Novel Preprocessing Approaches for Omics Data Types and Their Performance Evaluation

Dario Strbenac

A thesis submitted in fulfilment of the requirements for
the degree of Doctor of Philosophy

School of Mathematics and Statistics
The University of Sydney
2 December 2016
I certify that the content of this thesis is my own work. This thesis has not been submitted for any other degree or for other purposes. All of the assistance received in preparing this thesis and sources have been acknowledged.

D. Strickland
Abstract

A diverse range of high-dimensional datasets has recently become available to help elucidate the functioning of biological systems and defects within those systems leading to disease. This improved understanding will aid our knowledge of fundamental biology as well as increasing our comprehension of the processes that are altered in complex disease. All of these new technologies come with the challenges of determining how the raw data should be efficiently processed or normalised and, subsequently, how can the data best be summarised for more complex downstream analysis. There are many approaches to summarising and normalising omics data, with new methods frequently being developed. Different kinds of omics data may also be integrated, in order to provide more confidence in predictions. To date, there has not been a comprehensive evaluation of existing methods for many omics data types. This thesis focusses on systematically evaluating existing methods for three different types of omics data and, having identified limitations in the current methods, also proposes new approaches to improve their quality.

Firstly, CAGE-seq data are considered. This type of data has unique characteristics such that regional summarisation algorithms developed for similar experiments, such as ChIP-seq, are not directly applicable. Additionally, the raw data also contain artefactual measurements from confounding biological processes, and a comprehensive evaluation of region classification algorithms has not previously been carried out. A two-stage method based on a novel region-finding algorithm followed by a classifier that integrates sequence patterns surrounding the identified regions is shown to possess superior performance to two existing methods. Similarly, a novel data summarisation approach to gene expression data, which integrates changes in location and scale into a unified metric, demonstrates benefits in two-class classification problems. The error rates are found to be competitive with existing methods, and the feature selection has higher stability and increased biological relevance. Finally, in the proteomics setting, there are many choices for how to summarise peptides to proteins, as well as issues relating to batch effects and whether internal controls are necessary. By developing a broad variety of performance metrics that assess bias or variance, and an accompanying web-based framework for reproducible research, novel recommendations about peptide to protein summaries and batch correction algorithms are made, and a surprising result regarding the necessity of internal standards is revealed. The development and evaluation of novel dataset preprocessing approaches and the comprehensive evaluation of existing methods for three data types demonstrates the importance of systematic
performance evaluation of statistical bioinformatics methods for more accurate and precise knowledge generation in modern biology.
Publications and Presentations

Some of the research in this thesis has appeared in peer-reviewed journals or conferences.

Publications


Presentations


Statement of Contribution

For: Summarisation and Classification for CAGE-seq Data

The research presented in Chapter 2 has previously been published in the journal *BMC Genomics* and I am the corresponding author for the article. The project was motivated by Dario’s previous employment at the Garvan Institute, where he briefly worked on CAGE-seq analysis. Dario developed much of the two-stage TSS region algorithm and implemented it and the associated case studies in R. Features for the classifier were suggested and agreed upon by all participants. Classifier design involved input from all three participants. All publicly used datasets for integration were obtained and processed by Dario. The first draft of the journal article was written by Dario and it subsequently had major contributions from all three participants. Some additional unpublished evaluations are presented in Sections 2.2.1.1 and 2.3.

Corresponding Author: Prof. Jean Yang
Acknowledgements

In the academic context, I am thankful to my supervisors Professor Jean Yang and Dr. Nicola Armstrong for their inspiration and guidance on this long and challenging journey. They have always encouraged me to try more ideas when a proposed idea failed to meet expectations. Thanks also to Dr. John Ormerod for his guest supervision of the differential distribution project. His unique views of the problem helped to improve the work to a level which is of importance to a wide readership. The motivation for the project was partially provided by Professor Graham Mann, who has contributed many biological insights to the research, which ensures its translational relevance. All supervisors also provided detailed advice in regards to academic writing, which improved the clarity of the work and the associated results. Also, I would like to acknowledge the infrastructure provided by the Department of Mathematics and Statistics. Without the two large-memory, many-CPU servers available, computationally-intensive tasks such as large cross-validation would not be completed in good time. Finally, thanks to Sarah-Jane Schramm and Shila Ghazanfar for each proofreading a chapter.

The collaborators at the Bioanalytical Mass Spectrometry Facility at University of New South Wales have been a crucial part of the proteomics normalisation project. To complete the experiment, they had to dispense 21 proteins into 56 different tubes at a different volume in each tube. Such a difficult experiment has not been attempted before, and their patience and diligence enabled us to have the ideal dataset for methodology performance evaluation. Thanks are also due to Professor Susan Wilson for providing top-up salary funding, enabling the research to be properly and reproducibly done.

In the social context, I would like to thank my parents Drago and Darinka for the many visits they made to Sydney and the frequent assistance with domestic duties they provided. Last, but not least, I thank Jasmine, my girlfriend, for her patience and positive thinking.
# Table of Contents

1 Introduction .......................................................................................................................... 1
1.1 Fundamentals of Molecular Biology .............................................................................. 1
1.2 Overview of Measurement Technologies and Their Data Types .................................. 3
   1.2.1 Microarrays ............................................................................................................. 4
   1.2.2 High-throughput Sequencing ................................................................................ 4
   1.2.3 Mass Spectrometry-based Proteomics ................................................................. 6
   1.2.4 Biological Annotation of Omics Data ................................................................. 7
1.3 Omics Data Methods Evaluation Challenges ............................................................... 8
1.4 Thesis Outline .................................................................................................................. 10

2 Summarisation and Classification for CAGE-seq Data ...................................................... 11
2.1 Background .................................................................................................................... 12
   2.1.1 ChIP-seq ............................................................................................................. 13
   2.1.2 DNase-seq ........................................................................................................... 14
   2.1.3 RNA-seq ............................................................................................................. 15
   2.1.4 CAGE-seq ........................................................................................................... 15
   2.1.5 HTS Mapping ...................................................................................................... 19
2.2 Two-stage Preprocessing Algorithm ............................................................................ 20
   2.2.1 Region-Finding .................................................................................................... 20
   2.2.2 Classification of Regions .................................................................................... 22
2.3 Case Study: Prostate Cancer ......................................................................................... 28
2.4 Discussion and Conclusion ........................................................................................... 30

3 Differential Distribution for Binary Classification Problems ............................................ 33
3.1 Background .................................................................................................................... 34
3.2 Feature Selection and Classification .............................................................................. 38
3.3 Evaluation Metrics ......................................................................................................... 44
3.4 Simulation Evaluation of DV and DD Varieties ............................................................ 45
3.5 Evaluation of DE, DV, and DD ..................................................................................... 51
3.6 ClassifyR Classification Evaluation Framework ........................................................... 59
3.7 Discussion and Conclusion ........................................................................................... 61

4 Performance of Proteomics Summarisation and Normalisation Methods ...................... 64
4.1 Background .................................................................................................................... 66
   4.1.1 Mass Spectrometry-based Proteomics ................................................................. 66
   4.1.2 Evaluation of Quantitation .................................................................................. 69
List of Figures

Figure 1.1 **The Central Dogma of Molecular Biology.** The DNA in a cell exists as a pair of strands of nucleotides that are bonded (vertical grey lines). The sequences of nucleotides contain instructions for making RNA molecules. Once the information is transcribed to RNA form, it is usually translated into a sequence of amino acids termed a protein, although it can be converted back into DNA. Adapted from Fu et al. (2014, p. 294). .............................................................. 2

Figure 1.2 **The DNA double helix.** Each strand has a 5’ and a 3’ end, giving the two strands an antiparallel orientation. The two possible bonds are C to G and A to T Adapted from Becker et al. (2008, p.59). .............................................................. 2

Figure 1.3 **Two Ways to Measure DNA and RNA.** For RNA experiments, the molecules are firstly converted into DNA. Amplification of DNA is necessary to create an adequate number of measurable molecules. The molecules have to be broken into shorter fragments to allow them to be measured. Microarrays use fluorescent labelling of fragmented molecules and their binding to probes arrayed on a glass surface. HTS determines the sequence of the fragmented molecules. The count of sequences originating from a genomic region is proportional to its cellular abundance. This figure is hand-drawn.............................................................. 6

Figure 2.1 **Basic steps of histone ChIP-seq.** The black line is part of a chromosome. The grey cylinders are histones. The green circle represents a histone modification of interest. Firstly, the DNA is fragmented, and then only the histones that have the modification of interest are purified. The histones are then removed so that only the DNA remains and its sequence is determined. Adapted from Park (2009, p. 671)... 14
Figure 2.2 Basic steps of DNase-seq. The black line is part of a chromosome. The scissors are representative of DNase I. The enzyme cuts in location that are not bound to protein, represented by orange and grey shapes. The resulting DNA fragments are sequenced to determine regions of unbound DNA. Adapted from Zentner and Henikoff (2014, p. 816).

Figure 2.3 Key steps in the workflow for analysing CAGE-seq data. The cDNA sequences obtained from the sequencer (Step 1) are aligned to the reference genome (Step 2). Only the first position of each read is retained, and the retained positions are clustered into regions (Step 3). Lastly, a classification algorithm is used to label the regions as being from transcription initiation or not (Step 4). The focus of this study is on steps 3 and 4.

Figure 3.1 Summary of feature types and classifiers. For each of differential expression, differential variability, and differential distribution, a representative gene expression distribution is shown, along with an illustration of the classification process. In the left column, the dashed vertical lines represent the means of the class distributions. In the right column, the variables $x$ and $y$ denote two different genes in a dataset. Each point indicates a sample. The bottom right panel illustrates that each gene from the selected gene set votes independently in differential distribution classification. The black circle shows the position along the x-axis of the expression measurement value. The intervals show two kinds of distance weighting evaluated in the simulation study. The green interval corresponds to crossover distance voting and the black interval to height difference voting.
Figure 3.2 Feature selection proportions and balanced error rates of DV classification for seven simulated datasets. A. Proportions of selected genes. The average percentage of selected genes that are in the specified simulated change categories over all cross-validations is shown. The bottom row shows the proportions of simulated changes. B. Balanced error rates of class predictions. The distributions of error rates across all cross-validation iterations are shown as boxplots. 48

Figure 3.3 Feature selection proportions and balanced error rates of DD classification for seven simulated datasets. A. Proportions of selected genes. The average percentage of selected genes that are in the specified simulated change categories over all cross-validations is shown. The fifth row shows the proportions of simulated changes. B. Balanced error rates of class predictions. The distributions of error rates across all cross-validation iterations are shown as boxplots. 50

Figure 3.4 Cross-validation balanced error rates and sample-wise error rates. A. Distribution of balanced error rates over all iterations of cross-validation. B. Sample-wise error rates. Each patient is one column of a heatmap. Each classification type is one row of a heatmap. Details of the selection and classifier algorithms are provided in the Methods section. The error rates are binned into five equally sized bins. Colour scales are shaded by class colour, with a darker colour indicating less frequent misclassification than a lighter colour. 54

Figure 3.5 Feature ranking and selection overlaps. A. Overlaps between three feature selection types for three cancer datasets. The 50 highest-ranked genes of each method are used. All samples are used. B. Overlaps between three feature selection approaches for three cancer datasets. The genes selected by best resubstitution
error rate of each method are used. All samples are used. C. Cumulative MalaCards scores for the most frequently selected features in cross-validation. .......................... 56

Figure 3.6 **Cross-validation feature ranking and selection stability.** For each pair of comparisons, the number of genes in common is divided by the number of genes in the union and converted to a percentage. A. The average pairwise overlap of the top ranked genes is calculated for all iterations of cross-validation. Shapes represent datasets and colours represent different types of classification. B. The distribution of the pairwise overlaps of the selected genes is calculated for all iterations of cross-validation. From left to right, the number of data points which are greater than 20% and are not shown as points is: 475, 16879, 5301, 305, 593, 5884, 963, 2489, and 29384. .............................................................................................................................................. 58

Figure 4.1 **Flowchart of proteomics data preprocessing.** Steps of preprocessing are represented by boxes and particular methods to evaluate are listed on the right side. Grey steps or option lists are not evaluated in this study. ........................................ 69

Figure 4.2 **Overview of the web-based application for evaluation.** A. The New Processing Upload page allows users to describe various aspects of their data preprocessing and to upload the protein quantities or fold changes as a text file to the server. The web-based application calculates all of the performance metrics and then automatically switches to the Evaluation Matrix page. B. The Evaluation Matrix page displays the matrix of performance metrics and various tools to interact with it. The numbers in the table are summaries, either the mean or standard deviation of the metric, of either all experimental runs or all iTRAQ channels, where applicable. Filtering, sorting, row and column selection are used to subset the matrix to the
comparison of interest. Full metric names can be seen by hovering the mouse over the column number. Numerical and graphical evaluation tools are located below the metric matrix. ........................................................................................................................................ 82

Figure 4.3 **Performance metrics of regression for relative quantitation using the default processing parameters of ProteinPilot 5.** Horizontal red line represents the ideal metric value. Metrics are grouped either by iTRAQ channel (first row) or run (second row). ........................................................................................................................................ 85

Figure 4.4 **Performance metrics for relative quantitation.** Horizontal red line represents the ideal metric value. ........................................................................................................................................ 90

Figure 4.5 **Performance metrics for absolute quantitation.** Horizontal red line represents the ideal metric value. **A.** The distribution of metric values of a linear regression of protein summaries within a sample divided by the median protein group’s median summary of the sample, grouped either by iTRAQ channel or experimental run. **B.** Summaries for three variance metrics calculated using data from either one run or all runs. ........................................................................................................................................ 92

Figure A-1 **Raw counts and detected regions for sample GM12878.** The first two rows show the counts at each position of the first positions of CAGE reads. The Genes row shows a model of known genes in this region. The left-most edge of the blue rectangle is a known TSS. The thin blue lines are known introns. The bottom three rows show the regions determined by each of the three methods compared. The angle brackets show the strand on which the regions are. “>” represents the plus strand and “<” represents the minus strand. ........................................................................................................................................ 101
Figure A-2 **Principal components analysis of the scalar features surrounding the CAGE read regions.** The top panel shows the TSS category of regions and the bottom panel shows those regions which are labelled as Not TSS in the truth set. Dots represent the 100 observations in the lowest density regions.

Figure A-3 **Principal components analysis of the 4-mers surrounding the CAGE read regions.** The analysis was performed on the correlation matrix of the 512 4-mers. The top panel shows the TSS category of regions and the bottom panel shows those regions which are labelled as Not TSS in the truth set. Dots represent the 100 observations in the lowest density regions.

Figure A-4. **Precision and recall for three feature scenarios.** Precision and recall are calculated at each cost parameter value based on a LOOCV scheme. Blue lines are precision. Red lines are recall. Horizontal bars or dots represent the minimum and maximum value of all cell lines. Points on the line are averages across all six cell lines for the two leftmost panels and two cell lines for the rightmost panel.

Figure A-5. **CAGE count summaries for the genes with the best differential expression and TSS switching statistics.** The top two rows of data show normalised count values per genomic position in each of the two experimental conditions. The third row of data shows the structure of known genes. The bottom row shows the regions which have been classified as TSS regions.

Figure B-1 **Distribution of survival times for the three cancer datasets used for classification evaluation.** Densities are estimated with the default Gaussian kernel in R.
Figure B-2 **Most frequently chosen genes’ expression profiles in cross-validation for three datasets and three feature types.** The RefSeq symbol of the gene the densities are plotted for is shown above each density plot. DV selection is made by ranking of the Bartlett’s test statistic and DD selection is made by the proposed DMD statistic. …………………………………………………………………………………112

Figure C-1 **Relationship between length of spike-in protein and number of peptides detected in experiment.** The number of peptides associated to a spike-in protein based on matching by ProteinPilot’s Paragon algorithm is shown for all seven runs. ………………………………………………………………………………………………………117

Figure C-2 **Principal components analysis of background proteins.** Seven yeast proteins were quantified in all seven runs. For each sample, the dimensionality has been reduced to two dimensions. Each row of plots contains yeast protein measurements summarised from peptides in one of the three summary methods used in this study. The first column has samples coloured by experimental run and the second column has samples coloured by iTRAQ channel. ……………………………118
List of Tables

Table 2.1 **Number of regions detected for each cell line.** The proposed sliding window, Paraclu and F-seq are evaluated. ................................................................. 21

Table 2.2 **Percentage overlap between proposed method’s regions and those found by Paraclu and F-seq.** Regions must have at least one base overlap and be on the same genomic strand to be considered as overlapping. ........................................... 22

Table 2.3 **Features considered for TSS classification.** For each feature, the summarisation procedure, location of data points summarised, and the feature categorisation are shown. ............................................................................................................... 25

Table 2.3 **Precision and recall values for the internal features LOOCV classification.** For each cell line, classifier training was performed on the five other cell lines. A cost parameter of 0.1 is used for the linear SVM classifier. .............................................. 28

Table 4.1 **Experimental Design for Run 1.** The values in the table are volumes of diluted protein used, in units of microlitres. ........................................................................................................... 75

Table 4.2 **Summary of performance metrics and their characteristics.** The applicability of a metric to the quantitation type, preprocessing stage, and error characteristic is shown. (R): only affects relative quantitation. The number next to the metric name shows its column name in the web-based application......................................................... 79

Table A-1 **Precision and recall for three dataset scenarios.** Internal scenario uses selected 4-mers. Internal and Pooled External scenario uses selected 4-mers and Maximum H3K4me3, Maximum TFBS, Maximum Sensitivity, and Average Conservation features. Internal and Matched External scenario uses selected 4-mers and Poisson
test of flanking RNA-seq counts, matched to each cell line. Grey cells indicate evaluations for which data is not available. .............................................................105

Table B-1 **Classification runtime using ClassifyR.** DE and DV classification were evaluated in two cross-validation modes, using between 1 and 16 processing cores. ..........108

Table B-2 **Parameter classes and their required variables.** Classes that store parameters for each stage of classification and the meanings of each parameter. Any other variables can be stored and used, as long as the function which performs the particular stage knows how to use them. In addition, all classes use a parameter intermediate, which is a character vector of variable names generated internally by runTest that will be used in the classification stage the class represents. ............................................................................................................................................109

Table C-1 **The identities of purchased proteins and their associated information.** Group denotes the allocated group label of each protein, so that a variety of sizes are in each group. .............................................................................................................................................113

Table C-2 **Experimental design of all seven runs for the purchased proteins.** The volumes are in units of microlitres. ........................................................................................................................................114

Table C-3 **Percentage of peptides in a particular run detected in another run.** The percentage of peptides in each run that overlap with each other run is displayed. ............................................................................................................................................116

Table C-4. **Number of proteins identified in each run.** The number before the slash is for protein identification by at least one peptide. The number after the slash is the number of proteins with at least three matched peptides. ........................................................................117
Table C-5 **Performance metrics after normalisation of Top 3 peptide summaries of proteins.** The best value for a parameter combination is highlighted as a white metric value with a black background, if the difference between the lowest and highest value is at least 25% of the highest value. **A** For Simple Scaling and Two-stage Scaling considering four representations of the yeast proteins. **B** For RUV. $k$ is the rank parameter and $\nu$ is the regularisation parameter. Number shown in a cell is the mean of metrics, where there is more than one metric calculated. …121
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td>Balanced Error Rate</td>
</tr>
<tr>
<td>CAGE</td>
<td>Capped Analysis of Gene Expression</td>
</tr>
<tr>
<td>ChIP</td>
<td>Chromatin Immunoprecipitation</td>
</tr>
<tr>
<td>CV</td>
<td>Coefficient of Variation</td>
</tr>
<tr>
<td>DNA</td>
<td>Deoxyribonucleic Acid</td>
</tr>
<tr>
<td>KDE</td>
<td>Kernel Density Estimate</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>HTS</td>
<td>High-throughput Sequencing</td>
</tr>
<tr>
<td>IQR</td>
<td>Interquartile Range</td>
</tr>
<tr>
<td>iTRAQ</td>
<td>Isobaric Tags for Relative and Absolute Quantitation</td>
</tr>
<tr>
<td>LOOCV</td>
<td>Leave-One-Out Cross-validation</td>
</tr>
<tr>
<td>MS</td>
<td>Mass Spectrometry</td>
</tr>
<tr>
<td>µL</td>
<td>Microlitre</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Components Analysis</td>
</tr>
<tr>
<td>RNA</td>
<td>Ribonucleic Acid</td>
</tr>
<tr>
<td>RUV</td>
<td>Removing Unwanted Variation</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TFBS</td>
<td>Transcription Factor Binding Site</td>
</tr>
<tr>
<td>TSS</td>
<td>Transcription Start Site</td>
</tr>
</tbody>
</table>
Introduction

Major advancements to the understanding of living systems and complex diseases enabled by new developments in biotechnology and statistical bioinformatics are beginning to transform modern life. The developments in biotechnology often provide indirect measurements of the biological entities of interest and have particular types of noise or bias; problems which are essential to consider for knowledge discovery. The associated development of new statistical methods and their evaluation in this and other contexts has the potential to deliver improved biological understanding and advance medicine.

A basic introduction to the biology and biotechnology required for this thesis follows. Firstly, the key biological molecules that are measured and their purpose is described. Next, the technologies which are used to obtain the identities and quantities of the biological molecules are introduced. Thirdly, some challenges with evaluating statistical methods for omics data processing are highlighted. Readers familiar with molecular biology and the omics measurement technologies are recommended to skip to Section 1.3.

1.1 Fundamentals of Molecular Biology

Life is the continual flow of information from molecules to other types of molecules, to accomplish tasks such as cell growth, metabolism, and defence from disease. The major information pathway in all living cells is the encoding of information from deoxyribonucleic acid (DNA) sequences to ribonucleic acid (RNA) sequences to protein sequences and was first described over fifty years ago (Crick, 1958). The processing of DNA into RNA is termed transcription and the processing of RNA into protein is
referred to as *translation*. The theory left room for information to flow in the reverse direction from RNA to DNA, as shown in Figure 1.1, which was experimentally confirmed twelve years later by two research groups (Baltimore, 1970; Temin & Mizutani, 1970).

![DNA replication](image1)

**Figure 1.1 The Central Dogma of Molecular Biology.** The DNA in a cell exists as a pair of strands of nucleotides that are bonded (vertical grey lines). The sequences of nucleotides contain instructions for making RNA molecules. Once the information is transcribed to RNA form, it is usually translated into a sequence of amino acids termed a protein, although it can be converted back into DNA. Adapted from Fu et al. (2014, p. 294).

