

Contents

2	Methods, Techniques, and Resources	5
2.1	Bioinformatics Resources to Enable Genomics Research	5
2.1.1	Public Data and Software Packages	5
2.1.1.1	Cancer Genome Atlas Data	6
2.1.1.2	Reactome and Annotation Data	7
2.2	Data Handling	7
2.2.1	Normalisation (voom)	7
2.2.2	Sample triage	7
2.2.3	Pathway Metagenes and the Singular Value Decomposition . . .	9
2.2.3.1	Candidate Triage and Integration with Screen Data . .	9
2.3	Techniques	12
2.3.1	Statistical Procedures and Tests	12
2.3.2	Gene Set Over-representation Analysis	13
2.3.3	Clustering	14
2.3.4	Heatmap	14
2.3.5	Modeling and Simulations	14
2.3.5.1	Reciever Operator Curves (Statistical Performance) . .	15
2.3.6	Permutation and Resampling Analysis	16
2.4	Pathway Structure Methods	16
2.4.1	Network and Graph Analysis	16
2.4.2	Sourcing Graph Structure Data	18
2.4.3	Constructing Pathway Subgraphs	18
2.4.4	Network Analysis Metrics	18
2.5	Implementation	19
2.5.1	Computational Resources and Linux Utilities	20
2.5.2	R Language and Packages	21
2.5.3	High Performance and Parallel Computing	23
3	Methods Developed During Thesis	24
3.1	A Synthetic Lethal Detection Methodology	24
3.1.1	Supporting Data	26
3.2	Simulations and Modelling Synthetic Lethality in Expression Data . . .	27
3.2.1	A Model of Synthetic Lethality	27
3.2.2	Simulation Procedure	31
3.3	Detecting Simulated Synthetic Lethal Partners	33
3.3.1	Multivariate Normal Simulation of Synthetic lethality	34

3.3.1.1	Simulation with Directional Testing	34
3.3.1.2	Simulation with Query-Correlated Pathways	34
3.3.1.3	Simulated Expression Heatmaps	35
3.3.2	Replication Simulation Heatmap	35
3.4	Graph Structure Methods	35
3.4.1	Upstream and Downstream Gene Detection	35
3.4.1.1	Permutation Analysis for Statisical Significance	36
3.4.2	Simulating Gene Expression from Graph Structures	36
3.5	Customised methods and R packages developed	38
3.5.1	slipt	38
3.5.2	plotting	38
3.5.3	simulation from graph structures	38
3.5.4	igraph methods	38
	References	41
	A Software Used for Thesis	44

List of Figures

2.1	Read count density	8
2.2	Read count sample mean	8
2.3	Correlation profiles of removed samples	10
2.4	Correlation analysis and sample removal	11
3.1	Framework for synthetic lethal prediction	25
3.2	Synthetic lethal prediction adapted for mutation	26
3.3	A model of Synthetic Lethality	29
3.4	Modeling Higher-order Synthetic Lethality	30
3.5	Modeling Synthetic Lethal Gene Expression	32
3.6	Simulating gene function from gene expression	33
3.7	Simulating synthetic lethal genes with gene function	34
3.8	Simulating synthetic lethal gene expression	35
3.9	Simulating graph structures	36
3.10	Simulating expression from a graph structure	39
3.11	Simulating expression from graph structure with inhibitions	40

List of Tables

2.1	Excluded samples batch and clinical characteristics.	9
2.2	Computers Used During Thesis	19
2.3	Linux Utilities and Applications Used During Thesis	20
2.4	R Installations Used During Thesis	21
2.5	R Packages Developed During Thesis	21
2.6	R Packages Used During Thesis	22
3.1	Candidate synthetic lethal genes against secondary siRNA screen . . .	27
A.1	R Packages Used During Thesis	44

Chapter 2

Methods, Techniques, and Resources

Here we 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 or 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 the following chapter.

Bioinformatics Resources to Enable Genomics Research

Public Data and Software Packages

Various bioinformatics resources, such as databases and methods, have become an integral part of genetics and genomics research. Reference genomes, genotyped variants, gene expression, and epigenetics profiles are among the most commonly used resources. Gene expression in particular is widely available from many microarray and RNA-Seq studies, including Gene Expression Omnibus (GEO), caArray, and arrayExpress. Such profiles serve as an excellent resource to examine the changes of gene expression occurring in cancers and their variation. 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 further utilise the data compared to a single research group or

consortium. 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 (such as TCGA). This thesis is one such example of an investigation enabled by this wider movement and tools developed in various disciplines to generate, disseminate, and process genomic-scale data.

Along with databases, it is also becoming common practice for Bioinformatics researcher to release their code open-source or provide a software package to enable replication of the findings or further applications of the methods. 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. In addition to the R packages hosted on CRAN, the Bioconductor repositories 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.

Cancer Genome Atlas Data

Molecular profile data from normal and tumour samples was downloaded from publicly available sources, using the TCGA (Network, 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. The latest versions downloaded were on Aug 6th 2015 (Release 19) and May 2nd 2016 (Release 20) for breast and stomach cancer respectively via the ICGC data portal.

There remains debate regarding the optimal methods to perform an alignment. However, the statistical and biological aspects of Bioinformatics are the focus of this thesis, comparing alignment methods is outside the scope of this investigations. The TCGA project used the widely adopted “Bowtie” tools for alignment, with “mapslice” to detect splice sites, and the Reads Per Kilobase per Million mapped reads (RPKM) approach to quantify reads per transcript as a measure of gene expression. These are widely acceptable tools for processing RNA-Seq data and this is the raw quantitative expression data publicly available from TCGA.

Raw count and RSEM normalised TCGA expression data from Illumina RNA-Seq protocols were available from 1,177 samples (113 normal, 1,057 primary tumour,

and 7 metastases) for 20,501 genes. TCGA 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.

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 Gatz and colleagues (Gatz *et al.*, 2011; Gatz *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).

Data Handling

Normalisation (voom)

Apart from the PAM50 subtyping procedure (Bernard *et al.*, 2009) which required RSEM normalised data as (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 were trimmed as described above based on a correlation matrix (Euclidean distance), and the final dataset was TMM normalised (Robinson and Oshlack, 2010) and then processed using the `voom` function (Law *et al.*, 2014) in the `limma` R package (Ritchie *et al.*, 2015).

Sample triage

The TCGA 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 2.4. While no major batch effects were detectable between

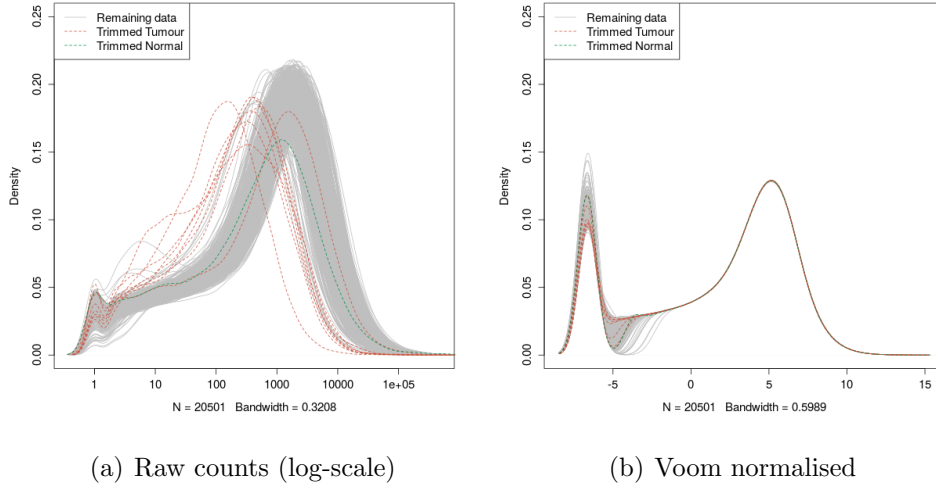


Figure 2.1: **Read count density.** Sample density plots of raw counts on log-scale and voom normalised showing samples removed due to quality concerns.

