Detecting Biomarkers among Subgroups with Structured Latent Features and Multitask Learning Methods

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Huanan Zhang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Rui Kuang

May, 2017

Acknowledgements

There are many people that have earned my gratitude for their contribution to my time in graduate school.

First and foremost, I would like to express my most sincere gratitude and appreciation to my advisor, Dr Rui Kuang. His insightful guidance and invaluable support guide me through my Ph.D studies. Not only his knowledge, but also his dedication and enthusiasm on research inspire me deeply and will assist me in my further work. This thesis would not have been possible without his support and encouragement.

The members of Computational Biology group contributed immensely to my personal and professional time at University of Minnesota. I am grateful for all the help from Dr. Ze Tian, Dr. Wei Zhang and Dr. Taehyun Hwang. I also want to express my appreciation to Dr. Nicholas Johnson, Dr. Yuguo Xiao, David Roe and Catherine Lee for all the great collaborations. I also want to thank my labmates: Raphael Petegrosso, Zhuliu Li, and Nishitha Paidimukkala who supported me in various ways.

For this thesis, I would like to thank my committee members Dr. Vipin Kumar, Dr. Chad Myers and Dr. Min Ni for their time, interest, insightful questions and helpful comments.

Lastly, I would like to thank my family who support me all the times with love, support and encouragement.

Dedication

To my parents, my wife Jin and my beautiful daughter Chloe.

Abstract

Because of disease progression and heterogeneity in samples and single cells, biomarker detection among subgroups is important as it provides better understanding on population genetics and cancer causative. In this thesis, we proposed several structured latent features based and multitask learning based methods for biomarker detection on DNA Copy-Number Variations (CNVs) data and single cell RNA sequencing (scRNA-seq) data. By incorporating prior known group information or taking domain heterogeneity into consideration, our models are able to achieve meaningful biomarker detection and accurate sample classification.

1. By cooperating population relationship from human phylogenetic tree, we introduced a latent feature model to detect population-differentiation CNV markers. The algorithm, named tree-guided sparse group selection ($treeSGS$), detects sample subgroups organized by a population phylogenetic tree such that the evolutionary relations among the populations are incorporated for more accurate detection of population-differentiation CNVs. 2. We applied transfer learning technic for cross-cancer-type CNV studies. We proposed Transfer Learning with Fused LASSO ($TLFL$) algorithm, which detects latent CNV components from multiple CNV datasets of different tumor types and distinguishes the CNVs that are common across the datasets and those that are specific in each dataset. Both the common and type-specific CNVs are detected as latent components in matrix factorization coupled with fused LASSO on adjacent CNV probe features. 3. We further applied multitask learning idea on scRNA-seq data. We introduced variance-driven multitask clustering on single-cell RNA-seq data ($scVDMC$) that utilizes multiple cell populations from biological replicates or related samples with significant biological variances. $scVDMC$ clusters single cells of similar cell types and markers but varies expression patterns across different domains such that the scRNA-seq data are adjusted for better integration.

We applied both simulations and several publicly available CNV and scRNA-seq datasets, including one in house scRNA-seq dataset, to evaluate the performance of our models. The promising results show that we achieve better biomarker prediction among subgroups.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Subgroup structures widely exist in genomic datasets. In population genetics, which focus on the study of genetic variations within and between populations, samples are normally analyzed in subgroups to learn distributions and changes in genotype and phenotype frequency [1]. Genetic biomarkers, such as single nucleotide polymorphism and copy number variation among sample subgroups contributes our understanding of how evolution acts on genetic variation and, with the help of advanced sequencing technology, even allows data to be attached to the points at which populations start to diverge [2,3]. In cancer genome, somatic alternations, including small indels, copy number variations, chromosomal rearrangements [4–6], are also observed differently on prognosis and frequency among patients with different tumor stages or tumor subtypes [7, 8]. It is critical to learn these subgroup specific biomarkers to better identify molecular-based therapies [9, 10]. Recently, single-cell RNA sequencing technology has emerged as a promising genome-wide mRNA expression quantification method in individual cells which identifies cell types by sub-populations of single cells [11]. Cell type specific genes serve as biomakers to characterize sub-population structure and understand disease progression and mechanisms of transcription regulation [12,13]. Overall, because of the discrepancies among samples, patients and single cells, biomarker detection among subgroups is not only reliable and accurate but also biological meaningful.