The three categories of biological molecules are comprised of different fundamental units and have vastly different structures, which is why the technologies used to identify and measure them are largely different.

![DNA double helix](image2)

**Figure 1.2 The DNA double helix.** Each strand has a 5’ and a 3’ end, giving the two strands an antiparallel orientation. The two possible bonds are C to G and A to T. Adapted from Becker et al. (2008, p.59).

**DNA:** The genome contains much of the information necessary for an organism to develop and live. The fundamental information units of DNA are the four bases: adenine (A), guanine (G), cytosine (C) and thymine (T). Two strands of DNA are bound to each other with only bonds between A and T, and G and C being possible (Figure 1.2). Each strand has a particular modification at its ends and the 3’ or 5’ notation denotes the position of the modification on the sugar
The segments of the DNA sequence which contain instructions for making RNA sequences are called *genes*.

**RNA:** Unlike DNA, which is the same in every cell of an organism, the collection of RNA sequences present, called *transcripts*, and their abundances are different between cells and change over time. The first base of synthesised RNA is said to be the 5’ end of the molecule and the last base is the 3’ end. The first base is also referred to as the *transcription start site*, or TSS. It has a particular cap structure on it. The template strand is the strand of DNA read by the RNA polymerase. Sequences towards the 3’ end of the coding strand of the DNA are said to be *downstream* of the TSS, whereas sequences towards the 5’ end are said to be *upstream*. The region upstream of the TSS which has a regulatory function is termed the *promoter*. Segments of the newly created RNA molecule which are excised are called *introns*. The contiguous genomic regions of the sequence that remains after the introns are removed are called *exons*.

**Proteins:** The functional entities which perform most of the tasks in a cell. Each cell is capable of expressing thousands of different proteins. All the proteins in an organism are referred to as the *proteome* (James, 1997). To translate a RNA sequence into a protein, the sequence is parsed in groups of three bases, always starting at AUG and ending at one of three base combinations. Parsing always occurs from the 5’ end towards the 3’ end of the RNA molecule.

### 1.2 Overview of Measurement Technologies and Their Data Types

The three types of biological molecules considered here (DNA, RNA and proteins) are comprised of different fundamental units and have vastly different structures. Hence, a range of experimental methods have been developed to measure and characterize different aspects of this system and in turn, a variety of complex data structures have been generated. Understanding and efficient processing of such
complex and high-dimensional data sets poses a range of challenges for modern statistics, including those of normalisation and summarisation.

In this thesis, the biotechnology platforms considered are microarrays and high-throughput sequencing (Chapters 2 and 3), and mass spectrometry (Chapter 4). These platforms, and the type of data generated, are described in detail below.

1.2.1 Microarrays

Microarrays are a grid of regularly-spaced probes. Each probe has a particular sequence of nucleotides that matches to a certain genomic location; this sequence is chosen to be unique to one location in the genome. The molecules are first amplified and then labelled with a chemical that fluoresces. The abundance of each probe can be determined by the intensity of the fluorescence at a particular place in the grid. Each probe represents a genomic region that surrounds the probe, whose boundaries are determined by the lengths of the nucleic acid molecule fragments created during sample preparation. The value is a continuous measurement. Molecules which have some mismatches to the probe sequence can also bind to the probe, meaning that some probes undesirably measure a combination of different genomic locations (Koltai & Weingarten-Baror, 2008). This is termed cross-hybridisation. Another drawback is that genes which have no associated probes on the microarray, but are discovered after the manufacture of the microarray, are unable to be measured unless the experiment is repeated with a newer model of microarray. Two of the datasets used for differential distribution evaluation are microarray datasets (Chapter 3).

1.2.2 High-throughput Sequencing

In contrast to microarrays where the probes are predefined by design, high-throughput sequencing (HTS) determines the nucleotide sequence of tens of millions of DNA or RNA fragments, often simply
referred to as *reads*. Using a predefined database of regions of interest and counting the number of reads mapped within them allows estimates of abundances to be calculated. HTS has some important advantages over microarrays (Wang, Gerstein, & Snyder, 2009). Firstly, it can allow the discovery of new genomic features that are not yet found in genome databases. Secondly, microarrays have both background fluorescence and can reach saturation if the number of molecules of a certain sequence is greater than the number of probes of that sequence. For HTS, there is no background signal, and saturation is also not an issue. Lastly, as explained in the previous section, cross-hybridisation is a problem of probe-based technology, which HTS is not. One disadvantage of HTS is that it is based on simple random sampling of fragments, which means that some lower abundance molecules may not be sequenced, depending on the abundance of the most common molecules. Secondly, fragments must be associated with biological features of interest, such as genes or proteins. A summary of the two technologies is shown in Figure 1.3. HTS datasets form the basis of CAGE-seq data analysis and dataset integration (Chapter 2) and one of the datasets that is utilised for evaluation of differential distribution (Section 3.5).
1.2 Overview of Measurement Technologies and Their Data Types

Figure 1.3 Two Ways to Measure DNA and RNA. For RNA experiments, the molecules are firstly converted into DNA. Amplification of DNA is necessary to create an adequate number of measurable molecules. The molecules have to be broken into shorter fragments to allow them to be measured. Microarrays use fluorescent labelling of fragmented molecules and their binding to probes arrayed on a glass surface. HTS determines the sequence of the fragmented molecules. The count of sequences originating from a genomic region is proportional to its cellular abundance. This figure is hand-drawn.

1.2.3 Mass Spectrometry-based Proteomics

Unlike the simpler structures of DNA and RNA, proteins have more complicated 3D forms, and their identification and quantitation are more difficult. RNA and protein levels are sometimes not highly
correlated (Gry et al., 2009), because of various regulatory mechanisms that affect the rates of translation and protein degradation. Therefore, it is desirable to be able to accurately identify proteins and measure their abundances. Mass spectrometry (MS) is the most popular technique for these tasks. It is based on propelling protein fragments through a magnetic field in a vacuum, which separates fragments based on their sizes (Weickhardt, Moritz, & Grotemeyer, 1996). The size of the signal created by protein fragments hitting the detector is proportional to the number of those fragments in the sample. A more detailed background is given in Section 4.1.1.

### 1.2.4 Biological Annotation of Omics Data

Microarrays, HTS, and MS interrogate a large number of molecular sequences. By design, all microarray probes are annotated to known biological features. To give biological meaning to HTS- and MS-derived sequences, it is necessary to match the sequences to known genes or proteins. There are large databases of experimentally characterised genes and proteins for well-studied organisms, such as human or mouse. For example, RefSeq and UniProt are two such databases (NCBI Resource Coordinators, 2015). Many of the genes and proteins in these databases have known functions, which enables biologists to suggest which cellular mechanisms might be important to their topic of study. The computational matching is largely different for HTS and MS data. A more detailed background is provided in Section 2.1.5 for HTS data and Section 4.1 for MS data.

A common challenge that is shared by various bioinformatic methods for mapping high-throughput data is that they need to be tolerant of minor differences between the experimental sequences and the database sequences. These can occur because the instrument makes an error determining the molecule’s sequence or because the sample under study has a genuine difference to the one found in the database, potentially related to the condition under study. Another challenge is the level of flexibility in handling ambiguity. The short sequences may match multiple locations in the genome, or multiple proteins in the
protein database. The software implementations of these mapping algorithms provide user-specified options to whether such sequences are reported, which affects summarisation of the raw data and its subsequent statistical analysis. Such ambiguities may bias the measurements, if ignored.

1.3 Omics Data Methods Evaluation Challenges

The widespread availability of complex biological data has necessitated a corresponding advancement in statistical methods. There exists a plethora of algorithms and methods that address issues from pre-processing tasks, such as data summarisation and normalisation, to more complex analyses, such as finding enriched networks of genes in a particular condition. The issue of finding an unbiased and meaningful way to accurately evaluate existing and newly proposed methods remains challenging. Also, accuracy and precision in many settings are suboptimal and may be improved with new approaches. Below, three issues that are related to the challenges explored in this thesis are described.

Absence of a Truth Set for Performance Comparison

For a particular biotechnology, many choices of data summary and normalisation methods are available. Some methods have been adapted from other types of datasets, while others have been developed with the specific technology in mind. For example, the boundaries of signal regions derived from sequencing data are difficult to evaluate, because no database contains a comprehensive set of regions and, indeed, the regions may differ by biological condition. This has resulted in efforts where domain experts have been tasked with manually viewing a small subset of the data and creating a database of regions based on their expert judgement (Hocking, Rigaill, & Bourque, 2015). The regions selected were found to be highly consistent between researchers. Other region-finding evaluations have compared the DNA sequence within the regions to known binding patterns (Wilbanks & Facciotti, 2010). Unfortunately, these kinds of methods are only applicable to proteins where information on binding patterns is readily
available and make no assessment about the suitability of the region boundaries. Thus, there is no comprehensive method for region evaluation available; only reasonable approximations.

*Generalisability of Findings*

For binary classification problems, cross-validation is a common technique used for performance assessment. By always using different samples to train models and to make predictions with, this identifies which of the fitted models are stable. Feature selection stability and prediction error rate distribution are two types of metrics which have previously been assessed by 10-fold cross-validation for two classes of breast cancer (Cun & Fröhlich, 2012). Apart from selection frequency, the overlap of the most frequently selected features and those in known gene pathways (Mann et al., 2013), is an alternative evaluation of feature selection. However, this approach may understate the importance of genes belonging to pathways not yet characterised in public databases. Cross-validation has been shown to have low stability for datasets with weak signals (Martinez, Carroll, Müller, Sampson, & Chatterjee, 2011), which motivates the use of dataset resampling with replacement together with cross-validation (Chapter 3). This approach gives a truer impression of classifier performance, rather than a point estimate. A drawback of cross-validation is that it produces better performance metrics than training and predicting on independent datasets (Bernau et al., 2014), a process described as *cross-study validation.* In contrast, another evaluation study found a number of examples where the error rates of traditional cross-validation are equivalent or better than in cross-study validation (Schramm, Campain, Scolyer, Yang, & Mann, 2012), suggesting that cross-validation does give realistic representations of classifier performance.
Optimal Design of Experiments

Evaluation of preprocessing is simpler than the above two biologically-driven scenarios and is ideally performed with designed datasets, such as from a dilution series or spike-in design experiment. A dilution series design involves creating a complex mixture of the molecules under study, followed by manual dilution of the starting mixture to create samples with certain chosen ratios of change. The main difference between spike-in designs and dilution series is that a spike-in experiment is constructed with the same complex background mixture for every sample, but a small variety of molecules are added at particular amounts to each sample. Both types of studies allow the assessment of bias and variance. However, the absolute amounts of individual molecules cannot be controlled in dilution studies, limiting the number of possible comparisons. A spike-in study where each amount is present at every combination of two factor levels is a Latin square design (Ryan, 2007) and is utilised in Chapter 4. A Latin square design allows two sources of unwanted variation to be accounted for with the minimum number of samples.

1.4 Thesis Outline

The remainder of this thesis is organised as follows. Chapter 2 presents a novel two-stage approach for CAGE-seq region-finding and classification, followed by a case study showing its ability to find existing and unknown changes of potential medical importance in a prostate cancer dataset. Chapter 3 proposes a new summarisation approach for gene expression data with performance improvements to existing methods and introduces a publicly available software implementation, ClassifyR. Then, in Chapter 4, a set of performance metrics to characterise bias and variance are created and a range of summarisation and normalisation approaches are evaluated by a newly developed web-based application which allows users to reproduce the research findings presented and also evaluate their own methods. Finally, the key contributions of this research are summarised in Chapter 5.
Summarisation and Classification for CAGE-seq Data

The introduction of HTS technologies provides new opportunities to characterise the mechanisms of disease. A broad range of applications have been developed, such sequencing of DNA to find mutations and copy number changes, enrichment of modified DNA and its sequencing to identify epigenetic deregulation, and sequencing of RNA to determine the functional regions of the genome. Capped Analysis of Gene Expression by sequencing (CAGE-seq) shows the diversity of transcription initiation across entire genomes. The identification of new transcription start sites will enable the discovery of new genes without any prior information and additionally may allow potential mechanisms of disease and development to be proposed. Two main challenges must be addressed to permit these discoveries.

Firstly, transcripts do not precisely start in a single genomic position, but are distributed across a region. This creates the need for a fast and simple method of identifying transcription start regions. Secondly, due to biological processes other than transcription which also add 5’ caps to RNA molecules, many of the regions observed are not actually transcription initiation events. Classification of the bioinformatically identified regions is an essential filtering step in the discovery of genuine transcriptional initiation. The first independent evaluation of the only existing method designed for this task is presented, as well as another method based on integration of complementary datasets. Finally, a new method is developed that results in an improvement of classification performance.
In this chapter, a two-stage approach is described to identify the regions of true transcription initiation. Firstly, regions of enriched signal are identified by a fast and simple algorithm using a sliding window for counting read start positions and comparing to an empirical null distribution for declaring enriched regions. Secondly, a linear support vector machine classifier (Cortes & Vapnik, 1995), herein abbreviated as SVM, is utilised to distinguish between regions that represent the initiation of transcription and regions that do not. Evaluation of classification performance shows significantly improved recall and similar precision to the two existing methods. Integration of external features derived from different cell lines to those with CAGE-seq data had similar precision and recall rates to using only features derived from CAGE-seq data. The addition of matched RNA sequencing data resulted in minor gains in recall while maintaining a similar level of precision.

The remainder of this chapter is organised as follows. Section 2.1 introduces CAGE-seq data and highlights the value of its analysis. Section 2.2 describes the new region-finding algorithm and makes comparisons to F-seq and Paraclu. It also explores the performance of the proposed region classifier and two competing methods. Having established the superior performance of the proposed region classifier, Section 2.3 demonstrates its use on a biological dataset of interest to determine changes in TSS regions between normal cells and prostate cancer cells.

2.1 Background

The locations of transcription start sites (TSSs) in the genome are of biological importance. There is rarely only a single TSS for a particular transcript (Frith et al., 2008), motivating their exhaustive enumeration. Clusters of TSS positions for a single transcript are referred to as TSS regions and frequently occur in close proximity to transcription factor binding sites (TFBS). For example, the Prkd2 promoter contains a Gabp binding site (Yang et al., 2013). TFBS are known to regulate the packing of nucleosomes (Cairns, 2009), which determines the accessibility of the TSS region to the process of
transcription. When there is a loss of Gabp, Prkd2 expression is much reduced, and can lead to the development of chronic myelogenous leukaemia. Knowing the locations of the TSS regions reduces the genomic regions in which to search for regulatory motifs and generate hypotheses about the cause of changes in gene expression. Correct usage of alternative TSSs is also important for healthy development of the nervous system (Pruunsild, Kazantseva1, Aid, Palm, & Timmusk, 2007). This highlights the importance of transcription start detection to health and development.

Biological features which have been previously associated with TSS regions provide motivation for their genome-wide measurement by techniques such as ChIP-seq and DNAse-seq. Also, RNA levels measured by RNA-seq can be informative for transcription initiation. First, a brief introduction to each experimental data type is given. Next, some important features of HTS mapping algorithms are discussed. Finally, a detailed introduction to the focus of this chapter - CAGE-seq - is provided.

2.1.1 ChIP-seq

For gene transcription to be initiated or paused, certain proteins - called transcription factors - can bind to DNA and be detached from it. Also, histones, which are proteins that are attached to DNA, can be modified by other enzymes attaching or removing small molecules to them. The presence or absence of these proteins or their modifications can allow genes to be transcribed or prevent transcription from occurring (Park, 2009). For instance, a modification that is commonly known as H3K4me3 has been associated with transcription start sites undergoing transcription (Barski et al., 2007).

A ChIP-seq experiment designed for detecting a histone modification of interest is used for illustration. The experiment begins by fragmenting the chromosomes into smaller segments, followed by enrichment for the modification of interest (Figure 2.1). The complexes that have been enriched for are then
2.1 Background

separated to keep only the DNA. Finally, the DNA is sequenced, which allows the characterisation of the regulatory functions of the modification.

Figure 2.1 Basic steps of histone ChIP-seq. The black line is part of a chromosome. The grey cylinders are histones. The green circle represents a histone modification of interest. Firstly, the DNA is fragmented, and then only the histones that have the modification of interest are purified. The histones are then removed so that only the DNA remains and its sequence is determined. Adapted from Park (2009, p. 671).

2.1.2 DNAse-seq

Like histone modifications and transcription factors, unwound DNA has also been associated with TSSs that are being transcribed (Sabo et al., 2006) and can be measured using DNAse-seq. Firstly, the DNA is exposed to the DNAse I enzyme. The enzyme cuts the locations of unbound DNA sequence into small fragments, while being unable to cut sections of DNA sequence obscured by proteins (Figure 2.2). Only the 5’ end of the sequenced fragment is used to build up a picture of where the cutting of DNA is taking place.
2.1 Background

Figure 2.2 Basic steps of DNase-seq. The black line is part of a chromosome. The scissors are representative of DNAse I. The enzyme cuts in location that are not bound to protein, represented by orange and grey shapes. The resulting DNA fragments are sequenced to determine regions of unbound DNA. Adapted from Zentner and Henikoff (2014, p. 816).

2.1.3 RNA-seq

RNA-seq data can also provide supporting evidence for TSS regions, because transcription of a particular gene only occurs in one direction along a chromosome. Therefore, it is expected that there are significantly more sequenced fragments on one side of the TSS region than the other. RNA-seq typically does not directly sequence RNA molecules, but their experimentally converted DNA representations, termed cDNA.

2.1.4 CAGE-seq

CAGE-seq can answer a number of biological questions of interest. Firstly, the analysis of its data allows the identification of the locations in the genome where the transcription of RNA molecules is initiated. This enables the discovery of new genes. Secondly, when samples from different biological conditions are available, tests for differential expression can be done. Genes may also undergo TSS switching. TSS switching describes the process where the starting location of a transcribed gene changes, irrespective of any changes in the transcript’s abundance. Changing the locations of the TSS usually alters which exons are present in the transcript, which results in a different protein product being made (Boley et al., 2014). The first nucleotide of a transcribed RNA molecule has a five-prime cap (5’ cap) attached to it. RNA molecules with a 5’ cap are extracted from cells and converted into DNA for
sequencing, which is the first step of the experiment. Once the reads are aligned to a genome, in the second analysis step, only the location of the first base is considered in further analysis. The third step involves finding regions of signal enrichment. Finally, in the fourth step, these regions must be classified as TSS regions or otherwise. The entire experimental process is summarised by Figure 2.3. CAGE-seq is inapplicable to some biological investigations, such as mitochondrial genes, which lack the 5’ cap (Grohmann, Amalric, Crews, & Attardi, 1978), and bacterial genes, which have a different cap modification to animals (Cahová, Winz, Höfer, Nübel, & Jäschke, 2015).

Figure 2.3 Key steps in the workflow for analysing CAGE-seq data. The cDNA sequences obtained from the sequencer (Step 1) are aligned to the reference genome (Step 2). Only the first position of each read is retained, and the retained positions are clustered into regions (Step 3). Lastly, a classification algorithm is used to label the regions as being from transcription initiation or not (Step 4). The focus of this study is on steps 3 and 4.
CAGE-seq signal regions may be broad or narrow and they only appear on one strand, unlike ChIP-seq, so the development of specific region-finding methods to this datatype is necessary. Also, a caveat of CAGE-seq is that it detects all 5’ RNA caps, even those not related to the transcription process (Otsuka, Kedersha, & Schoenberg, 2009; Mercer et al., 2010). Despite this major problem being known for many years, research publications using CAGE-seq data continue to overlook this issue (Kratz et al., 2014; Hashimoto et al., 2015).

**Challenges for Region-finding**

Various existing methods are available for the task of finding regions. The earliest method groups reads into regions if they overlap by at least one base (Carninci et al., 2006). This is likely to join positions that are thousands of bases away for highly expressed transcripts, which causes signals from different genes to be incorrectly merged into single regions. It also lacks any statistical basis. A later approach, using the Maximal Scoring Subsequences algorithm (Frith et al., 2008), is implemented in the software package Paraclu and relies on exhaustively using all possible values of a penalty parameter for the width of a candidate region. This alters the breakpoints of regions to obtain all regions possibly supported by the data. The sheer number of results it returns, many of which overlap multiple known genes, means that it requires manual post-processing to arrive at a sensible number of regions. A minor modification of Paraclu, called RECLU, has recently been published (Ohmiya et al., 2014). It reports the widest and narrowest regions, instead of all regions, but still requires time-consuming user intervention to arrive at a final set of regions. The other difference to Paraclu is that the number of reads per million reads for each genomic position is used, to allow the method to produce comparable results between samples which have a different number of total sequences. These minor modifications mean that RECLU retains most of the drawbacks of Paraclu. One major limitation is that the algorithm does not work without replicates, which is a common scenario for CAGE-seq datasets. A third approach is based on looking for
2.1 Background

adjoining positions with CAGE reads that have constant relative expression across multiple samples (Balwierz et al., 2009). Unlike the previously mentioned methods, this approach uses rigorous statistical methods for parameter estimation and hypothesis testing. However, this method also requires sample replicates. In contrast, F-seq is an approach that doesn’t require replicates and involves fitting a kernel density estimates (KDE) around each genomic location with non-zero read count (Boyle, Guinney, Crawford, & Furey, 2008). It has been applied to numerous DNase I datasets, but never for CAGE-seq data, motivating its evaluation in the present study. The method chooses regions of signal based on a threshold found by calculation of KDEs on a randomisation of the genomic positions of the reads.

Challenges for Region Classification

The results of the aforementioned region-finding algorithms depend on read density, and do not classify regions as originating from transcription initiation or confounding biological processes, despite the importance of this. The only algorithm specifically designed to classify CAGE regions is based on modelling nucleotide k-mer frequencies surrounding the regions using an unsupervised hidden Markov model (Djebali et al., 2012), herein called the ENCODE HMM method. Two models are trained. In the first, the k-mers used in training are weighted proportionally to the number of reads in a region. In the second model, all k-mers are weighted equally. The posterior probability of each cluster fitting to the main model is calculated using Bayes’ rule and regions are classified using a threshold on the probability. In other words, the algorithm biases towards learning the features of CAGE regions with high read counts, and against regions derived from lowly expressed genes. No validation of results from the classifier was performed in the original article. The authors also did not consider integrating external data in their model, which could potentially improve the classifier’s performance. Segway is another method that classifies regions into categories, such as TSS, Enhancer, and Gene End, but does not take CAGE-seq data as input (Hoffman et al., 2012). It is based on dynamic Bayesian networks. The simplest
approach that avoids region classification (and region calling) altogether is to make small counting windows around annotated TSS (Plessy et al., 2012), before performing an analysis of the amount of signal, such as differential expression. The drawback is that novel transcription starts, and even novel genes, are ignored.

