Contents

List of Figures

List of Tables

Chapter 2

Methods and Resources

In this chapter, I will outline the various existing resources and methods utilised throughout this project. This includes public data repositories, stable and development releases of software packages (mostly for the R programming environment), and implementation of bioinformatics methods and statistical concepts with Shell or R scripts developed for this purpose. Methods and packages developed specifically for this project will be covered in more detail along with preliminary data to demonstrate and support their use in chapter 3.

2.1 Bioinformatics Resources for Genomics Research

2.1.1 Public Data and Software Packages

Various bioinformatics resources, such as databases and methods, have become integral parts of genetics and genomics research. Reference genomes, genotyped variants, gene expression, and epigenetics profiles are among the most commonly used resources. Gene expression data in particular is widely available from many microarray and RNA-Seq studies, from repositories such as Gene Expression Omnibus (GEO) (Clough and Barrett, 2016), caArray (Heiskanen *et al.*, 2014), and ArrayExpress (Rustici *et al.*, 2013). Such profiles are excellent resources to examine the changes of gene expression occurring in cancers and the variation between samples. These microarray initiatives have set a precedent for data sharing, data mining, and the wider benefits of publicly available data for enabling the scientific community to further utilise the data rather than a single research group or consortium (Rung and Brazma, 2013). The practice of integrating findings from publicly available genomics data with the research questions

and experimental results of individual research groups has carried over into RNA-Seq datasets including the large-scale cancer genomics projects (Zhang *et al.*, 2011). This thesis is one such example of an investigation enabled by this wider movement and tools developed in various disciplines to generate, process, and disseminate genomic-scale data.

Along with databases, it is also becoming common practice for bioinformatics researchers to release their code as open-source or provide a software package to enable replication of the findings or further applications of the methods (Stajich and Lapp, 2006). This is part of a wider movement in software and data analysis with many tools to facilitate such work being released for use in Linux or the R programming environment (R Core Team, 2016). In addition to the R packages hosted on CRAN (CRAN, 2017), the Bioconductor repositories (Gentleman *et al.*, 2004) also contain many packages specifically for applications in bioinformatics, and the GitHub site hosts many packages in various stages of development and early release. Packages from these various sources have been used throughout this project and cited where-ever possible. Several R packages have been developed during this thesis project and either publicly released on GitHub or prepared to accompany a publication.

2.1.1.1 Cancer Genome Atlas Data

Molecular profile data from normal and tumour samples was downloaded from publicly available sources, using the TCGA (TCGA, 2012) and the International Cancer Genome Consortium (ICGC) web portals (Zhang *et al.*, 2011). These include gene expression (RNA-Seq), somatic mutations, and anonymous clinical data. These versions downloaded were on the 6$^{\text{th}}$ of August 2015 (Release 19) and the 2$^{\text{nd}}$ of May 2016 (Release 20) for breast and stomach cancer respectively via the ICGC data portal (`https://dcc.icgc.org/`).

Performing a genomic alignment in remains a challenge in bioinformatics and methods to do so may yet be improved (Chen and Tompa, 2010). However, the statistical and biological aspects of bioinformatics are the focus of this thesis, comparing alignment methods is outside the scope of these investigations. The TCGA project (TCGA, 2012) used widely adopted tools: "Bowtie" for alignment (Langmead *et al.*, 2009), "mapslice" to detect splice sites (Wang *et al.*, 2010), and the Reads Per Kilobase per Million mapped reads (RPKM) approach to qualify reads per transcript as a measure of gene expression (Mortazavi *et al.*, 2008). These are widely acceptable tools for processing RNA-Seq data which were used to produce the raw counts of mapped reads

(tier 1) and normalised expression data (tier 3) publicly available from TCGA.

Raw count and RSEM normalised TCGA expression data from Illumina RNA-Seq protocols were available from 1,177 breast samples (113 normal, 1,057 primary tumour, and 7 metastases) for 20,501 genes. TCGA breast somatic mutation data for 981 samples (976 primary tumours and 5 metastases) across 25,836 genes were available including 969 samples (964 primary tumours and 5 metastases) with corresponding RNA-Seq expression data and 19,166 genes mapped from Ensembl identifiers to gene symbols, of which 16,156 had corresponding gene expression information. Unless otherwise stated, the raw counts were used for further processing rather than the RSEM normalised data (provided by TCGA tier 3). Normalised protein expression was used (as provided by TCGA tier 3), generated from reverse phase protein arrays (RPPA) for 142 antibodies targeting 115 genes for 298 TCGA breast samples.

Raw count TCGA expression data (TCGA tier 1) from Illumina RNA-Seq was also available for 450 stomach samples (35 normal, 415 primary tumour) for 20,501 genes. TCGA stomach mutation data was also available for 289 samples across 25807 genes, corresponding to 19436 genes with expression data. Normalised protein expression (RPPA) data was also sourced (from TCGA tier 3) for 201 antibodies targeting 158 genes for 357 stomach samples.

Cell line data was downloaded from the Cancer Cell Line Encyclopaedia (CCLE) on the 7$^{\text{th}}$ of November 2014 (Barretina *et al.*, 2012; CCLE, 2014). This includes expression data for 1037 cell lines across 19544 genes (last updated on the 18$^{\text{th}}$ of October 2012), DNA copy number, somatic mutation, drug response, and sample information. Samples include 59 breast cell lines and 38 stomach cell lines.

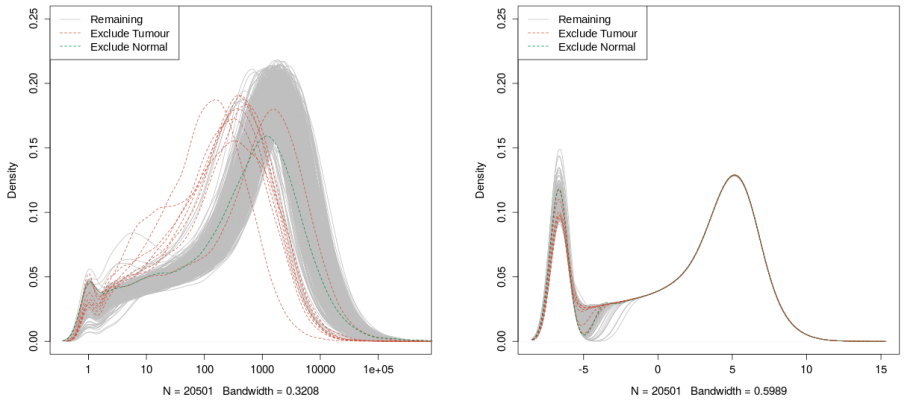
2.1.1.2 Reactome and Annotation Data

Unless otherwise specified, pathway analysis was performed for human pathway annotation from the Reactome database (version 52) with pathway gene sets derived from the `reactome.db` R package. Entrez identifiers were mapped to gene symbols or aliases to match to TCGA expression and mutation data using the `org.Hs.eg.db` R package. Further pathway analysis used breast cancer gene signatures from Gatza and colleagues (Gatza *et al.*, 2011; Gatza *et al.*, 2014). These gene symbols were matched to the relevant dataset and used to construct a matrix of category membership using the `safe` R package (Barry, 2016).

2.2 Data Handling

2.2.1 Normalisation

Apart from the PAM50 subtyping procedure (Parker *et al.*, 2009), which required RSEM normalised data (J.S. Parker personal communication), the analysis of the RNA-Seq data presented here was based on raw read count data. Raw read counts were log-scaled; samples removed for consistency (based on a Euclidean distance correlation matrix as described in section 2.2.2); and the final dataset was TMM normalised (Robinson and Oshlack, 2010) then processed using the `voom` function (Law *et al.*, 2014) in the `limma` R package (Ritchie *et al.*, 2015). Protein expression data generated from RPPA was normalised to dilution curves using the `SuperCurve` R package (Ju *et al.*, 2015; Neeley *et al.*, 2009).



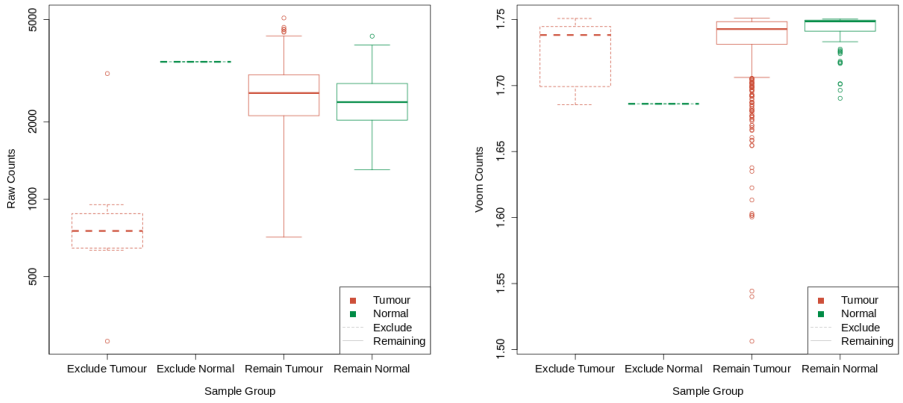
(a) Raw counts (log-scale)

(b) Voom normalised

Figure 2.1: **Read count density.** Sample density plots of raw counts on log-scale and voom normalised showing samples removed due to quality concerns.

2.2.2 Sample Triage

The TCGA breast RNA-Seq data were assessed for batch effects using a correlation matrix of the log-transformed raw counts for which a heatmap (Euclidean distance, complete linkage) is shown in Figure A.2. While no major batch effects were detectable between the samples, 9 samples were excluded due to poor correlation with



(a) Mean raw counts (log-scale)

(b) Mean voom normalised

Figure 2.2: **Read count sample mean.** Boxplots of sample means for raw counts on log-scale and voom normalised show removed tumour samples with low mean read count.

the remaining samples, as detailed in Table 2.1. These samples showed unusual density plots compared to the rest of the dataset, and exhibited low mean read count in Figures 2.1 and 2.2. A heatmap showing key clinical properties of these excluded samples and their correlation with the remainder of the samples is shown in Figure A.1, and a full correlation heatmap (Figure A.2) shows these samples as relatively poorly correlated outliers in the bottom rows and left columns. After removal of these samples, the TCGA dataset used for analysis consisted of the remaining 1168 samples (from 1040 patients): 1049 tumour samples, 112 normal tissue for matched samples, and 7 metastases.

Table 2.1: Excluded Samples by Batch and Clinical Characteristics.

Tissue Source	Type	Batch	Plate	Patient	Samples	p53	Subtype	Treatment (History)		Clinical Subtypes		(Stage)
A7 Christiana	Tumour	47	A227	A0DB	1 of 3	NA	Luminal A	Mastectomy	(no)	ER+	Ductal	(2)
A7 Christiana	Tumour	96	A220	A13D	1 of 3	Wildtype	Luminal A	Mastectomy	(no)	ER+	Ductal	(2)
A7 Christiana	Tumour	96	A227	A13E	1 of 3	NA	Basal	Lumpectomy	(no)	ER−	Ductal	(2)
A7 Christiana	Tumour	142	A277	A26E	1 of 3	NA	Basal	Lumpectomy	(no)	ER+	Ductal	(2)
A7 Christiana	Tumour	47	A277	A0DC	1 of 2	NA	Luminal A	Mastectomy	(yes)	ER+	Lobular	(3)
A7 Christiana	Tumour	142	A220	A26I	1 of 2	Mutant	Basal	Lumpectomy	(yes)	ER−	Ductal	(2)
AC Intl Genomics	Tumour	177	A18M	A2QH	2 of 2	Mutant	Basal	Radical Mastectomy	(no)	ER−	Metaplastic	(2)
AC Intl Genomics	Tumour	177	A220	A2QH	2 of 2	Mutant	Basal	Radical Mastectomy	(no)	ER−	Metaplastic	(2)
GI ABS IUPUI	Normal	177	A16F	A2C8	1 of 1	NA	Luminal A	Radical Mastectomy and Neoadjuvant	(no)	ER+	Ductal	(2)

Similarly, a correlation matrix of log-transformed raw counts was used to evaluate sample quality for TCGA stomach RNASeq. A tumour sample (patient 4294) was removed due to similar quality concerns leaving a final dataset for 449 samples (from

417 patients): 414 tumour samples and 35 normal tissue samples.

2.2.3 Metagenes and the Singular Value Decomposition

A "metagene" offers a consistent signal of pathway (expression) activation or inactivation by dimension reduction of a matrix, avoiding negatively correlated genes averaging out the signal of a mean-based centroid (Huang *et al.*, 2003). Constructing these pathway metagenes used gene sets for Reactome and Gatza signatures (Gatza *et al.*, 2011; Gatza *et al.*, 2014) as specified above (see Section 2.1.1.2). The singular-value decomposition was performed ($X = U^T DV$ where X is the data matrix of the gene set with genes $\times$ samples) and the leading eigenvector (first column of V) corresponding to the largest singular value was used as a metagene for the pathway gene set. To ensure consistent directionality of metagene signals, the median of the gene set in each sample was calculated and correlated against the metagene with the (arbitrary) metagene sign adjusted as needed to conform with the majority of the gene set (i.e., positive correlation between metagene and the median-based centroid). To ensure that genes and pathways were weighted equally, metagenes were derived from a z-transformed dataset of gene expression and samples were scaled (by fractional ranking) for each metagene so that they were comparable on a $[0, 1]$ scale.

2.2.3.1 Candidate Triage and Integration with Screen Data

Candidate triage in combination with the experimental data was intended to integrate findings of the SLIPT analysis with an ongoing experiment project (Chen *et al.*, 2014; Telford *et al.*, 2015). The first procedure to compare the SLIPT gene candidates for *CDH1* with an siRNA experimental screen (Telford *et al.*, 2015) was a direct comparison of the overlapping candidates, presented in a Venn diagram and tested with the χ^2 test. Since these candidates modestly overlapped at the gene level (even when excluding genes not contained in both datasets), further gene set over-representation analysis was performed for pathways specific to each detection approach and the intersection of the two.

The pathway composition of the intersection was further verified by a permutation resampling analysis (as described in section 2.3.6): the same number of genes detected by SLIPT were sampled randomly from the universe of genes tested by both approaches. These samplings were performed over 1 million iterations and the pathway over-representation was compared for each of the 1,652 reactome pathways. These

over-representation scores (χ^2) were compared the observed over-representation in the intersection of the SLIPT candidates, with the proportion of resamplings with higher χ^2 values used for empirical p-values of pathway composition. Pathways for which no resamplings were occurred as high as the observed were reported as $p < 10^{-6}$. These empirical p-values were adjusted for multiple comparisons (FDR). Intersection size was not assumed to be constant across resamplings so similarly with the proportion of resamplings with higher or lower intersection size were used to evaluate significance of enrichment or depletion respectively (of siRNA candidate among SLIPT candidate genes).

2.3 Techniques

Various statistical, computational, and bioinformatics techniques were performed throughout this thesis. This section describes these techniques and gives the parameters used unless otherwise specified. Where relevant, the R package implementation which provided the technique will be acknowledged.

2.3.1 Statistical Procedures and Tests

As described in sections 2.3.4 and 2.2.3, the z-transform has been used to generate z-scores in various analyses in this thesis. Each row of dataset (x) is transformed into a scores (z) using the mean (μ) and standard deviation (σ) of the data such that:

$$z = \frac{x - \mu(x)}{\sigma(x)}$$

This generates data where each row (gene) has a mean of 0 and standard deviation of 1. Where plotted as aa heatmap, any data more than 3 standard deviations above or below the mean is plotted as 3 or -3 respectively.

Empirical Bayes differential expression analysis was performed using the `limma` R package (Ritchie *et al.*, 2015). Where specified, the Fisher's exact test, χ^2 test, and correlation were used to measure associations between variables (as implemented in the `stats` R package (R Core Team, 2016)). Unless otherwise specified, Pearson's correlation was used for correlation analyses (r) and coefficient of determination (R^2). Where these comparisons are discussed in more detail, Fisher's exact test and χ^2 tests are supported by a table or Venn diagram, rendered with the `limma` R package (Ritchie *et al.*, 2015). In some analyses, correlation is furter supported by a scatter plot and a line of best fit dervied by least squares linear regression.

The `t.test` function (R Core Team, 2016) has also been used to implement the t-test to compare pairs of data. Where relevant, an analysis of variance (ANOVA) has been performed to report significance of multivariate predictors of outcomes, or least squares linear regression performed for the adjusted coefficient of determination (R^2) and F-statistic p-value to evaluate the fit of the predictor variables. For some analyses these are supported by boxplot or violinplot visualisation (rendered in R).

Multiple comparisons are adjusted by the Benjamini-Hochberg procedure to control the false discovery rate (FDR) unless otherwise specified (Benjamini and Hochberg, 1995). This procedure adjusts p-values to achieve an average of the proportion of false-positives among significant tests below a threshold, α. The more stringent Holm-Bonferroni (Holm) procedure (Holm, 1979) was also applied in some cases to adjust for multiple comparisons and control the family-wise error rate which adjusts p-values so that the probability that any one of the tests is a false-positive (type-1 error) below a threshold, α.

2.3.2 Gene Set Over-representation Analysis

Gene set enrichment over-representation was performed to test whether there is an enrichment of a gene set (such as a biological pathway) among a group of input genes. Such input genes may be predicted synthetic lethal candidates or a subset defined by clustering (in section 2.3.3) or comparison with experimental candidates (see section 2.2.3.1). Initially, these tests were performed using the GeneSetDB web tool (Araki *et al.*, 2012) hosted by the University of Auckland on the Reactome pathways (Croft *et al.*, 2014). Since the GeneSetDB tool used an older version of Reactome (version 40), it was difficult to directly compare with the results of other analysis (see sections 2.2.3.1 and 2.3.6) performed on version 52 (as described in section 2.1.1.2). Thus an implementation of the hypergeometric test in R (R Core Team, 2016) was used to test for over-representation against Reactome (version 52) pathways. Pathways containing less than 10 genes or more than 500 (as performed in GeneSetDB by Araki *et al.*, 2012) were excluded before adjusting for multiple comparisons.

2.3.3 Clustering

Clustering analysis when performed uses unsupervised hierarchical clustering with complete linkage (distance calculated from the furthest possible pairing). For correlation matrices or multivariate normal parameters (e.g., Σ), the distance metric used was

Euclidean distance. For empirical or simulated gene and pathway expression data correlation distance was used, calculated by $distance = 1 - cor(t(x))$ where cor is Pearson's correlation and $t(x)$ is the transpose of the expression matrix.

2.3.4 Heatmap

Standardised z-scores of the data were used to plot heatmaps on an appropriate scale. Raw (log-scale) read counts or voom normalised counts per gene (as specified) were plotted as normalised z-scores on a $[-3, +3]$ blue-red scale. Similarly, correlations were plotted on a $[-1, +1]$ blue-red scale. These heatmaps were performed using the linkage and distance specified for the clustering performed in Section 2.3.3. The `gplots` R package (Warnes *et al.*, 2015) was used to generate many of the heatmaps throughout this thesis, along with a customised heatmap function (released as `heatmap.2x`). Where clearly specified, data have been split into subsets with clustering performed separately on each subset with these plotted alongside each other.

2.3.5 Modeling and Simulations

Statistical modeling and simulations have been used to test various synthetic lethal detection procedures on simulated data. This involves constructing a statistical model of how synthetic lethality would appear in (continuous normally distributed) gene expression data. Where presented (in section 3.2.1), the assumptions of the model are stated clearly. The model allows sampling from a multivariate normal distribution (using the `mvtnorm` R package (Genz and Bretz, 2009; Genz *et al.*, 2016)) to generate simulated data with known underlying synthetic lethal partners (detailed in section 3.2.2). We can test whether statistical procedures, including those developed in this thesis (presented in section 3.1), are capable of detecting them upon this simulated data. This multivariate normal simulation procedure also enables the inclusion of correlation structure which is either given as correlated blocks of genes or derived from pathway structures (as detailed in section 3.4.2).

If this multivariate normal distribution was sampled once and the procedure to add known synthetic lethal partners was performed, it generates a simulated dataset. Performing this simulation procedure and testing with a synthetic lethal detection procedure iteratively, these simulations can be used to assess the statistical performance of the detection procedure. The number of iterations (`Reps`) will be given for each simulation result. Typically, these are performed 1000 or 10,000 times depending on

computational feasibility of doing so on larger datasets.

Several measures of statistical performance were used to assess the simulations. The following measures used the final classification of the detection procedure, statistical significance for χ^2, significance and directional criteria met for SLIPT (see section 3.1), and an arbitrary threshold: < -0.2 and $> +0.2$ for negative correlation and correlation respectively. Sensitivity (or "true positive rate") was measured as the proportion of known synthetic lethal partners predicted to be synthetic lethal. Specificity (or "true negative rate") was measured as the proportion of known non-synthetic lethal partners predicted not to be synthetic lethal. The "false discovery rate" (also used in adjusting for multiple comparisons) was measured here as the proportion of known non-synthetic lethal partners out of all putative partners predicted by the detection procedure. Statistical "accuracy" is the proportion of true predictions for a detection procedure, which is both the correctly predicted known synthetic lethal partners and correctly negative known non-synthetic lethal partners.

2.3.5.1 Receiver Operating Characteristic (Performance)

A more general procedure to measure the statistical performance of a simulation is the Receiver Operating Characteristic (ROC) curve which does not assume a threshold for classification of synthetic lethality but demonstrates the trade-off of sensitivity and specificity (Akobeng, 2007; Fawcett, 2006; Zweig and Campbell, 1993). These curves (implemented with the `ROCR` R package (Sing *et al.*, 2005)) plot the True Positive Rate (sensitivity) against the False Positive Rate (1−specificity) as the prediction threshold is varied. An ideal detection method will have a true positive rate of 1 and a false positive rate of 0, hence the Area Under the ROC curve (AUC or AUROC) is a measure of statistical performance for a detection procedure accounting for this trade-off. AUROC values are typically range from 0.5 the value expected by random chance to 1 for an optimal detection method, however it is possible for an AUROC below 0.5 for a poor detection method that performs worse than random chance. An cancer biology, an AUROC of approximately 0.8 is a predictive biomarker suitable for publication (Hajian-Tilaki, 2013) but predictors with lower AUROC values may still be informative depending on the context. In this thesis, the AUROC values varies widely across simulation parameters and a primarily used for comparisons across these parameters, although they can also be used to refine thresholds for optimal classification.

