Genome-Driven Personalized Medicine of Cancer via Machine Learning and Phylogenetic Models

Yifeng Tao

CMU-CB-21-103
August 2021

Computational Biology Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Russell Schwartz (Chair; Carnegie Mellon University)
Jian Ma (Carnegie Mellon University)
Xinghua Lu (University of Pittsburgh)
Adrian V. Lee (University of Pittsburgh)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Research reported in this thesis was supported by the National Human Genome Research Institute of the National Institutes of Health under award number R01HG010589, the National Library of Medicine of the National Institutes of Health under award number R01LM012011, the Pennsylvania Department of Health Commonwealth Universal Research Enhancement (CURE) program under award number 4100059192, the Center for Machine Learning and Health (CMLH) research funding, and the Mario Lemieux Foundation. Y. Tao also received support through a Center for Machine Learning and Health (CMLH) Fellowship.

The department specifically disclaims responsibility for any analyses, interpretations, or conclusions. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, donor or the U.S. Government.

Keywords: Interpretable Machine Learning, Deep Learning, Attention Mechanism, Cancer Genomics, Tumor Heterogeneity, Tumor Phylogenetics, Evolutionary Methods, Cancer Prognosis, Drug Response

To those who pave the way of cancer research and treatment

Abstract

Cancer proceeds from the accumulation of genomic alterations, and develops into heterogeneous cell populations in an evolutionary process. Therefore, the prognoses of cancer patients, such as survival profile, metastasis, and drug response, are encoded by the large-volume genome data. We first investigate the reliable phenotype inference of cancer through well-designed interpretable machine learning models. By leveraging the power of large-scale genomic data and external biomedical knowledge base, we utilize deep learning models for the accurate inference of cancer phenotypes, including transcriptome expression levels, transcription factor activities, and drug resistance. We address the interpretability of models through techniques such as attention mechanisms to identify driver mutations and critical biomarkers. Secondly, we reveal the intra-/inter-tumor heterogeneity and mechanism of tumor progression via robust deconvolution and phylogenetic algorithms. We formulate the deconvolution of bulk tumor molecular data mathematically as a biologically inspired matrix factorization problem, and propose a neural network and then an improved hybrid optimizer to solve the problem robustly and accurately. We develop and apply a Minimum Elastic Potential algorithm to reconstruct the evolutionary trajectory from the unmixed clones. Finally, we improve the prognostic prediction of cancer by incorporating machine learning and evolutionary methods. Clinicians traditionally focused on the pathological features and driver-level genomic profiles to facilitate the treatment. However, it is possible that critical clones, instead of the bulk tumor as a whole, affect the prognoses. We explore the questions by integrating both the evolutionary mutational features, driver-level features, and clinical features to improve the prognostic prediction of cancer. We develop an L0-regularized Cox regression model, and find that the evolutionary features account for roughly 1/3 of all the available features, depending on cancer types and sequencing techniques.

Acknowledgments

First and foremost, I would like to thank my advisor Russell Schwartz for introducing me to the exciting research area of tumor evolution. He granted me extensive freedom to conduct interesting research, while his knowledge and insights prevented me from going astray too many times.

I also would like to thank other thesis committee members for their close collaborations over years. I was fortunate to get a lot of guidance on machine learning in cancer genomics from Jian Ma in the prognosis prediction project. Adrian V. Lee was generous to share with us the private dataset and enlightened me new approach from the clinical aspects in the breast cancer metastasis project. I was grateful to be brought into the field of cancer genomics by Xinghua Lu and worked together on the phenotype inference project.

The presented work in this thesis would not be possible without our awesome and collaborative members from the Schwartz Lab: Haoyun Lei, Xuecong Fu, Xiaoyue Cui, Ziyi Cui, Jesse Eaton, Hannah Kim, Haoran Chen, and Yuanqi Zhao. I treasure the time working with brilliant students Rishi Verma, Alex Guo, and Alyssa Lee. I received helpful suggestions from Marcus Thomas and Arjun Srivatsa.

I was so lucky to work with a lot of fantastic collaborators in the past five years, including members from Jian Ma's lab: Ashok Rajaraman, Yang Zhang, and Wendy Yang; Xinghua Lu's lab: Shuangxia Ren, Yifan Xue, Chunhui Cai, Michael Q. Ding, Xueer Chen, and Qiao Jin; Adrian V. Lee's lab: Kai Ding and Fangyuan (Chelsea) Chen; Hatice Ulku Osmanbeyoglu's lab: Xiaojun Ma and Drake Palmer; Eric P. Xing's Lab: Haohan Wang, Benjamin Lengerich, and Pengtao Xie. I appreciate my early mentor William W. Cohen, and mentors during summer internships, including Christopher Potts, Guillaume Genthial, and Kimberly Gietzen.

Finally, I would like to thank my family for their consistent support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction[1]

Thirty-two years separated the development of the first DNA sequencing technology using primer extension in 1971 and the sequencing of all 3 billion nucleotides in the human genome as part of the Human Genome Project in 2001 [99, 238], completed at an estimated cost of 2.7 billion US dollars. Less than 20 years later, the estimated cost of whole-genome sequencing (WGS) crossed the milestone of $<$\$1000 per genome [157], with some startups offering direct-to-consumer WGS for \$299 in 2020 (`https://nebula.org/whole-genome-sequencing/`). The rate of decline in cost has been driven by tremendous advances in next-generation sequencing (NGS) technologies, some of which have been repurposed to interrogate genome-wide gene expression (transcriptomics), methylation and chromatin status (epigenomics), splice variants, and more [118]. The increased availability of sequencing technologies has led to a plethora of multiomic data in clinical research. The complexity of these data is compounded by extremely high heterogeneity both within [149] and between [180] tumors requiring advanced computational analysis and data mining tools to identify reproducible and clinically actionable patterns in the presence of multiple hypothesis testing on the order of 106-109 per phenotype. This challenge is being addressed by bringing together biological and clinical expertise with that from the computational sciences. Advances in machine learning (ML) have shown particular promise to address many of the limitations of more conventional statistical analyses by facilitating the identification of sparse signals in large, noisy data and predicting outcomes potentially with no a priori hypotheses or assumptions about data distribution [126].

Oncology has thus far been at the forefront of the genomic revolution, with large-scale projects identifying both germline and somatic variants and transcriptomic signatures to predict cancer risk [233], subtype histologically similar but clinically distinct cancers [51, 167, 175], predict tumor response to therapy [207], identify driving mutations [120] and pathways [234, 239], and nominate novel therapeutic targets [78, 172]. This effort has been facilitated by large-scale community efforts at data generation, including The Cancer Genome Atlas [29] and the International Cancer Genome Consortium [98] among others. Aside from tumor classification and characteristics, oncology has also been a driving force in the field of pharmacogenomics [69, 188], aimed at explaining and exploiting inter-patient variability in drug response by interrogating, for

[1] This chapter was developed from the unpublished material "Omar El-Charif, Russell Schwartz, Yifeng Tao, and Ye Yuan. Machine Learning Applications in Cancer Genomics. *Machine Learning in Clinical Radiation Oncology: A Guide for Clinicians*, edited by Barry Rosenstein, Tim Rattay, John Kang".

example, variants influencing the function of enzymes and transporters in pharmacokinetic pathways. Around 40% of the 400 drug-gene pairs on FDA labels are for oncological drugs, most of which were discovered using candidate gene approaches and conventional statistical methods prior to genome-wide interrogation and ML techniques. Furthermore, most are rarely used routinely in clinical settings, highlighting what is arguably the most significant challenge facing the field — translating cancer research into clinical implementation.

The promise of integrating ML with large scale multi-omic databases is only starting to come to fruition as costs of data collection decline and the baton is passed to data analytics. Today, the cost of sequencing, at least of bulk tumors, is no longer a significant obstacle to making precision genomic medicine a routine part of cancer treatment, but the computational tools and expertise to make use of these data in the clinic are still lacking. In this chapter, we will briefly review some current applications and successful examples of clinical biomarkers. We will also discuss some of the hurdles that remain in data-driven decisions of cancer genomics. Finally we will introduce our contributions to this field and go over the organization of the thesis.

1.1 Application of machine learning in cancer genomics

Personalized oncology Somatic variants are playing an ever-growing role in guiding cancer management, due to more granular tumor characterization beyond TNM staging and even chromosomal analysis, but also due to the success of mutation-targeting therapies. Among the earliest prototypes of targeted therapies is anti-EGFR drug gefitinib [7, 138, 150, 253]. More recently, Big Data initiatives have introduced a new paradigm, shifting oncological research towards high throughput, promising to create cheaper and more productive methods to reach personalized oncology. One such example is The Cancer Genome Atlas (TCGA), which characterized genomic and transcriptomic profiles for thousands of tumors along with clinical data and pathology reports [29, 95]. With millions of data points in each sample, the need to reduce dimensionality, identify relevant signals, and build understanding beyond single genes and pathways emerged. ML has been utilized for this purpose at various stages in the TCGA research pipeline [51, 77, 144]. The thesis work made use of deep learning approaches such as transfer learning, transformer, and attention mechanism to compress the large-scale multi-omic profiles of tumors into compact tumor embedding latent space. We showed the tumor embeddings predictive of phenotypes such as transcriptome expression [221] and drug resistance [224]. In addition, we introduced the evolutionary features, which reflect the tumor heterogeneity to further improve the tumor prognostic prediction [225].

Driver mutation discovery Although a single tumor may contain hundreds to thousands of somatic alterations, it is a general consensus that not all these mutations contribute equally to the initialization and progression of the tumor. Only a small amount of the somatic alterations happened at the beginning or critical time points of during tumor development and contribute to the tumor progression and selective advantages over the neighboring cell clones. These are named “driver” mutations/genes [76]. Meanwhile, “passenger” mutations happen along with the driver mutations but do not contribute to the tumor development. Although initial research focused on finding “driver genes” [82, 208], more recent research focused on the nonsynonymous “driver mutations". Researchers identify cancer driver mutations through various strategies: based on mutation frequency [55, 119], based on mutation location [30, 79, 168, 190], through causal inference [27, 244], or through ensembled methods of multiple tools [11, 80]. The thesis work built up a model based on the causal inference between somatic genomic alterations and differentially expressed genes [221]. It made use of self-attention mechanism to capture the contexutal effect of somatic mutations and identify crucial driver genes that have large impact to the downstream phenotypes.

Biomarkers of outcome in clinical practice Treatment strategies in modern oncology have evolved from one size fits all regimens to complex multimodality and increasingly personalized therapies. In the decades since, we have developed a better understanding of the biological pathways involved in cancer progression leading to the integration of histopathologic and molecular biomarkers in clinical risk-stratification. From a clinician's viewpoint, there is a pressing need for better prognostic, or better yet, predictive/prescriptive biomarkers that can help patients and clinicians make more personalized treatment decisions. Several validated genomic biomarkers, including Oncotype DX [63], DCISionRT [23], MammaPrint [233], and PAM50 [175] are currently being used in the clinic as decision support tools. The thesis work utilized attention mechanism and feature selection to identify the most crucial biomarkers of genes and evolutionary features that are indicative of clinical outcome [224, 225]. In addition, we identified the commonly perturbed pathways in breast cancer metastasis, providing the potential of early interventions in cancer treatment [222, 223].

Pharmacogenomics How to recommend existing or discover novel anti-cancer drugs with the facilitation of genomic profiles is a critical task in chemotherapy and pharmacogenomics for the precision medicine and personalized treatment of cancer patients [154]. Since the molecular mechanisms of patients with the same cancer type can be distinct, drugs effective in one patient may be ineffective in others [69, 188]. The extensive screening assays of cancer cell lines have made it possible to test the drug resistance of cancer cells to a panel of potential anti-cancer drugs. There exist a few popular public cancer cell line drug sensitivity datasets, e.g., NCI-60 [206], CCLE [12], and GDSC [257]. The assays conducted on cancer cell lines , however, do not provide additional MoAs of other potential drugs or molecules for cancer treatment. Computational researchers proposed a few solutions for this, including drug discovery [154] and drug repurposing/repositioning [270]. Another drawback with the cell lines data is the huge gap between the *in vitro* cell lines and *in vivo* real tumors, which consist of mixed cancer cell populations and the infiltrated healthy cells [134, 202]. The thesis work addressed the noisy and missing data of drug response data with collaborative filtering, and captured the interactions between genes and drugs using contextual attention [224].

1.2 Common hurdles of machine learning in genomics

ML can be a powerful and versatile set of tools, as the preceding discussion demonstrates. When bringing them to new applications or data sets, though, they can be easily applied in ways that are unsound or less than optimal. This section considers some of the common challenges to applying

ML successfully in genomics work, particularly for cancer genomics, and some strategies by which they may be approached.

1.2.1 Challenges in data acquisition

Obtaining sufficient quantities and quality of data plays an essential role in successfully applying ML in any context, and cancer genomics is no exception. With advances in sequencing techniques and the explosion of the scale of genomic data, a few recurring issues related to data acquisition and sharing have emerged. These include challenges in the availability of datasets, cleaning and summarizing raw datasets, and performing ML analysis while assuring the privacy and security of potentially sensitive genomic data.

There have been several unified efforts internationally in collecting comprehensive genomic and pathological data of cancer samples in the past years, such as The Cancer Genome Atlas (TCGA) and International Cancer Genome Consortium (ICGC) noted previously. In addition to the original datasets, which can contain petabytes of genomic data of various types [95], researchers and clinicians may also rely on summary statistics or aggregated results such as information on inferred driver mutations or specific subtypes of cancer data. Efforts to provide such data resources therefore typically involve considerable downstream processing and analysis to supplement them with informative derived data, associate them with relevant metadata, and make the raw data easier for the general research community to use. Table 1.1 provides a few examples in the form of a list, not meant to be comprehensive, of sources of cancer genomic data and associated derived data and metadata, most of which are accessible through easy to use web user interface.

1.2.2 Data sparsity

Although there has been a boom in genomic analysis of cancer cohorts in recent years, the public cancer cohorts usually are limited to tens of thousands of patients, or, for private ones, up to hundreds of thousands [34]. For more specialized kinds of questions, cohort sizes may range from just tens of samples to thousands of samples. In contrast, the ImageNet dataset in computer vision contains more than 14 million images [56] while the Yelp reviews dataset used in natural language processing (NLP) studies contains more than 5 million reviews [148]. At the same time, genomic data normally have high dimensional feature sets. For human beings, we have more than 20,000 genes and 324,000,000 known variants in total. The limited samples, high dimensions, and large noise characterizing the genomic data can lead to fragile machine learning models, e.g., overfitting and lack of interpretability. Researchers have tackled high dimensional data with various approaches, including feature selection and feature transformation.

Small sample sizes relative to the feature set present a challenge for almost any kind of ML. In some cases, dealing with small sample sizes may mean favoring different ML methods that are less data intensive, e.g., using support vector machines (SVMs) instead of popular deep learning approaches [25, 85]. Extra attention to protecting models from overfitting is also warranted. ML offers a variety of common methods for protecting a model from overfitting during learning, including Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), or other

Table 1.1: Examples of important cancer genomic data sets.

Databases	Pan-cancer	Tumor or cell line	URL	Comment
TCGA/GDC [101] / dbGaP [231]	Y	Tumor	https://portal.gdc.cancer.gov/	TCGA data were hosted through dbGaP before 2016, but they are now hosted through GDC.
ICGC [96]	Y	Tumor	https://icgc.org/	
EGA [117]	Y	Tumor	https://ega-archive.org/	
METABRIC [179]	N (Breast cancer)	Tumor	https://www.cbioportal.org/study/summary?id=brca_metabric	Large scale breast cancer dataset.
NCI-60 [206]	Y	Cell line	https://dtp.cancer.gov/discovery_development/nci-60/	Drug sensitivity data.
CCLE [12]	Y	Cell line	https://portals.broadinstitute.org/ccle	Drug sensitivity data.
GDSC [257]	Y	Cell line	https://www.cancerrxgene.org/	Drug sensitivity data.
CancerSEA [262]	Y	Tumor	http://biocc.hrbmu.edu.cn/CancerSEA/home.jsp	Single-cell data.

regularization strategies that bias toward simple models [114]. It is also important to validate robustness to data subsampling and to independent data sets post-hoc. The use of prior knowledge to bias a model to reflect expected outcomes can also help mitigate the effects of limited training data.

Another way to mitigate the model complexity is leveraging the effectiveness of other related data available, e.g., through transfer learning [249], multitask learning [31], and semi-supervised learning [86]. These methods utilize knowledge from other applications, with the hypothesis that the data entails specific structure and can improve performance when the number of samples is limited by capturing that structure.

One particular broadly important class of methods for dealing with data sparsity is dimensionality reduction, which refers to a set of strategies for shrinking the set of features from which we seek to learn. One simple version of data sparsity is straightforward: since a lot of features are redundant in the dataset, we can just select a subset of essential features that are important to the task. A few major categories of methods have been developed for this purpose [195], including filter [254], wrapper [111], and embedded methods [230]. Taking the cancer type classification task through microarray expression profiles of tumors as an example [271], the training process of a machine learning model is equivalent to optimizing an objective function. In the case of wrapper methods, we select the subset of genes that can achieve the best performance on the validation dataset. In practice, it is computationally infeasible to find the optimal subset of around 20,000 features, but many heuristic algorithms have been proposed to select suboptimal solutions, e.g., stepwise forward selection. In embedded methods, however, we add additional regularization terms on the model parameters to the original objective/loss function. A widely used version of this is the L1-regularization of the parameters, or Lasso [230]. The L1 regularization is equivalent to having a Laplacian prior to the model parameters, therefore enforcing a sparse solution, where most of the coefficients are zeros, i.e., not selected. Wrapper methods are in general computationally expensive and prone to overfit. Therefore it is necessary to have a proper split of the dataset during tuning and evaluation, e.g., nested cross-validation [32]. Embedded methods are faster and easier to implement in practices [229].

Apart from the data-driven dimension reduction methods, computational biologists also incorporate biological database for dimension reduction through a knowledge-driven approach. In cancer genomics, a common approach is to reduce the gene level expressions into pathway-level expressions [64, 174, 223]. A pathway is defined as a set of closely related genes that are adjacent in the genome, or participated in the same or similar molecular processes, and therefore are likely to coexpress. A few knowledge bases can be utilized, e.g., the DAVID database [94], KEGG pathway database [105], and Gene Ontology [152]. This kind of knowledge-driven dimension reduction method can especially be effective when the samples are limited.

1.2.3 Inter-tumor heterogeneity

Another particular challenge of cancer genomic data in machine learning is frequently high inter-tumor heterogeneity. Depending on the tumor type, there may be a number of subtypes with substantially different molecular mechanisms, confounding many forms of genomic analysis [174]. Where a subtyping is well defined and understood, pre-partitioning data by subtype [91] may avoid some of the challenges of confounding data at the cost of reducing cohort sizes and making it hard to infer cross-subtype effects. Even if a subtyping is not fully understood, unsupervised clustering approaches (e.g., *k*-means clustering) can be used as a preprocessing step to partition data into more homogeneous subgroups before further analysis [133]. Subtyping remains an area of active study, however, and meaningful subtypes continue to be discovered and refined. Furthermore, even within a defined subtype, most cancers commonly exhibit somatic hypermutability [119], leading typically to large amounts of idiosyncratic, functionally irrelevant passenger mutations that can expand data dimensionality, further confounding analyses and limiting the effectiveness of simple strategies for dimensionality reduction. General strategies for dealing with data sparsity and large feature sets discussed above may mitigate such problems. Specialized methods, such as training models on mutation burdens at the gene or pathway level rather than individual variants, can also reduce, but not eliminate, these challenges.

1.2.4 Intra-tumor heterogeneity

Another recurring challenge of cancer genomic data is intra-tumor heterogeneity (ITH), i.e., cell-to-cell variability within single tumors [100, 215]. ITH is a common feature of cancers arising from the process of clonal evolution within a tumor and from the somatic hypermutability processes frequently active in tumors during this process (Loeb 1991). As a result, cancer genomic data often must be interpreted as mixtures of cell types (clones) that may exhibit different mutations, epigenetic markers, or patterns of gene activity. This clonal heterogeneity is further exacerbated by contributions from infiltrating immune cells or other stromal contamination [274]. ITH is a confounding factor for many common ML analyses, as genetic or genomic signals that underlie tumor function are obscured by clones that lack those signals. The problem is particularly vexing for prognostic prediction because progression processes in cancers, such as metastasis or the development of drug resistance, frequently proceed from relatively rare clones within a tumor [88] and thus prediction based on the dominant genomic features of a tumor may poorly predict the behavior of the tumor as a whole. While ITH is problematic for ML in cancer genomics, it can be dealt with by computational or experimental approaches. Computational strategies for working with ITH typically involve the use of genomic deconvolution, a strategy for computationally separating mixed genomic signals to infer likely signals of specific clones within a tumor that can then be examined separately during machine learning [237]. While such approaches were initially developed specifically for resolving tumor impurity, by separating tumor and stromal contributions, the idea was later extended to resolve distinct clones within single tumors. Numerous deconvolution methods exist today, some relying on multiple samples from a single tumor to resolve clonal mixtures and others on comparison across tumors to resolve common features of progression across a cohort. Clonal deconvolution is often combined with tumor phylogenetics [199], i.e., inference of trees describing the evolution of clonal states in a tumor, to better resolve substructure [15, 200].

As single-cell genomics has become more practical and widespread, it has increasingly displaced deconvolution methods for resolving clonal mixtures computationally from bulk sequencing data. The peculiar error characteristics of different single-cell technologies — which may include high error rates, high rates of allelic dropout or other missing data, and aberrations such as doublet sequences — create distinct challenges for ML from single-cell data that generally must be resolved to adapt ML analysis methods from bulk to single-cell data [214]. A handful of methods now exist as well for combining bulk and single-cell data in common analyses to achieve some advantages of each method type [74, 125, 142, 143].

1.2.5 Other common data issues

Another issue that can plague many genomic applications is imbalanced data. Imbalanced data refers to datasets that are skewed to some possible outcomes over others. An example would be the challenge of predicting a rare complication or atypical progression outcome [62], for example predicting those patients with poor outcomes in cancers that are rarely fatal, because there are few examples from which to train a model. General methods for dealing with data sparsity — such as reliance on prior knowledge, model regularization, or strategies such as transfer learning — may also help address challenges of imbalanced data. In addition, specialized strategies including the upsampling of minor categories via the generation of artificial "decoy" data [21], or the downsampling of major categories can mitigate the problem of imbalance. When coping with imbalanced data, it is crucial to choose proper evaluation metrics. Common intuitive assessments such as accuracy and the receiver operating characteristic (ROC) curve do not describe the minor class properly. Instead, other measures such as the F1 score and precision-recall curve are in general better choices in these cases [50].

Missing or inconsistent annotation of data are likewise common problems of cancer genomic data [53]. While data quality has generally improved across genomic technologies over time, noisy assays and complex biology can lead to missing fields in subsets of data points, posing yet another problem for ML. This may be particularly a concern for clinical data, where standards for expert annotation may still be less precise and consistent than is ideal for automated inference. Furthermore, large consortium efforts, for all their value for the community, can introduce problems of standardization across partner sites. While changes in practice concurrent with the broader adoption of electronic health records (EHRs) may help, ML methods must be able to cope with all of these issues to make use of them. Cleaning data of poorly annotated data points or data fields can resolve some such problems [132]. ML may also rely on imputation, i.e., using simpler learning models or other heuristics to infer likely values for missing data, or be engineered directly to allow for unknown or uncertain values in data [19].

This remains far from an exhaustive list about the challenges of genomic data and cancer genomic data in particular. It is intended, however, to highlight some of the common issues and provide an overview of typical ML strategies for dealing with these or similar problems. When other difficulties arise, some of the same strategies discussed above may prove useful in achieving good performance in less-than-ideal conditions for machine learning inference.

1.3 Our contributions

As discussed in the previous sections, the application of genomics in cancer is impeded by the unique hurdles of the sparse and heterogeneous tumor data. To address the noisy high dimensional data, we developed explainable machine learning models that make use of biological knowledge. To cope with the intra- and inter-tumor heterogeneity, we investigated deconvolution and phylogenetic algorithms to recover the evolutionary trajectory of tumors.

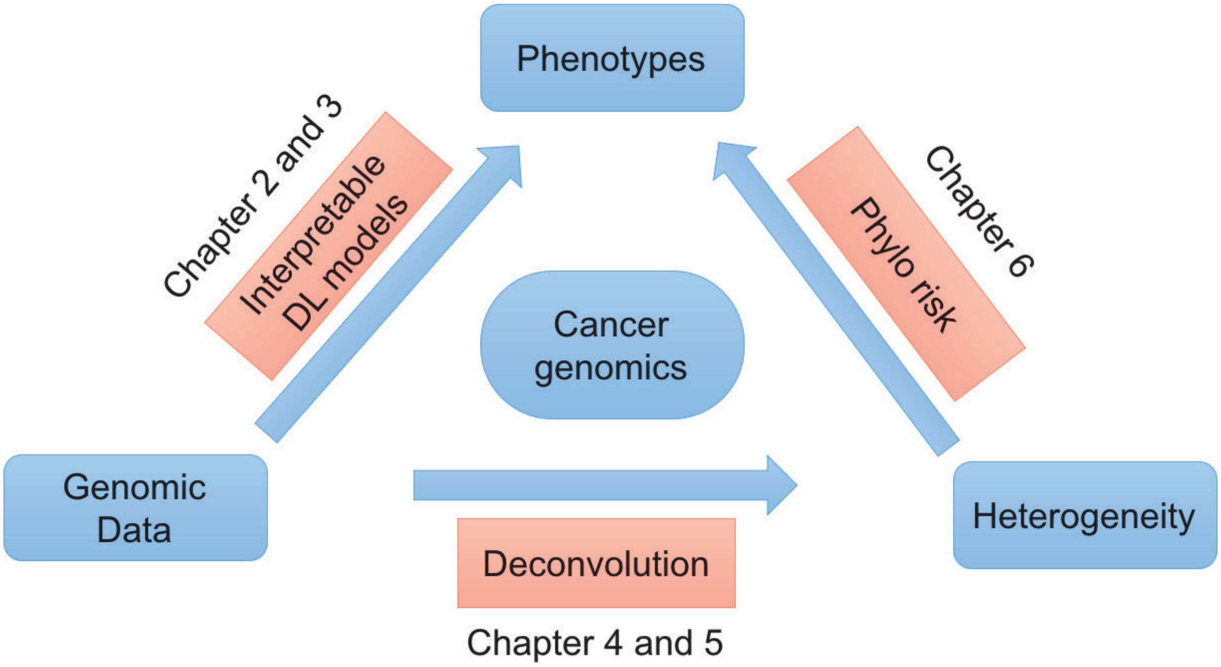

Figure 1.1: Thesis organization.

The thesis work aims to solve the crucial problems in personalized oncology, including driver mutation discovery, biomarker identification, drug resistance, clinical outcome prediction, and tumor heterogeneity inference in cancer genomics (Fig. 1.1), through state-of-the-art machine learning and phylogenetic methods:

Reliable phenotype inference of cancer through well-designed interpretable machine learning models By leveraging the power of large scale genomic data and external biomedical knowledge base, we have been working on deep learning models for the accurate inference of cancer phenotypes, including transcriptome expression levels (Genomic Impact Transformer; GIT) [221], transcription factor activities, and drug resistance (Contextual Attention-based Drug REsponse; CADRE) [224]. We addressed the interpretability of models through techniques such as attention mechanisms to identify driver mutations and critical biomarkers.

Revealing intra-/inter-tumor heterogeneity and mechanism of tumor progression via robust deconvolution and phylogenetic algorithms We formulated the deconvolution of bulk tumor molecular data mathematically as a biologically inspired matrix factorization problem, and proposed a neural network (Neural Network Deconvolution; NND) [223] and then an improved hybrid optimizer (Robust and Accurate Deconvolution; RAD) [222] to solve the problem robustly and accurately. We developed and applied a Minimum Elastic Potential (MEP) [219] algorithm to reconstruct the evolutionary trajectory from the unmixed clones.

Improving prognostic prediction of cancer by incorporating machine learning and evolutionary methods Clinicians traditionally focused on the pathological features and driver-level genomic profiles to facilitate the treatment. However, it is possible that critical clones, instead of the bulk tumor as a whole, affect the prognoses. We explored the questions by integrating both the evolutionary mutational features, driver-level features, and clinical features to improve the prognostic prediction of cancer. We developed an ℓ_0-regularized Cox regression model (Phylo-Risk) [225], and found that the evolutionary features account for roughly 1/3 of all the available features, depending on cancer types and sequencing techniques.

1.4 Thesis organization

The remaining chapters of the thesis will be organized in the following order:

Chapters 2 and 3 will describe the interpretable deep learning approach we take to for the accurate phenotype inference of cancer. Specifically, Chapter 2 will describe the inference of differentially expressed genes from somatic genomics alterations [221], while Chapter 3 will describe the inference of drug resistance using gene expression profiles [224].

Chapters 4 and 5 will describe the robust deconvolution and phylogenetic algorithms we develop for revealing inter- and intra-tumor heterogeneity. Chapter 4 will describe the first deconvolution algorithm (Neural Network Deconvolution) and phylogenetic algorithm (Minimum Elastic Entropy) for inferring tumor progression trajectory from bulk RNA-Seq data [219, 223]. Chapter 5 will describe an improved and accelerated version of the deconvolution algorithm (Robust and Accurate Deconvolution) [222].

Chapter 6 will describe the machine learning algorithms we investigate for evaluating the contribution of evolutionary features to prognostic prediction, and whether the evolutionary features improve the prognostic prediction [225].

Finally, Chapter 7 will summarize the findings of our work, and propose potential future directions on the topics of machine learning and phylogenetic modeling in cancer genomics.

Chapter 2

Inference of transcriptome expression levels in cancer through genomic impact transformer[1]

Cancer is mainly caused by the activation of oncogenes or deactivation of tumor suppressor genes (collectively called “driver genes”) as results of somatic genomic alterations (SGAs) [239], including somatic mutations (SMs) [104, 120], somatic copy number alterations (SCNAs) [45, 264], DNA structure variations (SVs) [213], and epigenetic changes [103]. Precision oncology relies on the capability of identifying and targeting tumor-specific aberrations resulting from driver SGAs and their effects on molecular and cellular phenotypes. However, our knowledge of driver SGAs and cancer pathways remains incomplete. Particularly, it remains a challenge to determine which SGAs (among often hundreds) in a specific tumor are drivers, which cellular signals or biological processes a driver SGA perturbs, and which molecular/cellular phenotypes a driver SGA affects.

Current methods for identifying driver genes mainly concentrate on identifying genes that are mutated at a frequency above expectation, based on the assumption that mutations in these genes may provide oncogenic advantages and thus are positively selected [55, 119]. Some works further focus on the mutations perturbing conserved (potentially functional) domains of proteins as indications they may be driver events [168, 190]. However, these methods do not provide any information regarding the functional impact of enriched mutations on molecular/cellular phenotypes of cells. Without the knowledge of functional impact, it is difficult to further determine whether an SGA will lead to specific molecular, cellular and clinical phenotypes, such as response to therapies. What’s more, while both SMs and SCNAs may activate/deactivate a driver gene, there is no well-established frequency-based method that combines different types of SGAs to determine their functional impact.

Conventionally, an SGA event perturbing a gene in a tumor is represented as a “one-hot” vector spanning gene space, in which the element corresponding to the perturbed gene is set to

[1] This chapter was developed from material published in “Yifeng Tao, Chunhui Cai, William W Cohen, and Xinghua Lu. From genome to phenome: Predicting multiple cancer phenotypes based on somatic genomic alterations via the genomicimpact transformer. In *Pacific Symposium on Biocomputing*, volume 25, pages 79-90. WorldScientific, 2020” [221].

"1". This representation simply indicates which gene is perturbed, but it does not reflect the functional impact of the SGA, nor can it represent the similarity of distinct SGAs that perturb a common signaling pathway. We conjecture that it is possible to represent an SGA as a low-dimensional vector, in the same manner as the "word embedding" [153, 178, 218] in the natural language processing (NLP) field, such that the representation reflects the functional impact of a gene on biological systems, and genes sharing similar functions should be closely located in such embedding space. Here the "similar function" is broadly defined, e.g., genes from the same pathway or of the same biological process [8]. Motivated by this, we propose a scheme for learning "gene embeddings" for SGA-affected genes, i.e., a mapping from individual genes to low-dimensional vectors of real numbers that are useful in multiple prediction tasks.

Based on the assumption that SGAs perturbing cellular signaling systems often eventually lead to changes in gene expression [27], we introduce an encoder-decoder architecture neural network model called "genomic impact transformer" (GIT) to predict DEGs and detect potential cancer drivers with the supervision of DEGs. While deep learning models are being increasingly used to model different bioinformatics problems [115, 121], to our knowledge there are few studies using the neural network to model the relationships between SGAs and molecular/cellular phenotypes in cancers. The proposed GIT model has the following innovative characteristics: 1) The encoder part of the transformer [236] first uses SGAs observed in a tumor as inputs, maps each SGA into a gene embedding representation, and combines gene embeddings of SGAs to derive a personalized "tumor embedding". Then the decoder part decodes and translates the tumor embedding to DEGs. 2) A multi-head self-attention mechanism [10, 255] is utilized in the encoder, which is a technique widely used in NLP to choose the input features that significantly influence the output. It differentiates SGAs by assigning different weights to them so that it can potentially distinguish SGAs that have an impact on DEG from those do not, i.e., detecting drivers from passengers. 3) Pooling inferred weighted impact of SGAs in a tumor produces a personalized tumor embedding, which can be used as an effective feature to predict DEGs and other phenotypes. 4) Gene embeddings are pre-trained by a "Gene2Vec" algorithm and further refined by the GIT, which captures the functional impact of SGAs on the cellular signaling system. Our results and analysis indicate that above innovative approaches enable us to derive powerful gene embedding and tumor embedding representations that are highly informative of molecular, cellular and clinical phenotypes.

2.1 Materials and methods

2.1.1 SGAs and DEGs pre-processing

We obtained SGA data, including SMs and SCNAs of 4,468 tumors consisting of 16 cancer types[2] directly from TCGA portal [163] and Firehose browser of the Broad Institute (`http://gdac.broadinstitute.org/`). For SMs: We considered all the non-synonymous mutation events of all genes and considered the mutation events at the gene level, where a mutated

[2] Instead of single cancer types, we used all the available samples of various cancer types, to find the common signaling mechanisms SGAs in cancer. In addition, the GIT model benefits from the large scale dataset. The heterogeneity of different cancer types was stratified by the additional cancer type feature as input to the model.

gene is defined as one that contains one or more non-synonymous mutations or indels. For SCNAs: TCGA network discretizes the gene SCNA into 5 different levels: homozygous deletion, single copy deletion, diploid normal copy, low copy number amplification, and high copy number amplification. We only included genes with homozygous deletion (potentially significant loss of gene function) or high copy number amplification (potentially significant gain of gene function) for further analysis, and filtered out the other three types of not-so-significant SCNAs. Therefore, we collectively designated all SGAs affecting a gene using the name of the gene being perturbed. Note that the preprocessing step of SMs and SCNAs excluded the obvious tumor passenger SGAs, since the functions of these mutated genes are not or only slightly perturbed. The remaining SGAs have the potential of being cancer drivers, such as oncogenes with gained functions, or tumor suppressor genes with lost functions. After processing genomic data from TCGA, we used a binary variable in a "one-hot" vector to indicate the genomic status of a gene. For example, we represented the genomic status of *TP53* as 1, if it is perturbed by one or more of SM/SCNA events in a tumor.

Gene expression data were pre-processed and obtained from the Firehose browser of the Broad Institute. We determined whether a gene is differentially expressed in a tumor by comparing the gene's expression in the tumor against a distribution of the expression values of the gene in the corresponding tissue-specific "normal" or control samples. For a given cancer type, assuming the expression of each gene (log-2-based) follows a Gaussian distribution in control sample, we calculated the *p*-values by determining the probability of observing an expression value from control distribution. Following the practice in previous work [27], if the *p*-value is equal or smaller than 0.005, the gene is considered as differentially expressed in the corresponding tumor. However, if a DEG is associated with an SCNA event affecting it, we remove it from the DEG list of the tumor.

2.1.2 The GIT neural network

GIT network structure: encoder-decoder architecture

Figure 2.1A shows the general structure of the GIT model with an overall encoder-decoder architecture. GIT mimics hierarchically organized cellular signaling system [37, 38], in which a neuron may potentially encode the signal of one or more signaling proteins. When a cellular signaling system is perturbed by SGAs, it often can lead to changes in measured molecular phenotypes, such as gene expression changes. Thus, for a tumor t, the set of its SGAs $\{g\}_{g=1}^{m}$ is connected to the GIT neural network as observed input (Fig. 2.1A bottom part squares). The impact of SGAs is represented as embedding vectors $\{\mathrm{e}_g\}_{g=1}^{m}$, which are further linearly combined to produce a tumor embedding vector e_t through an attention mechanism in the encoder (Fig. 2.1A middle part). We explicitly represent cancer type s and its influence on encoding system e_s of the tumor because tissue type influences which genes are expressed in cells of specific tissue as well. Finally, the decoder module, which consists of a feed-forward multi-layer perceptron (MLP) [193], transforms the functional impact of SGAs and cancer type into DEGs of the tumor (Fig. 2.1A top part).

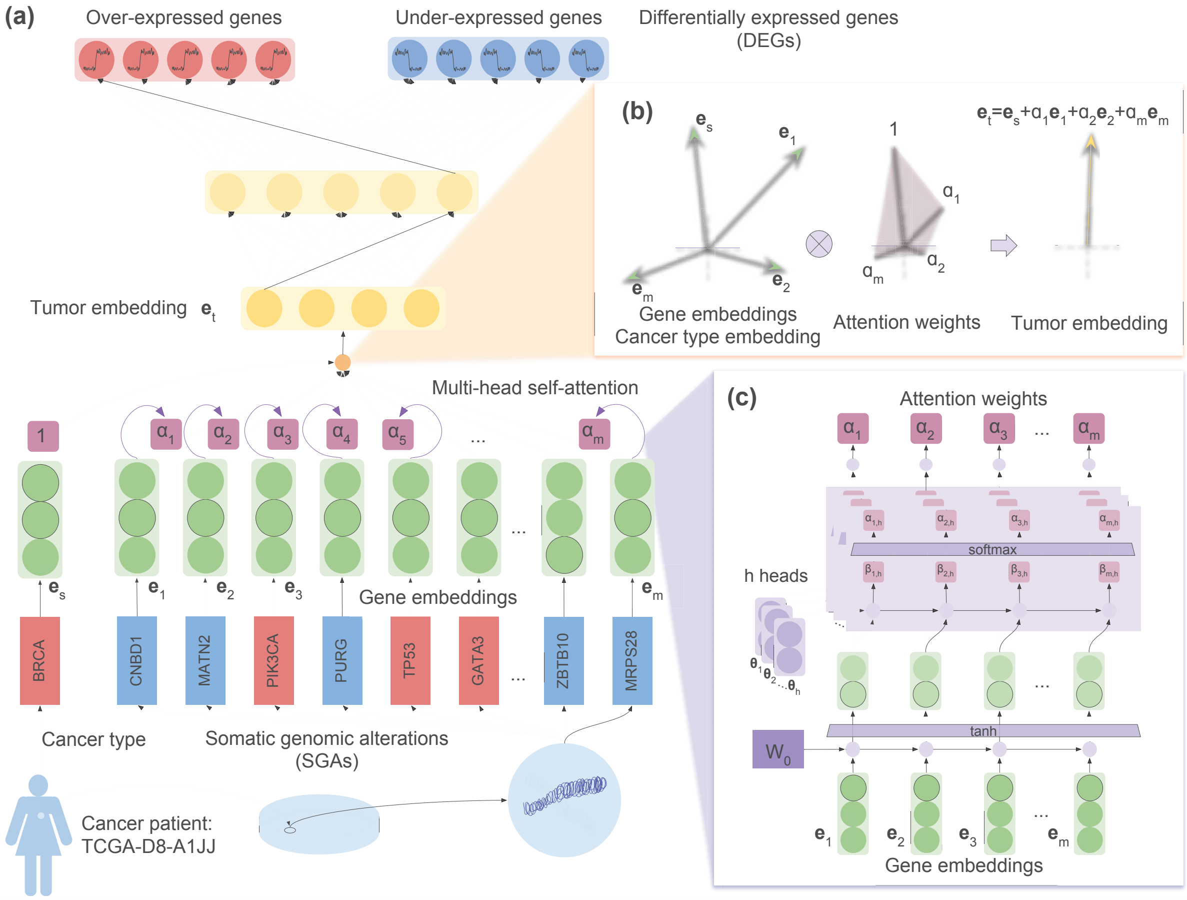

Figure 2.1: (A) Overall architecture of GIT. An example case and its detected drivers are shown. (B) A two-dimensional demo that shows how attention mechanism combines multiple gene embeddings of SGAs $\{\mathbf{e}_g\}_{g=1}^m$ and cancer type embedding $\mathbf{e}_s$ into a tumor embedding vector $\mathbf{e}_t$ using attention weights $\{\alpha_g\}_{g=1}^m$. (C) Calculation of attention weights $\{\alpha_g\}_{g=1}^m$ using gene embeddings $\{\mathbf{e}_g\}_{g=1}^m$.

Pre-training gene embeddings using Gene2Vec algorithm

In this study, we projected the discrete binary representation of SGAs perturbing a gene into a continuous embedding space, which we call "gene embeddings" of corresponding SGAs, using a "Gene2Vec" algorithm, based on the assumption of co-occurrence pattern of SGAs in each tumor, including mutually exclusive patterns of mutations affecting a common pathway [234]. These gene embeddings were further updated and fine-tuned by the GIT model with the supervision of affected DEGs.

The "Gene2Vec" is closely related the skip gram word2vec [153] pre-training algorithm. The biology rationale behind Gene2Vec algorithm is that we are able to portrait the co-occurrence pattern of SGAs in each tumor, i.e., mutually exclusive mutations [234], using gene embeddings and gene context embeddings.

Given the gene embedding $\mathbf{e}_g$ of an SGA-affected gene g and context embedding of any possible SGA-affected gene c': $\mathcal{V} = \{\mathbf{v}_{c'}\}_{c' \in \mathcal{G}}$, where $\mathcal{G}$ is the set of all possible SGA-affected genes, the skip gram paradigm assumes the probability that an alteration in gene c happens together with the alteration in gene g within a tumor with probability:

$$\Pr\left(c \in \text{Context}(g) \mid g\right) = \frac{\exp\left(\mathbf{e}_g^\intercal \mathbf{v}_c\right)}{\sum_{c' \in \mathcal{G}} \exp\left(\mathbf{e}_g^\intercal \mathbf{v}_{c'}\right)}. \tag{2.1}$$

We used the negative sampling (NS) technique to approximately maximize the log-likelihood of skip gram, which would otherwise be computationally expensive to optimize if directly following Eq. 2.1. Algorithm 1 shows implementation of Gene2Vec.

Encoder: multi-head self-attention mechanism

To detect the difference of functional impact of SGAs in a tumor, we designed a multi-head self-attention mechanism (Fig. 2.1A middle part).

For all SGA-affected genes $\{g\}_{g=1}^{m}$ and the cancer type s of a tumor t, we first mapped them to corresponding gene embeddings $\{\mathbf{e}_g\}_{g=1}^{m}$ and a cancer type embedding $\mathbf{e}_s$ from a look-up table $\mathcal{E} = \{\mathbf{e}_g\}_{g \in \mathcal{G}} \cap \{\mathbf{e}_s\}_{s \in \mathcal{S}}$, where $\mathbf{e}_g$ and $\mathbf{e}_s$ are real-valued vectors. From the implementation perspective, we treated cancer types in the same way as SGAs, except the attention weight of it is fixed to be "1".

The overall idea of producing the tumor embedding $\mathbf{e}_t$ is to use the weighted sum of cancer type embedding $\mathbf{e}_s$ and gene embeddings $\{\mathbf{e}_g\}_{g=1}^{m}$ (Fig. 2.1B) :

$$\mathbf{e}_t = 1 \cdot \mathbf{e}_s + \sum\nolimits_g \alpha_g \cdot \mathbf{e}_g = 1 \cdot \mathbf{e}_s + \alpha_1 \cdot \mathbf{e}_1 + ... + \alpha_m \cdot \mathbf{e}_m. \tag{2.2}$$

The attention weights $\{\alpha_g\}_{g=1}^{m}$ are calculated by employing multi-head self-attention mechanism, using gene embeddings of SGAs $\{\mathbf{e}_g\}_{g=1}^{m}$ in the tumor (Fig. 2.1C):

$$\alpha_1, \alpha_2, ..., \alpha_m = \text{Function}_{\text{Attention}}(\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m). \tag{2.3}$$

The attention function $\text{Function}_{\text{Attention}}$ is implemented as a sub-network. In the case of single-head attention, there is only one single head parameter $\boldsymbol{\theta}_j$, and the unnormalized weights $\{\beta_{g,j}\}_{g=1}^{m}$

```
Data: Genomic alterations in each tumor: $\mathcal{T} = \left\{T_i = \{g_{i1}, g_{i2}, ..., g_{im(i)}\}\right\}_{i=1,2,...,N}$.
Result: Pretrained gene embedding of each gene:
$\mathcal{E} = \{\mathbf{e}_g \in \mathbb{R}^n\}_{\ g\in\mathcal{G}}$.
Context gene embeddings:
$\mathcal{V} = \{\mathbf{v}_g \in \mathbb{R}^n\}_{\ g\in\mathcal{G}}$.
$f(g) \leftarrow \frac{1}{Z}\sum_{i=1}^{N} \nVdash(g \in T_i),\ g \in \mathcal{G}$;                        // Gene frequency
$f_n(g) \leftarrow \frac{1}{Z_n} f(g)^{3/4},\ g \in \mathcal{G}$;                                   // Normalized frequency
$\mathbf{e}_g \sim U\left(-\frac{0.5}{n}, \frac{0.5}{n}\right)^n,\ \mathbf{v}_g \leftarrow 0^n,\ g \in \mathcal{G}$;   // Initialize gene embeddings and context
 embeddings
while not converges do
    $l \leftarrow 0$;                                        // Total loss of a mini-batch samples
    for $b = 1, 2, ..., batch\_size$ do
        $g \sim f$ ;                                                    // Sample a gene
        $g_c \sim$ Context($g$ ; $\mathcal{T}$);                                // Sample a context gene
        $g_{nr} \sim f_n,\ r = 1, 2, ..., R$ ;                         // Sample negative context genes
        $l \leftarrow l + \text{NSLoss}\left(g, g_c, \{g_{nr}\}_{r=1}^{R}\ ;\ \mathcal{E}, \mathcal{V}\right)$ ;                   // Update
    end
    $(\mathcal{E}, \mathcal{V}) \leftarrow (\mathcal{E}, \mathcal{V}) - \eta \cdot \frac{\partial l}{\partial(\mathcal{E},\mathcal{V})}$ ;                        // Gradient descent
end
Function Context($g$ ; $\mathcal{T}$)
    $P_c \leftarrow U\left(\{g_c \mid g_c \in T_i, g \in T_i\}_{i=1,2,...,N}\right)$ ; // Uniform distribution on sequence of
     adjacent mutations
return $P_c$
Function NSLoss($g, g_c, \{g_{nr}\}_{r=1}^{R}$ ; $\mathcal{E}, \mathcal{V}$)
    $l \leftarrow \log\sigma\left(\mathbf{e}_g^\intercal \mathbf{v}_{g_c}\right) + \sum_{r=1}^{R}\log\sigma\left(-\mathbf{e}_g^\intercal \mathbf{v}_{g_{nr}}\right)$;      // Negative sampling loss of one
     sample
return $l$
```

Algorithm 1: Gene2Vec algorithm to pre-train the gene embeddings using skip gram with negative sampling loss. Given the context information of somatic genomic alterations (SGAs) in each cancer patient, i.e., whether two SGAs happened together in a single tumor, we pre-trained the gene embeddings (and context gene embeddings) using similar techniques to word2vec. Skip gram was used to predict the probability of co-occurred SGAs c given a known SGA g, as explained in Equation 2.1. Negative sampling loss was utilized to accelerate the maximization of log-likelihood in the skip gram assumption. Instead of original mutation frequency $f(g)$, the negative sampling frequency of SGA was sub-sampled by scaling to $f(g)^{3/4}$. In practice, the step size η in mini-batch gradient descent was decayed after training for every epoch to converge fast and prevent overfitting. Note that $\mathcal{E}$ here is defined slightly different from that in the main context, which contains both gene and cancer type embeddings.

can be derived as follows:

$$\beta_{g,j} = \boldsymbol{\theta}_j^{\intercal} \cdot \tanh(W_0 \cdot \mathbf{e}_g),\ g = 1, 2, ..., m, \tag{2.4}$$

which are further normalized to single-head weights $\{\alpha_{g,j}\}_{g=1}^{m}$:

$$\alpha_{1,j}, \alpha_{2,j}, ..., \alpha_{m,j} = \text{softmax}(\beta_{1,j}, \beta_{2,j}, ..., \beta_{m,j}), \tag{2.5}$$

where softmax function is defined as : $\alpha_g = \exp{(\beta_g)} / \sum_{g'=1}^{m} \exp{(\beta_{g'})}$. In the case of multi-head attention, there exist h different parameters $\Theta = \{\boldsymbol{\theta}_j\}_{j=1}^{h}$. Then multiple attention weights of each gene embedding are generated following Eq. 2.4,2.5 and summed up to be the final multi-head attention weight:

$$\alpha_g = \sum\nolimits_{j=1}^{h} \alpha_{g,j} = \alpha_{g,1} + \alpha_{g,2} + ... + \alpha_{g,h},\ g = 1, 2, ..., m. \tag{2.6}$$

Overall we have three parameters $\{W_0, \Theta, \mathcal{E}\}$ to train in the multi-head attention module using back-propagation [194]. The look-up table $\{\mathbf{e}_g\}_{g \in \mathcal{G}}$ was initialized with Gene2Vec pre-trained gene embeddings and refined by GIT here.