In this thesis, we designed several machine learning based algorithms for biomarker detection on two types of genomic data that have subgroup structures: DNA Copy Number Variation data and single-cell RNA sequencing data. In this introduction,

we briefly introduce these two datasets, present the current challenge, discuss related methods and lastly propose several structured latent features and multitask learning based algorithms to accurate detect biomarkers among subgroups.

1.1 Copy Number Variations

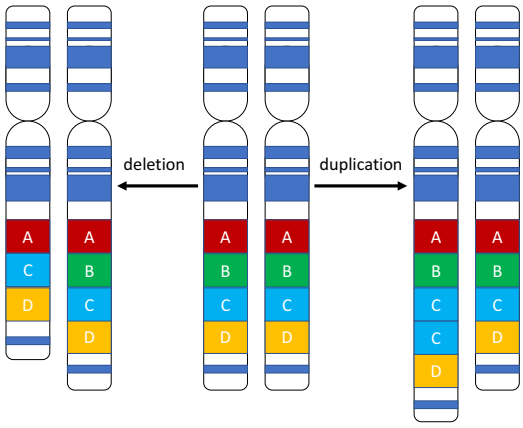


Figure 1.1: **Illustration of Copy Number Variation.** In Normal case, copy number is 2, as shown in the middle figure. The left figure shows a copy number deletion on region B while right figure shows a copy number duplication on region C.

Two copies of each gene are usually presented in a human genome. Variations of this copy number of genes due to large-scale DNA alternation, such as insertion, deletion and duplication (of a large portion of a gene) are called DNA Copy Number Variations (CNV). Figure 1.1 shows an example of copy number deletion and duplication.

CNVs account for a substantial proportion of human genetic variations: they are very common in human genome, affecting more nucleotide content per genome than single-nucleotide polymorphisms (SNPs). Previous studies show that CNV could happen in any region of genome and the percentage of an individual's chromosomes that exhibit CNVs varies from 6% to 19% [14]. It is well known that due to the heterogeneity on population level, human samples from different populations have different genetic variations and show different phenotypes. However, there is limited knowledge of preserved CNV patterns from specific population(s) as the population-specificity of CNVs are not well understood [15].

In our previous study [16], we introduced a tool called SubPatCNV, which is an approximate association pattern mining algorithm under a spatial constraint on the CNV probe features that exhaustively detect large, common CNV patterns across any

sample subsets. We applied it on Hapmap data [17] that contains 270 samples from 4 different populations and the results show that 55% to 70% of the patterns detected by SubPatCNV are population-specific. These highly population specific patterns indicate that, by incorporating populations information as a prior knowledge, it is possible that more accurate of population-differentiation CNVs can be detected and even the evolutionary relations among the populations could be depicted.

CNVs also have been found extremely common in human cancer genome [18,19] and it is believed that CNVs play significant roles in tumorigenesis [14, 20]. Identification and systematic analysis of CNVs can provide important insights into the cellular defects that are cancer causative and suggest potential therapeutic strategies.

In cancer research, one of the main tasks is to identify the CNVs and correlate them with diseases. Due to cancer heterogeneities among the patients [21, 22], even the genomic datasets from the same cancer type patients could be very different. For example, the patient samples grouped by different tumor grades, stages or survival and metastatic status exhibit different CNV patterns. The samples in each or some of the groups might be associated with CNVs that are only discovered from the samples in the same group(s). This is supported by previous study [8] which shows that low and medium grade tumors of bladder cancer generally contain few changes. Thus, it is more biologically interesting to identify CNV patterns for the samples under groups given by prior information to understand disease progression and discover personalized treatment.

1.2 Single Cell RNA-seq

In recent years, single cell RNA sequencing (scRNA-seq) technology has emerged as a promising individual cell genome-wide mRNA expression quantification method [11]. Traditionally, to measure molecular states, bulk RNA-seq methods take average of signal values from millions of cells. These bulk methods overlook the differences in cell population and treat cell population to be homogeneous. However, this could misrepresent signals of interest [23,24] as study [25] show that cell heterogeneity is not only attributed by mutation in tumor studies, but also observed in generically identical cells under the

same environment. To solve this problem, scRNA-seq technology are developed to identify cell heterogeneity. With the identification of cell types and measurement of gene expression distribution, we now have the chance to characterize subpopulation structure, understand disease progression and mechanisms of transcription regulation [25]. Furthermore, RNAs with b may be undetectable in traditional cell-averaging method as they may only express in a small number of cells which related with uncommon or short-time cell types. These RNAs may still play an important role and with the help of scRNA-seq technic with sufficient number of single cells, the measurement become possible [11].