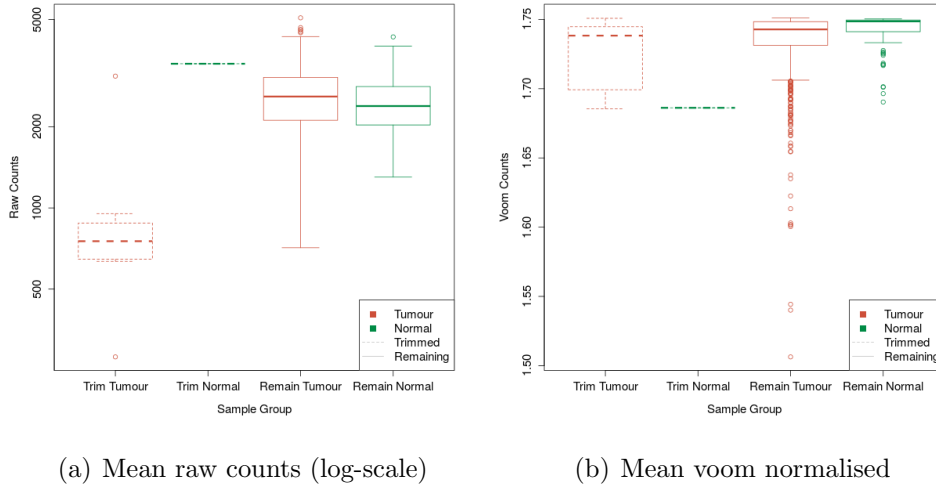


Figure 2.2: **Read count sample mean.** Sample boxplots of raw counts on log-scale and voom normalised showing samples removed due to quality concerns with low sample mean read count.

the samples, 9 samples were excluded due to poor correlation with the remaining samples as detailed in Table 2.1. These samples show unusual density plots compared to the rest of the dataset and 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

other rest of the samples is shown in Figure 2.3 and a full correlation heatmap (Figure 2.4) 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 batch and clinical characteristics.

Tissue Source	Type	Batch	Plate	Patient	Samples	p53	Subtype	Treatment (History)	Clinical (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	A2CS	1 of 1	NA	Luminal A	Radical Mastectomy and Neoadjuvant (no)	ER+ Ductal (2)

Pathway Metagenes and the Singular Value Decomposition

A “metagene” offers a consistent signal of pathway (expression) activation or inactivation by dimension reduction of a matrix, rather than negatively correlated genes averaging out the signal (Huang *et al.*, 2003). Construcing 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 ($X = U^T D V$ where X is the data matrix of the gene set with genes \times samples) and the leading eigenvector ($V[, 1]$) 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 direction is reversed to conform with the majority of the gene set. To ensure that genes and pathways were weighted equally, this was performed on a z-transformed dataset of gene expression and samples were ranked for each metagene so that each metagene was on a comparable $[0, 1]$ scale.

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

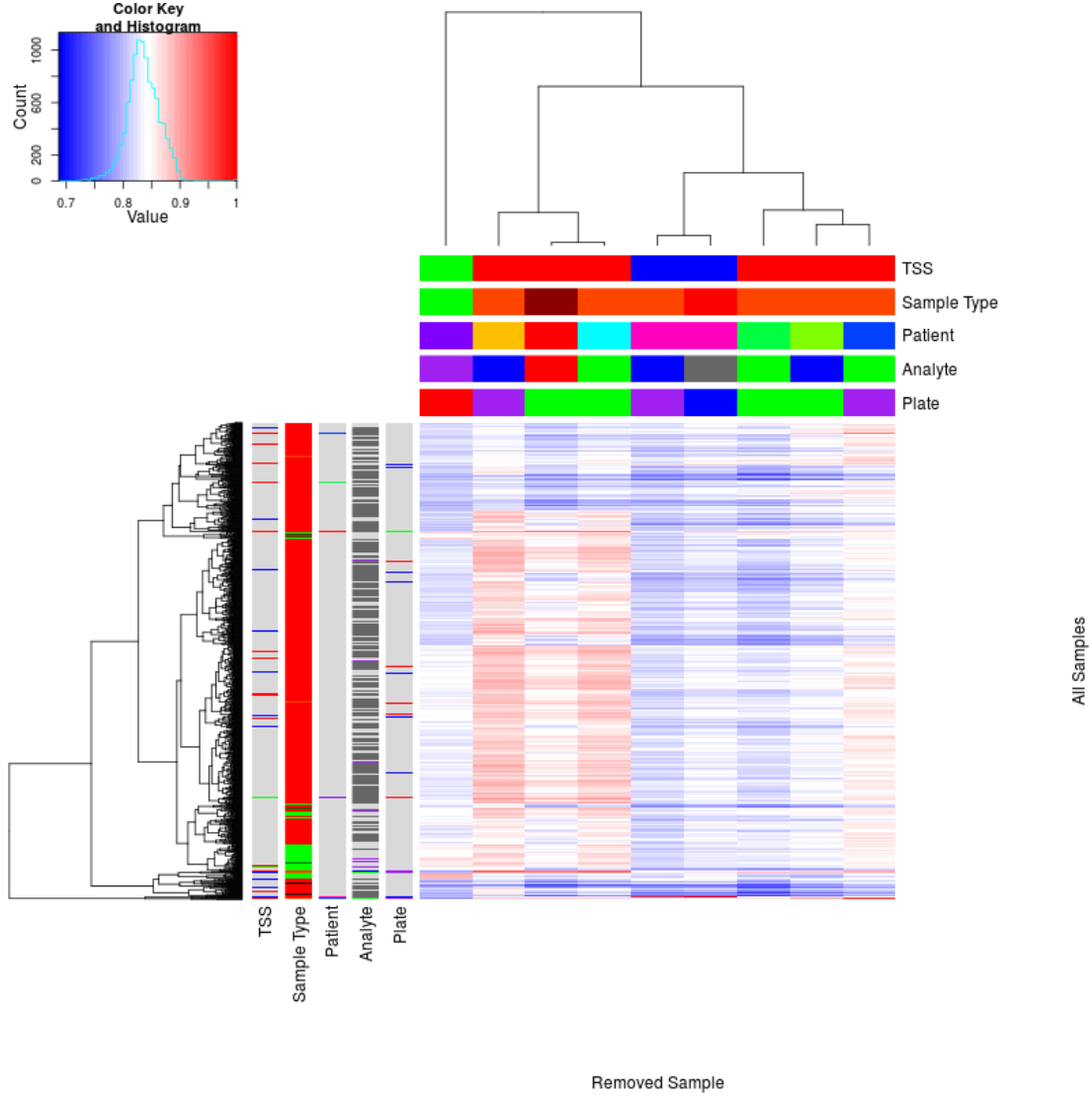


Figure 2.3: **Correlation profiles of removed samples.** Correlation matrix heatmap (Euclidean distance) of all samples in TCGA breast cancer dataset on the left (clustered with respect to all samples) against those removed annotated for sample codes: tissue source site (TSS), sample type with reds for tumour and greens for normal, patient showing both A2QH samples in pink, with various colours for analyte and plate (corresponding to batch in Table 2.1). Trimmed samples cluster at the bottom of the heatmap and the colour bars of the left therefore show which sample properties are shared with other samples in the dataset.)

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 were not very comparable at the gene level (even when excluding genes not contained in both datasets), further gene set over-representation