Decoder: multi-layer perceptron (MLP)

For a specific tumor t, we fed tumor embedding $\mathbf{e}_t$ into an MLP with one hidden layer as the decoder, using non-linear activation functions and fully connected layers, to produce the final predictions $\hat{y}$ for DEGs y (Fig. 2.1A top part):

$$\hat{y} = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot \text{ReLU}(\mathbf{e}_t) + b_1) + b_2). \tag{2.7}$$

where ReLU$(x) = \max(0, x)$ is rectified linear unit, and $\sigma(x) = (1+\exp(-x))^{-1}$ is sigmoid activation function. The output of the decoder and actual values of DEGs were used to calculate the ℓ_2-regularized cross entropy, which was minimized during training:

$$\min_{\mathcal{W}, \mathcal{E}, \Theta, b} \text{CrossEnt}(y, \hat{y}) + \ell_2(\mathcal{W}, \mathcal{E}, \Theta; \lambda_2), \tag{2.8}$$

where $\mathcal{W} = \{W_l\}_{l=0}^{2}$, cross entropy loss defined as

$$\text{CrossEnt}(y, \hat{y}) = -\sum\nolimits_i \left[(1 - y_i) \log(1 - \hat{y}_i) + y_i \log \hat{y}_i\right], \tag{2.9}$$

and ℓ_p regularizer defined as

$$\ell_p(\mathcal{W}; \lambda) = \lambda \cdot \sum\nolimits_l \|W_l\|_p\,, p \in \{1, 2\}. \tag{2.10}$$

2.1.3 Training and evaluation

We utilized PyTorch (`https://pytorch.org/`) to train, validate and test the Gene2Vec, GIT (variants) and other conventional models (Lasso and MLPs; Sec. 2.2.1). The training, validation and test sets were split in the ratio of 0.33:0.33:0.33 and fixed across different models. The hyperparameters were tuned over the training and validation sets to get best F1 scores, trained on training and validation sets, and finally applied to the test set for evaluation if not further mentioned below. The models were trained by updating parameters using backpropagation [194], specifically, using mini-batch Adam [109] with default momentum parameters. Gene2Vec used mini-batch stochastic gradient descent (SGD) instead of Adam. Dropout [210] and weight decay (ℓ_p-regularization) were used to prevent overfitting. We trained all the models 30 to 42 epochs until they fully converged. The output DEGs were represented as a sparse binary vector. We utilized various performance metrics including accuracy, precision, recall, and F1 score, where F1 is the harmonic mean of precision and recall. The training and test were repeated for five runs get the mean and variance of evaluation metrics. We designed two metrics in this chapter for evaluating the functional similarity among genes sharing similar gene embedding: "nearest neighborhood (NN) accuracy" and "GO enrichment".

2.2 Results

2.2.1 GIT statistically detects real biological signals

The task of GIT is to predict DEGs (dependent variables) using SGAs as input (independent variables). As a comparison, we trained and tested the Lasso (multivariate regression with ℓ_1-regularization) [230] and MLPs [193] as baseline prediction models to predict DEGs based on SGAs. The Lasso model is appealing in our setting because, when predicting a DEG, it can filter out most of the irrelevant input variables (SGAs) and keep only the most informative ones, and it is a natural choice in our case where there are 19.8k possible SGAs. However, in comparison to MLP, it lacks the capability of portraying complex relationships between SGAs and DEGs. On the other hand, while conventional MLPs have sufficient power to capture complex relationships–particularly, the neurons in hidden layers may mimic signaling proteins [38]–they can not utilize any biological knowledge extracted from cancer genomics, nor do they explain the signaling process and distinguish driver SGAs. We employed the precision, recall, F1 score, as well as accuracy to compare GIT and traditional methods (Tab. 2.1: 1st to 4th, and last rows). One can conclude that GIT outperforms all these other conventional baseline methods for predicting DEGs in all metrics, indicating the specifically designed structure of GIT is able to soar the performance in the task of predicting DEGs from SGAs.

In order to evaluate the utility of each module (procedure) in GIT, we conducted ablation study by removing one module at a time: the cancer type input ("can"), the multi-head self-attention module ("attn"), and the initialization with pre-trained gene embeddings ("init"). The impact of each module can be detected by comparing to the full GIT model. All the modules in GIT help to improve the prediction of DEGs from SGAs in terms of overall performance: F1 score and accuracy (Tab. 2.1: 5th to last rows).

Table 2.1: Performances of Genomic Impact Transformer (GIT) and baseline methods.

Methods	Precision	Recall	F1 score	Accuracy
Lasso	59.6±0.05	52.8±0.03	56.0±0.01	74.0±0.02
1 layer MLP	61.9±0.09	50.4±0.17	55.6±0.07	74.7±0.02
2 layer MLP	64.2±0.39	52.0±0.66	57.4±0.28	75.9±0.09
3 layer MLP	64.2±0.37	50.5±0.30	56.5±0.19	75.7±0.13
GIT - can	60.5±0.34	45.8±0.38	52.1±0.29	73.6±0.14
GIT - attn	67.6±0.32	55.3±0.77	60.8±0.35	77.7±0.05
GIT - init	**69.8**±0.28	54.1±0.37	60.9±0.16	78.3±0.06
GIT	69.5±0.09	**57.1**±0.18	**62.7**±0.08	**78.7**±0.01

2.2.2 Gene embeddings compactly represent the functional impact of SGAs

We examined whether the gene embeddings capture the functional similarity of SGAs, using mainly two metrics: NN accuracy and GO enrichment.

NN accuracy By capturing the co-occurrence pattern of somatic alterations, the Gene2Vec pre-trained gene embeddings improve 36% in NN accuracy over the random chance of any pair of the genes sharing Gene Ontology (GO) annotation [8] (Tab. 2.2). The fine-tuned embeddings by GIT further show a one-fold increase in NN accuracy. These results indicate that the learned gene embeddings are consistent with the gene functions, and they map the discrete binary SGA representation into a meaningful and compact space.

Table 2.2: Nearest Neighborhood (NN) accuracy with respect to Gene Ontology (GO) in different gene embedding spaces.

Gene embeddings	NN accuracy	Improvement
Random pairs	5.3±0.36	–
Gene2Vec	7.2	36%
Gene2Vec + GIT	**10.7**	100%

GO enrichment We performed clustering analysis of SGAs in embedding space using k-means clustering, and calculated GO enrichment, and we varied the number of clusters (k) to derive clusters with different degrees of granularity (Fig. 2.2A). As one can see, when the genes are randomly distributed in the embedding space, they get GO enrichment of 1. However, in the gene embedding space, the GO enrichment increases fast until the number of clusters reaches 40, indicating a strong correlation between the clusters in embedding space and the functions of the genes.

To visualize the manifold of gene embeddings, we grouped the genes into 40 clusters, and conducted the t-SNE [140] of genes (Fig. 2.2B left panel). Using PANTHER GO enrichment

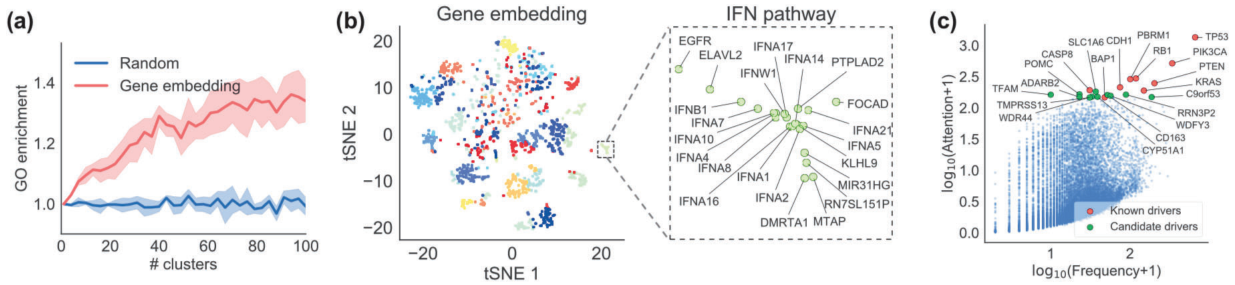

Figure 2.2: (A) GO enrichment of vs. number of groups in *k*-means clustering. (B) t-SNE visualization of gene embeddings. The different colors represent *k*-means (40 clusters) clustering labels. An enlarged inset of a cluster is shown, which contains a set of closely related genes which we refer to "IFN pathway". (C) Landscape of attention of SGAs based on attention weights and frequencies.

analysis [152], 12 out of 40 clusters are shown to be enriched in at least one biological process. Most of the gene clusters are well-defined and tight located in the projected t-SNE space. As a case study, we took a close look at one cluster (Fig. 2.2B right panel), which contains a set of functionally similar genes, such as that code a protein family of type I interferons (IFNs), which are responsible for immune and viral response [54].

2.2.3 Self-attention reveals impactful SGAs on cancer cell transcriptome

While it is widely accepted that cancer is mainly caused by SGAs, but not all SGAs observed in a cancer cell are causative [239]. Previous methods mainly concentrate on searching for SGAs with higher than expected frequency to differentiate candidate drivers SGAs from passenger SGAs. GIT provides a novel perspective to address the problem: identifying the SGAs that have a functional impact on cellular signaling systems and eventually lead DEGs as the *tumor-specific* candidate drivers. Here we compare the relationship of overall attention weights (inferred by GIT model) and the frequencies of somatic alterations (used as the benchmark/control group) in all the cancer types (Pan-Cancer) from our test data (Fig. 2.2C). In general, the attention weights are correlated with the alteration frequencies of genes, e.g., common cancer drivers such as *TP53* and *PIK3CA* are the top two SGAs selected by both methods [104]. However, our self-attention mechanism assigns high weights to many of genes previously not designated as drivers, indicating these genes are potential cancer drivers although their roles in cancer development remain to be further studied. Table 2.3 lists top SGAs ranked according to GIT attention weights in pan-cancer and five selected cancer types, where known cancer drivers from TumorPortal [120] and IntOGen [80] are marked as bold font. Apart from *TP53* and *PIK3CA* as drivers in the pan-cancer analysis [104], we also find the top cancer drivers in specific cancer types consistent with our knowledge of cancer oncology. For example, *CDH1* and *GATA3* are drivers of breast invasive carcinoma (BRCA) [161], *CASP8* is known driver of head and neck squamous cell carcinoma (HNSC) [212], *STK11*, *KRAS*, *KEAP1* are known drivers of lung adenocarcinoma (LUAD) [165], *PTEN* and *RB1* are drivers of glioblastoma (GBM) [24], and *FGFR3*, *RB1*, *HSP90AA1*, *STAG2* are known drivers in urothelial bladder carcinoma (BLCA) [164]. In contrast, the most frequently mutated genes (control group) are quite different from that using attention mechanism (experiment group), and only a few of them are known drivers.

Table 2.3: Top five SGA-affected genes ranked according to attention weight.

Rank	PANCAN	BRCA	HNSC	LUAD	GBM	BLCA
1	***TP53***	***TP53***	***TP53***	***STK11***	***TP53***	***TP53***
2	***PIK3CA***	***PIK3CA***	***CASP8***	***TP53***	***PTEN***	***FGFR3***
3	***RB1***	***CDH1***	***PIK3CA***	***KRAS***	*C9orf53*	***RB1***
4	***PBRM1***	***GATA3***	***CYLD***	*CYLC2*	***RB1***	***HSP90AA1***
5	***PTEN***	*MED24*	*RB1*	***KEAP1***	*CHIC2*	***STAG2***

2.2.4 Personalized tumor embeddings reveal distinct survival profiles

Besides learning the specific biological function impact of SGAs on DEGs, we further examined the utility of tumor embeddings $\mathbf{e}_t$ in two perspectives: (1) Discovering patterns of tumors potentially sharing common disease mechanisms across different cancer types; (2) Using tumor embedding to predict patient survival.

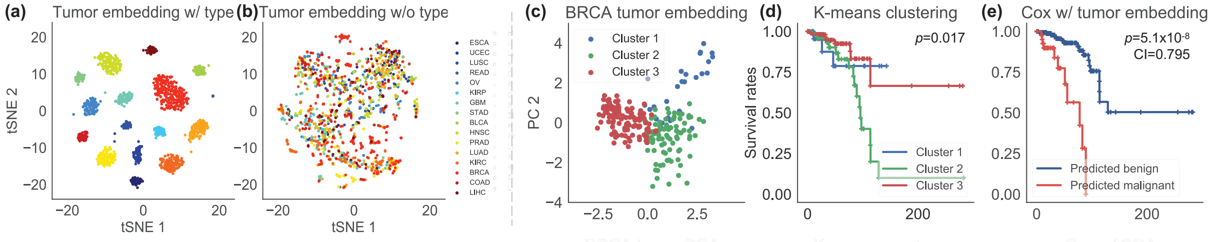

Figure 2.3: (A) t-SNE of full tumor embedding $\mathbf{e}_t$. (B) t-SNE of stratified tumor embedding ($\mathbf{e}_t$-$\mathbf{e}_s$). (C) PCA of tumor embedding shows internal subtype structure of BRCA tumors. Color lablels the group index of k-means clustering. (D) KM estimators of the three breast cancer groups. (E) Cox regression using tumor embeddings.

We first used the t-SNE plot of tumor embeddings to illustrate the common disease mechanisms across different cancer types (Fig. 2.3A). When cancer type embedding $\mathbf{e}_s$ is included in full tumor embedding $\mathbf{e}_t$, which has a much higher weight than any individual gene embedding (Fig. 2.1B, Eq. 2.2) and dominates the full tumor embedding, tumor samples are clustered according to cancer types. This is not surprising as it is well appreciated that expressions of many genes are tissue-specific [89]. To examine the pure effect of SGAs on tumor embedding, we removed the effect of tissue by subtracting cancer type embeddings $\mathbf{e}_s$, followed by clustering tumors in the stratified tumor embedding space (Fig. 2.3B). It is interesting to see that each dense area (potential tumor clusters) includes tumors from different tissues of origins, indicating SGAs in these tumors may reflect shared disease mechanisms (pathway perturbations) among tumors, warranting further investigations.

The second set of experiments was to test whether differences in tumor embeddings (thereby difference in disease mechanisms) are predictive of patient clinical outcomes. We conducted unsupervised *k*-means clustering using only breast cancer tumors from our test set, which reveals 3 three groups (Fig. 2.3C) with significant difference in survival profiles evaluated by log-rank test [145] (Fig. 2.3D; p-value=0.017). In addition, using tumor embeddings as input features, we trained $\ell_{1,2}$-regularized (elastic net) [276] Cox proportional hazard models [46] in a 10-fold cross-validation (CV) experiment. This led to an informative ranked list of tumors according to predicted survivals/hazards evaluated by the concordance index (CI) value (CI=0.795), indicating that the trained model is very accurate. We further split test samples into two groups divided by the median of predicted survivals/hazards, which also yields significant separation of patients in survival profiles (Fig. 2.3E; p-value=5.1×10^{-8}), indicating that our algorithm has correctly ranked the patients according to characteristics of the tumor.

As shown above, distinct SGAs may share similar embeddings if they share similar functional impact. Thus, two tumors may have similar tumor embeddings even though they do not share any SGAs, as long as the functional impact of distinct SGAs from these tumors are similar. Therefore, tumor embedding makes it easier to discover common disease mechanisms and their impact on patient survival.

2.2.5 Tumor embeddings are predictive of drug responses of cancer cell lines

Precision oncology concentrates on using patient-specific omics data to determine optimal therapies for a patient. We set out to see if SGA data of cancer cells can be used to predict their sensitivity to anti-cancer drugs. We used the CCLE dataset [12], which performed drug sensitivity screening over hundreds of cancer cell lines and 24 anti-cancer drugs. The study collects genomic and transcriptomic data of these cell lines, but in general, the genomic data (except the molecularly targeted genes) from a cell line are not sufficient to predict sensitivity its sensitivity to different drugs.

We discretized the response of each drug following the procedure in previous research [12, 61]. Since CCLE only contains a small subset of mutations in TCGA dataset (around 1,600 gene mutations), we retrained the GIT with this limited set of SGAs in TCGA, using default hyperparameters we set before. Cancer type input was removed as well, which is not explicitly provided in CCLE dataset. The output of tumor embeddings $\mathbf{e}_t$ was then extracted as feature. We formulated drug response prediction as a binary classification problem with ℓ_1-regularized cross entropy loss (Lasso), where the input can be raw sparse SGAs or tanh-curved tumor embeddings tanh$(\mathbf{e}_t)$. Following previous work [12], we performed 10-fold CV experiment training Lasso using either inputs to test the drug response prediction task of four drugs with distinct targets. Lasso regression using tumor embeddings consistently outperforms the models trained with original SGAs as inputs (Fig. 2.4). Specifically, in the case of Sorafenib, the raw mutations just give random prediction results, while the tumor embedding is able to give predictable results. It should be noted that it is possible that certain cancer cells may host SGAs along the pathways related to FGFR, RAF, EGFR, and RTK, rendering them sensitive to the above drugs. Such information can be implicitly captured and represented by the tumor embeddings, so that the information from raw SGAs are captured and pooled to enhance classification accuracy.

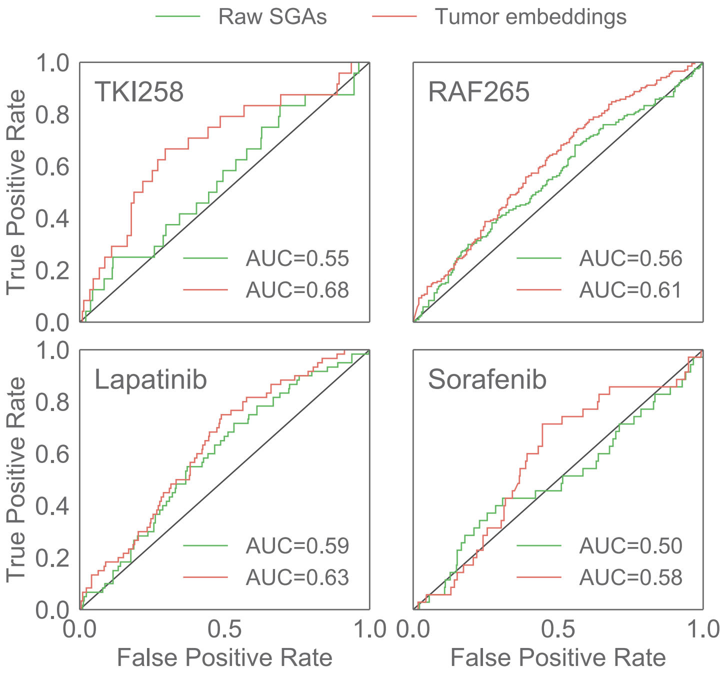

Figure 2.4: ROC curves and the areas under the curve (AUCs) of Lasso models trained with original SGAs and tumor embeddings representations on predicting responses to four drugs.

2.3 Discussion

Despite the significant advances in cancer biology, it remains a challenge to reveal disease mechanisms of each individual tumor, particularly which and how SGAs in a cancer cell lead to the development of cancer. Here we propose the GIT model to learn the general impact of SGAs, in the form of gene embeddings, and to precisely portray their effects on the downstream DEGs with higher accuracy. With the supervision of DEGs, we can further assess the importance of an SGA using multi-head self-attention mechanisms in each individual tumor. More importantly, while the tumor embeddings are trained with predicting DEGs as the task, it contains information for predicting other phenotypes of cancer cells, such as patient survival and cancer cell drug sensitivity. The key advantage of transforming SGA into a gene embedding space is that it enables the detection and representation of the functional impact of SGAs on cellular processes, which in turn enables detection of common disease mechanisms of tumors even if they host different SGAs. We anticipate that GIT, or other future models like it, can be applied broadly to gain mechanistic insights of how genomic alterations (or other perturbations) lead to specific phenotypes, thus providing a general tool to connect genome to phenome in different biological fields and genetic diseases. One should also be careful that despite the correlation of genomic alterations and phenotypes such as survival profiles and drug response, the model may not fully reveal the causalities and there may exist other confounding factors not considered.

There are a few future directions for further improving the GIT model. First of all, decades of biomedical research has accumulated a rich body of knowledge, e.g., Gene Ontology and gene regulatory networks, which may be incorporated as the prior of the model to boost the performance [139]. Secondly, we expect that by getting a larger corpus of tumor data with mutations and gene expressions, we will be able to train better models to minimize potential overfitting or variance. Lastly, more clinically oriented investigations are warranted to examine, when trained with a large volume of tumor omics data, the learned embeddings of SGAs and tumors may be applied to predict sensitivity or resistance to anti-cancer drugs based SGA data that are becoming readily available in contemporary oncology practice.

Chapter 3

Inference of drug sensitivity in cancer cell lines through collaborative filtering with contextual attention[1]

Precise prediction of drug sensitivities of tumors is one of the essential aspects of personalized treatment of cancers, which requires clinicians and researchers to assign the best potential anti-cancer drugs to individual patients [181]. With the rapid advancement of high-throughput sequencing technologies in the past decade, large amounts of tumor multi-omics data have become available at an acceptable cost [189]. However, inter- and intra-tumor heterogeneities [199] make tumor resistance to drugs a much more complex problem to resolve: Even cancer patients with the same cancer type may have distinct prognoses with the same clinical intervention [182]. Furthermore, it is expensive and impractical to conduct systematic drug sensitivity assays directly on human beings. Although there is still a gap between cell lines and *in vivo* tumors, large-scale cancer-cell-line pharmacogenomic data [12, 257] provide a reasonable basis for understanding how cells and drugs interact and how inter-tumor heterogeneity can lead to distinct sensitivity profiles across tumors.

Predicting the sensitivities of cell lines to a panel of potential molecules based on omics data of the cell lines is challenging in three primary aspects. 1) Drug sensitivity data are **noisy** and often contain many missing entries [131]. Therefore the model has to be robust, generalize well, and not otherwise overfit to the training data [261]. 2) The relationship between the molecular profiles of cell lines (such as gene expressions) and drug response is complex. Not every gene contributes equally to the response. In addition, a few genes may **interact** with each other to generate complex **contextual** effects [265]. 3) Cancer researchers and clinicians are especially concerned about the **interpretability** and clinical implications of the models, with an emphasis on how the critical biomarkers affect the final prediction results. Although deep learning models can achieve good to excellent performance in predicting sensitivities [39, 61], most of them behave like "black boxes" without achieving balanced performance and interpretability.

[1] This chapter was developed from material published in "Yifeng Tao, Shuangxia Ren, Michael Q. Ding, Russell Schwartz, and Xinghua Lu. Predicting drug sensitivity of cancer cell lines via collaborative filtering with contextual attention. volume 126 of *Proceedings of Machine Learning Research*, pages 660–684, Virtual, 07–08 Aug 2020" [224].

To address the three essential challenges mentioned above in cell line resistance inference, we proposed a model for accurate and interpretable drug sensitivity prediction, which we called CADRE (**C**ontextual **A**ttention-based **D**rug **RE**sponse). CADRE was built on the framework of a type of classical machine learning model called collaborative filtering [198] to impute the sensitivities of untested cell lines to a panel of known drugs using their molecular profiles, such as gene expressions (Sec. 3.1.1). Collaborative filtering captures shared features by jointly exploiting the similarities between drugs as well as similarities between cell lines, alleviating the significant **noise** in the sensitivity data [242]. Furthermore, we developed and employed a contextual attention mechanism to identify the crucial inputs, capture the interactions between genes and drug targets, and thus encode the cell line features from their expression profiles effectively (See Sec. 3.1.2-3.1.3 for implementation details). The attention mechanism is a family of deep learning modules/components which has been shown to be effective in encoding input features by assigning them different "attention weights" and thus further improving the model performance in many applications, such as computer vision [255], natural language processing [258], and computational biology [221]. Not only did contextual attention increase prediction accuracy by capturing the **contextual interactions** of genes and drug targets, but it also improved the model **interpretability** by assigning higher weights to the genes that have a more significant effect on drug response. Although classical models such as random forest and linear models are well studied theoretically and have better interpretability compared with attention mechanism, attention mechanism still improves both model interpretability and empirical performance in deep learning.

3.1 Materials and methods

3.1.1 Overall architecture: collaborative filtering

The overall architecture of CADRE is collaborative filtering (Fig. 3.1A,B; 198). Given a cell line c and a drug d, CADRE first maps the cell line and drug to two feature vectors, which we call cell line embedding $\vec{e}_c \in \mathbb{R}^s$ and drug embedding $\vec{e}_d \in \mathbb{R}^s$. CADRE then predicts the probability that the cell line c will be sensitive to drug d through the inner product and logistic function:

$$\hat{y}_{c,d} = \sigma\left(\langle \vec{e}_c, \vec{e}_d \rangle\right) = \frac{1}{1 + \exp(-\vec{e}_c^{\intercal} \vec{e}_d)}. \tag{3.1}$$

We define $\mathcal{W}$ as all the model parameters to be optimized/trained, such as gene embeddings, drug embeddings, target pathway embeddings, and neural network weights. At the training stage, we optimize the loss function:

$$\ell(\hat{y}_{c,d}, y_{c,d}; \mathcal{W}) = \text{CrossEnt}(\hat{y}_{c,d}, y_{c,d}) + \frac{\lambda_2}{2} \cdot \ell_2(\mathcal{W}), \tag{3.2}$$

where

$$\text{CrossEnt}(\hat{y}_{c,d}, y_{c,d}) = -\left[y_{c,d} \cdot \log \hat{y}_{c,d} + (1 - y_{c,d}) \cdot \log(1 - \hat{y}_{c,d})\right] \tag{3.3}$$

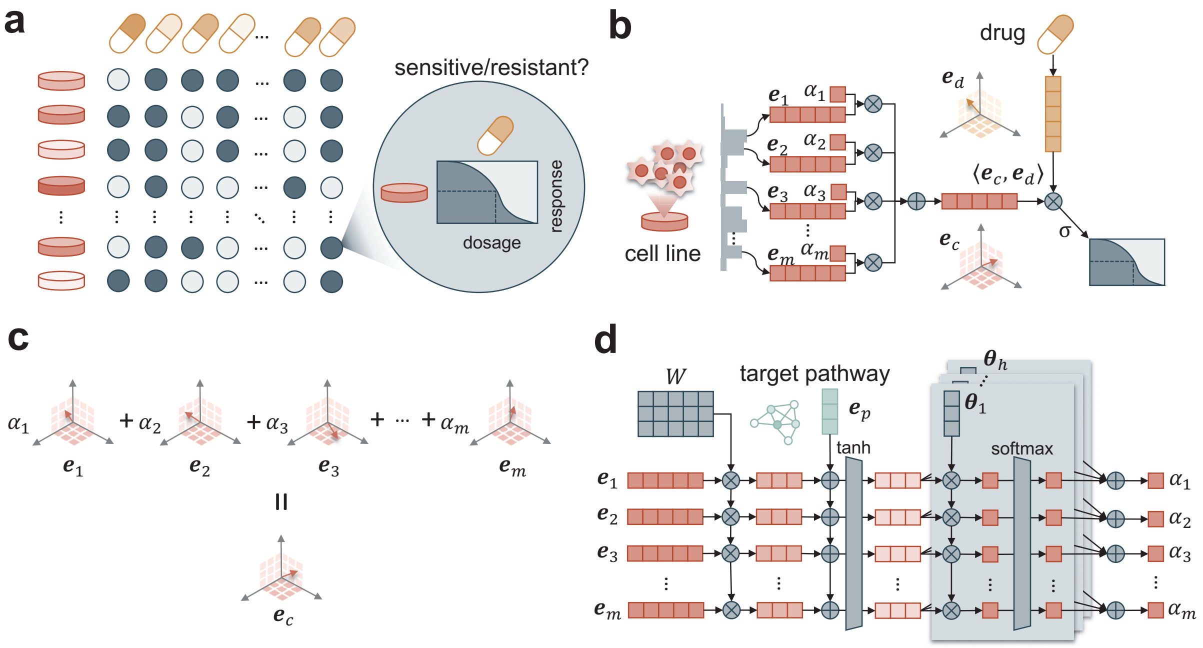

Figure 3.1: Diagram of the CADRE model. (A) The general goal of CADRE is to predict the responses of cancer cell lines to a panel of given anti-cancer drugs based on their gene expression profiles. (B) Given a pair of cell line c and drug d, CADRE first extracts the gene embeddings of m expressed genes in the cell line $\vec{e}_1$, $\vec{e}_2$, ..., $\vec{e}_m$ and the drug embedding $\vec{e}_d$. Then it generates the cell line embedding $\vec{e}_c$ as the weighted sum of gene embeddings. Finally, the predicted response will be $\sigma(\vec{e}_c^{\intercal}\vec{e}_d)$. (C) CADRE generates the cell line embedding $\vec{e}_c$ as a weighted sum of its consisting gene embeddings. (D) CADRE calculates the attention weights α_1, α_2, ..., α_m through the contextual attention mechanism, which was implemented as a sub neural network. It takes as input both the gene embeddings $\vec{e}_1$, $\vec{e}_2$, ..., $\vec{e}_m$ and drug target pathway embedding $\vec{e}_p$.

is the cross-entropy between predicted sensitivity $\hat{y}_{c,d}$ and ground truth sensitivity $y_{c,d}$, $\ell_2(\mathcal{W})$ is the ℓ_2-regularization term to prevent overfitting, λ_2 is the weight decay coefficient.

The mapping from drug d to its drug embedding $\vec{e}_d$ is direct, through a lookup table of drug embeddings $\mathcal{E}_\mathcal{D} = \{\vec{e}_d\}_{d \in \mathcal{D}}$, where $\mathcal{D}$ is the set of all the drugs in the dataset. We considered a total of 3,000 most varied genes, and for each cell line c, we had a set of m=1,500 genes that were highly expressed (Sec. 3.1.6). Instead of a binary 3,000-dimensional vector, the input of the CADRE and collaborative filtering models is similar to the "bag of words": the indices of the m expressed genes $\{1, 2, ..., m\}$. We then mapped these gene indices into their corresponding gene embeddings $\vec{e}_1, \vec{e}_2, ..., \vec{e}_m$ using a lookup table $\mathcal{E}_\mathcal{G} = \{\vec{e}_g\}_{g \in \mathcal{G}}$, where $\mathcal{G}$ is all the set of all the genes we considered ($|\mathcal{G}|$ = 3,000), and $\vec{e}_g \in \mathbb{R}^s$. We chose the parameter 3,000 and 1,500 following previous work (61; Sec. 3.1.6). The major difference between CADRE and "vanilla" collaborative filtering is in how to calculate the cell line embedding $\vec{e}_c$ from the gene embeddings $\vec{e}_1, \vec{e}_2, ..., \vec{e}_m$. As we will introduce in Sec. 3.1.2-3.1.3, SADRE (a CADRE variant) and CADRE incorporate attention mechanism to calculate $\vec{e}_c$. In vanilla collaborative filtering, however, we naively take the sum (or average) of all the gene embeddings $\vec{e}_1, \vec{e}_2, ..., \vec{e}_m$:

$$\vec{e}_c = \sum\nolimits_{i=1}^{m} 1 \cdot \vec{e}_i = 1 \cdot \vec{e}_1 + 1 \cdot \vec{e}_2 + ... + 1 \cdot \vec{e}_m. \tag{3.4}$$

We added a dropout layer [210] after the $\vec{e}_c$ to reduce the model complexity to prevent overfitting:

$$\vec{e}_c = \text{dropout}(\vec{e}_c; \rho), \tag{3.5}$$

where $\rho \in [0, 1]$ is the dropout rate. We use "vanilla" to refer to the standard collaborative filtering model, in contrast to more advanced models, such as CADRE in this chapter, which also build on the framework of collaborative filtering.

3.1.2 SADRE: Self-attention-based drug response

Instead of summing up all the m gene embeddings with equivalent weights in the vanilla collaborative filtering (Sec. 3.1.1; Eq. 3.4), we assumed that different genes should have different importance when we aggregate them into a single cell line embedding (Fig. 3.1B,C):

$$\vec{e}_c = \sum\nolimits_{i=1}^{m} \alpha_i \cdot \vec{e}_i = \alpha_1 \cdot \mathbf{e}_1 + \alpha_2 \cdot \mathbf{e}_2 + ... + \alpha_m \cdot \mathbf{e}_m. \tag{3.6}$$

We could calculate the weights $\alpha_1, \alpha_2, ..., \alpha_m$ ($\alpha_i > 0$, $i = 1, 2, ..., m$) through various attention mechanisms [171, 258], thus enabling a better feature representation of the cell line from its composing gene embeddings. We developed two attention-based collaborative filtering models in this chapter: SADRE and CADRE. CADRE uses a slightly different attention mechanism from SADRE, and will be described in Sec. 3.1.3. In SADRE (**S**elf-**A**ttention-based **D**rug **RE**sponse), the attention weights $\alpha_1, \alpha_2, ..., \alpha_m$ are the outputs of all the gene embeddings $\vec{e}_1, \vec{e}_2, ..., \vec{e}_m$ using a Self-Attention function:

$$\alpha_1, \alpha_2, ..., \alpha_m = \text{Self-Attention}\left(\vec{e}_1, \vec{e}_1, ..., \vec{e}_m\right). \tag{3.7}$$

This self-attention function captures the contextual impact of other expressed genes when we calculate the weight of i-th gene α_i: it can not be solely calculated using $\vec{e}_i$. We implemented the Self-Attention function via a sub neural network (Fig. 3.1D; 258), which first calculates the unnormalized attention weights:

$$\beta_{i,j} = \vec{\theta}_j^\intercal \tanh(W\vec{e}_i), \qquad i = 1, 2, ..., m, \ j = 1, 2, ..., h, \tag{3.8}$$

where $W \in \mathbb{R}^{q \times s}$, $\vec{\theta}_j \in \mathbb{R}^q$ are trainable model parameters. Then we normalize them:

$$\alpha_{1,j}, \alpha_{2,j}, ...\alpha_{m,j} = \text{softmax}(\beta_{1,j}, \beta_{2,j}, ...\beta_{m,j}), \qquad j = 1, 2, ..., h, \tag{3.9}$$

where h is the number of attention heads (discussed at the end of this paragraph). The softmax normalization is the key step of attention mechanisms to capture the interactions of all the input genes. The softmax function is defined as:

$$\alpha_{i,j} = \exp\left(\beta_{i,j}\right) \Big/ \sum\nolimits_{i'=1}^{m} \exp\left(\beta_{i',j}\right) , \qquad i = 1, 2, ..., m. \tag{3.10}$$

The final weights of the self-attention mechanism are:

$$\alpha_i = \sum\nolimits_{j=1}^{h} \alpha_{i,j} = \alpha_{i,1} + \alpha_{i,2} + ... + \alpha_{i,h}, \qquad i = 1, 2, ..., m. \tag{3.11}$$

Note that in Eq. 3.8-3.11 we implemented the multi-head self-attention mechanism with h attention heads, instead of single-head self-attention. We used multi-head because single-head often pays most weight to a single gene embedding, while in practice, a few genes might be all-important, which could be selected through multiple independent heads, thus improving both the performance and interpretability of the model.

3.1.3 CADRE: Contextual attention-based drug response

Although self-attention was already able to capture the contextual effects from other expressed genes to encode the cell line embedding, we hypothesized that by integrating the contextual information of drug targets, we could further improve the performance, leading to CADRE (**C**ontextual **A**ttention-based **D**rug **RE**sponse). Given the drug d and its target pathway p, instead of just using gene embeddings to calculate attention weights (Eq. 3.7), CADRE uses target pathway embedding $\vec{e}_p$ as well:

$$\alpha_1, \alpha_2, ..., \alpha_m = \text{Contextual-Attention}\left(\vec{e}_1, \vec{e}_1, ..., \vec{e}_m, \vec{e}_p\right). \tag{3.12}$$

All the steps to calculate attention weights in CADRE/Contextual-Attention are the same as SADRE/Self-Attention (Eq. 3.9-3.11), except Eq. 3.8, where CADRE calculates the unnormalized contextual-attention weights in the following way (Fig. 3.1D):

$$\beta_{i,j} = \vec{\theta}_j^\intercal \tanh(W\vec{e}_i + \vec{e}_p), \qquad i = 1, 2, ..., m, \ j = 1, 2, ..., h, \tag{3.13}$$

where $\vec{e}_p \in \mathbb{R}^q$ is the target pathway embedding of the pathway p. We mapped it from the lookup table of pathway embeddings $\mathcal{E}_\mathcal{P} = \{\vec{e}_p\}_{p \in \mathcal{P}}$, where $\mathcal{P}$ is the set of all possible pathways. Essentially, the drug target embedding $\vec{e}_p$ reflects the functional similarities of drugs, i.e., if two drugs share the same target, their target embeddings should be similar, leading to a similar function to calculate the attention weights from gene embeddings (Eq. 3.13). It is possible to directly use the drug embeddings here by replacing $\vec{e}_p$ with $V\vec{e}_d$ in Eq. 3.13, where $V \in \mathbb{R}^{q \times s}$ is a trainable model parameter. However, we did not find that this significantly improves performance.

3.1.4 Pretraining gene embeddings

Transfer learning research in the area of natural language processing and computational biology showed that word embeddings [153, 218] or gene embeddings [221] pretrained on large-scale external unlabeled datasets could improve related supervised learning tasks. To integrate the external knowledge of the co-expression pattern of genes, we utilized gene embeddings $\mathcal{E}_{\mathcal{G}} = \{\vec{e}_g\}_{g \in \mathcal{G}}$ pretrained on a large-scale Gene Expression Omnibus (GEO; 13) database. It is also possible to utilize other genetic association databases [241]. The 200-dimensional gene embeddings were pretrained using the gene2vec algorithm [65], which is a variant of the word2vec algorithm [153] in the scenario of gene expression. The co-expressed genes would also be close in the pretrained gene embedding space, and thus had a similar effect to the model. The full CADRE model directly used the fixed pretrained gene embeddings $\mathcal{E}_{\mathcal{G}}$ and did not optimize them at the time of training. If gene embeddings were randomly initialized and trained, the model reduced to CADREΔpretrain. To make a fair comparison, we also used fixed pretrained gene embeddings in the collaborative filtering and SADRE models.

3.1.5 Implementation and training procedure

We implemented the CADRE model using PyTorch [176]. We trained it using the one cycle policy with momentum gradient descent on an AWS GPU instance `g4dn.12xlarge` [209]. To facilitate a balanced training process, we used a training batch size of $8 \times |\mathcal{D}|$, where $\mathcal{D}$ is the set of all the drugs. One cycle policy enabled fast convergence while preventing overfitting (superconvergence) by adjusting the learning rate and momentum. In the first 45% training steps (warm-up), we increased the learning rate linearly from $\eta/10$ to η, and decreased the momentum linearly from 0.95 to 0.85. In the following 45% training steps (cool-down), we decreased the learning rate linearly from η to $\eta/10$, and increased the momentum linearly from 0.85 to 0.95. In the last 10% training steps (annihilation), we decreased the learning rate linearly from $\eta/10$ to $\eta/100$, while keeping the momentum as 0.95. See Sec. 3.2.1 for our tuning and evaluation protocols.

3.1.6 Datasets

Cohort selection

We focused on two large-scale pharmacogenomic datasets: GDSC [257] and CCLE [12]. Both datasets included drug response data between hundreds of cell lines and tens/hundreds of anti-cancer drugs. We extracted the transcriptome data, i.e., the expression profiles of around 20k genes, as they were shown in previous multi-omics research to be the most dominant features compared with genomic and epigenomic data [39, 202]. We also collected and summarized the targeted pathways of drugs of both datasets. Interested readers may refer to the original papers for more detailed characteristics of the two datasets [12, 257].

Drug sensitivity discretization and missing value imputation

GDSC and CCLE released both the half-maximal inhibitory concentration (IC50) and area under the curve (AUC)/activity area (AA) as the single continuous response value for each pair of cell line and drug. We discretized the AA into two categories of sensitive (one) vs. resistant (zero) using the waterfall algorithm for each drug [12], following parameters in the previous work [61]. Continuous IC50 [39] and continuous AA [12] were also widely used drug sensitivity measurements in the literature. However, AA is a more robust metric of sensitivity compared with IC50, which is crucial in our case since drug sensitivity data are usually noisy. In addition, binary sensitivity has a better clinical significance compared with the continuous one.

Although most of the response data of CCLE are available, 20% of response entries in the GDSC dataset are missing. At the time of training the model, if the sensitivity of a cell line to a drug was missing, we filled the missing value with the mode of the available sensitivities to this specific drug. At the time of evaluation, we skipped the missing values. There might exist alternative strategies, such as imputation with the mode of the k-nearest neighbors [19]. Another potential solution would be to modify the models by using a mask at the training phase to omit the unknown objective loss that resulted from them. However, our preliminary experiments found that filling the values at the training stage could improve the performance on the validation set slightly.

Gene expression data preprocessing

We downloaded the RNA expressions of cell lines in both GDSC and CCLE datasets. We calculated the variance of each gene using the quantile-normalized values in log-scale, and selected the 3,000 genes that had the largest variances. Within each cell line, we then annotated the top 1,500 highly expressed genes with ones, and the remaining genes with zeros. The parameters 3,000 and 1,500 were inherited from previous research [61].

Drug target pathway extraction

We directly extracted the “target pathway” of each drug in the GDSC dataset. For the CCLE dataset, 15 out of its 24 drugs were shared with the GDSC dataset. Therefore, we used the same “target pathways” values for these 15 drugs in CCLE. For the remaining 9 drugs, we used their “class” as target pathways.

3.2 Results

3.2.1 Evaluation approach

We trained and evaluated the models on the two datasets separately. For each dataset, we split the cell lines into three parts: training, validation, and test sets with a ratio of 60%, 20%, and 20%. All the models compared in this chapter shared the same split of datasets. We manually tuned all the models using the training and validation sets to optimize the overall F1 score. After we tuned the hyperparameters, we finally trained the models on the training and validation sets, and evaluated on the test set. Since the outputs of the models were slightly different across each run, we retrained and evaluated three times, and reported the mean and standard deviation. At the time of evaluation, we utilized multiple metrics in addition to the F1 score, including accuracy, area under the precision-recall curve (AUPR), and area under ROC (AUROC). AUPR and AUROC are more comprehensive evaluation metrics, which take as input the predicted probability instead of binary predictions, in contrast to the F1 score and accuracy.

Table 3.1: Performance of different models and variants on the GDSC and CCLE datasets. We calculated the means and standard deviations from three repeated experiments. CADRE, SADRE, and collaborative filtering each utilized pretrained gene emebeddings. CADRE outperformed all the other competing models using four different metrics, validating its superior performance from the contextual attention mechanism and pretrained gene embeddings.

Dataset	Model	F1 Score	Accuracy	AUPR	AUROC
GDSC	DeepDR	62.1±0.55	78.4±0.74	67.2±0.91	81.8±0.92
	Collaborative filtering	60.9±0.18	76.9±0.37	67.0±1.36	81.2±0.13
	SADRE	62.9±0.20	78.2±0.25	68.7±1.43	82.9±0.29
	CADREΔpretrain	62.6±0.64	77.4±0.25	69.8±2.53	82.3±0.46
	CADRE	**64.3**±0.22	**78.6**±0.34	**70.6**±1.30	**83.4**±0.19
CCLE	DeepDR	51.5±2.95	77.8±0.39	53.5±2.80	78.5±1.09
	Collaborative filtering	48.6±1.26	77.6±0.42	48.9±1.10	78.6±0.52
	SADRE	50.1±1.61	77.5±0.31	56.4±2.02	78.7±0.80
	CADREΔpretrain	50.8±1.95	76.6±0.71	49.7±2.85	74.1±0.98
	CADRE	**54.4**±2.15	**79.1**±0.56	**60.0**±2.43	**80.9**±0.25

3.2.2 CADRE outperforms competing models

We compared the overall performance of CADRE (Sec. 3.1.3) with other competing models and its ablated variants, including DeepDR [39], vanilla collaborative filtering (Sec. 3.1.1), SADRE (Sec. 3.1.2), and CADREΔpretrain (CADRE without pretrained gene embeddings; Sec. 3.1.4). DeepDR was previously shown to outperform both simple neural networks and classical models such as linear regression and SVM [39]. Note that CADRE, SADRE, and collaborative filtering each utilized pretrained gene embeddings in our experimental setting. DeepDR did not use pretrained gene embeddings, since we did not find it helpful in the preliminary experiments. As one can see from Tab. 3.1, CADRE outperforms all these other models and variants on both GDSC and CCLE datasets. The attention mechanisms can significantly improve the performance of collaborative filtering. The contextual attention performs better than the self-attention, indicating the improved performance brought by the extra contextual information from the drug target pathway. The use of pretrained gene embeddings is also a crucial module for the superior performance of CADRE, validating the transferred knowledge from external databases.