Currently, scRNA-seq protocol contain the following steps: isolation of single cell and RNA, reverse transcription, amplification, library generation and sequencing. However, a variety of noise and bias could be introduced in each step [11]. Besides those issues also exist in bulk RNA-seq, there are some distinct problems in scRNA-seq, both from biological sources, such differences among cells in cell-cycle stage or cell size, and technical/systematic sources, such as capture inefficiency, material degradation, sample contamination, amplification biases, GC content, sequencing depth, etc. For example, due to the tiny amount of cell materials [26], heavily amplification is need before sequencing. PCR are mostly popularly used, however, any bias introduced by PCR could be exponentially amplified. Other amplification technic, such as in vitro transcription [23, 27], which is proposed to avoid PCR sequence bias, also suffer from certain transcribed inefficiency and sequence drop-out. So noise and bias is unavoidable in current amplification step. These potential issues lead to uneven coverage on entire transcript and as result, abundance of zero regions are observed [28]. When multiple single cell populations are available. as shown in Figure 1.2. , there could be significant variances among them due to experiment technical bias. Variance could be more significant when different samples are used for generating each single cell populations as sample to sample confounder could be another issue need to be considered.

1.3 Challenges and Objectives

Currently, there are still many challenges on learning with CNV and scRNA-seq data, especially for biomarker detection among groups.

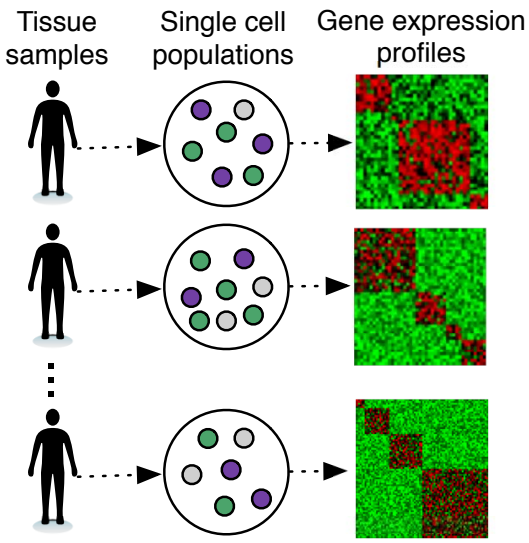


Figure 1.2: **Illustration of single cell RNA-seq data from multiple cell populations.** Each color circle dot in the middle column indicate one single cell, while different color indicate different cell types. Each sample on the left generate many single cells. Cell populations from different samples are normally inconsistent, as shown in gene expression profiles on the right.

1. Many DNA copy-number variations are known to lead to phenotypic variations and pathogenesis. With the increasing number of avalible samples, it is important to consider both the similarity and the heterogeneity among the samples to accurately detect CNV patterns. Existing methods such as FLLat [29] ignore the fact that patient samples with different phenotypes show different frequencies and patterns of CNVs. These methods tend to miss the CNVs specific to subsets of samples. Similarly, on population studies, despite the prevalence of CNVs in human genomes and previous studies showed that the reported CNVs tend to be more common in closely related human populations [14], only limited effort has been made on CNV analysis in the context of human population evolution [15, 30, 31]. Understanding the CNV diversities across populations is a computational challenge because CNV patterns are often present in several related populations and only occur in a subgroup of individuals within each of the population. Previous studies limit on pairwise comparison, so groups specific CNVs, which could be used as Ancestry Informative Markers, are still waited to be explored.

2. Application of transfer learning on CNV analysis across multiple cancer types is promising since CNVs are a hallmark of cancer genomes. However, it is still a challenge to study how CNVs play a role in driving tumorgenic mechanisms that are either universal or specific in different cancer types. Previous studies suggested that many copy number alternations might be found across different cancer types, but most previous computational research work focused on developing models for identifying individual

CNV events from CNV samples of a single cancer type. [32] studied 17 cancer types with at least 40 samples in each cancer type and reported that about 80% somatic copy number alternations found in one cancer type can also be found in pooled analysis excluding that cancer type. These common and type-specific CNVs can potentially reveal unknown cancer mechanisms in the light of cross-cancer-type analysis. However, currently there is no unified mathematical model to simultaneously detect the CNV events common or specific to multiple cancer types from CNV array datasets.