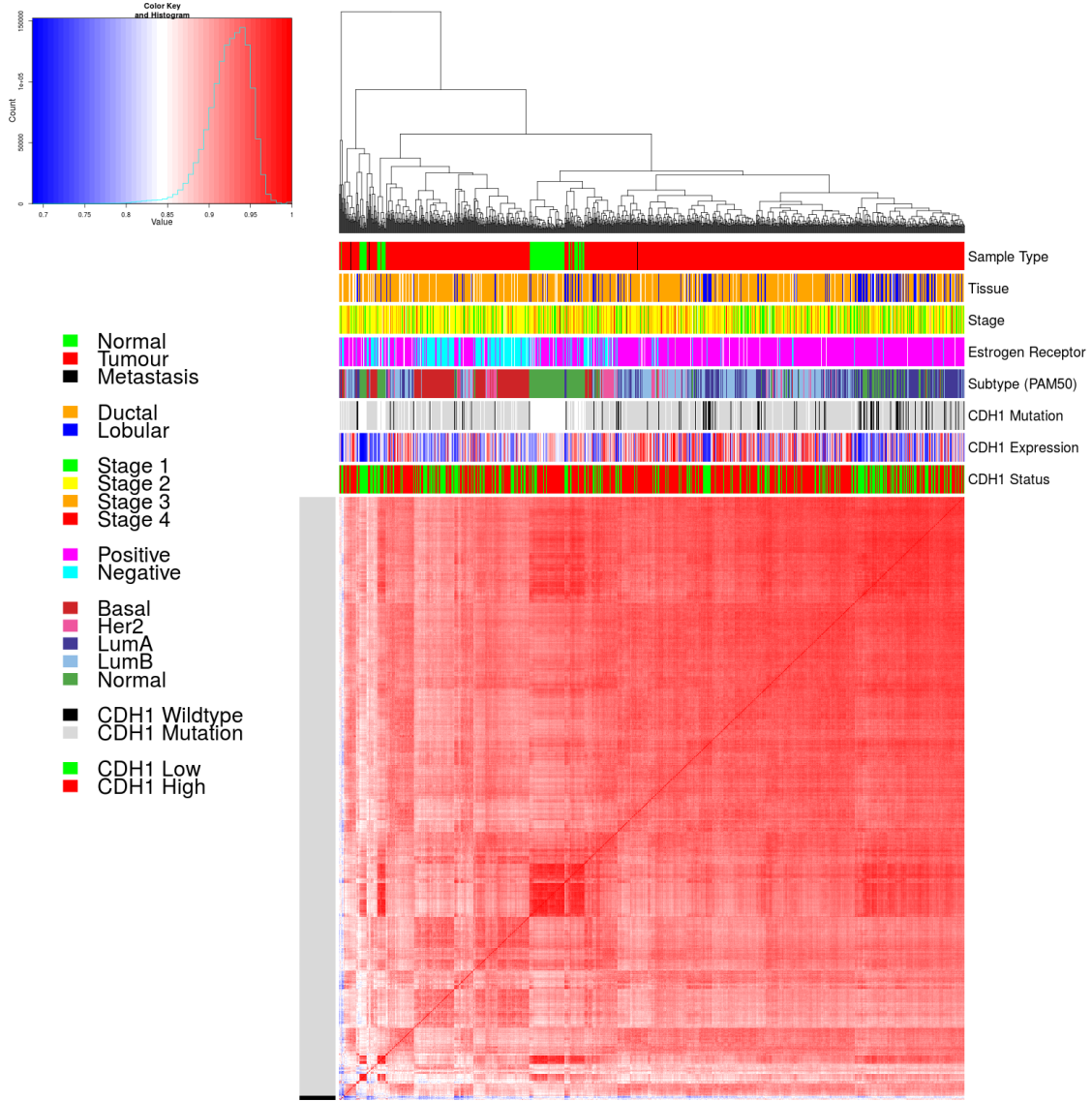


Figure 2.4: **Correlation analysis and sample removal.** Correlation matrix heatmap (Euclidean distance) of all samples in TCGA breast cancer dataset against each other annotated for sample clinical data: sample type, tissue type, tumour stage, Estrogen receptor (IHC) and intrinsic subtype (from the 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.)

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. Adjusting for multiple comparisons was not needed here in a permutation analysis as the test χ^2 statistics were directly used with the same degrees of freedom between expected and observed. 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. These empirical p-values were adjusted for multiple comparisons (FDR) and pathways for which no resamplings were observed as high as the observed were assumed to be zero for the purposes of FDR and reported as less than the adjusted value of one in a million resamplings. 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).

Techniques

Various statistical, computational, and bioinformatics techniques were performed throughout this thesis. This section presents these techniques and where relevant acknowledges the R package implementation which provided it and gives the parameters used in data presented in the thesis unless otherwise specified.

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. Here the 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 heatmap, any data more than 3 standard deviations from the mean is plotted as 3.

Differential expression analysis was performed with empirical Bayes, using the `limma` R package (Ritchie *et al.*, 2015). Where specified, the Fisher’s exact test, χ^2 test, and correlation are used to measure associations between variables (as implemented in the `stats` R package (R Core Team, 2016)). Unless otherwise specified, Pearson’s correlation is 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 will be supported by a table or Venn diagram, rendered with the `limma` R package (Ritchie *et al.*, 2015). Correlation analyses may be supported by a scatter plot and a line of best fit derived 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 these have been typically supported by visualisation of a boxplot rendered in R. 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.

Multiple comparisons are adjusted by Benjamini-Hochberg procedure to control the false discovery rate (FDR) unless otherwise specified (Benjamini and Hochberg, 1995). This procedure adjusts p-values to an estimate of the proportion of false-positives among tests with this p-value or lower. The more stringent Holm-Bonferroni (Holm) procedure (Holm, 1979) was also applied in some cases to adjust for multiple comparisons by the family-wise error rate which adjusts p-values to the probability that any one of the tests with this p-value or lower is negative.

Gene Set Over-representation Analysis

Gene set enrichment over-representation tests 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 as defined by clustering (in section 2.3.3) or comparison with experimental candidates (see section 2.2.3.1). Initially, these 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 (Araki *et al.*, 2012)) were excluded before adjusting for multiple comparisons.

Clustering

Clustering analysis when performed uses unsupervised hierarchical clustering with complete linkage (distance calculated from the furthest possible pairing). When data has been generated relates to correlations, such as correlation matrices or Sigma (multivariate normal) parameters, the distance measure used was Euclidean distance. For empirical or simulated gene and pathway expression data correlation distance is used, calculated by $distance = 1 - cor(t(x))$ where `cor` is Pearson’s correlation and $t(x)$ is the transpose of the expression matrix.

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 has been split with clustering performed separately on each with these plotted alongside each other.

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) 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 sampled once with the procedure to add known synthetic lethal partners, 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 of > 0.2 and < -0.2 for correlation and negative 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 important 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, in this case the total correctly predicted known synthetic lethal partners and correctly negative known non-synthetic lethal partners.

Receiver Operator Curves (Statistical Performance)

A more general procedure to measure the statistical performance of a simulation is the Receiver Operator Curve (ROC) which does not assume a threshold for classification of synthetic lethality but demonstrates the trade-off of sensitivity and specificity. These curves (implemented with the **ROCR** R package) plot the True Positive Rate (sensitivity) against the False Positive Rate ($1 - \text{specificity}$) 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. An ideal detection method will have an AUROC of 1 and a poor detection method performs worse than random chance if the AUROC is below 0.5. Generally an AUROC in the range of $[0.7, 0.8]$ is achievable and regarded as acceptable for publication. However, in this

thesis, the AUROC values varies widely across simulation parameters and a primarily used for comparisons across these parameters.

Permutation and Resampling Analysis

Permutation and resampling analyses are used to statistically test the significance of an observation without assuming the underlying distribution of expected test statistics. 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 as the observed synthetic lethal were to gain an expected value of the test statistic or network metrics. Sampling iteratively across many resampling procedures, these expected values form a null distribution of test statistics that would be observed if the null hypothesis could not be rejected. Thus the proportion of expected test statistics across these iterations greater or equal to that observed forms an empirically derived p-value which is used to determine the significance of pathway composition or number of overlapping genes (in section 2.2.3.1) and the significance of directional relationships within pathway structures (in section 3.4.1.1).

The number of iterations determines the accuracy of these p-values. For pathway composition (in section 2.2.3.1), 1 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), 1 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.

Pathway Structure Methods

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. Network theory is an interdisciplinary field which combines the approaches of com-

puter 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. 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.

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. 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 digraph.

The degree distribution of naturally occurring networks often follows a power law distribution with the majority of nodes having far fewer connections than average and a small subset of highly connected network ‘hubs’ (Barabási and Albert, 1999). Hubs further differentiate into ‘party’ hubs (which interact simultaneously with many partners) and ‘date’ hubs (which interact with different partners in different conditions) (Han *et al.*, 2004). Network hubs can also be classed as associative or dissociative depending on whether they tend toward or away from connecting directly to other network hubs (van Steen, 2010). The associative and dissociative properties can also be used to test whether nodes of a particular subgroup (e.g., gene function) associate

with each other.

Sourcing Graph Structure Data

Pathway Commons interaction data was sourced using the paxtools-4.3.0 java application on October 6th 2015. This utility was used to source ‘sif’ format interaction data into R, from which the human Reactome (version 52) of interactions was imported, 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.

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.

Network Analysis Metrics

Various existing network analysis measures were applied in this thesis, using an implementation in the `igraph` R package where available (Csardi and Nepusz, 2006). Various custom features were added for analysis of `iGraph` objects in R and released as `igraph.extensions` (as described in section 3.5.4).

Vertex degree is the number of edges a node has and is a fundamental measure of the importance and connectivity of a network. 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.

Another central concept in network analysis is shortest paths, that is the shortest route via edges between any two particular nodes in a network. These are computed by Dijkstra’s algorithm in `igraph` R package (Csardi and Nepusz, 2006). Where applicable paths will only use directed edges in a particular direction. Shortest paths are a useful as a measure of how close nodes are in a network, this is used to compute information

centrality here and later 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. 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.

