S., Paern, J., & Lopez, R. (2010). A new bioinformatics analysis tools framework at EMBL-EBI. *Nucleic Acids Research*, 38(SUPPL. 2), 695–699. <http://doi.org/10.1093/nar/gkq313>
- Gowda, H., Ivanisevic, J., Johnson, C. H., Kurczy, M. E., Benton, H. P., Rinehart, D., ... Siuzdak, G. (2014). Interactive XCMS online: Simplifying advanced metabolomic data processing and subsequent statistical analyses. *Analytical Chemistry*, 86(14), 6931–6939. <http://doi.org/10.1021/ac500734c>
- Grapov, D., Wanichthanarak, K., & Fiehn, O. (2015). Systems biology MetaMapR : pathway independent metabolomic network analysis incorporating unknowns, 31(April), 2757–2760. <http://doi.org/10.1093/bioinformatics/btv194>
- Greef, J. Van Der, Wietmarschen, H. Van, Ommen, B. Van, & Verheij, E. (2013). LOOKING BACK INTO THE FUTURE : 30 YEARS OF METABOLOMICS AT TNO, (1977), 399–415. <http://doi.org/10.1002/mas>
- Griffiths, W. J., Koal, T., Wang, Y., Kohl, M., Enot, D. P., & Deigner, H. (2010). Targeted Metabolomics for Biomarker Discovery *Angewandte*, 5426–5445. <http://doi.org/10.1002/anie.200905579>
- Guan, W., Zhou, M., Hampton, C. Y., Benigno, B. B., Walker, L. D., Gray, A., ... Fernández, F. M. (2009). Ovarian cancer detection from metabolomic liquid chromatography/mass spectrometry data by support vector machines. *BMC Bioinformatics*, 10, 259. <http://doi.org/10.1186/1471-2105-10-259>
- Hadi, J. A., Gutierrez, N., Alfaro, A. C., & Roberts, R. D. (2014). Use of probiotic bacteria to improve growth and survivability of farmed New Zealand abalone (*Haliotis iris*). *New Zealand Journal of Marine and Freshwater Research*, 48(3), 405–415. <http://doi.org/10.1080/00288330.2014.909857>
- Hao, J., Liebeke, M., Astle, W., De Iorio, M., Bundy, J. G., & Ebbels, T. M. D. (2014). Bayesian deconvolution and quantification of metabolites in complex 1D NMR spectra using BATMAN. *Nature Protocols*, 9(6), 1416–27. <http://doi.org/10.1038/nprot.2014.090>
- Hayashi, S., Akiyama, S., Tamaru, Y., Takeda, Y., & Fujiwara, T. (2009). Biochemical and Biophysical Research Communications A novel application of metabolomics in vertebrate development. *Biochemical and Biophysical Research Communications*, 386(1), 268–272. <http://doi.org/10.1016/j.bbrc.2009.06.041>
- He, Q., Johnston, J., Zeitlinger, J., City, K., & City, K. (2015). HHS Public Access, 33(4), 395–401. <http://doi.org/10.1038/nbt.3121>.ChIP-nexus
- He, Q. P., & Wang, J. (2010). Comparison of a new spectrum alignment algorithm with other methods. *American Control Conference*, 1260–1265.
- Hefke, F., Schmucki, R., & Güntert, P. (2013). Prediction of peak overlap in

- NMR spectra. *Journal of Biomolecular NMR*, 56(2), 113–123.
<http://doi.org/10.1007/s10858-013-9727-9>
- Heinemann, J., Mazurie, A., Tokmina-Lukaszewska, M., Beilman, G. J., & Bothner, B. (2014). Application of support vector machines to metabolomics experiments with limited replicates. *Metabolomics*, 1121–1128. <http://doi.org/10.1007/s11306-014-0651-0>
- Hendriks, M. M. W. B., Eeuwijk, F. A. va., Jellema, R. H., Westerhuis, J. A., Reijmers, T. H., Hoefsloot, H. C. J., & Smilde, A. K. (2011). Data-processing strategies for metabolomics studies. *TrAC Trends in Analytical Chemistry*, 30(10), 1685–1698. <http://doi.org/10.1016/j.trac.2011.04.019>
- Hohman, M., Gregory, K., Chibale, K., Smith, P. J., Ekins, S., & Bunin, B. (2009). Novel web-based tools combining chemistry informatics, biology and social networks for drug discovery. *Drug Discovery Today*, 14(5–6), 261–270. <http://doi.org/10.1016/j.drudis.2008.11.015>
- Holland, R. C. G., Down, T. A., Pocock, M., Prli, A., Huen, D., James, K., ... Schreiber, M. J. (2008). BioJava : an open-source framework for bioinformatics, 24(18), 2096–2097.