3. As a new technology, there are some challenging in performing scRNA-seq experiment and downstream analysis. Compared to bulk RNA-seq experiments, typical scRNA-seq experiments have more experimental bias and lower read coverage making it more difficult to discover relevant biological variation. Currently, there are existing studies on identifying cell sub-population and further characterizing differential expressed genes on learned cell clusters. Some of the methods directly came from traditional bulk RNA-seq analysis and classical dimension reduction algorithms, such as Principal Component Analysis [33–35], hierarchical clustering [36], t-SNE [37–39], Independent Component Analysis [40] and Multi-dimensional Scaling [41]. Other methods focus on special properties of scRNA-seq data, such as high variance and uneven expressions. For example, SNN-Cliq uses ranking measurement [42] to get reliable results on high dimension data; [43] proposed a special dimension reduction method to handle large amount zeros measurement on scRNA-seq; [44] propose a Latent Dirichlet Allocation based model with latent gene group to measure cell to cell distance. Mixed multiple batch strategy is also proposed [36,45] to reduce the technical variance but with limited improvement. Nevertheless, there is no model designed specifically to learn cell types all together on multiple cell population scRNA-seq data.

1.4 Related methods

To identify biomarkers across many samples, latent feature methods are widely used. Transfer learning technic is also applied when there are several related domains with heterogeneity, such as cross cancer CNV studies or single cell sequencing data with multiple cell populations In this section, we briefly discuss these two related methods.

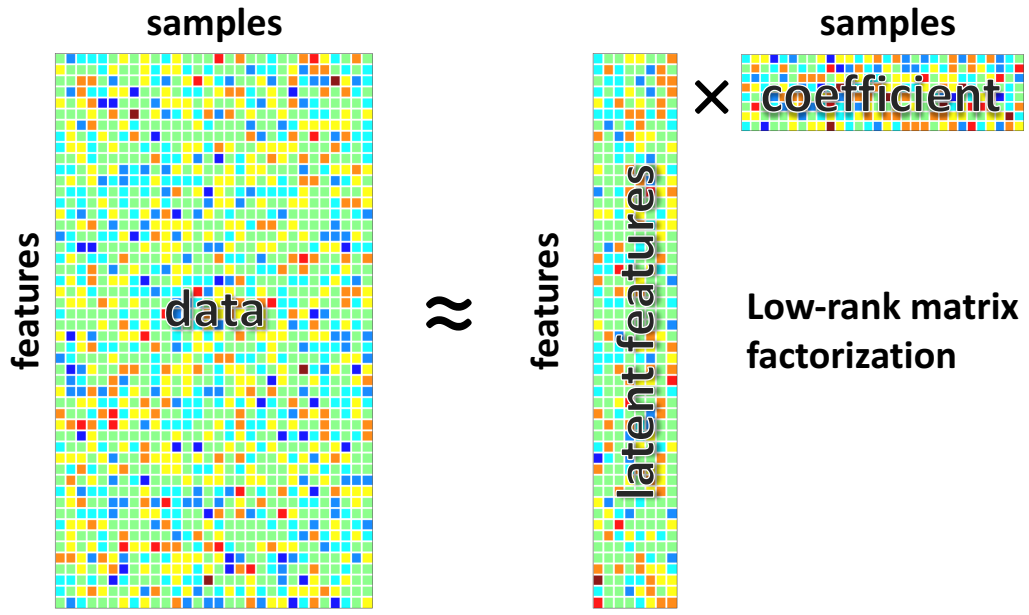


Figure 1.3: **Illustration of latent feature model.** A feature by sample genomic profile matrix can be factorized into latent features and corresponding coefficients with low-rank matrix factorization.

1.4.1 Latent feature learning with low-rank matrix factorization

For multi-sample CNV detection, all samples are analyzed simultaneously in one optimization framework. [46] and [47] proposed to identify the amplification or deletion regions shared across all samples as follows: for N samples with M copy number features, we can solve the following optimization problem:

$$\min_{U \in \mathbb{R}^{M \times N}} \|X - U\|^2 + \lambda \sum_{m=1}^{M-1} \|U_{m+1,\bullet} - U_{m,\bullet}\|,$$

where X is the $M \times N$ CNV profile matrix, U is the de-noised segmentation approximating X and $U_{m,\bullet}$ is the mth row of U. A fast group least-angle regression (LARS) algorithm can be applied to solve the optimization framework approximately to detect shared change-points from the multiple CNV profiles. Since the change-points are detected from all profiles in the framework, it is expected to be more accurate than detecting change-points independently from each CNV profile.

