UNIVERSITY OF CALIFORNIA,
IRVINE

# Machine Learning for High Throughput Genomic Data Analysis

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Yi Li

Dissertation Committee:
Professor Xiaohui Xie, Chair
Professor Pierre Baldi
Professor Yongsheng Shi

2016

# DEDICATION

To my dad and my wife

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

This thesis would not have been possible without help and support from many people.

First of all, I deeply thank my academic advisor Professor Xiaohui Xie. I thank him for taking me into the computer science department of UC Irvine. Without this chance, my life would have been much different. Xiaohui leads me into the field of machine learning and bioinformatics and has been intensively involved in all the work presented in this thesis. He shapes my taste and style of research and teaches me what is good science and what is worth pursuing. His critical thinking and strict requirements for my work builds up my machine learning foundation for not just bioinformatics but also any other fields. His insights help me get around each roadblock throughout my PhD research. Besides academia, he also helps me tremendously in getting connected with industry. I owe my deep gratitude to my advisor in many ways.

I thank all previous and current members in Xie lab. Jacob Biesinger was literally my second PhD adviser during my first year at the lab. Jake had helped me with all aspects, whether in general bioinformatics research problems or specific programming details. He also helped manage and support our lab in a lot of sense. I personally felt more motivated and productive when working with him. Daniel Newkirk gave me one of his personal computers, which I have used for my first three years with two papers published. Daniel Quang gave me lots of insightful comments on my research from biology side and always helped me proof reading my manuscripts. Lingjie Weng taught me lots of useful experiences in bioinformatics and referred me for internship at LinkedIn. Yifei Chen helped me finished most part of the gene expression inference project. The five years in Xie lab is one of my most memorable time.

Many thanks to my friends, Wei Ping, Mengfan Tang, Yuxiao Wang, Qiang Liu and Qi Lou. With all you guys being around, the journey towards PhD is not boring at all!

I would like to thank the committee members of my dissertation. Professor Pierre Baldi has taught an excellent course on deep learning during the time I was doing the research of gene expression inference with deep learning. I could easily practice all the materials covered in the course on my research and they have been very helpful. Professor Yongsheng Shi gave me very constructive advices on the research of polyadenylation analysis, without which the work would have not been published. I would also like to thank my collaborators, Ali Mortazavi, Ken Cho, Aravind Subramanian for their help on my different research projects.

To all my previous mentors/supervisors: Victor Chen at LinkedIn, Jared Maguire at Counsyl, Nirmal Keshava and Zhongwu Lai at AstraZeneca, and Weiliang Zhu at Shanghai Institute of Materia Medica, thank you so much for your guidance.

I gratefully acknowledge Oxford University Press and BioMed Central for giving me permission to incorporate my publications into my dissertation. The work in Chapter 2, Chapter 3 and Chapter 4 was partly supported by National Institute of Health grant R01HG006870. The work in Chapter 5 was partly supported by National Institutes of Health P50GM76516

and National Science Foundation DBI-0846218. I would also like to acknowledge dbGaP repository for providing the cancer sequencing datasets. The accession numbers for the breast cancer, pancreatic cancer and prostate cancer datasets presented in the thesis are phs000369.v1.p1, phs000516.v1.p1 and phs000447.v1.p1, respectively.

Finally, I owe my greatest thanks to my parents, my wife and my daughter. My dad has continuously supported me throughout my PhD program both financially and emotionally. I feel fearless to take any adventures ahead of me with he backing me up. My wife keeps everything in order in my life and faces any difficulties together with me in my life. My daughter gave me the kind of happiness I have never experienced before and has deeply changed me. I thank my family for all your unconditional love.

# CURRICULUM VITAE

## Yi Li

### EDUCATION

**Doctor of Philosophy in Computer Science** **2016**
University of California, Irvine *Irvine, California, US*

**Master of Science in Computer Science** **2013**
University of California, Irvine *Irvine, California, US*

**Bachelor of Engineering in Bioinformatics** **2009**
Huazhong University of Science and Technology *Wuhan, Hubei, China*

### RESEARCH EXPERIENCE

**Graduate Research Assistant** **2012–2016**
University of California, Irvine *Irvine, California, US*

### TEACHING EXPERIENCE

**ICS 031 Introduction to Programming** **Fall 2012**
**CS 141 Concepts in Programming Languages I** **Spring 2015**
University of California, Irvine *Irvine, California, US*

## REFEREED JOURNAL PUBLICATIONS

**Understanding sequence conservation with deep learning** **2016**
Y Li, D Quang, X Xie. Manuscript in preparation

**Gene expression inference with deep learning** **2016**
Y Li, Y Chen, R Narayan, A Subramanian, X Xie. Bioinformatics

**Poly (A) code analyses reveal key determinants for tissue-specific mRNA alternative polyadenylation** **2016**
L Weng, Y Li, X Xie, Y Shi. RNA

**MixClone: a mixture model for inferring tumor subclonal populations** **2015**
Y Li, X Xie. BMC Genomics

**Genome-wide view of TGF beta/Foxh1 regulation of the early mesendoderm program** **2014**
WT Chiu, RC Le, IL Blitz, MB Fish, Y Li, J Biesinger, X Xie, KWY Cho. Development

**Deconvolving tumor purity and ploidy by integrating copy number alterations and loss of heterozygosity** **2014**
Y Li, X Xie. Bioinformatics

**A mixture model for expression deconvolution from RNA-seq in heterogeneous tissues** **2013**
Y Li, X Xie. BMC Bioinformatics

## SOFTWARE

**DeepCons** `https://github.com/uci-cbcl/DeepCons`
*Sequence conservation analysis with deep learning.*

**D-GEX** `https://github.com/uci-cbcl/D-GEX`
*Gene expression inference with deep learning.*

**PyLOH** `https://github.com/uci-cbcl/PyLOH`
*Deconvolving tumor purity and ploidy by integrating copy number alterations and loss of heterozygosity.*

**MixClone** `https://github.com/uci-cbcl/MixClone`
*A mixture model for inferring tumor subclonal populations.*

**TEMT** `https://github.com/uci-cbcl/TEMT`
*Transcripts abundances estimation from heterogeneous tissue sample of RNA-Seq data.*

**polyAcode** `https://github.com/uci-cbcl/polyAcode`
*Tissue-specific Poly (A) code analyses.*

# ABSTRACT OF THE DISSERTATION

Machine Learning for High Throughput Genomic Data Analysis

By

Yi Li

Doctor of Philosophy in Computer Science

University of California, Irvine, 2016

Professor Xiaohui Xie, Chair

Machine learning methods have been successfully applied to computational biology and bioinformatics for decades with both unsupervised learning and supervised learning. Recent advancement in high throughput genomic data profiling, such as high throughput sequencing and large-scale gene expression profiling, has became a powerful tool for both fundamental biological research and medicine. For example, high throughput sequencing now is possible to sequence billions of bases both fast and cheap, such as Illumina's latest sequencer HiSeq X that can sequence 32 human genomes per week with each costing less than $1000. With the generation of millions or even billions of signals (e.g. sequencing reads) per experiment and thousands or even millions of experiments per study (e.g. large-scale gene expression profiling), there arises a great need for more advanced machine learning models for analysing high throughput genomic data using both unsupervised and supervised learning methods. In this thesis, we try to solve two main challenges in high throughput genomic data analysis, 1) deconvolving the sequencing data from more than one cell population, e.g. heterogeneous tumor tissues, using unsupervised probabilistic learning methods such as mixture models with latent variables; 2) modelling the nonlinear and hierarchical patterns within high throughput genomic data using supervised deep learning methods such as convolutional neural networks. We present five new models to solve these two challenges, each of them is applied to a specific problem. The first three models focus on deconvolving tumor heterogeneity: Chapter

2 presents a probabilistic model to deconvolve tumor purity and ploidy; Chapter 3 further extends the model to infer tumor subclonal populations; Chapter 4 presents a probabilistic model to deconvolve tumor transcriptome expression. The last two models focus on applying deep learning methods in analysing large scale genomic data: Chapter 5 presents a deep learning method for gene expression inference; Chapter 6 presents a deep learning method to understand sequence conservation.

# Chapter 1

# Introduction

One of the fundamental problems in computational biology is to extract useful information from biology experimental data and essentially summarize testable knowledge. Preferred methods for solving this problem should allow us to summarize knowledge in the format of abstract models with manageable parameters that are learned from the experimental data. It is also favorable that the learned models are able to make predictions about the biology system with reasonable accuracy given new data. With these two goals in mind, machine learning methods are often the choice to solve computational biology problems that are defined with rich datasets. Machine learning in general refers to the research and study of algorithms that can learn from data and subsequently make predictions about data by training a parametrized model [67]. Depending on the learning task, machine learning methods can be broadly divided into two categories, supervised learning and unsupervised learning. In supervised learning, the algorithm is presented with both input examples and desired outputs, such as categorical labels or numerical targets, and the goal is to learn a mapping from inputs to outputs. In unsupervised learning, the algorithm is presented with only input examples without desired outputs, and the goal is to discover structures and patterns within the input examples. Both supervised learning and unsupervised learning methods

have been successfully applied to computational biology and bioinformatics for decades [8]. In genomics, supervised learning methods have been used to predict genes, their locations and structures, e.g. translation initiation codon prediction using support vector machine (SVM) [86, 151]. In proteomics, supervised learning methods have been used to predict the secondary structures of proteins, e.g. nearest neighbour [65] and Bayesian classifier [142]. In gene expression analysis, unsupervised learning methods, e.g. hierarchical clustering, are widely used to discover expression patterns of certain disease such as cancers [126]. In systems biology, probabilistic graphical models, which are also advanced unsupervised learning methods, are used for cellar networks inference [43].

Recent advancement in high throughput genomic data profiling, such as high throughput sequencing (HTS) and large-scale gene expression profiling, has became a powerful tool for both fundamental biological research and medicine. For HTS, on January 2014, Illumina announced its new state-of-the-art sequencing instrument, the HiSeq X Ten Sequencing System, that is capable of sequencing human genomes at $1000 each, with a throughput of 600 billion base pairs per day. For large-scale gene expression profiling, researchers from Broad institute have developed the L1000 Luminex bead technology to measure the expression of about 1000 genes, with a fairly low cost of $5 per profile [100]. With the L1000 technology, the NIH LINCS program has generated ˜1.3 million gene expression profiles under a variety of experimental conditions. With the generation of millions or even billions of signals (e.g. sequencing reads) per experiment and thousands or even millions of experiments per study (e.g. large-scale gene expression profiling), there arises a great need for more advanced machine learning models for analysing high throughput genomic data using both unsupervised and supervised learning methods.

In this thesis, we try to solve two main challenges in high throughput genomic data analysis, 1) deconvolving the sequencing data from heterogeneous tumor tissues, using unsupervised probabilistic learning methods; 2) modelling the nonlinear and hierarchical patterns within

genomic data using supervised deep learning methods.

## 1.1 Tumor heterogeneity

One of the fundamental limitations of the current HTS techniques is that, a sufficient amount of DNA or RNA material is required for sequencing. Therefore, most of HTS applications have used a mixture of cells with different populations as the start material. Thus reads from the sequencer often come from sources of more than one cell population. This limitation is particularly prominent for tumor samples, that they consist of a mixture population of tumor cells and surrounding normal cells. More importantly, tumor cells themselves are also often heterogeneous that consist of multiple subclonal populations [22]. The landscape of both genomic and transcriptomic profiles of these different subclonal populations are often distinct. Characterizing genomic and transcriptomic features of each subclonal population is important for both understanding the evolution path of heterogeneous tumor tissues, and for designing more effective drug treatments as some subclones may have pre-existing mutations that could lead to drug resistance [44]. In this thesis, we present three new probabilistic models to solve different aspects of the tumor heterogeneity problem.

- **Chapter 2 focuses on deconvolving tumor purity and ploidy.** A prominent problem in the analysis of cancer genome sequencing data is deconvolving the mixture to identify the reads associated with tumor cells. Solving the problem is, however, challenging because of the so-called 'identifiability problem', where different combinations of tumor purity and ploidy often explain the sequencing data equally well. We propose a new model to resolve the identifiability problem by integrating two types of sequencing information, somatic copy number alterations and loss of heterozygosity, within a unified probabilistic framework. We derive algorithms to solve our model, and implement them in a software package called PyLOH. We benchmark the performance

of PyLOH using both simulated data and 12 breast cancer sequencing datasets and show that PyLOH outperforms existing methods in disambiguating the identifiability problem and estimating tumor purity. The PyLOH package is written in Python and is publicly available at `https://github.com/uci-cbcl/PyLOH`. This chapter is a revision of the original publication [81].

- **Chapter 3 focuses on inferring tumor subclonal populations.** In addition to estimate tumor purity in the scenario of mixture of normal and tumor cells within tumor samples, complete characterization of all subclonal types is a fundamental need in tumor genome analysis. With the advancement of next-generation sequencing, computational methods have been developed to infer tumor subclonal populations directly from cancer genome sequencing data. Most of these methods are based on sequence information from somatic point mutations, However, the accuracy of these algorithms depends crucially on the quality of the somatic mutations returned by variant calling algorithms, and usually requires a deep coverage to achieve a reasonable level of accuracy. We describe a novel probabilistic mixture model, MixClone, for inferring the cellular prevalences of subclonal populations directly from whole genome sequencing of paired normal tumor samples. MixClone integrates sequence information of somatic copy number alterations and allele frequencies within a unified probabilistic framework. We demonstrate the utility of the method using both simulated and real cancer sequencing datasets, and show that it significantly outperforms existing methods for inferring tumor subclonal populations. The MixClone package is written in Python and is publicly available at `https://github.com/uci-cbcl/MixClone`. This chapter is a revision of the original publication [82].

- **Chapter 4 focuses on deconvolving tumor transcriptome expression from RNA-Seq data.** Besides DNA sequencing data, we also developed a probabilistic model-based approach, Transcript Estimation from Mixed Tissue samples (TEMT), to

estimate the transcript abundances of tumor cells from RNA-seq data of heterogeneous tumor tissue samples. TEMT incorporates positional and sequence specific biases, and its online EM algorithm only requires a runtime proportional to the data size and a small constant memory. We test the proposed method on both simulation data and recently released ENCODE data, and show that TEMT significantly outperforms current state of the art methods that do not take tumor heterogeneity into account. TEMT is written in Python, and is publicly available at `https://github.com/uci-cbcl/TEMT`. This chapter is a revision of the original publication [80].

## 1.2 Deep learning

Recent successes in deep learning on many machine learning tasks have demonstrated its power in learning hierarchical nonlinear patterns on large scale datasets [14]. Deep learning in general refers to methods that learn a hierarchical representation of the data through multiple layers of abstraction (e.g. multi-layer feedforward neural networks). A number of new techniques have been developed recently in deep learning, including the deployment of General-Purpose Computing on Graphics Processing Units (GPGPU) [30, 32], new training methodologies, such as dropout training [56, 10]. With these advances, deep learning has achieved state-of-the-art performances on a wide range of applications, both in traditional machine learning tasks such as computer vision [69], natural language processing [125], speech recognition [55], and in natural science applications such as exotic particles detection [9] and protein structure prediction [36]. More recently, deep learning has also been successfully applied in solving sequence-based problems in genomics with convolutional neural networks [76, 107, 2, 150]. In this thesis, we present two new deep learning models to perform large-scale gene expression inference and sequence conservation analysis, respectively.

- **Chapter 5 focuses on gene expression inference with deep learning.** Large-

scale gene expression profiling has been widely used to characterize cellular states in response to various disease conditions, genetic perturbations, etc. Although the cost of whole-genome expression profiles has been dropping steadily, generating a compendium of expression profiling over thousands of samples is still very expensive. Recognizing that gene expressions are often highly correlated, researchers from the NIH LINCS program have developed a cost-effective strategy of profiling only ˜1000 carefully selected landmark genes and relying on computational methods to infer the expression of remaining target genes. However, the computational approach adopted by the LINCS program is currently based on linear regression (LR), limiting its accuracy since it does not capture complex nonlinear relationship between expressions of genes. We present a deep learning method (abbreviated as D-GEX) to infer the expression of target genes from the expression of landmark genes. We used the microarray-based Gene Expression Omnibus dataset, consisting of 111K expression profiles, to train our model and compare its performance to those from other methods. In terms of mean absolute error averaged across all target genes, deep learning significantly outperforms LR with 15.33% relative improvement. A gene-wise comparative analysis shows that deep learning achieves lower error than LR in 99.97% of the target genes. We also tested the performance of our learned model on an independent RNA-Seq-based GTEx dataset, which consists of 2921 expression profiles. Deep learning still outperforms LR with 6.57% relative improvement, and achieves lower error in 81.31% of the target genes. D-GEX is available at `https://github.com/uci-cbcl/D-GEX`. This chapter is a revision of the original publication [28].

- **Chapter 6 focuses on understanding sequence conservation with deep learning.** Comparative genomics has been very effective in finding functional elements across the human genome. However, understanding the functional roles of these sequences still remain a challenge, especially in noncoding regions. We present a deep learning approach, DeepCons, to understand sequence conservation. DeepCons is a

convolutional neural network that is trained to classify conserved and non-conserved sequences. We show that the learned convolution kernels of DeepCons can capture rich information with respect to sequence conservation: 1) they match motifs such as CTCF, JUND, RFX3 and MEF2A that are known to be widely distributed within conserved noncoding elements, 2) they have positional bias relative to transcription start sites, transcription end sites and miRNA, and 3) they have strand bias relative to transcription end sites. DeepCons could also be used to score sequence conservation at nucleotide level resolution. We rediscovered known motifs within a given sequence by highlighting each nucleotide regarding their scores. The source code of DeepCons and all the learned convolution kernels in motif format is publicly available online at `https://github.com/uci-cbcl/DeepCons`.

# Chapter 2

# Deconvolving tumor purity and ploidy

## 2.1 Introduction

The advent of next-generation sequencing (NGS) and launch of comprehensive cancer genome sequencing projects [59, 33] have yielded an unprecedented view on the complex landscape of cancer genomes, leading to the discovery of new cancer-causing genes and pathways, and novel therapeutic targets for treating cancers. Analysing the data from cancer genome sequencing remains, however, computationally challenging due to the shear size of the sequencing data and the complexity of the tumor genomes and samples.

Cancer genomes are often characterized by wide-spread somatic copy number alterations (CNAs), where genomic segments are deleted or duplicated one or more times. Identifying somatic copy number alterations associated with specific tumor genomes is of long-standing interest in the study of cancer genomes and is one of the focal points of the cancer genome analysis. Many computational methods have been proposed to discover copy number changes directly from DNA microarrays [103, 89, 83, 149, 20] or sequencing data [23, 29]. However, most of these methods aim at identifying the relative copy numbers of segments of the same

tumor genome. Discovering copy numbers in an absolute scale is biologically more relevant [25], but more challenging. This is due to the fact that the absolute copy number changes can be affected by two confounding factors: a) tumor purity - the fraction of all cancerous cells within a heterogeneous tumor sample, and b) tumor ploidy - the baseline copy number of genomic segments or entire chromosomes [25, 95], both of which are unknown and themselves need to be estimated in order to infer absolute copy number changes. It is possible to estimate tumor purity and ploidy using experimental techniques such as quantitative image analysis [147] and single-cell sequencing [93], however, these techniques are still too expensive or time-consuming to support large-scale studies. Hence, it is of great interest to use computational approaches to estimate tumor purity and ploidy, and consequently absolute copy number changes, directly from NGS data.

Tumor purity and ploidy affect not only copy number changes in different segments of genomes, but also the distribution of allele frequencies in these segments. In the NGS data, these two types of information can be summarized in terms of the total number of reads mapped to each segment (total read count), and the frequencies of reads matching B-alleles (B-allele frequencies) at different sites. Computational methods have been proposed to estimate tumor purity alone [75, 128] or jointly with tumor ploidy [25, 49, 95] based on these two types of information extracted from NGS data.

Depending upon how copy number changes and B-allele frequency information are used, the existing methods can be roughly grouped into two categories: one category of methods utilize B-allele frequencies (BAFs) at somatic mutation sites to estimate tumor purity, including PurityEst [128] and PurBayes [75]. These methods leverage the fact that the BAFs at somatic mutation sites are expected to be around 0.5 if the tumor purity is 100%, and any addition of normal cells will lead to a reduction in the observed BAFs at these sites. The second category of methods rely instead on copy number changes to estimate tumor purity and/or ploidy, including CNAnorm [49], THetA [95], and ABSOLUTE [25]. It has been

shown that the methods in the second category are often more accurate and robust than those in the first category due to the fact that a) the total read counts are very large in NGS data, and thus methods relying on copy number changes are statistically more stable than methods relying on BAFs at somatic mutation sites, the number of which is often very small, and b) the determination of somatic mutations is not perfect and the inclusion of false positives can significantly bias the estimation [95, 113, 66].

However, the utility of the methods relying on copy number changes to estimate tumor purity and ploidy is severely hindered by the so called "identifiability problem", where different combinations of tumor purity and ploidy can explain the observed data equally well [25, 95, 109]. This is because tumor purity and ploidy are often intertwined — changes in one can be offset by compensations from the other, allowing the same copy number to be explained by multiple combinations of tumor purity and ploidy. For example, a homozygous deletion combined with 30% tumor purity can also be explained as a heterozygous deletion combined with 60% tumor purity. Resolving this ambiguity is key to accurate estimation of tumor purity and ploidy. Existing methods try to solve this identifiability problem by either using heuristics, e.g., favoring solutions that have the smallest deviations from diploid (e.g., CNAnorm) [49], seeking additional experiential data (e.g., ABSOLUTE) [25], or simply outputting all possible solutions (e.g., THetA) [95].

Here we provide a more principled way to solve the identifiability problem by combining the information revealed from copy number changes and B-allele frequencies. Instead of using B-allele frequencies extracted from somatic mutation sites as in the previous cases, we use B-allele frequencies calculated at sites that are heterozygous with respect to the normal genomes, and most of which are common SNPs. These heterozygous sites are much more abundant [117] and easier to identify, leading to more statistically stable results. Copy number changes in the cancer genome often result in loss of heterozygosity (LOH) at these heterozygous sites, and the extent of LOH is closely related to absolute instead of relative

copy number changes. We will use BAFs to gauge the extent of LOH, and provide information on the absolute copy number changes by examining the patterns of BAFs at the heterozygous sites within the same genomic segment. For example, although a homozygous deletion with 30% tumor purity results in the same copy number as a heterozygous deletion with 60% tumor purity in a tumor sample, B-allele frequencies at heterozygous sites of the tumor sample cluster at different values in the two combinations, and therefore are able to distinguish these two cases. Based on this insight, we propose a full probabilistic model implemented as a software package called PyLOH to integrate the information gathered from CNAs and LOH. Estimations of tumor purity and absolute copy numbers are then formulated as an optimization problem in which we choose those values that maximally explain both total read counts and B-allele frequency information.

Our method is similar in spirit to some of the earlier methods proposed for SNP array analysis, where both the signal intensity and BAF of each SNP are used in estimating copy number changes. The combination of these two signals has been shown to improve the estimation accuracy of tumor ploidy [47], or both tumor purity and ploidy [134, 144, 108]. Recently, some of these methods have been extended to sequencing data, including OncoSNP-SEQ by Yau et al. [143] and Patchwork by Mayrhofer et al. [87]. However, the OncoSNP-SEQ algorithm only utilizes the reads mapped to the SNP sites, while our algorithm uses all reads, and thus should be able to yield a more accurate estimation of copy number changes. Similar to our work, the Patchwork algorithm also uses all reads, but it requires manual interpretation through data visualization to determine the initial copy numbers of clusters of genomic segments, which could be useful when the tumor genome is too complex for the algorithms to resolve different solutions by themselves. Here we seek an alterative approach that is based on a generative model and requires no manual intervention. In addition, the Patchwork algorithm requires the existence of copy-neutral loss of heterozygosity within the tumor genome in order to run the algorithm, while our algorithm has no such constraints.

The outline of this article is as follows: in the methods section, we describes the full probabilistic model of PyLOH. In the results section, we first present cluster patterns of BAFs in NGS data of paired tumor-normal samples, then introduce a visualization tool called "BAF heat map" to characterize such patterns. Finally, we compare tumor purity estimates of PyLOH and other methods on both simulated datasets and 12 breast cancer sequencing datasets. Our results show that explicitly incorporating both CNAs and LOH information can resolve the identifiability problem and significantly improve the accuracy of tumor purity estimation. Finally, we discuss the limitations of PyLOH and propose future directions in the discussion section.

## 2.2 Methods

In this section, we present the probabilistic model of PyLOH which combines CNAs and LOH information to infer absolute copy numbers and tumor purity. We first introduce some notations, then propose a generative mixture model incorporating both total read counts and B-allele frequency information, and finally introduce algorithms to solve the model.

### 2.2.1 Basic definitions and notations

Similar to previous work [95, 25], we assume the tumor genome has already been segmented into $J$ segments, each of which has the same copy number alterations. Denote the copy number of the $j$-th segment of the tumor genome by $C_j$ with $j = 1, \cdots, J$. In addition, we assume each segment has a number of heterozygous sites (single nucleotide changes) in the corresponding normal (i.e., control) genome. We use $(i, j)$ to index the $i$-th heterozygous site in segment $j$ with $i = 1, \cdots, I_j$, where $I_j$ is the total number of heterozygous sites in segment $j$.

The observed data are summarized and grouped into two categories: One category is the copy number information, represented as the total number of reads mapped to each segment. Let $D_j$ denote the number of reads mapped to segment $j$. The second category of observed data is the allele frequency information, represented by the total number of reads matching each of two alleles at a heterozygous site. For notational purpose, for each heterozygous site we define the *A allele* to be the allele matching the reference genome, and the *B allele* to be the corresponding unmatched one. Using the notation from [114], let $a_{ij}$ and $b_{ij}$ denote the number of reads matching A and B alleles, respectively, at site $(i, j)$. Since most of the data we consider are from paired tumor-normal samples, we use a superscript $N$ (from normal samples) and $T$ (from tumor samples) to denote the sample origin of the data. For example, $D_j^T$ and $D_j^N$ will denote the total number of reads mapped to segment $j$ from the tumor and normal samples, respectively.

To account for the contamination of normal cells, we assume the tumor sample yielding the sequence data consists of a mixture of normal and tumor cells. Denote the fraction of tumor cells within the tumor sample by $\phi$, which will also be called tumor purity. Consequently, the average copy number of each segment within the tumor sample is

$$\bar{C}_j = \phi C_j + (1 - \phi)2 \tag{2.1}$$

for $j = 1, \cdots, J$, assuming that the default copy number within normal cells is always 2. Our goal is to use both the total read count information and site-specific allele count information to infer both the absolute copy number $\{C_1, \cdots, C_J\}$ and the tumor purity $\phi$.

### 2.2.2 Modeling copy number alterations

Following the Lander-Waterman theory [72], the probability of a read originating from a specific segment depends on three main factors: 1) the copy number of the segment, 2) the

total genomic length of the segment, and 3) the mappability of the segment (depending on factors such as GC content, repetitive sequence, and so on) [95]. Borrowing the concept of interval weight factor from [95], we associate a coefficient $\theta_j$ to segment $j$ accounting for the effect of its genomic length and mappability. We assume the expected number of reads mapped to segment $j$, denoted by $\lambda_j$, in the tumor sample is proportional to $\bar{C}_j\theta_j$. That is, given two segments $a$ and $b$, we have

$$\frac{\lambda_a}{\lambda_b} = \frac{\bar{C}_a\theta_a}{\bar{C}_b\theta_b} \tag{2.2}$$

In Eq. (2.2), the mapping coefficient $\theta_j$'s matters only in their relative values. For simplicity, we take $\theta_a/\theta_b = D_a^N/D_b^N$, the ratio of the mapped read counts between these two segments in the normal sample, since it reflects intrinsic sequence properties of these segments and therefore should be the same between the normal and tumor samples.

The above formula determines the relative value of the expected number of reads mapped to each segment. To further specify the absolute value of $\lambda_j$ of segment $j$, we make use of the allele frequency information, and curate a list of segments that contain no loss of heterozygosity. Where there is no loss of heterozygosity, the only possible copy numbers at these segments in tumor cells must be even numbers. From the list, we further remove "outlier" segments whose copy numbers deviate from the bulk of the segments in the list based on the observed read counts at these segments. At the end, we are left with a set of segments (denoted by set $S$ containing the indices of these segments) that both contain no loss of heterozygosity and likely share the same copy number. Details are given in supplementary information *Data preprocessing* [81].

The set of segments in $S$ will be the baseline segments that we use to specify the expected read counts $\lambda_j$'s. To reduce complexity, we assume that the same even copy number $c_s$ shared by all segments in $S$ can only be either 2 or 4. (The other possible values are 0 for

homozygous deletion, which is unlikely since each segment in $S$ is supported by a certain amount of reads, or values that are greater than 4 for ploidy higher than tetraploid, which is likely to be rare.) Our algorithm will check both cases and select the one most compatible with the observed data (in terms of the likelihood function). Given the values of $c_s$'s for each $s \in S$, the average copy number of these segments in the tumor sample, taking the contamination of normal cells into account, is then given by Eq. (2.1).

With the average copy numbers in the baseline segments given, we then specify the expected read count for each segment $j = 1, \cdots, J$ in the tumor sample as follows

$$\lambda_j = \frac{1}{|S|} \sum_{s \in S} \frac{\bar{C}_j \theta_j}{\bar{C}_s \theta_s} D_s^T \tag{2.3}$$

which is the average expected read count suggested by the baseline segments through Eq. (2.2), where the observed read counts in segment $s$ of the tumor sample are denoted as $D_s^T$. Here $|S|$ denotes the number of segments in set $S$.

Given the expected read count at each segment, we model the probability of observing $D_j^T$ reads in segment $j$ as a Poisson distribution with parameter $\lambda_j$,

$$D_j^T \mid C_j, \phi \sim \text{Poisson}(\lambda_j) \tag{2.4}$$

for each $j = 1, \cdots, J$, where $\lambda_j$ is a parameter depending on the absolute copy numbers and is calculated based on Eq. (2.3). More discussion about using the Poisson distribution is given in supplementary information [81].