In summary, the CAGE-seq algorithms proposed previously do not sufficiently model features of the dataset to enable accurate region determination and their classification. The region-finding algorithms either overestimate the size of regions by not properly taking into account the noise distribution or provide millions of mostly overlapping and redundant regions. Also, the region classification algorithms have not been independently evaluated, which may identify possibilities for improvement.

2.1.5 HTS Mapping

Millions of short sequences obtained from the sequencing technologies introduced above must be accurately mapped to a large reference genome in a reasonable amount of time. The large number of reads generated and the size of genomes has motivated the development of fast and accurate algorithms for short sequence mapping. Unlike the classic Needleman-Wunsch algorithm for aligning pairs of sequences, the algorithms developed for HTS data use heuristics to find good matches, but don’t guarantee finding the best match of a sequence to a reference. The earliest of these is Bowtie (Langmead, Trapnell, Pop, & Salzberg, 2009), which uses a Burrows-Wheeler transform and a Ferragina-Manzini index to align about 25 million sequences per hour. Bowtie is sufficient for mapping CAGE-seq, ChIP-seq, and DNase-seq data. Most RNA-seq data has special requirements for its accurate mapping. RNA-seq mapping software needs to be able to map sequences across large gaps, because of introns. Specialised applications for mapping RNA sequences to the genome include TopHat (Trapnell, Pachter, & Salzberg, 2009) and STAR (Dobin et al., 2013), which use similarly efficient data structures and heuristics to Bowtie to map millions of reads in a short amount of time.
2.2 Two-stage Preprocessing Algorithm

To enable biologically valid insights to be derived from CAGE-seq data, a two-stage approach is proposed. In the first stage, a sliding window is applied across the genome and regions are identified based on comparison to a large number of randomly chosen regions. In the second stage, a linear SVM is applied to predict if the found regions are caused by transcription initiation or biological artefacts.

2.2.1 Region-Finding

In the first stage, a sliding window approach is used to detect regions of significant signal enrichment using a cut-off based on randomly sampled windows. The algorithm starts with background estimation, followed by window joining and region trimming. Assuming that genuine TSS signals are rare in the genome, 1000 candidate windows of width $w$ are generated at random locations in the genome. The CAGE reads within each window are counted and the 95th percentile of the counts is taken as the cut-off value, below which a window is deemed to have insufficient signal to form a region. Next, a candidate window of width $w$ is moved along each strand of each chromosome in increments of $w/2$. Here, $w = 50$ is used, which is typical of regions found previously (Carninci et al., 2006). For each candidate window, the count of CAGE read starts is made. If the count is above the background cut-off value, the window is added to a list of regional windows. The ends of regional windows are trimmed for outermost, contiguous positions that contain counts less than the cut-off value, divided by the window width. Finally, any adjacent regional windows separated by less than 30 base pairs are merged into a single region.

2.2.1.1 Comparison of Regions

Evaluation is based on publicly available CAGE-seq data of six cell lines (GM12878, H1-hESC, K562, HeLa-S3, HepG2, and HUVEC – the CAGE cell lines). Mapped BAM files were downloaded from the ENCODE data repository (ENCOD Project at UCSC, n.d.) on the UCSC Genome Browser website.
Preprocessing details are found elsewhere (Djebali et al., 2012). The unique Submission IDs are 3946, 2380, 2359, 2363, 2381, and 2376 and the number of mapped reads are 19677397, 24604761, 24319886, 24394908, 24604043, and 18717719, respectively.

All three algorithms discover tens or hundreds of thousands of regions in each sample (Table 2). The proposed method finds the least number of regions for each cell line. The greatest number of regions are found for the H1-hESC cell line by the proposed method, but HepG2 for the other two methods. H1-hESC is the only stem cell sample in the study; the other five are differentiated cells. F-seq finds over one million regions for GM12878 and HepG2 cells.

Table 2.1 **Number of regions detected for each cell line.** The proposed sliding window, Paraclu and F-seq are evaluated.

<table>
<thead>
<tr>
<th></th>
<th>Region-finding Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cell Line</td>
<td>Proposed</td>
</tr>
<tr>
<td></td>
<td>GM12878</td>
<td>83729</td>
</tr>
<tr>
<td></td>
<td>H1-ESC</td>
<td>266462</td>
</tr>
<tr>
<td></td>
<td>HeLa-S3</td>
<td>78719</td>
</tr>
<tr>
<td></td>
<td>HepG2</td>
<td>117421</td>
</tr>
<tr>
<td></td>
<td>HUVEC</td>
<td>76107</td>
</tr>
<tr>
<td></td>
<td>K562</td>
<td>68692</td>
</tr>
</tbody>
</table>

Tabulation of the overlaps of regions between the proposed method with the regions found by Paraclu and F-seq demonstrates that there is a lot of similarity between the regions found by the methods evaluated (Table 2.2). Indeed, every region found by the proposed method is also found by Paraclu. However, a visual exploration of the regions shows that Paraclu and F-seq are detecting many other regions which have minimum support – as little as one CAGE-seq read (Figure A-1) – or they span large regions of the genome, which contain multiple known genes.
2.2 Two-stage Preprocessing Algorithm

Table 2.2 Percentage overlap between proposed method’s regions and those found by Paraclu and F-seq. Regions must have at least one base overlap and be on the same genomic strand to be considered as overlapping.

<table>
<thead>
<tr>
<th>Cell Line</th>
<th>Region-finding Method</th>
<th>Paraclu</th>
<th>F-seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM12878</td>
<td></td>
<td>99.7%</td>
<td>100%</td>
</tr>
<tr>
<td>H1-ESC</td>
<td></td>
<td>99.8%</td>
<td>100%</td>
</tr>
<tr>
<td>HeLa-S3</td>
<td></td>
<td>99.7%</td>
<td>100%</td>
</tr>
<tr>
<td>HepG2</td>
<td></td>
<td>99.8%</td>
<td>100%</td>
</tr>
<tr>
<td>HUVEC</td>
<td></td>
<td>99.7%</td>
<td>100%</td>
</tr>
<tr>
<td>K562</td>
<td></td>
<td>99.7%</td>
<td>100%</td>
</tr>
</tbody>
</table>

2.2.2 Classification of Regions

A linear support vector machine (SVM) classifier is developed to classify regions found by the proposed method as TSS or otherwise with high precision and recall. Two kinds of features are calculated for each region; internal features and external features. Internal features are those that may be computed from the CAGE-seq data directly or the genome it is mapped to. External features are derived from associated datasets, such as ChIP-seq or RNA-seq. External features may be further divided into matched and pooled features. Matched features are those that are calculated on the same sample the CAGE-seq is performed on, whereas pooled features are those calculated from an aggregation of other samples distinct from the sample under consideration. Many features are computed in windows that are a certain distance upstream and downstream of the region’s summit. Definitions of these terms are provided in Section 1.1.

2.2.2.1 Features and Classes

Three internal features are considered:

Kurtosis: Pearson’s kurtosis of the CAGE read histogram of an identified region, based on the fourth standardised moment, is calculated. This feature is analysed to examine if any differences in shape would be discriminatory.
2.2 Two-stage Preprocessing Algorithm

**Read density:** The number of CAGE reads inside the boundaries of an identified region, divided by the width of the region. This feature is a combination of the shape and height of a region.

**4-mer counts:** Patterns of DNA bases surrounding the TSSs are also known to be different to other regions in the genome (Sonnenburg, Zien, & Rätsch, 2006). A 500 base pair window was created upstream and another downstream of the summit of each CAGE region. Frequencies of all 4-mers were calculated independently for the two windows. In the upstream window, there are $4^4 = 256$ distinct 4-mers, and similarly downstream, making a total of 512 4-mer features.

Furthermore, five external features are considered. All datasets were downloaded from the ENCODE Project Repository (ENCODE Project at UCSC, n.d.).

**Mammalian conservation:** Considered for its known association with regulatory regions, such as promoters, scores within the regions are used. For each region, the conservation values of each base are averaged. A small fraction of regions may not overlap with any bases with conservation scores, because the genomic sequence is not able to be multiply aligned to the other genomes. For these regions, an imputed value, equal to the minimum value of regions that had available conservation scores, is used.

**TFBS:** For each region, the maximum score in a window extending 100 base pairs from the region boundaries is assigned to the region. Rather than exclude cell type-specific signals, the measured maximum is used. Pooled measurements of transcription factor binding from 95 cell types of an unspecified number of transcription factors stored in the table wgEncodeRegTfbsClusteredV2.

**DNAseq I hypersensitivity:** This feature is considered because TSSs typically occur in open chromatin. Similar to TFBS, the maximum count within 100 base pairs from region boundaries is determined. Pooled DNAseq I hypersensitivity data using 74 cell lines was obtained from the table named wgEncodeRegDnaseClustered.
2.2 Two-stage Preprocessing Algorithm

**H3K4me3**: This histone modification is known to be found on the nucleosomes surrounding active TSSs. Again, the maximum score within 100 base pairs of the region boundaries is calculated. Seven files were downloaded with Submission IDs of 2806, 2815, 2846, 2878, 2890, 2909, and 2921.

**RNA-seq difference**: The number of RNA-seq reads on either side of the region is counted. One count is a 100 base wide flanking window immediately upstream of the 5’ edge of the region. The other is the same size, but downstream of the 3’ edge of the region. The feature calculated is $P(Y \leq y)$ of the Poisson distribution where $\lambda$ is equal to the downstream flank count and $y$ is the count in the upstream flank. Unmapped, total RNA-seq data for two of the six CAGE cell lines (GM12878 and K562) was downloaded. Total RNA-seq data is not available for the other four cell lines. The unique Submission IDs are 1502 and 1503. Raw reads were mapped to the human genome assembly hg19 with STAR (Dobin et al., 2013). Only uniquely mapping reads and no more than 3 mismatches to the reference sequence were allowed. 40 bases from the ends of each pair of reads were ignored, as these correspond to stretches of low-quality sequencing data. No splice junctions spanning more than 100000 bases were allowed.

All features are standardised to be between 0 and 1 by dividing by the maximum score for all regions, per feature type and per cell line. A concise overview of all features is provided by Table 2.3.
Table 2.3 **Features considered for TSS classification.** For each feature, the summarisation procedure, location of data points summarised, and the feature categorisation are shown.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summarisation</th>
<th>Location</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kurtosis</td>
<td>Directly used</td>
<td>Region</td>
<td>Internal</td>
</tr>
<tr>
<td>Read Density</td>
<td>Directly used</td>
<td>Region</td>
<td>Internal</td>
</tr>
<tr>
<td>4-mers Counts</td>
<td>Count</td>
<td>500 bases upstream and downstream of region summit</td>
<td>Internal</td>
</tr>
<tr>
<td>TFBS</td>
<td>Maximum</td>
<td>Region and 100 base extension of boundaries</td>
<td>External Pooled</td>
</tr>
<tr>
<td>DNAse I Hypersensitivity</td>
<td>Maximum</td>
<td>Region and 100 base extension of boundaries</td>
<td>External Pooled</td>
</tr>
<tr>
<td>H3K4me3 Hypersensitivity</td>
<td>Maximum</td>
<td>Region and 100 base extension of boundaries</td>
<td>External Pooled</td>
</tr>
<tr>
<td>Mammalian Conservation</td>
<td>Average</td>
<td>Region</td>
<td>External Pooled</td>
</tr>
<tr>
<td>RNA-seq Difference</td>
<td>Distribution function probability</td>
<td>100 bases flanks adjacent to region boundaries</td>
<td>External Matched</td>
</tr>
</tbody>
</table>

The class labels of regions determined by ENCODE HMM were also obtained from the repository, with Submission IDs 5610 and 5147. Five of the cell lines erroneously have the same submission identifier in the database, although manual inspection confirmed that all files do contain sample-specific results.

Segway’s labels were downloaded from the supplementary materials website accompanying ENCODE’s analysis of the same six cell lines (Hoffman et al., 2013).

### 2.2.2.2 Classifier Construction

Unlike typical classification datasets, where the true class membership is known in advance, TSS datasets require the assignment of inferred class labels to regions. Class labelling of regions was made by the same method used for Segway’s evaluation (Hoffman et al., 2012); Segway is, to date, the most comprehensive study of TSS region determination. It involves the use of dynamic Bayesian networks to create a segmentation of the genome, followed by manual assignment of biological categories to the segments by a domain expert. Briefly, 500 bases wide windows were made upstream and downstream of
the start position of each GENCODE transcript. If a detected region overlapped with any of the windows, it was labelled as a TSS region. Otherwise, it was assigned to the non-TSS class.

PCA (Mardia, Kent, & Bibby, 1979) on the labelled TSS regions was used to select features for the classifier. Firstly, the six scalar features of kurtosis, read density, TFBS, DNase I hypersensitivity, H3K4me3 hypersensitivity, and mammalian conservation were used (Figure A-2). The first dimension had visible separation between the two classes. Features with an absolute correlation to PC1 of more than 0.5 were used in the model. The features selected were: average mammalian sequence conservation ($\rho = -0.85$), DNase I hypersensitivity maximum ($\rho = -0.74$), TFBS maximum ($\rho = -0.90$), and H3K4me3 maximum ($\rho = -0.88$). In addition to the scalar features, the vector feature of 4-mer frequency was also evaluated. PCA was used on standardised counts of the 512 distinct 4-mers (Figure A-3). The first principal component had noticeable separation of the two classes. There were 148 4-mers that had $|\rho| \geq 0.5$ with PC1, and used subsequently in classification.

Classifier training is performed using the selected features with a L2-regularised L2-loss linear SVM in a leave-one-out cross-validation (LOOCV) scheme, using primal solving. In the primal form, there are as many variables (parameters which define the decision boundary of the classifier) to optimise as there are features, which is an efficient model training approach in this scenario. To handle imbalanced class sizes, error weights are provided for each class. Not adjusting for differences in set sizes of the smaller TSS and larger non-TSS set would result in high accuracy for the non-TSS set and low accuracy for the TSS set, since the default parameterisation of SVMs is to maximise the overall number of correct predictions (Akbani, Kwek, & Japkowicz, 2004). For the TSS class, the weight is calculated as the number of peaks in the non-TSS class divided by the number of peaks belonging to the TSS class. For the non-TSS class, the weight was 1.
2.2 Two-stage Preprocessing Algorithm

2.2.2.3 Classification Performance Evaluation

Classification performance is evaluated on the metrics of precision and recall. Precision is the fraction of TSS-classified regions that are labelled as TSS regions. Recall is the fraction of labelled TSS regions that are classified as TSS regions by the linear SVM.

A broad range of cost values was evaluated, to understand classification performance at different hardness levels of the SVM margin (Figure A-4). The classification procedure was reasonably robust to the cost value used and a cost of 0.1 is used in the following evaluations.

Three classification scenarios are considered:

**Internal features only:** This involves using only the selected 4-mers.

**Internal and pooled external features:** The selected 4-mers and the four previously-selected features derived from the public datasets are used.

**Internal and matched external features:** The selected 4-mers and RNA-seq counts are considered.

Considering classification with internal features, the proposed method has high precision and recall for all cross-validations (Table 2.3). The performance metrics obtained using the proposed SVM classifier are compared to those obtained by two other algorithms; Segway and ENCODE HMM. Details of Segway and ENCODE HMM algorithms are provided in Section 2.1.4. ENCODE HMM has the highest recall for all samples, however this is always accompanied by low precision. The method is biased to producing false positive predictions. Segway has a good balance between precision and recall, although the recall is, on average, 0.19 less than the proposed method’s recall. The addition of pooled external features has no systematic benefit to precision or recall (Table A-1). The use of the matched external feature derived from RNA-seq data shows a minor gain in recall and also an increase in precision of 0.13 for the H1-ESC sample.
Table 2.4 Precision and recall values for the internal features LOOCV classification. For each cell line, classifier training was performed on the five other cell lines. A cost parameter of 0.1 is used for the linear SVM classifier.

<table>
<thead>
<tr>
<th>Cell Line</th>
<th>Segway Precision</th>
<th>Segway Recall</th>
<th>ENCODE HMM Precision</th>
<th>ENCODE HMM Recall</th>
<th>Proposed Method Precision</th>
<th>Proposed Method Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM12878</td>
<td>0.70</td>
<td>0.64</td>
<td>0.25</td>
<td>0.92</td>
<td>0.77</td>
<td>0.81</td>
</tr>
<tr>
<td>H1-ESC</td>
<td>0.59</td>
<td>0.71</td>
<td>0.27</td>
<td>0.89</td>
<td>0.61</td>
<td>0.81</td>
</tr>
<tr>
<td>HeLa-S3</td>
<td>0.79</td>
<td>0.66</td>
<td>0.32</td>
<td>0.91</td>
<td>0.76</td>
<td>0.87</td>
</tr>
<tr>
<td>HepG2</td>
<td>0.59</td>
<td>0.59</td>
<td>0.23</td>
<td>0.93</td>
<td>0.69</td>
<td>0.79</td>
</tr>
<tr>
<td>HUVEC</td>
<td>0.82</td>
<td>0.67</td>
<td>0.26</td>
<td>0.94</td>
<td>0.81</td>
<td>0.85</td>
</tr>
<tr>
<td>K562</td>
<td>0.77</td>
<td>0.62</td>
<td>0.27</td>
<td>0.93</td>
<td>0.71</td>
<td>0.88</td>
</tr>
</tbody>
</table>

2.3 Case Study: Prostate Cancer

The proposed two-stage method is applied to a dataset consisting of one healthy prostate cell line (PrEC) and one cancerous prostate cell line (LNCaP) sample (Bert et al., 2013). A number of TSS regions with high-ranking changes in read counts for genes never previously associated with prostate cancer are found. Using version 17 of the GENCODE human gene database (Harrow et al., 2012), it is possible to annotate TSS regions to known genes and evaluate how well-known in the prostate cancer research field the differentially expressed genes are using the GeneValorization text mining tool (Brancotte et al., 2011).

Regions have been called using the algorithm described in Section 2.2.1 for each sample. 42445 regions were found for the PrEC sample and 31707 for the LNCaP sample. 29927 regions were classified as TSS regions for PrEC and 22585 for LNCaP. Regions which had any overlap between the two conditions were merged by calculating the union of regions’ genomic coordinates. This resulted in 38923 distinct regions used for further evaluation. A linear SVM using only the selected 4-mers’ counts and trained on the six ENCODE cell lines was applied to the prostate cancer dataset to predict each region’s class. GFOLD (Feng et al., 2012) was used to produce a ranking of shrunken (generalised) fold
changes, because no statistical analysis of differential expression is possible in this unreplicated dataset. The shrinkage applies more strongly to genes with low read counts, preventing the rankings from being biased towards genes which have a relatively small numerator and denominator in their fold change, which often result in large, but unstable, fold changes. The 100 regions which have the largest absolute GFOGLs were overlapped with known genes and 98 have a match to a gene. The gene with the best associated GFOGL statistic has clear differences between conditions in a plot of the count data (Figure A-5A). 88 distinct genes were text mined with GeneValorization for the term “prostate cancer” and a large spread of journal publications per gene was found. The gene EREG has been associated with prostate cancer in 20134 publications. The median number of publications per gene is eight. Seventeen of the 98 genes have never been associated with prostate cancer before.

Differential TSS usage evaluation finds even more changes in prostate cancer of biological interest. Genes which have at least two associated TSS regions are considered. There are 3119 such genes. A Fisher’s exact test is not used because of the lengthy computation time required. Instead, a TSS switching statistic for a gene is defined as the difference between the largest and smallest fold changes, on the log₂ scale, multiplied by the minimum of the average counts of the minimum or maximum fold change regions. Visual representation of the top ranked gene, DUT, illustrates that the proposed TSS switching method is choosing appropriate genes (Figure A-5B). GeneValorization text mining analysis of the top 100 genes shows that only a few of these genes have been previously associated with prostate cancer. The median value is 0 publications, while the maximum value, for gene ATPIF1, is 1451 publications.
2.4 Discussion and Conclusion

A novel two-stage approach comprised of a fast and simple region-finding method, followed by a linear SVM classifier has been developed to detect thousands of regions per sample and categorise whether they are TSS regions or not. This study is also the first time that F-seq has been evaluated with CAGE-seq data. Three dataset scenarios were assessed to determine if data integration can provide an improvement in precision and recall. Finally, the new approach was used on a prostate cancer dataset. Many previously known gene-disease associations were rediscovered, as well as a number of novel genes that warrant further biological investigation.

Although F-seq reliably found the boundaries of regions, it produced an excessively long list for the dataset analysed. Closer inspection of the results found that many genomic locations with only one CAGE read mapping to them were identified as regions. A possible reason for this is that F-seq takes the reads and randomises their position along the genome to find a cut-off value. In contrast, the proposed method keeps the locations of the original data values, but generates random windows and counts the number of reads within them. The null model used by the proposed method may be more appropriate for CAGE-seq data than the model used by F-seq.

It may initially be surprising that neither kurtosis nor read density were strongly associated with the first principal component of the PCA plot in Figure A-2, which was associated with region class. However, previous research on CAGE-seq region properties provides support for this disparity. CAGE regions have been able to be categorised into three different shapes; broad, sharp, and multi-modal (Carninci et al., 2006). All of these shape categories have many regions which belong to the promoter regions of known genes, providing support for the idea that TSS regions have a broad range of read densities and kurtoses. The selection of 4-mer features for their association with region classes also agrees with previous research work (Sonnenburg, Zien, and Rätsch 2006).
2.4 Discussion and Conclusion

Evaluation of integrating pooled external features showed there was no overall benefit to incorporating freely available datasets with CAGE-seq data. When an orthogonal dataset on the same sample was available, such as RNA-seq, there were minor gains in recall, while precision improved for one of the cell lines. In a practical setting, this suggests that integrating RNA-seq data should be done if it is already available, but it is not worth the effort of generating RNA-seq data for the sole purpose of integrating it with CAGE-seq data. No firm conclusions can be drawn about the effect of integrating matched RNA-seq data, however, because only two out of six samples have such data available.