<http://doi.org/10.1093/bioinformatics/btn397>
- Holman, J. D., Tabb, D. L., & Mallick, P. (2014). Employing ProteoWizard to Convert Raw Mass Spectrometry Data. *Current Protocols in Bioinformatics / Editorial Board, Andreas D. Baxevanis ... [et Al.]*, 46, 13.24.1-13.24.9.
<http://doi.org/10.1002/0471250953.bi1324s46>
- Horgan, R. P., & Kenny, L. C. (2011). SAC review “Omic” technologies : proteomics and metabolomics. *The Obstetrician & Gynaecologist*, 13, 189–195. <http://doi.org/10.1576/toag.13.3.189.27672>
- Jacobs, D. M., Gaudier, E., Duynhoven, J. Van, & Vaughan, E. E. (2009). Non-Digestible Food Ingredients , Colonic Microbiota and the Impact on Gut Health and Immunity : A Role for Metabolomics, (October 2016), 41–54.
<http://doi.org/10.2174/138920009787048383>
- Jae, K. K., Myoung, R. C., Hyung, J. B., Tae, H. R., Chang, Y. Y., Myong, J. K., ... Kobayashi, A. (2007). Analysis of metabolite profile data using batch-learning self-organizing maps. *Journal of Plant Biology*, 50(4), 517–521.
<http://doi.org/10.1007/BF03030693>
- Jewison, T., Su, Y., Disfany, F. M., Liang, Y., Knox, C., Maclejewski, A., ... Wishart, D. S. (2014). SMPDB 2.0: Big improvements to the small molecule pathway database. *Nucleic Acids Research*, 42(D1), 1–7.
<http://doi.org/10.1093/nar/gkt1067>
- Johnson, C. H., Ivanisevic, J., Benton, H. P., & Siuzdak, G. (2015). Bioinformatics: The next frontier of metabolomics. *Analytical Chemistry*, 87(1), 147–156. <http://doi.org/10.1021/ac5040693>
- Jones, O. A. H. (2014). *Metabolomics and Systems Biology in Human Health and Medicine*. CABI.
- Jung, J., Kim, S. H., Lee, H. S., Choi, G. S., Jung, Y. S., Ryu, D. H., ... Hwang, G. S. (2013). Serum metabolomics reveals pathways and biomarkers associated with asthma pathogenesis. *Clinical and Experimental Allergy*, 43(4), 425–433. <http://doi.org/10.1111/cea.12089>
- Kamburov, A., Cavill, R., Ebbels, T. M. D., Herwig, R., & Keun, H. C. (2011). Integrated pathway-level analysis of transcriptomics and metabolomics data with IMPaLA, 27(20), 2917–2918.