### 2.2.3 Modeling loss of heterozygosity

To model the loss of heterozygosity at heterozygous sites (i.e., with genotype AB in the normal cells), we need to consider the genotypes of these sites in tumor cells. Let

$$\mathcal{G} = \{\emptyset, \mathtt{A}, \mathtt{B}, \mathtt{AA}, \mathtt{AB}, \mathtt{BB}, \mathtt{AAB}, \mathtt{ABB}, \mathtt{AABB}\}$$

be the set of possible genotypes that we will consider at each heterozygous site in tumor cells. Note that by focusing on this set we have excluded some other genotypes that are less likely to occur in tumor cells. For instance, we will not consider genotypes AAA or BBB since any copy number change from AB to these two genotypes will involve at least one deletion and two insertions. Instead, all genotypes included in $\mathcal{G}$ can be derived from AB with a minimum of one operation on each allele. Although we formulated the set of possible genotypes here by assuming the maximum copy number of each allele is 2 in tumor cells, PyLOH allows the user to change this value. However, there is a trade-off in choosing the maximum copy number threshold. On one hand, increasing the threshold can accommodate genomes with high instability, but on the other hand, it can also significantly increase the complexity of the model and thus make it more susceptible to overfitting.

The corresponding copy number and frequency of the B alleles (BAFs) associated with each genotype in $\mathcal{G}$ are $\{0, 1, 1, 2, 2, 2, 3, 3, 4\}$ and $\{\frac{1}{2}, \epsilon, 1-\epsilon, \epsilon, \frac{1}{2}, 1-\epsilon, \frac{1}{3}, \frac{2}{3}, \frac{1}{2}\}$, respectively, written in the same order as the genotypes in set $\mathcal{G}$. Note that we have included a small $\epsilon \ll 1$ in the calculation of BAF to account for sequencing and/or read-mapping biases or errors. In practice, we choose $\epsilon = 0.01$, corresponding to a Phred quality score of 20 [41]. We will use $n_g$ and $\mu_g$ to denote the corresponding copy number and BAF, respectively, for genotype $g$.

Since the tumor sample consists of a mixture of normal and tumor cells, the fraction of B alleles in the tumor sample is the weighted average of BAFs between normal and tumor cells,

with weights depending upon tumor purity $\phi$ and copy numbers,

$$\bar{\mu}_g = \frac{\phi n_g \mu_g + (1-\phi)2\mu_0}{\phi n_g + (1-\phi)2} \tag{2.5}$$

where $\mu_0 = 0.5$ is the BAF at the heterozygous sites in normal cells.

Using the notation from [114], let $G_{ij}$ be a random variable denoting the genotype of site $(i, j)$ in tumor cells. Conditional on its genotype, we model the probability of the B allele count at each site as a binomial distribution, that is, given $d_{ij}^T = a_{ij}^T + b_{ij}^T$ reads mapped to site $(i, j)$, the chance of observing $b_{ij}^T$ reads matching B allele is

$$b_{ij}^T \mid G_{ij} = g, \phi \sim \text{Binomial}(d_{ij}^T,\ \bar{\mu}_g) \tag{2.6}$$

with the total number of trials specified by $d_{ij}^T$ and the chance of success at each trial specified by $\bar{\mu}_g$.

### 2.2.4 Combining CNAs and LOH information

For heterozygous sites located within the same segment, their genotypes are constrained by the underlying copy number associated with the segment. We model this constraint through a conditional probability distribution $P(G_{ij} = g|C_j = c) = Q_{gc}$ for all $i$ and $j$. Here $Q_{gc}$ is a predefined matrix specifying the chance of a site being genotype $g$ conditional on the underlying copy number being $c$. In practice, we assign a small probability $\sigma$ to any genotypes incompatible with the copy number $c$ conditional on the heterozygosity in normal cells, and equal probabilities to other compatible genotypes.

Conditional on the underlying copy number, we can then write down the probability of

observing B-allele read count at each site as follows

$$\mathbb{P}(b_{ij}^T|C_j = c, \phi) \;=\; \sum_{g\in\mathcal{G}} Q_{gc}\; \mathbb{P}(b_{ij}^T|G_{ij} = g, \phi)$$

We will assume that conditional on the underlying copy number, the B-allele read counts at different sites of the same segment are independent of each other and are independent of the total read count from the segment. Let $\mathbf{b}_j^T = (b_1^T, \cdots, b_{I_j}^T)$ denote all B-allele read counts at heterozygous sites of segment $j$. Under the conditional independence assumption outlined above, the joint probability of observing $D_j^T$ and $\mathbf{b}_j^T$ conditional on the underlying copy number $C_j = c$ and the tumor purity being $\phi$ is

$$\begin{aligned}\mathbb{P}(D_j^T, \mathbf{b}_j^T|C_j = c, \phi) &= \mathbb{P}(D_j^T|C_j = c, \phi) \\ &\times \prod_{i=1}^{I_j} \sum_{g\in\mathcal{G}} Q_{gc}\; \mathbb{P}(b_{ij}^T|G_{ij} = g, \phi) \end{aligned} \tag{2.7}$$

where the probability of $D_j^T$ conditional on $C_j$ and $\phi$ is the Poisson distribution (2.4), and the probability of $b_{ij}^T$ conditional on $G_{ij}$ and $\phi$ is the binomial distribution (2.6).

### 2.2.5 Likelihood model

So far, we have specified the probability of observing total read count and site-specific B-allele read counts at each segment conditional on the underlying copy number. Next we further treat the copy number $C_j$ at each segment as a random variable, and model its probability as a categorical distribution with support $\mathcal{C} = \{0, 1, 2, 3, 4\}$, denoting the range of considered copy numbers, and parameters $\rho_j = (\rho_{j0}, \cdots, \rho_{j4})$, where $\rho_{jc}$ denotes the probability of having $C_j = c$ in segment $j$. In other words, we have

$$C_j \mid \rho_j \sim \text{Categorical}(\mathcal{C}, \rho_j) \tag{2.8}$$

for each $j = 1, \cdots, J$.

We treat $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_J)$ and $\phi$ as parameters of our model $\Theta = (\phi, \boldsymbol{\rho})$, and the goal of our model is to infer the values of these parameters based on the total read count information in each segment and site-specific allele count information at each heterozygous site of these segments. Let $\mathbf{D} = (D_1, \cdots, D_J)$ and $\mathbf{b} = (\mathbf{b_1}, \cdots, \mathbf{b_J})$. By Eq. (2.7) and assuming the observations from different segments are conditionally independent conditional on the tumor purity $\phi$, the likelihood of observing the combined read count information is then

$$\begin{aligned}\mathbb{P}(\mathbf{D}, \mathbf{b}|\phi, \boldsymbol{\rho}) &= \prod_{j=1}^{J} \sum_{c \in \mathcal{C}} \mathbb{P}(C_j = c|\rho_j) \mathbb{P}(D_j^T, \mathbf{b}_j^T | C_j = c, \phi) \\ &= \prod_{j=1}^{J} \sum_{c \in \mathcal{C}} \rho_{jc} \frac{\lambda_j^{D_j^T} e^{-\lambda_j}}{D_j^T!} \left[ \prod_{i=1}^{I_j} \sum_{g \in \mathcal{G}} Q_{gc} \binom{d_{ij}^T}{b_{ij}^T} \bar{\mu}_g^{b_{ij}^T} (1 - \bar{\mu}_g)^{a_{ij}^T} \right].\end{aligned} \tag{2.9}$$

Given the likelihood function, we can then estimate the model parameters using maximum likelihood estimation. Alternatively, we can also add a prior into the model by incorporating our prior knowledge on the copy numbers and/or tumor purity. For instance, we can use the Dirichlet distribution to incorporate the prior on the distribution of copy numbers, and beta distribution to incorporate the prior on tumor purity,

$$\rho_j \sim \text{Dirichlet}(\boldsymbol{\omega}), \quad \phi \sim \text{Beta}(\alpha, \beta) \tag{2.10}$$

where $\boldsymbol{\omega}$ is a vector having the same dimension as $\rho_j$ and gives a weight to each copy number. If the priors are specified, we can then estimate the values of the parameters by maximizing their posterior probability, i.e., using the method of maximum a posteriori (MAP) estimation. In this article, we use non-informative prior for $\phi$ and a Dirichlet prior configured based on the compatible genotypes of each copy number for $\rho_j$. We solve the MAP problem using the

Expectation-Maximization (EM) framework [35]. An alternative approach would be to take a Bayesian approach to calculate the posterior probabilities of the tumor purity and copy number changes. We do not take the Bayesian approach due to computational considerations as it would require more time-consuming inference procedures. The complete details about prior configurations and EM updates are given in supplementary information [81].

## 2.3 Results

Next we demonstrate the utility of combining loss of heterozygosity with copy number alterations to infer tumor purity and ploidy. For this purpose, we first present a clustering pattern of B-allele frequencies derived from NGS data in paired tumor-normal samples. Then we show that this clustering pattern can be used to resolve the ambiguous combinations of tumor purity and copy number changes, using both a toy example and real data. Afterward, we apply our method PyLOH, developed to infer tumor purity and absolute copy numbers by integrating the information from total read counts and B-allele frequencies, to simulated data and compare its performance to exiting state-of-the-art methods, CNAnorm-1.4.0 [49], THetA-0.0.3 [95] and PurBayes-1.3 [75]. Finally, we test the performance of our and other methods on real data, consisting of 12 whole genome sequencing datasets from breast cancer samples [11].

### 2.3.1 BAFs patterns in NGS data and BAF heat map

As discussed in the introduction, the distribution of BAFs is closely related to the underlying copy number changes. In particular, copy number changes at sites that are heterozygous with respect to the normal genome may result in a deviation of the BAFs from 0.5 (loss of heterozygosity), and the extent of this deviation depends on the absolute copy number

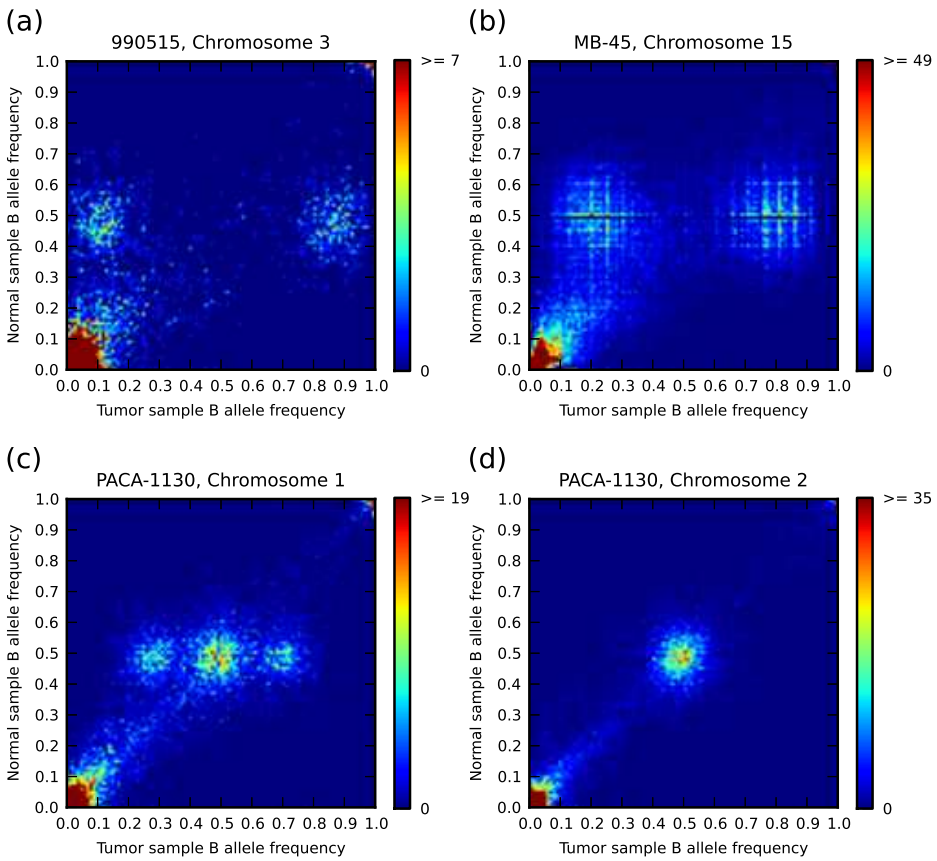



Figure 2.1: Frequencies of BAF pairs in paired tumor-normal samples shown as heat maps. (a) Chromosome 3 of patient 990515 [148]; (b) Chromosome 15 of patient MB-45 [11]; (c) Chromosome 1 of patient PACA-1130 [19]; (d) Chromosome 2 of patient PACA-1130 [19]. The x-axis and the y-axis are divided into 100 bins representing the BAF resolution of 1%, thus each BAF heat map is a 100 by 100 mesh grid. The color of each grid quantifies the number of sites that has such a paired BAF across a specific genomic segment.

changes and tumor purity. We illustrate this idea using a heat map plot (Figure 2.1), which shows the frequencies of BAF pairs, with one calculated from the normal sample and the other calculated from the matched tumor sample at the same site, coded in pseudo colors. Figure 2.1 shows the BAF heat maps of paired tumor-normal samples from three independent cancer genome NGS datasets, including both exome sequencing [19, 148] and whole genome sequencing [11].

The BAF heat maps demonstrate a clear cluster pattern on the distribution of BAF pairs

(Figure 2.1). Two clusters are shared by all four heat maps, including the bottom left cluster, containing sites with homozygous A-allele in both normal and tumor genomes, and the top right cluster, containing sites with homozygous B-allele in both normal and tumor genomes. (Note that the small deviations of BAFs away from 0 or 1 of sites in these two clusters are likely due to sequencing and/or read-mapping errors.) Without changes in BAFs, these two clusters reveal no information with regard to the underlying copy number changes. Thus, we focus our attention on the other clusters in the heat maps, which all have BAFs centering at 0.5 in the normal samples, and thus contain mostly the heterozygous sites that underscore our method. For this reason, these clusters will be referred to as *heterozygous clusters* in the following.

The heterozygous clusters demonstrate distinct cluster patterns in different genomic segments of same/different samples. Although the BAFs of these clusters all center at 0.5 in the normal samples, the BAFs of the corresponding matched tumor samples can center at 0.5 (Figure 2.1cd) or at values away from 0.5 (Figure 2.1abc). In fact, these values provide a measure on the extent of LOH in the segments of tumor samples. For example, the tumor BAFs of the heterozygous cluster center at 0.5 in Figure 2.1d, suggesting no loss of heterozygosity in this segment. Without LOH, the absolute copy number in this case can only be even numbers, with diploid being the most plausible answer. (We can eliminate homozygous deletion since there are reads mapped to this region.)

A different cluster pattern emerges in Figure 2.1ab, which show two heterozygous clusters, with tumor BAFs centering at 0.1 and 0.9 in Figure 2.1a, and at 0.2 and 0.8 in Figure 2.1b, suggesting significant loss of heterozygosity in these two segments. (Note the appearance of two heterozygous clusters in these two cases and the symmetry of the two clusters with respect to the tumor BAF=0.5. This is because the B-alleles are determined according to the human reference genome, which is not phased.) If the underlying copy number changes are a single-copy deletion in both cases, then the cluster with tumor BAFs centering at 0.1

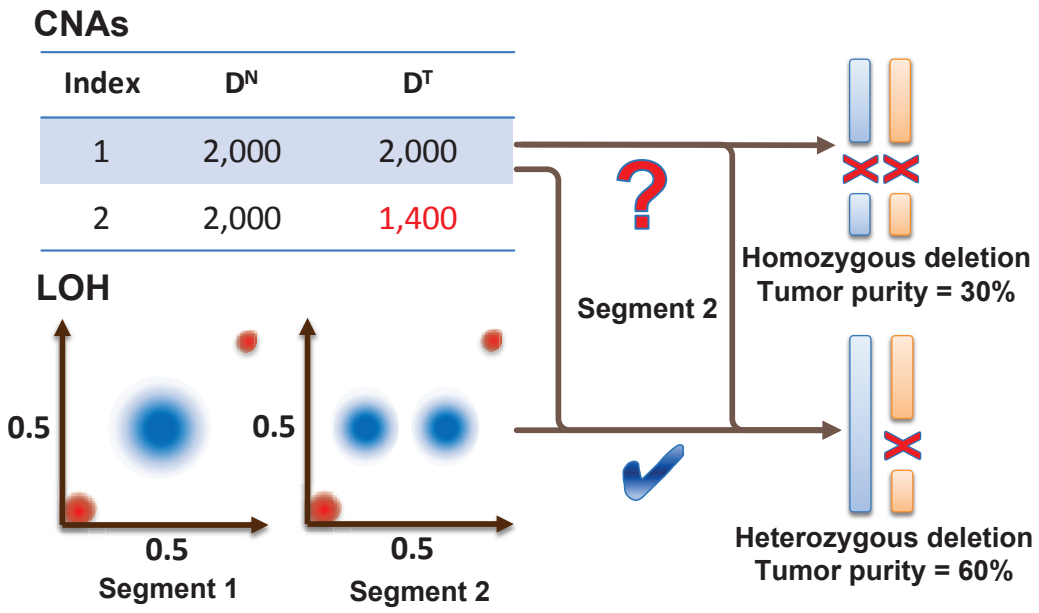



Figure 2.2: A toy example illustrating the utility of BAFs patterns in resolving the identifiability problem.

(Figure 2.1a) would correspond to a larger LOH, and consequently a higher tumor purity than the other case(Figure 2.1b).

Figure 2.1c shows an interesting case with three heterozygous clusters with one center cluster showing no LOH and two symmetric clusters suggesting LOHs. This more complex cluster pattern suggests more than one types of CNAs within the segment being considered, most likely due to the presence of both diploid and single-copy deletion changes.

Overall, these BAF heat maps provide a convenient and intuitive way to examine the overall CNAs of a chromosomal segment, and illustrate the utility of BAFs at heterozygous sites for inferring tumor purity and absolute copy numbers.

Table 2.1: Three SNP sites from the exome sequencing data of patient MB-154.

| Index | Pos | $d_N$ | $d_T$ | $BAF_N$ | $BAF_T$ | dbSNP ID |
|---|---|---|---|---|---|---|
| 1 | chr6:112,147,822 | 141 | 87 | 49% | 25% | rs28763978 |
| 2 | chr7:131,842,835 | 98 | 29 | 51% | 52% | rs156961 |
| 3 | chr7:82,225,896 | 317 | 352 | 50% | 51% | rs62465931 |

$BAF_N$ and $BAF_T$ denote BAFs of the normal and tumor sample, respectively. $d_N$ and $d_T$ denote the read depth at the SNP site of the normal and tumor sample, respectively.

### 2.3.2 Using BAFs to solve the identifiability problem

The BAFs patterns shown in Figure 2.1 can be used to resolve the identifiability problem, as the heterozygous clusters in each BAF heat map will center at different values with respect to different combinations of tumor purity and copy number changes. We demonstrate this idea using a toy example (Figure 2.2). In this example, we have total read counts in two segments of the genome from both normal and tumor samples. The segment 2 has much smaller total read counts from the tumor sample than the normal sample. The differences can be explained by either a heterozygous deletion with 60% tumor purity or a homozygous deletion with 30% tumor purity in this segment. The total read counts themselves cannot distinguish these two possibilities. However, if we add in the information from the BAFs of the sites in segment 2, an observation of heterozygous clusters centering at tumor BAFs away from 0.5 would eliminate the homozygous deletion solution (Figure 2.2).

We can observe similar cases in real cancer genome sequencing data as those in the above toy example. For instance, Table 2.1 shows the total read counts and BAFs at three SNP sites (dbSNP 130 ID listed [120]) observed in the exome sequencing of a breast cancer patient MB-154 [11]. The mean coverage of the exome sequencing data was 141X for the tumor samples and 133X for the normal samples, respectively [11]. The first SNP site shows an example of a heterozygous deletion as $d_T$ is significant lower than $d_N$ while $BAF_T$ significantly deviates from 0.5. The second site shows an example of a homozygous deletion as $d_T$ is significant lower than $d_N$ while $BAF_T$ is around 0.5. As a control, the third site shows an example

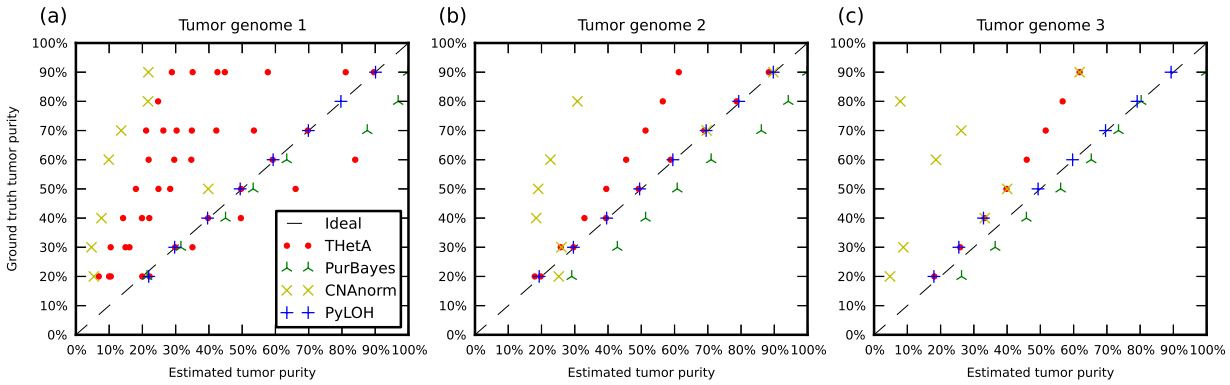



Figure 2.3: The tumor purity estimates of the first three simulated datasets given by THetA, CNAnorm, PurBayes and PyLOH. The x-axis is the estimated tumor purity and the y-axis is the ground truth tumor purity.

without LOH or CNAs.

### 2.3.3 Results from simulated data

We have developed a probabilistic model to infer tumor purity and absolute copy numbers by integrating the LOH information described above and the information based on total read counts (see methods). Next we benchmark the performance of our new method on simulated data and compare it with other algorithms. By using simulated data, we know the ground truth of both tumor purity and absolute copy numbers, thereby providing us an objective way of comparing the performance of different algorithms.

We first created an artificial diploid human genome by using the human reference genome as a template and inserting SNP sites with a frequency similar to those observed in the human population [117]. This diploid genome will be treated as the normal genome in our follow-up simulation and analysis. The tumor genome was generated by adding somatic mutations and copy number changes to the normal genome. NGS reads were then simulated from the tumor sample consisting of a mixture of the normal and tumor genome, with the fraction of

the tumor genome determined by the tumor purity. To reduce computational time, we use only data from chromosome 1 in our analysis. Details on how the genomes and reads were generated are described in supplementary information [81].

We created four tumor genomes that differ in their copy number configurations. The absolute copy numbers of each tumor genome were configured to introduce the identifiability problem and the detailed configurations are given in supplementary Table S1 [81]. For each tumor genome, we then simulated eight different sets of NGS reads from both normal and tumor samples by varying tumor purity. Overall, 32 sets of paired tumor-normal reads, each with 60X coverage, were generated. We applied PyLOH to each of these datasets, and compared its performance to three other methods, including PurBayes, CNAnorm, and THetA. THetA and PyLOH require a segmentation of the tumor genome based on relative CNAs as an input. To avoid issues related to genome segmentation, we used ground truth segmentations in our analyses. Similarly, we used ground truth somatic mutation sites as the input for PurBayes. Details on how reads were preprocessed are given in supplementary information [81].

The tumor purities estimated by PyLOH and three other existing methods are shown in Figure 2.3. Due to the space limitation, we only show the results of the first three simulated datasets in Figure 2.3. The complete tumor purity estimates for all the simulated datasets are shown in supplementary Table S2 [81]. The absolute copy numbers estimated by PyLOH and THetA for each simulated dataset are shown in supplementary Table S4,S5,S6 and S7 [81]. A few observations emerge from the figure and tables. First, PyLOH significantly outperforms the other three methods on these datasets, providing a more accurate estimation of both tumor purity and absolute copy numbers and returning ground true values in most of the tested cases. Second, the THetA method, based only on total read counts information, is able to identify the ground truth as one of its possible solutions for tumor genomes 1 and 2, but fails to resolve the identifiability problem. Third, the PurBayes method, based on information from somatic mutations, can return the true tumor purity in some cases, but

Table 2.2: The tumor purity estimates of the 12 breast cancer whole genome sequencing datasets given by THetA, CNAnorm, PurBayes and PyLOH.

| Patient ID | THetA | CNAnorm | PurBayes | PyLOH | ABSOLUTE |
|---|---|---|---|---|---|
| MB-15 | 0.288 | 0.245 | 0.999 | 0.589 | 0.22 |
| MB-45 | 0.526 | 0.291 | 0.999 | 0.566 | 0.25 |
| MB-50 | 0.193 | 0.224 | 0.999 | 0.532 | 0.47 |
| MB-82 | 0.129 | 0.274 | 0.999 | 0.192 | 0.74 |
| MB-98 | 0.598 | 0.437 | 0.999 | 0.698 | 0.54 |
| MB-106 | 0.409* | 0.135 | 0.999 | 0.831 | 0.89 |
| | 0.817* | | | | |
| MB-116 | 0.325 | 0.510 | 0.769 | 0.325 | 0.66 |
| MB-123 | 0.358 | 0.353 | 0.999 | 0.377 | 0.65 |
| MB-154 | 0.645 | 0.187 | 0.999 | 0.664 | 0.70 |
| MB-165 | 0.668* | 0.172 | 0.999 | 0.662 | 0.68 |
| MB-198 | 0.301 | 0.293 | 0.999 | 0.607 | 0.64 |
| MB-200 | 0.515 | 0.158 | 0.999 | 0.523 | 0.55 |
| MAE# | 0.220 | 0.320 | 0.397 | 0.186 | n/a |

*THetA outputted multiple solutions and here we only show the solutions with the smallest deviation from the diploid.
#For tumor purities reported by THetA with multiple solutions, we used the average of the solutions with the smallest deviation from the diploid to calculate the Mean Absolute Error (MAE).

has larger deviations than PyLOH, likely reflecting the statistical fluctuation associated with relatively small number of somatic mutation sites.

In addition to the four simulated tumor genomes discussed above, we simulated two additional tumor genomes based on copy number configurations derived from Sanger COSMIC v68 [42]. The tumor purities and absolute copy numbers of the two COSMIC samples estimated by PyLOH and other methods are shown in Supplementary Table S9,S10 and S11 [81]. PyLOH still outperformed the other methods on these two new datasets. Further details on these two dataset are described in supplementary information [81].

### 2.3.4 Results from breast cancer sequencing data

Having illustrated the utility of our method on simulated data, we proceed to test the performance of PyLOH on a real cancer genome dataset, consisting of whole-genome sequencing of 12 breast cancer samples [11]. As the ground truth tumor purities are unknown for this dataset, we used the tumor purities calculated by ABSOLUTE based on SNP array data and reported in the paper by Bajerji et al. [11] as a baseline for our comparison. Although this baseline is by no means absolutely correct, it offers clues on the performance of different algorithms since it was derived from SNP array data instead of NGS sequencing data as in our case. We used BIC-seq-1.2.1 [139] to obtain segmentation files for THetA and PyLOH, and used VarScan-2.3.5 [66] to call somatic mutation sites for PurBayes. Since THetA often outputs multiple solutions, we selected the ones with the smallest deviation from the diploid whenever this happens, as recommended by THetA [95]. Further details on this dataset are described in supplementary information [81].

The tumor purities estimated by PyLOH and three existing algorithms - THetA, CNAnorm and PurBayes, for the 12 breast cancer sequencing datasets are summarized in Table 2.2. If the tumor purities estimated by ABSOLUTE are used as our comparison baseline, we find PyLOH to be the most accurate algorithm among the four - it yields a mean absolute error (MAE) of 0.186, as compared to a MAE of 0.22 by the second best algorithm, THetA. PurBayes, which utilizes somatic mutations to estimate tumor purity, produced poorest results, likely due to the inclusion of false positives in the somatic mutation calling procedure.

Although PyLOH returned closer solutions to ABSOLUTE (as measured by MAE) than any of the other methods, the tumor purities estimated by PyLOH and ABSOLUTE deviate in six samples: MB-15, MB-45, MB-82, MB-98, MB-116 and MB-123. To find out why such a discrepancy arises, we carefully studied each of these six cases. In two of these cases (MB-15 and MB-45), we believe that the results obtained by PyLOH are more accurate because

the tumor purities and absolute copy numbers inferred by total read counts information are consistent with those inferred by BAFs information, and both support the results obtained by PyLOH. For sample MB-82, MB-116, and MB-123, the contribution of BAFs information to estimating tumor purities is not significant compared with the total read count information, likely due to a low tumor purity in these samples. As a result, the estimation of tumor purity is mainly contributed by information from total read counts, and in fact produces a similar estimation compared to THetA. For the remaining case MB-98, the estimated tumor purities given by the four algorithms are all inconsistent - one possible reason for this may be the existence of subclonal tumor populations in the tumor sample.

Aside from the accuracy comparison described above, we note that PyLOH is very fast, with a running time scaling linearly with the number of segments. This is in contrast to the THetA method, the running time of which scales exponentially with the number of segments since it explores all combinations of copy number changes across all segments [95]. As a result, THetA takes a prohibitively long time to run when the number of segments is above 150 and the maximum copy number is greater than 6, while PyLOH has no such constraints. Further details about the run time of each algorithm are given in supplementary information [81].

## 2.4 Discussion

In this paper, we examined the problem of estimating tumor purity and absolute copy number changes from NGS data, and, in particular, focused on solving the identifiability problem that has not been properly solved by the existing methods. We demonstrated that the distribution of B-allele frequencies at sites that are heterozygous with respect to the normal genome provides key, but underutilized, information to solve the identifiability problem. We further developed a full probabilistic model to integrate the copy number change and BAF

information, and derived a principled way to estimate tumor purity and absolute CNAs. We benchmarked the performance of our method, PyLOH, on both simulated data and real whole-genome sequencing data, showing that our method outperforms existing methods in both cases.

PyLOH requires a segmentation of the genome into segments with different CNAs as input. Many algorithms have been developed to segment genomes based on copy number changes and BAFs of SNP array data with varying levels of accuracy [134, 144, 129, 96]. A few of these array-based methods have recently been translated to the sequencing domain [87, 143]. A future direction of PyLOH would be to integrate these existing methodologies and combine them with the probabilistic model of PyLOH to carry out both genome segmentation and absolute copy number estimation.