A limitation is the lack of a truth set independent of any experimental data used by this study. It was necessary to generate the truth set by using the CAGE reads and the GENCODE gene annotation, as the authors of Segway did in their study. This potentially introduces an unwanted biases into the evaluation procedure. One such possible bias is for biologically real TSS regions missed by the CAGE technique being labelled as not being TSS regions in the truth set. This would lead to false negatives being counted as true negatives. Also, the GENCODE gene annotation, like any biological database, is sure to be incomplete. The cells being studied in the experiment may have real TSS regions which are not stored in the database. This leads to true positives incorrectly being counted as false positives. Lastly, the accuracy of the TSS positions in the GENCODE database is impossible to quantify. The database does not provide experimental evidence or a confidence score for its entries. When evaluating precision and recall, it is necessary to assume that most of the TSS positions are correct. The future development of a robust and independent truth set is important for the accuracy of any CAGE-seq evaluation study.

Another limitation is the lack of biological replicates available for each cell type, ruling out any formal hypothesis testing. It is desirable to evaluate whether the regions and their classifications are stable in a scenario where multiple replicates of each biological condition are available.
In summary, an effective two-stage algorithm comprising of a sliding window region-finding method and linear SVM classifier have been developed to accurately identify and classify CAGE-seq read regions. The algorithm has been compared to existing published methods and both the region-finding and classification stages demonstrated superior performance to existing methods.
Differential Distribution for Binary Classification Problems

In the last decade, technologies such as microarrays and high-throughput sequencing have enabled the simultaneous measurement of expression for tens of thousands of genes, which allows the hypothesis-free discovery of novel associations with disease, also known as biomarkers. Biomarkers routinely become the basis of patient care and disease classification. To date, a central theme in disease classification using omics data is using the change in average expression, often referred to as differential expression (DE), as the main measure of differences between classes of samples. For the task of survival class prediction, the error rate remains undesirably high and the stability of selected features for the classifier has not been characterised (Jayawardana et al., 2015). Recent studies have shown that when transcription factors or epigenetic signals become deregulated, a change in expression variability (DV) of target genes is frequently observed (Ho, Stefani, dos Remedios, & Charleston, 2008; Hulse & Cai, 2013). However, assessing the importance of genes by either differential expression or variability alone potentially misses sets of important biomarkers. Integrating both types of changes could lead to improved prediction and prognosis for patients.

A new approach is described here for assessing the importance of genes based on differential distribution (DD), which combines information from differential expression and differential variability into a unified metric. A comprehensive simulation study is carried out, to pick a representative method for DV and DD. The proposed statistic based on the differences of medians and absolute deviations is
found to choose a broad variety of features and a naïve Bayes kernel voting method is shown to have low classification error. Based on findings in the simulation study, a direct comparison between representative feature selection methods for the three types of distributional change is carried out on three cancer datasets. Feature ranking and selection stability based on differential distribution is shown to perform two to three times better than DE or DV alone, and also yield equivalent error rates to DE and DV. Finally, assessing genes via differential distribution produces a complementary set of selected genes to DE and DV, potentially opening up new categories of biomarkers. A novel classification execution and evaluation framework, ClassifyR, is implemented in R and used to carry out all evaluations in a fast and reproducible way.

The rest of this chapter is structured as follows. In Section 3.1, overviews of existing classifiers for omics data based on either DE or DV data features, and corresponding classification frameworks in R are provided. Section 3.2 outlines the procedures for feature selection, training, and prediction for the three change types evaluated. Section 3.3 defines the metrics used to compare competing methods. Section 3.4 provides a comparison of two varieties of DV selection and four varieties of DD selection using two novel and two existing varieties of the naïve Bayes classifier. The best-performing selection method for each change type is used in an evaluation of classifier performance on three biological datasets in Section 3.5. Section 3.6 introduces ClassifyR, a software which has been developed for parallelised classification and its comprehensive evaluation. Finally, the benefits of DD classification and opportunities for its improvement are summarised and discussed in Section 3.7.

3.1 Background

Accurate prediction of patient survival has the opportunity to optimise patient treatment. Those patients who have genetic features associated with lengthy survival can avoid treatment and its associated problems. Avoiding unnecessary treatment reduces costs to the healthcare system and also allows
clinicians to focus their efforts on patients who need immediate intervention. One of the earliest studies built a classifier for different subtypes of leukaemia with different severities, based on finding a set of genes that are uniformly high in one class and low in the other (Golub et al., 1999). Since then, a wide range of studies have been used to determine important biomarkers (features), with applications such as predicting survival outcomes (Eschrich et al., 2005; Jayawardana et al., 2015), disease subtypes (Mills et al., 2009; Marisa et al., 2013), drug sensitivity (Gunther, Stone, Gerwien, Bento, & Heyes, 2003; L. Li et al., 2014), and even behavioural characteristics (Takahashi et al., 2010). Apart from RNA expression, many studies have utilised other kinds of biological platforms, such as proteins (Li et al., 2013) and metabolites (Kim, Koo, Jung, Chung, & Lee, 2010). Reviews of classification methods based on differences in average expression levels can be found elsewhere (Ferté et al., 2013; Lin & Chen, 2013; Domany, 2014).

Recently, variances of expression (differential variability) have been found to differ in numerous gene expression datasets (Ho et al., 2008; Hulse & Cai, 2013). One biological interpretation for this is that increased variability of the RNA level of a particular gene, caused by the loss of precise regulation of its expression, may follow disruption of transcription factors or epigenetic signals by pathogenic processes, leading to greater variation of the expression level between samples within the affected class (Haraksingh & Snyder, 2013). The study by Ho and his colleagues found that highly variable genes are highly co-expressed with many more genes than are genes with lower variability in their expression, suggesting an increase of regulation in networks of genes. Genes with higher variability are typically associated with the disease state, although high variability per se may be evolutionarily conserved between species and serve a potentially beneficial purpose in some gene systems (Mostafavi et al., 2014).
Motivated by these biological insights, two main statistical approaches have been developed to assess the association of genes with disease phenotypes, such as prognosis. In the earliest proposal, mixture models, which can also measure differential expression, were developed (Bar, Booth, & Wells, 2014). Another method, diffVar (Phipson & Oshlack, 2014), based on testing absolute deviations from class means in a linear modelling framework, is limited to discovering changes in variability. Despite these methods showing the ability to find genes with differential variability between conditions, neither study developed a corresponding metric for feature selection or investigated its potential as a biomarker in a classification setting.

The potential of differential variability to aid in classification was recently demonstrated for the first time in a comprehensive study of DNA methylation in a number of cervical cancer datasets (Teschendorff & Widschwendter, 2012). A differential variability classifier based on adaptive index models (Tian & Tibshirani, 2011) outperformed a differential methylation classifier for predicting early-stage cancer, although there was no difference in classification performance of later stages. This suggests that traditional DE classifiers disregard important differences which are present in biological datasets.

Differential variability attempts to use the characteristics of deregulated networks to provide a new approach to assessing the importance of genes. However, it omits useful information from changes in locations between classes. To address this, a novel metric is proposed, based on identifying genes with differential distribution to enable simultaneously identifying genes that are differentially expressed, differentially variable, or both. As such, DD aims to avoid the need for ad-hoc DE and DV classifier aggregation algorithms, such as taking the intersection of selected features, by detecting features with both kinds of changes using the same metric and requiring only one classification algorithm, rather than two. Biologically, a change in distribution, such as from unimodality to multimodality, suggests that a
gene has an expression range which must be maintained for healthy cellular function. Increases in variability can be similarly interpreted. Furthermore, we extend such feature selection criteria into a classification setting.

To date, no research literature has examined DD classification for biological problems. In other fields, such as engineering, DD classification by a multidimensional kernel density estimate predictor has been shown to perform slightly better than methods like LDA (Ghosh, Chaudhuri, & Sengupta, 2006) on a simulated dataset and low-dimensional datasets from physics and chemistry, motivating its exploration in high-dimensional omics datasets. This demonstrates the need to develop and evaluate an alternative set of DD metrics as a type of discrimination measure for identifying candidate genes of interest as well as using those metrics in a novel classification scheme applied to high-dimensional datasets. DV and DD are, for the first time, characterised in terms of model stability, something which is known to be lacking for DE feature selection (Cun & Fröhlich, 2012). Additionally, the performance of all three classification schemes is systematically examined based on their prognostic error rates and biological relevance.

Existing Classification Software

Although a large collection of disparate classification software packages exist in R, there are no available frameworks for linking existing or custom classification functions with a comprehensive set of classification performance measures. An effective classification package should exhibit the characteristics of fast computational time, reproducible results (e.g. random number seed, storing algorithm parameters used), flexibility to incorporate user-defined classification functions and post-processing options for performance assessment. There are some R packages which provide only a subset of these desirable features. For example, SPRINT (Mitchell et al., 2014) provides a new framework for parallelizing classification algorithms in R, but requires new classification functions to be coded in C or
3.2 Feature Selection and Classification

Fortran. This is challenging for users who are only proficient in R. Many popular R packages, such as e1071 (Meyer et al., 2014) and PamR (Hastie, Tibshirani, Narasimhan, & Chu, 2013), provide a small number of classifiers but are not extensible. caret is a popular framework which provides parallelization and customisable classifiers, but lacks post-processing evaluation functions (Kuhn, 2008). Similar packages, such as MLInterfaces, CMA and MCRestimate are also available from Bioconductor, but are also not comprehensive classification solutions. A new framework to integrate classification and performance evaluation is required.

3.2 Feature Selection and Classification

The particular feature selection methods and classifier utilised are distinct for each type of change.

DE: The process differs for microarrays and RNA-seq datasets. (a) For microarray data, genes are ranked on their moderated t-statistics using the implementation in limma (Smyth, 2004). Training and prediction for the microarray datasets is performed using diagonal linear discriminant analysis (DLDA), originally developed for classification of cancer samples based on their gene expression measurements (Dudoit, Fridlyand, & Speed, 2002). (b) For RNA-seq data, genes are ranked based on a likelihood ratio test statistic of negative binomial generalised linear models using the implementation in edgeR (Robinson, McCarthy, & Smyth, 2010). Poisson linear discriminant analysis (PLDA) is used to determine a decision boundary and make predictions, as it has been demonstrated that DLDA finds suboptimal decision boundaries for count data, whereas PLDA finds the correct boundary (Witten, 2011). A power transformation was applied to eliminate overdispersion, making PLDA applicable to RNA-seq count data.

DV: For microarray data, the normalised values were directly used. For RNA-seq data, the mean-variance trend was removed by using the regularised logarithm transformation of DESeq2 (Love, Huber,
& Anders, 2014), to avoid detecting DV features simply caused by DE. Features were then ranked based on either their Bartlett or Levene statistic and selection was applied. The Bartlett test (Bartlett, 1937) tends to choose features with a small number of outliers, whereas the Levene statistic (Levene, 1960) is robust to outliers. Before training and prediction, feature values were calculated as the absolute value of the difference of each measurement with the median of all samples in the training set and taking the absolute value. Thus, if the values originally came from a normal distribution the transformed values will follow a half-normal distribution. Fisher’s linear discriminant analysis (FLDA) was used for classification.

DD: Four approaches for assessing the differences between two different distributions (class 1 and class 2) are considered. These are the differences of medians and deviations, the Kolmogorov-Smirnov distance, the log-likelihood ratio, and simply combining the results of individual DE and DV selections. Motivated by the success of finding DE genes by considering the absolute differences in medians for the melanoma dataset (Jayawardana et al., 2015), the Differences of Medians and Deviations (DMD) is defined as:

$$\text{DMD} = |\text{median}_1 - \text{median}_2| + |Q_{n_1} - Q_{n_2}|$$

where $\text{median}_1$ and $\text{median}_2$ represent the median expression values of class 1 and 2, respectively. The values $Q_{n_1}$ and $Q_{n_2}$ represent the robust scale estimator (Rousseeuw & Croux, 1993) for class 1 and 2, respectively. The Kolmogorov-Smirnov (KS) distance is simply defined as the greatest distance between the empirical cumulative distribution functions of the two classes. Thirdly, a log-likelihood ratio statistic with robust estimates of the location and scale was used:
3.2 Feature Selection and Classification

\[ LLR = -2 \left( \sum_{i=1}^{s_1} \log f(x_i | \mu = \text{median}, \sigma^2 = Q_{i}^2) \right) - \left( \sum_{i=1}^{s_2} \log f(x_i | \mu = \text{median}_2, \sigma^2 = Q_{i_2}^2) \right) \]

where subscript \( i \) denotes membership of class \( i \), \( s_i \) denotes the number of samples in class \( i \) and \( f \) is the probability density function of the normal distribution. The LLR is the log of the likelihood ratio statistic for testing whether the two classes come from the same normal distribution or two different normal distributions where robust estimators of the mean and standard deviation are used in place of the maximum likelihood estimators. Terms without subscripts use values from samples in both classes.

Finally, ensemble feature selection was performed by combining the selections which use the moderated t-test from \textit{limma} and Bartlett test by taking the union of selected features. This is a naïve way to jointly capture features that are changing means and also those that are changing variances.

For each of the chosen features, a kernel density estimate is built for each of the two classes using a Gaussian smoothing kernel and bandwidth calculated by Silverman’s rule, which are the default settings of the \texttt{density} function in R. For the RNA-seq dataset, counts were transformed by the regularised log method, to prevent feature selection being biased towards differentially expressed genes, because of overdispersion of count data. To predict a sample from the test set, a naïve Bayes classifier is used for each feature. Two types of the classifier are considered. Firstly, each feature votes once for the class that has the maximum \textit{a posteriori} estimate and the class with the largest number of votes is the predicted class. This is referred to as \textit{unweighted voting}. Also, the differences between class densities, the distances from the observation to the nearest non-zero crossover point of the two densities, and the sums of those two weights are calculated and summed over all selected features, with the sign of the sum determining the class prediction. This is termed \textit{weighted voting}. Intuitively, the crossover distance
weighting captures how far away a measurement is from the nearest substantial observation of the class with lower density at the measurement point.

To summarise the voting schemes, let + and – denote the two classes that a sample can belong to. Let \( n_+ \) and \( n_- \) represent the number of samples in the training set belonging to each class. Let \( d_+ \) and \( d_- \) represent the fitted densities of each class. Let \( g \) be the vector of measurements of a selected gene chosen in the feature selection stage. \( sgn \) is the sign function. Unweighted voting can be mathematically expressed as:

\[
\text{mode} \left( \text{foreach } g_i \in g \left( sgn \left( n_+ \times d_+ (g_i) \right \) - n_- \times d_- (g_i) \right) \right)
\]

Weighted voting has the form:

\[
sgn \left( \text{sum} \left( \text{foreach } g_i \in g \left( n_+ \times d_+ (g_i) \right \) - n_- \times d_- (g_i) \right) \right)
\]

A summary of the various feature characteristics and classifiers for them is presented in Figure 3.1.
Figure 3.1 **Summary of feature types and classifiers.** For each of differential expression, differential variability, and differential distribution, a representative gene expression distribution is shown, along with an illustration of the classification process. In the left column, the dashed vertical lines represent the means of the class distributions. In the right column, the variables $x$ and $y$ denote two different genes in a dataset. Each point indicates a sample. The bottom right panel illustrates that each gene from the selected gene set votes independently in differential distribution classification. The black circle shows the position along the x-axis of the expression measurement value. The intervals show two kinds of distance weighting evaluated in the simulation study. The green interval corresponds to crossover distance voting and the black interval to height difference voting.
All classifications in this chapter have the same basic structure. 100 variations of each dataset were created by resampling with replacement followed by 5-fold cross-validation, to obtain the distribution of metrics. The reasons for doing so are described in Section 1.3. For each iteration of cross-validation:

1. The training data is first processed by a feature selection function. In each case, the feature selection function chooses the set of features by testing each gene individually, ranking them by a score, and calculating resubstitution error rates for the top $x$ ranked features. Values of $x$ considered ranged from 10 to 150, in increments of 10. The value of $x$ which obtains the lowest balanced error rate determines the size of the set of top features selected.

Let $T$ be the set of samples in the training set. Let $\hat{C}(t)$ be a classifier function which predicts the class of sample $t$ and let $C(t)$ be a function that returns the known class of sample $t$. Then, the resubstitution error rate is defined by the expression:

$$\frac{1}{n} \sum_{t \in T} 1_{\text{TRUE}}(\hat{C}(t) \neq C(t))$$

$I$ is an indicator function.

2. For only the proposed DV classification, a transformation of expression values is made. For each feature, all samples’ expression levels are subtracted from the median expression level of the training set, and absolute values taken. This transforms the data into a form which allows a classifier that uses linear decision boundaries to be applied.

3. The samples assigned to the training set in the current iteration are used for model building.

4. The samples assigned to the test set in the current iteration have their classes predicted.

The choice of using the resubstitution error for feature selection is a pragmatic one. Ideally, a nested cross-validation would be used. However, this approach is computationally infeasible in practice.
3.3 Evaluation Metrics

Feature ranking stability is assessed by considering the feature rankings of every cross-validation iteration of a particular classification. The top $t$ ranked features are considered for each iteration, and every possible pair-wise intersection of feature identifiers is done. The average size of the intersections is converted to a percentage by dividing by $t$. A range of values between 10 and 100 were assessed. To assess feature selection stability, overlap percentages between all pairs of selected feature sets were calculated as:

$$overlap_{i,j} = \frac{|features_i \cap features_j|}{|features_i \cup features_j|} \times 100$$

The indices $i$ and $j$ are for different iterations of the cross-validation loop of a particular classification. Feature ranking commonality and feature selection commonality are similarly defined, except that the feature set comparison is between two kinds of classification, rather than within a classification.

Balanced error rate is the main performance measure considered, and can be thought of as the average error rate for the two classes, denoted positive and negative. This is defined as

$$BER = \frac{FP}{P} + \frac{FN}{N}$$

where $FP$ denotes the number of false positives and $P$ denotes the number of samples in the positive class. Similarly, $FN$ and $N$ represent the number of false negative and $N$ the number of samples in the negative class. It may be significantly different to ordinary error rates in situations where there is a major class imbalance.
3.4 Simulation Evaluation of DV and DD Varieties

Evaluation of the Bartlett test and Levene test for feature selection has not been carried out before, DV classification has not been characterised in a simulation study, and DD selection and classification methods have never been applied to high-dimensional data, motivating their characterisation.

Each simulated dataset was created by first taking the melanoma dataset used for biological dataset evaluation (Section 3.5) and filtering out any genes showing any differences between classes to create a background set. Then, a subset of genes in the background set were altered to create specific kinds of signals. The detailed steps are:

1) **Background expression:** To define features with differences between classes, six methods were used to rank features of the melanoma dataset in terms of DE, DV, and DD, from largest statistic to smallest. These were based on moderated t-statistic, a Bartlett statistic, a Levene statistic, a DMD distance, a KS test statistic, and a likelihood ratio statistic. Features that appeared in the top 20% of any of the six lists were excluded from the unchanged feature set. This gave 9453 unchanged features and 300 of these features were randomly chosen to be changed to create seven simulated datasets.

2) **Changed Features:** Each of the seven created datasets has varying proportions of changed features. These are DE, DV, differentially skewed (DS) and differentially modal (DM) features. For each dataset, the changing features and their magnitudes were chosen by randomly choosing a class and a direction of change, with both the classes and directions being equally likely. To add noise for DE features, 10% to 30% of randomly chosen samples in the unchanged class were also changed by the same amount. The random sampling of change magnitude was repeated for each feature. The same set of features which were altered for the changed class were altered for the samples belonging to unchanging class. Additionally,
(a) The amount of change for DE features was sampled from a log-normal distribution with mean 1 and standard deviation 1. The change was applied by adding or subtracting the change value from the measurements.

(b) Two varieties of DV features were simulated; consistent and outlier. Each variety was equally as likely to be applied to a feature.

  **Consistent:** For a particular feature, the standard deviation of the chosen samples was increased or decreased by a number sampled from a log-normal distribution with mean 1 and standard deviation 1. This enlarges or shrinks the spread of data symmetrically. Lastly, the values were shifted to keep the original mean.

  **Outlier:** Between 10% and 30% of samples in a randomly chosen class had their expression values increased or decreased by an amount sampled from a Uniform(2, 5) distribution. This simulates another observed pattern of DV in biological datasets.

(c) For DS features, the median expression value of the change class was calculated and either the values lower or higher than this value chosen to be changed. The change to be added or subtracted came from independent samples from a log-normal distribution with mean 1 and standard deviation 1. The change value was multiplied by the distance of each sample on the changing side from the median, before being added to median, to calculate the new value of the changed samples.

(d) DM features were created by calculating the mean of the change class and sampling two mean changes from a log-normal distribution with mean 1 and standard deviation 1. One change is the distance to the simulated lower mode’s mean and the other is the distance to the simulated higher mode’s mean. Additionally, two standard deviation values were sampled from a log-normal distribution with mean 1 and standard deviation 1.
normal distribution with mean 1 and standard deviation $\frac{1}{2}$. Once the changed means and standard deviations were found, random samples (of the same number originally above and below the median value) were drawn from the normal distribution with the mean and standard deviation of each mode. This creates a bimodal distribution.

Both DV feature selection methods select an undesirably large proportion of features from the background feature set; typically about 50% of selections (Figure 3.2A). As expected, features simulated as DE are rarely selected. DV features comprise almost all of the features that are selected. DM features are chosen in similar proportions to their presence in each simulated dataset. The classifier which uses the Bartlett statistic for feature selection has a consistently better median BER than the classifier which performs feature selection using the Levene statistic. (Figure 3.2B). The BERs are higher for datasets in which the simulated DV proportion is smaller. Therefore, feature selection based on the Bartlett statistic is used for biological dataset classification evaluation.
3.4 Simulation Evaluation of DV and DD Varieties

Figure 3.2 Feature selection proportions and balanced error rates of DV classification for seven simulated datasets. A. Proportions of selected genes. The average percentage of selected genes that are in the specified simulated change categories over all cross-validations is shown. The bottom row shows the proportions of simulated changes. B. Balanced error rates of class predictions. The distributions of error rates across all cross-validation iterations are shown as boxplots.