3.2.3 Effective attention-encoded cell line representation contributes to the major improvements of performance

Using aggregated and flattened predictions and ground truth sensitivities, we showed the overall superior performance of CADRE over competing models (Tab. 3.1). However, we were also concerned about the disentangled performance of individual cell lines to all drugs (per-cell-line performance), or individual drugs to all cell lines (per-drug performance). Since different cell lines or drugs could have distinct performances, we compared the distributions of their performances instead of single averaged values. As one can see from Fig. 3.2, all the methods had comparable and reasonably high AUPR per cell line. However, the attention-based models such as SADRE and CADRE significantly outperformed other models in per-drug ARPR. This indicated that the major improvements of the attention-based models might come from the useful cell line representations built by attention mechanisms, such that the cell lines with various expression profiles were appropriately distinguished.

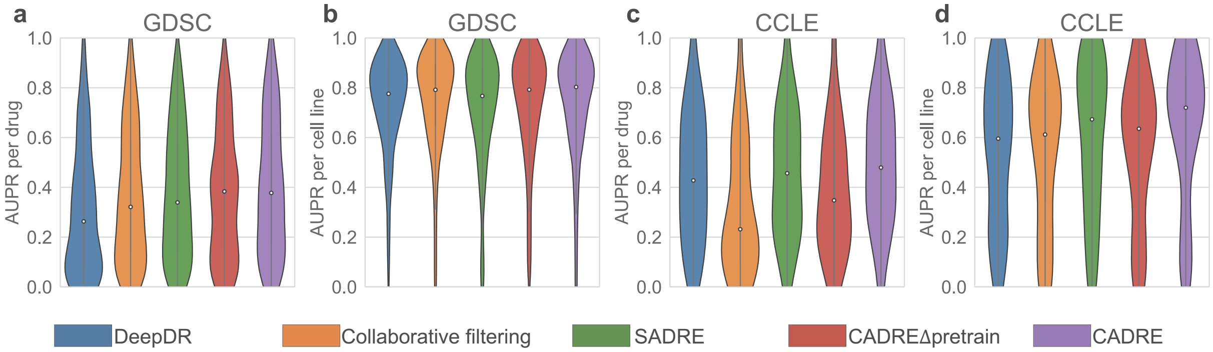

Figure 3.2: The distributions of dissected AUPR per cell line and AUPR per drug in different models on both GDSC and CCLE datasets. The per-cell-line performances of all models achieved reasonable high levels. Meanwhile, the major improvements of CADRE and SADRE came from the per-drug performance, indicating the well-designed attention mechanisms might encode the cell lines more effectively, so that different cell lines were more easily distinguished, leading to high per-drug AUPR.

To validate that attention-encoded cell line embeddings are more effective than non-attention-encoded vanilla cell line embeddings, we calculated the “correlation” of cell line embeddings and the origins of cell lines using “NN accuracy”. NN accuracy is defined as the following expectation:

$$\text{NN accuracy} = \mathbb{E}_{c':\vec{e}_{c'}=\text{NN}(\vec{e}_c)}\left[\text{Tissue}(c') = \text{Tissue}(c)\right], \tag{3.14}$$

where $\text{Tissue}(c)$ outputs the tissue of cell line c and $\text{NN}(\vec{e}_c)$ returns the closest gene embedding to $\vec{e}_c$ using unnormalized cosine similarity. We approximated this expectation by iterating c over all the cell lines in the dataset. The NN accuracy reflects how the distribution of cell line embeddings is consistent with their tissues. We focused on the GDSC dataset, since it had a larger sample size, and provided comprehensive annotations of the cell lines. As one can see from Tab. 3.2, the attention-based cell line embeddings had a better correlation with the tissue type (NN accuracy=36.1%), compared with vanilla cell line embeddings (NN accuracy=20.2%) and random cases (shuffled and repeated for five times; NN accuracy=12.9±2.1%). The t-SNE plot [232] of attention-encoded cell line embeddings revealed distinct clusters of embeddings within the same tissue, and similar embeddings across different tissues (Fig. 3.3A). In contract, non-attention-encoded cell line embeddings only reflected similarities of tissues (Fig. 3.3B), and did not discover the similar subgroups observed in attention-encoded embeddings.

Table 3.2: NN accuracy of differently encoded cell line embeddings with respect to tissue type. CADRE-encoded cell line embeddings improved the NN accuracy by around 80% compared with the embeddings encoded without attention, indicating the attention mechanism enabled the cell line embeddings to achieve a higher correlation with their corresponding tissues.

Cell line embedding	NN accuracy (%)
Random	12.9±2.1
Encoded w/o attention	20.2
Encoded w/ attention	**36.1**

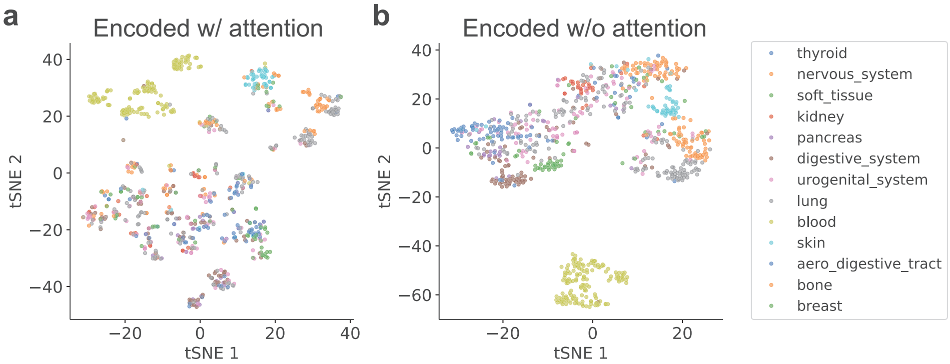

Figure 3.3: t-SNE visualization of cell line embeddings in the GDSC dataset. (A) CADRE-encoded cell line embeddings. The groups within the same tissue revealed the potential subtypes of the cancer that exhibited distinct drug response profiles. At the same time, even cell lines from different tissues can merge into the same clusters and thus share similar responses. (B) Cell line embeddings encoded without attention mechanism. These mainly reflected tissue-specific expression patterns rather than differences in response profiles, without finding subgroups of cell lines as in attention-encoded embeddings.

3.2.4 CADRE identifies the critical biomarkers related to drug resistance

CADRE assigned the heaviest weights to the genes that were most important to the drug response, providing a way to identify critical gene expression markers and biological processes. We extracted the attention weights from CADRE and counted expression frequency of the genes, and plotted the normalized attention weights vs. expression frequency. We shuffled the attention weights randomly 1,000 times to infer the significantly attended genes with a p-value threshold of 0.01 [265]. We mainly focused on the GDSC dataset due to its larger sample size. In general, the essential genes identified by CADRE are independent of the expression frequency (Fig. 3.4). The frequently expressed genes did not necessarily receive higher attention weights. We then conducted Gene Ontology (GO) enrichment analysis on these significant genes [152]. Two primary cell activities emerged (Tab. 3.3). First, functions related to exporting the molecules from the cells were enriched, which is consistent with previous research that many cancer cells acquired drug resistance by expelling the compounds using microvesicles [155]. Secondly, functions related to signaling receptor binding were enriched, reflecting the fact that a lot of anti-cancer drugs are targeted to the receptors of specific signaling pathways such as EGFR and RTK [257]. The clinically actionable genes of these CADRE-identified biomarkers could be potential targets of anti-cancer compounds for future exploration.

Table 3.3: Enriched biological functions of significantly weighted genes in GDSC dataset (red dots in Fig. 3.4). Genes related to intracelluar vesicles exporting compounds from cells, and signaling receptor bindings were heavily picked by CADRE.

GO domain	Enriched functions	FDR
biological process	export from cell	2.59e-3
biological process	secretion	2.84e-4
biological process	leukocyte activation	2.13e-2
molecular function	signaling receptor binding	6.24e-3
cellular component	intracelluar vesicle	9.64e-3
cellular component	vesicle lumen	4.79e-2
cellular component	extracellular region	1.05e-3

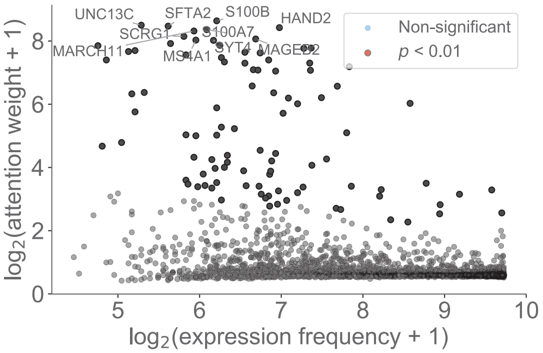

Figure 3.4: Landscape of the normalized contextual-attention weights of genes and their expression frequencies in the GDSC dataset. A higher expression frequency did not guarantee a higher attention weight.

3.3 Discussion

Personalized medicine in oncology requires that researchers and clinicians have tools to suggest effective anti-cancer drugs based on complex molecular profiles of the tumor and noisy pharmacogenomic assays, and to provide reasonable explanations for the recommendations. In pursuit of this goal, we created CADRE, an interpretable machine learning model that accurately predicted drug sensitivities of cancer cell lines from their expression levels. CADRE was built upon collaborative filtering, which is capable of dealing with noisy response assay data. The attention mechanism of CADRE improved both interpretability and performance of the model, by capturing the interactions and contextual effects of genes and drugs, and by encoding a better representation of cell lines from raw expression profiles. What is more, CADRE utilized gene representations transferred from an external database to boost its performance further. Through extensive evaluations and comparisons on the two primary pharmacogenomic datasets, we validated the superior performance of CADRE over competing models. Our results indicate that the genes assigned significant attention are involved in biological processes that can be expected to impact cellular responses to the presence of drugs. Thus these genes are potential novel biomarkers for designing more efficient test panels than whole-genome-scale sequencing.

Our model considered the simplified scenario of cancer cell lines, where each sample consists of only one cell population. However, a tumor tissue from a single cancer patient usually consists of multiple subpopulations, each exhibiting a distinct molecular profile [199, 222]. A clinically applicable anti-cancer drug resistance model should not only consider the inter-tumor differences between cell lines or patients (CADRE considered the similarities/differences between cell lines by grouping them into subtypes in the embedding space; Fig. 3.3A), but also take into account and deconvolve the intra-tumor heterogeneity of the cell populations within the same tumor tissue. Network matching [134] or single-cell techniques [125] could be promising directions in bridging both sides of *in vitro* cancer cell lines and *in vivo* tumors. A few other promising directions also warrant pursuing in the future. We mainly validated the effectiveness of CADRE using RNA expression data of cell lines in this chapter. However, we expect that models similar to CADRE, with slight modifications, could apply to genomic or epigenomic data, or the combination of these omic data in the future. Although we incorporated the knowledge of drugs through their target pathways, other drug representations such as drug embeddings well-represented from their structures, such as inferred from fingerprints or SMILES, might further improve our model [269]. Finally, since CADRE integrates pathway information to capture the contextual information, it would be helpful to conduct a case study of the essential genes captured by attention mechanism in specific drug target pathways, in addition to the aggregated analysis (Sec. 3.2.4; Fig. 3.4).

Chapter 4

Revealing tumor heterogeneity via neural network deconvolution[1]

Metastatic disease is the primary mechanism by which cancer results in patient mortality [33, 35]. By the time metastases have appeared, there are generally no viable treatment options [83]. Successful treatment thus depends on treating not just the primary tumor but also the seeds of metastasis that may linger after a seemingly successful remission. Identifying successful treatment options for metastasis is problematic, however, since the genomics of primary and metastatic tumors may be quite different even in single patients and metastatic cell populations may be poorly responsive to therapies effective on the primary tumor. Studies of cell-to-cell variation in cancers have revealed often substantial clonal heterogeneity in single tumors, with clonal populations sometimes dramatically shifting across progression stages [81]. Phylogenetic studies of clonal populations have been inconclusive on the typical evolutionary relationships between primary and metastatic tumors [199]. It remains a matter of debate whether changes in clonal composition occur primarily through ongoing clonal evolution, which results in novel clones with metastatic potential and resistance to therapy, or from selection on existing clonal heterogeneity already present at the time of first treatment [52, 60]. The degree to which either answer is true has important implications for prospects for early detection or prophylactic treatment of metastasis.

Brain metastases (BrMs) occur in around 10%–30% of metastatic breast cancers cases [128]. Although recent advances in the treatment of metastatic breast cancer have been able to achieve long-term overall survival, there are limited treatment options for BrMs and clinical prognoses are still disappointing [251]. Recent work examining transcriptomic changes between paired primary and BrM samples has demonstrated dramatic changes in expression programs over metastasis, including changes in tumor subtype with important implications for treatment options and

[1] This chapter was developed from material published in "Yifeng Tao, Haoyun Lei, Adrian V. Lee, Jian Ma, and Russell Schwartz. Phylogenies derived from matched transcriptome reveal the evolution of cell populations and temporal order of perturbed pathways in breast cancer brain metastases. In *Mathematical and Computational Oncology*, pages 3–28. Springer International Publishing, 2019" [219] and "Yifeng Tao, Haoyun Lei, Adrian V. Lee, Jian Ma, and Russell Schwartz. Neural network deconvolution method for resolving pathway-level progression of tumor clonal expression programs with application to breast cancer brain metastases. *Frontiers in Physiology*, 11:1055, 2020" [223].

prognosis [182, 235]. Some past research has sought to infer phylogenetic models to explain the development of brain metastases based on somatic genomic alterations [22, 112]. Such methods are challenged in drawing robust conclusions about recurrent progression processes, though, by the high heterogeneity within single tumors and across progression stages and patients. While single-cell methods are proving powerful for resolving such problems in other contexts [68, 184], such data is rarely available for studies of metastatic progression, which generally require working with samples archived years before metastases are discovered. Changes in the activity of particular genetic pathways or modules may provide a more robust measure of frequent genomic alterations across cancers.

In this chapter, we develop a strategy for tumor phylogenetics to explore how changes in clonal composition, via both novel molecular evolution and shifts in population dynamics of tumor clones and associated stroma, influence changes in expression programs across such progression stages. Our methods make use of multi-site bulk transcriptomic data to profile changes evident in gene expression programs between clones and progression stages. We break from past work in this domain in that we seek to study not clones *per se*, as is typical in tumor phylogenetics [66, 220], but what we dub “cell communities”: collections of clones or other stromal cell types that persist as a group with similar proportions across samples (Sec. 4.1.4). We accomplish this via a novel transcriptomic deconvolution approach designed to make use of multiple samples both within and between patients [200, 266] while improving robustness to inter- and intra-tumor heterogeneity by integrating deconvolution with pathway-based analyses of expression variation [174].

4.1 Materials and methods

4.1.1 Overview

Cell populations evolve due to genomic perturbations that can result in changes in the activity of various functional pathways between clones. Our overall method for deriving coarse-grained portraits of cell community evolution at the pathway level is illustrated by Fig. 4.1. After the preprocessing of transcriptome data (Sec. 4.1.2), the overall workflow consists of three main steps: First, the bulk expression profiles are mapped into the gene module and pathway space using external knowledge bases to reduce redundancy, noise, and sparsity, and to provide markers of expression variation for the subsequent analysis (Sec. 4.1.3). Second, a deconvolution step is implemented to resolve cell communities, i.e., coarse-grained mixtures of cell types presumed to represent an associated population of cancer clones and stromal cells, from the compressed pathway representation of samples (Sec. 4.1.4). Third, phylogenies of these cell communities are built based on the deconvolved communities as well as inferred ancestral (Steiner) communities to reconstruct likely trajectories of evolutionary progression by which cell communities develop — through a combination of genetic mutations, expression changes, and changes in population distributions — as a tumor progresses from healthy tissue to primary and potentially metastatic tumor (Sec. 4.1.5).

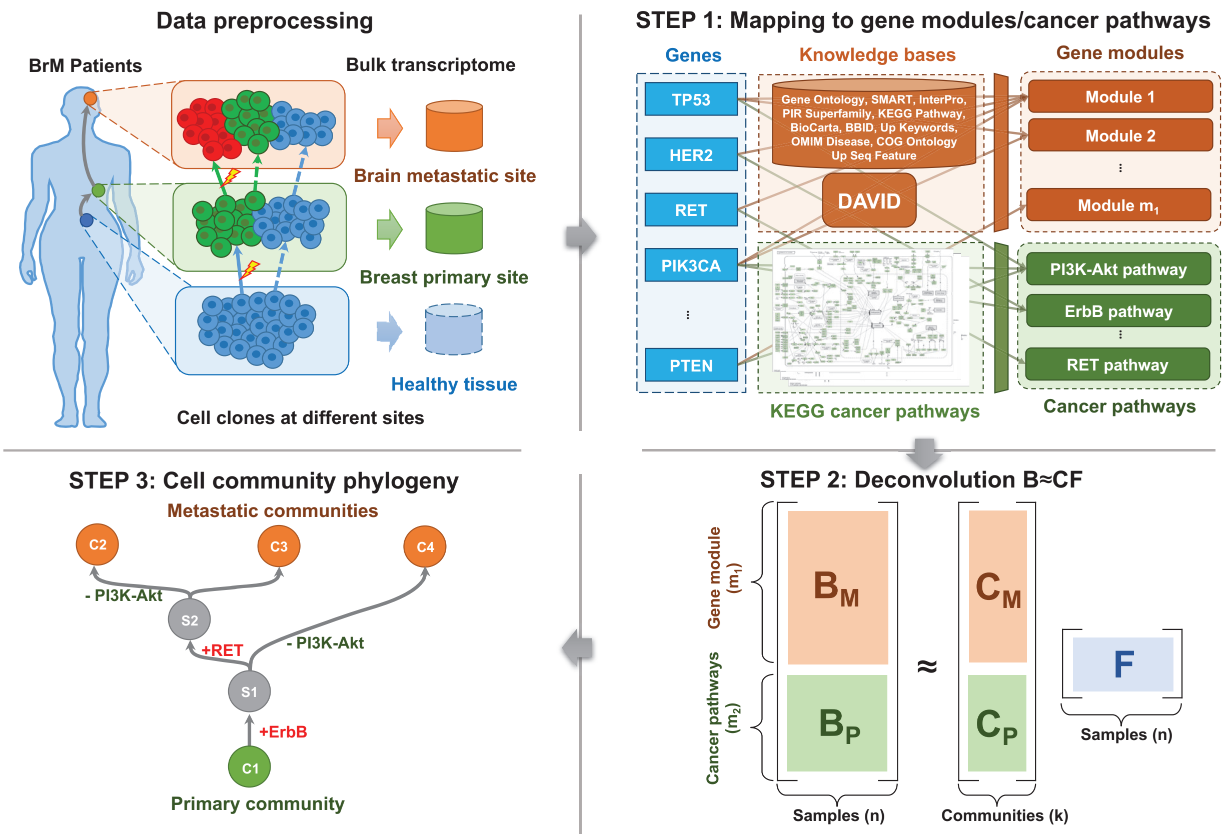

Figure 4.1: The pipeline of Brain Metastases (BrM) phylogenetics using matched bulk transcriptome.

4.1.2 Transcriptome data preprocessing

We applied our methods to raw bulk RNA-Sequencing data of 44 matched primary breast and metastatic brain tumors from 22 patients (each patient gives two samples) [182, 235], where six patients were from the Royal College of Surgeons (RCS) and sixteen patients from the University of Pittsburgh (Pitt). These data profiled the expression levels of approximately 60,000 transcripts. These can be represented in the format of a matrix, with rows corresponding to genes and columns to the samples (primary tumors or metastases). We removed the genes that are not expressed in any sample. We also considered only protein-coding genes in the present study. Approximately 20,000 genes remain after the filter. We conducted quantile normalization across samples using the geometric mean to remove possible artifacts [4]. The top 2.5% and bottom 2.5% of expressions were clipped to further reduce noise. Finally, we transformed the resulting bulk gene expression values into the log space and mapped those for each gene to the interval $[0, 1]$ by a linear transformation. The resulting preprocessed transcriptome data were used as the input of Step 1 (Sec. 4.1.3).

4.1.3 Mapping to gene modules and cancer pathways

The protein-coding gene expressions were mapped into both perturbed gene modules and cancer pathways, using the DAVID tool and external knowledge bases [94], as well as the cancer pathways in the KEGG database [105]. This step compresses the high dimensional data and provides markers of cancer-related biological processes (Fig. 4.1, Step 1). Note that although both gene module and cancer pathway representations capture recurrent features of metastatic progression, they serve different purposes in our analysis. Gene modules are an essential part of deconvolution in the following steps because they provide the major variance within the data. Cancer pathways serve primarily as probes for *post hoc* interpretation of the unmixed communities, but are biased relative to the gene module space by the focus only on genes with known relevance to cancer.

Gene modules Functionally similar genes are usually affected by a common set of somatic alterations [174] and therefore are co-expressed in the cells. These genes are believed to belong to the same "gene modules" [57]. Inspired by the idea of gene modules, we fed a subset of 3,000 most informative genes out of the approximately 20,000 genes that have the largest variances into the DAVID tool for functional annotation clustering using several databases [94]. DAVID maps each gene to one or more modules. We did not force the genes to be mapped into disjunct modules because a gene may be involved in several biological functions and therefore more than one gene module. We removed gene modules that were not enriched (fold enrichment < 1.0) and kept the remaining $m_1 = 109$ modules (and the corresponding annotated functions), where fold enrichment is defined as the EASE score of the current module to the geometric mean of EASE scores in all modules [92]. The gene module values of all the $n = 44$ samples were represented as a gene module matrix $\mathbf{B}_M \in \mathbb{R}^{m_1 \times n}$. The i-th gene module value in j-th sample, $(\mathbf{B}_M)_{i,j}$, was calculated by taking the sum of expressions of all the genes in the i-th module. Then $\mathbf{B}_M$ was rescaled row-wise by taking the z-scores across samples to compensate for the effect of variable module sizes.

Cancer pathways Although the gene module representation is able to capture the variances across samples and reduce the redundancy of raw gene expressions, it has two disadvantages. The first is a lack of interpretability. Specifically, some annotations assigned by DAVID are not directly related to biological functions, and the annotations of different modules may substantially overlap. The second is that the key perturbed cancer pathways or functions may not always be the ones that vary most across samples. For example, genes in cancer-related KEGG pathways (hsa05200; Kanehisa and Goto [105]) are not especially enriched in the top 3,000 genes with the largest expression variances. To make better use of prior knowledge on cancer-relevant pathways, we supplemented the generic DAVID pathway sets with a KEGG "cancer pathway" representation of samples $\mathbf{B}_P \in \mathbb{R}^{m_2 \times n}$, where the number of cancer pathways $m_2 = 24$. The cancer-related pathways in the KEGG database are cleaner and easier to explain, more orthogonal to each other, and contain critical signaling pathways to cancer development. We extracted the 23 cancer-related pathways from the following 3 KEGG pathway sets: *Pathways in cancer* (hsa05200), *Breast cancer* (hsa05224), and *Glioma* (hsa05214). An additional cancer pathway *RET pathway* was added, since it was found to be recurrently gained in the prior research [235]. See y-axis of Fig. 4.4D for the complete list of 24 cancer pathways. We considered all the $\sim$20,000 protein-coding genes other than top 3,000 genes. The following mapping of cancer pathways and transformation to z-scores were similar to that we did to map the gene modules.

Until this step, the raw gene expressions of n samples were transformed into the compressed gene module/pathway representation of samples $\mathbf{B} = \left[\mathbf{B}_M^\intercal, \mathbf{B}_P^\intercal\right]^\intercal \in \mathbb{R}^{m \times n}$, where $m = m_1 + m_2$. The gene module representation $\mathbf{B}_M$ serves for accurately deconvolving and unmixing the cell communities, while the pathway representation $\mathbf{B}_P$ serves as markers/probes and for interpretation purpose.

4.1.4 Deconvolution of bulk data

We applied a type of matrix factorization (MF) with constraints on the pathway-level expression signatures to deconvolve the communities/populations from primary and metastatic tumor samples (Fig. 4.1, Step 2) [113]. Note that common alternatives, such as principal components analysis (PCA) and non-negative matrix factorization (NMF) are not amenable to this case [122], since PCA does not provide a feasible solution to the constrained problem, and the NMF does not apply to our mixture data, which can be either positive or negative.

Cell communities We define a cell community to be a set of clones/clonal subpopulations and other cell types that propagate as a group during the evolution of a tumor. A community may be just a single subpopulation/clone, but is a more general concept in the sense that it usually involves multiple related clones and their associated stroma. For example, a set of immunogenic clones and the immune cells infiltrating them might collectively form a community that has a collective expression signature mixing signatures of the clones and associated immune cells, even if the individual cell types are not distinguishable from bulk expression data alone. While much work in this space has classically aimed to separate individual clones, or perhaps individual cell types more broadly defined, we note that deconvolution may be unable in principle to resolve distinct cell types if they are always co-located in similar proportions. It is particularly true when data is sparse and cell types are fit only approximately, as in this chapter, that a model with large complexity to deconvolve the fine-grained populations is prone to overfit. The community concept is intended in part to better describe the results we expect to achieve from the kind of data examined here and in part because identifying these communities is itself of interest in understanding how tumor cells coevolve with their stroma during progression and metastasis. Single-cell methods may provide an alternative, but are not amenable to preserved samples such as are needed when retrospectively studying primary tumors and metastases that may have been biopsied years apart.

Formulation of deconvolution With a matrix of bulk pathway values $\mathbf{B} \in \mathbb{R}^{m \times n}$, the deconvolution problem is to find a component matrix $\mathbf{C} = \left[\mathbf{C}_M^\intercal, \mathbf{C}_P^\intercal\right]^\intercal \in \mathbb{R}^{m \times k}$ that represents the inferred fundamental communities of tumors, and the corresponding set of mixture fractions $\mathbf{F} \in \mathbb{R}_+^{k \times n}$:

$$\begin{aligned} \min_{\mathbf{C},\mathbf{F}} \quad & \|\mathbf{B} - \mathbf{CF}\|_{\text{Fr}}^2 , && && (4.1) \\ \text{s.t.} \quad & \mathbf{F}_{lj} \geq 0, && l = 1, ..., k,\ j = 1, ..., n, && (4.2) \\ & \sum\nolimits_{l=1}^{k} \mathbf{F}_{lj} = 1, && j = 1, ..., n, && (4.3) \end{aligned}$$

where $||\mathbf{X}||_{\text{Fr}}$ is the Frobenius norm. The column-wise normalization in Eq. 4.3 aims for recovering the biologically meaningful cell communities. In addition, they are equivalent to applying ℓ_1 regularizers and therefore enforce sparsity to the fraction matrix $\mathbf{F}$.

Neural Network Deconvolution Although it is possible to build new algorithms for solving MF by adapting previous work [122], the additional but necessary constraints of Eq. 4.2-4.3 make the optimization much harder to solve. For the problem of Eq. 4.1-4.3, one can prove that it does not generally guarantee convexity. A slightly modified version of the algorithm to solve NMF with constraints may guarantee neither good fitting nor convergence [124, 125]. Therefore, instead of revising existing MF algorithms, such as ALS-FunkSVD [17, 75, 113], we developed an algorithm which we call "neural network deconvolution" (NND) to solve the optimization problem using gradient descent. Specifically, the NND was implemented using backpropagation in the form of a neural network (Fig. 4.2A) with the PyTorch package (`https://pytorch.org/`) [109, 194], based on the revised constraints:

$$\begin{aligned} \min_{\mathbf{C},\mathbf{F}_{\text{par}}} \quad & \|\mathbf{B} - \mathbf{CF}\|_{\text{Fr}}^2 , && (4.4) \\ \text{s.t.} \quad & \mathbf{F} = \text{cwn}\left(|\mathbf{F}_{\text{par}}|\right), && (4.5) \end{aligned}$$

where $|\mathbf{X}|$ applies element-wise absolute value and $\text{cwn}\left(\mathbf{X}\right)$ is column-wise normalization, so that each column sums up to 1. The two operations of Eq. 4.5 naturally rephrase and remove the two constraints in Eq. 4.2-4.3, and meanwhile fit the framework of neural networks. An alternative to the absolute value operation $|\mathbf{X}|$ might be rectified linear unit $\text{ReLU}(\mathbf{X}) = \max\left(\mathbf{0}, \mathbf{X}\right)$. However, this activation function is unstable and leads to inferior performance in our case, since $\mathbf{X}_{lj}$ will be fixed to zero once it becomes negative and will lose the chance to get updated in the following iterations. One may also want to replace the column-wise normalization cwn $(\mathbf{X})$ with softmax operation softmax$(\mathbf{X})$. However, the nonlinearity introduced by softmax actually changes the original optimization problem Eq. 4.1-4.3 and the fitted $\mathbf{F}$ is therefore not sparse.

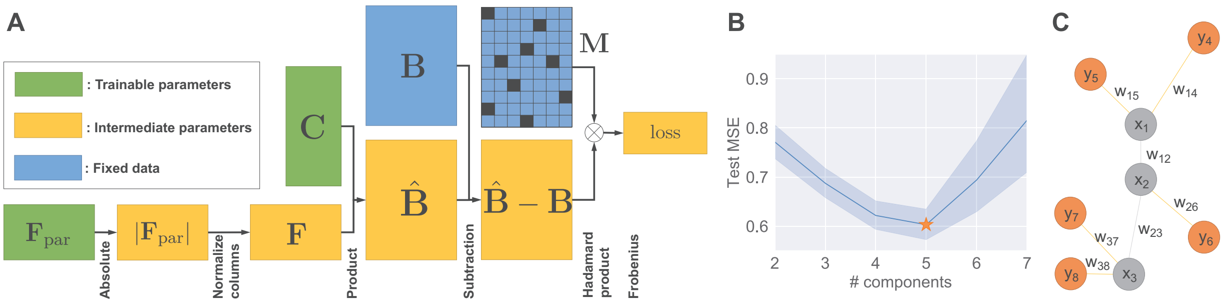

Figure 4.2: Method details. (A) Neural network architecture of NND. (B) Test errors of NND using 20-fold CV. Errors in unit of mean square error (MSE). (C) Illustration of a phylogeny with five extant nodes and three Steiner nodes.

Based on the revised NND optimization problem Eq. 4.4-4.5, we built the neural network with the architecture shown in Fig. 4.2A. An Adam optimizer other than vanilla gradient descent was used with default momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate of 1×10^{-5} [109]. The mini-batch technique is not required since the data size in our application is small enough not to require it ($\mathbf{B} \in \mathbb{R}^{m \times n}$, $m = 133$, $n = 44$). The training is run until convergence, which is defined as when the relative decrease of training loss is smaller than $\epsilon = 1 \times 10^{-10}$ every 20,000 iterations. This implementation has two main advantages: First, the method can be easily adapted to a wide range of optimization scenarios with various constraints, when existing methods do not or are hard to apply. Second, the NND has the flexibility of allowing for cross-validation, which is important for us in choosing the number of components k and preventing overfitting.

Cross-validation of NND In order to find the best tradeoff between model complexity and overfitting, we used cross-validation (CV) with the "masking" method to choose the optimal number of components/communities $k = 5$ that has the smallest test error (Fig. 4.2B). In each fold of the CV, we used estimated $\hat{\mathbf{B}}$ to only fit some randomly selected elements of $\mathbf{B}$, and then the test error was calculated using the other elements of $\mathbf{B}$. This was implemented by introducing two additional mask matrices $\mathbf{M}_{\text{train}}, \mathbf{M}_{\text{test}} \in \{0, 1\}^{m \times n}$, which are in the same shape of $\mathbf{B}$, and $\mathbf{M}_{\text{train}} + \mathbf{M}_{\text{test}} = 1^{m \times n}$. During the training time, with the same constraints in Eq. 4.5, the optimization goal is:

$$\min_{\mathbf{C}, \mathbf{F}_{\text{par}}} \|\mathbf{M}_{\text{train}} \odot (\mathbf{B} - \mathbf{C}\mathbf{F})\|_{\text{Fr}}^2 , \tag{4.6}$$

where $\mathbf{X} \odot \mathbf{Y}$ is the Hadamard (element-wise) product. At the time of evaluation, given optimized $\hat{\mathbf{C}}$, $\hat{\mathbf{F}}_{\text{par}}$, and therefore optimized $\hat{\mathbf{F}} = \text{cwn}\left(\left|\hat{\mathbf{F}}_{\text{par}}\right|\right)$ for the optimization problem during training, the test error was calculated on the test set: $\left\|\mathbf{M}_{\text{test}} \odot \left(\mathbf{B} - \hat{\mathbf{C}}\hat{\mathbf{F}}\right)\right\|_{\text{Fr}}^2$. We used 20-fold cross-validation on the NND, so in each fold 95% of positions of $\mathbf{M}_{\text{train}}$ and 5% of positions of $\mathbf{M}_{\text{test}}$ were 1s. Note that the actual number of cell populations is probably considerably larger than 5, and therefore each one of the five communities may contain multiple cell populations. Furthermore, it is likely that with sufficient numbers and precision of measurements, these communities could be more finely resolved into their constituent cell types. However $k = 5$ represents the largest hypothesis space of NND model that can be applied to the current dataset without severe overfitting.

4.1.5 Phylogeny of inferred cell subcommunities and pathway inference of Steiner nodes

We built "phylogenies" of cell subcommunities and estimated the pathway representation of unobserved (Steiner) nodes [136] inferred to be ancestral to them, with the goal of discovering critical communities that appear to be involved in the transition to metastasis and identifying the important changes of functions and expression pathways during this transition (Fig. 4.1, Step 3). Note that we are using the term "phylogeny" loosely here, as these trees are intended to capture evolution of populations of cells not just by accumulation of mutations from a single ancestral clone but also via changes in community structure, for example, due to generating or suppressing an immune response or migrating to a metastatic site. Although an abuse of terminology, we use the term phylogeny here due to the methodological similarity to more proper phylogenetic methods in wide use for analyzing mutational data in cancers [199].

Phylogeny of communities

Given the pathway profiles of the extant communities at the time of collecting tumor samples $\mathbf{C} \in \mathbb{R}^{m \times k}$, a phylogeny of the k extant cell communities was built using the neighbor-joining (NJ) algorithm [160], which inferred a tree that contains k extant nodes/leaves, $k-2$ unobserved Steiner nodes, and edges connecting two Steiner nodes or a Steiner node and an extant node. We estimated an evolutionary distance for any pair of two communities u, v as the input of NJ using the Euclidean distance between their pathway vectors $\|\mathbf{C}_{\cdot u} - \mathbf{C}_{\cdot v}\|_2$, similar to that in a prior work [174].

Inference of pathways: Setting and approach

Denote the phylogeny of cell subcommunities as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and $\mathcal{V} = \mathcal{V}_S \cup \mathcal{V}_C$, where the indices of Steiner node $\mathcal{V}_S = \{1, 2, ..., k-2\}$ ($|\mathcal{V}_S| = k-2$), the indices of extant nodes $\mathcal{V}_C = \{k-1, k, ..., 2k-2\}$ ($|\mathcal{V}_C| = k$). For each edge $(u, v) \in \mathcal{E}$, where $1 \leq u < v \leq 2k-2$, the first node of edge $u \leq k-2$ is always a Steiner node. The second node v can be either a Steiner node ($v \leq k-2$) or extant node ($v \geq k-1$). Denote the set of weights $\mathcal{W} = \{w_{uv} = 1/d_{uv} \mid (u, v) \in \mathcal{E}\}$ (inverse distance), where the edge length d_{uv} is the output of NJ. For each dimension i of the pathway vectors, we consider them independently and separately, so that each dimension of the Steiner nodes can be solved in the same way. Now let us consider the i-th dimension (and omit the subscript i for brevity) of extant nodes $\mathcal{V}_C$: $\mathbf{y} = [y_{k-1}, y_k, ..., y_{2k-2}]^\intercal = \mathbf{C}_{i\cdot}^\intercal \in \mathbb{R}^k$ and Steiner nodes $\mathcal{V}_S$: $\mathbf{x} = [x_1, x_2, ..., x_{k-2}]^\intercal \in \mathbb{R}^{k-2}$. Figure 4.2C illustrates a phylogeny where $k = 5$.

The inference of the i-th element in the pathway vector of the Steiner nodes can be formulated as minimizing the following elastic potential energy $U(\mathbf{x}, \mathbf{y}; \mathcal{W})$:

$$\min_{\mathbf{x}} \quad U(\mathbf{x}, \mathbf{y};\ \mathcal{W}) = \sum_{\substack{(u,v)\in\mathcal{E}\\ v\leq k-2}} \frac{1}{2} w_{uv}(x_u - x_v)^2 + \sum_{\substack{(u,v)\in\mathcal{E}\\ v\geq k-1}} \frac{1}{2} w_{uv}(x_u - y_v)^2, \tag{4.7}$$

which can be rephrased as a quadratic programming problem and solved easily, as we show below.

Inference of pathways: Derivation of quadratic programming, $\mathbf{P}(\mathcal{W})$, and $\mathbf{q}(\mathcal{W}, \mathbf{y})$

Theorem Equation 4.7 can be further rephrased as a quadratic programming problem:

$$\min_{\mathbf{x}} \quad \frac{1}{2}\mathbf{x}^{\intercal}\mathbf{P}(\mathcal{W})\mathbf{x} + \mathbf{q}(\mathcal{W}, \mathbf{y})^{\intercal}\mathbf{x}, \tag{4.8}$$

where $\mathbf{P}(\mathcal{W})$ is a function that takes as input edge weights $\mathcal{W}$ and outputs a matrix $\mathbf{P} \in \mathbb{R}^{(k-2)\times(k-2)}$, $\mathbf{q}(\mathcal{W}, \mathbf{y})$ is a function that takes as input edge weights $\mathcal{W}$ and vector $\mathbf{y}$ and outputs a vector $\mathbf{q} \in \mathbb{R}^{k-2}$.

Proof Based on Eq. 4.7, $U(\mathbf{x}, \mathbf{y}; \mathcal{W}) \geq 0$. Each term inside the first summation ($v \leq k-2$) can be written as:

$$\frac{1}{2} w_{uv}(x_u - x_v)^2 = \frac{1}{2}\mathbf{x}^{\intercal}\mathbf{P}(w_{uv})\mathbf{x}, \tag{4.9}$$

where

$$\mathbf{P}(w_{uv}) = \begin{array}{r|ccccc|} & & u\text{-th col} & & v\text{-th col} & \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ u\text{-th row} & \mathbf{0} & w_{uv} & \mathbf{0} & -w_{uv} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ v\text{-th row} & \mathbf{0} & -w_{uv} & \mathbf{0} & w_{uv} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}. \tag{4.10}$$

Each term ($v \geq k-1$) inside the second summation can be rephrased as:

$$\frac{1}{2} w_{uv}(x_u - y_v)^2 = \frac{1}{2}\mathbf{x}^{\intercal}\mathbf{P}(w_{uv})\mathbf{x} + \mathbf{q}(w_{uv}, y_v)^{\intercal}\mathbf{x} + C(w_{uv}, y_v), \tag{4.11}$$

where

$$\mathbf{P}(w_{uv}) = \begin{array}{r|ccccc|} & & u\text{-th col} & & & \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ u\text{-th row} & \mathbf{0} & w_{uv} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}, \quad \mathbf{q}(w_{uv}, y_v) = \begin{array}{r|c|} & \mathbf{0} \\ u\text{-th row} & -w_{uv}y_v \\ & \mathbf{0} \\ & \mathbf{0} \\ & \mathbf{0} \end{array}, \tag{4.12}$$

and $C(w_{uv}, y_v) = \frac{1}{2}w_{uv}y_v^2$ is independent of $\mathbf{x}$. Therefore the optimization in Eq. 4.7 can be calculated and written as below:

$$\min_{\mathbf{x}} \quad \sum_{\substack{(u,v)\in\mathcal{E}\\ v\leq k-2}} \frac{1}{2}\mathbf{x}^\intercal\mathbf{P}(w_{uv})\mathbf{x} + \sum_{\substack{(u,v)\in\mathcal{E}\\ v\geq k-1}} \left(\frac{1}{2}\mathbf{x}^\intercal\mathbf{P}(w_{uv})\mathbf{x} + \mathbf{q}(w_{uv}, y_v)^\intercal\mathbf{x}\right), \tag{4.13}$$

$$\Leftrightarrow \min_{\mathbf{x}} \quad \frac{1}{2}\mathbf{x}^\intercal\left(\sum_{\substack{(u,v)\in\mathcal{E}\\ v\leq k-2}} \mathbf{P}(w_{uv}) + \sum_{\substack{(u,v)\in\mathcal{E}\\ v\geq k-1}} \mathbf{P}(w_{uv})\right)\mathbf{x} + \sum_{\substack{(u,v)\in\mathcal{E}\\ v\geq k-1}} \mathbf{q}(w_{uv}, y_v)^\intercal\mathbf{x}, \tag{4.14}$$

$$\Leftrightarrow \min_{\mathbf{x}} \quad \frac{1}{2}\mathbf{x}^\intercal\mathbf{P}(\mathcal{W})\mathbf{x} + \mathbf{q}(\mathcal{W}, \mathbf{y})^\intercal\mathbf{x}. \quad \square \tag{4.15}$$

Remark The optimal $\mathbf{x}^\star$ of the Eq. 4.7, or the solution to the quadratic programming problem Eq. 4.8 can be solved by setting the gradient to be $\mathbf{0}$:

$$\mathbf{P}(\mathcal{W})\mathbf{x}^\star + \mathbf{q}(\mathcal{W}, \mathbf{y}) = \mathbf{0}. \tag{4.16}$$

Therefore,

$$\mathbf{x}^\star = -\mathbf{P}(\mathcal{W})^{-1}\mathbf{q}(\mathcal{W}, \mathbf{y}). \tag{4.17}$$

Remark Based on the proof, we can derive how to calculate the matrix $\mathbf{P}(\mathcal{W})$ and vector $\mathbf{q}(\mathcal{W}, \mathbf{y})$.

Initialize the matrix and vector with zeros:

$$\mathbf{P} \leftarrow 0^{(k-2)\times(k-2)}, \quad \mathbf{q} \leftarrow 0^{k-2}. \tag{4.18}$$

For each edge $(u, v) \in \mathcal{E}$ with weight w_{uv}, there are two possibilities of nodes u and v: First, if both of them are Steiner nodes ($u \leq k-2$, $v \leq k-2$), we update $\mathbf{P}$ and keep $\mathbf{q}$ the same:

$$\mathbf{P}_{uu} \leftarrow \mathbf{P}_{uu} + w_{uv},\ \mathbf{P}_{vv} \leftarrow \mathbf{P}_{vv} + w_{uv},\ \mathbf{P}_{uv} \leftarrow \mathbf{P}_{uv} - w_{uv},\ \mathbf{P}_{vu} \leftarrow \mathbf{P}_{vu} - w_{uv}. \tag{4.19}$$

Second, if u is Steiner node and v is an extant node ($u \leq k-2$, $v \geq k-1$), we update both $\mathbf{P}$ and $\mathbf{q}$:

$$\mathbf{P}_{uu} \leftarrow \mathbf{P}_{uu} + w_{uv}, \quad \mathbf{q}_u \leftarrow \mathbf{q}_u - y_v \cdot w_{uv}. \tag{4.20}$$

We apply the same procedure to all dimension of pathways $i = 1, 2, ..., m$ to get the full pathway values for each Steiner node.

4.2 Results

4.2.1 NND deconvolves the bulk RNA accurately

Before we applied our deconvolution algorithm NND to the breast cancer brain metastatic samples, we first validated our algorithm on a semi-simulated dataset where the ground truth expressions and fractions of each cell clone in the mixture samples are known.

Semi-simulated GSE11103 dataset The semi-simulated dataset is based on the real data of pure clones from the GSE11103 dataset [1, 13]. Expression profiles of four different cells were measured using microarrays: Raji (B cell), IM-9 (B cell), THP-1 (monocyte), Jurkat (T cell). Each experiment was repeated three times. We took the average of the three replicates to get the expression data of the four pure cell clones. The top 300 genes that varied most across cell types were selected as the ground truth real data of pure cell clones: $\mathbf{C} \in \mathbb{R}_+^{300\times 4}$. We then created 100 mixture samples of the four pure clones *in silico* $\mathbf{B} \in \mathbb{R}_+^{300\times 100}$ by randomly generating the fraction matrix $\mathbf{F} \in \mathbb{R}_+^{4\times 100}$. The fraction matrix was generated in the following way:

$$\mathbf{F}_{lj} \leftarrow U(0,1), \qquad l = 1, ..., 4,\ j = 1, ..., 100, \tag{4.21}$$

$$\mathbf{F}_{lj} \leftarrow \frac{\mathbf{F}_{lj}}{\sum_{l'=1}^{4} \mathbf{F}_{l'j}}, \qquad j = 1, ..., 100, \tag{4.22}$$

where $U(0,1)$ is a uniform distribution in the interval $[0,1]$. The semi-simulated bulk expression matrix $\mathbf{B}$ was then generated from $\mathbf{C}$, $\mathbf{F}$, with a log-normal noise:

$$(\mathbf{B})_{ij} = (\mathbf{CF})_{ij} + 2^{\mathcal{N}\left(0,(s\sigma)^2\right)}, \qquad i = 1, ..., 300,\ j = 1, ..., 100, \tag{4.23}$$

where $\mathcal{N}\left(0,(s\sigma)^2\right)$ is a Gaussian distribution; s controls the noise level, which we set to 0, 0.4, 0.9, and 1.3 for test; σ is the standard deviation of $\log_2$-transformed original GSE11103 data.

Performance evaluation Given the bulk matrix $\mathbf{B}$, we applied NND and other two algorithms to infer the estimated $\hat{\mathbf{C}}$, $\hat{\mathbf{F}}$ and $\hat{\mathbf{B}} = \hat{\mathbf{C}}\hat{\mathbf{F}}$, and compared the accuracy between estimated and actual values using the following metrics. For $\mathbf{C}$, we used L_1 loss [275]:

$$L_1 \text{ loss}(\mathbf{C}) = \frac{\|\hat{\mathbf{C}} - \mathbf{C}\|_1}{\|\mathbf{C}\|_1}. \tag{4.24}$$

For $\mathbf{F}$ and $\mathbf{B}$, we used root mean square error (RMSE):

$$\text{RMSE}(\mathbf{F}) = \sqrt{\|\hat{\mathbf{F}} - \mathbf{F}\|_{\text{Fr}}^2}, \tag{4.25}$$

$$\text{RMSE}(\mathbf{B}) = \sqrt{\frac{\|\hat{\mathbf{B}} - \mathbf{B}\|_{\text{Fr}}^2}{\|\mathbf{B}\|_{\text{Fr}}^2}} \tag{4.26}$$

Different levels of noise s were added to test the robustness of models and the performance of different models under different conditions. We repeated all the experiments for 10 times to get the boxplot.