Another important future direction is to use our model to study tumor heterogeneity. So far, we have focused on separating genetic changes from a mixture of normal and tumor cells. It is well known that multiple tumor clonal types may coexist in the tumor sample, each with an associated mutation landscape [99]. To further model intra-tumor heterogeneity on top of the current probabilistic framework, we can assume there are multiple populations of tumor cells. Thus the model likelihood given by Eq. (2.9) can be extended to account for subclonal tumor populations (details in supplementary information [81]). We plan to further extend PyLOH in this direction to tackle the more challenging problem of deconvolving tumor heterogeneity by combining copy number change and allele frequency information.

# Chapter 3

# Inferring tumor subclonal populations

## 3.1 Introduction

Tumor genomes have been shown to present extensive cellular heterogeneity for decades since Nowell's original clonal theory for tumor progression [94]. Identifying tumor subclonal populations is important for both understanding the evolution of tumor cells, and for designing more effective treatments as pre-existing mutations occurring in some subclones could lead to drug resistance [44]. For example, a research in lymphocytic leukemia has shown links between the presences of driver mutations within subclones and adverse clinical outcomes [71].

With the advancement of next-generation sequencing (NGS) and launch of large-scale cancer genome sequencing projects [59], computational methods have recently been developed to infer tumor subclonal populations based on cancer genome sequencing data [115, 3, 61, 50, 95].

Most of these methods rely on sequence information from somatic point mutations, such as

PyClone [115], EXPANDS [3], PhyloSub [61] and rec-BTP [50]. Methods in this category leverage the cluster pattern of allele frequencies at somatic point mutations to detect distinct subclonal populations. However, as the determination of somatic point mutations is imperfect and the inclusion of false-positives is unavoidable [113], deep sequencing with more than 100X coverage is often required for subclonal inferences with high sensitivity and specificity [115, 61, 50].

Other approaches utilizing the read depth information from genomic segments with somatic copy number alterations (SCNAs) to infer the cellular prevalences of subclonal populations have also been developed, such as THetA [95]. THetA explores all combinations of copy number changes across all segments to infer the most likely collection of subclonal populations [95]. However, with the copy number information alone, THetA suffers from the "identifiability problem", where distinct combinations of tumor purity and ploidy are able to explain the read depth information from SCNAs equally well [95]. Additionally, the running time of THetA scales exponentially with the number of genomic segments [95], and often takes a prohibitively long time to run under certain parameter settings.

In this article, we present a novel probabilistic mixture model, MixClone, to infer the cellular prevalences of subclonal populations. MixClone integrates both read depth information from genomic segments with SCNAs and allele frequency information from heterozygous single-nucleotide polymorphism (SNP) sites within a unified probabilistic framework. Such integrative framework has been shown to significantly improve the accuracy of tumor purity estimation in our previous work [81]. Here, we present that MixClone achieves two major advantages compared to the existing methods that (i) it does not require deep sequencing data, (ii) it resolves the identifiability problem. To demonstrate MixClone's utility, we conducted simulation studies and showed that it outperforms existing methods. We also applied MixClone on a breast cancer sequencing dataset [11], and showed that it was able to discover subclonal events not reported before.

## 3.2 Methods

In this section, we introduce the generative mixture model of MixClone, which is an extension of our previous work on tumor purity estimation[81]. First, we introduce the notations for input data. Then, we describe the probabilistic models for sequence information of both SCNAs and allele frequencies. Finally, we combine these two types of data into a single likelihood model, and describe an algorithm to solve the model.

### 3.2.1 Basic notations

The raw input data for MixClone are two aligned whole genome sequencing read sets of paired normal-tumor samples and a genome segmentation file based on the tumor sample. Following the notations from our previous work [81], we assume the tumor genome has been partitioned into $J$ segments. We also assume there are $I_j$ heterozygous SNP sites within segment $j$ in the corresponding normal genome, and use $(i, j)$ to index SNP site $i$ within segment $j$. For each SNP site $(i, j)$ we define the `A` allele to be the reference allele and the `B` to be the alternative allele, with respect to the reference genome. We also use a superscript `N` to denote data from normal samples and superscript `T` to denote data from tumor samples. Overall, the observed data are summarized in the following notations [114]:

$b_{ij}^N$ = number of reads mapped to the B allele in the normal sample at site $(i, j)$.

$d_{ij}^N$ = reads depth of the normal sample at site $(i, j)$.

$D_j^N$ = total number of reads mapped to segment $j$ of the normal sample.
The notations for the observed data from tumor samples are similarly defined, e.g. $D_j^T$ denotes total number of reads mapped to segment $j$ of the tumor sample.

### 3.2.2 Modeling SCNAs

Next, we describe the probabilistic model for SCNAs data. For each segment $j$, we define an allelic configuration $H_j$ to represent its underlying allele-specific copy number status. For example, if the absolute copy number of segment $j$ is 2, then the compatible allelic configurations are `PP`, `MM` and `PM`, where `P` and `M` denotes the paternal and maternal allele of the tumor genome, respectively. Since `PP` and `MM` are not distinguishable based on sequence information alone as the reference human genome is not phased, we define the set of all possible allelic configuration as

$$H_j \in \mathcal{H} = \{\emptyset, \texttt{P/M}, \texttt{PP/MM}, \texttt{PM}, \texttt{PPP/MMM}, \texttt{PPM/PMM}\} \tag{3.1}$$

assuming the maximum copy number for each segment is 3. The corresponding copy number associated with each allelic configuration in $\mathcal{H}$ is then

$$n_h = \{0, 1, 2, 2, 3, 3\} \tag{3.2}$$

MixClone allows the user to specify the maximum copy number and the default value is 6 in the released package [81]. We further assume there are $K$ subclonal populations within the tumor sample, each of which has an associated cellular prevalence $\phi_k \in [0, 1]$. The subclonal type of each segment $j$ is denoted as

$$Z_j \in \mathcal{Z} = \{1, 2, \cdots, K\} \tag{3.3}$$

representing one of the $K$ possible subclonal populations. Given the allelic configuration $H_j = h$ and the subclonal type $Z_j = k$, the average copy number of segment $j$ within the tumor sample, taking into account the subclonal cellular prevalence $\phi_k$, is

$$\bar{C}_j = \phi_k n_h + (1 - \phi_k)2 \tag{3.4}$$

Based on the Lander-Waterman model [72], the probability of sampling a read from a given segment $j$ depends on three main factors: 1) its copy number, 2) its total genomic length, and 3) its mappability, which depends on factors such as repetitive sequence and GC content [95]. For each segment $j$, we associate a coefficient $\theta_j$ to account for the effect of its mappability and genomic length. Thus the expected read counts mapped to segment $j$, which is denoted as $\lambda_j$, is proportional to $\bar{C}_j\theta_j$. For example, for segment $x$ and segment $y$, we have

$$\frac{\lambda_x}{\lambda_y} = \frac{\bar{C}_x\theta_x}{\bar{C}_y\theta_y} \tag{3.5}$$

Because the mappability coefficients ($\theta_j$'s) matter only in a relative sense, we take $\theta_x/\theta_y = D_x^N/D_y^N$, as these segments should have the same sequence properties between the normal and tumor samples.

Additionally, to determine the absolute value of $\lambda_j$, we curate a list of segments which contain no loss of heterozygosity according to their allele frequencies information. Based on the observed number of reads mapped to each segment, we further remove "outlier" segments from the list if their copy numbers are different from the bulk of the segments' copy numbers in the list. Finally, we call the remaining segments in the list as "baseline segments" and denote the set of these segments as $S$. We assume the allelic configurations of all the baseline

segments are PM with copy number $n_s = 2$. Other possible allelic configurations for baseline segments, which have equal copy numbers for each allele (e.g. $\emptyset, \texttt{PPMM}$), are likely to be rare, and currently we do not model them. Then based on $n_s$, we specify $\lambda_j$ as follows

$$\lambda_j = \frac{1}{|S|} \sum_{s \in S} \frac{\bar{C}_j \theta_j}{n_s \theta_s} D_s^T \tag{3.6}$$

where $D_s^T$ denotes the number of reads mapped to segment $s$ of the tumor sample. Finally, we model the number of reads mapped to segment $j$ in the tumor sample as a Poisson distribution, given $H_j$ and $Z_j$

$$D_j^T \mid H_j, Z_j \sim \text{Poisson}(\lambda_j) \tag{3.7}$$

Details on curating the baseline segments are given in Supplementary [82].

### 3.2.3 Modeling allele frequencies

Next, we describe the probabilistic model used for allele frequencies of heterozygous SNP data. For each SNP site $i$ within segment $j$, we denote its tumor genotype as $G_{ij}$, which is selected from the set of all possible tumor genotypes up to a maximum copy number alteration, e.g.

$$\mathcal{G} = \{\emptyset, \texttt{A}, \texttt{B}, \texttt{AA}, \texttt{AB}, \texttt{BB}, \texttt{AAA}, \texttt{AAB}, \texttt{ABB}, \texttt{BBB}\} \tag{3.8}$$

assuming the maximum copy number is 3. The corresponding B allele frequencies (BAF) for all the genotypes in $\mathcal{G}$ are

$$\mu_g = \{\frac{1}{2}, \epsilon, 1-\epsilon, \epsilon, \frac{1}{2}, 1-\epsilon, \epsilon, \frac{1}{3}, \frac{2}{3}, 1-\epsilon\} \tag{3.9}$$

in which, $\epsilon \ll 1$ is a small random deviation accounting for general sequencing errors. We choose $\epsilon = 0.01$, which is equivalent to a Phred quality of 20 [41].

Given the tumor genotype $G_{ij} = g$, the allelic configuration $H_j = h$, and the subclonal type $Z_j = k$, the average BAF of site $(i, j)$ within the tumor sample, taking into account the subclonal cellular prevalence $\phi_k$, is

$$\bar{\mu}_{ij} = \frac{\phi_k n_h \mu_g + (1-\phi_k) 2\mu_0}{\phi_k n_h + (1-\phi_k) 2} \tag{3.10}$$

in which $\mu_0 = 0.5$ is the BAF of heterozygous SNP sites in the normal sample. Finally, we model the distribution of the B allele count $b_{ij}^T$ at site $(i, j)$ as a binomial distribution, given $G_{ij}, H_j$ and $Z_j$

$$b_{ij}^T \mid d_{ij}^T,\ G_{ij}, H_j, Z_j \sim \text{Binomial}(d_{ij}^T,\ \bar{\mu}_{ij}) \tag{3.11}$$

### 3.2.4 Combining SCNAs and allele frequencies

Now, we combine sequence information from both SCNAs and heterozygous SNP sites. For all the heterozygous SNP sites within the same segment, their genotypes should be consistent

with the underlying allelic configuration of the segment. We model this consistency through a predefined conditional probability $Q_{gh} = \mathbb{P}(G_{ij} = g|H_j = h)$. If the genotype $g$ is inconsistent with the allelic configuration $h$, e.g. `AA` is inconsistent with `PM`, we assign a small probability $\sigma$ as $Q_{gh}$, otherwise we assign equal probabilities to genotypes that are consistent with the allelic configuration.

Conditional on the underlying allelic configuration $H_j$ and subclonal type $Z_j$, the probability of observing B allele read count $b_{ij}^T$ at site $(i, j)$ is given as

$$\mathbb{P}(b_{ij}^T|H_j = h, Z_j = k) = \sum_{g\in\mathcal{G}} Q_{gh}\mathbb{P}(b_{ij}^T|G_{ij} = g, H_j = h, Z_j = k) \tag{3.12}$$

We assume that conditional on the allelic configuration $H_j$, the B allele read counts $\{b_{ij}^T\}_{i=1}^{I_j}$ at different sites within the same segment $j$ are independent of each other, and are also independent of the total read count $D_j^T$ of the segment. Then, the joint probability of observing the two types of read counts information of segment $j$ is

$$\begin{aligned} & \mathbb{P}(D_j^T, \{b_{ij}^T\}_{i=1}^{I_j}|H_j = h, Z_j = k) \\ = \; & \mathbb{P}(D_j^T|H_j = h, Z_j = k) \times \prod_{i=1}^{I_j} \sum_{g\in\mathcal{G}} Q_{gh}\mathbb{P}(b_{ij}^T|G_{ij} = g, H_j = h, Z_j = k) \end{aligned} \tag{3.13}$$

### 3.2.5 Likelihood model

We have specified the joint distribution of the two types of read counts information of segment $j$. We then further model the allelic configuration $H_j$ and the subclonal type $Z_j$ of segment

$j$ as random variables that follow categorical distributions

$$H_j \mid \rho_j \sim \text{Categorical}(\rho_j) \tag{3.14}$$

$$Z_j \mid \pi \sim \text{Categorical}(\pi) \tag{3.15}$$

$\rho_j = (\rho_{j\emptyset}, \cdots, \rho_{j\texttt{PPM/PMM}})$, where $\rho_{jh} = \mathbb{P}(H_j = h)$ is the probability of observing $h$ as the allelic configuration of segment $j$. $\pi = (\pi_1, \cdots, \pi_K)$, where $\pi_k = \mathbb{P}(Z_j = k)$ is the probability of observing subclonal type $k$ for all the segments. The model parameters $\Theta$ is defined as

$$\Theta = (\{\rho_j\}_{j=1}^{J}, \{\pi_k\}_{k=1}^{K}, \{\phi_k\}_{k=1}^{K}) \tag{3.16}$$

And the model likelihood of observing all the data is then

$$\begin{aligned}
& \mathbb{P}(\{D_j^T\}_{j=1}^{J}, \{b_{ij}^T\}_{i=1,j=1}^{I_j,J} | \Theta) \\
= & \prod_{j=1}^{J} \sum_{k=1}^{K} \sum_{h \in \mathcal{H}} \mathbb{P}(Z_j = k) \mathbb{P}(H_j = h) \mathbb{P}(D_j^T | H_j = h, Z_j = k) \\
\times & \prod_{i=1}^{I_j} \sum_{g \in \mathcal{G}} Q_{gh} \mathbb{P}(b_{ij}^T | G_{ij} = g, H_j = h, Z_j = k) \\
= & \prod_{j=1}^{J} \sum_{k=1}^{K} \sum_{h \in \mathcal{H}} \pi_k \rho_{jh} \frac{\lambda_j^{D_j^T} e^{-\lambda_j}}{D_j^T!} \\
\times & \prod_{i=1}^{I_j} \sum_{g \in \mathcal{G}} Q_{gh} \binom{d_{ij}^T}{b_{ij}^T} \bar{\mu}_{ij}^{b_{ij}^T} (1 - \bar{\mu}_{ij})^{d_{ij}^T - b_{ij}^T}
\end{aligned} \tag{3.17}$$

We use Expectation-Maximization (EM) algorithm [35] to find the maximum likelihood estimation of $\Theta$. The complete details of the EM updates are given in Supplementary [82].

### 3.2.6 Model selection

One of the key issues in subclonal analysis is to determine the number of subclonal populations $K$. PyClone and PhyloSub use posterior sampling methods to estimate $K$ [115, 61], while THetA requires users to specify $K$ as an input [95]. Since the probabilistic model of MixClone is a generative mixture model, the model complexity and the corresponding log-likelihood increases as $K$ increases. Therefore, we use a criterion based on the increase of the log-likelihood to select $K$. Practically, MixClone allows the user to specify $K$. If $K$ is not specified, MixClone runs the mixture model five times with different $K$ in range of 1 to 5. We denote the log-likelihoods under the five different settings as $\{L_K\}_{K=1}^{5}$, and the total log-likelihood increase as

$$\Delta = L_5 - L_1 \tag{3.18}$$

If $|\Delta/L_1| < 0.01$, which means the ratio of total log-likelihood increase is less than 0.01, MixClone predicts there is no subclonal event in the tumor sample and selects $K = 1$ as the number of subclonal populations. If $|\Delta/L_1| \geq 0.01$, MixClone further calculates another quantity

$$\delta_i = |L_i - L_1|/\Delta,\ i \in [2, 5] \tag{3.19}$$

which is the cumulative log-likelihood increase from $K = 1$ to $K = i$ as a percentage regarding to the total increase $\Delta$. If $\delta_i \geq 0.9$ and $\delta_{i-1} < 0.9$, MixClone selects $K = i$ as the number of subclonal populations.

In practice, we suggest users use this criterion as a heuristic guide when analyzing real data,

and determine the number of subclonal populations in conjunction with regard to other external information.

### 3.2.7 MixClone software package

Figure 3.4 is the general workflow of MixClone. MixClone is a comprehensive software package, including subclonal cellular prevalences estimation, allelic configuration estimation, absolute copy number estimation and a few visualization tools. This package is implemented in Python and is built on top of the PyLOH package, previously released by us [81]. It also utilizes some features from the software package JointSNVMix [114], which have been explicitly indicated in the source code.

## 3.3 Results

In this section, we evaluate the performance of MixClone on both simulated and real datasets and compare its performance with two published algorithms: (i) PyClone, a method based on somatic point mutations, and (ii) THetA, a method based on somatic copy number alterations.

### 3.3.1 Results from simulated data

To generate simulation data, we simulated ten sets of NGS reads from chromosome 1 of artificial paired normal-tumor samples, each with 60X coverage. Heterozygous SNP sites from dbSNP [120] were inserted to the reference human genome to create the artificial normal genome. Both heterozygous SNP sites and somatic point mutations from [16] were inserted to the reference human genome to create artificial tumor genomes. Five of the artificial

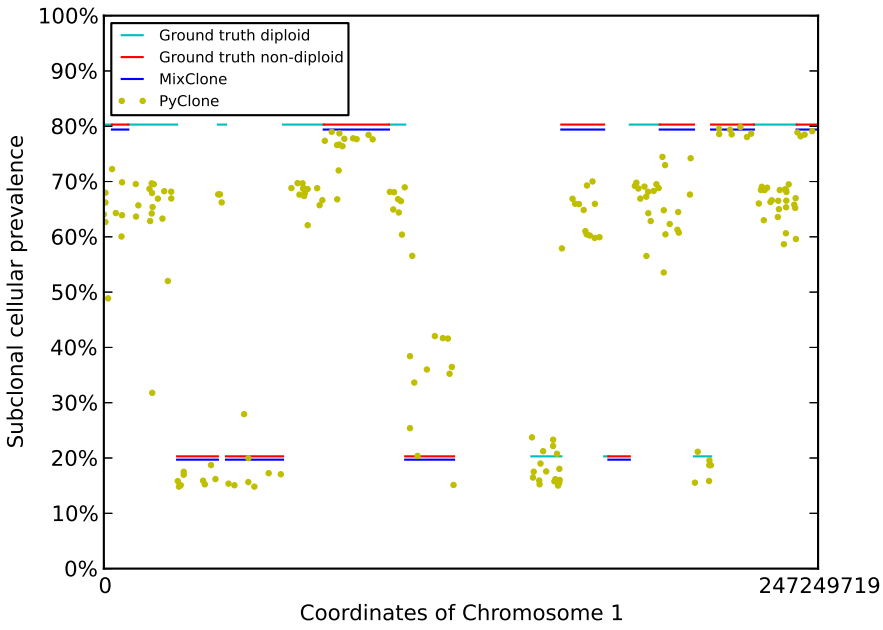



Figure 3.1: Subclonal inference results by MixClone and PyClone on a simulated dataset with two subclonal populations. The x-axis are the coordinates of Chromosome 1, and the y-axis are subclonal cellular prevalences. The blue horizontal bars represent the subclonal cellular prevalences estimated by MixClone based on non-diploid segments. Cyan and red horizontal bars represent the ground truth subclonal cellular prevalences of diploid and non-diploid segments. Yellow dots represent the subclonal cellular prevalences estimated by PyClone based on somatic point mutations.

tumor genomes contain two subclonal populations and the other five contain three subclonal populations. Each artificial tumor genome was randomly assigned with segmentations, allelic configurations and subclonal cellular prevalences. We used segmentations based on both ground truth and BIC-seq [139] as the input for MixClone. We used ground truth somatic point mutation sites and copy numbers as the input for PyClone and THetA. Details on how reads were simulated and preprocessed are given in Supplementary [82].

MixClone is able to identify the correct subclonal populations for all the simulated datasets based on ground truth segmentations. Figure 3.1 shows the result of simulated dataset with two subclonal populations. MixClone also correctly estimates the subclonal cellular prevalences of all the segments with SCNAs except for one small segment in tumor genome case 4 with three subclonal populations. For results based on BIC-seq segmentations, MixClone

still correctly estimates the subclonal cellular prevalences of the majority of the segments with SCNAs, except for those with copy-neutral loss of heterozygosity. This is likely due to the incorrect segmentations of BIC-seq, as BIC-seq relies on copy number changes and is unable to detect segments with copy-neutral loss of heterozygosity when they are adjacent to diploid segments. The complete results of all the simulated datasets based on both ground truth and BIC-seq segmentations are shown online through the github website associated with MixClone. As a comparison, we also run PyClone and THetA on the same datasets. We were unable to obtain THetA results after running it for more than 72 hours, likely due to its exponential scalability with the number of segments. In Figure 3.1, PyClone detects one of the two subclonal populations, whose ground truth cellular prevalence is 20%, but misestimates the other subclonal population, whose ground truth cellular prevalence is 80%, except for a few segments. The performance of MixClone on the other simulated datasets also significantly outperforms PyClone. One possible reason might be that the reads coverage of simulated datasets is not deep enough to support PyClone's non-parametric method [115], thus PyClone tends to report more subclonal populations due to the statistical variance.

### 3.3.2 Results from breast cancer sequencing data

We also applied MixClone on a whole-genome breast cancer sequencing dataset [11]. The details on data preprocessing are described in Supplementary [82].

Figure 3.2a shows the subclonal inference results of sample MB-116. One estimated subclonal cellular prevalence 32% is consistent with the tumor purities estimated by PyLOH and THetA [81], and another estimated cellular prevalence 66% is consistent with the tumor purity estimated by ABSOLUTE [25] reported in [11].

Figure 3.2b shows the five log-likelihoods of MB-116 under different numbers of subclonal populations. The magenta, red and yellow curves represent the log-likelihoods corresponding

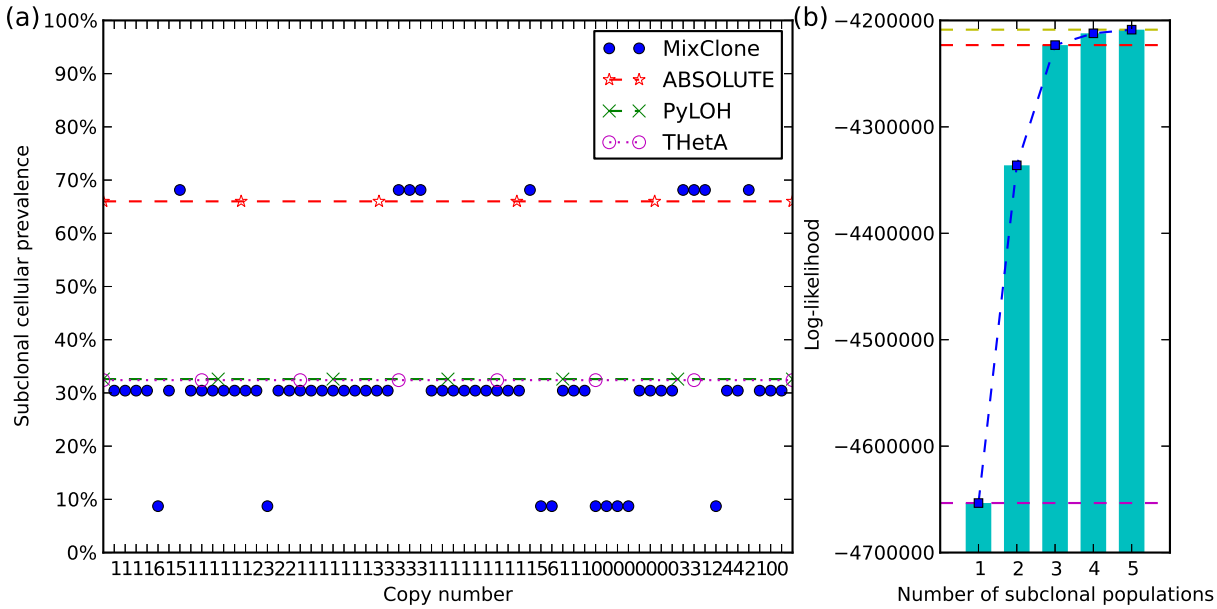



Figure 3.2: Subclonal inference results of sample MB-116. (a) The subclonal cellular prevalences estimated by MixClone, the tumor purities estimated by PyLOH, THetA [81], and the tumor purities estimated by ABSOLUTE [25] reported in [11] of sample MB-116. Each blue dot represents a segment. The x-axis is the estimated absolute copy number of the segment, and the y-axis is the estimated subclonal cellular prevalence of the segment. (b) The five log-likelihoods of MB-116 under different number of subclonal populations.

to number 1, 3, and 5, respectively. Because the distance between the magenta and red curves (the cumulative log-likelihood increase from 1 to 3) is greater than 0.9 of the distance between the magenta and yellow curves (the total log-likelihood increase from 1 to 5), MixClone selected $K = 3$ as the number of subclonal populations for MB-116.

For samples without significant subclonal events, MixClone selected one as the number of subclonal populations, e.g. MB-106 (Figure 3.3). In Figure 3.3b, the ratio of total log-likelihood increase from 1 to 5 is $1.4 \times 10^{-4}$, which is less than the threshold of 0.01. Therefore, MixClone selected $K = 1$ as the number of subclonal populations for MB-106. The estimated cellular prevalence of this single population is 83%, which is also consistent with the tumor purities estimated by PyLOH, ABSOLUTE and one result of THetA [81] (Figure 3.3a).

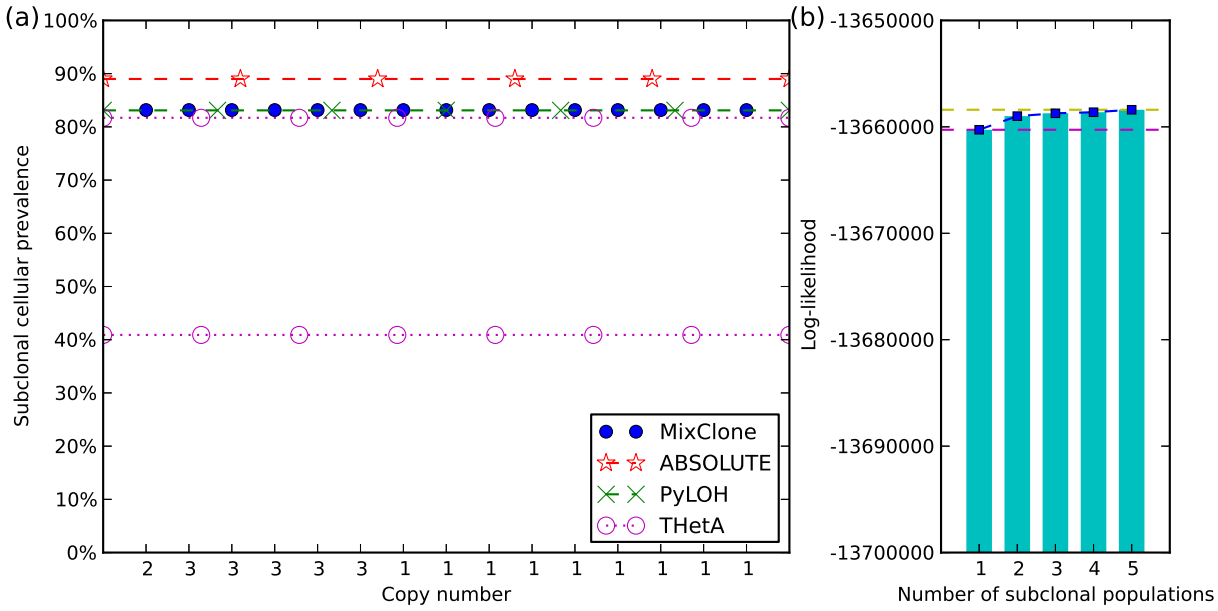



Figure 3.3: Subclonal inference results of sample MB-106. (a) The subclonal cellular prevalences estimated by MixClone, the tumor purities estimated by PyLOH, THetA [81], and the tumor purities estimated by ABSOLUTE [25] reported in [11] of sample MB-106. Each blue dot represents a segment. The x-axis is the estimated absolute copy number of the segment, and the y-axis is the estimated subclonal cellular prevalence of the segment. (b) The five log-likelihoods of MB-106 under different number of subclonal populations.

Besides MB-116, MixClone also detected significant subclonal events in MB-45 and MB-123. Results of MB-45 and MB-123 are given in Supplementary [82].

## 3.4 Discussion

In this article, we demonstrated MixClone's utility using whole genome sequencing data. However, most of the existing cancer genome sequencing data are from exome sequencing. An important future direction is to extend the current methodology to handle the exome sequencing data. Yet, extending MixClone to whole exome sequencing data is not trivial, as reads coverage on targeted exonic regions are no longer randomly distributed due to probe's variable efficiency [119]. Instead of Poisson distribution, using Gaussian distribution to model

reads depth ratios between tumor and normal samples might be more appropriate to account for such additional variances, which has been demonstrated in whole exome sequencing based copy number analysis [119].

Another important future direction to extend MixClone is to implement joint analysis based on multiple samples, which is supported by PyClone and PhyloSub [115, 61]. Multiple samples have been obtained for a single heterogeneous tumor tissue both temporally and spatially, and joint analysis based on these samples may reveal additional patterns of the history of tumor progression [115].

Currently, MixClone runs the subclonal analysis five times with different number of subclonal populations in range of 1 to 5 by default. In reality, larger numbers of subclonal populations may coexist within one tumor sample, but in this case some of the populations are very likely to share similar cellular prevalences. Since MixClone defines different subclonal populations based on distinct cellular prevalences, those populations with similar cellular prevalences may not be differentiated by MixClone. To achieve finer resolution of subclonal populations, subclonal lineages information would be necessary to further differentiate each population in addition to cellular prevalences. And phylogenetic methods may be possible solutions to explicitly incorporate subclonal lineages information [61].

