Computational Methods for Systems Biology Data of Cancer

by

Jiarui Ding

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

May 2016

Abstract

High-throughput genome sequencing and other techniques provide a cost-effective way to study cancer biology and seek precision treatment options. In this dissertation I address three challenges in cancer systems biology research: 1) predicting somatic mutations, 2) interpreting mutation functions, and 3) stratifying patients into biologically meaningful groups.

Somatic single nucleotide variants are frequent therapeutically actionable mutations in cancer, e.g., the 'hotspot' mutations in known cancer driver genes such as *EGFR*, *KRAS*, and *BRAF*. However, only a small proportion of cancer patients harbour these known driver mutations. Therefore, there is a great need to systematically profile a cancer genome to identify all the somatic single nucleotide variants. I develop methods to discover these somatic mutations from cancer genomic sequencing data, taking into account the noise in high-throughput sequencing data and valuable validated genuine somatic mutations and non-somatic mutations.

Of the somatic alterations acquired for each cancer patient, only a few mutations 'drive' the initialization and progression of cancer. To better understand the evolution of cancer, as well as to apply precision treatments, we need to assess the functions of these mutations to pinpoint the driver mutations. I address this challenge by predicting the mutations correlated with gene expression dysregulation. The method is based on hierarchical Bayes modelling of the influence of mutations on gene expression, and can predict the mutations that impact gene expression in individual patients.

Although probably no two cancer genomes share exactly the same set of somatic mutations because of the stochastic nature of acquired mutations across the three billion base pairs, some cancer patients share common driver mutations or disrupted pathways. These patients may have similar prognoses and potentially benefit from the same kind of treatment options. I develop an efficient clustering algorithm to cluster high-throughput and high-dimensional biological datasets, with the potential to put cancer patients into biologically meaningful groups for treatment selection.

Preface

A version of Chapter 2 has been published in *Bioinformatics* as an original paper [46]. I implemented the `mutationSeq` algorithm and performed all computational analyses except DNA sequencing and most of the work of aligning short DNA sequences to a reference human genome. I co-wrote the text with Dr. Sohrab Shah and Dr. Anne Condon. The genome sequencing data were provided by project leaders Drs. Samuel Aparicio and Sohrab Shah as part of the study "Shah *et al.* (2012). The clonal and mutational evolution spectrum of primary triple-negative breast cancers. *Nature*, 486(7403):395-399."

A version of Chapter 3 has been published in *Nature Communications* as an article [47]. Most of the data used in this chapter were from The Cancer Genome Atlas project. The METABRIC data were from the study "Curtis, *et al.* (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346-352." I implemented the `xseq` software and performed all computational analyses described in this chapter. I co-wrote the text with Dr. Sohrab Shah and Dr. Anne Condon.

A version of Chapter 4 has been published in *Bioinformatics* as an original paper [49]. I implemented the `densityCut` software and performed all the computational analyses described in this chapter. I co-wrote the text with Dr. Anne Condon and Dr. Sohrab Shah. All the data used in this chapter were from previously published works.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I'm greatly indebted to my supervisor Professor Sohrab Shah, who introduced me to computational cancer biology when I was a novice in the field. During the years in his lab, Sohrab inspired me to gain a deep understanding of cancer biology and statistical modelling. I'm extremely excited to have the opportunity to work with real patient data and to know that my research could benefit patient care. I would like to extend my deepest gratitude to my co-supervisor Professor Anne Condon, for all her guidance, dedication, as well as the challenging questions in our meetings. Her intellectual rigour pushes me to think critically of the research problems.

I would also like to express my gratitude and appreciation for Professors Sam Aparicio, Will Evans, and David Huntsman for their additional supervision, insightful comments, questions, and feedback. Special thanks go to collaborators Melissa McConechy, Gavin Ha, Ali Bashashati, Andrew Roth, Fong Chun Chan, Anthony Mathelier, Calvin Lefebvre, Hugo Horlings, Tyler Funnell, Sarah Mullaly, Ryan Morin, Alicia Tone, Gholamreza Haffari, Jüri Reimand, Gary Bader, Pier-Luc Clermont, and other coauthors. Thanks also go to the people help me out in one way or another, especially Chris Thachuk from the Condon lab and Professor Laks Lakshmanan.

Finally, I would like to thank my wife, Rachel, for putting up with my nights and weekends in the office for so long, and for giving me the motivation and encouragement to finish this journey.

Dedication

To my parents

and

to Rachel and Anna Luxin

Chapter 1

Overview of the thesis

"It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only."

– Charles Dickens, A Tale of Two Cities, 1859

1.1 Cancer and cancer genome sequencing studies

In a multicellular organism, the cells behave in a social, cooperative manner: they differentiate, grow, proliferate, rest or die for the benefit of the organism as a whole [3]. Cancer cells, on the contrary, violate these fundamental rules of cell behaviour by which multicellular organisms are built and maintained [3]: they grow selfishly at the expense of normal cells; they invade neighbour tissues and build colonies in other parts of the body; eventually they destroy the entire organism. Douglas Hanahan and Robert Weinberg summarized these cancer cell behaviours into ten hallmarks such as sustaining proliferative signals, evading growth inhibitory signals and resisting cell deaths [84].

Cancer cells obtain these hallmarks by accumulating a set of (as small as three in adult solid tumours) critical heritable genetic and epigenetic aberrations [9, 196, 213, 214]. For many types of cancer (originated in different sites of the human body), it takes years for a cell to finally obtain the set of critical genetic and epigenetic aberrations. Therefore, cancer is an evolutionary process involving successive rounds of random mutations (or epigenetic alterations, but mutations appear to be the fundamental and universal feature) followed by natural selection [8, 79, 154, 213]: a cell randomly acquires a mutation (as a result of an un-repaired DNA replication error or an un-repaired DNA damage) which gives the cell some growth advantages; this cell grows and

proliferates slightly more vigorously than its neighbours to form an adenoma (benign tumour); the second round of clonal expansion is triggered by an additional critical mutation in a cell of the adenoma; several rounds of a critical mutation followed by clonal expansion transform a normal cell to a cancerous cell. The progression of colorectal cancer is a classic example of clonal evolution in cancer [213]: for example, it takes about 20 years for a normal epithelial cell to sequentially acquire mutations in genes *APC*, then *KRAS*, and finally *PIK3CA* to become cancerous.

Because of the limitations of various DNA repair mechanisms, a daughter cell can accumulate a few mutations [202, 230]. Many years of cell proliferation produces a full-blown cancer consisting of billions of cells with heterogeneous genomes in different spatial sections of a solid cancer. It is also possible that spatially distinct regions of a primary solid tumour harbour different sets of critical mutations that drive the development of cancer. Similarly, heterogeneities are common within a metastasis and among metastases of a patient, as well as between two cancer genomes of different patients [4, 213]. Given the universal feature of genetic heterogeneity in cancer, what is extraordinary about cancer is that billions of cancer cells of a typical tumour share the same set of critical mutations that drive the initialization and progression of cancer [193, 214].

Our understanding of cancer biology has grown enormously thanks to technical advances such as sequencing technologies. Current high-throughput massively parallel sequencing devices provide a cost-effective way to sequence the whole genomes of a bulk of cancer cells to identify all the genetic aberrations. These instruments typically generate billions of short reads of dozens to hundreds of nucleotides. Each read corresponds to a DNA segment from the original cancer cells (Fig. 1.1). By aligning the short reads to a reference genome [113, 114, 121, 122], different kinds of genetic abnormalities can be detected [54, 143]: 1) Single nucleotide variants (SNVs) – nucleotide substitution mutations. SNVs in specific sites (mutation 'hotspots') of genes such as *IDH1*, *KRAS*, *NRAS*, *BRAF*, *PIK3CA* and in two hotspots of the promoter region of the gene *TERT* have been demonstrated to promote cancer. 2) Small insertions and deletions (indels) of a DNA segment. Indels in genes such as *PTEN*, *APC*, *ARID1A*, *CDH1*, and *BRCA1/2* have been shown to promote cancer [70]. 3) Large copy number alterations (CNAs) – deletions or gains of extra copies of DNA segment spanning a gene, multiple genes or a whole chromosome [83]. Copy number homozygous deletions of gene *CDKN2A* are frequent events in the aggressive brain cancer – Glioblastoma [137], and loss of *CDKN2A* is a shared event in Glioblastoma patients

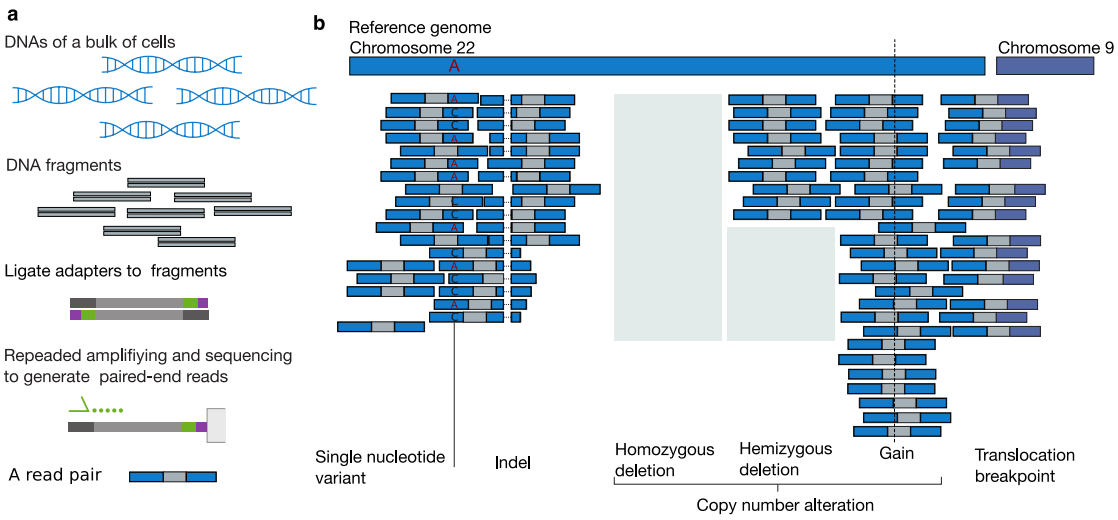


Figure 1.1: DNA sequencing and genetic aberration detection. a) A DNA pool from multiple cells is fragmented. The ends of each double strand DNA fragment are repaired. Then adapters are ligated to both ends of each DNA fragment. These DNA fragments are attached to a flowcell and amplified. Finally to sequence to generate paired-end reads. b) Aligning the paired-end reads to a reference genome to detect various genetic aberrations. Figure modified from Meyerson *et al* [143]

harbouring other various copy number alterations driving the progression of Glioblastoma, and thus represents an early event in primary Glioblastoma [191]. 4) Genomic rearrangements – inversions or translocations of DNA segments. The classic example is the *BCR-ABL* fusion gene that results from the translocation between chromosome 9 and chromosome 22 in a subtype of blood cancer – chronic myeloid leukemia [55].

The genes with cancer promoting mutations ('driver') can be broadly grouped into two categories based on whether inactivation or activation of these genes help drive a cell toward cancer. Oncogenes, when activated by mutations or epigenetic events, promote cancer formation. On the contrary, tumour suppressor genes prevent cancer formation in their active form - anti-oncogenes. Most oncogenes require only one hit or mutation to promote cancer (they tend to be genetically dominant) whereas tumour suppressor genes require biallelic inactivation to promote cancer. (They are genetically recessive.) Oncogenes harbour mostly gain-of-function mutations (e.g., some missense mutations, in-frame indels, or copy number gains), which confer new functions or enhance the activities of the final gene products. Tumour suppressor genes frequently accumulate loss-of-function mutations (e.g., some nonsense mutations, frame-shift indels, splice-site mutations, or copy number deletions), which reduce or abrogate the functions

of the final gene products. Oncogenes and tumour suppressor genes can contribute to the onset of cancer indirectly. For example, defects in genes normally repairing DNA damages (e.g., *BRCA1*) confer cell growth advantages by allowing cells that have chromosomal changes favoring growth to survive and divide [213]. In addition, some genes such as *NOTCH1* can act as either oncogenes or tumour suppressor genes depending on mutation types and cell types.

1.2 Systems biology approach to profile cancer

Genome sequencing could become routine in the future standard of cancer care as the sequencing cost keeps decreasing [219]. However, it is a challenging or impossible task to detect all the 'driver' mutations of each cancer patient based solely on statistical analysis of mutation data from DNA sequencing alone because of the far larger number of 'passenger' mutations that are immaterial to neoplasia [71, 195, 196, 213] co-acquired during tumourigenesis [116]. More importantly, we still need to understand the mechanism of how the driver mutations lead to tumorigenesis and the progression of cancer in each patient, e.g., how a dysfunctional protein resulting from a driver mutation confers the cell growth advantages, and how a cancer cell obtains the 'hallmarks' through the set of driver mutations. This knowledge is essential to understand the mechanism of drug resistance and seek better treatment options [77]. We suggest that a systems biology [28, 88, 102, 159, 224] approach to integrative analysis of various high-throughput data such as mutation and gene expression data could be invaluable to both nominate candidate cancer driver mutations and interpret mutation functions. Below I briefly discuss some high-throughput data used in this dissertation.

Transcriptome data Different types of cells of a single multicellular organism (e.g., a liver cell and a muscle cell) express different set of genes, and the resulting RNA and protein products determine the physical characters and the functions of the cells. Directly measuring all the final protein products of protein-coding genes is possible by techniques such as mass spectrometry but currently is limited by the data quality (reproducibility issue [220]) and the lack of efficient computational annotation tools [226]. In contrast, microarray and more expensive but higher resolution RNA sequencing (RNA-Seq), which directly measure the abundance of each RNA molecule by counting the number of reads mapped to the exon region of each gene, can generate highly reproducible measurements. In addition, although a cell can control protein content after RNA molecules are produced by directly degrading RNA or protein molecules or controlling RNA

translation, transcription regulation is paramount because it can minimize the energy wasted to produce unnecessary molecules [3]. Correspondingly, mRNA gene expression data have been widely used to study biological problems. In cancer research, mRNA gene expression data have been used to stratify morphologically (under the microscope) indistinguishable tumours into clinically relevant subgroups (e.g., the expression of a panel of 50 genes to stratify breast cancer patients into five subgroups).

Single-cell transcriptome data Instead of analyzing a bulk of cells to measure the population average, current high-throughput single-cell techniques have made it possible to efficiently measure the levels of all mRNA molecules in a specific cell [59, 90, 103, 128, 194]. Typically, these techniques include several steps: first cells are separated and the mRNA molecules of each cell are separately captured; then the mRNA molecules of each cell are reverse transcribed to cDNA, which is amplified (e.g., by polymerase chain reaction) to generate enough materials for sequencing [194]. Inaccuracy is introduced by the low capture ratio (the proportion of mRNA molecule captured) and PCR-amplification bias [90] (e.g., different cDNA molecules are amplified differently.) Another factor influencing single-cell mRNA measurements is the cell state at the time of sequencing (a cell can be at rest, in different stages of the cell cycle or in apoptosis).

Single-cell mass cytometry data Mass cytometry (CyTOF) uses antibodies labeled with rare-earth element isotopes (with different masses) to bind to target epitopes of protein antigens (marker proteins) of cells [15, 90]. Cells with bound antibody-isotope conjugates are then nebulized into single cell droplets, and each cell is further vaporized to produce ions of its atomic constituents. The ions are mass filtered to only keep the rare-earth element ions, which are then mass measured by passing through a time-of-flight mass spectrometer (heavier ions take longer to 'fly through' a fixed distance to reach a detector). The abundance of rare-earth elements is thus a measure of the marker protein expression. Mass cytometry is limited by the number of proteins that can be simultaneously measured (around 40 at present) because of the limited number of rare-earth element isotopes. Compared to single-cell gene expression, mass cytometry has higher throughput to measure protein expression of tens of thousands to even millions of cells (currently can process hundreds of cell per second) [15].

Molecular networks A typical human cell expresses thousands of genes, and the protein products interact with other molecules (e.g., O^2, glucose, ATP) or ions (e.g., Na^+) to perform

different reactions necessary for the cell, e.g., to break down foodstuffs or synthesize molecules. An ordered series of reactions (the products of one reaction serve as inputs or substrates of the next) are connected to form a pathway, e.g., the glycolysis pathway takes many steps to convert glucose to pyruvate and release ATP. Three kinds of pathways have been widely studied and documented in databases: metabolic pathways [99, 100, 132] about a cell obtaining energy and synthesizing molecules, gene regulation pathways [96] to turn on a set of genes and shut down another set of genes at the right time and the right place, and signalling transduction pathways [132, 180] to sense intracellular or extracellular signals and respond accordingly. These pathway databases are still far from complete [18, 28, 54], miss tissue context information, and are biased by manual curation [28]. Different pathways are inter-linked to form a network. One of the most widely studied biological networks is the protein-protein interaction (PPI) network where a node in the network represents a protein and a link between two nodes represent physical binding of the two proteins. A protein typically binds to other molecules to participate in biological processes. Therefore, the PPI networks are useful in interpreting protein functions because two bound proteins tend to share common functions [3]. Large scale detecting of PPI networks can be performed by yeast two-hybrid screens [172] (the standard yeast two-hybrid screen system has low sensitivity and specificity and can only detect nucleus interactions) or affinity purification coupled to mass spectrometry [3]. Computational methods have long been used to predict functional PPIs [64], which means two predicted interacted proteins may not physically interact but participate in the same biological process.

1.3 Modelling and prediction

In cancer systems biology research, we frequently reason under uncertainty [176], e.g., to infer whether a signalling pathway is highly active in a patient given noisy measurements of DNA mutations and mRNA expression in that patient. We can summarize the uncertainty in data from measurement noise using probability distributions, e.g., the observed expression of a gene in skin across people is modelled by a Gaussian distribution. The Bayesian approach allows us to further describe the uncertainty in model parameters through prior distributions and even the uncertainty in choosing different kinds of models (e.g., a spherical or an unconstrained two dimensional Gaussian distribution) [75]. A joint distribution of all the random variables specifies the space of possible outcomes and the corresponding probability masses or probability densities

for continuous distributions. After observing the values of a subset of random variables (evidence variables), we can calculate the marginal distributions of some query variables of interest, e.g., after observing a *CTNNB1* gene mutation in a patient, we can update the marginal distribution of Wnt signalling pathway activity in that patient.

For distributions that cannot be specified by a small number of parameters, even for simple binary random variables, directly enumerating and assigning probabilities to the combinations of outcomes take time and space exponential in the number of random variables. Unfortunately the joint distribution of a real problem is typically flexible, we therefore need to explore the structures (conditional independences) in the joint distribution by factoring it into the products of distributions, each over a small subset of random variables. Then we only need to specify the set of local distributions each involving a small subset of random variables to fully determine the joint distribution. The probabilistic graphical model framework [20, 21, 22, 107, 150] provides a pictorial way for modulators (e.g., domain experts) to explore the conditional independent structures in the joint distribution.

One type of probabilistic graphical models, the directed graphical model (Bayesian network) is a directed acyclic graph $\mathcal{G}(V, E)$ whose vertices V correspond to random variables. A vertex $X \in V$ and its parents $\text{Parent}(X)$ (directly connected to X by an edge $\in E$ and can be empty) consist a family, which has a conditional distribution $p(x \mid \text{Parent}(x))$ that quantifies the effects of the parents on X. The joint distribution encoded by the graph is the products of the local conditional distributions:

$$p(\mathbf{x} \mid \mathcal{G}) = \prod_{v=1}^{|V|} p(x_v \mid \text{Parent}(x_v)) \tag{1.1}$$

where $|V|$ means the cardinality of the vertex set V. The vector $\mathbf{x} = (x_1, x_2, \ldots, x_v, \ldots, x_{|V|})^T$ is any possible values of random variables $(X_1, X_2, \ldots, X_v, \ldots, X_{|V|})^T$, where '$T$' means transpose of a vector. We typically specify the structures (topology) of a directed probability graphical model, and learn the parameters (the conditional distributions) given data.

Many classic probabilistic models can be represented as probabilistic graphical models. These classic models could serve as building blocks to develop more sophisticated probabilistic models.

For example, the mixture model used throughput this thesis has the following joint distribution:

$$p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x} \mid \boldsymbol{\theta}_k) \tag{1.2}$$

where $p(\mathbf{x} \mid \boldsymbol{\theta}_k)$ is the distribution of the mixture component k governed by parameter $\boldsymbol{\theta}_k \in \boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\} = \{\boldsymbol{\theta}_k\}_{k=1}^K$, and π_k is the mixture coefficient of component k. We use the vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)^T$ to denote the mixture coefficients, which are nonnegative and sum to one. Here $(\pi_1, \ldots, \pi_k)^T$ means the transpose of the row vector $(\pi_1, \ldots, \pi_k)$. In the literature, a continuous distribution such as $p(\mathbf{x} \mid \boldsymbol{\theta}_k)$ in Equation 1.2 is commonly written as

$$f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x} \mid \boldsymbol{\theta}_k) \tag{1.3}$$

In addition, Equation 1.2 (the notations used in this thesis) has the same meaning as Equation 1.4 with slightly different notations:

$$p(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}; \boldsymbol{\theta}_k) \tag{1.4}$$

The density function in Equation 1.2 is not in the conventional form of the products of conditional distributions as in Equation 1.1. If we introduce a latent variable Z, such that $Z = z$ indicates that the source component of data point $\mathbf{x}$ is z, then the joint distribution is

$$p(\mathbf{x}, z \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) = p(z \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) p(\mathbf{x} \mid z, \boldsymbol{\pi}, \boldsymbol{\Theta}) = p(z \mid \boldsymbol{\pi}) p(\mathbf{x} \mid z, \boldsymbol{\Theta}) = p(z \mid \boldsymbol{\pi}) p(\mathbf{x} \mid \boldsymbol{\theta}_z) \tag{1.5}$$

where $p(z \mid \boldsymbol{\pi})$ is the categorical distribution such that $P(Z = k \mid \boldsymbol{\pi}) = \pi_k$. Then we can get the marginal distribution $p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_z p(\mathbf{x}, z \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{z=1}^K p(z \mid \boldsymbol{\pi}) p(\mathbf{x} \mid \boldsymbol{\theta}_z) = \sum_{k=1}^K \pi_k p(\mathbf{x} \mid \boldsymbol{\theta}_k)$. The probabilistic graphical model in Fig. 1.2 (a) specifies the mixture model with latent variable Z.

We can use a probabilistic graphical model to generate artificial data by sequentially drawing samples of random variables from parents to children, using the conditional distribution at each random variable. For example, for the mixture model in Fig. 1.2(a), given the mixture coefficients $\boldsymbol{\pi}$ and the component parameters $\boldsymbol{\Theta}$, an artificial data point $\mathbf{x}$ is generated in two steps: 1) the component indicator variable z is sampled from the categorical distribution $p(z \mid \boldsymbol{\pi})$, and 2) the actuarial data point $\mathbf{x}$ is generated from the distribution $p(\mathbf{x} \mid \boldsymbol{\theta}_z)$.

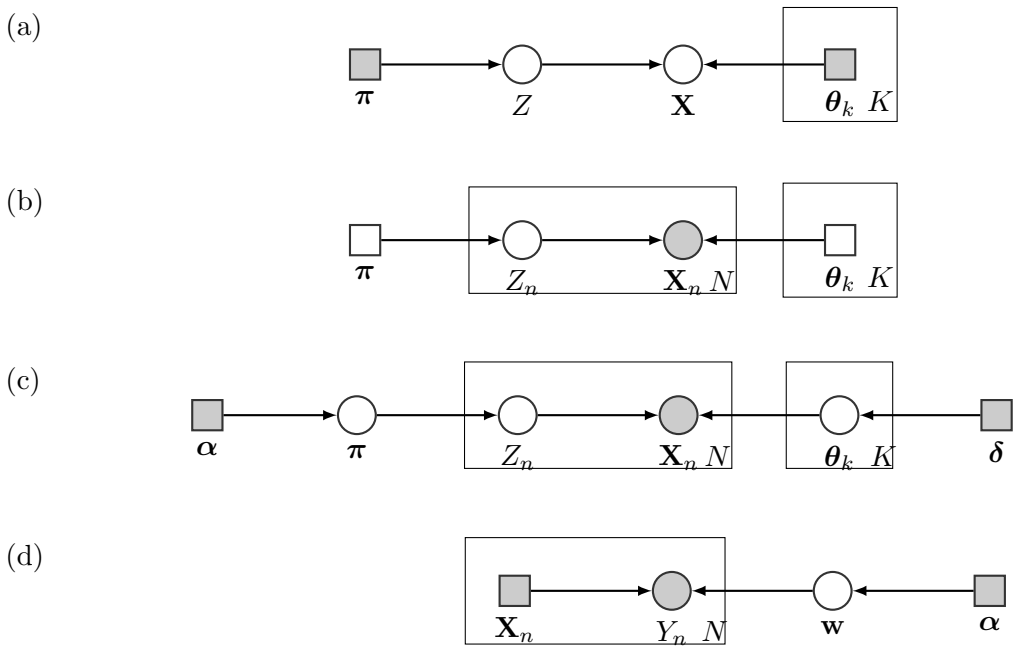


Figure 1.2: Example probabilistic graphical models for both generative and discriminative models. Circles represent random variables. Squares represent deterministic parameters. Shaded nodes are observed, and unshaded nodes are hidden. Here we use the plate notation, i.e., nodes inside each box will get repeated when the node is unpacked (the number of repeats is on the bottom right corner of each box). Each node and its parents constitute a family. Given the parents, a random variable is independent of the ancestors. Therefore, the joint distribution of all the random variables is the products of the family conditional distributions. (a) The structure of a mixture model with known and fixed parameters. (b) A mixture model with K mixture components. We assume that the mixture coefficients $\boldsymbol{\pi}$ and the parameters of each mixture component $\boldsymbol{\theta}_k$ are unknown but fixed parameters. (c) A Bayesian mixture model where both $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are random vectors and follow specific distributions with hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$, respectively. (d) A discriminative regression model with observed training data. Although parameters are also random variables from a Bayesian perspective, they are represented by lower case letters in the graphical models.

In addition to as a knowledge representation system (to encode conditional independence relationships and to specify the joint distribution), graphical models also serve as powerful reasoning engines to infer unknown quantities after observing some variables (evidence variables) [176]. Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} = \{\mathbf{x}_n\}_{n=1}^N$, where a data point is a D dimensional column vector $\mathbf{x}_n = (x_{n,1}, x_{n,2}, \ldots, x_{n,D})^T$ whose component $x_{n,j}$ is a real number, we can add those data points as observed random variables as in the probabilistic graphical model of Fig. 1.2(b). The data point specific latent variables $\mathbf{z} = \{z_1, z_2, \ldots, z_N\} = \{z_i\}_{n=1}^N$. Both $\boldsymbol{\pi}$ and $\boldsymbol{\Theta}$ are unknown but fixed parameters of the mixture model. Then we can write the

joint distribution for the mixture model in Fig. 1.2(b) as

$$p(\mathbf{z}, \mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) = \prod_{n=1}^{N} p(z_n \mid \boldsymbol{\pi}) p(\mathbf{x}_n \mid \boldsymbol{\Theta}, z_n) \tag{1.6}$$

For probabilistic graphical models, we typically separate inference from learning. The inference problem is to compute the marginal distributions of latent variables of interest, and the learning problem is to learn the model parameters. For example, in Fig. 1.2(b), the inference is to compute $p(z_n \mid \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\Theta})$, and the learning problem is to estimate $\hat{\boldsymbol{\pi}}$ and $\hat{\boldsymbol{\Theta}}$, typically by maximizing the likelihood. For the mixture model in Fig. 1.2(b), given the parameters, the inference problem is simple because Z_n is independent of $\{Z_i, \mathbf{X}_i\}_{i \neq n}$. Therefore, we can process each data point independently as follows:

$$\begin{aligned} P(Z_n = k \mid \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\Theta}) &= \frac{P(Z_n = k \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) p(\mathbf{x}_n \mid Z_n = k, \boldsymbol{\pi}, \boldsymbol{\Theta})}{\sum_{c=1}^{K} P(Z_n = c \mid \boldsymbol{\pi}, \boldsymbol{\Theta}) p(\mathbf{x}_n \mid Z_n = c, \boldsymbol{\pi}, \boldsymbol{\Theta})} \\ &= \frac{\pi_k p(\mathbf{x}_n \mid \boldsymbol{\theta}_k)}{\sum_{c=1}^{K} \pi_c p(\mathbf{x}_n \mid \boldsymbol{\theta}_c)} \end{aligned} \tag{1.7}$$

For a general graphical model with many latent variables and observed variables, a simple inference algorithm is by summing out (or integrating out) the latent random variables that are not interested one by one given the factorized joint distribution (the variable elimination algorithm). A more efficient inference algorithm (the belief propagation algorithm) uses dynamic programming to store and reuse intermediate results, and it can compute the marginal distributions of all the latent variables by passing messages (distributions) along the graph twice. For complex models when exact influence (e.g., both the variable elimination algorithm and the belief propagation algorithm) is intractable, we have to use approximate inference algorithms.

For a graphical model without latent variables, maximum likelihood learning of the model parameters is simple. For example, for the mixture model in Fig. 1.2(b) when Z_n is given, $\hat{\pi}_k = \frac{\sum_{n=1}^{N} Z_n = k}{N}$, and $\hat{\boldsymbol{\theta}}_k$ can be estimated from the data points $\{\mathbf{x}_n \mid z_n = k\}$ the same way as estimating the parameters of a single component distribution from data. For a graphical model with latent variables, the learning problem is typically solved iteratively by the Expectation-Maximization (EM) algorithm, which involves first solving the inference problem in each iteration. For example, for the mixture model in Fig. 1.2(b), given the current guess of the model parameters $\boldsymbol{\pi}^t$ and $\boldsymbol{\Theta}^t$, we can first compute the distribution $p(z_n \mid \mathbf{x}_n, \boldsymbol{\pi}^t, \boldsymbol{\Theta}^t)$. Then the mixture coefficient can be estimated as $\hat{\pi}_k^{t+1} = \frac{\sum_{n=1}^{N} P(Z_n = k \mid \mathbf{x}_n, \boldsymbol{\pi}^t, \boldsymbol{\Theta}^t)}{N}$. The parameter $\hat{\boldsymbol{\theta}}_k^{t+1}$ can be

updated similarly for the case without latent variables by considering a 'soft' assignment of data points to clusters. This process is repeated until the likelihood increases very small in each iteration.

We can take a Bayesian perspective to treat parameters as random variables and further specify prior distributions govern by hyperparameters for these random variables. A Bayesian version of the mixture model is in Fig. 1.2(c), which specifies a joint distribution

$$p(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z}, \mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\delta}) = p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \prod_{k=1}^{K} p(\boldsymbol{\theta}_k \mid \boldsymbol{\delta}) \prod_{n=1}^{N} p(z_n \mid \boldsymbol{\pi}) p(\mathbf{x}_n \mid \boldsymbol{\Theta}, z_n) \tag{1.8}$$

In the Bayesian setting, there is no distinctions between inference and learning since the parameters are also random variables. By so doing, we can quantify the uncertainty of the parameters through the posterior distributions. For example, we can compute the posterior distribution

$$p(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z} \mid \boldsymbol{\alpha}, \boldsymbol{\delta}, \mathbf{X}) = \frac{p(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z}, \mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\delta})}{\int_{\boldsymbol{\Theta}} \int_{\boldsymbol{\pi}} \sum_{\mathbf{z}} p(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z}, \mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\delta}) d\boldsymbol{\pi} d\boldsymbol{\Theta}} \tag{1.9}$$

The integral (summation) in Equation 1.9 is typically difficult to compute. Instead of computing the posterior distribution analytically, the Markov chain Monte Carlo (MCMC) method [20, 127, 150] draws samples from the posterior distribution. Specifically, MCMC constructs a Markov chain whose stationary distribution is the target posterior distribution in Equation 1.9. Then we can collect (approximately independent) samples from the Markov chain, these samples are used to construct an empirical estimation of the posterior distribution as these samples can be considered as from the target posterior distribution. However, MCMC tends to be slow. Instead of drawing samples from the target distribution, the variational inference [20, 23, 127, 150] picks a distribution in a family $q(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z})$ that most 'similar' to $p(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z} \mid \boldsymbol{\alpha}, \boldsymbol{\delta}, \mathbf{X})$ by minimizing the Kullback-Leibler divergence

$$q^*(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z}) = \arg \min_{q(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z})} \mathbb{KL}(q(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z}) || p(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z} \mid \boldsymbol{\alpha}, \boldsymbol{\delta}, \mathbf{X}) - \log(p(\mathbf{X})) \tag{1.10}$$

$$= \arg \min_{q(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z})} \iint_{\boldsymbol{\pi}, \boldsymbol{\Theta}} \sum_{\mathbf{z}} q(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z}) \log \frac{q(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z})}{p(\boldsymbol{\pi}, \boldsymbol{\Theta}, \mathbf{z} \mid \boldsymbol{\alpha}, \boldsymbol{\delta}, \mathbf{X})} d\boldsymbol{\pi} d\boldsymbol{\theta} - \log(p(\mathbf{X})) \tag{1.11}$$

Compared to MCMC, variational inference is relatively new and how the algorithm behaves is still not well understood [23]. Instead of computing the posterior distribution, a computationally efficient alternative is to do maximum a posteriori (MAP) estimation to find a mode of the pos-

terior distribution. In the EM framework, the expectation step is the same as that in maximum likelihood estimation of parameters. The prior distribution only influence the maximization step in updating model parameters. Other optimization algorithms other than EM can also be used to find a local maximum of the posterior distribution.

Sometimes, we are given training data with ground truth information, i.e., a set of input-output pairs $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$. Our task is to learn a mapping $f(\mathbf{x})$ from $\mathbf{x}$ to y. From Bayes theorem we can get $p(y \mid \mathbf{x}) = \frac{p(\mathbf{x},y)}{p(\mathbf{x})} = \frac{p(y)p(\mathbf{x}|y)}{\sum_y p(y)p(\mathbf{x}|y)}$ for classification problems where y is a categorical variable. This is called a generative model since we implicitly model the joint distribution $p(\mathbf{x}, y)$. To solve these problems, we can use discriminative models, i.e., directly modelling the conditional distribution of hidden variables given observed variables $p(y \mid \mathbf{x})$, or simply finding a discriminative function $f(\mathbf{x})$ (so called algorithmic modelling by Leo Breiman [27], including popular algorithms such as support vector machines, artificial neural networks [119], and random forests). Because the joint distribution $p(\mathbf{x} \mid y)$ is typically a multivariante distribution and is difficult to model, discriminative approaches bypass the difficulties of modelling $p(\mathbf{x} \mid y)$ and solve simpler problems of either modelling $p(y \mid \mathbf{x})$ or adopting the algorithmic modelling to find a discriminative function $f(\mathbf{x})$. Compared to generative models, labelled ground-truth data are typically required to train discriminative models. For example, Fig. 1.2(d) shows a regression model. Let $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$ and $\mathbf{w} = (w_0, w_1, \ldots, w_D)^T$, where D is the dimensionality of each data point $\mathbf{x}_n$, the joint distribution for the statistical model in Fig. 1.2(d) is

$$p(\mathbf{y}, \mathbf{w} \mid \boldsymbol{\alpha}, \mathbf{X}) = \prod_{n=1}^{N} p(y_n \mid \mathbf{x_n}, \mathbf{w}) p(\mathbf{w} \mid \boldsymbol{\alpha}) \tag{1.12}$$

For ease of presenting the results, we add a constant one to each data point $\mathbf{x}_n$, e.g., $\mathbf{x}_n = (1, x_{n,1}, x_{n,2}, \ldots, x_{n,D})^T$. For binary logistic regression (Chapter 2) where $y \in \{0, 1\}$, $p(y_n \mid \mathbf{x_n}, \mathbf{w}) = \text{Ber}(y_n \mid \text{sigm}(\mathbf{w}^T\mathbf{x}_n))$, which is a Bernoulli distribution with parameter $\text{sigm}(\mathbf{w}^T\mathbf{x}_n) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x}_n)}$. For linear regression where $y \in \mathbb{R}$, $p(y_n \mid \mathbf{x_n}, \mathbf{w}) = \mathcal{N}(y_n \mid \mathbf{w}^T\mathbf{x}_n, \sigma^2)$, which is a univariate normal distribution with mean $\mathbf{w}^T\mathbf{x}_n$ and variance σ^2.

1.4 Challenges in detecting and interpreting mutations from genomic sequencing data

Although in principle we can sequence the whole genome of a cancer call to detect all the genetic aberrations, in practice we only observe part of these aberrations because of technical issues and experimental design limitations in current high-throughput sequencing [213] (Chapter 2). First, typical sequencing platforms such as Illumina (Solexa), ABI SOLiD, Ion Torrent, SMRT of Pacific Bioscience and Roche 454 have platform-specific bias [170, 173], e.g., SMRT platform yields long single molecular reads that are subjected to insertions and deletions errors [170]. Sequence errors can originate from sample preparation, e.g., low tumour cell content or formalin fixation that causes nucleotide mutations in samples for sequencing. Each step in sequencing library preparation can introduce bias [170, 206]. Subclonal mutations (mutations only in some of the cancer cells) and mutations in copy number alterations regions can have very low variant allele frequencies (of the reads covering a mutation, the fraction of these reads with the variant allele) and make it difficult to detect them. Short reads from repetitive regions of the genome are difficult to unambiguously map to the reference genome [46]. GC-rich regions such as the first exon of many genes can be difficult to fragment (Fig. 1.1(a)) or amplify by polymerase chain reaction (PCR) and are thus underrepresented in the sequence reads [17, 173]. Inadequate sequencing depth (or coverage, the average number of times a site is sequenced) can be a major factor in missing variants. Griffith *et al* show that current strategies for whole genome sequencing studies missed many somatic mutations (mutations in the somatic cells but not the germ line of a patient) [80]. By increasing the sequencing depths from 30x in their original study [53] to 300x and using a consensus of somatic single nucleotide variant (SNV) callers, the number of identified SNVs increased from 118 to 1343. Moreover, an additional 2500 SNVs were highly likely to be genuine somatic SNVs but still without enough evidence even at 300x coverage.

Current cancer genome sequencing studies detected dozens to hundreds of mutations for a typical cancer genome, and even tens of thousands of mutations for the genomes with DNA-repair deficiency [80, 213]. To detect and interpret the functions of the small number of 'driver' mutations is thus the focus of current cancer genomic sequencing studies [14, 29, 44, 117, 130, 142, 164, 168, 203, 207, 208, 235] and of targeted therapies informed by patient genomic information. If the driver mutation directly produces an overactive, or neomorphic (result in a dominant gain of gene function that is different from the normal function) mutant protein or

excessive amount of protein products, drugs can possibly be designed to inhibit the activities of the mutant protein, e.g., *BRAF* inhibitors to treat melanoma patients with activating *BRAF* mutations [187]. For a driver mutation that results in non-functional protein products, the inactivation of this protein is expected to activate some growth-related proteins downstream of a signal pathway [213]. The cancer cells are thus dependent on these proteins to survive and therefore pathway information and synthetic lethality (mutations in two or more genes combined cause cell death, but a single mutated gene does not cause cell death) can be harnessed to selectively kill the cancer cells [138]. For example, ovary and breast cancer cells losing *BRCA1* or *BRCA2* depend on *PARP1* to repair single-strand DNA breaks, and inhibiting *PARP* causes accumulated double-strand breaks in cancer cells, and leads to cancer cell death. To overcome the almost inevitable developed resistance to single drug based treatments, multiple drugs can be applied simultaneously to block several cancer signaling pathways to increase the possibilities of eliminating all the cancer cells [25]. However, to successfully apply these therapies, we need to separate driver mutations from passenger mutations in individual patients (Chapter 3).

Although mutations in hundreds of genes are implicated in promoting cancer, these genes are components of a much smaller number of signal pathways through which cancer cells confer growth advantages [213]. Thus, it is common to observe similar messenger RNA (mRNA) expression profiles for two genetically different cancers as they share the same set of disrupted signaling pathways (and show similar clinical outcomes). The large numbers of passenger mutations do not contribute to cancer development, but some of them result in aberrant mutant proteins that are foreign to the patient immune system. Therefore, the mutation data could act as markers to stratify patients for immunotherapy [118, 169, 181, 186, 192]. Recent technology advances, especially single cell technologies, enable us to profile cancer at single cell resolutions, e.g., mutations, copy number alterations, mRNA expression measurements from single cell sequencing data, and protein expression from mass cytometry data. These data could revolutionize our understanding of tumour heterogeneity and tumour microenvironment, and possibly help patient stratification (Chapter 4).

1.5 Research contributions

Below, I summarize the research contributions of my work (Fig. 1.3). These points are expanded in subsequent chapters.

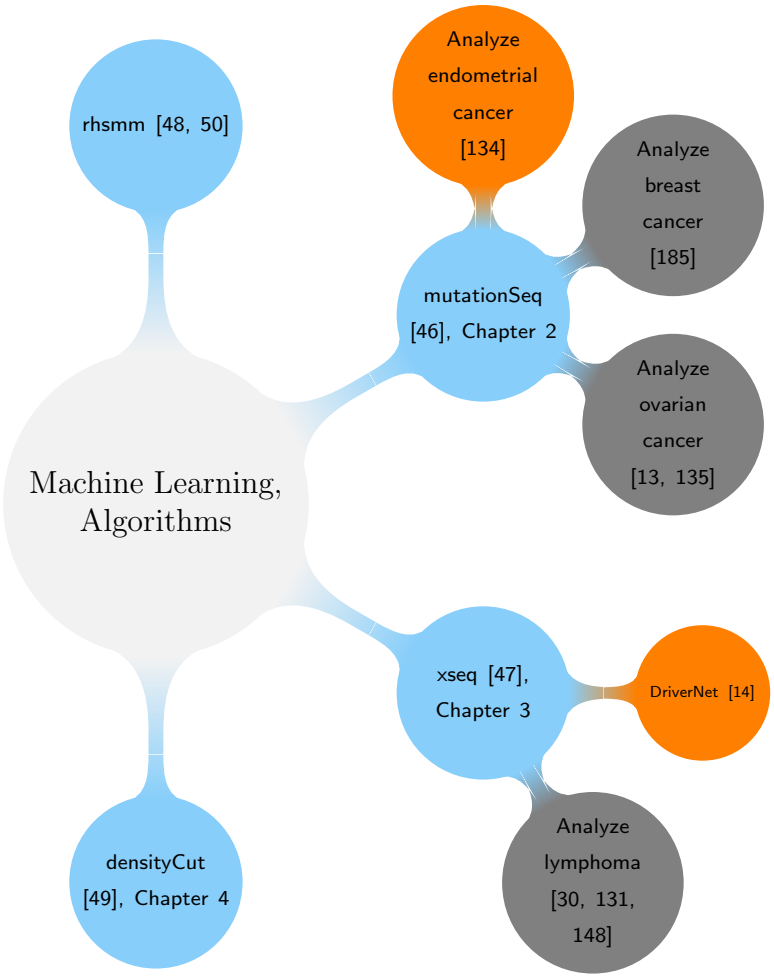


Figure 1.3: Overview of the work done during my PhD study. First author publications are in blue [46, 47, 48, 49, 50], co-first author publications in orange [14, 134], and other co-authored papers in grey [13, 135, 185].

1.5.1 Predicting somatic single nucleotide variants

Advances in high-throughput sequencing technology allow for rapid sequencing of tumour-normal pairs of DNA, where both tumour DNA and adjacent healthy tissue (or blood) DNA from the same individual are sequenced. These tumour-normal matched pairs allow for comparing billions of DNA bases from a whole genome to identify somatic mutations. `mutationSeq` (`http://compbio.bccrc.ca/software/mutationseq/`) is among the first available tools specifically designed for the prediction of somatic single nucleotide variants – by integrating both normal and tumour sequencing data. `mutationSeq` is a feature-based supervised discriminative classifier approach that takes advantage of valuable validated somatic mutations to predict novel single

nucleotide variants. We further identified and classified the technical noise in high-throughput DNA sequencing data. `mutationSeq` has been used to profile 70 triple-negative breast cancers [185], high-grade serous ovarian cancer [13], endometrial cancer [134, 135] and many others. Since the publication of `mutationSeq`, large numbers of tools have been developed and are under development for discovering (calling) somatic single nucleotide variants from paired normal-tumour sequencing data. `mutationSeq` remains one of the best methods for single nucleotide variant calling. `mutationSeq` has undergone three years of active development and has been extended in several ways, and has been used in our lab and Canada's Michael Smith Genome Science Centre to profile thousands of cancer genomes and exomes.

1.5.2 Predicting somatic mutations that correlate with gene dysregulation

We present a novel hierarchical Bayes statistical model, `xseq` (`http://compbio.bccrc.ca/software/xseq/`), to systematically quantify the impact of somatic mutations on expression profiles. We establish the theoretical framework and robust inference characteristics of the method using computational benchmarking. We then apply `xseq` for the analyses of thousands of tumour datasets available through The Cancer Genome Atlas, to systematically quantify somatic mutations impacting expression profiles. We identify 30 novel cis-effect tumour suppressor gene candidates, enriched in loss-of-function mutations and bi-allelic inactivation. Analysis of trans-effects of mutations and copy number alterations with `xseq` identifies mutations in 150 genes impacting expression networks, with 89 novel predictions. We reveal two important novel characteristics of mutation impact on expression: (1) patients harbouring known driver mutations exhibit different downstream gene expression consequences; (2) expression patterns for some mutations are stable across tumour types. These results have critical implications for identification and interpretation of mutations with consequent impact on transcription in cancer.

1.5.3 Clustering high-dimensional, high-throughput biological datasets

As measurement technology advances have drastically enhanced our ability to generate various high-throughput datasets, there is a great need to develop efficient and robust clustering algorithms to analyze large N (the number of data points), large D (the dimentionality of data points) datasets, with the ability to detect arbitrary shape clusters and automatically determine the number of clusters. Therefore, in this chapter, we developed and implemented a novel clustering algorithm called `densityCut` (`https://bitbucket.org/jerry00/densitycut_dev/`

`overview`) to cluster high-dimensional, high-throughput biological datasets. Experimental results on synthetic benchmark datasets demonstrate the robustness of `densityCut` in clustering datasets consisting of complex shape clusters. We apply `densityCut` to three real-world applications, namely to cluster variant allele frequencies of somatic mutations to reveal clonal architectures of individual tumours, to cluster single-cell gene expression data to uncover cell population compositions, and to cluster single-cell mass cytometry data to detect communities of cells of the same functional states or types.

Chapter 2

Predicting somatic single nucleotide variants from paired normal-tumour sequencing data

"It's difficult to make predictions, especially about the future"

– Niels Bohr

2.1 Introduction

The genome-wide search for functionally important somatic mutations in cancer by emergent, cost-effective high throughput sequencing (HTS) technology has revolutionized our understanding of tumour biology. The discovery of diagnostic mutations [183], new cancer genes (ARID1A [225], PBRM1 [209], PPP2R1A [133], IDH1 [234], EZH2 [146]), insights into tumour evolution and progression [51, 184] and definitions of mutational landscapes in tumor types (CLL [163], myeloma [32], lymphoma [147]) amongst many others, provide important examples of the power and potential of HTS in furthering our knowledge of cancer biology.

Using HTS to interrogate cancers for somatic mutations usually involves sequencing tumour DNA and DNA derived from non-malignant (or normal) tissue (often blood) from the same patient. Consequently, cancer-focused HTS experiments differ considerably in experimental design from the study of Mendelian disorders or normal human variation. In cancer studies, sequence reads from the two matched samples are aligned to a reference human genome, and lists of predicted variants using single nucleotide variant (SNV) predictors (callers) (e.g., Samtools [123], SOAPsnp [124], VarScan [105], SNVMix [78], GATK [136], VipR [5]) are compared in the tumour and normal data. Using naive approaches, those variants appearing in the tumour, but not the normal sample would be considered putative somatic mutations and provide the investigator with a list of candidates to follow up for functional impact and clinical relevance. Unfortunately, such naive approaches often result in false predictions and we suggest herein that

the problem of computational identification of somatic mutations from HTS data derived from tumour and matched normal DNA remains an unsolved challenge. As a result, labour-intensive and often costly validation experiments are still required to confirm the presence of predicted somatic mutations for both research purposes and clinical interpretation.

Although some false predictions may be due to under-sampled alleles, most can be attributed to detectable artefacts that we argue can be leveraged in principled inference techniques to improve computational predictions. Many different approaches to the problem of SNV discovery from HTS have been implemented. Model-based methods such as SNVMix and SOAPsnp aim to probabilistically model the allelic distributions present in the data and infer the most likely genotype from allelic counts. These methods avoid imposing ad-hoc depth-based thresholds on allelic distributions, but their accompanying software packages do not explicitly handle known sources of technical artefacts and they must rely on pre- or post-processing of the data to produce reliable predictions. Examples of features that can indicate artefacts include strand bias whereby all variant reads are sequenced in the same orientation [32], mapping quality - how well each read aligns to its stated position, base quality - the signal to noise ratio of the base call and average distance of mismatched bases to the end of the read, amongst many others (see Methods). Many of these features are readily available from aligned data in packages such as GATK and Samtools and it is generally accepted that applying filters on these quality metrics is necessary to remove false signals. Some software tools such as VarScan, Samtools, and GATK aim to leverage these features in their SNV prediction routines; however, they are often guided by heuristics, whereby somewhat arbitrary decision boundaries are implemented.

We propose that training feature-based classifiers using robust classification methods from the machine learning literature will better optimize the contribution of each feature to the discrimination of true and false positive somatic mutation predictions. Fitting such classifiers to large sets of ground truth data should allow us to discriminate classes of false positives that may be predicted for different reasons, enabling a more thorough understanding of HTS machine, alignment and biology related artefacts that are informed by data. We suggest that features that best identify somatic mutations will differ in importance in the normal data compared to the tumour data, and so integrated analysis of the tumour and normal data will yield better results than independent treatment of the two datasets. To our knowledge, this notion is currently not considered in any published somatic mutation detection method. Finally, flexible feature-based classifiers that can use any number of features can combine features from different software

packages and therefore leverage newly discovered discriminative features to continually improve somatic mutation prediction accuracy as the bioinformatics literature and methodology matures.

In this chapter, we study the use of discriminative, feature-based classifiers and investigate computational features from aligned tumour *and* normal data that can best separate somatic SNVs from non-somatic SNVs. We implemented four standard machine learning algorithms: random forest, Bayesian additive regression tree, support vector machine, and logistic regression, and compared their performance to each other and to standard methods for somatic mutation prediction. We trained the classifiers on a set of 106 features computed from tumour and normal data on a set of ~ 3400 ground truth positions from 48 primary breast cancer genomes sequenced with exome capture technology, while simultaneously estimating the importance of features. Classifiers were evaluated in a cross validation scheme using robust quantitative accuracy measurements of sensitivity and specificity on labeled training data, and on independent held-out test data derived from four additional cases sequenced with a different technology. We show that principled, feature-based classifiers significantly improve somatic mutation prediction in both sensitivity and specificity over standard approaches such as Samtools and GATK. Finally, using discriminative features, we show how false positive (wildtype) positions can be segregated into several distinct types of systematic artefacts that contribute to false positive predictions.