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 is 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.

Implementation

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

Computational Resources and Linux Utilities

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 (L ^A T _E X)	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	T _E X	3.1415926	3.14159265		
	TexLive (L ^A T _E X)	2013	2015		
	PDF _T E _X	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			

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). 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 2.4: 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)	

R Language and Packages

Table 2.5: R Packages Developed During Thesis

	Package Name	Description and GitHub Repository	Description in Thesis
	<code>slipt</code>	Synthetic lethal detection by SLIPT (to accompany publication) https://github.com/TomKellyGenetics/slipt	Section 3.1
visualisation	<code>vioplotx</code>	Customised violin plots (based on <code>vioplot</code>) https://github.com/TomKellyGenetics/vioplotx	
	<code>heatmap.2x</code>	Customised heatmaps (based on <code>gplots</code>) https://github.com/TomKellyGenetics/heatmap.2x	Section 2.3.4
igraph.extensions	<code>igraph.extensions</code>	Meta-package to install the follow iGraph functions https://github.com/TomKellyGenetics/igraph.extensions	Section 3.5.4
	<code>plot.igraph</code>	Custom plotting of directed graphs https://github.com/TomKellyGenetics/plot.igraph	Section 2.4.4
	<code>info.centraliity</code>	Computing information centrality from network efficiency https://github.com/TomKellyGenetics/info.centraliity	Section 3.4.2
	<code>pathway.structure.permutation</code>	Testing pathway structure with resampling analysis https://github.com/TomKellyGenetics/pathway.structure.permutation	Section 3.4.1.1
	<code>graphsim</code>	Generating simulated expression from graph structures https://github.com/TomKellyGenetics/graphsim	Section 3.4.2

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 relevant. Various R packages were used throughout this thesis (as detailed in Table 2.6 with versions specified) these have not been updated if they would change the functionality of the 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 to their dependencies) have been used throughout this project. Where a package implementation has been central to the methods applied, they are described

in the relevant sections as well. A full table of packages used in this thesis can be found in the Appendix (Table A.1).

Table 2.6: R Packages Used During Thesis

hline			
Package	Latest Version	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

High Performance and Parallel Computing

Another enabling technology for bioinformatics is parallel computing, performing independent operations in separate cores, such “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 once such computational resource providing the Intel Pan cluster (hosted by the University of Auckland) used throughout this thesis project to optimise and perform computations which would have otherwise been infeasible in the timeframe of thesis. This is another example of how technological developments and infrastructure has enabled research including this project.

Chapter 3

Methods Developed During Thesis

Here we 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 this describes the Synthetic Lethal Interaction Prediction Tool (SLIPT), a bioinformatics approach to synthetic lethal candidate triage. 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 will be provided alongside description of the method. This includes the construction of a statistical model of synthetic lethality in (continuous multivariate Gaussian) gene expression data. This 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 here. Various R packages have been developed to support this project, most notably the `slipt` package to implement the SLIPT methodology. Numerous other 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.

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, the p-values for the χ^2 test were 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 are less frequent than expected; and (iii) observed low-query, high-candidate and high-query, low-candidate samples are more frequent than expected.

		Candidate Gene (e.g. <i>SVIL</i>)		
		Low	Medium	High
Query Gene (e.g. <i>CDH1</i>)	Low	Observed less than expected	→	Observed more than expected
	Medium	↓		
	High	Observed more than expected		

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 is low in both genes of a synthetic lethal pair is unlikely 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 direction applies strictly and synthetic lethal genes will predicted to be partners of each other.

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

disrupting 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. Where these are compared, we contrast those generated entirely from expression data (exprSL) with those using mutation data for the query gene (mtSL). Unless otherwise specified, computationally predicted synthetic lethal gene candidates from SLIPT used the exprSL procedure.

		Candidate Gene (e.g. <i>SVIL</i>)		
		Low	Medium	High
Query Gene (e.g. <i>CDH1</i>)	Mutation	Observed less than expected	→	Observed more than expected
	Wild-type	↓ Observed more than expected		

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. Synthetic lethal genes are partners of each other commutatively with synthetic lethal genes will predicted to be partners of each other.

Supporting 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 with the strongest validation

being independent replication with 4 gene knockdowns with different targeting siRNA. As shown in Table 3.1, 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 3.1: Candidate synthetic lethal genes against secondary siRNA screen

		Secondary Screen					total
		0/4	1/4	2/4	3/4	4/4	
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

Simulations and Modelling Synthetic Lethality in Expression Data

A statistical model of Synthetic Lethality was developed upon which to test the SLIPT procedure on simulated data. 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 a later chapter, including in combination with simulations from graph structures.

A Model of Synthetic Lethality

We first construct a conceptual model of synthetic lethality (see Figure 3.3). This will be used to build a statistical model of synthetic lethal gene expression from which to

simulate expression data to test SLIPT and various potential synthetic lethal prediction methods upon. In the model, synthetic lethality arises from related functions and manifests in a cell death phenotype.

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. All of these potential mechanisms of gene inactivation could lead to synthetic lethality but 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 performed poorly in previous studies as a predictor of synthetic lethality, 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. This does not preclude mutual exclusivity, compensating expression, or co-loss under-representation as discussed in previous studies.

The first condition of the synthetic lethal model is that if there are only 2 synthetic lethal genes (1 true partner), then they will not both be non-functional in the same sample (in an ideal model). The frequency of gene function is an arbitrary level for non-functional and functional samples, for the purposes of modeling synthetic lethality we have used 30%, an estimate of the number of non-functional E-cadherin samples in the TCGA breast cancer dataset. In this ideal case, we would not expect to observe any samples lowly expressing both of these genes. 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.

A synthetic lethal pair of genes is unlikely to act in isolation, higher-order synthetic lethal interactions (or 3 or more genes) must be considered in the model as shown in Figure 3.4. Even when testing pairwise interactions, modelling higher level interactions that may interfere with it is important. If there are additional synthetic lethal partners, there are two possibilities for adding these: 1) that they are independent partners of the query genes interacting pairwise (and not with each other) or 2) that it interacts with both of the synthetic lethal genes already in the system and any of the 3 (or more) are required to be functional for the cell to survive. The signal in gene expression data will be weaker for this latter case and thus we make the more stringent assumption that all synthetic lethal partner genes interact with one and other: that only one of