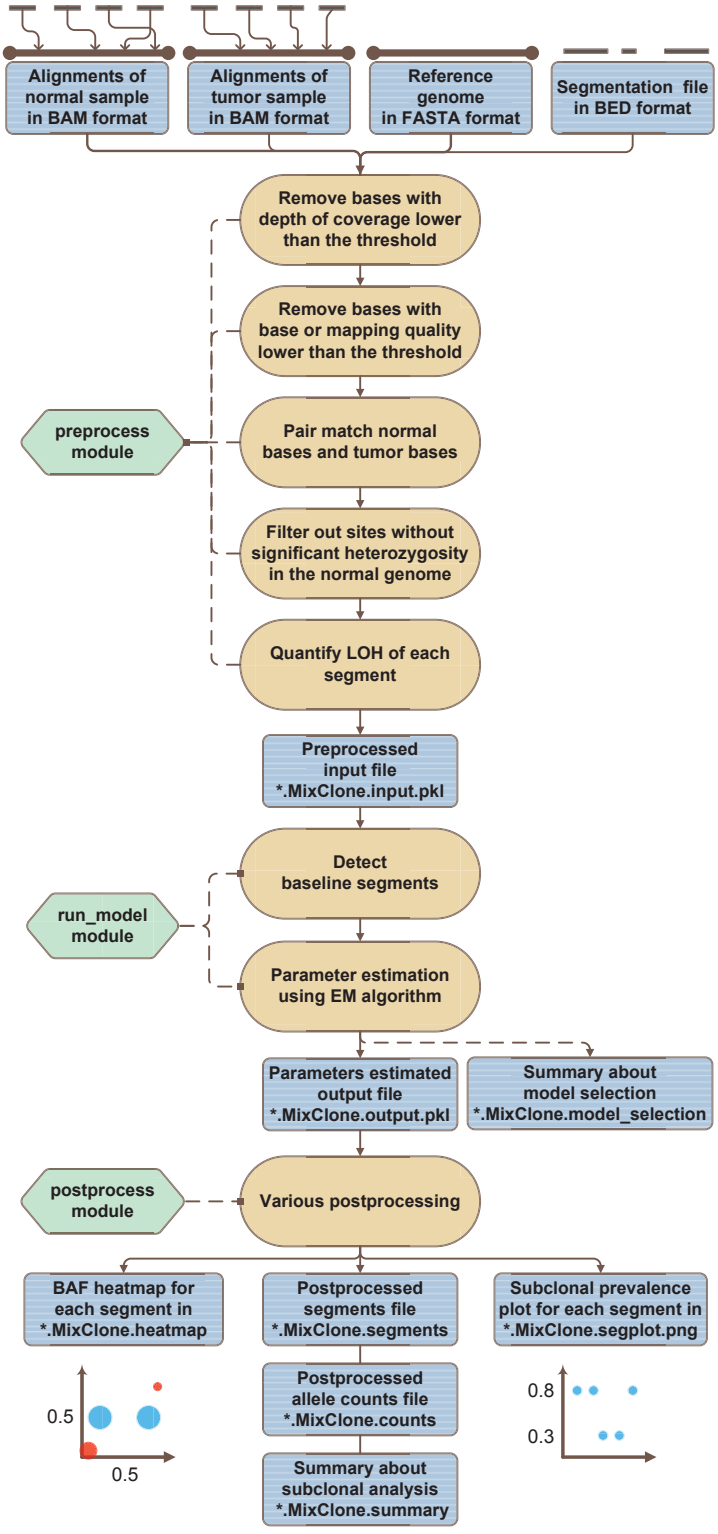



Figure 3.4: The general workflow of MixClone.

# Chapter 4

# Deconvolving tumor transcriptome expression

## 4.1 Introduction

The rapidly advancing next-generation sequencing based transcriptome analysis tool, RNA-seq, provides a comprehensive and accurate method for analyzing the entire RNA components of the transcriptome [85]. The efficiency and sensitivity of RNA-seq make it a primary method for detecting alternatively-spliced forms and estimating their abundances [133, 110]. However, estimating transcript abundances in heterogeneous tissues by RNA-seq remains an unsolved, outstanding problem because of the confounding effect from different cell types [26]. Many tissue samples from native environments are heterogeneous. For example, tumor samples are usually composed of tumor cells and surrounding normal cells [93]. Therefore, reads from an RNA-seq experiment of tumor samples will consist of contributions from both tumor and normal cells. Additionally, tumor tissues themselves are often heterogeneous, consisting of different subclones (e.g. breast cancer subtypes [91]), leading to even more

complicated tissue environments.

Experimental methods have been proposed to address issues arising from contamination of different cell types, such as laser-capture microdissection [39], which allows dissection of morphologically distinguishable cell types. The mRNA content yield by this technology is consequently lowered, and needs to be compensated for, usually by molecular amplification. However, the nonlinearity induced by amplifying mRNA [97] has its own problems, and can make the expression profiles of distinct cell types less distinguishable, weakening the sensitivity of RNA-seq technology. Other experimental approaches, including cell purification and enrichment, are comparatively expensive and laborious [31]. Therefore developing alternative *in silico* approaches to resolving the tissue heterogeneity problem, especially in cancer research, remains a major problem in RNA-seq analysis [90].

Research in computational approaches to resolving the tissue heterogeneity problem of different biotechnologies has a fairly long history [135, 49, 40, 146]. The first attempt to computationally micro-dissect heterogeneous tissues for microarray expression data was based on a linear model [135], which estimated both cell-type proportion and gene expression level. Prior information regarding "marker genes", which are genes uniquely expressed in each cell-type, was incorporated into the linear model to identify distinct cell types. The linear model was extended with Bayesian prior densities of cell-type proportions [40], and a posterior sampling approach was then constructed for cell-type-specific expression profiling. A statistical testing method [146] was proposed for single nucleotide polymorphism (SNP) array based copy number alterations analysis from heterogeneous tissue samples. In this method, Bayesian differentiation between hemizygous deletion and homozygous deletion were used to infer the underlying normal cell proportion and copy number profiles of both normal cells and tumor cells. One common feature shared by these methods is that they all adopted probabilistic models, not only allowing prior information about different cell types to be smoothly incorporated into the models, but also taking advantages of the flexibility of

probabilistic model to capture specific aspects of each data type.

To the best of our knowledge, no computational approaches have been proposed to resolve the tissue heterogeneity problem from RNA-seq data in a probabilistic fashion. Typically, researchers apply transcriptional profiling tools designed for homogeneous tissue samples directly to RNA-seq data from heterogeneous tissue samples. Subsequent estimation results are interpreted as transcriptional profiling of a particular single cell type of interest. Therefore, we ask whether it is possible to estimate trancriptive abundances of individual cell types from RNA-seq of heterogeneous tissues, by decoupling the contributions from multiple cell types. We propose a probabilistic model-based approach, Transcript Estimation from Mixed Tissue samples (TEMT) to address this question. Currently, TEMT requires two sets of single-end RNA-seq reads. One read set is from a heterogeneous tissue sample composed of two cell types, while the other is from a pure tissue sample composed of one of the two cell types. TEMT incorporates prior information of cell type proportion and can calculate probabilities of RNA-seq reads sampled from each cell type. Because TEMT implements an online EM algorithm [24], it has a time requirement proportional to the data size and a constant memory requirement. To further improve the estimation accuracy, TEMT also implements a bias module, which incorporates both positional bias [21, 78, 79] and sequence-specific bias [112, 51].

To assess the performance of TEMT, we analyzed a series of both simulation and real data from ENCODE [64], and compared the transcript relative abundances estimation from TEMT to those obtained from other methods that do not take the tissue heterogeneity into account. Our results show that explicitly accounting for tissue heterogeneity can significantly improve transcript abundance estimation accuracy.

## 4.2 Methods

In this section, we first introduce the generative mixture model of TEMT. Combined with cell type proportion as prior information, we propose a maximum a posteriori estimation approach for finding model parameters. Next, we explain how to incorporate a positional and sequence-specific bias module into the model. Finally, we introduce an online EM algorithm for parameter estimation, reducing the time complexity to be proportional to the data size and the space complexity to be constant.

### 4.2.1 Model

**Basic definition** We focus on transcript abundance estimation. Denote $\mathcal{T}$ as a set of reference transcripts, which we assume is known and complete. Let $l_t$ denote the length of transcript $t$ in the set with $t = 1, \cdots, T$, where $T$ is the total number of transcripts in the reference set. Suppose we are interested in transcriptome analysis in two cell types: $a$ and $b$. Let $\rho_t^a$ and $\rho_t^b$ denote the relative transcript abundance of transcript $t$ in cell type $a$ and $b$, respectively, with $t = 1, \cdots, T$. We assume $\{\rho_t^a\}_{t=1}^T$ and $\{\rho_t^b\}_{t=1}^T$ are properly normalized such that $\sum_{t=1}^T \rho_t^a = 1$ and $\sum_{t=1}^T \rho_t^b = 1$.

We assume RNA-seq reads are available in two samples: one consisting of cells of only type $a$, which we call the "pure sample", and the other consisting of cells of both type $a$ and $b$ with percentage $\tau^a$ from cell type $a$ and $\tau^b$ from cell type $b$, which we call the "mixed sample." In the cancer transcriptome analysis, cell type $a$ can represent normal cells as it is usually easy to obtain a pure tissue sample, while cell type $b$ can represent tumor cells as most tumor tissue samples are contaminated by normal cells.

Because the pure sample consists of only cell type $a$, its relative transcript abundance $\rho_t^p$ is described by $\rho_t^p = \rho_t^a$ for all $t$. However, the relative abundance of transcript $t$ within the

mixed sample is a weighted sum of the transcript abundance of both cell type $a$ and $b$

$$\rho_t^m = \tau^a \rho_t^a + \tau^b \rho_t^b, \tau_t^a + \tau_t^b = 1 \tag{4.1}$$

Denote the read set from the pure sample by $\mathcal{R}^p$ and the read set from the mixed sample by $\mathcal{R}^m$. Our goal is to estimate the relative abundance of each transcript in the reference set $\mathcal{T}$ from the RNA-seq read data $\mathcal{R}^p$ and $\mathcal{R}^m$ in both cell type $a$ and $b$.

**Alignment representation** We first map reads to the reference transcript set $\mathcal{T}$ and convert the raw read data into a corresponding alignment representation. Denote the alignment representation of the read set $\mathcal{R}^p$ by $\mathcal{Y}^p = \{y_{i,t}^p | i = 1, \cdots, N^p, t = 1, \cdots, T\}$, where $y_{i,t}^p = 1$ if read $i$ from $\mathcal{R}^p$ aligns to transcript $t$ and 0 otherwise, and $N^p$ is the total number of reads in read set $\mathcal{R}^p$. The alignment representation $\mathcal{Y}^m = \{y_{i,t}^m | i = 1, \cdots, N^m, t = 1, \cdots, T\}$ is similarly defined for read set $\mathcal{R}^m$ from the mixed sample. Note that one read might map to multiple transcripts due to alternative splicing, sequence similarity shared by homologous genes, or other reasons. As a result, the summation of $y_{i,t}^p$ over all transcripts may be bigger than 1 for some $i$. These "ambiguous reads" introduce a major source of uncertainty into transcript abundance estimation.

**Generative model** We model the sequencing of reads as a sampling process, randomly chooses a transcript $t$ from the reference transcript set $\mathcal{T}$ according to its relative abundance and effective length, and then generates a read from a random location of the chosen transcript. Under this model, the probability of a read originating from transcript $t$ is

$$\alpha_t^s = \frac{\rho_t^s \tilde{l}_t}{\sum_{k=1}^T \rho_k^s \tilde{l}_k} \tag{4.2}$$

with $s$ being either $p$ for the pure sample or $m$ for the mixed sample. Here, $\tilde{l}_t$ is the

effective length of transcript $t$, which quantifies the number of positions at which a read can start within transcript $t$. Different methods have been proposed to model the effective length [112, 98]. In TEMT, the effective length is modelled with consideration to the length distribution of RNA-seq fragments [112]

$$\tilde{l}_t = \sum_{x=1}^{l_t} \frac{\phi(x; \mu, \sigma^2)}{\sum_{x'=1}^{l_t} \phi(x'; \mu, \sigma^2)} (l_t - x + 1) \tag{4.3}$$

We assume the fragment length $x$ has a normal distribution with mean $\mu$ and variance $\sigma^2$, and $\phi(x; \mu, \sigma^2)$ is the normal probability density function of. By renormalizing $\phi(x; \mu, \sigma^2)$, we obtain the discrete distribution of all possible fragment lengths. The effective length $\tilde{l}_t$ is then the expectation of the number of positions a read can start within transcript $t$, based on the discrete distribution of fragment length.

Suppose a read is generated uniformly from each location covered by the effective length of each transcript. Then the probability of observing read $i$ as represented by its alignment map is

$$\mathbb{P}(\{y_{i,t}^s\}_{t=1}^T) = \sum_{t=1}^{T} y_{i,t}^s \frac{\alpha_t^s}{\tilde{l}_t} \tag{4.4}$$

for $s = p$ or $m$.

Assume each read is generated independently in both the pure and the mixed samples. The likelihood of observing the read set $\mathcal{R}^p$ from the pure sample and $\mathcal{R}^m$ from the mixed sample is then described by

$$\mathbb{P}(\mathcal{R}^p, \mathcal{R}^m | \{\alpha_t^p\}_{t=1}^T, \{\alpha_t^m\}_{t=1}^T) = \prod_{i=1}^{N^p} \sum_{t=1}^{T} y_{i,t}^p \frac{\alpha_t^p}{\tilde{l}_t} \prod_{i=1}^{N^m} \sum_{t=1}^{T} y_{i,t}^m \frac{\alpha_t^m}{\tilde{l}_t} \tag{4.5}$$

We are interested in estimating the relative transcript abundances set $\{\rho_t^a\}_{t=1}^T, \{\rho_t^b\}_{t=1}^T$, but

since it can be uniquely defined by the read sampling probability set $\{\alpha_t^a\}_{t=1}^T, \{\alpha_t^b\}_{t=1}^T$

$$\rho_t^a = \frac{\frac{\alpha_t^a}{\tilde{l}_t}}{\sum_{k=1}^T \frac{\alpha_k^a}{\tilde{l}_k}}, \rho_t^b = \frac{\frac{\alpha_t^b}{\tilde{l}_t}}{\sum_{k=1}^T \frac{\alpha_k^b}{\tilde{l}_k}} \tag{4.6}$$

We can directly estimate the read sampling probability set $\{\alpha_t^a\}_{t=1}^T, \{\alpha_t^b\}_{t=1}^T$ from the likelihood function Eq. (4.5) instead. Note that, again $\alpha_t^p = \alpha_t^a$ for all $t$ as it is the parameter of pure sample, but unlike the linear form in Eq. (4.1), $\alpha_t^m$ in terms of $\alpha_t^a, \alpha_t^b$ is given as a nonlinear form

$$\alpha_t^m = \Lambda^a \tau^a \alpha_t^a + \Lambda^b \tau^b \alpha_t^b \tag{4.7}$$

$$\Lambda^a = \frac{\sum_{k=1}^T \rho_k^a \tilde{l}_k}{\sum_{k=1}^T \rho_k^m \tilde{l}_k}, \Lambda^b = \frac{\sum_{k=1}^T \rho_k^b \tilde{l}_k}{\sum_{k=1}^T \rho_k^m \tilde{l}_k} \tag{4.8}$$

Where, the factor $\Lambda^a, \Lambda^b$ induce the nonlinearity. But due to the averaging effect of the large number of transcripts, practically $\Lambda^a, \Lambda^b$ lies within $1 \pm 0.05$. So we approximate $\alpha_t^m$ with the linear form

$$\alpha_t^m \approx \tau^a \alpha_t^a + \tau^b \alpha_t^b \tag{4.9}$$

As it brings computational convenience in the following learning step.

Finally, we define

$$\Theta = \{\{\alpha_t^a\}_{t=1}^T, \{\alpha_t^b\}_{t=1}^T, \tau^a, \tau^b\} \tag{4.10}$$

as the parameters of our model. The likelihood in Eq. (4.5) can be then expressed as

$$\mathbb{P}(\mathcal{R}^p, \mathcal{R}^m|\Theta) = \prod_{i=1}^{N^p}\sum_{t=1}^{T} y_{i,t}^p \frac{\alpha_t^a}{\tilde{l}_t} \prod_{i=1}^{N^m}\sum_{t=1}^{T} y_{i,t}^m \frac{(\tau^a\alpha_t^a + \tau^b\alpha_t^b)}{\tilde{l}_t} \tag{4.11}$$

### 4.2.2 Maximum a posteriori estimation

Several analysis have noticed the identifiability problem [49, 40] in estimating cell type specific expression in heterogeneous tissue samples. Ideally, if the proportion information for some cell types is missing, we can then pool these cell types as one type, making the expression of each individual cell type inside unidentifiable. Previously, prior constraints have been used to resolve the problem [49, 40]. In our model, the prior knowledge of cell type proportions is combined with the model likelihood, and we subsequently use maximum a posteriori (MAP) estimation to find the optimal parameters.

Specifically, we place a $Beta(\beta^a, \beta^b)$ distribution as the prior for cell proportions of type $a$ and type $b$. The parameter $\beta^a, \beta^b$ quantify the location and sharpness of the prior. Practically, we found setting $\beta^a, \beta^b$ 10 times as the data size gave a good convergence rate and accuracy. Combining the prior with the likelihood given in Eq. (4.11), the posterior distribution of the model is proportional to

$$\mathbb{P}(\Theta\ |\mathcal{R}^p, \mathcal{R}^m) \propto \left(\prod_{i=1}^{N^p}\sum_{t=1}^{T} y_{i,t}^p \frac{\alpha_t^a}{\tilde{l}_t}\right)\left[\prod_{i=1}^{N^m}\sum_{t=1}^{T} y_{i,t}^m \frac{(\tau^a\alpha_t^a + \tau^b\alpha_t^b)}{\tilde{l}_t}\right](\tau^a)^{\beta^a-1}(\tau^b)^{\beta^b-1} \tag{4.12}$$

### 4.2.3 Incorporating sequencing bias

Both positional [21, 78, 79] and sequence-specific [112, 51] sequencing biases have been observed in next generation sequencing data. These biases mainly result from non-uniformly distributed cDNA fragments during the RNA-seq library preparation [51]. Under positional

bias, reads positioning is not uniformly distributed across the effective length of the target transcript, but preferentially distributed around either the $5'$ end or the $3'$ end of the target transcript. Under sequence-specific bias, the sequences near the two ends of the fragments affect their probability to be sequenced. To account for these non-uniformity effects during transcript abundance estimation, we incorporate the bias module of [112] into our model.

In order to further describe the local alignment context, we define another two sets of variables. Specifically, for read $i$ from either read set $\mathcal{R}^p$ or $\mathcal{R}^m$, we denote $b^s_{i,t} \in [0, \tilde{l}_t]$ as the starting position of the alignment within transcript $t$ relative to the $5'$ end of the strand. We also denote $\pi^s_{i,t} \in \Sigma^L$, where $\Sigma = \{A, C, G, T\}$, as the local sequence of transcript $t$ with length $L$ and centered at $b^s_{i,t}$ .Then we define the bias weight $w^s_{i,t}$ as

$$w^s_{i,t} = \frac{\mathbb{P}(b^s_{i,t}|\text{bias})\mathbb{P}(\pi^s_{i,t}|\text{bias})}{\mathbb{P}(b^s_{i,t}|\text{uniform})\mathbb{P}(\pi^s_{i,t}|\text{uniform})} \tag{4.13}$$

for s=$p$ or $m$.

This bias weight $w^s_{i,t}$ is essentially the ratio of the probability of observing $b^s_{i,t}$ and $\pi^s_{i,t}$ under the bias model to the probability under the uniform model. If no bias exists, the weight $w^s_{i,t}$ reduces to 1. The bias re-weighted Eq. (4.4) is then:

$$\mathbb{P}(\{y^s_{i,t}\}^T_{t=1}) = \sum_{t=1}^{T} y^s_{i,t} \frac{\alpha^s_t}{\tilde{l}_t} w^s_{i,t} \tag{4.14}$$

To calculate the bias weight, we use the bin method and Markov chain for positional bias and sequence-specific bias respectively. Complete details can be found in the Supplementary [80]. The final unnormalized posterior distribution of the model is then described as

$$\mathbb{P}(\Theta\,|\mathcal{R}^p, \mathcal{R}^m) \propto \left(\prod_{i=1}^{N^p}\sum_{t=1}^{T} y^p_{i,t}\frac{\alpha^a_t}{\tilde{l}_t}w^p_{i,t}\right)\left[\prod_{i=1}^{N^m}\sum_{t=1}^{T} y^m_{i,t}\frac{(\tau^a\alpha^a_t + \tau^b\alpha^b_t)}{\tilde{l}_t}w^m_{i,t}\right](\tau^a)^{\beta^a-1}(\tau^b)^{\beta^b-1} \tag{4.15}$$

Where $w^p_{i,t}$ and $w^m_{i,t}$ are the bias weights computed based on read set $\mathcal{R}^p$ and $\mathcal{R}^m$. The

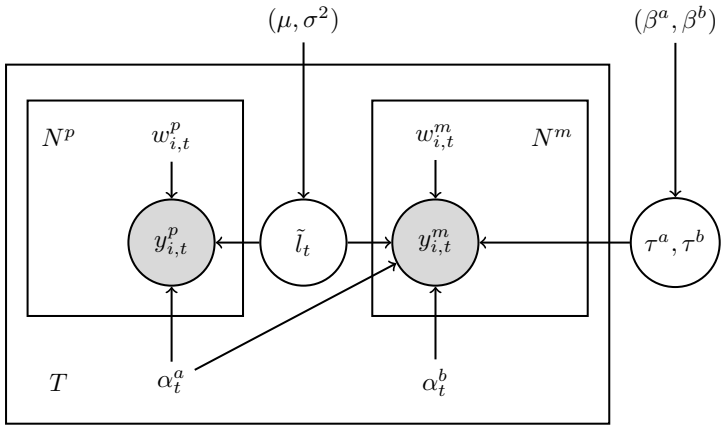



Figure 4.1: The representative graphical model of TEMT.

directed graphical model of TEMT is shown in Figure 4.1. The estimated parameters are given by

$$\hat{\Theta} = \underset{\theta}{arg\,max} \log \mathbb{P}(\Theta \mid \mathcal{R}^p, \mathcal{R}^m) \tag{4.16}$$

### 4.2.4 Online EM algorithm for learning

We solve the maximum a posteriori problem in Eq. (4.16) using the Expectation-Maximization (EM) [35] framework. For each read $i$ from read set $\mathcal{R}^p$ of pure sample, we denote the latent variable of the transcript alignment representation as $\mathcal{Z}^p_i = \{z^p_{i,t} | t = 1, \cdots, T\}$, where $z^p_{i,t} = 1$ if read $i$ aligns to transcript $t$ and 0 otherwise. But now $\sum_{t=1}^{T} z^p_{i,t} = 1$, which means only one $z^p_{i,t} = 1$, indicating read $i$ is actually originating from transcript $t$. Similarly, for each read $i$ from read set $\mathcal{R}^m$ of mixed sample, we denote the latent variable of the transcript alignment representation as $\mathcal{Z}^m_i = \{z^{ma}_{i,t}, z^{mb}_{i,t} | t = 1, \cdots, T\}$, where $z^{ma}_{i,t} = 1$ if read $i$ aligns to transcript $t$ and is originating from cell type $a$ within the mixed sample, and 0 otherwise. $z^{mb}_{i,t} = 1$ or 0 is similar defined for cell type $b$. Thus $\sum_{t=1}^{T}(z^{ma}_{i,t} + z^{mb}_{i,t}) = 1$ means read $i$ is actually originat-

ing from only one transcript, and either from cell type $a$ or $b$ within the mixed sample. We also define the auxiliary variable $q^p_{i,t} = \mathbb{P}(z^p_{i,t} = 1|\Theta, \mathcal{Y}^p, \mathcal{Y}^m), q^{ma}_{i,t} = \mathbb{P}(z^{ma}_{i,t} = 1|\Theta, \mathcal{Y}^p, \mathcal{Y}^m)$ and $q^{mb}_{i,t} = \mathbb{P}(z^{mb}_{i,t} = 1|\Theta, \mathcal{Y}^p, \mathcal{Y}^m)$ as the conditional probability weight of each latent variable $z^p_{i,t} = 1, z^{ma}_{i,t} = 1$ and $z^{mb}_{i,t} = 1$ conditional on model parameters $\Theta$ and the observed read alignment representations $\mathcal{Y}^p, \mathcal{Y}^m$. Then based on Jensen's inequality [60], the complete posterior distribution, which is also the lower bound of Eq. (4.15) can be written as

$$\begin{aligned} & \mathbb{P}(\Theta \mid \mathcal{R}^p, \mathcal{R}^m) \\ \geq \; & \frac{1}{\mathcal{C}} \left[ \prod_{i=1}^{N^p} \prod_{t=1}^{T} (\frac{\alpha^a_t}{\tilde{l}_t} w^p_{i,t})^{q^p_{i,t}} \right] \left[ \prod_{i=1}^{N^m} \prod_{t=1}^{T} (\frac{\tau^a \alpha^a_t}{\tilde{l}_t} w^m_{i,t})^{q^{ma}_{i,t}} (\frac{\tau^b \alpha^b_t}{\tilde{l}_t} w^m_{i,t})^{q^{mb}_{i,t}} \right] (\tau^a)^{\beta^a - 1} (\tau^b)^{\beta^b - 1} \end{aligned} \tag{4.17}$$

In which $\mathcal{C}$ is a normalizing constant and the equality holds only if the conditional probabilities $q^p_{i,t}, q^{ma}_{i,t}, q^{mb}_{i,t}$ are the true posterior distributions of latent variables $\{\mathcal{Z}^p_i\}_{i=1}^{N^p}, \{\mathcal{Z}^m_i\}_{i=1}^{N^m}$.

The EM framework maximizes Eq. (4.17) by iteratively applying the expectation step and the maximization step to update both the conditional probabilities $q^p_{i,t}, q^{ma}_{i,t}, q^{mb}_{i,t}$ and model parameters $\Theta$ until convergence. The expectation step of typical batch EM algorithm has to fetch all the data points into memory, and calculates the conditional probabilities based on the average of all the data points. While this batch method guarantee's the log-likelihood function to monotonically increase, it also induces inefficiency in both time and space complexity. Considering the high-throughput nature of next-generation sequencing technology as well as its huge data size, we implemented the EM algorithm in an online fashion [24] to both lower the memory requirement and boost the convergence rate.

The main difference between the batch EM and the online EM is in the E-step. The E-step of the online EM algorithm first calculates the conditional probabilities of only one new data point, and then updates the conditional probabilities of all the current data points by interpolating between the conditional probabilities of all the previous data points and the

conditional probabilities of the new data point, with a forgetting factor $\sigma$ controlling the convergence rate.

It is shown in [24] that with the constraint $0.5 < \sigma \leq 1$, the online EM algorithm is asymptotically equivalent to stochastic gradient ascent, and is guaranteed to converge to the maximum likelihood estimator, which is extended to the maximum a posteriori estimator in our model.

Specifically, the online EM updates in our model is given by

**E-step**

$$q^p_{i+1,t} = \frac{y^p_{i+1,t}\frac{\alpha_t^{a(n)}}{\tilde{l}_t}w^p_{i,t}}{\sum_{k=1}^T y^p_{i+1,k}\frac{\alpha_k^{a(n)}}{\tilde{l}_k}w^p_{i,k}} \tag{4.18}$$

$$q^{ma}_{i+1,t} = \frac{y^m_{i+1,t}\frac{\tau^{a(n)}\alpha_t^{a(n)}}{\tilde{l}_t}w^m_{i,t}}{\sum_{k=1}^T y^m_{i+1,k}\frac{\tau^{a(n)}\alpha_k^{a(n)}+\tau^{b(n)}\alpha_k^{b(n)}}{\tilde{l}_k}w^m_{i,k}} \tag{4.19}$$

$$q^{mb}_{i+1,t} = \frac{y^m_{i+1,t}\frac{\tau^{b(n)}\alpha_t^{b(n)}}{\tilde{l}_t}w^m_{i,t}}{\sum_{k=1}^T y^m_{i+1,k}\frac{\tau^{a(n)}\alpha_k^{a(n)}+\tau^{b(n)}\alpha_k^{b(n)}}{\tilde{l}_k}w^m_{i,k}} \tag{4.20}$$

$$q^{p\ (n+1)}_{*,t} = \left[1-\frac{1}{(n+2)^\sigma}\right]q^{p\ (n)}_{*,t} + \frac{1}{(n+2)^\sigma}q^p_{i+1,t} \tag{4.21}$$

$$q^{ma(n+1)}_{*,t} = \left[1-\frac{1}{(n+2)^\sigma}\right]q^{ma(n)}_{*,t} + \frac{1}{(n+2)^\sigma}q^{ma}_{i+1,t} \tag{4.22}$$

$$q^{mb(n+1)}_{*,t} = \left[1-\frac{1}{(n+2)^\sigma}\right]q^{mb(n)}_{*,t} + \frac{1}{(n+2)^\sigma}q^{mb}_{i+1,t} \tag{4.23}$$

We compute the conditional probabilities $q^p_{i+1,t}, q^{ma}_{i+1,t}, q^{mb}_{i+1,t}$ of just one new read $i+1$ based on previous parameter estimation $\{\alpha_t^{a(n)}\}_{t=1}^T, \{\alpha_t^{b(n)}\}_{t=1}^T, \tau^{a(n)}, \tau^{b(n)}$; Then, we compute the new conditional probabilities average $q^{p\ (n+1)}_{*,t}, q^{ma(n+1)}_{*,t}, q^{mb(n+1)}_{*,t}$ by interpolating between the previous conditional probabilities average $q^{p\ (n)}_{*,t}, q^{ma(n)}_{*,t}, q^{mb(n)}_{*,t}$ and $q^p_{i+1,t}, q^{ma}_{i+1,t}, q^{mb}_{i+1,t}$. $n$ is the index of iteration step and $i$ is the index of data points. $\sigma$ is the forgetting factor which

controls the convergence rate, with the constraint $0.5 < \sigma \leq 1$.