<http://doi.org/10.1093/bioinformatics/btr499>
- Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., & Tanabe, M. (2012). KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic*

- Acids Research*, 40(D1), 1–6. <http://doi.org/10.1093/nar/gkr988>
- Kanz, C., Aldebert, P., Althorpe, N., Baker, W., Baldwin, A., Bates, K., ... Apweiler, R. (2005). The EMBL Nucleotide Sequence Database, 33, 29–33. <http://doi.org/10.1093/nar/gki098>
- Karnovsky, A., Weymouth, T., Hull, T., Tarcea, V. G., Scardoni, G., Laudanna, C., ... Omenn, G. S. (2012). Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data, 28(3), 373–380. <http://doi.org/10.1093/bioinformatics/btr661>
- Kazmi, S. A., Ghosh, S., Shin, D. G., Hill, D. W., & Grant, D. F. (2006). Alignment of high resolution mass spectra: Development of a heuristic approach for metabolomics. *Metabolomics*, 2(2), 75–83. <http://doi.org/10.1007/s11306-006-0021-7>
- Kelder, T., Iersel, M. P. Van, Hanspers, K., Kutmon, M., Conklin, B. R., Evelo, C. T., & Pico, A. R. (2012). WikiPathways : building research communities on biological pathways, 40(November 2011), 1301–1307. <http://doi.org/10.1093/nar/gkr1074>
- Koek, M. M., Jellema, R. H., van der Greef, J., Tas, A. C., & Hankemeier, T. (2011). Quantitative metabolomics based on gas chromatography mass spectrometry: Status and perspectives. *Metabolomics*, 7(3), 307–328. <http://doi.org/10.1007/s11306-010-0254-3>
- Kuhl, C., Tautenhahn, R., Böttcher, C., Larson, T. R., & Neumann, S. (2012). CAMERA: An integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets. *Analytical Chemistry*, 84(1), 283–289. <http://doi.org/10.1021/ac202450g>
- Kuo, T.-C., Tian, T.-F., Tseng, Y., Kolbe, A., Oliver, S., Fernie, A., ... Siuzdak, G. (2013). 3Omics: a web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data. *BMC Systems Biology*, 7(1), 64. <http://doi.org/10.1186/1752-0509-7-64>
- Lankadurai, B. P., Nagato, E. G., & Simpson, M. J. (2013). Environmental metabolomics: an emerging approach to study organism responses to environmental stressors. *Environmental Reviews*, 21(3), 180–205. <http://doi.org/10.1139/er-2013-0011>
- Lei, Z., Huhman, D. V., & Sumner, L. W. (2011). Mass Spectrometry Strategies in Metabolomics. <http://doi.org/10.1074/jbc.R111.238691>
- Li, W. (2012). VOLCANO PLOTS IN ANALYZING DIFFERENTIAL EXPRESSIONS WITH mRNA MICROARRAYS. *Journal of Bioinformatics and Computational Biology*, 10(6), 1231003. <http://doi.org/10.1142/S0219720012310038>
- Liebeke, M., & Bundy, J. G. (2012). Tissue disruption and extraction methods for metabolic profiling of an invertebrate sentinel species. *Metabolomics*, 8(5), 819–830. <http://doi.org/10.1007/s11306-011-0377-1>
- Liland, K. H. (2011). Multivariate methods in metabolomics - from pre-processing to dimension reduction and statistical analysis. *TrAC - Trends in Analytical Chemistry*, 30(6), 827–841. <http://doi.org/10.1016/j.trac.2011.02.007>
- Lin, X., Yang, F., Zhou, L., Yin, P., Kong, H., Xing, W., ... Xu, G. (2012). A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information. *Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences*, 910, 149–155. <http://doi.org/10.1016/j.jchromb.2012.05.020>

- Lloyd, G. R., Wongravee, K., Silwood, C. J. L., Grootveld, M., & Brereton, R. G. (2009). Self Organising Maps for variable selection: Application to human saliva analysed by nuclear magnetic resonance spectroscopy to investigate the effect of an oral healthcare product. *Chemometrics and Intelligent Laboratory Systems*, 98(2), 149–161. <http://doi.org/10.1016/j.chemolab.2009.06.002>
- Lommen, A., & Kools, H. J. (2012). MetAlign 3.0: Performance enhancement by efficient use of advances in computer hardware. *Metabolomics*, 8(4), 719–726. <http://doi.org/10.1007/s11306-011-0369-1>
- Lu, H., Liang, Y., Dunn, W. B., Shen, H., & Kell, D. B. (2008). Comparative evaluation of software for deconvolution of metabolomics data based on GC-TOF-MS. *TrAC Trends in Analytical Chemistry*, 27(3), 215–227. <http://doi.org/10.1016/j.trac.2007.11.004>
- Lukas, C. K., Unker, B. H. J., & Chreiber, F. S. (2006). The VANTED software system for transcriptomics , proteomics and metabolomics analysis, 31(3), 289–292.