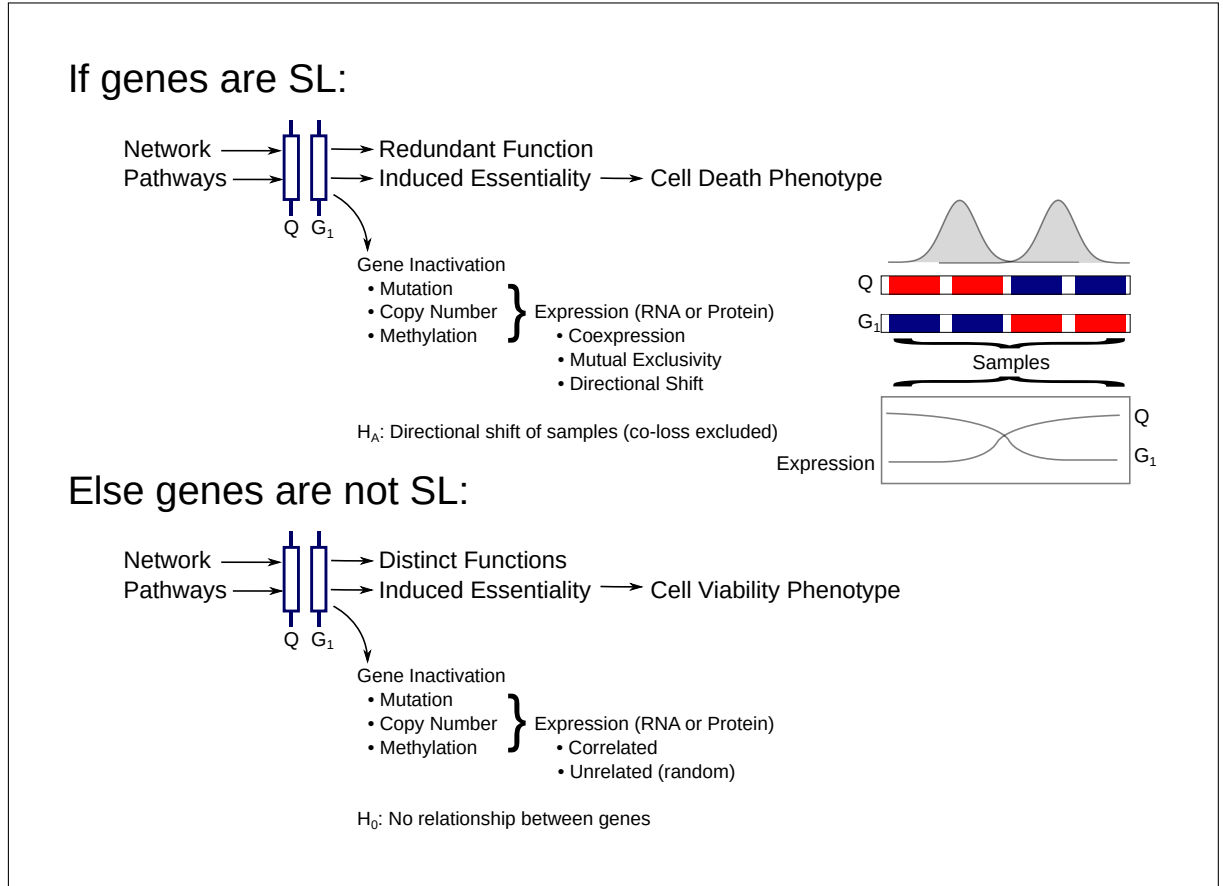


Figure 3.3: **A model of Synthetic Lethality.** A conceptual model of synthetic lethal interactions between a Query gene (Q) and partner gene (G_1). Genes that are synthetic lethal may not be non-functional in the same sample without another gene compensating for the loss of function. This is most likely to be detectable as lowly expressed gene, whether they are lost by mutation, copy number loss, 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 exclude co-loss samples while the null hypothesis (H_0) is that in 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 probably of detecting samples with combined gene loss.

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

Higher-order Synthetic Lethality

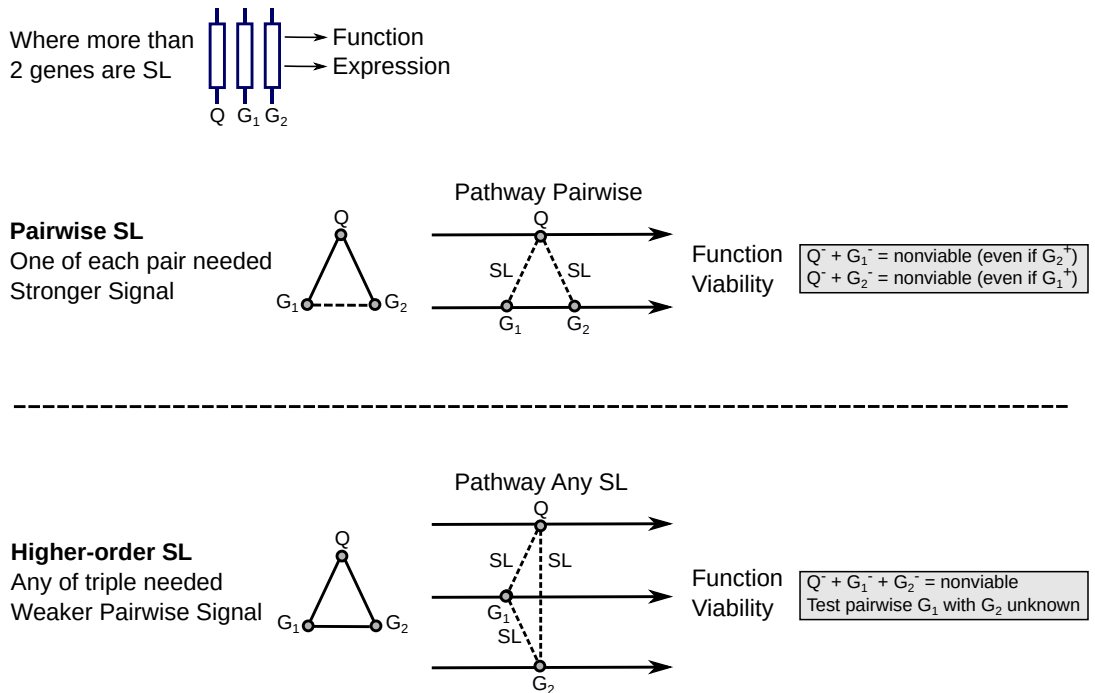


Figure 3.4: **Modeling Higher-order Synthetic Lethality.** Higher order synthetic lethal interactions may occur between 3 or more genes, which may impact the simulated expression pattern (or observed synthetic lethal predictions) whether or not that these higher-order interactions are known to a pair-wise observer. This situation is illustrated with synthetic lethal interactions between a Query gene (Q) two partner genes (G_1 and G_2). Each partner gene may interact with Q pairwise with synthetic lethality manifesting as cellular inviability when either gene pair is lost. Alternatively, the synthetic lethal genes may form a higher-order interaction such as the “synthetic lethal triplet” presented here. In this case, any of the genes are able to provide the essential function of the synthetic lethal group, with loss of all required for synthetic lethality to manifest as cellular inviability. Both of these higher-order interactions are biological plausible, for instance depending on whether the partner genes are directly upstream or downstream of each other in a biological pathway.

function of the others, allowing for higher-level synthetic lethal partners to compensate for loss a synthetic lethal gene pair. While samples lowly expression synthetic lethal

gene pairs will be under-represented, they may not be completely excluded from the dataset due to these higher-level interactions.

To model synthetic lethal gene expression, several synthetic lethal genes are considered. In Figure 3.5, an example of 3 synthetic lethal genes is presented, where any of the higher-order synthetic lethal interaction are required for cell viability. Expression data is assumed to be continuous, normally distributed data with an arbitrary threshold to define gene function. As discussed above this threshold is $pr = 0.3$ throughout this thesis but is amenable to further modifications. Thus, if we consider synthetic lethal pairs, only a subset of these samples will be excluded, leading to an under-representation.

In practice, samples are not removed from a dataset, rather the query gene is added to satisfy the synthetic lethal condition as outlined in section 3.2.2 below. This is performed to maintain a comparable samples size across simulations and the preserve the assumed (multivariate) normal distribution of the data.

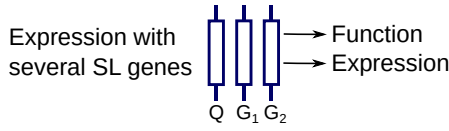
Simulation Procedure

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 gives a more comparable test of SLIPT (and other methods) on simulated data compared to normalised empirical gene expression data. This also has the added advantage of being amenable to simulating correlation structures with the Multivariate normal distribution. In Figure 3.6, an example of such a 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. This is done by ordering the samples by functional data (without assuming correlation of underlying expression values) with the query gene in one direction and the remaining dataset ordered by the selected synthetic lethal partner.

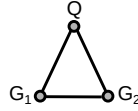
This results in the simulated dataset where sample with non-functional query gene

Synthetic Lethal Gene Expression



Higher-order SL

Any of triple needed
Weaker Pairwise Signal



$Q^- + G_1^- + G_2^- = \text{nonviable}$
Test pairwise G₁ with G₂ unknown

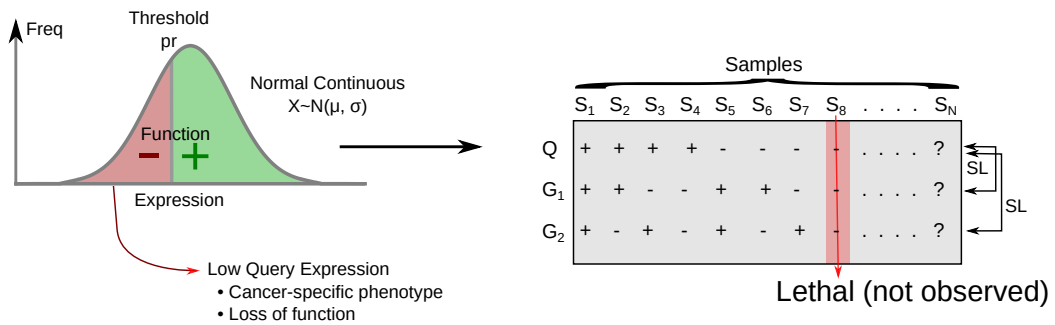


Figure 3.5: Modeling Synthetic Lethal Gene Expression. Here we model synthetic lethal interactions between a Query gene (Q) two partner genes (G₁ and G₂) where any of the triplet are able to maintain cellular viability when the other genes are lost. Expression is modeled a normally (Gaussian) distributed continuous data which could be RNA (microarray or RNA-Seq) or Protein expression levels. 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. Genes with loss of function may also lead to cancer specific phenotypes if they are tumour suppressors (although these thresholds are not necessarily the same). In the case of a synthetic lethal triple there are 8 potential combinations of functional groups, however only 7 of these are permitted under the SL model as inactivation of all genes in inviable. However, pairwise inactivation may be observed if the additional partner genes are functional, whether or not these are known or observed. Note that these combinations will not be observed in equal proportions but will depend on the functional threshold (pr).