2.2 Methods

Fig. 2.1 shows the workflow of the feature-based classifier for somatic mutation prediction. We used supervised machine learning methods fit to validated, ground truth training data originally predicted using naive methods (see below for details). Using deep sequencing to validate predictions, we define positions as *somatic* mutations where the variant was found in the tumour but not the normal, *germline* variants where the variant was found in the tumour and the normal, or *wildtypes* (no variants found in either the tumour or the normal, i.e., false positive predictions). The germline and wildtype positions are classed as non-somatic positions so that binary classifiers can be used. Features are constructed for each SNV in the training data using the exome capture `bamfiles` from the tumour and normal alignments. As explained below, we use features available in Samtools, GATK and a set of features we have defined ourselves. These features along with their somatic/non-somatic labels are the inputs to train classifiers. Given

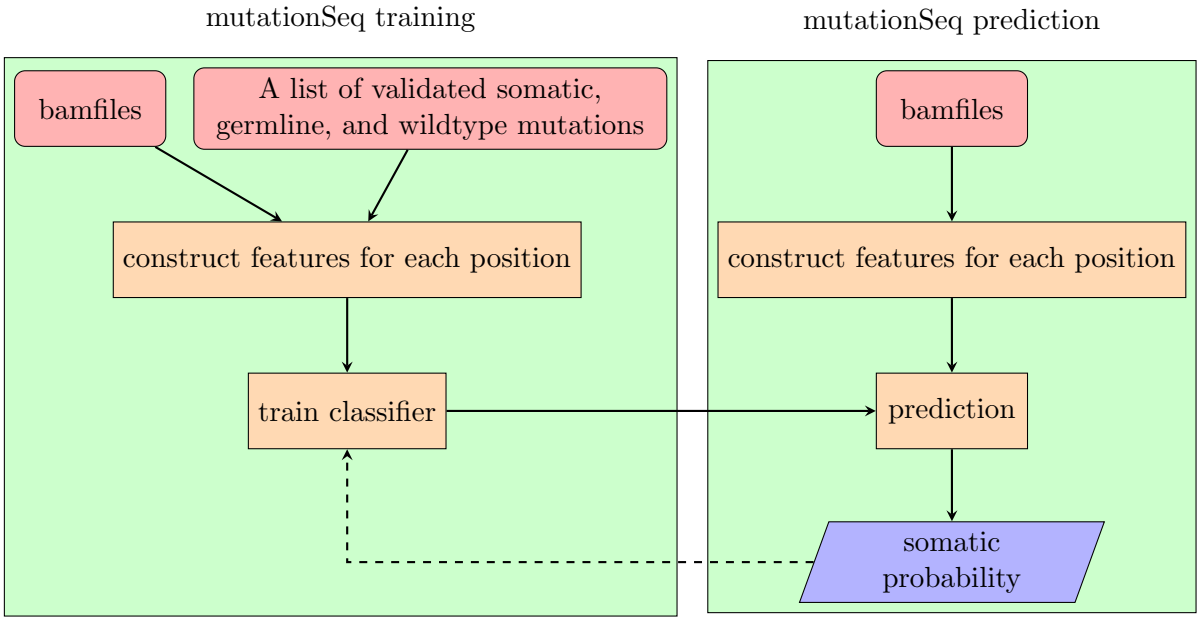


Figure 2.1: The workflow of the feature-based classifier for somatic mutation prediction from DNA sequencing data. Given test bamfiles, each candidate site is represented by a feature vector, and the classifier trained on validated ground truth data is applied to make a prediction. The classifier outputs the probability of each site being somatic.

test bamfiles, we construct features for each candidate site, and apply the trained classifier to predict the somatic mutation probability for each site.

2.2.1 Feature construction

We formalize the somatic mutation prediction problem as a classification problem. Each candidate mutation site of the genome is represented by a feature vector $\mathbf{x}$ with 106 feature components $\{x_1, \ldots, x_{106}\}$. The problem is to predict the label y of the feature represented site. y is defined as "1" if the site is a somatic mutation, and "0" otherwise. Below we first define each component of the feature vector in detail, and then compare different models to predict y given $\mathbf{x}$.

Features x_1 to x_{20} are constructed from the normal data and their definitions are given in Table 2.1. This table is based on the table in `http://samtools.sourceforge.net/mpileup.shtml`. Features x_{21} to x_{40} have the same definitions but are constructed from the tumour. Features x_{41} to x_{60} are constructed from the normal data and their definitions are given in Table 2.2. These features are constructed based on GATK. Features x_{61} to x_{80} have the same definitions but are constructed from the tumour. We show in Section 2.3.3 how simultaneous treatment of the tumour and normal data allow the classifiers to differentially weight corresponding features

Table 2.1: The definitions of features x_1 to x_{20}. Q13 means base quality bigger or equal to Phred score 13; D represents the three-dimensional vector (depth, number of reference bases and number of non-reference bases) at the current site; $G_i \in \{aa, ab, bb\}$ means the genotype at site i, where $a, b \in \{A, C, T, G\}$ and a is the reference allele and b is the non-reference allele. These features are constructed from Samtools.

1. number of reads covering or bridging the site	12. sum of non-reference mapping qualities
2. number of reference Q13 bases on the forward strand	13. sum of squares of non-reference mapping qualities
3. number of reference Q13 bases on the reverse strand	14. sum of tail distances for reference bases
4. number of non-reference Q13 bases on the forward strand	15. sum of squares of tail distance for reference bases
5. number of non-reference Q13 bases on the reverse strand	16. sum of tail distances for non-reference bases
6. sum of reference base qualities	17. sum of squares of tail distance for non-reference bases
7. sum of squares of reference base qualities	18. $P(D \mid G_i = aa)$, phred-scaled, i.e., x is transformed to $-10\log(x)$
8. sum of non-reference base qualities	19. $\max_{G_i \neq aa}(P(D \mid G_i))$, phred-scaled
9. sum of squares of non-reference base qualities	20. $\sum_{G_i \neq aa}(P(D \mid G_i))$, phred-scaled
10. sum of reference mapping qualities	
11. sum of squares of reference mapping qualities	

Table 2.2: The definitions of features x_{41} to x_{60}. These features are constructed from GATK.

41. QUAL: phred-scaled probability of the call given data	51. QD: variant confidence/unfiltered depth
42. allele count for non-ref allele in genotypes	52. SB: strand bias (the variation being seen on only the forward or only the reverse strand)
43. AF: allele frequency for each non-ref allele	53. sumGLbyD
44. total number of alleles in called genotypes	54. allelic depths for the ref-allele
45. total (unfiltered) depth over all samples	55. allelic depths for the non-ref allele
46. fraction of reads containing spanning deletions	56. DP: read depth (only filtered reads used for calling)
47. HRun: largest contiguous homopolymer run of variant allele in either direction	57. GQ: genotype quality computed based on the genotype likelihood
48. HaplotypeScore: estimate the probability that the reads at this locus are coming from no more than 2 local haplotypes	58. $P(D \mid G_i = aa)$, phred-scaled
49. MQ: root mean square mapping quality	59. $P(D \mid G_i = ab)$, phred-scaled
50. MQ0: total number of reads with mapping quality zero	60. $P(D \mid G_i = bb)$, phred-scaled

so as to emphasize tumour-specific and normal-specific features that best discriminate between real and false predictions.

To account for variance in depth across the data, features that scale with depth (e.g., feature x_2 to x_{17}) are first normalized by dividing by the depth. In addition to Samtools and GATK we added several features that we noticed may contribute to systematic errors. For example,

Table 2.3: The definitions of x_{98} to x_{106}. These features are used to boost weak mutation signals in the tumour and decrease the influence of germline polymorphism. In this table, F_i means the normalized version of the ith feature.

98. Forward strand non-reference base ratio F_{24}/F_4	103. Sum of squares of non-reference mapping quality ratio F_{33}/F_{13}
99. Reverse strand non-reference base ratio F_{25}/F_5	104. Sum of non-reference tail distance ratio F_{36}/F_{16}
100. Sum of non-reference base quality ratio F_{28}/F_8	105. Sum of squares of non-reference tail distance ratio F_{37}/F_{17}
101. Sum of squares of non-reference base quality ratio F_{29}/F_9	106. Non-reference allele depth ratio F_{75}/F_{55}
102. Sum of non-reference mapping quality ratio F_{32}/F_{12}	

in [139], the authors found that GGT sequences are often erroneously sequenced as GGG. To capture this artefact, we computed the difference between the sum of the base qualities of the current site and the next site, the sum of the square of the base qualities of the current site and the next site, for both normal and tumour. These features are defined as features x_{81-84}. In addition, the reference base, the alternative base of the normal as well the alternative base of the tumour are included as features x_{85-95} (by dummy representation of categorical variables). In addition, to combine strand bias effects from the tumour and normal data, we define feature x_{96} and feature x_{97} to estimate the strand bias from the pooled normal and tumour data.

To boost weak signals such as rare somatic mutations that may be undersampled or represent a mutation occurring in a small proportion of cells in the tumour, and to decrease the influence of germline polymorphism, another nine features are introduced. The definitions of these features are given in Table 2.3. Note in the table, F_i means the normalized version of the ith feature (dividing by the depth feature). All features are standardized to have zero mean and unit variance prior to training and testing.

2.2.2 Models

After constructing the feature value vector $\mathbf{x}$ for each candidate somatic position, the problem is to find a discriminative function $f(\mathbf{x})$ which optimally separates the true somatic positions from false somatic positions. In so doing, we wished to simultaneously learn the features that best discriminate the two classes. Numerous tools have been developed to solve this problem in the statistics and machine learning community. Here we compare four methods: logistic regression (Logit), support vector machines (SVM), random forests (RF) [89], and Bayesian additive regression tree (BART) [35]. These methods (described below) differ in their underlying

methodology and generally represent broad classes of classifiers present in the machine learning literature. We set out to compare performance of the different approaches in the specific context of predicting somatic mutations from HTS data.

L1 regularized logistic regression

Binary logistic regression is a generalization of the linear regression for classification problems and models the conditional probability of $y \in \{0, 1\}$ given the feature vector $\mathbf{x}$, representing a candidate mutation site, as

$$p(y \mid \mathbf{w}, \mathbf{x}) = \mathrm{Ber}(y \mid \mathrm{sigm}(\mathbf{w}^T\mathbf{x} + w_0)) \tag{2.1}$$

where $\mathbf{w} = (w_1, \ldots, w_D)$ is the weight for each feature, $D = 106$ is the dimensionality of each feature vector and the function $\mathrm{sigm}(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$ is the logistic function.

Here we put a prior $p(\mathbf{w})$ on the weight of the logit model, and do maximum a posterior estimation (MAP) of the parameter $\mathbf{w}$. We focus on the factorized Laplace prior

$$p(\mathbf{w}) = \prod_{j=1}^{D} p(w_j|\rho) = \prod_{j=1}^{D} \frac{1}{2\rho} \exp(-\frac{|w_j|}{\rho}) \tag{2.2}$$

Given N independent and identically distributed (*i.i.d.*) validated mutation pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^N\}$, the posterior of $\mathbf{w}$ is:

$$\begin{aligned} p(\mathbf{w} \mid \mathcal{D}) &\propto p(\mathbf{w})p(\mathcal{D} \mid \mathbf{w}) \\ &= p(\mathbf{w}) \prod_{n=1}^{N} \mathrm{sigm}(\mathbf{w}^T\mathbf{x}_n + w_0) \qquad (2.3) \\ -\log(p(\mathbf{w} \mid \mathcal{D})) &\propto D\log(2\rho) + \sum_{j=1}^{D} \frac{|w_j|}{\rho} \\ &\quad - \sum_{n=1}^{N} \big(y_n \log(p(y_n|\mathbf{w}, \mathbf{x}_n)) + (1 - y_n)\log(1 - p(y_n|\mathbf{w}, \mathbf{x}_n))\big) \qquad (2.4) \end{aligned}$$

The Laplace prior introduces a penalty term $\frac{1}{\rho}\sum_{j=1}^{D}|w_j|$ to the negative log-likelihood function. Because the L_1 norm is defined as $\|\mathbf{w}\|_1 := \sum_{j=1}^{D}|w_j|$, the above logistic regression model with Laplace prior on the weights is called L_1 regularized (penalized) logistic regression model. The L_1 regularized logistic regression model has the property of shrinking the weights of irrele-

vant features to zero.

Support vector machine

The support vector machine (SVM) classifier is a generalization of logistic regression by basis expansion and finds a linear discriminative function of the form

$$f(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\Phi}(\mathbf{x}) + w_0 \tag{2.5}$$

where $\boldsymbol{\Phi}$ is a basis function which maps $\mathbf{x}$ from a 106 dimension space to a new feature space. Although $f(\mathbf{x})$ is a linear function of $\boldsymbol{\Phi}(\mathbf{x})$, it may be a nonlinear function of $\mathbf{x}$. When $\boldsymbol{\Phi}(\mathbf{x}) = \mathbf{x}$, the discriminative function is a hyperplane in the original space spanned by the set of training vectors $\mathbf{x}$, and the corresponding classifier is called a linear SVM.

The SVM assumes that the optimal discriminative function is the one which leaves the largest possible margin on both sides of the feature space (for binary classification; The feature space is the original 106 dimensional space for a linear SVM). When the data points are not linear separable in the feature space, the points that are misclassified or inside the margin are penalized. A parameter C is used to control the trade-off between the margin width and the number of points that are misclassified or inside the margin (A large C tends to produce a narrow margin).

Random forests

Random forests (RF) learns the set of basis functions $\boldsymbol{\Phi}(\cdot)$ from data and is a tree-based method for classification and regression analyses. Given a training dataset consisting N input-output pairs $\{(\mathbf{x}_n, y_n)_{n=1}^N\}$, where $\mathbf{x}_n = (x_{n,1}, \ldots, x_{n,j}, \ldots, x_{n,D})^T$ is a feature vector and $y_n \in \{0, 1\}$ is the corresponding output label, a classification tree g is grown by repeatedly binarily splitting the feature space into disjoint hyper-rectangle regions $\{U_m\}_{m=1}^M$. The splitting rules are typically based on a single component $x_{,j}$ of the whole training feature vectors and are of the form $\{x_{,j} \leq s\}$ vs $\{x_{,j} > s\}$, where s is a real number determined in training. After the classification tree is grown, a leaf node m represents a region U_m with N_m observations from the training data, and the points in this region are assigned a label based on the majority vote:

$$k_m = \arg\max_k \frac{1}{N_m} \sum_{\mathbf{x}_n \in U_m} \mathbb{I}(y_n = k) \tag{2.6}$$

where $k \in \{0, 1\}$ in our case, and $\mathbb{I}(\cdot)$ is the indicator function. Here a leaf node m defines a basis function $\Phi_m(\cdot)$. Let $\mathbf{x} = (x_1, \ldots, x_D)^T$ is a D-dimensional feature vector,

$$\Phi_m(\mathbf{x} \mid \boldsymbol{\theta}_m) = \begin{cases} 1, & \text{if } \mathbf{x} \in U_m. \\ 0, & otherwise \end{cases} \tag{2.7}$$

where $\boldsymbol{\theta}_m$ is the parameter vector for the leaf node m of the tree (an ordered list of splitting rules on the path from the root to the leaf node). The discriminative function can be written in terms of the basis functions $f(\mathbf{x}) = \sum_{m=1}^{M} k_m \Phi_m(\mathbf{x} \mid \boldsymbol{\theta}_m)$. RF makes a prediction based on the outputs of an ensemble of B trees:

$$p(y = k \mid \mathbf{x}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_b, \ldots, \boldsymbol{\Theta}_B) = \frac{\sum_{b=1}^{B} \mathbb{I}(g_b = k)}{B} \tag{2.8}$$

where $\boldsymbol{\Theta}_b$ is the set of parameters of tree b, and $g_b = k$ means that tree b's prediction is k. Specifically, B bootstrap samples (random sampling with replacement, each sample includes N data points) are drawn from the training data, and a tree is grown on each bootstrap sample. To reduce the dependence of the B trees, only p features out of all the features (106 for our case) are selected as candidates for splitting at each intermediate node to grow a tree.

Bayesian additive regression tree

Generally speaking, the Bayesian additive regression tree (BART) can be considered as a Bayesian version of the random forests model and is based on regression trees. As for the classification trees used for random forests classification, a regression tree is grown by repeated binarily splitting the feature space into disjoint hyper-rectangle regions $\{U_m\}_{m=1}^{M}$. The discriminative function of a regression tree with M leaf nodes is

$$f(\mathbf{x}) = g(\mathbf{x} \mid T, \boldsymbol{\Theta}) + \epsilon = \sum_{m=1}^{M} \mu_m \mathbb{I}(\mathbf{x} \in U_m) + \epsilon \tag{2.9}$$

where $\epsilon \sim \mathcal{N}(\epsilon \mid 0, \sigma^2)$, T represents the structure of the binary regression tree, $\boldsymbol{\Theta}$ represents the set of M parameter vectors, each for a leaf node (an ordered list of splitting rules on the path from the root to the leaf node), and μ_m is the mean response in region U_m (i.e., $\mu_m = \frac{1}{N_m} \sum_{\mathbf{x}_n \in U_m} y_n$).

For binary classification, e.g., to classify a site $\mathbf{x}$ as somatic ($y = 1$) or non-somatic ($y = 0$),

the class probability is defined as:

$$p(y \mid \mathbf{x}, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_B) = \Phi(\sum_{b=1}^{B} g_b) \tag{2.10}$$

where $\Phi(.)$ is the standard Gaussian cumulative distribution function.

The BART model differentiates from RF because BART is a fully Bayesian model. All the parameters (tree structures T and leaf parameters $\boldsymbol{\Theta}$) are given hyper-priors and Markov chain Monte Carlo (MCMC) sampling is used for inference.

2.2.3 Datasets

We used two independent datasets to train and test the performance of the models for somatic mutation prediction. The first dataset (*exome capture data*) consists of 48 triple negative breast cancer Agilent SureSelect v1 exome capture tumour/normal pairs sequenced using the Illumina genome analyzer as 76bp pair-end reads. These data were generated as part of a large-scale sequencing project [185] whereby 3369 variants were predicted using only allelic counts and liberal thresholds. Follow-up re-sequencing experiments achieving ∼6000x coverage for the targeted positions revalidated 1015 somatic mutations, 471 germline and 1883 wildtype (false positive) positions. The exome capture data are subdivided into two groups consisting of non-overlapping positions:

- SeqVal1 (somatic: 775 germline: 101 wildtype: 487, total: 1363)
- SeqVal2 (somatic: 269, germline: 428, wildtype: 1410, total: 2107)

SeqVal1 positions were obtained by aligning the reads to the whole human genome, while SeqVal2 positions were obtained by aligning the reads to a reference limited to the targeted human exons. SeqVal2 was considerably noisier due to misalignments. (Note that 101 positions overlapped in the two datasets therefore we removed redundant sites from the combined dataset.)

The second dataset (*whole genome shotgun data*) is from four whole human genome tumour/normal pairs sequenced using the Life Technologies SOLiD system as $25 - 50$bp pair-end reads. These data were aligned to the human genome by using the BioScope aligner. Ground truth for these samples was obtained from orthogonal Illumia exome capture experiments followed by targeted resequencing on the same DNA samples resulting in 113 somatic mutations, 57 germline mutations and 337 wildtypes. These four samples are completely independent of

the training samples in SeqVal1+SeqVal2.

2.2.4 Experimental design

For each of the four classifiers, we used the exome capture data for classifier training, and tested on the whole genome shotgun data. For training, we used the following procedure. Since each of the models accepts hyper-parameters, we applied a 10-fold cross-validation analysis on a range of hyper-parameters to approximate the optimal settings by the one-standard error rule. We applied the resulting settings on all of the exome capture data in the final training step. We obtained a set of discriminative features using ensemble feature selection [1] after training using 40 bootstrap samples and finally computed a feature set aggregated from these 40 samples for each classifier. (For each bootstrap sample, Logit selected the features with nonzero weights, SVM and RF used background elimination to gradually remove the least informative features, and BART selected the set of features with appearance frequencies great than average of $1/106$.) To test the robustness of each classifier to the input set of features, we trained each classifier using each of the other classifiers' feature sets, producing $4 \times 4 = 16$ results, which were then assessed using sensitivity, specificity and accuracy metrics.

To compare our classification methods to standard approaches for SNV detection, we used two popular methods: GATK v1.0.5543M and Samtools v1.16. Samtools `mpileup` and `bcftools` were run independently on the tumour and normal bamfiles to produce SNV calls at the 3369 positions in the exome capture data. Those SNVs present in the tumour list, but not the normal were considered as somatic mutations, otherwise they were considered as non-somatic. For GATK, we used the `UnifiedGenotyper` tool in a similar fashion to classify the positions in the exome capture data. We also compared the results after removing small indel-induced artefacts using GATKs local realignment and base quality recalibration tool. We then compared all methods using accuracy and receiver operator characteristic curves (ROC).

2.3 Results

2.3.1 Classifiers outperform standard approaches

To visually investigate the discriminative ability of the features, we used principal component analysis (PCA) to project the 106-dimensional feature vectors to a three-dimensional space spanned by the first three principal components. Fig. 2.2 shows that somatic mutations were

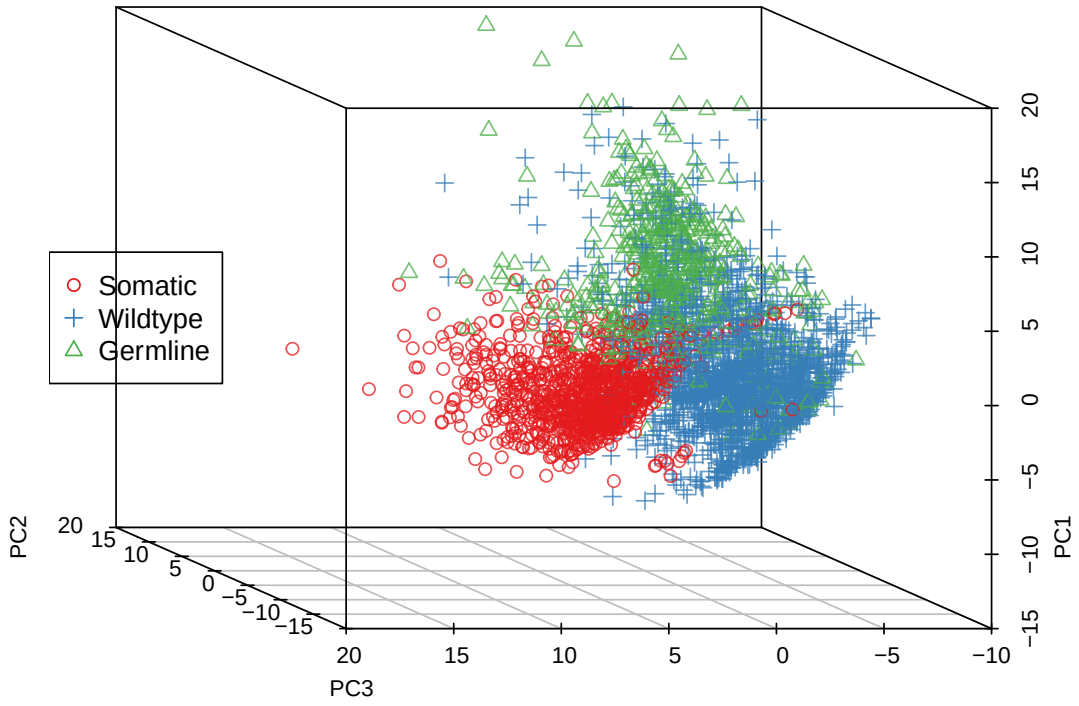


Figure 2.2: The training data projected onto the three-dimensional space spanned by the first three principal components. The somatic mutations (red) are reasonably well-separated from non-somatic mutations (germline - green, wildtype - blue).

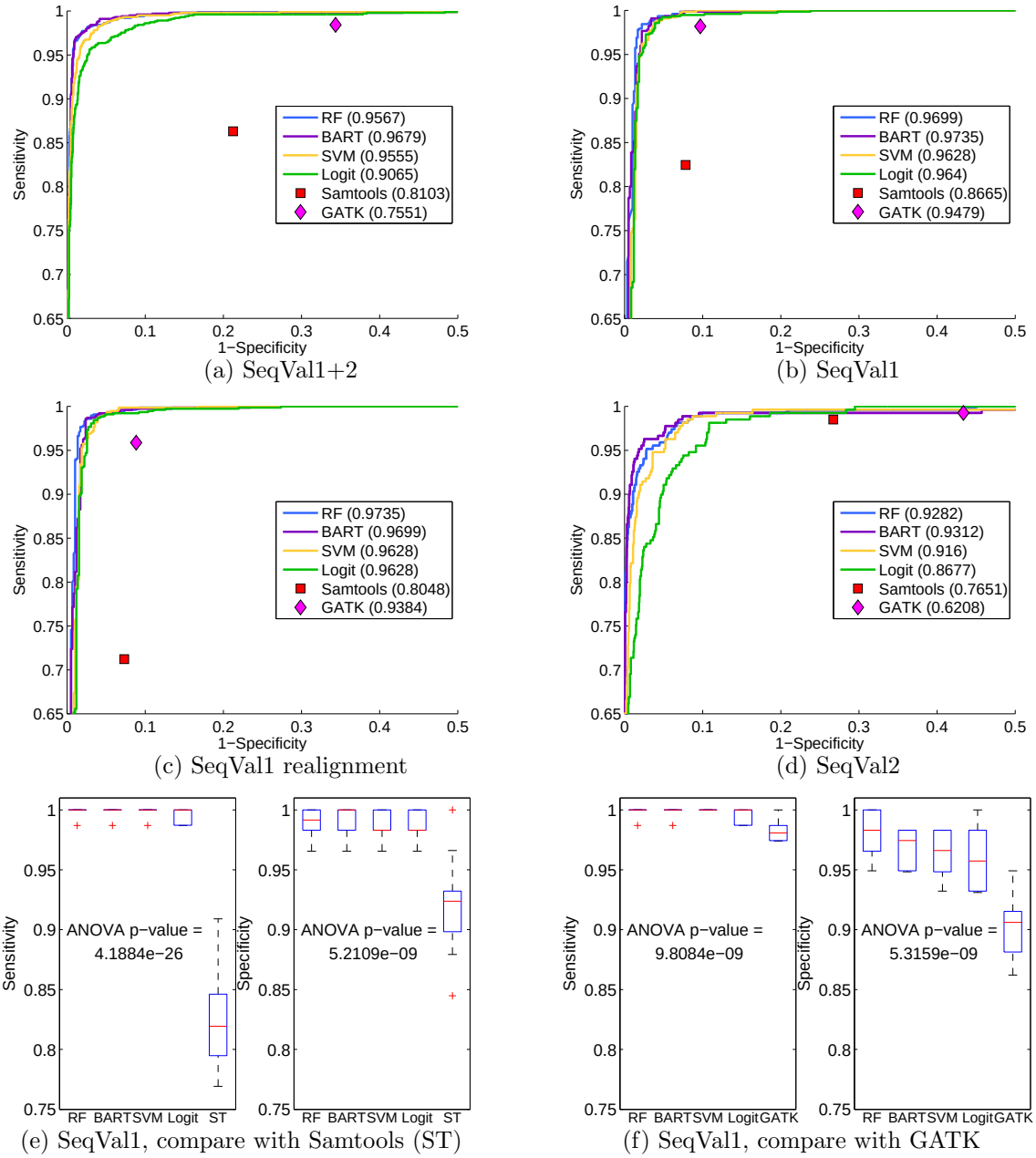
reasonably separated from non-somatic mutations, suggesting that accurate classifiers could potentially be learned from the set of features we chose to examine. Comparison of accuracy on the combined dataset SeqVal1+2 of the different classifiers, GATK's `UnifiedGenotyper` and Samtools `bcftools` (Fig. 2.3(a)) showed that BART was most accurate (0.9679) followed by RF (0.9567), SVM (0.9555) and Logit (0.9065). All classifiers were better than Samtools (0.8103) and GATK (0.7551). We next evaluated the contribution of specificity to the accuracy results using a high fixed sensitivity of 0.99 to establish a probability threshold for each of the classifiers. Specificity at this threshold was 0.9584, 0.9422, 0.9405 and 0.8704 for BART, RF, SVM and Logit respectively, suggesting that Logit had less discriminative power than the other classifiers. The comparative sensitivity and specificity breakdown for Samtools was 0.8631 and 0.7876, and for GATK it was 0.9842 and 0.6563. Thus, Samtools had more balanced misclassifications, whereas GATK was very sensitive but with a lower specificity than the other methods.

Similar patterns were observed for independent analysis of SeqVal1 (Fig. 2.3(b)), although

results for GATK (sens: 0.9819, spec: 0.9031) and Samtools (sens: 0.8245, spec: 0.9218) were better than for the SeqVal1+2 results. We also tested whether local realignment reads around insertions and deletions and base quality re-calibration (post alignment processing tools in the GATK package) improved results. The classifier results were nearly identical to those shown in Fig. 2.3(b) for SeqVal1 (Fig. 2.3(c)). However, while results for Samtools and GATK both showed an improvement in specificity, there was a substantial reduction in sensitivity (Fig. 2.3(c)). We also assessed results on SeqVal2 independently (Fig. 2.3(d)) and found that accuracy was highest for BART (0.9312) followed by RF (0.9282), SVM (0.9160), Logit (0.8677), Samtools (0.7651) and GATK (0.6208). All methods were worse on this dataset than on SeqVal1, although the difference for the classifiers was more moderate than the other methods. The markedly worse performance of Samtools and GATK for this dataset was mainly due to considerably decreased specificity; this dataset was generated from constrained alignments to exons that likely induce many false alignments, thus the classifier methods may be more robust to artefacts introduced by misalignments.

We then assessed the statistical significance of the observed differences between methods using the best performing results for Samtools and GATK (SeqVal1). For each cross validation fold, we fixed sensitivity according to the Samtools results and computed the specificity of the other methods. We then compared the specificity distributions of the methods over all folds using a one-way ANOVA test. Similarly, we evaluated sensitivity distributions by fixing specificity. A similar procedure was then applied to the GATK results. The classifiers were not statistically different from each other in any comparison. However, all classifiers were statistically significantly higher in specificity and sensitivity (ANOVA, $p < 0.00001$) than Samtools and GATK.

To test the generalization performance of the trained classifiers, we applied them to the test data: four cases with whole genome shotgun sequencing from tumour and normal DNA on the SOLiD platform. Despite being trained on exome data, the classifiers performed extremely well and recapitulated the results seen in the cross validation experiments (Fig. 2.4). The accuracy for the classifiers by using the same thresholds for the training data was 0.9487, 0.9487, 0.9369 and 0.9191 for BART, SVM, RF and Logit respectively. Samtools accuracy was 0.9053 followed by GATK at 0.8738. These results indicate that, on a limited dataset, the trained parameters should generalise well to other platforms and are likely robust to overfitting. On the orthogonal test data, all classifiers outperformed GATK and Samtools in both sensitivity and specificity



(a) SeqVal1+2 (b) SeqVal1

(c) SeqVal1 realignment (d) SeqVal2

(e) SeqVal1, compare with Samtools (ST) (f) SeqVal1, compare with GATK

Figure 2.3: (a) Accuracy results from cross-validation experiments on all the exome capture data (SeqVal1+2). All classifiers showed better results than Samtools and GATK's prediction results in terms of ROC comparison. The numbers in parentheses are the prediction accuracy by fixing the sensitivity at 0.99, except for Samtools and GATK's prediction results because their outputs are deterministic. (b) Accuracy results from cross-validation experiments on the exome capture data of SeqVal1. (c) Accuracy results from cross-validation experiments on the exome capture data of SeqVal1 after GATK's local realignment around indels and base quality recalibration. (d) Accuracy results from cross-validation experiments on the exome capture data of SeqVal2. (e) Comparison of classifiers and Samtools's (ST) performance at the specificity and sensitivity level given by Samtools. (f) Comparison of classifiers and GATK's performance at the specificity and sensitivity level given by GATK.

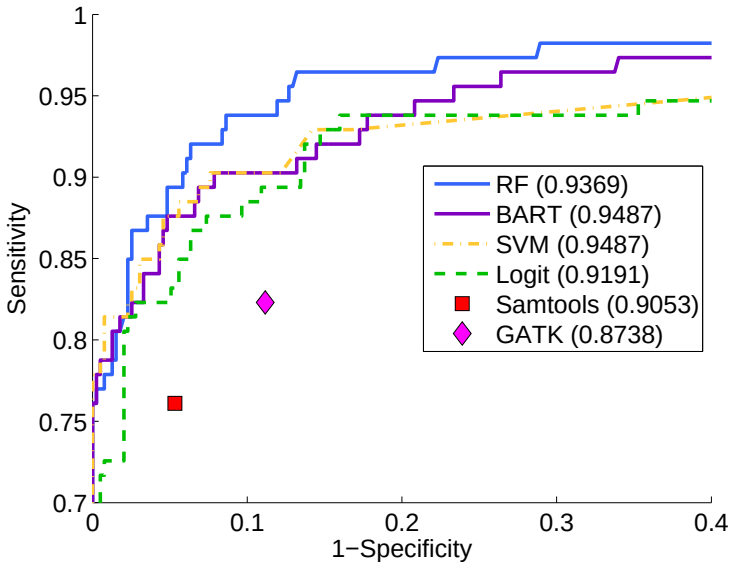


Figure 2.4: ROC curves derived from the held-out whole genome shotgun independent test data from four cases show different classifiers' prediction results as well as Samtools and GATK's prediction results. The numbers in the parentheses are the prediction accuracy by using the same threshold as for the exome capture data (except for Samtools and GATK's prediction results.)

with BART exhibiting the best overall performance.

We next investigated whether the use of classifiers or the expanded set of features contributed to increased performance of our methods. Using the SeqVal1 dataset, we restricted the analysis to only Samtools-derived features (x_{1-40}) and compared the results of classifiers with those of the Samtools caller. All classifiers performed statistically better than the Samtools caller. For the second experiment, we restricted the analysis to only the GATK features (x_{41-80}). As for Samtools, the classifiers showed statistically significantly better results than those of the GATK caller. These results suggest that the classifiers on the same set of features for both Samtools and GATK better approximated the "optimal" decision boundary without the use of heuristic thresholds employed by the naive methods and demonstrate the clear advantages of the machine learning approaches we used.

Finally, we studied the effect of the additional 26 features we introduced (Table 2.3, x_{81-106}) to the Samtools and GATK features in order to boost weak mutation signals in the tumour and decrease the influence of germline polymorphisms. We compared the performance of RF and BART on different feature sets: Samtools alone (x_{1-40}), GATK alone (x_{41-80}), our 26 features alone (x_{81-106}), and all features combined (x_{1-106}). As shown in Fig. 2.5, by using all the features, RF and BART showed the best performance in terms of accuracy. (Although the

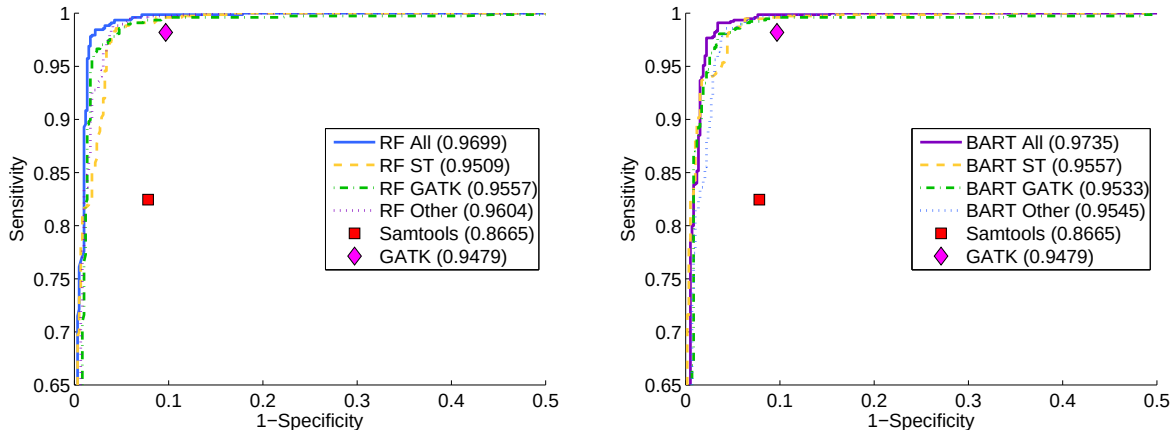


Figure 2.5: (a) The performance of RF on different feature sets. RF All: RF's accuracy by using all the features, RF ST: RF's accuracy by using only the Samtools features, RF GATK: RF's accuracy by using only the GATK features, RF Other: RF's accuracy by using all the new constructed 26 features. (2) The performance of BART on different feature sets. BART All means BART's accuracy by using all the features. BART ST, BART GATK and BART Other are similarly defined.

26 novel features did not dramatically increase the performance, partially because the original feature sets already captured some information in the new feature sets.) We therefore suggest that while the use of the machine learning classifiers accounts for the majority of improvement over naive methods, further improvement is achievable with the introduction of novel features thus illustrating the power of the flexible framework we used in this study.

2.3.2 Robustness of classifiers to different feature sets

We used ensemble feature selection to output a set of the most salient, discriminative features for each classifier, leading to four feature sets overall. We then fit each classifier to the exome capture data using only the four selected feature sets output from the ensemble feature selection method and classified the SOLiD data. We noted (Table 2.4) that overall, BART and RF were most robust to the initial set of features and performed similarly for each of the four sets, showing stable performance in the presence of variable initial feature sets. Interestingly, all four methods performed equally well on the features chosen by ensemble feature selection applied to BART (Table 2.4) with a mean accuracy of 0.9453 compared to $0.9359, 0.9339$ and 0.9157 for SVM, Logit, and RF chosen features, respectively.

In summary, all classifiers performed better than the naive methods for somatic SNV prediction, with RF and BART showing the best performance. RF classifier is slightly more sensitive (Fig. 2.4), while BART has slightly higher specificity (Fig. 2.3(d)). The performance of SVM

Table 2.4: The accuracy of classifiers on the SOLiD data by using different feature sets. Here RF_Feature means the feature selected by RF classifier. BART_Feature, SVM_Feature and Logit_Feature are similarly defined. The numbers in parentheses are the number of feature selected.

Model \ Feature	RF_Feature (18)	BART_Feature (23)	SVM_Feature (17)	Logit_Feature (17)
RF	0.9369	0.9487	0.9448	0.9329
BART	0.9369	0.9428	0.9369	0.9310
SVM	0.9034	0.9408	0.9369	0.9408
Logit	0.8856	0.9487	0.9250	0.9310
Mean	0.9157	**0.9453**	0.9359	0.9339

and Logit is relatively poor, especially in the presence of outliers as can be seen from Fig. 2.3(d). Importantly, both RF and BART are less sensitive to different feature sets compared to SVM and Logit (Table 2.4). Overall, the data support using RF and BART over SVM and Logit and suggest that RF may achieve better sensitivity while BART will achieve higher specificity, though both methods are extremely comparable.

2.3.3 Discriminative features are different for tumour and normal data

The description of the set of features selected by BART is given in Table 2.5. The features fell into five broad categories: i) allelic count distribution *likelihoods*: provided by both Samtools and GATK; ii) *base qualities* such as the sum of reference base qualities, sum of non-reference base qualities, sum of squares of non-reference base quality ratio; iii) *strand bias* such as sum of the pooled estimation of strand bias on both strands; iv) *mapping qualities* such as the mean square mapping quality; and v) *tail distance* such as sum of squares of tail distance for non-reference bases and sum of squares of non-reference tail distance (min distance of variant base to the ends of the read) ratio. Notably, the features are often different in the tumour and normal. For example, the *reference* base qualities (x_6) are selected in the normal, but for tumour both reference (x_{26}) and non-reference base qualities (x_{28}) are selected. Other tumour-specific features included sum of tail distance of the non-reference bases (x_{37}), allele frequency for each non-reference allele (x_{63}), and variant confidence normalized by depth (x_{71}). Therefore, BART assigned unequal weights to the features in the normal and tumour, suggesting that the improved accuracy is due to treating the tumour and normal data differentially to optimize the contribution of the discriminant features. We note in Table 2.5 that BART selected several of the new features we designed (x_{83}, x_{96}, x_{97}, x_{99}, x_{101}, x_{102}, x_{105}). These were not in Samtools

Table 2.5: The BART model selected features. As expected, the likelihoods provided by both Samtools and GATK, the base quality, mapping quality, strand bias, tail distance features are relevant. The features selected from the normals and tumours are different.

Index	Feature definition	Tumour	Samtools	GATK
6	sum of reference base qualities		√	
10	sum of reference mapping qualities		√	
19	$\max_{G_i \neq aa}(P(D \mid G_i))$		√	
26	sum of reference base qualities	√	√	
28	sum of non-reference base qualities	√	√	
37	sum of squares of tail distance for non-reference bases	√	√	
38	$P(D \mid G_i = aa)$	√	√	
41	QUAL: phred-scaled probability of the call given data			√
53	sumGLbyD			√
57	$P(D \mid G_i = aa)$			√
60	$P(D \mid G_i = bb)$			√
63	AF: allele frequency for each non-ref allele	√		√
69	MQ: root mean square mapping quality	√		√
71	QD: variant confidence/unfiltered depth	√		√
73	sumGLbyD	√		√
77	GQ: genotype quality computed based on the genotype likelihood	√		√
83	the difference between the sum of the base qualities of the current site and the next site			
96	sum of the pooled estimation of strand bias on both strands max(forward, reverse)			
97	sum of the pooled estimation of strand bias on both strands $\sum$(forward, reverse)			
99	Reverse strand non-reference base ratio			
101	sum of squares of non-reference base quality ratio			
102	Sum of non-reference mapping quality ratio			
105	Sum of squares of non-reference tail distance ratio			

or GATK, and some were a combined calculation from the tumour and normal data. This illustrates the advantage of the classifiers' ability to add arbitrary features and the importance of simultaneous (not independent) treatment of the tumour and normal data.

2.3.4 Sources of errors and sub-classification of wildtypes

We subgrouped the wildtype positions (false positives from the original predictions) by their feature vectors in order to characterize false positives due to distinct sources of error. Using

the wildtype positions from Seqval1, we identified the features which were not unimodal with the dip statistic [87] and selected 28 features with p-value < 0.1. We then used PCA to project the features to a seven-dimensional space spanned by the first seven principal components, and modelled the wildtypes in the seven-dimensional space (the first seven principal components account for about 95% of the variance) using a Gaussian mixture model based clustering algorithm fit with the Expectation-Maximization algorithm (EM) [63] (Chapter 1). We used the Bayesian information criteria (BIC) score to select six clusters. The number of wildtypes in Group 1 to Group 6 was 37, 189, 43, 181, 6 and 31, respectively. We attributed the six events in Group 5 to outliers and excluded this group from further analysis. We then identified discriminant features of the different groups, using an ANOVA test followed by a multiple comparison test on each feature.

Broadly the groups had the following characteristics. Group 1 (black) featured high values for x_{102} and x_{103} indicating disproportionate mapping qualities in the tumour compared to the normal. Thus, the tumour reads harbouring variants mapped with higher qualities than the normal reads harbouring variants at the same genomic location. In addition, Group 1 exhibited strand bias as shown by high values of (x_{96} and x_{97}). The events in this group had low values for x_{57} and x_{77} which indicated low genotype qualities (confidence in the genotype call). Taken together, these data suggest that the combination of poor mapping quality of the normal reads and the strand bias may be affecting the callers' ability to accurately call these variants.

Group 2 (red) is characterized by high values of x_{96} suggesting strand bias (Fig. 2.6). We examined the surrounding sequence content around these variants and found the majority of the variants in this group had a common tri-nucleotide sequence GGT, changing to GGG (Fig. 2.7), a pattern which has been discovered in whole genome methyl-Seq experiments [139]. Thus, we expect the false positive events in this group to be induced by systematic artefacts owing to sequencing errors at specific tri-nucleotide sequences.

The discriminative features for Group 3 (green) events were characterized by mapping quality-related features (x_{10}, x_{11}, x_{50}, x_{70}, x_{49}, x_{69}). Thus, these wildtypes may be the result of misaligned reads, or simply repetitive regions for unambiguously aligning short reads to. To investigate these wildtypes, we computed the UCSC mapability (`http://genome.ucsc.edu/cgi-bin/hgTrackUi?db=hg18&g=wgEncodeMapability`) of each site. The mappability of a site depicts the uniqueness of the reference genome in a window size of 35. Overall, Group 3 wildtypes have considerably lower mappability scores than the other groups and therefore can be best explained

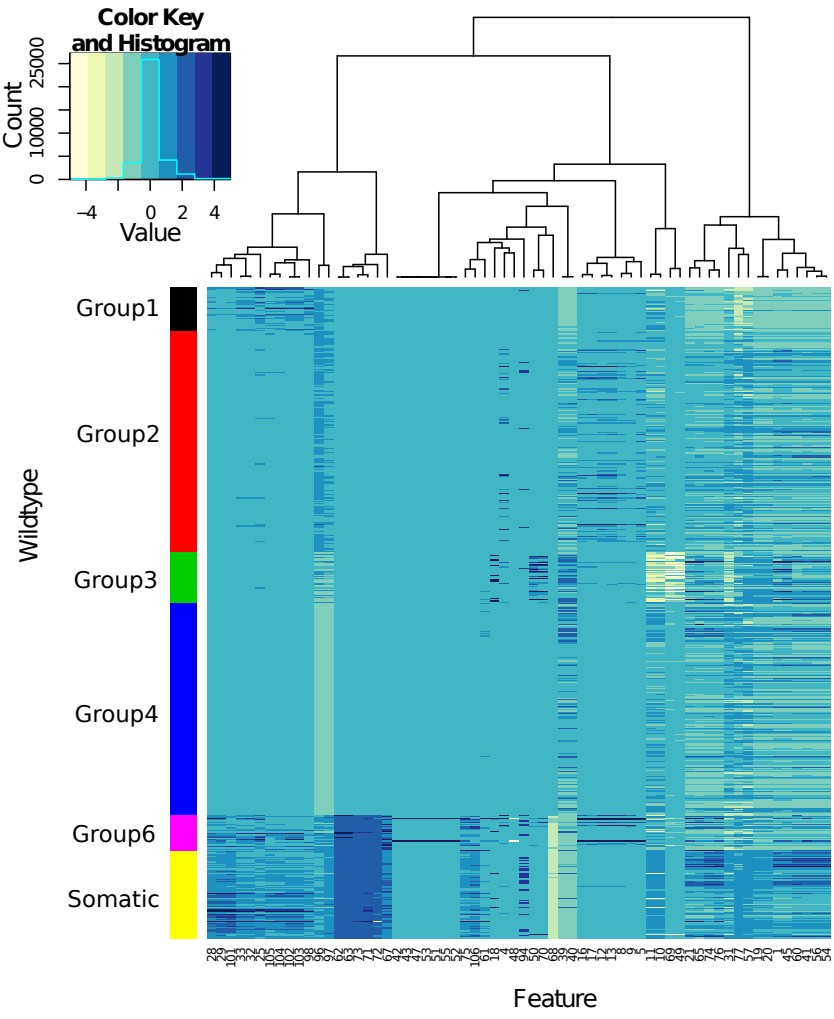


Figure 2.6: The feature heatmap obtained with $K = 6$ clusters. The six events in Group 5 were not shown here.

by characteristics of the genome at these positions that make variant calling error prone.

For the wildtypes in Group 4 (blue), the pooled estimated strand-biases are zero (x_{96} and x_{97}). This is because these two features were computed after passing Samtools internal base quality filter threshold of 13, and the variant alleles had small base qualities so they didn't pass this filter. The Samtools caller utilizes this base quality filter and therefore did not call these positions as variants (large x_{39} and x_{40}). Group 4 wildtypes were also characterized by the GGT to GGG systematic sequencing artefact we observed for Group 2 and therefore is fundamentally similar to Group2, but may be easier to detect owing to poor base qualities at the site of the sequencing error.

Interestingly, Group 6 (magenta) exhibited very similar patterns to the true somatic muta-

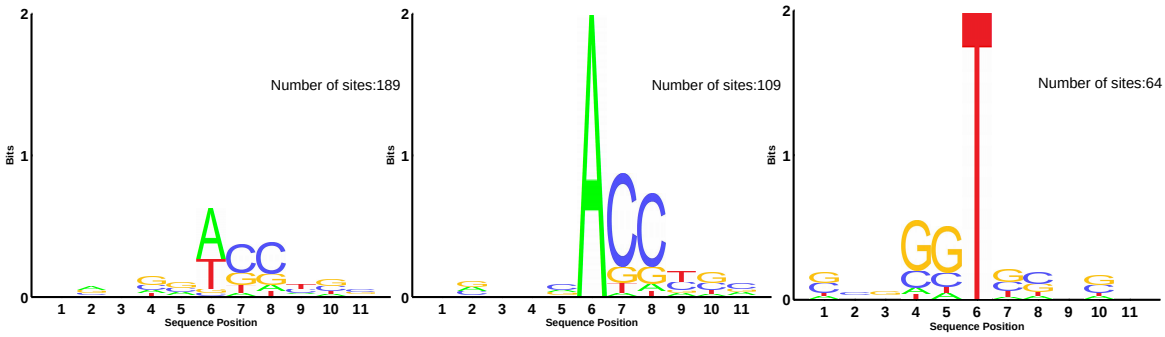


Figure 2.7: Group two wildtype sequence motifs centred at error sites. Errors occur at the $6th$ bases. (a) The logo for all the group 2 wildtypes, (b) the wildtypes whose reference base is 'A', (c) the wildtypes whose reference base is 'T'.

tions (yellow) and thus made them challenging to interpret. Upon inspection, many of these positions had weak signals for a variant in the normal data, but perhaps not enough to induce a variant call. The tumour data, conversely (as shown by (x_{62}, x_{63}, x_{71}, x_{73})) exhibited strong signals for a variant. Thus, the weak signals in the normal data were likely being prematurely thresholded out by the naive methods. Indeed, the Samtools caller called 13 of the 31 Group 6 events as somatic while GATK called 29 of the 31 events as somatic. The characteristics of the positions in Group 6 underscore the strength of simultaneously considering the tumour and normal features that we suspect enhances the ability of the classifier to choose better decision boundaries.

2.4 Discussion

We studied the use of feature-based classifiers for the purpose of somatic mutation detection in tumour/normal pair HTS data. Using an extensive set of ground truth positions, we trained four different machine learning classifiers using features extracted from existing software tools and novel features we computed ourselves. All four classifiers statistically significantly outperformed popular software packages that use a naive way to detect somatic mutations, treating the tumour and normal data independently. Results were consistent between a cross-validation analysis of the training data and a completely independent test data set derived from an orthogonal sequencing platform. Our results encapsulate three key results: i) classifiers can be trained using principled machine learning techniques to significantly improve somatic mutation detection; ii) feature selection analysis revealed that our classification method selects different features in the tumour and normal datasets to optimize classification ability, underscoring that simultaneous

rather than independent analysis of the paired data is important; and iii) we identified five distinct groups of false positive results. This last result indicates that feature-based analysis of 'negative' or wildtype positions can be helpful to guide future developments in software pipelines that operate upstream of variant calling.