Under the same motivation that CNVs are usually shared by multiple samples, instead of approximating the profile matrix X by a segmentation matrix of the same size, another more advanced modeling is to detect the shared CNVs as latent fused features by low-rank matrix factorization decomposed from X, as shown in Figure 1.3. In this model, each samples is approximated as linear combination of latent features. Another widely used dimensionality reduction method principal component analysis (PCA) can decompose X into orthogonal principle components. The projection of X to a low-dimensional space obtains coefficients of the principle components to preserve the variance. However, practically it is not feasible to interpret the principle components as CNVs since the principle components cannot be explained as CNV patterns without fusing the adjacent features with lasso.

More recently, a Fused Lasso Latent Feature Model ($FLLat$) was proposed by [29] for detecting latent CNV components. Again, for the profile matrix X with N samples and M probes, $FLLat$ decomposes it as a weighted sum of a fixed number of latent feature components, which are smoothed by fused lasso. The corresponding optimization problem for $FLLat$ is

$$\min_{U,V} \sum_{n=1}^{N}\sum_{m=1}^{M}\left(X_{mn} - \sum_{k=1}^{K} \mathrm{U}_{mk}V_{kn}\right)^2 + \lambda_1 \sum_{k=1}^{K}\sum_{m=1}^{M}|U_{mk}| + \lambda_2 \sum_{k=1}^{K}\sum_{m=2}^{M}|U_{mk} - U_{m-1,k}|$$

subject to $\sum_{n=1}^{N} V_{kn}^2 \leq 1$ for each k, where K is the number of latent features and X_{mn} is the log intensity ratio of the mth probe for the nth sample. This model minimizes the sum of the square errors as well as the fused lasso penalties on the latent feature U. It is clear that the model does not assume any structure on the weights of the latent fused lasso components V so each learned latent components are still according to all the samples which are not subgroup specific.

1.4.2 Multitask learning/Transfer learning

Traditional machine learning system works within one domain: it either makes predictions by learned models from training data or direct learns with unlabeled data. However, in real world application, it is often expensive or impossible to collect labels

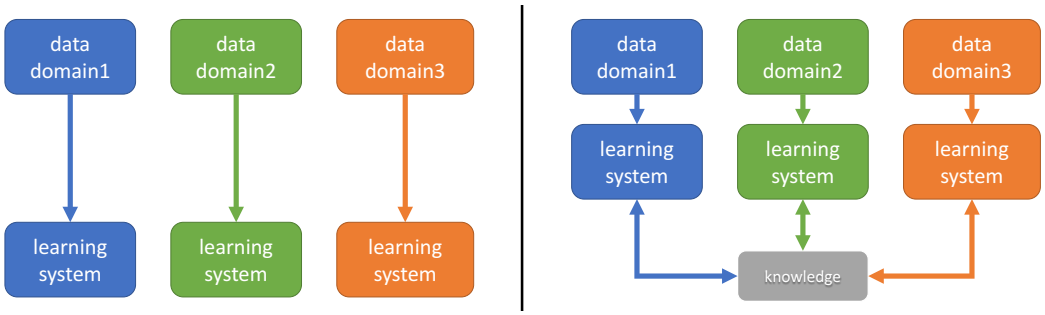


Figure 1.4: **Illustration of Multitask Learning.** On the left, traditional machine learning methods are shown which apply learning technic on each domain individually. On the right, multitask learning uses knowledge as bridge to connect all the domains and improves learning system for each domain.

from certain domain data, but information from similar or related domains are available. Transfer learning, which solve a task, such as classification or clustering in one domain by utilizing information from other domains that may be in a different feature space or follow a different data distribution, can greatly improve the performance of learning.

Commonly, transfer learning are refer to the case when knowledge is transferred from source domains to target domains. Technique that learn task all simultaneously among all the domains are called multitask learning and sometimes is considered to be a special case of transfer learning [48]. In this thesis, we use term "transfer learning" and "multitask learning" interchangeably. The difference between traditional machine learning and multitask learning is shown in Figure 1.4.

As the feature space and data distribution could be different among domains, so a feature selection or feature reduction procedure is needed to extract the common and sharable information among them to minimize domain divergence. By utilizing shared knowledge, classification or regression error in each task could be reduced as well. Currently there are many transfer learning studies: some are designed for improvement only on single target domain [49], some work for all the domains [50,51]. Knowledge transfer combines with sparse feature learning [52], SVM [53], kernel-based method [54] and Procrustes analysis-based method [55] are also developed. [56] proposed a method that extract the discriminative information from labelled data and use it for unsupervised dimensionality reduction. By repeating this procedure, this method iteratively updates the clustering results to get most discriminative subspace and optimal clustering result.