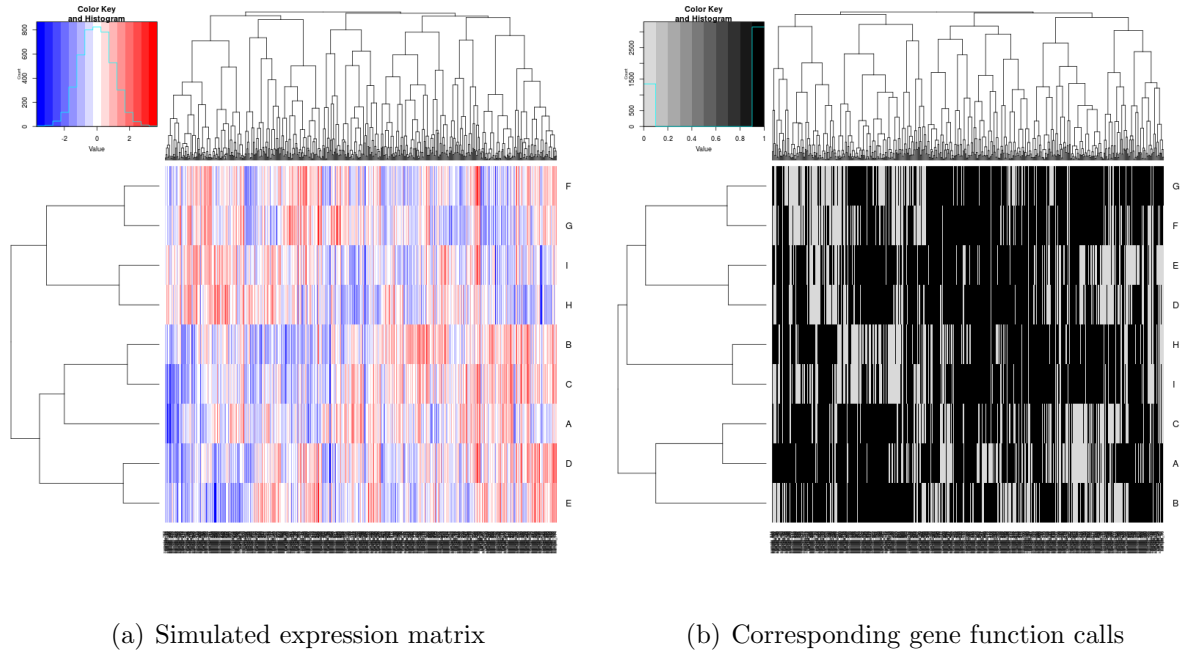
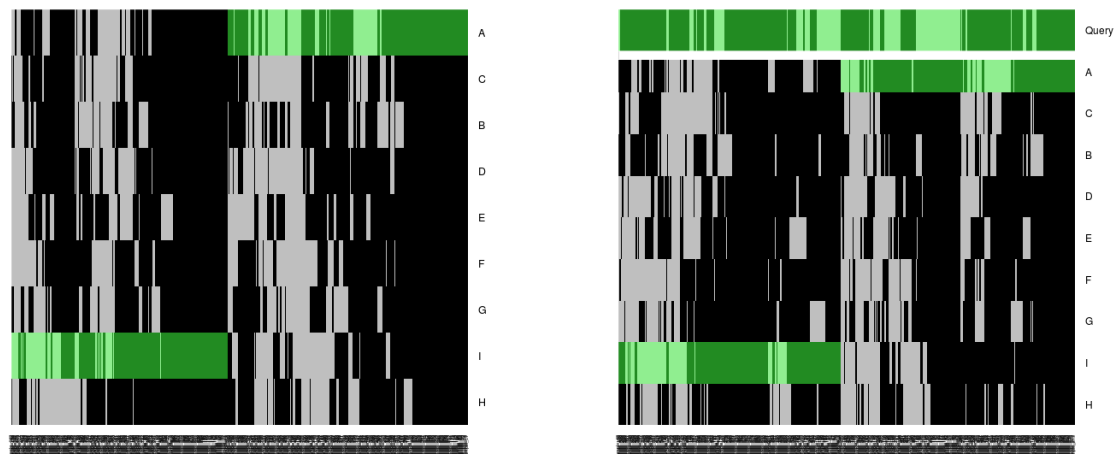


Figure 3.6: **Simulating gene function from gene expression.** Simulated data with samples (columns) and genes (row) showing how a simulated dataset is transformed from a continuous dataset (on a blue to red colour scale) to a discrete matrix of gene function (shaded in black for samples with functional gene levels).

has loss of function in all of the synthetic lethal partners as at least one has been required to be functional in each sample, similarly the query gene is functional where the synthetic lethal partner genes are non-functional. Therefore we have generated a dataset 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.

Detecting Simulated Synthetic Lethal Partners

Test



(a) Simulated gene function with SL (partner) genes

(b) Query gene added with SL condition

Figure 3.7: Simulating synthetic lethal genes with gene function. Simulated data with samples (columns) and genes (row) in a discrete matrix of gene function (shaded in black for sample with functional gene levels). Genes A and I are selected to be synthetic lethal partners of a “Query” gene, which of these genes will be the true partner in each sample is selected randomly and indicated in green which samples are considered for the purposes of simulating synthetic lethality (shaded in forest green for samples with functional gene levels). Note that samples are ordered such that either the query gene or selected partner are functional in any particular sample.

Multivariate Normal Simulation of Synthetic lethality

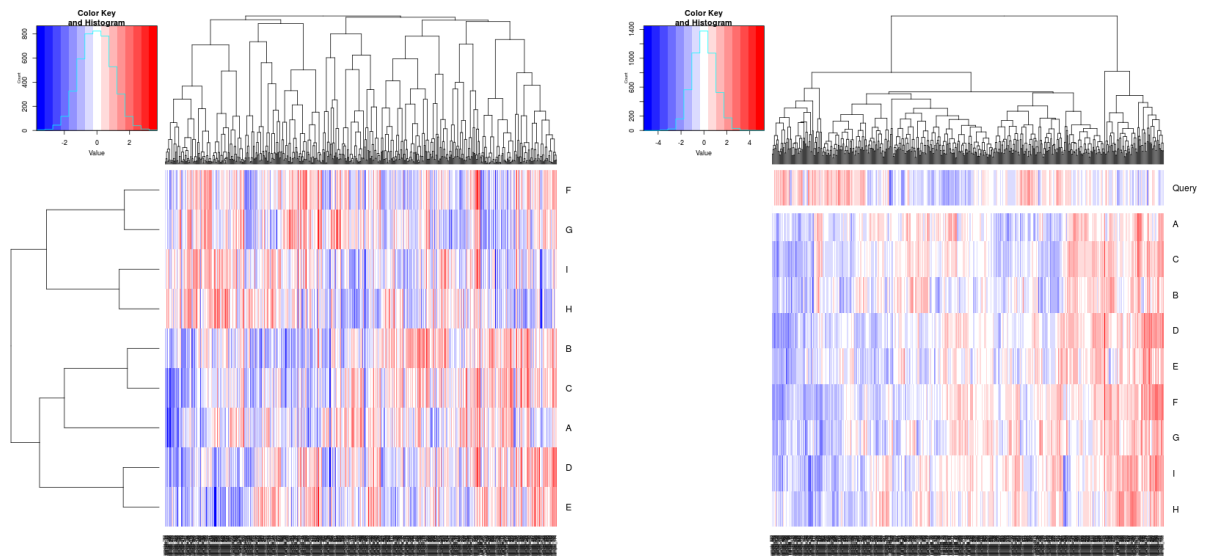
Test

Simulation with Directional Testing

Test

Simulation with Query-Correlated Pathways

Test



(a) Initial expression matrix

(b) Simulated synthetic lethal dataset

Figure 3.8: **Simulating synthetic lethal gene expression.** Simulated data with samples (columns) and genes (row) showing how a simulated continuous dataset (on a blue to red colour scale) 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.