2.4.1 Progress in SNV prediction

Since the publication of `mutationSeq`, dozens of tools have been developed to call somatic SNVs from HTS data [37, 73, 106, 111, 115, 174, 179]. Probabilistic model based approached such as JoinSNVMix [174] may have difficulties in modelling the combinations of errors in HTS sequencing data, thus discriminative features such as strand-bias are typically ignored in the modelling framework. To achieve high specificity for these methods, heuristic filters based on some features can be used to remove false-positives in HTS data [37, 179]. Some other methods are designed for special cases, e.g., calling mutations from targeted high-coverage sequencing data [73]. An interesting direction is to systematically compare different SNV callers on extensive simulation data through a challenge competition [58]. However, we should notice that the results may not only reflect the performance of different algorithms, but also how much efforts each team puts on the competition since the results are also correlated with the number of participants in each team [58]. A promising approach for getting high performance may be to aggregate the results from different somatic mutation callers [80].

Chapter 3

Predicting mutations that influence gene expression

"What I cannot create, I do not understand."

– Richard Feynman, 1988

3.1 Introduction

Human cancers acquire malignant properties following a stepwise accumulation of somatic genomic alterations [213] and subsequent evolutionary selection on resultant phenotypic changes. Genomic mutations (loosely classified as single nucleotide variants (SNVs), small insertions and deletions (indels), copy number alterations and genomic rearrangements) show widespread variation in their functional impacts on gene products, biochemical pathways, and phenotypic properties. Consequently, the effect of a mutation is often difficult to predict. Previous computational approaches to predict functional effects of mutations include: evolutionary conservation of the mutation sites across species, the chemical properties of amino acid substitutions [168], and the frequency of mutations of a gene of interest relative to its expected background rate of mutations [117]. These approaches rely on interpretation of DNA sequences alone and do not consider other molecular measurements such as gene expression, methylation or proteome measurements that are co-acquired from the same tumour samples. Thus, histological or molecular context of mutations is often ignored in their interpretation. To address this deficiency, we propose that additional patterns representing functional consequences of mutations can be determined through simultaneous analysis of mutation *and* gene expression data.

In this chapter, we study the impact of mutations on gene expression as a means of quantifying their potential functional effects. This concept is motivated by biological hypotheses predicting that some functional mutations will exhibit a "transcriptional shadow". For example, loss-of-function mutations such as nonsense mutations, frame-shift small insertions and deletions (indels), and splice-site mutations can result in loss of expression of genes harbouring these

mutations (cis-effects). This is primarily due to nonsense-mediated mRNA decay mechanism to eliminate transcripts with premature stop codons [152]. By contrast, some mutations can disrupt the expression of numerous other genes in the same biochemical pathway (trans-effects). For example, functional mutations in genes that encode signalling proteins or transcription factors can dysregulate downstream target genes. Mutations in chromatin modifying and methylation factors can promote or inhibit the accessibility of DNAs to proteins thus influencing transcription. These functional mutations tend to cast a long transcriptional shadow over the genes across the genome [39]. The dysregulation of gene expression indicates the disruptions of specific genes or pathways, and may help pinpoint functional mutations, and even cancer driver mutations. Furthermore, association of mutation with gene expression in individual tumours may identify patient-specific characteristics that could be exploited with a personalized treatment approach.

There are few computational tools available [164] to systematically identify mutations impacting gene expression. CONEXIC [2] is a probabilistic approach to detect driver copy number regulators and their target genes. EPoC [97] derives driver copy number alterations and their target genes by using differential equations to model the expression synthesis rate of a gene as a function of its copy number and the regulatory effects of other genes. MOCA [130] detects differently expressed genes in the presence of mutations in a gene, and tests the significance of the correlation (between mutation and gene differential expression). PARADIGM [210] integrates copy number and expression to identify disrupted pathways. DriverNet [14] uses a combinatorial approach and a greedy algorithm to nominate cancer driver genes. However, none of these methods can identify individual mutations that correlate with dysregulated gene networks.

We present a novel statistical model, `xseq` using a hierarchical Bayes approach and apply it to the analysis of thousands of tumour datasets available through TCGA, systematically examining the impact of somatic mutations on expression profiles across 12 tumour types. We demonstrate the robustness of `xseq` by conducting extensive computational benchmarking, and by testing `xseq` on an independent breast cancer dataset. We identify 30 novel cis-effect tumour suppressor gene candidates, statistically enriched in loss-of-function mutations and frequent bi-allelic inactivations. We identify 150 genes from trans-effect analysis impacting expression networks in the 12 cancer types, with 60 known cancer genes and 89 novel predictions. Notably, 29 of these newly predicted genes are known interacting partners of cancer driver genes. Based on the trans-analysis, we find two important characteristics of mutations impacting gene expression that could not be revealed with other methods: (1) a stratification (partition) of patients

harbouring known driver mutations, but that exhibit different downstream gene expression consequences; (2) identification of mutations driving expression patterns that are stable across tumour types, thereby nominating important molecular targets for therapeutic intervention, transcending anatomic sites of origin.

3.2 Methods

3.2.1 Modelling the effects of mutations on expression with `xseq`

The `xseq` model is predicated on the idea that mutations with functional effects on transcription will exhibit measurable signals in mRNA transcripts biochemically related to the mutated gene –thus imposing a transcriptional shadow across part (or all) of a pathway. To infer this property, three key inputs are required for the model (Fig. 3.1(a)): a patient-gene matrix encoding the presence/absence of a mutation (any form of somatic genomic aberrations that can be ascribed to a gene, e.g., SNVs, indels, or copy number alterations); a patient-gene expression matrix encoding continuous value expression data (e.g., from RNA-Seq or microarrays); and a graph structure encoding whether two genes are known to be functionally related (e.g., obtained through literature, databases, or co-expression data). `xseq` uses a precomputed 'influence graph' [207] as a means to incorporate prior gene-gene relationship knowledge into its modelling framework. For analysis of mutation impact in-*cis*, the graph reduces to the simple case where the mutated gene is only connected to itself. Given the inputs, we get the `xseq` structures and the actual expression value of the nth gene connected to mutated gene g in patient m, denoted by $Y_{g,m,n}$ (Table 3.1).

The output of `xseq` consists of: a) the probability that a mutated gene g influences gene expression across the population of patients with mutations in g (denoted by $P(D_g = 1 \mid \mathcal{D})$, where $\mathcal{D}$ represents the observed gene expression as the Y variable in Fig. 3.1(b); Table 3.1); and b) the probability that a mutation in gene g in an individual patient m influences expression within that patient (denoted by $P(F_{g,m} = 1 \mid \mathcal{D})$).

In addition to the random variables D_g, $F_{g,m}$, and $Y_{g,m,n}$, `xseq` also models the gene expression distribution over the patient population with gene-specific three component mixture models of Student's t emission densities. The three mixture components represent down-regulation, neutral, or up-regulation, respectively (Fig. 3.1(a)). $G_{g,m,n} \in \{\mathcal{L} = -1, \mathcal{N} = 0, \mathcal{G} = 1\}$ denotes the status of the nth gene connected to gene g in patient m, where $\mathcal{L}, \mathcal{N}$, and $\mathcal{G}$ represent down-

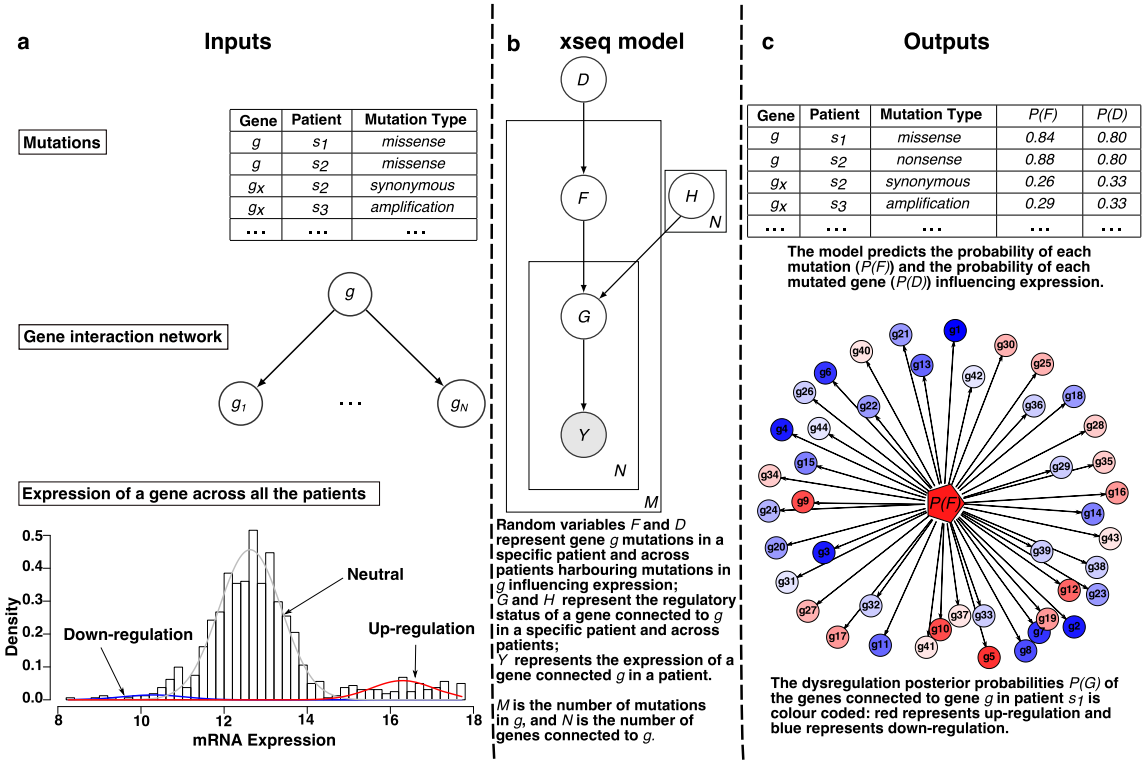


Figure 3.1: Overview of the `xseq` modelling framework. (a) The inputs to the `xseq` model: a mutation matrix typically from next generation sequencing, a gene-interaction network and a gene expression matrix. `xseq` models the expression of a gene across all the patients by mixture distributions. The three mixture components represent down-regulation, neutral, and up-regulation, respectively. (b) The graphical model representation of `xseq` with the plate notation. Circles represent random variables and arrows denote dependencies between variables. Boxes are plates which represent replicates. For example, the graph represents a gene mutated in M patients (we assume that a gene is mutated only once in a patient), and the gene is connected to N genes. (c) `xseq` predicts the conditional probabilities (given observations of Y) of each gene ($P(D)$), each mutation ($P(F)$) influencing expression and the regulatory probabilities of the genes connected to the mutated gene in a patient ($P(G)$). $P(D) = P(D = 1 \mid \mathcal{D})$; $P(F) = P(F = 1 \mid \mathcal{D})$; $P(G) = P(G = g \mid \mathcal{D}), g \in \{-1, 0, 1\}$.

regulation, neutral, and up-regulation, respectively. The number of genes connected to g is obtained from the influence graph. The central assumption is that a mutation in gene g of patient m impacting gene expression (denoted by $F_{g,m}$=1) more frequently co-associates with non-neutral states in its connected genes, compared to the mutations that do not impact expression. The specific direction of expression is encoded by $H_{g,n} \in \{\mathcal{L} = -1, \mathcal{G} = 1\}$ to denote the nth gene connected to gene g is up-regulated or down-regulated when mutations in g influence expression. (We also consider a simplified model, `xseq-simple` without modelling the directionality of gene-regulation for a specific gene, i.e., without the H variable in Fig. 3.1(b) for

Table 3.1: Description of the random variables in `xseq`

Name	Description	Range
D_g	Whether mutations in gene g across patients impact expression	Binary (0 means not impacting expression, and 1 means impacting expression)
$F_{g,m}$	Whether gene g's mutation specifically in patient m impact expression	Binary (0 means not impacting expression, and 1 means impacting expression)
$G_{g,m,n}$	The nth gene connected to g in patient m is up-regulated, neutral, or down-regulated	Ternary ($\mathcal{L} = -1$, $\mathcal{N} = 0$ and $\mathcal{G} = 1$ represent expression down-regulation, neutral and up-regulation, respectively)
$H_{g,n}$	The nth gene connected to g is up-regulated or down-regulated across the patients harbouring gene g mutations that correlated with connected gene dysregulation	Binary ($\mathcal{L} = -1$ and $\mathcal{G} = 1$ mean down- and up-regulation of expression, respectively)
$Y_{g,m,n}$	The expression of the nth gene connected to g in patient m	Real (observed)

simplicity of inference.) To represent a recurrent pattern of expression impact across multiple patients, we consider information across all patients with a mutation in gene g. This allows for borrowing of statistical strength across multiple gene expression patterns associated with mutations in order to generalize whether a mutated gene impacts expression across the population (denoted by $D_g = 1$). Fig. 3.2(a) shows a simple `xseq` model for a mutated gene.

Based on the `xseq` model structure in Fig. 3.1(b), for a mutated gene g, `xseq` specifies a joint distribution [150] with both discrete and continuous random variables (assuming that g is mutated in M patients and g has N connected genes):

$$\begin{aligned} & P(D=d)\prod_{m=1}^{M} P(F_m = f_m \mid D=d) \\ & \prod_{n=1}^{N} p(y_{m,n} \mid G_{m,n} = \mathcal{g}_{m,n}) P(G_{m,n} = g_{m,n} \mid F_m = f_m, H_n = h_n) \\ =& \theta_{D=d} \prod_{m=1}^{M} \theta_{F=f_m|D=d} \prod_{n=1}^{N} p(y_{m,n} \mid G_{m,n} = \mathcal{g}_{m,n}) \theta_{G=\mathcal{g}_{m,n}|F=f_m,H=h_n} \end{aligned} \tag{3.1}$$

In Equation 3.1, we write the joint distribution in terms of the model parameters (the θs in Table 3.2). Notice that when $f_m = 0$, $\theta_{G=\mathcal{g}_{m,n}|F=0,H=h_n} = \theta_{G=\mathcal{g}_{m,n}|F=0}$ as in Table 3.2. For simplicity, we consider the case with only one mutated gene and we remove 'g' in the notations.

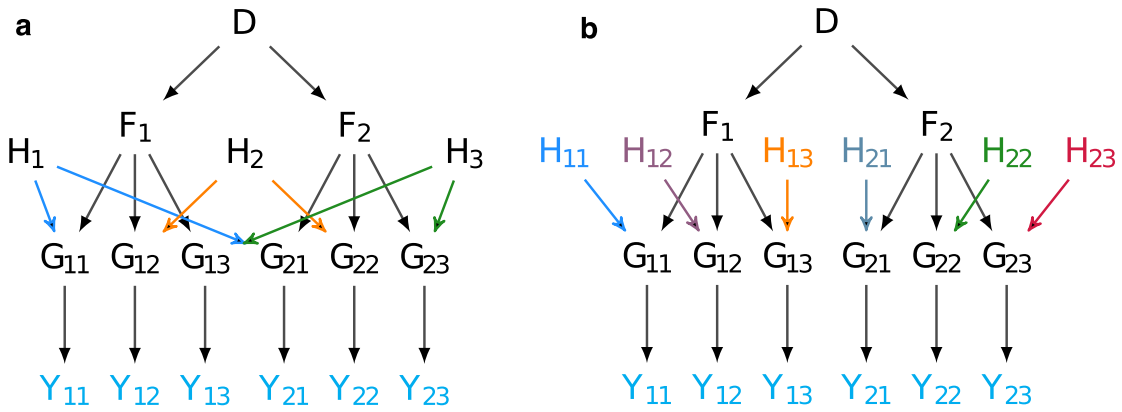


Figure 3.2: A simple `xseq` model. This `xseq` model has just one mutated gene g, two patients harbouring mutations in gene g, and three genes connected to g. Here we drop 'g' in the figure for simplicity. (a) H is hidden and (b) H is observed.

Here we assume that a gene is just mutated once in a specific patient, i.e., M equals to the number of accumulated mutations in gene g. Therefore, m is the patient (mutation) index, and n is the gene index, e.g., $Y_{m,n}$ represents the expression of the n-th gene connected to g in the m-th patients harbouring mutations in g. We now explain how we execute parameter learning and inference over this joint distribution.

3.2.2 Inference and learning in `xseq`

We use the Belief Propagation algorithm [158] for inference and the Expectation-Maximization (EM) algorithm for parameter learning [45]. The inference problem is to compute the conditional distributions (given observations of the gene expression Y) $p(d_g \mid \mathcal{D})$, $p(f_{g,m} \mid \mathcal{D})$, and $p(g_{g,m,n} \mid \mathcal{D})$ given the expression data. The learning problem is to estimate the conditional probabilities of a variable given its parents, e.g., $\theta_{F=1|D=1}$ –the probability of a mutation impacting expression in a specific patient given that this gene's mutations impact expression across patients (Fig. 3.1(b), Table 3.2). For clarity of presentation, we have removed the subscripts and directly refer to D, F, and G. For example, we use $P(D)$, $P(F)$ and $P(G)$ to denote the conditional probabilities $P(D_g = 1 \mid \mathcal{D})$, $P(F_{g,m} = 1 \mid \mathcal{D})$, and $P(G_{g,m,n} = g_{g,m,n} \mid \mathcal{D})$, where $g_{g,m,n} \in \{-1, 0, 1\}$, respectively.

Pearl's Belief Propagation algorithm (polytree algorithm) was first proposed by Judea Pearl in 1982 [158] to solve inference problems for tree structure Bayesian networks. (There is only

Table 3.2: Description of the conditional distributions in `xseq`. Here g refers to a mutated gene in general.

Name	Description	Range
$\theta_{D=1} \stackrel{\text{def}}{=} P(D=1)$	Probability of gene g's mutations impacting expression across patients	Real
$\theta_{F=1\|D=0} \stackrel{\text{def}}{=} P(F=1\|D=0)$	Probability of gene g's mutation impacting expression in patient m given that gene g's mutations do not impact expression across patients	Real
$\theta_{F=1\|D=1} \stackrel{\text{def}}{=} P(F=1\|D=1)$	Probability of gene g's mutation impacting expression in patient m given that gene g's mutations impact expression across patients	Real
$\theta_{G=\mathcal{L}\|F=0} \stackrel{\text{def}}{=} P(G=\mathcal{L}\|F=0)$	Probability of a gene connected to g is down-regulated in patient m given that g's mutation in m do not impact expression	Real
$\theta_{G=\mathcal{N}\|F=0} \stackrel{\text{def}}{=} P(G=\mathcal{N}\|F=0)$	Probability of a gene connected to g is neutral in patient m given that g's mutation in m do not impact expression	Real
$\theta_{G=\mathcal{L}\|F=1,H=0} \stackrel{\text{def}}{=} P(G=\mathcal{L}\|F=1,H=0)$	Probability of a gene connected to g is down-regulated in patient m given that g's mutation in m impact expression and this gene is generally down-regulated when mutations in g impact expression	Real
$\theta_{G=\mathcal{N}\|F=1,H=0} \stackrel{\text{def}}{=} P(G=\mathcal{N}\|F=1,H=0)$	Probability of a gene connected to g is neutral in patient m given that g's mutation in m impact expression and this gene is generally down-regulated when mutations in g impact expression	Real
$\theta_{G=\mathcal{L}\|F=1,H=1} \stackrel{\text{def}}{=} P(G=\mathcal{L}\|F=1,H=1)$	Probability of a gene connected to g is down-regulated in patient m given that g's mutation in m impact expression and this gene is generally up-regulated when mutations in g impact expression	Real
$\theta_{G=\mathcal{N}\|F=1,H=1} \stackrel{\text{def}}{=} P(G=\mathcal{N}\|F=1,H=1)$	Probability of a gene connected to g is neutral in patient m given that g's mutation in m impacts expression and this gene is generally up-regulated when mutations in g impact expression	Real
$p(y\|G=\mathcal{L})$	The likelihood of observing expression of y in a gene given that this gene is down-regulated (the conditional probability density function at y)	Real
$p(y\|G=\mathcal{N})$	The likelihood of observing expression of y in a gene given that this gene is neutral	Real
$p(y\|G=\mathcal{G})$	The likelihood of observing expression of y in a gene given that this gene is up-regulated	Real

Although in probability theory, the notation $D=1$ represents the set of events $\{D=1\}$, here for simplicity we use the random variables to index the conditional probabilities.

one undirected path between any two nodes. In addition, a child node has only one parent.) It was then extend to do inference for polytrees [101]. (A child node can have multiple 'parents'.) Notice that `xseq-simple` is a tree, while `xseq` is not a tree or a polytree, e.g., for the `xseq`

model in Fig. 3.2(a), there is an undirected loop $D \to F_1 \to G_{11} \to H_1 \to G_{21} \to F_2 \to D$. Therefore, generally speaking inference in `xseq` is difficult, i.e., the time and memory complexity is exponential in the number of mutations in g. However, if H is given, `xseq` is equivalent to a polytree because we can "break" H to convert the toy `xseq` model in Fig. 3.2(a) to a polytree as given in Fig. 3.2(b).

In addition to simplify the inference problem, there are many situations where H is observed. First, from pathway information, we may know that gene g up-regulates gene a. The directionality of gene regulation information can be added to the `xseq` model as an observed node H. Then we can use `xseq` to search the mutations in gene g that correlate with gene a's up-regulation. Secondly, if we observe that gene g up-regulates gene a in a discovery dataset, given a validation dataset, we also want to inspect whether the mutations in gene g up-regulate gene a.

The Belief Propagation algorithm computes the conditional distributions $p(d_g \mid \mathcal{D})$, $p(f_{m,n} \mid \mathcal{D})$ and $p(g_{g,m,n} \mid \mathcal{D})$ in two phases of message passing: 1) A leaf node (an observed gene expression value node $Y_{g,m,n}$) sends to its parent $G_{g,m,n}$ a message, namely the likelihoods $p(y_{g,m,n} \mid G_{g,m,n})$ over parent $G_{g,m,n}$ (for different values that $G_{g,m,n}$ can take). After receiving all the messages from its children, a node can construct and send messages to its parents. Again, a message sent to a parent is a distribution over the parent, e.g., the message $G_{g,m,n}$ sent to $F_{g,m}$ is a distribution over $F_{g,m}$. The detailed definitions of these messages are described below. This process is repeated until the roots of the tree - the nodes without any parents receive all the messages from their children. 2) A root node can update its conditional distribution by normalizing the product of all the incoming messages and its own prior distribution. Then a root node can construct and send messages down to its children. A top-down message is also a distribution of the parent, e.g., the message D_g sent to $F_{g,m}$ is the conditional distribution $p(d_g \mid \mathcal{D})$ divided by the message $F_{g,m}$ sent to D_g before. This process is again repeated until the leaf nodes have received all the messages from their parents. After receiving all the incoming messages from its neighbours (both parents and children), a node can update its belief – the conditional distributions.

Details of the message passing steps in xseq To simplify our description of the inference algorithm, we use a simple example with just one mutated gene, and H is given. In addition, this gene is mutated in M patients and this gene is connected to N genes. Formally, at the

first message collection phase, the bottom-up messages (with superscript $(-)$) consist of (in sequential order):

$$m^{(-)}_{Y_{m,n}\to G_{m,n}} = p(y_{m,n} \mid G_{m,n}) \tag{3.2}$$

$$m^{(-)}_{G_{m,n}\to F_m} = \sum_{g_{m,n}} \theta_{G=g_{m,n}|F=f_m,H=h_n} \mathrm{bel}^{(-)}_{G_{m,n}} \tag{3.3}$$

$$m^{(-)}_{F_m\to D} = \sum_{f_m} \theta_{F=f_m|D=d} \mathrm{bel}^{(-)}_{F_m} \tag{3.4}$$

where $\mathrm{bel}^{(-)}_{G_{m,n}}$ and $\mathrm{bel}^{(-)}_{F_m}$ are the bottom-up belief states of $G_{m,n}$ and F_m, respectively. The bottom-up belief of a node is updated after the node receives all the bottom-up messages from its children, for example,

$$\mathrm{bel}^{(-)}_{G_{m,n}} \propto m^{(-)}_{Y_{m,n}\to G_{m,n}} \tag{3.5}$$

$$\mathrm{bel}^{(-)}_{F_m} \propto \prod_{n=1}^{N} m^{(-)}_{G_{m,n}\to F_m} \tag{3.6}$$

Once the root D receives all the incoming messages, it can update its belief by normalizing the product of these messages as well as the 'prior' distribution:

$$\mathrm{bel}_D = p(d \mid \mathcal{D}) \propto \theta_{D=d} \prod_m m^{(-)}_{F_m\to D} \tag{3.7}$$

Then D sends top-down messages to its children (with superscript $(+)$):

$$m^{(+)}_{D\to F_m} \propto \frac{\mathrm{bel}_D}{m^{(-)}_{F_m\to D}} \tag{3.8}$$

$$m^{(+)}_{F_m\to G_{m,n}} \propto \frac{\mathrm{bel}_{F_m}}{m^{(-)}_{G_{m,n}\to F_m}} \tag{3.9}$$

Here the 'division' operator for distributions is only defined for two distributions of the same set of random variables, and it is defined as element-wise divisions. The conditional distributions

of F_m and $G_{m,n}$ can be updated by

$$\text{bel}_{F_m} = p(f_m \mid \mathcal{D}) \propto \sum_d \theta_{F=f_m|D=d} * m^{(+)}_{D\to F_m} \prod_n m^{(-)}_{G_{m,n}\to F_m} \tag{3.10}$$

$$\text{bel}_{G_{m,n}} = p(g_{m,n} \mid \mathcal{D}) \propto \sum_{f_m} \theta_{G=g_{m,n}|F=f_m,H=h_n} * m^{(+)}_{F_m\to G_{m,n}} m^{(-)}_{Y_{m,n}\to G_{m,n}} \tag{3.11}$$

Details of parameter learning The EM algorithm [45] is used to learn the parameters (Table 3.2) in `xseq`. Given an initial guess of the model parameters (the θs in Table 3.2), the EM algorithm iterates between the E-step (computing the conditional distributions of hidden random variables D_g, $F_{g,m}$, and $G_{g,m,n}$ given the current guess of model parameters and the observations $\mathcal{D}$) and the M-step (update the model parameters) to find a local maximum of the objective likelihood function. To describe the EM algorithm for paramter learning, we first extend the joint distribution in Equation 3.1 to the case with T mutated genes:

$$\prod_{g=1}^{T} \theta_{D=d_g} \prod_{m=1}^{M_g} \theta_{F=f_{g,m}|D=d_g} \prod_{n=1}^{N_{g,m}} p(y_{g,m,n} \mid G_{g,m,n} = g_{g,m,n}) \theta_{G=g_{g,m,n}|F=f_{g,m},H=h_{g,n}} \tag{3.12}$$

The corresponding complete log-likelihood (since we assume the hidden variables are given) function given data $\mathcal{D}$ is

$$\begin{aligned}\log p(\mathcal{D} \mid \boldsymbol{\theta}) = &\sum_{g=1}^{T} \log\left(\theta_{D=d_g}\right) + \sum_{g=1}^{T}\sum_{m=1}^{M_g} \log\left(\theta_{F=f_{g,m}|D=d_g}\right) + \\ &\sum_{g=1}^{T}\sum_{m=1}^{M_g}\sum_{n=1}^{N_{g,m}} \left(\log\left(\theta_{G=g_{g,m,n}|F=f_{g,m},H=h_{g,n}}\right) + \log\left(p(y_{g,m,n} \mid G_{g,m,n} = g_{g,m,n})\right)\right)\end{aligned} \tag{3.13}$$

Here $\boldsymbol{\theta}$ is the vector of model parameters as in Table 3.2. Given the current 'guess' of model parameters $\boldsymbol{\theta}^{old}$, we can compute the conditional distributions of the unknown variables in the `xseq` model. Then we take the expectation of the complete log-likelihood function with respect

to the conditional distribution of the hidden variables to get

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) =& \mathbb{E}\left(\log p(\mathcal{D} \mid \boldsymbol{\theta})\right) \\
=& \sum_{i=0}^{1} \sum_{g=1}^{T} \log(\theta_{D=i}) P(D_g = i | \mathcal{D}, \boldsymbol{\theta}^{old}) + \\
& \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{g=1}^{T} \sum_{m=1}^{M_g} \log(\theta_{F=j|D=i}) P(F_{g,m} = j, D_g = i | \mathcal{D}, \boldsymbol{\theta}^{old}) + \\
& \sum_{j=0}^{1} \sum_{k \in \{\mathcal{L}, \mathcal{N}, \mathcal{G}\}} \sum_{l \in \{\mathcal{L}, \mathcal{G}\}} \sum_{g=1}^{T} \sum_{m=1}^{M_g} \sum_{n=1}^{N_{g,m,n}} \\
& \log(\theta_{G=k|F=j,H=l}) P(G_{g,m,n} = k, F_{g,m} = j, H_{g,n} = l | \mathcal{D}, \boldsymbol{\theta}^{old})
\end{aligned} \tag{3.14}
$$

We can maximize the above log likelihood function by Lagrange multipliers. For example, for the first term of Equation 3.14,

$$
L(\boldsymbol{\theta}_D, \lambda) = \sum_{i=0}^{1} \sum_{g=1}^{T} \log(\theta_{D=i}) P(D_g = i \mid \mathcal{D}, \boldsymbol{\theta}^{old}) + \lambda(\theta_{D=0} + \theta_{D=1} - 1) \tag{3.15}
$$

$$
\begin{cases}
\theta_{D=0} + \theta_{D=1} = 1 \\
\sum_{g=1}^{T} \frac{1}{\theta_{D=0}} P(D_g = 0 \mid \mathcal{D}, \boldsymbol{\theta}^{old}) + \lambda = 0 \\
\sum_{g=1}^{T} \frac{1}{\theta_{D=1}} P(D_g = 1 \mid \mathcal{D}, \boldsymbol{\theta}^{old}) + \lambda = 0
\end{cases} \tag{3.16}
$$

We can easily solve Equation 3.16 to get

$$
\theta_{D=1} = \frac{\sum_{g=1}^{T} P(D_g = 1 \mid \mathcal{D}, \boldsymbol{\theta}^{old})}{\sum_{g=1}^{T} P(D_g = 0 \mid \mathcal{D}, \boldsymbol{\theta}^{old}) + \sum_{g=1}^{T} P(D_g = 1 \mid \mathcal{D}, \boldsymbol{\theta}^{old})} \tag{3.17}
$$

In summary, the expectation step is to compute the conditional distributions of latent variables given current guess of model parameters using the introduced belief propagation algorithm, and

specifically, to compute the following terms:

$$\hat{N}_{D=i} = \sum_{g=1}^{T} P(D_g = i|\mathcal{D}) \tag{3.18}$$

$$\hat{N}_{F=j|D=i} = \sum_{g=1}^{T} \sum_{m=1}^{M_g} P(F_{g,m} = j, D_g = i|\mathcal{D}) \tag{3.19}$$

$$\hat{N}_{G=k|F=j,H=l} = \sum_{g=1}^{T} \sum_{m=1}^{M_g} \sum_{n=1}^{N_{m,g}} P(G_{g,m,n} = k, F_{g,m} = j, H_{g,n} = l|\mathcal{D}) \tag{3.20}$$

By solving a series of constraint optimization problems to maximize Equation 3.14 to get the maximization step equations:

$$\hat{\theta}_{D=i} = \frac{\hat{N}_{D=i}}{\sum_i \hat{N}_{D=i}} \tag{3.21}$$

$$\hat{\theta}_{F=j|D=i} = \frac{\hat{N}_{F=j|D=i}}{\sum_j \hat{N}_{F=j|D=i}} \tag{3.22}$$

$$\hat{\theta}_{G=k|F=0} = \frac{\sum_l \hat{N}_{G=k|F=0,H=l}}{\sum_k \sum_l \hat{N}_{G=k|F=0,H=l}} \tag{3.23}$$

$$\hat{\theta}_{G_k|F_j=1,H_l} = \frac{\hat{N}_{G=k|F=1,H=l}}{\sum_k \hat{N}_{G=k|F=1,H=l}} \tag{3.24}$$

Pseudo-counts are added to both the numerators and denominators of the above maximization steps to do maximum a posteriori estimation of parameters.

Adding constraints in learning `xseq` parameters Parameter learning is a challenging problem in using Bayesian networks to solve real world problems. When the Bayesian networks have multiple hidden nodes, parameter learning is extremely difficult, and learning algorithms can easily get trapped into local maxima. One solution to mitigate these local maximum problems is to add prior distributions for parameters and use maximum a posteriori estimation of parameters. However, we still need to set the hyper-parameters of the prior distributions. We may have very limited knowledge of these parameters. Here in addition to selecting good hyper-parameters for the prior distributions, we also constrain the parameter space in learning `xseq` conditional distributions. More specifically, in trans analysis, we require the parameters of the

`xseq-simple` model:

$$\theta_{G=\mathcal{L}|F} = \theta_{G=\mathcal{G}|F} \tag{3.25}$$

These constraints indicate that the genes connected to a mutated gene g are equally likely to be up-regulated or down-regulated, given the F status of the mutation in g of a patient. For the `xseq` model, we require the following constraints in analyzing the TCGA data:

$$\theta_{G=\mathcal{L}|F=1,H=\mathcal{L}} = \theta_{G=\mathcal{G}|F=1,H=\mathcal{G}} \tag{3.26}$$

$$\theta_{G=\mathcal{G}|F=1,H=\mathcal{L}} = \theta_{G=\mathcal{L}|F=1,H=\mathcal{G}} \tag{3.27}$$

Incorporating network connection weights in `xseq` inference Our inference graph is a directed weighted graph. We can easily incorporate the weight between two genes/proteins into `xseq` inference. As before, we consider a simple case with just one mutated gene g. The conditional distribution of D given gene expression Y (by summarizing out F, G, and H) can be written as:

$$\begin{aligned}
&P(D = d \mid Y_{1,1} = y_{1,1}, \ldots, Y_{1,N} = y_{1,N}, \ldots, Y_{M,1} = y_{M,1}, \ldots, Y_{M,N} = y_{M,N}) \\
\propto &P(D = d, Y_{1,1} = y_{1,1}, \ldots, Y_{1,N} = y_{1,N}, \ldots, Y_{M,1} = y_{M,1}, \ldots, Y_{M,N} = y_{M,N}) \\
= &\sum_{f_m, g_{m,n}, h_n} \theta_{D=d} \prod_{m=1}^{M} \theta_{F=f_m|D=d} \prod_{n=1}^{N} p(y_{m,n} \mid G = g_{m,n}) \theta_{G=g_{m,n}|F=f_m,H=h_n}
\end{aligned} \tag{3.28}$$

The condotional distribution $p(d \mid \mathcal{D})$ of observing expression $y_{m,n}, m \in 1, \ldots, M, n \in 1, \ldots, N$ is determined by each factor (or "bucket"), e.g., $p(y_{m,n} \mid G_{m,n})$. Each factor can be considered as a feature, and the features that are most relevant should play a larger role in computing the conditional probabilities. Assuming that the connection strength between gene g and the nth gene connected to g is $w_{g,n}$, then we can replace the factor $p(y_{m,n} \mid G = g_{m,n})$ by $p(y_{m,n} \mid G = g_{m,n})^{w_{g,n}}$ in Equation 3.28.

Let's look at how the weight influences predictions. When $w_{g,n} = 0$, the factor $p(y_{m,n} \mid G = g_{m,n})^{w_{g,n}}$ has no influence on the conditional probabilities. When $0 < w_{g,n} \leq 1$, the relative values among $p(y_{m,n} \mid G_{m,n} = \mathcal{G})$, $p(y_{m,n} \mid G_{m,n} = \mathcal{N})$, and $p(y_{m,n} \mid G_{m,n} = \mathcal{L})$ will decrease

after weighting, for example,

$$\frac{p(y_{m,n} \mid G_{m,n} = \mathcal{L})}{\sum_k p(y_{m,n} \mid G_{m,n} = k)} \leq \left(\frac{p(y_{m,n} \mid G_{m,n} = \mathcal{L})}{\sum_k p(y_{m,n} \mid G_{m,n} = k)} \right)^{w_{g,n}} \leq 1 \tag{3.29}$$

As the conditional distribution $p(d \mid \mathcal{D})$ should be normalized so they sum to 1 ($\sum_d P(D = d \mid \mathcal{D}) = 1$), the conditional distribution is only sensitive to the relative magnitude among $p(y_{m,n} \mid G_{m,n} \in \{\mathcal{L}, \mathcal{N}, \mathcal{G}\})$. That means, the more discriminative of $p(y_{m,n} \mid G_{m,n} \in \{\mathcal{L}, \mathcal{N}, \mathcal{G}\})$, the more important the nth is gene in predicting the conditional distribution $p(d \mid \mathcal{D})$. Therefore, a small weight for $w_{g,n}$ means that the nth gene is less important.

3.2.3 Conditional distributions of gene expression values

The conditional distributions $p(y \mid G)$[1] are modelled as Student's t-distributions and estimated offline. For example, the conditional distribution of gene g expression distribution is modelled as a Student's t-distribution when gene g is down-regulated:

$$p(y \mid G = \mathcal{L}) = \int \mathcal{N}(y \mid \mu_{\mathcal{L}}, \sigma_{\mathcal{L}}/z) \text{Gamma}(z \mid \frac{\nu}{2}, \frac{\nu}{2}) dz \tag{3.30}$$

where y is the expression level of gene g, $\mu_{\mathcal{L}}$, $\sigma_{\mathcal{L}}$ and ν are the parameters of the Student's t-distribution. As the parameter ν increases, the Student's t distribution approaches a Gaussian distribution $\mathcal{N}(y \mid \mu_{\mathcal{L}}, \sigma_{\mathcal{L}})$. Compared to Gaussian distributions, the Student's t-distributions are more robust to outliers, especially when ν is small. Now the gene expression distribution is a mixture of three Student's t-distribution:

$$p(y) = \sum_{k \in \{\mathcal{L}, \mathcal{N}, \mathcal{G}\}} \omega_k p(y \mid G = k) \tag{3.31}$$

where ω_k is the mixture coefficient of mixture component k. We also use the EM algorithm to uncover the parameters of the Student's t-distributions.

3.2.4 Influence graph

The influence graph can be any such graph encoding gene regulation. The i^{th} vertex of the graph represents gene (protein) g_i and edge $w_{i,j}$ represents the association strength between

[1] For a given $G = g$, the probability density function of the expression of a gene. More formally written as $f_{Y|G}(y|g)$

gene (protein) g_i and g_j. For analyses presented in this thesis, we constructed a combined functional gene association network by merging the STRING v9.1 [64] functional protein association network, the pathway datasets from KEGG [99], WikiPathway [160] and BioCyc [100], and transcription factors-targets networks. The pathways have already been integrated into the IntPath database [240]. The transcription factors-targets network [236] is download through the transcription factor encyclopedia web API. The ENCODE [72] transcription factor 'proximal' and 'distal' networks are also included in the combined network (download from `http://encodenets.gersteinlab.org/`). The majority of these interactions are transcription factor–target gene interactions (about 1% between transcription factor interactions in the 'proximal' network [72]).

For each dataset, we construct a weighted network. The weight of an interaction represents our prior confidence of the interaction. For the datasets that do not provide weights for interactions, a default weight of 0.8 is used. The STRING protein-protein interaction network is already a weighted network so we use their provided weights. To merge these networks, for a specific interaction, we take the largest weight for this interaction across different networks. We then only keep the interactions with at least median confidence (threshold of 0.4, the default threshold suggested for the STRING database). In this combined network, $17,258$ genes (proteins) connect to $19,070$ genes (proteins) through $898,032$ interactions. This network is almost weakly connected (22 genes do not connect to rest genes.)

Then for each mutated gene, we test whether the genes connected to it are differentially expressed with adjusted p-value (BH method) threshold of 0.05. If there exist differentially expressed genes, we only keep these genes and set their connection weights to 1. In addition, if a gene is not differently expressed in a specific tumour type but differently expressed in other tumour types based on Fishers' combined p-value FDR ≤ 0.05, we also set their connection weights to 1. If no differentially expressed genes exist for a given mutated gene, we use the network from the original weighted network.

3.2.5 Collecting *bona fide* driver genes

We collected the genes from the manually curated, and widely used Cancer Gene Census (CGC) database [70] and two major recent review papers [116, 213] as our reference *bona fide* cancer driver genes. Specifically, we collected 519 genes from CGC [70], 125 genes predicted by the "20/20 rule" [213] (more details below), 66 recently discovered frequently mutated genes collected

in the Supplementary Table 4 of Lawrence *et al* [116], and 33 genes predicted by MutSig and have strong and consistent connections to cancer [116]. In summary, these datasets include 603 unique genes in total. The 127 significantly mutated genes predicted by the MuSiC suite [98] are not counted because we use this dataset for several comparisons. Notice that in our analysis, we use the samples with expression, copy number alterations, and mutations. Therefore `xseq` analyzes a subset of mutations used by the MuSiC suite.

3.2.6 Modelling loss-of-function mutations and hotspot mutations

The mutation patterns of most known tumour suppressor genes and oncogenes are highly characteristic and non-random [213]. In a recent review [213], a "20/20 rule" is used to identify driver genes: for oncogenes, at least 20% of all the mutations are required to be hotspot missense mutations or in-frame indels; for tumour suppressor genes, at least 20% of all the mutations are required to be loss-of-function mutations.

Here we extend the "20/20 rule" by using mixture-of-binomial modelling of loss-of-function mutations and hotspot mutations. We analyze oncogenes and tumour suppressor genes separately. When predicting oncogenes, we first count the number of hotspot mutations $n_{g,rec}$ (recurrent missense mutations and in-frame indels) in gene g, and the total number of mutations N_g in gene g. Then we model the mutation count distribution as a mixture of two binomial distributions: one component for oncogenes and the other component for non-oncogenes:

$$P(n_{g,rec} \mid N_g) = \omega_1 \text{Binomial}(n_{g,rec} \mid N_g, p_1) + \omega_2 \text{Binomial}(n_{g,rec} \mid N_g, p_2)$$
$$P(\text{OCG}) = \omega_1 \text{Binomial}(n_{g,rec} \mid N_g, p_1) / P(n_{g,rec} \mid N_g)$$

Then $P(\text{OCG})$ (or $P(\text{OCG} = 1)$) is defined as the probability of $n_{g,rec}$ in the mixture component with higher success rate, namely p_1 here. The mixture parameters $\boldsymbol{\omega} = (\omega_1, \omega_2)$ and success rates $\boldsymbol{p} = (p_1, p_2)$ are estimated by the EM algorithm.

Similarly when predicting tumour suppressor genes, we first count the number of loss-of-function mutations $n_{g,loss}$ in gene g, and N_g. Then we model the count distribution as a mixture of two binomial distributions: one component representing tumour suppressor genes

and the other representing non-tumour suppressor genes:

$$P(n_{g,loss} \mid N_g) = \omega_1 \text{Binomial}(n_{g,loss} \mid N_g, p_1) + \omega_2 \text{Binomial}(n_{g,loss} \mid N_g, p_2)$$
$$P(\text{TSG}) = \omega_1 \text{Binomial}(n_{g,loss} \mid N_g, p_1) / P(n_{g,loss} \mid N_g)$$

$P(\text{TSG})$ (or $P(\text{TSG} = 1)$) is defined as the probability of $n_{g,loss}$ in the mixture component with higher success rate (p_1 here). Again, the mixture parameters $\boldsymbol{\omega}$ and success rates $\boldsymbol{p}$ are estimated by the EM algorithm.

The mixture-of-binomial approach can be considered as a generation of the "20/20 rule" because of its ability to estimate the parameters from data, and to account for the total number of mutations to compute probabilities of genes to be oncogenes or tumour suppressor genes. To make the estimated parameters more accurate, we also added extra genome wide screen data from COSMIC v64 [62], downloaded from syn1855816, and the Pan-Cancer data downloaded from syn1710680. Fig. 3.3 shows the Binomial mixture modelling of oncogenes and tumour suppressor genes for all the genome-wide screen somatic mutation data. We used a threshold of 0.2 for $P(\text{TSG})$ and $P(\text{OCG})$ to call genes with tumour suppressor gene properties and oncogene properties, respectively.

3.2.7 Expression information in predicting driver mutations

Some mutated genes are not expressed in cancer cells, and therefore the mutations in these genes are less likely to be pathogenic. Currently the expression information has not yet been fully explored for the identification of driver mutations [40], and only a few methods take expression information into account to assess the background mutation rates [117].

We adopt a mixture modelling approach to predict whether a gene is "highly-expressed" in a tumour type. We first $\log_2$ transform the tumour gene RSEM abundance estimation values. To prevent taking the $\log_2$ of 0, we remove the gene expression values if they are less than 0 before $\log_2$-transformation. We then compute the 90^{th} percentile of the expression of a gene across patients in a tumour type to represent the overall expression of that gene. Here we use the 90^{th} percentile instead of median considering the gene dosage effects of copy number deletions on expression in cancer. It is unlikely that a gene is deleted in 90% of all the analyzed samples (e.g., for the Pan-Cancer datasets, the mostly frequently homozygous deleted gene is *CDKN2A*, which is deleted in 57% of patients in GBM). If we also consider heterozygous deletions, then

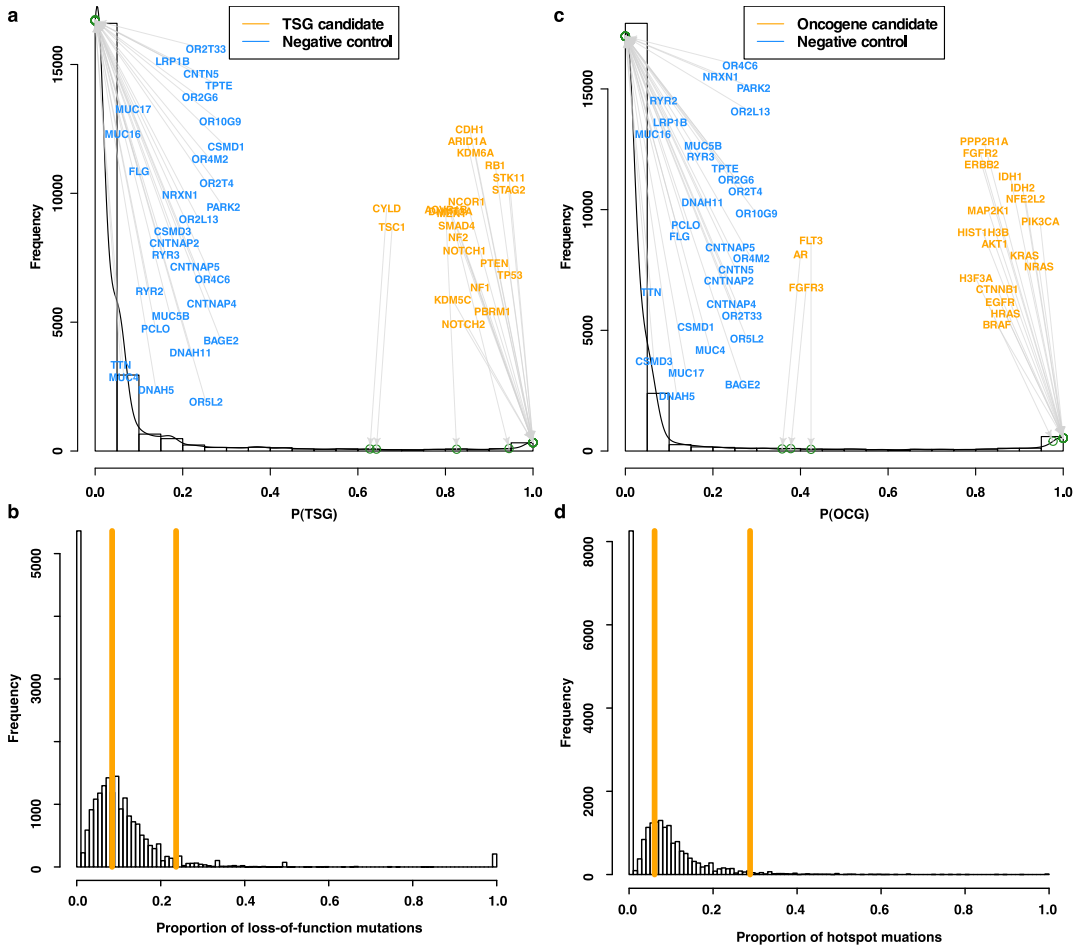


Figure 3.3: Mixture-of-Binomial modelling of loss-of-function mutations and hotspot mutations. (a) $P(\text{TSG} = 1)$ (tumour suppressor gene) histogram. We labeled the $P(\text{TSG})$ of the 30 negative control genes (blue colour, Results section), as well as 21/23 predicted cis-effect tumour suppressor genes with high $P(\text{TSG})$ (orange colour, Results section). (b) The proportion of loss-of-function mutations distribution. The two vertical lines represent the estimated success rates of the two binomial distributions. (c) $P(\text{OCG} = 1)$ (oncogene) histogram. We also labeled the 30 negative control genes (blue colour), as well as 20 selected oncogenes for illustration purpose (orange colour). (d) The proportion of recurrent mutation distribution. The two vertical lines represent the estimated success rates of the two binomial distributions.

the most frequently deleted gene is *EBF3*, which is deleted in 90% of patients in GBM. The 90^{th} percentile expression of a gene may overestimate the expression level of the gene in the studied tumour type (since we are more concerned about losing important genes). Next, we model the 90^{th} percentile expression of genes as a mixture of two Gaussian distributions: one component representing "highly-expressed" and the other component representing "lowly-expressed". A gene is considered to be "highly-expressed" if its probability in the "highly-expressed" group is

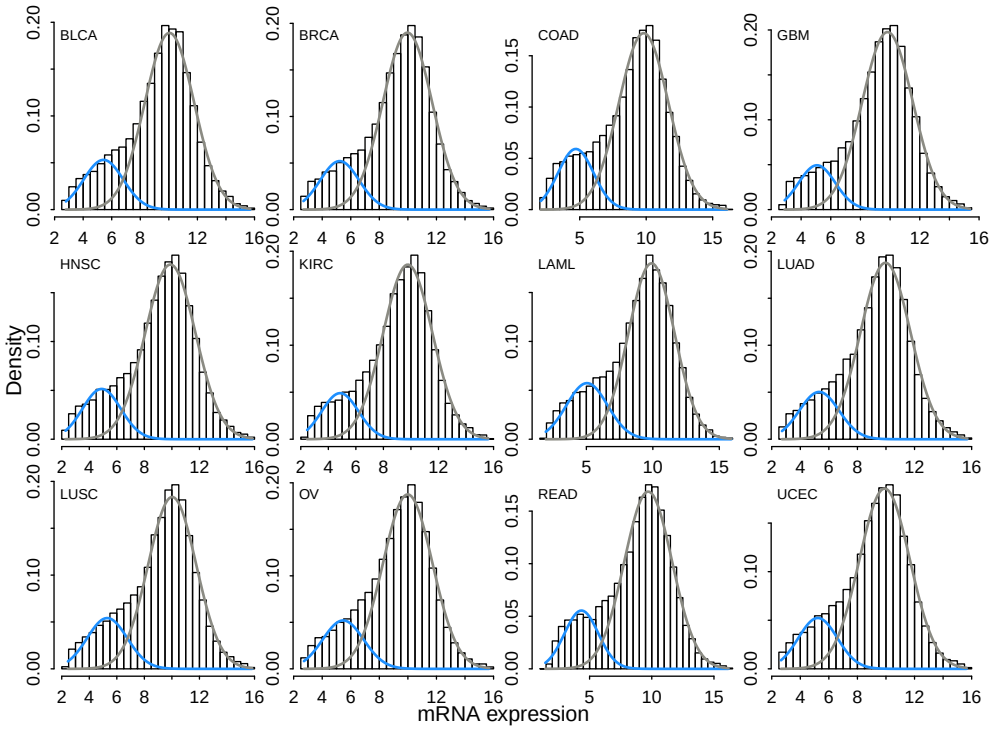


Figure 3.4: Detecting highly-expressed genes based on mixture-of-Gaussian distributions. For each plot, the left blue curve is the Gaussian fit to the "lowly-expressed" genes, and the right grey curve is the Gaussian fit to the "highly-expressed" genes.

greater or equal to 0.8. In the presence of outliers (some genes are expressed at extremely high or low levels), we first remove outliers based on the boxplot rule, and then fit the data. We assign a probability of 1 to the highly-expressed outliers, and a probability of 0 to the lowly-expressed outliers. Fig. 3.4 shows Gaussian mixture modelling of expression across the 12 cancer types.