**M-step**

$$\tau^{a(n+1)} = \frac{\sum_{t=1}^{T} q_{*,t}^{ma(n+1)} + \frac{\beta^a - 1}{N^m}}{1 + \frac{\beta^a + \beta^b - 2}{N^m}} \tag{4.24}$$

$$\tau^{b(n+1)} = \frac{\sum_{t=1}^{T} q_{*,t}^{mb(n+1)} + \frac{\beta^b - 1}{N^m}}{1 + \frac{\beta^a + \beta^b - 2}{N^m}} \tag{4.25}$$

$$\alpha_t^{a(n+1)} = \frac{q_{*,t}^{p\ (n+1)} + q_{*,t}^{ma(n+1)}}{1 + \tau^{a(n+1)}} \tag{4.26}$$

$$\alpha_t^{b(n+1)} = \frac{q_{*,t}^{mb(n+1)}}{\tau^{b(n+1)}} \tag{4.27}$$

In the subsequent M-step, parameters $\{\alpha_t^{a(n+1)}\}_{t=1}^{T}, \{\alpha_t^{b(n+1)}\}_{t=1}^{T}, \tau^{a(n+1)}, \tau^{b(n+1)}$ are updated according to new conditional probabilities average $q_{*,t}^{p\ (n+1)}, q_{*,t}^{ma(n+1)}, q_{*,t}^{mb(n+1)}$.

## 4.3 Results

Next we test the performance of the proposed method on both simulation data and the recently released ENCODE data [64]. For both datasets, we used the following three-step protocol and parameters to construct the analysis:

**1.** We aligned the raw read set from either simulation or the ENCODE data to a given transcript set using bowtie-0.12.7 [73]. For each read, we allowed 2 mismatches and reported at most 10 candidate alignments.

**2.** The abundance of each transcript in terms of estimated counts was estimated via both TEMT and a control model. Estimated counts is defined as the estimated number of reads generated from the target transcript. In TEMT, the prior of each cell type proportion was

set to the same as the proportion used in simulation and ENCODE data respectively, and $\beta^a, \beta^b$ was set to 10 times the size of the read set $\mathcal{R}^m$. $\mu = 200, \sigma = 80$ were used as the mean and standard deviation of the RNA-seq fragment length distribution. We chose eXpress-0.9.4 [111] as the control model, as it is the state-of-the-art method for transcript abundance estimation and also utilizes an online-EM algorithm. Note that, to run TEMT, we need two read sets, in which one is for the pure sample and the other is for the mixed sample, as previously mentioned. In contrast, to run eXpress, we only need one read set from either the pure sample or the mixed sample. The forgetting factor for the on-line EM algorithms in both TEMT and eXpress was set to be $\sigma = 0.85$, and the error-model in eXpress was disabled for comparison.

**3.** To measure the model accuracy, we used the Error Fraction (EF) measure introduced by [78] to quantify the discrepancy between the model estimates and the ground truth estimates. The Error Fraction is defined as the fraction of transcripts for which the estimates are significantly different (percent error $> 10\%$ in our case) from the ground truth.

### 4.3.1 Simulation

**Data preparation** To show the utility of TEMT, we first carried out a series of simulation studies. To obtain simulated read sets, we used FluxSimulator [118], a software for transcriptome and read generation by simulating the biochemical processes underlying the library preparation. FluxSimulator requires a reference transcript set to start the simulation process, so we manually downloaded 406 transcripts of 208 alternatively spliced genes in human from Alternative Splicing Structural Genomics Project (AS3D) [5], and used these 406 transcripts as the reference transcript set. We first simulated the transcript expression process twice producing two sets of relative transcript abundances, corresponding to cell type a and b respectively. Based on these two transcript abundance sets, we then simulated

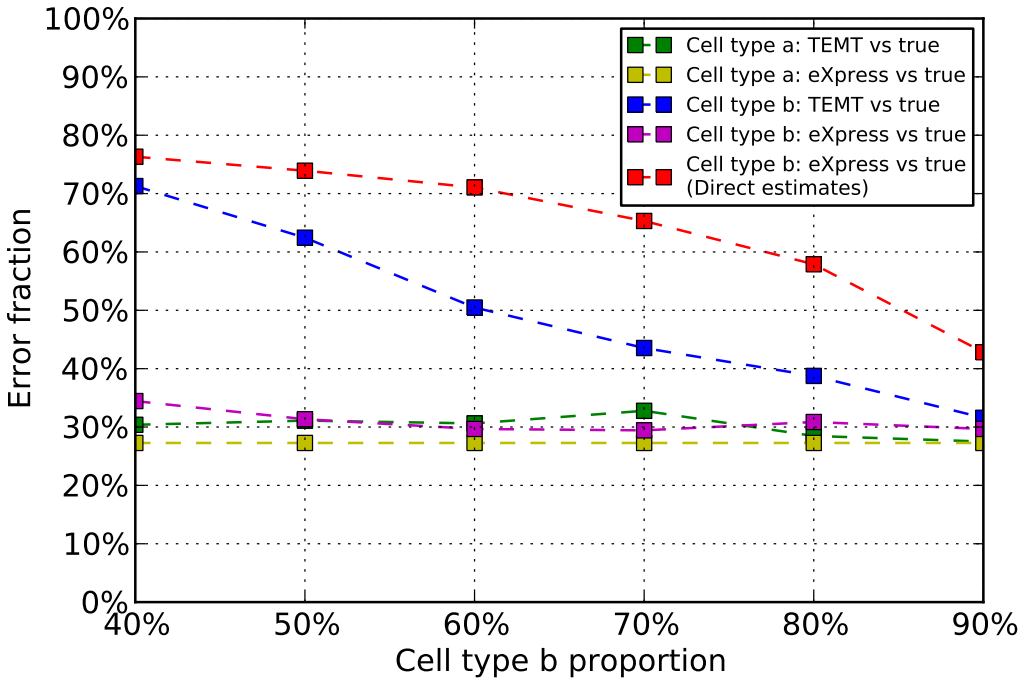



Figure 4.2: Analysis results of simulated data of 6 different cell type $b$ proportions with the bias module disabled. The x-axis is the cell type b proportions, and the y-axis is the Error Fraction of the corresponding estimates. The green and blue lines are the estimates from TEMT for cell type $a$ and cell type $b$, based on the two read sets of the cell type $a$ pure sample and the mixed sample. The yellow and magenta lines are the estimates from eXpress for cell type $a$ and cell type $b$, based on the two read sets of the cell type $a$ pure sample and the cell type $b$ pure sample. The red line is the direct estimates from eXpress for cell type $b$, based on the read set of the mixed sample.

6 pairs of 1 million 75-bp single-end read sets corresponding to six different cell type $b$ proportions from 40% up to 90%. The relative transcript abundances of cell type $a$ and $b$ were kept the same throughout these simulations. For each paired read set, one read set is for the pure sample composed of only cell type $a$, whereas the other read set is for the mixed sample composed of both cell type $a$ and $b$, mixed with the cell type $b$ proportion. Within the mixed-sample read set, we also extracted the reads simulated purely from cell type $b$, which was used for control model eXpress.

**Analysis** The simulated data are analyzed with the bias module both enabled and disabled. Surprisingly, the positional and sequence-specific bias module did not improve the accuracy

of the transcript abundance estimation as measured by the Error Fraction of estimated counts in both TEMT and eXpress. This result may due to the stochasticity during the simulation of FluxSimulator. So we only present the results with the bias module disabled in both TEMP and eXpress in Figure 4.2.

We note that the estimates of cell type $a$ from TEMT achieve roughly the same accuracy, compared with the estimates from eXpress based on the read set of the pure sample of cell type $a$. Also, this accuracy does not change significantly under the effect of different cell type b proportions. This is mainly due to the pure sample read set of cell type $a$ within the input data for TEMT.

The accuracy of the estimates of cell type $b$ from TEMT is also shown in Figure 4.2, which shows that TEMT generally outperforms the direct estimation method. To the best of our knowledge, there are no computational tools similar to our model that can estimate the relative transcript abundances of cell type $b$ via RNA-seq data generated from mixed samples. Typically, computational methods are applied directly to the noisy data of mixed samples and results are interpreted as the estimates of cell type $b$. To compare the estimates of cell type $b$ from TEMT with direct estimates using the current method, we applied the control model eXpress directly to the read set of the mixed sample. The estimated counts from eXpress were then compared with the true counts from another 1 million simulated read set purely of cell type $b$, while keeping the same relative transcript abundance as the previous simulations. The corresponding Error Fractions are shown as the red line in Figure 4.2 regarding different cell type $b$ proportions. Although the accuracy of cell type $b$ estimates from TEMT is affected by different cell type $b$ proportions, it is generally better than the direct estimates. This can be further illustrated in Figure 4.3, which shows that the direct estimated counts of cell type $b$ from eXpress deviate more from the true counts as the cell type $b$ proportion decrease, while the estimates of TEMT have much reduced deviation. We notice that as the cell type $b$ proportion gradually decreases, the accuracy of the estimates

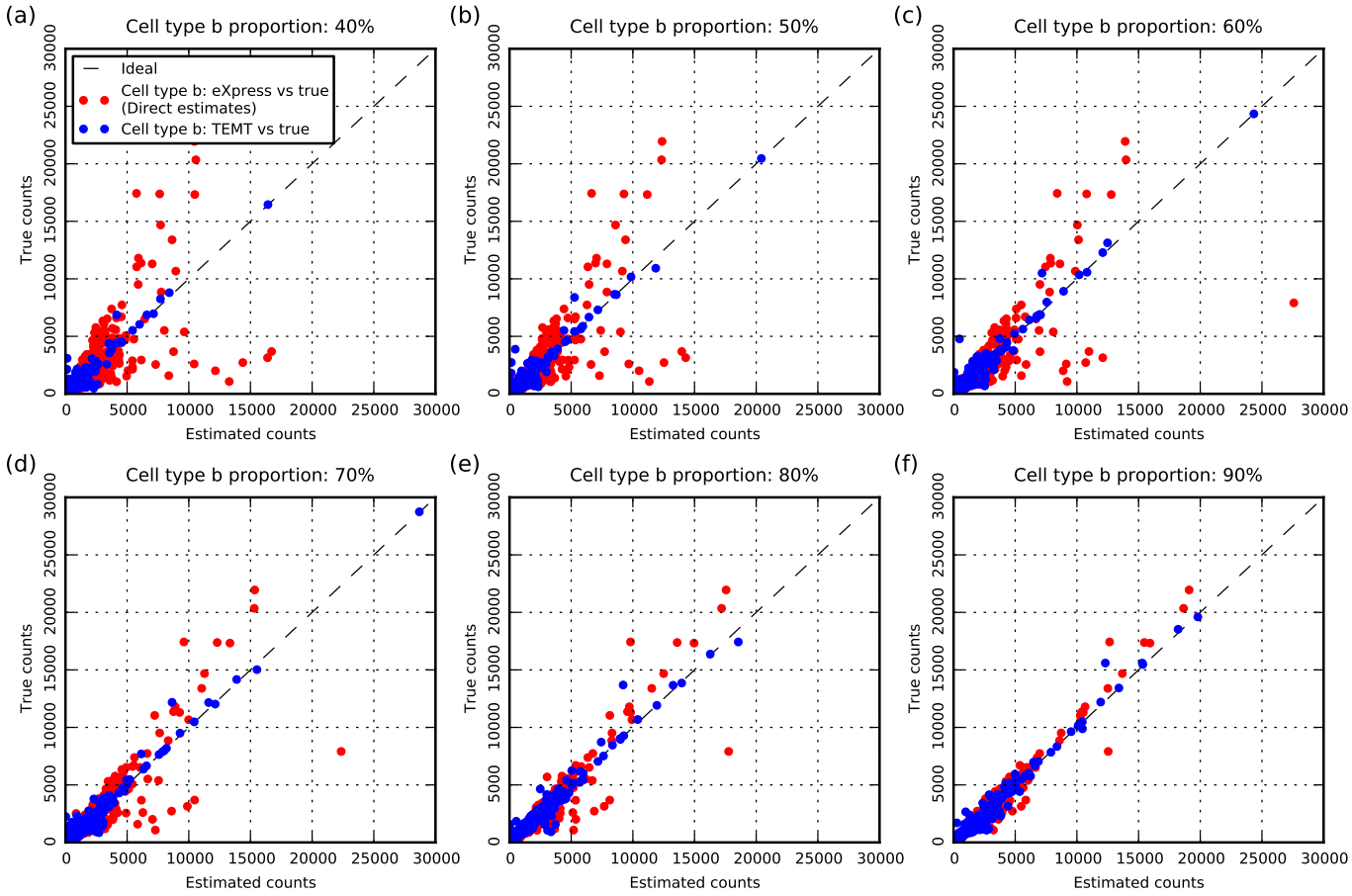



Figure 4.3: Comparisons between indirect estimates from TEMT and direct estimates from eXpress for cell type $b$ in terms of estimated counts. The x-axis is the estimated counts from the two models, and the y-axis is the true counts. Each point in the figure is a comparison between the estimated count and true count. The red points are the direct estimates from eXpress, while the blue points are the indirect estimates from TEMT. Figure (**a**)-(**f**) are each comparison with cell type $b$ proportions from 40% to 90%.

of cell type $b$ from TEMT also decreases. This is the result of the contamination effect from the cell type $a$ within the mixed sample. A recent paper [26] also observed this similar phenomenon when studying copy number aberrations from heterogeneous tumor tissue.

### 4.3.2 ENCODE data

**Data preparation** Next we analyzed the recently released ENCODE data. Due to the lack of RNA-seq data sampled from mixed tissue samples with known cell type proportions, we artificially generated the mixed-sample read sets by mixing reads obtained from two different cell types. Specifically, we chose two Tier 1 cell lines, GM12878 and K562, and treated them as cell type $a$ and cell type $b$ respectively. The corresponding single-end RNA-seq data of these two cell lines, GM78 1x75D A 1 (UCSC Accession: wgEncodeEH000125) and K562 1x75D A 1 (UCSC Accession: wgEncodeEH000126) from the Wold lab [92] at Caltech, were download from ENCODE (2012). The data downloaded from the same lab under similar protocols is intended to reduce the deviation resulting from experiments. We then randomly selected 10 million reads from GM12878 cells to form the read set of the pure sample, and 10 million reads from both GM12878 and K562 cells using different K562 cells proportions to form the read set of the mixed sample. Similar to the previous simulation study, we extracted the reads purely selected from K562 cells within the mixed sample, and used them for the eXpress control model. We studied 6 different K562 cells proportions from 40% to 90% in order to compare with the previous simulation study. 36908 human RefSeq [104] transcripts from UCSC known genes [58] were used as the transcript set for the ENCODE data.

**Analysis** One major issue in studying the ENCODE data is that the ground truth of relative transcript abundance in each cell type is unknown. We used the estimates from eXpress based on the GM12878 and K562 pure samples as the ground truth. Again, the bias module was disabled for both TEMT and eXpress. The general result of ENCODE data is shown in Figure 4.4. Similar to the simulated data, the indirect estimates for K562 cells from TEMT generally outperforms the direct estimates from eXpress based on the read set of the mixed sample. The contamination effect from cell type $a$ within the mixed sample observed in Figure 4.3 is also seen in the eXpress analysis of ENCODE data, while TEMT does not have

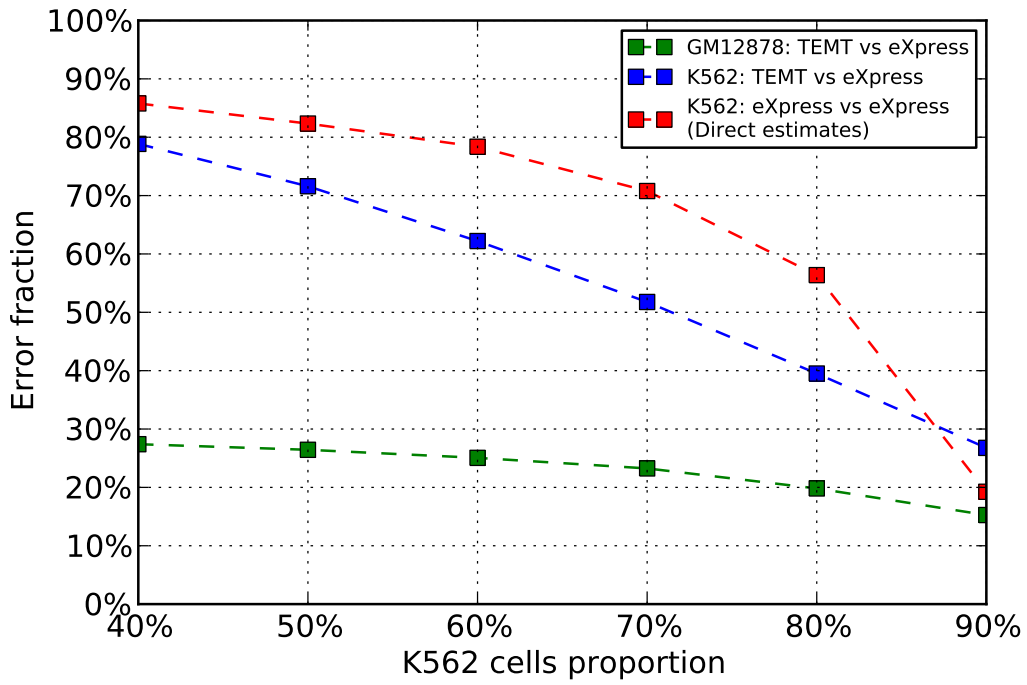



Figure 4.4: Analysis results of the ENCODE data of 6 different K562 cells proportions with the bias module disabled. The x-axis is the different K562 cells proportions, and the y-axis is the Error Fraction of the corresponding estimates. The green and blue lines are the estimates from TEMT for GM12878 and K562 cells, based on the read sets of the GM12878 cells pure sample and the mixed sample. The red line is the direct estimates from eXpress for K562 cells, based on the read set of the mixed sample.

this issue. Note that the measure of relative transcript abundances as shown in the red line of Figure 4.4 is no longer estimated counts, but reads per kilobase of transcript per million mapped reads (RPKM), as the total number of reads from K562 cells within the mixed sample is less than the total number of reads of the mixed sample, so that normalization is necessary for comparison. We notice TEMT underperforms direct estimates from eXpress when K562 cells proportion equals 90%. Possibly the contamination effect of GM12878 cells within the mixed sample is not severe enough at this point, as we can imagine the red line in Figure 4.4 will finally reach 0% Error Fraction when K562 cells proportion reaches 100%. On the other hand, since the estimates from eXpress based on the pure sample are considered the ground truth, the lower bound Error Fraction of K562 cells estimates from TEMT should be the same as the Error Fraction of GM12878 cells estimates, which is around 20% to 30%

in Figure 4.4.

## 4.4 Discussion

We formulated our model under the assumption that the heterogeneous tissue is only composed of two cell types, but in reality, a heterogeneous tissue might be much more complicated, consisting of multiple cell types. To relax this constraint, our model needs to be further extended to analyze more complex cases in which each cell type may have its own subtypes, e.g. breast cancer subtypes, leading to a more sophisticated heterogeneous tissue environment. Further dissecting cell subtype heterogeneity is the next step in refining our model. Moving from two cell types to arbitrarily many cell types is of great interest, since it may substantially facilitate transcriptome study of heterogeneous tissues.

One critical component necessary to make our model work is the prior information of cell type $b$ proportion, which is necessary to resolve the identifiability problem of mixed samples. In real experiments, precise prior information regarding cell type proportions may be unavailable. One solution in the context of our model is to down weight the effect of the prior by decreasing the parameter $\beta^a, \beta^b$, which adds more uncertainty to the cell mixture proportion. However, this approach may decrease the performance of the model as the uncertainty in cell mixture proportion can not be distinguished from the uncertainty in transcript abundance estimation. This observation suggests another direction to further improving our model which is to solely estimate cell type $b$ proportion without the prior information. To fulfill this requirement, the identifiability problem needs to be resolved as mentioned in section 2.3, which turns out to be comparatively hard for RNA-seq data. Unlike the heterozygous and homozygous deletions in [146], which can be utilized to differentiate between the SNP array data generated by normal cells and tumor cells, there are no such explicit differences between the reads generated by distinct cell types in RNA-seq data, thus making the gener-

ative mixture model unconstrained. The "marker genes" method proposed by [135], which tries to distinguish distinct cell types by utilizing genes uniquely expressed in each cell type, provides a future potential direction to extend the current model.

# Chapter 5

# Gene expression inference with deep learning

## 5.1 Introduction

A fundamental problem in molecular biology is to characterize the gene expression patterns of cells under various biological states. Gene expression profiling has been historically adopted as the tool to capture the gene expression patterns in cellular responses to diseases, genetic perturbations and drug treatments. The Connectivity Map (CMap) project was launched to create a large reference collection of such patterns and has discovered small molecules that are functionally connected using expression pattern-matching (e.g., HDAC inhibitors and estrogen receptor modulators) [70].

Although recent technological advances, whole-genome gene expression profiling is still too expensive to be used by typical academic labs to generate a compendium of gene expression over a large number of conditions, such as large chemical libraries, genome-wide RNAi screening and genetic perturbations. The initial phase of the CMap project produced only

564 genome-wide gene expression profiles using Affymetrix GeneChip microarrays [70].

Despite the large number of genes (˜22,000) across the whole human genome, most of their expression profiles are known to be highly correlated. Systems biologists have leveraged this idea to construct gene regulatory networks and to identify regulator and target genes [12]. Researchers from the LINCS program (http://www.lincsproject.org/) analyzed the gene expression profiles from the CMap data using principal component analysis. They found that a set of ˜1,000 carefully chosen genes can capture approximately 80% of the information in the CMap data (http://support.lincscloud.org/hc/en-us/articles/202092616-The-Landmark-Genes). Motivated by this observation, researchers have developed the L1000 Luminex bead technology to measure the expression profiles of these ˜1,000 genes, called the *landmark genes* (http://support.lincscloud.org/hc/en-us/articles/202092616-The-Landmark-Genes), with a much lower cost (˜$5 per profile) [100]. Therefore, researchers can use the expression signatures of landmark genes to characterize the cellular states of samples under various experimental conditions. If researchers are interested in the expression of a specific gene other than landmark genes, the expression profiles of the remaining ˜21,000 genes, called the *target genes*, can be then computationally inferred based on landmark genes and existing expression profiles. With the L1000 technology, the LINCS program has generated ˜1.3 million gene expression profiles under a variety of experimental conditions.

However, computationally inferring the expression profiles of target genes based on landmark genes is challenging. It is essentially a large scale multi-task machine learning problem, with the target dimension (˜21,000) significantly greater than the feature dimension (˜1,000). The LINCS program currently adopts linear regression as the inference method, which trains regression models independently for each target gene based on the Gene Expression Omnibus (GEO) [38] data. While linear regression is highly scalable, it inevitably ignores the nonlinearity within gene expression profiles that has been observed [52]. Kernel machines can represent dexterous nonlinear patterns and have been applied to similar problems [145]. Un-

fortunately, they suffer from poor scalability to growing data size. Thus, a machine learning method enjoying both scalability and rich representability is ideal for large scale multi-task gene expression inference.

Recent successes in deep learning on many machine learning tasks have demonstrated its power in learning hierarchical nonlinear patterns on large scale datasets [14]. Deep learning in general refers to methods that learn a hierarchical representation of the data through multiple layers of abstraction (e.g. multi-layer feedforward neural networks). A number of new techniques have been developed recently in deep learning, including the deployment of General-Purpose Computing on Graphics Processing Units (GPGPU) [30, 32], new training methodologies, such as dropout training [56, 10] and momentum method [130]. With these advances, deep learning has achieved state-of-the-art performances on a wide range of applications, both in traditional machine learning tasks such as computer vision [69], natural language processing [125], speech recognition [55], and in natural science applications such as exotic particles detection [9], protein structure prediction [36], RNA splicing prediction [77] and pathogenic variants identification [106].

Here we present a deep learning method for gene expression inference (D-GEX). D-GEX is a multi-task multi-layer feedforward neural network. We evaluated the performances of D-GEX, linear regression (with and without different regularizations) and k-nearest neighbor (KNN) regression on two types of expression data, the microarray expression data from the GEO and the RNA-Seq expression data from the Genotype-Tissue Expression (GTEx) project [84, 4]. GPU computing was used to accelerate neural network training so that we were able to evaluate a series of neural networks with different architectures. Results on the GEO data show that D-GEX consistently outperforms other methods in terms of prediction accuracy. Results on the GTEx data further demonstrate D-GEX, combined with the dropout regularization technique, achieves the best performance even where training and prediction were performed on datasets obtained from different platforms (microarray verse

RNA-Seq). Such cross platforms generalizability implies the great potential of D-GEX to be applied to the LINCS program where training and prediction were also done separately on the microarray data and the L1000 data. Finally, we attempted to explore the internal structures of the learned neural networks with two different strategies and tried to interpret the advantages of deep learning compared to linear regression.

## 5.2 Methods

In this section, we first introduce three expression datasets we used in this study and formulate gene expression inference as a supervised learning problem. We then present D-GEX for this problem and explain a few key deep learning techniques to train D-GEX. Finally, we introduce several common machine learning methods that we used to compare with D-GEX.

### 5.2.1 Datasets

1. *The GEO expression data* was curated by the Broad Institute from the publicly available GEO database. It consists of 129,158 gene expression profiles from the Affymetrix microarray platform. Each profile comprises of 22,268 probes, corresponding to the 978 landmark genes and the 21,290 target genes. The original GEO data was accessed from the LINCS Cloud (http://www.lincscloud.org/), which has been quantile normalized into a numerical range between 4 and 15. Some of the expression profiles in the GEO dataset are biological or technical replicates. To avoid complications in the learning procedure, we removed duplicated samples (see Supplementary [28]), leaving 111,009 profiles in the end.

2. *The GTEx expression data* consists of 2,921 gene expression profiles of various tissue samples obtained from the Illumina RNA-Seq platform [4]. The expression level of each gene was measured based on Gencode V12 annotations [4] in the format of Reads Per Kilobase

per Million (RPKM).

3. *The 1000 Genomes expression data* consists of 462 gene expression profiles of lymphoblastoid cell line samples from the Illumina RNA-Seq platform [74]. The expression level of each gene was also measured based on Gencode V12 annotations [74] in the format of RPKM.

Since the gene expression values of the microarray platform and the RNA-Seq platform were measured in different units (probes vs Gencode annotations) and different numerical scales, we quantile normalized the three expression datasets jointly to retain the maximum information cross platforms. Because one Gencode annotation may include multiple microarray probes, 943 landmark genes and 9,520 target genes in terms of Gencode annotations were left after joint quantile normalization. Details of joint quantile normalization are given in Supplementary [28]. Finally, all the datasets were standardized by subtracting the mean and dividing by the standard deviation of each gene.

### 5.2.2 Gene expression inference as multi-task regression

Assume there are $L$ landmark genes, $T$ target genes, and $N$ training samples (i.e. profiles); the training dataset is expressed as $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{R}^L$ denotes the expression values of landmark genes and $\mathbf{y}_i \in \mathcal{R}^T$ denotes the expression values of target genes in the $i$-th sample. Our goal is to infer the functional mapping $\mathcal{F} : \mathcal{R}^L \to \mathcal{R}^T$ that fits $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, which can be viewed as a multi-task regression problem.

We use Mean Absolute Error (MAE) to evaluate the predictive performance at each target gene $t$,

$$\text{MAE}_{(t)} = \frac{1}{N'} \sum_{i=1}^{N'} \left| y_{i(t)} - \hat{y}_{i(t)} \right| \tag{5.1}$$

where $N'$ is the number of testing samples and $\hat{y}_{i(t)}$ is the predicted expression value for target gene $t$ in sample $i$. We define the overall error as the average MAE over all target genes, and use it to evaluate the general predictive performance.

For the microarray platform, we used the GEO data for training, validation and testing. Specifically, we randomly partitioned the GEO data into ˜80% for training (88,807 samples denoted as GEO-tr), ˜10% for validation (11,101 samples denoted as GEO-va) and ˜10% for testing (11,101 samples denoted as GEO-te). The validation data GEO-va was used to do model selection and parameter tuning for all the methods.

For the RNA-Seq platform, we used GEO-tr for training, the 1000 Genomes data for validation (denoted as 1000G-va), and the GTEx data for testing (denoted as GTEx-te). The validation data 1000G-va was used to do model selection and parameter tuning for all the methods.

### 5.2.3 D-GEX

D-GEX is a multi-task multi-layer feedforward neural network. It consists of one input layer, one or multiple hidden layers, and one output layer. All the hidden layers have the same number of hidden unites. Units between layers are all fully connected. A hidden unit $j$ in layer $l$ takes the sum of weighted outputs plus the bias from the previous layer $l-1$ as the input, and produces a single output $o_j^l$ using a nonlinear activation function $f$.

$$o_j^l = f(\sum_{i=1}^{H} w_{i,j}^{l-1} o_i^{l-1} + b_j^{l-1}) \tag{5.2}$$

$H$ is the number of hidden units. $\{w_{i,j}^{l-1}, b_j^{l-1}\}_{i=1}^{H}$ are the weights and the bias associated with unit $j$ that need to be learned. We adopt the hyperbolic tangent (TANH) activation

function to hidden units, which naturally captures the nonlinear patterns within the data. Linear activation function is applied to output units for the regression purpose. The loss function for training is the sum of mean squared error at each output unit, namely,

$$\mathcal{L} = \sum_{t=1}^{T} \left[ \frac{1}{N} \sum_{i=1}^{N} \left( y_{i(t)} - \hat{y}_{i(t)} \right)^2 \right] \tag{5.3}$$

D-GEX contains 943 units in the input layer corresponding to the 943 landmark genes. Ideally, we should also configure D-GEX with 9,520 units in the output layer corresponding to the 9,520 target genes. However, each of our GPUs has only 6 GB of memory, thus we cannot configure hidden layers with sufficient number of hidden units if all the target genes are included in one output layer. Therefore, we randomly partitioned the 9,520 target genes into 2 sets that each contains 4,760 target genes. We then built 2 separate neural networks with each output layer corresponding to one half of the target genes. With this constraint, we were able to build a series of different architectures containing 1˜3 hidden layers each and each hidden layer contains 3,000, 6,000 or 9,000 hidden units. Supplementary Figure S1 [28] shows an example architecture of D-GEX with 3 hidden layers.