2.3.6 Resampling Analysis

Resampling analyses (e.g., "permutation" analysis) are used to statistically test the significance of an observation without assuming the underlying distribution of expected test statistics Collingridge (2013). Instead these are derived from randomly shuffling test statistics or randomly sampling predicted candidates. For the purposes of this thesis, this involved randomly sampling genes from those tested to be analysed as putative synthetic lethal candidates. This was performed both for testing the significance of pathway composition in the intersection with experimental gene candidates (section 2.2.3.1) and for assessing the significance of pathway structure among synthetic lethal candidates (section 3.4.1.1).

These were analysed to compare the observed synthetic lethal genes against values derived from randomly sampling the same number of genes as observed by synthetic lethal from among the genes tested. Sampling iteratively across many resampling procedures, these resampling-based values form a null distribution that would be observed if the null hypothesis were true. Thus the proportion of resampling-based values across these iterations that are greater than or equal to that observed, forms an empirically derived p-value to test significance.

Resampling was performed for comparison (in section 2.2.3.1) with fixed experimental screen candidates (Telford *et al.*, 2015) both resampling the number of genes overlapping with the screen candidates and test statistics for pathway enrichment. Resampling analysis was also applied to shortest paths and network metrics (in section 3.4.1.1) to test significance of directional relationships between synthetic lethal candidate genes within pathway structures.

The number of iterations determines the accuracy of these p-values. For pathway composition (in section 2.2.3.1), a million iterations were performed using high performance computing (as detailed in section 2.5.3) to provide sufficient accuracy after adjusting for multiple comparisons across pathways. For the purposes of network analysis (in section 3.4.1.1), a thousand iterations were sufficient to reject the null hypothesis for the majority of pathways tested before adjusting for multiple comparisons, and thus further iterations were not performed.

2.4 Pathway Structure Methods

2.4.1 Network and Graph Analysis

Networks are important in considering the structure of relationships in molecular biology, including gene regulation, kinase cellular signaling, and metabolic pathways (Barabási and Oltvai, 2004). Network theory is an interdisciplinary field which combines the approaches of computer science with the metrics and fundamental principles of graph theory, an area of pure mathematics dealing with relationships between sets of discrete elements. The vast amounts of molecular and cellular data from high-throughput technologies have enabled the application of network-based and genome-wide bioinformatics analysis to examine the complexity of a cell at the molecular level and understand aberrations in cancer. This thesis uses various metrics and analysis procedures developed in Graph and Network theory to analyse graph structure of biological pathways. Where feasible, these have been implemented using the `igraph` R package with such procedures described below (Csardi and Nepusz, 2006). Custom R functions to perform more complex analysis and visualisation of iGraph data objects will be described later.

Graph theory is a branch of pure mathematics which deals with the properties of sets of discrete objects (referred to as a 'node' or 'vertex') with some pairs are joined (by a 'link' or an 'edge'). While a seemingly reductionist abstraction to mathematically study relationships, graph theory serves has applications in a wide range of studies including life sciences. Network theory is the sub-discipline of graph theory which deals with networks which has become popular due to the vast potential for applications of networks (van Steen, 2010).

Applications vary depending on the situation modelled, particularly in how the edges between vertices are defined, whether they are directed or weighted, and whether multiple redundant edges between a pair of vertices (referred to as 'parallel edges') or edges connecting a vertex to itself (referred to as 'loops') are permitted in the model. Networks are defined such that the edges represent a relationship between the vertices and may be directed, weighted, or contain parallel edges or loops depending on the application (van Steen, 2010). Unless otherwise stated, graph structures and networks in thesis will be unweighted and have no parallel edges or loops. Where a directional relationship is known or modelled, it will be represented with a directed edge in a directed graph.

2.4.2 Sourcing Graph Structure Data

Pathway Commons interaction data was sourced using the paxtools-4.3.0 Java application on October 6th 2015 (Cerami *et al.*, 2011; Demir *et al.*, 2013). This utility was used to source 'sif' format interaction data into R (R Core Team, 2016), from which the human Reactome (version 52) dataset of interactions was imported (Croft *et al.*, 2014), matching those used for pathway enrichment analysis. These interactions were used to construct an adjacency matrix for the Reactome network and subnetworks corresponding to each relevant biological pathway.

2.4.3 Constructing Pathway Subgraphs

Subgraphs for each relevant pathway were constructed by matching the nodes in the complete Reactome network to the pathway gene sets (as derived in section 2.1.1.2). A subgraph with adjacent nodes was constructed by adding nodes which have an edge with a gene in the pathway gene set. The pathways these adjacent nodes belong to were added to form a "meta-pathway" to account for the possibility for nodes within the pathway being linked by the surrounding graph structure.

2.4.4 Network Analysis Metrics

The existing network analysis measures applied in this thesis (as described below) used an implementation in the `igraph` R package where it was available (Csardi and Nepusz, 2006). Otherwise, custom features were developed for analysis of iGraph objects in R and released as `igraph.extensions` (as described in section 3.5.3).

Vertex degree is the number of edges a node has and is a fundamental measure of the importance and connectivity of a network (van Steen, 2010). More connected nodes, such as network hubs, will have a higher vertex degree relative to other nodes. For the purposes of this thesis, vertex degree ignored edge direction with loops (edges with itself) and double edges to the same node excluded.

A fundamental concept in network analysis is a "shortest path", that is the shortest route via edges between any two particular nodes in a network. These are computed by Dijstra's algorithm (Dijkstra, 1959) in the `igraph` R package (Csardi and Nepusz, 2006). Where applicable paths will only use directed edges in a particular direction. Shortests paths are a useful measure of how close nodes are in a network. This is used to compute information centrality, and for further analysis of pathway structure (as

described in section 3.4.1).

Network centrality is an alternative measure of the importance or influence of a node to the graph structure (Borgatti, 2005). Various strategies are used to derive centrality, typically based on how connected the node is or the impact of node removal on the connectivity of the network. One of the most notable is the "PageRank" algorithm, a refinement of eigenvector centrality based on the eigenvectors of the adjacency matrix (Brin and Page, 1998). This is implemented in the `igraph` R package (Csardi and Nepusz, 2006).

Another network centrality measure that has been previously applied to biological protein interaction networks (Kranthi *et al.*, 2013) is the "information centrality". The information centrality of a node is the relative impact on efficiency (transmission of information via shortest paths) of the network when the node is removed. That is the centrality (C) (Kranthi *et al.*, 2013) for node n in graph G is defined as:

$$C_n = \frac{E(G) - E(G')}{E(G)}$$

where G' is the subgraph with the node removed and E is the efficiency (Latora and Marchiori, 2001) derived from shortest paths (d_{ij} between nodes i and j) as:

$$E(G) = \frac{2}{N(N-1} \sum_{i<j \in G} \frac{1}{d_{ij}}$$

The efficiency of the network can be derived from shortest paths implemented in the `igraph` R package and the iterative network centrality computation of each node has been released as an R package (`info.centrality`) and included in the `igraph.extensions` package.

2.5 Implementation

2.5.1 Computational Resources and Linux Utilities

Several computers were used to process and store data during this thesis (as summarised in Table 2.2), running different versions of Linux operating systems, including a personal laptop computer, laboratory desktop machine, departmental server, and the New Zealand eScience Infrastructure Intel Pan high-performance computing cluster (a supercomputer based at the University of Auckland). Each of these systems support a 64-bit architecture. Current workflows on local machines use Elementary OS (based

on the Ubuntu versions given in Table 2.2) and interacting with these via ZSH shell. However, Ubuntu OS and the Bourne Again SHell (bash) were used at the inception of this project and bash is continues to be used for running scripts. Various Linux applications and command-line utilities were used on these machines (as summarised in Table 2.3). As such, the workflows developed in this project should be backwards-compatible with Ubuntu Linux (and other derivatives). The majority of novel methodology and implementations were performed in R which is a cross-platform language, packages developed in R will be available for users of Linux, Mac, and Windows machines.

Table 2.2: Computers used during Thesis

	Viao Laptop	**Lab Machine**	**Biochem Server**	**NeSI Pan Cluster**
Operating System (OS)	Elementary OS Freya 0.3.2	Elementary OS Loki 0.4	Red Hat Enterprise Maipo 7.2	Cent OS Final 6.4
Upstream OS	Ubuntu LTS Trusty 14.04	Ubuntu LTS Xenial 16.04		
Linux Kernel	3.19.0-65-generic	4.4.0-36-generic	3.10.0-327.36.2.el7.x86_64	2.6.32-504.16.2.el6.x86_64
Shell: bash	4.3.11(1)	4.3.46(1)	4.2.46(1)	4.2.1(1)
Shell: zsh	5.0.2	5.1.1	5.0.2	5.2

2.5.2 R Language and Packages

The R programming language has been used for the majority of this thesis. Current R installations across the machines used are given in Table 2.4. Local machines currently run the latest version of the R (at the time of writing) and remote machines run the versions and modules as managed by the system administrator. Various scripts and packages in this thesis were developed or run in previous versions of RStudio and R but these run without error in the current version of R (and the older versions on remote machines). The R packages developed during this thesis are given in Table 2.5 with the relevant sections describing their implementation and use where appropriate, in addition to further details on these functions in section 3.5. Various R packages were used throughout this thesis (as detailed in Table 2.6 with versions specified) installed from the Comprehensive R Archive Network (CRAN, 2017), Bioconductor (Gentleman *et al.*, 2004, version 3.4; BiocInstaller 1.24.0) , or GitHub. These packages were not updated when they would change the functionality of scripts or functions in packages, in particular imported data from annotation packages (used to define gene sets) have been saved as local files to continue using stable versions of these pathway data (across machines). This is a summary of the key packages which (in addition

Table 2.3: Linux Utilities and Applications used during Thesis

		Viao Laptop	**Lab Machine**	**Biochem Server**	**NeSI Pan Cluster**
	OS	Elementary OS Freya 0.3.2	Elementary OS Loki 0.4	Red Hat Enterprise Maipo 7.2	Cent OS Final 6.4
	Linux Kernel	3.19.0-65-generic	4.4.0-36-generic	3.10.0-327.36.2.el7.x86_64	2.6.32-504.16.2.el6.x86_64
Scripting	Shell bash	4.3.11(1)	4.3.46(1)	4.2.46(1)	4.2.1(1)
	Shell zsh	5.0.2	5.1.1	5.0.2	5.2
Programming	Python	2.7.6	2.7.12	2.7.5	
	Java	1.8.0_101	9-ea	1.8.0_101	
	C++	4.8.4	5.4.0	4.8.5	4.4.7
Text Editor	nano	2.2.6	2.5.3	2.3.1	2.0.9
	kile (LaTeX)	2.1.3	2.1.3		
Version Control	git	1.9.1	2.11.0	1.7.1	1.8.3.1
Shell Utilities	sed	4.4.2	4.4.2	4.4.2	4.4.1
	grep	2.16-1	2.25-1	2.20	2.6.3
	nohup	8.21	8.25	8.22	8.4
Typesetting	TeX	3.1415926	3.14159265		
	TexLive (LaTeX)	2013	2015		
	PDFTeX	2.5-1	2.6		
	pandoc	1.12.2.1	1.16.0.2		
Remote Computing	slurm scheduler				16.05.6
	OpenSSH	7.2p2	7.2p2	6.6.1	5.3p1
	OpenSSL	1.0.2g	1.0.2g	1.0.01e-fips	1.0.01e-fips
	rsync	3.1.0p31	3.1.1p31	3.0.9p30	
	Globus Online Transfer			3.1	3.1
	Cisco AnyConnect VPN		3.1.05170		
Image Processing	Inkscape	0.48.4	0.91		
	GIMP	2.8.10	2.8.16		
	ImageMagick	6.7.7.10-6			

Table 2.4: R Installations used during Thesis

		Viao Laptop	**Lab Machine**	**Biochem Server**	**NeSI Pan Cluster**
	OS	Elementary OS Freya 0.3.2	Elementary OS Loki 0.4	Red Hat Enterprise Maipo 7.2	Cent OS Final 6.4
Programming	R	3.3.2	3.3.2	3.3.1	3.3.0-intel (module)
Development	RStudio	1.0.136	1.0.136	1.0.136 (server)	

to their dependencies) have been used throughout this project. Where a package implementation has been central to the methods applied, they are described in more detail in the relevant section. A full table of packages used in this thesis can be found in the Appendix (Table B.1).

Table 2.6: R Packages used during Thesis

Package	**Version Used**	**Built**	**Repository**
colorspace	1.3-2	3.3.1	CRAN
curl	2.3	3.3.1	CRAN
data.table	1.9.6	3.3.1	CRAN

dendextend	1.4.0	3.3.2	CRAN
DBI	0.5-1	3.3.1	CRAN
devtools	1.12.0	3.3.1	CRAN
dplyr	0.5.0	3.3.1	CRAN
ggplot2	2.2.1	3.3.1	CRAN
git2r	0.18.0	3.3.1	CRAN
gplots	3.0.1	3.3.1	CRAN
gtools	3.5.0	3.3.1	CRAN
igraph	1.0.1	3.3.1	CRAN
matrixcalc	1.0-3	3.3.1	CRAN
mclust	5.2.2	3.3.1	CRAN
mvtnorm	1.0-6	3.3.1	CRAN
org.Hs.eg.db	3.1.2	3.1.2	Bioconductor
openssl	0.9.6	3.3.1	CRAN
plyr	1.8.4	3.3.1	CRAN
purrr	0.2.2	3.3.1	CRAN
reactome.db	1.52.1	3.2.1	Bioconductor
RColorBrewer	1.1-2	3.3.1	CRAN
Rcpp	0.12.9	3.3.1	CRAN
ROCR	1.0-7	3.3.1	CRAN
roxygen2	6.0.1	3.3.2	CRAN
shiny	1.0.0	3.3.1	CRAN
snow	0.4-2	3.3.1	CRAN
testthat	1.0.2	3.3.2	CRAN
tidyr	0.6.1	3.3.2	CRAN
tidyverse	1.1.1	3.3.2	GitHub (hadley)
sm	2.2-5.4	3.3.1	CRAN
Unicode	9.0.0-1	3.3.2	CRAN
vioplot	0.2	3.3.1	CRAN
viridis	0.3.4	3.3.2	CRAN
xml2	1.1.1	3.3.2	CRAN
xtable	1.8-2	3.3.1	CRAN
zoo	1.7-14	3.3.1	CRAN
graphics	3.3.2	3.3.2	base
grDevices	3.3.2	3.3.2	base

cluster	2.0.5	3.3.1	base
graphics	3.3.2	3.3.2	base
grDevices	3.3.2	3.3.2	base
Matrix	1.2-8	3.3.1	base
stats	3.3.2	3.3.2	base

2.5.3 High Performance and Parallel Computing

Another enabling technology for bioinformatics is parallel computing, performing independent operations in separate cores: this "multithreading" is widely used to increase the time to compute results. Bioinformatics is particularly amenable to this since performing multiple iterations of a simulation or testing separate genes is often "embarrassingly parallel", being completely independent of the results of each other. As such parallel computing is offered by many high-performance "supercomputers" including national research infrastructure.

The New Zealand eScience Infrastructure (NeSI) is a computating resource providing the Intel Pan cluster hosted by the University of Auckland (NeSI, 2017). The Pan cluster used throughout this thesis project to optimise and perform computations which would have otherwise been infeasible in the timeframe of thesis. Such technological developments and infrastructure initiatives have enabled bioinformatics research including this project. High performance computing on the Pan cluster was used extensively in this project including for resampling analysis (in sections 2.3.6 and 3.4.1.1), calculating information centrality (in section 2.4.4), and in simulations (in sections 2.3.5, 3.2, and 3.4.2)

Scripts and data were transferred between the Pan cluster and University of Otago computing resources by `rsync` or the Globus file transfer service (Globus, 2017). R scripts (R Core Team, 2016) were run in parallel with the "simple network of workstations" `snow` R package Tierney *et al.* (2015). This utilised the "message passing interface" (Yu, 2002) when it was feasible with memory requirements to run in parallel across multiple compute nodes, otherwise SOCKS was used to access multiple cores within an instance of R and pass input data to them. R jobs were submitted to queue for available resources and run on the Pan cluster via the Slurm (Simple Linux Utility for Resource Management) workload manager (Slurm, 2017). When running R scripts across many parameters or for memory-intensive jobs, Slurm array job submission and independent submission of different parameters via shell commands with

Table 2.5: R Packages Developed during Thesis

	Package Name	Description and GitHub Repository	Section
	slipt	Synthetic lethal detection by SLIPT (to accompany publication) https://github.com/TomKellyGenetics/slipt	3.1
visualisation	vioplotx	Customised violin plots (based on vioplot) https://github.com/TomKellyGenetics/vioplotx	
	heatmap.2x	Customised heatmaps (based on gplots) https://github.com/TomKellyGenetics/heatmap.2x	2.3.4
igraph.extensions	igraph.extensions	Meta-package to install the follow iGraph functions https://github.com/TomKellyGenetics/igraph.extensions	3.5.3
	plot.igraph	Custom plotting of directed graphs https://github.com/TomKellyGenetics/plot.igraph	2.4.4
	info.centrality	Computing information centrality from network efficiency https://github.com/TomKellyGenetics/info.centrality	3.4.2
	pathway.structure.permutation	Testing pathway structure with resampling analysis https://github.com/TomKellyGenetics/pathway.structure.permutation	3.4.1.1
	graphsim	Generating simulated expression from graph structures https://github.com/TomKellyGenetics/graphsim	3.4.2

arguments passed to R. In some cases, this submission was automated across a range of parameters with Bash scripts.

Chapter 3

Methods Developed During Thesis

In this chapter, I will outline the rationale and development of various methods used throughout this thesis to examine synthetic lethality in gene expression data, graph structures, models and simulations. First by describing the Synthetic Lethal Interaction Prediction Tool (SLIPT), a bioinformatics approach to triage of synthetic lethal candidate genes. This is considered one of the main research outputs of the thesis, which is supported by comparisons to an experimental screen from a related project and performance on simulated data. These supporting data will be covered in further chapters but preliminary data to support the use and design of SLIPT are provided alongside description of the method. This includes the construction of a statistical model of synthetic lethality in (continuous multivariate Gaussian) gene expression data, which enables testing SLIPT upon simulated data with known synthetic lethal partners. Another key component of the simulation pipeline used later is the generation of simulated data from a known graph structure or simulated biological pathway. The development of this simulation procedure and other statistical treatment of graph and network structures will also be covered. Various R packages have been developed to support this project, most notably the `slipt` package to implement the SLIPT methodology. The additional R packages for handling graph structures, simulations, and custom plotting features will also be described as research outputs of this thesis, methods applied throughout, and contributions to the open-source software community that made this project feasible.

3.1 A Synthetic Lethal Detection Methodology

The SLIPT methodology identifies gene expression patterns consistent with synthetic lethal interactions between a query gene and a panel of candidate interacting partners. Gene expression is called low, medium, or high by separating samples into tertiles (3-quantiles) for each gene. Genes with insufficient expression across all samples were excluded by requiring that the first tertile of raw counts is above zero. Then a χ^2 test is performed between the query gene and each candidate partner, with the p-values for the χ^2 test being corrected for multiple testing using false discovery rate (FDR) error control to reduce false positives for large candidate gene panels (Benjamini and Hochberg, 1995). Significance was called only if FDR adjusted p-values were below the threshold $p < 0.05$. A synthetic lethal interaction is predicted (as shown in Figure 3.1) when (i) the χ^2 test is significant; (ii) observed low-query, low-candidate samples

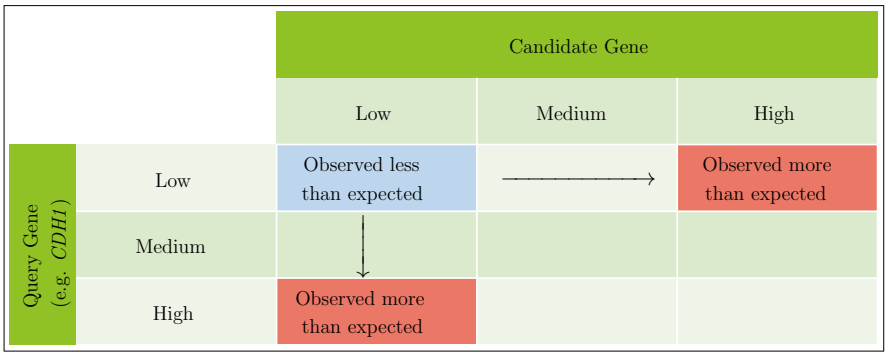


Figure 3.1: **Framework for synthetic lethal prediction.** Synthetic Lethal Interaction Prediction Tool (SLIPT) was designed to identify candidate interacting genes from gene expression data using the χ^2 test against a query gene. Samples are sorted into low, medium, and high expression quantiles for each gene to test for a directional shift. A sample being low in both genes of a synthetic lethal pair is unlikely, since loss of both genes will be deleterious, and is expected to be statistically under-represented in a gene expression dataset. We expect a corresponding (symmetric) increase in frequency of sample with low-high gene pairs. Synthetic lethal candidate (exprSL) partners of a gene are identified by running this procedure on all possible partner genes, selecting those with an FDR-adjusted χ^2 p-value of $p < 0.05$, and meeting the directional criteria. Since synthetic lethal genes are partners of each other commutatively, the symmetric direction criteria are all required such that synthetic lethal genes will predicted to be partners of each other.

are less frequent than expected; and (iii) observed low-query, high-candidate and high-query, low-candidate samples are more frequent than expected.