Competing algorithms There are two competing algorithms for the deconvolution problem. Geometric unmixing is an algorithm that borrows the intuition from computational geometry [200], which first identifies the corners of a simplex containing all the mixture sample points, and then infers the fraction matrix. However, the algorithm does not directly optimize the problem equation Eq. 4.1-4.3. Another intuitive algorithm is based on the popular multiplicative update (MU) rule that solves general NMF problem [122]: an additional update step of

$\mathbf{F}_{lj} \leftarrow \frac{\mathbf{F}_{lj}}{\sum_{l'=1}^{k}\mathbf{F}_{l'j}}, j = 1, ..., n$ can be added to the loop. Although the original MU rule guarantees the non-increasing of the objective function, this additional update step can lead to an increasing objective and we need to stop the iteration once this happened. Since the two competing algorithms work on non-negative space, we adapted the NND by adding an element-wise absolute value operator after the $\mathbf{C}$ in the network (Fig. 4.2A).

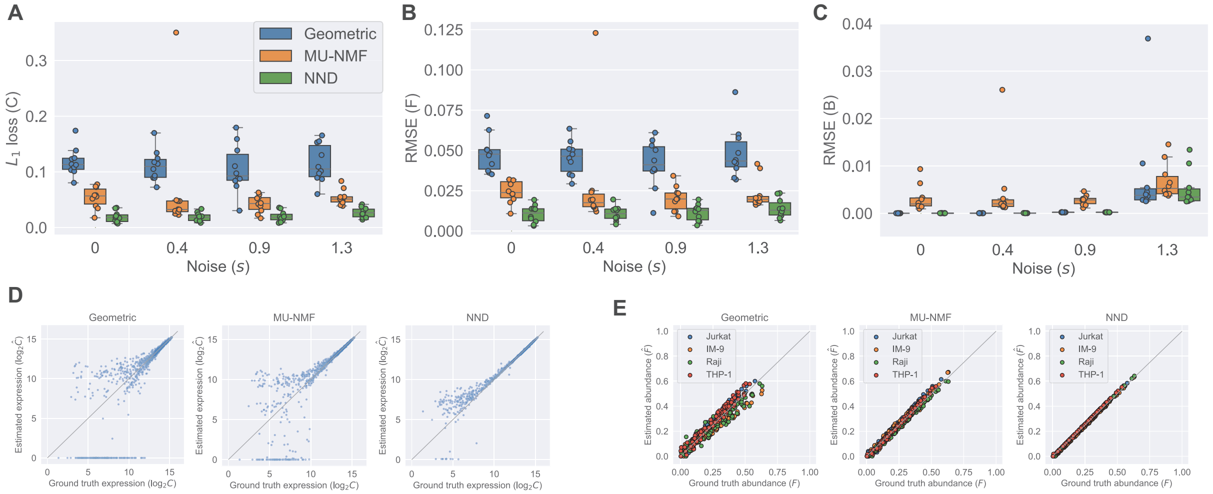

Figure 4.3: Comparison of NND and other algorithms on the semi-simulated GSE11103 dataset shows the better accuracy and robustness of NND algorithm. (A-C) Accuracies in estimated $\mathbf{C}$, $\mathbf{F}$ and $\mathbf{B}$ with three different algorithms. We tested four noise levels and repeated the experiments for ten replicates. (D) Estimated $\hat{\mathbf{C}}$ and ground truth $\mathbf{C}$ using three different algorithms with noise level s of 1.3. (E) Estimated $\hat{\mathbf{F}}$ and ground truth $\mathbf{F}$ using three algorithms with noise level s of 1.3.

Superiority of NND We show the results in Fig. 4.3. Figure 4.3A-C show the accuracies of both $\mathbf{C}$, $\mathbf{F}$, and $\mathbf{B}$ using the three algorithms under various noise levels. One can easily see that NND achieves lower L_1 loss of $\mathbf{C}$, RMSE of $\mathbf{F}$, and RMSE of $\mathbf{B}$. What is more, it is also much more robust than the geometric and MU-NMF algorithms, as there are fewer outliers that have huge errors. MU-NMF has a reasonable estimation accuracy of $\mathbf{C}$ and $\mathbf{F}$. However, its overall fitting ability is limited due to its non-convergence-guaranteed MU optimization algorithm. We can also visualize the estimation accuracy by plotting the estimated values and ground truth values at a specific noise level, as is shown in Fig. 4.3D-E. One can see the supriority of NND qualitatively over the other two algorithms in estimating expression profiles and fractions of individual pure clones.

4.2.2 Gene modules/Pathways provide an effective representation

Gene expressions of samples were mapped into the gene module and pathway space in order to reduce the noise of raw transcriptome data and reduce redundancy (Sec. 4.1.3). We verified that the gene module/pathway representation is effective in the sense that it captures distinguishing features of primary/metastatic sites and individual samples well and is able to identify recurrently gained or lost pathways.

Feature space of the gene module and pathway representation As one can see in Fig. 4.4A, the first principal component analysis (PCA) dimension of the gene module and pathway representation accounts for the difference between primary and metastatic samples, while the second and third PCA dimensions mainly capture variability between patients. This observation suggests the feasibility of using the gene module/pathway representation to distinguish recurrent features of metastatic progression across patients despite heterogeneity between patients. To make a direct comparison of the noise and redundancy between the gene module/pathway and raw gene expression representations, we applied hierarchical clustering to the 44 samples using Ward's minimum variance method [246]. Two hierarchical trees were built based on the two different representations (Fig. 4.4B). The gene module/pathway features more effectively separate the primary and metastatic samples into distinct clusters (Fig. 4.4B, right panel) than do the raw gene expression values (Fig. 4.4B, left panel). This is consistent with the PCA results that the largest mode of variance in the pathway representation distinguishes primary from metastatic samples. We do notice that in a few cases, matched primary and metastatic samples from the same patient are neighbors with pathway-based clustering. For example, 29P_Pitt:29M_Pitt and 51P_Pitt:51M_Pitt are grouped in the same clades using the pathway representation, showing that in a minority of cases, features of individual patients dominate over primary vs. metastatic features. Following previous work [174], we quantified the ability of the hierarchical tree to group the samples of the same labels using four metrics. 1. MSD: Mean square distance of edges that connect nodes of the same label (primary vs. metastatic). 2. z_{MSD}: The labels of all nodes were shuffled and the MSD is recalculated for 1,000 times to get the mean μ_{MSD} and standard deviation σ_{MSD}, which were used to get the z-score of the current assignment $z_{\text{MSD}} = (\text{MSD} - \mu_{\text{MSD}})/\sigma_{\text{MSD}}$. 3. rMSD: The ratio of MSD of edges that connect same label nodes and MSD of edges that connect distinct label nodes. 4. z_{rMSD}: as with MSD, a z-score of rMSD was calculated by shuffling labels for 1,000 times. Intuitively, the smaller values the MSD, z_{MSD}, rMSD, and z_{rMSD} are, the better is the feature representation at grouping same label samples together. The shortest paths and distances between all pairs of nodes were calculated using the Floyd-Warshall algorithm [71, 247]. All the edge lengths were considered as 1.0 to account for the different scales of pathway and gene representations. The pathway representation has significantly lower values for all four metrics (Tab. 4.1), indicating its strong grouping ability.

Table 4.1: Quantitative performance of hierarchical clustering.

Feature representation	MSD	rMSD	z_{MSD}	z_{rMSD}
Gene expression	99.62	0.93	-2.60	-2.57
Gene module/pathway	**86.23**	**0.66**	**-13.37**	**-11.42**

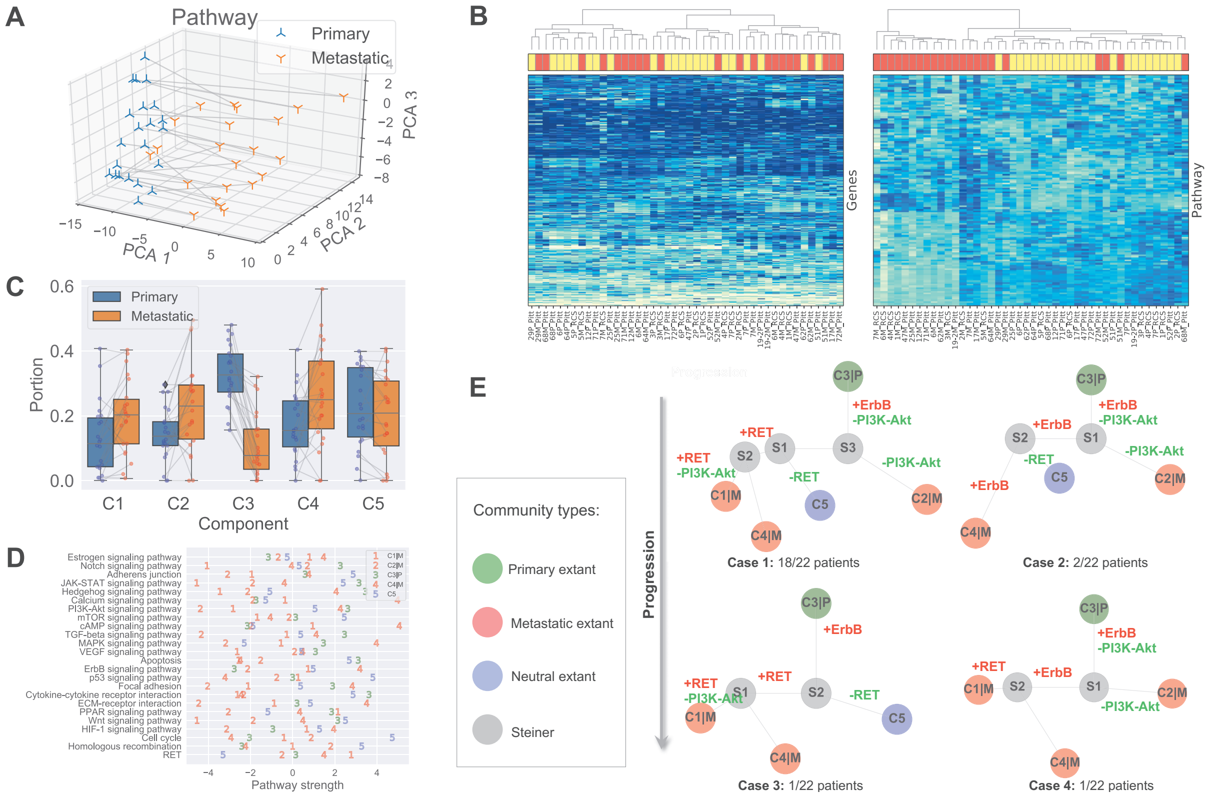

Figure 4.4: Results and analysis. (A) First three gene module/pathway representation PCA dimensions of matched primary and metastatic samples. Matched samples are connected. (B) Hierarchical clustering of tumor samples based on raw gene expressions (left panel) and compressed gene module/pathway representation (right panel). Metastatic samples are shown in red rectangles and primary ones in yellow. (C) Portions and changes of the five communities in primary and metastatic sites. Each gray line connects the portions of a community in the primary site (blue node) and metastatic site (red node) from the same patient. (D) Pathway strengths across cell communities. (E) Phylogeny of cell subcommunities.

Recurrently perturbed cancer pathways We next identified differentially expressed pathways in the primary and metastatic tumors using bulk data $\mathbf{B}_P \in \mathbb{R}^{24\times 44}$, prior to deconvolving cellular subcommunities. We conducted the Student's *t*-test followed by FDR correction on each of the 24 pathways. Eleven pathways are significantly different between the two sites (FDR<0.05; Tab. 4.2). The signaling pathways related to neurotransmitter and calcium homeostasis, including *cAMP* and *Calcium* [90], are enriched in metastatic samples, which we can suggest may reflect stromal contamination by neural cells in the brain metastatic samples. We also observed recurrent gains in *ErbB* pathway, as indicated by the primary studies [182, 235]. Three pathways related to immune activity are under-expressed in metastatic samples, including *Cytokine-cytokine receptor interaction* [123], *JAK-STAT* [123], and *Notch* [9], consistent with the previous inference of reduced immune cell expression in metastases in general and brain metastasis most prominently [274]. We can suggest that this result similarly may reflect expression changes in infiltrating immune cells, due to the immunologically privileged environment of the brain, rather than expression changes in tumor cell populations. Five other signalling pathways, including *Apoptosis* [252], *Wnt* [267], *Hedgehog* [84], *PI3K-Akt* [22], and *TGF-beta* [147], show reduction in metastatic samples and in each case, their loss or dysregulation has been reported to promote the tumor growth and brain metastasis. Note that the primary references for these data define pathways using the co-expression pattern of genes [182, 235], while our work uses external knowledge bases. Previous research also used somatic mutations or copy number variation to analyze perturbed genes [22, 182], while we focus exclusively on the transcriptome. Despite large differences in data types and pathway definitions, our observations are consistent with the prior analysis, especially with respect to variation in the *HER2/ErbB2* and *PI3K-Akt* pathways.

Table 4.2: Differentially expressed cancer pathways between primary and metastatic samples (FDR<0.05).

Gain/Loss after metastasis	Differentially expressed pathways	FDR
Relative gain	cAMP signaling pathway	6.88e-03
Relative gain	ErbB signaling pathway	2.09e-02
Relative gain	Calcium signaling pathway	4.39e-02
Relative loss	Cytokine-cytokine receptor interaction	4.37e-06
Relative loss	Apoptosis	8.53e-04
Relative loss	JAK-STAT signaling pathway	8.53e-04
Relative loss	Wnt signaling pathway	3.97e-03
Relative loss	Hedgehog signaling pathway	4.50e-03
Relative loss	PI3K-Akt signaling pathway	1.35e-02
Relative loss	TGF-beta signaling pathway	4.56e-02
Relative loss	Notch signaling pathway	4.56e-02

4.2.3 Landscape of deconvolved cell communities in tumors

We unmixed the bulk data $\mathbf{B}$ into five components using NND (Sec. 4.1.4). The deconvolution enables us to produce at least a coarse-grained landscape of major cell communities $\mathbf{C}$ and their distributions in primary and metastatic tumors $\mathbf{F}$. The number of components ($k = 5$) was chosen through 20-fold cross-validation (Sec. 4.1.4; Fig. 4.2B). Although the true heterogeneity of the samples may be much larger, we fit k to provide a balance between excessively coarse-grained communities if k is too small versus excessively high variance and thus unstable deconvolution if k is too large.

Community distributions across samples F The portions of the 5 components in all the 44 samples are represented as the mixture fraction matrix $\mathbf{F} \in \mathbb{R}^{5\times 44}$ (Fig. 4.4C). A primary or metastatic community is one inferred to change proportions substantially (magnitude>0.05) in the tumor samples after metastasis, or perhaps to be entirely novel to or extinct in the metastatic sample (denoted by a $|P$ or $|M$ suffix). Otherwise, the component is classified as a neutral community. Three components ($C1|M$, $C2|M$, $C4|M$) are classified as metastatic communities; one ($C3|P$) as primary; and one ($C5$) as neutral (Fig. 4.4C). Some components may be missing in both samples of some patients, e.g., $C1|M$, $C2|M$, $C5|M$ are absent in two, one, and one patient. We note that these five communities represent rough consensus clusters of cell populations inferred to occur frequently, but not universally, among the samples. Based on this rule, we can define four basic cases of patients in total. Twelve subcases can be found using a more detailed classification method based on the existence of communities in both primary and metastatic samples.

Pathway values of communities C We are especially interested in the pathway part $\mathbf{C}_P$ of the cell community inferences, since it serves as the marker and provides results easier to interpret. The pathway values of five subcommunities using $\mathbf{C}_P$ provides a much more fine-grained description of samples (Fig. 4.4D), compared with that in Sec. 4.2.2, which is only able to distinguish the differentially expressed pathways in bulk samples. As noted in Sec. 4.1.4, it is likely that true cellular heterogeneity is greater than the methods are able to discriminate and that communities inferred by our model may each conflate one or more distinct cell types and clones. We observe that the metastatic community $C4|M$ most prominently contributes to the enrichment for functions related to neurotransmitter and ion transport, since its strongest pathways (*cAMP*, *Calcium*) are greatly enriched relative to those of the other four communities. We might interpret this community as reflecting at least in part stromal contamination from neural cells specific to the metastatic site. $C4|M$ also contributes most to the gains of *ErbB* in brain samples. The metastatic subcommunity $C1|M$ is probably most closely related to the loss of immune response in metastatic samples as it has the lowest pathway values of *Notch*, *JAK-STAT*, and *Cytokine-cytokine receptor interaction*. This component might thus in part reflect the effect of relatively greater immune infiltration in the primary versus the metastatic site. $C1|M$ also has the lowest pathway values of *Apoptosis*, *Wnt*, and *Hedgehog*. The metastatic community $C2|M$ is most responsible for the loss of *PI3K-Akt* and *TGF-beta* pathways. We also note that although *RET* does not show up in the list of Tab. 4.2, it seems to be quite over-expressed in the metastatic communities $C1|M$ and $C4|M$ but not in the metastatic community $C2|M$.

4.2.4 Phylogenies of BrM communities reveal common order of perturbed pathways

We built phylogenies of cell communities and calculated the pathway representations of their Steiner nodes (Sec. 4.1.5). The phylogenies' topologies provide a way to infer a likely evolutionary history of cancer cell communities and thus their constitutive cell types. At the same time, the perturbed pathways along their edges suggest the order of genomic alterations or changes in community composition.

Topologically similar BrM phylogenies All five cell components do not appear in each BrM patient. We analyze the distribution of communities in each patient based on whether the community is inferred to be present in the patient. There are four different cases in general (Fig. 4.4E). Case 1: all five communities are found in the patient (majority; 18/22 patients). Case 2: only $C1|M$ missing (minority; 2/22). Case 3: only $C2|M$ missing (minority; 1/22). Case 4: only $C5$ missing (minority; 1/22). Although not all communities exist in Case 2-4, the topologies are similar to that of Case 1 and can be seen as special cases of Case 1, representing some inferred common mechanisms of progression across all the BrM patients.

Common order of altered cancer pathways After inferring the pathway values for Steiner nodes, the most perturbed pathways can also be found by subtracting the pathway vectors of nodes that share an edge. We focus on the top five gained or lost pathways along the evolutionary trajectories and the changes of magnitude larger than 1.0. We further examine those perturbed cancer pathways that were specifically proposed in the study that generated the data examined here, as well as others that are clinically actionable [22, 182, 235], i.e., *ErbB*, *PI3K-Akt*, and *RET* (Fig. 4.4E). As one may see from Case 1, the primary community $C3|P$ first evolves to community $S3$ by gaining expressions in *ErbB* and losing functions in *PI3K-Akt*. Then, if it continues to lose *PI3K-Akt* activity, it will evolve into the metastatic community $C2|M$. If it gains in *RET* activity, it will instead evolve into metastatic communities $C1|M$ and $C4|M$. The perturbed pathways along the trajectories of Cases 2-4 are similar to those of Case 1, with minor differences. We therefore draw to the conclusion that the evolution of BrMs follows a specific and common order of pathway perturbations. Specifically, the gain of *ErbB* reproducibly happens before the loss of *PI3K-Akt* and the gain of *RET*. Different subsequently perturbed pathways lead to different metastatic tumor cell communities. These inferences are consistent with the hypothesis that at least some major changes in expression programs between primary and metastatic communities occur by selecting for heterogeneity present early in tumor development rather than solely deriving from novel functional changes immediately prior to or after metastasis.

4.3 Discussion

Cancer metastasis is usually a precursor to mortality with no successful treatment options. Better understanding mechanisms of metastasis provides a potential pathway to identify new diagnostics or therapeutic targets that might catch metastasis before it ensues, treat it prophylactically, or provide more effective treatment options once it occurs. This chapter developed a computational approach intended to better reconstruct mechanisms of functional adaption from multisite RNA-Seq data to help us understand at the level of cancer pathways the mechanisms by which progression frequently proceeds across a patient cohort. Our method compresses expression data into a gene module/pathway representation using external knowledge bases, deconvolves the bulk data into putative cell communities where each community contains a set of associated cell types or subclones, and builds evolutionary trees of inferred communities with the goal of reconstructing how these communities evolve, adapt, and reconfigure their compositions across metastatic progression. Results on semi-simulated data show the method to yield improved accuracy in mixture deconvolution relative to prior deconvolution algorithms. We applied the pipeline to matched transcriptome data from 22 BrM patients and found that although there are slight differences of tumor communities across the cohort, most patients share a similar mechanism of tumor evolution at the pathway level. Specifically, the methods infer a fairly conserved mechanism of early gain of *ErbB* prior to metastasis, followed post-metastasis gain of *RET* or loss of *PI3K-Akt* resulting in intertumor heterogeneity between samples. Our methods provide a novel way of viewing the development of BrM with implications for basic research into metastatic processes and potential translational applications in finding markers or drug targets of metastasis-producing clones prior to the metastatic transition.

The results suggest several possible avenues for future development. In part, they suggest a need for better separating phylogenetically-related mixture components (i.e., distinct tumor cell clones) from unrelated infiltrating cell types (e.g., healthy stroma from the primary or metastatic site or infiltrating immune cells). The methods are likely finding only a small fraction of the true clonal heterogeneity of the tumors and stroma, and might benefit from algorithms capable of better resolution or from integration of multi-omics data (e.g., RNA-Seq, DNA-Seq, methylation) that might have complementary value in finer discrimination of cell types. The present methods are also using only a limited form of temporal constraint in considering a two-stage progression process and without use of quantitative time measurements. Models might be extended in future work to consider true time-series data, such as is becoming available through "liquid biopsy" technologies. In addition, we know of no data with known ground truth that models the kind of progression process studied here nor of other tools designed for modeling similar progression processes from expression data, leaving us reliant on validating based on consistency with prior research on brain metastasis [22, 182, 235]. Future work might compare to prior approaches for reconstruction of clonal evolution from expression data more generically [58, 191, 200] and seek replication on additional real or simulated expression data or artificial mixtures of different cell types [184] designed to mimic metastasis-like progression. The general approach might also have broader application than studying metastasis, for example in reconstructing mechanisms of other progression processes, such as pre-cancerous to cancerous, as well as to other tumor types or independent data sets. Finally, much remains to be done to exploit the translational potential of the method in better identifying diagnostic signatures and therapeutic targets, and what type of effective and safe clinical strategies can be taken to prevent metastasis at an early stage.

Chapter 5

Revealing tumor heterogeneity via robust and accurate deconvolution[1]

In this chapter, we build on our past work in developing deconvolution methods for understanding the metastatic transition (Chapter 4) [219]. Our contributions in this chapter are two-fold. Methodologically, we proposed and developed a tool kit called **R**obust and **A**ccurate **D**econvolution (RAD), the core algorithm of which takes bulk RNA as input, and infers the expressions and proportions of groups of associated tumor cells, which we denote *cell communities*. We refer readers to Sec. 5.1.2 and Sec. 5.2.3 for the major novelty and advancement of RAD, and the differences between RAD and previous deconvolution algorithms such as NMF [122], Geometric Unmixing [200], LinSeed [265], and NND [219]. We show that RAD can automatically identify correct numbers of cell populations and identify perturbed biomarkers, such as cancer pathways. We validated its superiority over alternative deconvolution algorithms through comprehensive validations on both simulated and real datasets. Biologically, we applied the RAD algorithm to transcriptomic data from matched breast cancer primary and metastatic samples [14, 182, 183, 274], extending our prior analysis of brain metastasis specifically [219] to consider variations across multiple metastatic sites. We further applied a refined phylogeny inference algorithm to trace the evolutionary trajectories from the primary tumor to different metastatic sites [219]. Our analysis showed that although the breast tumors of different metastatic types encompass heterogeneous and distinct cell populations, there exist common patterns of perturbed pathways detected at the early stage of primary breast cancer, suggesting that our framework might shed light on early diagnostic and treatment options.

[1] This chapter was developed from material published in "Yifeng Tao, Haoyun Lei, Xuecong Fu, Adrian V Lee, Jian Ma, and Russell Schwartz. Robust and accurate deconvolution of tumor populations uncovers evolutionary mechanisms of breast cancer metastasis. *Bioinformatics*, 36:i407–i416, jul 2020" [222].

5.1 Materials and methods

5.1.1 Overview

Figure 5.1 illustrates the general approach for unmixing bulk RNA from paired breast cancer metastatic samples and inferring underlying tumor evolutionary processes. To reduce the noise of bulk RNA, we first compress the expression levels of individual genes into the modules using external knowledge bases (Fig. 5.1A; Sec. 5.1.2). We then apply a novel robust and accurate deconvolution algorithm to unmix the compressed expression data into expression profiles and fractions of individual cell communities (Fig. 5.1B; Sec. 5.1.2). Finally, we infer the evolutionary trajectories of the unmixed tumor communities. We then analyze perturbed biomarkers, such as activity of cancer-related regulatory pathways, during tumor metastasis (Fig. 5.1C; Sec. 5.1.3).

Note that our overall goal is to deconvolve tumor *cell communities* [219], as opposed to necessarily single cells, and describe their evolution across the metastatic transition. A cell community is defined as one or more cell types with potentially distinct expression profiles that share common mixture proportions across samples, indicative of their possible interaction or co-association in the tumor. For example, a set of immunogenic tumor clones and the immune cell types infiltrating them may form a community exhibiting an overall expression pattern that is a mixture of those of its constituent cell types. Although our deconvolution algorithm can identify correct numbers of distinct clones when they are mixed in distinct proportions in different samples (Sec. 5.2.2), it is more proper to interpret results as describing cell communities when the sample size is small or when tumor cells may associate or coevolve with their stroma [16].

5.1.2 RAD: Toolkit for robust and accurate deconvolution

We proposed and developed the toolkit called **R**obust and **A**ccurate **D**econvolution: RAD. RAD is a set of tools that solves the problem of 1) estimating the number of cell communities from bulk RNA, 2) unmixing the cell communities from bulk data, and 3) inferring other biomarkers such as pathways from the deconvolved communities. Table 5.1 shows an example of applying RAD to a common RNA deconvolution problem.

RAD is different from the traditional non-negative matrix factorization (NMF; Lee and Seung [122]) widely used in this problem domain. First, RAD considers a biologically meaningful unmixing problem (Sec. 5.1.2), which has additional constraints that make it much harder to solve than the general NMF problem. Although there exist many NMF variants that consider different regularizations, such as NMF with ℓ_1-norm [203] and sparseness [93], these NMF algorithms mainly consider the prior of data distribution instead of the biological feasibility. As far as we know, the only algorithm that solves this specific deconvolution problem is our gradient descent-based NND method [219]. Algorithms derived from the widely used multiplicative update (MU) rules do not necessarily guarantee the convergence or accuracy in this new scenario [124, 125], a problem that RAD tackles by using a hybrid solver. Second, NMF is fragile since the unmixed matrices may be distinct given a different initialization seed. In contrast, RAD employs both the prior knowledge of gene modules and the minimum similarity criteria to generate robust and reliable output. Lastly, RAD is a toolkit that aims to resolve a series of problems in the deconvolution of bulk RNA, while NMF mainly focuses on optimizing the objective function

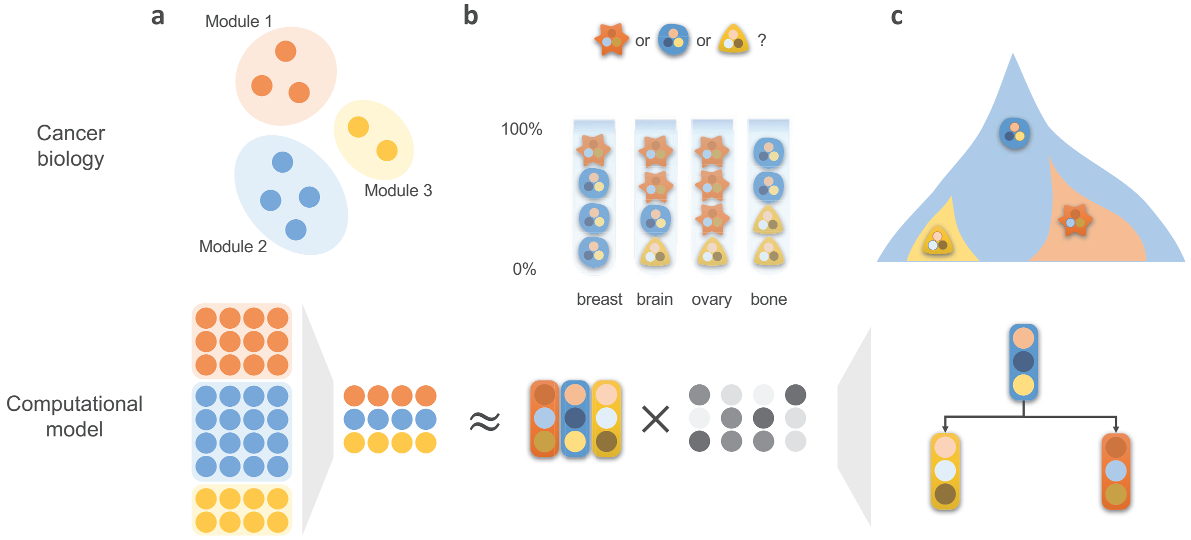

Figure 5.1: Illustration of discovering underlying cancer evolutionary mechanisms of metastatic progression using the proposed computational models. (A) Gene module compression. Genes from the same modules have similar co-expression patterns. By mapping the individual genes into modules, we can get a cleaner representation of bulk RNA data. (B) Deconvolution of bulk RNA. Using the RAD algorithm, we unmixed the bulk data into two matrices: expression matrix and fraction matrix, which represent the expression profile and fractions of individual cell populations. (C) Phylogeny inference of cell communities. The inferred phylogeny represents the most likely evolutionary trajectories of cell populations.

efficiently.

Table 5.1: Functions in the RAD package. The five-line demo shows a typical application of RAD package, which takes as input the bulk RNA data $\mathbf{B}$, bulk marker data $\mathbf{B}_P$, and gene module knowledge, and outputs deconvolved RNA $\mathbf{C}$, biomarker $\mathbf{C}_P$, and fractions $\mathbf{F}$.

Function	Demo code
`compress_module`	`B_M = compress_module(B, module)`
`estimate_number`	`k = estimate_number(B_M)`
`estimate_clones`	`C_M, F = estimate_clones(B_M, k)`
`estimate_marker`	`C_P = estimate_marker(B_P, F)`
	`C = estimate_marker(B, F)`

Deconvolution problem formulation

Given a non-negative bulk RNA-Seq expression matrix $\mathbf{B} \in \mathbb{R}_+^{m \times n}$, where each row i is a gene, each column j is a tumor sample, our goal is to infer an expression profile matrix $\mathbf{C} \in \mathbb{R}_+^{m \times k}$, where each column l is a cell community, and a fraction matrix $\mathbf{F} \in \mathbb{R}_+^{k \times n}$, such that: $\mathbf{B} \approx \mathbf{CF}$. To be more concrete, we formulated the problem, as in our prior work [219], as follows:

$$\min_{\mathbf{C},\mathbf{F}} \quad \|\mathbf{B} - \mathbf{CF}\|_{\text{Fr}}^2 , \tag{5.1}$$

$$\text{s.t.} \quad \mathbf{C}_{il} \geq 0, \quad i = 1, ..., m, \ l = 1, ..., k, \tag{5.2}$$

$$\mathbf{F}_{lj} \geq 0, \quad l = 1, ..., k, \ j = 1, ..., n, \tag{5.3}$$

$$\sum\nolimits_{l=1}^{k} \mathbf{F}_{lj} = 1, \qquad j = 1, ..., n, \tag{5.4}$$

where $\|\mathbf{X}\|_{\text{Fr}}$ is the Frobenius norm. The column-wise normalization of Eq. 5.4 ensures that the total fractions of all cell communities in the same sample sum up to one. This optimization problem is non-convex and non-trivial to resolve. In addition, the bulk data $\mathbf{B}$ is noisy, so even an optimal solution does not necessarily fit the ground truth $\mathbf{C}$ and $\mathbf{F}$. Our overall approach to solve for this problem consists of three phases — a randomized warm-start procedure to develop an initial guess as to a solution, coordinate descent optimization to improve fit to the objective, and a minimum similarity selection procedure to identify the most informative partitioning among a set of random restarts — as described in more detail below.

Knowledge-driven gene module compression

By default, RAD unmixes the raw expression matrix $\mathbf{C}$ directly. However, gene module compression (`compress_module`) is a suggested option if we have prior information on what genes belong to the same gene modules. Gene expressions within the same gene modules are highly correlated, which can be explained by their shared genomic context, similar biological functions, or participation in the same interaction network. Intuitively, we can compress the noisy expressions of individual genes into cleaner "gene modules" [57, 174], to reduce the signal-to-noise ratio (SNR).

Here we utilize the functional annotation clustering tool in DAVID to group the top 3,000 most varied genes into around 100 to 200 modules [94], where multiple external knowledge databases are used to facilitate the clustering. The gene expressions within the same module are averaged to get the module expression value. The resulting bulk compressed module matrix is $\mathbf{B}_M \in \mathbb{R}_+^{m_1 \times n}$, where m_1 is the total number of modules.

The gene module compression in our work is knowledge-driven, which is reliable even when only limited tumor samples are available, in contrast to prior work using data-driven clustering of coexpressed genes [265], which is more dependent on large sample sizes. We validate the superiority of knowledge-driven compression over data-driven compression on a real dataset (Fig. 5.4B,E), and the effect of such knowledge-driven compression (Fig. 5.4D,E) in Sec. 5.2.3.

Core algorithm of RAD

The core algorithm of RAD (`estimate_clones`) unmixes the compressed bulk RNA data $\mathbf{B}_M$ into expression profiles $\mathbf{C}_M$ and fractions $\mathbf{F}$ of individual communities. The method works in three phases – warm-start, coordinate descent, and minimum similarity selection – to achieve accuracy and robustness. RAD can directly unmix the original bulk RNA data $\mathbf{B}$ as well, and we will remove the subscript $_M$ in this section for simplicity.

Warm-start – The warm-start phase borrows its idea from the multiplicative update (MU) rules for the general NMF problem [122]. It first randomly initializes $\mathbf{C}$ and $\mathbf{F}$ from the uniform distribution $U(0, 1)$ then iterates the following loop until the objective function converges:

$$\mathbf{C} \leftarrow \mathbf{C} \odot (\mathbf{B}\mathbf{F}^\intercal) \oslash (\mathbf{C}\mathbf{F}\mathbf{F}^\intercal)\,, \tag{5.5}$$

$$\mathbf{F} \leftarrow \mathbf{F} \odot (\mathbf{C}^\intercal\mathbf{B}) \oslash (\mathbf{C}^\intercal\mathbf{C}\mathbf{F})\,, \tag{5.6}$$

$$\mathbf{F}_{lj} \leftarrow \mathbf{F}_{lj} \Big/ \sum_{l'=1}^{k} \mathbf{F}_{l'j}\,, \qquad l = 1, ..., k,\ j = 1, ..., n, \tag{5.7}$$

where $\odot$ and $\oslash$ are element-wise product and element-wise division operators. We add an additional MU step Eq. 5.7 to the original two MU steps Eq. 5.5-5.6 to satisfy the constraint Eq. 5.4. This deteriorates the convergence guarantee of the original MU rules. Therefore, we have to stop the update once the objective stops decreasing. However, the revised MU rules in the warm-start phase still have the advantage of fast convergence even when the initial values of $\mathbf{C}$ and $\mathbf{F}$ are far from optimal solution, and can provide a reasonable starting point for the second phase.

Coordinate descent – After the warm-start phase, the coordinate descent phase optimizes over the two coordinates $\mathbf{C}$ and $\mathbf{F}$ iteratively until the objective function converges:

$$\mathbf{C} \leftarrow \arg\min_{\mathbf{C}} \quad \|\mathbf{B} - \mathbf{C}\mathbf{F}\|_{\text{Fr}}^2\,, \tag{5.8}$$

$$\text{s.t.} \quad \mathbf{C}_{il} \geq 0, \qquad i = 1, ..., m,\ l = 1, ..., k, \tag{5.9}$$

and

$$\mathbf{F} \leftarrow \arg\min_{\mathbf{F}} \quad \|\mathbf{B} - \mathbf{C}\mathbf{F}\|^2_{\text{Fr}}, \tag{5.10}$$

$$\text{s.t.} \quad \mathbf{F}_{lj} \geq 0, \quad l = 1, ..., k, \; j = 1, ..., n, \tag{5.11}$$

$$\sum\nolimits_{l=1}^{k} \mathbf{F}_{lj} = 1, \qquad j = 1, ..., n, \tag{5.12}$$

Although the original optimization problem Eq. 5.1-5.4 is non-convex, the two sub-problems of the coordinate descent are convex and can be solved using general quadratic programming. We used the Python package `cvxopt` [6]. The coordinate descent phase can usually further reduce the loss function by around 5% to 30% after the warm-start phase, as evaluated on a real dataset GSE19830 (Sec. 5.1.5; Shen-Orr et al. [204]).

Minimum similarity selection – Since the deconvolution problem Eq. 5.1-5.4 is non-convex, different initialization values of $\mathbf{C}$ and $\mathbf{F}$ may converge to distinct solutions. We reran the initialization, warm-start, and coordinate descent for multiple times (10 in our experiments) and selected the solution with minimum similarity of expression profiles. We defined the unnormalized cosine similarity of a specific solution $\mathbf{C}$ as:

$$\text{cosim}(\mathbf{C}) = \sum_{l=1}^{k-1} \sum_{l'=l+1}^{k} \mathbf{C}_{\cdot l}^{\intercal} \mathbf{C}_{\cdot l'}. \tag{5.13}$$

Biologically, the use of minimum similarity is motivated by the assumption that individual cell communities should be distinct from each other. Minimum similarity performs slightly better than minimum loss on GSE19830 dataset, although the two criteria often overlap empirically.

Estimating number of communities

The core algorithm of RAD infers the $\mathbf{C}$ and $\mathbf{F}$ from $\mathbf{B}$ given a specific number of communities k, which in practice is unknown to us in advance. RAD estimates the number of cell components (`estimate_number`) through cross-validation (CV). The estimated/optimal k reflects the trade-off between model bias and the variance.

We take a 20-fold CV as an example: In each fold, 5% of the elements in $\mathbf{B}$ are unseen at the time of training/optimization. At the time of test, the loss is calculated only on these unseen 5% elements. Technically, we realized it by utilizing the mask matrices with the same shape of $\mathbf{B}$: $\mathbf{M}, \mathbf{M}_{\text{test}} \in \{0, 1\}^{m \times n}$ ($\mathbf{M}$ is $\mathbf{M}_{\text{train}}$. We omit the subscript $_{\text{train}}$ for simplicity.), and $\mathbf{M} + \mathbf{M}_{\text{test}} = 1^{m \times n}$. $\mathbf{B}_{ij}$ is blocked when the corresponding position of $\mathbf{M}_{ij} = 0$. In each fold, 5% of the $\mathbf{M}_{\text{test}}$ are 1's, and 95% of the $\mathbf{M}$ are 1's. Note that the positions of the 1's and 0's are randomly distributed across the whole matrices $\mathbf{M}$ and $\mathbf{M}_{\text{test}}$, rather than column-wise or row-wise.

At the time of validation, we can calculate the "normalized MSE" as the CV error on validation set:

$$\|\mathbf{M}_{\text{test}} \odot \left(\mathbf{B} - \hat{\mathbf{C}}\hat{\mathbf{F}}\right)\|_2^2 \Big/ \|\mathbf{M}_{\text{test}} \odot \mathbf{B}\|_2^2 \tag{5.14}$$

At the time of training, we want to optimize the following objective function with the same constraints to Eq. 5.2-5.4:

$$\min_{\mathbf{C},\mathbf{F}} \quad \|\mathbf{M} \odot (\mathbf{B} - \mathbf{C}\mathbf{F})\|_{\mathrm{Fr}}^2 . \tag{5.15}$$

There exist corresponding algorithms for MU warm-start and coordinate descent phases when a mask $\mathbf{M}$ exists. The MU rules to optimize the masked objective Eq. 5.15 is similar to Eq. 5.5-5.7. However, Equation 5.5,5.6 are revised to the following rules:

$$\mathbf{C} \leftarrow \mathbf{C} \odot \left(\left(\mathbf{M} \odot \mathbf{B} \right) \mathbf{F}^{\intercal} \right) \oslash \left(\left(\mathbf{M} \odot \left(\mathbf{C}\mathbf{F} \right) \right) \mathbf{F}^{\intercal} \right) , \tag{5.16}$$

$$\mathbf{F} \leftarrow \mathbf{F} \odot \left(\mathbf{C}^{\intercal} \left(\mathbf{M} \odot \mathbf{B} \right) \right) \oslash \left(\mathbf{C}^{\intercal} \left(\mathbf{M} \odot \left(\mathbf{C}\mathbf{F} \right) \right) \right) . \tag{5.17}$$

The coordinate descent iterations of the masked version are the same to the unmasked version Eq. 5.8-5.12, except that $(\mathbf{B} - \mathbf{C}\mathbf{F})$ in Eq. 5.8,5.10 are replaced with $\mathbf{M} \odot (\mathbf{B} - \mathbf{C}\mathbf{F})$. The subproblems of the two masked coordinate descent steps are still quadratic programming problems.

Cancer-related pathway annotation

We further processed the bulk data to derive aggregate biomarkers profiling activity of cancer-related pathways. Although this is not part of the RAD toolkit, we make use of bulk biomarkers and estimates of their activity in individual cell components (Sec. 5.1.2) to interpret the results of the RAD. We extracted 24 cancer-related pathways and the corresponding genes from the *pathways in cancer* (hsa05200), *breast cancer* (hsa05224), and *glioma* (hsa05214) of the KEGG database [105]. The value of each pathway is the average expression of all genes within it. We mapped the original bulk gene matrix $\mathbf{B}$ into the cancer pathway probes matrix $\mathbf{B}_P \in \mathbb{R}_+^{24 \times n}$. Readers can find the list of pathways in Fig. 5.5.

Cancer pathways are distinct from gene modules, which capture the major variance across samples and the co-expression patterns using prior knowledge to facilitate more reliable deconvolution. However, co-expression modules are often weakly linked to biological functions and not informative for downstream analysis. In contrast, biomarkers such as cancer-related pathways facilitate functional interpretation of results. We did not use cancer-related pathways or other biomarkers for deconvolution, since they are not representative of the expression profiles.

Pathway estimation of cell communities

Given the bulk cancer pathway data $\mathbf{B}_P$, RAD utilizes the deconvolved $\mathbf{F}$ to infer the pathway values of cell populations $\mathbf{C}_P$ (`estimate_marker`), similar to the paradigm of the Digital Sorting Algorithm (DSA; Zhong et al. [273]), by replacing $\mathbf{B}$ and $\mathbf{C}$ with $\mathbf{B}_P$ and $\mathbf{C}_P$ in Eq. 5.8,5.9. The unmixed $\mathbf{C}$ or $\mathbf{C}_M$ may not be easy to interpret. However, other biomarkers, such as cancer pathways, provide a possible way to explain the biological process of each cell community, and can be used as input to infer the phylogeny and perturbed pathways during progression (Sec. 5.1.3).

5.1.3 Phylogeny inference through minimum elastic potential

Given the deconvolved cell clone profiles $\mathbf{C} \in \mathbb{R}_+^{n\times k}$ of k cell components, we inferred trees describing the observed extant and unobserved ancestral Steiner communities. We use the term "phylogeny" to refer to these trees to highlight their connection to tumor phylogenetics methods often used for similar purposes and on similar data [199], although we recognize that these trees describe changes in mean transcriptomic states of groups of associated cells rather than strictly clonal evolution and thus are not proper phylogenies.

RAD works in the original non-log linear space, which recovers underlying components with higher accuracy and lower bias [272]. In the phylogeny inference algorithm, however, we instead used the log space to focus on the fold change of expressions $\mathbf{C} \leftarrow \log_2(\mathbf{C}+1)$. We also normalize the expression to make them have zero mean value for each gene.

We used a variant of the minimum elastic potential (MEP) method to infer the phylogeny structure and expression values of Steiner nodes [219]. Taking $\mathbf{C}$ as input, the MEP first builds the phylogeny tree that includes k extant nodes and $(k-2)$ ancestral Steiner nodes using the neighbor-joining (NJ) algorithm [160]. To identify the perturbed biological processes and gene expression during tumor evolution, MEP infers expression values of the unknown Steiner nodes by minimizing an "elastic potential energy". For a specific pathway or gene, this is equivalent to a quadratic programming problem:

$$\min_{\mathbf{x}} \quad \frac{1}{2}\mathbf{x}^\intercal\mathbf{P}(\mathcal{W})\mathbf{x} + \mathbf{q}(\mathcal{W},\mathbf{y})^\intercal\mathbf{x}, \tag{5.18}$$

where $\mathbf{x} \in \mathbb{R}^{k-2}$, $\mathbf{y} \in \mathbb{R}^k$ are the expression values of Steiner and extant nodes; $\mathbf{P}(\mathcal{W})$ is a function that takes as input the tree edge weights $\mathcal{W}$ and outputs a matrix $\mathbf{P} \in \mathbb{R}^{(k-2)\times(k-2)}$; $\mathbf{q}(\mathcal{W},\mathbf{y})$ is a function that takes as input edge weights $\mathcal{W}$ and vector $\mathbf{y}$ and outputs a vector $\mathbf{q} \in \mathbb{R}^{k-2}$. We further added an ℓ_2-regularization $\lambda\mathbf{I}^{(k-2)\times(k-2)}$ to the $\mathbf{P}(\mathcal{W})$. In practice we use $\lambda = 0.001$, which helps more stable inference when the total number of nodes is large. Interested readers can find detailed problem formulation, derivatives, and proofs of MEP in the previous work [219].

Although we used the notation $\mathbf{C}$ for the explanation in this section, in our application, we used the pathway probes $\mathbf{C}_P$ to infer the cancer pathway values of each Steiner nodes for downstream interpretation.

5.1.4 Evaluation

The deconvolution algorithm outputs both the estimated expression profiles $\hat{\mathbf{C}}$ and the fractions $\hat{\mathbf{F}}$ of each cell communities. We utilized four different metrics to measure the accuracy and error of the two estimators with the ground truth $\mathbf{C}$ and $\mathbf{F}$ following previous research [166, 265, 275]: R_C^2 (Pearson coefficient of $\hat{\mathbf{C}}$ and $\mathbf{C}$), L_1 loss ($\|\hat{\mathbf{C}}-\mathbf{C}\|_1/\|\mathbf{C}\|_1$), R_F^2 (Pearson coefficient of $\hat{\mathbf{F}}$ and $\mathbf{F}$), MSE (Mean square error of $\hat{\mathbf{F}}$ and $\mathbf{F}$).