Varieties of DD classification (Section 3.2) were examined that incorporate different choices of how distance between classes is measured to compare weighted and unweighted voting schemes. DD classification is found to perform well across a range of possible simulation settings. Firstly, it was examined whether the four different selection approaches select differing proportions of simulated changed features. The features chosen by DMD mirror the pattern of simulated changes most closely (Figure 3.3A). The KS statistic selects no DV features for most simulations. Regardless of how common
they are, it favours the selection of features that are simulated to be DE. The likelihood ratio selection chooses features in proportions similar to which they were simulated in, but always chooses more unchanged features than the DMD method. About half of the features chosen are those which are simulated to be unchanged. In terms of BER (Figure 3.3B), the DMD selection statistic with weighted classification using the crossover distance has the lowest error rate for most of the simulated datasets. Likelihood ratio and ensemble selection have quite variable balanced error rates between datasets, particularly for the height and sum of differences weightings. Because it selects features in the desired proportions and has good error rates across all seven datasets, the DMD statistic combined with crossover distance weighted voting is used for biological dataset classification evaluation.
Figure 3.3 Feature selection proportions and balanced error rates of DD classification for seven simulated datasets. A. Proportions of selected genes. The average percentage of selected genes that are in the specified simulated change categories over all cross-validations is shown. The fifth row shows the proportions of simulated changes. B. Balanced error rates of class predictions. The distributions of error rates across all cross-validation iterations are shown as boxplots.
3.5 Evaluation of DE, DV, and DD

Three experimental datasets are used for comparison of selection and classification performance. All values from microarrays were transformed to the $\log_2$ scale. Cases in each dataset were partitioned into a good prognosis class and a poor prognosis class. An external database (MalaCards) was obtained for evaluation of feature selection in terms of previously disease-associated genes.

**Melanoma:** The raw microarray expression and clinical data are available from GEO as GSE54467. The samples were assayed on the Illumina Human WG-6 BeadChip microarray, version 3. Previously defined classes for this dataset (Jayawardana et al., 2015) of poor prognosis as death less than one year from metastasis ($n = 22$) and good prognosis as survival of more than four years with no signs of recurrence ($n = 25$) are considered. Raw data was NEQC normalised (Shi, Oshlack, & Smyth, 2010) and probes which had less than ten samples with a detection $p$-value of $< 0.01$ were removed from further analysis. The detection $p$-value is calculated by the microarray scanner’s control software and gives the probability that the gene is not expressed. This leaves 17699 probes to use for classification.

**Serous Ovarian Cancer:** Processed microarray data generated by the study GSE13876 were obtained from the Bioconductor package curatedOvarianData (Ganzfried et al., 2013). Gene expression was measured with Operon Human (version 3) microarrays developed by the Netherlands Cancer Institute. 20577 genes are measured in this study. Poor prognosis is defined as death within two years ($n = 22$) and good prognosis as survival of five or more years ($n = 25$).

**Lung Adenocarcinoma:** The processed data was obtained from TCGA Data Portal (The Cancer Genome Atlas - Data Portal, n.d.) on 16 May 2014. Poor prognosis cases were defined as those who died less than one year from diagnosis and good prognosis cases as those who lived for over four years, with no signs of recurrence. This resulted in a total of 18 poor prognosis and 18 good prognosis samples.
Sequencing was performed on an Illumina HiSeq 2000 instrument. Normalised gene count values were used. Genes that had fewer than ten counts in less than ten samples were removed from further consideration. This left 16321 genes for use in classification.

The follow-up time densities using all samples from each dataset show that the TCGA lung cancer dataset has relatively few cases with long follow-up times, while the other two datasets have a greater variety of follow-up times (Figure B-1).

**MalaCards:** Cards for melanoma, ovarian cancer, and lung cancer were downloaded from the MalaCards website (http://www.malacards.org) on 25 March 2014. These are gene lists for particular diseases with scores for each gene proportional to that gene’s association to the disease in the published literature (26).

In the ovarian cancer dataset, some features were annotated with multiple alternative gene names. For those genes, the maximum MalaCard score of matching names was chosen.

DE, DV, and DD classification all display a similar error profile for all three cancers. Figure 3.4A shows the balanced error rates for all three datasets classified by all three types of classification. For melanoma, the median balanced error rate for the resampling and folding validation is 23%, 26% and 24% for DE, DV, and DD, respectively. A similar pattern is observable for the other two cancers. Ovarian cancer is the most difficult to classify in all cases whereas lung cancer has the lowest median error rate for each classification type. The spread of the BER values is large. For example, each of the BER distributions has values as low as 5% or as high as 50%, depending on the iteration of the cross-validation. By chance, it would be expected to obtain BERs of 50%, so all three classifications almost always perform much better than classification at random.

The error rates of individual patients are also similar between methods. That is, some samples are systematically classified poorly for all three gene assessment types, while others are typically classified
correctly. The percentage of misclassifications of every sample was calculated for each method and plotted as an error map (Figure 3.4B). Darker shading indicates correct predictions are made in more cross-validations. For example, melanoma patient TB52 is classified poorly by all three methods, whereas TB36 is always classified well. There are more misclassified samples in the smaller class for ovarian cancer. The lung cancer dataset has balanced class sizes and no tendency in misclassification is observed.
3.5 Evaluation of DE, DV, and DD

Figure 3.4 Cross-validation balanced error rates and sample-wise error rates. A. Distribution of balanced error rates over all iterations of cross-validation. B. Sample-wise error rates. Each patient is one column of a heatmap. Each classification type is one row of a heatmap. Details of the selection and classifier algorithms are provided in the Methods section. The error rates are binned into five equally sized bins. Colour scales are shaded by class colour, with a darker colour indicating less frequent misclassification than a lighter colour.
The genes selected are largely different for each of the three selection types. Considering all the samples for each dataset, the top DD-ranked 50 genes ranked have only minor overlaps with both DE- and DV-ranked genes (Figure 3.5A). Ovarian cancer has the smallest overlaps between methods, with only one gene in common between possible pair of selections. For the ovarian cancer and lung cancer datasets, one gene is common to all three selection types. Regardless of the dataset considered, all pairwise overlaps between selection types are 12% or less of the size of the set union.

The selected features also have little overlap (Figure 3.5B). For melanoma and lung cancer, only one feature is chosen by all three assessment types. For melanoma and lung cancer, the DE and DV selections have just one gene in common. There are no common genes between DE and DV selection for ovarian cancer. The size of the gene list chosen also varies widely from dataset to dataset. DE selection gives the most compact sets, ranging from 10 to 30 features. DD selection for lung cancer gives the biggest set of features, choosing all 150 that were considered. It also chooses 110 features for the melanoma dataset, while the other two methods choose 10 or 20 features.

Although these sets of genes had little overlap, the DD-selected genes are enriched for those of biological significance (Figure 3.5C). Considering the top-ranked genes, DD selection provides the most disease-associated genes for the melanoma and ovarian cancer datasets over different subsets of top-ranked genes. For lung cancer, it selects almost as many high-scoring genes as DE. DV selection provides the lowest cumulative score of previously disease-associated genes for every dataset. The low recall of DV is clearest for the lung cancer dataset, where the cumulative sum plateaus at top rankings no higher than 50. The top-ranking DV features of ovarian cancer are almost entirely unassociated with the disease.
Figure 3.5 **Feature ranking and selection overlaps.** **A.** Overlaps between three feature selection types for three cancer datasets. The 50 highest-ranked genes of each method are used. All samples are used. **B.** Overlaps between three feature selection approaches for three cancer datasets. The genes selected by best resubstitution error rate of each method are used. All samples are used. **C.** Cumulative MalaCards scores for the most frequently selected features in cross-validation.
In cross-validation using the cancer datasets, differential distribution selection yields more features in common over all pairs of cross-validations. Considering the highest ranking genes, the DD-selected features are typically two to three to times more stably ranked than DE features, depending on the dataset (Figure 3.6A). For lung cancer, the stability is as much as six times higher. DD feature ranking is also more stable than DV feature ranking for the ovarian and lung cancer datasets. The lung cancer dataset benefits the most from DD selection, with stabilities ranging between 30 and 35 percent. The lung cancer dataset has the highest selection stability overall, whereas the ovarian cancer dataset achieves the lowest.

Feature selection based on differential distribution is also the most stable (Figure 3.6B). The DD median selection score is about twice as large as the second highest median score, except for melanoma, where it is nearly identical to DV’s score. For the ovarian and lung cancers, the second highest median score is from DV selection. The median selection score of DE selection is the lowest in every dataset. The interquartile range for DD scores is the largest, except for melanoma, where DV has a slightly larger spread. The expression distribution of the most frequently selected feature in each dataset and of each feature type is illustrated (Figure B-2). The most stable genes for ovarian cancer have much less noticeable differences between survival classes than the most stable genes for the other two cancers.
Figure 3.6 **Cross-validation feature ranking and selection stability.** For each pair of comparisons, the number of genes in common is divided by the number of genes in the union and converted to a percentage. A. The average pairwise overlap of the top ranked genes is calculated for all iterations of cross-validation. Shapes represent datasets and colours represent different types of classification. B. The distribution of the pairwise overlaps of the selected genes is calculated for all iterations of cross-validation. From left to right, the number of data points which are greater than 20% and are not shown as points is: 475, 16879, 5301, 305, 593, 5884, 963, 2489, and 29384.
3.6 ClassifyR Classification Evaluation Framework

All of the previous results were generated with ClassifyR, a customisable framework for classification and its evaluation (Strbenac, Mann, Ormerod, & Yang, 2015). ClassifyR has been assessed and accepted into the Bioconductor project. It is publicly available for download at http://www.bioconductor.org/packages/release/bioc/html/ClassifyR.html. A tutorial with runnable code based on a public benchmarking dataset is also available from the same webpage. The framework is implemented as four modular components. The data transformation stage takes a matrix of data and returns a matrix of transformed data. This stage is optional. The feature selection stage takes a matrix of training data and returns a vector of indices indicating which features were selected and a ranking of all features. Training uses the training samples and the selected features to build a classifier. Finally, the prediction stage takes a matrix of test data and returns a vector of class predictions. The order that stages are executed in is user-defined, which allows some unusual classifiers, such as nearest shrunken centroids (Tibshirani, Hastie, Narasimhan, & Chu, 2003), to work with the framework. Nearest shrunken centroids is unusual in that it requires classifier training before feature selection. All of the feature selection and classification algorithms evaluated in previous sections have defined functions in the package, as well as some classifiers not used in this study, such as the nearest shrunken centroids classifier and fitting of mixtures of normals followed by posterior probability class prediction. If the user doesn’t provide any specification of feature selection and training parameters, by default, a limma moderated t-test is done and a DLDA classifier is used to perform 100 resamples and 5-fold cross-validation.

Three varieties of cross-validation are available to the user through the function runTests. $r$ resamples and $k$-fold involves resampling the samples which comprise the dataset with replacement $r$ times. Similarly, $r$ resample and $s\%$ split resamples the samples with replacement, but it splits the newly
created datasets such that $s\%$ of randomly chosen samples form the training set and $100 - s\%$ of samples form the test set. In other words, unlike the $r$ resamples and $k$-fold scheme, not all samples in each resampled dataset are used for prediction. Thirdly, leave-$k$-out uses all combinations of $k$ samples from the dataset as the test set and the remainder of the samples as the test set. Values of 1 or 2 are typically used, as larger values of $k$ are computationally intractable. Each iteration of cross-validation is independent from all others. Therefore, cross-validation is implemented to easily run on multiple processor cores. Classifications of the melanoma dataset demonstrate that the time savings of using multiple cores are substantial (Table B-1). Apart from cross-validation, the situation of independent training and test sets is handled by the function `runTest`. In fact, the cross-validation functionality available from `runTests` involves repeated use of `runTest` for each iteration of cross-validation.

Virtually any feature selection method or classifier can be used with ClassifyR. This is enabled by a formalisation of what parameters each of the four stages requires (Table B-2). Also, the function which processes each stage has rules about the order of variables passed into it (Section B.2). Many classification functions available from other R packages already coincidentally conform to these rules. However, even those functions which do not can easily be used by writing a short wrapper function around them. An example of a wrapper for the $k$ nearest neighbours classifier from the R package `class` is shown in Section B.2. A number of wrappers are provided by ClassifyR, such as the `classifyInterface` function which wraps the PLDA implementation in the package PoiClaClu.

Feature selection and class prediction may be easily evaluated by a number of convenience functions. These evaluate feature ranking and selection stability, feature ranking and selection commonality between classifiers, sample-wise error rate, cross-validation error rate, receiver operating characteristic curves (for classifiers with a continuous predictor), and calculation of over 20 performance metrics (e.g. Matthews Correlation Coefficient, $F_1$ score). Details are available in the software documentation. All of
these evaluation functions takes as input the direct output of the classification procedure or a list of such outputs. Classification evaluation is streamlined and reproducible by this style of implementation.

3.7 Discussion and Conclusion

Stable and accurate prediction of sample classes from gene expression signatures or other kinds of omics datasets remains a challenging problem in cancer prognosis and omics research. Previous research (Phipson & Oshlack, 2014), as well as this study, has found that DE and DV methods select mostly different sets of genes. Feature selection using all samples in a dataset had a minimal overlap, ranging from 1% to 5% for the cancers considered. These large differences in selected sets motivated the development of measures that seamlessly combine the characteristics of both DE and DV methods. Here, a kernel density-based DD measure has been developed with a corresponding prognostic algorithm and evaluation done to demonstrate that it performs well in terms of classification and stability on both simulated and three sets of high-dimensional transcriptome datasets.

Feature selection stability is an important problem, because if a feature is selected infrequently in a resampling procedure, that feature may have been selected by chance and not related to the outcome of interest, which limits its translational potential. The sources of selection instability have been previously characterised and ensemble feature selection proposed as a solution (Yang et al. 2013). This, however, is computationally costly, as it requires the training of many similar models. Also, the combining of features from different models is subjective, and depends on a user-specified parameter of how many models the feature should be selected in to be used in the final model. In comparison, differential distribution selection by the DMD method has been shown to always be more stable than the popular moderated t-statistic and equally or more stable than the Bartlett statistic, without requiring the generation of multiple models and subjectively aggregating them. As shown by Figure 3.5C, for each
dataset, the DMD selection type chose more genes in common with MalaCards than DE or DV in two out of three datasets, suggesting that DD has more power than either of the alternatives.

Assessment of genes via DV remains a potentially desirable type of classification when one seeks to classify samples using mostly experimentally-unexplored genes. The top ranked DV genes for each dataset had the least overlap with currently well-known disease-associated genes, as defined by MalaCards (Figure 3.5C). Although this observation may seem concerning at first, it is actually expected and can be explained by publication bias. Almost all prior research work on biomarkers has focussed on obtaining markers that have a systematic change in expression between conditions. Only one study has attempted classification with differential variability (Teschendorff & Widschwendter, 2012) and the lack of variability-associated disease genes in public databases is likely to end, once more studies begin to consider DV or DD classification.

Figure 3.4 shows that ovarian cancer had a higher BER than the other two datasets for all three classification types, as well as more patients with high patient-specific error rates. The difficulty of ovarian cancer survival prediction has been demonstrated recently (Lisowska et al., 2014). The lowest FDR value obtained from fitting a Cox regression model to each gene was 0.85. Selecting the four best genes simply based on odds ratio magnitudes and testing them on an independent dataset found that none of them were transferrable to an independent dataset.

Considering Figure 3.4B, it is evident that error rates for a minority of patients were very high when assessing genes via DE, DV or DD. Although most patients in each class were classified correctly at least 80% of the time, a small number of patients were classified incorrectly in the majority of cross-validations. The frequent incorrect classification happened regardless of the type of classification done. This could be as a result of differences in medical treatment or other unspecified confounding factors. For example, two patients could each have a gene signature that is associated with poor prognosis, but
one patient may have received better surgical treatment than the other and, therefore, survive a long time. Therefore, although both patients have a gene signature indicative of early death, one patient may survive a long time. This issue has recently been explored for invasive breast carcinoma where prognosis prediction was shown to be confounded by oestrogen receptor status, causing some samples to be systematically misclassified. Grouping patients by aspects of their clinical data before creating separate classifiers for those sample groups is a promising new research direction. It provides extra motivation for researchers not only to increase their sample sizes, but to obtain thorough clinical data, in order to make these analyses possible.

In summary, compared to DE and DV, assessing genes via DD chooses different genes, selects features in a more stable manner, and provides competitive balanced error rates. DE classification only detects changes in means, and misses signatures of transcriptional deregulation. However, if the biological conditions being studied are rare, DE classification may be the only approach possible. The results presented here show that the DD approach selects a different set of features with greater biological relevance than DE, while maintaining good prognostic accuracy. DD classification is, therefore, a superior approach for assessing genes, which provides good classification accuracy and a complimentary set of biological features to DE or DV selection for biologists to pursue experimentally.
Performance of Proteomics Summarisation and Normalisation Methods

Proteomics promises to reveal the complexity underlying the mechanisms of disease and development by directly assaying the abundance of thousands of proteins, the functional end products of the central dogma of molecular biology. As with all omics technologies, preprocessing is crucial for obtaining accurate and precise measurements. To date, only a limited number of studies provide comparisons of preprocessing methods for proteomics data and none are comprehensive. The development of comparison studies which evaluate a specific platform is an on-going challenge and often requires specially designed experiments. A recently released experiment (P. Wang, Yang, Raftery, Zhong, & Wilson, 2013) based on a replicated Latin squares design consisting of 21 spike-in proteins ranging in abundance over 64 folds and a yeast proteome background has been utilised to evaluate a range of preprocessing options. This experiment now enables, for the first time in proteomics, the development of a comprehensive evaluation framework, similar to one created for Affymetrix gene expression arrays (Irizarry, Wu, & Jaffee, 2006).

The evaluation of statistical methods for proteomics data is made possible by two main contributions. Firstly, a comprehensive set of performance metrics for a range of analysis scenarios have been developed, based on a replicated Latin squares experiment, enabling the thorough examination of sources of bias and variance. Secondly, a web-based application has been created that allows easy
summarisation of a large number of combinations of performance metrics and their distributions across various experimental aspects. This enables the scenarios in which a preprocessing method performs well to be identified. Notably, the web-based application is publicly accessible to anyone with an internet browser; users can upload their own preprocessed dataset to compare their own statistical methods in an unbiased way against those previously characterised. These developments allow the performance comparison of a range of important preprocessing decisions. Similar to the previous evaluation of types of changes by differential expression, differential variability, and differential distribution for classification problems (Chapter 3), combinations of numeric summarisation strategies are evaluated for the first time and recommendations about the most suitable approaches made. Additionally, the effect of between-sample correction methods is characterised. A large number of datasets for a variety of biological systems is becoming available to researchers through raw dataset repositories such as ProteomeXchange (Vizcaíno et al., 2014) and RefData (Gatto et al., 2016). Optimal preprocessing of a public dataset may reveal new findings which were missed in the original study.

The remainder of this chapter is organised as follows. In Section 4.1, an introduction to proteomics is followed by a survey of existing evaluation datasets, metrics used, and benchmarking frameworks. Section 4.2 introduces the new performance metrics to evaluate bias and variance. Section 4.3 demonstrates the key capabilities of the web-based application, such as user-preprocessed data upload and interactive performance metric summaries. Section 4.4 uses the web-based application to compare bias and variance of protein summaries in three different quantitation scenarios. Finally, in Section 4.5, the performance of the statistical preprocessing methods is discussed, and ideas for future work are proposed.
4.1 Background

Firstly, an overview of protein identification and quantitation by mass spectrometry is made. Then, various designs for performance evaluation are introduced, their strengths and weaknesses are discussed, and their use in previous studies is highlighted. One specific replicated Latin squares design is focussed on, which forms the basis of the three case study evaluations presented in Section 4.4.

4.1.1 Mass Spectrometry-based Proteomics

Experiments to characterise the proteome begin by cutting the proteins into shorter amino acid sequences, called peptides (Rauniyar & Yates, 2014). The peptides are firstly separated based on their masses. Next, the peptides are subjected to two stages of mass spectrometry. In the first stage, the peptides are separated by their mass and charge. Before the second stage, the peptides are captured and broken in random positions. Ideally, each peptide is broken in a single location. The peptide sequences remaining after breakage are again measured in a mass spectrometer, and because there will ideally be a breakage between every amino acid, the difference between the masses of a sequence of length \( l \) and length \( l + 1 \) indicates what the \( l + 1^{th} \) amino acid is. This allows the reconstruction of the entire sequence of the peptide. The determination of peptide sequence is difficult for many reasons. Some peptides are not broken in every possible position, which leads to ambiguities in the mass differences between consecutive peaks. This is because any two amino acids have the same total mass, no matter which order they are present in. Also, there are six types of breakages possible, and spectra tend to contain between two and three of them.

Peptides can be quantified by a technique such as iTRAQ, which adds molecules of a known mass to the fragmented peptides. The labels are measured in a separate region of the spectrum to the peptides. This type of proteomics is called labelled and enables multiple biological samples to be measured together in a single instrument run. Two versions of the iTRAQ labels are available; 4-plex, which has four labels,
and 8-plex, which has eight. The other alternative, known as unlabelled proteomics, involves calculations (e.g. peak height, peak area) using the determined peaks in the first stage of mass spectrometry. Clearly, for unlabelled proteomics, only one biological sample may be analysed per instrument run.

Determining the amino acid sequence for a spectrum, commonly known as Peptide-Spectrum Matching (PSM), is necessary to associate a spectrum with a known protein. The first software to perform this task was SEQUEST (Eng, McCormack, & Yates, 1994). The method is able to account for modified amino acids but has no statistical framework for controlling false discoveries. A few years later Mascot was released (Perkins, Pappin, Creasy, & Cottrell, 1999). It was the first software to have a statistical model that assigns a probability to each inference. More modern PSM algorithms, such as Paragon (Shilov et al., 2007) and MS Amanda (Dorfer et al., 2014), also have the ability to assign peptides to proteins when ambiguity exists. Peptide sequences may match multiple proteins in the database, but may be assigned to a particular protein after considering other peptides matching to unique portions of proteins. A comparison of a range of these algorithms and their performance on eight benchmarking datasets has recently been made (Tu et al., 2015), finding that no search algorithm had consistently superior performance.

Once peptides are identified, uniquely matched to proteins and have quantities associated with them, they need to be summarised to the biological units of interest - proteins. There are many potential ways to summarise numerous peptide values into a single summary value for a protein. Popular approaches include the median peptide (Sturm et al., 2008), sum of the three peptides with the largest quantities (Ning, Fermin, & Nesvizhskii, 2012), and sum of all peptides divided by the number of theoretically observable peptides (Schwanhäusser et al., 2011). Summarisation methods can have significant impact on inferred protein quantities, but their evaluation has been limited in scope to date.
Finally, normalisation must be performed to remove sample-to-sample and batch-to-batch variability. Both of these unwanted kinds of variability would decrease the number of proteins statistically determined to be differentially abundant, if not accounted for. Many normalisation methods, such as the exponentially modified protein abundance index (Ishihama et al., 2005) and normalised spectral index (Griffin et al., 2010), are developed on label-free data and use spectral counts or peptide counts, which discard the rich quantitative information contained in iTRAQ experiments. Normalisation procedures such as median centering or z-score transformation of the precursor ion intensity (Webb-Robertson, Matzke, Jacobs, Pounds, & Waters, 2011), have also been infrequently used. However, this evaluation of normalisation approaches was based on unlabelled data and biological samples were used, which means the truth set of differentially abundant proteins is unknown.