- Luscombe, N. M., Greenbaum, D., & Gerstein, M. (2001). Review What is bioinformatics ? An. *Gene Expression*, 40(4), 83–100. <http://doi.org/10.1053/j.ro.2009.03.010>
- Magrane, M., & Consortium, U. (2011). Original article UniProt Knowledgebase : a hub of integrated protein data, 2011, 1–13. <http://doi.org/10.1093/database/bar009>
- Mahadevan, S., Shah, S. L., Marrie, T. J., & Slupsky, C. M. (2008). Analysis of metabolomic data using support vector machines. *Anal Chem*, 80(19), 7562–7570. <http://doi.org/10.1021/ac800954c>
- Martens, L., Chambers, M., Sturm, M., Kessner, D., Levander, F., Shofstahl, J., ... Deutsch, E. W. (2011). mzML — a Community Standard for Mass Spectrometry Data *, 1–7. <http://doi.org/10.1074/mcp.R110.000133>
- Mushtaq, M. Y., Choi, Y. H., Verpoorte, R., & Wilson, E. G. (2014). Extraction for metabolomics: Access to the metabolome. *Phytochemical Analysis*, 25(4), 291–306. <http://doi.org/10.1002/pca.2505>
- Nagana Gowda, G. A., Zhang, S., Gu, H., Asiago, V., Shanaiah, N., & Raftery, D. (2008). Metabolomics-Based Methods for Early Disease Diagnostics: A Review. *Expert Review of Molecular Diagnostics*, 8(5), 617–633. <http://doi.org/10.1586/14737159.8.5.617>
- Nordstrom, a, O'Maille, G., Qin, C., & Siuzdak, G. (2006). Nonlinear data alignment for UPLC MS and HPLC MS based metabolomics: quantitative analysis of endogenous and exogenous metabolites in human serum. *Analytical Chemistry*, 78(10), 3289–3295. <http://doi.org/10.1021/ac060245f>
- Nováková, L., & Vlčková, H. (2009). A review of current trends and advances in modern bio-analytical methods: Chromatography and sample preparation. *Analytica Chimica Acta*, 656(1–2), 8–35. <http://doi.org/10.1016/j.aca.2009.10.004>
- O’Gorman, A., Gibbons, H., & Brennan, L. (2013). Metabolomics in the Identification of Biomarkers of Dietary Intake. *Computational and Structural Biotechnology Journal*, 4(5), 1–7. <http://doi.org/10.5936/csbj.201301004>
- Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., & Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 27(1), 29–34. <http://doi.org/10.1093/nar/27.1.29>
- Oh, J. H., Craft, J. M., Townsend, R., Deasy, J. O., Bradley, J. D., & Naqa, I. El. (2011). A Bioinformatics Approach for Biomarker Identification in Radiation-Induced Lung Inflammation from Limited Proteomics Data. *Proteome*,

1406–1415.

- Okuda, S., Yamada, T., Hamajima, M., Itoh, M., Bork, P., Goto, S., & Kanehisa, M. (2008). KEGG Atlas mapping for global analysis of metabolic pathways, 36(May), 423–426. <http://doi.org/10.1093/nar/gkn282>
- Ozdemir, V., Suarez-kurtz, G., Stenne, R., Somogyi, A. A., Someya, T., & Og, S. (2009). Risk Assessment and Communication Tools for Genotype Associations with Multifactorial Phenotypes : The Concept of “ Edge Effect ” and Cultivating an Ethical Bridge between Omics Innovations and Society, 13(1). <http://doi.org/10.1089/omi.2009.0011>
- Patel, S., & Ahmed, S. (2015). Emerging field of metabolomics: Big promise for cancer biomarker identification and drug discovery. *Journal of Pharmaceutical and Biomedical Analysis*, 107, 63–74. <http://doi.org/10.1016/j.jpba.2014.12.020>
- Pedrioli, P. G. A., Eng, J. K., Hubley, R., Vogelzang, M., Deutsch, E. W., Raught, B., ... Zhu, W. (2004). A common open representation of mass spectrometry data and its application to proteomics research, 22(11), 1459–1466. <http://doi.org/10.1038/nbt1031>
- Perl, A., Hanczko, R., Lai, Z. W., Oaks, Z., Kelly, R., Borsuk, R., ... Phillips, P. E. (2015). Comprehensive metabolome analyses reveal N-acetylcysteine-responsive accumulation of kynurenine in systemic lupus erythematosus: implications for activation of the mechanistic target of rapamycin. *Metabolomics*, 11(5), 1157–1174. <http://doi.org/10.1007/s11306-015-0772-0>
- Pluskal, T., Castillo, S., Villar-Briones, A., & Oresic, M. (2010). MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics*, 11, 395. <http://doi.org/10.1186/1471-2105-11-395>
- Ponnusamy, K., Choi, J. N., Kim, J., Lee, S. Y., & Lee, C. H. (2011). Microbial community and metabolomic comparison of irritable bowel syndrome faeces. *Journal of Medical Microbiology*, 60(6), 817–827. <http://doi.org/10.1099/jmm.0.028126-0>
- Raamsdonk, L. M., Teusink, B., Broadhurst, D., Zhang, N., Hayes, A., Walsh, M. C., ... Oliver, S. G. (2001). A functional genomics strategy that uses metabolome data to reveal the phenotype of silent mutations. *Nat Biotechnol*, 19(1), 45–50. <http://doi.org/10.1038/83496>
- Reimand, J., Arak, T., Adler, P., Kolberg, L., Reisberg, S., Peterson, H., & Vilo, J. (2016). g:Profiler—a web server for functional interpretation of gene lists (2016 update). *Nucleic Acids Research*, 1–7. <http://doi.org/10.1093/nar/gkw199>
- Reimers, A. C. (2015). Hierarchical decomposition of metabolic networks using k-modules. *Biochemical Society Transactions*, 43(6), 1146–1150.