3.2.8 Compensating for the cis-effects of copy number alterations

Before analyzing the trans-effects of somatic mutations, we first remove the cis-effects of copy number alterations on expression; copy number alterations are common in cancer and the majority have cis-effects on expression. Numerous studies have done integrative analysis of copy number and gene-expression data [10, 74, 92, 145]. Here we use the Gaussian process (GP) regression to model the expression y_i of a gene in a patient i, as a function of its copy number log2 value x_i. GP regression is flexible to add extra variables such as DNA methylation data as independent variables if necessary, and can capture nonlinear relationships between copy number alterations and expression.

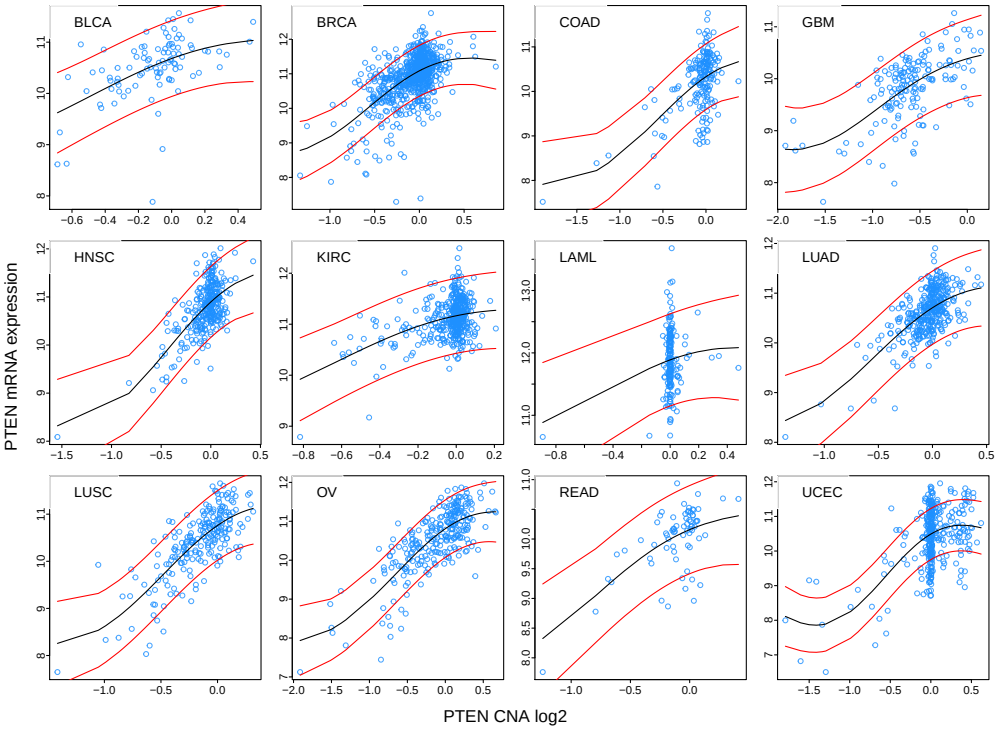


Figure 3.5: Scatter plots of *PTEN* copy number alterations and expression across 12 cancer types. The black curves are the regression curves estimated by Gaussian process regression, and the region between the two red curves for each plot is the 95% confidence interval.

GP regression models the joint distribution of y_i as a joint Gaussian distribution. The covariance matrix is constructed based on the given copy number data, $\text{cov}(x_i, x_j) = k(x_i, x_j)$, where k is the squared exponential kernel function. The hyper-parameters of the kernel function are computed by optimizing the log-marginal likelihood function using scaled conjugate gradient algorithms. To remove the cis-effects of copy number alterations, we subtract the regression values from the original expression values to get the residuals that are considered to be regulated by trans-effect mutations. Fig. 3.5 shows the scatter plots of copy number alteration and expression values for *PTEN* across the 12 cancer types. The GP regression lines and the 95% confidence intervals of the regression lines are overlaid on the scatter plots. Fig. 3.6 shows the scatter plots of *PTEN* expression across cancer types after removing the cis-effects of copy number alterations on expression.

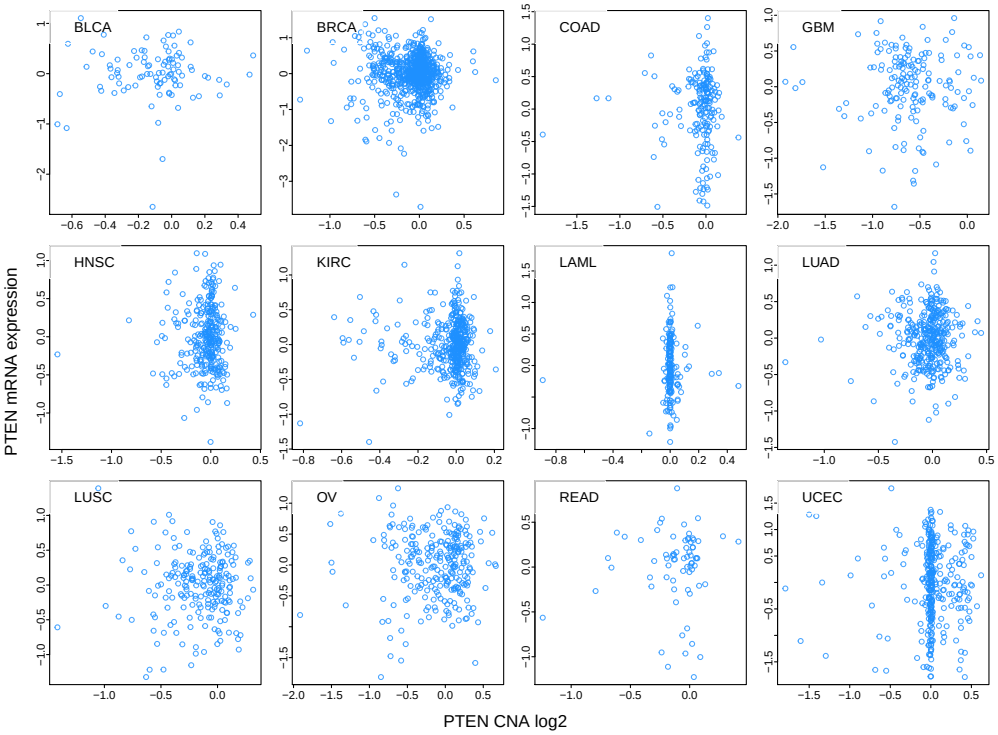


Figure 3.6: Scatter plots of *PTEN* copy number alterations and expression after cis-effect removing. The cis-effects of copy number alterations on gene expression have been removed by subtracting the regression values from the original expression values.

3.3 Results

3.3.1 Datasets

We used somatic point mutation, copy number alteration, and gene expression data in 12 cancer types, from the TCGA Pan-Cancer project [223] (Table 3.3). A total of 2,786 patients with all three types of data were included in our analyses. For trans-analysis, focal copy number homozygous deletions and amplifications (with four or more copies) predicted by GISTIC [142, 237] were also encoded as inputs to `xseq`. We did not analyze the cis-effects of copy number alterations since the majority of them have cis-effects on gene expression [199].

3.3.2 Computational benchmarking and validation of `xseq`

Simulation analysis We examined the theoretical performance of `xseq` via simulation and permutation analyses. To investigate the performance of `xseq` under different noise levels, we simulated data (Fig. 3.7) from nine hyper-parameter sets, and ten independent realizations

Table 3.3: List of the twelve cancer types analyzed

Data	Mutation	RNA-Seq	SNP6.0	Overlap
BLCA	99	96	125	94
BRCA	772	822	879	743
COAD	155	192	414	149
GBM	291	167	576	144
HNSC	306	303	306	295
KIRC	417	428	452	390
LAML	196	173	197	167
LUAD	230	355	358	169
LUSC	178	220	342	177
OV	316	266	581	159
READ	69	71	163	65
UCEC	248	333	493	235

The numbers are the sample counts.
BLCA, Bladder urothelial carcinoma; BRCA, Breast invasive carcinoma; COAD, Colon adenocarcinoma; GBM, Glioblastoma multiforme; HNSC, Head and neck squamous cell carcinoma; KIRC, Kidney renal clear cell carcinoma; LAML, Acute myeloid leukemia, also denoted as AML; LUAD, Lung adenocarcinoma; LUSC, Lung squamous cell carcinoma; OV, Ovarian serous cystadenocarcinoma; READ, Rectum adenocarcinoma; UCEC, Uterine corpus endometrioid carcinoma
Totally 563,024 somatic mutations in the overlapped samples (363,676 missense mutations, 132,981 synonymous mutations, 33,838 nonsense mutations, 13,260 frameshift indels, 6,952 non-coding RNA mutations, 8,699 splice-site mutations, 3,141 in-frame indels, and 477 stop-gain mutations).
In trans-analysis, we added the 37,308 homozygous deletions in 2,084 genes (focal copy number deletion peaks), and 69,643 amplifications in 960 genes (focal copy number amplification peaks).

of data for each hyper-parameter set. We used `xseq` (Fig. 3.7) to generate synthetic data. The number of mutated genes T was fixed to 300, and the number of patients was fixed to 200. The number of accumulated mutations for gene g was sampled from a shifted geometric distributions: $P(M_g = m) = 0.1^{m-15} \times 0.9$ with $m \geq 15$. The number of genes connected to g was also sampled from a shifted geometric distribution: $P(N_g = n) = 0.1^{n-10} \times 0.9$ with $n \geq 10$. The hyper-parameters of the beta distribution for θ_D were set to $(70, 30)$, which means that around 30% of the genes accumulated mutations impacting expression. The hype-parameters of the beta distribution for $\theta_{F|D=0}$ were set to $(105, 5)$, and $(10, 100)$ for $\theta_{F|D=1}$. The Dirichlet hyper-parameters for $\theta_{G|F=0}$ were set to $(10, 180, 10)$. The gene expression data were sampled from Gaussian distributions. The variances of the Gaussian distributions were sampled from an inverse-gamma distribution: $\mathrm{IG}(\sigma^2 \mid \nu, \sigma_0^2)$. We fixed the hyper-parameter $\nu = 3$, $\sigma_0^2 = 0.12$.

We first analyzed the influence of the degree of gene regulation (up- or down-regulation) correlated with mutations. For high degree of gene regulation correlated with mutations, we set the Dirichlet parameters to $(143, 55, 2)$ for $\theta_{G|F=1,H=\mathcal{L}}$. For $\theta_{G|F=1,H=\mathcal{G}}$, the Dirichlet parameters were set to $(2, 55, 143)$ (Fig. 3.8, column 1). For moderate gene regulation correlated with

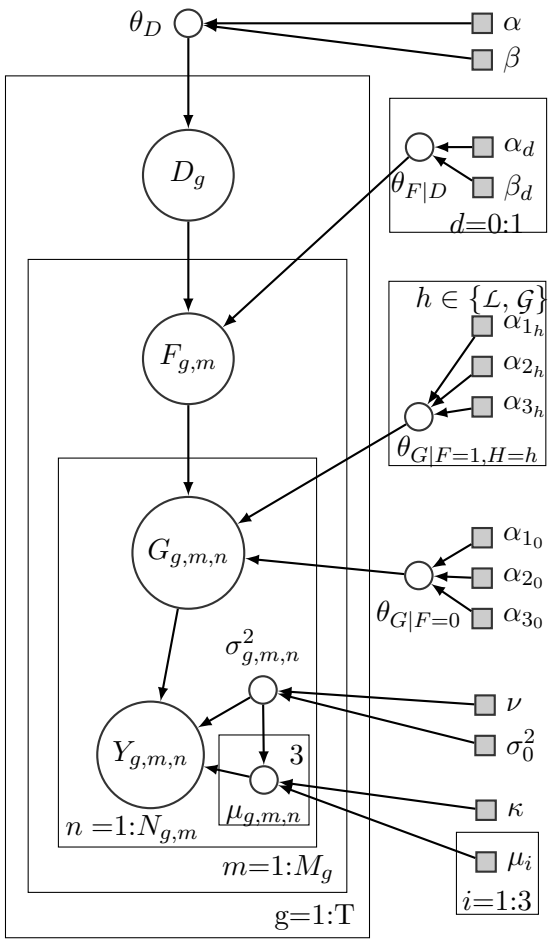


Figure 3.7: The `xseq` model used to sample T mutated genes. The gth gene has M_g mutations. Of the mth patient with a mutation in gene g, $N_{g,m}$ genes connected to g. Here the shaded nodes represent observed variables and the unshaded nodes represent hidden variables. Circles represent random variables and shaded squares represent hyper-parameters.

mutations, we set the Dirichlet parameters to $(93, 105, 2)$ for $\theta_{G|F=1,H=\mathcal{L}}$. For $\theta_{G|F=1,H=\mathcal{G}}$, the Dirichlet parameters were set to $(2, 105, 93)$ (Fig. 3.8, column 2). Finally, for low level gene regulation correlated with mutations, we set the Dirichlet parameters to $(43, 155, 2)$ for $\theta_{G|F=1,H=\mathcal{L}}$. For $\theta_{G|F=1,H=\mathcal{G}}$, the Dirichlet parameters were set to $(2, 155, 43)$ (Fig. 3.8, column 3). As expected, as we decreased the degree of gene regulation correlated with mutations, the `xseq` prediction performance declined, especially in recovering the specific mutation variables F (the ROC curves in Fig. 3.8).

We then analyzed the influence of the discrimination of expression of down-regulation, neutral, and up-regulation on predictions. For high discriminative down-regulation, neutral and up-regulation expression distributions, we set the mean parameters μ_i of Gaussian distributions

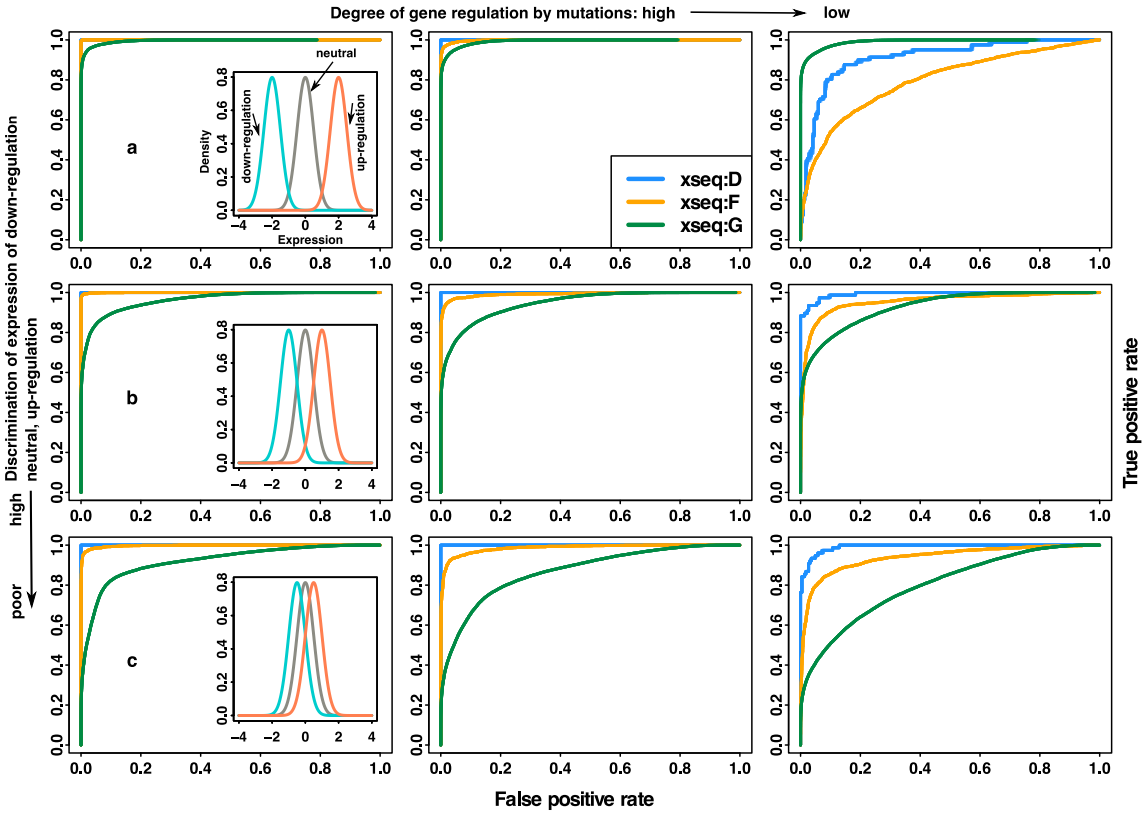


Figure 3.8: Theoretical performance of `xseq` on simulated datasets. Each plot depicts a receiver operating characteristic (ROC) curve, which displays the true positive rate as a function of false positive rate. (a) The expression of genes which are down-regulated, neutral, and up-regulated is highly discriminative (first row), (b) moderately discriminative (second row), and (c) poorly discriminative (third row, see the enclosed figures, where cyan is down-regulation, grey is neutral, and red is up-regulation, respectively). The ROC curves in the first column, second column, and the third column were computed when the degree of dysregulation of the expression of connected genes by mutations was high, moderate, and low, respectively.

to (-2, 0, 2) (Fig. 3.8, the first row). For moderate discriminative down-regulation, neutral, and up-regulation expression distributions, we set the mean parameters of Gaussian distributions to (-1, 0, 1) (Fig. 3.8, the second row). For poor discriminative down-regulation, neutral, and up-regulation expression distributions, we set the mean parameters of Gaussian distributions to (-0.5, 0, 0.5) (Fig. 3.8, the third row). The decreasing in the discrimination of down-regulation, neutral, and up-regulation expression distributions mostly declined the performance of recovering the gene regulation variables G. We also drew the trace plots for the most challenging case by given the true H (Fig. 3.9) and estimating H offline (Fig. 3.10). When H was given, the EM algorithm found the parameters that were very close to the true values. While H was unknown, the estimated parameters could have high bias because the three mixture models were highly

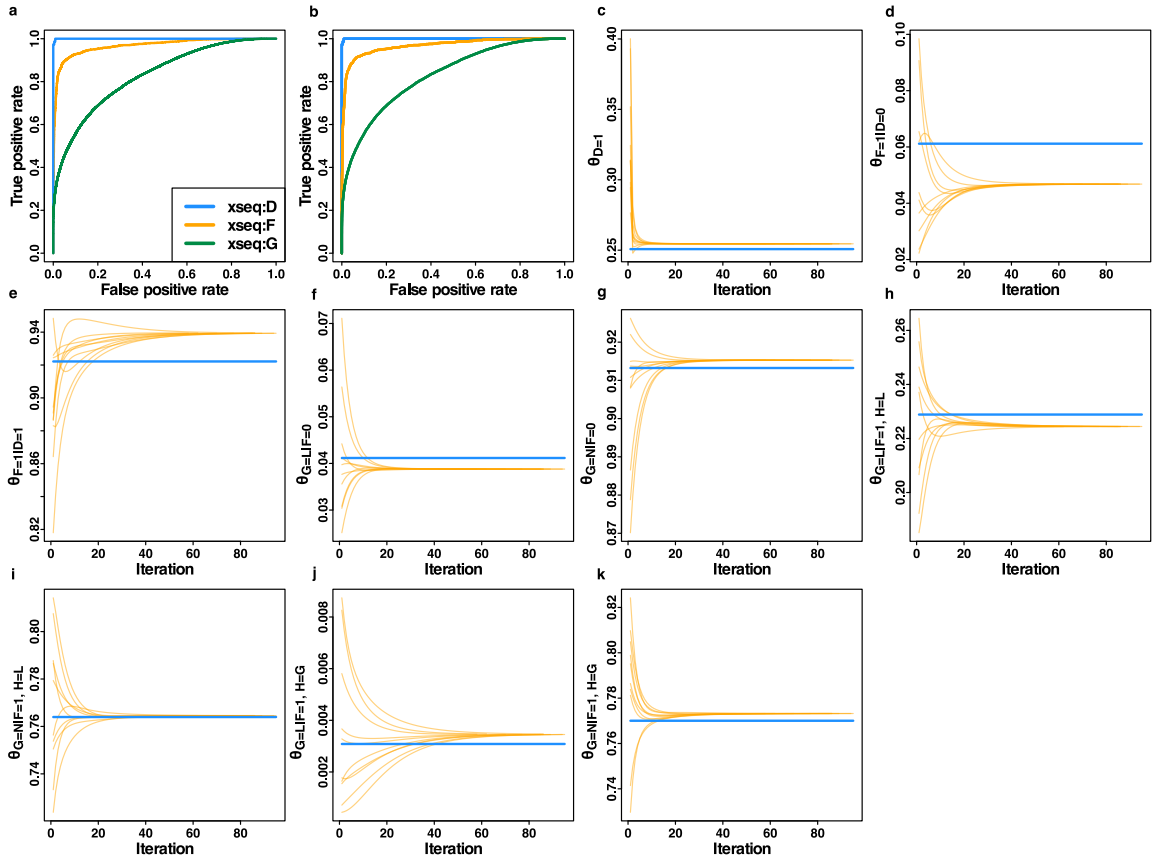


Figure 3.9: `xseq` parameter trace plots during EM-iterations for the most challenging case when H is known. (a) The ROC curves of the predictions based on the true parameters. (b) The ROC curves of `xseq` predictions based on the parameters learned from the EM algorithm. (c)-(k) The trace plots of `xseq` parameters during EM iterations.

overlap (Fig. 3.8(c), enclosed figures).

We next used the `xseq-simple` model to analyze the simulation datasets (Fig. 3.11). Generally speaking, the `xseq-simple` performed quite well. Compared to `xseq` predictions given the true values of H variables, we could see a slightly drop in performance, especially for the most challenging case (Fig. 3.8 and Fig. 3.11, bottom-right ROC curves). Because `xseq-simple` does not model the H variables, it substituted the two conditional distributions $\theta_{G=\mathcal{L}|F=1,H=\mathcal{L}}$ and $\theta_{G=\mathcal{L}|F=1,H=\mathcal{G}}$ with a single conditional distribution $\theta_{G=\mathcal{L}|F=1}$. We could see this phenomenon from the parameter trace plots during EM algorithm iterations (Fig. 3.12).

Permutation analysis We performed several permutations to investigate the influence of each component of `xseq` on the final predictions. First, we switched the patient names within

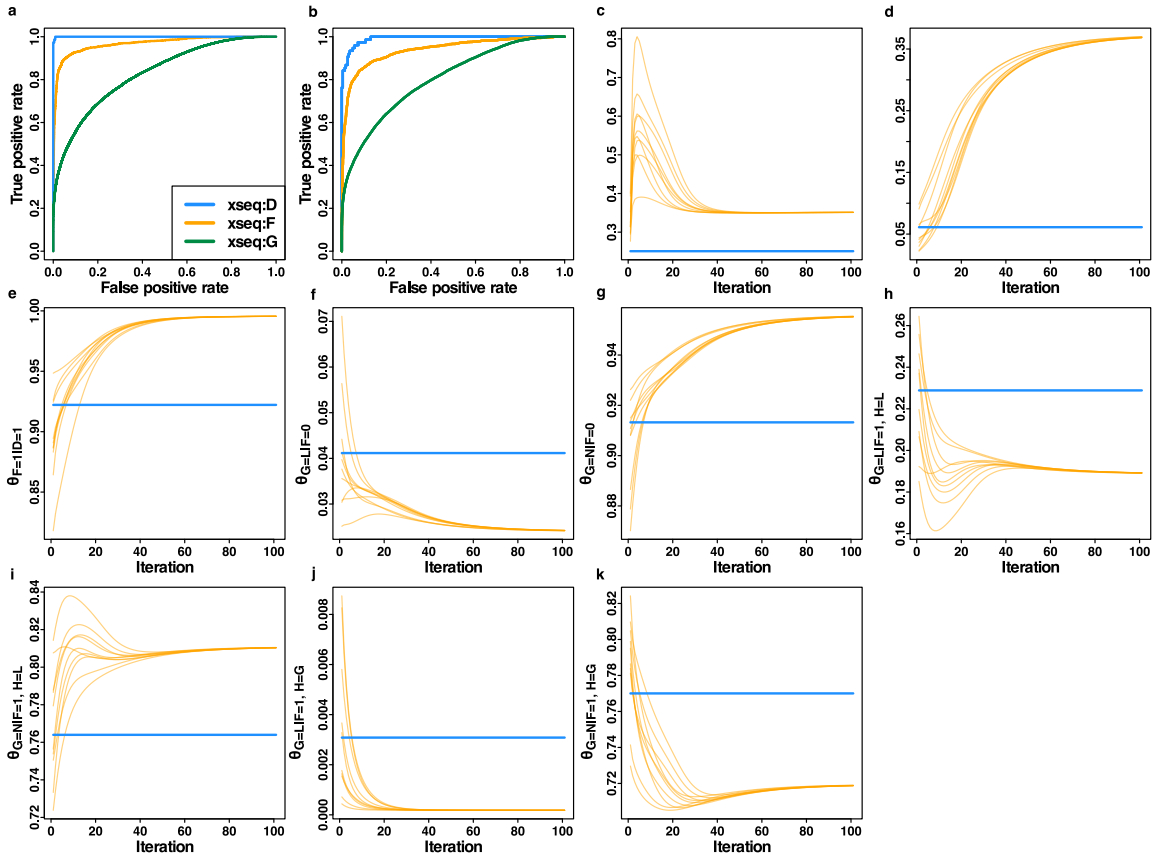


Figure 3.10: `xseq` parameter trace plots during EM-iterations for the most challenging case (H is estimated offline). (a) The ROC curves of the predictions based on the true parameters. (b) The ROC curves of `xseq` predictions based on the parameters learned from the EM algorithm. (c)-(k) The trace plots of `xseq` parameters during EM iterations. The estimated parameters could have high bias because the three mixture models were highly overlap (Fig. 3.8(c), enclosed figures).

the mutation matrix (Fig. 3.13(a), permute sample). Even after permutation, some mutations were still predicted to have high probabilities $P(F)$. To help explain this phenomenon, we generated an expression heatmap (Fig. 3.13(b)), which showed the expression of genes connected to *RUNX1*. We can see that some patients without *RUNX1* mutations still showed similar expression pattern to those with *RUNX1* mutations. This "phenocopy" [222] effect could result in some patients without mutations but high predicted probabilities $P(F)$. Phenocopying may be a common event in cancer because of DNA methylation and other epigenetic alterations, and it may suggest novel treatment opportunities. For example, there is increasing evidence that treating of patients based on phenotypes (expression) instead of genotypes (DNA mutations)

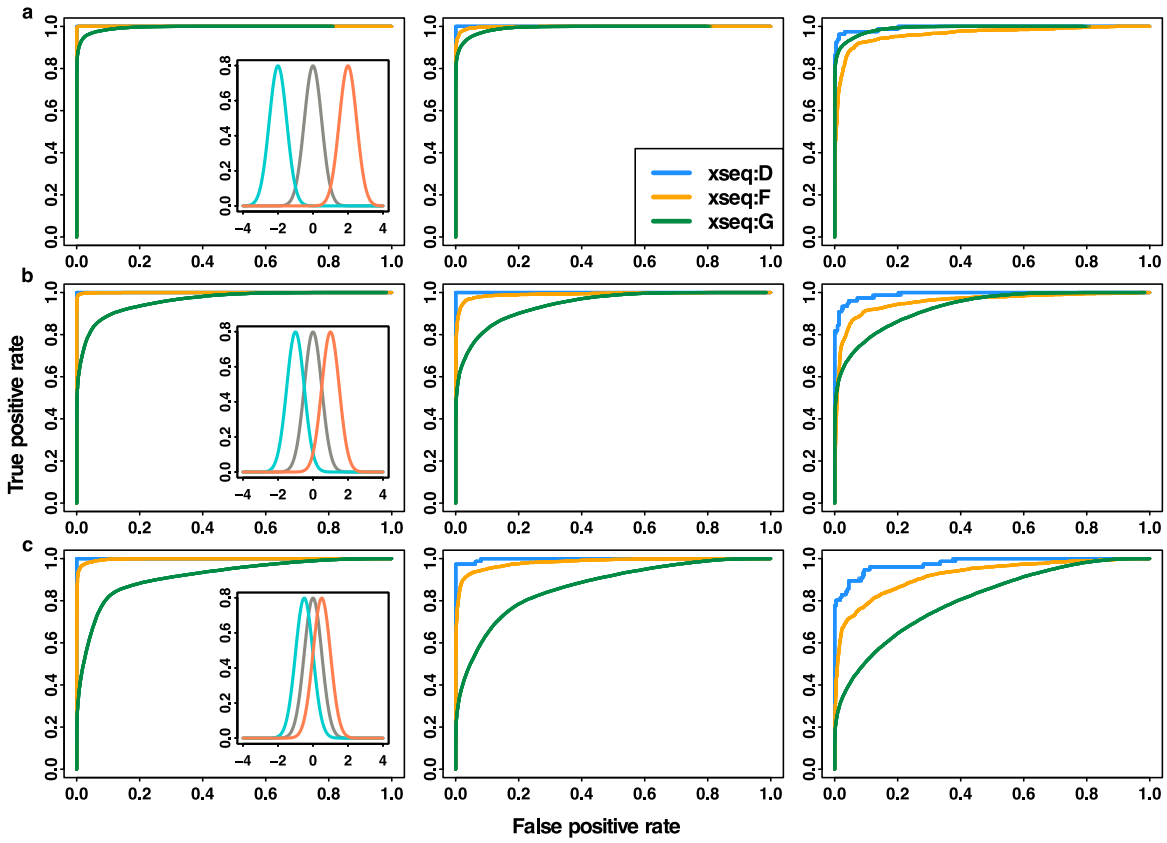


Figure 3.11: The `xseq-simple` model prediction ROC curves from different simulated datasets. (a) The expression of genes which are down-regulated, neutral, and up-regulated is highly discriminative (first row), (b) moderately discriminative (second row), and (c) poorly discriminative (third row, see the enclosed figures, where blue is down-regulation, grey is neutral, and red is up-regulation, respectively). The ROC curves in the first column, second column, and the third column were computed when the degree of dysregulation of the expression of connected genes by mutations was high, moderate, and low, respectively.

produces better outcomes in some types of cancer [231]. We also switched the gene names within the mutation matrix (Fig. 3.13(a), permute gene), and the results showed similar performance to those by switching patients.

Next, we randomly drew the same number of connected genes as given by the combined network (Fig. 3.13(a), permute network). Because the gene-regulation information is sparse and some master regulators can influence the expression of huge number of genes, the model may still predict a few mutations with high probability $P(F)$ because these "randomly-drawn" genes might be truly regulated by the mutated genes. Finally, if both the mutation matrix and the network were shuffled, there were very few predicted high probability mutations (dashed red

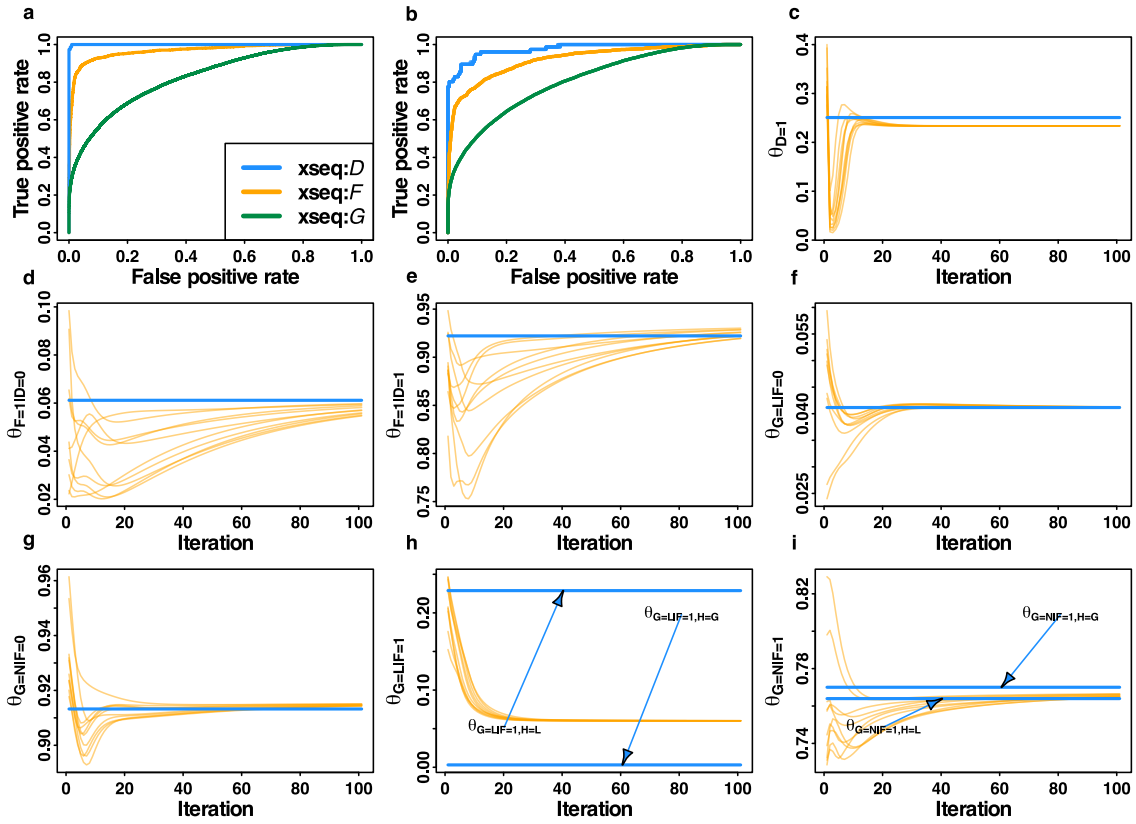


Figure 3.12: `xseq-simple` parameter trace plots during EM-iterations for the most challenging case. (a) The ROC curves of the predictions based on the true parameters. (b) The ROC curves of `xseq-simple` predictions based on the parameters learned from the EM algorithm. (c)-(i) The trace plots of `xseq-simple` parameters during EM iterations.

curves in Fig. 3.13(a), permute all). We performed the same processing steps after permutations thus minimizing the possibility of introducing bias. In addition, we kept the expression matrix the same in all permutation analyses.

Cross-validation analysis We executed a cross-validation analysis by splitting each TCGA dataset into approximately equally sized discovery and validation datasets. We trained a model on the discovery dataset, and used the trained model to predict the validation dataset, with ten repeats for each tumour type. We defined the validation rate as the proportion of high probability predicted genes ($P(D) \geq 0.8$, see next section on picking the threshold) in the training data also predicted to have high probabilities in the validation data. For *bona fide* cancer genes, the median validation rate was 0.625 across all the 12 tumour types. For all of the predictions from

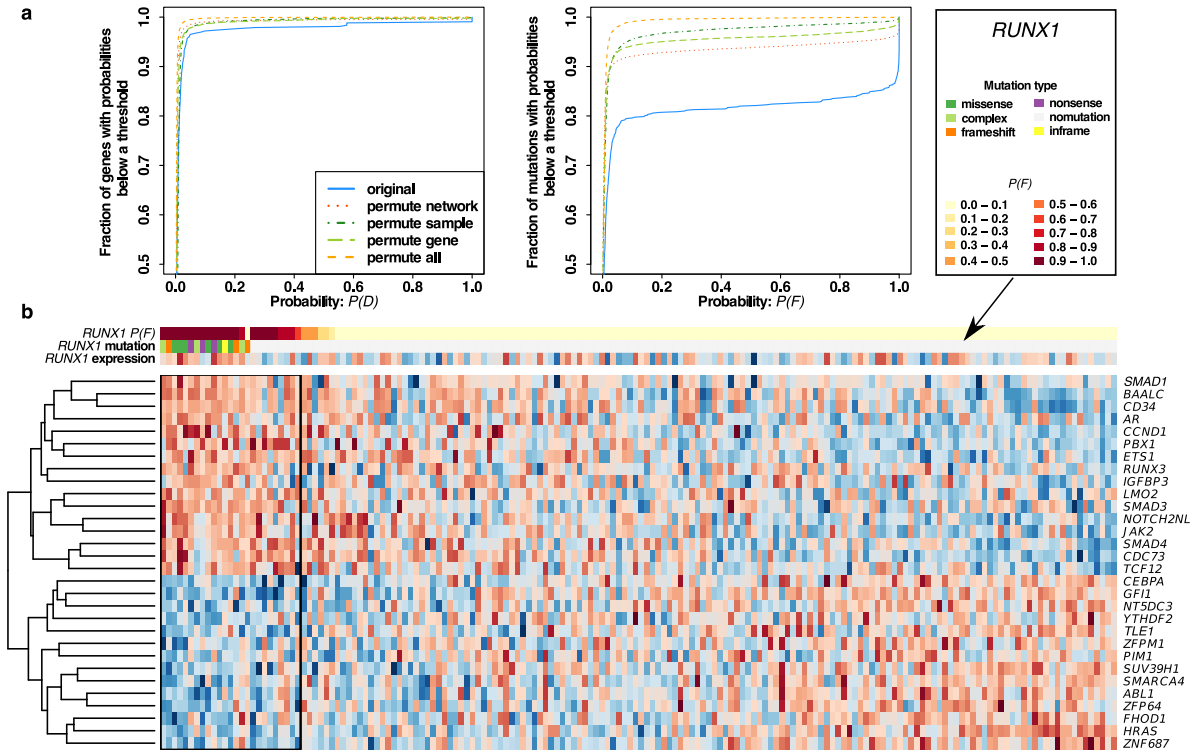


Figure 3.13: Permutation analysis of the TCGA acute myeloid leukemia datasets. (a) Left panel shows the empirical distribution functions of $P(D)$, and the right panel shows the empirical distribution functions of $P(F)$ estimated from different permuted datasets. (b) Heatmap shows the expression of genes connected to *RUNX1*: red represents high expression and blue represents low expression. Here columns represent patients and rows represent genes. For the patients without *RUNX1* mutations, we "assume" the mutations still exist and estimate the probabilities of individual mutations $P(F)$. The mutation type 'complex' of a gene in a patient represents the gene harbouring multiple types of mutations in the patient.

the discovery data, the median validation rate was 0.492. The validation rate is sensitive to the number of patients (e.g., the median validation rate for the predicted *bona fide* cancer genes in COAD and BRCA was 0 and 0.73, respectively). Restricting analysis to genes with at least five mutations in both the discovery and validation datasets, median validation rates for the *bona fide* cancer genes and all the predicted genes increased to 0.68 and 0.63, respectively.

Tested on independent datasets To examine how the model would translate to independently generated data, we used the METABRIC data [39] to validate the predicted copy number alterations from TCGA in breast cancer. METABRIC copy number alterations [39] were generated with Affymetrix SNP 6.0, however gene expression was generated using Illumina microarrays. We applied the `xseq` model trained on the TCGA breast cancer data to analyze

the METABRIC breast cancer data. This analysis generated 14 genes with high probability ($P(D) \geq 0.8$), representing a strict subset of the 42 genes predicted in the TCGA breast cancer data [47].

Method comparison Finally, we quantitatively benchmarked `xseq` against CONEXIC [2] and `xseq-simple`. We ran CONEXIC algorithm on the TCGA breast cancer copy number and RNA-Seq expression data. We first used the default parameter setting and initialized the clustering using K-means with $K = 50$ clusters. The candidate driver CNAs were the focal copy number deletions and amplifications from the Pan-Cancer analysis [237]. Because of the large number of candidate driver genes (focal copy number alterations in 2891/3044 genes which have Entrez gene symbols), CONEXIC generated 1385 modules, with median module size of six and modulator number of three. These modules selected uniquely 313 modulators, and 21/313 genes were putative cancer driver genes. Another 46/292 genes were directly connected partners of the *bona fide* cancer driver genes.

To compare with CONEXIC, we re-ran `xseq` on the TCGA breast cancer data using only DNA copy number and RNA-Seq expression data. This analysis generated 40 high-probability genes ($P(D) \geq 0.8$), and 13/40 genes were *bona fide* cancer driver genes. Another 13/27 genes were directly connected partners of the *bona fide* cancer driver genes. Ten genes overlapped with CONEXIC predictions: *BYSL, CALML3, CALML5, CCNE1, EGFR, ERBB2, EXOSC4, KPNA2, MRPS28*, and *PPP2R2A*. Therefore, xseq was more specific but potentially less sensitive than CONEXIC in predicting copy number alterations influencing expression.

`xseq` increased sensitivity without loss of specificity of results relative to `xseq-simple`. Application of `xseq-simple` to the Pan-Cancer datasets predicted a strict subset of 106 genes (relative to the 150 genes predicted by `xseq`. An example gene predicted only by `xseq` is *CCNE1* in OV cancer as can seen from Fig. 3.14 and Fig. 3.15. The `xseq-simple` model only picked the *CCNE1* amplifications in 5/30 patients correlated with extreme expression dysregulation ($P(F) \geq 0.5$). By considering the direction of gene regulation, `xseq` picked 19/30 amplification in *CCNE1* ($P(F) \geq 0.5$). In addition, considering the direction of gene-regulation does not sacrifice specificity, e.g., 89/149 predicted genes were either putative cancer genes or directly connected parters of putative cancer genes for the `xseq` model, compared to 65/105 `xseq-simple` predicted genes were either putative cancer genes or directly connected partners of putative cancer genes (p-value = 0.80).

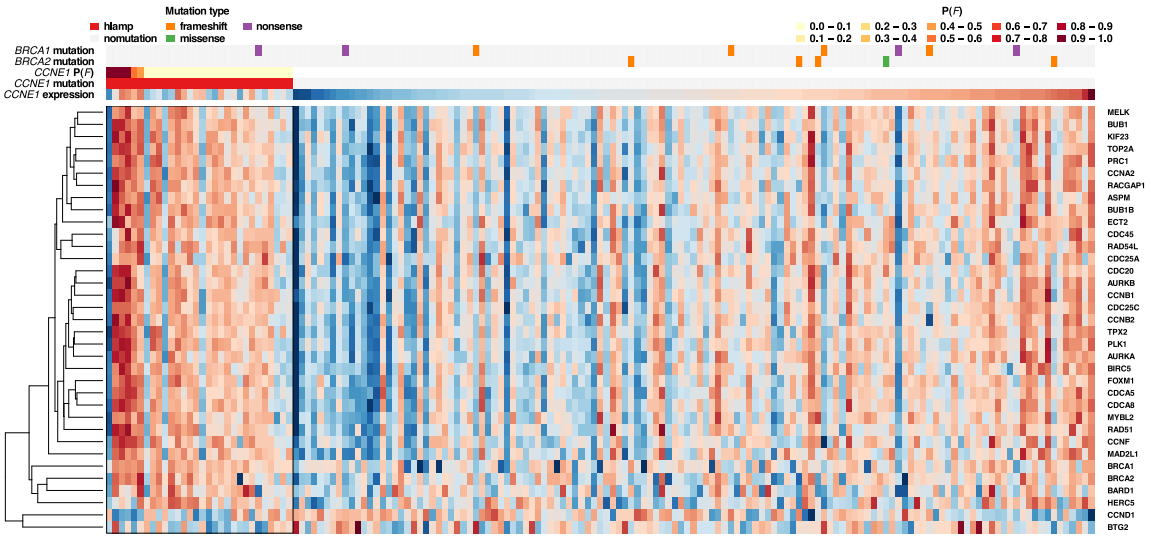


Figure 3.14: *CCNE1* amplifications in high-grade serous ovarian cancer predicted by `xseq-simple`. *CCNE1* amplification probabilities $P(F)$ predicted by the simplified version of `xseq` without considering the directionality of gene regulation.

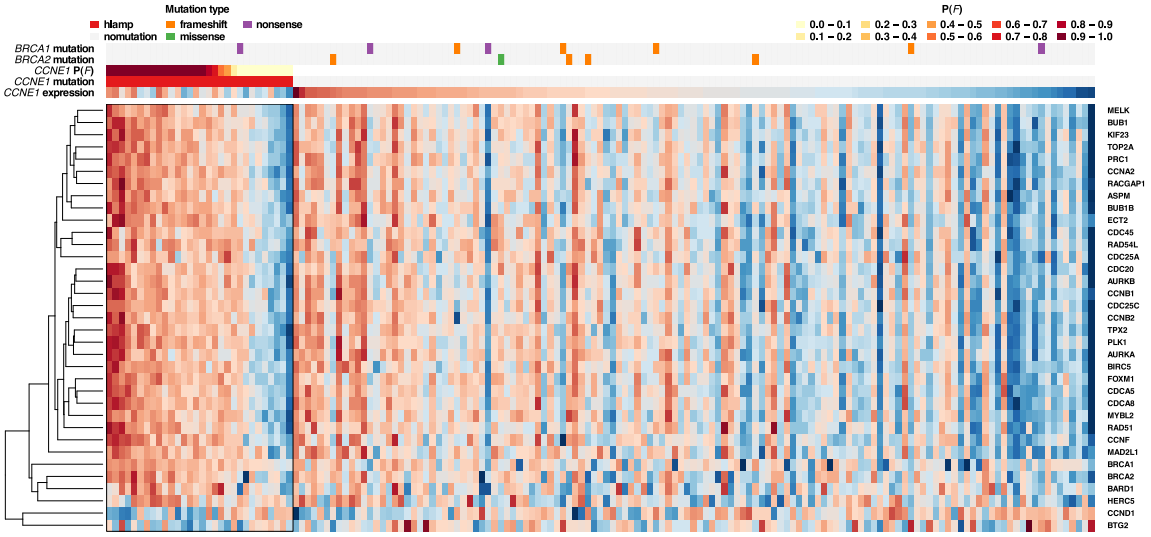


Figure 3.15: *CCNE1* amplifications in high-grade serous ovarian cancer predicted by `xseq`. *CCNE1* amplification probabilities $P(F)$ predicted by `xseq` when the directionality of gene regulation was considered.

3.3.3 Cis-effects loss-of-function mutations across the TCGA data

We began analysis of the TCGA data by focusing on the cis-effect impacts of loss-of-function mutations (frameshift, nonsense, and splice-site mutations) on gene expression, yielding 65 genes across the 12 datasets with $P(D) \geq 0.8$ (Fig. 3.16(a)). (We chose the threshold of 0.8 for $P(D)$ to balance prediction of novel genes with introduction of false positives. Changes to the

results with thresholds in the range 0.75 to 0.85 were minor.) To place these predictions in the context of known cancer genes, we compiled a list of 603 *bona fide* cancer genes from the Cancer Gene Census (CGC) database [70] (Fig. 3.16(a), black coloured genes), Vogelstein *et al* [213], and Lawrence *et al* [116] (Fig. 3.16(a), blue coloured genes). In total, 34/65 `xseq` predictions overlapped *bona fide* cancer genes. We compared `xseq` predictions to those [98] predicted by an orthogonal method, MuSiC [44], which computes the statistical significance of the population mutation frequency of a gene above an expected background mutation rate to predict its role as a cancer gene. As MuSiC uses only mutation data, and not expression data, we used it as a benchmark to determine the effect of integrating gene expression data on cancer gene discovery. MuSiC predicted 22/65 genes as significantly mutated. Importantly, 13/43 of the `xseq` genes that were not predicted by MuSiC were present in the list of *bona fide* cancer genes, suggesting that integrating gene expression information can complement the existing mutation frequency-based methods to identify mutated cancer genes.

We next characterized the tumour suppressor properties of the 65 `xseq` cis-effect predictions for consistency with known patterns of enrichment for loss-of-function mutations. We found 51/65 genes with tumour suppressor characteristics ($P(\text{TSG}) \geq 0.2$). Results were robust to a more conservative threshold, yielding 47/65 genes with $P(\text{TSG}) >= 0.5$. The cis-effect loss-of-function mutations were co-associated with genomic copy number (one-way ANOVA test p-value < 0.001, Fig. 3.16(c-d)), with `xseq` cis-effect genes enriched for coincidence with hemizygous deletion. (Fisher's exact test p-value < 0.001, Fig. 3.16(c-d). The statistical test method when reporting p-values is omitted if Fisher's exact test is used.)

Additional biological characterization of the cis-effect genes suggested strong enrichment for transcription factors, phosphoproteins, and X chromosome genes. Nearly half (30/65) of the cis-effect genes encode transcription factors (Fig. 3.16(a), p-value < 0.001), as annotated in the Checkpoint database [204]. Most of the cis-effect genes (54/65, p-value < 0.001) encode human phosphoproteins, consistent with recent work predicting cancer driver genes based on enriched mutations in phosphorylation regions [166]. Finally, cis-effect genes were disproportionately found on chromosome X [42] (8/65, p-value < 0.01, Fig. 3.16(a)). Taken together, these data indicate `xseq` cis-effect predicted genes' properties are well aligned with known characteristics of tumour suppressor genes.