Unlike label-free experiments, labelled experiments inherently have batches of samples. Only one study has proposed a solution to remove such batch effects (Hill et al., 2008). A log-linear model with terms for various aspects of the experiment, such as iTRAQ channel and run number, as well as the experimental design, is fitted. Then, a normalised protein value is found by subtracting away the fitted coefficients for the technical effects. The performance of this method has not been evaluated independently. Other possible approaches include calculating a scaling factor for each sample, as ProteinPilot does, or using modern techniques such as Removing Unwanted Variation, commonly abbreviated to RUV (Jacob, Gagnon-Bartsch, & Speed, 2016). More details of these methods are provided in Section 4.4.2. These techniques, however, have not previously been evaluated in the proteomics field. In genomics, the MaQC-II study (Luo et al., 2010) corrected six datasets with different kinds of batch effects with five algorithms and the classification performance was found to be improved in the majority of scenarios for a subset of the correction algorithms.
The general process of converting the raw data obtained from a mass spectrometer into useful protein measurements is summarised by Figure 4.1.

Figure 4.1 **Flowchart of proteomics data preprocessing.** Steps of preprocessing are represented by boxes and particular methods to evaluate are listed on the right side. Grey steps or option lists are not evaluated in this study.

4.1.2 Evaluation of Quantitation

All of the described preprocessing steps can have substantial effects on the protein measurement summary and motivate the need for evaluation of competing methods to determine which method, if any, is superior.
4.1 Background

Evaluation in Genomics and Transcriptomics

Much effort has been placed into designing experiments that facilitate the evaluation of competing algorithms in genomics and transcriptomics. The first study of its kind, AffyComp, came in response to the confusion and uncertainty surrounding the best-performing normalisation algorithm out of the many published (Cope, Irizarry, Jaffee, Wu, & Speed, 2004). It compared a variety of microarray normalisation methods on the basis of fourteen different performance metrics, using a spike-in Latin square design. The experimental design used had sixteen different transcripts spiked in at fourteen different concentrations. The concentration of a particular transcript differed on each of the fourteen microarrays. Metrics to evaluate bias were all based on the difference between observed and expected fold changes between consecutive spike-in amounts. Variance was evaluated by summarising the mean standard deviation of biological replicates and the IQR of log₂-fold changes of genes unchanged in expression. This framework lead to the subsequent discovery that the background correction algorithm used is a critical step of data preprocessing (Irizarry et al., 2006).

Another spike-in Latin square design enabled the bias and variance of different copy number prediction algorithms to be evaluated for the first time (Halper-Stromberg et al., 2011). The evaluation mostly consisted of an expected to observed fold change regression slope and replicate variability. Systematic bias such as the intercept of the regression line was not considered. This work is notable for comparing six different microarray models from three different manufacturers. Important conclusions to arise from this study were that there are strong artefacts in the raw data not corrected for by all manufacturer software except one, but corrected for by almost all academically-developed software. Between-manufacturer evaluation also found that arrays made by NimbleGen had the best overall performance. This illustrates the importance of comparing data preprocessing software provided by the instrument’s manufacturer to software developed in academic research.
An immense variety of performance metrics have been used to recently evaluate RNA-sequencing experiments (SEQC/MAQC-III Consortium, 2014). Two samples, the Universal Human Reference and the Human Brain Reference, were mixed in ratios of 1:3 or 3:1 and distributed to seven research institutes around the world. This design creates a range of expected fold changes between 1/1000 and 1000. Bias was evaluated by comparing the expected to measured fold changes. Variance was evaluated between samples at the same facility using the same or different instruments and also between facilities. The generated data was also processed with six different bioinformatics tools. This allowed a detailed characterisation of the sources of variability. The evaluation performed found that no approach could accurately quantitate absolute gene expression, but all approaches had similarly good performance for relative quantitation. A novel finding from this study was that gene expression microarrays showed higher reproducibility than RNA-sequencing between research institutes, which has implications for collaborative clinical research. Other recent evaluative efforts (Germain et al., 2016; Teng et al., 2016) provide similarly detailed recommendations about the best analytical choices for various scenarios.

*Evaluation in Proteomics*

Comprehensive evaluation of protein quantitation, rather than protein identification, has been rare for proteomics data. The best methods for most situations are still unknown. The earliest evaluation study consisted of six proteins spiked into a complex mixture at six different concentrations to form six samples, each measured by a separate unlabelled experiment (Mueller et al., 2007). Considered together, the six designed samples form a Latin square. Each sample was analysed three times. Performance evaluation was minimal; variance was indirectly assessed by clustering the samples using k-means clustering and also as the standard deviation of the three technical replicates for each protein. Bias was indirectly addressed by plotting the protein quantitation and expected quantitation of every sample.
Other limitations of the study are that no comparisons of any algorithms or parameters were done and the size of the spike-in proteins was not systematically varied in the experimental design.

A small number of other datasets which make use of spike-ins at specific concentrations have been published (Mahoney et al., 2011; Tuli et al., 2012; Matzke et al., 2013; Pursiheimo et al., 2015; Shalit, Elinger, Savidor, Gabashvili, & Levin, 2015), but they have used simple dilutions of particular samples or compared two conditions which have a presence or absence of spike-in proteins. These studies do not allow the exploration of the interaction between protein amount and experimental quantitation, since it has not been controlled for. For instance, in Mahoney and others’ study, there are five quantity levels in the experimental design, but each protein only appears at two of the five levels. Another difference is that iTRAQ experiments have different quantitation characteristics to unlabelled experiments. Almost all prior evaluation studies are unlabelled experiments. iTRAQ experiments differ to these, as they have inherent experimental batches of either four or eight samples per batch and a downward bias of measured fold changes (Karp et al., 2010), particularly with single quadrupole time-of-flight instruments (Rauniyar & Yates, 2014). This means that findings from unlabelled experiments should not be extrapolated to labelled experiments. Additionally, the performance metrics evaluated have been limited. For example, Matzke and colleagues made conclusions focussed on method run time and software user interfaces, rather than the accuracy or precision of the normalisation methods compared. Tuli and others’ evaluation was only slightly more informative; it compared the number of statistically significant protein differences between two conditions processed by four publicly available preprocessing methods.

*Existing Evaluation Software*

Due to the rarity of comprehensively designed datasets in the proteomics field, there is limited software available to evaluate and compare preprocessing algorithm performance. Normalizer (Chawade, Alexandersson, & Levander, 2014) is the first software that attempts to enable such comparisons. The
4.1 Background

Software allows users to load their proteomics dataset into R and applies twelve normalisation methods to it, such as LOESS normalisation, before producing simple dataset summaries, such as mean-variance plots. Normalyzer has a number of limitations. Firstly, it forces twelve normalisation methods to be applied to the dataset, disregarding whether the methods are applicable to the experimental design of the study. It provides no flexibility for the user to provide a dataset normalised using their own novel method. Secondly, Normalyzer is intended for generic biological experiments rather than designed evaluation studies, so the evaluations (e.g. coefficient of variation, quantile-quantile plots) are restricted to examining unwanted variance. Finally, unlike a web-based evaluation application, it is standalone software, and comparisons between methods of different researchers cannot be centrally disseminated.

These limited comparisons of summarisation approaches and unevaluated batch correction methods motivate the systematic comparison of performance metrics with a well-designed dataset.

*Latin Square Evaluation Dataset*

A recently public dataset which allows for thorough examination of the sources of technical variability is utilised. The raw mass spectra and peptide-spectrum matches are available from ProteomeXchange as dataset PXD003608. The underlying experimental design and laboratory work were performed in 2012 and the laboratory-based component of the project was done at the Bioanalytical Mass Spectrometry Facility at the University of New South Wales.

The experiment consists of 21 known proteins, each purchased from Sigma-Aldrich (Table C-1) and the yeast proteome, to add background complexity to the protein mixture. The non-yeast proteins are referred to as *spike-in proteins* and yeast proteins are referred to as *background proteins*. The spike-in proteins selected for purchasing were based on characteristics such as protein purity, lack of overlapping
4.1 Background

segments with background proteins, protein solubility, and the practical considerations of price and delivery time. More details about the choices made are available elsewhere (Wang et al., 2013).

All proteins were labelled in each run with eight iTRAQ channels. For each experimental run, each spike-in protein was added at a different concentration in each iTRAQ channel, except in channel 121, referred to as the internal reference channel. The other seven channels are referred to as dynamic channels. A sample is defined as a particular iTRAQ channel of a particular experimental run. The proteins are grouped into seven groups, such that each group contains one small, one medium, and one large protein. The amount of each protein group used in each channel is shown for Run 1 (Table 4.1). The full design matrix is available as Table C-2. The spectral measurements were made with an AB SCIEX QStar Elite mass spectrometer. Spectra were acquired with Analyst QS 2.0 instrument software.
Table 4.1 **Experimental Design for Run 1.** The values in the table are volumes of diluted protein used, in units of microlitres.

<table>
<thead>
<tr>
<th>iTRAQ Channel</th>
<th>Protein Group</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>113</td>
<td>113</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>114</td>
<td>114</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>115</td>
<td>115</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>116</td>
<td>116</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>117</td>
<td>117</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>118</td>
<td>118</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>119</td>
<td>119</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>121</td>
<td>121</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

The Latin square dataset has been evaluated previously with the use of default normalisation settings in ProteinPilot (Wang et al., 2013), but only the regression slope of the entire dataset was evaluated, ignoring all aspects of variance and the effect of runs and channels on the slope metric. Therefore, the web-based application and proposed performance metrics are needed to comprehensively evaluate summarisation and normalisation methods of ProteinPilot and a variety of other methods.

To summarise the raw data to normalised protein-level quantities, five out of six key stages of preprocessing were performed. Missing data treatment is not included because this dataset has a low percentage of missing peptides (Table C-3). Mass spectra were acquired from an ABI QStar Elite instrument. The spectra were searched using ProteinPilot’s Paragon method (Shilov et al., 2007) against a custom protein database, which contains all proteins for all of the source organisms in the March 2015 release of SwissProt (Consortium, 2015) database. Version 5.0 of ProteinPilot was used. Parameter settings used by the algorithm were:
Cysteine alkylation: Iodoacetamide

Protein digestion: Trypsin

Special factors: Carbamidomethylation of cysteine, oxidation of methionine.

Search effort: Thorough

Approximately the same number of proteins were identified from the background set and the spike-in set (Table C-4).

For peptide quantitation, the reporter ion areas were corrected for ion impurity using the default impurity table provided by ProteinPilot. Bias factor scaling and background correction were both turned off. Quantitated spectra were exported from ProteinPilot to text files using the Peptide Summary feature of the Export panel. Spectra were summarised to peptides by summing the areas for each channel for all spectra which had the same peptide sequence within each run and had confidence of more than 95%. These peptide summaries form the basis of subsequent summarisation and normalisation analyses in Sections 4.4.2 and 4.4.3. The data was also processed with the default bias and background correction steps, and evaluated in Section 4.4.1.

4.2 Performance Metrics

To comprehensively evaluate statistical methods in terms of bias and variance, eight distinct performance metrics are proposed for absolute and relative quantitation scenarios, providing a total of fifteen performance metrics. The metrics may further be categorised as regression metrics and summation metrics.

**Regression Metrics**

The regression metrics examine the relationship between expected $\log_2$-fold change vs. experimental $\log_2$-fold change based on a linear regression fit between these two fold changes. They are:
4.2 Performance Metrics

- **Intercept**: Ideally, the value is zero, which means that proteins which have no change are also measured as having no change. The further the intercept is from zero, the more systematically inaccurate the fold changes are.

- **Slope**: An ideal value of one means that the measured fold change increase at the same rate as they do in the experimental design. Values between zero and one suggest that the fold changes are systematically underestimated.

- **Line Difference Sum**: This variance-based metric is defined as the sum of the absolute differences between the fold change values predicted by the regression line and the expected values from the experimental design. It combines biases which affect either the slope or the intercept into a single value and provides an overall goodness of fit between the observed and expected fold changes. In a perfect situation, it is zero.

- **Residual Sum**: This variance-based metric is the traditional sum of absolute residuals and is ideally zero. It could be close to ideal, even if the slope and intercept are far from their ideal values.

Two kinds of regression lines may be constructed for this evaluation. All four of the above metrics may be calculated for both lines. In the absolute quantitation scenario, each protein summary in a sample is divided by the median summary of the proteins present at the median design amount of 8 µL in that sample. In this way, the measured amount of different proteins are directly compared to each other. The main experimental factors to affect these metrics are precursor ion coelution and different measurement efficiencies of different peptides. The second kind of linear regression is for relative quantitation. The protein summary of each protein is divided by the same protein’s summary in the internal reference channel. The key factors to affect these metrics are any systematic difference between the samples and also precursor ion coelution. These two regression lines provide a total of eight performance metrics.
**Summation Metrics**

An additional set of metrics examines unwanted variance by considering the total amount of protein per channel, protein group, or experimental run based on summation within the particular grouping, followed by calculation of the coefficient of variation (C.V.). The ideal value for all summation metrics is zero. By design, all spike-in protein summaries for each dynamic channel have the same total amount of protein used. Also, every protein is present at the same total amount across all channels within a run. The metrics which make use of these important properties of the experimental design are:

- **C.V. Channel Sums:** Protein summaries are summed within each iTRAQ channel and the C.V. is calculated. It provides an indication how similar channels are with each run and also in general.

- **C.V. Protein Group Sums:** This metric is calculated by firstly summing the protein summaries of proteins within each protein group across all dynamic channels. Factors which affect this metric are proteins which have systematically high or low signal, undetected proteins, and binding of variable efficiency between peptides and particular iTRAQ labels.

- **C.V. Size Group Sums:** Instead of grouping proteins by their allocated group in the experimental design, proteins are grouped into three size groups; small, medium, and large. Seven proteins belong to each group. This metric is particularly affected by the decreasing number of peptides available to calculate protein summaries, as protein size decreases (Figure C-1).

- **C.V. Median Run Proteins:** The median spike-in protein summary is found for each run and the C.V. of those summaries is calculated. The factors which has the most effect on this metric are any extrinsic differences between runs, such as if the instrument has been recalibrated or differences in laboratory environmental conditions, such as room temperature.
The first three summation metrics are calculated within each run and also across all runs, resulting in six separate metrics. Therefore, there are seven summation metrics in total.

None of the metrics use any protein summaries from the internal reference channel or the background proteins to ensure that metrics are independent of the protein summaries used for normalisation. All metrics, the scenarios they are relevant to, the preprocessing stages they are affected by, and the undesirable characteristic they evaluate are summarised in Table 4.2.

Table 4.2 Summary of performance metrics and their characteristics. The applicability of a metric to the quantitation type, preprocessing stage, and error characteristic is shown. (R): only affects relative quantitation. The number next to the metric name shows its column name in the web-based application.

<table>
<thead>
<tr>
<th>Metric (Column ID)</th>
<th>Quantitation</th>
<th>Preprocessing Stage</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Absolute</td>
<td>Relative</td>
<td>Summarisation</td>
</tr>
<tr>
<td>Intercept (1, 5)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Slope (2, 6)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Line Difference</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sum (3, 7)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Residual Sum (4, 8)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C.V. Channel Sums</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(9, 13)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C.V. Protein Group Sums (10, 14)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C.V. Size Group Sums (11, 15)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C.V. Median Run Proteins (12)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

4.3 Web-based Application

To enable the unbiased and reproducible evaluation of three novel case studies (Section 4.4) and also to allow researchers world-wide to develop their own data preprocessing methods and compare their method’s performance to existing approaches, a web-based application has been developed. Its URL is

The web-based application has three sections; a help page, an evaluation matrix page, and a dataset upload page. To enable researchers to compare their novel methods with existing approaches, the New Processing Upload page (Figure 4.2A) enables users to upload their own preprocessing of the dataset for automated performance metric evaluation for the fifteen metrics defined in Section 4.2. It consists of a number of drop-down selection lists that are used to describe each of the main steps of standard proteomics data processing. If the user has a new method for a particular preprocessing stage, it can be typed in and will be added to the list of options. The application expects the input data to be a tab-separated text file with one protein record per row, and columns for the protein identification, experimental run and quantity summary amount of each iTRAQ channel or alternatively the fold change to the internal reference channel. Absolute quantitation and summation metrics are not calculated if fold changes are provided. When the details of preprocessing and the quantitation table have been uploaded to the server, all possible performance metrics are calculated and the display is switched to the Evaluation Matrix page (Figure 4.2B).

The Evaluation Matrix page is centred on an interactive performance metrics matrix. Each unique set of preprocessing parameters is referred to as a *method set* and forms one row of the matrix. The top row of the matrix is the row of ideal values for each metric, and has a golden background colour to provide a visual cue that the numbers are for the gold standard. The metrics are grouped by whether they are calculated on within-channel, within-run, or between-runs protein measurements. They are further grouped by whether they measure aspects of bias or variance. Each metric occupies one column of the matrix. To enable users to interactively make comparisons of their interest, the rows and columns of the matrix are selectable by left clicking the method set number or the metric number in the footer of the table. Once a subset of the matrix has been selected, the method sets may be evaluated either numerically or graphically.
Numerical comparison uses the selected metrics of the selected method sets to calculate the distance of each selected method set to the ideal metric value. The best method set has either the smallest sum of absolute distances to the ideal values (weighted voting), or the largest number of metrics closest to the ideal values (unweighted voting). The result text also provides the distance value or the metrics for which the method set was best for. Ties are handled by displaying all method sets which have the best set of metrics.

Two kinds of graphical summaries are available. Firstly, the Plot PCA button makes a PCA plot of the method sets selected, based on the metrics selected. The method sets are projected into two dimensions. This enables the compact summarisation of method set performance similarity. Secondly, the Plot Comparison button causes either boxplots or bar charts to be generated of the selected subset of the matrix, depending on whether there are one or more summarised values, for each value of the x-axis variable. The plots can be organised into multiple rows and columns, by selecting one of the Methodology column names from the Rows By and Columns By drop-down lists. The Colour By list adds a third way to group metrics by in the plot. Therefore, the application allows up to three methodology factors to be visualised simultaneously.
4.3 Web-based Application

Figure 4.2 **Overview of the web-based application for evaluation.** A. The New Processing Upload page allows users to describe various aspects of their data preprocessing and to upload the protein quantities or fold changes as a text file to the server. The web-based application calculates all of the performance metrics and then automatically switches to the Evaluation Matrix page. B. The Evaluation Matrix page displays the matrix of performance metrics and various tools to interact with it. The numbers in the table are summaries, either the mean or standard deviation of the metric, of either all experimental runs or all iTRAQ channels, where applicable. Filtering, sorting, row and column selection are used to subset the matrix to the comparison of interest. Full metric names can be seen by hovering the mouse over the column number. Numerical and graphical evaluation tools are located below the metric matrix.
4.4 Case Studies

The utility of the proposed evaluation metrics and web-based application are illustrated by their application to three analysis scenarios, each examining the performance of preprocessing methods for either relative or absolute quantitation. The first scenario involves the use of a commercial software, ProteinPilot, to calculate protein-level relative quantitation. Secondly, the performance characteristics of custom relative quantitation are evaluated, using peptide summarisation methods and between-sample normalisation methods popular in academic research. Finally, a novel scenario of performing absolute quantitation without use of internal standards is examined.

Between-sample and between-run normalisation are of interest because clustering of samples by iTRAQ channel and, to a lesser extent, run can be seen in PCA plots of the background protein summaries (Figure C-2). Samples measured by the internal reference channel are expected to be located away from the other seven channels, where the proportion of yeast proteins (1 µL:57 µL) is larger than for the other channels (1 µL:128 µL), so the PCA plot is consistent with theoretical expectations.

4.4.1 Case 1: Preprocessing with ProteinPilot

Two key steps of ProteinPilot processing are bias and background correction. The bias correction step aims to make samples more comparable to each other by calculating the ratio to the internal reference channel for all proteins within a sample and representing a sample by the median ratio. By a process not described in the software documentation, scaling factors are calculated to make the median ratio of all samples equivalent. From the design matrix (Table C-2), because each dynamic channel contains the same protein volumes in a different permutation, the theoretical median fold change of each dynamic sample to the reference sample is the same, so this normalisation approach is applicable to the dataset. The following step, background correction, is intended to make fold changes closer to their true values. It affects each protein within a sample differently. In the software documentation, it is stated to be
problematic with datasets where the proteins with the highest expression amount are mostly differentially expressed. This is true for the experimental design used by this case study, where the yeast proteins are present at much lower abundances than any of the spike-in proteins. Only the regression line for relative quantitation is evaluated, since the software does not output the numerator and denominator it uses, only the calculated fold change.

Most distributions of the slope of the regression line have a median close to the ideal value of 1 (Figure 4.3). The slopes for channels 113 and 118 are particularly good, having little variance and being the very close to 1. Channel-specific biases are evident, with the Intercept metric of channels 114 and 115 having intercepts around -2. The biases also adversely affect the Line Difference Sum metric of those two channels. There is noticeably less variability between runs than between channels. The low bias comes at a cost of increased variance. The medians of the Residual Sum metric are increased in comparison to the unnormalised dataset, presented in the following section.
4.4 Case Studies

Figure 4.3 **Performance metrics of regression for relative quantitation using the default processing parameters of ProteinPilot 5.** Horizontal red line represents the ideal metric value. Metrics are grouped either by iTRAQ channel (first row) or run (second row).

4.4.2 Case 2: Custom Relative Quantitation

Motivated by their usage in published proteomics research (Section 4.1.1), the summarisation approaches of peptides to a particular protein evaluated in this case study are:

**Top 1 Peptide:** The peptide with the maximum signal is used.

**Top 3 Peptides:** The peptides with the three highest signals are summed.

**Median Peptide:** The peptide with the median signal is used.

The normalisation methods between samples are:

**Simple Scaling:** This simple method is based on correcting the sample-to-sample variability within each run by calculating scaling factors using only the yeast protein summaries. Every channel has the yeast proteome added at the same quantity (e.g. Table 4.1). Firstly, a representative control quantity value for each sample is calculated. Four options are considered:
Top 1 Yeast Protein: The yeast protein with the highest summary in each sample.