- Roberts, L. D., Souza, A. L., Gerszten, R. E., & Clish, C. B. (2012). Targeted Metabolomics. *Current Protocols in Molecular Biology*, CHAPTER, Unit30.2-Unit30.2. <http://doi.org/10.1002/0471142727.mb3002s98>
- Robertson, D. G., & Frevert, U. (2013). Metabolomics in drug discovery and development. *Clinical Pharmacology and Therapeutics*, 94(5), 559–61. <http://doi.org/10.1038/clpt.2013.120>
- Robertson, D. G., Watkins, P. B., & Reilly, M. D. (2011). Metabolomics in toxicology: Preclinical and clinical applications. *Toxicological Sciences*, 120(SUPPL.1), 1–85. <http://doi.org/10.1093/toxsci/kfq358>
- Romero, R., Espinoza, J., Gotsch, F., Kusanovic, J. P., Friel, L. A., Erez, O., ... Tromp, G. (2006). The use of high-dimensional biology (genomics,

- transcriptomics, proteomics, and metabolomics) to understand the preterm parturition syndrome. *BJOG: An International Journal of Obstetrics and Gynaecology*, 113(SUPPL. 3), 118–135. <http://doi.org/10.1111/j.1471-0528.2006.01150.x>
- Römisch-Margl, W., Prehn, C., Bogumil, R., Röhring, C., Suhre, K., & Adamski, J. (2012). Procedure for tissue sample preparation and metabolite extraction for high-throughput targeted metabolomics. *Metabolomics*, 8(1), 133–142. <http://doi.org/10.1007/s11306-011-0293-4>
- Rubingh, C. M., Bijlsma, S., Derks, E. P. P. A., Bobeldijk, I., Verheij, E. R., Kochhar, S., & Smilde, A. K. (2006). Assessing the performance of statistical validation tools for megavariate metabolomics data. *Metabolomics*, 2(2), 53–61. <http://doi.org/10.1007/s11306-006-0022-6>
- Sato, Y., Suzuki, I., Nakamura, T., Bernier, F., Aoshima, K., & Oda, Y. (2012). Identification of a new plasma biomarker of Alzheimer's disease using metabolomics technology. *The Journal of Lipid Research*, 53(3), 567–576. <http://doi.org/10.1194/jlr.M022376>
- Shulaev, V. (2006). Metabolomics technology and bioinformatics, 7(2), 128–139. <http://doi.org/10.1093/bib/bbl012>
- Smith, C. A., Maille, G. O., Want, E. J., Qin, C., Trauger, S. A., Brandon, T. R., ... Siuzdak, G. (2005). METLIN: A Metabolite Mass Spectral Database. *Therapeutic Drug Monitoring*, 27(6). Retrieved from http://journals.lww.com/drug-monitoring/Fulltext/2005/12000/METLIN__A_Metabolite_Mass_Spectral_Database.16.aspx
- Smith, C. A., Want, E. J., Maille, G. O., Abagyan, R., & Siuzdak, G. (2006). <Smith_2006_XCMS.pdf>, 78(3), 779–787. <http://doi.org/10.1021/ac051437y>
- Smoot, M. E., Ono, K., Ruscheinski, J., Wang, P., & Ideker, T. (2011). Cytoscape 2 . 8 : new features for data integration and network visualization, 27(3), 431–432. <http://doi.org/10.1093/bioinformatics/btq675>