For the 30 novel predictions (not in our *bona fide* cancer driver gene list, nor significantly mutated based on MuSiC analysis), we searched for literature in support of their tumour sup-

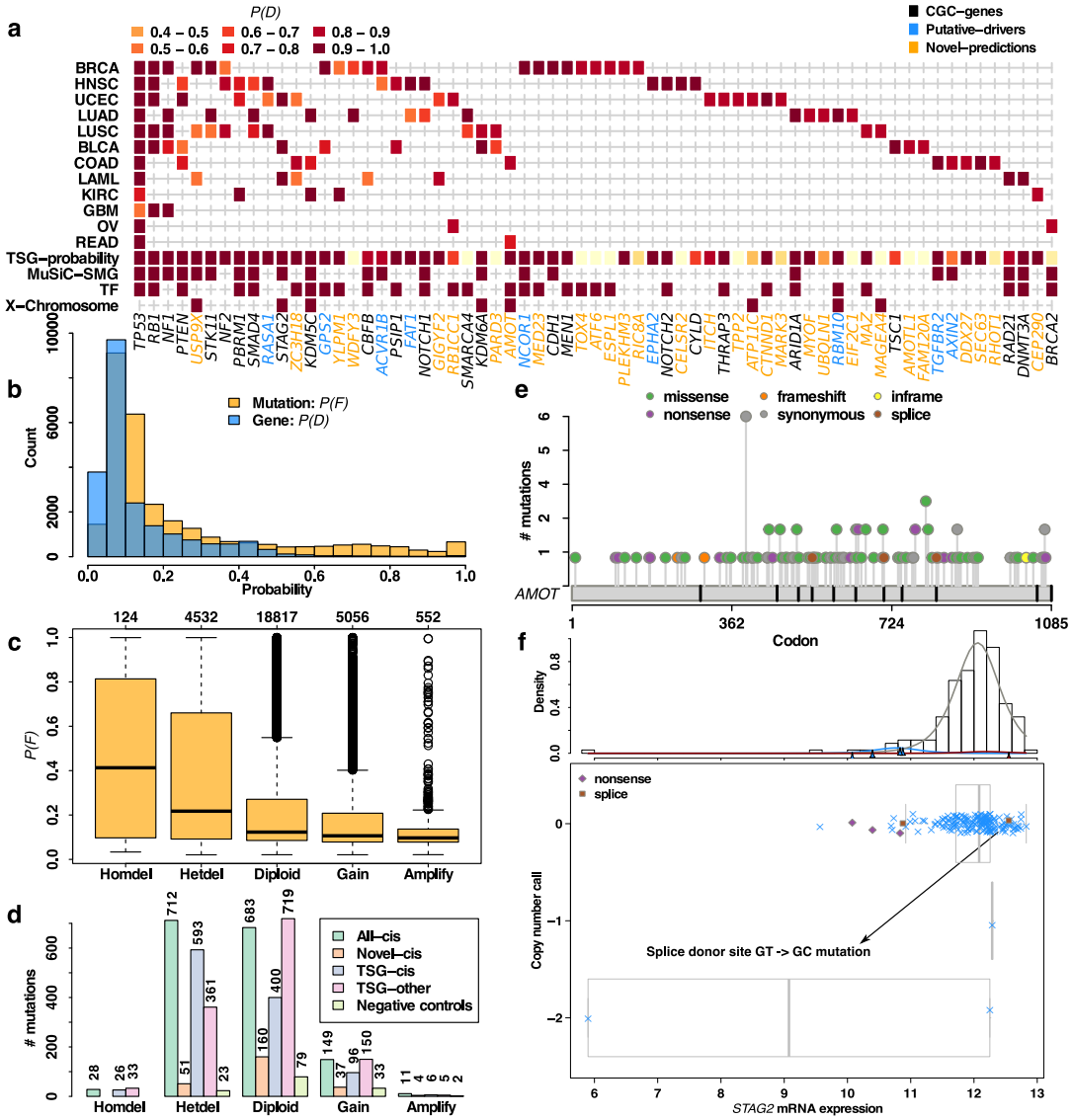


Figure 3.16: The 65 genes harboured loss-of-function mutations with strong cis-effects on the expression of these genes. (a) The predicted cis-effect loss-of-function mutations across 12 tumour types ($P(D) \geq 0.8$ in at least one tumour type). (TSG-probability, tumour suppressor gene probability; MuSiC-SMG, significantly mutated genes predicted by MuSiC; TF, transcription factors). (b) The histograms of conditional probabilities of mutations and genes given gene expression data across tumour types. (c) The conditional probabilities of mutations given gene expression data separated based on copy number status. (d) The loss-of-function mutations in the 65 cis-effect genes (All-cis), 30 novel predictions (Novel-cis), 23 cis-effect tumour suppressor genes (TSG-cis), 108 non cis-effect TSGs (TSG-other), and 30 negative control genes (Negative controls) segregated based on copy number status. (e) A "novel" tumour-suppressor gene *AMOT* is not significantly mutated based on frequency-based methods, but *AMOT* is enriched in loss-of-function mutations (tumour suppressor gene probability $P(\text{TSG}) = 0.92$). (f) The loss-of-function mutations in *STAG2* typically correlate with lower expression, except for a splice donor site mutation GT $\rightarrow$ GC mutation. (both GT and GC are used by the splicing machinery.)

pressor roles in cancer. In total, we found strong connections to tumour suppressor genes for at least 17 genes. The tumour suppressor roles of several of these genes have recently been elucidated (e.g., *UBQLN1* [182] and *MED23* [189]). Notably, three genes (*AMOT*, *AMOTL1*, and *ITCH*) encode proteins in the Hippo signalling pathway, involved in restraining cell division and promoting apoptosis. We further characterized the 30 genes according to criteria presented above for all the 65 genes. Of the 30 novel predictions, 18 genes accumulated enriched loss-of-function mutations ($P(\text{TSG}) \geq 0.2$, p-value < 0.001), ten genes encode transcription factors (p-value < 0.05), 21 genes (p-value < 0.001) encode human phosphoproteins, three genes reside on the X chromosome (p-value < 0.1). All of the novel genes were rarely mutated in the 12 studied cancer types (based on MuSiC results). 51/252 loss-of-function mutations in these genes were in hemizygous deletion regions (p-value< 0.05).

As a comparison to a negative control group of genes, we used the 30 genes flagged as false-positive cancer driver genes in a recent study [117]. These genes are not significantly mutated after correction for gene length, DNA replication time, and other factors in estimating the background mutation rates [117]. All 30 genes had $P(\text{TSG}) < 0.1$. In addition, loss-of-function mutations in these genes were not enriched in hemizygous deletion regions (p-value = 0.7, Fig. 3.16(c-d)) and all 30 were predicted to have probabilities $P(D) < 0.6$ by the `xseq` model, suggesting that the false discovery rate for `xseq` is relatively low in the TCGA data, as shown in the permutation analysis.

We next estimated the proportion of known tumour suppressor genes harbouring cis-effect loss-of-function mutations. We began by enumerating a set of 131 known tumour suppressor genes from both CGC [70] and Vogelstein *et al* [213]. We found that 23/131 genes (significant enrichment of cis-effect genes, p-value < 0.001) were predicted to exhibit cis-expression effects indicating that loss-of-function mutations in $\sim$17.6% of tumour suppressor genes yield concomitant changes in mRNA expression levels.

3.3.4 Trans-effect mutations across the TCGA data

Application of `xseq` to predict mutations impacting expression *in-trans* resulted in a total of 150 genes across the 12 cancer types ($P(D) \geq 0.8$). Sixty of the 150 (40%) genes are *bona fide* cancer genes. We characterized these 60 trans-effect genes with annotated roles in cancer according to biological functions and found that 30/60 genes encode transcription factors (p-value < 0.001), 14/60 genes encode protein kinases (p-value < 0.001) and 4/60 genes (*ATRX, BAP1, KDM5A,*

SETD2, p-value < 0.01) are chromatin regulatory factors. Moreover, 26/60 genes encode cell cycle proteins (gene ontology term: GO:0007049). By comparison, MuSiC only predicted 35/60 of these genes. One gene (*ACVR2A*) was predicted by both `xseq` and MuSiC but was not in the *bona fide* cancer gene list. Taken together, `xseq` uniquely predicted 89 novel genes through trans-impacting expression analysis.

Further investigation revealed that 29/89 of the novel predicted genes were known interacting partners of previously characterized *bona fide* cancer genes. The gene (protein) interactions were assessed based on the high-quality protein-protein interaction networks [172] (downloaded from `http://interactome.dfci.harvard.edu/H_sapiens/index.php?page=download`). As for the 60 genes above, 23/89 genes encode transcription factors (p-value < 0.1), 7/89 genes encode protein kinases (p-value < 0.05), 3/89 genes are chromatin regulatory factors (p-value < 0.1), 18/89 genes encode proteins of the cell cycle process (p-value < 0.01). Pathway analysis indicated these genes encode proteins involved in major cancer pathways such as cell proliferation, apoptotic process, mitotic cell cycle, chromatin modification, cell migration, and focal adhesion. Nineteen genes were predicted to have $P(\text{TSG})$ or $P(\text{OCG}) \geq 0.2$. The gene which harboured the largest number of high probability mutations was *KPNA2* in breast cancer. (Fig. 3.17, 43 mutations with $P(F) \geq 0.5$. Results are similar if choosing slightly different thresholds for $P(F)$.)

To examine other sources of evidence of functional effects, we analyzed high-probability missense mutations across all the tumour types, and computed the enrichment of phosphorylation-related SNVs (pSNVs) [166, 167]. (We found 839 missense mutations with $P(F) \geq 0.5$ in the genes with $P(D) \geq 0.8$, and also overlapped the set of missense mutations analyzed [167].) Of these mutations, 620 were unique when the same amino acid residue replacement in different patients was considered. We performed two analyses where the same amino acid substitution missense mutations (from different patients) were considered as separate events or collapsed into the same event (unique). The Pan-Cancer dataset included 241,700 (236,367 unique) missense mutations. Among them, 16,840 (16,074 unique) mutations were pSNVs. Of the high probability SNVs, 232/839 of them were pSNVs (134/620 unique). The high probability missense mutations were highly enriched in pSNVs in both analyses (p-value < 0.001). These results provided additional data to support functional activity of the `xseq` predictions specifically related to impact on phosphorylation.

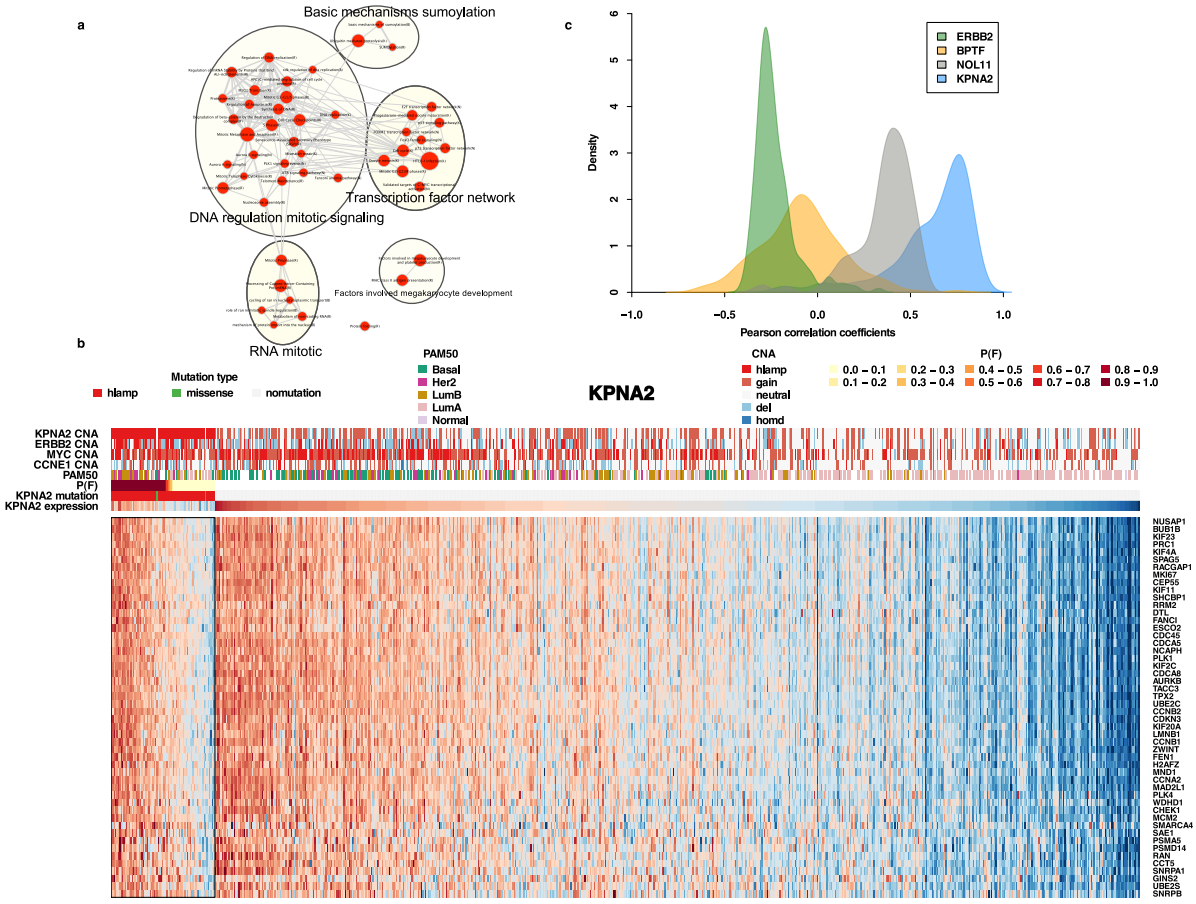


Figure 3.17: *KPNA2* amplifications in breast cancer correlated with a set of gene up-regulations. (a) These up-regulated genes play roles in DNA-regulation, mitotic signalling, transcription factor networks. (b) The tumours harbouring high-probability *KPNA2* amplifications were mostly Lumina B breast cancer. *KPNA2* expression was highly correlated with the expression of the up-regulated genes. (c) Compared to *KPNA2*, expression of genes in the same loci as *KPNA2* (e.g, *ERBB2, BPTF, NOL11*) had lower correlation with the genes up-regulated with *KPNA2* amplification.

Expression dysregulation across tumour types Certain genes are frequently mutated in multiple tumour types [98]. We asked whether these mutations across tumour types correlated with the dysregulation of the same set of genes. We focused on those genes whose mutations were predicted to influence gene expression in multiple tumour types. For each gene connected to the mutated gene g in a tumour type, we counted how many times this gene was dysregulated ($P(G =$ 'up-regulation') ≥ 0.5 or $P(G =$ 'down-regulation') ≥ 0.5) in the presence of high probability mutations ($P(F) \geq 0.5$). We analyzed down-regulation and up-regulation independently using a binomial exact test to test the significance of this correlation (high probability mutations

and gene dysregulation). The binomial distribution parameters were obtained by maximum likelihood estimation from all count data. From this analysis we found 17/20 recurrent genes had at least one gene up-regulated or down-regulated in two tumour types. Mutations in *RB1* correlated with the same group of gene dysregulations across several tumour types. In particular, we observed that *RB1* mutations correlated with E2F family gene up-regulations (e.g., *E2F1*) in BLCA, BRCA, GBM, LUSC, OV and UCEC (Table 3.3), as well as genes encoding mini-chromosome maintenance proteins, e.g., *MCM5* in BRCA, GBM, LUAD, LUSC and OV. To confirm these correlations, for each gene connected to *RB1* in the original full influence graph (Methods), in each tumour type, we compared the expression of this gene in the patients with *RB1* mutations to the patients without *RB1* mutations using the Limma package [190]. We then aggregated all the obtained p-values from the genes connected to *RB1* across tumour types, and computed the FDRs [16]. We found that *E2F1* was up-regulated in BLCA, BRCA, GBM, LUSC, and UCEC (FDR < 0.1). We performed a similar analysis for *MCM5* and found it was up-regulated in BRCA, GBM, LUSC, OV and UCEC (FDR < 0.1).

In addition, aberrations (mutations and amplifications) in the transcription factor *NFE2L2* in six different tumour types (BLCA, HNSC, KIRC, LUAD, LUSC, and UCEC) exhibited trans-effects on gene expression ($P(D) \geq 0.8$). Two genes, *MAFG* and *FECH* (Figs 3.18-3.19) were significantly up-regulated in five and four tumour types, respectively, in the presence of *NFE2L2* aberrations (FDR < 0.1). As *MAFG*, *FECH* and *NFE2L2* reside on chromosome 17, 18, and two, respectively, the correlations are not likely cased by gene dosage effects. Several other genes were also up-regulated in the presence of *NFE2L2* aberrations, e.g., *NQO1, TXNRD1, PRDX1, GSR, GPX2, GCLM, FTL, AKR1C1, TXN, SQSTM1, GSTA1, KEAP1, GSTA4, ABCC1*, and *GCLC* were up-regulated in six to three tumour types.

Stratifying patients harbouring the same gene mutations We investigated whether `xseq` probabilities $P(F)$ could stratify patients harbouring mutations in the same cancer driver gene. We analyzed each of the 127 genes from Kandoth *et al* [98] in each tumour type for the presence of bimodal `xseq` $P(F)$ distributions over patients harbouring mutations in the genes of interest. Twenty-two commonly mutated genes exhibited bimodal distributions in at least one tumour type. This was particularly evident for *CTNNB1* mutations in UCEC (Fig. 3.20(a)); 53/72 patients harboured high probability *CTNNB1* mutations ($P(F) \geq 0.5$), with all 53 patients harbouring *CTNNB1* hotspot mutations (mutations hitting codons between 31 and 45).

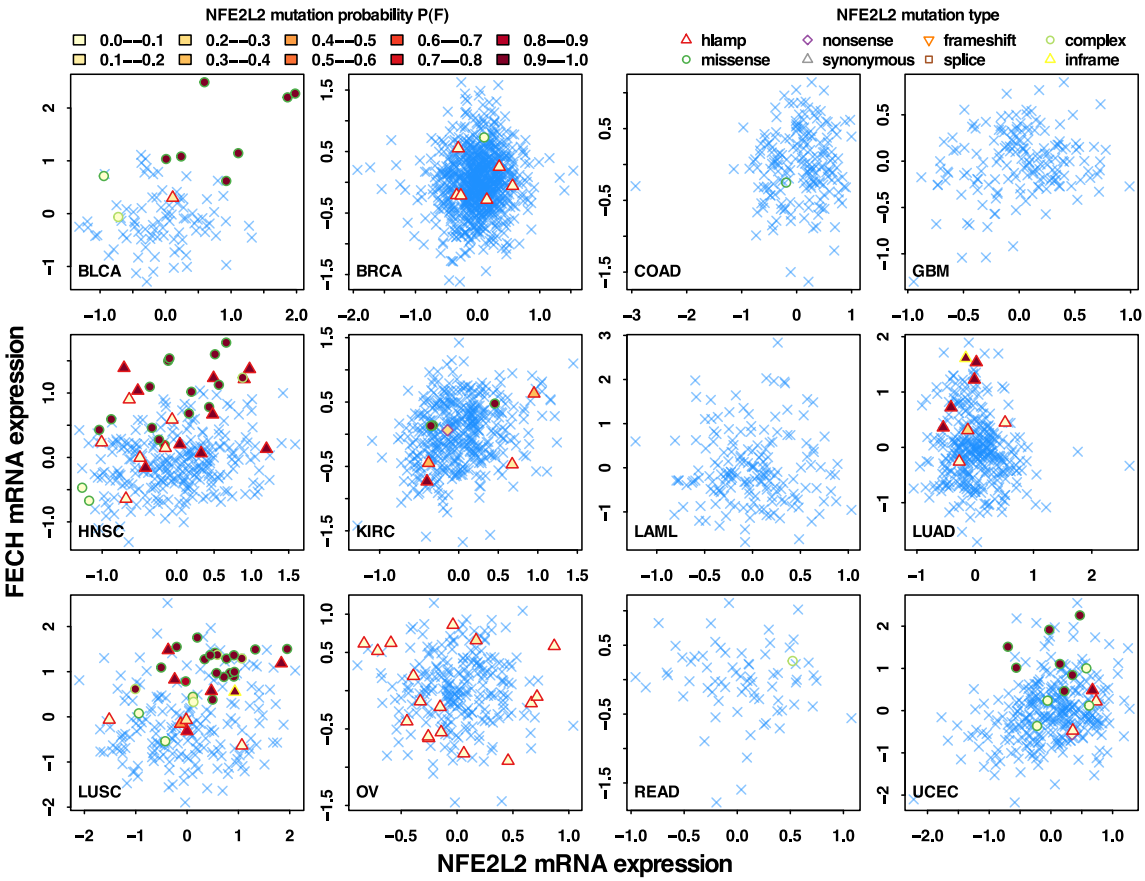


Figure 3.18: *NFE2L2* mutations and *FECH* up-regulation. *NFE2L2* mutations were predicted to correlate with *FECH* expression up-regulation in five types of cancer: BLCA, HNSC, LUAD, LUSC, and UCEC. Each dot in the scatterplots represents the expression of *NFE2L2* and *FECH* in a patient. A blue cross '×' means the patient does not have *NFE2L2* mutations. Other types of symbols represent different kinds of mutations (lamp, copy number amplifications). The filled colours encode the estimated mutation probability $P(F)$ from trans-analysis (both *FECH* expression and the expression of other *NFE2L2* interaction partners determine $P(F)$).

By contrast, only 9/19 patients without high `xseq` probability *CTNNB1* mutations harboured hotspot mutations (Fig. 3.20(c)). In addition, 9/19 patients harboured *POLE* mutations, or were annotated as "ultramutated" (tumours with more mutations than Q3+IQR $*$ 4.5, where Q3 is the third quartile of mutation counts across a corresponding tumour type, and IQR is the interquartile range, as defined in syn1729383), suggesting that the *CTNNB1* mutations were inconsequential passenger mutations. Moreover, patients in the $P(F) \geq 0.5$ group were significantly younger than patients with $P(F) < 0.5$ (mean age 57.5 versus 65.7 years old, one-sided t-test p-value < 0.01).

Similar results for *RB1* mutations in UCEC are shown in Fig. 3.20(d). All 11 loss-of-function

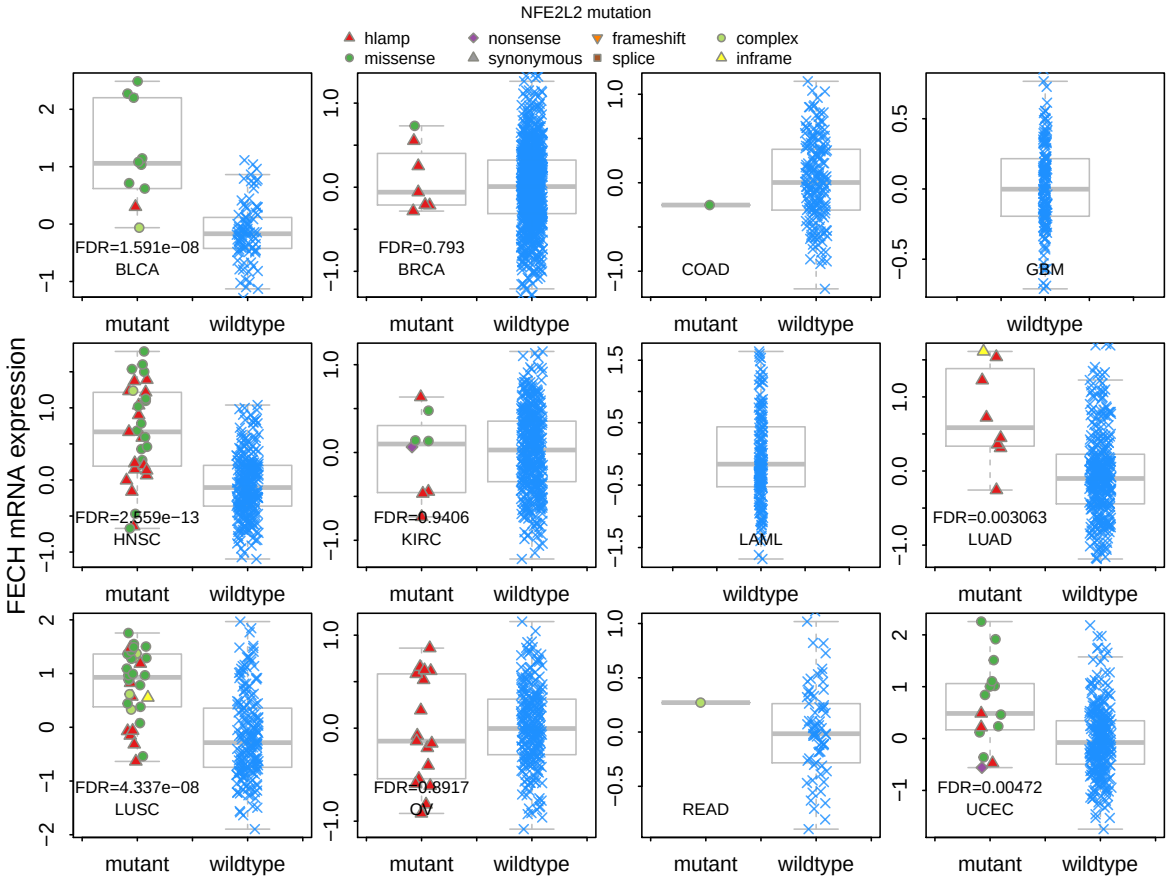


Figure 3.19: Boxplots showing *NFE2L2* mutations and *FECH* up-regulation. In BLCA, HNSC, LUAD, LUSC, and UCEC, *FECH* was up-regulated in the patients with *NFE2L2* mutations or amplifications (Limma FDR < 0.1).

mutations (in eight patients) were predicted to have high probabilities ($P(F) \geq 0.5$), however, only 2/13 patients that did not harbour loss-of-function mutations were predicted to accumulate high probability mutations ($P(F) \geq 0.5$, Fig. 3.20(d)). Taken together, although genes such as *CTNNB1* and *RB1* frequently harbour driver mutations, they still likely accumulate passenger mutations without impact on gene expression. As such, patients' tumours with these 'inert' mutations do not exhibit expected pathway dysregulation. `xseq` is therefore capable of sub-stratifying patients into meaningful phenotypic groups, separating patients with mutations *and* dysregulated pathways from those patients with mutations, but normal pathway activities.

TP53 mutations in UCEC also showed bimodal distributions. *TP53* frequently accumulates both loss-of-function mutations and missense mutations. The variation in $P(F)$ cannot be explained by the types and positions of the mutations. However, patients with $P(F)$ >= 0.5 were more likely to harbour copy number hemizygous deletions. (36 patients harboured co-

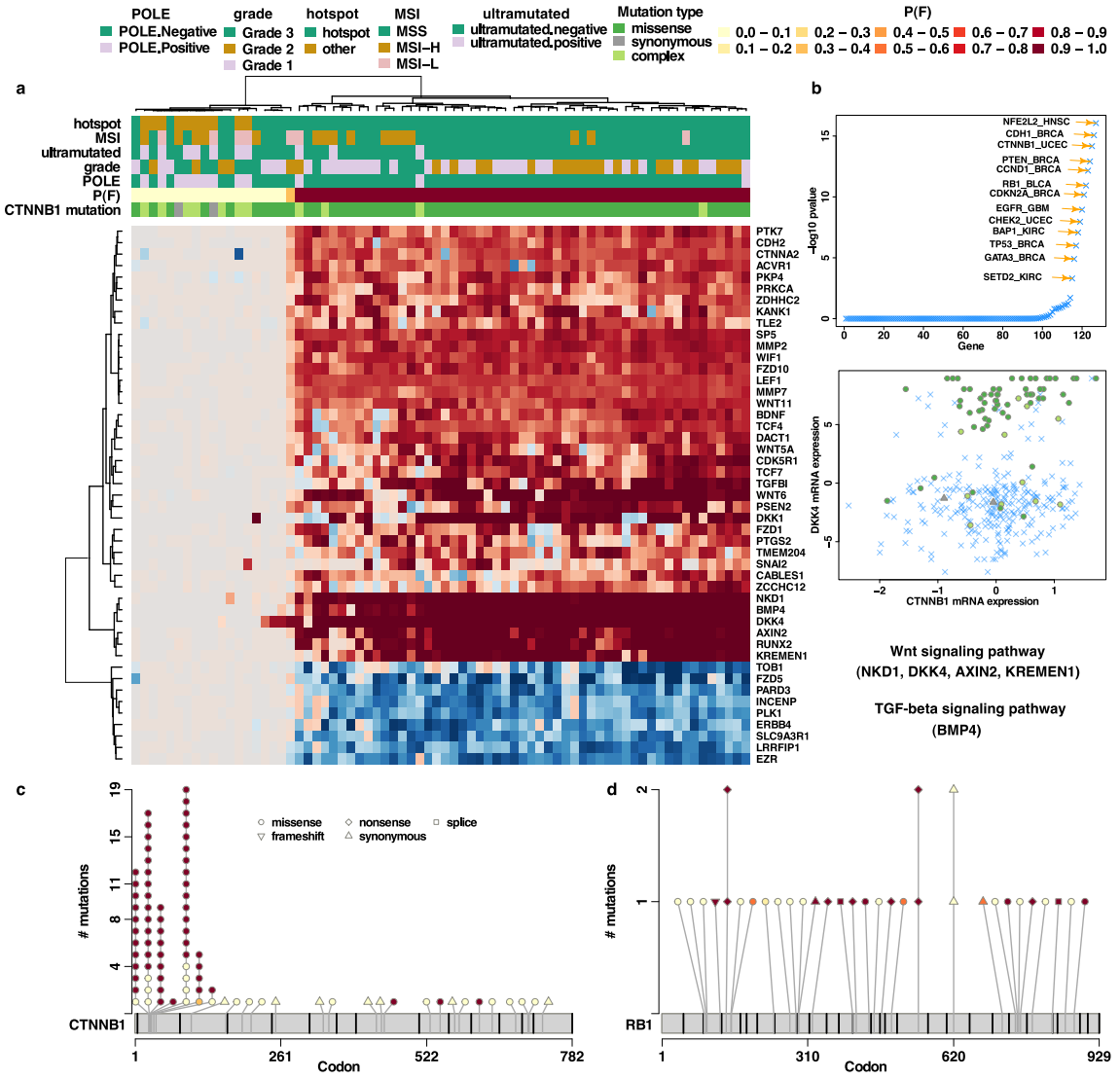


Figure 3.20: Patients harbouring the same gene mutations but with variations in trans-associated gene expression. (a) In UCEC, *CTNNB1* mutations correlated with the up-regulation of a set of genes, and down-regulation of another set of genes. The most extreme up-regulated genes included *BMP4* (in the TGF-β signalling pathway), *NKD1, AXIN2, DKK4*, and *KREMEN1* (in the Wnt signalling pathway). The down-regulated genes included Wnt signalling pathway gene *FZD5*. Here red colour in the heatmap represents gene up-regulation and blue colour represents gene down-regulation. (MSI, microsatellite instability; MSS microsatellite stable; MSI-H, MSI-high; MSI-L, MSI-low). (b) The smallest unimodality dip-test p-values of $P(F)$ of the 127 significantly mutated genes across tumour types. (c) The mutation sites, mutation types, and $P(F)$ (filled colours) of *CTNNB1* mutations (d) and *RB1* mutations in UCEC.

occurring hemizygous deletions, compared to 8 patients with $P(F) < 0.5$; only 9 patients lacked copy number alterations in the group with $P(F) \geq 0.5$ compared to 12 in the group with $P(F) < 0.5$, p-value < 0.005.)

3.4 Discussion

We developed a statistical model, `xseq` to quantitatively assess the association of mutations with dysregulated gene expression in 12 tumour types. Computational benchmarking and assessment of independent datasets have demonstrated the robustness of `xseq`. Our results have implications for the interpretation of somatic mutations in retrospective, discovery-based studies.

Systematic analysis of mutation and expression landscapes from more than 2,700 tumours uncovered several novel patterns. We revealed 30 novel tumour suppressor candidate genes by cis-effect loss of expression analysis. These genes showed the hallmarks of tumour suppressor genes including a distribution of loss-of-function mutations, and biallelic inactivation through loss-of-function mutations and heterozygous deletions. In addition, we assessed the landscape of mutations impacting gene expression in-trans across the 12 tumour types. These results implicated 89 novel genes with mutations impacting gene expression. In total, 33% of these genes had functional relationships with cancer genes in core tumourigenic processes. These genes were not nominated by mutation analysis alone, suggesting that integrated analysis of mutations and gene expression is a complementary approach towards comprehensive identification of functional mutations. Recent synthesis of mutation rates and discovery 'saturation' in genome-wide sequencing studies has indicated that current standard of study design has under-sampled important mutations, and that for some 50 tumour types, sequencing of >2000 cases are needed to reach comprehensive sampling [116]. The combined cis- and trans-analyses led to the elucidation of >100 novel candidate cancer genes predicted to impact expression. Integration of gene expression data directly into analysis of mutations will therefore help to bridge the discovery gap left by DNA mutation analysis alone.

Results from `xseq` analysis identified two important characteristics for biological interpretation of mutations. The trans-analysis revealed that the same mutated gene in different patients can exhibit distinct expression impacts. In our analysis, constitutive activation of Wnt signalling genes due to *CTNNB1* mutation segregates almost exclusively with known hotspot mutations. However, several cases exhibited mutation in *CTNNB1* without evidence of Wnt activation, resulting in low xseq probabilities. These cases were primarily phenotyped as hypermutators due to mismatch repair deficiency and/or *POLE* mutations [201], and patients were statistically older at diagnosis. Thus, a real phenotypic distinction associates with low and high xseq $P(F)$ probabilities, providing evidence for integrative analysis of mutations and expression as a route

to stratifying phenotypically distinct tumours in the context of the same mutations.

`xseq` analysis identified several genes that had consistent expression impact across tumour types. Despite distinct histologies and cell contexts of source tumours, *RB1* loss-of-function mutations and *NFE2L2* mutations/amplifications exhibited similar expression patterns. *RB1* binds and inhibits the E2F transcription factor family. Accordingly, we observed that *RB1* mutations correlated with E2F family gene up-regulation across tumour types. *NFE2L2* binds to its regulator *KEAP1* and regulates the expression of antioxidant-related genes to protect against oxidative damage. We observed *NFE2L2* mutations correlated with up-regulation of *KEAP1*, as well as of oxidative stress genes (e.g., *GCLM, GCLC, TXNRD1, GPX2, and NQO1*). While therapeutic responses to targeted inhibitors administered against the same mutation can have variable effects due to intrinsic gene expression context (e.g., *BRAF* inhibition in melanoma and colorectal cancer [162]), the mutations we outlined (such as *RB1* and *NFE2L2* mutations) exhibit stable profiles and represent important targets for future development of broadly applicable therapeutics. An intriguing evolutionary implication arises from these mutations: phenotypic impact is selected for in multiple heterogeneous tumour micro-environments, indicating independent convergence of phenotype transcending cell context.

In conclusion, this work provides a route towards closing the cancer gene discovery gap in the field of cancer genome sequencing. Direct, model-based integration of mutations and co-acquired gene expression measurements from tumour samples enhances interpretation capacity of discovered mutations leading to optimal selectivity of targets for functional studies and development of novel therapeutics.

3.4.1 Limitations

`xseq` is not able to distinguish different mutations of a gene within a specific patient –a limitation, as these mutations may result in different functional impacts. Although genes are rarely mutated multiple times within a single patient, some large tumour suppressor genes (such as *ARID1A*) accumulate multiple mutations, as a result of their long coding sequences. Similarly, in glioblastoma and lung cancers, *EGFR* is frequently mutated multiple times in single patients, often due to the emergence of clonal populations following the administration of *EGFR* inhibitors [104]. Examining the expression impact properties of such mutations in clonal populations would likely require advanced single cell methods [157].

Inference for the `xseq-simple` model is efficient (time and space linear in the number of

nodes in the model) since it is a tree. For the `xseq` model given the gene regulation variables H, the inference is also efficient as it is a polytree with tree width of two. The most computational resources gain in calculating and storing the doubled in size tables in Equation 3.20 and in estimating the parameter in Equation 3.24. However, when the number of patients is large, the Gaussian process regression used to remove the cis-effects of copy number alteration on gene expression could be slow. As we process each gene independently, this step can easily be parallelized.

Each input (mutations, gene expression, and the influence graph) influence the `xseq` predictions as shown in the permutation analyses. Although permuting the influence graph does not completely remove the signals because we select subnetworks from the influence graph as inputs to the `xseq` model. The mixture of Student's t-distribution modelling of gene expression distribution is quite robust to outliers and the number of patients with measured gene expression values. However, for some genes (e.g, genes not expressed in the studied tumour type), their expression may not follow a mixture of three Student's t-distribution. The expression of these genes is better removed from further analyses.

Chapter 4

densityCut: an efficient and versatile topological approach for automatic clustering of biological data

"Nothing truly valuable can be achieved except by the unselfish cooperation of many individuals."

– Albert Einstein, 1940

4.1 Introduction

In the previous chapters, we have investigated the problems of predicting mutations from high-throughput DNA sequencing data and estimating the impacts of these mutations on gene expression. Interestingly, we found that some patients, although their mutations were quite different, showed high similarity in gene expression patterns. In addition, recent studies have demonstrated that for some cancer, the treatments based on phenotype (gene expression) produce better outcomes than the treatments based on genotype (mutations). Because of the complexity of the working mechanisms of human cancer, it is necessary to integrate all the measurements, e.g., mutation, gene expression, DNA methylations and micro-RNA expression to provide a comprehensive view of a specific tumour. Furthermore, to deliver the best treatment options to a patient, it is advantageous to divide (cluster) patients into distinct subgroups of clinical relevance, and apply similar treatment options to a group of patients.

Clustering analysis (unsupervised machine learning), which organizes data points into sensible and meaningful groups, has been increasingly used in the analysis of high-throughput biological datasets. For example, The Cancer Genome Atlas project has generated multiple omics data for individual patients. One can cluster the omics data of individuals into subgroups of potential clinical relevance. To study clonal evolution in individual cancer patients, we can cluster variant allele frequencies of somatic mutations [185], such that mutations in the same cluster are accumulated during a specific stage of clonal expansion. Emerging technologies such

as single-cell sequencing have made it possible to cluster single-cell gene expression data to detect rare cell populations, or to reveal lineage relationships [161, 232]. One can cluster single-cell mass cytometry data to study intratumor heterogeneity [120]. As measurement technology advances have drastically enhanced our abilities to generate various high-throughput datasets, there is a great need to develop efficient and robust clustering algorithms to analyze large N (the number of data points), large D (the dimensionality of data points) datasets, with the ability to detect arbitrary shape clusters and automatically determine the number of clusters.

The difficulties of clustering analysis lie in part with the definition of a cluster. Of the numerous proposed clustering algorithms, the density-based clustering algorithms [63, 69] are appealing because of the probabilistic interpretation of a cluster generated by these algorithms. Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^D$ be drawn from an unknown density function $f(\mathbf{x}), \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$. For model-based approaches such as Gaussian mixture models $f(\mathbf{x}) = \sum_{c=1}^C \pi_c \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, a cluster is considered as the points generated from a mixture component, and the clustering problem is to estimate the parameters of the density function from $\mathcal{D}$ [63]. To analyze datasets consisting of complex shape clusters, nonparametric methods such as kernel density estimation can be used to estimate $\hat{f}(\mathbf{x}) = \sum_{i=1}^N K_h(\mathbf{x}, \mathbf{x}_i)$, where $K_h(\cdot)$ is the kernel function with bandwidth h. Here a cluster is defined as the data points associated with a 'mode' of the density function $f(\mathbf{x})$ [227]. The widely used 'mean-shift' algorithm [34, 38, 69] belongs to this category, and it locates the modes of the kernel density function $\hat{f}(\mathbf{x})$ by iteratively moving a point along the density gradient until convergence. This algorithm, however, is computationally expensive, having time complexity $O(N^2T)$, where T is the number of iterations, typically dozens of iterations are sufficient for most cases. A more efficient, non-iterative graph-based approach [108] constructs trees such that each data point $\mathbf{x}_i$ represents a node of a tree, the parent of node $\mathbf{x}_i$ is a point $\mathbf{x}_j$ which is in the direction closest to the gradient direction $\nabla\hat{f}(\mathbf{x}_i)$, and the root of a tree corresponds to a mode of $\hat{f}(\mathbf{x})$. Then each tree constitutes a cluster. This algorithm has been used to reduce the time complexity of the mean-shift algorithm to $O(N^2)$ [211], and has been extended in several ways, e.g., constructing trees after filtering out noisy modes [171].

Nonparametric clustering methods have been generalized to produce a hierarchical cluster tree [86]. Consider the λ level set of a density function $f(x)$:

$$L(\lambda; f(\mathbf{x})) = \{\mathbf{x} \mid f(\mathbf{x}) \geq \lambda\}.$$

The 'high level clusters' at level λ are the connected components of $L(\lambda; f(\mathbf{x}))$ (in the topological sense, the maximal connected subsets of $L(\lambda; f(\mathbf{x}))$). As λ goes from 0 to $\max f(\mathbf{x})$, the high level clusters at all levels constitute the level set tree, where the leaves of the tree correspond to the modes of $f(\mathbf{x})$ [197]. The widely used DBSCAN algorithm [57] extracts the high level clusters at just one given level λ. Many original approaches for level set tree construction in statistics [141] take the straightforward 'plug-in' approach to estimating the level set tree from $\hat{f}(\mathbf{x})$ by partitioning the feature space, i.e., $\mathcal{X}$. Therefore, they are computationally demanding, especially for high-dimensional data. Recently, efficient algorithms have been proposed to partition the samples $\mathcal{D}$ directly [33, 109]. Recovering the level set tree from a finite dataset is more difficult than partitioning the dataset into separate clusters. Correspondingly, theoretical analyses show that for these algorithms to identify salient clusters from finite samples, the number of data points N needs to grow exponentially in the dimension D [33, 109]. Moreover, although the level set tree provides a more informative description of the structure of the data, many applications still need the cluster membership of each data point, which is not available directly from the level set tree.

The spectral clustering algorithm [153, 188] works on an N by N pairwise data similarity matrix $\mathbf{S}$, where each element $S_{i,j}$ measures the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. The similarity matrix can be considered as the adjacency matrix of a weighted graph $\mathcal{G} = (V, E)$, where vertex v_i represents $\mathbf{x}_i$ and the edge weight $E_{i,j} = S_{i,j}$. Given the number of clusters C, the spectral clustering algorithm partitions the graph $\mathcal{G}$ into C disjoint, approximately equal size clusters, such that the points in the same cluster are 'similar', while points in different clusters are 'dissimilar'. In contrast to density-based methods, the spectral clustering algorithm does not make assumptions on the probabilistic model which generates data $\mathcal{D}$ [215]. Therefore, selecting the number of clusters is a challenging problem for spectral clustering algorithms, especially in the presence of outliers or when the number of clusters is large. In addition, the spectral clustering algorithm is time-consuming because it needs to compute the eigenvalues and eigenvectors of the row-normalized similarity matrix $\mathbf{S}$, requiring $\Theta(N^3)$ time. Instead of using single value decomposition to calculate the eigenvalues and eigenvectors, the power iteration clustering algorithm (PIC) [125] iteratively smoothes a random initial vector by the row-normalized similarity matrix, such that the points in the same cluster will be similar in value. Then the k-means algorithm is used to partition the smoothed vector into C clusters. Although PIC has a time complexity of $O(N^2T)$, where T is the number of iterations, PIC

may encounter many difficulties in practice. First, the points from two quite distinct clusters may have very similar 'smoothed' densities, and therefore they may not be distinguishable by k-means. Second, the points in a non-convex shape cluster can break into several clusters. As the number of clusters increases, these problems become more severe [125].

In this chapter, we introduce a simple and efficient clustering algorithm, `densityCut`, which shares some advantages of both density-based clustering algorithms and spectral clustering algorithms. As for spectral clustering algorithms, `densityCut` works on a similarity matrix; thus it is computationally efficient, even for high-dimensional data. Using a sparse K-nearest neighbour graph further reduces the time complexity. Besides, we can use a random walk on the K-nearest neighbour graph to estimate densities at each point. As for many density-based clustering algorithms, `densityCut` is simple, efficient, and there is no need to specify the number of clusters as an input. Moreover, `densityCut` inherits both methods' advantage of detecting arbitrarily shaped clusters. Finally, `densityCut` offers a novel way to build a hierarchical cluster tree and to select the most stable clustering. We first benchmark `densityCut` against ten widely used simulation datasets and two microarray datasets to demonstrate its robustness. We then use `densityCut` to cluster variant allele frequencies of somatic mutations to infer clonal architectures in tumours, to cluster single-cell gene expression data to uncover cell population compositions, and to cluster single-cell mass cytometry data to detect communities of cells of the same functional states or types.

4.2 Methods

The `densityCut` method consists of four major steps (Algorithm 1): (1) density estimation: given a set of data points $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^D$, form a directed unweighted K-nearest neighbour graph and estimate the densities of data points by using the K-nearest neighbour density estimator; (2) density refinement: refine the initial densities via a random walk on the unweighted K-nearest neighbour graph; (3) local-maxima based clustering: detect local maxima of the estimated densities, and assign the remaining points to the local maxima; (4) hierarchical stable clustering: refine the initial clustering by merging neighbour clusters. This cluster merging step produces a hierarchical clustering tree, and the final clustering is obtained by choosing the most stable clustering as the threshold in merging clusters varies. Fig. 4.1 demonstrates how `densityCut` works on a toy example [68]. Below we discuss each step in detail.

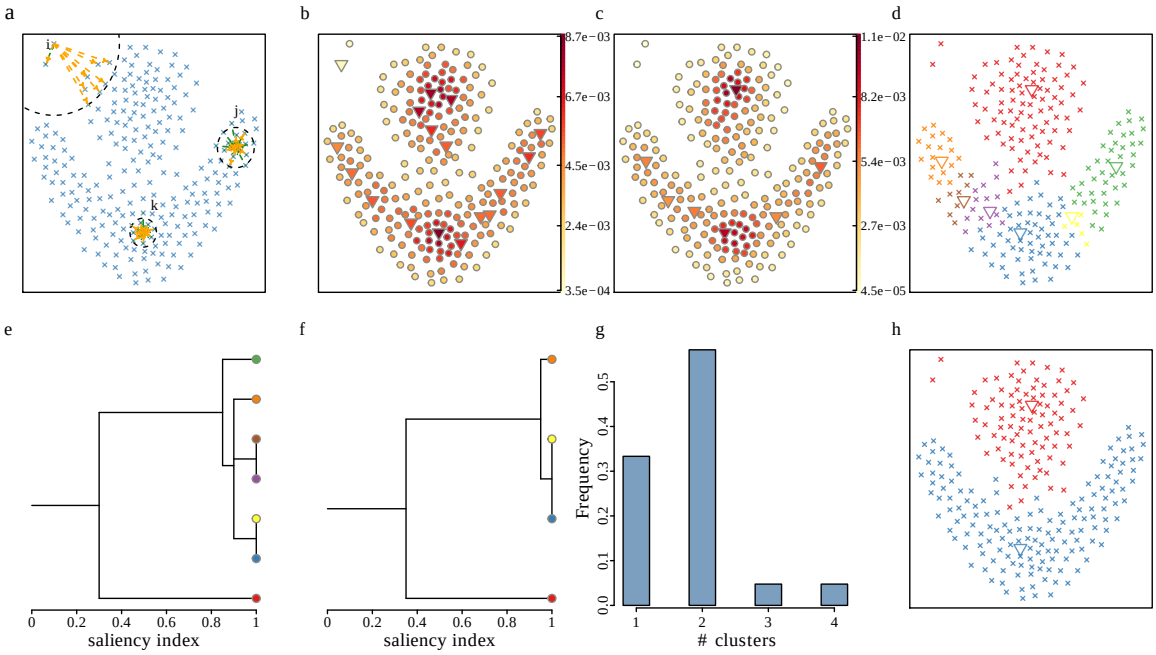


Figure 4.1: Major steps of the `densityCut` algorithm. (a) Data points in $\mathcal{D}$. (This dataset was introduced by Fu *et al* [68].) The dotted black circles represent the balls containing $K = 8$ points from $\mathcal{D}$ centred at three example points, i, j and k, whose densities to be estimated. Each point connects to its $K = 8$ nearest-vertices by orange arrows. Other points connect to i, j and k by green arrows if i, j and k are among the K in-vertices of these points. Notice the asymmetry of in-vertices and out-vertices of a vertex in a Knn graph, e.g., vertex v_i has one in-vertex but $K = 8$ out-vertices. (b) Colour coded Knn estimated densities at points from $\mathcal{D}$. The modes of densities are represented by triangles '$\triangledown$'. (c) The refined densities based on a random walk. (d) Initial clustering by assigning data points to modes. (e) The tree created by merging clusters without adjusting valley heights. (f) The tree created by merging clusters based on the saliency index using the adjusted valley heights. (g) The cluster number frequency plot. (h) The final clustering results.

4.2.1 Density estimation

We adopt the K-nearest neighbour density estimator to estimate the density at $\mathbf{x} \in \mathbb{R}^D$ (Fig. 4.1(a)):

$$f_K(\mathbf{x}) = \frac{(K-1)/N}{V_K(\mathbf{x})} \tag{4.1}$$

where $V_K(\mathbf{x}) = V_D \times (r_K(\mathbf{x}))^D$ is the volume of the smallest ball centred at $\mathbf{x}$ containing K points from $\mathcal{D}$. V_D is the volume of the unit ball in the D-dimensional space, and $r_K(\mathbf{x})$ is the distance from $\mathbf{x}$ to its Kth nearest neighbour. Compared to the widely used kernel density estimator, K-nearest neighbour density estimates are easier to compute, and also the parameter K is more intuitive to set than the kernel bandwidths for kernel density estimators. Here for simplicity,

Algorithm 1 The `densityCut` algorithm. Unless otherwise specified, all results in this paper are obtained with $K = \log_2(N)$ and $\alpha = 0.9$.

Input

- A set of data points $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^D$
- The number of nearest neighbours K
- The damping factor α

1. Density estimation

$$f_i^0 = \frac{(K-1)/N}{V_K(\mathbf{x}_i)}$$

$$W_{i,j} = \begin{cases} 1 & \mathbf{x}_j \in K\text{nn}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases}$$

2. Density refinement

$$P_{i,j} = \frac{W_{i,j}}{\sum_j W_{i,j}}$$

Iterate until $||\mathbf{f}^{t+1} - \mathbf{f}^t|| \leq \epsilon$ (default value: $\epsilon = 10^{-6}$)

$$\mathbf{f}^{t+1} = \alpha \mathbf{f}^t \mathbf{P} + (1-\alpha)\mathbf{f}^0$$

3. Local-maxima based clustering

Detect modes, i.e., local maxima, of the underling density function from
$\{v_j \mid \forall P_{i,j} > 0, f_i < f_j\}$
Build trees of points rooted at the modes, using
$\text{Parent}(v_i) = \arg\min_{v_j \in \mathcal{N}_i} (\left|d_j - d_i\right| \mid f_i < f_j)$
Build one cluster per tree, containing all points in that tree

4. Hierarchical stable clustering

Calculate heights of trees and valleys
(Optional) adjust valley heights based on Equation 4.10
Compute the saliency index ν for a pair of adjacent trees (Equation 4.9)
Merge clusters to generate a hierarchical tree by varying ν,
$\nu \in \{0.0, 0.05, 0.10, \ldots, 0.95, 1.0\}$
Select the most stable clustering

we only calculate the densities at data points from $\mathcal{D}$, represented by $\mathbf{f}^0 = (f_1^0, \ldots, f_N^0)^T$, where $f_i^0 = f_K(\mathbf{x}_i)$ and the superscript '0' indicates that this is the initial Knn density estimate since we will refine this density in the next section. Fig. 4.1(b) shows the estimated densities at each data point.