Training D-GEX follows the standard back-propagation algorithm [116] and mini-batch gradient descent, supplemented with advanced deep learning techniques. Detailed parameter configurations are given in Supplementary Table S1 [28]. For more descriptions about neural networks and their background please see [27]. We discuss a few key training techniques as follows:

1. *Dropout* is a technique to perform model averaging and regularization [56] for neural networks. At the training time, each unit along with its edges is temporarily dropped out with probability $p$ for each training sample. Then the forward- and back-propagation are performed on a particularly "thinned" network. For an architecture with $n$ units performing

dropout, there are $O\left(\frac{1}{(1-p)^n}\right)$ such thinned networks. At the testing time, all the units are retained with weights multiplied by $1-p$. Therefore, dropout can be seen as model averaging of exponentially many different neural networks in an approximate but efficient framework. Dropout has been shown to suppress co-adaptation among units and force each unit to learn patterns that are more generalizable [127]. The dropout rate $p$ serves as a tuning parameter that controls the intense of regularization. We applied dropout to all the hidden layers of D-GEX except for the outgoing edges from the input layer. The dropout rate was set to [0%, 10%, 25%] to compare the effect of different degrees of regularization.

2. *Momentum method* is a technique to accelerate gradient-based optimization. It accumulates a velocity in directions of gradients of the loss function across iterations and uses the velocity instead of the gradient to update parameters [130]. Given a loss function $\mathcal{L}$ with respect to the parameters $\Theta$ of the neural network, the momentum is given by

$$\begin{aligned} V^{(k+1)} &= \mu V^{(k)} - \eta^{(k)} \nabla \mathcal{L}(\Theta^{(k)}) \\ \Theta^{(k+1)} &= \Theta^{(k)} + V^{(k+1)} \end{aligned} \tag{5.4}$$

where $\mu \in [0, 1]$ is the momentum coefficient, $\eta$ is the learning rate, $V$ is the velocity, and $\nabla \mathcal{L}(\Theta)$ is the gradient of the loss function. Momentum method has been shown to improve the convergence rate particularly for training deep neural networks [130].

3. *Normalized initialization* is a technique to initialize the weights of deep neural networks [45]. The weights of a unit is sampled from a uniform distribution defined by,

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_i + n_o}}, \frac{\sqrt{6}}{\sqrt{n_i + n_o}}\right] \tag{5.5}$$

where $n_i, n_o$ denote the number of fan-ins and fan-outs of the unit. It is designed to stabilize the variances of activation and back-propagated gradients during training [45]. The uniform distribution of the output layer of D-GEX was set to be within a smaller range of $[-1 \times 10^{-4}, 1 \times 10^{-4}]$ as it was adopted with the linear activation function.

4. *Learning rate* was initialized to $5 \times 10^{-4}$ or $3 \times 10^{-4}$ depending on different architectures, and was decreased according to the training error on a subset of GEO-tr for monitoring the training process. Specifically, the training error was checked after each epoch, if the training error increased, the learning rate was multiplied by a decay factor of 0.9 until it reached a minimum learning rate of $1 \times 10^{-5}$.

5. *Model selection* was performed based on GEO-va for the GEO data and 1000G-va for the GTEx data. Training was run for 200 epochs. The model was evaluated on GEO-va and 1000G-va after each epoch, and the model with the best performance was saved respectively.

D-GEX was implemented based on two Python libraries, Theano [18] and Pylearn2 [46]. Training was deployed on an Nvidia GTX TITAN Z graphics card with dual GPUs. The largest architecture of D-GEX (3 hidden layers with 9,000 hidden units in each hidden layer) contains ˜427 million parameters. Training half of the target genes with the largest architecture took around 6 hours. D-GEX is publicly available at https://github.com/uci-cbcl/D-GEX.

### 5.2.4 Linear regression

Linear regression (LR) for multi-task gene expression inference trains a model, $\mathcal{F}_{(t)}(\mathbf{x}) = \mathbf{w}_{(t)}^T \mathbf{x} + b_{(t)}$, independently for each target gene $t$. $\mathbf{w}_{(t)} \in \mathcal{R}^L, b_{(t)} \in \mathcal{R}$ are the model param-

eters associated with each target gene $t$, and

$$\left(\mathbf{w}_{(t)}, b_{(t)}\right) = \arg\min_{\mathbf{w},b} \frac{1}{N} \sum_{i=1}^{N} \left(y_{i(t)} - \mathbf{w}_{(t)}^T \mathbf{x}_i - b_{(t)}\right)^2 \tag{5.6}$$

L1 or L2 penalties can be further introduced for regularization purpose. In these cases,

$$\left(\mathbf{w}_{(t)}, b_{(t)}\right) = \arg\min_{\mathbf{w},b} \frac{1}{N} \sum_{i=1}^{N} \left(y_{i(t)} - \mathbf{w}_{(t)}^T \mathbf{x}_i - b_{(t)}\right)^2 + \lambda \|\mathbf{w}_{(t)}\|_1 \tag{5.7}$$

or

$$\left(\mathbf{w}_{(t)}, b_{(t)}\right) = \arg\min_{\mathbf{w},b} \frac{1}{N} \sum_{i=1}^{N} \left(y_{i(t)} - \mathbf{w}_{(t)}^T \mathbf{x}_i - b_{(t)}\right)^2 + \lambda \|\mathbf{w}_{(t)}\|_2 \tag{5.8}$$

Linear regression (5.6) is currently adopted by the LINCS program. In our study, we evaluated both (5.6) and (5.7), (5.8) using scikit-learn [102]. The regularization parameter $\lambda$ was tuned based on the performance on GEO-va and 1000G-va.

### 5.2.5 K-nearest neighbor regression

K-nearest neighbor (KNN) regression is a non-parametric and instance-based method. In standard KNN regression, a spatial data structure $\mathcal{T}$ such as the KD tree [15] is built for training data in the feature space. Then, for any testing data, the k nearest training samples based on a certain distance metric are queried from $\mathcal{T}$. The average of their values is computed as the prediction.

However, the standard KNN regression may be biased when duplicated samples frequently exist in the data, such as the GEO microarray data. Therefore, in gene expression inference, a commonly adopted alternative is to query the k nearest genes rather than the k nearest

samples. Specifically, for each target gene, its euclidean distances to all the landmark genes were calculated using the training samples. The k landmark genes with the least euclidean distances are determined as the k nearest landmark genes of the target gene. Then the average of their expression values in the testing samples is computed as the prediction for the target gene. Such algorithm is also consistent with the basic assumption of the LINCS program that, the expression of target genes can be computationally inferred from landmark genes. We call this algorithm the gene-based KNN (KNN-GE).

Due to the non-parametric and instance-based nature, KNN-GE does not impose any prior assumptions on the learning machine. Therefore, it is very flexible to model nonlinear patterns within the data. However, as performing prediction involves building and querying data structures that have to keep all the training data, KNN-GE suffers from poor scalability to growing data size and dimension. We evaluated KNN-GE in our study. The optimal k was selected based on the performance on GEO-va and 1000G-va.

## 5.3 Results

We have introduced two types of gene expression data, namely the GEO microarray data and the GTEx/1000G RNA-Seq data. We have formulated the gene expression inference as a multi-task regression problem, using the GEO data for training and both the GEO and the GTEx data for testing. We have also described our deep learning method D-GEX, and another two methods, linear regression and k-nearest neighbour regression, to solve the problem. Next, we show the predictive performances of the three methods on both the GEO data and the GTEx data.

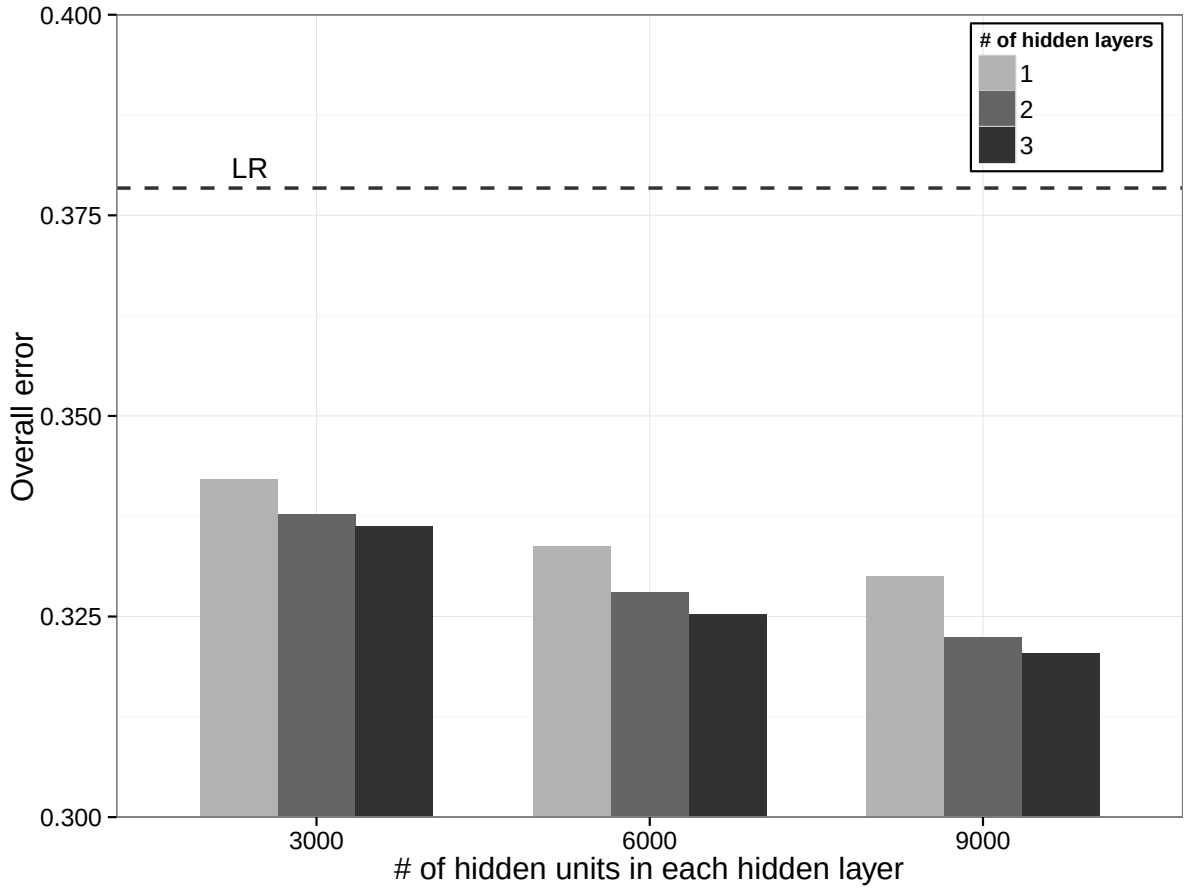



Figure 5.1: The overall errors of D-GEX-10% with different architectures on GEO-te. The performance of LR is also included for comparison.

Table 5.1: The overall errors of LR, LR-L1, LR-L2, KNN-GE and D-GEX-10% with different architectures on GEO-te. Numerics after "±" are the standard deviations of prediction errors over all target genes. The best performance of D-GEX-10% is shown in bold font. The performance selected using model selection by GEO-va of D-GEX-10% is underscored.

| # of hidden units | | 3000 | 6000 | 9000 |
|---|---|---|---|---|
| # of hidden layers | 1 | 0.3421±0.0858 | 0.3337±0.0869 | 0.3300±0.0874 |
| | 2 | 0.3377±0.0854 | 0.3280±0.0869 | 0.3224±0.0879 |
| | 3 | 0.3362±0.0850 | 0.3252±0.0868 | **0.3204±0.0879** |
| | LR | | 0.3784±0.0851 | |
| | LR-L1 | | 0.3782±0.0844 | |
| | LR-L2 | | 0.3784±0.0851 | |
| | KNN-GE | | 0.5866±0.0698 | |

### 5.3.1 Performance on the GEO data

D-GEX achieves the best performance on both GEO-va and GEO-te with 10% dropout rate (denoted as D-GEX-10%). Figure 5.1 and Table 5.1 show the overall performances of D-GEX-10% and the other methods on GEO-te. The complete performances of D-GEX with other dropout rates on both GEO-va and GEO-te are given in Supplementary Table S2 and S3 [28]. The largest architecture of D-GEX-10% (3 hidden layers with 9,000 hidden units in each hidden layer, denoted as D-GEX-10%-9000×3) achieves the best performance on both GEO-va and GEO-te. The relative improvements of D-GEX-10%-9000×3 are 15.33% over LR and 45.38% over KNN-GE. Besides D-GEX-10%-9000×3, D-GEX-10% consistently outperforms LR and KNN-GE on all the other architecture as shown in Figure 5.1. One possible explanation is that deep architectures enjoy much richer representability than shallow architectures, thus learning complex features is much easier from the perspective of optimization [13].

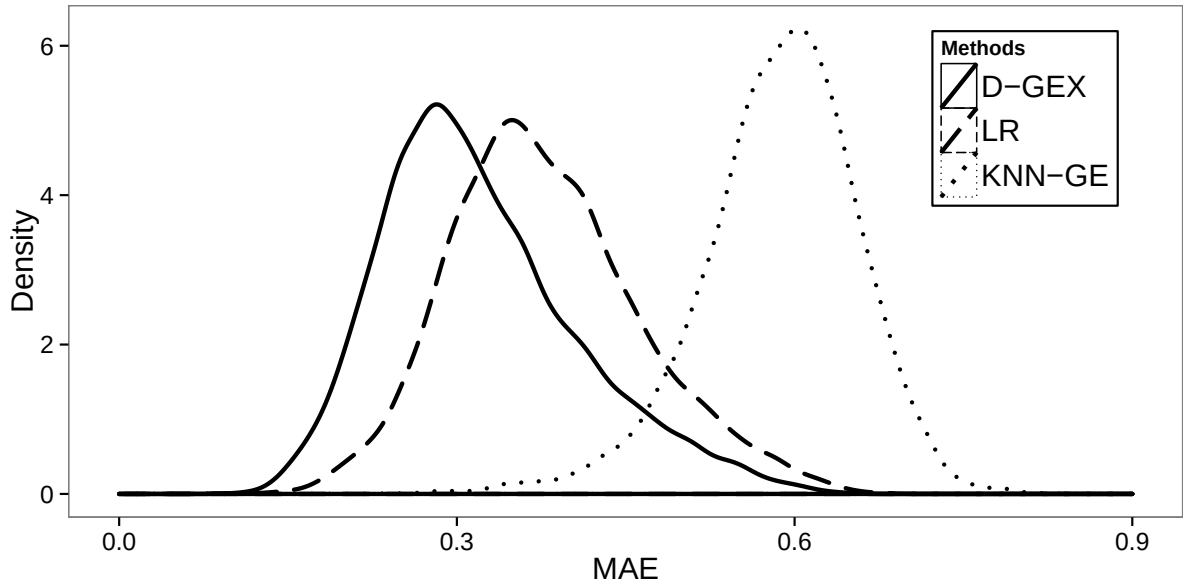
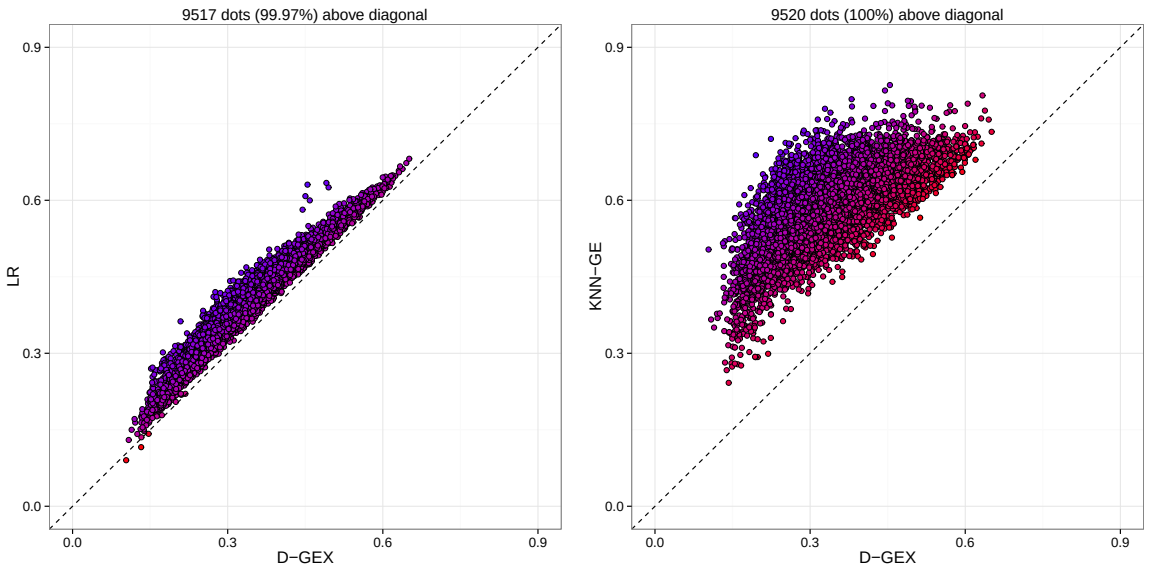



Figure 5.2: The density plots of the predictive errors of all the target genes by LR, KNN-GE and GEX-10%-9000×3 on GEO-te.

D-GEX also outperforms LR and KNN-GE for almost all of the target genes. Figure 5.2




(a) (b)

Figure 5.3: The predictive errors of each target gene by GEX-10%-9000×3 compared to LR and KNN-GE on GEO-te. Each dot represents 1 out of the 9,520 target genes. The x-axis is the MAE of each target gene by D-GEX, and the y-axis is the MAE of each target gene by the other method. Dots above diagonal means D-GEX achieves lower error compared to the other method. (a)D-GEX verse LR; (b)D-GEX verse KNN-GE.

shows the density plots of the predictive errors of all the target genes by LR, KNN-GE and GEX-10%-9000×3. Figure 5.3 shows a gene-wise comparative analysis between D-GEX-10%-9000×3 and the other two methods. D-GEX-10%-9000×3 outperforms LR in 99.97% of the target genes and outperforms KNN-GE in all the target genes. These results seem to suggest that D-GEX captured some intrinsic nonlinear features within the GEO data where LR and KNN-GE didn't.

Regularization methods do not improve LR significantly. Table 5.1 shows the relative improvements of LR-L1 and LR-L2 over LR are 0.05% and 0.00%. Thus, it is most likely that LR is underfitting which means linear model is not complex enough to represent the data. Therefore, regularization techniques that reduce model complexity are not helpful.

KNN-GE performs significantly worse than the other methods. One possible explanation is that the k nearest landmark genes for each target gene based on GEO-tr and GEO-te may not be fully consistent.

### 5.3.2 Performance on the GTEx data

Results on the GEO data demonstrate the significant improvement of D-GEX over LR and KNN-GE on the microarray platform. Yet in practice, the LINCS program trains regression models with the GEO data and performs gene expression inference on the L1000 data, which was generated with a different platform. Whether the significance of D-GEX preserves cross platforms requires further investigation. To explore the cross platforms scenario, we trained D-GEX with GEO-tr and evaluated its performances on GTEx-te which was generated with the RNA-Seq platform [84].

However, new challenges arise in this scenario as the intrinsic distributions of the training data and the testing data may be similar but not exactly equivalent. Particularly in gene

expression profiling, discrepancies between microarray and RNA-Seq data have been systematically studied [137]. Such discrepancies bring specific challenges to deep learning as the complex features it learns in the training data may not generalize well to the testing data, which leads to overfitting and reduces the prediction power. Therefore, more aggressive regularization may be necessary for deep learning to retain the maximum commonality cross platforms while avoiding platform-dependent discrepancies.

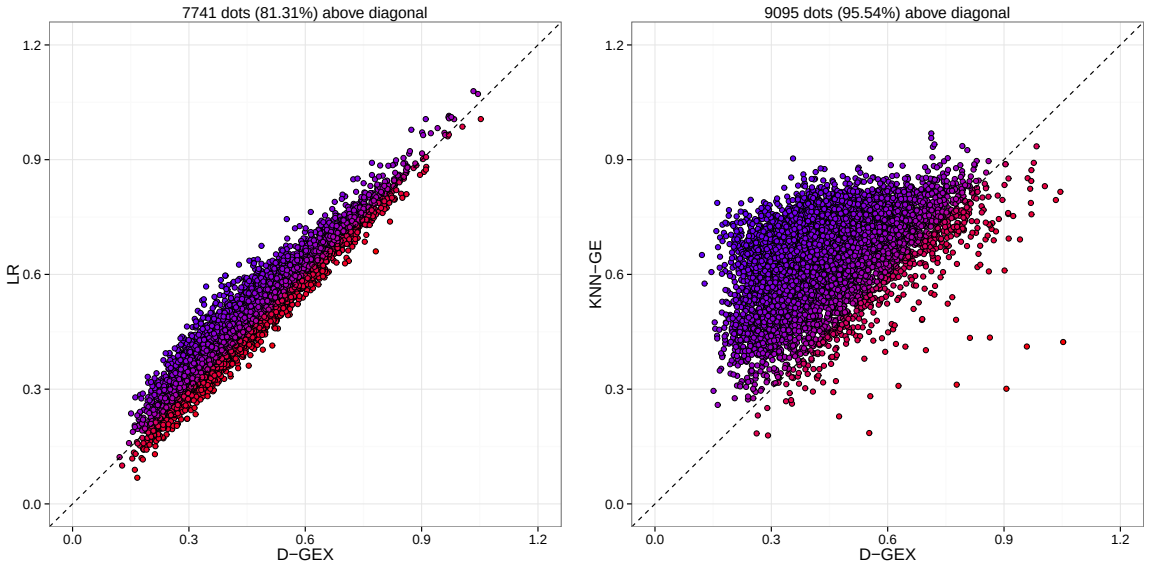
D-GEX-25%-9000×2 (with 25% dropout rate, two hidden layers with 9000 hidden units in each layer) achieves the best performance on both 1000G-va and GTEx-te. The relative improvements of D-GEX-25%-9000×2 are 6.57% over LR and 32.62% over KNN-GE. Table 5.2 shows the overall performances of D-GEX-25% and the other methods on GTEx-te. The complete performances of D-GEX with other dropout rates on both 1000G-va and GTEx-te are given in Supplementary Table S4 and S5 [28].

Table 5.2: The overall errors of LR, LR-L1, LR-L2, KNN-GE and D-GEX-25% with different architectures on GTEx-te. Numerics after "±" are the standard deviations of prediction errors over all target genes. The best performance of D-GEX-25% is shown in bold font. The performance selected using model selection by 1000G-va of D-GEX-25% is underscored.

| # of hidden units | | 3000 | 6000 | 9000 |
|---|---|---|---|---|
| # of hidden layers | 1 | 0.4507±0.1231 | 0.4428±0.1246 | 0.4394±0.1253 |
| | 2 | 0.4586±0.1194 | 0.4446±0.1226 | **0.4393±0.1239** |
| | 3 | 0.5160±0.1157 | 0.4595±0.1186 | 0.4492±0.1211 |
| | LR | | 0.4702±0.1234 | |
| | LR-L1 | | 0.5667±0.1271 | |
| | LR-L2 | | 0.4702±0.1234 | |
| | KNN-GE | | 0.6520±0.0982 | |

D-GEX still outperforms LR and KNN-GE in most of the target genes. Figure 5.4 also shows the gene-wise comparative analysis between D-GEX-25%-9000×2 and the other two methods. D-GEX-25%-9000×2 outperforms LR in 81.31% of the target genes and outperforms KNN-GE in 95.54% of the target genes. Therefore, the significance of D-GEX on the microarray platform basically preserves on the RNA-Seq platform. However, unlike the results on the

GEO data, there is a noticeable number of target genes that D-GEX gets higher error than the other methods on the GTEx data. Thus, the expression patterns of these target genes D-GEX learned on the GEO data may be platform dependent and do not generalize well to the GTEx data. It is noteworthy that although the general performance of KNN-GE is still poor on the GTEx data, its errors on some of the target genes are significantly lower than D-GEX (dots in bottom right part of Figure 5.4(b)). This is likely due to the gene-based aspect of KNN-GE that the numerical values predicted on target genes were not computed based on GEO-tr but based on GTEx-te itself. Therefore, the expression patterns captured by KNN-GE may be cross platforms invariant.




(a) (b)

Figure 5.4: The predictive errors of each target gene by GEX-25%-9000×2 compared to LR and KNN-GE on GTEx-te. Each dot represents 1 out of the 9,520 target genes. The x-axis is the MAE of each target gene by D-GEX, and the y-axis is the MAE of each target gene by the other method. Dots above diagonal means D-GEX achieves lower error compared to the other method. (a)D-GEX verse LR; (b)D-GEX verse KNN-GE.

Dropout regularization effectively improves the performance of D-GEX on the GTEx data as shown in Figure 5.5. Without dropout, the overall error of D-GEX-9000×2 on GTEx-te slightly decreases at the beginning of training and then quickly increases, clearly implying overfitting. However, with 25% dropout rate, D-GEX-9000×2 achieves the best performance

on both 1000G-va and GTEx-te.

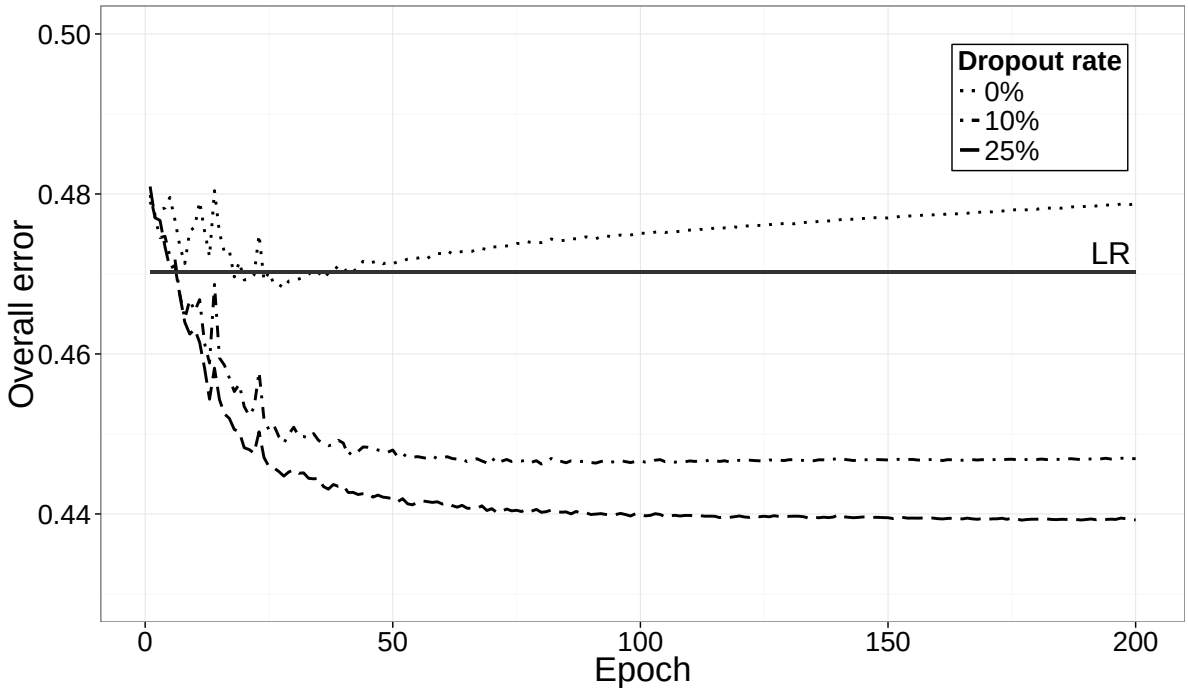



Figure 5.5: The overall error decreasing curves of D-GEX-9000×2 on GTEx-te with differant dropout rates. The x-axis is the training epoch and the y-axis is the overall error. The overall error of LR is also included for comparison.

### 5.3.3 Interpreting the learned neural network

We have demonstrated the performance of our deep learning method D-GEX on both the GEO microarray data and the GTEx RNA-Seq data. D-GEX outperforms linear regression on both types of expression data. On the other hand, interpreting the learned linear model from linear regression is straightforward as coefficients with large absolute value indicate strong dependencies between landmark genes and target genes. But for deep learning, currently there are no established methods to interpret the neutral networks learned from gene expression data. Next, we attempt to explore the learned neural networks with two strategies, a) visualizing the major weights of the learned neural networks and b) examining the nonlinearity captured by the hidden layers.

1. *Visualizing the major weights* is a strategy inspired by the method of interpreting linear model that coefficients with large absolute value indicate strong dependencies between inputs and targets. Similarly, we examined the weights of the learned neural network of D-GEX-10%-3000×1 that was trained based on half of the target genes of GEO-tr and GEO-va. The weights from input to hidden units were randomly initialized with dense connections. However, after learning, the connections became so sparse that each input unit was primarily connected to only a few hidden units with the weights to the rest of hidden units decayed to near zero. Similar patterns were also observed for connections from the hidden to the output layer. Therefore, we created a visualization map of the learned connections by removing those with weights near zeros. Specifically, for each input unit (landmark gene), we calculated the mean and the standard deviation of the weights of the connections between the input unit and the 3,000 hidden units. Then we only retained the major weights that were 4 standard deviations away from the mean. Likewise, we used a threshold of 5 standard deviations to retain the major weights of the connections between the output units (target genes) and the hidden units. We colored the weights differently so that red indicates positive weights and blue indicates negative weights. Supplementary Figure S3 [28] shows the final visualization map. From the visualization map, we noticed two interesting observations: a) Most of the units in the input layer and the output layer have connections to the hidden layer. In contrast, only a sparse number of units in the hidden layer have connections to the input and the output layer. Specially, the connections to the output layer are dominated by a few hidden units, which we refer to as the "hub units". b) Lots of the "hub units" seem to have only one type of connections to the output layer, e.g. some of them only have positive connections (red edges), while some other units only have negative connections (blue edges). It seems that these "hub units" may have captured some strong local correlations between the landmark genes and target genes.