- Staab, J. M.; O'Connell, T. M.; Gomez, S. M. (2010). Enhancing metabolomic data analysis with Progressive Consensus Alignment of NMR Spectra (PCANS). *BMC Bioinf.*, 11, 123. <http://doi.org/10.1186/1471-2105-11-123>
- Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigan, C., ... Birney, E. (2002). The Bioperl Toolkit : Perl Modules for the Life Sciences, 1611–1618. <http://doi.org/10.1101/gr.361602.the>
- Steuer, R., Morgenthal, K., Weckwerth, W., & Selbig, J. (2007). A gentle guide to the analysis of metabolomic data. *Methods in Molecular Biology (Clifton, N.J.)*, 358, 105–126. http://doi.org/10.1007/978-1-59745-244-1_7
- Sumner, S. J., Burgess, J. P., Snyder, R. W., Popp, J. A., & Fennell, T. R. (2010). Metabolomics of urine for the assessment of microvesicular lipid accumulation in the liver following isoniazid exposure. *Metabolomics*, 6(2), 238–249. <http://doi.org/10.1007/s11306-010-0197-8>
- Sun, X., & Weckwerth, W. (2013). Using COVAIN to Analyze Metabolomics Data. In *The Handbook of Plant Metabolomics* (pp. 305–320). Wiley-VCH Verlag GmbH & Co. KGaA. <http://doi.org/10.1002/9783527669882.ch17>
- szántay, C. (2007). NMR and the uncertainty principle: How to and how not to interpret homogeneous line broadening and pulse nonselectivity. I. The fundamentals. *Concepts in Magnetic Resonance Part A*, 30A(6), 309–348. <http://doi.org/10.1002/cmr.a.20098>
- Tapp, H. S., & Kemsley, E. K. (2009). Notes on the practical utility of OPLS. *TrAC - Trends in Analytical Chemistry*, 28(11), 1322–1327.

- <http://doi.org/10.1016/j.trac.2009.08.006>
- Tautenhahn, R., Patti, G. J., Rinehart, D., & Siuzdak, G. (2012). XCMS online: A web-based platform to process untargeted metabolomic data. *Analytical Chemistry*, 84(11), 5035–5039. <http://doi.org/10.1021/ac300698c>
- Trupp, M., Altman, T., Fulcher, C., Caspi, R., Krummenacker, M., Paley, S., & Karp, P. (2010). Beyond the genome (BTG) is a (PGDB) pathway genome database: HumanCyc. *Genome Biol.*, 11(Suppl 1), O12. <http://doi.org/10.1186/gb-2010-11-s1-o12>
- Trygg, J., & Wold, S. (2002). Orthogonal projections to latent structures (O-PLS). *Journal of Chemometrics*, 16(3), 119–128. <http://doi.org/10.1002/cem.695>
- Tsuji, H., Kurokawa, K., & Asahi, H. (2006). DPCLus : A density-periphery based graph clustering software mainly focused on detection of protein complexes in interaction networks Instruction Manual 1 . Starting DPCLus.
- Ummanni, R., Mundt, F., Pospisil, H., Venz, S., Scharf, C., Barrett, C., ... Balabanov, S. (2011). Identification of clinically relevant protein targets in prostate cancer with 2D-DIGE coupled mass spectrometry and systems biology network platform. *PLoS ONE*, 6(2). <http://doi.org/10.1371/journal.pone.0016833>
- Urbanek, S. (2003). A fast way to provide R functionality to applications.