5.1.5 Datasets and preprocessing

We utilized three datasets throughout this chapter: one real dataset of breast cancer metastasis (BrM; Zhu et al. [274]); one real dataset of both pure and mixed transcriptome of liver, brain, and lung (GSE19830; Shen-Orr et al. [204]); and one simulated dataset. All three datasets contain bulk transcriptome $\mathbf{B}$, which is the input of RAD and downstream analysis. Ground truth $\mathbf{C}$, $\mathbf{F}$ are only available in simulated and GSE19830 datasets, and unknown in BrM dataset. Ground truth module knowledge of genes are only available in simulated dataset, GSE19830 and BrM datasets instead used DAVID to infer gene module.

Simulated dataset

We simulated a series of datasets with different parameters of module size (1, 2, 4, ..., 512) and noise level $\sigma \in [0, 5]$ to validate the effectiveness of RAD and module compression. We selected most of the parameters following [265], with selected parameters consistent with the distribution of the real dataset GSE19830.

We considered the expression of 2,048 interested genes and assumed three pure cell clones $\mathbf{C} \in \mathbb{R}_+^{2048\times 3}$. In the case where each module consists of just one gene, $\mathbf{C}$ was drawn from a log-normal distribution independently [18]:

$$\mathbf{C}_{il} \sim 2^{\mathcal{N}\left(6,2.5^2\right)}, \quad i = 1, 2, ..., 2048,\ l = 1, 2, 3. \tag{5.19}$$

We assumed that genes from the same module are highly correlated and prone to co-express. When module size is greater than one, e.g., there are four genes in a module, for each gene module $\mathbf{C}_S \in \mathbb{R}_+^{4\times 3}$ (a subblock of $\mathbf{C}$), we draw each gene module as follows:

$$\left(\mathbf{C}_S\right)_{\cdot l} \sim 2^{\mathcal{N}\left([6,6,6,6]^\intercal, 2.5^2\Sigma\right)}, \quad l = 1, 2, 3, \tag{5.20}$$

where Σ is the covariance matrix of four genes in the module:

$$\Sigma = \begin{bmatrix} 1 & 0.95 & 0.95 & 0.95 \\ 0.95 & 1 & 0.95 & 0.95 \\ 0.95 & 0.95 & 1 & 0.95 \\ 0.95 & 0.95 & 0.95 & 1 \end{bmatrix}. \tag{5.21}$$

We assumed 100 mixture samples of bulk RNA, and drew the fractions of mixture samples uniformly from a unit simplex:

$$\mathbf{F}_{\cdot j} \sim U[\Delta^3], \quad j = 1, 2, ..., 100, \tag{5.22}$$

Finally, we added the noise of magnitude σ to get the final bulk data:

$$\mathbf{B}_{ij} \sim (\mathbf{CF})_{ij} + 2^{\mathcal{N}\left(0,\sigma^2\right)},\ i = 1, 2, ..., 2048,\ j = 1, 2, ..., 100. \tag{5.23}$$

GSE19830 dataset

GSE19830 contains RMA-normalized Affymetrix expressions of cells from rat brain, liver, and lung biospecimens [204]. It mixed the three pure tissues in different predefined proportions, leading to 33 mixture samples. Expression profiles of each pure and mixed sample were measured three times.

We removed the non-protein coding genes, conducted quantile normalization, and took the average of three replicates. This yielded three matrices for the GSE19830 data: expression profiles of the three pure clones $\mathbf{C} \in \mathbb{R}_+^{13741\times 3}$, fractions of three clones in all mixture data $\mathbf{F} \in \mathbb{R}_+^{3\times 33}$, and bulk data $\mathbf{B} \in \mathbb{R}_+^{13741\times 33}$.

BrM dataset

The BrM dataset contains matched transcriptome data of breast cancer metastasis patients [274]. There are 102 samples from 51 patients in total. There are four possible different metastatic sites for each breast cancer patient in the dataset: brain (BR; 22 patients), ovary (OV; 13 patients), bone (BO; 11 patients), and gastrointestinal tract (GI; 5 patients). A sample from the primary breast site has a matched sample from the metastatic site of the same patients. RNA-Seq of around 60,000 genes are available. We will denote the sample from the metastatic site and the corresponding primary site as MBR/PBR, MOV/POV, MBO/PBO, MGI/PGI. We removed the non-protein coding genes and conducted quantile normalization to get the final bulk data: $\mathbf{B} \in \mathbb{R}_{+}^{19689\times 102}$.

5.2 Results

5.2.1 Gene modules facilitate robust deconvolution

RAD can act directly on the bulk gene expression $\mathbf{B}$ or on the knowledge-based compressed bulk gene module expression $\mathbf{B}_M$ (Sec. 5.1.2). We validated that the gene module compression is beneficial to RAD through both simulated data and real datasets.

As a proof of concept, we first assume the knowledge of gene modules is correct and known to us. We generated simulated bulk data with a noise level $\sigma = 4$ and various module sizes from 1 to 512 (Sec. 5.1.5). We calculated four metrics to evaluate the accuracy of RAD estimation on these data, and repeated all experiments 100 times (Fig. 5.2). A moderate module size of 32 makes RAD the most accurate (highest median accuracy) and robust (smallest variance), which is helpful to deconvolution. RAD is less robust (high variance) on the original uncompressed bulk gene data (module size of one). However, when module size is too large, RAD has both inaccurate and unstable performance.

We further examined whether the module knowledge from DAVID generates a reasonable module size and facilitates robust RAD deconvolution. We applied RAD to both the uncompressed and compressed GSE19830 dataset. As one can see in Fig. 5.4D,E, module compression based on DAVID improve the RAD estimation of both $\mathbf{C}$ and $\mathbf{F}$.

5.2.2 RAD detects the correct number of cell components

RAD utilizes cross-validation (CV) to identify the number of underlying cell populations (Sec. 5.1.2). To validate its correctness, we applied a 20-fold CV to the GSE19830 data. As one can see in Fig. 5.3A, the CV error Eq. 5.14 drops quickly when the number of cell components k increases from one to three, and flattens after k goes over three. In this case, we identify the correct number of cell clones to be three[2].

For the BrM dataset, we applied a 20-fold CV as well and found the CV error drops when k is smaller than seven, and increases with substantial variance because of overfitting when k is higher than seven. We therefore used k=7 as the number of communities in the BrM data.

[2] Although k=8 gives minimum CV error, it is due to the small noise or artifacts in the samples.

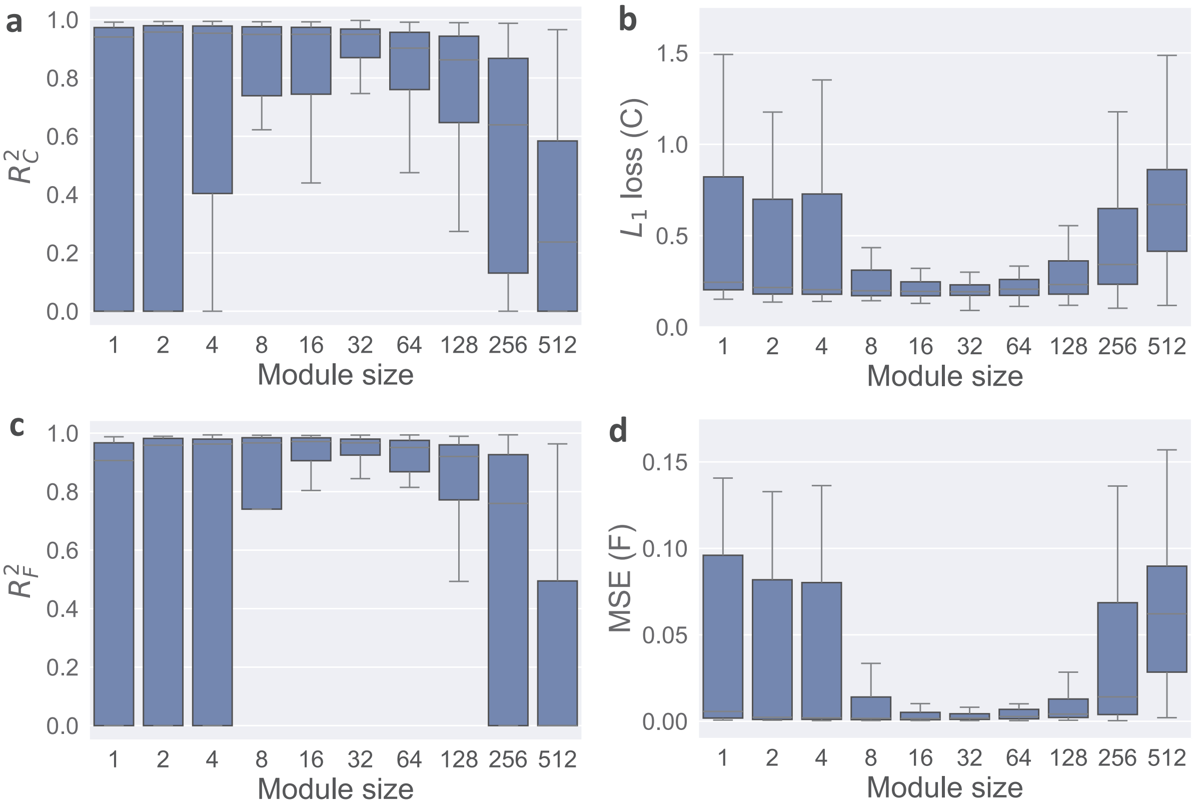

Figure 5.2: Effectiveness of gene module representation. Compressing the expression of individual genes into gene modules can promote more robust deconvolution with proper module size. We tested on the simulated bulk data with the noise level of $\sigma = 4$, and repeated all the experiments for 100 times to get the boxplot. We evaluated four metrics that measure the accuracy of deconvolution with different gene module sizes. Note that “module size” of one is equivalent to the original gene expression without module compression. (A) Accuracy of expression matrix estimation: R_C^2. (B) Error of expression matrix estimation: L_1 loss. (C) Accuracy of fraction matrix estimation: R_F^2. (D) Error of fraction matrix estimation: MSE.

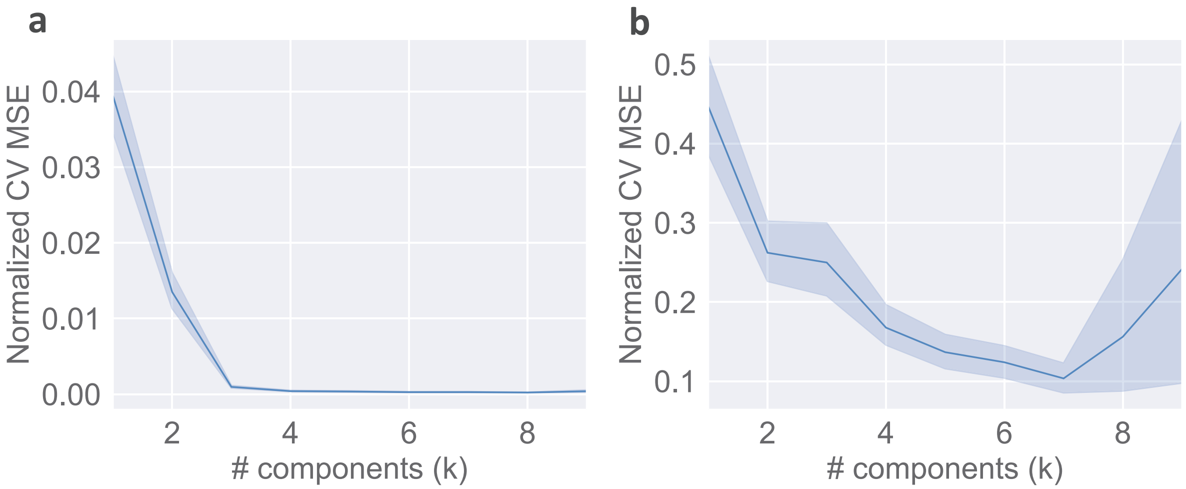

Figure 5.3: Cross-validation to automatically select the number of cell communities. We conducted a 20-fold cross-validation to infer the optimal number of cell component k as the input of RAD algorithm. (A) Cross-validation on GSE19830 dataset. The actual number of cell clones is three. (B) Cross-validation on BrM dataset. The inferred number of cell components is seven, which is used as the input parameter for Fig. 5.5-5.7.

Previous research using a subset of the BrM dataset (only breast cancer brain metastasis samples) identified only five cell populations [219]. With a larger sample size, and more heterogeneous data, we can identify more fine-grained unmixed cell communities.

5.2.3 RAD estimates cell populations robustly and accurately

We then evaluated the accuracy and error of RAD relative to alternative deconvolution algorithms. There are many algorithms to solve the "partial deconvolution" problem, where the expression profiles of clones $\mathbf{C}$ are available at the time of deconvolution, e.g., DSA [273]. For tumor samples, however, the underlying cell types are often unknown, and partial deconvolution can be unstable. We consider the "complete deconvolution" problem here [265], where the expression profiles of populations $\mathbf{C}$ are not available and must be inferred.

There are a few existing algorithms that seem to be suitable for the complete deconvolution problem, e.g., principal components analysis (PCA), independent components analysis (ICA), and NMF. However, none of these algorithms account for the fraction normalization constraints Eq. 5.4, and PCA and ICA do not guarantee the non-negativity of expression profiles Eq. 5.2. A few more well-designed algorithms have been developed for the specific complete deconvolution problem, including Geometric Unmixing [200], LinSeed [265], and NND [219]:

- Geometric Unmixing: It poses unmixing as a problem from computational geometry by assuming all the bulk samples are located in a simplex, where the corners of the simplex represent the expression profiles of pure clones.
- LinSeed: Based on the observation that genes in the same module are highly correlated to each other. It first identifies these anchor genes through linear correlation. Then it

uses a partial deconvolution algorithm DSA to infer the fractions $\mathbf{F}$ and expressions of non-anchor genes $\mathbf{C}$ [273].

- NND: It converts the complete deconvolution problem equivalently into an optimization problem, which can be implemented as a neural network. It uses backpropagation (gradient descent) to optimize.

For RAD, we can directly take as input the bulk gene expression matrix $\mathbf{B}$ (Fig. 5.4D), or use the more noise-free bulk module expression matrix $\mathbf{B}_M$ (Fig. 5.4E). Although the NND can also take $\mathbf{B}$ or $\mathbf{B}_M$ in principle, we only included results of "NND w/ Module" (Fig. 5.4C), due to the intractable training time of "NND w/ Gene".

Figure 5.4 compares the performance of the five deconvolution algorithms and module-compressed variants on the GSE19830 dataset. The gene module compression improves the accuracy of RAD significantly (Fig. 5.4D,E), consistent with the observations in Fig. 5.2. The RAD on the compressed module data outperforms the other three algorithms in metrics R_C^2, R_F^2, and MSE (Fig. 5.4A-C,E). It also has comparable L_1 loss with the "NND w/ Module" algorithm (Fig. 5.4C,E). These results reveal the superiority and accuracy of the RAD algorithm and module compression.

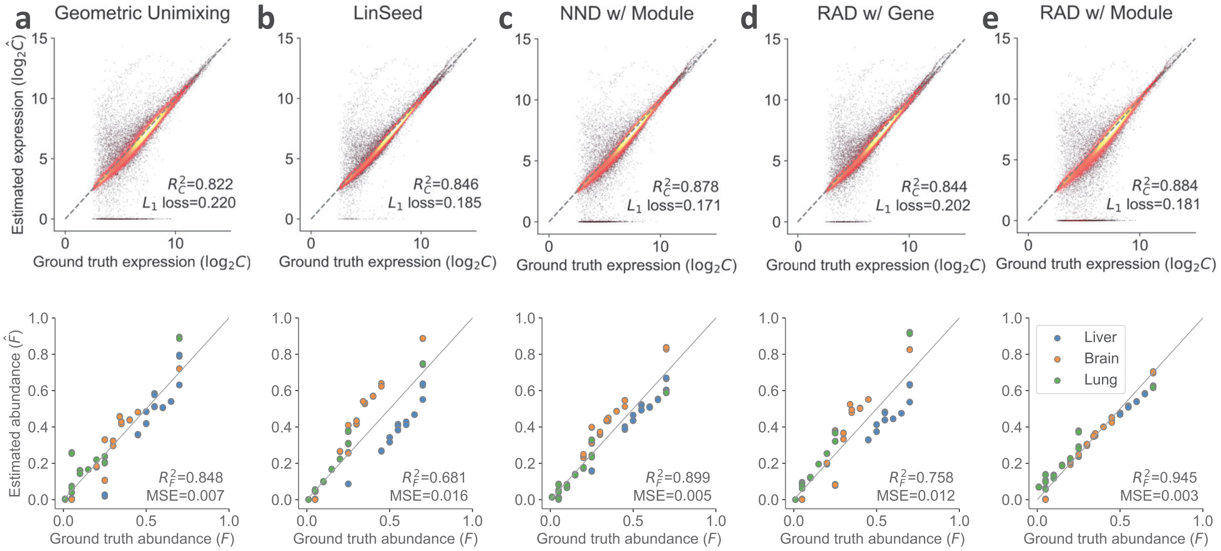

Figure 5.4: Performance of different deconvolution algorithms on GSE19830 dataset. We compared the accuracy of both estimated $\mathbf{C}$ and $\mathbf{F}$ across four different deconvolution algorithms. RAD with the module achieves best the performance among all the deconvolution algorithms evaluated in R_C^2, R_F^2, and MSE. The module information facilitates the RAD to achieve better performance. (A) Geometric Unmixing. (B) LinSeed. (C) NND (with module). (D) RAD (with gene). (E) RAD (with module).

The elevated accuracy and robustness of RAD over competing algorithms is crucial for downstream analyses such as phylogeny inference. For example, RAD reveals a more detailed portrait of perturbed pathways (Sec. 5.2.5) during metastasis than our previous NND algorithm [219].

5.2.4 Landscape of tumor cell communities

We derived the fractions of cell communities in each sample $\mathbf{F}$ using RAD, and further inferred the pathway values of each cell community from $\mathbf{B}_P$ and $\mathbf{F}$ (Sec. 5.1.2). Although there exists inter-tumor heterogeneity across cancer samples, RAD aims to separate the shared features of cell populations across these tumors from sample-specific features as in prior cross-cohort deconvolutional phylogeny studies [192, 200] and prior oncogenetic tree methods that do not include a deconvolution step [58, 191].

Expression profiles of cell communities Figure 5.5 shows the pathway values of each cell population in log scale $\log_2(\mathbf{C}_P + 1)$ after applying RAD to the BrM dataset. $C5$ is the most abnormal community, having lost half of the pathways completely (almost zero expression), including *PI3K-Akt*, *ECM-receptor*, and *Calcium*. Another unusual cell community is $C2$, which is specifically enriched in neurotransmitter and calcium homeostasis functions (*Calcium* and *cAMP*; Hofer and Lefkimmiatis [90]). We hypothesize that $C2$ might reflect a cell community combining both neural cells and metastatic tumor cells. In contrast, $C6$, which we infer to approximate the primary breast tumor community, has a relatively high expression of *PI3K-Akt* [22] and immune function (*Cytokine-cytokine receptor*; Zhu et al. [274]).

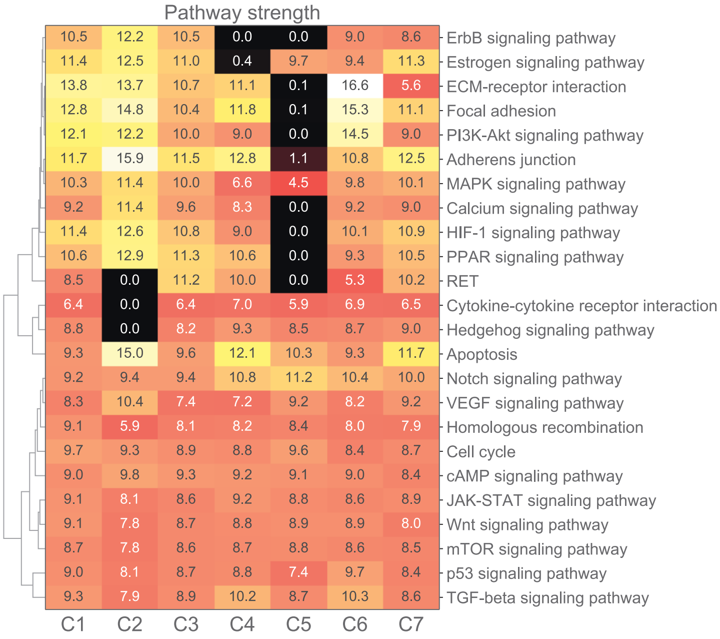

Figure 5.5: Cancer-related pathway strengths of each cell component from BrM dataset. The strengths are shown in log-scale $\log_2(\mathbf{C}_P + 1)$.

Distribution of cell communities Figure 5.6 shows the distribution of cell components across different metastatic sites. We classified the tumor sites into eight categories: MBR/PBR, MOV/POV, MBO/PBO, and MGI/PGI (Sec. 5.1.5). We observe that $C6$ is always decreased in the metastatic samples, from which we infer it may approximate the primary clones and capture features that distinguish primary clones from metastatic ones in general. Other components are increased in specific metastasis types. For example, $C1$ in OV and BO; $C2$ in BR, OV, and GI; $C3$ in BR, OV, and GI; $C4$ in BO; $C5$ in BR, OV, and BO; and $C7$ in BR. This indicates that there exist different cell population mixtures in different metastatic sites, likely in part reflecting site-specific stroma but also revealing commonalities across metastatic sites. We further note that the distribution of cell clones across primary sites is related to their eventual sites of metastasis. This result is suggestive that there may be a signal in the primary clonal composition of whether a primary tumor is likely to metastasize to a particular site, although that suggestion requires further evaluation and validation.

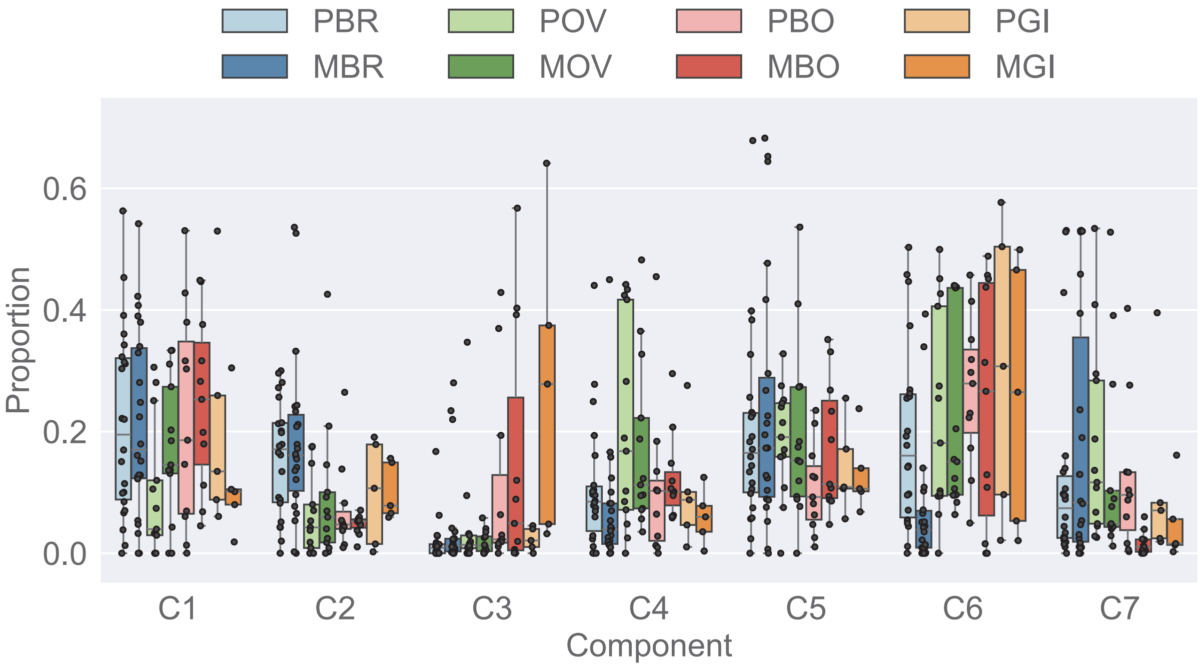

Figure 5.6: Fractions of communities in both primary and metastatic sites of four different metastasis types from BrM dataset. The differences of cell distribution exist both between primary (lighter) and metastatic (darker) sites, and across four metastatic cases (different colors).

5.2.5 Common evolutionary mechanisms of breast cancer metastasis

Using the unmixed cell clones $\mathbf{C}_P$, we built a phylogeny and inferred the pathway values of Steiner nodes using the MEP algorithm (Sec. 5.1.3). Since there are vast differences across the four metastasis types (Fig. 5.6), we inferred a phylogeny tree for each metastasis type (Fig. 5.7A-D). We presented $C6$ as the common root node, as it consistently decreases in all four metastasis types, and identified the communities whose average fractions increase in the metastatic communities of specific metastasis types. Figure 5.7 shows the top five most differentially expressed pathways for more than one fold along each edge.

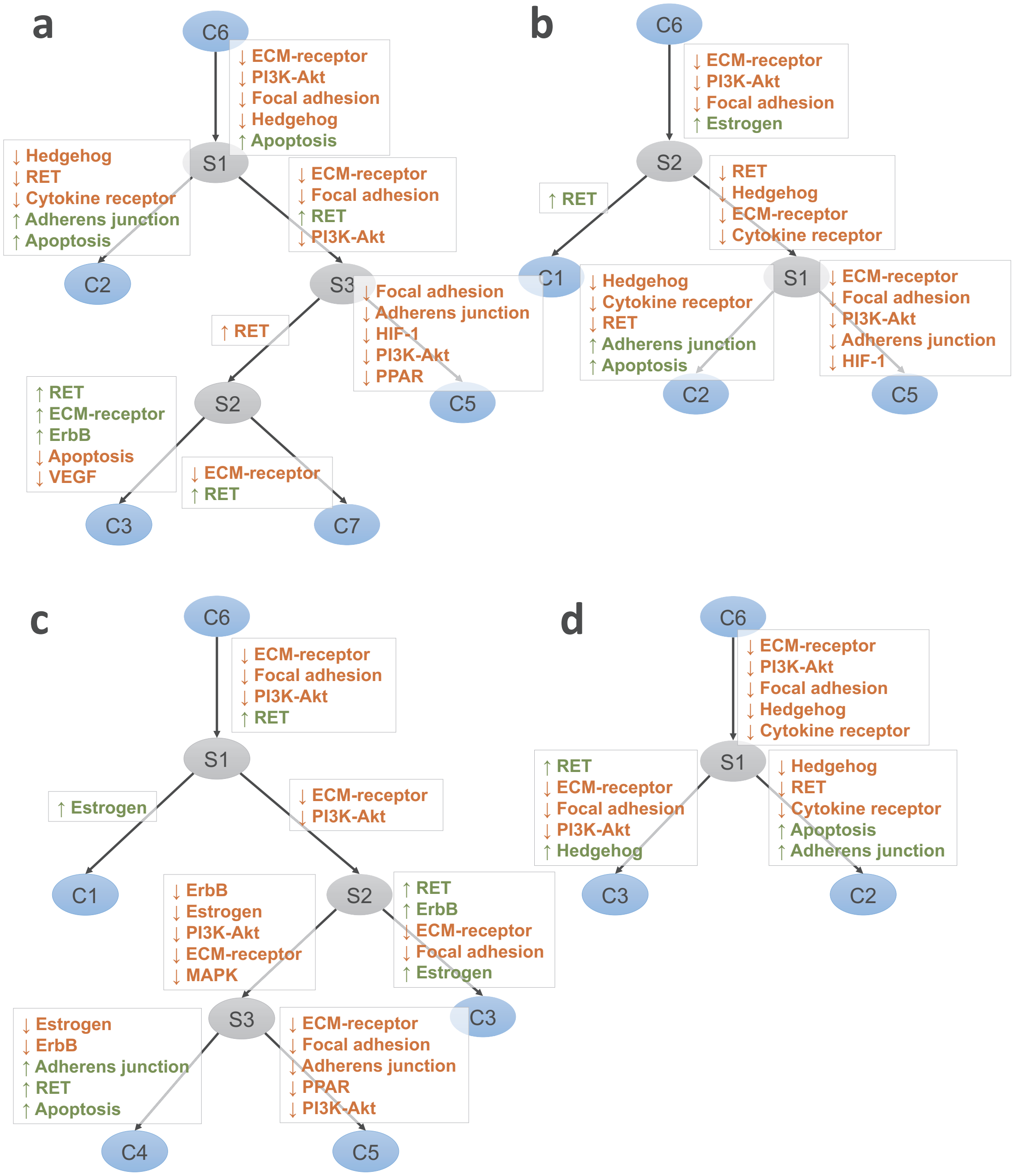

Figure 5.7: Phylogenies of four different metastatic cases. Although there are large differences in tumor communities across the four metastasis sites, there exists common mechanisms, such as the early events of perturbed *PI3K-Akt*, *ECM-receptor*, and *focal adhesion* pathways. (A) Breast cancer brain metastasis. (B) Breast cancer ovary metastasis. (C) Breast cancer bone metastasis. (D) Breast cancer GI metastasis.

As one can see, there are common patterns at the early stage of metastasis, e.g., the decrease of *PI3K-Akt*, *ECM-receptor interaction*, and *focal adhesion*. The loss of *PI3K/Akt/mTOR* in metastatic tumors has already been identified in brain metastasis research based on both genomic and transcriptomic data [22, 219]. Our result indicates the loss of *PI3K-Akt* pathway is a common event among the general metastasis types as well, not limited to brain metastasis. Loss of *ECM-receptor interaction* and *focal adhesion* also plays a critical role in tumor cell migration generically [156]. Tumor cells adhere to the extracellular matrix (ECM), forming the structures called focal adhesions, and loss of these interactions is a key step in enabling metastatic migration.

There are also substantial differences across metastatic sites that may suggest potential markers of incipient site-specific metastasis. The dysregulation of some perturbed pathways have already been shown to be closely related to tumor progression (*Hedgehog*, *Apoptosis*; Gupta et al. [84]). *RET* and *ErbB* have been shown recurrently perturbed in metastasis [182]. The reduction of *Cytokine-cytokine receptor* may reflect the reduced immune cell recruitment in metastatic samples [274].

5.3 Discussion

We developed a tool called RAD for deconvolution of multi-stage transcriptomic data corresponding to primary and metastatic tumor samples. We have shown that RAD can robustly and accurately estimate the number of cell populations, unmix the cell populations, and infer biomarkers from bulk RNA-Seq of tumor samples, while showing improved reliability and accuracy over other deconvolution algorithms on both simulated and real RNA datasets. We applied RAD with gene module compression and a phylogeny inference algorithm to bulk transcriptome data collected from matched breast primary and four different metastatic sites to characterize similarities and variations in tumor clonal populations by eventual site of metastasis. Significant perturbations of cancer-related pathways, such as *PI3K-Akt*, *ECM-receptor*, and *focal adhesion* emerge as common early events across sites of breast cancer metastasis, showing the potential of the method to reveal recurrent evolution mechanisms of breast cancer metastasis.

It has been observed that the noise of RNA expression grows with its amplitude, suggesting that a more principled probabilistic model instead of Frobenius norm could potentially further improve the deconvolution accuracy [275]. Furthermore, we applied RAD by considering a limited two-stage progression process, without use of the time-series information. One future direction might be extending RAD into a temporal model to take advantage of more precise information on time to metastasis when available or more extensive time-series data on multiple time points such as might be produced by “liquid biopsy” technologies. We mainly focused on transcriptome data in this chapter, but we expect the RAD algorithm to be versatile and potentially applicable to other types of continuous biological data, such as epigenome and proteome. With moderate adaptations, it is also possible to apply RAD to genome data, which is one major focus of cancer phylogenetics, such as copy number variations [66]. While much of the motivation for this chapter is the difficulty of acquiring scRNA for primary tumor samples when examining metastases years later, we do anticipate that this problem will lessen over time. It is thus worth considering for the future whether our methods might be adapted for working on limited and noisy scRNA with matched bulk data [68].

Chapter 6

Improving prognostic prediction of cancer by incorporating machine learning and evolutionary methods[1]

Cancers are typically caused by somatic genomic alterations accumulating under the forces of evolutionary diversification and selection that ultimately lead to uncontrolled cell growth [169]. In most cases, cancer progression is accelerated by somatic hypermutability, where defects in DNA replication or repair mechanisms cause the rapid acquisition of mutations across generations of cell growth [135]. Tumor cell populations thus typically undergo substantial genetic diversification over time, most of it likely selectively neutral but some with phenotypic effects [250], resulting in profound intra-tumor heterogeneity (ITH) [146], i.e., cell-to-cell variation in genetic makeup. Such heterogeneity in turn creates an opportunity for selection for mutations that promote uncontrolled cell growth or escape from normal controls on growth, leading ultimately to tumor development and potentially subsequent metastasis and patient mortality [81, 169]. This process of evolutionary "diversification" and "selection" further underlies the development of cancer recurrence and resistance to therapeutics [72]. Understanding the processes of somatic evolution that act in cancers is thus crucial to understanding why some precancerous lesions progress to cancer while others do not, why some cancers are highly aggressive while others are indolent, and why some respond robustly to treatment and others do not [199].

One of the key insights into cancer progression to derive from high-throughput sequencing studies is that mechanisms of somatic evolution can differ widely across cancers. Mechanisms of somatic hypermutability may differ between distinct patients for a single cancer type [161] and even between distinct cell lineages [177] or over time [107] in a single tumor. Different cancers may be prone to varying degrees of point mutation hypermutability, microsatellite instability, or chromosome instability [119]. Even within these broad classes, there are now numerous recognized mutational phenotypes presumed to be caused by distinct hypermutability mutations. For example, approximately thirty point mutation signatures [2, 3] are known to exhibit variability in different cancers, with several either known to be connected to specific kinds of hypermutability

[1] This chapter was developed from material published in "Yifeng Tao, Ashok Rajaraman, Xiaoyue Cui, Ziyi Cui, Haoran Chen, Yuanqi Zhao, Jesse Eaton, Hannah Kim, Jian Ma, and Russell Schwartz. Assessing the contribution of tumormutational phenotypes to cancer progression risk. *PLOS Computational Biology*, 17(3):1–29, 03 2021" [225].

defects (e.g., pol-ϵ defect [205], APOBEC defect [216], or various DNA mismatch repair defects such as those are induced by germline *BRCA1* or *BRCA2* mutations [108]), as well as distinct signatures of copy number or structural variation mechanisms [141, 240], such as those due to *TP53* dysfunction [26, 227]. At present, a number of these hypermutability signatures remain of unknown origin [2]. It remains elusive whether others might be detected as we gain better power to resolve broader classes of mutations and precisely quantify them via deep sequencing [110]. A variety of lines of evidence have suggested that these distinct hypermutability phenotypes have important implications for how a tumor is likely to evolve in the future. For example, it has been shown that tumors prone to copy number alterations (CNAs) and aneuploidy via whole genome duplication (WGD) have significantly worse prognoses than similar tumors only prone to focal CNAs [170, 185, 260]. Similar observations have appeared anecdotally for a variety of specific mutation classes.

Here, we sought to explore a vital implication of these past studies: how a tumor is likely to progress in the future is influenced by, and in principle predictable from, the mechanisms by which it has evolved so far, independent of the specific spectrum of driver mutations those mechanisms have so far produced. That is, the patient-specific spectrum of mutational phenotypes acting in a given tumor has predictive power for its future progression. For example, evolutionary statistics based on fluorescence *in situ* hybridization (FISH) data [116] are predictive of whether a tumor will go on to metastasize [43], with more nuanced models of mechanism and variation rate leading to enhanced predictive power [41, 42]. Conceptually, the use of mutational mechanisms as predictors is distinct from, and complementary to, the standard "driver gene" model of prediction – that we predict likelihoods of tumor progression based on its specific pattern of mutations or expression changes in genes of known functional significance in cancer [80] – which is the basis of much of the current work in genomic diagnostics for cancer. While conventional approaches to finding genomic predictors focus primarily on markers of the "selection" component of clonal evolution, seeking mutations with putative functional effects on clonal fitness, we here seek to understand the role of the "diversification (drift)" component of clonal evolution by profiling phenotypes that affect the degree and kind of mutations a tumor is prone to generate. We develop this idea of evolutionary predictors of progression by applying "tumor phylogenetics", i.e., the reconstruction of cell lineage histories in single tumors, to derive quantitative estimates of the evolutionary processes acting on those tumors from their phylogenies. The combination of mechanisms of mutation and the degrees to which they act in a given cancer should then, we propose, have independent predictive power from its specific driver mutations for future progression that we can quantify and harness via machine learning frameworks.

The concept "mutational phenotype" (which is captured by "evolutionary feature") proposed in this chapter is related to the concepts of tumor mutational burden (TMB) [34], mutational signature [2], and "mutator phenotype" [135], although distinct from each. Previous research sought to characterize hypermutability in tumors by inferring the TMB from either somatic or germline mutation counts, and found that tumors with higher TMB tend to be more responsive to immunotherapies [197]. Our focus here is on characterizing the ensemble of processes generating mutations and the degree to which they act in a given tumor and not on the resulting mutation burden per se. Mutational signatures capture this idea of an ensemble of processes generating distinguishable mutational patterns at the level of simple somatic mutations. Both TMB and mutational signatures consider the cumulative mutation counts in tumors. The evolutionary features proposed in this chapter are those that quantify differences in how a tumor generates mutational and clonal diversity independent of the specific mutations it has accumulated. Examples include variation across tumors in rates of mutation-generating processes, such as those captured by mutational signatures, and higher-level quantifications of evolutionary trajectories revealed from clonal lineage trees. More concrete examples of evolutionary features can be found in Tab. 6.2 and Tab. 6.3. We also note that we use the term "mutational phenotype" here rather than Loeb's term "mutator phenotype" [135] to describe the range of phenotypes of mutability we seek to quantify, despite the importance of Loeb's work in establishing the central idea of a phenotype of hypermutability in cancers, in order to avoid confusion with the more specific understanding of Loeb's mutator phenotype hypothesis.

The remainder of this chapter is devoted to implementing and demonstrating a realization of this idea of progression prediction from somatic hypermutability phenotypes with the goal of examining the degree to which a tumor's future progression is predicted by its mutational phenotypes independent of specific driver mutations. Below, we describe a general framework, which calls specific somatic variations from either whole exome sequence (WES) data or whole genome sequence (WGS) data, and uses a combination of tumor phylogeny models to extract features for use in predicting progression through regularized Cox regression. We demonstrate that mutational phenotypes are indeed significant predictors of progression outcomes, specifically via prediction of overall survival (OS) and recurrence/disease-free survival (DFS) on data from the Cancer Genome Atlas (TCGA) [29] and the International Cancer Genome Consortium (ICGC) [268], including breast invasive carcinoma (BRCA) [161] and lung carcinoma (LUCA) [162, 165]. In each case, we show that predictions from phylogeny-derived features quantifying mutational phenotypes have significant predictive power for progression outcomes and, further, that these phylogeny-derived evolutionary features provide additional predictive power relative to predictions from clinical or traditional driver-centric genomic features alone. The results demonstrate that variability in mutational phenotypes, and thus evolutionary diversification mechanisms, underlie a substantial portion of the risk of future progression of these cancers.

6.1 Materials and methods

6.1.1 Overall workflow

We assume in the analysis below the future progression of tumors depends on four categories of risk factors: evolutionary, driver, environmental, and unknown factors (Fig. 6.1A). We define evolutionary features as measures quantifying generic preferences for the tumor genome to evolve independent of specific genes affected by these preferences, for example, the overall point mutation rate or the preference for branched versus linear evolution. Driver features capture mutations in specific genes of known cancer relevance, such as point mutations in *TP53* or amplification of *ERBB2*. Environmental features capture additional measurable information about the patient beyond that derivable from the genome, such as demographic features. Unknown factors are presumed risk factors for which we lack measurements.

The general goal of this chapter is to evaluate the contribution to prognosis predictions from

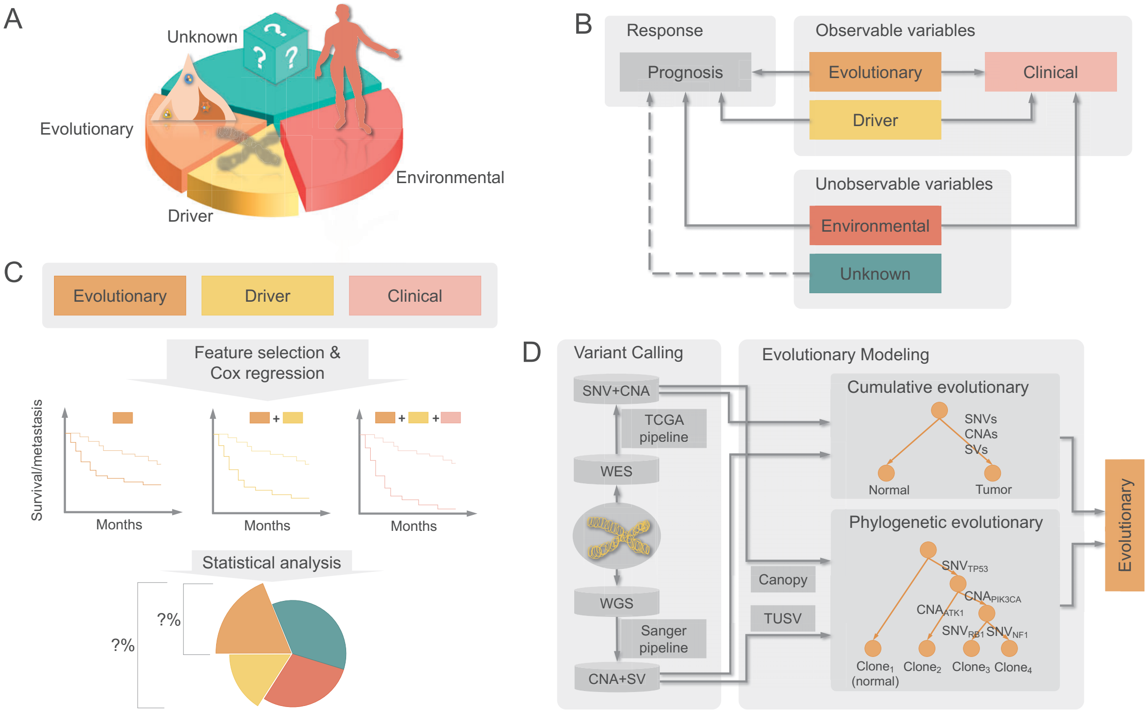

Figure 6.1: Overall workflow for evaluating the contribution from evolutionary features to tumor progression risk. (A) Various factors may account for future prognoses. We categorize these into evolutionary, driver, environmental, and unknown factors. (B) Only part of these factors are observable, making the evaluation of contributions complex. In addition, all three known factors have effects on outcomes as well as mutual correlations. (C) We conducted Cox regression to predict outcomes of patients, such as survival and recurrence, using combinations of evolutionary, driver, and clinical features, and analyzed their regression results to estimate the contribution of evolutionary features specifically to tumor progression risk. (D) The general framework utilizes either WES data followed by standard the TCGA variant calling pipeline or WGS data followed by Sanger variant callers to derive measures of mutational preferences from phylogenetic models of clonal evolution and cumulative mutation burdens.

evolutionary features describing mutational phenotypes relative to the independent contributions from other genomic features and to the full set of risk factors. The problem itself is challenging in large part because of the complex correlations between these variables, and the fact that interdependencies between many factors are unknown to us (Fig. 6.1B). For example, we might expect our evolutionary features to include measures of mutational process that are consequences of chromosome instability. These measures will thus be strongly correlated with driver features such as *TP53* mutation that cause chromosome instability. Our goal is to evaluate the portion of such risk attributable specifically to the mutational phenotypes, independent of other consequences of those mechanisms.

We propose, indirectly, to first evaluate the relative contributions of these features to progression risk, through their predictive power in a machine learning analysis using Cox regression, either alone or in combination with additional clinical features and driver-centric genomic features (Fig. 6.1C). We then can use the results of these predictive tests to infer how each kind of feature contributes to overall progression risk. We utilize the ℓ_0-regularized Cox regression model, which is tuned heuristically through step-wise forward feature selection, throughout the chapter to exploit the power entailed by different sets of features. It achieves better performance than the widely used lasso Cox regression model. We also employ bootstrapping in the Cox model to identify the essential features and capture the interactions across them. The bootstrapping assigns interpretability to the models and is robust to correlated factors.

The process of extracting evolutionary features begins with tumor genome sequencing data (Fig. 6.1D). In principle, these data might be whole exome sequence (WES) or whole genome sequence (WGS), and could be either single-sample (solely tumor data), paired tumor/normal, or multi-sample (multiple distinct tumor sites or regions as well as possibly paired normal). While some study designs might alternatively use targeted deep sequencing data, we would generally consider those data not suited to the present methods, which benefit from profiling larger fractions of the genome to estimate better aggregate mutation rates. We consider here only inference from bulk tumor data [222, 223], although we note that the strategy might be applied to single-cell sequence data [158] or combinations of bulk and single-cell data [124, 125, 142], should such data become available for sufficiently large cohorts. The genomic data is preprocessed and passed to one or more variant callers, ideally including single nucleotide variations (SNVs) and copy number alterations (CNAs) calls as well as calls for diverse classes of structural variations (SVs) to produce a variant call format (VCF) file with detected variants and their variant allele frequencies (VAFs) per sample. These variant calls are then fed into tumor evolution algorithms, which deconvolve aggregate bulk data into multiple clonal evolutionary states and infer a cellular lineage tree connecting those states and predicting their likely ancestry. Next, a variety of quantitative measures of the evolutionary process, i.e., “evolutionary features”, corresponding to distinct mutation mechanisms, are extracted. These features intend to approximate the degree to which distinct “mutational phenotypes” are activated in a tumor.

In order to validate the effectiveness and generality of our hypothesis and approach, we compiled two sets of data testing various conditions under which the approach might be applied (Tab. 6.1). These datasets cover: two data sources: TCGA [29], ICGC [263, 268]; two cancer types: BRCA [161], LUCA [162, 165]; two sequencing strategies: WES, WGS; two variant callers: TCGA pipeline [29, 44, 70, 151], Sanger pipeline [28]; two phylogenetic methods: Canopy [102] (SNV+CNA), TUSV [66] (CNA+SV); two prognostic prediction tasks: OS, DFS.

We select breast and lung cancers for validation primarily because they are the most common cancer types and have relatively large TCGA and ICGC cohorts. We treat the two datasets separately and assume all the samples within the same dataset are processed with the same experimental protocol. This avoids the potential confounding factors that may hide behind the datasets. For example, samples from WGS ICGC cohort generally exhibit much greater genome coverage than those from the WES TCGA.

Table 6.1: Statistics of the experiments and datasets in the study. We collect two sets of data covering two cancer types (BRCA, LUCA), two experiment strategies (WES, WGS), two variant callers (TCGA pipeline, Sanger pipeline), two phylogenetic models, and two prediction tasks (OS, DFS).