The synthetic lethal prediction procedure has also been adapted to utilise somatic mutation data for the query gene. This is intended to utilise a query gene known to be recurrently mutated in the disease (and dataset), with the majority of mutations inactivating gene function (such as null or frameshift mutations). A synthetic lethal interaction is predicted (as shown in Figure 3.2) when (i) the χ^2 test is significant; (ii) observed mutant-query, low-candidate samples are less frequent than expected; and (iii) observed mutant-query, high-candidate and wild-type-query, low-candidate samples are more frequent than expected. Unless otherwise specified, computationally predicted synthetic lethal gene candidates from SLIPT used expression data (exprSL) for both genes (as shown in Figure 3.1) rather than mutation data (mtSL) for the query gene (as shown in Figure 3.2).

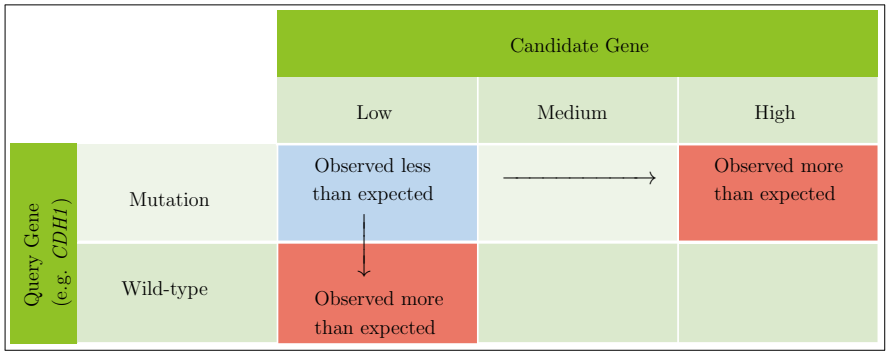


Figure 3.2: **Synthetic lethal prediction adapted for mutation.** Synthetic Lethal Interaction Prediction Tool (SLIPT) was also adapted to identify candidate interacting genes using (somatic) mutation data of the query gene in the χ^2 test. Samples are sorted into low, medium, and high expression quantiles for each candidate gene and tested for a directional shift against mutation status of the query gene. A sample having low expression or mutation for the synthetic lethal pair is expected to be unlikely with a corresponding increase in frequency of sample with mutant-high or wild-type-low gene pairs. Synthetic lethal candidate (mtSL) partners of a gene are identified by running this procedure on all possible partner genes, selecting those with an FDR-adjusted χ^2 p-value of $p < 0.05$, and meeting the directional criteria.

3.2 Synthetic Lethal Simulation and Modelling

A statistical model of Synthetic Lethality was developed to generate simulated data to test the SLIPT procedure. This section will describe the synthetic lethal model and the simulation procedure for generating gene expression data with known synthetic lethal partners. Some preliminary results to support usage of the SLIPT methodology throughout this thesis will be presented here. The simulation procedure will be applied in more depth in chapter 6, including in combination with simulations from graph structures.

3.2.1 A Model of Synthetic Lethality in Expression Data

A conceptual model of synthetic lethality was constructed (see Figure 3.3), which will be used to build a statistical model of synthetic lethal gene expression from which to simulate expression data to on which test SLIPT and various potential synthetic lethal prediction methods. In the model, synthetic lethality arises between genes with related functions as a cell death phenotype when these functions are removed.

This model suggests that synthetic lethality is detectable in measures of gene inactivation across a sample population, namely mutation, DNA copy number, DNA methylation, and suppression of expression. While any of these mechanisms of gene inactivation could lead to synthetic lethality, expression data is readily available and changes in these alternative mechanisms are likely to impact on the amount of expressed (functional) RNA or protein detectable. There are several ways that functional relationships between genes could manifest in expression data, including coexpression, mutual exclusivity and directional shifts. Co-expression is overly simplistic and has previously performed poorly as a predictor of synthetic lethality (Jerby-Arnon *et al.*, 2014), although this will still be tested with correlation measures in later simulations. Here the alternative hypothesis is that synthetic lethality will lead to a detectable directional shift in the number of samples exhibiting low or high expression of either gene. This model does not preclude mutual exclusivity (Wappett *et al.*, 2016), compensating expression or co-loss under-representation (Lu *et al.*, 2015) as previously postulated to occur between synthetic lethal genes.

The first condition of the synthetic lethal model is that if there are only two synthetic lethal genes (e.g., *CDH1* and one SL partner), then they will not both be non-functional in the same sample (in an ideal model). Gene function is thus determined for each sample in a model of synthetic lethal with the proportion of samples with a

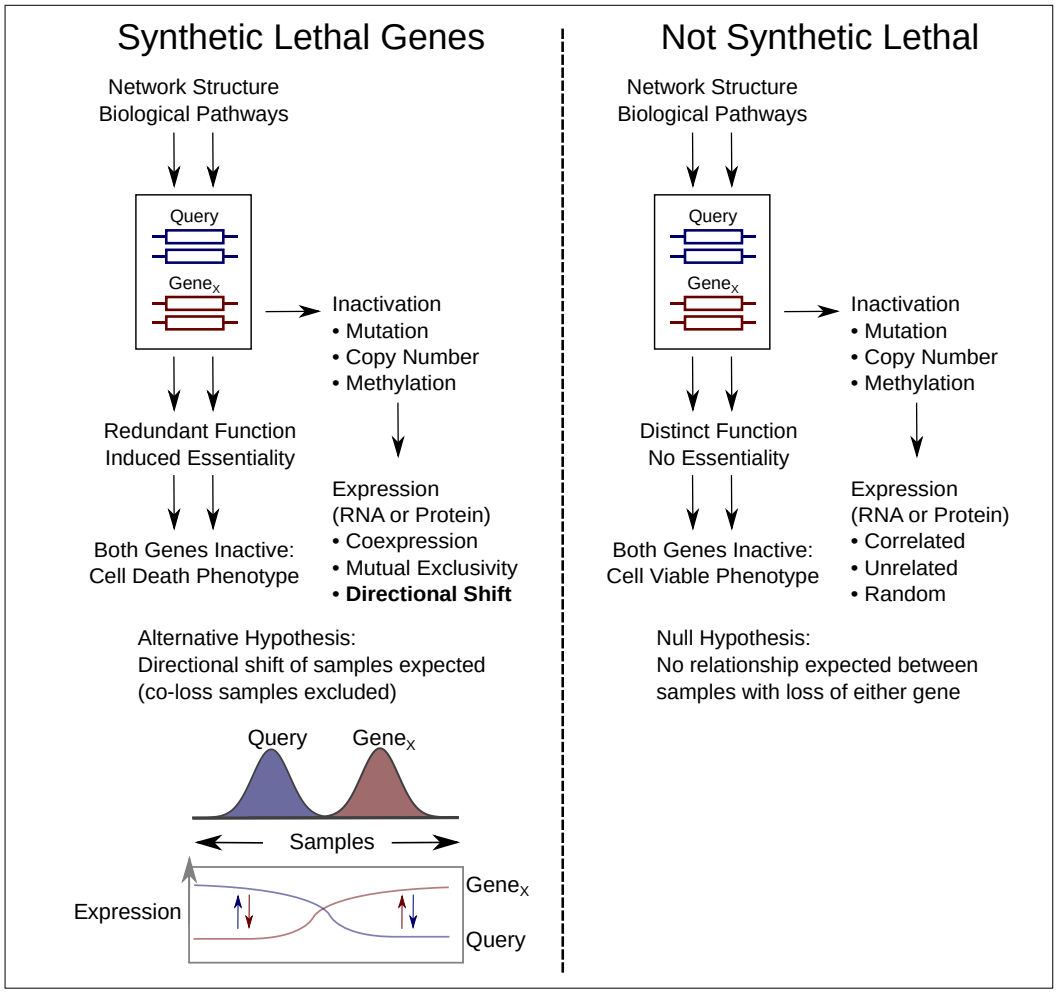


Figure 3.3: **A model of synthetic lethal gene expresion.** A conceptual model of synthetic lethal interactions between a Query gene and partner gene (G_X). Genes that are synthetic lethal may not both be non-functional in the same sample without another gene compensating for the loss of function. This is most likely to be detectable as low gene expression, whether they are lost by mutation, deletion, DNA methylation, or suppressing regulatory signals. This could manifest as coexpression, mutual exclusivity, or directional shifts in sample frequency. Thus the alternative hypothesis (H_A) is that synthetic lethal genes will have a reduced frequency of co-loss samples while the null hypothesis (H_0) is that non-synthetic lethal gene pairs would show no such relationship, even if they may be correlated for other means such as pathway relationships. In this model synthetic lethal genes may compensate for the loss of each other but this is not assumed, only that loss of both is unfavourable to cell viability and probability of detecting samples with combined gene loss.

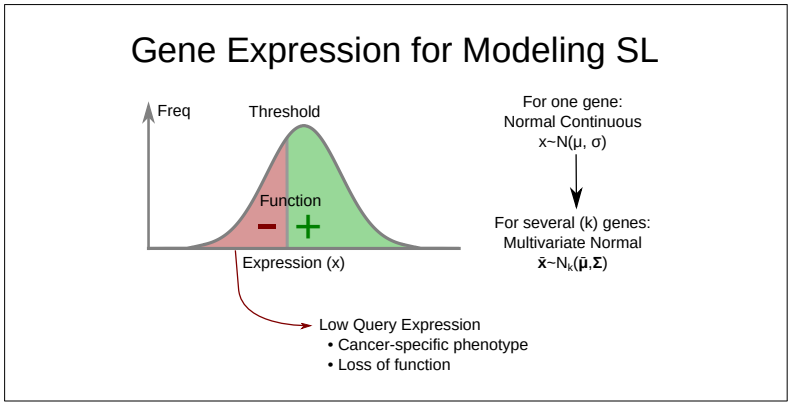


Figure 3.4: **Modeling synthetic lethal gene expression.** When modeling synthetic lethal interactions between a Query gene with partner genes (G_X and G_Y) above, cellular viability requires that at least of genes is not inactivated. Expression below a threshold is used as a model of loss of function, where genes are regarded as non-functional for the purposes of modelling synthetic lethality. Tumour suppressor genes with loss of function also have cancer specific phenotypes (although these thresholds are not necessarily the same). Expression is modeled by a normally (Gaussian) distributed continuous data such as (log-scale) data from RNA (microarray or RNA-Seq), protein, or pathway metagenes. This rationale generalises for several genes on a multivariate normal distribution.

functional or non-functional gene being arbitrary. Whether a gene is functional can similarly be modelled by an arbitrary threshold of continuous and normally distributed gene expression data to define gene function (as shown in Figure 3.4). For the purposes of modeling synthetic lethality in breast cancer expression data, a threshold of the 30^{th} percentile of the expression levels was used because approximately 30% of samples analysed had *CDH1* inactivation. This was generalised for a model of the proportion of samples inactivated for each gene. In this ideal case, no samples lowly expressing both of these genes are expected to be observed. While this is not observed, that is to be expected as it is unlikely that only 2 genes will have an exclusive synthetic lethal partnership. The threshold of the 0.3 quantile was used in simulations dervied from this model throughout this thesis.

A synthetic lethal pair of genes is unlikely to act in isolation, therefore higher-order synthetic lethal interactions (i.e., 3 or more genes) must be considered in the model as shown in Figure 3.5. Even when testing pairwise interactions, modelling higher level interactions that may interfere is important. If there are additional synthetic lethal partners, there are two possibilities for adding these: 1) that they are independent

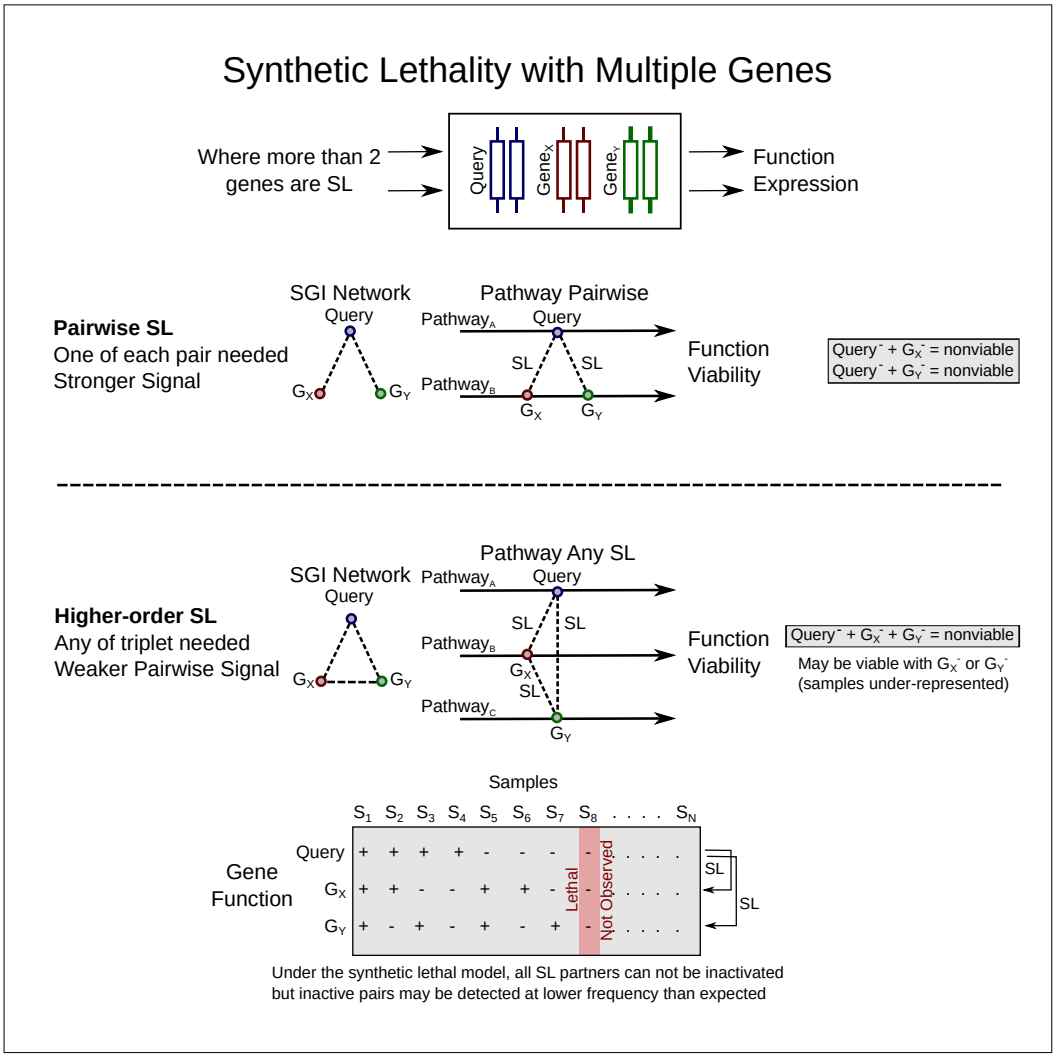


Figure 3.5: **Synthetic lethality with multiple genes.** Higher order synthetic lethal interactions may occur between 3 or more genes, affecting the simulated expression (or synthetic lethal predictions) even if undetected when observed pairwise. Consider interactions between a Query gene and two partner genes (G_X and G_Y). They may interact with the Query pairwise (inviable when either gene pair is lost) or form a higher-order interaction such as the "synthetic lethal triplet" if any of the genes provide an essential function (inviable only when all are lost). Either is plausible with the potential pathway structures. A synthetic lethal triple has 8 potential combinations of gene functional but one is not expected to be observed (due to inviability) but pairwise inactivation may be observed if additional partner genes are functional. The proportion of these combinations vary depending on the functional threshold.

partners of the query genes interacting pairwise (and not with each other) or 2) that an addition partner gene interacts with both of the synthetic lethal genes already in the system and any of the three (or more) are required to be functional for the cell to survive.

The signal (in terms of gene expression data) will be weaker for this latter case and this model has the more stringent assumption that all synthetic lethal partner genes interact with each other: that only one of these must be expressed to satisfy the model of synthetic lethality. In this model any of the synthetic lethal genes in a higher-order interaction is able to provide the missing function of the others, allowing for higher-level synthetic lethal partners to compensate for loss a synthetic lethal gene pair. While samples expressing low levels of the synthetic lethal gene pairs will be under-represented, they may not be completely absent from the dataset due to these higher-level interactions.

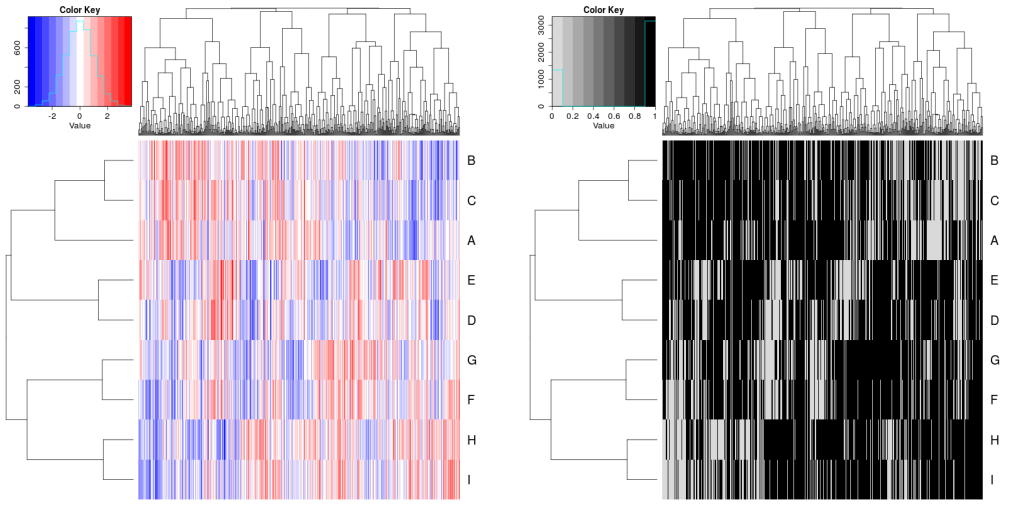
In the example of 3 synthetic lethal genes 3.5, only one of genes involved in the higher-order synthetic lethal interaction is required for cell viability. For synthetic lethal pairs, only a subset of these samples will be inviable (i.e., removed from simulated data), leading to an under-representation.

In practice, samples are not removed from a simulated dataset, rather the expression and function of the query gene is generated across samples separately from the pool of potential partner genes. The query gene data is matched to simulated samples (as shown in Figure 3.7), satisfying the synthetic lethal condition with the procedure described in section 3.2.2. This is performed to maintain a comparable samples size across simulations and the preserve the assumed (multivariate) normal distribution of the data.

3.2.2 Simulation Procedure

Simulations were developed to simulate normal distributions of expression data and define function with a threshold cut-off. This is the reverse to the procedure of SLIPT to predict synthetic lethal partners (although the threshold is assumed to be unknown when testing upon simulated data). While gene function is used as an intermediary step in modelling synthetic lethal genes in expression data, the normal distribution is sampled for simulated data to represent normalised empirical gene expression data for which SLIPT (and other methods) will be applicable.

Sampling a distribution for expression profiles has the added advantage of being amenable to simulating correlation structures with the multivariate normal distribu-



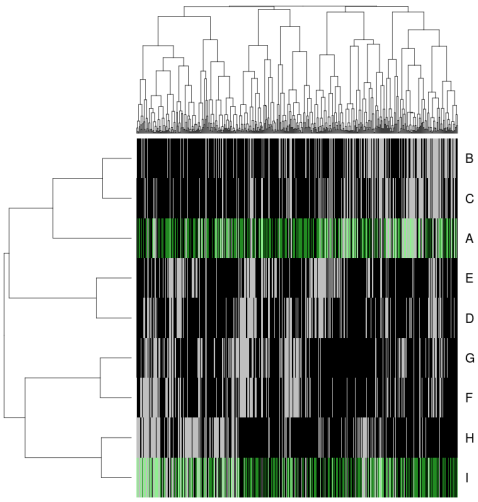
(a) Simulated expression matrix (b) Corresponding gene function calls

Figure 3.6: **Simulating gene function.** A simulated dataset with samples (columns) and genes A–H (rows) is transformed from a continuous (coloured blue–red) scale to a discrete matrix of gene function (black for functional levels and grey for non-functional).

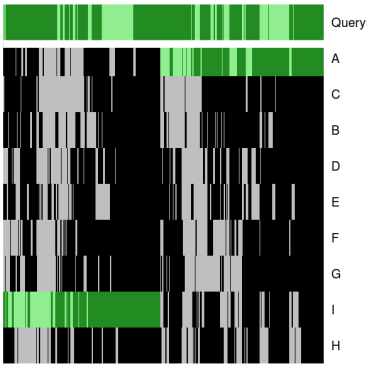
tion (using the `mvtnorm` R package (Genz and Bretz, 2009; Genz *et al.*, 2016)). The parameter Σ is a covariance matrix defines the correlation structure between simulated genes being sampled. With a diagonal of one, this Σ matrix simulates genes with a standard deviation of one and the covariance parameters between them are the correlations between each gene. In Figure 3.6, an example of such a simulated multivariate normal dataset is shown with the functional threshold applied.

Once we have generated a simulated dataset, the samples are compared by gene function (as derived from a functional threshold). Known underlying synthetic lethal partners are selected within the dataset and a query gene is generated by sampling from the normal distribution. These are matched (as shown for 2 synthetic lethal partners in Figure 3.7) such that the synthetic lethal condition is met: that at least one of the synthetic partner genes and the query gene are functional in any particular cell. The samples are ordered by functional data (without assuming correlation of underyling expression values) with the query gene in one direction and the remaining dataset ordered by the selected synthetic lethal partner.