When computing the Knn densities, we can get the K-nearest neighbour graph $\mathcal{G}$ as a

byproduct with the following adjacency matrix:

$$W_{i,j} = \begin{cases} 1 & \mathbf{x}_j \in K\text{nn}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

As $\mathbf{x}_i$ maps to node v_i in the Knn graph $\mathcal{G}$, we next use 'points' and 'nodes' interchangeably.

The Knn graph $\mathcal{G}$ is a directed unweighted graph. While the sets of in-vertices and out-vertices of many nodes may overlap significantly, some outliers may have few, if any in-vertices. In addition, points at the boundary of density changes may also have quite different sets of in-vertices and out-vertices because their in-vertices commonly consist of points from low-density regions while out-vertices are usually from high-density regions.

4.2.2 A random walk based density refinement

As Knn density estimates are based on order statistics and tend to be noisy, we next introduce a way to refine the initial Knn density vector $\mathbf{f}^0 = (f_1^0, \ldots, f_N^0)^T$. Our refinement is based on the intuition that 1) a high-density vertex belongs to one of the K-nearest neighbours of many vertices, and 2) a vertex tends to have high density if its in-vertices also have high densities. For example, point k in Fig. 4.1(a) has nine in-vertices, and these vertices are in high-density regions, and indeed, the density of k is 0.0087 which is higher than average for this example. Based on the above assumptions, we get the following recursive definitions of densities:

$$f_j^{t+1} = \alpha \sum_i P_{i,j} f_i^t + (1-\alpha) f_j^0 \tag{4.3}$$

where $\alpha \in [0, 1]$ specifies the relative importance of information from v_j's in-vertices and the initial density estimate f_j^0, which is the K-nearest neighbour density estimate. If each row of $\mathbf{P}$ sums to one, $\mathbf{P}$ is a Markov transition matrix, which contains the transition probability from a vertex to its K out-vertices. Therefore, Equation 4.3 defines a random walk with restart. The refined density vector is the stationary distribution of the random walk on the Knn graph with adjacency matrix $\mathbf{P}$. We can row-normalize matrix $\mathbf{W}$ to get $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$, where $\mathbf{D} = \text{diag}(\sum_j \mathbf{W}_{1,j}, \ldots, \sum_j \mathbf{W}_{N,j})$. Compared to the original Knn densities in Fig. 4.1(b), the refined densities in Fig. 4.1(c) have fewer 'local maxima' (shown as triangle points, see next section for details). Similar methods have been used in information retrieval and semi-supervised learning applications [155].

Equation 4.3 can be solved exactly in the limit by $\mathbf{f} = (1-\alpha)\mathbf{f}^0(\mathbf{I}-\alpha\mathbf{P})^{-1}$ if $\alpha < 1$. Therefore, the above iterations guarantee convergence. When $\alpha = 1$, $\mathbf{f}^t$ converges to a left eigenvector of $\mathbf{P}$ with the maximum eigenvalue of 1. In our computational experiments, when $\alpha < 1$, e.g., $\alpha = 0.90$, Equation 4.3 typically converges within a few dozen iterations, and can be much faster than the case when $\alpha = 1$. This rough density estimation process is faster than methods which attempt to solve density estimation to a high degree of precision since density estimation is well known to be a difficult problem. Moreover, our method can be applied to data that are presented in the form of a graph, rather than as data points over the reals.

4.2.3 Local-maxima based clustering

After obtaining densities for points, we estimate the 'modes' of the underlying probability density function. The modes are the 'local maxima' of the density function with zero gradients. For finite samples, the modes are rarely located exactly at points $\mathbf{x}_i \in \mathcal{D}$, so we use the points close to the modes instead. Mathematically, modes can be approximated by points $\{\mathbf{x}_i \mid \max_{|\mathbf{x}_j-\mathbf{x}_i|<\epsilon} f_j \leq f_i\}$, where ϵ is a small distance threshold. The distance ϵ is dataset dependent and difficult to choose in practice. Instead, we can define a mode as a vertex whose density is the largest among all of its in-vertices:

$$\{v_j \mid \forall P_{i,j} > 0, f_i < f_j\} \tag{4.4}$$

Here we use in-vertices instead of out-vertices in order to be able to detect small cluster with less than K data points. As the vertices of a small cluster with less than K vertices can form a clique (or are highly connected to each other) in the Knn graph, we can detect this small cluster based on the definition of local maxima above if these points are not among the K-nearest neighbours of points outside this cluster. A potential problem is that some outliers with very few in-vertices could be detected as local maxima. We simply remove those local maxima whose numbers of in-vertices are less than $K/2$. In other words, we treat a cluster less than $K/2$ in size as an 'outlier' cluster, and `densityCut` is unlikely to detect this small cluster.

Data points that fall into the basin of attraction of each mode constitute a cluster. This process can be done by moving each point along its gradient direction to reach a mode. We modify the efficient graph-based hill-climbing algorithm [108] to build a unique forest (a set of

trees) for a given dataset. The parent of vertex v_i is defined as

$$\text{Parent}(v_i) = \arg\min_{v_j \in \mathcal{N}_i} (\left| d_j - d_i \right| \mid f_i < f_j) \tag{4.5}$$

where $\mathcal{N}_i$ is the set of in-vertices of v_i. In other words, the parent of v_i is the vertex which is closest to v_i among all of v_i's in-vertices that have higher densities than v_i. From the construction of the trees, we can see that each vertex is associated with just one tree. Therefore, each tree can be considered as a cluster. Fig. 4.1(d) shows the clusters by assigning data points to the seven local maxima.

4.2.4 Hierarchical stable clustering

We then generate a hierarchical cluster tree and select the most stable clustering. First the density of the root of a tree $\mathcal{T}$ generated above is called the height of this tree, denoted by $h_{\mathcal{T}}$, which has the largest density among all the vertices in $\mathcal{T}$. Then we define the boundary points between trees $\mathcal{T}_1$ and $\mathcal{T}_2$:

$$B(\mathcal{T}_1, \mathcal{T}_2) = \{v \in \mathcal{T}_1 \mid \exists u \in \mathcal{N}_v \cap \mathcal{T}_2, f_v < f_u\} \tag{4.6}$$

Sets $B(\mathcal{T}_1, \mathcal{T}_2)$ and $B(\mathcal{T}_2, \mathcal{T}_1)$ are not the same: $B(\mathcal{T}_1, \mathcal{T}_2) \subset \mathcal{T}_1$ and $B(\mathcal{T}_2, \mathcal{T}_1) \subset \mathcal{T}_2$. The valley separating two trees is:

$$\text{Valley}(\mathcal{T}_1, \mathcal{T}_2) = B(\mathcal{T}_1, \mathcal{T}_2) \cup B(\mathcal{T}_2, \mathcal{T}_1), \tag{4.7}$$

The height of the valley separating $\mathcal{T}_1$ and $\mathcal{T}_2$ is defined as

$$h_{\text{Valley}(\mathcal{T}_1, \mathcal{T}_2)} = \max_{v \in \text{Valley}(\mathcal{T}_1, \mathcal{T}_2)} f_v \tag{4.8}$$

The saliency index ν of a valley represents the relative height of the valley compared to the shorter tree:

$$\nu(\mathcal{T}_1, \mathcal{T}_2) = \frac{h_{\text{Valley}(\mathcal{T}_1, \mathcal{T}_2)}}{\min(h_{\mathcal{T}_1}, h_{\mathcal{T}_2})} \tag{4.9}$$

Fig. 4.2(a) shows the height of the valley (the length of the grey arrow) separating two adjacent trees, and the saliency index is the ratio between the length of the grey arrow and the black

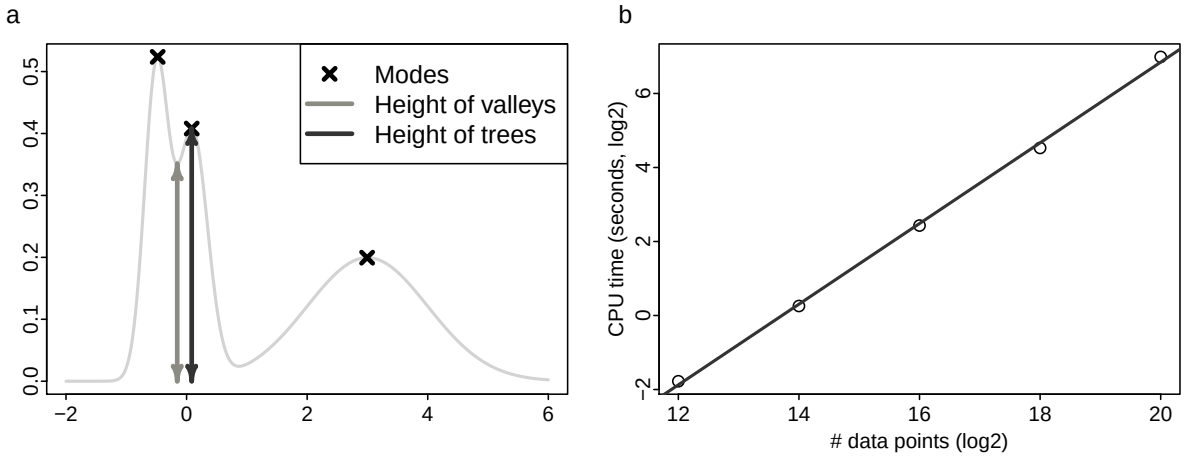


Figure 4.2: (a) Merging the first two trees (clusters) based on the relative height of the valley separating the two trees. (b) The time (in seconds, $\log_2$ transformed) increases almost linearly with the number of data points ($\log_2$ transformed).

arrow.

The saliency index defined in Equation 4.9 has several properties. First, $0 \leq \nu \leq 1$, and ν is invariant under scaling of the densities by a positive constant factor. This 'scale-free' property is very useful for us to select a threshold for merging trees. Second, it automatically scales to the local densities of trees, e.g., to get the same saliency index, the height of the valley separating two trees in low-density regions is shorter than that separating two trees in high-density regions.

We can merge two clusters if the saliency index between them is above a threshold ν. When the saliency index $\nu = 1$, no clusters get merged, and when $\nu = 0$, all the connected clusters get merged to form a single cluster. Therefore, by gradually decreasing the saliency index threshold, we can get a hierarchical clustering tree, which is useful for us to interpret the structure of data, especially for high-dimensional data. Fig. 4.1(e) shows the cluster tree from merging neighbouring clusters in Fig. 4.1(d).

For a dataset consisting of clusters whose densities are considerably different, a single K for density estimation may be insufficient. This is because the points from a high-density cluster need a larger K to estimate their densities, compared to the points from a low-density cluster. The high-density cluster could 'break' into several small clusters when K is small. Instead of picking the parameter K on a data point basis, we can adjust the height of a valley by:

$$\hat{h}_{\text{Valley}(\mathcal{T}_1,\mathcal{T}_2)} = \Big(1 + \sum_i \frac{f_i < h_{\text{Valley}(\mathcal{T}_1,\mathcal{T}_2)}}{N}\Big) h_{\text{Valley}(\mathcal{T}_1,\mathcal{T}_2)} \tag{4.10}$$

The intuition behind this density adjustment step is that a high-density valley could be an

artefact caused by splitting a high-density cluster. Fig. 4.1(f) shows the cluster tree using the saliency indexes based on the adjusted valley heights. A potential problem of this step is that it also increases the valley height between two genuine clusters and increases their likelihood to merge. For example, as can be seen from Fig. 4.1(e-f), the two clusters merge at saliency index around 0.30 before valley adjustment, but merge at saliency index around 0.35 after valley adjustment. We will further investigate the influence of this density adjustment step on clustering in later sections.

We finally introduce a method to determine the number of clusters to produce the final clustering from the hierarchical cluster tree. The basic idea is that by gradually decreasing the saliency index, clusters will get merged. Noisy, non-salient clusters will get merged quickly, and true clusters will exist for a long period of time. We therefore can calculate the length of saliency index change for producing a fixed number of clusters, and select the cluster number which spans the longest saliency index changes. In our current implementation of `densityCut`, we decrease the saliency index evenly and therefore we can interpret the saliency index change interval as 'Frequency'. Fig. 4.1(g) shows the cluster number frequency plot, and Fig. 4.1(h) shows the final clustering by merging the initial clustering to produce two clusters as selected by the cluster number frequency plot.

4.2.5 Complexity analysis and implementation

`densityCut` has been implemented in the statistical computation language `R`. `densityCut` has a worst-case time complexity of $O(NK + C^2)$ and a space complexity of $O(NK + C^2)$, where N is the number of data points, K is the number of neighbours and C is the number of clusters (local maxima). In practice, as a majority of clusters are only adjacent to few clusters, the time and space complexity is typically of $O(NK + C)$. We did not consider the time used to compute the Knn graph in `densityCut` as numerous algorithms have been developed for efficient Knn search with different complexities, and typically it takes less time to compute the Knn graph compared to cluster the data. To build the Knn graph given a data matrix, efficient algorithms such as kd-trees can be used in low-dimensional spaces ($D \leq 20$) with time complexity $O(N \log(N))$ [149]. To build the Knn graph in high-dimensional spaces ($D < 1000$), efficient software libraries based on random projection exist to repeatedly partition the data to build a tree (`https://github.com/spotify/annoy`). This algorithm can run in $O(NDT)$ time , where T is the number of trees, and typically dozens of trees are enough to preserve the

accuracy of Knn search.

To demonstrate the scalability of `densityCut`, we tested `densityCut` on a Mac desktop computer running OS X Version 10.9.5. The computer has 32 GB of RAM and a 3.5GHz four-core Intel i7 processor with 8MB cache. We carried out all the experiments presented in the paper on this computer.

We sampled $\{2^{12}, 2^{14}, 2^{16}, 2^{18}, 2^{20}\}$ data points from a mixture of 64 two-dimensional Gaussian distributions. As can be seen from Figure 4.2(b), the running time increased almost linearly in the number of data points. It took about 127 CPU seconds to cluster a million data points ($N = 2^{20}$).

4.2.6 Parameter setting

Our algorithm has two parameters: the number of nearest neighbours K, and the damping factor α in density refinement. K should be small enough to detect local maxima, e.g., smaller than the number of data points in a cluster. However, very small K can result in poor density estimates and produce large numbers of clusters, thus 'overfitting' the data, and there may not exist a 'gap' in the cluster number frequency plot for us to select the number of clusters. On the contrary, for large K, `densityCut` may fail to detect detailed structures thus 'underfitting' the data.

Theoretical analysis for spectral clustering shows that K should be $\Omega(\log(N))$ to produce a connected graph [215], and limit results are obtained under conditions $K/\log(N) \to \infty$ and $K/N \to 0$. K is also dependent on the dimensionality D. For the density estimate at $\mathbf{x}$ ($f(\mathbf{x})$ is Lipchitz smooth in a neighbour of $\mathbf{x}$) from its K-nearest neighbours, under conditions $k/N^{2/(2+D)} \to 0$ and $k \to \infty$, we can get $|\hat{f}(\mathbf{x}) - f(\mathbf{x})| \lesssim f(\mathbf{x})/\sqrt{k}$ [41].

In practice, K should be dataset dependent. For example, if the Euclidean distance is used, K should be sufficiently small such that the Euclidean distance is a good measure of the distance between two close data points even the data lie in a manifold. If the number of clusters is small, K should increase to prevent generating too many local maxima. We therefore conducted an empirical study of the influence of K and α on clustering the data in Fig. 4.1, for which $N = 240$ (Fig. 4.3-4.4). First, when $K = \log_2(N) = 8$, `densityCut` correctly detected the two clusters given different values for α (Fig. 4.3). Small $K = \log_2(N) = 4$ produced 'spiky' density estimates and resulted in many local maxima (Fig. 4.3). On the contrary, large K produced flat density estimates, and the two true clusters tended to merge because of no deep valley between them

(Fig. 4.3). We therefore used a default value of $K = \log_2(N)$. In addition, when $\alpha = 0.9$ or 0.99, `densityCut` correctly detected the two clusters given different values for K. Increasing α produced better clustering results but it took much longer for the density refinement step to converge, e.g., median 176 iterations when $\alpha = 0.99$ compared to 41 iterations when $\alpha = 0.90$. We set the default value for $\alpha = 0.90$ as it made a good compromise between accuracy and execution time.

The valley height adjustment step plays a role of 'smoothing' the density estimates. This functionality is especially useful for small K. For example, even when $K = 0.5\log_2(N) = 4$, `densityCut` correctly detected the two clusters after adjusting the heights of valleys (Fig. 4.4). For all the results presented in the paper, we used the default parameter setting ($K = \log_2(N)$ and $\alpha = 0.9$) with the valley height adjustment step.

4.2.7 Comparing clustering

Three kinds of measures have been developed in the literature to assess the similarity between two clusterings [140, 217]. The first type of measure is based on set overlaps, i.e., to match two clusterings such that the absolute or relative overlap is maximized. The second type of measure originates in information theory and is based on mutual information, i.e., our knowledge about one clustering increasing when we are told the other clustering. The third type of measure is based on counting pairs, i.e., to consider $\binom{N}{2}$ pair of decisions of assigning a point from clustering one and a point from clustering two to separate clusters or the same cluster. In this study, we used three measures, one from each type to compare clusterings. In addition, we can calculate these measures between a clustering and the ground truth (if available) to assess the accuracy of the clustering.

We first introduce the notations used in defining different measures. Let a clustering $\mathcal{C} = \{C_1, \ldots, C_L\}$ is a partition of the dataset $\mathcal{D} = \{\mathbf{x}^i\}_{i=1}^N$ into L mutually disjoint subsets $C_i, i \in 1, \ldots, L$. Let $\mathcal{C}' = \{C'_1, \ldots, C'_R\}$ denotes a second clustering of $\mathcal{D}$. Let element $M_{i,j} = |C_i \cap C'_j|$ denotes the $(i,j)^{th}$ entry of the confusion matrix $\mathbf{M}^{L\times R}$ between $\mathcal{C}$ and $\mathcal{C}'$. In other words, $M_{i,j}$ denotes the number of points that are common in cluster C_i and C'_j.

The maximum-matching measure (MMM) [140, 217] is based on set overlaps, and it is a generalization of the accuracy in classification applications. It defines a mapping between $\mathcal{C}$ and $\mathcal{C}'$, such that the sum of the number of common points between $\mathcal{C}$ and $\mathcal{C}'$ ($\bar{M}$) is maximized, under the constraint that only one entry in $\mathcal{C}$ can match one entry in $\mathcal{C}'$. Then the MMM

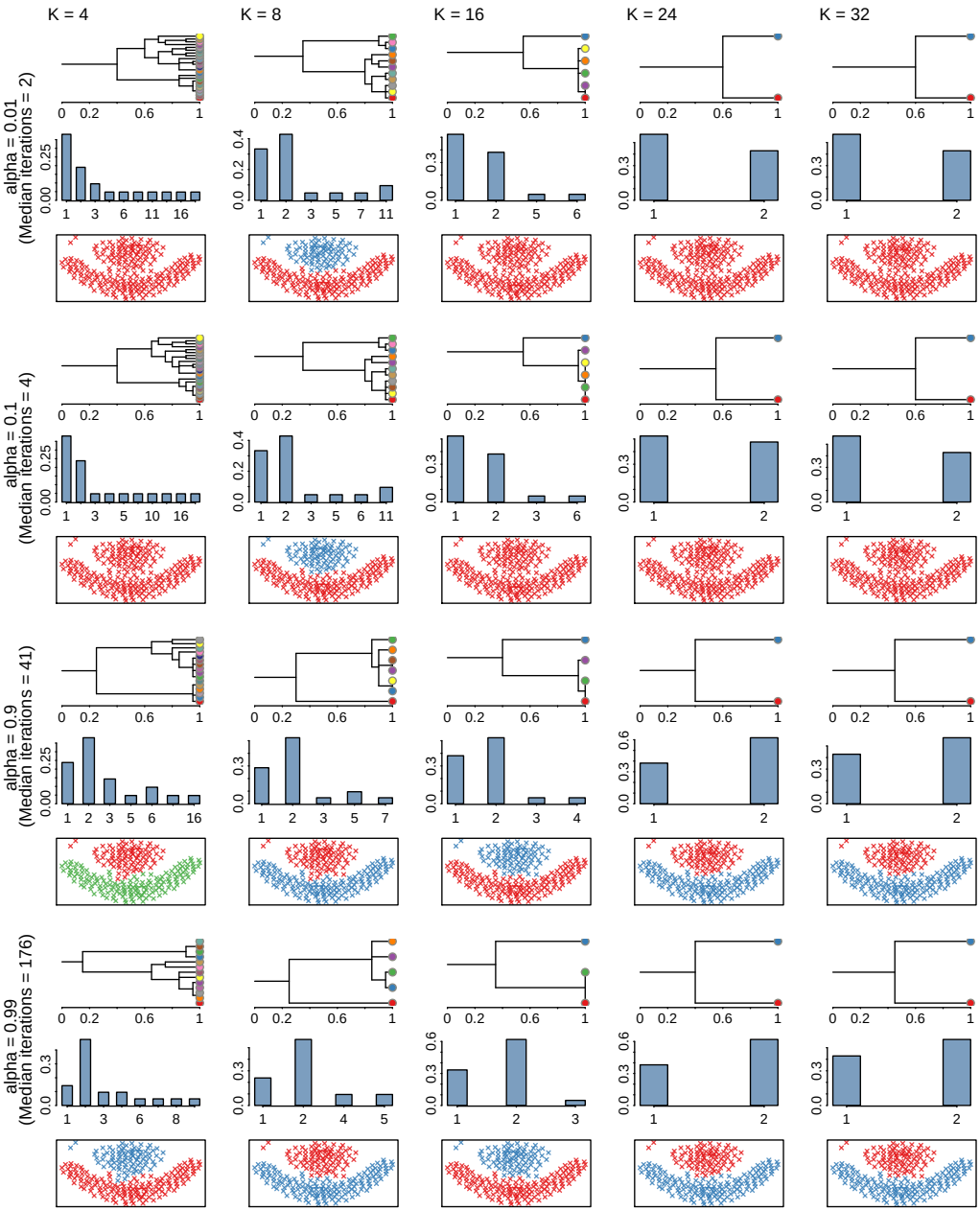


Figure 4.3: The influence of densityCut parameter K and α on the final clustering results. When $K = \log_2(N) = 8$, `densityCut` correctly detected the two clusters given different values for α. Small $K = \log_2(N) = 4$ produced 'spiky' density estimates and resulted in many local maxima. Large K produced flat density estimates, and the two true clusters tended to merge because of no deep valley between them. In addition, when $\alpha = 0.9$ or 0.99, `densityCut` correctly detected the two clusters given different values for K. Increasing α produced better clustering results but it took much longer for the density refinement step to converge, e.g., median 176 iterations when $\alpha = 0.99$ compared to 41 iterations when $\alpha = 0.90$.

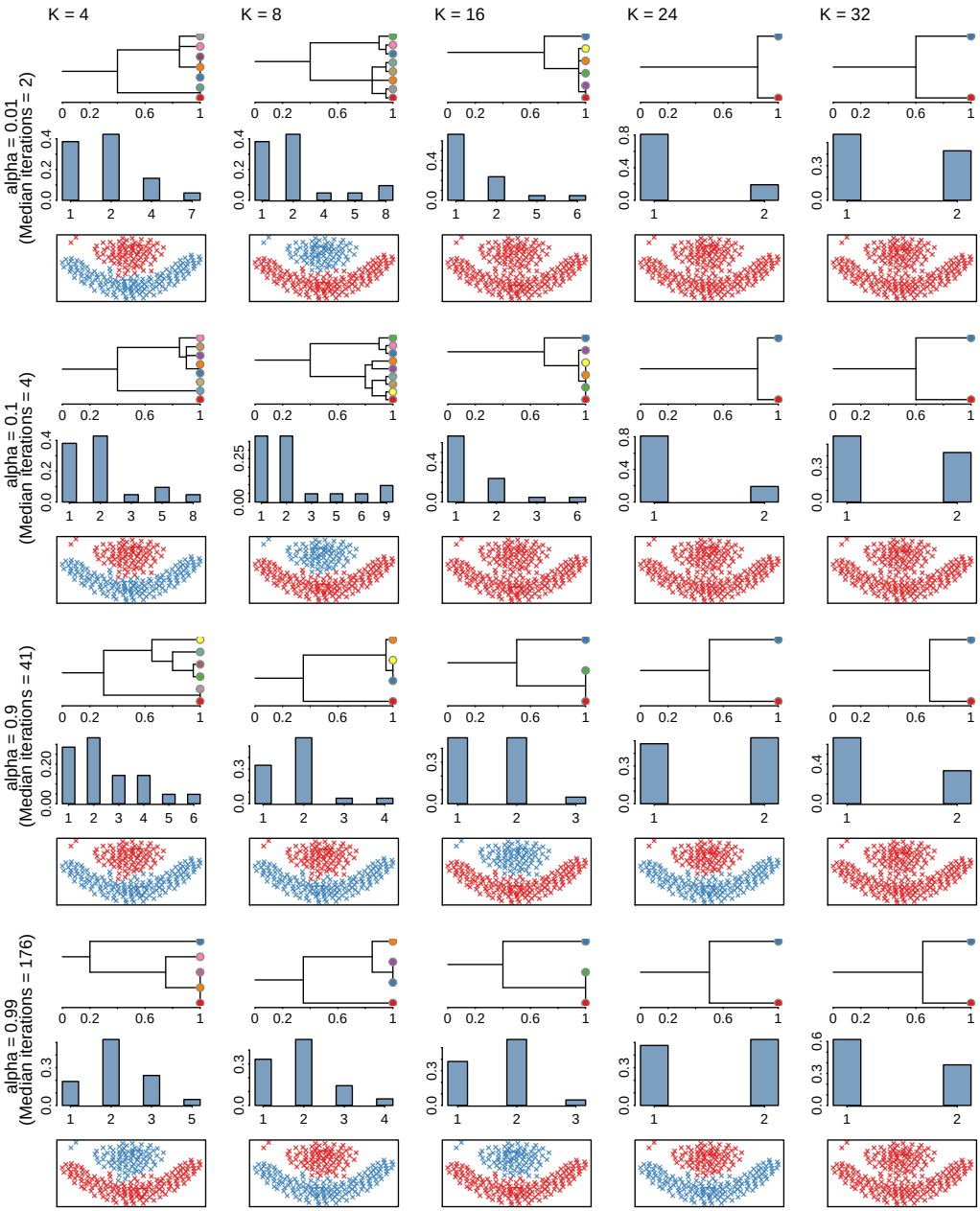


Figure 4.4: The influence of `densityCut` parameter K and α on the final clustering results (with the valley height adjustment step). The valley height adjustment step plays a role of smoothing the density estimates. This functionality is especially useful for small K. For example, when $K = 0.5 \log 2(N) = 4$, densityCut correctly detected the two clusters after adjusting the heights of valleys.

between $\mathcal{C}$ and $\mathcal{C}'$ is defined as:

$$\text{MMM}(\mathcal{C}, \mathcal{C}') = \frac{\bar{M}}{N} \tag{4.11}$$

For perfect match $\text{MMM}(\mathcal{C}, \mathcal{C}') = 1$. However, unlike accuracy for classification, the maximum-matching measure between two random clusterings is not zero. In fact, the minimum maximum-matching measure is $1/N$ under the extrem condition that $L = N$ and $R = 1$.

The normalized mutual information (NMI) between $\mathcal{C}$ and $\mathcal{C}'$ is defined as follows [66]:

$$\text{NMI}(\mathcal{C}, \mathcal{C}') = \frac{2I(\mathcal{C}, \mathcal{C}')}{H(\mathcal{C}) + H(\mathcal{C}')} \tag{4.12}$$

where $H(\mathcal{C}) = -\sum_{i=1}^{L} \frac{|C_i|}{N} \log_2(\frac{|C_i|}{N})$ is the entropy associated with clustering $\mathcal{C}$. The mutual information between clustering $\mathcal{C}$ and $\mathcal{C}'$ is defined as $I(\mathcal{C}, \mathcal{C}') = \sum_i \sum_j \frac{|C_i \cap C_j|}{N} \log_2(\frac{|C_i \cap C_j|/N}{|C_i|/N * |C_j|/N})$. The normalized mutual information is a number between 0 and 1. For perfect match, $\text{NMI}(\mathcal{C}, \mathcal{C}') = 1$, and $\text{NMI}(\mathcal{C}, \mathcal{C}') = 0$ if the joint distribution $P_{i,j} = \frac{|C_i \cap C_j|}{N}$ is independent. Because of the strong independent requirement, the NMI between two random clusterings is typically a small number but not zero.

The adjusted Rand index (ARI) compares pair of assignments form $\mathcal{C}$ and $\mathcal{C}'$, and is defined as:

$$\text{ARI}(\mathcal{C}, \mathcal{C}') = \frac{\sum_{i=1}^{L} \sum_{j=1}^{R} \binom{M_{i,j}}{2} - t_3}{(t_1 + t_2)/2 - t_3} \tag{4.13}$$

where $t_1 = \sum_{i=1}^{k} \binom{|C_i|}{2}$, $t_2 = \sum_{j=1}^{l} \binom{|C'_j|}{2}$, and $t_3 = t_1 t_2 / \binom{N}{2}$. Compared to MMM and NMI, ARI has been corrected for chance, i.e., $\text{ARI}(\mathcal{C}, \mathcal{C}') = 0$ when the elements of the confusion matrix $\mathbf{M}$ follow a generalized geometric distribution (the two clusterings $\mathcal{C}$ and $\mathcal{C}'$ are picked at random, subjected to having the original number of elements in each cluster [93]. In other words, the marginal distributions of the confusion matrix $\mathbf{M}$ are the same as the originals.) An undesired property of the adjusted Rand index is that negative values can occur. For perfect match, $\text{ARI}(\mathcal{C}, \mathcal{C}') = 1$.

4.2.8 Comparing algorithms

We compared densityCut with three best algorithms reported in [229], i.e., the hierarchical clustering algorithm (HC, from the R stats package) with average linkage, the partitioning around medoids (PAM, from the R cluster package) algorithm, and the density-based clustering algorithm OPTICS [7] (from the R dbscan package). Notice that in [229], two density-based algorithms (DBSCAN [57] and clusterdp [171]) were tested and showed good performance. Cur-

rently the clusterdp algorithm needs some human interactions to select the cluster centers, and unfortunately there is no agreed way to automatically set this parameter. Similarly, DBSCAN is very sensitive to the parameter epsilon, which is the radius used to define the neighbours for each data point. We therefore used the OPTICS algorithm, which is similar to DBSCAN, but is more robust because essentially there is no need to set the epsilon parameters. We extracted clusters from OPTICS outputs based on the methods of Sander et al, 2003 [178]. The points considered as outliers by OPTICS were assigned to other clusters by a K-nearest neighbour classifier (where K is the same as the MinPts parameter of OPTICS).

We did not compare densityCut to one of the best clustering tools reported in [229], transitivity clustering [228], because we could not find an easy to use software package for clustering large datasets represented as matrices. We also compared densityCut with the Gaussian mixture model (GMM, implemented in the R mclust package [63]) based clustering algorithm and the normalized cut (NCut, implemented in the kernlab package [239]) spectral clustering algorithm. These algorithms generally represent broad classes of methods for clustering analysis (i.e., hierarchical, partition, density-based, model-based, and graph-based) [6]

4.3 Results

4.3.1 Benchmarking against state-of-the-art algorithms

Synthetic datasets

We used ten synthetic datasets in our study (downloaded from `http://cs.joensuu.fi/sipu/datasets/`). The fourth column figures in Fig. 4.5 shows these synthetic datasets: Aggregation [76], Compound [238], Flame [68], Spiral [31], Jain [95], Pathbased [31], R15 [212], D31 [212], S3 [65], and S4 [65]. The number of data points in each dataset is 788, 399, 240, 312, 373, 300, 600, 3100, 5000, and 5000, respectively. Seven out of the ten datasets have been used in [229] to compare various clustering algorithms (except for Jain, D31, and S4).

As shown in Fig. 4.5, `densityCut` performed the best in terms of the above evaluation measures (with mean MMM, NMI, and ARI of 0.911, 0.897, and 0.854). Overall, the clustering results on these synthetic benchmark datasets demonstrated that `densityCut` can produce excellent results if the high-density clusters were separated by low-density valleys. For the datasets in Fig. 4.5(a,c-e, g-j), `densityCut` detected the right number of clusters and revealed the structures of these datasets. The red colour cluster in Fig. 4.5(b) was considered as two separated

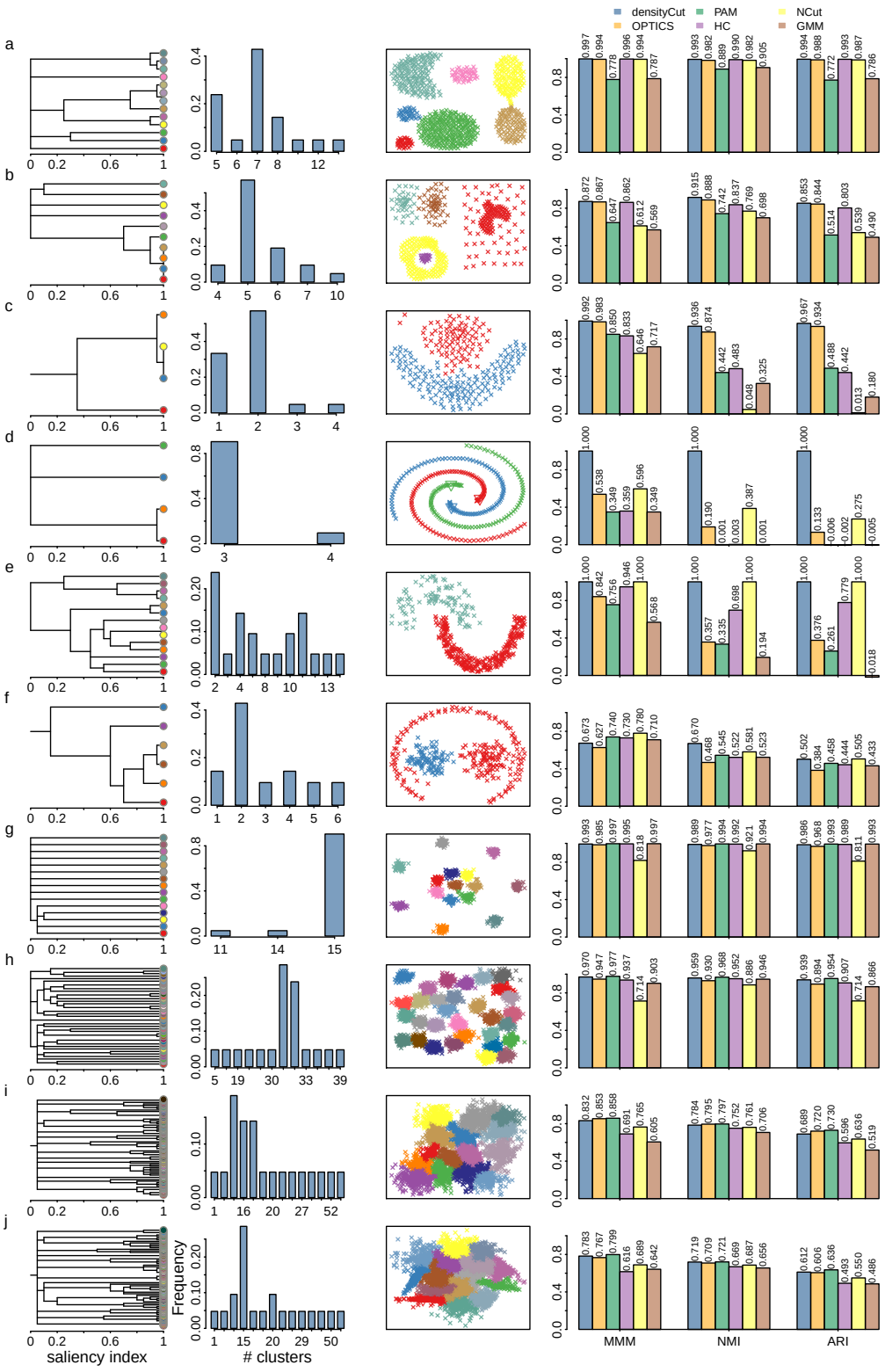


Figure 4.5: Results on the synthetic benchmark datasets consisting of irregular, non-convex, or overlapped clusters. The first-column figures show the clustering trees, the second-column figures show the cluster number frequency plots, the third-column figures show the final clustering results, and the fourth-column figures show the maximum-matching measure (MMM), the normalized mutual information (NMI), and the adjusted Rand index (ARI) from comparing clustering results of each algorithm to the ground truth.

clusters originally [238]. However, the sparse background points and the centre high-density points could be considered as from the same cluster for density-based clustering. For the dataset in Fig. 4.5(f), `densityCut` failed to detect the three clusters because there were no 'deep' valleys between the outer arc cluster and the two Gaussian clusters. Therefore, the outer arc cluster got merged to the right Gaussian cluster. Without the valley height adjustment step, `densityCut` generated the same clustering. The density-based clustering algorithm OPTICS ranked second with mean MMM, NMI, and ARI of 0.840, 0.717, and 0.685, respectively (Fig. 4.5, last column).

PAM and GMM performed poorly on the datasets consisting of irregular shape clusters (Fig. 4.5 (a-f), last column). PAM results on these datasets had mean MMM, NMI, and ARI of 0.687, 0.492, and 0.414, respectively, and GMM results on these datasets had mean MMM, NMI, and ARI of 0.617, 0.441, and 0.311, respectively. However, PAM did very well on the datasets where the points in each cluster were sampled from a two-dimensional Gaussian distribution with mean MMM, NMI, and ARI of 0.908, 0.870, and 0.828 (Fig. 4.5 (g-j), last column). In contrast, GMM performed inferior to PAM on these datasets with mean MMM, NMI, and ARI of 0.787, 0.826, and 0.716. For example, for 'D31' in Fig. 4.5(h), cluster two consisted of only a single data point, and cluster nine consisted of two data points. Cluster three and 18 consisted of points from multiple Gaussian distributions. One major reason for the failure of GMM was that the relatively large number of clusters (31) compared to the limited number of data points (3100) and the overlapped clusters resulted in many local maxima in its objective log-likelihood function, while the Expectation-Maximization algorithm for fitting GMM only searched for a local maximum of the objective function. By contrast, `densityCut` directly located the high-density peaks and selected the most stable clustering, and thus it was less likely influenced by spurious density peaks.

Both HC and NCut can cluster datasets consisting of arbitrary shape clusters. On the datasets consisting of non-convex shape clusters, HC (with mean MMM, NMI, and ARI of 0.788, 0.589, and 0.577) and NCut (with mean MMM, NMI, and ARI of 0.771, 0.628, and 0.55) performed slightly better than PAM and GMM (Fig. 4.5 (a-f), last column). On the datasets consisting of convex shape clusters, HC (with mean MMM, NMI, and ARI of 0.810, 0.841, and 0.746) and NCut (with mean MMM, NMI, and ARI of 0.746, 0.814, and 0.678) performed slightly worse than PAM and GMM (Fig. 4.5 (g-j), last column). Compared to HC and NCut, `densityCut` performed better in both the datasets consisting of arbitrary shape clusters (with mean MMM, NMI, and ARI of 0.922, 0.919, and 0.886) and the datasets consisting

of convex shape clusters (with mean MMM, NMI, and ARI of 0.895, 0.863, and 0.807). Moreover, `densityCut` had low complexity and was scalable to large datasets.

Microarray gene expression data

Gene expression data have been used to stratify cancer patients into biologically or clinically meaningful subtypes, e.g., different subtypes of patients have distinct prognosis. Here we tested `densityCut` on the two microarray gene expression datasets as in [11]. The first microarray gene expression dataset consists of the expression of 1543 genes from four types of lung cancer tissues (186 snap-frozen tumours) and 17 normal lung tissues [19]. These lung tumours include 139 adenocarcinomas, 21 squamous cell lung carcinomas, 20 pulmonary carcinoids, 6 small cell lung cancer. This dataset was downloaded from `http://bioinformatics.rutgers.edu/Static/Supplements/CompCancer/Affymetrix/bhattacharjee-2001/bhattacharjee-2001_database.txt`.

The second microarray gene expression dataset consists of the expression of 182 genes from the mixture of breast cancer tissues and colon cancer tissues [36]. The breast tumours consist of 32 pairs of snap-frozen tumours and the corresponding preserved tumours, and the colon tumours consist of 20 pairs of snap-frozen tumours and the corresponding preserved tumours. This dataset was downloaded from `http://bioinformatics.rutgers.edu/Static/Supplements/CompCancer/Affymetrix/chowdary-2006/chowdary-2006_database.txt`.

The results in Fig. 4.6 show that `densityCut` performs the best on these two datasets (with MMM, NMI, and ARI of 0.926, 0.780, and 0.853 on dataset one, and 0.981, 0.860, and 0.924 on dataset two) compared with PAM, HC, OPTICS, NCut, and GMM. Although OPTICS produced good results on the previous two dimensional synthetic datasets, it performed poorly on these high-dimensional gene expression datasets (ten dimensions are considered as high dimensions for density-based clustering in [110]). OPTICS produced just one cluster for each dataset. Although the absolute distances between data points are not discriminative in high-dimensional spaces (the curse of dimensionality, the distances between any two points are approximately the same), the relative distances (the order of closeness) could still be meaningful, and could be captured by the Knn graph. `densityCut` explores the topology of the Knn graph thus performed better on high-dimensional spaces than OPTICS. GMM (with MMM, NMI, and ARI of 0.626, 0.621, and 0.408 on dataset one, and 0.962, 0.765, and 0.850 on dataset two) and PAM (with MMM, NMI, and ARI of 0.714, 0.583, and 0.411 on dataset one, and 0.952, 0.727, and 0.815 on dataset two)

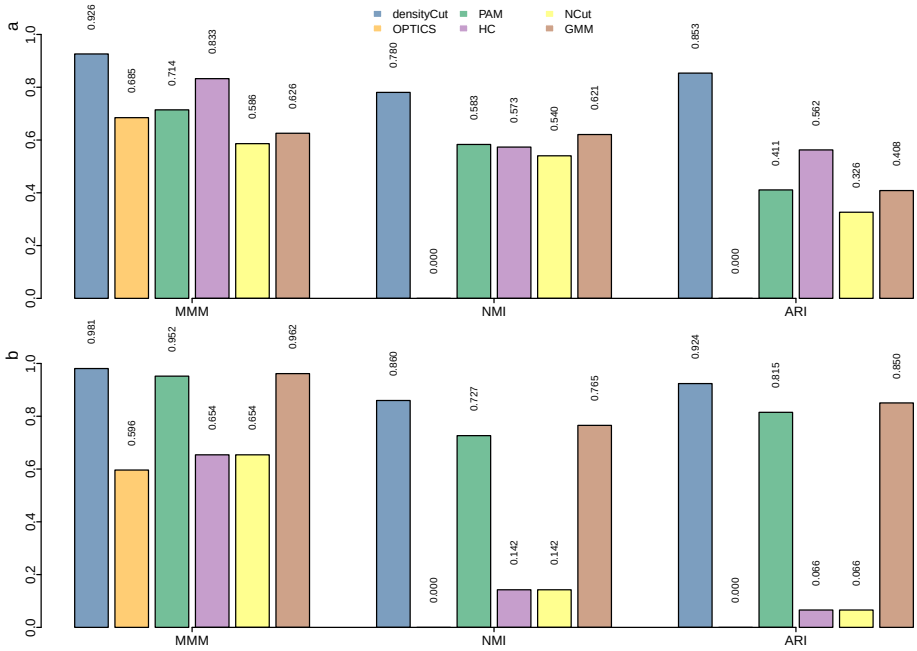


Figure 4.6: Clustering microarray gene expression data. Clustering results on (a) the lung cancer dataset, and (b) the mixture of breast cancer and colon cancer dataset.

performed relatively well on these datasets. The results were consistent with the results from the previous study that GMM and PAM (more precisely, the k-means algorithm) performed well on clustering gene expression data [43].

4.3.2 Inferring clonal architectures of individual tumours

Cancer cells are heterogeneous, and a subpopulation of cancer cells of the same patient could harbour different sets of mutations [144]. Moreover, cancer cells frequently accumulate additional mutations after treatment or in metastasis. Understanding the clonal architecture of each tumour provides insights into tumour evolution and treatment responses. We used `densityCut` to cluster the somatic variant allele frequencies (VAF) measured from DNA sequencing of multiple tumour biopsies. The mutations in each cluster were accumulated during a specific stage of clonal expansion. The clustering results provide valuable information of the clonal architectures of tumours.

We first tested `densityCut` on the mutation data from a primary myelofibrosis (PMF) patient [56]. This patient was first diagnosed with PMF, and seven years later, this patient's tumour transformed to acute myeloid leukaemia (AML). After chemotherapy treatment, the

patient underwent complete remission. However, 1.5 years later, the patient redeveloped PMF but no evidence of AML relapse. A total of 649 single nucleotide variants detected in whole genome sequencing of either PMF, AML, or relapse PMF genomes were validated by targeted high-coverage sequencing. We used `densityCut` to jointly cluster the targeted sequencing VAFs from PMF, AML, and relapse PMF tissues. Figure 4.7(a) shows that `densityCut` grouped the mutations into four clusters.

Overall, `densityCut` clustering results matched those presented in the original study. However, to produce the results, the authors [56] used different algorithms and several pre-processing steps. For example, the authors used DBSCAN [57] to detect outliers (the mutations with circles '$\circ$' in Fig. 4.7(a)), and then used Mclust [63] for model selection and final clustering analysis. The maximum number of clusters was limited to four, and each cluster had to contain at least seven mutations [56]. In contrast, we directly used `densityCut` to cluster the VAFs and produced exactly the same results (MMM=1, the outliers were not considered in calculating MMM.) We also changed the parameter K from the default $\log_2(N) = 10$ to $2\log_2(N)$ until $10\log_2(N)$. Only after K was set to $8\log_2(N)$, the red colour cluster and the violet colour cluster got merged, as can be seen from Fig. 4.7(b). For $K < 8\log_2(N)$, `densityCut` produced the same four clusters. OPTICS, PAM, HC, NCut, and GMM produced the same results as `densityCut` results (to be exact, only PAM assigned one point to different clusters, Fig. 4.8(c)). However, except for OPTICS, these algorithms either need the number of clusters as input parameters or cut the dendrogram to produce the desired number of clusters (HC).

Next, we tested `densityCut` on the acute myeloid leukaemia sample AML28 [53]. We jointly clustered the VAFs from sequencing both the primary tumour and the relapse tumour after 26 months of chemotherapy [53]. Figure 4.7(c) shows that `densityCut` grouped the 804 detected somatic mutations into five clusters. The results matched those predicted by sciClone [144], a variational Bayesian mixture model based clustering algorithm. Only one mutation (with circle '$\circ$' in Fig. 4.7(c)) was assigned to the red cluster by `densityCut`, but originally assigned to the cyan cluster by sciClone (MMM=0.999). `densityCut` is much more efficient than sciClone as can be seen from Fig. 4.7(d). sciClone took a median of 48.12 seconds to run on the AML28 dataset while `densityCut` took a median of 0.074 second to run. Because it took less than a CPU second to run `densityCut`, we ran both algorithms ten times to get a more accurate estimation of the time used. For competing algorithms, PAM split the large cluster into two clusters because it tends to generate equal-size clusters (with MMM, NMI, and ARI of 0.619, 0.509, and 0.767;

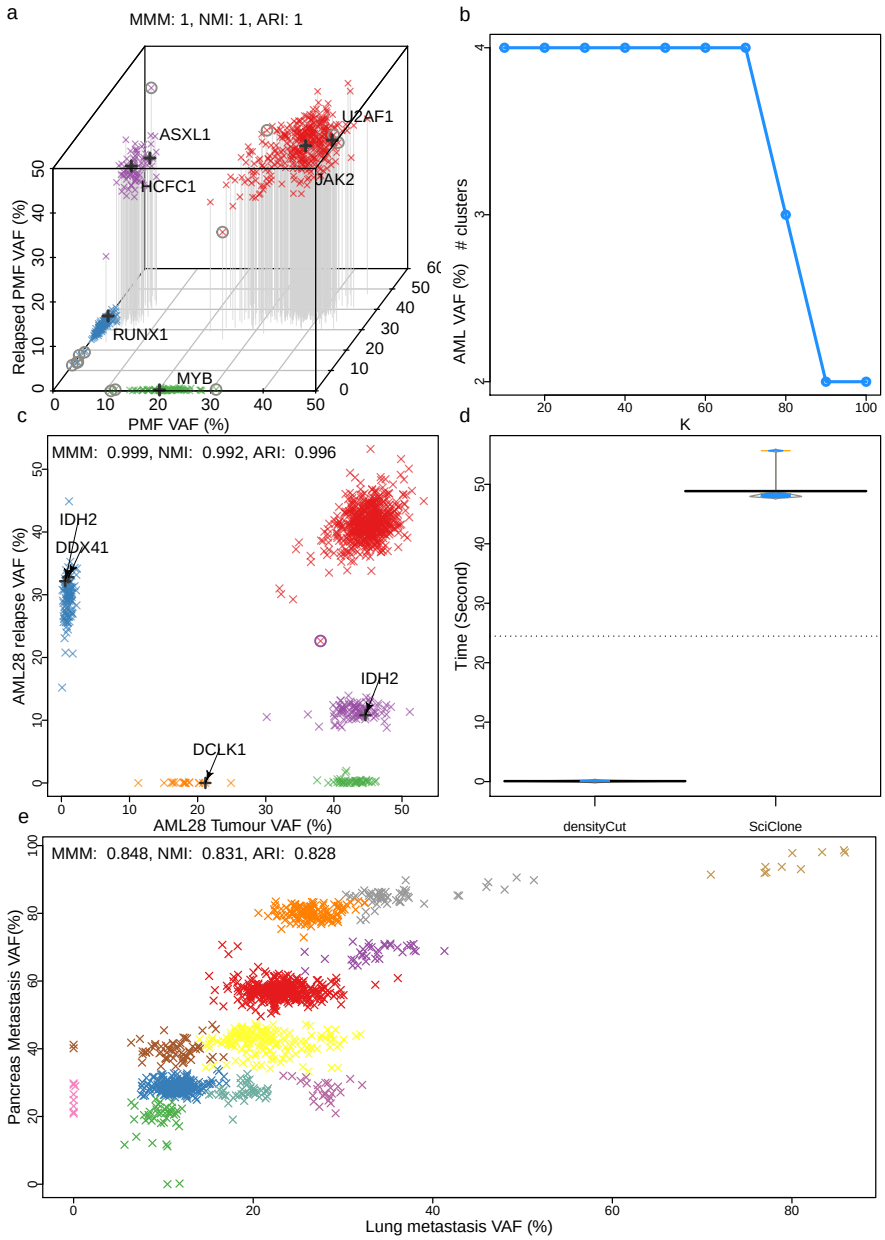


Figure 4.7: Clustering variant allele frequencies (VAF) of somatic mutations. (a-b) Clustering multi-time sample data from initial primary myelofibrosis (PMF), later acute myeloid leukaemia (AML), and after treatment relapsed PMF using `densityCut`. (c-d) Clustering the somatic mutations from sequencing a primary/relapse pair of an AML patient. (e) Clustering the somatic mutations from sequencing a lung/pancreas metastasis pair of a melanoma patient. The possible 'driver' mutations in each cluster are labeled with a black plus sign '+'. The clustering validation indices (MMM, NMI, and ARI) were from comparing `densityCut` results with sciClone results or the results reported in the original studies. (a) Three-dimensional VAF plot. The mutations in each cluster were assigned a unique colour. The mutations with a circle '○' were considered as outliers in the original publication [56] before clustering analysis. (b) The number of clusters produced by `densityCut` as we gradually increased K from $\log_2(N)$ to $10\log_2(N)$. (c) The mutation assigned to the violet colour cluster by sciClone but assigned to the red colour cluster by `densityCut` was labeled with a circle '○'. (d) `densityCut` and sciClone execution time based on repeated ten runs.