	Seq	Cancer	Variant	Phylogenetic		#Event/#Censored	
Dataset	Strategy	Type	Caller	Model	Size	OS	DFS
TCGA	WES	BRCA	TCGA	Canopy	1044	145/897	102/764
		LUCA	TCGA	Canopy	512	180/323	180/266
ICGC	WGS	BRCA	Sanger	TUSV	90	17/73	14/59
		LUCA	Sanger	TUSV	89	44/43	29/38

6.1.2 Variant calling and evolutionary modeling

For each cancer sample, somatic genomic variants were first called by a range of possible tools as discussed below, then the SNVs, CNAs, and SVs were integrated and converted into a single VCF file, which is the input format required by both evolutionary tree methods considered, Canopy [102] and TUSV [66]. Both methods output fractions of clones in each tumor sample, the inferred phylogenetic tree connecting clones, and the acquired somatic variants of clones during evolution along the tree edges.

We made use of calls from two different variant callers in the experiments, according to the sequencing strategy of the samples. For the WES BRCA and LUAD samples from TCGA corpus (Tab. 6.1 TCGA), we downloaded SNVs and CNAs from the TCGA Genomic Data Commons Data Portal (GDC) [159]. It provides calls from the TCGA pipeline [29], which uses a consensus of standard variant callers such as MuSE [70], MuTect2 [44] and GISTIC2 [151]. We made use of WGS samples from the ICGC/PCAWG project [268], which provides one of the largest public corpora of WGS samples for breast cancer and lung cancer (Tab. 6.1 ICGC). For these samples, we downloaded SNV, CNA, and SV calls [97], which had been computed for this project using the Sanger pipeline [28]. The users can also use other variant callers such as Weaver [127, 186] and novoBreak [40] for the WGS samples. We pooled the two cohorts of lung cancer: LUAD and LUSC into a single LUCA category to increase the size of the dataset.

We applied two general approaches to derive tumor phylogenetic trees, based on the availability of WGS and WES data. We consider the two data types primarily to assess whether the major conclusions of the analysis are robust to the differences in data type, and particularly whether they are manifest in more widely available WES data despite its more limited utility for assessing mutational phenotypes. WGS data provides a much better ability to call SVs. To assess the predictive value of SVs, we sought to capitalize on this capability by building phylogenies using a customized version of the TUSV phylogeny software [66], which infers phylogenies from SVs and CNAs and is, to our knowledge, the only tumor phylogeny program currently able to incorporate SVs into its trees. For WES data, we instead used the third-party tool Canopy [102], which makes inferences from SNVs and CNAs.

Note that both the clonal phylogenetic models, including Canopy and TUSV, infer the most likely trees based on the sequencing data available. Since the samples available for the patient are limited, the estimation of phylogenies may be noisy and inaccurate. However, We can still take advantage of the useful information hidden behind these phylogenetic evolutionary features of large scale samples through the well-designed machine learning model proposed in this chapter.

We finally added a trivial evolutionary model, which we dub the "cumulative evolutionary" model, in contrast to the more complex tree models inferred by Canopy or TUSV. This cumulative model is intended as an aggregate approximate model of evolutionary preferences derived from overall mutation burdens. For this model, we assume that there is a single branch of evolution from normal to cancer, resulting in a diploid tree of a "normal" root and single derived cancer state. This two-node cumulative evolutionary tree gives a crude approximation to evolutionary rates that requires minimal assumptions about the underlying evolutionary model and data types.

6.1.3 Feature extraction and preprocessing

Our experiments made use of two types of genomic features (evolutionary and driver features) and clinical features, assumed to be data that would typically be available for diagnosis in clinical practice.

Evolutionary features We first derived a set of cumulative evolutionary features of samples from overall mutation burdens subdivided by mutation class, and analyzed them as if they were derived from two-node evolutionary trees (Tab. 6.2). Although we might have directly selected the known mutational signatures as part of the evolutionary features (e.g., *G*→*C* using the COSMIC database), we instead chose to include as many potential evolutionary features as possible including trinucleotide frequencies not corresponding to any common signature. In part, this enabled us to identify the useful features outside of our current knowledge from literature. It also allowed us to consider that informative evolutionary features might be different across different platforms due to different sequencing methods and variant calling pipelines. In addition to the total SNV mutation rates (*snv rate*), we used mutation rates for different types of SNVs, e.g., *T*→*A*. We took *G*→*T*, *G*→*C*, *G*→*A*, *A*→*T*, *A*→*G*, *A*→*C* equivalent to *C*→*A*, *C*→*G*, *C*→*T*, *T*→*A*, *T*→*C*, *T*→*G* since they are equivalent. For example, when a mutation *G*→*T* is observed in one DNA strand, there must be another mutation *C*→*A* in the complementary strand, and we treat the two as equivalent. There are therefore $4 \times (4-1)/2 = 6$ such mutation patterns in total. We also broke these down further into the trinucleotide context, as is typically done in mutational signature analyses [2]. A trinucleotide mutation has the general form of $N_l N_x N_r \rightarrow N_l N_y N_r$, which represents the muation of $N_x \rightarrow N_y$ with the left and right contextual nucleotide N_l and N_r. There are $4 \times 6 \times 4 = 96$ trinucleotide mutation features in total. Apart from the mutation rates related to simple somatic mutations, we also estimated various aggregate mutation rates of CNAs and SVs, e.g., *cna rate*, *sv rate*. Similar to the mutational signatures of SNVs, we tried to characterize the copy number-level equivalents by taking into account the size of CNA region and duplication vs. deletion of the CNA, e.g., rates of CNA above 500,000 nt (*cna lg rate*), CNA deletion rates (*cna del rate*) etc. In each case, we treated mutations as occurring on a single evolutionary tree edge spanning from normal to tumor. When the normal state is unknown, we screened out sites of common germline single nucleotide polymorphisms (SNPs) using germline SNP data from the 1000 Genomes Project [228], and used deviation from a standard human reference (GRCh38/hg38 for TCGA, GRCh37/hg19 for ICGC) [99]. We assumed a fixed edge length of ten years as an estimated time from the appearance of the first ancestral tumor cell to the time of sequencing in order to provide a scaling factor to convert mutation counts into estimated rates, although we note that the scale is arbitrary and does not affect the machine learning inference.

Table 6.2: List of cumulative evolutionary features. The mutation rates related to SNVs, CNAs and SVs of samples are included. All cumulative evolutionary features are in continuous value. We have 6 mutation rates and 96 trinucleotide mutation rates as features in total.

		Included	
Cumulative Feature	Definition	WES	WGS
$T \to A$	mutation rates	✓	✓
$T \to G$	mutation rates	✓	✓
$T \to C$	mutation rates	✓	✓
$C \to A$	mutation rates	✓	✓
$C \to T$	mutation rates	✓	✓
$C \to G$	mutation rates	✓	✓
$N_l N_x N_r \to N_l N_y N_r$	trinucleotide mutation rates (96 in total)	✓	✓
snv_rate	total SNV rates	✓	✓
cna_rate	total CNA rates	✓	✓
cna_amp_rate	CNA duplication rates	✓	✓
cna_del_rate	CNA deletion rates	✓	✓
cna_lg_rate	rates of CNA above 500,000 nt	✓	✓
cna_sm_rate	rates of CNA below 500,000 nt	✓	✓
sv_rate	total SV rates		✓

Second, we added a set of features derived from the more involved evolutionary trees built by Canopy or TUSV. After a phylogenetic evolutionary tree was built for WES by Canopy, or for WGS by the extended TUSV, measures that quantify topological features of the tree were extracted (Tab. 6.3). Since the outputs of TUSV include extra SV information, we have some additional phylogenetic features for WGS samples. However, there are still some common tree features, such as the clone number (*num clone*), the height of the phylogeny (*height*) and the average of edge lengths (*branch mean*) that are conserved between phylogeny inference methods.

Table 6.3: List of phylogenetic evolutionary features. Due to the different output of Canopy (phylogenetic model for WES) and TUSV (phylogenetic model for WGS), the sets of phylogenetic features are slightly different. The WGS data contain additional features related to CNA and SV rates.

		Included	
Phylogenetic Feature	Definition	WES	WGS
num_clone	clone number	✓	✓
diversity	diversity of clone proportions	✓	✓
lg_clone_proportion	proportion of the largest clone	✓	✓
lg_clone_snv	SNV rates in the largest clone	✓	
lg_clone_cna	CNA rates in the largest clone	✓	✓
lg_clone_sv	SV rates in the largest clone		✓
height_topology	topological height of phylogeny	✓	✓
height	height of phylogeny	✓	✓
height_cna	height of phylogeny in unit of CNA rates		✓
height_sv	height of phylogeny in unit of SV rates		✓
branch_num	number of edges in phylogeny	✓	✓
branch_len	total edge lengths	✓	✓
branch_mean	average of edge lengths	✓	✓
branch_mean_cna	average of edge lengths in unit of CNA rates		✓
branch_mean_sv	average of edge lengths in unit of SV rates		✓
branch_max	maximum edge length	✓	✓
branch_max_cna	maximum edge length in unit of CNA rates		✓
branch_max_sv	maximum edge length in unit of SV rates		✓
branch_var	variance of edge lengths	✓	✓
branch_var_cna	variance of edge lengths in unit of CNA rates		✓
branch_var_sv	variance of edge lengths in unit of SV rates		✓

Driver features The potential drivers of BRCA and LUCA came from two sources. First, we used the IntOGen database [80], where the top 20 drivers based on mutation counts in samples of each cancer type were collected. Second, we used the COSMIC database [226], where 20 common drivers were collected. See Tab. 6.4 for the full list of potential drivers of both BRCA and LUCA. We counted the times that a driver was perturbed by SNVs, indels, CNAs or SVs. Examples of common drivers are *TP53*, *PIK3CA*, and *GATA3*.

Table 6.4: List of driver features. The potential drivers come from both IntOGen and COSMIC databases. BRCA and LUCA share a large portion of drivers. We count the somatic mutation rates of both SNVs, indels, CNAs, and SVs in all drivers as the driver features. These features are in continuous value.

	Included	
Driver Feature	BRCA	LUCA
TP53, *PIK3CA*, *GATA3*, *MLL3*, *CDH1*, *NCOR1*, *MAP2K4*, *PTEN*, *AKT1*, *RUNX1*, *NF1*, *RB1*, *ARID1A*, *TBX3*, *MLL2*, *SPEN*, *LRP1B*, *ESR1*, *KMT2C*, *KMT2D*, *FOXA1*, *ERBB2*	✓	✓
MAP3K1, *MACF1*, *MED12*, *ATM*, *AKAP9*	✓	
KEAP1, *CDKN2A*, *KRAS*, *EGFR*, *STK11*, *KDR*, *FAT1*, *SVEP1*, *NFE2L2*, *FN1*, *NOTCH1*, *MLL*		✓

Clinical features All the clinical data for the TCGA samples were extracted from TCGA-reported clinical metadata downloaded from GDC. The clinical features of ICGC/PCAWG samples also came from GDC, as the samples sequenced using WGS are a subset of the TCGA samples. We collected the clinical feature set which provides a consensus representation of information likely to be available to clinicians at the time of diagnosis. We then removed the features that are available for fewer than half of the samples (Tab. 6.5). We note that a large portion of these clinical features are in common between BRCA and LUCA. Examples of preserved clinical features are *person neoplasm cancer status*, *pathologic stage*, and cancer subtype (*histological type*).

Feature engineering We preprocessed, encoded, and imputed the clinical features according to their value types. For continuous values, missing values were filled with the median value of the cohort. For binary values, missing values were filled with the mode of the cohort, and the features were encoded by 0/1. For non-binary categorical values, missing values were filled with the mode, and a feature of *k* categories was mapped into *k* mutually exclusive binary features. We removed the category that appears least frequently to avoid collinearity. We made $\log_2(x+1)$ transformation to the features that have long-tail distributions, including driver, cumulative evolutionary, and most of the phylogenetic evolutionary features, so that the resulting feature distribution is close to a normal distribution. We removed the sparse clinical and genomic

Table 6.5: List of clinical features. BRCA and LUCA samples share a large portion of similar clinical features. Three data types are available: binary, categorical and continuous. Cancer subtype is denoted as *histological type*.

		Included	
Clinical Feature	Data Type	BRCA	LUCA
person neoplasm cancer status	binary	✓	✓
gender	binary	✓	✓
history of neoadjuvant treatment	binary	✓	✓
ethnicity	binary	✓	✓
pathologic stage	categorical	✓	✓
histological type	categorical	✓	✓
race	categorical	✓	✓
age at initial pathologic diagnosis	continuous	✓	✓
lab procedure her2 neu in situ hybrid outcome type	binary	✓	
breast carcinoma progesterone receptor status	categorical	✓	
breast carcinoma estrogen receptor status	categorical	✓	
lab proc her2 neu immunohistochemistry receptor status	categorical	✓	
margin status	categorical	✓	
her2 immunohistochemistry level result	categorical	✓	
number of lymphnodes positive by he	continuous	✓	
cytokeratin immunohistochemistry staining method micrometastasis indicator	continuous	✓	
menopause status	categorical	✓	
her2 neu chromosone 17 signal ratio value	continuous	✓	
her2 immunohistochemistry level result	continuous	✓	
her2 erbb pos finding cell percent category	continuous	✓	
fluorescence in situ hybridization diagnostic procedure chromosome 17 signal result range	continuous	✓	
anatomic neoplasm subdivision	categorical		✓

features that are non-zero in fewer than 5% samples. Finally, we mapped all features into the interval $[0, 1]$ linearly.

6.1.4 Cox regression and its regularized variants

We utilized ℓ_0-regularized Cox regression model throughout this chapter, using step-wise forward feature selection as a heuristic strategy to tune hyperparameters and select features.

Proportional hazards and Cox regression The clinical prognostic outcomes of cancer patients, such as OS and DFS, are censored data, meaning that the time to a death event or recurrence event was not observed for some samples due to the limited follow-up time. Therefore, instead of conventional classification or regression methods, we performed Cox regression [46] and evaluated the prediction performance with metrics for survival analysis, which are specifically designed to cope with these censored data. In our formulation, the samples are

$$\{(X_i, y_i, \delta_i)\}_{i=1}^{N}, \tag{6.1}$$

where N is the total number of samples, $X_i \in \mathbb{R}^m$ is the feature vector of sample i, $\delta_i \in \{0, 1\}$ indicates the status of patient i at the last follow-up time $y_i = \min(T_i, C_i)$: If $\delta_i = 1$, the event happened and was observed at time $y_i = T_i$. If $\delta_i = 0$, the event had not happened at the censoring time $y_i = C_i$.

Cox regression is a semi-parametric regression method based on the proportional hazards (PH) assumption:

$$h_i(t) = h(t) \cdot \exp\left(-\beta^\intercal X_i\right), \tag{6.2}$$

where $h_i(t) \coloneqq \lim_{\Delta t \to 0} \Pr(t < T_i \leq t + \Delta t \mid T_i > t)/\Delta t$ is the hazard of patient i at time t, or in another word, the probability of death if the patient has survived to time point t (for OS; similarly it is the hazard of recurrence for DFS), $h(t)$ is the non-parametric part calculated from the training data, X_i is the feature vector of sample i, $\beta \in \mathbb{R}^m$ is the model parameter to be estimated. Instead of predicting whether the patient will be dead or alive, Cox regression estimates β and provides the hazard of the patient following Eq. 6.2. When the total number of samples is large, we can roughly assume $h(t)$ to be close enough at the time of cross-validation (CV). The comparison of $h_i(t)$ thus reduces to the comparison of risk score $\eta_i = -\beta^\intercal X_i$, i.e., the logarithm of hazard's parametric part $\exp\left(-\beta^\intercal X_i\right)$, which is independent of time t [243]. There exist other approaches as well, e.g., by comparing the expectation of the survival time. However, we found that the comparison of risk score is more robust and reliable in our experiments. We implemented the Cox regression and its variants using the Python package `lifelines` [49].

Cox regression without penalty terms At the time of training, we optimized the negative log-partial likelihood function of the Cox model:

$$\min_{\beta} \; l\left(\beta \;\middle|\; \{(X_i, y_i, \delta_i)\}_{i=1}^{N}\right). \tag{6.3}$$

Since the dimension of the features is large and the features are correlated in our application, the `lifelines` package is unstable in optimizing the objective function Eq. 6.3. In addition, such large feature dimension can lead to severe overfitting, meaning that we can achieve perfect performance on the training set, but the performance is not generalizable to the test set. We considered two Cox variants in this chapter.

ℓ_1-regularized Cox regression (lasso) Lasso is one of the most widely used sparsity-promoting regularization methods to cope with the high-dimensional feature space. It adds an ℓ_1 regularization term to Eq. 6.3:

$$\min_{\beta} \; l\left(\beta \;\middle|\; \{(X_i, y_i, \delta_i)\}_{i=1}^{N}\right) + \lambda \left\|\beta\right\|_1 , \tag{6.4}$$

where the coefficient $\lambda \in \{0.03, 0.1, 0.3, 1.0, 3, 10, 30\}$ was tuned through inner CV. We showed that lasso generally performed worse than ℓ_0-regularized Cox regression (Tab. 6.7).

ℓ_0-regularized Cox regression Instead of the ℓ_1 penalty, ℓ_0-regularized Cox regression selects the subset of features by optimizing the following objective:

$$\min_{\beta} \; l\left(\beta \;\middle|\; \{(X_i, y_i, \delta_i)\}_{i=1}^{N}\right), \quad \text{s.t. } \left\|\beta\right\|_0 \leq k, \tag{6.5}$$

where $\left\|\beta\right\|_0$ is the ℓ_0-norm that counts the number of nonzero elements in coefficient vector β. The integer parameter k is tuned at the training and validation phase to achieve the highest inner CV performance. As we will discuss in the Section "Tuning ℓ_0-regularized Cox model: subset selection", it is computationally infeasible to find the exact solution of Eq. 6.5 and we therefore took a heuristic step-wise forward selection strategy to find a possibly suboptimal solution.

6.1.5 Evaluation metrics and two-loop cross-validation

With the predictions of risk scores, we can evaluate prediction results based on an assessment of concordance index (CI) [73, 211], and an assessment of the statistical significance of separating censored survival data using a logrank test [145]. The CI is defined as the following:

$$\text{CI} = \frac{\sum_{i,j} \delta_i \cdot \mathbb{1}\left(y_j > y_i,\ \eta_j < \eta_i\right)}{\sum_{i,j} \delta_i \cdot \mathbb{1}\left(y_j > y_i\right)}, \tag{6.6}$$

where $\mathbb{1}$(statement) is the indicator function. CI is a value similar to area under curve (AUC). A model reaches perfect prediction when CI $= 1$ and random guess when CI $= 0.5$. The logrank test uses a statistical test to accept or reject the null hypothesis $\mathcal{H}_0$: two groups of samples share the same survival profile. It calculates a statistic z^2 from observations of two groups of censored data. While $z \xrightarrow{d} \mathcal{N}(0, 1)$, we can get the *p*-value to accept or reject $\mathcal{H}_0$. In our experiments, we split the samples in the test sets into two groups with the median of risk scores $\{\eta_i\}_{i=1}^{N}$, and used the logrank test to evaluate the differences between these two cohorts of predicted malignant and benign samples.

We used a two-loop cross-validation (CV) protocol to tune hyperparameters and evaluate the performance of different models, which consists of the inner CV and outer CV (Fig. 6.2). It is also referred to as "nested cross-validation" in the literature [32]. The inner CV is used for tuning parameters on the training set, i.e., λ in lasso model, k (and selected features) in the ℓ_0-regularized model. The outer CV is used for **unbiased evaluation** on the test sets. This means that although the model may overfit at the inner CV time on the training sets due to larger model complexity or feature dimension, the final test sets at the outer CV are unseen to the model. By reporting the evaluation results only on the outer CV test sets, the evaluation protocol avoids the problem that a model with larger complexity tends to overfit and perform better through only inner CV. This is especially a concern when we compare the same model type with different features (Tab. 6.6 "clinical" vs. "full"), and when we compare the performance of different models such as ℓ_0- and ℓ_1-regularized Cox regression models (Tab. 6.6 vs. Tab. 6.7). Note that we did not use a fixed training and test set split to replace the 3-fold outer CV. To some extent, the outer CV avoids the problem that test set may not be representative of the full data since we evaluated the full data during the outer CV.

6.1.6 Tuning ℓ_0-regularized Cox model: subset selection

Heuristic subset selection: step-wise forward feature selection After extracting the three types of features, we aimed to utilize the ℓ_0-regularized Cox regression for modeling and prediction. However, the hyperparameter k in the objective function Eq. 6.5 is hard to tune. Given the $\beta \in \mathbb{R}^m$, where m ranges between 100 and 200 in our case, we have to train and validate the 2^m models in a brute force way (or slightly faster using mixed integer linear programming [20]) to find the model with the optimal hyperparameter k, which is computationally intractable. Therefore, we propose using step-wise feature selection as a confrontation approach to finding suboptimal solutions in Eq. 6.5. It is hard in our case to assess how well the forward selection heuristic approaches an optimal solution, since training and tuning the model optimally is a computationally intractable problem. However, a recent study on simulated data found step-wise forward selection performs only marginally worse than, if not as well as the optimal subset selection [87]. The validation tests of our method described below also suggest that it provides good solutions in practice. Careful readers may find the ℓ_0-regularized problem will be slightly modified here if we use the heuristic step-wise feature selection method, because at the end of the step-wise feature selection through inner cross-validation (CV), we find/tune both the hyperparameter k and selected features, instead of just k. The selected features will be fixed at the time of evaluating the test set, not flexible.

In step-wise forward feature selection [111], we first traverse all the individual features in the candidate feature set, and evaluate the performance in concordance index (CI) of these features through the inner CV. The single feature with the best CV performance is selected. Secondly, we continue to traverse all the other unselected individual features in the candidate set, concatenate it with the first selected feature and evaluate the joint performance through the inner CV. The feature with the best performance is then taken as the second selected feature. We continue such a process to find the third, fourth, and fifth selected features until the inner CV performance does not increase. We implemented the step-wise forward selection of evolutionary, driver, clinical, genomic, and full feature sets using inner 3-fold CV in TCGA dataset and inner leave-one-out

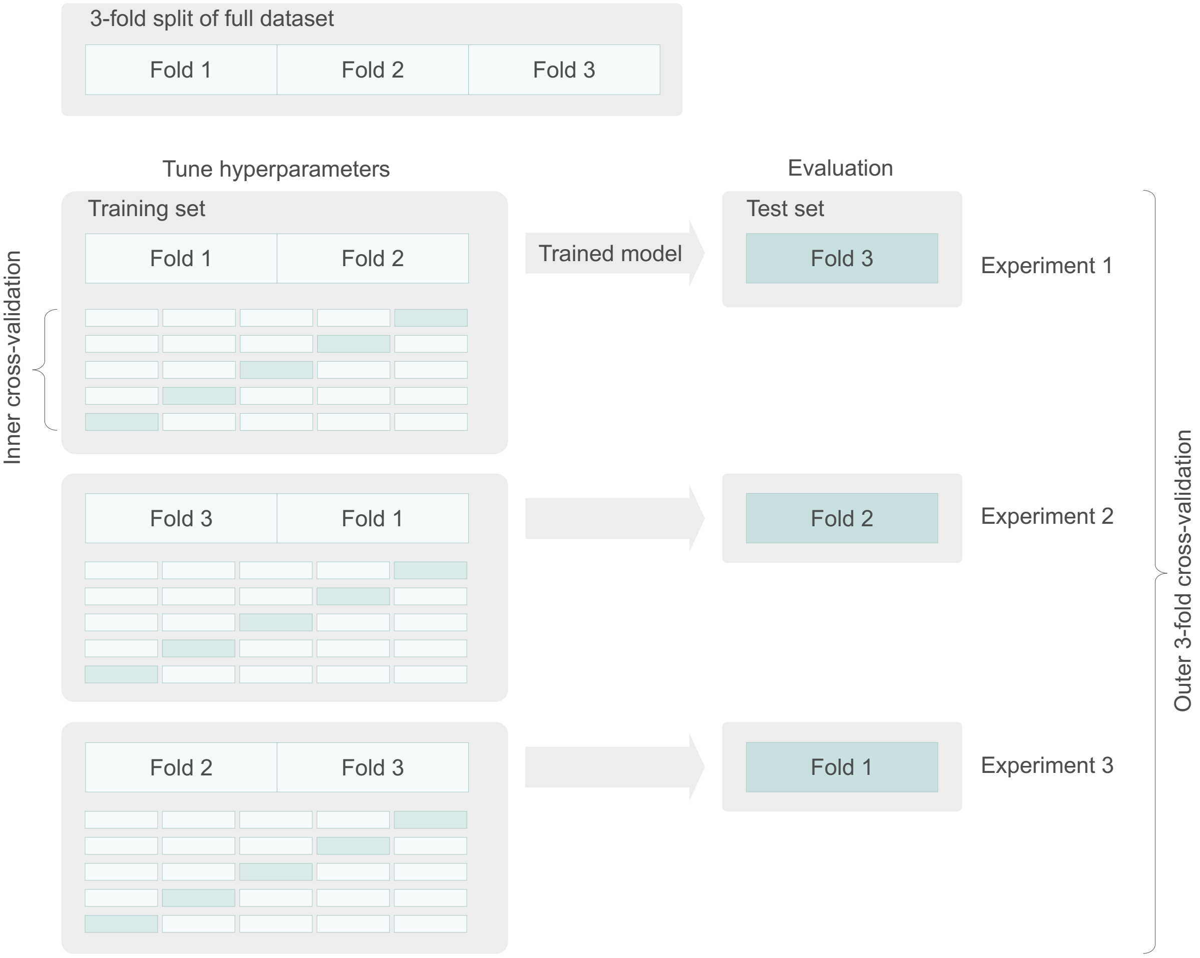

Figure 6.2: Procedure of training, tuning, and unbiased evaluation through two-loop cross-validation. The whole dataset is split into three parts and evaluated through 3-fold outer cross-validation (CV) on the test sets. In each experiment of the outer CV, the ℓ_1- or ℓ_0-regularized model is tuned using an inner CV only on the training set. We used 3-fold inner CV for TCGA and leave-one-out inner CV (LOOCV) for ICGC throughout the work. We employed LOOCV for ICGC because it has a much smaller sample size. The utilization of two-loop CV prevents the problem of bias when we evaluate models with different complexities, e.g., Cox model using clinical features vs. Cox model using both clinical and genomic features, or ℓ_1-regularized model vs. ℓ_0-regularized model. In contrast, the model with larger complexity tends to perform "better" due to overfitting if using the single-loop CV.

CV in ICGC dataset [73, 211]. We notice such a protocol is prone to overfit at the phase of inner CV compared with lasso. Other heuristics such as BIC may be appropriately revised and applied (though not trivial in Cox regression) to stop the forward steps early [201], which may potentially improve the model performance further by reducing the model complexity. However, even with the inner CV overfitting, ℓ_0-regularized Cox model still outperforms a more conservative lasso Cox model on test sets of the outer CV (Tab. 6.6 vs. Tab. 6.7).

Feature collinearity, bootstrapping, model interpretability and robustness The correlations across features may lead to multicollinearity and fragile models, making the models hard to interpret. Prior to the step-wise forward selection tuning phase, we first randomly removed one of the features if any two features are highly correlated. We did not keep them both because technically the multicollinearity across features [20] could cause the numerical problems when taking the inverse of the covariance matrix, and lead to the `lifelines` optimizer failing to solve the problem. From the aspect of model performance, two features that are highly correlated capture similar information and it is less likely for the model to improve predictive power by taking both features as input. From the aspect of interpretability, the coefficients of the two correlated variables estimated from multivariable regression are unstable and inaccurate, making it hard to evaluate the importance of the feature. We used a Pearson coefficient threshold of 0.8 in absolute value to filter features, which is a commonly used threshold empirically [20]. Although a more carefully chosen threshold may potentially further improve the model performance, in practice, the bootstrapping (discussed below) is robust to the different thresholds in evaluating feature importance.

We noticed that such removal of features did not fully resolve the problem of model interpretability. For example, if both features are informative to the model but highly correlated, only one of the features will be kept. To better identify which individual features are predictive, we conducted bootstrapping. We drew 80% candidate features and 80% samples with replacement for 1,000 trials, and randomly selected one of the two features every time. The informative/essential features will be the ones selected frequently in bootstrapping. We refer readers to Sec. "Informative features for predicting tumor future progression" for two other approaches to evaluate feature importance in the context of feature collinearity, and their disadvantages compared with the bootstrapping.

Apart from the bootstrapping approach, there could be other solutions to address the collinearity problem, e.g., by pre-clustering the features into independent groups. However, the clustering method may also suffer in situations where one of the features can be represented as a complex linear combination of others, but no two features are highly correlated.

6.1.7 Estimation of contribution from evolutionary features to progression risk

In multivariable regression, the different features may be intertwined in a complicated way. Therefore, the contributions of different features may be different when be placed together with other relevant features. We proposed a log-hazard-ratio-based fraction metric to calculate the fractions of contributions from evolutionary or genomic features among all the available data to the prognoses predictions [245]. The fraction metric normally lies within the range $[0, 1]$, where zero means the evolutionary or genomic features do not contribute to the prediction at all, while one means the evolutionary or genomic features can explain all of the predictions even if other available features are provided. Theoretically, the fraction metric can be greater than one, which means the additional clinical or driver features are harmful to the model performance. However, empirically this does not happen in our experiments as clinical features are always helpful.

We define the log-hazard-ratio-based fraction of evolutionary features as below:

$$\text{fraction(evolutionary)} = \frac{\log \mathbf{HR}(\text{evolutionary})}{\log \mathbf{HR}(\text{evolutionary+driver+clinical})}, \tag{6.7}$$

where hazard ratio (HR) is defined as a value in $[1, +\infty)$ when the feature set is informative of survival/recurrence:

$$\text{HR} = \exp\left(-\sum\nolimits_{i \in \mathcal{I}_M} \beta^\intercal X_i\right) \Big/ \exp\left(-\sum\nolimits_{i \in \mathcal{I}_B} \beta^\intercal X_i\right). \tag{6.8}$$

where $\mathcal{I}_M$ is the index set of samples predicted to be malignant (hazard larger than the median hazard), and $\mathcal{I}_B$ is benign (hazard smaller than the median values).

We can use similar rules to define the contribution fractions of genomic features:

$$\text{fraction(genomic)} = \frac{\log \mathbf{HR}(\text{evolutionary+driver})}{\log \mathbf{HR}(\text{evolutionary+driver+clinical})}, \tag{6.9}$$

We use the log-scale HR instead of raw HR because the HR itself is calculated in the exponential scale of features. Taking the log allows us to directly compare the metric in the scale of the original feature space. Note that the fraction and CI are two different ways of evaluating the contribution and effectiveness of evolutionary features that would be expected to yield quantitatively different outcomes. The fraction aims to explain the feature contribution in the linear space of features, while CI aims to evaluate the effectiveness of features through performance. The calculation of fraction does not take into consideration unobservable variables, while the value of CI is affected by how much information is available in all the observed variables relative to unknown factors.

We calculated the HRs on the test sets at the outer CV phase for fraction estimation. The experiments were replicated five times to derive the mean HRs for the subsequent fraction estimation.

6.2 Results

We first describe the overall workflow by which we extracted evolutionary and other features, and the methods we used to predict tumor progression and estimate the contribution of each feature class to the progression risk. We then provide an analysis of the genomic and clinical features predictive of tumor progression derived from our analysis pipeline. We then take a closer look into the overall feature landscape and interactions among the different feature types. These components then bring us to the major conclusion of this chapter, namely, features describing how a tumor is evolving contribute substantially to tumor progression risk, underlying a large fraction of the predictive power of other feature types as well as augmenting it. Finally, we expand on that main conclusion to examine the degree to which the additional information provided by evolutionary features can further improve prognostic prediction beyond what is available from traditional clinical and driver features alone.

6.2.1 Informative features for predicting tumor future progression

We have around 100 to 200 features in total in each dataset (Tab. 6.2-6.5). However, not all these features may be predictive of the future progression of tumors. Therefore, we first set out to evaluate their importance individually. We provided evolutionary, driver, and clinical feature sets separately as candidate features, and fed them into the ℓ_0-regularized Cox regression model for 3-fold cross-validation and feature selection. We conducted bootstrapping by drawing 80% features and 80% samples with replacement for 1,000 trials. These replicates provide the frequency with which each feature is selected. We treat this frequency as a measure of the degree to which a specific feature is informative in the model (Fig. 6.3,6.4), instead of performance or regression coefficient.

There exist alternative ways to evaluate feature importance. A straightforward approach is to conduct univariate regressions for all features and take the concordance index performance as the importance metric. However, a feature that helps achieve the best performance together with other features may not be significantly effective alone in univariate regression. Therefore, it is necessary to use multivariate regression to identify these features. A second solution is to conduct a round of cross-validation using multivariate Cox regression to find the selected subset of features and take their regression coefficients as the effect on the prediction. However, if two features are highly correlated and both are important, the optimal feature subsets are not unique. Moderate correlation can problematic as well, leading to unreliable estimation of the coefficients. The bootstrapping combined with the multivariate regression take advantage of both capturing complex interactions between features and responses (compared with univariate regression) and robustness of selecting features (compared with a single run of the multivariate model). Bootstrapping also has the advantage when potential strong factors exist, such as tumor status, in preventing those strong features from obscuring effects of weaker features. The importance order of most other features will change a little after removing it.

Evolutionary features The most strongly predictive evolutionary features vary by dataset, tumor type, and task. Some trinucleotide SNV mutation rates show up as significant in most cases examined (Fig. 6.3,6.4 evolutionary bars) [135]. Trinucleotide SNV rates of the form $N_lCN_r \rightarrow N_lTN_r$ are especially important for the breast cancer prognostic prediction. $C \rightarrow T$ (and, equivalently, $G \rightarrow A$) preferences are associated with several mutational signatures [2] and we would hypothesize that their association with age-related signatures largely accounts for their predictive power here (Fig. 6.7A).

There exist systematic variances across the TCGA and ICGC datasets, which reflect the differences in the sequencing techniques (WES vs. WGS), variant caller pipelines (TCGA pipeline vs. Sanger pipeline), and phylogenetic models (Canopy vs. TUSV). SNV-related trinucleotide

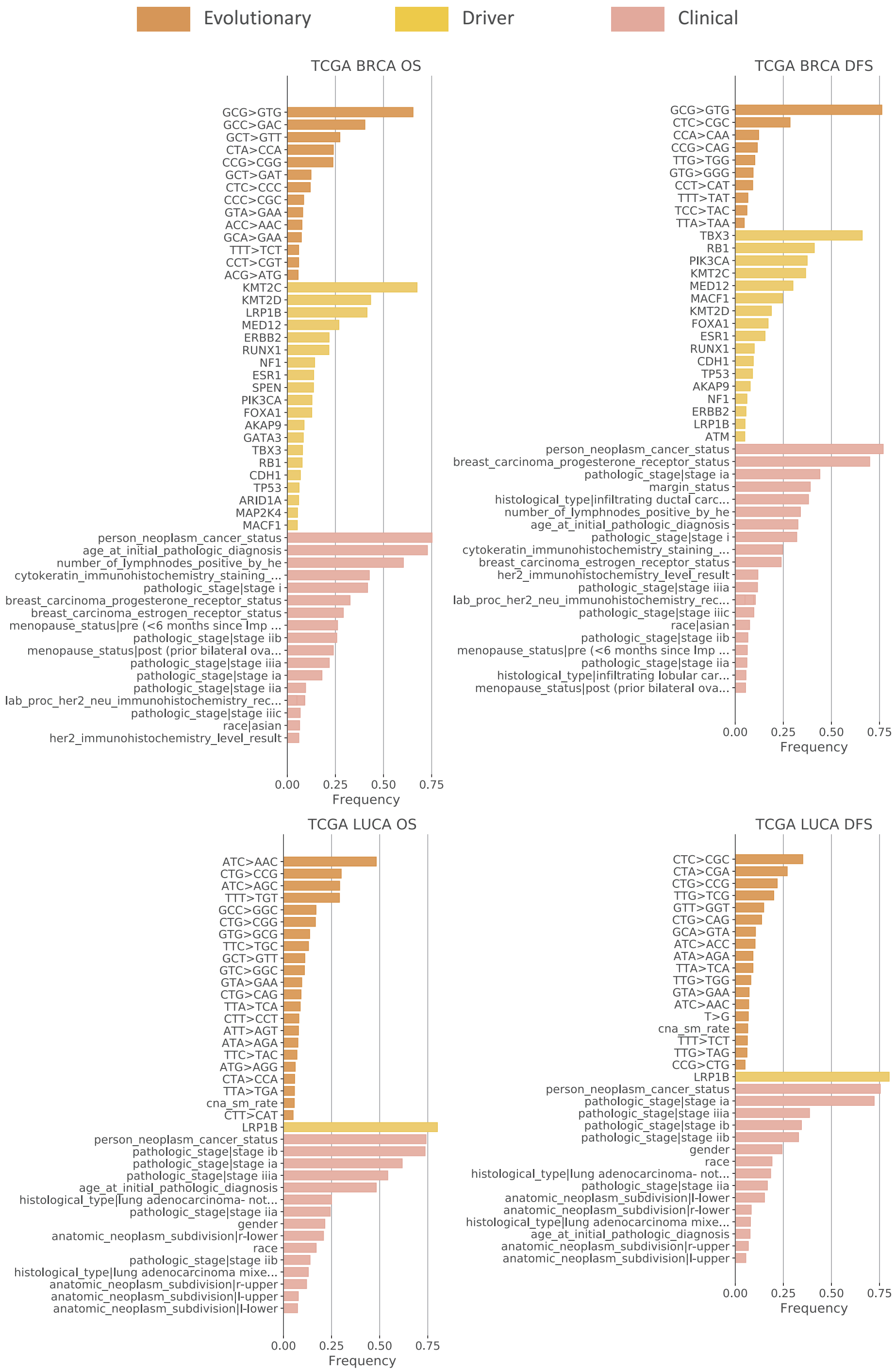

Figure 6.3: List of important features predictive of prognoses (TCGA dataset). The evolutionary, driver, and clinical features were fed separately as the candidate features for subset selection. We bootstrapped through cross-validation for 1,000 trials, each time with 80% features and 80% samples drawn with replacement. The importance of features is reflected as their frequency of being selected. We show those features among the top ten or selected more than 50 times (frequency>5%).

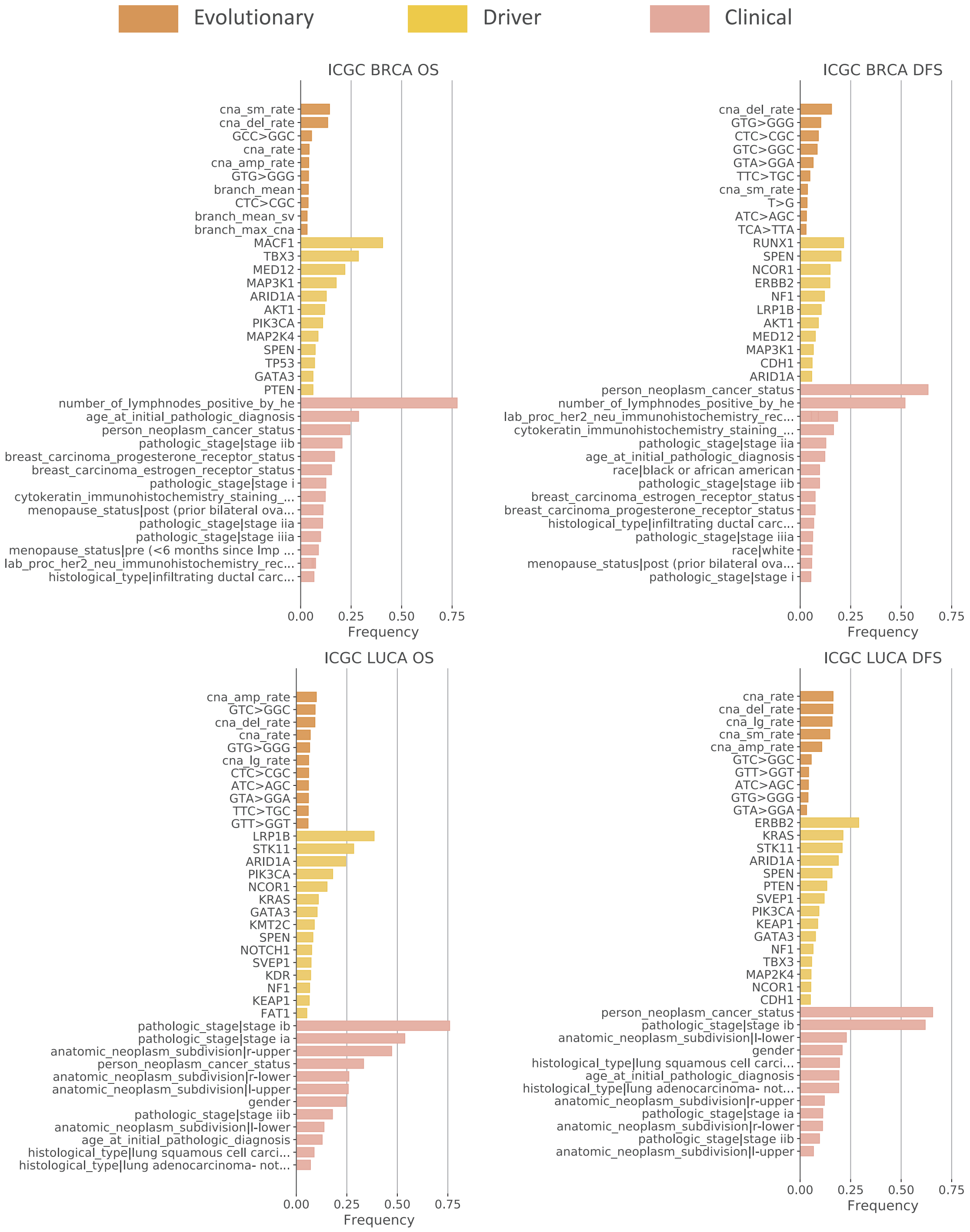

Figure 6.4: List of important features predictive of prognoses (ICGC dataset).

features seem to be less effective predictors for ICGC than TCGA. It might be caused by the different variant callers used with the two datasets. ICGC utilizes the Sanger pipeline to call simple somatic alterations, which identifies most of these alterations as indels instead of SNVs. Therefore the estimation of the SNV-related evolutionary features is less accurate and informative in the ICGC dataset. One this other hand, the WGS-based ICGC dataset allows for better coverage and profiling, especially with respect to structural variations (SVs) and the copy number alterations (CNAs) that result from them. In the ICGC data, we observe several measures related to SV and CNA rates, which are difficult to capture accurately with WES data but visible in WGS experiments (Fig. 6.4 evolutionary bars), such as average branch length in unit of SV rates (*branch mean sv*), CNA duplication/deletion rates (*cna amp rate*, *cna del rate*), and rates of CNA above/below 500,000 nt (*cna lg rate*, *cna sm rate*). Their appearance indicates that the CNA rates and various sub-categories of it are broadly important predictors of progression. It is consistent with prior knowledge that SVs are a crucial mechanism of tumor progression and functional adaptation through their role in creating CNAs as well as fusion genes [264] and contribute substantially to tumor evolution [66]. In addition, phylogenetic evolutionary features such as the average of branch lengths (*branch mean*) are informative in the ICGC BRCA dataset, suggesting that overall measures of evolutionary heterogeneity inferred by TUSV are important predictors of breast cancer progression and prognoses [43, 173]. Frequencies with which predictive features are sampled are generally lower for the ICGC versus TCGA analyses across feature classes, which likely reflects the much smaller cohort available for the WGS analysis.

Driver features A few driver features, such as *TBX3* and *MED12*, are informative for both OS and DFS prediction in breast cancer (Fig. 6.3,6.4 BRCA driver bars) [80, 226]. Some of the most well-established breast driver genes with fairly high population frequencies are not selected with high frequency in all analyses (e.g., *ERBB2* does not appear in the ICGC OS list while *TP53* does not appear in the ICGC DFS list), which likely reflects at least in part the fact that other features can serve as proxies for these and thus they are not consistently chosen by the machine learning feature selection.

The driver feature *LRP1B* is the most informative for both prediction tasks in lung cancer (Fig. 6.3,6.4 LUCA driver bars) [129]. We observe a big difference of selected drivers in the LUCA samples between TCGA and ICGC datasets, though. This is mainly caused by the sparsity of the driver mutation rates in WES-based TCGA data. Most of the drivers are mutated at less than 5% in frequency (zero values in more than 95% samples) and are therefore excluded at the time of feature engineering.

Clinical features Tumor neoplasm status (*person neoplasm cancer status*), *pathologic stage*, *age at initial pathologic diagnosis*, and cancer subtype (*histological type*) emerge as the top clinical features across the eight analyses (Fig. 6.3,6.4 clinical bars). The neoplasm status reports whether the patient has a tumor or is tumor-free. The high frequency of inclusion of pathologic stage and subtype in successful predictors reflects the known value of expert staging and classification in predicting cancer outcome [5], and their inclusion is thus also unsurprising. The importance of age may in part reflect the cancer-intrinsic effects of somatic mutation processes that are likely to have been active longer in tumors of older versus younger patients. However, older patients are more likely to die of other competing risks, such as heart attack, than their cancer during the time of follow-up [217]. Teasing apart the various confounding factors introduced by age at diagnosis is a complicated question, however, beyond the scope of the present study [137].

OS and DFS prediction tasks extensively share clinical features within the same cancer type. For example, PR status (*progesterone receptor status*) and ER status (*estrogen receptor status*) are highly informative for both tasks in BRCA, which is consistent with the current subtype classification method of breast cancer based on hormone receptor status [67]. The clinical features *number of lymphnodes positive by he* and *cytokeratin immunohistochemistry staining method micrometastasis indicator* are consistent with previous research on the value of these factors as predictors of breast cancer survival [256] and metastasis [48]. While in the present work, we directly take cytokeratin immunohistochemistry staining as a feature, one should note the possible artifacts that could be introduced in practice [130]. In lung cancer, both the clinical features *anatomic neoplasm subdivision* and *gender* are predictive for both OS and DFS, consistent with the previous research that the right lung and male gender are usually associated with worse prognoses [196]. The male and female samples are balanced (53% vs. 47%) in the dataset.

Full feature sets The combinations of features that lead to the best performance in TCGA and ICGC when all feature classes are treated collectively during bootstrapping are collected in Fig. 6.5,6.6. In all cases, the high-ranking features in the full feature sets include mixtures of evolutionary, driver, and clinical features. For the most part, the selected features that appear in the full feature bootstrapping are subsets of those found when bootstrapping on individual feature sets. This indicates that the features from the three sets in general provide at least some orthogonal information to one another, and all contribute to the prediction of prognoses. While the top-ranking features in the collective sets are usually drawn from the clinical feature set, the frequencies of many clinical features decrease and some drop out of the list in the full experiments. This observation indicates that the effectiveness of these features diminishes and they can become redundant when some useful genomic features are included. The importance order of features is also frequently changed in the full feature sets, again suggesting correlations between features from distinct classes. Some genomic features are not selected as predictive when combined only with other genomic features, but proved predictive in combination with clinical features, e.g., *ATG*→*ACG* in the TCGA BRCA DFS task (Fig. 6.5).