Simulated Expression Heatmaps

Test

Replication Simulation Heatmap

Test

Graph Structure Methods

Test

Upstream and Downstream Gene Detection

Test

Permutation Analysis for Statistical Significance

Test

Simulating Gene Expression from Graph Structures

A further refinement of the simulation procedure is to generate expression data with correlation structure derived from a known graph structure. This enables modelling of synthetic lethal partners within a biological pathway and the impact of pathway structure on synthetic lethal prediction. First we construct a simulated pathway with a graph structure with the `igraph` R package with the added provision of including the state of the edges, i.e., whether the activate or inhibit downstream pathway members. Here we consider purely whether biological pathway members would be expected to have correlated gene expression (higher than the background of genes in other pathways) but this framework is as applicable to modelling protein levels in a kinase regulation cascade or metabolic pathway with related substrates and products as it is to gene regulation as it has been applied here.

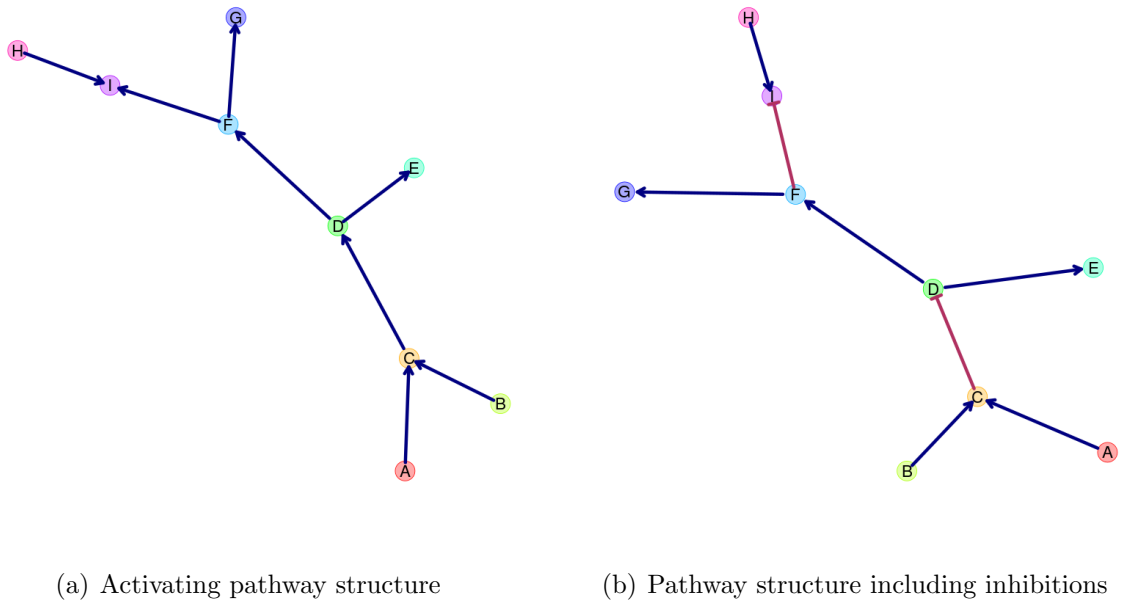


Figure 3.9: **Simulating graph structures.** An example graph structure which will be used throughout demonstrating the simulation procedure from graph structures. Here activating links are denoted by blue arrows and inhibiting links by red edges.

First we must consider the graph structure upon which simulated data will be generated (by sampling a multivariate normal distribution). Throughout this section, the simulation procedure will be demonstrated with the relatively simple constructed graph structure shown in Figure 3.9. This graph structure visualisation was developed specifically for (directed) `igraph` objects in R and has been released in the `plot.igraph` package and `igraph.extensions` library. The `plot_directed` function allows customisation of plot parameters for each node or edge and mixed (directed) edge types for indication activation or inhibition. These inhibition links (which often occur in biological pathways) are demonstrated in Figure 3.9(b).

The simulation procedure is designed to use such graph structures to inform development of a “Sigma” correlation-covariate matrix for sampling a multivariate normal distribution (using the `mvtnorm` R package). Given a graph structure (or adjacency matrix), such as Figure 3.10(a), a relation matrix is calculated based on distance such that nearer nodes are weighted higher than farther nodes. For the purposes of this thesis a geometrically decreasing (relative) distance weighting is used, with each more distant node being related by $\frac{1}{2}$ compared to the next nearest as shown in Figure 3.10(b). However, an arithmetically decreasing (absolute) distance weighting is also available in the `graphsim` R package release of this procedure.

A Sigma matrix is derived from this distance weighting matrix by creating a matrix where each node is correlated with itself (diagonal of 1), correlated to adjacent nodes by a correlation parameter (`cor`), 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, `cor` = 0.8, unless otherwise specified as used for the example in Figure 3.10(c). This Sigma matrix is used to sample from a multivariate normal distribution where each node has a mean of 0, standard deviation 1 and covariates within the range [0, 1]. This generates a simulated (continuous normally distributed) expression profile for each node (as shown in Figure 3.10(d)) with corresponding correlation structure (Figure 3.10(e)). This simulated correlation structure closely resembles the expected correlation structure (Sigma in 3.10(c)) even for the relatively modest sample size ($N = 100$) illustrated in 3.10. Once a simulated gene expression dataset has been generated (as in Figure 3.10(d)) then a discrete matrix of gene function can be constructed with a functional threshold quantile (the parameter `pr`) to simulate functional relationships of synthetic lethality. For the purposes of this thesis the parameter `pr` is set to the 0.3 quantile which generates functional discrete matrices such as those used for synthetic lethal simulation in section

3.2.2 (as shown Figure 3.10(f))

The simulation procedure discussed in Figure 3.10 is amenable to pathways containing inhibition links (as shown in Figure 3.11) with a few refinements. With the inhibition links (as shown in Figure 3.11(a)), the distances are calculated in the same manner as before (Figure 3.11(b)) but the inhibitions are accounted for by iteratively multiplying downstream nodes by -1 to form blocks of negative correlations (as shown in Figures 3.11(c) and 3.11(e)). As before, a multivariate normal distribution with these negative correlations can be sampled to generate simulated data (as shown in Figures 3.11(d) and 3.11(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.