This results a simulated dataset where samples with non-functional query gene have at least one functional partner gene. Similarly, the query gene is functional in all samples where all of the synthetic lethal partner genes are non-functional. There-

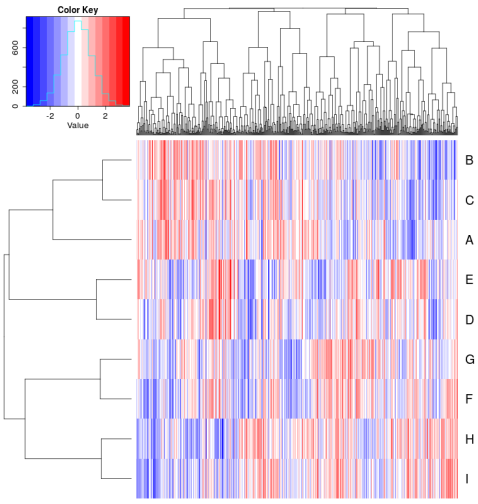


(a) Simulated gene function with SL genes

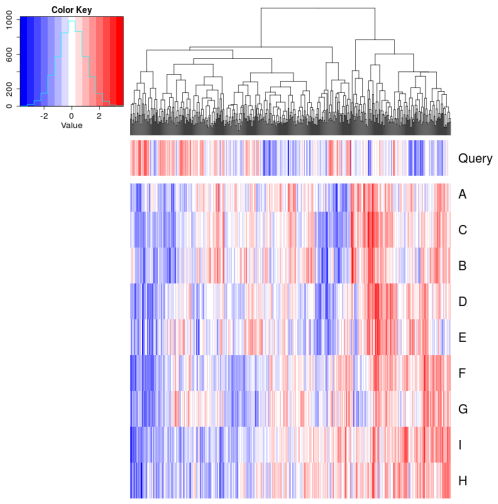


(b) Query gene added with SL condition

Figure 3.7: **Simulating synthetic lethal gene function.** In a discrete simulated gene function dataset (shaded for functional levels and pale otherwise) with samples (columns) and genes (rows), genes A and I are SL partners of a "Query" gene. A partner is selected (highlighted in green) randomly in each sample for simulating synthetic lethality, then ordered such that the query gene or an SL partner are functional in each sample.



(a) Initial expression matrix



(b) Simulated synthetic lethal dataset

Figure 3.8: **Simulating synthetic lethal gene expression.** A simluated continuous expression dataset (blue–red scale) with samples (columns) and genes (rows) is matched to a query gene such that at least one synthetic lethal partner is above a functional threshold when the query gene is below it satisfying the synthetic lethal model.

fore a dataset has been generated with known synthetic lethal partners (see Figure 3.8) by as few assumptions about the relationships between the each synthetic lethal pair as possible (and allowing compensating functions from higher-order interactions). This has been designed to have the most stringent (least detectable) synthetic lethal relationships where higher-order interactions are possible for the purposes of testing pairwise detection procedures such as SLIPT.

3.3 Detecting Simulated Synthetic Lethal Partners

The synthetic lethal detection methodology (SLIPT), as described in section 3.1, was tested on simulated data with known synthetic lethal partners, generated using the procedure described in section 3.2.2. This section will present basic simulations to demonstrate the methodology and support it's use throughout this thesis. These will be performed with sampling from basic statistical distributions as described, including multivariate normal distribution with correlated blocks of genes, with the Σ matrix show in the plots where relevant. A more complex multivariate normal sampling procedure based on pathway graph structures, as described in section 3.4.2, will be applied in Chapter 6.

3.3.1 Binomial Simulation of Synthetic lethality

A previous version of the synthetic lethal simulation procedure (described in section 3.2.2), used gene function sampled directly from a binomial distribution using the binomial probability of observing functional gene levels ($p = 0.3$) in one observation ($n = 1$) for each samples:

$$X \sim Bin(n, p)$$

Once a query gene consistent with synthetic lethality has been added, these functional levels were passed directly into SLIPT as "low" and "high" categories.

The simulation procedure was performed with 20,000 total genes (as feasible in the human genome and expression datasets) with a variable number of true synthetic lethal partners and sample sizes of 500, 1000, 2000, and 5000. Each ROC curve was derived from the results of 10,000 replicate simulations. The statistical performance (as shown in Figure 3.9) of such an approach based on the χ^2 p-value declines towards random predictions (an AUROC of 0.5) with an increasing number of underlying true synthetic lethal partners to detect. However, increased sample size mitigates this decline to some

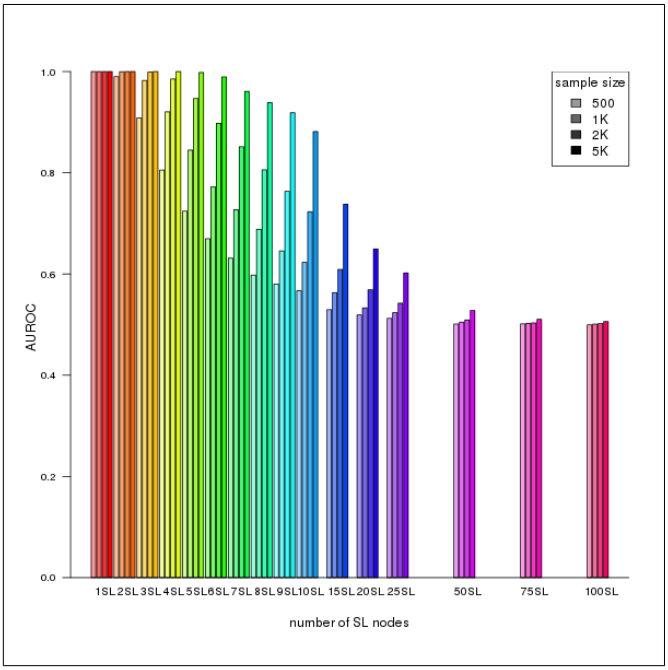


Figure 3.9: **Performance of binomial simulations.** Gene function was simulated by binomial sampling and tested for synthetic lethal genes. Statistical performance declines with additional known synthetic partners but this is mitigated by increased sample sizes.

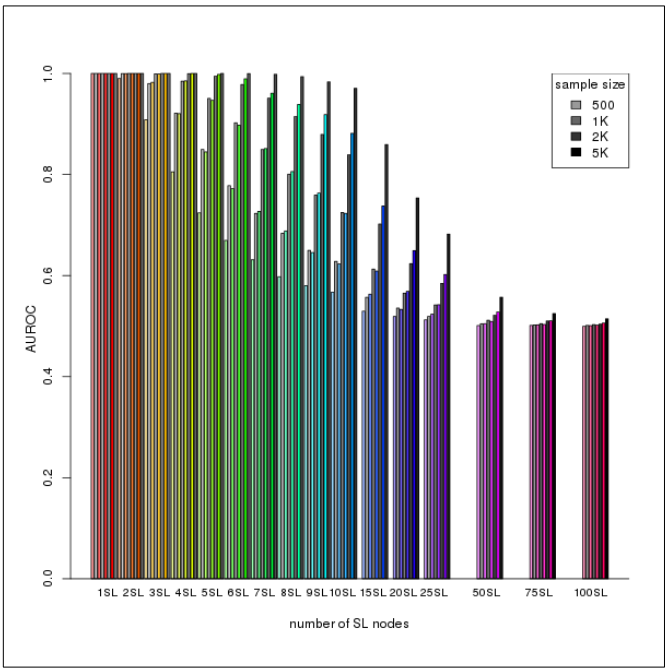


Figure 3.10: **Comparison of statistical performance.** Binomial simulation of synthetic lethality (in colour) is compared (in greyscale) to multivariate normal simulations (detailed below) which consistently outperforms binomial simulation across parameters.

extent, as expected with a statistical predictor, particularly for moderate numbers of synthetic lethal partners.

Simulations based on a simple binomial model of synthetic lethality are limited but form a basis for builing a more complex model including expression and correlation structures. While this does not represent the data that SLIPT will be applied to, binomial simulations do demonstrate that SLIPT is able to distinguish small numbers of synthetic lethal partners in a simplistic simulated system with behavious expected with resepect to sample size. This supported further development of the synthetic lethal model and simulation pipeline (as described in section 3.2) using the multivariate normal distribution.

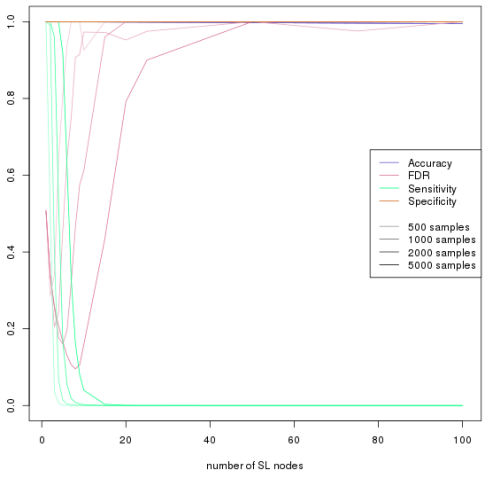
The multivariate normal simulation procedure is more representative of the (normalised) expression data SLIPT is intended for and enables the prediction procedure to be tested without changes to the methodology (presented in more detail in section 3.3.2). Sampling continuous expression values from a normal distribution allows the expression threshold for gene function to differ from the categorical "low" and "high" expression binning performed by SLIPT (as discussed in section 3.2.1) which represents that the SLIPT procedure does not assume a known threshold for expression but rather uses expression as an estimate of gene function. This functionality can be included in the multivariate normal simulation without compromising the statistical performance of the SLIPT, rather the performance estimates (shown in Figure 3.10) were a marked improvement over the binomial simulation procedure across simulation parameters in an equivalent simulation (without correlation structure). This imprivement may be due to binomial model defining the synthetic lethal condition in a way that, while ensuring at least on synthetic lethal partner is active in query deficient samples, disrupts the number of samples with functional synthetic lethal genes compared to other genes affecting the expected sample proportions of χ^2 test.

3.3.2 Multivariate Normal Simulation of Synthetic lethality

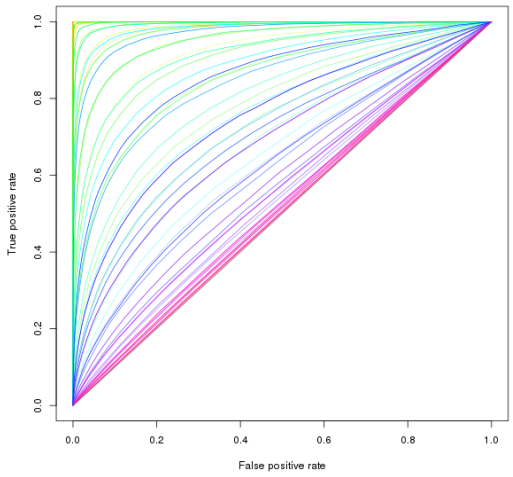
The multivariate normal simulation procedure was initially performed using the `mvtnorm` R package (Genz and Bretz, 2009; Genz *et al.*, 2016) (as described in section 3.2) without correlation structure.

Expression is sampled from multivariate normal distribution with a mean ($\mu = 0$), standard deviation ($\sigma = 1$), and no correlation between genes ($r = 0$):

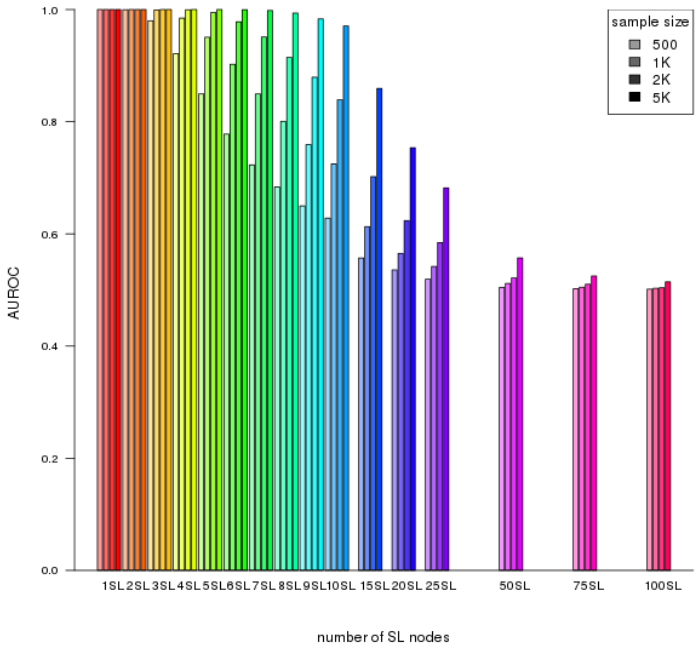
$$X \sim N(\bar{\mu}, \Sigma)$$



(a) Statistical evaluation



(b) Receiver operating characteristic



(c) Statistical performance

Figure 3.11: **Performance of multivariate normal simulations.** Simulation of synthetic lethality was performed sampling from a multivariate normal distribution (without correlation structure). Performance of SLIPT declines for more synthetic partners but this is mitigated by increased sample sizes (in darker colours). This generally occurs as the sensitivity decreases for a greater number of true positives to detect, leading to a trade off in accuracy as seen in a trough for false discovery rate and the ROC curves.

Once a query gene consistent with synthetic lethality has been added, the simulated expression values are tested by SLIPT exactly as described in section 3.1.

As shown in Figure 3.11(a), the statistical accuracy of SLIPT as a binary classifier is considerably high across simulations of a full human dataset of 20,000 genes. However, with the χ^2 p-value as a threshold for prediction, this is largely to desirable specificity: the majority of non-SL genes are distinguished from the few underlying synthetic lethal genes. In this regard, the SLIPT methodology generally performs better with larger datasets with more expected negatives and thus the results of simulations of smaller numbers of genes (such as the graph structures analysed in Chapter 6) can be applied to larger datasets where they are expected to perform comparably or better with a lower false negative rate. Accordingly, key results will be supported by replication with larger numbers of non-SL genes added to the simulations.

However, with higher numbers of synthetic lethal genes to detect, the sensitivity (in Figure 3.11(a)) of SLIPT as a binary classifier of synthetic lethality declines, although this is somewhat mitigated by higher sample sizes (shown in darker colours). Thus the minority of true synthetic lethal partners are more difficult to distinguish when there are more of them (and a weaker expression signal from each). While a reasonable reduction of the false discovery rate can be achieved for moderate numbers of underlying synthetic lethal partners, we can not be sure how many partners are expected to be detected in analyses of expression data. However this simulation procedure is amenable to assessing the performance of SLIPT across simulation parameters, graph structures and comparisons to other approaches (presented in more detail in Chapter 6).

Not all of the genes detected by SLIPT will be true synthetic lethals but these will be among the strongest candidates and it performs better with fewer underlying synthetic lethals to detect. This supports a focus on pathway analyses, in particular detecting pathways for further investigation. Since individually gene candidates are not necessarily gene synthetic lethal themselves, pathway over-representation analysis will be performed to detect functional groups recurrently detected by SLIPT as these detection of functionally related genes further support their role in synthetic lethal relationships in addition to being biologically informative. Alternatively, pathway meta-genes will reduce the number of underlying synthetic lethals to identify synthetic lethal pathways. Both of these approaches will be applied in Chapter 4 to identify and replicate synthetic pathways of *CDH1*. Pathways are also more likely to replicate across experimental models as demonstrated by Dixon *et al.* (2008).

The receiver operating characteristic curves (in Figure 3.11(b)) demonstrate that

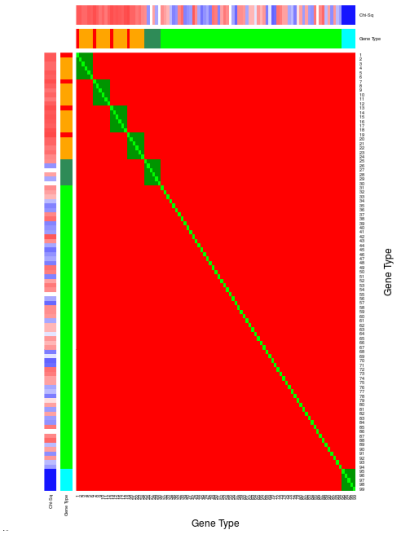
SLIPT is subject to near equal trade-off between sensitivity and specificity across threshold values. The lower sensitivity and higher specificity with a binary classifiation (in Figure 3.11(a)) stems from stringent testing by SLIPT with (FDR) p-values adjusted for multiple tests. The area under these curves is also used to compare statistical performce (in Figure 3.11(c)), with declining performance across increased underlying synthetic lethal partners and increased performance with sample size in multivariate normal simulations.

3.3.2.1 Multivariate Normal Simulation with Correlated Genes

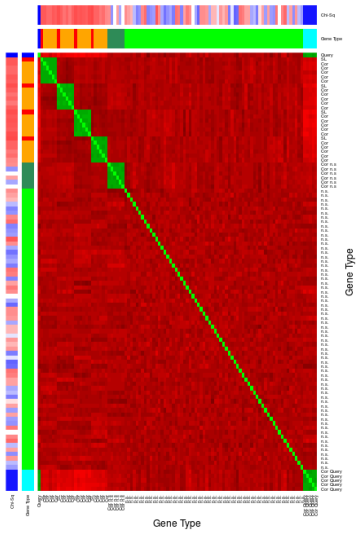
Correlation structures can be added to the simulation procedure (as discussed in section 3.2), starting with simple correlated blocks of genes as the Σ parameter depicted in Figure 3.12(a). These correlated blocks represent genes with correlated expression such as that expected by coregulation or biological pathways. Figure 3.12 gives an example of 4 synthetic lethal genes (out of 100), each with 5 correlated genes that are not themselves synthetic lethal partners of the query gene. This serves to test whether synthetic lethal genes are distinguishable from correlated partners. This Σ matrix produces a similar correlation structure (Figure 3.12(b)) in the resulting expression profiles (Figure 3.12(c)) where apart from correlated blocks of genes ($r = 0.8$), the remaining genes have only slight variations due to random sampling. The structure of the dataset, particularly between synthetic lethal genes and the query, is shown at the gene expression (Figure 3.12(c)) and function (Figure 3.12(d)). These are ordered by the SLIPT results and the synthetic lethal genes are ranked high, with the majority of them being distinguishable from highly correlated genes.

The use of correlation structures generalises to larger datasets, such as 1000 genes shown in Figure 3.13. Synthetic lethal genes are highly ranked by SLIPT and still largely distinguishable from correlated genes. As previously discussed in section 3.3.2, these synthetic lethal genes are still detectable among a larger number of true negatives and the SLIPT methodology performs better on such datasets.

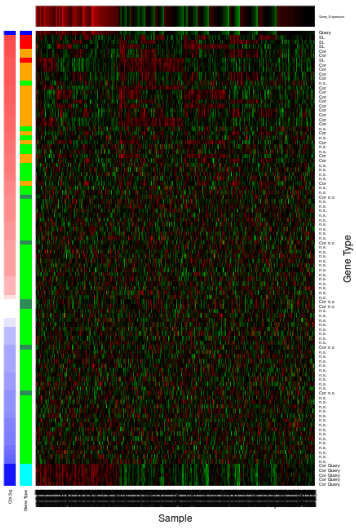
These plots (Figures 3.12 and 3.13) also show similar correlated blocks with a non-synthetic lethal gene (true negative) and the query gene (which is not synthetic lethal with itself). Neither of these should be synthetic lethal (or detected to be) but they may impact upon the performance of the model, particularly the specificity as correlated negative genes may be distinguishable from true synthetic lethals. The non-synthetic lethal correlated block has no impact on synthetic lethal detection but the impact of query correlated genes will be discussed in section 3.3.2.2 and Chapter 6.



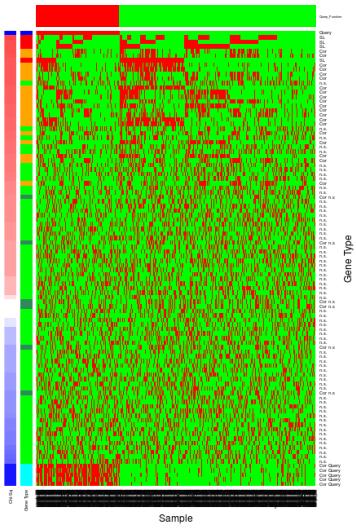
(a) Input Σ matrix parameter



(b) Simulated correlation matrix

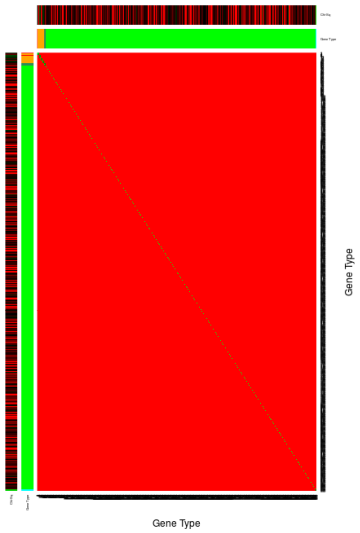


(c) Simulated gene expression

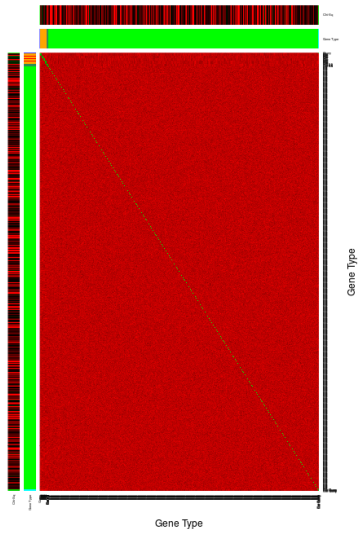


(d) Simulated gene function

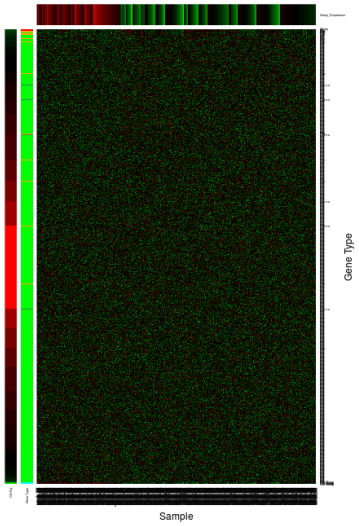
Figure 3.12: **Simulating expression with correlated gene blocks.** A Σ matrix (a) is used generate a multivariate normal distribution with of 100 genes correlated blocks of genes (correlated by 0.8) with a comparable structure (b) to the input Σ, as shown by correlation on a red–green scale. The annotation bars for genes give the χ^2 (in blue if the direction of SLIPT is met or red otherwise) and the gene category (blue for query, cyan for query-correlated, red for SL, orange for SL-correlated, forest green for non-SL-correlated, and green for non-SL). The simulated gene expression (c) and function (d) generated are ordered by χ^2 showing the functional structure of synthetic lethal genes and that they are among the strongest SLIPT results.