6.2.2 Landscape of evolutionary, driver, and clinical feature spaces

After identifying and analyzing the important features related to future tumor progression, we explored the landscape of evolutionary, driver, and clinical features in their individual and joint feature space. We were especially interested in understanding the relationship between the three types of features and characterizing any internal structure embedded in the tumor evolutionary feature space. We first explored the overall correlation structure of the full feature space to identify the orthogonal features and provide biological insight into subgroups of features. We calculated the Pearson correlation coefficient between each pair of important features (Fig. 6.3,6.4), and performed hierarchical clustering based on Ward distance [246] to group them. We mainly

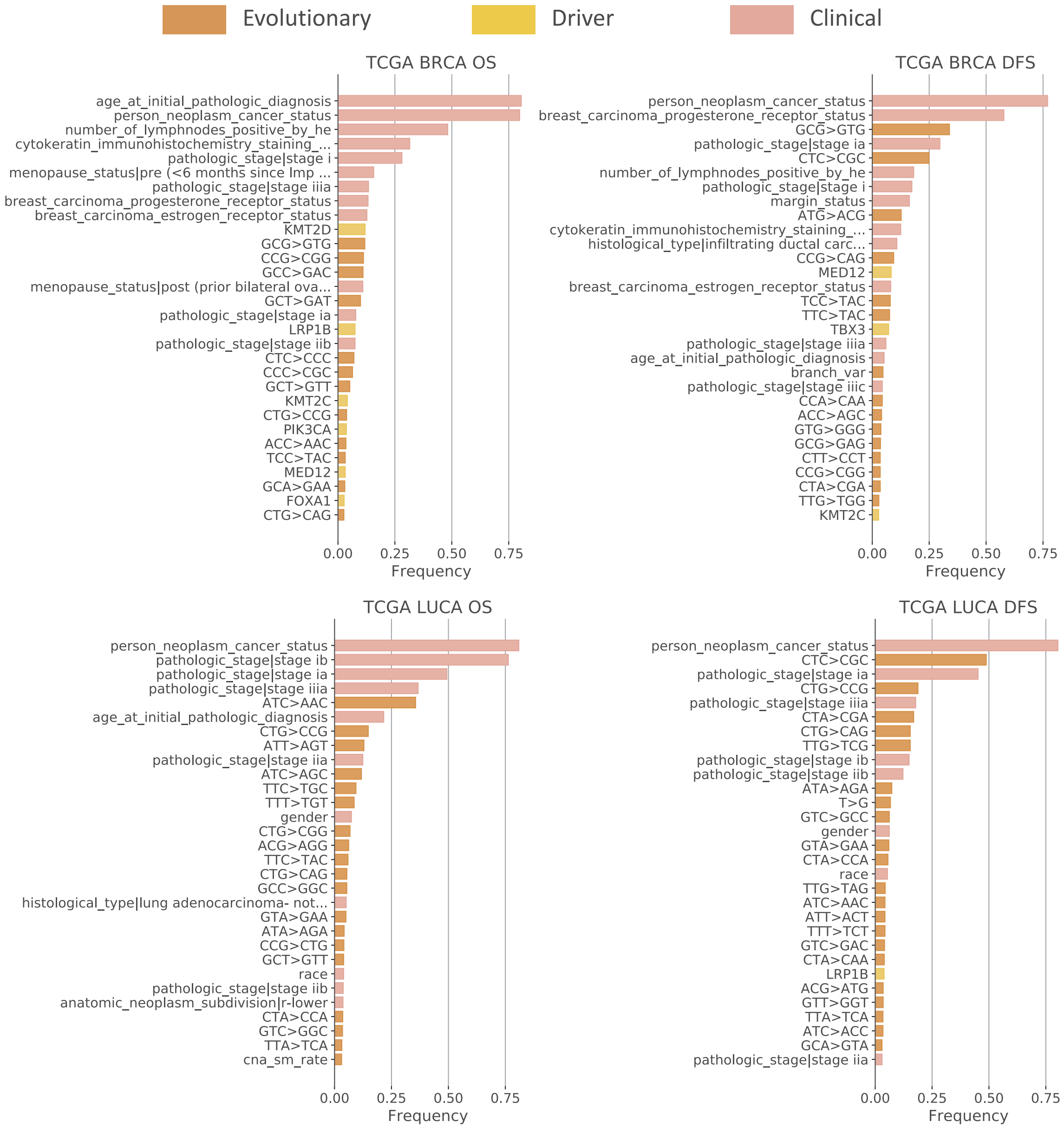

Figure 6.5: List of important features predictive of prognoses when evolutionary, driver, and clinical features are all available (TCGA dataset).

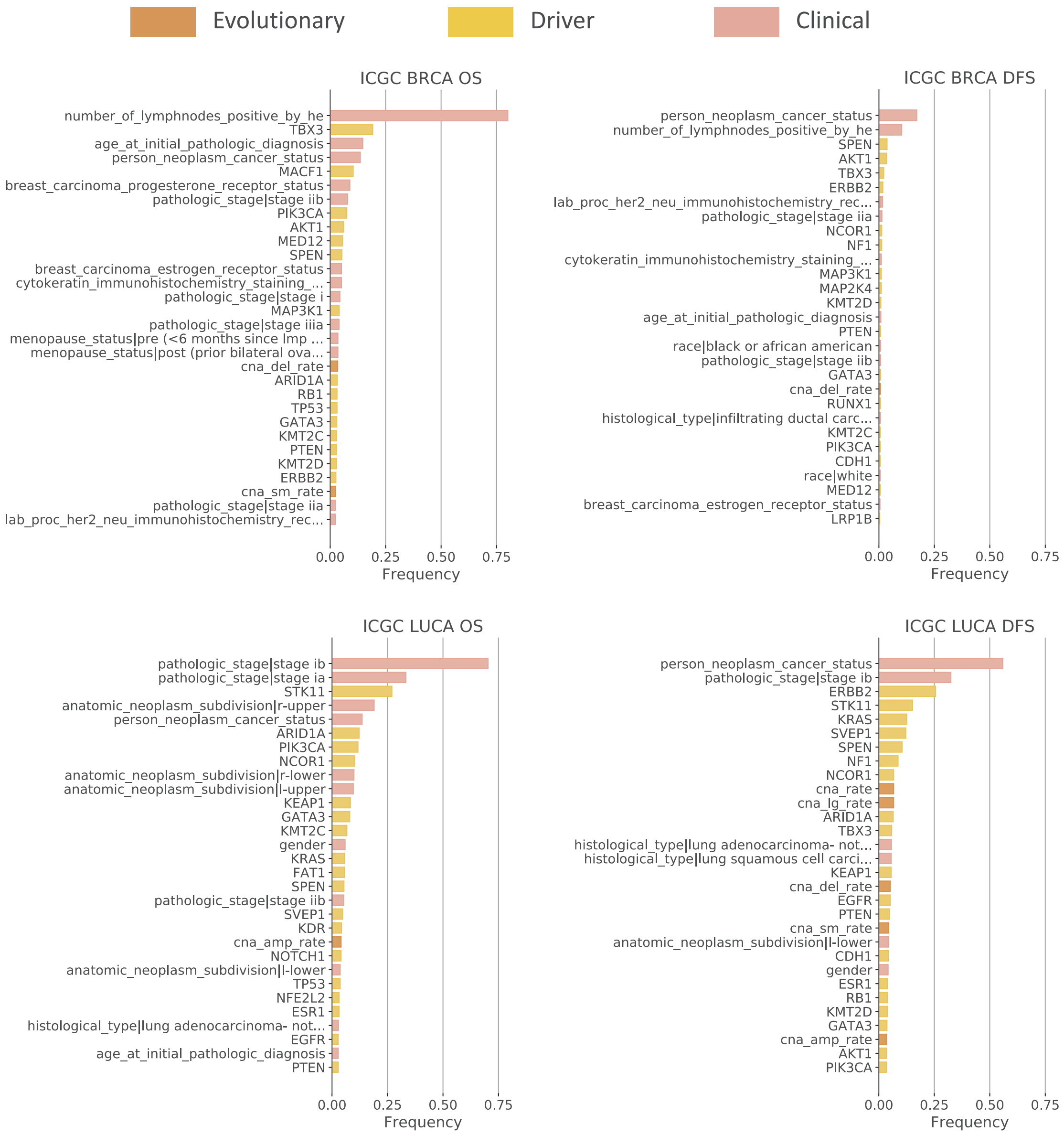

Figure 6.6: List of important features predictive of prognoses when both evolutionary, driver, and clinical features are available (ICGC dataset).

focus here on BRCA samples in both TCGA (WES) and ICGC (WGS) datasets (Fig. 6.7) as they are qualitatively similar to the LUCA datasets (Fig. 6.8).

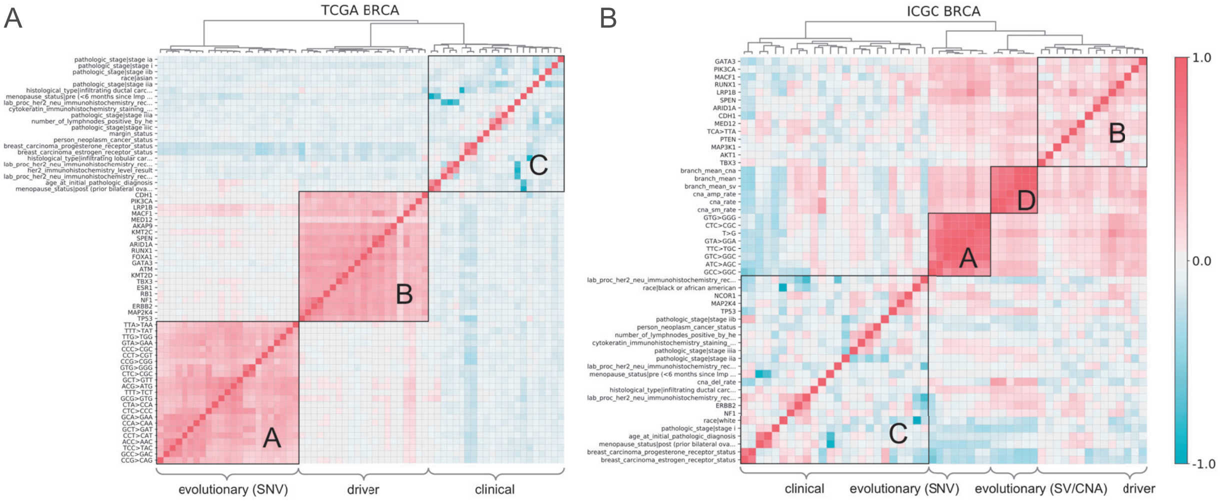

Figure 6.7: Pearson correlation heatmap for evolutionary, driver, and clinical features of BRCA samples. (A) TCGA dataset, (B) ICGC dataset. We merge the important features from Fig. 6.3 (TCGA BRCA OS/DFS) and from Fig. 6.4 (ICGC BRCA OS/DFS). Therefore, these shown features can effectively predict either OS or DFS in BRCA samples. We only show the feature name along each row, while the features of columns are in the same order as rows, due to the display limit. (A) Strong correlation within each feature type is observed: block-A (driver), block-B (SNV-related evolutionary), and block-C (clinical). Meanwhile, genomic features are more independent and orthogonal to clinical features. (B) One can observe an additional block-D (CNA-related evolutionary) in the ICGC data.

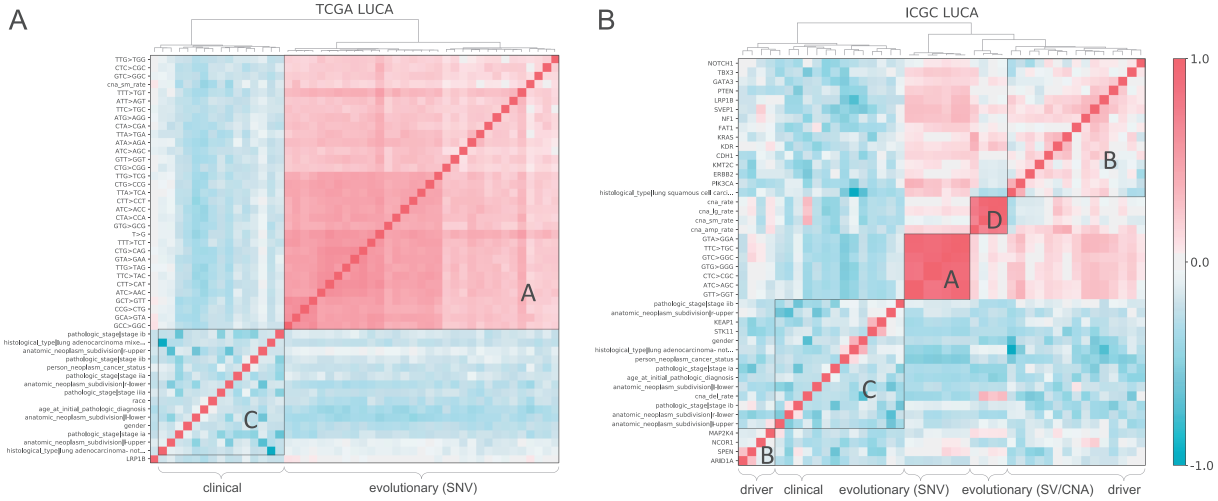

Figure 6.8: Pearson correlation heatmap for evolutionary, driver, and clinical features of LUCA samples. (A) TCGA dataset, (B) ICGC dataset.

The correlation landscape of BRCA features in TCGA (Fig. 6.7A) and ICGC datasets (Fig. 6.7B) allows one to distinguish three or four closely related blocks of features along the diagonal, corresponding to SNV-related evolutionary features (block-A), SV/CNA-related evolutionary features (block-D), driver features (block-B), and clinical features (block-C), which we analyze in turn.

SNV-related evolutionary features (block-A) Most of the SNV-derived evolutionary features are collapsed into a single high-correlation block, essentially corresponding to point mutation rates. This correlation can be easily understood, since all of the features in the block capture overall SNV rate in various ways, although there is some substructure consistent with distinct point mutation processes. These also show some cross-correlation with both driver and SV/CNA features.

SV/CNA-related evolutionary features (block-D) This block primarily collects CNA rate features, which we note are highly correlated with one another and partially correlated to both driver and SNV features. This block emerges only in the ICGC datasets, as we would expect since we require WGS data to profile these features accurately. It shows cross-correlation with both SNV and driver features. The correlation landscape is similar to the sub-blocks of block-A and block-D when only evolutionary features are used for generating the heatmap.

Driver features (block-B) Driver features mostly form a distinct block of high mutual correlation. This block is moderately correlated with both classes of evolutionary rate features. We hypothesize that the general pattern of positive correlation among driver features reflects generic differences in background SNV or CNA rates from different somatic hypermutability phenotypes [135]. For example, we propose that tumors exhibiting one CNA driver mutation are more likely to exhibit other CNA driver mutations because they are more likely to have an elevated rate of CNA mutations generically. This interpretation is confirmed by the positive cross-correlations between driver features and CNA rate features (Fig. 6.7B) and between driver and SNV rate features (Fig. 6.7A,B).

Clinical features (block-C) Most clinical features are negatively, rather than positively, correlated with other clinical features outside of small, highly correlated sub-blocks. The main reason comes from the data processing and feature extraction step when we map categorical clinical features into one-hot vectors and therefore introduce collinearity. For example, the *menopause status* | *pre* and *menopause status* | *post* originally came from the same categorical clinical feature *menopause status* to yield two anti-correlated binary features. We can break this type of collinearity while maintaining the model interpretability by dropping one of the anti-correlated categories and bootstrapping. Strong positive correlation is also sometimes observed within clinical features for biological reasons, such as that between PR status (*progesterone receptor status*) and ER status (*estrogen receptor status*) reflecting a common association with luminal breast cancer subtypes [67], or that between *age* and *menopause status*.

In general, we may conclude that: 1) The features from each of the four main feature blocks are highly correlated to other features of the same block, validating that it is necessary to prevent collinearity by employing feature pruning and bootstrapping. 2) The genomic features are, in general, slightly negatively correlated to the clinical ones. This relationship is generally stronger in the ICGC data (Fig. 6.7 A vs. B; Fig. 6.8 A vs. B). This indicates that the genomic features capture some of the information within the clinical features. 3) Each of the major feature blocks carries independent information despite the correlations between blocks. This independence is especially true for genomic features vs. clinical features. Therefore, one may conclude that the generical genomic features are roughly orthogonal to the clinical features and can be used as complementary in the task of prognostic prediction to improve the performance possibly. 4) SNV and SV/CNA features carry largely orthogonal information to one another despite some cross-correlation. This analysis is concordant with findings of Dawson et al. [51] that suggested a partitioning of breast cancers into distinct classes of SNV-driven and CNA-driven tumors (Fig. 6.7B), although our findings support a subtly different model that these are two orthogonal feature classes both of which may act to different degrees in the same tumors as opposed to defining two orthogonal classes of tumors.

6.2.3 Evolutionary features contribute substantially to progression risk

In this section, we focus on the central question of this chapter: to what degree is the future progression risk of the tumor determined by its mutational phenotypes, as captured by evolutionary features. To ask this question, we need to account for the fact that correlations among feature classes will cause non-evolutionary features to serve as at least partial proxies for the evolutionary features we wish to assess. We thus need to assess the contribution of evolutionary features in the context of other genomic features (driver) and other environment-affected features (clinical features). We evaluate the contributions of evolutionary features to the prognostic prediction using a fraction metric inferred from the hazard ratio (HR) between predicted benign and malignant cohorts of tumors on the test set (Fig. 6.1C). We focus on the log-HR-based fraction metric in this chapter, since it is linear in the scale of feature space. Although it is possible to conduct the similar analysis based on the results from concordance index (CI; Tab. 6.6), the nonlinearity of the CI makes the results hard to compare and interpret.

We conducted two-loop cross-validation experiments using evolutionary features, genomic features, or full features, replicating each experiment five times. The HRs were then calculated on the test sets that used the three feature sets following Eq. 6.8. We finally estimated the log-HR-based fractions of either evolutionary or genomic features from the log-HRs following Eq. 6.7,6.9.

Fig. 6.9 shows that the evolutionary features contribute around 40% of the risk captured by the predictive models for both the BRCA and LUCA in the TCGA dataset and around 25%-35% of the risk for BRCA and LUCA in the ICGC dataset. We believe the difference largely occurs, despite the better genome coverage in the WGS-based ICGC data, because the Sanger variant calling pipeline identifies most of the simple somatic mutations as indels instead of SNVs, preventing us from effectively extracting the trinucleotide features that proved to be among the most powerful evolutionary features for tumor progression prediction (Fig. 6.3,6.4). While the exact numbers are thus contingent on the data available and how it is processed, the data suggest that mutational phenotypes, corresponding to variability in mechanisms of evolutionary drift, account for approximately a third of the total progression risk identified.

Genomic features in total contribute around 50% of the risk in the TCGA dataset, and around

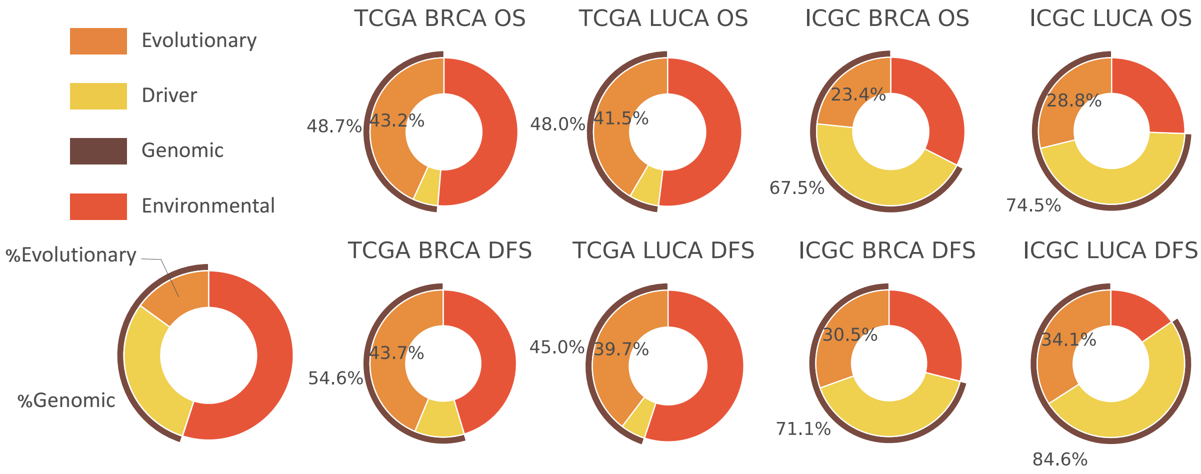

Figure 6.9: Contribution percentage of evolutionary and genomic features to tumor progression risk prediction. Evolutionary features contribute to around 40% in TCGA data, and 25-35% in ICGC data. The genomic features contribute to around half in TCGA, and 70-80% in ICGC dataset.

70%-80% in the ICGC data. The discrepancy is mainly because driver features proved substantially more predictive for ICGC than TCGA. It appears to occur because of the sparsity of the driver features in the TCGA WES dataset. This is especially true for LUCA, where in most cases only a single driver *LRP1B* has more than 5% non-zero values (Fig. 6.3 TCGA LUCA driver bars). In addition, the driver features in TCGA are more highly intra-correlated than those in ICGC (Fig. 6.7 A vs. B). ICGC WGS has a higher coverage than TCGA WES, and since all the somatic alterations that happened within the genome region of drivers, including SNVs, indels, CNAs, and SVs are counted as driver mutation, ICGC has a substantially richer representation of the driver features. While the exact numbers are again dependent on the specific data, the results suggest that very roughly half of the contribution of genomic features to progression risk comes from variability in mutational phenotypes and half from other independent genomic factors.

We note here that we do not provide the contribution fractions of drivers by themselves. The summation of evolutionary fractions and driver fractions does not necessarily equal the genomic fractions, since we have to take into account the correlation and interaction between the two feature sets. That is, some portion of the driver risk is explained by driver mutations that induce hypermutability phenotypes, and our approach is intended to separate that information from driver risk that is not caused by mutational phenotypes.

6.2.4 Evolutionary and genomic features improve prognostic prediction

While improving predictive power for progression is not the central goal of this study, the guiding hypothesis nonetheless suggests that evolutionary features should provide predictive power for future progression, an inference that we test in this section. Although it is to be expected that clinical features will have the strongest individual predictive value among feature classes, we are interested here in establishing whether evolutionary features provide any additional predictive value for progression outcomes beyond that provided by clinical features alone or clinical features supplemented by driver genomic features. We use the two groups of evaluation datasets (TCGA WES data with more samples; ICGC WGS data but with smaller cohort size) to explore different settings and possibilities in clinical practice (Tab. 6.1), such as different cancer types, sequencing methods, and prediction tasks, to answer that question. The results indicate that evolutionary features can incrementally enhance the predictive power in most cases (Tab. 6.6) and that genomic features (evolutionary + driver) collectively enhance predictive power relative to clinical alone.

Table 6.6: Performance of prognostic prediction with different features in WES-based TCGA and WGS-based ICGC samples. Clinical feature set performs best among the evolutionary, driver, and clinical feature sets. However, the additional genomic features are synergistic and promote the prediction of prognoses ("full" features), except the ICGC LUCA OS task, where two feature sets are on par with each other. We evaluate the performance using two-loop cross-validation (CV). Both TCGA and ICGC utilize 3-fold outer CV to calculate the concordance index (CI) for evaluation. We repeat the outer CV for five times to calculate the means and standard deviations to measure the variation on each test. As for inner CV, we use 3-fold CV for TCGA, and leave-one-out CV (LOOCV) for ICGC since it has a smaller sample size. "Genomic" means all genomic features, including driver and evolutionary features. "Full" means both evolutionary, driver, and clinical features. Statistical significance notation for the "full" vs. "clinical" is defined by the one-sided test p-value. $^{\text{ns}}$: not significant; $^{\cdot}$: $p < 0.10$; *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$. We refer interested readers to Fig. 6.5 and Fig. 6.6 for features selected in the "full" experiments (last row of the table).

	TCGA (WES)				ICGC (WGS)			
	BRCA		LUCA		BRCA		LUCA	
CI	OS	DFS	OS	DFS	OS	DFS	OS	DFS
evolutionary	$56.9_{\pm 0.56}$	$53.3_{\pm 0.44}$	$51.8_{\pm 0.26}$	$50.5_{\pm 0.25}$	$51.7_{\pm 0.74}$	$54.0_{\pm 1.21}$	$52.9_{\pm 0.73}$	$53.2_{\pm 0.66}$
driver	$54.9_{\pm 0.54}$	$55.4_{\pm 0.59}$	$53.6_{\pm 0.05}$	$53.6_{\pm 0.03}$	$53.1_{\pm 0.97}$	$51.1_{\pm 0.46}$	$51.2_{\pm 0.59}$	$50.4_{\pm 0.59}$
genomic	$59.2_{\pm 0.49}$	$56.2_{\pm 1.01}$	$53.2_{\pm 0.35}$	$51.7_{\pm 0.38}$	$57.5_{\pm 2.15}$	$56.2_{\pm 2.30}$	$52.8_{\pm 1.28}$	$54.5_{\pm 1.03}$
clinical	$78.9_{\pm 0.29}$	$74.3_{\pm 0.51}$	$67.4_{\pm 0.31}$	$63.1_{\pm 0.35}$	$75.0_{\pm 1.47}$	$70.3_{\pm 2.47}$	$\mathbf{62.1}_{\pm 1.14^{\text{ns}}}$	$58.1_{\pm 1.54}$
full	$\mathbf{79.7}_{\pm 0.32^{**}}$	$\mathbf{75.1}_{\pm 0.69^{*}}$	$\mathbf{67.9}_{\pm 0.47^{\cdot}}$	$\mathbf{64.4}_{\pm 0.35^{***}}$	$\mathbf{80.7}_{\pm 1.34^{***}}$	$\mathbf{80.8}_{\pm 3.22^{**}}$	$61.7_{\pm 1.09}$	$\mathbf{61.5}_{\pm 2.65^{*}}$

We utilized two-loop cross-validation (CV) to tune, train and unbiasedly evaluate the performance of ℓ_0-regularized Cox regression on different sets of features with the performance metric of concordance index (CI). We used 3-fold CV in both TCGA and ICGC for outer loop CV; 3-fold CV in TCGA, and leave-one-out CV (LOOCV) in ICGC for inner loop CV, since ICGC has far fewer samples. In order to quantify the uncertainty resulting from random choices in splitting the datasets and assess the robustness of the methods, we randomly shuffled the samples and repeated the experiments for five times to derive the means and standard deviations of prediction outcomes.

As expected, the clinical predictors provide the strongest predictive information among any single predictive class (evolutionary, driver, and clinical), as assessed by CI for both WES-based TCGA and WGS-based ICGC data (Tab. 6.6). Evolutionary and driver features alone are predictive as well but substantially less than clinical features individually. Evolutionary and driver features perform comparably to one another, with each performing slightly better than the other in half of the scenarios. In most cases, genomic features collectively performed better than either evolutionary or driver alone, suggesting that each carries some orthogonal information. In all but one case, the full feature set outperformed clinical features alone. Among all the eight cases, six show statistically significant improvements (p-value$<$0.05), and two show effective ties in performance between clinical and full features (TCGA LUCA OS, ICGC LUCA OS) suggesting that the additional genomic features are already largely explained by the information captured by the clinical features. Collectively, the results suggest that each of the three feature classes carries at least some predictive power not captured by the other classes.

The selected features from the full feature sets show that the clinical features are always the most essential in the model (Fig. 6.5,6.6). However, the SNV-related evolutionary features mainly facilitate the model for the TCGA dataset while driver features and CNA-related features mainly facilitate the model in ICGC dataset. Similar to the above section, we hypothesize that this reflects the advantages of WGS in capturing the somatic alterations in the driver regions. However, the Sanger pipeline takes most of the simple somatic mutations as indels instead of SNVs, leading to a loss of information when we infer the SNV-related evolutionary features. This variability between the different variant calling pipelines used by the two sequencing projects suggests the importance of feature engineering for the prognostic prediction task.

Finally, we evaluated the ability of the best combinations of full features in both TCGA and ICGC to stratify patients with distinct survival/non-recurrence time as assessed by the logrank test (Fig. 6.10). Patients were split into two groups: malignant and benign, based on the predicted hazards of the events (decease or recurrence) using ℓ_0-regularized Cox model at the outer CV phase. Samples with predicted hazards larger than the median of predictions were classified as malignant, otherwise as benign. Figure 6.10A illustrates the separations by OS and DFS time between the predicted malignant and predicted benign tumors in the WES-based TCGA through distinct Kaplan-Meier estimator curves. The separations are statistically significant for both breast and lung cancers. Similar significant separations exist for tumors in the WGS-based ICGC dataset (Fig. 6.10B).

We were also curious whether other Cox regression variants might perform better than ℓ_0-regularized Cox model on our datasets, e.g., lasso. We therefore performed the experiments following the same two-loop CV protocol using lasso for feature selection as well. We tuned the ℓ_1 coefficient of lasso ($\lambda \in [0.03, 30.0]$) in the inner CV. One can see that our ℓ_0-regularized Cox regression model performs better than the lasso in most cases (Tab. 6.6 vs. Tab. 6.7). In addition, we noticed that it is difficult for the lasso model to fully utilize the additional genomic features when integrating them with the clinical features (Tab. 6.7 row “full” vs. “clinical”). From our observations on the lasso regression results, we find they are more sparse than the ℓ_0-regularized method, indicating that lasso might be too strict in its feature selection to make full use of the small gains from individual genomic features. When both genomic and clinical features are provided, lasso tends only to keep the clinical features.

In summary, these experiments indicate that genomic and clinical features act synergistically to improve the prediction of prognoses, and that in general, adding evolutionary features that

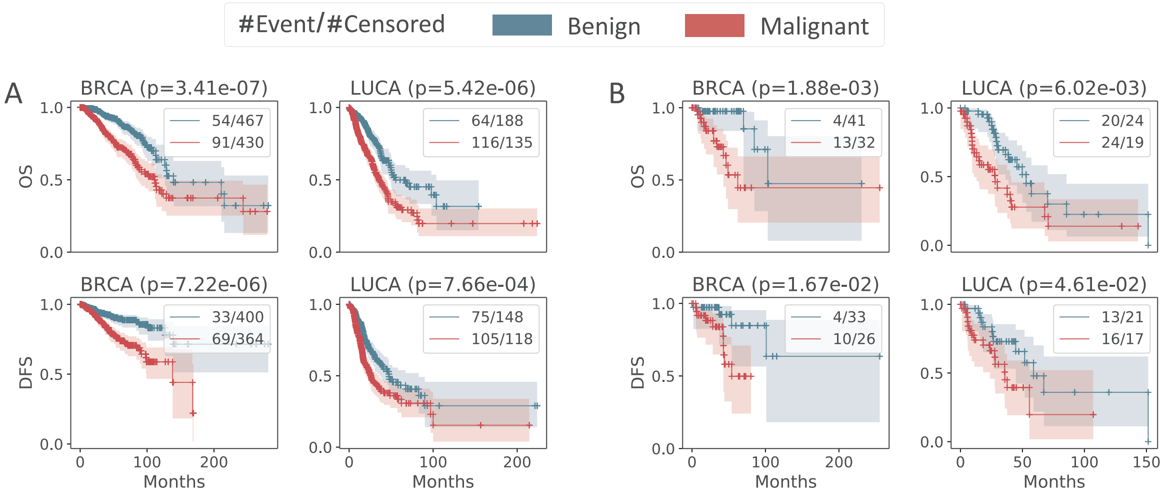

Figure 6.10: Kaplan-Meier estimators of predicted malignant and benign cohorts using both clinical and genomic features. (A) TCGA, (B) ICGC. Logrank test shows significant distinct survival and recurrence profiles between the two cohorts across both datasets and cancer types.

Table 6.7: Performance of prognoses prediction using lasso (ℓ_1-regularized Cox model) instead of ℓ_0-regularized Cox regression model. The lasso model follows the same two-loop cross-validation evaluation protocol as ℓ_0-regularized model, and replicates for five times to get the mean and standard deviation values of performance in concordance index (CI). One can find that ℓ_0-regularized Cox model performs better than lasso in most cases (29/40), while lasso performs better only in 6 cases.

	TCGA (WES)				ICGC (WGS)			
	BRCA		LUCA		BRCA		LUCA	
CI	OS	DFS	OS	DFS	OS	DFS	OS	DFS
evolutionary	$58.0_{\pm 0.12}$	$52.8_{\pm 0.39}$	$50.8_{\pm 0.26}$	$50.4_{\pm 0.15}$	$51.0_{\pm 0.58}$	$53.9_{\pm 1.03}$	$52.8_{\pm 0.59}$	$50.4_{\pm 0.50}$
driver	$55.3_{\pm 0.40}$	$55.1_{\pm 0.57}$	$53.6_{\pm 0.06}$	$53.6_{\pm 0.03}$	$52.7_{\pm 1.44}$	$51.1_{\pm 0.74}$	$50.0_{\pm 0.08}$	$50.1_{\pm 0.21}$
genomic	$58.0_{\pm 0.19}$	$54.4_{\pm 0.30}$	$51.6_{\pm 0.41}$	$50.6_{\pm 0.19}$	$51.1_{\pm 0.71}$	$54.2_{\pm 0.90}$	$52.2_{\pm 0.45}$	$50.5_{\pm 0.43}$
clinical	$78.2_{\pm 0.47}$	$74.8_{\pm 0.52}$	$67.6_{\pm 0.72}$	$63.1_{\pm 0.55}$	$70.9_{\pm 3.91}$	$73.7_{\pm 4.02}$	$62.1_{\pm 1.55}$	$55.6_{\pm 2.07}$
full	$79.6_{\pm 0.19}$	$72.6_{\pm 0.34}$	$67.5_{\pm 0.47}$	$64.5_{\pm 0.42}$	$77.9_{\pm 1.76}$	$69.5_{\pm 3.50}$	$57.3_{\pm 1.15}$	$58.4_{\pm 1.79}$

capture variation in mutational phenotypes of tumors enhances predictive power relative to clinical features and driver features. Note that this analysis asks a different question than the hazard ratio analysis in the previous section. In the preceding section, our goal was to estimate the portion of risk entailed by mutational phenotypes specifically, which is captured by evolutionary features but may also be redundantly captured by the other feature classes. The analysis in this section explores the non-redundant information captured by each feature class beyond what is carried by the others. Collectively, these two results suggest that a large fraction of progression risk is captured by evolutionary features, and while much of that information about mutational phenotypes is also captured by driver and/or clinical features, there is additional information in the evolutionary features that manifests as somewhat enhanced predictive power.

6.3 Discussion

Our work supports the central hypothesis that the variation in mutational phenotypes accounts for a large fraction of cancer progression risk, under various confounding environmental factors, beyond the risk for which independent of clinical factors and specific driver gene mutations account. The recognition that cancer is a product of somatic evolution has proven greatly influential in our understanding of how cancers appear and progress. However, practical application of evolutionary theory in cancer has predominantly focused on the "selection" side of evolution, e.g., identifying functional mutations under selection as biomarkers or drug targets, and less so on variations cancer-to-cancer in mechanisms of "diversification", or the drift component of evolution. The exact quantification of various sources of risks presented here must necessarily be considered preliminarily, as it depends on the cohorts examined, the data sources available, the kinds of mutability mechanisms and other features profiled, and the details of the machine learning analysis. Nonetheless, the results suggest that the processes by which a particular tumor generates non-specific genetic diversity (the drift component of tumor evolution) are of comparable importance to the specific functional mutations that diversity has produced (the selection component of tumor evolution) in determining whether or not the tumor progresses. This observation has implications for how we pursue diagnostics for cancers or precancerous conditions, prognostic prediction, and treatment strategies.

Nonetheless, large gaps remain in exploring in detail the space of predictors, the mechanisms by which they act, and the best strategies to realize their translational potential. Some of these questions remain data-limited and likely cannot be answered without larger cohorts and richer clinical metadata. Our work suggests that WGS provides an advantage over WES in improving prediction power beyond that of clinical and driver-centric features, particularly in allowing us to better capture the important role of SV/CNA-driven mutability in parallel to SNV-driven mutability. Available cohorts of patients with WGS data are still limited, an issue that proved a limitation to this chapter despite our use of some of the largest cancer WGS corpora that have been made accessible to researchers. Insufficient sample sizes could lead to fragile models and statistically insignificant performance.

Another area where the work can likely be advanced is with respect to describing patient-specific mutational phenotypes effectively through a set of partially independent variables. Our results suggest that SV/CNA and SNV mutator phenotypes act largely independently of one another (block-A and block-D in Fig. 6.7,6.8). Prior literature suggests a finer structure underlying these two broad types of processes exists. In the SNV domain, mutational signature analysis has demonstrated the existence of approximately 30 independent point mutation processes acting to different degrees on different cancers [2]. We likewise know that distinct mechanisms of SVs and CNAs act on a tumor genome [248] and at least some of these, such as whole genome duplication (WGD), are independently predictive of outcome [170, 185, 260]. Our analysis of the evolutionary feature space suggests that there is a structure of the manifolds describing mutability phenotypes related to progression outcomes, but more work is needed to develop more interpretable models of this manifold structure and translate that into both better predictions and mechanistic insights.

We observe systematic variances of evolutionary and driver features in the same cancer type across the two datasets (Fig. 6.3,6.4). Such variances mainly came from different sequencing coverages, variant callers, and phylogenetic models at the time of data extraction and collection. Our conclusions on the selected informative features are thus mainly restricted to the data extracted with the same platforms. We might further generalize the informative features, reduce variations of models, and improve the overall performance through ensemble learning [59], by integrating the results from different sequencing techniques, variant calling pipelines, and clonal inference algorithms. Second, through unbiased evaluations of the two most common cancer types under various settings, we found qualitatively similar results on the contributions of evolutionary features. However, pan-cancer analysis across different cancer types given sufficiently large cohorts can be a future direction to consolidate and extend this observation, and enable us to evaluate the different tumor evolution mechanisms acting in each cancer type. Third, more advanced methods for survival analysis might be warranted. For example, deep learning has been applied to survival analysis recently [106, 187, 259], demonstrating the feasibility of using a neural network as an effective feature extractor for predicting prognoses if the evolutionary, driver and clinical features of all the available samples are integrated properly. In summary, we believe that future work with better feature extraction protocols, improved phylogeny reconstruction algorithms, more extensive and diverse datasets, and more powerful machine learning models will provide us with greater understanding and a more accurate estimation of the risk contribution from the mutational phenotypes.

Chapter 7

Conclusions and future directions

The reduced cost of sequencing technologies enables the research community to share large-scale genomic data of tumors, which makes it possible to utilize powerful machine learning (ML) methods to address crucial problems of oncology. However, due to the high dimensions and complex interactions of the noisy genomic data, normal ML models can not be easily transferred from successful applications of other domains. In the thesis work, we showed that the integration of prior knowledge such as cancer pathways and gene representations, and techniques such as transfer learning and multi-task learning, can alleviate the problems of limited sample size. Proper regularization and evaluation methods are crucial in avoiding overfitting issues. The ML models in cancer genomics are closely related to clinical decisions, such the interpretability and robustness will be a much more important concern in this area compared to other applications. Due to the mechanism of tumor progression, the bulk genomic data usually consist of heterogeneous cell populations. This kind of heterogeneity may lead to severe outcomes such as drug resistance. Although single-cell sequencing could be a promising approach to resolve this issue, we showed that proper deconvolution algorithms can facilitate us to have a better understanding of the cell clones.

Biologically, we found although different cancer types are caused by various factors, they share some common signaling pathways and evolutionary mechanisms. For example, we discovered a set of mutations common in different cancer types, genes of intracellular vesicles correlated with drug resistance in different cell lines, reduced *PI3K-Akt*, *ECM-receptor interaction*, and *focal adhesion pathways* in different breast cancer metastasis cases. In addition, we found the mechanism of tumor progression is indicative of future prognosis, e.g., mutational signature and structural variations indicative of prognoses in both breast and lung cancer, therefore it may be helpful in translational research to validate these evolutionary features.

In Chapter 2 and Chapter 3, we addressed one of the most important tasks in cancer genomics: phenotype inference using the genomic alteration profile of the cancer patient. We started with the phenotype of transcriptome expression levels in Chapter 2, because of the large scale of mRNA we have and its crucial connections with both the upstream somatic alterations and downstream phenotypes. We developed an interpretable deep learning model called GIT, which utilized the self-attention mechanism to capture the contextual interactions of mutations and accurately predict the differentially expressed genes from somatic mutations. As the side products of GIT, gene embeddings and tumor embeddings prove to reflect the biological similarities between mutations and tumor patients. In Chapter 3, we focused on a more challenging problem of predicting drug resistance of cancer cell lines with RNA-seq data. We further improved the self-attention mechanism developed in the GIT model to be a contextual-attention mechanism to capture the complex interaction between both genes and drugs. To address the problem of noisy and missing data, we employed the framework of collaborative filtering, instead of the widely used encoder-decoder architecture. The proposed CADRE model outperformed both the classical model as well as the deep learning model in the drug response prediction task.

In Chapter 4 and Chapter 5, we focused on another crucial problem in cancer genomics: intra- and inter-tumor heterogeneity of the tumor and its progression mechanism. In Chapter 4, we proposed a mathematical formulation of the deconvolution problem and transformed the problem equivalently into a neural network problem. The method additionally incorporated the pathway and gene module information to make it the state-of-the-art method in the "complete deconvolution" problem. The NND method proposed in Chapter 4, however, suffered from the slow training speed and did not scale well to large dimensions of data. Therefore, in Chapter 5, we further improved it into the RAD model. The RAD model essentially shared similar mathematical formulations with NND, but employed a three-phase optimization process to accelerate the training of the model (because of the first warmstart phase) and achieved even better accuracy (because of the second coordinate descent phase). It is also more reliable and robust (because of the third minimum similarity phase). We developed a Python package based on RAD model and provided an easy-to-use interface for researchers.

In Chapter 6, we answered the question of whether the heterogeneity and evolution can help the prognosis of cancer, in addition to the classical pathological and driver information; and what is the contribution of these evolutionary features to the future progression risk of tumors. We developed an ℓ_0-regularized Cox regression model, which utilized features generated from the phylogenetic models such as TUSV or Canopy to predict both the survival profiles and recurrence patterns of breast cancer and lung cancer patients. Our results showed that the inclusion of evolutionary features can improve the prognostic prediction in most cases. The evolutionary features contribute to a significant portion of around 1/3 of the progression risk.

A lot of future work can be explored based on the thesis work (Fig. 7.1). For the task of predicting phenotype using genomic data, we may explore fine-grained models that consider not only the entity of mutated genes but also the mutation location and mutation type as well. For the task of tumor deconvolution, we are currently mainly focusing on bulk RNA data, however, the integration of single-cell data will likely provide us a more clear landscape of the cell types and progression trajectory [36]. For the effect of heterogeneity on the phenotype, we are currently mainly working on phylogenetic model inferred features. It will be a promising question of connecting the large scale *in vitro* cancer cell line data to the *in vivo* tumors. In the long run, the application of machine learning in cancer genomics will likely to benefit from the publicly large-scale single-cell sequencing data. The spatial scRNA-Seq and corresponding algorithms may emerge as powerful tools for elucidating the heterogeneity of tumor cells and the complex signaling pathways in cancer [47]. However, due to the expensive cost and strict time restrictions for samples, the utilization of bulk data may still be a major approach in the foreseen future.

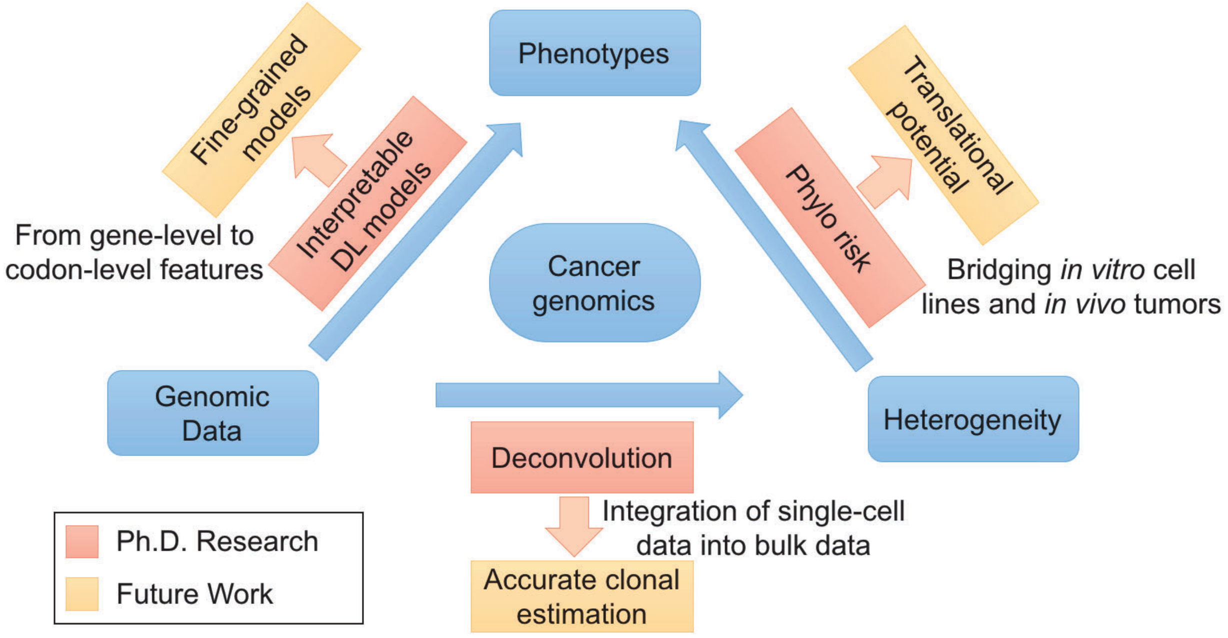

Figure 7.1: Future work.