Top 3 Yeast Proteins: The proteins with the three largest summaries in each sample are summed.

Median Yeast Protein: The median of the yeast protein summaries in each sample.

All Yeast Proteins: The sum of summaries of all yeast proteins in each sample.

Next, a scaling factor is calculated by dividing the representative value of the internal reference channel to the representative value of each dynamic channel. Every protein in a dynamic channel is multiplied by the calculated factor for that channel. This procedure is done independently for each run.

Two-stage Scaling: The first step is identical to Simple Scaling. In the second stage, another set of scaling factors to multiply entire runs by is calculated. The four ways of calculating a representative value are used, this time considering all yeast proteins within a run. A scaling factor is calculated by arbitrarily picking one run to be the reference run and then dividing the representative value of its representative value by each other run’s representative value. Lastly, all proteins within a run are multiplied by the calculated factor for that run.

Linear Model: A more complicated approach involves fitting a linear model to the subset of the yeast proteins which were measured in every run of the experiment (Mahoney et al., 2011). Here, a modified version of the published model is used that does not include a peptide effect:

\[ y_{ijk} = \mu + \text{run}_i + \text{channel}_j + \text{runChannel}_{ij} + \text{protein}_k + \epsilon, \]

\( y \) is the log\(_2\)-transformed protein summary value of a protein in run \( i \), measured in channel \( j \) of protein \( k \).

The term \( \mu \) is the intercept term, \( \text{run}_i \) is the effect of the \( i \)\(^{th} \) run \((i \in \{1..7}\)\), \( \text{channel}_j \) is the effect of the \( j \)\(^{th} \) channel \((j \in \{113, 114, 115, 116, 117, 118, 119, 121\})\), the interaction term \( \text{runChannel}_{ij} \) is the
sample-specific effect, $protein_k$ is a factor for each of the yeast proteins detected in all runs, and $\epsilon$ is for the other sources of technical variability. The corrected protein summary value, denoted $z$, is simply corrected by subtracting the unwanted fitted effects from the observed summary value of spike-in protein $k$:

$$z_{ijk} = y_{ijk} - run_i - channel_j - runChannel_{ij}$$

**RUV**: The most complex method for making corrections is a recent extension to the RUV technique (Jacob et al., 2016), which can correct a dataset when the unwanted experimental factors are unknown in advance and neither is the factor of interest (e.g. disease status). Like the two-stage scaling and linear model approaches, this method makes use of control features present in each sample. The model being fitted is:

$$Y = X\beta + W\alpha + \epsilon,$$

where $Y$ is a $m \times n$ matrix of $m$ samples and $n$ feature measurements, $X$ is a $m \times p$ matrix where the $p$ columns contain the factors of interest (e.g. disease subtype), $W$ is a $m \times k$ matrix that contains the $k$ unwanted factors causing noise (e.g. experimental batch), and $\epsilon$ is some random noise, assumed to be normally distributed. This version of RUV differs to previous versions in that the algorithm to estimate the $W\alpha$ does not require classes to be assigned to the samples.

Any systematic bias will be present in all proteins, including those known to be unrelated to actual differences between samples. For this reason, the yeast proteins are key to RUV normalisation. The user has to provide an estimate of the number of independent factors thought to be biasing the measurements, termed the rank parameter, and a second parameter to fine-tune how much of the estimated unwanted variation is removed from the dataset, termed the regularisation parameter.
All of these procedures mainly aim to improve relative quantitation bias, because systematic differences between samples are accounted for. Variance of relative quantitation and some of the summation metrics which use summaries from multiple samples is also ideally reduced, because the variability between samples is reduced. Apart from Simple Scaling, all of these procedures also aim to reduce variability between runs.

Each of the normalisation methods with parameter choices was evaluated independently, to reduce the complexity of comparison between them (Section C-3). For the two scaling methods, Median Yeast summarisation of a sample performed best in the most number of categories (Table C-5A) and will be used as the summary for each sample to calculate a scaling factor with. For RUV, the performance metrics were rather sensitive to the choice of parameters (Table C-5B). The RUV software uses a default of $k$ equal to the number of samples and $\nu$ equal to 0.001 and these values are used in subsequent comparisons.

The Simple Scaling, Linear Model and RUV methods all produce similar biases to fold changes (Figure 4.4). For example, channels 114 and 115 have a median intercept of about -2, and the variability of this across runs is small. Also, channel 117 has an intercept of around 1, with little variation across runs. In contrast, when the intercepts are grouped by run, the distributions are centred close to 0, although the spread is much larger. The unnormalised intercepts are all around -2, due to the different total volume in the internal reference channel to the dynamic channels and the compositional nature of the data. The spread of the intercepts between channels for the unnormalised summary is much less than for any of the normalised summaries, demonstrating that normalising using the yeast protein summaries introduces additional unwanted variation when removing bias.

The slope coefficient is particularly far from ideal for channels 114 and 115, with a median value of about 0.25. When considering the slopes being grouped by runs, all runs are rather consistent with
medians close to 0.5. The Residual Sum’s median is typically around 15, with little variation between groupings. The peptide to protein summary type has no substantial effect on any of the metrics.

The intercept is the only metric affected by the normalisations, so the distribution of intercept metrics of all four approaches is shown. The Slope and Residual Sum are unaffected by between-sample normalisation methods and only the metrics for the unnormalised dataset are shown. Applying a multiplicative scaling factor to a sample causes the points used to fit the regression line to be translated up or down because the fold changes are represented on the logarithmic scale. Without loss of generality, a proof is provided in Section C.4 for the case of two vectors consisting of two measurements each.
4.4 Case Studies

4.4.3 Case 3: Absolute Quantitation

The linear regression of expected to measured fold changes for this novel scenario has an intercept and slope similar to what is theoretically expected (Figure 4.5A). No normalisation methods are applied, because they normalise between samples whereas these metrics use only within-sample protein summaries. The intercepts of the linear fit tend to be slightly positively biased, which suggests that the median protein summary used as the denominator of the fold change is typically lower than expected. As for custom relative quantitation, the slopes are all well below 1, regardless of the run, channel or summary method used. The Median summarisation has the worst median slope estimates for all seven

Figure 4.4 **Performance metrics for relative quantitation.** Horizontal red line represents the ideal metric value.
runs and five out of seven channels. The variability of the metrics is not greatly different between the summarisation varieties, as shown by similar IQRs within each figure panel.

Regarding the summation metrics which sum across different samples, no normalisation method performs consistently better than the others (Figure 4.5B). RUV results in the greatest improvement of the within channel C.V.s, but only for runs four, five, and six. For the other four runs, all methods result in similar C.V.s. Simple Scaling is the best method, but by only a small margin, for the C.V. Median Proteins metric. The C.V. of all of the measurements in a channel across all runs is approximately the same as the C.V. within each run, showing that there is little additional unwanted variability present between runs. Notably, the C.V. increases with run number for Top 1 and Top 3 summarisations, suggesting a relationship between the length of time the protein mixture was stored before analysis with the mass spectrometer and the magnitude of measurement variability.
Figure 4.5 **Performance metrics for absolute quantitation.** Horizontal red line represents the ideal metric value. **A.** The distribution of metric values of a linear regression of protein summaries within a sample divided by the median protein group’s median summary of the sample, grouped either by iTRAQ channel or experimental run. **B.** Summaries for three variance metrics calculated using data from either one run or all runs.
4.5 Discussion and Conclusion

This study has made two main contributions to proteomics. Firstly, a range of performance metrics which allow the systematic evaluation of bias and variance were defined. Secondly, a web-based application for performance evaluation has been made available for the first time to the proteomics research community that allows exploration of a large variety of performance metrics applicable to different research questions. A number of interesting findings resulted from comparisons made with this web-based application.

The web-based application is notable because it allows the upload and comparison of new preprocessing method summaries, unlike other recent proteomics performance evaluation efforts (Mahoney et al., 2011; Tuli et al., 2012; Goeminne, Gevaert, & Clement, 2016). It also calculates a broader range of metrics than existing studies, allowing more detailed comparisons of runs, channels, and proteins to be made. Another advantage is that the results are publicly accessible, which provides unhindered access to scientists to search for the best preprocessing method for a particular analysis scenario. The raw dataset is available from ProteomeXchange and the file input requirements are clearly explained in the application’s documentation, enabling summaries from novel methods to easily be added to the existing metrics table. No software installation or computer programming knowledge is required, providing accessibility to the broader research community.

The application of the web-based application in three case studies resulted in a number of interesting observations. Firstly, there are no major differences between protein summary methods or between-sample normalisation methods. A previous analysis of protein summaries lacked a statistical experimental design and made no conclusions (Matzke et al., 2013), leaving the research community uncertain about which approach is the best. The results of Section 4.4.2 demonstrate that there is little difference for most performance metrics between all three summarisation approaches. Metrics for
Median summarisation tend to be closer to zero than for the other two approaches, suggesting that it is discarding some useful information. Also, complex methods for between-sample normalisation like RUV perform almost identically to the simple scaling approaches, as seen in Figure 4.4. These results suggest that RUV captures the same sources of unwanted variation that a computationally fast to calculate scaling factor does. RUV has only been evaluated on a dataset chosen by the authors who developed the method and QPEP suggests that the performance of the algorithm may not be as good as claimed when independently evaluated. There appears to be no good reason to use RUV for relative quantitation when a much simpler method works equally as well.

A second surprising observation is that ProteinPilot’s background correction works well in a dataset where the majority of detected proteins are differentially expressed. The method is not described in mathematical detail by ProteinPilot’s documentation and is suggested by the documentation to perform poorly when the majority of proteins change in abundance. Moreover, it has never before been independently evaluated. The slopes of the regression between expected and observed log₂-fold changes improved from about one half for all custom normalisation methods tried to nearly one when background correction was used. This improvement in bias did come with an increase in variance, however, which would adversely affect differential protein expression studies in practice, where the increased variance could result in too many false negative findings.

Thirdly, a longstanding rule in omics research that one should not compare measurements between different genes or proteins appears to be unnecessary. The current approach to absolute quantitation is to add stable-isotope labelled standards to every sample and create a linear model for the known concentration of peptide and experimentally-measured amount (Scott, Turko, & Phinney, 2015). This is time-consuming and expensive, because internal standards must be created for every peptide of interest (Villanueva, Carrascal, & Abian, 2014). It is infeasible to take this approach for shotgun proteomics
experiments because of the number of distinct standards which would be necessary, so almost all published work takes the ratio of each protein summary to its summary in the reference channel. Comparing Figure 4.4 to Figure 4.5 suggests that this daunting task may not be necessary. The slopes of the regression between expected and observed log_2-fold changes are essentially the same as for relative quantitation. Moreover, the intercepts are noticeably better for absolute quantitation than for any of the normalisation methods applied in the relative quantitation setting. Remarkably, the distribution of the intercepts were strongly associated with the iTRAQ channel, although the yeast proteins were scaled to have a fold change of 1. This suggests that different iTRAQ labels may have different measurement characteristics. The performance of absolute quantitation without internal standards suggests that reserving one channel for an internal standard to perform relative quantitation is unnecessary and more biological samples could be analysed in less runs of the instrument by not having reference channels in an experimental design.

Finally, there is much less observed variability between instrument runs than there is between iTRAQ channels. This has important implications for experimental design. Most published articles do not use the statistical principles of randomisation and blocking when designing their iTRAQ-labelled experiment (Leong et al., 2012; Xu et al., 2015; Kim et al., 2016). Unless the findings of these projects are validated by an independent technique, such as Western blotting or reverse phase protein arrays, it is unclear whether the differences in protein amount are the result of the biological conditions studied or simply an artefact of the iTRAQ channels inadvertently used for each condition.

In summary, a range of performance metrics and a web-based application have been described, which allow for the comprehensive evaluation of bias and variance in a carefully designed iTRAQ spike-in experiment. A number of interesting observations were found in three case studies, which have potential consequences for existing and future proteomics experiments. In the future, it would be interesting to
compare an LC-MS/MS experiment that has been background corrected by ProteinPilot to an LC-MS/MS/MS, or MS\textsuperscript{3}, experiment of the same samples, which is a laboratory approach to addressing the underestimation of fold changes in iTRAQ experiments (Ting, Rad, Gygi, & Haas, 2011). Which approach results in measurements with smaller variance is also unknown. Another outstanding issue is that ProteinPilot does not have any user options regarding peak quantitation. Methods such as peak area, peak summit height, and different types of peak area integration could be implemented and their evaluation carried out. Additionally, the poorer performance of iTRAQ channels 114 and 115 relative to the other six channels warrants further investigation. It may have only occurred for the reagent batch used by this study or may be systematically affecting all experiments which utilise iTRAQ labels, the majority which do not randomise experimental conditions to channels.
Conclusion

This thesis has demonstrated that each type of omics technology generates data with a plethora of summarisation and normalisation options. This presents the opportunity for the development of novel methods to improve classification or numerical summarisation performance. Each data type has unique characteristics which had to be understood in order to develop a new method or to systematically compare existing methods and select the superior method in each scenario. The three main contributions have been the development of a two-stage algorithm to perform region-finding and region classification for CAGE-seq data, the characterisation of feature selection and prediction by differential distribution for the first time with a novel software framework (ClassifyR), and the development of a large variety of performance metrics and a web-based application that allows anyone to verify the performance of a method and to compare their own methods against methods which are currently widely used by researchers.

The CAGE-seq region finding algorithm is notable for its simplicity and speed, while using an improved background model compared to F-seq, and not returning a lengthy list of regions and forcing the user to manually choose the most representative regions, like Paraclu (Section 2.2.1.1). The previous work on classification of regions was also found to be limited, with only two methods attempting to provide a solution to the problem of classifying regions as TSS or other artefacts. The proposed method outperformed both ENCODE-HMM and Segway (which does not use CAGE-seq data), when evaluated by precision and recall. One surprising finding was that integrating CAGE-seq data with matched RNA-seq data provided little benefit to classification performance.
Prognosis classification is another area of statistical bioinformatics where the accuracy of existing methods leaves room for improvement. Classification has traditionally been focussed on changes in location between classes (Section 3.1), but the present study found that changes in scale, or the combination of changes in location and scale, modelled by a novel differential distribution framework, had equivalent balanced error rates to existing methods, but provided significant advantages in terms of feature selection (Section 3.5). These findings suggest that classifiers built on differential distribution may provide more informative and more stable features in terms of biological relevance. The median cross-validated error rates of around 25% are better than those based only on clinical data, which is the current standard in clinical care, but are still higher than what clinicians desire. Future work could involve vertical data integration, when more complementary datasets (e.g. metabolomics, proteomics) of the same patients become available as data acquisition becomes simpler and cheaper with technological advancements. Biological systems are complex, and more of the variability between samples and their prognosis could potentially be explained with more orthogonal measurements of the same patient sample.

Much like the comprehensive evaluations for CAGE-seq and gene expression in earlier chapters, novel comparisons have provided evidence for performance improvements in proteomics data analysis (Chapter 4). No other evaluation has developed as many different performance metrics to characterise bias and variance or used an experimental design where the factors were exhaustively varied. Moreover, the novel web-based application is a substantial improvement over the numerous published research studies that do not provide reproducible code to replicate their findings or the possibility to compare future findings with theirs in a streamlined way. This framework enabled the discovery that commercial software performs quite well in comparison to academic approaches (Section 4.4.1) and that complex statistical approaches have no benefit over the simplest of scaling approaches (Section 4.4.2). It was also
surprising that performing absolute quantitation without internal standards performed just as well as relative quantitation (Section 4.4.3). This casts doubt on a widespread rule in omics research about not comparing measurements of different features directly and creates opportunities for new types of within-sample analysis.

The statistical approaches proposed in the preceding chapters have provided new ways to summarise, normalise and evaluate omics datasets that measure various stages of the central dogma of molecular biology, from transcription (Chapter 2) to proteins (Chapter 4). Their implementations have been made available through online software repositories or web applications, to enable others to use them on their own datasets and be able to improve upon them. Novel findings which improve classification or numeric summarisation have been presented across all three topics and challenges which remain unresolved have been outlined.
A.1 Region Finding Visual Evaluation

It has been shown in Section 2 that almost all of the regions found by the proposed method are found by Paraclu and F-seq. This is because the proposed method finds a subset of regions using its null model. A representative genomic region is illustrated in Figure A-3, which shows the histogram of read counts, along with a track of regions for each method. Paraclu returns a large number of overlapping results, leaving it to the scientist to decide which region they think is the most representative of the data. This is infeasible to manually do for all of the regions found by Paraclu. Paraclu also includes large regions with no signal in the raw data, such as the left half of the figure. F-Seq identifies the key region in the centre of the figure, but also identifies many regions that are supported by only one read. These regions are not found by the proposed algorithm, perhaps because of the effective determination of the null distribution. These issues are observed for all six cell lines.
A.2 Feature Selection

Features were chosen based on having an absolute correlation of above 0.5 to the first principal component of the PCA. The 2D PCA plot using the six scalar features and another plot using the 512 4-mer sequence counts are shown below. The scalar features used here are kurtosis, read density, TFBS, DNAse I hypersensitivity, H3K4me3 hypersensitivity, and mammalian conservation.
Figure A-2 Principal components analysis of the scalar features surrounding the CAGE read regions. The top panel shows the TSS category of regions and the bottom panel shows those regions which are labelled as Not TSS in the truth set. Dots represent the 100 observations in the lowest density regions.

Figure A-3 Principal components analysis of the 4-mers surrounding the CAGE read regions. The analysis was performed on the correlation matrix of the 512 4-mers. The top panel shows the TSS category of regions and the bottom panel shows those regions which are labelled as Not TSS in the truth set. Dots represent the 100 observations in the lowest density regions.
A.3 Parameter Tuning

Linear SVMs have a single tuning parameter; the cost parameter. It controls how much incorrectly classified samples are penalised during the fitting of the separating hyperplane parameters. A grid of cost values was chosen and the precision and recall metrics are shown to be stable for a large range of the cost parameter.

![Image showing Precision and recall for three feature scenarios](image)

**Figure A-4.** Precision and recall for three feature scenarios. Precision and recall are calculated at each cost parameter value based on a LOOCV scheme. Blue lines are precision, Red lines are recall. Horizontal bars or dots represent the minimum and maximum value of all cell lines. Points on the line are averages across all six cell lines for the two leftmost panels and two cell lines for the rightmost panel.
A.4 Classification Performance

The precision and recall metrics for the six cell lines are shown for three dataset scenarios. Minor improvements in recall are made by incorporating the matched external feature.

Table A-1 **Precision and recall for three dataset scenarios.** Internal scenario uses selected 4-mers. Internal and Pooled External scenario uses selected 4-mers and Maximum H3K4me3, Maximum TFBS, Maximum Sensitivity, and Average Conservation features. Internal and Matched External scenario uses selected 4-mers and Poisson test of flanking RNA-seq counts, matched to each cell line. Grey cells indicate evaluations for which data is not available.

<table>
<thead>
<tr>
<th>Cell Line</th>
<th>Precision</th>
<th>Recall</th>
<th>Precision</th>
<th>Recall</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>0.77</td>
<td>0.81</td>
<td>0.76</td>
<td>0.81</td>
<td>0.75</td>
<td>0.84</td>
</tr>
<tr>
<td>H1-ESC</td>
<td>0.61</td>
<td>0.81</td>
<td>0.63</td>
<td>0.80</td>
<td>0.74</td>
<td>0.84</td>
</tr>
<tr>
<td>HeLa-S3</td>
<td>0.76</td>
<td>0.87</td>
<td>0.79</td>
<td>0.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HepG2</td>
<td>0.69</td>
<td>0.79</td>
<td>0.76</td>
<td>0.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HUVEC</td>
<td>0.81</td>
<td>0.85</td>
<td>0.81</td>
<td>0.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K562</td>
<td>0.71</td>
<td>0.88</td>
<td>0.82</td>
<td>0.81</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A.5 Prostate Cancer Case Study

Plots of the counts in the regions identified for the gene with the best differential expression and TSS switching statistics are shown.
Figure A-5. **CAGE count summaries for the genes with the best differential expression and TSS switching statistics.** The top two rows of data show normalised count values per genomic position in each of the two experimental conditions. The third row of data shows the structure of known genes. The bottom row shows the regions which have been classified as TSS regions.
B.1 ClassifyR Runtime

Two-class classification using ClassifyR was done on the melanoma dataset introduced in Section 3.5. DLDA was used for DE classification and FLDA was used for DV classification. DE feature selection used the limma moderated t-statistic. DV feature selection used the Levene statistic, which involves repeatedly calculating medians for every feature, which explains the much slower timings of DV classification than for DE. The runtimes approximately halves as the processors are doubled from one to eight processors, but the improvement is much diminished for 16 processors.
Table B-1 **Classification runtime using ClassifyR.** DE and DV classification were evaluated in two cross-validation modes, using between 1 and 16 processing cores.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Cross-validation</th>
<th>Cores</th>
<th>Time (minutes:seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>100 resamples, 5 folds</td>
<td>1</td>
<td>14:54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>7:51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>4:21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>2:30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>2:03</td>
</tr>
<tr>
<td></td>
<td>Leave-2-out</td>
<td>1</td>
<td>32:42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>17:32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>9:14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>5:34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>4:18</td>
</tr>
<tr>
<td>Variability</td>
<td>100 resamples, 5 folds</td>
<td>1</td>
<td>669:55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>357:02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>207:46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>125:24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>91:31</td>
</tr>
<tr>
<td></td>
<td>Leave-2-out</td>
<td>1</td>
<td>1446:41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>782:13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>408:50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>262:55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>210:00</td>
</tr>
</tbody>
</table>

**B.2 ClassifyR Parameters**

Each of the four stages of classification require some specific parameters. All parameters for a particular stage are stored in a class. These parameters are summarised in the table above.
Table B-2 **Parameter classes and their required variables.** Classes that store parameters for each stage of classification and the meanings of each parameter. Any other variables can be stored and used, as long as the function which performs the particular stage knows how to use them. In addition, all classes use a parameter `intermediate`, which is a character vector of variable names generated internally by `runTest` that will be used in the classification stage the class represents.