2. *Examining the nonlinearity* is a strategy to show that the intermediate hidden layers have captured some nonlinearity within the raw expression data. The neural networks we

used are quite complex, containing several layers and many hidden units, each of which is activated through a nonlinear transfer function. To dissect the nonlinear contribution, we took a relatively simple approach by focusing on the representation (activations) from the last hidden layer. Each of the hidden unit in that layer can be viewed as a feature generated through some nonlinear transformation of the landmark genes. We then studied whether a linear regression based on these nonlinear features can achieve better performance than a linear regression based solely on the landmark genes. For this purpose, we measured the linear correlation between the activations from the last hidden layer of D-GEX-10%-9000×3 and the final targets (the expression of target genes), and compared it with the linear correlation between the raw inputs and the final targets. Normally, coefficient of determination ($R^2$) is used to compare the fitnesses of different linear models. Since the dimensionality has changed from the raw inputs to the transformed activations, we used adjusted $R^2$ [132] to specifically account for the change in dimensionality. We calculated the adjusted $R^2$ of both the raw inputs and the transformed activations for each target gene based on GEO-tr. Supplementary Figure S2 [28] shows the gene-wise comparison of adjusted $R^2$ between the raw inputs and the transformed activations. The transformed activations have a larger adjusted $R^2$ than the raw inputs in 99.99% of the target genes. It seems to indicate that the intermediate hidden layers have systematically captured some nonlinearity within the raw expression data that would be ignored by simple linear regression. After the nonlinear transformation through the hidden layers, the activations fit the final targets significantly better than the raw inputs using a simple linear model. The analysis seems to suggest that most of the target genes benefit from the additional nonlinear features, although to a different extent as characterized by the adjusted $R^2$.

### 5.3.4 Inference on the L1000 data

The LINCS program has used the L1000 technology to measure the expression profiles of the 978 landmark genes under a variety of experimental conditions. It currently adopts linear regression to infer the expression values of the 21,290 target genes based on the GEO data. We have demonstrated our deep learning method D-GEX achieved significantly improvement on prediction accuracy over linear regression on the GEO data. Therefore, we have re-trained GEX-10%-9000×3 using all the 978 landmark genes and the 21,290 target genes from the GEO data and inferred the expression values of unmeasured target genes from the L1000 data. The full dataset consists of 1,328,098 expression profiles and can be downloaded at `https://cbcl.ics.uci.edu/public_data/D-GEX/l1000_n1328098x22268.gctx`. We hope this dataset will be of great interest to researchers who are currently querying the LINCS L1000 data.

## 5.4 Discussion

Revealing the complex patterns of gene expression under numerous biological states requires both cost-effective profiling tools and powerful inference frameworks. While the L1000 platform adopted by the LINCS program can efficiently profile the ˜1,000 landmark genes, the linear-regression-based inference does not fully leverage the nonlinear features within gene expression profiles to infer the ˜21,000 target genes. We presented a deep learning method for gene expression inference that significantly outperforms linear regression on the GEO microarray data. With dropout as regularization, our deep learning method also preserves cross platforms generalizability on the GTEx RNA-Seq data. In summary, deep learning provides a better model than linear regression for gene expression inference. We believe it achieves more accurate predictions for target gene expressions of the LINCS dataset generated from the L1000 platform.

Interpreting the internal representation of deep architectures is notoriously difficult. Unlike other machine learning tasks such as computer vision, where we can visualize the learned weights of hidden units as meaningful image patches, interpreting the deep architectures learned by biological data requires novel thinking. We attempted to interpret the internal structures of the neural networks learned from gene expression data using strategies that were inspired by linear model. Yet, more systematic studies may require advanced computational frameworks that are specifically designed for deep learning. Unsupervised feature learning methods, such as autoencoder [136] and restricted Boltzmann machine [54] may provide some insights on this problem.

In the current setting, target genes were randomly partitioned into multiple sets, and each set was trained separately using different GPUs due to hardware limitations. Alternatively, we could first cluster target genes based on their expression profiles, and then partition them accordingly rather than randomly. The rationale is that target genes sharing similar expression profiles share weights in the context of multi-task neural networks. Ultimately, the solution is to jointly train all target genes, either by using GPUs with larger memory such as the more recent Nvidia Tesla K80, or by exploiting multi-GPU techniques [32].

# Chapter 6

# Understanding sequence conservation with deep learning

## 6.1 Introduction

Numerous conserved elements have been detected through comparative genomics [122, 34]. This is because conserved elements tend to be functional and are believed to be under negative (purifying) selection. Thus they evolve at a significant slower rate than other non-conserved (neutral) sequences, and develop distinct sequence patterns. Studies based on human and rodent genomes estimate that about 5% bases of mammalian genomes are under negative selection, among which coding regions only account for 1.5% [122]. Therefore, extensive studies and methods have been focused on understanding the functional roles of conserved sequences in noncoding regions. Nevertheless, the exact function of conserved non-coding sequences remains elusive.

Recent advances in deep learning, specifically in solving sequence-based problems in genomics with convolutional neural networks [76, 107, 2, 150], provide a new powerful method to study

sequence conservation. Deep learning refers to algorithms that learn a hierarchical nonlinear representation of large datasets through multiple layers of abstraction (e.g. convolutional neural networks, multi-layer feedforward neural networks, and recurrent neural networks). It has achieved state-of-the-art performances on several machine learning applications such as speech recognition [55], natural language processing [125], and computer vision [69]. Most recently, deep learning methods have also been adapted to solve genomics problems such as motif discovery [107, 2, 150], pathogenic variants identification [106], and gene expression inference [28].

Here we present a deep learning method for studying sequence conservation (DeepCons). DeepCons is a convolutional neural network trained to predict whether a given DNA sequence is conserved or not. By learning to discriminate between conserved and non-conserved sequences, DeepCons can capture rich information about conserved sequences, such as motifs. Specifically, we show that, 1) the learned convolution kernels significantly match to known motifs, such as regulatory motifs CTCF and the RFX family, that are known to be widely distributed within conserved noncoding elements [141], 2) the kernels have positional bias relative to transcription start sites (TSS), transcription end sites (TES) and miRNA, indicating their potential roles in post-translational regulation, and 3) the kernels that are close to TES display strand bias, suggesting their RNA level regulatory effects. We further demonstrate that DeepCons could be used to score sequences at nucleotide level resolution in terms of conservation. We rediscovered known motifs, such as CTCF, JUND, RFX3 and MEF2A, within a given sequence by highlighting each nucleotide regarding their scores. Finally, we show that the learned convolution kernels represents a large variety of motifs, and we have made all the kernels publicly available online in the MEME [6] format. We hope researchers may draw new biological insights from these motifs.

## 6.2 Methods

### 6.2.1 DeepCons

DeepCons is a convolutional neural network [76] composed of one input layer, three hidden layers and one output layer. The first input layer uses one hot encoding to represent each input sequence as a 4-row binary matrix, with the number of columns equal to the length of the sequence. The second layer is a convolution layer composed of 1,000 convolution kernels of 10 bp length and 500 convolution kernels of 20 bp length. All the convolution kernels use rectified linear function as the activation function. Each convolution kernel acts as a motif detector that scans across input matrices and produces different strengths of signals that are correlated to the underlying sequence patterns. The third layer is a max pooling layer that takes the maximum output signal of each convolution kernel along the whole sequence. The fourth layer is a fully connected layer of 1,500 hidden units with rectified linear function as the activation function. The last layer performs a non-linear transformation with sigmoid activation and produces a value between 0 and 1 that represents the probability of a sequence being conserved. DeepCons contains ∼2.3 million parameters. Figure 6.1 shows the neural network architecture of DeepCons.

DeepCons was trained using the standard back-propagation algorithm [116] and mini-batch gradient descent with the Adagrad [37] variation. Dropout [56] and early stopping were used for regularization and model selection.

DeepCons was implemented based on two Python libraries, Theano [17] and Keras `http://keras.io/`. Training was performed on an Nvidia GTX TITAN Z graphics card. DeepCons is publicly available at `https://github.com/uci-cbcl/DeepCons`.

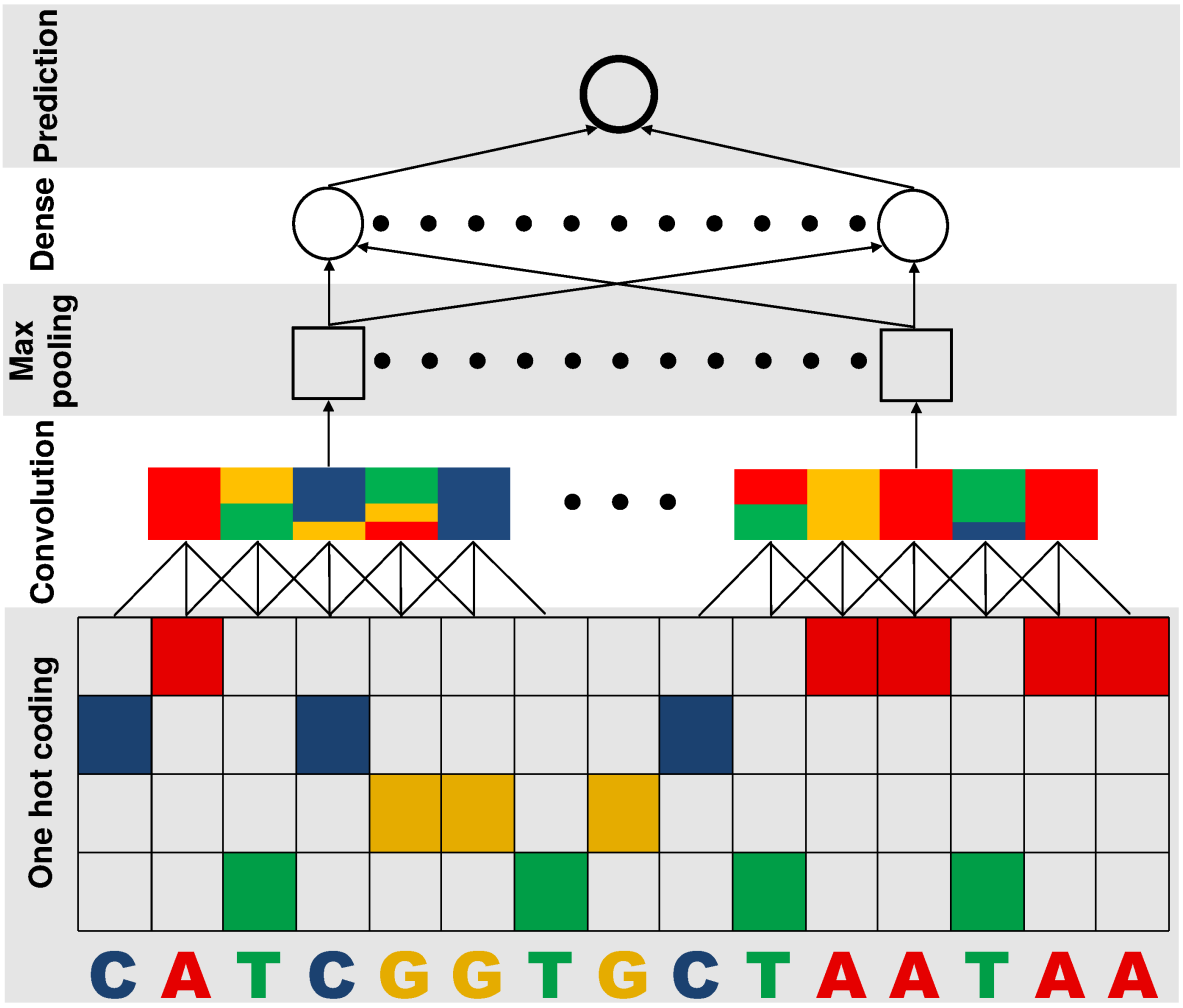



Figure 6.1: The neural network architecture of DeepCons.

### 6.2.2 Logistic regression

We also trained a baseline model using logistic regression (LR) for benchmarking purpose. Instead of using raw sequences as inputs, we first computed the counts of different k-mers of length 1-5 bp [107]. We then normalized the counts by subtracting the mean and dividing by the standard deviation and used these values as features. We also added a small L2 regularization of $1e$-6 to the cross entropy loss function of LR during training. LR was implemented with the scikit-learn [101] library.

## 6.3 Results

In this section, we first introduce the dataset of conserved and non-conserved sequences we used in this study and show the performances of both DeepCons and LR on this dataset. Next, we demonstrate DeepCons captures rich information within the conserved sequences by showing that the learned convolution kernels correspond to known motifs, have positional bias relative to TSS, TES and miRNA, and display strand bias relative to TES. We further demonstrate that the learned model could be used to score the importance of each nucleotide within a given sequence in terms of conservation, and rediscovered known motifs with these scores. Finally, we clustered all the convolution kernels and show that they represents a large variety of informative motifs.

### 6.3.1 Classifying conserved and non-conserved sequences

We first show that DeepCons can accurately discriminate between conserved and non-conserved sequences. To build the dataset of conserved sequences, we downloaded the 46-way phastCons conserved elements [122] under mammal category from UCSC genome browser

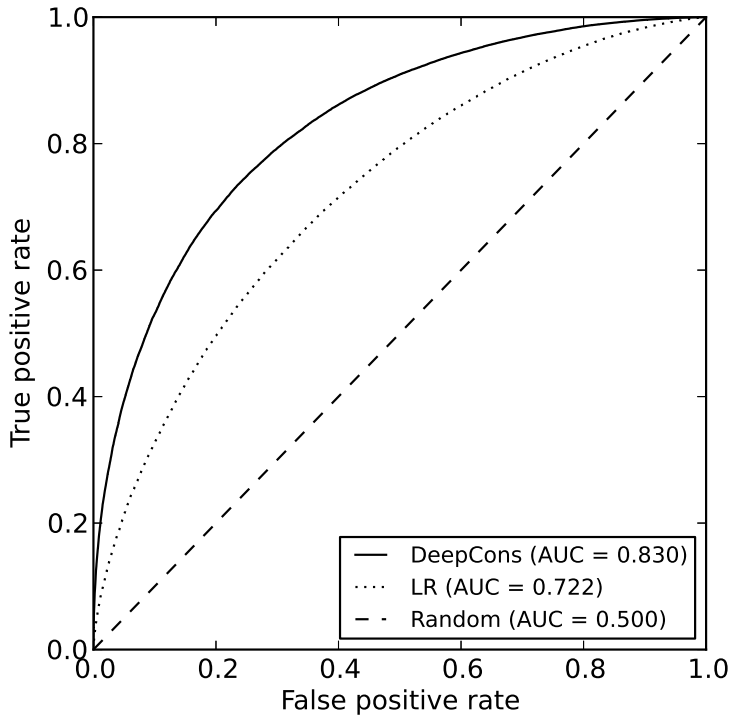



Figure 6.2: The ROC curves of DeepCons and LR on classifying conserved and non-conserved sequences on the testing dataset.

[62] based on hg19. We excluded conserved sequences that overlap with repetitive sequences (`http://www.repeatmasker.org/`). or coding exons. We then filtered away conserved sequences that were either shorter than 30 bp or longer than 1,000 bp for training the model, leaving 887,577 sequences in the end. 75% of the nucleotides were preserved after the length filtering. To build the dataset of non-conserved sequences, we randomly shuffled the 887,577 conserved sequences on hg19, excluding repetitive sequences, coding exons and conserved sequences themselves. After combining both conserved and non-conserved sequences, we randomly set aside ∼80% for training (1,415,154 sequences), ∼10% for validation (180,000 sequences) and ∼10% for testing (180,000 sequences).

DeepCons achieved 74.9% accuracy and an area under the curve (AUC) of 0.830 on the testing dataset. The baseline LR model achieved 65.9% accuracy and 0.722 AUC on the testing dataset. Figure 6.2 shows the receiver operating characteristic (ROC) curves of both DeepCons and LR on the testing dataset. DeepCons outperforms the baseline LR model

significantly on classifying conserved and non-conserved sequences.

### 6.3.2 Known motifs

Previous results have shown that regulatory motifs are widely distributed within conserved noncoding elements across the human genome, such as CTCF and the RFX family [141]. We observed similar results when examining the learned convolution kernels of DeepCons. Specifically, we converted the kernels from the convolution layer to position weight matrices, using the method described in DeepBind [2]. Then, we aligned these kernels to known motifs using TOMTOM [48]. 69 kernels match known motifs significantly ($E < 1e-2$), including the CTCF and RFX families. Figure 6.3 shows four examples of identified known motifs.

### 6.3.3 Positional bias

We observed that many of the convolution kernels have display bias relative to TSS, TES and miRNA. Specifically, we downloaded RefSeq gene models[105] and obtained 4,000 bp sequences centered on the TSS or the TES of each gene. Then, we used CentriMo [7] to assess the positional bias of each kernel relative to TSS and TES. 264 and 779 kernels have significant ($E < 1e-5$) positional bias relative to TSS and TES, respectively, indicating these kernels have potential roles in post-translational regulation. Figure 6.4 shows the positional distributions of the top four biased kernels relative to TSS and TES. We note that, the well known polyadenylation signal AATAAA and its reverse compliment TTTATT are among the top four positional biased kernels relative to TES. Previous results have also reported that motifs discovered in conserved sequences have positional bias relative to TSS and TES [140].

Besides positional bias relative to TSS and TES, we also observed several kernels have

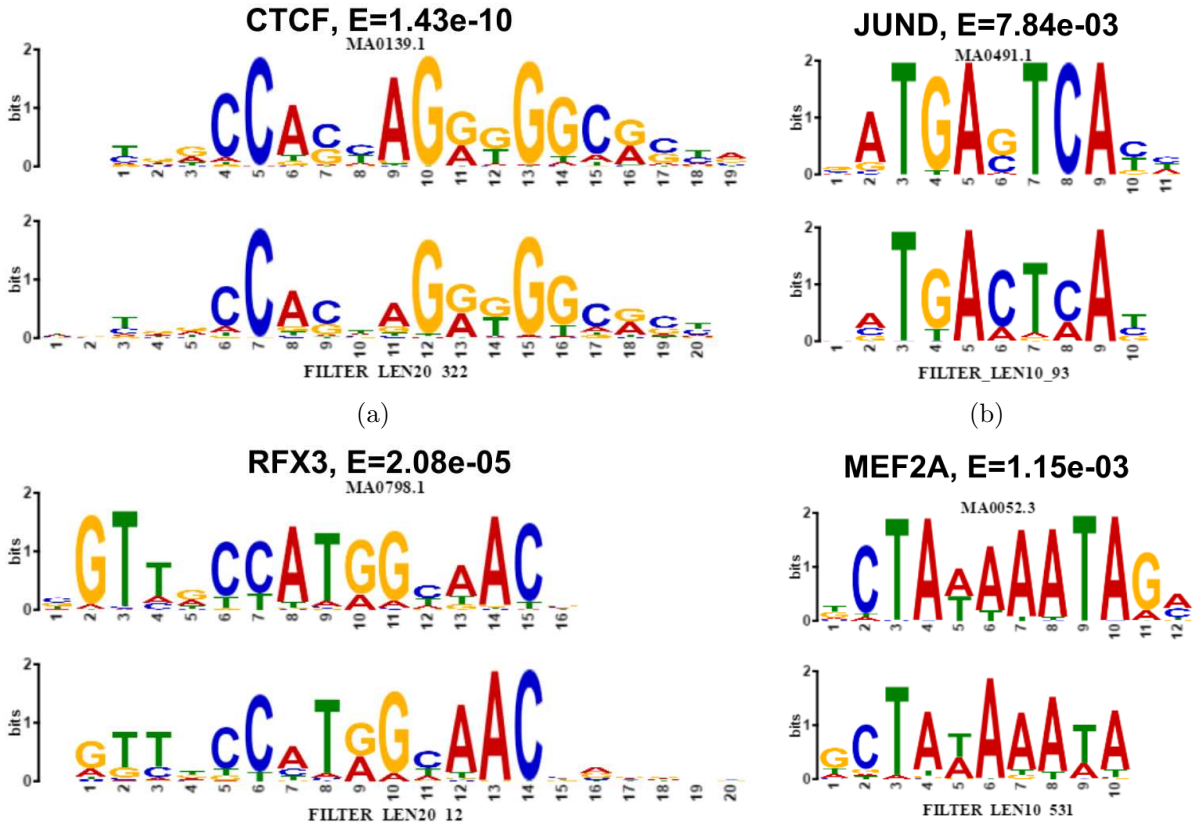



Figure 6.3: Four known motifs (top) aligned with convolution kernels (bottom). E-values of the match are displayed. (a)CTCF; (b)JUND; (c)RFX3; (d)MEF2A.

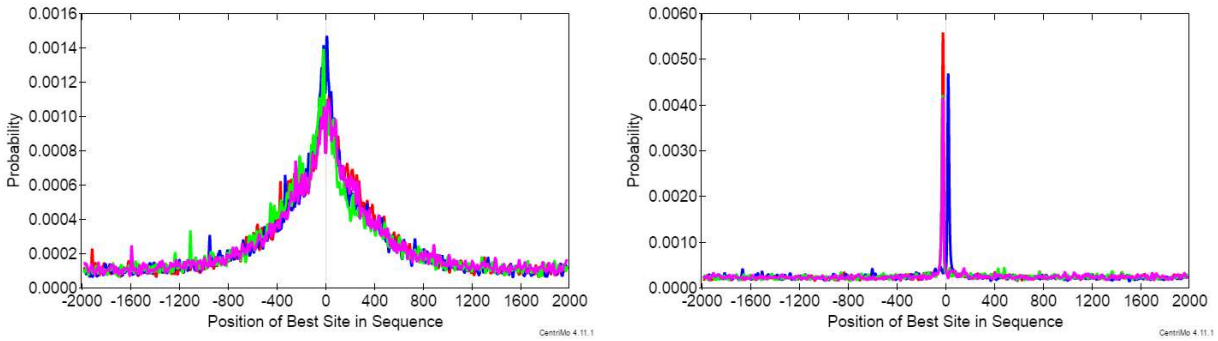



Figure 6.4: The positional distributions of the top four biased kernels relative to TSS and TES. (a)TSS; (b)TES.

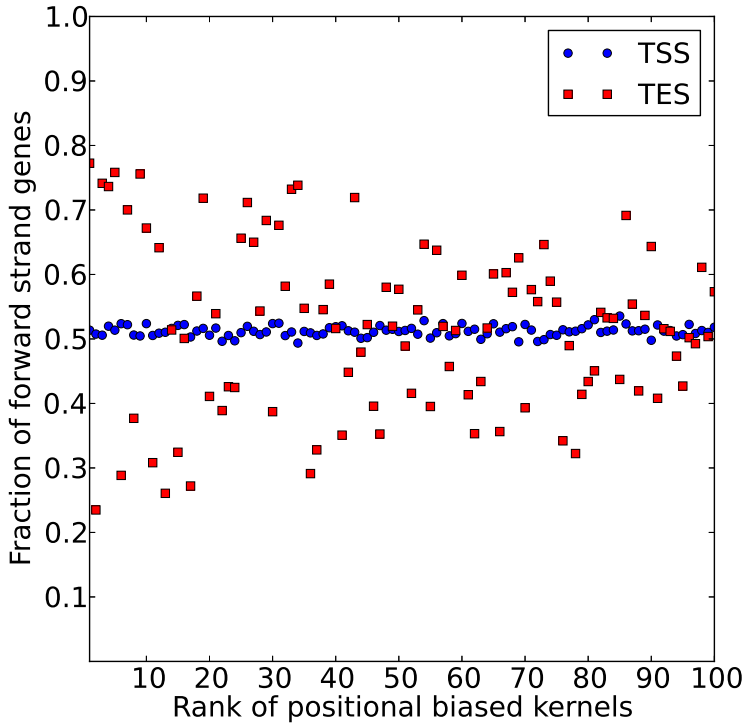



Figure 6.5: The strand bias of the top 100 positional biased kernels relative to TSS and TES. The x-axis is the rank of each kernel. The y-axis is the fraction of forward strand genes that each kernel is positional biased to.

positional bias relative to miRNA. We downloaded all the 1,881 human hairpin miRNA from miRBase [68] and used CentriMo [7] to test the positional bias of each kernel relative to miRNA. 122 kernels have significant ($E < 1e-5$) positional bias relative to the first 10 positions of miRNA. Previous results have also reported that 95% of 8-mers discovered in conserved sequences match the first 10 positions of miRNA [140].

### 6.3.4 Strand bias

In addition to positional bias, we also observed the convolution kernels that are close to TES have strand bias. Specifically, for each of the top 100 positional biased kernels relative to TSS and TES, we looked at the strand of the genes that the kernel is close to. Figure 6.5 shows the fractions of forward strand genes for each kernel. We found the fractions of

forward strand genes is tightly distributed around 0.5 for kernels that positional biased to TSS, while the fractions significantly deviate from 0.5 for kernels that are positional biased to TES, suggesting those kernels also have strand bias and their RNA level regulatory effects.

### 6.3.5 Scoring sequences at nucleotide level resolution

We adopted the method of saliency maps [124, 121] to compute the gradient of a given sequence and used it as a score to annotate each nucleotide within the sequence. Negative gradients were clipped to 0. Figure 6.6 shows the saliency maps of four conserved sequences. The motifs of CTCF, JUND, RFX3 and MEF2A are clearly recovered in this example, demonstrating their relevancy to conservation.

### 6.3.6 Motifs summary

Finally, we clustered all 1,500 convolution kernels into 820 clusters, using RSAT motif hierarchical clustering tool [88]. The clustering results suggest that DeepCons learned a large variety of informative motifs (Figure 6.7). The complete RSAT clustering results and the 1,500 kernels in the format of MEME [6] are publicly available online at `https://github.com/uci-cbcl/DeepCons`.

## 6.4 Discussion

Comparative genomics is an powerful tool in finding functional elements across the human genome. However, our understanding of the functional roles of these conserved sequences remains incomplete, especially in noncoding regions. Here we present a deep learning approach, DeepCons, to understand sequence conservation by training a convolutional neural

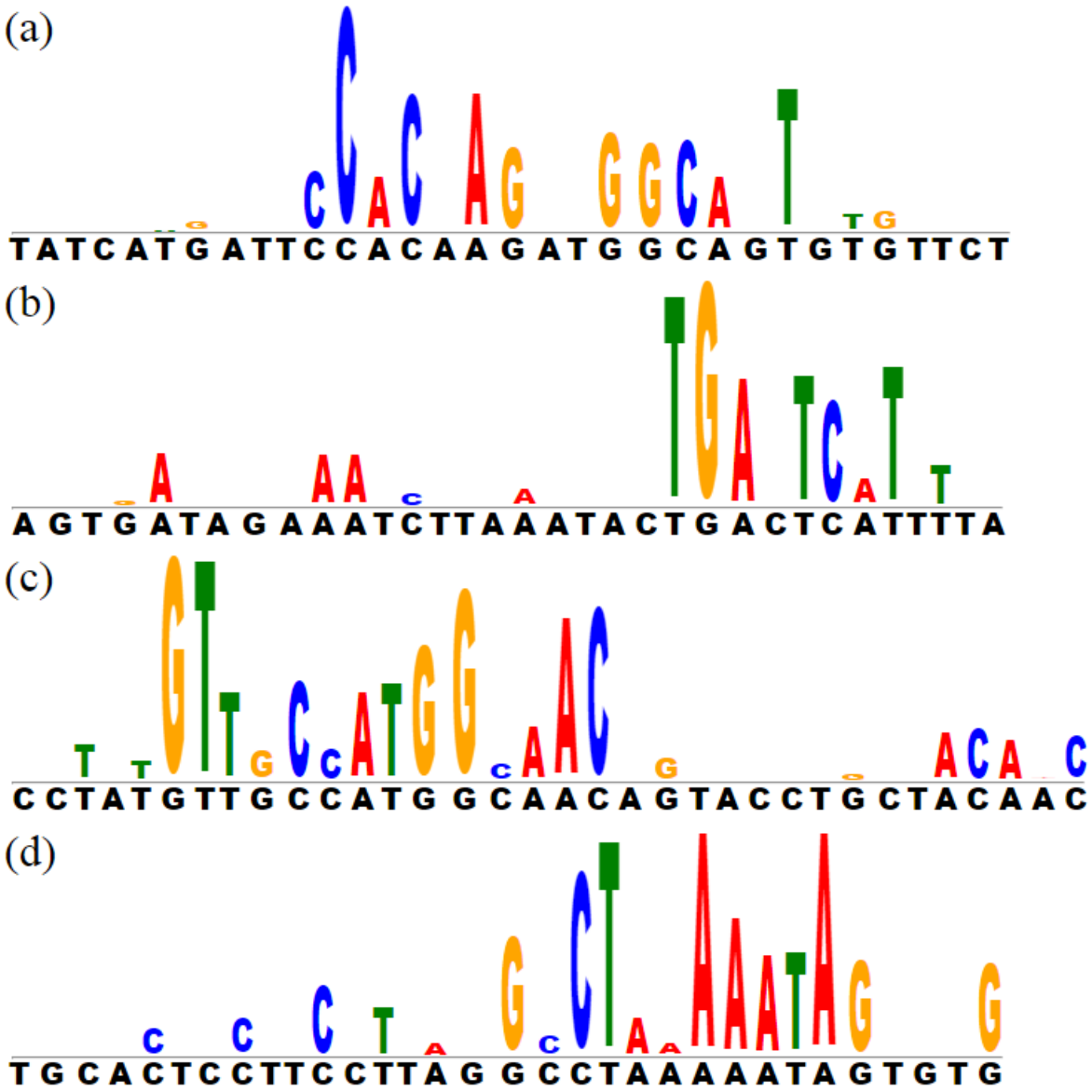



Figure 6.6: The saliency maps of four conserved sequences. The black letters below the gray line are the nucleotides of each sequence. The colored letters above the gray line are the nucleotides highlighted by their gradients, with the height proportional to the gradient. Four motifs are rediscovered in this example. (a)CTCF; (b)JUND; (c)RFX3; (d)MEF2A.