Bibliography

[1] Abbas, A. R. et al. Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. *PloS one*, 4(7):e6098, jul 2009. 4.2.1

[2] Alexandrov, L. B. and Stratton, M. R. Mutational signatures: the patterns of somatic mutations hidden in cancer genomes. *Current Opinion in Genetics & Development*, 24: 52–60, 2014. 6, 6.1.3, 6.2.1, 6.3

[3] Alexandrov, L. et al. The repertoire of mutational signatures in human cancer. *BioRxiv*, page 322859, 2018. 6

[4] Amaratunga, D. and Cabrera, J. Analysis of data from viral DNA microchips. *Journal of the American Statistical Association*, 96(456):1161–1170, 2001. 4.1.2

[5] Amin, M. B. et al. The eighth edition AJCC cancer staging manual: Continuing to build a bridge from a population-based to a more "personalized" approach to cancer staging. *CA: A Cancer Journal for Clinicians*, 67(2):93–99, 2017. 6.2.1

[6] Andersen, M. S., Dahl, J., and Vandenberghe, L. CVXOPT: A python package for convex optimization, version 1.1.6. *Available at cvxopt.org*, 54, 2013. 5.1.2

[7] Arteaga, C. L. ErbB-targeted therapeutic approaches in human cancer. *Exp. Cell Res.*, 284(1):122–130, mar 2003. 1.1

[8] Ashburner, M. et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25:25, may 2000. 2, 2.2.2

[9] Aster, J. C., Pear, W. S., and Blacklow, S. C. The varied roles of Notch in cancer. *Annual Review of Pathology*, 12:245–275, jan 2017. 4.2.2

[10] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*, 2015. 2

[11] Bailey, M. H. et al. Comprehensive characterization of cancer driver genes and mutations. *Cell*, 173(2):371—-385.e18, apr 2018. 1.1

[12] Barretina, J. et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483:603, mar 2012. 1.1, 1.1, 2.2.5, 3, 3.1.6, 3.1.6

[13] Barrett, T. et al. NCBI GEO: archive for functional genomics data sets–update. *Nucleic Acids Research*, 41(Database issue):D991–D995, jan 2013. 3.1.4, 4.2.1

[14] Basudan, A. et al. Frequent ESR1 and CDK Pathway Copy-Number Alterations in Metastatic Breast Cancer. *Molecular cancer research : MCR*, 17(2):457–468, feb 2019. 5

[15] Beerenwinkel, N. et al. Mtreemix: a software package for learning and using mixture models of mutagenetic trees. *Bioinformatics*, 21(9):2106–2107, 2005. 1.2.4

[16] Beerenwinkel, N., Greenman, C. D., and Lagergren, J. Computational cancer biology: an evolutionary perspective. *PLoS computational biology*, 12(2), 2016. 5.1.1

[17] Bell, R. M. and Koren, Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 43–52, 2007. 4.1.4

[18] Bengtsson, M., Ståhlberg, A., Rorsman, P., and Kubista, M. Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. *Genome research*, 15(10):1388–1392, oct 2005. 5.1.5

[19] Beretta, L. and Santaniello, A. Nearest neighbor imputation algorithms: a critical evaluation. *BMC Medical Informatics and Decision Making*, 16 Suppl 3(Suppl 3):74, jul 2016. 1.2.5, 3.1.6

[20] Bertsimas, D. and King, A. Logistic regression: From art to science. *Statistical Science*, 32:367–384, aug 2017. 6.1.6, 6.1.6

[21] Blagus, R. and Lusa, L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14:106, mar 2013. 1.2.5

[22] Brastianos, P. K. et al. Genomic characterization of brain metastases reveals branched evolution and potential therapeutic targets. *Cancer discovery*, 5(11):1164–1177, nov 2015. 4, 4.2.2, 4.2.4, 4.3, 5.2.4, 5.2.5

[23] Bremer, T. et al. A biological signature for breast ductal carcinoma in situ to predict radiotherapy benefit and assess recurrence risk. *Clinical Cancer Research*, 24(23):5895–5901, dec 2018. 1.1

[24] Brennan, C. W., Verhaak, R. G. W., McKenna, A., et al. The somatic genomic landscape of glioblastoma. *Cell*, 155(2):462–477, oct 2013. 2.2.3

[25] Brown, M. P. et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1): 262–267, jan 2000. 1.2.2

[26] Burrell, R. A. and Swanton, C. The evolution of the unstable cancer genome. *Current Opinion in Genetics & Development*, 24:61–67, 2014. 6

[27] Cai, C. et al. Systematic discovery of the functional impact of somatic genome alterations in individual tumors through tumor-specific causal inference. *PLOS Computational Biology*, 15(7):e1007088, jul 2019. 1.1, 2, 2.1.1

[28] Campbell, P. J. et al. Pan-cancer analysis of whole genomes. *Nature*, 578(7793):82–93, 2020. 6.1.1, 6.1.2

[29] Cancer Genome Atlas Research Network et al. The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics*, 45(10):1113–1120, oct 2013. 1, 1.1, 6, 6.1.1, 6.1.2

[30] Carter, H. et al. Cancer-specific high-throughput annotation of somatic mutations: Computational prediction of driver missense mutations. *Cancer Research*, 69(16):6660 LP –

6667, aug 2009. 1.1

[31] Caruana, R. Multitask Learning. In Thrun, S. and Pratt, L., editors, *Learning to Learn*, pages 95–133. Springer US, Boston, MA, 1998. 1.2.2

[32] Cawley, G. C. and Talbot, N. L. C. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11: 2079–2107, aug 2010. 1.2.2, 6.1.5

[33] Chaffer, C. L. and Weinberg, R. A. A perspective on cancer cell metastasis. *Science*, 331 (6024):1559–1564, 2011. 4

[34] Chalmers, Z. R. et al. Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden. *Genome Medicine*, 9(1):34, 2017. 1.2.2, 6

[35] Chambers, A. F., Groom, A. C., and MacDonald, I. C. Dissemination and growth of cancer cells in metastatic sites. *Nature Reviews Cancer*, 2(8):563–572, 2002. 4

[36] Chen, F. et al. Single-cell transcriptomic heterogeneity in invasive ductal and lobular breast cancer cells. *Cancer Research*, 81(2):268–281, 2021. 7

[37] Chen, L., Cai, C., Chen, V., and Lu, X. Trans-species learning of cellular signaling systems with bimodal deep belief networks. *Bioinformatics*, 31(18):3008–3015, sep 2015. 2.1.2

[38] Chen, L., Cai, C., Chen, V., and Lu, X. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics*, 17 Suppl 1: 9, jan 2016. 2.1.2, 2.2.1

[39] Chiu, Y.-C. et al. Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC Medical Genomics*, 12(1):18, 2019. 3, 3.1.6, 3.1.6, 3.2.2

[40] Chong, Z. et al. novoBreak: local assembly for breakpoint detection in cancer genomes. *Nature Methods*, 14:65, nov 2016. 6.1.2

[41] Chowdhury, S. A. et al. Algorithms to model single gene, single chromosome, and whole genome copy number changes jointly in tumor phylogenetics. *PLoS Computational Biology*, 10(7):e1003740, 2014. 6

[42] Chowdhury, S. A. et al. Inferring models of multiscale copy number evolution for single-tumor phylogenetics. *Bioinformatics*, 31(12):i258–i267, 2015. 6

[43] Chowdhury, S. A. et al. Phylogenetic analysis of multiprobe fluorescence in situ hybridization data from tumor cell populations. *Bioinformatics*, 29(13):i189–i198, 2013. 6, 6.2.1

[44] Cibulskis, K. et al. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology*, 31:213, feb 2013. 6.1.1, 6.1.2

[45] Ciriello, G., Miller, M. L., Aksoy, B. A., et al. Emerging landscape of oncogenic signatures across human cancers. *Nat. Genet.*, 45:1127, sep 2013. 2

[46] Cox, D. R. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972. 2.2.4, 6.1.4

[47] Crosetto, N., Bienko, M., and van Oudenaarden, A. Spatially resolved transcriptomics and beyond. *Nature Reviews Genetics*, 16(1):57–66, 2015. 7

[48] Cserni, G. et al. The value of cytokeratin immunohistochemistry in the evaluation of axillary sentinel lymph nodes in patients with lobular breast carcinoma. *Journal of Clinical Pathology*, 59(5):518–522, 2006. 6.2.1

[49] Davidson-Pilon, C. et al. CamDavidsonPilon/lifelines: 0.15.3, dec 2018. 6.1.4

[50] Davis, J. and Goadrich, M. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240, New York, NY, USA, jun 2006. Association for Computing Machinery. 1.2.5

[51] Dawson, S.-J., Rueda, O. M., Aparicio, S., and Caldas, C. A new genome-driven integrated classification of breast cancer and its implications. *The EMBO Journal*, 32(5): 617–628, 2013. 1, 1.1, 6.2.2

[52] de Bruin, E. C. et al. Spatial and temporal diversity in genomic instability processes defines lung cancer evolution. *Science (New York, N.Y.)*, 346(6206):251–256, oct 2014. 4

[53] de Souto, M. C. P., Jaskowiak, P. A., and Costa, I. G. Impact of missing data imputation methods on gene expression clustering and classification. *BMC Bioinformatics*, 16:64, feb 2015. 1.2.5

[54] de Weerd, N. A. and Nguyen, T. The interferons and their receptors–distribution and regulation. *Immunol. Cell Biol.*, 90(5):483–491, 2012. 2.2.2

[55] Dees, N. D. et al. MuSiC: identifying mutational significance in cancer genomes. *Genome research*, 22(8):1589–1598, aug 2012. 1.1, 2

[56] Deng, J. et al. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, jun 2009. 1.2.2

[57] Desmedt, C. et al. Biological processes associated with breast cancer clinical outcome depend on the molecular subtypes. *Clinical Cancer Research*, 14(16):5158–5165, 2008. 4.1.3, 5.1.2

[58] Desper, R., Khan, J., and Schäffer, A. A. Tumor classification using phylogenetic methods on expression data. *Journal of Theoretical Biology*, 228(4):477–496, 2004. 4.3, 5.2.4

[59] Dietterich, T. G. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer, 2000. 6.3

[60] Ding, L., Raphael, B. J., Chen, F., and Wendl, M. C. Advances for studying clonal evolution in cancer. *Cancer letters*, 340(2):212–219, nov 2013. 4

[61] Ding, M. Q. et al. Precision oncology beyond targeted therapy: Combining omics data with machine learning matches the majority of cancer cells to effective therapeutics. *Molecular Cancer Research*, 16(2):269–278, feb 2018. 2.2.5, 3, 3.1.1, 3.1.6, 3.1.6

[62] Diz, J., Marreiros, G., and Freitas, A. Applying data mining techniques to improve breast cancer diagnosis. *Journal of Medical Systems*, 40(9):203, sep 2016. 1.2.5

[63] Dowsett, M. et al. Comparison of PAM50 risk of recurrence score with oncotype DX and IHC4 for predicting risk of distant recurrence after endocrine therapy. *Journal of Clinical Oncology*, 31(22):2783–2790, aug 2013. 1.1

[64] Drier, Y., Sheffer, M., and Domany, E. Pathway-based personalized analysis of cancer.

Proceedings of the National Academy of Sciences, 110(16):6388–6393, 2013. 1.2.2

[65] Du, J. et al. Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics*, 20(1):82, 2019. 3.1.4

[66] Eaton, J., Wang, J., and Schwartz, R. Deconvolution and phylogeny inference of structural variations in tumor genomic samples. *Bioinformatics*, 34(13):i357–i365, 2018. 4, 5.3, 6.1.1, 6.1.2, 6.2.1

[67] Elledge, R. M. et al. Estrogen receptor (ER) and progesterone receptor (PgR), by ligand-binding assay compared with ER, PgR and pS2, by immuno-histochemistry in predicting response to tamoxifen in metastatic breast cancer: A Southwest Oncology Group study. *International Journal of Cancer*, 89(2):111–117, 2000. 6.2.1, 6.2.2

[68] Elyanow, R., Dumitrascu, B., Engelhardt, B. E., and Raphael, B. J. netNMF-sc: leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis. *Genome research*, 30(2):195–204, feb 2020. 4, 5.3

[69] Evans, W. E. and Relling, M. V. Pharmacogenomics: translating functional genomics into rational therapeutics. *Science*, 286(5439):487–491, oct 1999. 1, 1.1

[70] Fan, Y. et al. MuSE: accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data. *Genome Biology*, 17(1):178, 2016. 6.1.1, 6.1.2

[71] Floyd, R. W. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):344–348, June 1962. 4.2.2

[72] Foo, J. and Michor, F. Evolution of acquired resistance to anti-cancer therapy. *Journal of Theoretical Biology*, 355:10–20, 2014. 6

[73] Fotso, S. Deep neural networks for survival analysis based on a multi-task framework. *arXiv preprint arXiv:1801.05512*, 2018. 6.1.5, 6.1.6

[74] Fu, X. et al. Joint clustering of single cell sequencing and fluorescence in situ hybridization data for reconstructing clonal heterogeneity in cancers. *Journal of Computational Biology*, 2021. 1.2.4

[75] Funk, S. Netflix update: Try this at home. Technical report, 2006. 4.1.4

[76] Futreal, P. A. et al. A census of human cancer genes. *Nat. Rev. Cancer*, 4(3):177–183, mar 2004. 1.1

[77] Geurts, P., Ernst, D., and Wehenkel, L. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006. 1.1

[78] Goldman, J. M. and Melo, J. V. Chronic myeloid leukemia—advances in biology and new approaches to treatment. *New England Journal of Medicine*, 349(15):1451–1464, oct 2003. 1

[79] Gonzalez-Perez, A. and Lopez-Bigas, N. Functional impact bias reveals cancer drivers. *Nucleic Acids Research*, 40(21):e169–e169, nov 2012. 1.1

[80] Gonzalez-Perez, A. et al. IntOGen-mutations identifies cancer drivers across tumor types. *Nature Methods*, 10:1081, sep 2013. 1.1, 2.2.3, 6, 6.1.3, 6.2.1

[81] Greaves, M. and Maley, C. C. Clonal evolution in cancer. *Nature*, 481(7381):306–313, 2012. 4, 6

[82] Greenman, C. et al. Patterns of somatic mutation in human cancer genomes. *Nature*, 446: 153, mar 2007. 1.1

[83] Guan, X. Cancer metastases: Challenges and opportunities. *Acta Pharmaceutica Sinica B*, 5(5):402–418, sep 2015. 4

[84] Gupta, S., Takebe, N., and Lorusso, P. Targeting the Hedgehog pathway in cancer. *Therapeutic Advances in Medical Oncology*, 2(4):237–250, jul 2010. 4.2.2, 5.2.5

[85] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, jan 2002. 1.2.2

[86] Hady, M. F. A. and Schwenker, F. Semi-supervised Learning. In Bianchini, M., Maggini, M., and Jain, L. C., editors, *Handbook on Neural Information Processing*, pages 215–239. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. 1.2.2

[87] Hastie, T., Tibshirani, R., and Tibshirani, R. Best subset, forward stepwise or lasso? analysis and recommendations based on extensive comparisons. *Statistical Science*, 35 (4):579–592, November 2020. 6.1.6

[88] Heselmeyer-Haddad, K. et al. Single-cell genetic analysis of ductal carcinoma in situ and invasive breast cancer reveals enormous tumor heterogeneity yet conserved genomic imbalances and gain of *MYC* during progression. *American Journal of Pathology*, 181(5): 1807–1822, 2012. 1.2.4

[89] Hoadley, K. A., Yau, C., Wolf, D. M., et al. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158(4):929–944, aug 2014. 2.2.4

[90] Hofer, A. M. and Lefkimmiatis, K. Extracellular Calcium and cAMP: Second messengers as "Third Messengers"? *Physiology*, 22(5):320–327, oct 2007. 4.2.2, 5.2.4

[91] Hofree, M. et al. Network-based stratification of tumor mutations. *Nature Methods*, 10 (11):1108–1115, nov 2013. 1.2.3

[92] Hosack, D. A. et al. Identifying biological themes within lists of genes with EASE. *Genome Biology*, 4(10):R70–R70, 2003. 4.1.3

[93] Hoyer, P. O. Non-Negative Matrix Factorization with Sparseness Constraints. *J. Mach. Learn. Res.*, 5:1457–1469, dec 2004. 5.1.2

[94] Huang, D. W., Sherman, B. T., and Lempicki, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, 4(1):44–57, 2009. 1.2.2, 4.1.3, 4.1.3, 5.1.2

[95] Hutter, C. and Zenklusen, J. C. The Cancer Genome Atlas: Creating lasting value beyond its data. *Cell*, 173(2):283–285, apr 2018. 1.1, 1.2.1

[96] ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium. Pan-cancer analysis of whole genomes. *Nature*, 578(7793):82–93, feb 2020. 1.1

[97] International Cancer Genome Consortium. ICGC Data Portal. `https://dcc.icgc.`

org/repositories, 2019. Accessed 6 February 2019. 6.1.2

[98] International Cancer Genome Consortium et al. International network of cancer genome projects. *Nature*, 464(7291):993–998, apr 2010. 1

[99] International Human Genome Sequencing Consortium et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860, 2001. 1, 6.1.3

[100] Jamal-Hanjani, M., Quezada, S. A., Larkin, J., and Swanton, C. Translational implications of tumor heterogeneity. *Clinical Cancer Research*, 21(6):1258–1266, mar 2015. 1.2.4

[101] Jensen, M. A., Ferretti, V., Grossman, R. L., and Staudt, L. M. The NCI Genomic Data Commons as an engine for precision medicine. *Blood*, 130(4):453–459, jul 2017. 1.1

[102] Jiang, Y., Qiu, Y., Minn, A. J., and Zhang, N. R. Assessing intratumor heterogeneity and tracking longitudinal and spatial clonal evolutionary history by next-generation sequencing. *Proceedings of the National Academy of Sciences*, 113(37):E5528–E5537, 2016. 6.1.1, 6.1.2

[103] Jones, P. A. and Baylin, S. B. The fundamental role of epigenetic events in cancer. *Nat. Rev. Genet.*, 3:415, jun 2002. 2

[104] Kandoth, C., McLellan, M. D., Vandin, F., et al. Mutational landscape and significance across 12 major cancer types. *Nature*, 502:333, oct 2013. 2, 2.2.3

[105] Kanehisa, M. and Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, jan 2000. 1.2.2, 4.1.3, 4.1.3, 5.1.2

[106] Katzman, J. L. et al. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1): 24, 2018. 6.3

[107] Kim, Y.-A., Madan, S., and Przytycka, T. M. WeSME: Uncovering mutual exclusivity of cancer drivers and beyond. *Bioinformatics*, 33(6):814–821, 2017. 6

[108] King, M.-C., Marks, J. H., and Mandell, J. B. Breast and ovarian cancer risks due to inherited mutations in BRCA1 and BRCA2. *Science*, 302(5645):643–646, 2003. 6

[109] Kingma, D. P. and Ba, J. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2.1.3, 4.1.4, 4.1.4

[110] Kishikawa, T. et al. Empirical evaluation of variant calling accuracy using ultra-deep whole-genome sequencing data. *Scientific Reports*, 9(1):1784, 2019. 6

[111] Kohavi, R. and John, G. H. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997. 1.2.2, 6.1.6

[112] Körber, V. et al. Evolutionary trajectories of IDHWT glioblastomas reveal a common path of early tumorigenesis instigated years ahead of initial diagnosis. *Cancer Cell*, mar 2019. 4

[113] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, aug 2009. 4.1.4, 4.1.4

[114] Kuha, J. AIC and BIC: Comparisons of assumptions and performance. *Sociological Methods & Research*, 33(2):188–229, nov 2004. 1.2.2

[115] Lan, K., Wang, D., Fong, S., et al. A survey of data mining and deep learning in bioinformatics. *J. Med. Syst.*, 42(8):139, 2018. 2

[116] Langer-Safer, P. R., Levine, M., and Ward, D. C. Immunological method for mapping genes on Drosophila polytene chromosomes. *Proceedings of the National Academy of Sciences of the United States of America*, 79(14):4381–4385, jul 1982. 6

[117] Lappalainen, I. et al. The European Genome-phenome Archive of human data consented for biomedical research. *Nature Genetics*, 47(7):692–695, jul 2015. 1.1

[118] Lappalainen, T., Scott, A. J., Brandt, M., and Hall, I. M. Genomic analysis in the age of human genome sequencing. *Cell*, 177(1):70–84, 2019. 1

[119] Lawrence, M. S. et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, 499(7457):214, jun 2013. 1.1, 1.2.3, 2, 6

[120] Lawrence, M. S. et al. Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature*, 505:495, jan 2014. 1, 2, 2.2.3

[121] Lee, B., Min, S., and Yoon, S. Deep learning in bioinformatics. *Brief. Bioinform.*, 18(5): 851–869, 2016. 2

[122] Lee, D. D. and Seung, H. S. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 535–541, Cambridge, MA, USA, 2000. MIT Press. 4.1.4, 4.1.4, 4.2.1, 5, 5.1.2, 5.1.2

[123] Lee, S. and Margolin, K. Cytokines in cancer immunotherapy. *Cancers*, 3(4):3856–3893, oct 2011. 4.2.2

[124] Lei, H. et al. Tumor copy number deconvolution integrating bulk and single-cell sequencing data. *Journal of Computational Biology*, 27(4):565–598, mar 2020. 4.1.4, 5.1.2, 6.1.1

[125] Lei, H. et al. Tumor heterogeneity assessed by sequencing and fluorescence in situ hybridization (FISH) data. *Bioinformatics*, jul 2021. 1.2.4, 3.3, 4.1.4, 5.1.2, 6.1.1

[126] Leung, M. K., Delong, A., Alipanahi, B., and Frey, B. J. Machine learning in genomic medicine: A review of computational problems and data sets. *Proceedings of the IEEE*, 104(1):176–197, 2016. 1

[127] Li, Y., Zhou, S., Schwartz, D. C., and Ma, J. Allele-specific quantification of structural variations in cancer genomes. *Cell Systems*, 3(1):21–34, 2016. 6.1.2

[128] Lin, N. U., Bellon, J. R., and Winer, E. P. CNS metastases in breast cancer. *Journal of Clinical Oncology*, 22(17):3608–3617, sep 2004. 4

[129] Liu, C. X. et al. The putative tumor suppressor LRP1B, a novel member of the low density lipoprotein (LDL) receptor family, exhibits both overlapping and distinct properties with the LDL receptor-related protein. *The Journal of Biological Chemistry*, 276:28889–28896, 2001. 6.2.1

[130] Liu, H. Application of immunohistochemistry in breast pathology: a review and update. *Archives of Pathology & Laboratory Medicine*, 138(12):1629–1642, dec 2014. 6.2.1

[131] Liu, H., Zhao, Y., Zhang, L., and Chen, X. Anti-cancer drug response prediction using

neighbor-based collaborative filtering with global effect removal. *Molecular Therapy - Nucleic Acids*, 13:303–311, 2018. 3

[132] Liu, J. et al. An integrated TCGA Pan-Cancer clinical data resource to drive High-Quality survival outcome analytics. *Cell*, 173(2):400—-416.e11, apr 2018. 1.2.5

[133] Liu, J. et al. Distance-based clustering of CGH data. *Bioinformatics*, 22(16):1971–1978, aug 2006. 1.2.3

[134] Liu, Q. et al. Network-based matching of patients and targeted therapies for precision oncology. *Pacific Symposium on Biocomputing*, 25:623–634, 2020. 1.1, 3.3

[135] Loeb, L. A. A mutator phenotype in cancer. *Cancer Research*, 61(8):3230–3239, 2001. 6, 6.2.1, 6.2.2

[136] Lu, C. L., Tang, C. Y., and Lee, R. C.-T. The full Steiner tree problem. *Theoretical Computer Science*, 306(1):55–67, sep 2003. 4.1.5

[137] Lyman, G. H. et al. Age and the risk of breast cancer recurrence. *Cancer Control*, 3(5): 421–427, 1996. 6.2.1

[138] Lynch, T. J. et al. Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib. *N. Engl. J. Med.*, 350(21):2129–2139, may 2004. 1.1

[139] Ma, J. et al. Using deep learning to model the hierarchical structure and function of a cell. *Nature Methods*, 15:290, mar 2018. 2.3

[140] Maaten, L. and Hinton, G. Visualizing high-dimensional data using t-SNE. *J. Mach. Learn. Res.*, 9:2579–2605, 01 2008. 2.2.2

[141] Macintyre, G. et al. Copy number signatures and mutational processes in ovarian carcinoma. *Nature Genetics*, 50(9):1262, 2018. 6

[142] Malikic, S. et al. Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data. *Nature Communications*, 10(1):2750, 2019. 1.2.4, 6.1.1

[143] Malikic, S. et al. PhISCS: a combinatorial approach for subperfect tumor phylogeny reconstruction via integrative use of single-cell and bulk sequencing data. *Genome Research*, 29(11):1860–1877, nov 2019. 1.2.4

[144] Malta, T. M. et al. Machine learning identifies stemness features associated with oncogenic dedifferentiation. *Cell*, 173(2):338—-354.e15, apr 2018. 1.1

[145] Mantel, N. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemotherapy Reports*, 50(3):163–170, March 1966. 2.2.4, 6.1.5

[146] Marusyk, A. and Polyak, K. Tumor heterogeneity: causes and consequences. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1805(1):105–117, 2010. 6

[147] Massagué, J. TGFβ in cancer. *Cell*, 134(2):215–230, 2008. 4.2.2

[148] McAuley, J. and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, RecSys '13, pages 165–172, New York, NY, USA, oct 2013. Association for Computing Machinery. 1.2.2

[149] McGranahan, N. and Swanton, C. Biological and therapeutic impact of intratumor heterogeneity in cancer evolution. *Cancer Cell*, 27(1):15–26, jan 2015. 1

[150] Mendelsohn, J. and Baselga, J. The EGF receptor family as targets for cancer therapy. *Oncogene*, 19(56):6550–6565, dec 2000. 1.1

[151] Mermel, C. H. et al. GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome Biology*, 12 (4):R41, 2011. 6.1.1, 6.1.2

[152] Mi, H., Muruganujan, A., Casagrande, J. T., and Thomas, P. D. Large-scale gene function analysis with the PANTHER classification system. *Nature Protocols*, 8:1551, jul 2013. 1.2.2, 2.2.2, 3.2.4

[153] Mikolov, T. et al. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc. 2, 2.1.2, 3.1.4

[154] Moffat, J. G., Rudolph, J., and Bailey, D. Phenotypic screening in cancer drug discovery - past, present and future. *Nature Reviews Drug Discovery*, 13(8):588–602, aug 2014. 1.1

[155] Muralidharan-Chari, V. et al. Microvesicle removal of anticancer drugs contributes to drug resistance in human pancreatic cancer cells. *Oncotarget*, 7(31):50365–50379, aug 2016. 3.2.4

[156] Nagano, M. et al. Turnover of Focal Adhesions and Cancer Cell Migration. *International Journal of Cell Biology*, 2012:310616, 2012. 5.2.5

[157] Nakagawa, H. and Fujita, M. Whole genome sequencing analysis for cancer genomics and precision medicine. *Cancer science*, 109(3):513–522, 2018. 1

[158] Navin, N. et al. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341): 90, 2011. 6.1.1

[159] NCI Genomic Data Commons. Genomic Data Commons Data Portal. `https://portal.gdc.cancer.gov`, 2018. Accessed 22 October 2018. 6.1.2

[160] Nei, M. and Saitou, N. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, jul 1987. 4.1.5, 5.1.3

[161] Network, T. C. G. A. et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490:61, sep 2012. 2.2.3, 6, 6.1.1

[162] Network, T. C. G. A. R. et al. Comprehensive genomic characterization of squamous cell lung cancers. *Nature*, 489:519, sep 2012. 6, 6.1.1

[163] Network, T. C. G. A. R. et al. The cancer genome atlas pan-cancer analysis project. *Nat. Genet.*, 45:1113, sep 2013. 2.1.1

[164] Network, T. C. G. A. R. et al. Comprehensive molecular characterization of urothelial bladder carcinoma. *Nature*, 507:315, jan 2014. 2.2.3

[165] Network, T. C. G. A. R. et al. Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, 511:543, jul 2014. 2.2.3, 6, 6.1.1

[166] Newman, A. M. et al. Robust enumeration of cell subsets from tissue expression profiles. *Nature methods*, 12(5):453–457, may 2015. 5.1.4

[167] Nielsen, T. O. et al. A comparison of PAM50 intrinsic subtyping with immunohistochemistry and clinical prognostic factors in tamoxifen-treated estrogen receptor-positive breast cancer. *Clin. Cancer Res.*, 16(21):5222–5232, nov 2010. 1

[168] Niu, B. et al. Protein-structure-guided discovery of functional mutations across 19 cancer types. *Nature Genetics*, 48:827, jun 2016. 1.1, 2

[169] Nowell, P. C. The clonal evolution of tumor cell populations. *Science*, 194(4260):23–28, 1976. 6

[170] Oltmann, J. et al. Aneuploidy, TP53 mutation, and amplification of MYC correlate with increased intratumor heterogeneity and poor prognosis of breast cancer patients. *Genes, Chromosomes and Cancer*, 57(4):165–175, 2018. 6, 6.3

[171] Oskooei, A. et al. PaccMann: Prediction of anticancer compound sensitivity with multi-modal attention-based neural networks, 2018. 3.1.2

[172] Paez, J. G. et al. EGFR mutations in lung cancer: correlation with clinical response to gefitinib therapy. *Science*, 304(5676):1497–1500, jun 2004. 1

[173] Park, S. Y. et al. Cellular and genetic diversity in the progression of in situ human breast carcinomas to an invasive phenotype. *The Journal of Clinical Investigation*, 120(2):636–644, 2010. 6.2.1

[174] Park, Y., Shackney, S., and Schwartz, R. Network-based inference of cancer progression from microarray data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(2):200–212, apr 2009. 1.2.2, 1.2.3, 4, 4.1.3, 4.1.5, 4.2.2, 5.1.2

[175] Parker, J. S. et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, 27(8):1160–1167, mar 2009. 1, 1.1

[176] Paszke, A. et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019. 3.1.5

[177] Pennington, G., Smith, C. A., Shackney, S., and Schwartz, R. Reconstructing tumor phylogenies from heterogeneous single-cell data. *Journal of Bioinformatics and Computational Biology*, 5(02a):407–427, 2007. 6

[178] Pennington, J., Socher, R., and Manning, C. D. GloVe: global vectors for word representation. In *Proc. of EMNLP*, volume 14, pages 1532–1543, 2014. 2

[179] Pereira, B. et al. The somatic mutation profiles of 2,433 breast cancers refines their genomic and transcriptomic landscapes. *Nature Communications*, 7:11479, may 2016. 1.1

[180] Polyak, K. et al. Breast cancer: origins and evolution. *The Journal of clinical investigation*, 117(11):3155–3163, 2007. 1

[181] Prasad, V. Perspective: The precision-oncology illusion. *Nature*, 537(7619):S63–S63, 2016. 3

[182] Priedigkeit, N. et al. Intrinsic subtype switching and acquired ERBB2/HER2 amplifications and mutations in breast cancer brain metastases. *JAMA Oncology*, 3(5):666–671, may 2017. 3, 4, 4.1.2, 4.2.2, 4.2.4, 4.3, 5, 5.2.5

[183] Priedigkeit, N. et al. Exome-capture RNA sequencing of decade-old breast cancers and matched decalcified bone metastases. *JCI insight*, 2(17), sep 2017. 5

[184] Qiu, P. et al. Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature Biotechnology*, 29(10):886–891, 2011. 4, 4.3

[185] Quigley, D. A. et al. Genomic hallmarks and structural variation in metastatic prostate cancer. *Cell*, 174(3):758–769, 2018. 6, 6.3

[186] Rajaraman, A. and Ma, J. Toward recovering allele-specific cancer genome graphs. *Journal of Computational Biology*, 25(7):624–636, 2018. 6.1.2

[187] Ranganath, R., Perotte, A., Elhadad, N., and Blei, D. Deep survival analysis. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 101–114, Children's Hospital LA, Los Angeles, CA, USA, 18–19 Aug 2016. 6.3

[188] Relling, M. V. and Evans, W. E. Pharmacogenomics in the clinic. *Nature*, 526(7573): 343–350, oct 2015. 1, 1.1

[189] Reuter, J. A., Spacek, D. V., and Snyder, M. P. High-throughput sequencing technologies. *Molecular Cell*, 58(4):586–597, may 2015. 3

[190] Reva, B., Antipin, Y., and Sander, C. Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic Acids Research*, 39(17):e118–e118, sep 2011. 1.1, 2

[191] Riester, M. et al. A differentiation-based phylogeny of cancer subtypes. *PLOS Computational Biology*, 6(5):e1000777, may 2010. 4.3, 5.2.4

[192] Roman, T., Nayyeri, A., Fasy, B. T., and Schwartz, R. A simplicial complex-based approach to unmixing tumor progression data. *BMC bioinformatics*, 16:254, aug 2015. 5.2.4

[193] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, pages 65–386, 1958. 2.1.2, 2.2.1

[194] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323:533, oct 1986. 2.1.2, 2.1.3, 4.1.4

[195] Saeys, Y., Inza, I., and Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, oct 2007. 1.2.2

[196] Sahmoun, A. E., Case, L. D., Santoro, T. J., and Schwartz, G. G. Anatomical distribution of small cell lung cancer: effects of lobe and gender on brain metastasis and survival. *Anticancer Research*, 25(2A):1101–1108, 2005. 6.2.1

[197] Samstein, R. M. et al. Tumor mutational load predicts survival after immunotherapy across multiple cancer types. *Nature genetics*, 51(2):202–206, feb 2019. 6

[198] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. *Collaborative Filtering Recommender Systems*, pages 291–324. Springer-Verlag, 2007. 3, 3.1.1

[199] Schwartz, R. and Schäffer, A. A. The evolution of tumour phylogenetics: principles and practice. *Nature Reviews Genetics*, 18:213, feb 2017. 1.2.4, 3, 3.3, 4, 4.1.5, 5.1.3, 6

[200] Schwartz, R. and Shackney, S. E. Applying unmixing to gene expression data for tumor phylogeny inference. *BMC Bioinformatics*, 11(1):42, jan 2010. 1.2.4, 4, 4.2.1, 4.3, 5, 5.2.3, 5.2.4

[201] Schwarz, G. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, 1978. 6.1.6

[202] Sharifi-Noghabi, H., Zolotareva, O., Collins, C. C., and Ester, M. MOLI: multi-omics late integration with deep neural networks for drug response prediction. *Bioinformatics*, 35 (14):i501–i509, jul 2019. 1.1, 3.1.6

[203] Shen, B., Liu, B., Wang, Q., and Ji, R. Robust nonnegative matrix factorization via L1 norm regularization by multiplicative updating rules. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5282–5286, 2014. 5.1.2

[204] Shen-Orr, S. S. et al. Cell type-specific gene expression differences in complex tissues. *Nature methods*, 7(4):287–289, apr 2010. 5.1.2, 5.1.5, 5.1.5

[205] Shinbrot, E. et al. Exonuclease mutations in DNA polymerase epsilon reveal replication strand specific mutation patterns and human origins of replication. *Genome Research*, 24: 1740–1750, 2014. 6

[206] Shoemaker, R. H. The NCI60 human tumour cell line anticancer drug screen. *Nature Reviews Cancer*, 6(10):813–823, 2006. 1.1, 1.1

[207] Sicklick, J. K. et al. Molecular profiling of cancer patients enables personalized combination therapy: the I-PREDICT study. *Nat. Med.*, 25(5):744–750, may 2019. 1

[208] Sjöblom, T. et al. The consensus coding sequences of human breast and colorectal cancers. *Science*, 314(5797):268–274, oct 2006. 1.1

[209] Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018. 3.1.5

[210] Srivastava, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 2.1.3, 3.1.1

[211] Steck, H. et al. On ranking in survival analysis: Bounds on the concordance index. In *Advances in Neural Information Processing Systems*, pages 1209–1216, 2008. 6.1.5, 6.1.6

[212] Stransky, N., Egloff, A. M., Tward, A. D., et al. The mutational landscape of head and neck squamous cell carcinoma. *Science*, 333(6046):1157–1160, aug 2011. 2.2.3

[213] Stransky, N. et al. The landscape of kinase fusions in cancer. *Nature Communications*, 5: 4846, sep 2014. 2

[214] Suvà, M. L. and Tirosh, I. Single-Cell RNA sequencing in cancer: Lessons learned and emerging challenges. *Molecular Cell*, 75(1):7–12, jul 2019. 1.2.4

[215] Swanton, C. Intratumor heterogeneity: evolution through space and time. *Cancer Research*, 72(19):4875–4882, oct 2012. 1.2.4

[216] Swanton, C., McGranahan, N., Starrett, G. J., and Harris, R. S. APOBEC enzymes:

mutagenic fuel for cancer evolution and heterogeneity. *Cancer Discovery*, 5(7):704–712, 2015. 6

[217] Tan, K. S., Eguchi, T., and Adusumilli, P. S. Competing risks and cancer-specific mortality: why it matters. *Oncotarget*, 9(7):7272–7273, dec 2017. 6.2.1

[218] Tao, Y., Godefroy, B., Genthial, G., and Potts, C. Effective feature representation for clinical text concept extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 1–14, Minneapolis, Minnesota, USA, jun 2019. Association for Computational Linguistics. 2, 3.1.4

[219] Tao, Y. et al. Phylogenies derived from matched transcriptome reveal the evolution of cell populations and temporal order of perturbed pathways in breast cancer brain metastases. In *Mathematical and Computational Oncology*, pages 3–28. Springer International Publishing, 2019. 1.3, 1.4, 1, 5, 5.1.1, 5.1.2, 5.1.2, 5.1.3, 5.1.3, 5.2.2, 5.2.3, 5.2.3, 5.2.5

[220] Tao, Y. et al. Improving personalized prediction of cancer prognoses with clonal evolution models. *bioRxiv*, 2019. 4

[221] Tao, Y., Cai, C., Cohen, W. W., and Lu, X. From genome to phenome: Predicting multiple cancer phenotypes based on somatic genomic alterations via the genomic impact transformer. In *Pacific Symposium on Biocomputing*, volume 25, pages 79–90. World Scientific, 2020. 1.1, 1.1, 1.3, 1.4, 1, 3, 3.1.4

[222] Tao, Y. et al. Robust and accurate deconvolution of tumor populations uncovers evolutionary mechanisms of breast cancer metastasis. *Bioinformatics*, 36:i407–i416, jul 2020. 1.1, 1.3, 1.4, 3.3, 1, 6.1.1

[223] Tao, Y. et al. Neural network deconvolution method for resolving pathway-level progression of tumor clonal expression programs with application to breast cancer brain metastases. *Frontiers in Physiology*, 11:1055, 2020. 1.1, 1.2.2, 1.3, 1.4, 1, 6.1.1

[224] Tao, Y. et al. Predicting drug sensitivity of cancer cell lines via collaborative filtering with contextual attention. In Doshi-Velez, F. et al., editors, *Proceedings of the 5th Machine Learning for Healthcare Conference*, volume 126 of *Proceedings of Machine Learning Research*, pages 660–684, Virtual, 07–08 Aug 2020. PMLR. 1.1, 1.1, 1.1, 1.3, 1.4, 1

[225] Tao, Y. et al. Assessing the contribution of tumor mutational phenotypes to cancer progression risk. *PLOS Computational Biology*, 17(3):1–29, 03 2021. 1.1, 1.1, 1.3, 1.4, 1

[226] Tate, J. G. et al. COSMIC: the catalogue of somatic mutations in cancer. *Nucleic Acids Research*, 47(D1):D941–D947, 2018. 6.1.3, 6.2.1

[227] Te Raa, G. D. and Kater, A. P. TP53 dysfunction in CLL: Implications for prognosis and treatment. *Best Practice & Research: Clinical Haematology*, 29(1):90–99, mar 2016. 6

[228] The 1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68, 2015. 6.1.3

[229] Tibshirani, R. The lasso method for variable selection in the Cox model. *Stat. Med.*, 16 (4):385–395, feb 1997. 1.2.2

[230] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal*

Statistical Society, Series B, 58:267–288, 1994. 1.2.2, 2.2.1

[231] Tryka, K. A. et al. NCBI's Database of Genotypes and Phenotypes: dbGaP. *Nucleic Acids Research*, 42(Database issue):D975—-9, jan 2014. 1.1

[232] van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 3.2.3

[233] van 't Veer, L. J. et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, jan 2002. 1, 1.1

[234] Vandin, F., Upfal, E., and Raphael, B. J. De novo discovery of mutated driver pathways in cancer. *Genome research*, 22(2):375–385, feb 2012. 1, 2.1.2

[235] Vareslija, D. et al. Transcriptome characterization of matched primary breast and brain metastatic tumors to detect novel actionable targets. *Journal of the National Cancer Institute*, jun 2018. 4, 4.1.2, 4.1.3, 4.2.2, 4.2.4, 4.3

[236] Vaswani, A., Shazeer, N., Parmar, N., et al. Attention is all you need. In *Proc. of NIPS*. Curran Associates, Inc., 2017. 2

[237] Venet, D., Pecasse, F., Maenhaut, C., and Bersini, H. Separation of samples into their constituents using gene expression data. *Bioinformatics*, 17 Suppl 1:S279—-87, 2001. 1.2.4

[238] Venter, J. C. et al. The sequence of the human genome. *Science*, 291(5507):1304–1351, feb 2001. 1

[239] Vogelstein, B. et al. Cancer genome landscapes. *Science*, 339(6127):1546–1558, 2013. 1, 2, 2.2.3

[240] Wala, J. A. et al. Selective and mechanistic sources of recurrent rearrangements across the cancer genome. *BioRxiv*, page 187609, 2017. 6

[241] Wang, H. et al. Automatic human-like mining and constructing reliable genetic association database with deep reinforcement learning. In *Pacific Symposium on Biocomputing*, volume 24, pages 112–123. World Scientific, 2019. 3.1.4

[242] Wang, L., Li, X., Zhang, L., and Gao, Q. Improved anticancer drug response prediction in cell lines using matrix factorization with similarity regularization. *BMC Cancer*, 17(1): 513, aug 2017. 3

[243] Wang, P., Li, Y., and Reddy, C. K. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, 51(6):110, 2019. 6.1.4

[244] Wang, Z. et al. Cancer driver mutation prediction through Bayesian integration of multiomic data. *PLoS One*, 13(5):e0196939, may 2018. 1.1

[245] Wangsa, D. et al. Phylogenetic analysis of multiple FISH markers in oral tongue squamous cell carcinoma suggests that a diverse distribution of copy number changes is associated with poor prognosis. *International Journal of Cancer*, 138(1):98–109, jan 2016. 6.1.7

[246] Ward, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, mar 1963. 4.2.2, 6.2.2

[247] Warshall, S. A theorem on boolean matrices. *Journal of the ACM*, 9(1):11–12, jan 1962.

4.2.2

[248] Waszak, S. M. et al. Germline determinants of the somatic mutation landscape in 2,642 cancer genomes. *BioRxiv*, page 208330, 2017. 6.3

[249] Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big Data*, 3(1):9, may 2016. 1.2.2

[250] Williams, M. J. et al. Identification of neutral tumor evolution across cancer types. *Nature Genetics*, 48(3):238–244, 2016. 6

[251] Witzel, I. et al. Breast cancer brain metastases: biology and new clinical perspectives. *Breast Cancer Research*, 18(1):8, jan 2016. 4

[252] Wong, R. S. Y. Apoptosis in cancer: from pathogenesis to treatment. *Journal of Experimental & Clinical Cancer Research*, 30(1):87, sep 2011. 4.2.2

[253] Woodburn, J. R. The epidermal growth factor receptor and its inhibition in cancer therapy. *Pharmacol. Ther.*, 82(2-3):241–250, may 1999. 1.1

[254] Xing, E. P., Jordan, M. I., and Karp, R. M. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML ’01, pages 601–608, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. 1.2.2

[255] Xu, K. et al. Show, attend and tell: neural image caption generation with visual attention. In Bach, F. and Blei, D., editors, *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR. 2, 3

[256] Yang, J. et al. The value of positive lymph nodes ratio combined with negative lymph node count in prediction of breast cancer survival. *Journal of Thoracic Disease*, 9(6): 1531, 2017. 6.2.1

[257] Yang, W. et al. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Research*, 41(Database issue):D955–61, jan 2013. 1.1, 1.1, 3, 3.1.6, 3.2.4

[258] Yang, Z. et al. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, jun 2016. 3, 3.1.2, 3.1.2

[259] Yao, J., Zhu, X., Zhu, F., and Huang, J. Deep correlational learning for survival prediction from multi-modality data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 406–414. Springer, 2017. 6.3

[260] Yates, L. R. et al. Genomic evolution of breast cancer metastasis and relapse. *Cancer Cell*, 32(2):169–184, 2017. 6, 6.3

[261] Yuan, H. et al. Multitask learning improves prediction of cancer drug sensitivity. *Scientific Reports*, 6(1):31619, 2016. 3

[262] Yuan, H. et al. CancerSEA: a cancer single-cell state atlas. *Nucleic Acids Research*, 47

(D1):D900—-D908, jan 2019. 1.1

[263] Yung, C. K. et al. Large-scale uniform analysis of cancer whole genomes in multiple computing environments. *BioRxiv*, page 161638, 2017. 6.1.1

[264] Zack, T. I. et al. Pan-cancer patterns of somatic copy number alteration. *Nature Genetics*, 45(10):1134–1140, sep 2013. 2, 6.2.1

[265] Zaitsev, K., Bambouskova, M., Swain, A., and Artyomov, M. N. Complete deconvolution of cellular mixtures based on linearity of transcriptional signatures. *Nature Communications*, 10(1):2209, 2019. 3, 3.2.4, 5, 5.1.2, 5.1.4, 5.1.5, 5.2.3

[266] Zare, H. et al. Inferring Clonal Composition from Multiple Sections of a Breast Cancer. *PLOS Computational Biology*, 10(7):e1003703, jul 2014. 4

[267] Zhan, T., Rindtorff, N., and Boutros, M. Wnt signaling in cancer. *Oncogene*, 36:1461, sep 2016. 4.2.2

[268] Zhang, J. et al. International Cancer Genome Consortium Data Portal–a one-stop shop for cancer genomics data. *Database: The Journal of Biological Databases and Curation*, 2011:bar026–bar026, sep 2011. 6, 6.1.1, 6.1.2

[269] Zhang, N. et al. Predicting anticancer drug responses using a dual-layer integrated cell line-drug network model. *PLoS Computational Biology*, 11(9):e1004498, 2015. 3.3

[270] Zhang, Z. et al. Overcoming cancer therapeutic bottleneck by drug repurposing. *Signal Transduction and Targeted Therapy*, 5(1):113, jul 2020. 1.1

[271] Zhao, G. and Wu, Y. Feature subset selection for cancer classification using weight local modularity. *Scientific Reports*, 6:34759, oct 2016. 1.2.2

[272] Zhong, Y. and Liu, Z. Gene expression deconvolution in linear space. *Nature Methods*, 9 (1):8–9, 2012. 5.1.3

[273] Zhong, Y. et al. Digital sorting of complex tissues for cell type-specific gene expression profiles. *BMC Bioinformatics*, 14(1):89, 2013. 5.1.2, 5.2.3

[274] Zhu, L. et al. Metastatic breast cancers have reduced immune cell recruitment but harbor increased macrophages relative to their matched primary tumors. *Journal for ImmunoTherapy of Cancer*, 7(1):265, 2019. 1.2.4, 4.2.2, 5, 5.1.5, 5.1.5, 5.2.4, 5.2.5

[275] Zhu, L., Lei, J., Devlin, B., and Roeder, K. A unified statistical framework for single cell and bulk rna sequencing data. *The annals of applied statistics*, 12(1):609–632, mar 2018. 4.2.1, 5.1.4, 5.3

[276] Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. B*, 67(2):301–320, 2005. 2.2.4