<table>
<thead>
<tr>
<th>Class</th>
<th>Mandatory Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransformParams</td>
<td>transform</td>
<td>Any function that transforms a matrix of numbers and returns the transformed matrix of numbers.</td>
</tr>
<tr>
<td>SelectParams</td>
<td>featureSelection</td>
<td>Any function that takes a matrix of numbers and returns a vector of indices, representing the features that were chosen.</td>
</tr>
<tr>
<td></td>
<td>subsetExpressionData</td>
<td>If TRUE, subsets the expression values. If FALSE, only returns the vector of indices of chosen features.</td>
</tr>
<tr>
<td>TrainParams</td>
<td>classifier</td>
<td>Any function that builds a classifier.</td>
</tr>
<tr>
<td></td>
<td>transposeExpression</td>
<td>Some existing classifiers in R require the features to be columns. If set to TRUE, the data matrix is transposed before being dispatched to classifier.</td>
</tr>
<tr>
<td></td>
<td>doesTests</td>
<td>If set to TRUE, notifies the framework that classifier should also be used as the prediction function. Some functions do both training and testing.</td>
</tr>
<tr>
<td>PredictParams</td>
<td>predictor</td>
<td>A function that uses the model built in the training stage to make predictions on test data.</td>
</tr>
<tr>
<td></td>
<td>getClasses</td>
<td>A function that takes as input the output of predictor, or classifier if doesTests is TRUE, and extracts a vector of predicted classes.</td>
</tr>
</tbody>
</table>

Each of the functions used in the four stages also has rules about arguments it must conform to.

**Transform Function:** The first argument must be a matrix of data. It returns a transformed matrix of the same dimension as the input matrix.

**Selection Function:** The first argument must be a matrix of data. It returns a `SelectResult` object.

**Training Function:** The first argument must be a `matrix`. This is because most other R classifiers on CRAN take matrices. The second argument must be a factor of classes of the training samples. The
function returns a trained model. If \texttt{doesTests} is TRUE because the function also makes predictions using the test set samples, the third argument must be a matrix of test set values.

**Prediction Function**: The first argument must be a trained model that was generated by the training step. The second argument must be a matrix of test data. It returns an object containing predictions.

Apart from these mandatory arguments, the functions may have any other arguments they wish to.

An example of a function which does not conform to these rules is \( k \) nearest neighbours. It has the classes of the training set as the third argument, whereas ClassifyR expects it to be the second argument and the matrix of test values to be the third. However, a simple wrapper function enables it to work with the framework, simply by referring to the wrapper function, instead of the original function.

```r
knnWrapper <- function(exprTrain, classesTrain, exprTest, verbose, ...) {
  knn(train = exprTrain, test = exprTest, cl = classesTrain, ...)
}

training <- TrainParams(knnWrapper,
    transposeExpression = TRUE, doesTests = TRUE)
result <- runTests(aDataset, "Dataset Name", "Classifier Name",
    params = list(SelectionParams(), training,
                  PredictParams()
        )
    )
```

**B.3 Properties of Biological Datasets**

In the biological study, two classes were created by using a cut-off on the number of years since the patient was diagnosed and whether they are alive or dead (Section 3.5). This excludes a number of samples for the evaluation. The survival times of all samples in the datasets are alternatively presented as two groups, based on whether they are alive or dead. The lung cancer dataset has very few patients who were followed for more than three years.
B.4 Most Stable Feature

For the three cancer datasets, the gene which was selected the greatest number of times in cross-validation was found. The plot of the class distributions is shown below. For the DV and DD rows, the variety of selection found to perform the best in the simulation study (Section 3.4) is used. That is, Bartlett’s test is used for DV and the DMD statistic is used for DD. As expected, Bartlett’s test chooses genes which have a small number of outliers in one class.

Figure B-1 **Distribution of survival times for the three cancer datasets used for classification evaluation.** Densities are estimated with the default Gaussian kernel in R.
Figure B-2 Most frequently chosen genes’ expression profiles in cross-validation for three datasets and three feature types. The RefSeq symbol of the gene the densities are plotted for is shown above each density plot. DV selection is made by ranking of the Bartlett’s test statistic and DD selection is made by the proposed DMD statistic.
C

Supplementary Material For Chapter 4

C.1 Full Specification of Latin Squares Experimental Design

The experiment consists of 21 proteins of known identity and the yeast proteome which has an unknown composition. The assigned group, name, database ID, length, and source organism are specified below.

Table C-1 The identities of purchased proteins and their associated information. Group denotes the allocated group label of each protein, so that a variety of sizes are in each group.

<table>
<thead>
<tr>
<th>Group</th>
<th>Protein Name</th>
<th>SwissProt ID</th>
<th>Amino Acids</th>
<th>Organism</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Thioredoxin</td>
<td>P10599</td>
<td>105</td>
<td>Human</td>
</tr>
<tr>
<td>A</td>
<td>Glutathione S-Transferase</td>
<td>P08515</td>
<td>222</td>
<td><em>Schistosoma japonicum</em></td>
</tr>
<tr>
<td>A</td>
<td>Antithrombin III</td>
<td>P41361</td>
<td>465</td>
<td>Cow</td>
</tr>
<tr>
<td>B</td>
<td>Polyubiquitin B</td>
<td>P0CG47</td>
<td>76</td>
<td>Human</td>
</tr>
<tr>
<td>B</td>
<td>DNase I</td>
<td>P00639</td>
<td>282</td>
<td>Cow</td>
</tr>
<tr>
<td>B</td>
<td>Catalase</td>
<td>P00432</td>
<td>527</td>
<td>Cow</td>
</tr>
<tr>
<td>C</td>
<td>Alpha-lactalbumin</td>
<td>P00711</td>
<td>142</td>
<td>Cow</td>
</tr>
<tr>
<td>C</td>
<td>Quinone Oxidoreductase 2</td>
<td>P16083</td>
<td>231</td>
<td>Human</td>
</tr>
<tr>
<td>C</td>
<td>Gelsolin</td>
<td>P06396</td>
<td>782</td>
<td>Human</td>
</tr>
<tr>
<td>D</td>
<td>Beta-2-microglobulin</td>
<td>P61769</td>
<td>119</td>
<td>Human</td>
</tr>
<tr>
<td>D</td>
<td>Ubiquitin-conjugating enzyme E2 E1</td>
<td>P51965</td>
<td>193</td>
<td>Human</td>
</tr>
<tr>
<td>D</td>
<td>Amylase</td>
<td>P06278</td>
<td>512</td>
<td><em>Bacillus licheniformis</em></td>
</tr>
<tr>
<td>E</td>
<td>Cytochrome c</td>
<td>P00004</td>
<td>105</td>
<td>Horse</td>
</tr>
<tr>
<td>E</td>
<td>C Reactive Protein</td>
<td>P02741</td>
<td>224</td>
<td>Human</td>
</tr>
<tr>
<td>E</td>
<td>Serum Albumin</td>
<td>P02768</td>
<td>609</td>
<td>Human</td>
</tr>
<tr>
<td>F</td>
<td>Hemoglobin subunit alpha</td>
<td>P69905</td>
<td>142</td>
<td>Human</td>
</tr>
<tr>
<td>F</td>
<td>Peroxiredoxin 1</td>
<td>Q06830</td>
<td>199</td>
<td>Human</td>
</tr>
<tr>
<td>F</td>
<td>Sero transferrin</td>
<td>P02787</td>
<td>698</td>
<td>Human</td>
</tr>
<tr>
<td>G</td>
<td>Cytochrome b5</td>
<td>P00167</td>
<td>134</td>
<td>Human</td>
</tr>
<tr>
<td>G</td>
<td>Carbonic anhydrase 2</td>
<td>P00918</td>
<td>260</td>
<td>Human</td>
</tr>
<tr>
<td>G</td>
<td>Lactotransferrin</td>
<td>P02788</td>
<td>710</td>
<td>Human</td>
</tr>
</tbody>
</table>
The experiment is composed of seven runs of varying volume for each protein in a particular iTRAQ channel. Across the seven runs in the seven dynamic channels, each spike-in protein is present at one of the seven volumes exactly once.

Table C-2 **Experimental design of all seven runs for the purchased proteins.** The volumes are in units of microlitres.

<table>
<thead>
<tr>
<th>Run</th>
<th>Channel</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>113</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>114</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>115</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>116</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>117</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>118</td>
<td>32</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>119</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>121</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>113</td>
<td>2</td>
<td>16</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>8</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>114</td>
<td>16</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>8</td>
<td>32</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>115</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>8</td>
<td>32</td>
<td>2</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>116</td>
<td>64</td>
<td>4</td>
<td>8</td>
<td>32</td>
<td>2</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>117</td>
<td>4</td>
<td>8</td>
<td>32</td>
<td>2</td>
<td>16</td>
<td>1</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>118</td>
<td>8</td>
<td>32</td>
<td>2</td>
<td>16</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>119</td>
<td>32</td>
<td>2</td>
<td>16</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>121</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>113</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>32</td>
<td>64</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>114</td>
<td>8</td>
<td>2</td>
<td>32</td>
<td>64</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>115</td>
<td>2</td>
<td>32</td>
<td>64</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>116</td>
<td>32</td>
<td>64</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>117</td>
<td>64</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>118</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>32</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>119</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>32</td>
<td>64</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>121</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
# C.1 Full Specification of Latin Squares Experimental Design

<table>
<thead>
<tr>
<th></th>
<th>113</th>
<th>114</th>
<th>115</th>
<th>116</th>
<th>117</th>
<th>118</th>
<th>119</th>
<th>121</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>64</td>
<td>16</td>
<td>4</td>
<td>32</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>16</td>
<td>4</td>
<td>32</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>4</td>
<td>32</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>32</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>64</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>64</td>
<td>16</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>64</td>
<td>16</td>
<td>4</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8</td>
<td>64</td>
<td>16</td>
<td>4</td>
<td>32</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>113</th>
<th>114</th>
<th>115</th>
<th>116</th>
<th>117</th>
<th>118</th>
<th>119</th>
<th>121</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>16</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>64</td>
<td>4</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>4</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>113</th>
<th>114</th>
<th>115</th>
<th>116</th>
<th>117</th>
<th>118</th>
<th>119</th>
<th>121</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>32</td>
<td>4</td>
<td>64</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>64</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>16</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>16</td>
<td>32</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>16</td>
<td>32</td>
<td>4</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2</td>
<td>16</td>
<td>32</td>
<td>4</td>
<td>64</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>16</td>
<td>32</td>
<td>4</td>
<td>64</td>
<td>1</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>32</td>
<td>4</td>
<td>64</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>113</th>
<th>114</th>
<th>115</th>
<th>116</th>
<th>117</th>
<th>118</th>
<th>119</th>
<th>121</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>64</td>
<td>1</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>64</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>64</td>
<td>1</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>64</td>
<td>1</td>
<td>32</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>8</td>
<td>64</td>
<td>1</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>64</td>
<td>1</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>
C.2 Exploratory Analysis of Latin Squares Dataset Characteristics

In Section 4.4 it was stated that the missing peptide percentage between runs is low. The full table of percentages is presented below. For each run, the peptides found were compared to each of the six other runs. The percentage of peptides found in each other run was calculated by matching the peptide sequence, allowing for matches between two sequences if one sequence is entirely contained within the other. This commonly occurs when a position in a protein that could be cut by trypsin was not. For example, in Run 1 the peptide DSSLCK is detected, but in Run 2 only the peptide KDSSLCK is. It would be inappropriate to consider KDSSLCK as a missing value in Run 1 and DSSLCK as another missing value in Run 2. The simple explanation is that, in Run 2, the cleavage site after the K in the first position was missed by the enzyme. The presence of co-occurring peptides between pairs of runs was examined. The percentage of matched peptides between any two runs is high.

Table C-3 Percentage of peptides in a particular run detected in another run. The percentage of peptides in each run that overlap with each other run is displayed.

<table>
<thead>
<tr>
<th>Peptide Source Run</th>
<th>Peptides Matched In Run (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>89</td>
</tr>
<tr>
<td>2</td>
<td>93</td>
</tr>
<tr>
<td>3</td>
<td>92</td>
</tr>
<tr>
<td>4</td>
<td>89</td>
</tr>
<tr>
<td>5</td>
<td>91</td>
</tr>
<tr>
<td>6</td>
<td>94</td>
</tr>
<tr>
<td>7</td>
<td>90</td>
</tr>
</tbody>
</table>

The number of proteins identified by at least one or at least three peptides was summarised for each of the seven experimental runs. The number of spike-in proteins is 21. The expected number of background proteins is unknown, but recent studies have identified more than 2000 proteins in a standard sample of yeast (Jung et al., 2015).
Table C-4. **Number of proteins identified in each run.** The number before the slash is for protein identification by at least one peptide. The number after the slash is the number of proteins with at least three matched peptides.

<table>
<thead>
<tr>
<th>Run</th>
<th>Background Proteins</th>
<th>Spike-in Proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18/11</td>
<td>20/20</td>
</tr>
<tr>
<td>2</td>
<td>16/8</td>
<td>20/20</td>
</tr>
<tr>
<td>3</td>
<td>17/8</td>
<td>20/20</td>
</tr>
<tr>
<td>4</td>
<td>17/9</td>
<td>21/20</td>
</tr>
<tr>
<td>5</td>
<td>13/10</td>
<td>20/20</td>
</tr>
<tr>
<td>6</td>
<td>14/9</td>
<td>20/20</td>
</tr>
<tr>
<td>7</td>
<td>18/11</td>
<td>21/21</td>
</tr>
</tbody>
</table>

In the explanation of performance metrics in Section 4.2, it was stated that the number of peptides detected for a protein increases as the length of the protein increases. The plot below demonstrates such a trend using the replicated Latin squares dataset.

**Figure C-1 Relationship between length of spike-in protein and number of peptides detected in experiment.** The number of peptides associated to a spike-in protein based on matching by ProteinPilot’s Paragon algorithm is shown for all seven runs.
Section 4.4 claimed that there is some clustering evident in a principal components plot of the background proteins. The figure below illustrates those patterns. The plots in the left column are coloured by run. In the right column, the plots colour by the iTRAQ channel.

![Principal Components Analysis of Yeast Proteins](image)

Figure C-2 **Principal components analysis of background proteins.** Seven yeast proteins were quantified in all seven runs. For each sample, the dimensionality has been reduced to two dimensions. Each row of plots contains yeast protein measurements summarised from peptides in one of the three summary methods used in this study. The first column has samples coloured by experimental run and the second column has samples coloured by iTRAQ channel.
C.3 Parameter Selection For Methods Comparisons

Each of the between-sample normalisation methods has a choice of one or more parameters. Results are shown for the Top 3 protein summary, although the results are qualitatively the same for the other two summarisation options. For Simple Scaling and Two-stage Scaling, the yeast protein summary is varied. The maximum, sum of top 3, the median, and sum of all yeast proteins were tried. The metrics calculated on measurements within a sample are unchanged and not shown. Also, the metrics which sum across all samples are unaffected and not shown (but may be conveniently viewed with the web-based application). The Median summary yields the largest number of metrics closest to their ideal values.

RUV’s performance is not robust to the value of the regularisation parameter chosen. The dependence was evaluated by choosing one of four ranks \( k \in \{2, 3, 4, 5\} \) and one of five regularisation values \( \nu \in 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2 \). All combinations of the two parameters were evaluated. Most performance metrics change noticeably for different values of the regularisation parameter but not the rank parameter. Larger values of the regularisation parameter have adverse effects on relative quantitation.

C.4 Scaling Factor Unchanged Slope Proof

Let \( x_1 \) and \( x_2 \) be two expected fold changes and \( y_1 \) and \( y_2 \) be experimental measurements paired with the \( x \) measurements. Let \( r_1 \) and \( r_2 \) be two measurements from a common reference sample used as denominators of the fold changes. The slope of the line between the observed and expected log-fold changes on the Cartesian plane, denoted by \( S \), is:

\[
S = \frac{\log y_2 - \log y_1}{\log x_2 - \log x_1} - \frac{\log r_2 - \log r_1}{\log x_2 - \log x_1}
\]

and the intercept, denoted by \( I \), is:
Now, consider the situation where the \( y \) measurements are multiplied by a constant, \( c \). The slope is

\[
S_{\text{scaled}} = \frac{\log \frac{cy_2}{r_2} - \log \frac{cy_1}{r_1}}{\log x_2 - \log x_1} = \frac{\log c + \log \frac{y_2}{r_2} - \log c + \log \frac{y_1}{r_1}}{\log x_2 - \log x_1} = \frac{\log \frac{y_2}{r_2} - \log \frac{y_1}{r_1}}{\log x_2 - \log x_1} = S
\]

and the intercept, after scaling is:

\[
I_{\text{scaled}} = \frac{\log \frac{cy_1}{r_1}}{S \log x_1} = \frac{c \log \frac{y_1}{r_1}}{S \log x_1} = cl
\]

Therefore, the slope of the line between the points does not change, but the intercept does by a factor of \( c \).
Table C-5 **Performance metrics after normalisation of Top 3 peptide summaries of proteins.** The best value for a parameter combination is highlighted as a white metric value with a black background, if the difference between the lowest and highest value is at least 25% of the highest value. **A** For Simple Scaling and Two-stage Scaling considering four representations of the yeast proteins. **B** For RUV. $k$ is the rank parameter and $\nu$ is the regularisation parameter. Number shown in a cell is the mean of metrics, where there is more than one metric calculated.

### A

<table>
<thead>
<tr>
<th>Quantitation</th>
<th>Scaling</th>
<th>Metric</th>
<th>Top 1</th>
<th>Top 3</th>
<th>Median</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>Simple</td>
<td>C.V. Median Proteins</td>
<td>0.171</td>
<td>0.145</td>
<td>0.119</td>
<td>0.136</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C.V. Channel Sums (within-run sums)</td>
<td>0.776</td>
<td>0.850</td>
<td>0.856</td>
<td>0.814</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C.V. Channel Sums (between-run sums)</td>
<td>0.529</td>
<td>0.696</td>
<td>0.716</td>
<td>0.671</td>
</tr>
<tr>
<td></td>
<td>Two-stage</td>
<td>C.V. Median Proteins</td>
<td><strong>0.200</strong></td>
<td><strong>0.200</strong></td>
<td>0.318</td>
<td>0.222</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C.V. Channel Sums (between-run sums)</td>
<td>0.537</td>
<td>0.711</td>
<td>0.710</td>
<td>0.673</td>
</tr>
<tr>
<td>Relative</td>
<td>Simple or Two-stage</td>
<td>Intercept</td>
<td>-1.190</td>
<td>-0.714</td>
<td><strong>-0.422</strong></td>
<td><strong>-0.561</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line Difference Sum</td>
<td>29.4</td>
<td>26.0</td>
<td>24.9</td>
<td>25.1</td>
</tr>
</tbody>
</table>
## C.4 Scaling Factor Unchanged Slope Proof

### B

<table>
<thead>
<tr>
<th>Quantitation</th>
<th>Metric</th>
<th>$k$</th>
<th>$10^{-2}$</th>
<th>$10^{-1}$</th>
<th>$10^{0}$</th>
<th>$10^{1}$</th>
<th>$10^{2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>C.V. Median</td>
<td>2</td>
<td>0.288</td>
<td>0.261</td>
<td>0.216</td>
<td>0.165</td>
<td>0.137</td>
</tr>
<tr>
<td>Proteins</td>
<td></td>
<td>3</td>
<td>0.294</td>
<td>0.260</td>
<td>0.216</td>
<td>0.165</td>
<td>0.137</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.285</td>
<td>0.258</td>
<td>0.216</td>
<td>0.165</td>
<td>0.137</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.283</td>
<td>0.258</td>
<td>0.216</td>
<td>0.165</td>
<td>0.137</td>
</tr>
<tr>
<td>C.V. Channel</td>
<td>Sums (within-runs sums)</td>
<td>2</td>
<td>0.761</td>
<td>0.704</td>
<td>0.511</td>
<td>0.725</td>
<td>0.798</td>
</tr>
<tr>
<td>Sums</td>
<td></td>
<td>3</td>
<td>0.739</td>
<td>0.702</td>
<td>0.511</td>
<td>0.725</td>
<td>0.798</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.727</td>
<td>0.700</td>
<td>0.510</td>
<td>0.725</td>
<td>0.798</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.726</td>
<td>0.700</td>
<td>0.510</td>
<td>0.725</td>
<td>0.798</td>
</tr>
<tr>
<td>C.V. Channel</td>
<td>Sums (between-runs sums)</td>
<td>2</td>
<td>0.732</td>
<td>0.674</td>
<td>0.366</td>
<td>0.238</td>
<td>0.250</td>
</tr>
<tr>
<td>Sums</td>
<td></td>
<td>3</td>
<td>0.708</td>
<td>0.671</td>
<td>0.366</td>
<td>0.238</td>
<td>0.250</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.694</td>
<td>0.670</td>
<td>0.366</td>
<td>0.238</td>
<td>0.250</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.693</td>
<td>0.670</td>
<td>0.366</td>
<td>0.238</td>
<td>0.250</td>
</tr>
<tr>
<td>Relative</td>
<td>Intercept</td>
<td>2</td>
<td>-0.057</td>
<td>-0.237</td>
<td>-1.09</td>
<td>-1.93</td>
<td>-2.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>-0.073</td>
<td>-0.238</td>
<td>-1.09</td>
<td>-1.93</td>
<td>-2.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>-0.082</td>
<td>-0.240</td>
<td>-1.09</td>
<td>-1.93</td>
<td>-2.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>-0.074</td>
<td>-0.239</td>
<td>-1.09</td>
<td>-1.93</td>
<td>-2.10</td>
</tr>
<tr>
<td></td>
<td>Line Difference Sum</td>
<td>2</td>
<td>23.9</td>
<td>23.6</td>
<td>27.3</td>
<td>39.6</td>
<td>42.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>23.9</td>
<td>23.6</td>
<td>27.3</td>
<td>39.6</td>
<td>42.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>23.7</td>
<td>23.6</td>
<td>27.3</td>
<td>39.6</td>
<td>42.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>23.7</td>
<td>23.6</td>
<td>27.3</td>
<td>39.6</td>
<td>42.6</td>
</tr>
</tbody>
</table>
References


References


Dauer Stage to L4 Stage in Caenorhabditis elegans. *Journal of Proteome Research, 15*(2), 531–539. https://doi.org/10.1021/acs.jproteome.5b00884


References


(e1071), TU Wien (Version 1.6-4). Retrieved from http://cran.r-project.org/web/packages/e1071/index.html

https://doi.org/10.1128/MCB.01325-08


https://doi.org/10.1186/s13059-014-0465-4

https://doi.org/10.1101/gr.126201.111


Spectrometry. *Analytical Chemistry*, 87(8), 4429–4435. https://doi.org/10.1021/acs.analchem.5b00331


2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 65–67).
https://doi.org/10.1109/BIBM.2013.6732765