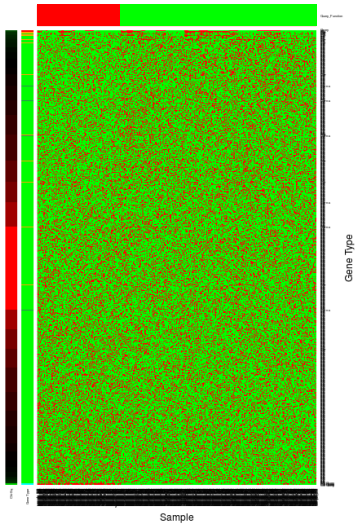
(a) Input Σ matrix parameter



(b) Simulated correlation matrix

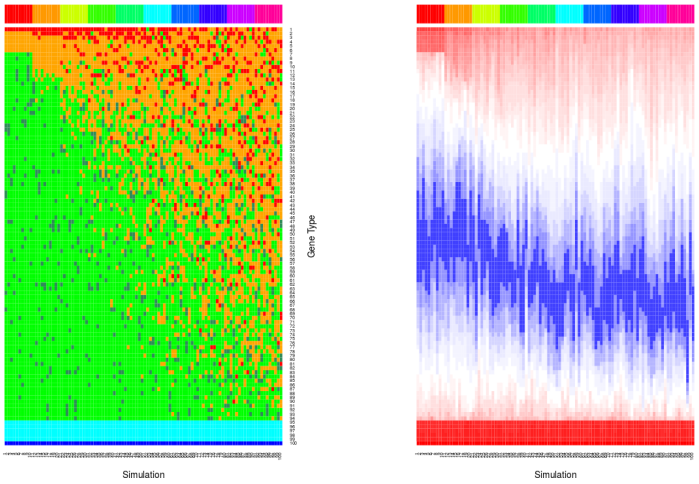


(c) Simulated gene expression



(d) Simulated gene function

Figure 3.13: **Simulating expression with correlated gene blocks.** Using the (a) Σ matrix, sampling from a multivariate normal distribution with of 1000 genes produced (b) correlated blocks of genes (correlated by 0.8) on a red–green scale. The simulated gene expression (c) and function (d) generated are ordered by χ^2 and SLIPT direction show that synthetic lethal genes are among the strongest SLIPT results with high specficity against many potential false positives. These are annotated for χ^2 (on a red–green scale) and category (blue for query, cyan for query-correlated, red for SL, orange for SL-correlated, forest green for non-SL-correlated, and green for non-SL) for each gene.

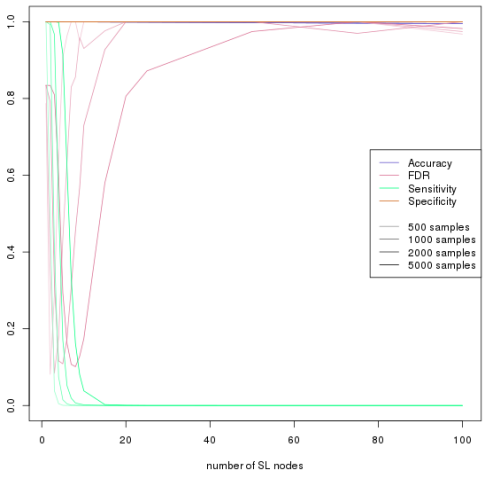


(a) Gene category in simulations (b) Corresponding χ^2 values

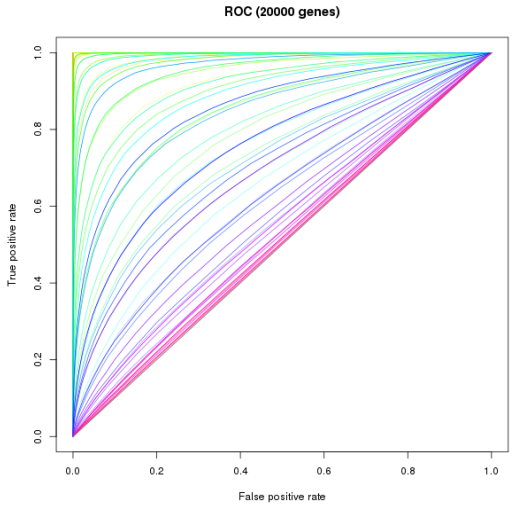
Figure 3.14: **Synthetic lethal prediction across simulations.** The gene category (blue for query, cyan for query-correlated, red for SL, orange for SL-correlated, forest green for non-SL-correlated, and green for non-SL) ordered by χ^2 signed by the SLIPT directional condition is shown across simulations. For each of 1–10 SL partners, 10 simulations demonstrate that the increasing numbers of SL partners become harder detect. The χ^2 values show a clear threshold for SL and correlated genes when there are fewer of them, distinguishable from correlated genes in this case.

These simulations (on 100 genes) were repeated to examine the variation between detection on different samples and varying the number of underlying synthetic lethal partners, in simulated gene expression data with correlations structure. A small nuber (10 for each) simulations are shown in Figure 3.14 to demonstrate the variation between replicate simulations, with iterative sampling from the same multivariate normal distribution. These simulations show synthetic lethal genes are not only highly ranked by SLIPT when there are few of them but also that they are fairly consistent across replicate simulations. Whereas they become less consistent for increasing numbers of true synthetic lethal partners to detect and thus more difficult to distinguish from other geners, particularly those correlated with them. Similarly, the χ^2 values show a marked stepwise increase with clear thresholds for SL and correlated genes in simple simulations, whereas these become less evident for higher numbers of SL partners.

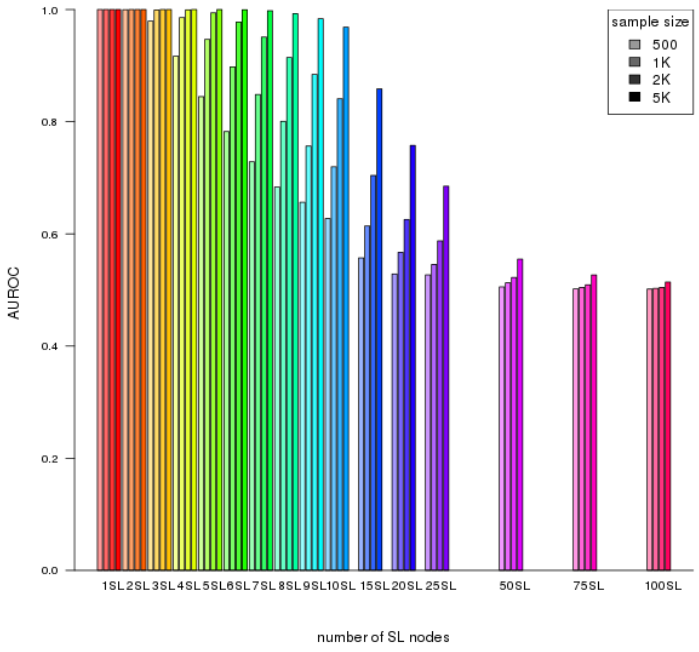
Whether the synthetic lethal genes detected in simple simulations (in Figure 3.14) are robustly detectable across greater number of simulations, in addition to further



(a) Statistical evaluation



(b) Receiver operating characteristic



(c) Statistical performance

Figure 3.15: **Performance with correlations.** Simulation of synthetic lethality was performed sampling from a multivariate normal distribution (with correlation structure). Performance of SLIPT declines for more synthetic partners but this is mitigated by increased sample sizes (darker colours). This generally occurs as the sensitivity decreases for a greater number of true positives to detect, leading to a trade off in accuracy as seen in a trough for false discovery rate and the ROC curves.

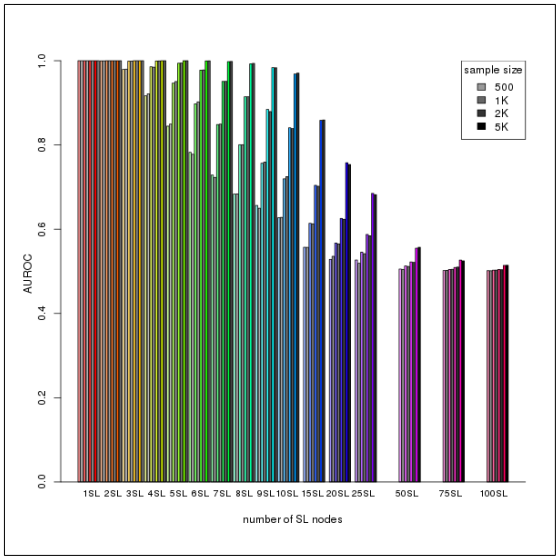
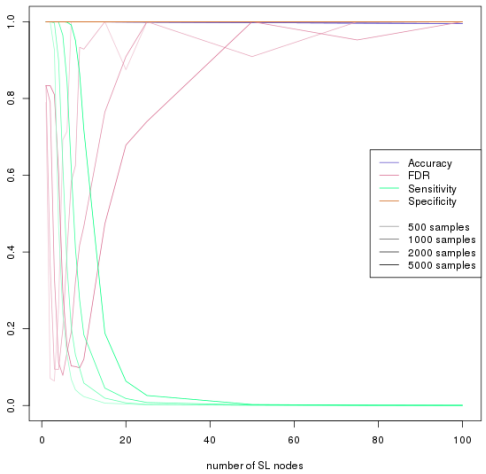


Figure 3.16: **Comparison of statistical performance with correlation structure.** Multivariate simulation of synthetic lethality with correlation structure (in colour) has comparable performance to simulation without correlations (in greyscale) with known synthetic partners across parameters.

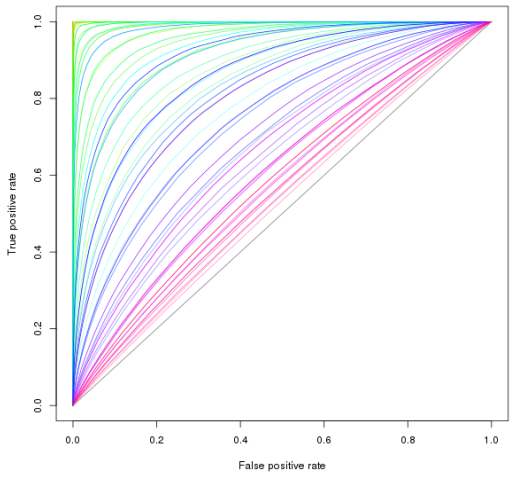
comparions, was tested with a supporting ROC analysis. These results (in Figure 3.15) are very similar to simulations without correlation structure, with SLIPT as a binary classifier having a poor sensitivity with increasing numbers of synthetic lethal partners to detect but high specificity in a total of 20,000 genes with the vast majority being true negatives. This is reflected in a similar decline in statistical performance for increasing numbers of synthetic lethal partners and a compensating increase in performance with higher sample size. Overall, the statistical performance is very similar to simulations without correlation structure (as shown in Figure 3.16).

Thus SLIPT is robust across correlation structures and applicable to real gene expresssion data where pathway structures and correlations are a consideration. These correlation structures are not intended to model specific biological pathways or represent them, rather they serve to test the impact of correlation structure on the performance of SLIPT with an extreme example of closely correlated ($r = 0.8$) gene blocks. More complex correlation structures, such as genes postively correlated with the query gene and derived from pathway graph structures (as described in 3.4.2) will be examined below (in section 3.3.2.2) and in Chapter 6 respectively.

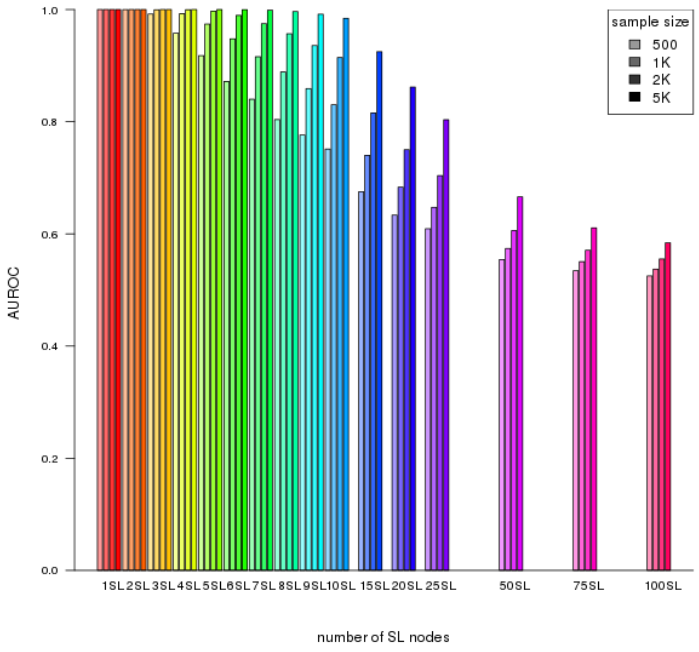
In particular, genes correlated with true synthetic lethal genes have little impact on the performance of SLIPT detection: synthetic lethal genes are as distinguishable



(a) Statistical evaluation



(b) Receiver operating characteristic



(c) Statistical performance

Figure 3.17: **Performance with query correlations.** Simulation of synthetic lethality was performed sampling from a multivariate normal distribution (with correlation structure including correlated genes with non-SL and query genes). As before, performance of SLIPT declines for more synthetic partners and is mitigated by increased sample sizes (darker colours)but the sensitivity remains higher for a greater number of true positives with corresponding improvements in ROC curves.

from true negative genes as without correlated genes. Synthetic lethal correlated genes will not interfere detect of true synthetic lethals, although they may be ranked next below them and be biologically informative with related gene functions.

3.3.2.2 Specificity with Query-Correlated Pathways

Another consideration for correlation structures is postively correlated genes with the query that are not synthetic lethal. As described in section 3.3.2.1, 5 highly correlated ($r = 0.8$) with the query gene were added. These simulations perform similarly to before (in Figure 3.17) with a higher specficity and a lower false discovery rate being feasible (as shown in 3.17(a)).

3.3.2.2.1 Importance of Directional Testing

It is important to notice here that the directional criteria of the SLIPT procedure is enhancing it's performance, particularly in distinguishing positively correlated true negatives. The multivariate normal simulation results, with 20,000 genes including all of the correlation structures discussed (SL, non-SL, and query correlated genes), are compared here for SLIPT with and without (χ^2) directional testing. There is a marked improvement in statistical performance with directional criteria, particularly with increased sensitivity and lower false discovery rate (as shown in Figure 3.18).

This is encouraging for the application of SLIPT to empirical expression datasets as postively correlated genes are likely to occur and the directional condition robustly improves the performance of SLIPT across simulation parameters. Without assuming the underlying number of synthetic lethal genes, SLIPT will perform better than the χ^2 test alone at detecting them. This is further supported irrespective of significance threshold for the χ^2 test by the ROC analysis in Figure 3.19. The directional SLIPT methodology outperforms the ordinary χ^2 test at detecting synthetic lethal partners with some predictive power (above random and AUROC of 0.5) even up to 100 synthetic lethal genes.

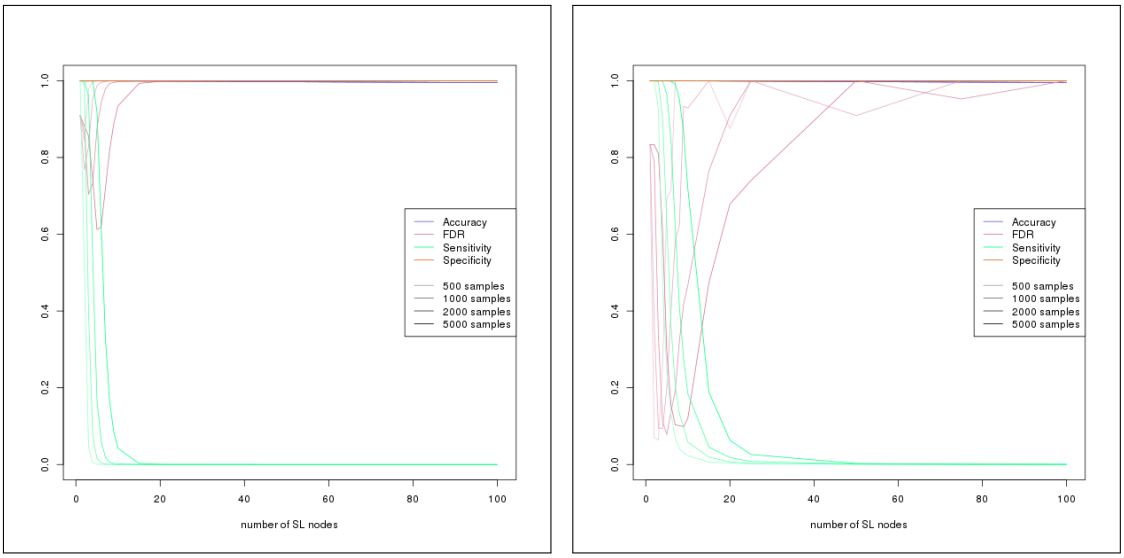
Together these simulation results support the application of the SLIPT methodology as it has been performed throughout Chapter 4 and 5. However, the methodology and simulation procedure will explored in more detail in Chapter 6, with the inclusion of graph structures and comparison to other synthetic lethal detection approaches.

3.4 Graph Structure Methods

Graph structures have been used in several ways in this project with novel approaches to analysis and simulations. Procedures were developed for statistical and network analysis of gene states in pathway structures. Specifically, the relationships between siRNA and SLIPT genes were tested within biological pathways in Chapter 5. These graph structures were also used in Chapter 6 for the simulation of synthetic lethality to derive correlation structure between simulated gene expression profiles in manner that resembles biological pathways.

3.4.1 Upstream and Downstream Gene Detection

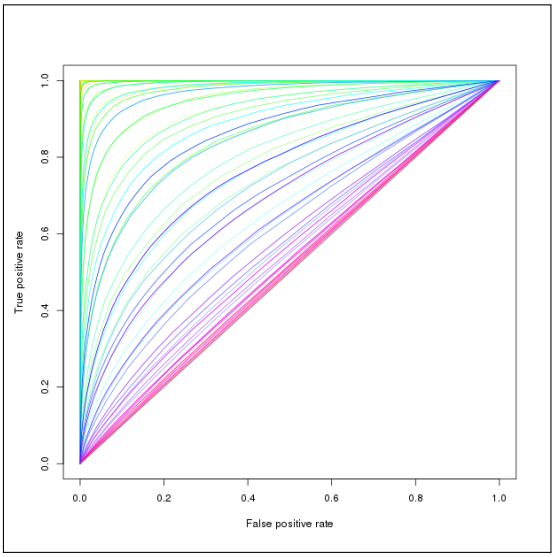
Comparison of experimental and computational candidate synthetic lethal partner genes within pathway structures arose from the hypothesis that these sets of genes were related by pathway structure. Due to differences in how these candidates were generated, it should not be expected that they detect the identical genes within the



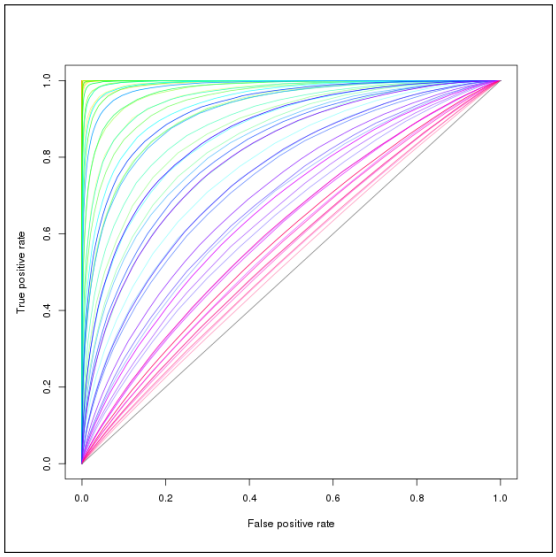
(a) χ^2 testing without direction

(b) SLIPT with directional criteria

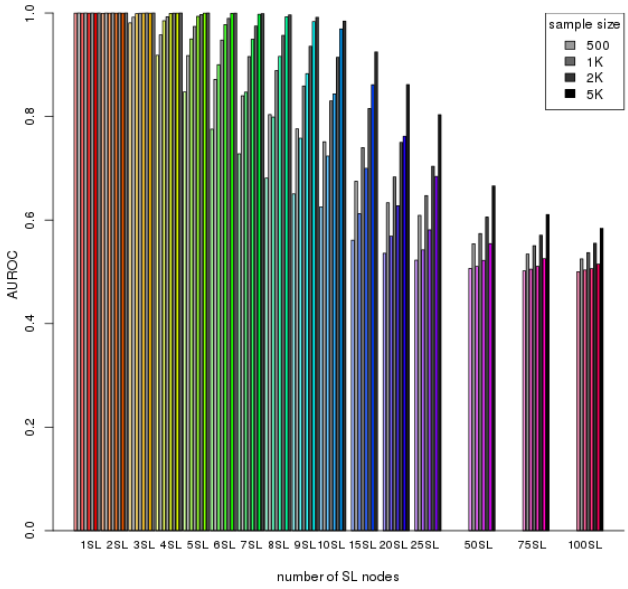
Figure 3.18: **Statistical evaluation of directional criteria.** A simulated multivariate normal dataset of 20,000 genes with correlation structures was tested by SLIPT with the directional condition and and the equivalent χ^2 test without. SLIPT exhibited a consistently higher sensitivity and lower false discovery rate.