- Van Emon, J. M. (2016). The Omics Revolution in Agricultural Research. *Journal of Agricultural and Food Chemistry*, 64(1), 36–44. <http://doi.org/10.1021/acs.jafc.5b04515>
- Vey, S., & Voigt, A. (2007). AMDiS: Adaptive multidimensional simulations. *Computing and Visualization in Science*, 10(1), 57–67. <http://doi.org/10.1007/s00791-006-0048-3>
- Vinayavekhin, N., & Saghatelian, A. (2001). Untargeted Metabolomics. In *Current Protocols in Molecular Biology*. John Wiley & Sons, Inc. <http://doi.org/10.1002/0471142727.mb3001s90>
- Vlaanderen, J., Moore, L. E., Smith, M. T., Lan, Q., Zhang, L., Skibola, C. F., ... Vermeulen, R. (2010). Application of OMICS technologies in occupational and environmental health research; current status and projections. *Occupational and Environmental Medicine*, 67(2), 136–43. <http://doi.org/10.1136/oem.2008.042788>
- Vu, T. N., & Laukens, K. (2013). Getting your peaks in line: a review of alignment methods for NMR spectral data. *Metabolites*, 3(2), 259–76. <http://doi.org/10.3390/metabo3020259>
- Vuckovic, D. (2012). Current trends and challenges in sample preparation for global metabolomics using liquid chromatography-mass spectrometry. *Analytical and Bioanalytical Chemistry*, 403(6), 1523–1548. <http://doi.org/10.1007/s00216-012-6039-y>
- Wang, X., Zhang, A., & Sun, H. (2013). Power of metabolomics in diagnosis and biomarker discovery of hepatocellular carcinoma. *Hepatology*, 57(5), 2072–2077. <http://doi.org/10.1002/hep.26130>
- Weckwerth, W. (2008). Integration of metabolomics and proteomics in molecular plant physiology - Coping with the complexity by data-dimensionality reduction. *Physiologia Plantarum*, 132(2), 176–189. <http://doi.org/10.1111/j.1399-3054.2007.01011.x>
- Weljie, A. M., Newton, J., Mercier, P., Carlson, E., & Slupsky, C. M. (2006). Targeted profiling: quantitative analysis of ¹H NMR metabolomics data. *Analytical Chemistry*, 78(13), 4430–4442. <http://doi.org/10.1021/ac060209g>
- Whitley, E., & Ball, J. (2002). Statistics review 1: presenting and summarising data. *Critical Care (London, England)*, 6(1), 66–71.

- <http://doi.org/10.1186/cc1820>
- Wishart, D. S. (2008). Quantitative metabolomics using NMR. *TrAC - Trends in Analytical Chemistry*, 27(3), 228–237.
<http://doi.org/10.1016/j.trac.2007.12.001>
- Wishart, D. S., Tzur, D., Knox, C., Eisner, R., Guo, A. C., Young, N., ... Querengesser, L. (2007). HMDB : the Human Metabolome Database, 35, 521–526. <http://doi.org/10.1093/nar/gkl923>
- Worley, B., & Powers, R. (2015). WorleyPowers-CurrMetab-2013, 1(1), 92–107. <http://doi.org/10.2174/2213235X11301010092>
- Wrzodek, C., Eichner, J., Büchel, F., & Zell, A. (2013). InCroMAP: Integrated analysis of cross-platform microarray and pathway data. *Bioinformatics*, 29(4), 506–508. <http://doi.org/10.1093/bioinformatics/bts709>
- Xi, B., Gu, H., Baniasadi, H., & Raftery, D. (2014). Statistical analysis and modeling of mass spectrometry-based metabolomics data. *Methods in Molecular Biology*, 1198, 333–353. http://doi.org/10.1007/978-1-4939-1258-2_22
- Xi, Y., & Rocke, D. M. (2008). Baseline correction for NMR spectroscopic metabolomics data analysis. *BMC Bioinformatics*, 9(1), 324. <http://doi.org/10.1186/1471-2105-9-324>
- Xia, J., Broadhurst, D. I., Wilson, M., & Wishart, D. S. (2013). Translational biomarker discovery in clinical metabolomics: An introductory tutorial. *Metabolomics*, 9(2), 280–299. <http://doi.org/10.1007/s11306-012-0482-9>
- Xia, J., Mandal, R., Sinelnikov, I. V., Broadhurst, D., & Wishart, D. S. (2012). MetaboAnalyst 2.0-a comprehensive server for metabolomic data analysis. *Nucleic Acids Research*, 40(W1), 127–133. <http://doi.org/10.1093/nar/gks374>