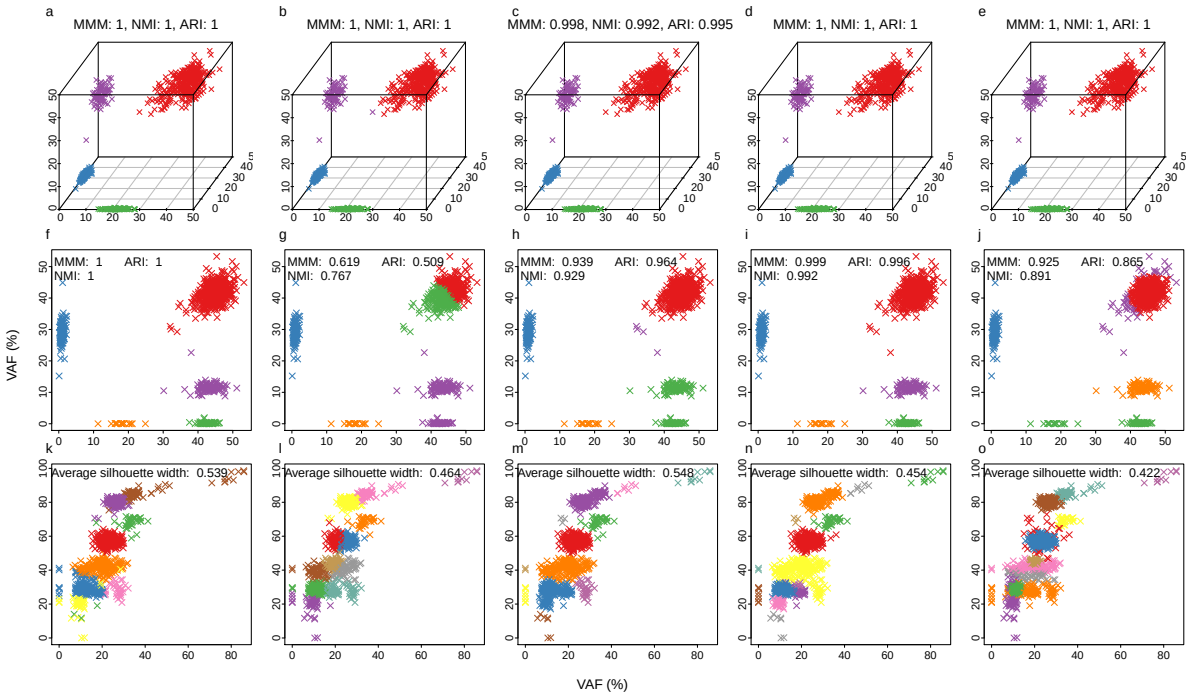


Figure 4.8: Clustering variant allele frequencies of somatic mutations using competing algorithms. (a-e) Clustering multi-time sample data from initial primary myelofibrosis (PMF), acute myeloid leukaemia (AML), and after treatment relapsed PMF. (f-j) Clustering the somatic mutations from sequencing a primary/relapse pair of an AML patient. (k-o) Clustering the somatic mutations from sequencing a lung/pancreas metastasis pair of an melanoma patient. First column figures show the OPTICS clustering results, the second column figures show the PAM clustering results, the third column figures show the HC clustering results, the fourth column figures show the NCut clustering results, and the fifth column figures show the GMM clustering results,.

Supplementary Fig. 4.8(g)). HC assigned some 'outliers' to a distinct clusters, and merged the points from two clusters (with MMM, NMI, and ARI of 0.939, 0.964, and 0.929; Fig. 4.8(h)). Similarly, GMM modelled the outliers using a Gaussian component (with MMM, NMI, and ARI of 0.925, 0.865, and 0.891; Fig. 4.8(j)). OPTICS results differed from `densityCut` results by the assignment of only one data point (Fig. 4.8(f)), and NCut produced the same results as `densityCut` results (Fig. 4.8(i)).

Finally, we used `densityCut` to cluster the somatic mutations from whole genome sequencing of the lung/pancreatic metastasis pair from the same melanoma patient MEL5 [52]. Compared to blood cancer genomes, melanoma genomes are much more complex, frequency harbouring copy number alterations. The combinations of copy number alterations, homozygous mutations, and heterozygous mutations make it a challenging task to develop a model to uncover the clonal

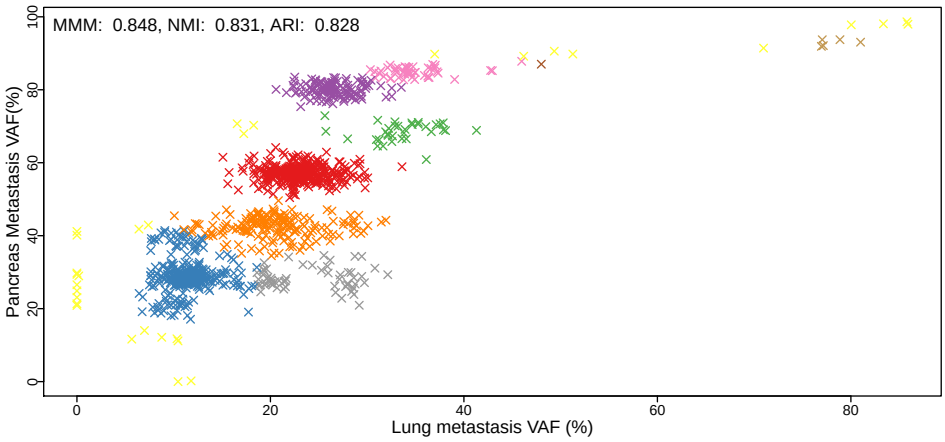


Figure 4.9: Clustering the somatic mutations from sequencing a lung/pancreas metastasis pair of a melanoma patient using sciClone (without considering copy number alterations). The yellow colour cluster may be meaningful in terms of clustering because it models the 'outliers'. However, it may not have biological meaning because the mutations in this cluster could from different clones. The MMM, NMI, and ARI were computed from comparing sciClone results (ten clusters) with the `densityCut` results (12 clusters).

structure of these cancer genomes [175]. The `densityCut` clustering results in Fig. 4.7(e) show that the mutations in MEL5 could be grouped into 12 clusters, providing the starting point for detailed inspection of the clonal structure of this cancer genome. Additional information such as copy number alterations would be required to fully interpret the clonal architectures. We also ran sciClone, which produced ten clusters (Fig. 4.9). Both algorithms agreed in clustering 84.8% of the mutations (MMM: 0.848, Fig. 4.7(e)). `densityCut` clustering had an average silhouette width of 0.58 (Fig. 4.10), which was higher than sciClone clustering average silhouette width of 0.55. Other competing algorithms performed inferrer to `densityCut` with PAM and OPTICS performed second and third with average silhouette widths of 0.548 and 0.539, respectively (Fig. 4.8(m, k)).

4.3.3 Clustering single-cell gene expression datasets

We used `densityCut` to cluster two single-cell mRNA gene expression datasets. The first dataset consists of the low-coverage mRNA expression of 23,730 genes in 301 cells from 11 populations [161]. The second dataset consists of the single-cell mRNA expression of 43,309 genes in 223 stem cells from the subventricular zone of eight-week-old male mice [126]. We did several pre-processing steps to only select a subset of genes [161] for clustering analysis because

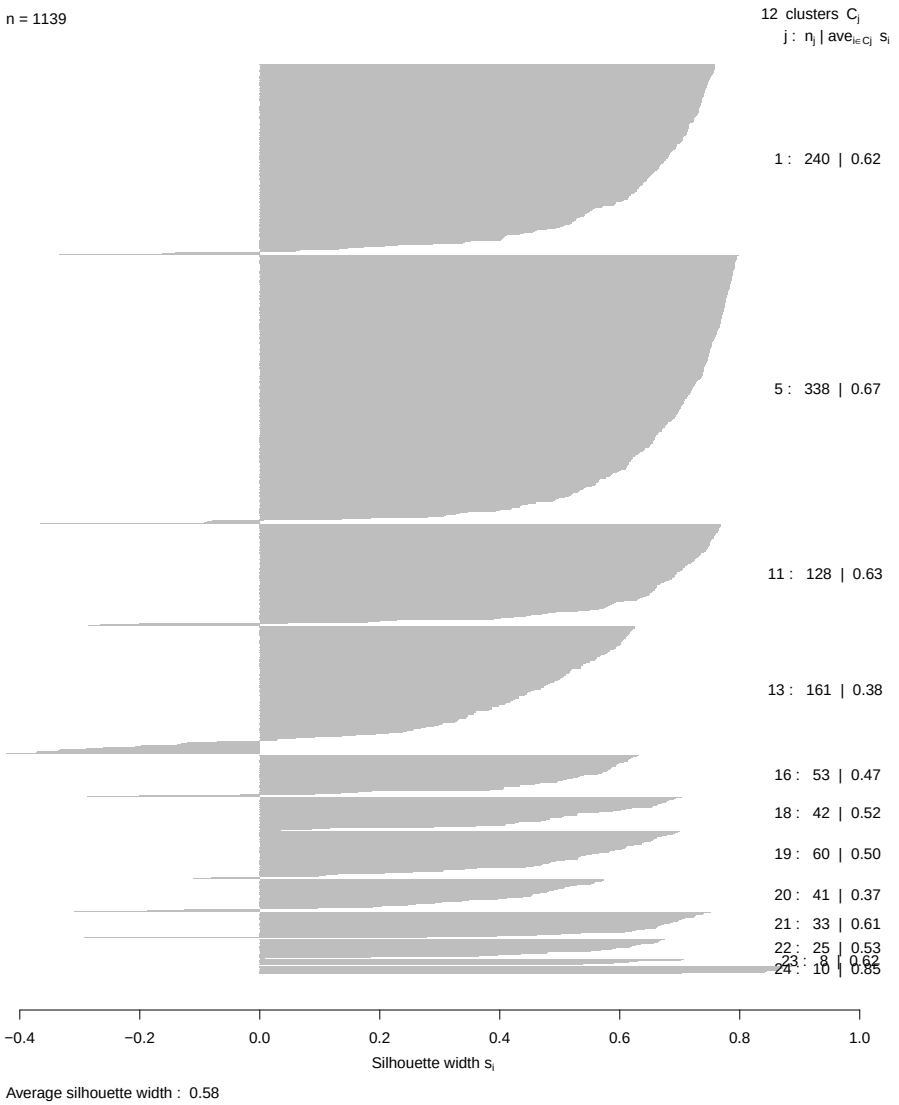


Figure 4.10: Plot of silhouette values from clustering the variant allele frequencies of somatic mutations from sequencing a melanoma lung/pancreas metastasis pair by `dencityCut`. The high average silhouette width of 0.58 suggests that this dataset could contain 12 clusters.

single-cell gene expression data have high technical noises (e.g., loss of cDNA in reverse transcription and bias in cDNA amplification) and Knn search in high dimensional spaces is still time-consuming. Specifically, we only kept the genes expressed in more than five cells because it is difficult to detect clusters less than five in size given the relatively large number of cells. Here a gene was considered to be expressed in a cell if its reads per kilobase per million (RPKM) value (or fragments per kilobase per million (FPKM) value for dataset two [126]) was greater than or equal to one in the cell. We then further normalized the RPKM values by log transformation:

$\log_2(x+1)$. Here x was the original RPKM value of a gene in a cell. A small value of one was added to prevent taking the log of zero or generating very small numbers.

Figure 4.11(a) shows that `densityCut` produced nine clusters for dataset one (MMM: 0.917, NMI: 0.953, and ARI: 0.918). `densityCut` cannot distinguish the cells from GW16, GW21, and GW21.2 based on the 1,000 genes. These cells were quite similar as they were all from the human cortex (GW16 cells were from the germinal zone of human cortex at gestational week 16, GW21 cells were from GW21, and GW21.2 cells were cultured cells of a subset of the GW21 cells [161]). These cells could possibly be separated by selecting a better set of features for clustering analysis. In addition, one GW21 cell was misclassified as a neural progenitor cell (NPC), and one NPC was in the human-induced pluripotent stem (iPS) cell cluster. For the other seven types of cells, `densityCut` perfectly put them into separate clusters. Other clustering algorithms such as OPTICS, PAM, HC, NCut, and GMM had inferior performance compared to `densityCut` results with PAM ranked second with MMM, NMI, and ARI of 0.877, 0.916, and 0.854, respectively (Fig. 4.12(a)).

`densityCut` grouped the 223 stem cells of dataset two into four clusters (Fig. 4.11(b)). Glutamate aspartate transporter$^+$/Prominin1$^+$ (GP) cells and polysialylated-neural cell adhesion molecule$^+$ (PSA) cells were in separate clusters (except for one PSA cell). The GP cells were subdivided into three clusters, consistent with the original finding that the GP cells consisted of at least three subtypes of stem cells. We next used t-SNE [205] to project the 1000-dimensional single-cell gene expression data to a two-dimensional space (Fig. 4.13). The results also show four very distinct clusters. Compared with the original analysis using hierarchical clustering coupled with principle component analysis feature section [126], `densityCut` can be used in a more unbiased way to cluster single-cell gene expression data and produce the same results. On this dataset, `densityCut`, PAM, HC, and GMM results had average silhouette widths of 0.190, 0.190, 0.191, and 0.189, respectively (Fig. 4.12(b); silhouette widths have less discriminative power in high dimensional spaces).

4.3.4 Clustering single-cell mass cytometry datasets

Finally, we used `densityCut` to cluster two benchmark single-cell mass cytometry (aka CyTOF) datasets [120]. The first dataset consists of CyTOF data of bone marrow mononuclear cells from a healthy individual. Manually gating (labelling) assigned 81,747 cells to 24 cell types based on 13 measured surface protein markers [15]. Dataset two contains CyTOF data from two healthy

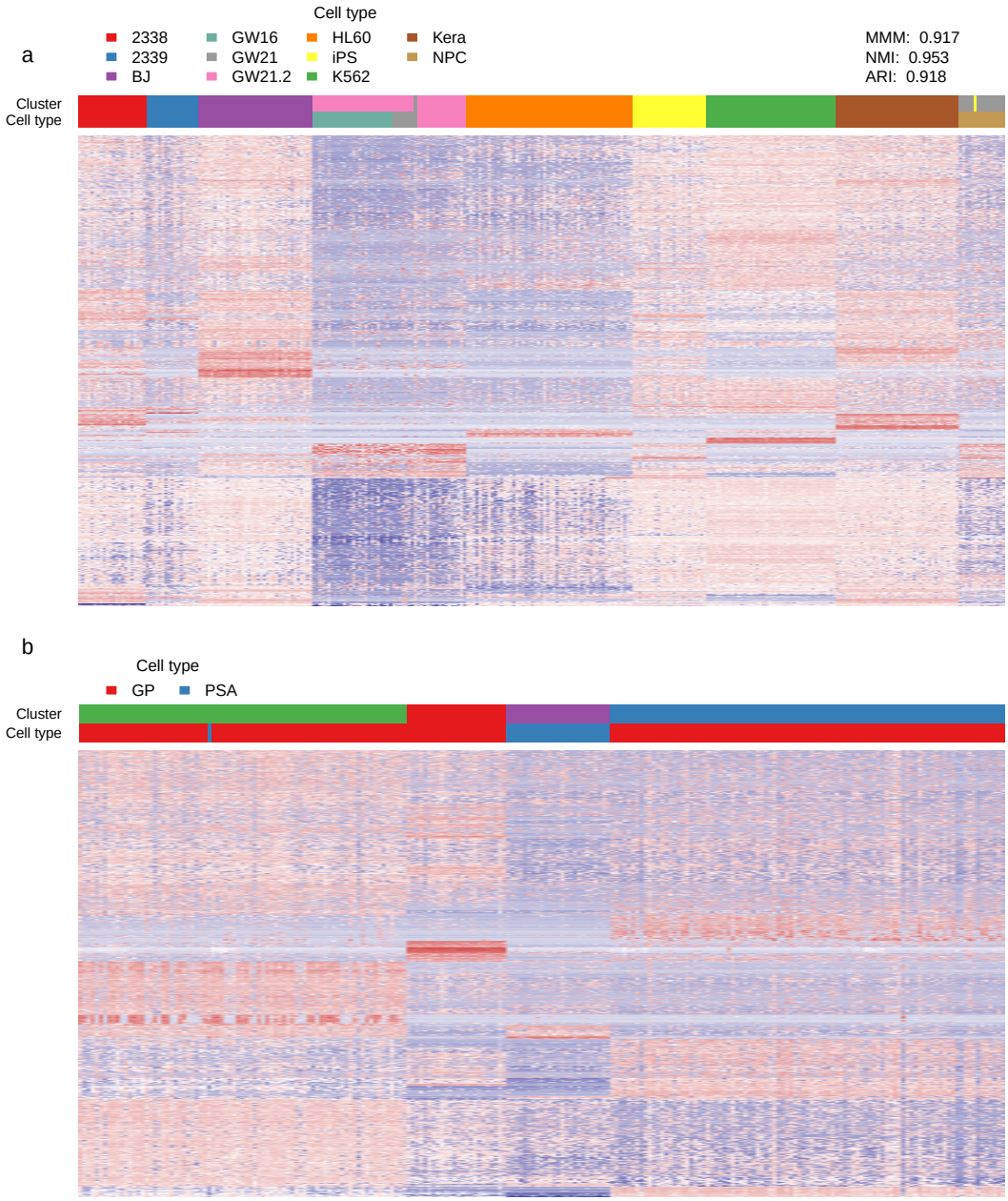


Figure 4.11: Clustering single-cell gene expression data. Each row is a gene and each column is a cell. The cell type and cluster membership of each cell are colour coded. Heatmaps show clustering (a) 301 cells from 11 populations, and (b) 223 stem cells from the subventricular zone of eight-week-old male mice.

adult donors H1 and H2. For H1, manual gating assigned 72,463 cells to 14 cell types based on 32 measured surface protein markers. Manual gating assigned 31,721 cells to the same 14 cell populations from H2 based on the 32 surface protein markers. These manually identified cell populations were used as ground truth to test `densityCut`.

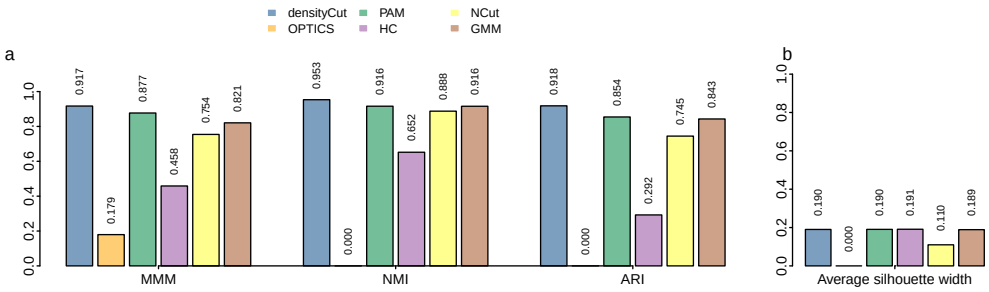


Figure 4.12: Performance measures on clustering single-cell gene expression data. (a) The clustering measures from clustering the gene expression data of 301 cells. (b) Average silhouette widths from clustering the expression data of 223 stem cells from the sub ventricular zone of eight-week-old mice.

We compared `densityCut` to the recently proposed algorithm, the PhenoGraph algorithm [120], in clustering the benchmark single-cell CyTOF datasets. As both `densityCut` and PhenoGraph first build a Knn graph, we used the same $K = \log_2(N)$ for both algorithms. As can be seen from Fig. 4.14, `densityCut` detected 12 distinct cell types (clusters) in dataset one, 9 cell types in H1, and 12 cell types in H2. The PhenoGraph algorithm detected 18, 24, and 20 clusters in dataset one, H1, and H2, respectively. Based on MMM, NMI, and ARI, `densityCut` performed slightly worse on dataset one (MMM: 0.879 vs. 0.883, NMI: 0.878 vs. 0.900, and ARI: 0.857 vs. 0.893), but performed better on H1 (MMM: 0.941 vs. 0.682, AMI: 0.935 vs. 0.833, and ARI: 0.96 vs. 0.669) and H2 (MMM: 0.953 vs. 0.67, NMI: 0.945 vs. 0.829, and ARI: 0.977 vs. 0.638). As for efficiency, `densityCut` was around two times faster than PhenoGraph based on the current implementations (Fig. 4.15). Other clustering algorithms such as PAM, HC, NCut, and GMM are not scalable to these relatively large datasets. For example, we can only run OPTICS on the first dataset (OPTICS took about 17 minutes while `densityCut` took only 24 seconds).

4.4 Conclusions and discussion

We developed `densityCut`, a simple and efficient clustering algorithm. `densityCut` effectively clustered irregular shape synthetic benchmark datasets. We have successfully used `densityCut` to cluster variant allele frequencies of somatic mutations, single-cell gene expression data, and single-cell CyTOF data. `densityCut` is based on density estimation on graphs. It could be considered as a variation of the spectral clustering algorithms but is much more time- and

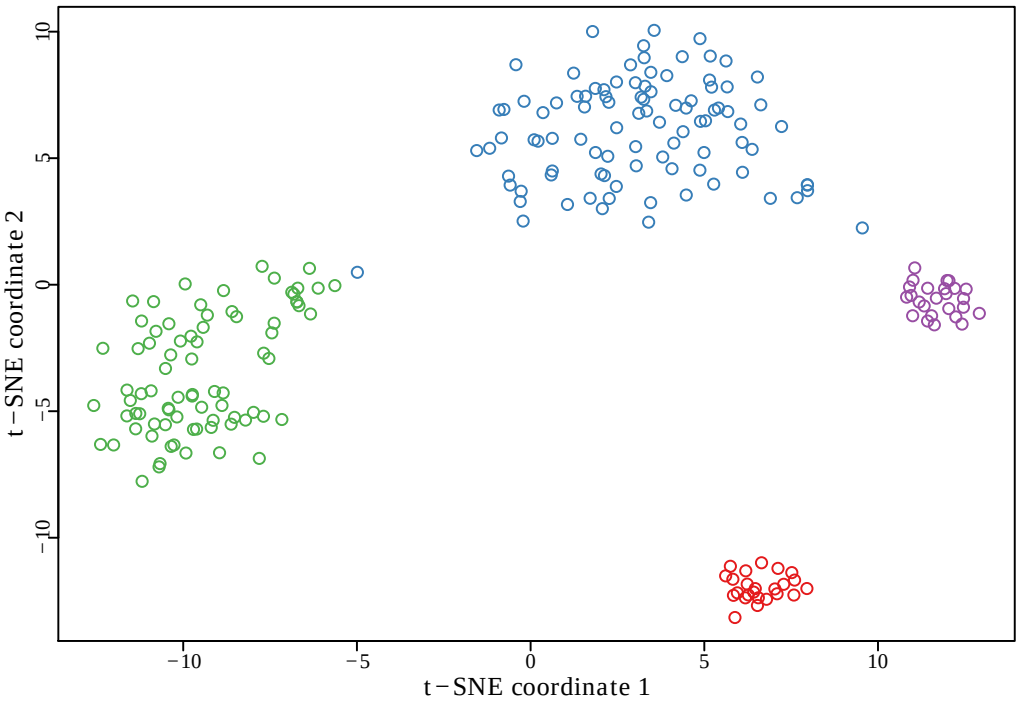


Figure 4.13: Visualizing the mouse brain stem cell expression data by t-Distributed Stochastic Neighbor Embedding (t-SNE). We can see four distinct clusters.

space-efficient. Moreover, it automatically selects the number of clusters and works for the datasets with a large number of clusters. In summary, `densityCut` does not make assumptions about the shape, size, and the number of clusters, and can be broadly applicable for exploratory data analysis.

A recent study has shown that current strategies for whole genome sequencing studies missed many somatic mutations [80]. By increasing the sequencing depths from 30x in their original study [53] to 300x and using a consensus of somatic single nucleotide variant (SNV) callers, the number of identified SNVs increased from 118 to 1343. Based on the 1343 SNVs, they identified two extra sub-clones [80]. Moreover, an additional 2500 SNVs were highly likely to be genuine somatic SNVs but still without enough evidence even at 300x coverage. For more complex genomes such as melanoma and breast cancer genomes, the number of SNVs could be much larger. Therefore, efficient algorithms such as `densityCut` are necessary to infer the clonal structures in individual tumours as more genomes are sequenced at higher coverage in the near future.

In recent years, single-cell techniques have empowered scientists to investigate cellular het-

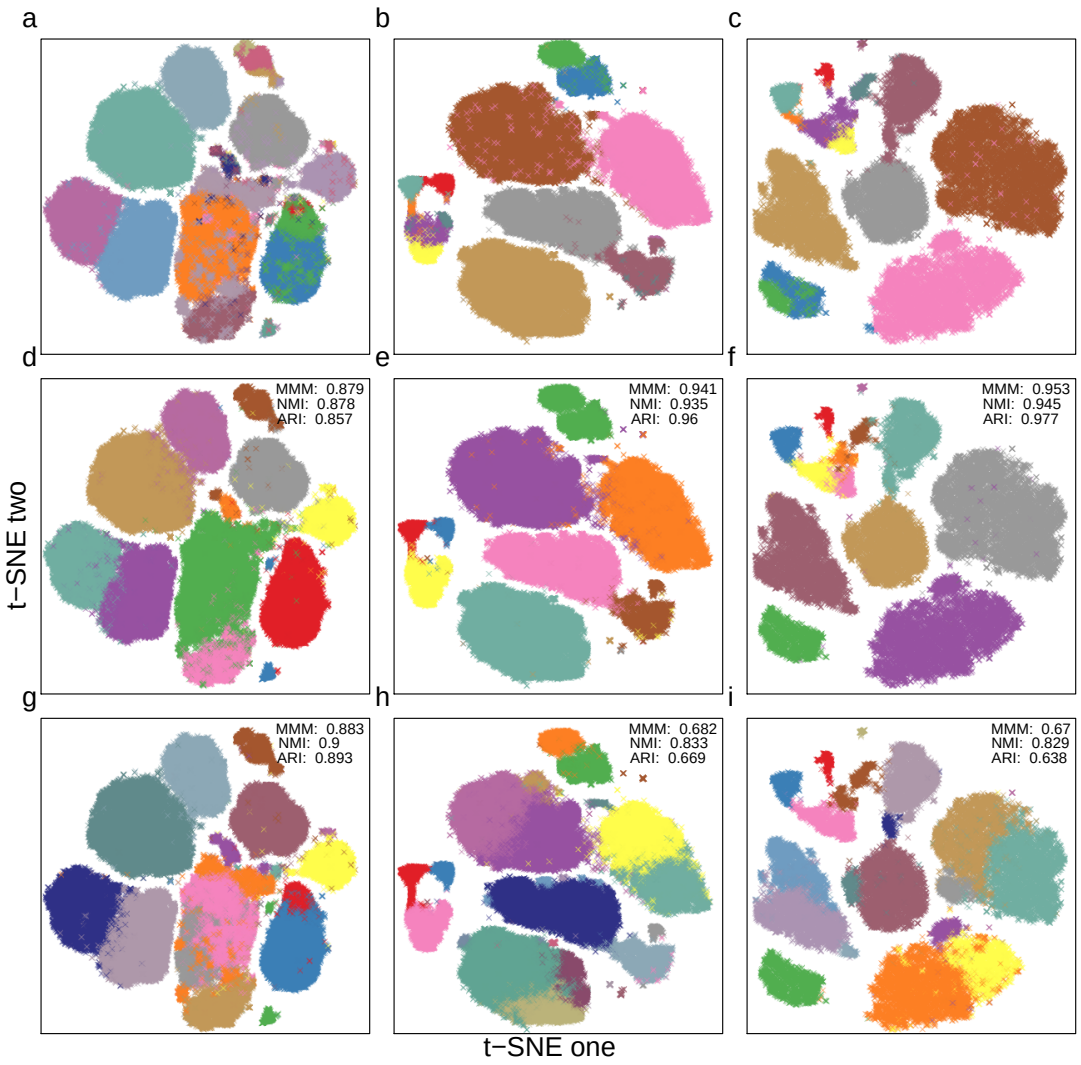


Figure 4.14: Comparing `densityCut` and PhenoGraph in clustering single-cell mass cytometry data. The original high-dimensional mass cytometry data were projected onto two dimensional spaces by t-SNE just for visualization purpose. Cell types and clustering memberships of data points were colour coded. (a-c) The ground truth. (d-f) `densityCut` clustering results. (g-i) PhenoGraph clustering results.

erogeneity. Computational tools are necessary to analyze these single-cell measurements with high dimensionality and large numbers of cells. Efficient algorithms such as `densityCut` whose computational complexities are independent of the dimensionality of data, and can cluster millions of points in a few minutes can be valuable tools to process these datasets to distill single cell biology.

Current technology advances have made it possible to simultaneously measure a tumour from different angles, e.g., mutations, gene expression, and DNA methylation. Each biological

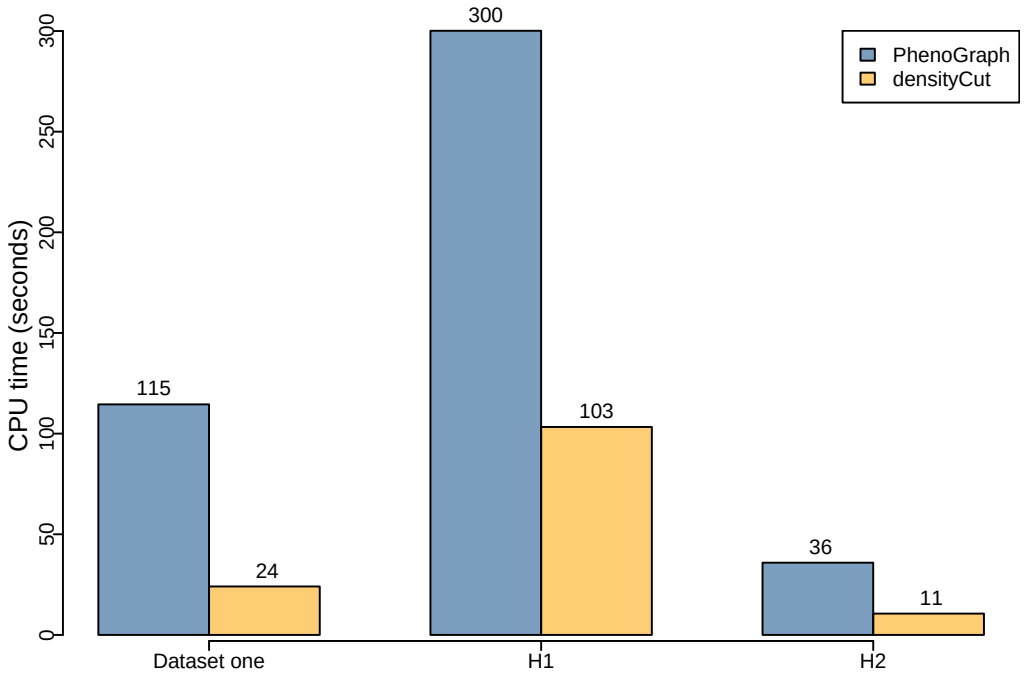


Figure 4.15: Comparing the time used by PhenoGraph and `densityCut` in clustering the benchmark CyTOF datasets. Although directly comparison is difficult since PhenoGraph is implemented in Python and `densityCut` is implemented in R, `densityCut` is around two times faster than PhenoGraph based on the current implementations.

measurement provides a different view of the tumour. For complex diseases such as cancer, it is necessary to combine multi-view datasets to provide a comprehensive view of these diseases. A future extension of `densityCut` is to integrate multi-view data for cancer patient stratification, i.e., putting patients into different groups such that the patients in the same group are similar to each other in molecular features. As `densityCut` works on a K-nearest neighbour graph, we need to effectively construct a combined Knn graph by integrating different datasets.

Currently `densityCut` simply clusters data points without probability information of a point belonging to a cluster. It is possible to a do a 'soft' assignment of a data point to clusters at the expense of efficiency. For example, for a point if its neighbours in a graph have been assigned to different clusters, this point could be assigned to different clusters as well. On the contrary, if all the neighbouring points have the same label, the point could be assigned to the same label with high probability.

Chapter 5

Conclusions

"It is a far, far better thing that I do, than I have ever done;
it is a far, far better rest that I go to than I have ever known."
– Charles Dickens, A Tale of Two Cities, 1859

5.1 Summary of contributions

This dissertation contributes three computational methods in the analyses of cancer systems biology data.

Extendable feature based discriminative classifiers to predict somatic SNVs from tumour/normal DNA sequencing data To predict somatic SNVs from DNA sequencing data is a challenging task because of the various errors introduced during sample preparation, library preparation, or the alignment of short reads to reference genome as well as platform-specific errors. As it is difficult to explicitly model the combinations of all sources of errors, we therefore developed versatile and extendable feature based discriminative classifiers trained on validated somatic SNVs to predict somatic SNVs given paired tumour/normal genome sequencing data. We further categorized the discriminative features in separating somatic SNVs from non-somatic SNVs. We finally discussed some systematic errors in high-throughput sequencing data.

A generative model-based approach to predicting the somatic mutations that impact gene expression To identify the small set of critical mutations that transform a normal cell to a cancer cell is crucial for revealing the biology behind the transformation and designing therapies. We used a hierarchical Bayes approach to model the impacts of a mutation on gene expression in a specific patient and the impacts of all the mutations in a gene across patients on gene expression. By analyzing the TCGA pan-cancer datasets from 12 types of cancer, we identified loss-of-function mutations in 65 genes correlated with expression down-regulation.

Furthermore, mutations in 150 genes correlated with pathway dysregulations. These genes were candidate cancer driver genes for further biological functional studies.

An efficient and versatile algorithm for clustering high-throughput systems biology data In addition to sequencing the whole genomes, current technology advances have made it possible to rapidly and accurately monitor cellular changes, even within a single-cell. The resulting data could possibly be used to stratify patients into clinically similar subgroups, e.g., with similar prognostic values. We developed an efficient and versatile clustering algorithm to put similar objects into the same group. We applied it to uncover the clonal structures in individual patients by clustering mutation variant allele frequencies and to reveal the cell population structures by clustering single-cell gene expression data and single-cell mass cytometry data.

5.2 Conclusions and future work

Advances in high-throughput DNA sequencing technologies have revolutionized cancer genome studies to detect somatic mutations in individual genomes. Additional computational algorithms have proven to be useful to detect somatic mutations and interpret mutation functions [54]. Cancer genome sequencing data typically have specific bias originating from tumour/normal contamination, intratumour heterogeneity, and copy number alterations [82]. All these factors could result in low variant allele frequencies, which are challenging for SNV prediction. Therefore, ideally tumours should be sequenced at a high coverage (e.g, 300x) to detect somatic SNVs [80]. Specific classifiers could be trained to detect low variant allele frequency SNVs. One major bias causing non-uniform coverage in high-throughput sequencing data is the PCR amplification bias, which could be removed by PCR-free sequencing [151]. As new generations of DNA sequencing machines to produce longer reads uniformly covering the whole genome with higher depth, our ability to detect SNVs will be greatly improved.

Although our understanding of the cancer signalling pathways is still fragmented, the pathway knowledge has already been translated to clinical care of cancer patients. By revealing the central role of the phosphoinositide 3-kinase (PI3K)/AKT/mTOR intracellular signalling pathway in regulating cell growth, many inhibitors have been developed to shut down this pathway in cancer. For example, clinical trials have shown the success of a PI3K inhibitor (which inhibits the delta isoform of the PI3K protein - PI3Kδ) to treat B cell cancer [67]. Notice that the inhibitor targets wildtype PI3Kδ because it is not mutated but essential for B cell survive.

The RTK/RAS/MAP-kinase signalling pathway receives extracellular signals from the cell surface and conveys the signals to the nucleus. Cancer cells frequently manipulate this pathway to generate endogenous mitogenic signals. For example, melanoma patients frequently harbour a hotspot missense mutation that changes the $600th$ amino acid valine (V600) of the protein encoded by the *BRAF* gene. In clinical trials, the majority of these patients had partial or complete responses to *BRAF* inhibitors [60]. However, colon cancer patients with exactly the same *BRAF* hotspot mutation did not respond to the same inhibitor. An RNAi knockdown screen pinpoints a feedback activating the cell surface growth factor receptor *EGFR* by inhibiting *BRAF* specificly in colon cancer where *EGFR* is highly expressed [162]. The gain in *EGFR* expression is also one of the reasons why some melanoma patients become resistance to *BRAF* inhibitors [198]. These results demonstrate that the molecular context plays an important role for the success of targeted therapies. Therefore, direct interpretation of the molecular context through systematic incorporation of gene expression data could be valuable for delivering effective targeted treatments to cancer patients.

In contrast, glioblastoma and especially low-grade glioma patients frequently have a hotspot *IDH1* R132 missense mutation [26, 156]. Since *IDH1* mutation in diffuse glioma is an early event in tumorigenesis and shared by all cancer cells [221], it could be an effective drug target as *BCR-ABL* in chronic myeloid leukemia. Since the discovery of *IDH1* hotspot mutations in 2008 [156], we have only partially revealed how the mutant protein contributes to tumorigenesis [61, 200]. *IDH1* encodes an enzyme to catalyze a metabolic process of oxidative decarboxylation of isocitrate to 2-oxoglutarate. *IDH1* mutation results in the massive production of 2-hydroxyglutarate (2-HG), which inhibits DNA demethylation through the TET family enzymes [233]. Eight years after detecting *IDH1* mutations in glioblastoma, Flavahan *et al* provides a mechanistic explanation about how *IDH1* mutations alter metabolism to manipulate DNA methylation and finally to promote cell proliferation through *PDGFRA* up-regulation in glioblastoma: *IDH1* mutant elevates methyl groups which prevent the isolator protein CTCF binding to DNA thus alters the organization of DNA [61, 81]. Altered DNA structure could change gene expression, e.g., up-regulation of *PDGFRA* from gene enhancer-promoter binding (enhancer of gene *FIP1L1* binds to the promoter of *PDGFRA*). Interestedly, `xseq` found the correlation between *IDH1* mutations and *PDGFRA*, *TET2* dysregulation from the STRING functional protein interaction network (Fig. 5.1). These results suggest that although detailed pathway information is not available, integrated analysis of multiple 'omics' datasets such as mutation, expression, and protein inter-

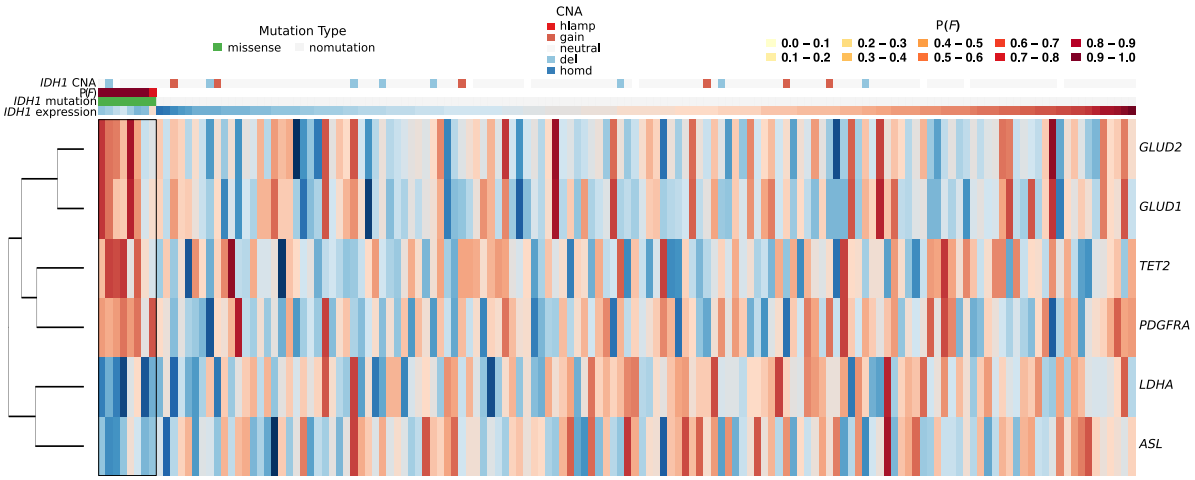


Figure 5.1: Genes connected to *IDH1* and dysregualted in *IDH1* mutated glioblastoma patients. The functional protein interaction network was from the STRING database.

action network data could help reveal part of the cancer pathways, and thus generate hypotheses for designing validation experiments.

Current targeted treatments (various protein kinase inhibitors and immunotherapies) do not produce durable responses in all patients. These results reflect our partial or fragmented understanding of the cancer cell signalling pathways. On the one hand, even for very aggressive metastasized tumours such as melanoma (with the *BRAF* hotspot mutations), we already can stop cancer in its tracks (at least transiently), sometimes even by an agent to target a single gene [213]. On the other hand, drug resistance [77] is almost inevitable for monotherapy. We are still at the beginning of uncovering the details of how cancer cells metastasize to other parts of the human body to build secondary colonies [91, 222]. A cancer cell, no matter how aggressive it is, is still a living cell and must have a set of proteins to drive basic cellular functions such as metabolism, transcription, translation, and DNA repair. Therefore, a human cancer cell still needs to express around 2000 'essential' genes to survive [24, 85, 218]. Now a problem is to find the set of tumour specific 'essential' genes for targeted treatment. We think that this kind of 'Achilles' heel' exists for cancer as demonstrated by the success (at least transiently) of targeted therapies, the universally detected mutated oncogenes or tumour suppressor genes in individual patients, and the much large number of synthetic lethal interactions in cells. Two challenges remain: 1) Cancer cells can have a different set of tumour specific essential genes because of intra-tumour heterogeneity [8]; therefore we need to find the essential 'trunk essential genes' and apply multiple drugs; 2) Cancer cell could evolve to resist targeted treatments thus we need to

monitor cancer progression and use immunotherapies.

In this dissertation, I focused on computational predictions without biological experimental validations. The CRISPR-Cas (clustered regularly interspaced short palindromic repeats - CRISPR-associated) systems enable rapidly and precisely knockout of genes or introducing mutations [112, 177]. These systems coupled with other measurements such as gene expression enable rapidly quantification of genetic perturbations on molecular phenotypes such as gene expression, and therefore can be used for validating `xseq` predictions. The `xseq` model can be extended to other applications, e.g., analyzing combinations of events that impact expression and synthetic lethalities.

Clustering analysis has been widely used to analyze various 'omics' data such as mRNA gene expression data, miRNA gene expression data, protein expression data, and DNA methylation data. A good framework for clustering analysis is density-based clustering. Interestingly, biological systems seem to adopt strategies similar to non-parametric density estimation to communicate with each other. For example, a haploid **a** yeast cell produces chemicals called pheromones [165]. A haploid $\boldsymbol{\alpha}$ yeast cell senses the concentration of the pheromones and grows toward the **a** yeast cell for mating to produce a diploid yeast cell [12]. In other words, the $\boldsymbol{\alpha}$ yeast cell moves (grows) along the pheromone gradient direction to reach the pheromone density peak. Therefore, grouping points to the basin of attraction of each density peak represents a natural choice for clustering analysis. Our `densityCut` algorithm has been proved to be effective in different settings. However, it does have limitations. For example, for some datasets, the clustering results could be sensitive to the parameter setting when two clusters are 'similar' to each other. This is a problem for many other clustering algorithms as well. An extension of the `densityCut` algorithm is for integrated analysis of multiple 'omic' datasets.

Oncologists and scientists have observed some patients with exceptional responses to treatments. For example, patients with inactivating mutations in *TSC1* or *TSC2* resulted in mTOR pathway activation, and are sensitive to mTOR inhibitors [94, 129, 216]. Tumours with DNA repair deficiency tend to accumulate many more somatic mutations than DNA repair proficiency tumours, and some of the mutant proteins could be recognized by the human immune systems for destruction. Therefore, patients with DNA repair deficiency tumour tend to respond to checkpoint blockade immunotherapies [118, 169, 192]. Since PI3Kα and pan-PI3K inhibitors induce DNA damage, triple negative breast cancer and high-grade serous ovary cancer patients with nonfunctional *TP53* (some also with nonfunctional *BRCA1*) tend to have long response

to the combinations of PARP inhibitors and PIK3 inhibitors. We hope that the current exceptional 'outlier' responders with become common in the coming years of cancer patient care. To achieve this goal, we need to develop efficient computation algorithms to detect all the genomic abnormalities, interpret these abnormalities in the context of other molecular features such as gene expression, DNA methylations, and gene interaction networks as well as tumour microenvironment, detect the dysregulated pathways, and cluster these patients into subgroups for treatment.

Biology is always an inspiration for developing computational algorithms. In Chapter 2, we formalized the SNV prediction problem as a supervised classification problem, and this problem could act as a test case for developing efficient supervised machine learning algorithms for big datasets. (A whole human genome sequenced at 30x depth produces about 300Gb of compressed data.) The `xseq` model presented in Chapter 3 is designed for a specific application. However, a probabilistic graphical model represents a family of distributions and is very flexible to be extended to analyze new problems by introducing new variables or changing the structure of the model. The `densityCut` algorithms can be readily applied for other clustering analysis applications.

In the field of cancer systems biology, we can increasingly access massive biological datasets. We need to analyze and interpret these data to discover biology and collaborate with wet-lab scientists to validate the discoveries. Of particular importance is to develop statistical models and the corresponding efficient inference algorithms for the integrative analyses of biological datasets, to understand the mechanisms of how the genetic and epigenetic alterations transform a normal cell into a cancer cell. This probabilistic modelling approach could explore the dependencies between different measurements (e.g., mutations and gene expression) and provide informative descriptions of the data. A second approach is to transform each dataset into a similarity matrix and combines these similarity matrices for further analyses, e.g., patient stratifications.

The biology we can learn from the massive datasets largely depends on the efficiency and effectiveness of our computational methods. With the newly learned biology, we will keep inventing new technologies for much easier and more accurately measuring biological signals in large scale, as well as measuring new biological signals. Our computational models will keep improving as new data are added and new biology is learned. Computational power will also increase to alleviate the challenges to process these data. As more informative data are extracted and collected for individual patients, e.g., time course measurement of driver gene

mutations and protein biomarker expression, we can better monitor patient disease progression and provide more effective treatments. Eventually, I hope one day, with the biology we learned, and the effective signals we measured and processed for individual patients, we can develop models to recommend drugs to cancer patients the same way as Google recommends webpages and Amazon recommends books to users.