(a) χ^2 testing without direction



(b) SLIPT with directional criteria



(c) Statistical performance

Figure 3.19: **Performance with directional criteria.** A simulated multivariate normal dataset of 20,000 genes with correlation structures was tested by SLIPT with the directional condition and and the equivalent χ^2 test without. SLIPT has higher performance across simulation parameters, clearly differing from random (grey diagonal) in ROC curves up to 100 SL genes (b). The performance (c) of SLIPT (in greyscale) was consistently higher than the χ^2 test (in color).

candidate biological pathways, rather they may be related by being upstream or downstream of each other.

Using the Reactome version 52 data (Croft *et al.*, 2014) as described in section 2.4.2, genes identified by each synthetic lethal discovery approach were mapped to the graph structure for the candidate pathways identified in Chapter 4 (with subgraphs defined as described in section 2.4.3). To test whether siRNA candidate genes were upstream of SLIPT candidate genes, shortest paths were traced between each potential pair of these genes in a directed network. The number of genes where the siRNA candidate was upstream were scored "up" and where the siRNA candidate was downstream were scored "down". This procedure enabled counting the total number of shortest paths which supported siRNA genes being upstream or downstream of the SLIPT genes and measuring the difference between these to determine if there is an imbalance in a particular direction. While this difference is indicative of the number of paths between the gene candidate groups in either direction, alone it is not sufficient to statistically support structure or relationships between siRNA and SLIPT genes. However, it may be combined with a permutation resampling procedure (as described in section 3.4.1.1) to test for directional relationships in either direction.

The original version of this procedure excluded gene detected by both approaches since they would count in both directions. Upon further consideration, the intersection genes were restored to being accounted for by the shortest paths counts since they may count unequally to being upstream or downstream of each gene set if there are unequal numbers above or below them in the pathway structure.

3.4.1.1 Permutation Analysis for Statistical Significance

A permutation procedure was developed to randomly assign members of the pathway to siRNA and/or SLIPT groups, with the same number of each candidate partner gene set as observed in the pathway. These permuted genes are measured for pathway structure between the permuted gene groups as performed for the observed candidates (as performed in section 3.4.1). A distribution of pathway structure relationships expected by chance is generated by permuting iteratively over these pathways. This null distribution can be compared to the observed counts of relationships (in either direction), which yields a permutation p-value as the proportion of permutations in which had value or greater or more extreme magnitude than the observed value.

The null hypothesis is that there is no relationship between these gene groups that would not have occurred had the genes been selected at random. Thus we can test

both the alternate hypothesis that the siRNA genes were upstream of the SLIPT genes or that they are downstream of them.

The permutation procedure does not assume the underlying distribution of the data under the null hypothesis and accounts for the total number of nodes, edges, siRNA, and SLIPT genes in each pathway network structure. The intersection size of the siRNA and SLIPT genes was originally not accounted for under the shortest path counts procedure that excluded them. A refined version of this procedure ensured that the number of intersecting genes was equal to the number observed to test for pathway structure without changing the intersection size, the subject of prior analyses.

3.4.1.2 Ranking Based on Biological Context

An alternative approach to pathway structure was performed based on the biological context that genes at the upstream and downstream ends of a pathway perform different functions, such as a kinase signalling cascade recieving signals from external stimuli and passes these on ribosomes or the nucleus. The genes were ranked to determine if genes of either candidate group (or those with stronger support for either group) performed upstream or downstream functions disproportionately.

A network-based approach was used to determine the pathway ranking of genes in a computationally rational way when applied to different biological pathways with a directed graph structure, G (without loops). The diameter of the network (i.e., the length of longest possible shortest path between the most distant genes) was used to identify a gene (z) at the downstream end of the pathway (at the end of a diameter spanning shortest path), assigned a rank of:

$$rank(z) = 1 + diameter(G)$$

Having identified the downstream end of the pathway, genes upstream (e.g., gene i) of this are assigned a rank by the length of their shortest path to this gene, z.

$$rank(i) = rank(z) - d_{iz}$$

The remaining unassigned genes (e.g., gene j) gain the rank of the length of the shortest path downstream from the nearest assigned gene if possible.

$$rank(j) = rank(i) + d_{ij}$$

This process may be performed iteratively to fill in pathway ranking but it was not necessary to perform further iterations for the candidate synthetic lethal pathways investigated (amenable to this procedure) which exhibited clear directional structure and

the small world property (with a low diameter). Thus genes in a pathway graph structure were assigned integer valued rankings upstream to downstream by this procedure:

$$rank \in \{1, 2, 3, ..., 1 + diameter(G)\}$$

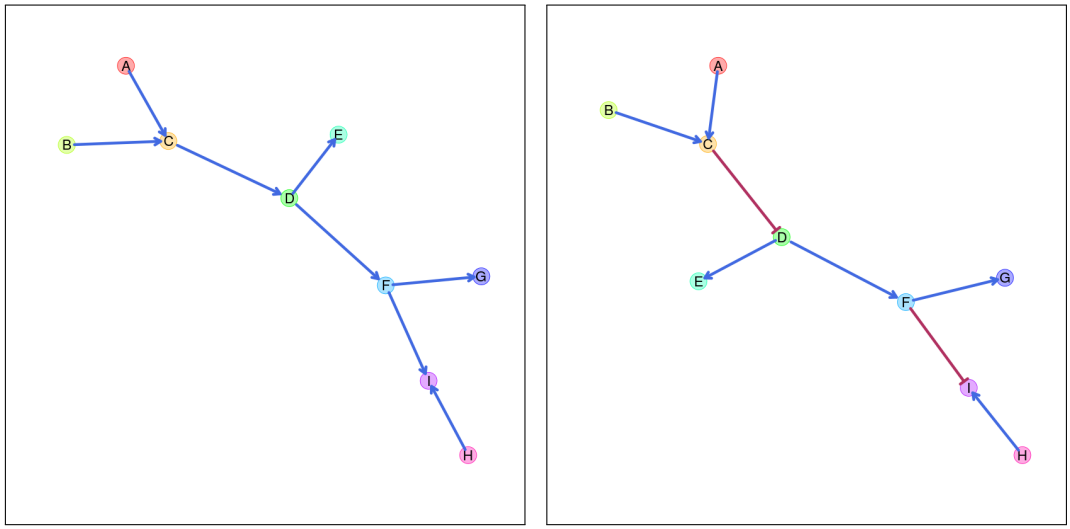
This ranking of pathway directionality can be used for comparison with measures of the number of genes of each candidate group and the support for being synthetic lethal partners with either approach.

3.4.2 Simulating Gene Expression from Graph Structures

A further refinement of the simulation procedure generated expression data with correlation structure, derived from a known graph structure. This enables modelling of synthetic lethal partners within a biological pathway and the investigation of impact of pathway structure on synthetic lethal prediction. A simulated pathway is first constructed as a graph structure, with the `igraph` R package Csardi and Nepusz (2006), with the added annotation of the state of the edges (i.e, whether they activate or inhibit downstream pathway members). This simulation procedure was intended for biological pathway members with correlated gene expression (higher than the background of genes in other pathways) but it may also be applicable to modelling protein levels (in a kinase regulation cascade) or substrates and products (in a metabolic pathway).

First, the graph structure is constructed for simulated data to be generated from (by sampling from a multivariate normal distribution using the `mvtnorm` R package (Genz and Bretz, 2009; Genz *et al.*, 2016)). Throughout this section, the simulation procedure will be demonstrated with the relatively simple constructed graph structure shown in Figure 3.20. This graph structure visualisation was specifically developed for (directed) iGraph objects in R and has been released in the `plot.igraph` package and `igraph.extensions` library (see Table 2.5 and section 3.5.3). The `plot_directed` function allows customisation of plot parameters for each node or edge and mixed (directed) edge types for indicating activation or inhibition. These inhibition links (which often occur in biological pathways) are demonstrated in Figure 3.20(b).

The simulation procedure is designed to use such graph structures to inform development of a "Sigma" variance-covariance matrix (Σ) for sampling from a multivariate normal distribution (using the `mvtnorm` R package). Given a graph structure (or adjacency matrix), such as Figure 3.21(a), a relation matrix is calculated based on distance such that nearer nodes are given higher weight than farther nodes. For the purposes of this thesis a geometrically decreasing (relative) distance weighting is used, with each

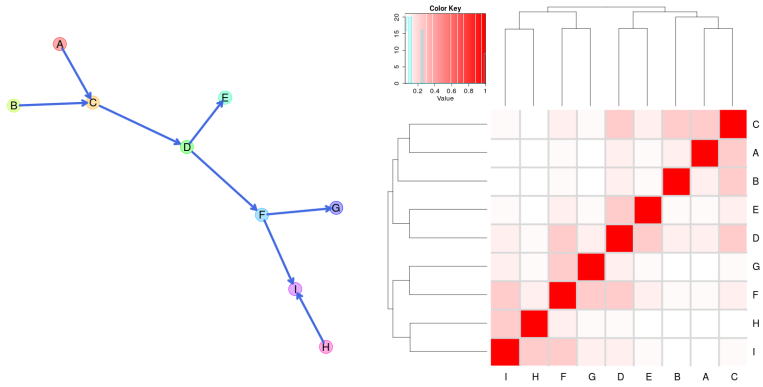


(a) Activating pathway structure (b) Pathway structure including inhibitions

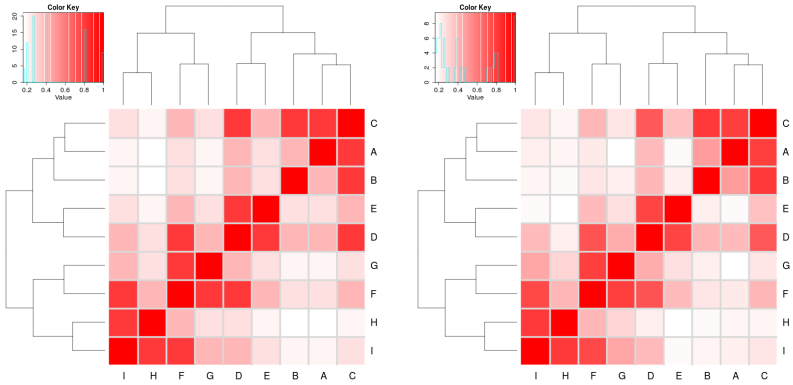
Figure 3.20: **Simulated graph structures.** A constructed graph structure used as an example to demonstrate the simulation procedure. Activating links are denoted by blue arrows and inhibiting links by red edges.

more distant node being related by $1/2$ compared to the next nearest as shown in Figure 3.21(b). However, an arithmetically decreasing (absolute) distance weighting is also available in the `graphsim` R package release of this procedure.

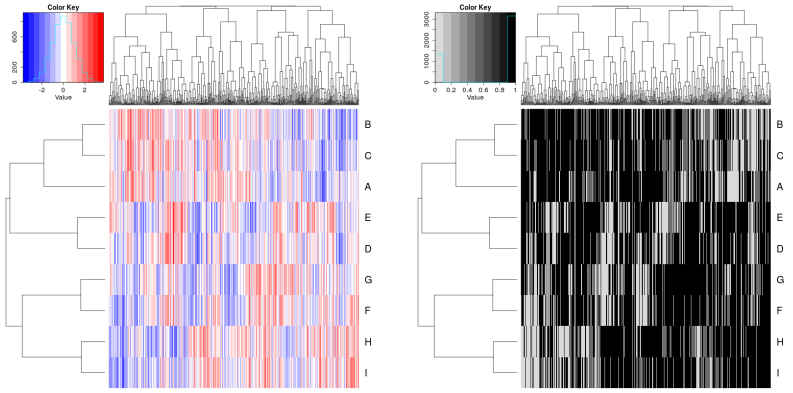
A Σ matrix is derived from this distance weighting matrix, creating a matrix (with a diagonal of 1) where each node has a variance and standard deviation of 1. Thus covariances between adjacent nodes are assigned by a correlation parameter and the remaining matrix based on weighting these correlations with by the distance matrix (or the nearest "positive definite" matrix). For the purposes of this thesis, the correlation parameter is 0.8 unless otherwise specified (as used for the example in Figure 3.21(c)). This Σ matrix is used to sample from a multivariate normal distribution with each gene having a mean of 0, standard deviation 1, and covariance within the range $[0, 1]$ such that they are correlations. This procedure generates a simulated (continuous normally distributed) expression profile for each node (as shown in Figure 3.21(e)) with corresponding correlation structure (Figure 3.21(d)). The simulated correlation structure closely resembles the expected correlation structure (Sigma in 3.21(c)) even for the relatively modest sample size ($N = 100$) illustrated in 3.21. Once a simulated gene expression dataset has been generated (as in Figure 3.21(e)), then a discrete matrix of gene function can be constructed with a functional threshold quantile to



(a) Activating pathway structure (b) Distance matrix



(c) Sigma, Σ (expected correlation) (d) Simulated correlation structure



(e) Simulated expression data (f) Simulated gene function calls

Figure 3.21: **Simulating expression from a graph structure.** An example graph structure is used to derive a correlation structure from the relative distances between nodes and simulate continuous gene expression with sampling from the multivariate normal distribution.

simulate functional relationships of synthetic lethality (as shown in Figure 3.4). For the purposes of this thesis, this threshold is the 0.3 quantile (as discussed in section 3.2.1) which generates functional discrete matrices such as those used for synthetic lethal simulation in section 3.2.2 (as shown Figure 3.21(f))

The simulation procedure (depicted in Figure 3.21) is amenable to pathways containing inhibition links (as shown in Figure 3.22) with several refinements. With the inhibition links (as shown in Figure 3.22(a)), distances are calculated in the same manner as before (Figure 3.22(b)) with inhibitions accounted for by iteratively multiplying downstream nodes by -1 to form blocks of negative correlations (as shown in Figures 3.22(c) and 3.22(d)). As before, a multivariate normal distribution with these negative correlations can be sampled to generate simulated data (as shown in Figures 3.22(e) and 3.22(f)).

These simulated datasets are amenable to simulating synthetic lethal partners of a query gene within a graph network. The query gene is assumed to be separate from the graph network pathway and is added to the dataset using the procedure in Section 3.2.2. Thus we can simulate known synthetic lethal partner genes within a synthetic lethal partner pathway structure.

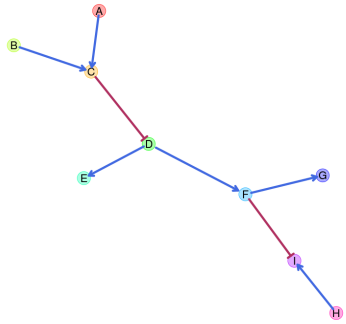
3.5 Customised Functions and Packages Developed

[Move to Appendix?]

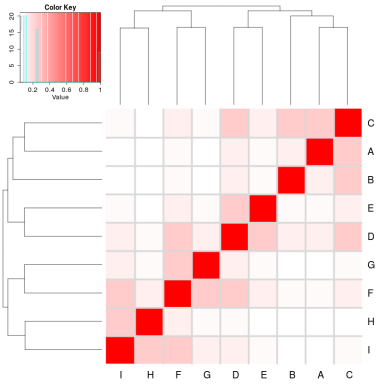
Various R packages have been developed throughout this thesis using `devtools` (Wickham and Chang, 2016) and `roxygen` (Wickham *et al.*, 2017) to enable reproducibility of customised analysis and visualisation. Many of these have the added benefit of the functions being documented, demonstrated in example vignettes, and released on GitHub to enable the research community to access utilise them in their own analysis. These are summarised in Table 2.5, along with the corresponding urls for their GitHub repository which contains a README file with instructions for installation with the `devtools` R package (Wickham and Chang, 2016) and links to the relevant vignette(s) where available.

3.5.1 Synthetic Lethal Interaction Prediction Tool

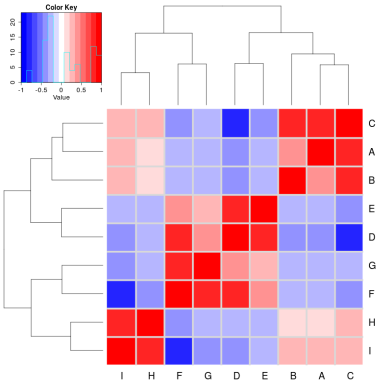
The statistical methodology for detection of synthetic lethality in gene expression data (SLIPT) is one of the main novel procedures developed in this thesis, as described in



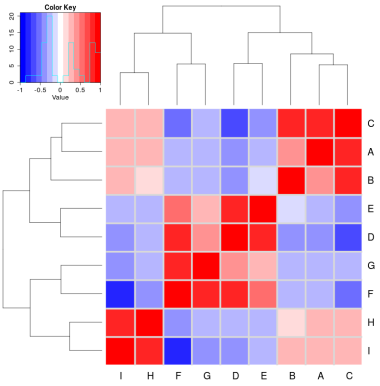
(a) Pathway structure with inhibition



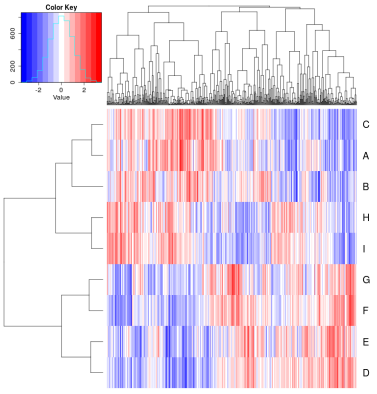
(b) Distance matrix



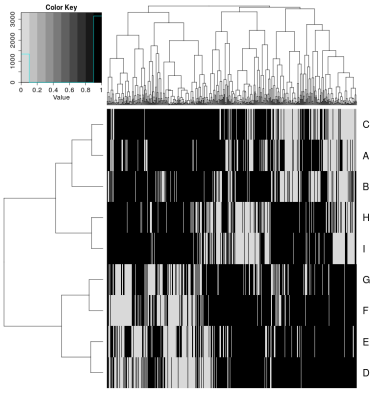
(c) Sigma, Σ (expected correlation)



(d) Simulated correlation structure



(e) Simulated expression data



(f) Simulated gene function calls

Figure 3.22: **Simulating expression from graph structure with inhibitions.** An example graph structure is used to derive a correlation structure from the relative distances between nodes and simulate continuous gene expression with sampling from the multivariate normal distribution.

section 3.1. The `slipt` R package has been prepared for release to accompany a publication demonstrating the applications of the methodology for identifying candidate interacting genes and pathways with *CDH1* in breast cancer (TCGA, 2012).

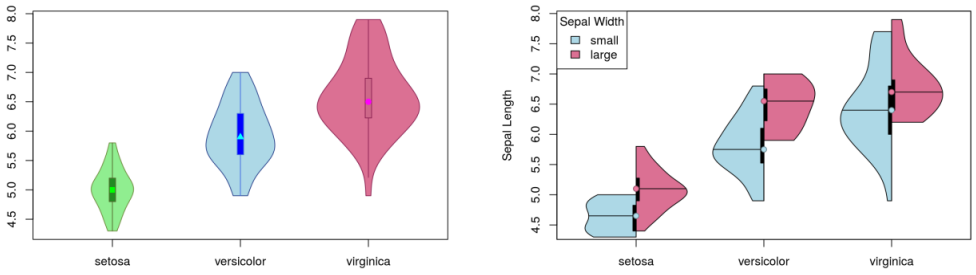
SLIPT is amenable to analysis of any effectively continuous measure of gene activity (e.g., microarray, RNA-Seq, protein abundance, or pathway metagenes). Executing `slipt` is straightforward: the `prep_data_for_SL` function scores samples as "low", "medium", or "high" for each gene, then the `detect_SL` function tests a given query gene against all potential partners by performing the chi-squared test and directional conditions. This function returns a table summarising the observed and expected sample numbers used for the directional criteria, the χ^2 values, and corresponding p-values including adjusting for multiple comparisons. The `count_of_SL` and `table_of_SL` functions serve to facilitate summary and extraction of the positive SLIPT hits, respectively, from the table of predictions of synthetic lethal partners.

The SLIPT methodology in this package release has been used in later analyses rather than the corresponding source R code, including use on remote machines and upon simulated data. In particular, the functions in the package facilitate alterations to parameters, such as the proportion of samples called as exhibiting low or high gene activity. This release support reproducibility and enables wider use of SLIPT in future investigations into other disease genes.

3.5.2 Data Visualisation

Customisations to existing data visualisations in R have been developed to present data throughout this thesis. The `vioplotx` and `heatmap.2x` packages are enhancements of the `vioplot` package (Adler, 2005) and `heatmap.2` provided by the `gplots` package (Warnes *et al.*, 2015).

The `vioplotx` package provides an alternative visualisation (of continuous variables against categories) to the more familiar boxplot, showing variability of the data by the width of the plots. As demonstrated in Figure 3.23, the customised version enables separate plotting parameters for each violin with vector inputs for colour, shape, and size of various elements of the median point, central boxplot, borders, and fill colour for the violin. Scaling violin width to adjust violin area and splitting data by a second categorical variable is also enabled. This function is intended to be backwards compatible with the inputs of `vioplot` (applying scalar inputs across all violins) and `boxplot` (by enabling formula inputs as an S3 method). Each of these features is demonstrated with examples in respective vignettes on the package GitHub repository.



(a) Customised violin plot

(b) Split violin plot

Figure 3.23: **Demonstration of violin plots with custom features.** An example of the `iris` dataset is plotted to show the custom features of the `vioplotx` package including a) individual colour, shape and size parameters of each violin, scaling violin widths by area, and b) splitting violins to compare subsets of data.