- Xia, J., Psychogios, N., Young, N., & Wishart, D. S. (2009). MetaboAnalyst: A web server for metabolomic data analysis and interpretation. *Nucleic Acids Research*, 37(SUPPL. 2), 652–660. <http://doi.org/10.1093/nar/gkp356>
- Xia, J., Sinelnikov, I. V., Han, B., & Wishart, D. S. (2015a). MetaboAnalyst 3.0-making metabolomics more meaningful. *Nucleic Acids Research*, 43(W1), W251–W257. <http://doi.org/10.1093/nar/gkv380>
- Xia, J., Sinelnikov, I. V., Han, B., & Wishart, D. S. (2015b). MetaboAnalyst 3 . 0 — making metabolomics more meaningful, 43(April), 251–257. <http://doi.org/10.1093/nar/gkv380>
- Xia, J., & Wishart, D. S. (2010). MSEA: A web-based tool to identify biologically meaningful patterns in quantitative metabolomic data. *Nucleic Acids Research*, 38(SUPPL. 2), 71–77. <http://doi.org/10.1093/nar/gkq329>
- Xia, J., Wishart, D. S., & Valencia, A. (2011). MetPA: A web-based metabolomics tool for pathway analysis and visualization. *Bioinformatics*, 27(13), 2342–2344. <http://doi.org/10.1093/bioinformatics/btq418>
- Yan, S.-K., Liu, R.-H., Jin, H.-Z., Liu, X.-R., Ye, J., Shan, L., & Zhang, W.-D. (2015). “Omics” in pharmaceutical research: overview, applications, challenges, and future perspectives. *Chinese Journal of Natural Medicines*, 13(1), 3–21. [http://doi.org/10.1016/S1875-5364\(15\)60002-4](http://doi.org/10.1016/S1875-5364(15)60002-4)
- Yang, Z., Nakabayashi, R., Okazaki, Y., Mori, T., Takamatsu, S., Kitanaka, S., ... Saito, K. (2014). Toward better annotation in plant metabolomics: Isolation and structure elucidation of 36 specialized metabolites from *Oryza sativa* (rice) by using MS/MS and NMR analyses. *Metabolomics*, 10(4), 543–555. <http://doi.org/10.1007/s11306-013-0619-5>
- Young, T., & Alfaro, A. (2016a). Metabolomic strategies for aquaculture research : A primer Metabolomic strategies for aquaculture research : a

- primer, (May). <http://doi.org/10.1111/raq.12146>
- Young, T., & Alfaro, A. C. (2014). New Zealand Journal of Marine and Freshwater Research Identification of candidate biomarkers for quality assessment of hatchery- reared mussel larvae via GC / MS-based metabolomics, (May 2015), 37–41.
<http://doi.org/10.1080/00288330.2014.958504>
- Young, T., & Alfaro, A. C. (2016b). Metabolomic strategies for aquaculture research : a primer, 1–31. <http://doi.org/10.1111/raq.12146>
- Young, T., Alfaro, A. C., & Villas-bo, S. G. (2016). Metabolic profiling of mussel larvae : effect of handling and culture conditions, 843–856.
<http://doi.org/10.1007/s10499-015-9945-0>
- Young, T., Alfaro, A., & Villas-Bôas, S. (2015). Identification of candidate biomarkers for quality assessment of hatchery-reared mussel larvae via GC/MS-based metabolomics. *New Zealand Journal of Marine and Freshwater Research*, 49(1), 87–95.
<http://doi.org/10.1080/00288330.2014.958504>
- Zhang, A., Qiu, S., Xu, H., Sun, H., & Wang, X. (2014). Clinica Chimica Acta Metabolomics in diabetes. *Clinica Chimica Acta*, 429, 106–110.
<http://doi.org/10.1016/j.cca.2013.11.037>
- Zhang, A., Sun, H., Wang, P., Han, Y., & Wang, X. (2012). MINIREVIEW Modern analytical techniques in metabolomics analysis †, 293–300.
<http://doi.org/10.1039/c1an15605e>
- Zhang, A., Sun, H., & Wang, X. (2013). Power of metabolomics in biomarker discovery and mining mechanisms of obesity. *Obesity Reviews*, 14(4), 344–349. <http://doi.org/10.1111/obr.12011>
- Zhang, W., Li, F., & Nie, L. (2010). Integrating multiple “omics” analysis for microbial biology: Application and methodologies. *Microbiology*, 156(2), 287–301. <http://doi.org/10.1099/mic.0.034793-0>
- Zhang, Z., Chen, S., & Liang, Y. (2010). Baseline correction using adaptive iteratively reweighted penalized least squares, 13(2), 1138–1146.
<http://doi.org/10.1039/b922045c>