Bibliography

[1] Thomas Abeel, Thibault Helleputte, Yves Van de Peer, Pierre Dupont, and Yvan Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2010. 28

[2] Uri David Akavia, Oren Litvin, Jessica Kim, Felix Sanchez-Garcia, Dylan Kotliar, Helen C Causton, Panisa Pochanard, Eyal Mozes, Levi A Garraway, and Dana Pe'er. An integrated approach to uncover drivers of cancer. *Cell*, 143:1005–1017, 2010. 41, 69

[3] Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 6 edition, 2014. 1, 5, 6

[4] Ash A Alizadeh, Victoria Aranda, Alberto Bardelli, Cedric Blanpain, Christoph Bock, Christine Borowski, Carlos Caldas, Andrea Califano, Michael Doherty, Markus Elsner, et al. Toward understanding and exploiting tumor heterogeneity. *Nat. Med.*, 21(8):846–853, 2015. 2

[5] Andre Altmann, Peter Weber, Carina Quast, Monika Rex-Haffner, Elisabeth B Binder, and Bertram Müller-Myhsok. vipR: variant identification in pooled DNA using R. *Bioinformatics*, 27(13):i77–i84, 2011. 18

[6] Bill Andreopoulos, Aijun An, Xiaogang Wang, and Michael Schroeder. A roadmap of clustering algorithms: finding a match for a biomedical application. *Brief. Bioinform.*, 10(3):297–314, 2009. 99

[7] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: ordering points to identify the clustering structure. In *SIGMOD Rec.*, volume 28, pages 49–60. ACM, 1999. 98

[8] Samuel Aparicio and Carlos Caldas. The implications of clonal genome evolution for cancer medicine. *N. Engl. J. Med.*, 368(9):842–851, 2013. 1, 118

[9] Peter Armitage and Richard Doll. The age distribution of cancer and a multi-stage theory of carcinogenesis. *Br. J. Cancer*, 8(1):1–12, 1954. 1

[10] Miriam Ragle Aure, Israel Steinfeld, Lars Oliver Baumbusch, Knut Liestøl, Doron Lipson, Sandra Nyberg, Bjørn Naume, Kristine Kleivi Sahlberg, Vessela N Kristensen, Anne-Lise Børresen-Dale, et al. Identifying in-trans process associated genes in breast cancer by integrated analysis of copy number and expression data. *PloS one*, 8(1):e53014, 2013. doi: 10.1371/journal.pone.0053014. 58

[11] Jangsun Baek and Geoffrey J McLachlan. Mixtures of common t-factor analyzers for clustering high-dimensional microarray data. *Bioinformatics*, 27(9):1269–1276, 2011. 102

[12] Lee Bardwell. A walk-through of the yeast mating pheromone response pathway. *Peptides*, 25(9):1465–1476, 2004. 119

[13] Ali Bashashati, Gavin Ha, Alicia Tone, Jiarui Ding, Leah M Prentice, Andrew Roth, Jamie Rosner, Karey Shumansky, Steve Kalloger, Janine Senz, et al. Distinct evolutionary trajectories of primary high-grade serous ovarian cancers revealed through spatial mutational profiling. *J. Pathol.*, 231(1):21–34, 2013. 15, 16

[14] Ali Bashashati, Gholamreza Haffari, Jiarui Ding, Gavin Ha, Kenneth Lui, Jamie Rosner, David G Huntsman, Carlos Caldas, Samuel A Aparicio, and Sohrab P Shah. DriverNet: uncovering the impact of somatic driver muta- tions on transcriptional networks in cancer. *Genome Biol.*, 12(4):R41, 2012. doi: 10.1186/gb-2012-13-12-r124. 13, 15, 41

[15] Sean C Bendall, Erin F Simonds, Peng Qiu, D Amir El-ad, Peter O Krutzik, Rachel Finck, Robert V Bruggner, Rachel Melamed, Angelica Trejo, Olga I Ornatsky, et al. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, 332(6030):687–696, 2011. 5, 109

[16] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 57(1):289–300, 1995. 76

[17] Yuval Benjamini and Terence P Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Res.*, 40(10):e72, 2012. doi: 10.1093/nar/gks001. 13

[18] Bonnie Berger, Jian Peng, and Mona Singh. Computational solutions for omics data. *Nat. Rev. Genet.*, 14(5):333–346, 2013. 6

[19] Arindam Bhattacharjee, William G Richards, Jane Staunton, Cheng Li, Stefano Monti, Priya Vasa, Christine Ladd, Javad Beheshti, Raphael Bueno, Michael Gillette, et al. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci. USA*, 98(24):13790–13795, 2001. 102

[20] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006. 7, 11

[21] Christopher M Bishop. Model-based machine learning. *Phil. Trans. R. Soc. A*, 371(1984), 2013. doi: 10.1098/rsta.2012.0222. 7

[22] David M Blei. Build, compute, critique, repeat: Data analysis with latent variable models. *Annu. Rev. Stat. Appl.*, 1:203–232, 2014. 7

[23] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*, 2016. 11

[24] Vincent A Blomen, Peter Májek, Lucas T Jae, Johannes W Bigenzahn, Joppe Nieuwenhuis, Jacqueline Staring, Roberto Sacco, Ferdy R van Diemen, Nadine Olk, Alexey Stukalov, et al. Gene essentiality and synthetic lethality in haploid human cells. *Science*, 350(6264):1092–1096, 2015. 118

[25] Ivana Bozic, Johannes G Reiter, Benjamin Allen, Tibor Antal, Krishnendu Chatterjee, Preya Shah, Yo Sup Moon, Amin Yaqubie, Nicole Kelly, Dung T Le, et al. Evolutionary dynamics of cancer in response to targeted combination therapy. *Elife*, 2:e00747, 2013. doi: 10.7554/eLife.00747. 14

[26] Daniel J Brat, RG Verhaak, Kenneth D Aldape, WK Yung, Sofie R Salama, LA Cooper, Esther Rheinbay, C Ryan Miller, Mark Vitucci, Olena Morozova, et al. Comprehensive, integrative genomic analysis of diffuse lower-grade gliomas. *N. Engl. J. Med.*, 372(26):2481–2498, 2015. 117

[27] Leo Breiman. Statistical modeling: The two cultures. *Stat. Sci.*, 16(3):199–231, 2001. 12

[28] Andrea Califano. Cancer systems biology: The future. In John Mendelsohn, Peter M Howley, Mark A Israel, Joe W Gray, and Craig B Thompson, editors, *The Molecular Basis of Cancer*, chapter 20, pages 297–314. Elsevier, 4 edition, 2015. 4, 6

[29] Hannah Carter, Sining Chen, Leyla Isik, Svitlana Tyekucheva, Victor E Velculescu, Kenneth W Kinzler, Bert Vogelstein, and Rachel Karchin. Cancer-specific high-throughput annotation of somatic mutations: computational prediction of driver missense mutations. *Cancer Res.*, 69(16):6660–6667, 2009. 13

[30] Fong Chun Chan, Adele Telenius, Shannon Healy, Susana Ben-Neriah, Anja Mottok, Raymond Lim, Marie Drake, Sandy Hu, Jiarui Ding, Gavin Ha, et al. An RCOR1 loss–associated gene expression signature identifies a prognostically significant DLBCL subgroup. *Blood*, 125(6):959–966, 2015. 15

[31] Hong Chang et al. Robust path-based spectral clustering. *Pattern Recog.*, 41(1):191–203, 2008. 99

[32] Michael A Chapman, Michael S Lawrence, Jonathan J Keats, Kristian Cibulskis, Carrie Sougnez, Anna C Schinzel, Christina L Harview, Jean-Philippe Brunet, Gregory J Ahmann, Mazhar Adli, et al. Initial genome sequencing and analysis of multiple myeloma. *Nature*, 471(7339):467–472, 2011. 18, 19

[33] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for the cluster tree. In *NIPS*, pages 343–351, 2010. 85

[34] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995. 84

[35] Hugh A Chipman, Edward I George, and Robert E McCulloch. BART: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4(1):266–298, 2010. 23

[36] Dondapati Chowdary, Jessica Lathrop, Joanne Skelton, Kathleen Curtin, Thomas Briggs, Yi Zhang, Jack Yu, Yixin Wang, and Abhijit Mazumder. Prognostic gene expression signatures can be measured in tissues collected in rnalater preservative. *J. Mol. Diagn.*, 8(1):31–39, 2006. 102

[37] Kristian Cibulskis, Michael S Lawrence, Scott L Carter, Andrey Sivachenko, David Jaffe, Carrie Sougnez, Stacey Gabriel, Matthew Meyerson, Eric S Lander, and Gad Getz. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.*, 31(3):213–219, 2013. 39

[38] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002. 84

[39] Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012. 41, 68

[40] Matteo D'Antonio and Francesca D Ciccarelli. Integrated analysis of recurrent properties of cancer genes to identify novel drivers. *Genome Biol.*, 14(5):R52, 2013. doi: 10.1186/gb-2013-14-5-r52. 56

[41] Sanjoy Dasgupta and Samory Kpotufe. Optimal rates for k-nn density and mode estimation. In *NIPS*, pages 2555–2563, 2014. 94

[42] Teresa Davoli, Andrew Wei Xu, Kristen E Mengwasser, Laura M Sack, John C Yoon, Peter J Park, and Stephen J Elledge. Cumulative haploinsufficiency and triplosensitivity drive aneuploidy patterns and shape the cancer genome. *Cell*, 155(4):948–962, 2013. 71

[43] Marcilio CP de Souto, Ivan G Costa, Daniel SA de Araujo, Teresa B Ludermir, and Alexander Schliep. Clustering cancer gene expression data: a comparative study. *BMC bioinform.*, 9:497. doi: 10.1186/1471-2105-9-497. 103

[44] Nathan D Dees, Qunyuan Zhang, Cyriac Kandoth, Michael C Wendl, William Schierding, Daniel C Koboldt, Thomas B Mooney, Matthew B Callaway, David Dooling, Elaine R Mardis, et al. MuSiC: identifying mutational significance in cancer genomes. *Genome Res.*, 22(8):1589–1598, 2012. 13, 71

[45] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 39(1):1–38, 1977. 45, 49

[46] Jiarui Ding, Ali Bashashati, Andrew Roth, Arusha Oloumi, Kane Tse, Thomas Zeng, Gholamreza Haffari, Martin Hirst, Marco A Marra, Anne Condon, et al. Feature-based classifiers for somatic mutation detection in tumour–normal paired sequencing data. *Bioinformatics*, 28(2):167–175, 2012. iii, 13, 15

[47] Jiarui Ding, Melissa K McConechy, Hugo M Horlings, Gavin Ha, Fong Chun Chan, Tyler Funnell, Sarah C Mullaly, Jüri Reimand, Ali Bashashati, Gary D Bader, et al. Systematic analysis of somatic mutations impacting gene expression in 12 tumour types. *Nat. Commun.*, 6:8554, 2015. doi: 10.1038/ncomms9554. iii, 15, 69

[48] Jiarui Ding and Sohrab Shah. A robust hidden semi–Markov model with application to aCGH data processing. *Int. J. Data Min. Bioinform.*, 8(4):427–442, 2013. 15

[49] Jiarui Ding, Sohrab Shah, and Anne Condon. densitycut: an efficient and versatile topological approach for automatic clustering of biological data. *Bioinformatics*, 2016. doi: 10.1093/bioinformatics/btw227. iii, 15

[50] Jiarui Ding and Sohrab P Shah. Robust hidden semi-Markov modeling of array CGH data. In *IEEE BIBM*, pages 603–608. IEEE, 2010. 15

[51] Li Ding, Matthew J Ellis, Shunqiang Li, David E Larson, Ken Chen, John W Wallis, Christopher C Harris, Michael D McLellan, Robert S Fulton, Lucinda L Fulton, et al. Genome remodelling in a basal-like breast cancer metastasis and xenograft. *Nature*, 464(7291):999–1005, 2010. 18

[52] Li Ding, Minjung Kim, Krishna L Kanchi, Nathan D Dees, Charles Lu, Malachi Griffith, David Fenstermacher, Hyeran Sung, Christopher A Miller, Brian Goetz, et al. Clonal architectures and driver mutations in metastatic melanomas. *PLoS ONE*, 9(11):e111153, 2014. doi: 10.1371/journal.pone.0111153. 106

[53] Li Ding, Timothy J Ley, David E Larson, Christopher A Miller, Daniel C Koboldt, John S Welch, Julie K Ritchey, Margaret A Young, Tamara Lamprecht, Michael D McLellan, et al. Clonal evolution in relapsed acute myeloid leukaemia revealed by whole-genome sequencing. *Nature*, 481(7382):506–510, 2012. 13, 104, 112

[54] Li Ding, Michael C Wendl, Joshua F McMichael, and Benjamin J Raphael. Expanding the computational toolbox for mining cancer genomes. *Nat. Rev. Genet.*, 15(8):556–570, 2014. 2, 6, 116

[55] Brian J Druker, François Guilhot, Stephen G O'Brien, Insa Gathmann, Hagop Kantarjian, Norbert Gattermann, Michael WN Deininger, Richard T Silver, John M Goldman, Richard M Stone, et al. Five-year follow-up of patients receiving imatinib for chronic myeloid leukemia. *N. Engl. J. Med.*, 355(23):2408–2417, 2006. 3

[56] EK Engle, DAC Fisher, CA Miller, MD McLellan, RS Fulton, DM Moore, RK Wilson, TJ Ley, and ST Oh. Clonal evolution revealed by whole genome sequencing in a case of primary myelofibrosis transformed to secondary acute myeloid leukemia. *Leukemia*, 29(4):869–876, 2015. 103, 104, 105

[57] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996. 85, 98, 104

[58] Adam D Ewing, Kathleen E Houlahan, Yin Hu, Kyle Ellrott, Cristian Caloian, Takafumi N Yamaguchi, J Christopher Bare, Christine P'ng, Daryl Waggott, Veronica Y Sabelnykova, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nature methods*, 12(7):623–630, 2015. 39

[59] H Christina Fan, Glenn K Fu, and Stephen PA Fodor. Combinatorial labeling of single cells for gene expression cytometry. *Science*, 347(6222):628–637, 2015. 5

[60] Keith T Flaherty, Igor Puzanov, Kevin B Kim, Antoni Ribas, Grant A McArthur, Jeffrey A Sosman, Peter J O'Dwyer, Richard J Lee, Joseph F Grippo, Keith Nolop, et al. Inhibition of mutated, activated BRAF in metastatic melanoma. *N. Engl. J. Med.*, 363(9):809–819, 2010. 117

[61] William A Flavahan, Yotam Drier, Brian B Liau, Shawn M Gillespie, Andrew S Venteicher, Anat O Stemmer-Rachamimov, Mario L Suvà, and Bradley E Bernstein. Insulator dysfunction and oncogene activation in IDH mutant gliomas. *Nature*, 529(7584):110–114, 2016. 117

[62] Simon A Forbes, Nidhi Bindal, Sally Bamford, Charlotte Cole, Chai Yin Kok, David Beare, Mingming Jia, Rebecca Shepherd, Kenric Leung, Andrew Menzies, et al. COSMIC: mining complete cancer genomes in the catalogue of somatic mutations in cancer. *Nucleic Acids Res.*, 39(suppl 1):D945–D950, 2011. 56

[63] Chris Fraley and Adrian E Raftery. Model-based methods of classification: using the mclust software in chemometrics. *J. Stat. Softw.*, 18(6):1–13, 2007. 36, 84, 99, 104

[64] Andrea Franceschini, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, et al. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.*, 41(D1):D808–D815, 2013. 6, 54

[65] Pasi Fränti and Olli Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recog.*, 39(5):761–775, 2006. 99

[66] Ana L.N. Fred and Anil K Jain. Robust data clustering. In *CVPR*, volume 2, pages 128–133, 2003. 98

[67] David A Fruman and Lewis C Cantley. Idelalisiba PI3Kδ inhibitor for B-cell cancers. *N. Engl. J. Med.*, 370(11):1061–1062, 2014. 116

[68] Limin Fu and Enzo Medico. FLAME, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC Bioinform.*, 8:3, 2007. doi: 10.1186/1471-2105-8-3. 86, 87, 99

[69] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory*, 21(1):32–40, 1975. 84

[70] P Andrew Futreal, Lachlan Coin, Mhairi Marshall, Thomas Down, Timothy Hubbard, Richard Wooster, Nazneen Rahman, and Michael R Stratton. A census of human cancer genes. *Nat. Rev. Cancer*, 4(3):177–183, 2004. 2, 54, 71, 73

[71] Levi A Garraway and Eric S Lander. Lessons from the cancer genome. *Cell*, 153(1):17–37, 2013. 4

[72] Mark B Gerstein, Anshul Kundaje, Manoj Hariharan, Stephen G Landt, Koon-Kiu Yan, Chao Cheng, Xinmeng Jasmine Mu, Ekta Khurana, Joel Rozowsky, Roger Alexander, et al. Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489(7414):91–100, 2012. 54

[73] Moritz Gerstung, Christian Beisel, Markus Rechsteiner, Peter Wild, Peter Schraml, Holger Moch, and Niko Beerenwinkel. Reliable detection of subclonal single-nucleotide variants in tumour cell populations. *Nat. Commun.*, 3:811, 2012. doi: 10.1038/ncomms1814. 39

[74] Olivier Gevaert, Victor Villalobos, Branimir I Sikic, and Sylvia K Plevritis. Identification of ovarian cancer driver genes by using module network integration of multi-omics data. *Interface Focus*, 3(4), 2013. doi: 10.1098/rsfs.2013.0013. 58

[75] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015. 6

[76] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007. 99

[77] Michael S Glickman and Charles L Sawyers. Converting cancer therapies into cures: lessons from infectious diseases. *Cell*, 148(6):1089–1098, 2012. 4, 118

[78] Rodrigo Goya, Mark GF Sun, Ryan D Morin, Gillian Leung, Gavin Ha, Kimberley C Wiegand, Janine Senz, Anamaria Crisan, Marco A Marra, Martin Hirst, et al. SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors. *Bioinformatics*, 26(6):730–736, 2010. 18

[79] Mel Greaves and Carlo C Maley. Clonal evolution in cancer. *Nature*, 481(7381):306–313, 2012. 1

[80] Malachi Griffith, Christopher A Miller, Obi L Griffith, Kilannin Krysiak, Zachary L Skidmore, Avinash Ramu, Jason R Walker, Ha X Dang, Lee Trani, David E Larson, et al. Optimizing cancer genome sequencing and analysis. *Cell Systems*, 1(3):210–223, 2015. 13, 39, 112, 116

[81] Matthew R Grimmer and Joseph F Costello. Cancer: Oncogene brought into the loop. *Nature*, 529(7584):34–35, 2016. 117

[82] Gavin Ha, Andrew Roth, Jaswinder Khattra, Julie Ho, Damian Yap, Leah M Prentice, Nataliya Melnyk, Andrew McPherson, Ali Bashashati, Emma Laks, et al. Titan: inference of copy number architectures in clonal cell populations from tumor whole-genome sequence data. *Genome Res.*, 24(11):1881–1893, 2014. 116

[83] Gavin Ha, Andrew Roth, Daniel Lai, Ali Bashashati, Jiarui Ding, Rodrigo Goya, Ryan Giuliany, Jamie Rosner, Arusha Oloumi, Karey Shumansky, et al. Integrative analysis of genome-wide loss of heterozygosity and monoallelic expression at nucleotide resolution reveals disrupted pathways in triple-negative breast cancer. *Genome Res.*, 22(10):1995–2007, 2012. 2

[84] Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674, 2011. 1

[85] Traver Hart, Megha Chandrashekhar, Michael Aregger, Zachary Steinhart, Kevin R Brown, Graham MacLeod, Monika Mis, Michal Zimmermann, Amelie Fradet-Turcotte, Song Sun, et al. High-resolution CRISPR screens reveal fitness genes and genotype-specific cancer liabilities. *Cell*, 163(6):1515–1526, 2015. 118

[86] John A Hartigan. *Clustering algorithms*. Wiley, New York, 1975. 84

[87] PM Hartigan. Algorithm as 217: Computation of the dip statistic to test for unimodality. *J. R. Stat. Soc. Ser. C Appl. Stat.*, 34(3):320–325, 1985. 36

[88] Leland Hartwell, Leroy Hood, Michael Goldberg, Ann Reynolds, and Lee Silver. *Genetics: from genes to genomes*. McGraw-Hill Education, New York, 4 edition, 2011. 4

[89] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Verlag, 2 edition, 2009. 23

[90] James R Heath, Antoni Ribas, and Paul S Mischel. Single-cell analysis tools for drug discovery and development. *Nat. Rev. Drug Discov.*, 15(3):204–216, 2016. 5

[91] Ayuko Hoshino, Bruno Costa-Silva, Tang-Long Shen, Goncalo Rodrigues, Ayako Hashimoto, Milica Tesic Mark, Henrik Molina, Shinji Kohsaka, Angela Di Giannatale, Sophia Ceder, et al. Tumour exosome integrins determine organotropic metastasis. *Nature*, 527(7578):329–335, 2015. 118

[92] Norman Huang, Parantu K Shah, and Cheng Li. Lessons from a decade of integrating cancer copy number alterations with gene expression profiles. *Brief. Bioinform.*, 13(3):305–316, 2012. 58

[93] Lawrence Hubert and Phipps Arabie. Comparing partitions. *J. classif.*, 2(1):193–218, 1985. 98

[94] Gopa Iyer, Aphrothiti J Hanrahan, Matthew I Milowsky, Hikmat Al-Ahmadie, Sasinya N Scott, Manickam Janakiraman, Mono Pirun, Chris Sander, Nicholas D Socci, Irina Ostrovnaya, et al. Genome sequencing identifies a basis for everolimus sensitivity. *Science*, 338(6104):221–221, 2012. 119

[95] Anil K Jain et al. Data clustering: a users dilemma. In *Pattern Recog. Mach. Intell.*, volume 3776 of *LNCS*, pages 1–10. Springer-Verlag, 2005. 99

[96] C Jiang, Zhenyu Xuan, Fang Zhao, and Michael Q Zhang. TRED: a transcriptional regulatory element database, new entries and other development. *Nucleic Acids Res.*, 35(suppl 1):D137–D140, 2007. 6

[97] Rebecka Jörnsten, Tobias Abenius, Teresia Kling, Linnéa Schmidt, Erik Johansson, Torbjörn EM Nordling, Bodil Nordlander, Chris Sander, Peter Gennemark, Keiko Funa, et al. Network modeling of the transcriptional effects of copy number aberrations in glioblastoma. *Mol. Syst. Biol.*, 7(1):486, 2011. doi: 10.1038/msb.2011.17. 41

[98] Cyriac Kandoth, Michael D McLellan, Fabio Vandin, Kai Ye, Beifang Niu, Charles Lu, Mingchao Xie, Qunyuan Zhang, Joshua F McMichael, Matthew A Wyczalkowski, et al. Mutational landscape and significance across 12 major cancer types. *Nature*, 502(7471):333–339, 2013. 55, 71, 75, 76

[99] Minoru Kanehisa and Susumu Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28(1):27–30, 2000. 6, 54

[100] Peter D Karp, Christos A Ouzounis, Caroline Moore-Kochlacs, Leon Goldovsky, Pallavi Kaipa, Dag Ahrén, Sophia Tsoka, Nikos Darzentas, Victor Kunin, and Núria López-Bigas. Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Res.*, 33(19):6083–6089, 2005. 6, 54

[101] Jin H Kim and Judea Pearl. A computational model for causal and diagnostic reasoning in inference systems. In *Proceedings of the Eighth international joint conference on Artificial intelligence-Volume 1*, pages 190–193. Morgan Kaufmann Publishers Inc., 1983. 46

[102] PDW Kirk, AC Babtie, and MPH Stumpf. Systems biology (un) certainties. *Science*, 350(6259):386–388, 2015. 4

[103] Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015. 5

[104] Susumu Kobayashi, Titus J Boggon, Tajhal Dayaram, Pasi A Jänne, Olivier Kocher, Matthew Meyerson, Bruce E Johnson, Michael J Eck, Daniel G Tenen, and Balázs Halmos. EGFR mutation and resistance of non–small-cell lung cancer to gefitinib. *N. Engl. J. Med.*, 352(8):786–792, 2005. 81

[105] Daniel C Koboldt, Ken Chen, Todd Wylie, David E Larson, Michael D McLellan, Elaine R Mardis, George M Weinstock, Richard K Wilson, and Li Ding. VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics*, 25(17):2283–2285, 2009. 18

[106] Daniel C Koboldt, Qunyuan Zhang, David E Larson, Dong Shen, Michael D McLellan, Ling Lin, Christopher A Miller, Elaine R Mardis, Li Ding, and Richard K Wilson. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.*, 22(3):568–576, 2012. 39

[107] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 7

[108] Warren LG Koontz, Patrenahalli M Narendra, and Keinosuke Fukunaga. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Trans. Comput.*, 100(9):936–944, 1976. 84, 90

[109] Samory Kpotufe and Ulrike V Luxburg. Pruning nearest neighbor cluster trees. In *ICML*, pages 225–232, 2011. 85

[110] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 1(3):231–240, 2011. 102

[111] Zhongwu Lai, Aleksandra Markovets, Miika Ahdesmaki, and Justin Johnson. Vardict: a novel and versatile variant caller for next-generation sequencing in cancer research. *Cancer Res.*, 75(15 Supplement):4864–4864, 2015. 39

[112] Eric S Lander. The heroes of CRISPR. *Cell*, 164(1):18–28, 2016. 119

[113] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, 9(4):357–359, 2012. 2

[114] Ben Langmead, Cole Trapnell, Mihai Pop, Steven L Salzberg, et al. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biol*, 10(3):R25, 2009. doi: 10.1186/gb-2009-10-3-r25. 2

[115] David E Larson, Christopher C Harris, Ken Chen, Daniel C Koboldt, Travis E Abbott, David J Dooling, Timothy J Ley, Elaine R Mardis, Richard K Wilson, and Li Ding. SomaticSniper: identification of somatic point mutations in whole genome sequencing data. *Bioinformatics*, 28(3):311–317, 2012. 39

[116] Michael S Lawrence, Petar Stojanov, Craig H Mermel, James T Robinson, Levi A Garraway, Todd R Golub, Matthew Meyerson, Stacey B Gabriel, Eric S Lander, and Gad Getz. Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature*, 505:495–501, 2014. 4, 54, 55, 71, 80

[117] Michael S Lawrence, Petar Stojanov, Paz Polak, Gregory V Kryukov, Kristian Cibulskis, Andrey Sivachenko, Scott L Carter, Chip Stewart, Craig H Mermel, Steven A Roberts, et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, 499:214–218, 2013. 13, 40, 56, 73

[118] Dung T Le, Jennifer N Uram, Hao Wang, Bjarne R Bartlett, Holly Kemberling, Aleksandra D Eyring, Andrew D Skora, Brandon S Luber, Nilofer S Azad, Dan Laheru, et al. PD-1 blockade in tumors with mismatch-repair deficiency. *N. Engl. J. Med.*, 372(26):2509–2520, 2015. 14, 119

[119] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 12

[120] Jacob H Levine et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197, 2015. 84, 109, 111

[121] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009. 2

[122] Heng Li and Richard Durbin. Fast and accurate long-read alignment with Burrows–Wheeler transform. *Bioinformatics*, 26(5):589–595, 2010. 2

[123] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, et al. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009. 18

[124] Ruiqiang Li, Yingrui Li, Xiaodong Fang, Huanming Yang, Jian Wang, Karsten Kristiansen, and Jun Wang. SNP detection for massively parallel whole-genome resequencing. *Genome Res.*, 19(6):1124–1132, 2009. 18

[125] Frank Lin and William W Cohen. Power iteration clustering. In *ICML*, pages 655–662, 2010. 85, 86

[126] Enric Llorens-Bobadilla et al. Single-cell transcriptomics reveals a population of dormant neural stem cells that become activated upon brain injury. *Cell Stem Cell*, 17(3):329–340, 2015. 107, 108, 109

[127] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. 11

[128] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015. 5

[129] Vivien Marx. Cancer: A most exceptional response. *Nature*, 520(7547):389–393, 2015. 119

[130] David L Masica and Rachel Karchin. Correlation of somatic mutation and expression identifies genes important in human glioblastoma progression and survival. *Cancer Res.*, 71(13):4550–4561, 2011. 13, 41

[131] Anthony Mathelier, Calvin Lefebvre, Allen W Zhang, David J Arenillas, Jiarui Ding, Wyeth W Wasserman, and Sohrab P Shah. Cis-regulatory somatic mutations and gene-expression alteration in B-cell lymphomas. *Genome Biol.*, 16:84, 2015. doi: 10.1186/s13059-015-0648-7. 15

[132] Lisa Matthews, Gopal Gopinath, Marc Gillespie, Michael Caudy, David Croft, Bernard de Bono, Phani Garapati, Jill Hemish, Henning Hermjakob, Bijay Jassal, et al. Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res.*, 37(suppl 1):D619–D622, 2009. 6

[133] Melissa K McConechy, Michael S Anglesio, Steve E Kalloger, Winnie Yang, Janine Senz, Christine Chow, Alireza Heravi-Moussavi, Gregg B Morin, Anne-Marie Mes-Masson, Mark S Carey, et al. Subtype-specific mutation of PPP2R1A in endometrial and ovarian carcinomas. *J. Pathol.*, 223(5):567–573, 2011. 18

[134] Melissa K McConechy, Jiarui Ding, Maggie CU Cheang, Kimberly C Wiegand, Janine Senz, Alicia A Tone, Winnie Yang, Leah M Prentice, Kane Tse, Thomas Zeng, et al. Use of mutation profiles to refine the classification of endometrial carcinomas. *J. Pathol.*, 228(1):20–30, 2012. 15, 16

[135] Melissa K McConechy, Jiarui Ding, Janine Senz, Winnie Yang, Nataliya Melnyk, Alicia A Tone, Leah M Prentice, Kimberly C Wiegand, Jessica N McAlpine, Sohrab P Shah, et al. Ovarian and endometrial endometrioid carcinomas have distinct CTNNB1 and PTEN mutation profiles. *Mod. Pathol.*, 27(1):128–134, 2013. 15, 16

[136] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, et al. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, 20(9):1297–1303, 2010. 18

[137] Roger McLendon, Allan Friedman, Darrell Bigner, Erwin G Van Meir, Daniel J Brat, Gena M Mastrogianakis, Jeffrey J Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008. 2

[138] Donal P McLornan, Alan List, and Ghulam J Mufti. Applying synthetic lethality for the selective targeting of cancer. *N. Engl. J. Med.*, 371(18):1725–1735, 2014. 14

[139] Frazer Meacham, Dario Boffelli, Joseph Dhahbi, David IK Martin, Meromit Singer, and Lior Pachter. Identification and correction of systematic error in high-throughput sequence data. *BMC Bioinform.*, 12:451, 2011. doi: 10.1186/1471-2105-12-451. 23, 36

[140] Marina Meilă. Comparing clusterings–an information based distance. *J. Multivar. Anal.*, 98(5):873–895, 2007. 95

[141] Giovanna Menardi and Adelchi Azzalini. An advancement in clustering via nonparametric density estimation. *Stat. Comp.*, 24(5):753–767, 2013. 85

[142] Craig H Mermel, Steven E Schumacher, Barbara Hill, Matthew L Meyerson, Rameen Beroukhim, Gad Getz, et al. GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome Biol*, 12(4):R41, 2011. doi: 10.1186/gb-2011-12-4-r41. 13, 60

[143] Matthew Meyerson, Stacey Gabriel, and Gad Getz. Advances in understanding cancer genomes through second-generation sequencing. *Nat. Rev. Genet.*, 11(10):685–696, 2010. 2, 3

[144] Christopher A Miller et al. SciClone: Inferring clonal architecture and tracking the spatial and temporal patterns of tumor evolution. *PLoS Comput. Biol.*, 10(8):e1003665, 2014. doi: 10.1371/journal.pcbi.1003665. 103, 104

[145] Qianxing Mo, Sijian Wang, Venkatraman E Seshan, Adam B Olshen, Nikolaus Schultz, Chris Sander, R Scott Powers, Marc Ladanyi, and Ronglai Shen. Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proc. Natl Acad. Sci. USA*, 110(11):4245–4250, 2013. 58

[146] Ryan D Morin, Nathalie A Johnson, Tesa M Severson, Andrew J Mungall, Jianghong An, Rodrigo Goya, Jessica E Paul, Merrill Boyle, Bruce W Woolcock, Florian Kuchenbauer, et al. Somatic mutations altering EZH2 (Tyr641) in follicular and diffuse large B-cell lymphomas of germinal-center origin. *Nat. Genet.*, 42(2):181–185, 2010. 18

[147] Ryan D Morin, Maria Mendez-Lago, Andrew J Mungall, Rodrigo Goya, Karen L Mungall, Richard D Corbett, Nathalie A Johnson, Tesa M Severson, Readman Chiu, Matthew Field, et al. Frequent mutation of histone-modifying genes in non-hodgkin lymphoma. *Nature*, 476(7360):298–303, 2011. 18

[148] Ryan D Morin, Karen Mungall, Erin Pleasance, Andrew J Mungall, Rodrigo Goya, Ryan D Huff, David W Scott, Jiarui Ding, Andrew Roth, Readman Chiu, et al. Mutational and structural analysis of diffuse large B-cell lymphoma using whole-genome sequencing. *Blood*, 122(7):1256–1265, 2013. 15

[149] David M Mount. ANN programming manual. Technical report, Dept. of Computer Science, U. of Maryland, 1998. 93

[150] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, Cambridge, 2012. 7, 11, 44

[151] Masao Nagasaki, Jun Yasuda, Fumiki Katsuoka, Naoki Nariai, Kaname Kojima, Yosuke Kawai, Yumi Yamaguchi-Kabata, Junji Yokozawa, Inaho Danjoh, Sakae Saito, et al. Rare variant discovery by deep whole-genome sequencing of 1,070 Japanese individuals. *Nat. Commun.*, 6:8018, 2015. doi: 10.1038/ncomms9018. 116

[152] Eszter Nagy and Lynne E Maquat. A rule for termination-codon position within intron-containing genes: when nonsense affects rna abundance. *Trends Biochem. Sci.*, 23(6):198–199, 1998. 41

[153] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2:849–856, 2002. 85

[154] Peter C Nowell. The clonal evolution of tumor cell populations. *Science*, 194(4260):23–28, 1976. 1

[155] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1999. 89

[156] D Williams Parsons, Siân Jones, Xiaosong Zhang, Jimmy Cheng-Ho Lin, Rebecca J Leary, Philipp Angenendt, Parminder Mankoo, Hannah Carter, I-Mei Siu, Gary L Gallia, et al. An integrated genomic analysis of human glioblastoma multiforme. *Science*, 321(5897):1807–1812, 2008. 117

[157] Anoop P Patel, Itay Tirosh, John J Trombetta, Alex K Shalek, Shawn M Gillespie, Hiroaki Wakimoto, Daniel P Cahill, Brian V Nahed, William T Curry, Robert L Martuza, et al. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401, 2014. 81

[158] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the second national conference on artificial intelligence*, pages 133–136. AAAI Press, 1982. 45

[159] Dana Pe'er and Nir Hacohen. Principles and strategies for developing network models in cancer. *Cell*, 144(6):864–873, 2011. 4

[160] Alexander R Pico, Thomas Kelder, Martijn P Van Iersel, Kristina Hanspers, Bruce R Conklin, and Chris Evelo. WikiPathways: pathway editing for the people. *PLoS Biol.*, 6(7):e184, 2008. doi: 10.1371/journal.pbio.0060184. 54

[161] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat. Biotechnol.*, 32(10):1053–1058, 2014. 84, 107, 109

[162] Anirudh Prahallad, Chong Sun, Sidong Huang, Federica Di Nicolantonio, Ramon Salazar, Davide Zecchin, Roderick L Beijersbergen, Alberto Bardelli, and René Bernards. Unresponsiveness of colon cancer to BRAF (V600E) inhibition through feedback activation of EGFR. *Nature*, 483(7388):100–103, 2012. 81, 117

[163] Xose S Puente, Magda Pinyol, Víctor Quesada, Laura Conde, Gonzalo R Ordóñez, Neus Villamor, Georgia Escaramis, Pedro Jares, Sílvia Beà, Marcos González-Díaz, et al. Whole-genome sequencing identifies recurrent mutations in chronic lymphocytic leukaemia. *Nature*, 475(7354):101–105, 2011. 18

[164] Benjamin J Raphael, Jason R Dobson, Layla Oesper, and Fabio Vandin. Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine. *Genome Med.*, 6(1):5, 2014. doi: 10.1186/gm524. 13, 41

[165] Jane Reece, Lisa A Urry, Noel Meyers, Michael L Cain, Steven A Wasserman, Peter V Minorsky, Robert B Jackson, and Bernard N Cooke. *Campbell biology*. Pearson Higher Education AU, 10 edition, 2013. 119

[166] Jüri Reimand and Gary D Bader. Systematic analysis of somatic mutations in phosphorylation signaling predicts novel cancer drivers. *Mol. Syst. Biol.*, 9(1):637, 2013. doi: 10.1038/msb.2012.68. 71, 74

[167] Jüri Reimand, Omar Wagih, and Gary D Bader. The mutational landscape of phosphorylation signaling in cancer. *Sci. Rep.*, 3:2651, 2013. doi: 10.1038/srep02651. 74

[168] B. Reva, Y. Antipin, and C. Sander. Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic Acids Res.*, 39(17):e118, 2011. doi: 10.1093/nar/gkr407. 13, 40

[169] Naiyer A Rizvi, Matthew D Hellmann, Alexandra Snyder, Pia Kvistborg, Vladimir Makarov, Jonathan J Havel, William Lee, Jianda Yuan, Phillip Wong, Teresa S Ho, et al. Mutational landscape determines sensitivity to PD-1 blockade in non–small cell lung cancer. *Science*, 348(6230):124–128, 2015. 14, 119

[170] Kimberly Robasky, Nathan E Lewis, and George M Church. The role of replicates for error mitigation in next-generation sequencing. *Nat. Rev. Genet.*, 15(1):56–62, 2014. 13

[171] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. 84, 98

[172] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, et al. A proteome-scale map of the human interactome network. *Cell*, 159(5):1212–1226, 2014. 6, 74

[173] Michael G Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J Lennon, Ryan Hegarty, Chad Nusbaum, and David B Jaffe. Characterizing and measuring bias in sequence data. *Genome Biol*, 14:R51, 2013. doi: 10.1186/gb-2013-14-5-r51. 13

[174] Andrew Roth, Jiarui Ding, Ryan Morin, Anamaria Crisan, Gavin Ha, Ryan Giuliany, Ali Bashashati, Martin Hirst, Gulisa Turashvili, Arusha Oloumi, et al. JointSNVMix: a probabilistic model for accurate detection of somatic mutations in normal/tumour paired next-generation sequencing data. *Bioinformatics*, 28(7):907–913, 2012. 39

[175] Andrew Roth, Jaswinder Khattra, Damian Yap, Adrian Wan, Emma Laks, Justina Biele, Gavin Ha, Samuel Aparicio, Alexandre Bouchard-Côté, and Sohrab P Shah. Pyclone: statistical inference of clonal population structure in cancer. *Nat. Methods*, 11(4):396–398, 2014. 107

[176] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, 3 edition, 2010. 6, 9

[177] Francisco J Sánchez-Rivera and Tyler Jacks. Applications of the CRISPR-Cas9 system in cancer biology. *Nature Reviews Cancer*, 15:387–395, 2015. 119

[178] Jörg Sander, Xuejie Qin, Zhiyong Lu, Nan Niu, and Alex Kovarsky. Automatic extraction of clusters from hierarchical clustering representations. In *Advances in knowledge discovery and data mining*, pages 75–87. Springer, 2003. 99

[179] Christopher T Saunders, Wendy SW Wong, Sajani Swamy, Jennifer Becq, Lisa J Murray, and R Keira Cheetham. Strelka: accurate somatic small-variant calling from sequenced tumor–normal sample pairs. *Bioinformatics*, 28(14):1811–1817, 2012. 39

[180] Carl F Schaefer, Kira Anthony, Shiva Krupa, Jeffrey Buchoff, Matthew Day, Timo Hannay, and Kenneth H Buetow. PID: the pathway interaction database. *Nucleic Acids Res.*, 37(suppl 1):D674–D679, 2009. 6

[181] Ton N Schumacher and Robert D Schreiber. Neoantigens in cancer immunotherapy. *Science*, 348(6230):69–74, 2015. 14

[182] Parag P Shah, William W Lockwood, Kumar Saurabh, Zimple Kurlawala, Sean P Shannon, Sabine Waigel, Wolfgang Zacharias, and Levi J Beverly. Ubiquilin1 represses migration and epithelial-to-mesenchymal transition of human non-small cell lung cancer cells. *Oncogene*, 34(13):1709–1717, 2015. 73

[183] Sohrab P Shah, Martin Köbel, Janine Senz, Ryan D Morin, Blaise A Clarke, Kimberly C Wiegand, Gillian Leung, Abdalnasser Zayed, Erika Mehl, Steve E Kalloger, et al. Mutation of FOXL2 in granulosa-cell tumors of the ovary. *N. Engl. J. Med.*, 360(26):2719–2729, 2009. 18

[184] Sohrab P Shah, Ryan D Morin, Jaswinder Khattra, Leah Prentice, Trevor Pugh, Angela Burleigh, Allen Delaney, Karen Gelmon, Ryan Guliany, Janine Senz, et al. Mutational evolution in a lobular breast tumour profiled at single nucleotide resolution. *Nature*, 461(7265):809–813, 2009. 18

[185] Sohrab P Shah, Andrew Roth, Rodrigo Goya, Arusha Oloumi, Gavin Ha, Yongjun Zhao, Gulisa Turashvili, Jiarui Ding, Kane Tse, Gholamreza Haffari, et al. The clonal and mutational evolution spectrum of primary triple-negative breast cancers. *Nature*, 486(7403):395–399, 2012. 15, 16, 27, 83

[186] Padmanee Sharma and James P Allison. The future of immune checkpoint therapy. *Science*, 348(6230):56–61, 2015. 14

[187] Alice T. Shaw, Manabu Soda, Yoshihiro Yamashita, Toshihide Ueno, Junpei Takashima, Takahiro Nakajima, Yasushi Yatabe, Kengo Takeuchi, Toru Hamada, Hidenori Haruta, et al. Resensitization to Crizotinib by the lorlatinib ALK resistance mutation L1198F. *N. Engl. J. Med.*, 363(18):1734–1739, 2015. 14

[188] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000. 85

[189] Jianxin Shi, Qingqi Han, Heng Zhao, Chenxi Zhong, and Feng Yao. Downregulation of MED23 promoted the tumorigenecity of esophageal squamous cell carcinoma. *Mol. Carcinog.*, 53(10):833–840, 2014. 73

[190] Gordon K Smyth. Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer, 2005. 76

[191] Matija Snuderl, Ladan Fazlollahi, Long P Le, Mai Nitta, Boryana H Zhelyazkova, Christian J Davidson, Sara Akhavanfard, Daniel P Cahill, Kenneth D Aldape, Rebecca A Betensky, et al. Mosaic amplification of multiple receptor tyrosine kinase genes in glioblastoma. *Cancer cell*, 20(6):810–817, 2011. 3

[192] Alexandra Snyder, Vladimir Makarov, Taha Merghoub, Jianda Yuan, Jesse M Zaretsky, Alexis Desrichard, Logan A Walsh, Michael A Postow, Phillip Wong, Teresa S Ho, et al. Genetic basis for clinical response to CTLA-4 blockade in melanoma. *N. Engl. J. Med.*, 371(23):2189–2199, 2014. 14, 119

[193] Andrea Sottoriva, Haeyoun Kang, Zhicheng Ma, Trevor A Graham, Matthew P Salomon, Junsong Zhao, Paul Marjoram, Kimberly Siegmund, Michael F Press, Darryl Shibata, et al. A big bang model of human colorectal tumor growth. *Nat. Genet.*, 47(3):209–216, 2015. 2

[194] Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.*, 16(3):133–145, 2015. 5

[195] Michael R Stratton. Exploring the genomes of cancer cells: progress and promise. *Science*, 331(6024):1553–1558, 2011. 4

[196] Michael R Stratton, Peter J Campbell, and P Andrew Futreal. The cancer genome. *Nature*, 458(7239):719–724, 2009. 1, 4

[197] Werner Stuetzle and Rebecca Nugent. A generalized single linkage method for estimating the cluster tree of a density. *J. Comp. Graph. Stat.*, 19(2):397–418, 2010. 85

[198] Chong Sun, Liqin Wang, Sidong Huang, Guus JJE Heynen, Anirudh Prahallad, Caroline Robert, John Haanen, Christian Blank, Jelle Wesseling, Stefan M Willems, et al. Reversible and adaptive resistance to BRAF (V600E) inhibition in melanoma. *Nature*, 508(7494):118–122, 2014. 117

[199] Yun-Chi Tang and Angelika Amon. Gene copy-number alterations: A cost-benefit analysis. *Cell*, 152(3):394–405, 2013. 60

[200] Kensuke Tateishi, Hiroaki Wakimoto, A John Iafrate, Shota Tanaka, Franziska Loebel, Nina Lelic, Dmitri Wiederschain, Olivier Bedel, Gejing Deng, Bailin Zhang, et al. Extreme vulnerability of IDH1 mutant cancers to NAD+ depletion. *Cancer cell*, 28(6):773–784, 2015. 117

[201] The Cancer Genome Atlas Research Network. Integrated genomic characterization of endometrial carcinoma. *Nature*, 497(7447):67–73, 2013. 80

[202] Cristian Tomasetti and Bert Vogelstein. Variation in cancer risk among tissues can be explained by the number of stem cell divisions. *Science*, 347(6217):78–81, 2015. 2

[203] Ali Torkamani and Nicholas J Schork. Prediction of cancer driver mutations in protein kinases. *Cancer Res.*, 68(6):1675–1682, 2008. 13

[204] Sushil Tripathi, Karen R Christie, Rama Balakrishnan, Rachael Huntley, David P Hill, Liv Thommesen, Judith A Blake, Martin Kuiper, and Astrid Lægreid. Gene ontology annotation of sequence-specific DNA binding transcription factors: setting the stage for a large-scale curation effort. *Database*, 2013:bat062, 2013. doi: 10.1093/database/bat062. 71

[205] Laurens Van der Maaten et al. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(85):2579–2605, 2008. 109

[206] Erwin L van Dijk, Yan Jaszczyszyn, and Claude Thermes. Library preparation methods for next-generation sequencing: tone down the bias. *Exp. Cell. Res.*, 322(1):12–20, 2014. 13

[207] Fabio Vandin, Eli Upfal, and Benjamin J Raphael. Algorithms for detecting significantly mutated pathways in cancer. *J. Comp. Biol.*, 18(3):507–522, 2011. 13, 42

[208] Fabio Vandin, Eli Upfal, and Benjamin J Raphael. De novo discovery of mutated driver pathways in cancer. *Genome Res.*, 22(2):375–385, 2012. 13

[209] Ignacio Varela, Patrick Tarpey, Keiran Raine, Dachuan Huang, Choon Kiat Ong, Philip Stephens, Helen Davies, David Jones, Meng-Lay Lin, Jon Teague, et al. Exome sequencing identifies frequent mutation of the SWI/SNF complex gene PBRM1 in renal carcinoma. *Nature*, 469(7331):539–542, 2011. 18

[210] Charles J Vaske, Stephen C Benz, J Zachary Sanborn, Dent Earl, Christopher Szeto, Jingchun Zhu, David Haussler, and Joshua M Stuart. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM. *Bioinformatics*, 26(12):i237–i245, 2010. 41

[211] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *ECCV*, pages 705–718. Springer, 2008. 84

[212] Cor J. Veenman et al. A maximum variance cluster algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1273–1280, 2002. 99

[213] Bert Vogelstein, Nickolas Papadopoulos, Victor E Velculescu, Shibin Zhou, Luis A Diaz, and Kenneth W Kinzler. Cancer genome landscapes. *Science*, 339(6127):1546–1558, 2013. 1, 2, 4, 13, 14, 40, 54, 55, 71, 73, 118

[214] Bert Vogelstein and Kenneth W. Kinzler. The path to cancer –three strikes and you're out. *N. Engl. J. Med.*, 373(20):1895–1898, 2015. 1, 2

[215] Ulrike Von Luxburg. A tutorial on spectral clustering. *Stat. Comp.*, 17(4):395–416, 2007. 85, 94

[216] Nikhil Wagle, Brian C Grabiner, Eliezer M Van Allen, Ali Amin-Mansour, Amaro Taylor-Weiner, Mara Rosenberg, Nathanael Gray, Justine A Barletta, Yanan Guo, Scott J Swanson, et al. Response and acquired resistance to everolimus in anaplastic thyroid cancer. *N. Engl. J. Med.*, 371(15):1426–1433, 2014. 119

[217] Silke Wagner and Dorothea Wagner. Comparing clusterings – an overview. Technical Report 2006-04, Universität Karlsruhe, 2007. 95

[218] Tim Wang, Kıvanç Birsoy, Nicholas W Hughes, Kevin M Krupczak, Yorick Post, Jenny J Wei, Eric S Lander, and David M Sabatini. Identification and characterization of essential genes in the human genome. *Science*, 350(6264):1096–1101, 2015. 118

[219] Lukas D Wartman. A case of me: clinical cancer sequencing and the future of precision medicine. *Molecular Case Studies*, 1(1):a000349, 2015. doi: 10.1101/mcs.a000349. 4

[220] Michael P Washburn, Dirk Wolters, and John R Yates. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nat. Biotechnol.*, 19(3):242–247, 2001. 4

[221] Takuya Watanabe, Sumihito Nobusawa, Paul Kleihues, and Hiroko Ohgaki. IDH1 mutations are early events in the development of astrocytomas and oligodendrogliomas. *Am. J. Pathol.*, 174(4):1149–1153, 2009. 117

[222] Robert Weinberg. *The biology of cancer*. Garland Science, 2 edition, 2013. 65, 118

[223] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nat. Genet.*, 45(10):1113–1120, 2013. 60

[224] Henrica MJ Werner, Gordon B Mills, and Prahlad T Ram. Cancer systems biology: a peek into the future of patient care? *Nat. Rev. Clin. Oncol.*, 11(3):167–176, 2014. 4

[225] Kimberly C Wiegand, Sohrab P Shah, Osama M Al-Agha, Yongjun Zhao, Kane Tse, Thomas Zeng, Janine Senz, Melissa K McConechy, Michael S Anglesio, Steve E Kalloger, et al. ARID1A Mutations in Endometriosis-Associated Ovarian Carcinomas. *N. Engl. J. Med.*, 363(16):203–211, 2010. 18

[226] Mathias Wilhelm, Judith Schlegl, Hannes Hahne, Amin Moghaddas Gholami, Marcus Lieberenz, Mikhail M Savitski, Emanuel Ziegler, Lars Butzmann, Siegfried Gessulat, Harald Marx, et al. Mass-spectrometry-based draft of the human proteome. *Nature*, 509(7502):582–587, 2014. 4

[227] David Wishart. Mode analysis: a generalization of nearest neighbor which reduces chaining effects. In A.J. Cole, editor, *Numerical Taxonomy*, pages 282–311. Academic Press, 1969. 84

[228] Tobias Wittkop, Dorothea Emig, Sita Lange, Sven Rahmann, Mario Albrecht, John H Morris, Sebastian Böcker, Jens Stoye, and Jan Baumbach. Partitioning biological data with transitivity clustering. *Nat. Methods*, 7(6):419–420, 2010. 99

[229] Christian Wiwie, Jan Baumbach, and Richard Röttger. Comparing the performance of biomedical clustering methods. *Nature methods*, 12(11):1033–1038, 2015. 98, 99

[230] Song Wu, Scott Powers, Wei Zhu, and Yusuf A. Hannun. Substantial contribution of extrinsic risk factors to cancer development. *Nature*, 347(6217):78–81, 2015. 2

[231] Murry W Wynes, Trista K Hinz, Dexiang Gao, Michael Martini, Lindsay A Marek, Kathryn E Ware, Michael G Edwards, Diana Böhm, Sven Perner, Barbara A Helfrich, et al. FGFR1 mRNA and protein expression, not gene copy number, predict FGFR TKI sensitivity across all lung cancer histologies. *Clin. Cancer Res.*, 20(12):3299–3309, 2014. 66

[232] Chen Xu and Zhengchang Su. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12):1974–1800, 2015. 84

[233] Wei Xu, Hui Yang, Ying Liu, Ying Yang, Ping Wang, Se-Hee Kim, Shinsuke Ito, Chen Yang, Pu Wang, Meng-Tao Xiao, et al. Oncometabolite 2-hydroxyglutarate is a competitive inhibitor of α-ketoglutarate-dependent dioxygenases. *Cancer cell*, 19(1):17–30, 2011. 117

[234] Hai Yan, D Williams Parsons, Genglin Jin, Roger McLendon, B Ahmed Rasheed, Weishi Yuan, Ivan Kos, Ines Batinic-Haberle, Siân Jones, Gregory J Riggins, et al. IDH1 and IDH2 mutations in gliomas. *N. Engl. J. Med.*, 360(8):765–773, 2009. 18

[235] Ahrim Youn and Richard Simon. Identifying cancer driver genes in tumor genome sequencing studies. *Bioinformatics*, 27(2):175–181, 2011. 13

[236] Dimas Yusuf, Stefanie L Butland, Magdalena I Swanson, Eugene Bolotin, Amy Ticoll, Warren A Cheung, Xiao Y Cindy Zhang, Christopher TD Dickman, Debra L Fulton, Jonathan S Lim, et al. The transcription factor encyclopedia. *Genome Biol.*, 13(3):R24, 2012. doi: 10.1186/gb-2012-13-3-r24. 54

[237] Travis I Zack, Steven E Schumacher, Scott L Carter, Andrew D Cherniack, Gordon Saksena, Barbara Tabak, Michael S Lawrence, Cheng-Zhong Zhang, Jeremiah Wala, Craig H Mermel, et al. Pan-cancer patterns of somatic copy number alteration. *Nat. Genet.*, 45(10):1134–1140, 2013. 60, 69

[238] Charles T Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 100(1):68–86, 1971. 99, 101

[239] Achim Zeileis, Kurt Hornik, Alex Smola, and Alexandros Karatzoglou. kernlab-an S4 package for kernel methods in R. *J. Stat Softw.*, 11(9):1–20, 2004. 99

[240] Hufeng Zhou, Jingjing Jin, Haojun Zhang, Bo Yi, Michal Wozniak, and Limsoon Wong. IntPath–an integrated pathway gene relationship database for model organisms and important pathogens. *BMC Syst. Biol.*, 6(Suppl 2):S2, 2012. doi: 10.1186/1752-0509-6-S2-S2. 54