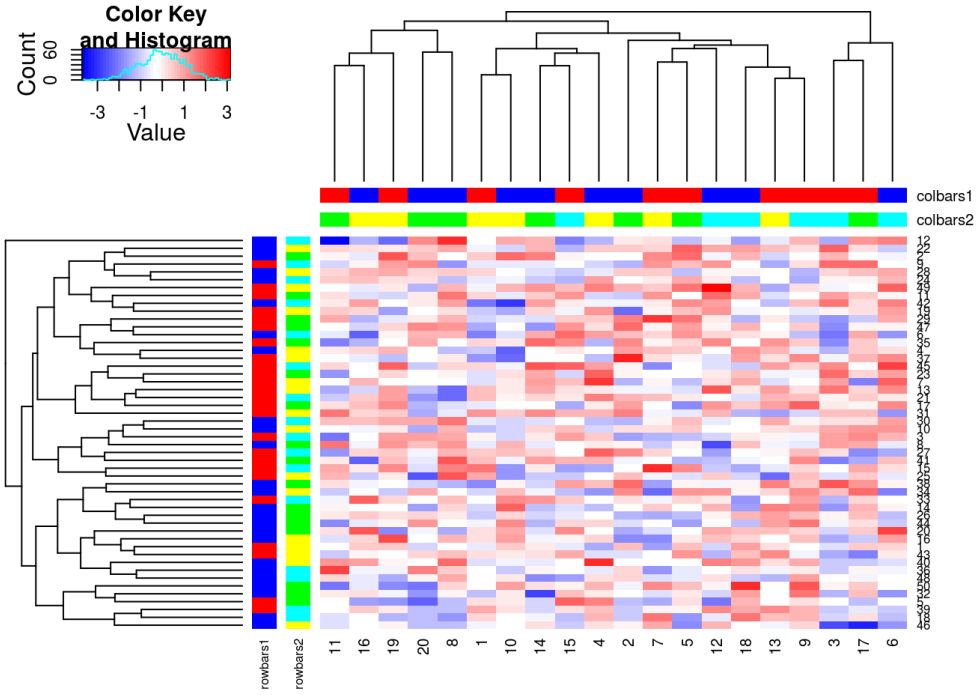


Figure 3.24: **Demonstration of annotated heatmap**. The example heatmap depicts the additional row and column annotation bars enabled by `heatmap.2x`, extending the features of `gplots` with backwards compatible inputs.

The `heatmap.2x` provides extensions for annotation colour bars for both the rows and columns (as shown in Figure 3.24). Multiple bars are enabled on both axes with matrix inputs (rather than single vector for `heatmap.2`) which facilitates additional plotting of gene and sample characteristics for comparison with correlation matrices, expression profiles, or pathway metagenes. Annotation bar inputs correspond to their orientation on the plot, each colour bar is provided as a column for the row annotation on the left of the heatmap and as a row for the column annotation on top of the heatmap. Row and column annotation bars are labelled with the column or row names respectively. Additional parameters enable resizing of these annotation bar labels and control of reordering columns for if samples are ordered in advance (e.g., ranked by a metagene or split into groups clustered separately). These features were used through this thesis and are provided in a package GitHub repository.

3.5.3 Extensions to the iGraph Package

The following features were developed during this thesis using "iGraph" data objects, building upon the `igraph` package (Csardi and Nepusz, 2006). These have been released as separate packages for each respective procedure and can be installed together as a collection of extensions to the `igraph` package.

3.5.3.1 Sampling Simulated Data from Graph Structures

The `graphsim` package implements the procedure for simulating gene expression from graph structures (as described in section 3.4.2). By default, this derives a matrix with a geometrically decreasing weighting by distance (by shortest paths) between each pair of nodes with. An absolute decreasing weighting is also available with the option of to derive correlation structures from adjacency matrices or the number of links common partners (i.e., size of the shared "neighbourhood" (Hell, 1976)) between each pair of nodes. Functions to compute these are called directly by passing parameters to them when running the `generate_expression` or `make_sigma_mat` commands. This package enables simulating expression data directly from a graph structure (with the intermediate steps automated) or generating Σ parameters for `mvtnorm` from graph structures or matrices derived from them. These functions support assigning activating or inhibiting to each edge (with a `state` parameter).

3.5.3.2 Plotting Directed Graph Structures

The `plot.igraph` package provides the `plot_directed` function specifically developed for directed graph structures and to plot activating or inhibiting for each edge (as described in section 3.4.2). As shown in Figure 3.25, this function supports separate plotting parameters for each node, node label, and edge. This includes colours of node fill, border, label text, and edges and size of nodes, edge widths, arrowhead lengths, and font size of labels. The `state` parameter for assigning activating or inhibiting to each edge determines whether edges are dipected with 30° or 90° arrowheads. Colours are assigned separately and so they may be customised. Vectorised parameters are applied across each node or edge, whereas scalar parameters apply the same plotting parameters across them. The default layout function is `layout.fruchterman.reingold` but any layout function supported by `plot` function in `igraph` (Csardi and Nepusz, 2006) is compatible such as `layout.kamada.kawai` used to implement the Kamada–Kawai algorithm (Kamada and Kawai, 1989) for graph plots throughout this thesis.

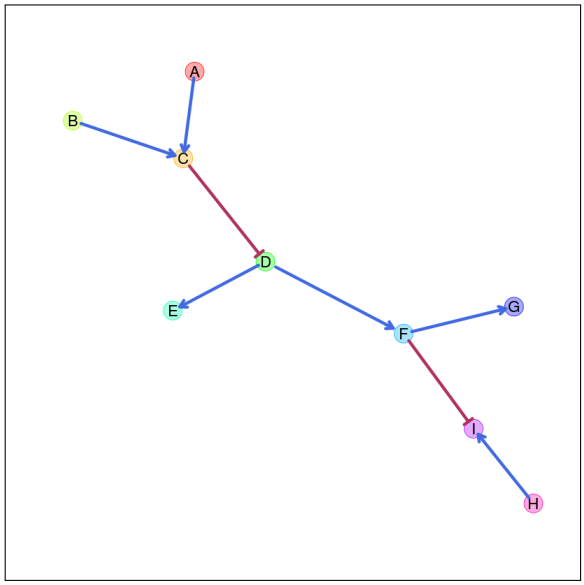


Figure 3.25: **Simulating graph structures.** An example graph structure which will be used throughout demonstrating the simulation procedure from graph structures. Here activating links are denoted by blue arrows and inhibiting links by red edges.

3.5.3.3 Computing Information Centrality

The shortest paths of a network are computed by the `igraph` package Csardi and Nepusz (2006) which can be extended to calculate the network efficiency but is not provided by the package itself (ss described in section 2.4.4). The "information centrality" of a vertex is computed as the relative change in the network efficiency when the vertex is removed. Information centrality is calculated iteratively for each node and the sum of information centrality for each vertex is the information centrality for the network. These metrics are provided by the `info.centrality` package.

3.5.3.4 Testing Pathway Structure with Permutation Testing

A network-based procedure developed was used for comparison of siRNA and SLIPT candidate genes in a pathway structure. Such pathway structure relationships were tested by computing the number of shortest paths between two different groups of nodes in either direction within a graph . This pathway relationship metric was implemented in the `pathway.structure.permutation` package with permutation testing (as described in sections 3.4.1 and 3.4.1.1).

3.5.3.5 Metapackage to Install iGraph Functions

These features may be installed together with `igraph.extensions` which can be accessed from a GitHub repository. This meta-package installs `igraph` (Csardi and Nepusz, 2006) and the packages described in section 3.5.3 including their dependencies for matrix operations and statistical procedures: `Matrix`, `matrixcalc`, and `mvtnorm` (Bates and Maechler, 2016; Genz and Bretz, 2009; Genz *et al.*, 2016; Novomestky, 2012).

References

Adler, D. (2005) *vioplot: Violin plot.* R package version 0.2.

Akobeng, A.K. (2007) Understanding diagnostic tests 3: receiver operating characteristic curves. *Acta Pdiatrica*, **96**(5): 644–647.

Araki, H., Knapp, C., Tsai, P., and Print, C. (2012) GeneSetDB: A comprehensive meta-database, statistical and visualisation framework for gene set analysis. *FEBS Open Bio*, **2**: 76–82.

Barabási, A.L. and Oltvai, Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat Rev Genet*, **5**(2): 101–13.

Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A.A., Kim, S., Wilson, C.J., Lehar, J., Kryukov, G.V., Sonkin, D., *et al.* (2012) The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, **483**(7391): 603–607.

Barry, W.T. (2016) *safe: Significance Analysis of Function and Expression.* R package version 3.14.0.

Bates, D. and Maechler, M. (2016) *Matrix: Sparse and Dense Matrix Classes and Methods.* R package version 1.2-7.1.

Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Methodological)*, **57**(1): 289–300.

Borgatti, S.P. (2005) Centrality and network flow. *Social Networks*, **27**(1): 55 – 71.

Brin, S. and Page, L. (1998) The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, **30**(1): 107 – 117.

Cancer Cell Line Encyclopedia (CCLE) (2014) Broad-Novartis Cancer Cell Line Encyclopedia. `http://www.broadinstitute.org/ccle`. Accessed: 07/11/2014.

Cerami, E.G., Gross, B.E., Demir, E., Rodchenkov, I., Babur, O., Anwar, N., Schultz, N., Bader, G.D., and Sander, C. (2011) Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Res*, **39**(Database issue): D685–690.

Chen, A., Beetham, H., Black, M.A., Priya, R., Telford, B.J., Guest, J., Wiggins, G.A.R., Godwin, T.D., Yap, A.S., and Guilford, P.J. (2014) E-cadherin loss alters cytoskeletal organization and adhesion in non-malignant breast cells but is insufficient to induce an epithelial-mesenchymal transition. *BMC Cancer*, **14**(1): 552.

Chen, X. and Tompa, M. (2010) Comparative assessment of methods for aligning multiple genome sequences. *Nat Biotechnol*, **28**(6): 567–572.

Clough, E. and Barrett, T. (2016) The Gene Expression Omnibus Database. *Methods Mol Biol*, **1418**: 93–110.

Collingridge, D.S. (2013) A primer on quantitized data analysis and permutation testing. *Journal of Mixed Methods Research*, **7**(1): 81–97.

Croft, D., Mundo, A.F., Haw, R., Milacic, M., Weiser, J., Wu, G., Caudy, M., Garapati, P., Gillespie, M., Kamdar, M.R., *et al.* (2014) The Reactome pathway knowledgebase. *Nucleic Acids Res*, **42**(database issue): D472D477.

Csardi, G. and Nepusz, T. (2006) The igraph software package for complex network research. *InterJournal*, **Complex Systems**: 1695.

Demir, E., Babur, O., Rodchenkov, I., Aksoy, B.A., Fukuda, K.I., Gross, B., Sumer, O.S., Bader, G.D., and Sander, C. (2013) Using biological pathway data with Paxtools. *PLoS Comput Biol*, **9**(9): e1003194.

Dijkstra, E.W. (1959) A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**(1): 269–271.

Dixon, S.J., Fedyshyn, Y., Koh, J.L., Prasad, T.S., Chahwan, C., Chua, G., Toufighi, K., Baryshnikova, A., Hayles, J., Hoe, K.L., *et al.* (2008) Significant conservation of synthetic lethal genetic interaction networks between distantly related eukaryotes. *Proc Natl Acad Sci U S A*, **105**(43): 16653–8.

Fawcett, T. (2006) An introduction to ROC analysis. *Pattern Recognition Letters*, **27**(8): 861 – 874. {ROC} Analysis in Pattern Recognition.

Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., *et al.* (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*, **5**(10): R80.

Genz, A. and Bretz, F. (2009) Computation of Multivariate Normal and t Probabilities. In *Lecture Notes in Statistics*, volume 195. Springer-Verlag, Heidelberg.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., and Hothorn, T. (2016) *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.0-5. URL.

Globus (Globus) (2017) Research data management simplified. `https://www.globus.org/`. Accessed: 25/03/2017.

Hajian-Tilaki, K. (2013) Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. *Caspian J Intern Med*, **4**(2): 627–635.

Heiskanen, M., Bian, X., Swan, D., and Basu, A. (2014) caArray microarray database in the cancer biomedical informatics grid$^{\mathrm{TM}}$ (caBIG$^{\mathrm{TM}}$). *Cancer Research*, **67**(9 Supplement): 3712–3712.

Hell, P. (1976) Graphs with given neighbourhoods i. problémes combinatorics at theorie des graphes. *Proc Coil Int CNRS, Orsay*, **260**: 219–223.

Holm, S. (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, **6**(2): 65–70.

Huang, E., Cheng, S., Dressman, H., Pittman, J., Tsou, M., Horng, C., Bild, A., Iversen, E., Liao, M., Chen, C., *et al.* (2003) Gene expression predictors of breast cancer outcomes. *Lancet*, **361**: 1590–1596.

Jerby-Arnon, L., Pfetzer, N., Waldman, Y., McGarry, L., James, D., Shanks, E., Seashore-Ludlow, B., Weinstock, A., Geiger, T., Clemons, P., *et al.* (2014) Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality. *Cell*, **158**(5): 1199–1209.

Ju, Z., Liu, W., Roebuck, P.L., Siwak, D.R., Zhang, N., Lu, Y., Davies, M.A., Akbani, R., Weinstein, J.N., Mills, G.B., *et al.* (2015) Development of a robust classifier for quality control of reverse-phase protein arrays. *Bioinformatics*, **31**(6): 912.

Kamada, T. and Kawai, S. (1989) An algorithm for drawing general undirected graphs. *Information Processing Letters*, **31**(1): 7–15.

Kranthi, S., Rao, S., and Manimaran, P. (2013) Identification of synthetic lethal pairs in biological systems through network information centrality. *Mol BioSyst*, **9**(8): 2163–2167.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S.L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, **10**(3): R25.

Latora, V. and Marchiori, M. (2001) Efficient behavior of small-world networks. *Phys Rev Lett*, **87**: 198701.

Law, C.W., Chen, Y., Shi, W., and Smyth, G.K. (2014) voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol*, **15**(2): R29.

Lu, X., Megchelenbrink, W., Notebaart, R.A., and Huynen, M.A. (2015) Predicting human genetic interactions from cancer genome evolution. *PLoS One*, **10**(5): e0125795.

Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., and Wold, B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods*, **5**(7): 621–628.

Neeley, E.S., Kornblau, S.M., Coombes, K.R., and Baggerly, K.A. (2009) Variable slope normalization of reverse phase protein arrays. *Bioinformatics*, **25**(11): 1384.

Novomestky, F. (2012) *matrixcalc: Collection of functions for matrix calculations.* R package version 1.0-3.

Parker, J., Mullins, M., Cheung, M., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., *et al.* (2009) Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology*, **27**(8): 1160–1167.

R Core Team (2016) *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. R version 3.3.2.

Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, **43**(7): e47.

Robinson, M.D. and Oshlack, A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol*, **11**(3): R25.

Rung, J. and Brazma, A. (2013) Reuse of public genome-wide gene expression data. *Nat Rev Genet*, **14**(2): 89–99.

Rustici, G., Kolesnikov, N., Brandizi, M., Burdett, T., Dylag, M., Emam, I., Farne, A., Hastings, E., Ison, J., Keays, M., *et al.* (2013) ArrayExpress update–trends in database growth and links to data analysis tools. *Nucleic Acids Res*, **41**(Database issue): D987–990.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005) Rocr: visualizing classifier performance in r. *Bioinformatics*, **21**(20): 7881.

Slurm development team (Slurm) (2017) Slurm workload manager. `https://slurm.schedmd.com/`. Accessed: 25/03/2017.

Stajich, J.E. and Lapp, H. (2006) Open source tools and toolkits for bioinformatics: significance, and where are we? *Brief Bioinformatics*, **7**(3): 287–296.

Telford, B.J., Chen, A., Beetham, H., Frick, J., Brew, T.P., Gould, C.M., Single, A., Godwin, T., Simpson, K.J., and Guilford, P. (2015) Synthetic lethal screens identify vulnerabilities in gpcr signalling and cytoskeletal organization in e-cadherin-deficient cells. *Mol Cancer Ther*, **14**(5): 1213–1223.

The Cancer Genome Atlas Research Network (TCGA) (2012) Comprehensive molecular portraits of human breast tumours. *Nature*, **490**(7418): 61–70.

The Comprehensive R Archive Network (CRAN) (2017) Cran. `https://cran.r-project.org/`. Accessed: 24/03/2017.

The New Zealand eScience Infrastructure (NeSI) (2017) NeSI. `https://www.nesi.org.nz/`. Accessed: 25/03/2017.

Tierney, L., Rossini, A.J., Li, N., and Sevcikova, H. (2015) *snow: Simple Network of Workstations*. R package version 0.4-2.

van Steen, M. (2010) *Graph Theory and Complex Networks: An Introduction*. Maarten van Steen, VU Amsterdam.

Wang, K., Singh, D., Zeng, Z., Coleman, S.J., Huang, Y., Savich, G.L., He, X., Mieczkowski, P., Grimm, S.A., Perou, C.M., *et al.* (2010) MapSplice: accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Res*, **38**(18): e178.

Wappett, M., Dulak, A., Yang, Z.R., Al-Watban, A., Bradford, J.R., and Dry, J.R. (2016) Multi-omic measurement of mutually exclusive loss-of-function enriches for candidate synthetic lethal gene pairs. *BMC Genomics*, **17**: 65.

Warnes, G.R., Bolker, B., Bonebakker, L., Gentleman, R., Liaw, W.H.A., Lumley, T., Maechler, M., Magnusson, A., Moeller, S., Schwartz, M., *et al.* (2015) *gplots: Various R Programming Tools for Plotting Data*. R package version 2.17.0.

Wickham, H. and Chang, W. (2016) *devtools: Tools to Make Developing R Packages Easier*. R package version 1.12.0.

Wickham, H., Danenberg, P., and Eugster, M. (2017) *roxygen2: In-Line Documentation for R*. R package version 6.0.1.

Yu, H. (2002) Rmpi: Parallel statistical computing in r. *R News*, **2**(2): 10–14.

Zhang, J., Baran, J., Cros, A., Guberman, J.M., Haider, S., Hsu, J., Liang, Y., Rivkin, E., Wang, J., Whitty, B., *et al.* (2011) International cancer genome consortium data portala one-stop shop for cancer genomics data. *Database: The Journal of Biological Databases and Curation*, **2011**: bar026.

Zweig, M.H. and Campbell, G. (1993) Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, **39**(4): 561–577.

Appendix A

Sample Correlation

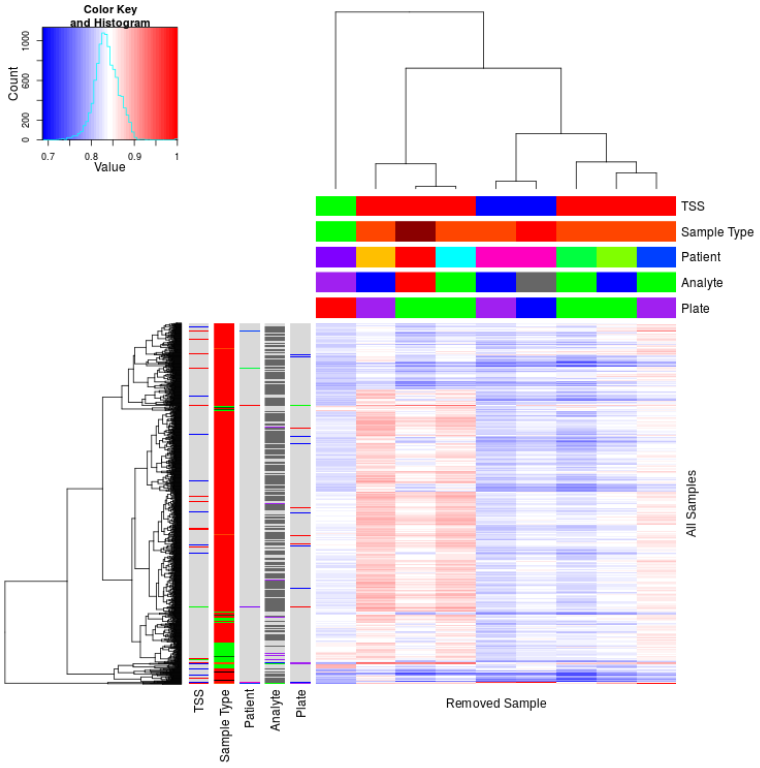


Figure A.1: **Correlation profiles of removed samples.** Correlation matrix heatmap (Euclidean distance) of all samples in TCGA breast cancer dataset (left) clustered for all samples against removed samples (top): tissue source site (TSS), sample type with reds for tumour and greens for normal, patient (A2QH in pink), with varied analyte and plate (corresponding to batch in Table 2.1). Excluded samples cluster at the bottom and annotation (left) show shared properties between samples in the dataset.