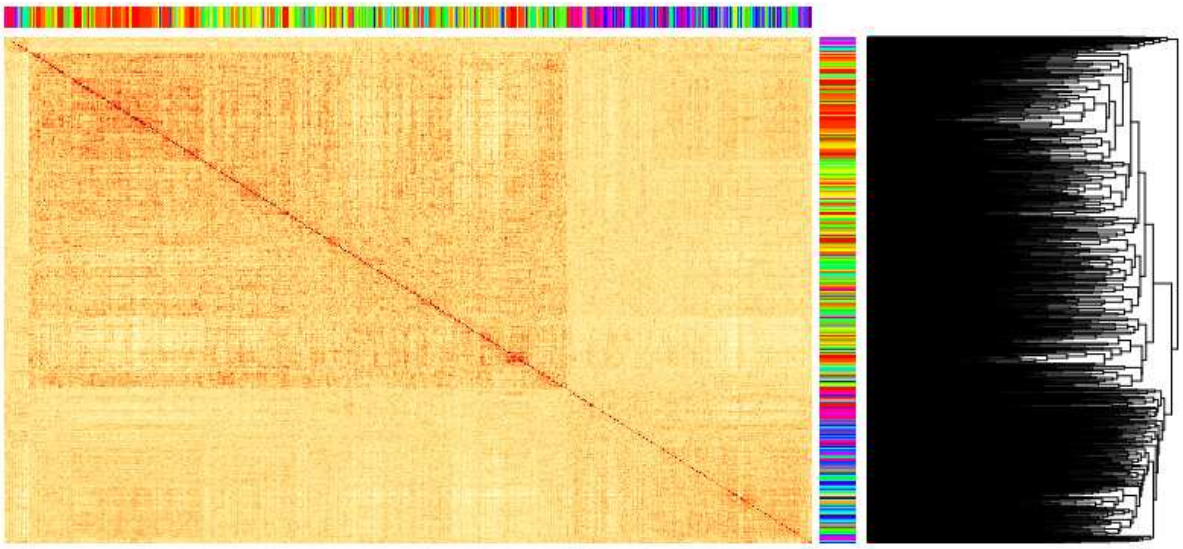


Figure 6.7: The hierarchical clustering heatmap of all the 1,500 kernels using RSAT motif clustering tool [88]

network to classify conserved and non-conserved sequences. The learned convolution kernels of DeepCons captured rich information with respect to sequence conservation that 1) they match to known motifs that are known to be widely distributed within conserved noncoding elements, 2) they have positional bias relative to both transcription start sites (TSS), transcription end sites (TES) and miRNA, indicating their potential roles in post-translational regulation, and 3) they also have strand bias relative to TES, suggesting their RNA level regulatory effects. We further demonstrate that DeepCons could be used to score sequence conservation at nucleotide level resolution. We rediscovered known motifs, such as CTCF, JUND, RFX3 and MEF2A, within a given sequence by highlighting each nucleotide regarding their scores. We have made all the kernels publicly available online at `https://github.com/uci-cbcl/DeepCons` as motifs, and we hope researchers may discover new biology by studying these motifs.

Convolutional neural networks are very effective at finding local sequence patterns through its kernels, but the kernels will typically fail to find long range sequence patterns that correspond to complex regulatory mechanisms. The size of the pattern mostly depends on the length of the convolution kernel, which typically ranges from a few bases to less than

one hundred bases. Using multiple convolutional layers may help to capture broader ranges of sequence patterns, but interpreting kernels at top layers that are not directly connected to the input sequences remains difficult. Long short term memory (LSTM) networks [57], on the other hand, are specifically designed to capture long term sequential patterns, and have been widely applied to analysis natural languages [131]. However, LSTM is also very inefficient to train since its backpropagation step is equivalent to passing the error through dozens, even hundreds, of layers. We applied LSTM to classify conserved and non-conserved sequences, but due to the large training set the algorithm took prohibitively long time to just finish even one epoch. Next, we plan to investigate multi-GPU training schemes that are now supported by TensorFlow [1], and hopefully this solution will speed up training LSTM to within an acceptable time range. Interpreting LSTM trained on sequence data also requires novel thinking. Visualizing the memory cell activities [63] may shed some lights on revealing long term sequence patterns.

# Chapter 7

# Conclusion

In this thesis, we have presented five machine learning models to solve different problems in analysing high throughput genomic data. The first three models focused on deconvolving high throughput sequencing data from heterogeneous tumor samples using unsupervised probabilistic learning methods. The last two models focused on modelling the nonlinear and hierarchical patterns within large scale genomic data using supervised deep learning methods. For the tumor heterogeneity problem, we first developed a probabilistic model to estimate tumor purity based on somatic copy number alterations observed in whole genome sequencing data. We then extended the model to further estimate the cellular prevalences of different subclonal populations within heterogeneous tumor samples. In addition to DNA sequencing data, we also developed a probabilistic model to estimate the transcriptome expression of tumor cells within heterogeneous tumor samples using RNA-Seq data. For analysing large scale genomic data using deep learning methods, we developed a multi-task deep neural network to infer the expression of ˜21,000 target genes given the expression of ˜1,000 landmark genes. We also developed a convolutional neural network to study conserved DNA sequences in non-coding regions of the human genome. The source code of all the five machine learning models are available at `https://github.com/uci-cbcl`.

High throughput technologies such as high throughput sequencing, have significantly advanced in the past decade. The cost of sequencing one human genome has extremely decreased for 100K fold, from about $100M in 2001 to about $1K in 2015. And enormous genomic data has been generated with these technological advances, such as large-scale cancer genome projects launched by International Cancer Genome Consortium (ICGC) [59] and The Cancer Genome Atlas (TCGA) [138]. To analyze this genomic "big data", advanced machine learning techniques that are both scalable and capable of modelling complex patterns are needed. Fortunately, machine learning research especially research in deep learning has also progressed rapidly in the past few years. Deep learning based artificial intelligence has surpassed human-level performances in various applications, such as image recognition [53] and the classical board game Go [123]. User-friendly software packages for training deep neural networks are also available from open-source projects, such as TensorFlow [1]. Therefore, more research and applications in genomics are expected to move from hypothesis-driven approaches to data-driven approaches, and machine learning plays an essential role in this move.

# Bibliography


[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 2015.

[3] N. Andor, J. V. Harness, S. Müller, H. W. Mewes, and C. Petritsch. Expands: expanding ploidy and allele frequency on nested subpopulations. *Bioinformatics*, 30(1):50–60, 2014.

[4] K. G. Ardlie, D. S. Deluca, A. V. Segrè, T. J. Sullivan, T. R. Young, E. T. Gelfand, C. A. Trowbridge, J. B. Maller, T. Tukiainen, M. Lek, et al. The genotype-tissue expression (gtex) pilot analysis: Multitissue gene regulation in humans. *Science*, 348(6235):648–660, 2015.

[5] AS3D. Alternative splicing structural genomics project. http://www.as3d.org/. 2012.

[6] T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble. Meme suite: tools for motif discovery and searching. *Nucleic acids research*, page gkp335, 2009.

[7] T. L. Bailey and P. Machanick. Inferring direct dna binding from chip-seq. *Nucleic acids research*, 40(17):e128–e128, 2012.

[8] P. Baldi and S. Brunak. *Bioinformatics: the machine learning approach*. MIT press, 2001.

[9] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5, 2014.

[10] P. Baldi and P. J. Sadowski. Understanding dropout. In *Advances in Neural Information Processing Systems*, pages 2814–2822, 2013.

[11] S. Banerji, K. Cibulskis, C. Rangel-Escareno, K. K. Brown, S. L. Carter, A. M. Frederick, M. S. Lawrence, A. Y. Sivachenko, C. Sougnez, L. Zou, et al. Sequence analysis of mutations and translocations across breast cancer subtypes. *Nature*, 486(7403):405–409, 2012.

[12] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. Di Bernardo. How to infer gene networks from expression profiles. *Molecular systems biology*, 3(1), 2007.

[13] Y. Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[14] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[15] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[16] M. F. Berger, M. S. Lawrence, F. Demichelis, Y. Drier, K. Cibulskis, A. Y. Sivachenko, A. Sboner, R. Esgueva, D. Pflueger, C. Sougnez, et al. The genomic complexity of primary human prostate cancer. *Nature*, 470(7333):214–220, 2011.

[17] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7, 2010.

[18] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[19] A. V. Biankin, N. Waddell, K. S. Kassahn, M.-C. Gingras, L. B. Muthuswamy, A. L. Johns, D. K. Miller, P. J. Wilson, A.-M. Patch, J. Wu, et al. Pancreatic cancer genomes reveal aberrations in axon guidance pathway genes. *Nature*, 491(7424):399–405, 2012.

[20] G. R. Bignell, J. Huang, J. Greshock, S. Watt, A. Butler, S. West, M. Grigorova, K. W. Jones, W. Wei, M. R. Stratton, et al. High-resolution analysis of dna copy number using oligonucleotide microarrays. *Genome research*, 14(2):287–295, 2004.

[21] R. Bohnert and G. Rätsch. rquant. web: a tool for rna-seq-based transcript quantitation. *Nucleic acids research*, 38(suppl 2):W348–W351, 2010.

[22] R. A. Burrell, N. McGranahan, J. Bartek, and C. Swanton. The causes and consequences of genetic heterogeneity in cancer evolution. *Nature*, 501(7467):338–345, 2013.

[23] P. J. Campbell, P. J. Stephens, E. D. Pleasance, S. O'Meara, H. Li, T. Santarius, L. A. Stebbings, C. Leroy, S. Edkins, C. Hardy, et al. Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nature genetics*, 40(6):722–729, 2008.

[24] O. Cappé and E. Moulines. Online em algorithm for latent data models. *Journal of the Royal Statistical Society*, 2008.

[25] S. L. Carter, K. Cibulskis, E. Helman, A. McKenna, H. Shen, T. Zack, P. W. Laird, R. C. Onofrio, W. Winckler, B. A. Weir, et al. Absolute quantification of somatic dna alterations in human cancer. *Nature biotechnology*, 30(5):413–421, 2012.

[26] K. Chan, P. Jiang, Y. Zheng, G. Liao, H. Sun, J. Wong, S. Siu, W. Chan, S. Chan, A. Chan, et al. Cancer genome scanning in plasma: Detection of tumor-associated copy number aberrations, single-nucleotide variants, and tumoral heterogeneity by massively parallel sequencing. *Clinical Chemistry*, 2012.

[27] Y. Chen. *Machine Learning for Large-Scale Genomics: Algorithms, Models and Applications*. PhD thesis, University of California, Irvine, ProQuest, UMI Dissertations Publishing, 12 2014.

[28] Y. Chen, Y. Li, R. Narayan, A. Subramanian, and X. Xie. Gene expression inference with deep learning. *Bioinformatics*, page btw074, 2016.

[29] D. Y. Chiang, G. Getz, D. B. Jaffe, M. J. O'Kelly, X. Zhao, S. L. Carter, C. Russ, C. Nusbaum, M. Meyerson, and E. S. Lander. High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nature methods*, 6(1):99–103, 2008.

[30] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[31] R. Clarke, H. Ressom, A. Wang, J. Xuan, M. Liu, E. Gehan, and Y. Wang. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews Cancer*, 8(1):37–49, 2008.

[32] A. Coates, B. Huval, T. Wang, D. Wu, B. Catanzaro, and N. Andrew. Deep learning with cots hpc systems. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1337–1345, 2013.

[33] F. Collins and A. Barker. Mapping the cancer genome. *Scientific American Magazine*, 296(3):50–57, 2007.

[34] E. V. Davydov, D. L. Goode, M. Sirota, G. M. Cooper, A. Sidow, and S. Batzoglou. Identifying a high fraction of the human genome to be under selective constraint using gerp++. *PLoS Comput Biol*, 6(12):e1001025, 2010.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[36] P. Di Lena, K. Nagata, and P. Baldi. Deep architectures for protein contact map prediction. *Bioinformatics*, 28(19):2449–2457, 2012.

[37] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[38] R. Edgar, M. Domrachev, and A. E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002.

[39] M. Emmert-Buck, R. Bonner, P. Smith, R. Chuaqui, Z. Zhuang, S. Goldstein, R. Weiss, L. Liotta, et al. Laser capture microdissection. *Science*, 274(5289):998–1001, 1996.

[40] T. Erkkilä, S. Lehmusvaara, P. Ruusuvuori, T. Visakorpi, I. Shmulevich, and H. Lähdesmäki. Probabilistic analysis of gene expression measurements from heterogeneous tissues. *Bioinformatics*, 26(20):2571–2577, 2010.

[41] B. Ewing and P. Green. Base-calling of automated sequencer traces usingphred. ii. error probabilities. *Genome research*, 8(3):186–194, 1998.

[42] S. A. Forbes, N. Bindal, S. Bamford, C. Cole, C. Y. Kok, D. Beare, M. Jia, R. Shepherd, K. Leung, A. Menzies, et al. Cosmic: mining complete cancer genomes in the catalogue of somatic mutations in cancer. *Nucleic acids research*, 39(suppl 1):D945–D950, 2011.

[43] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[44] L. A. Garraway and E. S. Lander. Lessons from the cancer genome. *Cell*, 153(1):17–37, 2013.

[45] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[46] I. J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*, 2013.

[47] C. D. Greenman, G. Bignell, A. Butler, S. Edkins, J. Hinton, D. Beare, S. Swamy, T. Santarius, L. Chen, S. Widaa, et al. Picnic: an algorithm to predict absolute allelic copy number variation with microarray cancer data. *Biostatistics*, 11(1):164–175, 2010.

[48] S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey, and W. S. Noble. Quantifying similarity between motifs. *Genome biology*, 8(2):R24, 2007.

[49] A. Gusnanto, H. M. Wood, Y. Pawitan, P. Rabbitts, and S. Berri. Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. *Bioinformatics*, 28(1):40–47, 2012.

[50] I. Hajirasouliha, A. Mahmoody, and B. J. Raphael. A combinatorial approach for analyzing intra-tumor heterogeneity from high-throughput sequencing data. *Bioinformatics*, 30(12):i78–i86, 2014.

[51] K. Hansen, S. Brenner, and S. Dudoit. Biases in illumina transcriptome sequencing caused by random hexamer priming. *Nucleic acids research*, 38(12):e131–e131, 2010.

[52] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins. Computational studies of gene regulatory networks: in numero molecular biology. *Nature Reviews Genetics*, 2(4):268–279, 2001.

[53] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[54] G. Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

[55] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[56] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[57] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[58] F. Hsu, W. Kent, H. Clawson, R. Kuhn, M. Diekhans, and D. Haussler. The ucsc known genes. *Bioinformatics*, 22(9):1036–1046, 2006.

[59] T. J. Hudson, W. Anderson, A. Aretz, A. D. Barker, C. Bell, R. R. Bernabé, M. Bhan, F. Calvo, I. Eerola, D. S. Gerhard, et al. International network of cancer genome projects. *Nature*, 464(7291):993–998, 2010.

[60] J. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.

[61] W. Jiao, S. Vembu, A. G. Deshwar, L. Stein, and Q. Morris. Inferring clonal evolution of tumors from single nucleotide somatic mutations. *BMC Bioinformatics*, 15(1):35, 2014.

[62] D. Karolchik, A. S. Hinrichs, T. S. Furey, K. M. Roskin, C. W. Sugnet, D. Haussler, and W. J. Kent. The ucsc table browser data retrieval tool. *Nucleic acids research*, 32(suppl 1):D493–D496, 2004.

[63] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[64] J. Khatun. An integrated encyclopedia of dna elements in the human genome. *Nature*, 2012.

[65] S. Kim. Protein $\beta$-turn prediction using nearest-neighbor method. *Bioinformatics*, 20(1):40–44, 2004.

[66] D. C. Koboldt, Q. Zhang, D. E. Larson, D. Shen, M. D. McLellan, L. Lin, C. A. Miller, E. R. Mardis, L. Ding, and R. K. Wilson. Varscan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome research*, 22(3):568–576, 2012.

[67] R. Kohavi and F. Provost. Glossary of terms. *Machine Learning*, 30(2-3):271–274, 1998.

[68] A. Kozomara and S. Griffiths-Jones. mirbase: annotating high confidence micrornas using deep sequencing data. *Nucleic acids research*, 42(D1):D68–D73, 2014.

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[70] J. Lamb, E. D. Crawford, D. Peck, J. W. Modell, I. C. Blat, M. J. Wrobel, J. Lerner, J.-P. Brunet, A. Subramanian, K. N. Ross, et al. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *science*, 313(5795):1929–1935, 2006.

[71] D. A. Landau, S. L. Carter, P. Stojanov, A. McKenna, K. Stevenson, M. S. Lawrence, C. Sougnez, C. Stewart, A. Sivachenko, L. Wang, et al. Evolution and impact of subclonal mutations in chronic lymphocytic leukemia. *Cell*, 152(4):714–726, 2013.

[72] E. S. Lander and M. S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2(3):231–239, 1988.

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

[74] T. Lappalainen, M. Sammeth, M. R. Friedländer, P. ACt Hoen, J. Monlong, M. A. Rivas, M. Gonzàlez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira, et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511, 2013.

[75] N. B. Larson and B. L. Fridley. Purbayes: estimating tumor cellularity and subclonality in next-generation sequencing data. *Bioinformatics*, 29(15):1888–1889, 2013.

[76] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[77] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129, 2014.

[78] B. Li, V. Ruotti, R. Stewart, J. Thomson, and C. Dewey. Rna-seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500, 2010.

[79] J. Li, H. Jiang, and W. Wong. Method modeling non-uniformity in short-read rates in rna-seq data. *Genome Biol*, 11(5):R25, 2010.

[80] Y. Li and X. Xie. A mixture model for expression deconvolution from rna-seq in heterogeneous tissues. *BMC bioinformatics*, 14(5):1, 2013.

[81] Y. Li and X. Xie. Deconvolving tumor purity and ploidy by integrating copy number alterations and loss of heterozygosity. *Bioinformatics*, page btu174, 2014.

[82] Y. Li and X. Xie. Mixclone: a mixture model for inferring tumor subclonal populations. *BMC genomics*, 16(Suppl 2):S1, 2015.

[83] K. Lindblad-Toh, D. M. Tanenbaum, M. J. Daly, E. Winchester, W.-O. Lui, A. Villapakkam, S. E. Stanton, C. Larsson, T. J. Hudson, B. E. Johnson, et al. Loss-of-heterozygosity analysis of small-cell lung carcinomas using single-nucleotide polymorphism arrays. *Nature biotechnology*, 18(9):1001–1005, 2000.

[84] J. Lonsdale, J. Thomas, M. Salvatore, R. Phillips, E. Lo, S. Shad, R. Hasz, G. Walters, F. Garcia, N. Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580–585, 2013.

[85] S. Marguerat and J. Bähler. Rna-seq: from technology to biology. *Cellular and Molecular Life Sciences*, 67(4):569–579, 2010.

[86] C. Mathé, M.-F. Sagot, T. Schiex, and P. Rouzé. Current methods of gene prediction, their strengths and weaknesses. *Nucleic acids research*, 30(19):4103–4117, 2002.

[87] M. Mayrhofer, S. DiLorenzo, and A. Isaksson. Patchwork: allele-specific copy number analysis of whole genome sequenced tumor tissue. *Genome biology*, 14(3):R24, 2013.

[88] A. Medina-Rivera, M. Defrance, O. Sand, C. Herrmann, J. A. Castro-Mondragon, J. Delerce, S. Jaeger, C. Blanchet, P. Vincens, C. Caron, et al. Rsat 2015: regulatory sequence analysis tools. *Nucleic acids research*, page gkv362, 2015.

[89] R. Mei, P. C. Galipeau, C. Prass, A. Berno, G. Ghandour, N. Patil, R. K. Wolff, M. S. Chee, B. J. Reid, and D. J. Lockhart. Genome-wide detection of allelic imbalance using human snps and high-density dna arrays. *Genome Research*, 10(8):1126–1137, 2000.

[90] M. Meyerson, S. Gabriel, and G. Getz. Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews Genetics*, 11(10):685–696, 2010.

[91] M. MLL and P. PIK3R1. Comprehensive molecular portraits of human breast tumours. 2012.

[92] A. Mortazavi, B. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628, 2008.

[93] N. Navin, J. Kendall, J. Troge, P. Andrews, L. Rodgers, J. McIndoo, K. Cook, A. Stepansky, D. Levy, D. Esposito, et al. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90–94, 2011.

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

[95] L. Oesper, A. Mahmoody, and B. J. Raphael. Theta: Inferring intra-tumor heterogeneity from high-throughput dna sequencing data. *Genome biology*, 14(7):R80, 2013.

[96] A. B. Olshen, E. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, 5(4):557–572, 2004.

[97] Y. Otsuka, Y. Ichikawa, C. Kunisaki, G. Matsuda, H. Akiyama, M. Nomura, S. Togo, Y. Hayashizaki, and H. Shimada. Correlating purity by microdissection with gene expression in gastric cancer tissue. *Scandinavian Journal of Clinical & Laboratory Investigation*, 67(4):367–379, 2007.

[98] L. Pachter. Models for transcript quantification from rna-seq. *arXiv preprint arXiv:1104.3889*, 2011.

[99] B. L. Parsons. Many different tumor types have polyclonal tumor origin: evidence and implications. *Mutation Research/Reviews in Mutation Research*, 659(3):232–247, 2008.

[100] D. Peck, E. D. Crawford, K. N. Ross, K. Stegmaier, T. R. Golub, and J. Lamb. A method for high-throughput gene expression signature analysis. *Genome biology*, 7(7):1, 2006.

[101] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[102] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[103] D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W.-L. Kuo, C. Chen, Y. Zhai, et al. High resolution analysis of dna copy number variation using comparative genomic hybridization to microarrays. *Nature genetics*, 20(2):207–211, 1998.

[104] K. Pruitt, T. Tatusova, and D. Maglott. Ncbi reference sequences (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic acids research*, 35(suppl 1):D61–D65, 2007.

[105] K. D. Pruitt, G. R. Brown, S. M. Hiatt, F. Thibaud-Nissen, A. Astashyn, O. Ermolaeva, C. M. Farrell, J. Hart, M. J. Landrum, K. M. McGarvey, et al. Refseq: an update on mammalian reference sequences. *Nucleic acids research*, 42(D1):D756–D763, 2014.

[106] D. Quang, Y. Chen, and X. Xie. Dann: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics*, page btu703, 2014.

[107] D. Quang and X. Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, page gkw226, 2016.

[108] M. Rasmussen, M. Sundström, H. G. Kultima, J. Botling, P. Micke, H. Birgisson, B. Glimelius, A. Isaksson, et al. Allele-specific copy number analysis of tumor samples with aneuploidy and tumor heterogeneity. *Genome Biol*, 12(10):R108–R108, 2011.

[109] O. Reiersøl. Identifiability of a linear relation between variables which are subject to error. *Econometrica: Journal of the Econometric Society*, pages 375–389, 1950.

[110] S. Ren, Z. Peng, J. Mao, Y. Yu, C. Yin, X. Gao, Z. Cui, J. Zhang, K. Yi, W. Xu, et al. Rna-seq analysis of prostate cancer in the chinese population identifies recurrent gene fusions, cancer-associated long noncoding rnas and aberrant alternative splicings. *Cell Research*, 22(5):806–821, 2012.

[111] A. Roberts and L. Pachter. Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature Methods*, 2012.

[112] A. Roberts, C. Trapnell, J. Donaghey, J. Rinn, L. Pachter, et al. Improving rna-seq expression estimates by correcting for fragment bias. *Genome Biol*, 12(3):R22, 2011.

[113] N. D. Roberts, R. D. Kortschak, W. T. Parker, A. W. Schreiber, S. Branford, H. S. Scott, G. Glonek, and D. L. Adelson. A comparative analysis of algorithms for somatic snv detection in cancer. *Bioinformatics*, 29(18):2223–2230, 2013.

[114] A. Roth, J. Ding, R. Morin, A. Crisan, G. Ha, R. Giuliany, A. Bashashati, M. Hirst, G. Turashvili, A. 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.

[115] A. Roth, J. Khattra, D. Yap, A. Wan, E. Laks, J. Biele, G. Ha, S. Aparicio, A. Bouchard-Côté, and S. P. Shah. Pyclone: statistical inference of clonal population structure in cancer. *Nature methods*, 11(4):396–398, 2014.

[116] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1988.

[117] R. Sachidanandam, D. Weissman, S. C. Schmidt, J. M. Kakol, L. D. Stein, G. Marth, S. Sherry, J. C. Mullikin, B. J. Mortimore, D. L. Willey, et al. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409(6822):928–933, 2001.

[118] M. Sammeth. The flux simulator. http://sammeth.net/confluence/display/sim/home. 2012.

[119] J. F. Sathirapongsasuti, H. Lee, B. A. Horst, G. Brunner, A. J. Cochran, S. Binder, J. Quackenbush, and S. F. Nelson. Exome sequencing-based copy-number variation and loss of heterozygosity detection: Exomecnv. *Bioinformatics*, 27(19):2648–2654, 2011.

[120] S. T. Sherry, M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. dbsnp: the ncbi database of genetic variation. *Nucleic acids research*, 29(1):308–311, 2001.

[121] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[122] A. Siepel, G. Bejerano, J. S. Pedersen, A. S. Hinrichs, M. Hou, K. Rosenbloom, H. Clawson, J. Spieth, L. W. Hillier, S. Richards, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome research*, 15(8):1034–1050, 2005.

[123] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[124] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[125] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011.

[126] T. Sørlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. Van De Rijn, S. S. Jeffrey, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19):10869–10874, 2001.

[127] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[128] X. Su, L. Zhang, J. Zhang, F. Meric-Bernstam, and J. N. Weinstein. Purityest: estimating purity of human tumor samples using next-generation sequencing data. *Bioinformatics*, 28(17):2265–2266, 2012.

[129] W. Sun, F. A. Wright, Z. Tang, S. H. Nordgard, P. Van Loo, T. Yu, V. N. Kristensen, and C. M. Perou. Integrated study of copy number states and genotype calls using high-density snp arrays. *Nucleic acids research*, 37(16):5365–5377, 2009.

[130] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1139–1147, 2013.

[131] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[132] H. Theil. Economic forecasts and policy. 1958.

[133] C. Trapnell, B. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. Van Baren, S. Salzberg, B. Wold, and L. Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515, 2010.

[134] P. Van Loo, S. H. Nordgard, O. C. Lingjærde, H. G. Russnes, I. H. Rye, W. Sun, V. J. Weigman, P. Marynen, A. Zetterberg, B. Naume, et al. Allele-specific copy number analysis of tumors. *Proceedings of the National Academy of Sciences*, 107(39):16910–16915, 2010.

[135] D. Venet, F. Pecasse, C. Maenhaut, and H. Bersini. Separation of samples into their constituents using gene expression data. *Bioinformatics*, 17(suppl 1):S279–S287, 2001.

[136] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[137] C. Wang, B. Gong, P. R. Bushel, J. Thierry-Mieg, D. Thierry-Mieg, J. Xu, H. Fang, H. Hong, J. Shen, Z. Su, et al. The concordance between rna-seq and microarray data depends on chemical treatment and transcript abundance. *Nature biotechnology*, 32(9):926–932, 2014.

[138] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

[139] R. Xi, A. G. Hadjipanayis, L. J. Luquette, T.-M. Kim, E. Lee, J. Zhang, M. D. Johnson, D. M. Muzny, D. A. Wheeler, R. A. Gibbs, et al. Copy number variation detection in whole-genome sequencing data using the bayesian information criterion. *Proceedings of the National Academy of Sciences*, 108(46):E1128–E1136, 2011.

[140] X. Xie, J. Lu, E. Kulbokas, T. R. Golub, V. Mootha, K. Lindblad-Toh, E. S. Lander, and M. Kellis. Systematic discovery of regulatory motifs in human promoters and 3 utrs by comparison of several mammals. *Nature*, 434(7031):338–345, 2005.

[141] X. Xie, T. S. Mikkelsen, A. Gnirke, K. Lindblad-Toh, M. Kellis, and E. S. Lander. Systematic discovery of regulatory motifs in conserved regions of the human genome, including thousands of ctcf insulator sites. *Proceedings of the National Academy of Sciences*, 104(17):7145–7150, 2007.

[142] C. Yan, D. Dobbs, and V. Honavar. A two-stage classifier for identification of protein–protein interface residues. *Bioinformatics*, 20(suppl 1):i371–i378, 2004.

[143] C. Yau. Oncosnp-seq: a statistical approach for the identification of somatic copy number alterations from next-generation sequencing of cancer genomes. *Bioinformatics*, 29(19):2482–2484, 2013.

[144] C. Yau, D. Mouradov, R. N. Jorissen, S. Colella, G. Mirza, G. Steers, A. Harris, J. Ragoussis, O. Sieber, C. C. Holmes, et al. A statistical approach for detecting genomic aberrations in heterogeneous tumor samples from single nucleotide polymorphism genotyping data. *Genome Biol*, 11(9):R92–R92, 2010.

[145] G. Ye, M. Tang, J.-F. Cai, Q. Nie, and X. Xie. Low-rank regularization for learning gene expression programs. *PloS one*, 8(12):e82146, 2013.

[146] G. Yu, B. Zhang, G. Bova, J. Xu, Y. Wang, et al. Bacom: in silico detection of genomic deletion types and correction of normal cell contamination in copy number data. *Bioinformatics*, 27(11):1473–1480, 2011.

[147] Y. Yuan, H. Failmezger, O. M. Rueda, H. R. Ali, S. Gräf, S.-F. Chin, R. F. Schwarz, C. Curtis, M. J. Dunning, H. Bardwell, et al. Quantitative image analysis of cellular heterogeneity in breast tumors complements genomic profiling. *Science translational medicine*, 4(157):157ra143–157ra143, 2012.

[148] Z. J. Zang, I. Cutcutache, S. L. Poon, S. L. Zhang, J. R. McPherson, J. Tao, V. Rajasegaran, H. L. Heng, N. Deng, A. Gan, et al. Exome sequencing of gastric adenocarcinoma identifies recurrent somatic mutations in cell adhesion and chromatin remodeling genes. *Nature genetics*, 44(5):570–574, 2012.

[149] X. Zhao, C. Li, J. G. Paez, K. Chin, P. A. Jänne, T.-H. Chen, L. Girard, J. Minna, D. Christiani, C. Leo, et al. An integrated view of copy number and allelic alterations in the cancer genome using single nucleotide polymorphism arrays. *Cancer research*, 64(9):3060–3071, 2004.

[150] J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature methods*, 12(10):931–934, 2015.

[151] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807, 2000.