Customised methods and R packages developed

test

slipt

test

plotting

test

simulation from graph structures

test

igraph methods

test

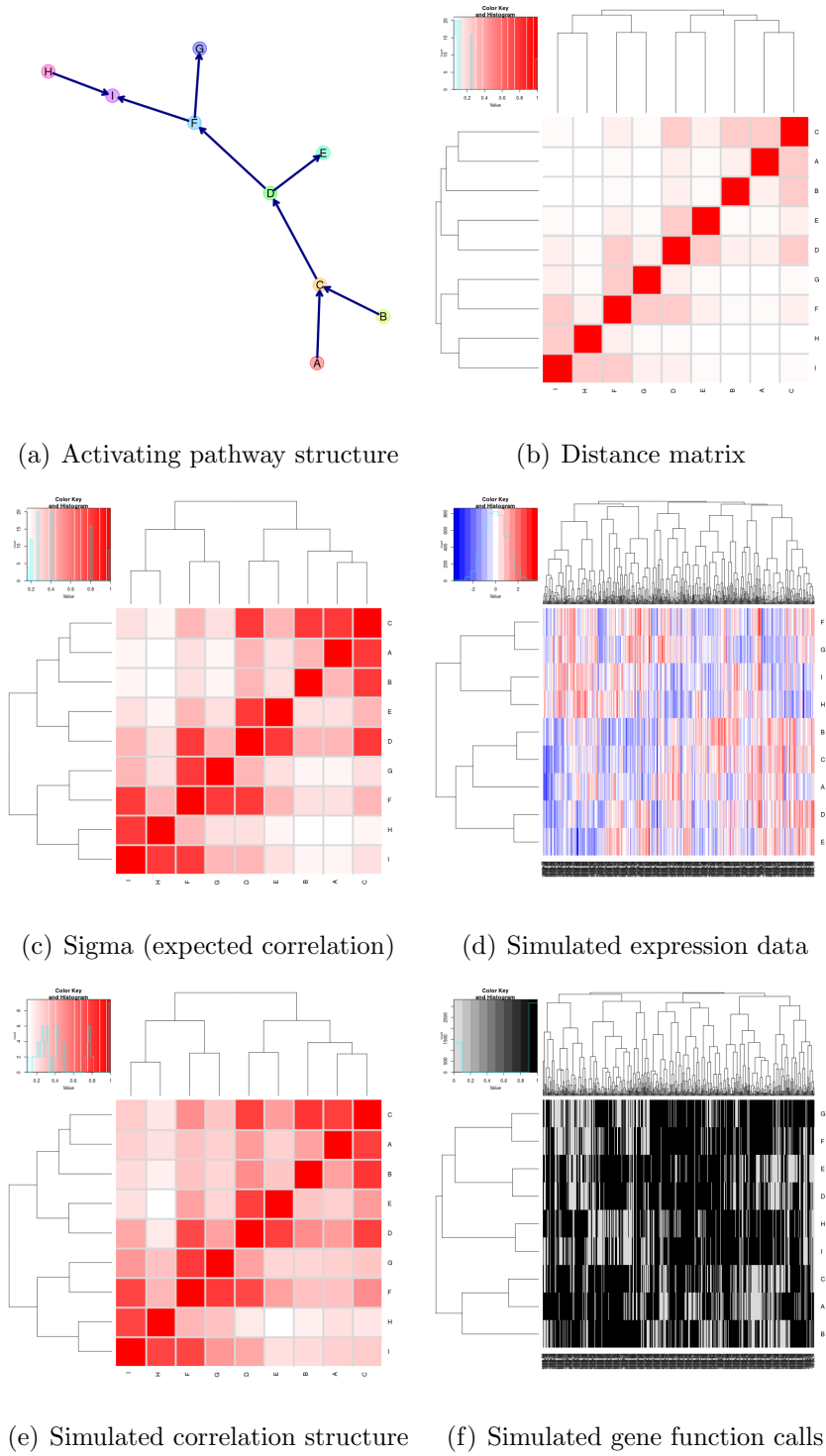


Figure 3.10: **Simulating expression from a graph structure.** An example graph structure is used to derive a correlation structure from the relative distances between nodes and simulate continuous gene expression with sampling from the multivariate normal distribution.

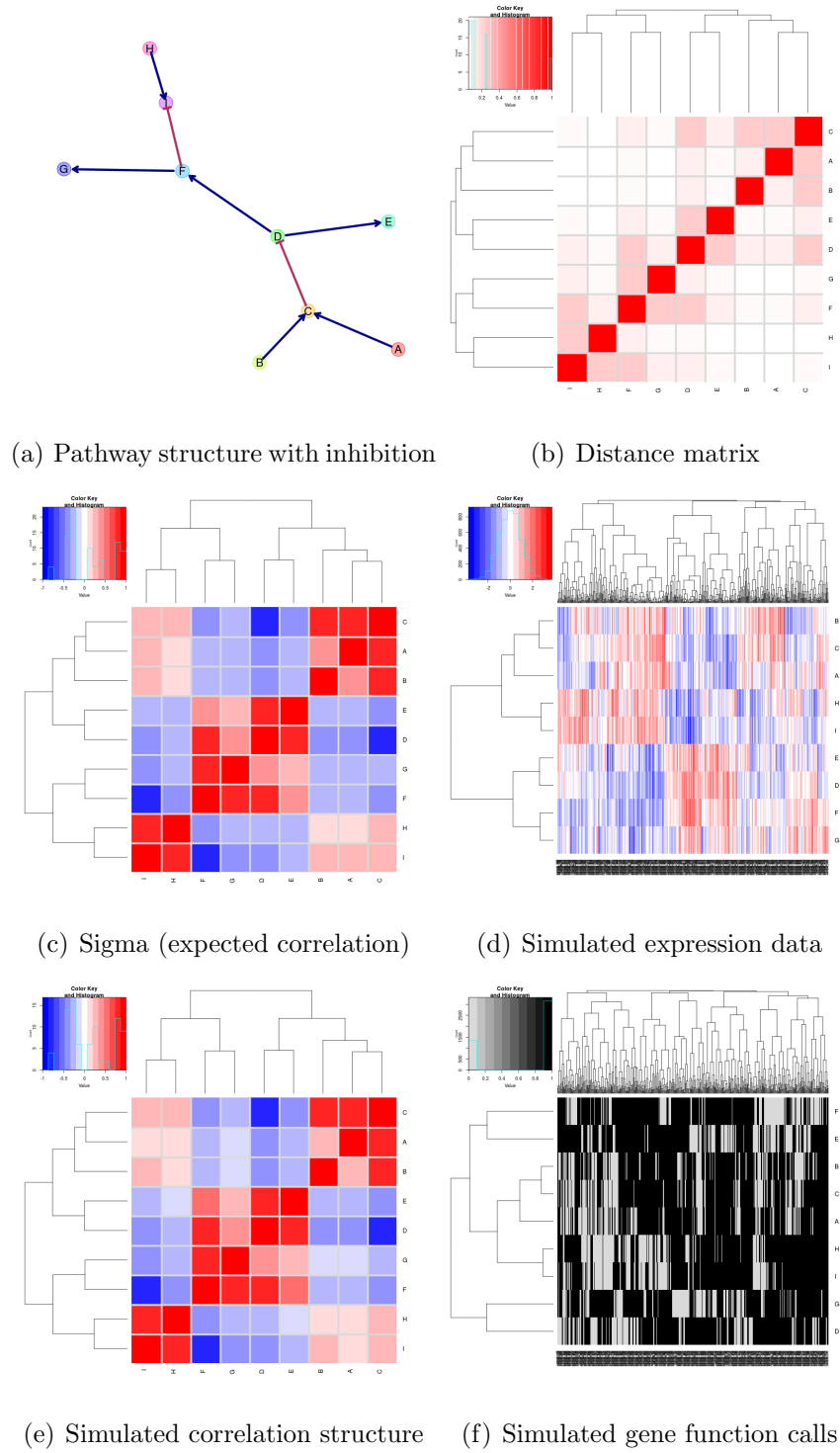


Figure 3.11: **Simulating expression from graph structure with inhibitions.** An example graph structure is used to derive a correlation structure from the relative distances between nodes and simulate continuous gene expression with sampling from the multivariate normal distribution.

References

- 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 Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509–12.
- Barry, W. T. (2016). *safe: Significance Analysis of Function and Expression*. R package version 3.14.0.
- 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.
- Bernard, P., Parker, J., Mullins, M., Cheung, M., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., Quackenbush, J., Stijleman, I., Palazzo, J., Matron, J., Nobel, A., Mardis, E., Nielsen, T., Ellis, M., and Perou, C. (2009). Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology*, 27(8), 1160–1167.
- 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.
- 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.
- Croft, D., Mundo, A. F., Haw, R., Milacic, M., Weiser, J., Wu, G., Caudy, M., Garapati, P., Gillespie, M., Kamdar, M. R., Jassal, B., Jupe, S., Matthews, L., May,

- B., Palatnik, S., Rothfels, K., Shamovsky, V., Song, H., Williams, M., Birney, E., Hermjakob, H., Stein, L., and D'Eustachio, P. (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.
- Han, J.-D. J., Bertin, N., Hao, T., Goldberg, D. S., Berriz, G. F., Zhang, L. V., Dupuy, D., Walhout, A. J. M., Cusick, M. E., Roth, F. P., and Vidal, M. (2004). Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430(6995), 88–93.
- 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., West, M., Nevins, J., and Huang, A. (2003). Gene expression predictors of breast cancer outcomes. *Lancet*, 361, 1590–1596.
- 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.
- 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.
- Network, C. G. A. R. (2012). Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418), 61–70.
- R Core Team (2016). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. 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.
- 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.
- van Steen, M. (2010). *Graph Theory and Complex Networks: An Introduction*. VU Amsterdam: Maarten van Steen.
- Warnes, G. R., Bolker, B., Bonebakker, L., Gentleman, R., Liaw, W. H. A., Lumley, T., Maechler, M., Magnusson, A., Moeller, S., Schwartz, M., and Venables, B. (2015). *gplots: Various R Programming Tools for Plotting Data*. R package version 2.17.0.
- Zhang, J., Baran, J., Cros, A., Guberman, J. M., Haider, S., Hsu, J., Liang, Y., Rivkin, E., Wang, J., Whitty, B., Wong-Erasmus, M., Yao, L., and Kasprzyk, A. (2011). International Cancer Genome Consortium Data Portala one-stop shop for cancer genomics data. *Database: The Journal of Biological Databases and Curation*, 2011, bar026.

Appendix A

Software Used for Thesis

Table A.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