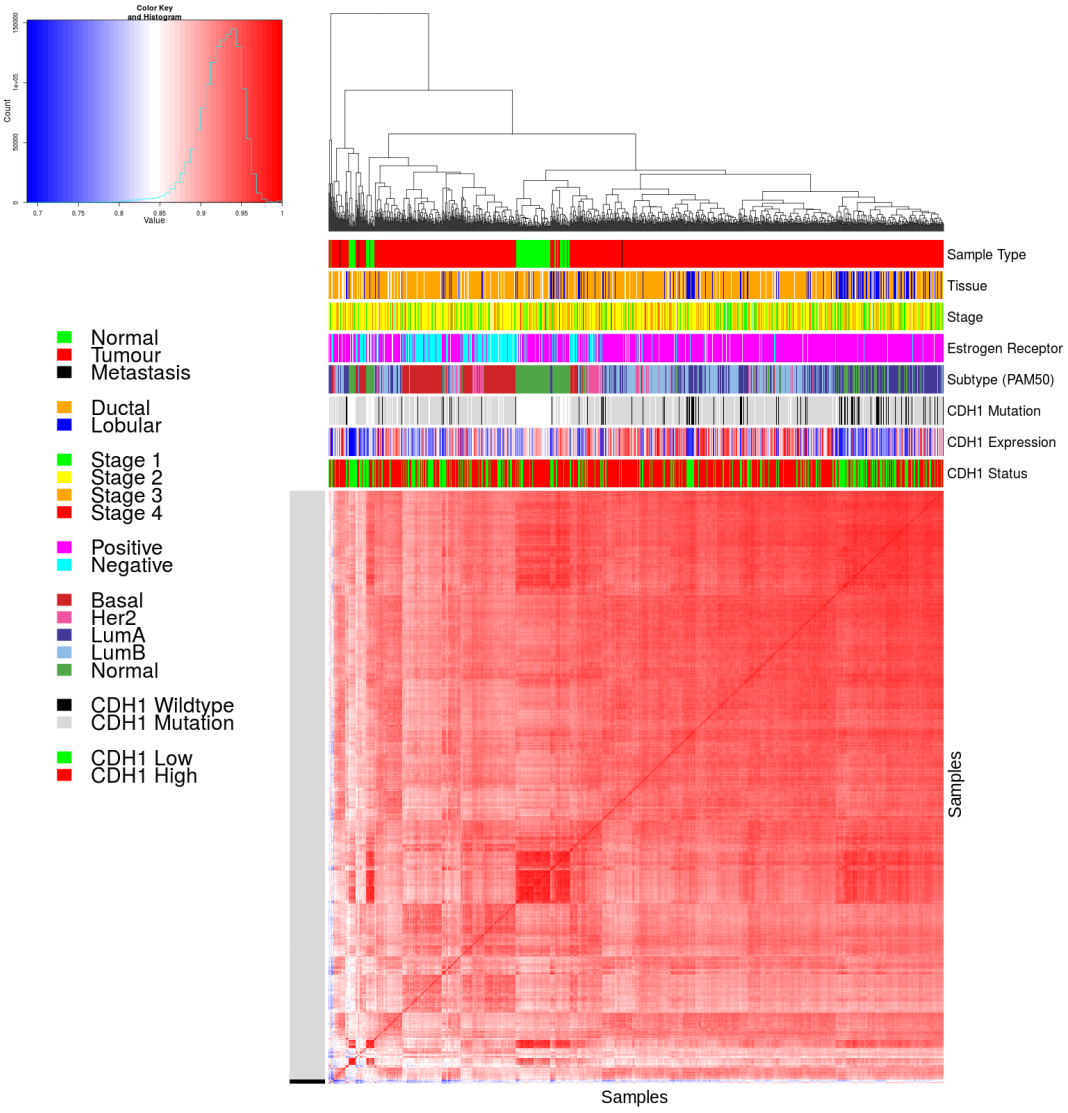


Figure A.2: **Correlation analysis and sample removal.** Correlation matrix heatmap (Euclidean distance) of all samples in TCGA breast cancer dataset against each other annotated for sample clinical data: sample type, tissue type, tumour stage, Estrogen receptor (IHC) and intrinsic subtype (from the PAM50 method). CDH1 somatic mutation, gene expression, and status for SLIPT prediction are also annotated. Discrete variables are coloured as displayed in the legend and continuous variables on a blue-red scale as shown in the colour key. Trimmed samples cluster at the bottom of the heatmap and the colour bars of the left show which were removed for quality concerns.

Appendix B

Software Used for Thesis

Table B.1: R Packages used during Thesis

Package	Repository	Laptop	Lab	Server	NeSI
base	base	3.3.2	3.3.2	3.3.1	3.3.0
abind	CRAN		1.4-5		1.4-3
acepack	CRAN		1.4.1		1.3-3.3
ade4	CRAN		1.7-5		
annaffy	Bioconductor		1.46.0		
AnnotationDbi	Bioconductor		1.36.0	1.36.0	1.34.4
apComplex	CRAN		2.40.0		
ape	CRAN		4		3.4
arm	CRAN		1.9-3		
assertthat	CRAN	0.1	0.1	0.1	0.1
backports	CRAN	1.0.5	1.0.4	1.0.5	1.0.2
base64	CRAN			2	2
base64enc	CRAN		0.1-3		0.1-3
beanplot	CRAN		1.2	1.2	1.2
BH	CRAN	1.60.0-2	1.62.0-1	1.62.0-1	1.60.0-2
Biobase	Bioconductor		2.34.0	2.34.0	2.32.0
BiocGenerics	Bioconductor		0.20.0	0.20.0	0.18.0
BiocInstaller	Bioconductor		1.24.0	1.20.3	1.22.3
BiocParallel	Bioconductor		1.8.1	1.8.1	
Biostrings	Bioconductor		2.42.1	2.42.0	
BiSEp	Bioconductor		2.0.1	2.0.1	2.0.1

bitops	CRAN	1.0-6	1.0-6	1.0-6	1.0-6
boot	base	1.3-18	1.3-18	1.3-18	1.3-18
brew	CRAN	1.0-6	1.0-6	1.0-6	1.0-6
broom	CRAN	0.4.1			
caTools	CRAN	1.17.1	1.17.1	1.17.1	1.17.1
cgdsr	CRAN		1.2.5		
checkmate	CRAN		1.8.2		1.7.4
chron	CRAN	2.3-47	2.3-48	2.3-50	2.3-47
class	base	7.3-14	7.3-14	7.3-14	7.3-14
cluster	base	2.0.5	2.0.5	2.0.5	2.0.4
coda	CRAN		0.19-1		0.18-1
codetools	base	0.2-15	0.2-15	0.2-15	0.2-14
colorRamps	CRAN		2.3		
colorspace	CRAN	1.2-6	1.3-2	1.3-2	1.2-6
commonmark	CRAN	1.1		1.2	
compiler	base	3.3.2	3.3.2	3.3.1	3.3.0
corpcor	CRAN		1.6.8	1.6.8	1.6.8
Cprob	CRAN		1.2.4		
crayon	CRAN	1.3.2	1.3.2	1.3.2	1.3.2
crop	CRAN		0.0-2	0.0-2	
curl	CRAN	1.2	2.3	2.3	0.9.7
d3Network	CRAN		0.5.2.1		
data.table	CRAN	1.9.6	1.10.0	1.10.1	1.9.6
data.tree	CRAN		0.7.0	0.7.0	
datasets	base	3.3.2	3.3.2	3.3.1	3.3.0
DBI	CRAN	0.5-1	0.5-1	0.5-1	0.5-1
dendextend	CRAN	1.4.0	1.4.0	1.4.0	
DEoptimR	CRAN	1.0-8	1.0-8	1.0-8	1.0-4
desc	CRAN	1.1.0		1.1.0	
devtools	CRAN	1.12.0	1.12.0	1.12.0	1.12.0
DiagrammeR	CRAN		0.9.0	0.9.0	
dichromat	CRAN	2.0-0	2.0-0	2.0-0	2.0-0
digest	CRAN	0.6.10	0.6.11	0.6.12	0.6.9
diptest	CRAN	0.75-7	0.75-7	0.75-7	
doParallel	CRAN	1.0.10	1.0.10	1.0.10	1.0.10

dplyr	CRAN	0.5.0	0.5.0	0.5.0	0.5.0
ellipse	CRAN		0.3-8	0.3-8	0.3-8
evaluate	CRAN		0.1	0.1	0.9
fdrtool	CRAN		1.2.15		
fields	CRAN		8.1		
flexmix	CRAN	2.3-13	2.3-13	2.3-13	
forcats	CRAN	0.2.0			
foreach	CRAN	1.4.3	1.4.3	1.4.3	1.4.3
foreign	base	0.8-67	0.8-67	0.8-67	0.8-66
formatR	CRAN		1.4	1.4	1.4
Formula	CRAN		1.2-1		1.2-1
fpc	CRAN	2.1-10	2.1-10	2.1-10	
futile.logger	CRAN		1.4.3	1.4.3	1.4.1
futile.options	CRAN		1.0.0	1.0.0	1.0.0
gdata	CRAN	2.17.0	2.17.0	2.17.0	2.17.0
geepack	CRAN		1.2-1		
GenomeInfoDb	Bioconductor		1.10.2	1.10.1	
GenomicAlignments	Bioconductor		1.10.0	1.10.0	
GenomicRanges	Bioconductor		1.26.2	1.26.1	
ggm	CRAN		2.3		
ggplot2	CRAN	2.1.0	2.2.1	2.2.1	2.1.0
git2r	CRAN	0.15.0	0.18.0	0.16.0	0.15.0
glasso	CRAN		1.8		
GO.db	Bioconductor		3.4.0	3.2.2	3.3.0
GOSemSim	Bioconductor		2.0.3	1.28.2	1.30.3
gplots	CRAN	3.0.1	3.0.1	3.0.1	3.0.1
graph	Bioconductor		1.52.0		
graphics	base	3.3.2	3.3.2	3.3.1	3.3.0
graphsim	GitHub TomKellyGenetics	0.1.0	0.1.0	0.1.0	0.1.0
grDevices	base	3.3.2	3.3.2	3.3.1	3.3.0
grid	base	3.3.2	3.3.2	3.3.1	3.3.0
gridBase	CRAN	0.4-7	0.4-7	0.4-7	0.4-7
gridExtra	CRAN	2.2.1	2.2.1	2.2.1	2.2.1
gridGraphics	CRAN		0.1-5		

gtable	CRAN	0.2.0	0.2.0	0.2.0	0.2.0
gtools	CRAN	3.5.0	3.5.0	3.5.0	3.5.0
haven	CRAN	1.0.0			
heatmap.2x	GitHub TomKellyGenetics	0.0.0.9000	0.0.0.9000	0.0.0.9000	0.0.0.9000
hgu133plus2.db	Bioconductor		3.2.3		
highr	CRAN		0.6	0.6	0.6
Hmisc	CRAN		4.0-2	4.0-2	3.17-4
hms	CRAN	0.2	0.3		
htmlTable	CRAN		1.8	1.9	
htmltools	CRAN	0.3.5	0.3.5	0.3.5	0.3.5
htmlwidgets	CRAN		0.8	0.8	
httpuv	CRAN	1.3.3		1.3.3	
httr	CRAN	1.2.1	1.2.1	1.2.1	1.1.0
huge	CRAN		1.2.7		
hunspell	CRAN		2.3		2
hypergraph	CRAN		1.46.0		
igraph	CRAN	1.0.1	1.0.1	1.0.1	1.0.1
igraph.extensions	GitHub TomKellyGenetics	0.1.0.9001	0.1.0.9001	0.1.0.9001	0.1.0.9001
influenceR	CRAN		0.1.0	0.1.0	
info.centrality	GitHub TomKellyGenetics	0.1.0	0.1.0	0.1.0	0.1.0
IRanges	Bioconductor		2.8.1	2.8.1	2.6.1
irlba	CRAN	2.1.1	2.1.2	2.1.2	2.0.0
iterators	CRAN	1.0.8	1.0.8	1.0.8	1.0.8
jpeg	CRAN		0.1-8		
jsonlite	CRAN	1.1	1.2	1.3	0.9.20
KEGG.db	Bioconductor		3.2.3		
kernlab	CRAN	0.9-25	0.9-25	0.9-25	
KernSmooth	base	2.23-15	2.23-15	2.23-15	2.23-15
knitr	CRAN		1.15.1	1.15.1	1.14
labeling	CRAN	0.3	0.3	0.3	0.3
lambda.r	CRAN		1.1.9	1.1.9	1.1.7
lattice	base	0.20-34	0.20-34	0.20-34	0.20-33

latticeExtra	CRAN		0.6-28		0.6-28
lava	CRAN		1.4.6		
lavaan	CRAN		0.5-22		
lazyeval	CRAN	0.2.0	0.2.0	0.2.0	0.2.0
les	CRAN		1.24.0		
lgtdl	CRAN		1.1.3		
limma	Bioconductor		3.30.7	3.30.3	
lme4	CRAN		1.1-12		1.1-12
lubridate	CRAN	1.6.0			
magrittr	CRAN	1.5	1.5	1.5	1.5
maps	CRAN		3.1.1		
markdown	CRAN		0.7.7	0.7.7	0.7.7
MASS	base	7.3-45	7.3-45	7.3-45	7.3-45
Matrix	base	1.2-7.1	1.2-7.1	1.2-8	1.2-6
matrixcalc	CRAN	1.0-3	1.0-3	1.0-3	1.0-3
mclust	CRAN	5.2	5.2.1	5.2.2	5.2
memoise	CRAN	1.0.0	1.0.0	1.0.0	1.0.0
methods	base	3.3.2	3.3.2	3.3.1	3.3.0
mgcv	base	1.8-16	1.8-16	1.8-17	1.8-12
mi	CRAN		1		
mime	CRAN	0.5	0.5	0.5	0.4
minqa	CRAN		1.2.4		1.2.4
mnormt	CRAN	1.5-5	1.5-5		1.5-4
modelr	CRAN	0.1.0			
modeltools	CRAN	0.2-21	0.2-21	0.2-21	
multtest	Bioconductor		2.30.0	2.30.0	
munsell	CRAN	0.4.3	0.4.3	0.4.3	0.4.3
mvtnorm	CRAN	1.0-5	1.0-5	1.0-6	1.0-5
network	CRAN		1.13.0		
nlme	base	3.1-128	3.1-128	3.1-131	3.1-128
nloptr	CRAN		1.0.4		1.0.4
NMF	CRAN	0.20.6	0.20.6	0.20.6	0.20.6
nnet	base	7.3-12	7.3-12	7.3-12	7.3-12
numDeriv	CRAN		2016.8-1		2014.2-1
openssl	CRAN	0.9.4	0.9.6	0.9.6	0.9.4

org.Hs.eg.db	Bioconductor		3.1.2		3.3.0
org.Sc.sgd.db	Bioconductor		3.4.0		
parallel	base	3.3.2	3.3.2	3.3.1	3.3.0
pathway.structure .permutation	GitHub TomKellyGenetics	0.1.0	0.1.0	0.1.0	0.1.0
pbivnorm	CRAN		0.6.0		
PGSEA	Bioconductor		1.48.0		
pkgmaker	CRAN	0.22	0.22	0.22	0.22
PKI	CRAN		0.1-3		
plogr	CRAN		0.1-1	0.1-1	
plot.igraph	GitHub TomKellyGenetics	0.0.0.9001	0.0.0.9001	0.0.0.9001	0.0.0.9001
plotrix	CRAN		3.6-4		
plyr	CRAN	1.8.4	1.8.4	1.8.4	1.8.3
png	CRAN		0.1-7		0.1-7
prabclus	CRAN	2.2-6	2.2-6	2.2-6	
praise	CRAN	1.0.0	1.0.0		1.0.0
pROC	CRAN		1.8	1.9.1	
prodlim	CRAN		1.5.7		
prof.tree	CRAN		0.1.0		
proftools	CRAN		0.99-2		
progress	CRAN			1.1.2	
psych	CRAN	1.6.12	1.6.12		
purrr	CRAN	0.2.2	0.2.2	0.2.2	0.2.2
qgraph	CRAN		1.4.1		
quadprog	CRAN		1.5-5	1.5-5	1.5-5
R.methodsS3	CRAN		1.7.1		1.7.1
R.oo	CRAN		1.21.0		1.20.0
R.utils	CRAN		2.5.0		
R6	CRAN	2.1.3	2.2.0	2.2.0	2.1.3
RBGL	CRAN		1.50.0		
RColorBrewer	CRAN	1.1-2	1.1-2	1.1-2	1.1-2
Rcpp	CRAN	0.12.7	0.12.9	0.12.9	0.12.7
RcppArmadillo	CRAN			0.7.700.0.0	0.6.700.6.0
RcppEigen	CRAN		0.3.2.9.0		0.3.2.8.1

RCurl	CRAN		1.95-4.8	1.95-4.8	1.95-4.8
reactome.db	Bioconductor		1.52.1	1.52.1	
reactometree	GitHub TomKellyGenetics		0.1		
readr	CRAN	1.0.0	1.0.0		
readxl	CRAN	0.1.1			
registry	CRAN	0.3	0.3	0.3	0.3
reshape2	CRAN	1.4.1	1.4.2	1.4.2	1.4.1
rgexf	CRAN		0.15.3	0.15.3	
rgl	CRAN			0.97.0	0.95.1441
Rgraphviz	CRAN		2.18.0		
rjson	CRAN		0.2.15		
RJSONIO	CRAN		1.3-0		
rmarkdown	CRAN		1.3	1.3	1
Rmpi	CRAN		0.6-6		0.6-5
rngtools	CRAN	1.2.4	1.2.4	1.2.4	1.2.4
robustbase	CRAN	0.92-7	0.92-7	0.92-7	0.92-5
ROCR	CRAN	1.0-7	1.0-7	1.0-7	1.0-7
Rook	CRAN		1.1-1	1.1-1	
roxygen2	CRAN	6.0.1	5.0.1	6.0.1	5.0.1
rpart	base	4.1-10	4.1-10	4.1-10	4.1-10
rprojroot	CRAN	1.2	1.1	1.2	
Rsamtools	Bioconductor		1.26.1	1.26.1	
rsconnect	CRAN		0.7		
RSQLite	CRAN		1.1-2	1.1-2	1.0.0
rstudioapi	CRAN	0.6	0.6	0.6	0.6
rvest	CRAN	0.3.2			
S4Vectors	Bioconductor		0.12.1	0.12.0	0.10.3
safe	Bioconductor		3.14.0	3.10.0	
scales	CRAN	0.4.0	0.4.1	0.4.1	0.4.0
selectr	CRAN	0.3-1			
sem	CRAN		3.1-8		
shiny	CRAN	0.14		1.0.0	
slipt	GitHub TomKellyGenetics	0.1.0	0.1.0	0.1.0	0.1.0

sm	CRAN	2.2-5.4	2.2-5.4		
sna	CRAN		2.4		
snow	CRAN	0.4-1	0.4-2	0.4-2	0.3-13
sourcetools	CRAN	0.1.5		0.1.5	
SparseM	CRAN		1.74		1.7
spatial	base	7.3-11	7.3-11	7.3-11	7.3-11
splines	base	3.3.2	3.3.2	3.3.1	3.3.0
statnet.common	CRAN		3.3.0		
stats	base	3.3.2	3.3.2	3.3.1	3.3.0
stats4	base	3.3.2	3.3.2	3.3.1	3.3.0
stringi	CRAN	1.1.1	1.1.2	1.1.2	1.0-1
stringr	CRAN	1.1.0	1.1.0	1.2.0	1.0.0
Summarized Experiment	Bioconductor		1.4.0	1.4.0	
survival	base	2.39-4	2.40-1	2.40-1	2.39-4
tcltk	base	3.3.2	3.3.2	3.3.1	3.3.0
testthat	CRAN	1.0.2	1.0.2		1.0.2
tibble	CRAN	1.2	1.2	1.2	1.2
tidyr	CRAN	0.6.1	0.6.1	0.6.1	
tidyverse	GitHub hadley	1.1.1			
timeline	CRAN		0.9		
tools	base	3.3.2	3.3.2	3.3.1	3.3.0
tpr	CRAN		0.3-1		
trimcluster	CRAN	0.1-2	0.1-2	0.1-2	
Unicode	CRAN	9.0.0-1	9.0.0-1	9.0.0-1	
utils	base	3.3.2	3.3.2	3.3.1	3.3.0
vioplot	CRAN		0.2		
vioplotx	GitHub TomKellyGenetics	0.0.0.9000	0.0.0.9000		
viridis	CRAN	0.3.4	0.3.4	0.3.4	
visNetwork	CRAN		1.0.3	1.0.3	
whisker	CRAN	0.3-2	0.3-2	0.3-2	0.3-2
withr	CRAN	1.0.2	1.0.2	1.0.2	1.0.2
XML	base	3.98-1.3	3.98-1.1	3.98-1.5	3.98-1.4

xml2	CRAN	1.1.1		1.1.1	1.0.0
xtable	CRAN	1.8-2	1.8-2	1.8-2	1.8-2
XVector	Bioconductor		0.14.0	0.14.0	
yaml	CRAN		2.1.14	2.1.14	2.1.13
zlibbioc	CRAN		1.20.0	1.20.0	
zoo	CRAN	1.7-13	1.7-14		1.7-13

Appendix C

Secondary Screen Data

A series of experimental genome-wide siRNA screens have been performed on synthetic lethal partners of *CDH1* (Telford *et al.*, 2015). The strongest candidates from a primary screen were subject to a further secondary screen for validation by independent replication with 4 gene knockdowns with different targeting siRNA. As shown in Table C.1, there is significant ($p = 7.49 \times 10^{-3}$ by Fisher's exact test) association between SLIPT candidates and stronger validations of siRNA candidates. Since there were more SLIPT− genes among those not validated and more SLIPT+ genes among those validated with several siRNAs, this supports the use of SLIPT as a synthetic lethal discovery procedure which may augment such screening experiments.

Table C.1: Comparing SLIPT genes against Secondary siRNA Screen in breast cancer

		Secondary Screen					
		0/4	1/4	2/4	3/4	4/4	**Total**
SLIPT+	Observed	70	46	31	8	2	157
	Expected	85	44	10	4	2	
SLIPT−	Observed	190	90	31	10	4	325
	Expected	175	91	42	12	4	
	Total	280	136	52	18	6	482

Similar analysis with mtSLIPT, comparing SLIPT against *CDH1* somatic mutation with siRNA validation results was not significant ($p = 7.02 \times 10^{-1}$ by Fisher's exact test). However, as shown in Table C.2, the observed and expected values were in a direction consistent with that observed above for SLIPT against low *CDH1* expression.

It is not unexpected that this result does not have comparable statistical support due to the lower sample size for mutation data.

Table C.2: Comparing mtSLIPT genes against Secondary siRNA Screen in breast cancer

		Secondary Screen					
		0/4	1/4	2/4	3/4	4/4	**Total**
mtSLIPT+	Observed	54	35	17	4	6	111
	Expected	60	31	14	4	1	
mtSLIPT−	Observed	206	101	45	14	5	371
	Expected	200	105	48	14	4	
	Total	269	143	63	19	6	482

This analysis was replicated on a (smaller) stomach cancer dataset but it was less conclusive ($p = 2.36 \times 10^{-1}$ by Fisher's exact test). As shown in Table C.3, fewer SLIPT candidates were validated than expected statistically. However, these results in stomach cancer may not be directly comparable to experiments in a breast cell line. Genes validated by 0 or 1 siRNA behave consistently with the results above.

Table C.3: Comparing SLIPT genes against Secondary siRNA Screen in stomach cancer

		Secondary Screen					
		0/4	1/4	2/4	3/4	4/4	**Total**
SLIPT+	Observed	67	47	13	4	1	132
	Expected	71	37	17	5	2	
SLIPT−	Observed	195	90	50	14	5	354
	Expected	190	100	46	13	4	
	Total	262	137	63	19	6	486