However, this proposed method required some labelled information in source domain to work. In paper [57], a feature reduction method is proposed to minimize the distance between distributions of the data in source and target domains. Even though the feature reduction method do not require label information, it cannot easily extended to multiple domains. Also, when data in each domain are severely suffered by systematic bias, a simple feature selection or reduction is not enough as domains may have no clear shared knowledge. For example, in multiple cell population single cell data, each domains may contain several cell types but assuming the data of certain cell type to be similar across different domains is a hypothesis that could be too strong to be true.

1.5 Contributions

Considering that 1) subgroup structures exist in both CNV data and scRNA-seq data; 2) cross-domain heterogeneities such as human population, cancer types and single cell samples; we proposed several structured latent features and multitask learning based methods as follows:

First, we cooperated prior-known sample relationship and developed a structured latent features based method for high accurate CNV pattern detection on population study. We proposed a tree-guided machine learning algorithm to detect population-differentiation CNVs among populations organized by a phylogenetic tree of human populations. Utilizing the evolutionary relation in the human population tree, the algorithm *treeSGS* discovers sets of CNV markers associated with the branches of the tree such that there exists a subgroup of individuals in each population below the branch exhibiting the preserved CNV patterns from the ancestral population. In the study of 1179 samples from the 11 populations in Hapmap3 and 1000-genome-project data, we validated the accuracy of the algorithm in detecting a list of candidate AIM CNV markers that not only are population-differentiation but also depict the evolutionary relations among the populations.

Then, to study CNVs across different cancer domains, we proposed a Transfer Learning with Fused Lasso model $TLFL$ to detect latent CNV features from CNV datasets of multiple cancer types, in which each cancer type can be regarded as one domain in transfer learning. Common latent CNV features are used as a bridge to transfer

knowledge among different cancer domains along with the domain-specific components for each cancer type to explain the observed CNV datasets. To represent the pattern of CNV events, fused lasso is applied on each latent CNV features to preserve the sparsity and block structure. By using alternating optimization to solve the $TLFL$ model, common latent features and domain specific features could be detected from multiple domains. Compared with a baseline method without using knowledge transfer, $TLFL$ is more robust and identifies more accurate latent CNV components in simulations and experiments on real arrayCGH CNV datasets and SNP genotyping array datasets.

Finally, we applied multitask learning method on single cell RNA sequencing data with multiple cell populations. We introduced a multitask learning method with an embedded feature selection to capture most the differentially expressed genes among cell clusters across all cell populations to achieve better single-cell clustering simultaneously. The key to doing this is the use of multiple single-cell populations available from biological replicates or related samples with significant biological variances such as samples cultured independently or obtained from different patients. We proposed a variance-driven multitask clustering of single-cell RNA-seq data ($scVDMC$) algorithm that utilizes expression patterns of different single-cell populations with shared cell-type markers for better integration. Applied to two real single-cell RNA-seq datasets with several replicates, $scVDMC$ detected more accurate cell populations and known cell markers than pooled clustering and several other recently proposed scRNA-seq clustering methods. $scVDMC$, applied to in-house Recessive Dystrophic Epidermolysis Bullosa (RDEB) scRNA-seq data, revealed several interesting cell types and markers that were previously unknown.

1.6 Outline

The rest of the thesis will be organized into four chapters:

- In Chapter 2, we describe *treeSGS* algorithm which use population tree as prior knowledge to discover population-different CNV patterns.
- Chapter 3 describes $TLFL$ algorithm that identify common and specific CNVs on cross-cancer studies.

- Chapter 4 describes a variance driven multitask clustering method $scVDMC$ on multiple cell population scRNA-seq datasets.
- Finally, we summarized all these algorithms and models and then discussed possible future work in Chapter 5.

Chapter 2

Tree-guided group selection for CNV detection

2.1 Introduction

Two copies of each gene are usually present in a human genome. Variations of this copy number of genes due to larger-scale DNA alternation, such as insertion, deletion and duplication (of a large portion of a gene) are called DNA copy number variants (CNVs). CNVs are very common in human genome, affecting more nucleotide content per genome than single-nucleotide polymorphisms (SNPs). Previous studies showed that CNV could happen in any region of genome and the percentage of an individual's chromosomes that exhibit CNVs varies from 6% to 19% [14].

In the literature [58, 59], there have been intensive studies on the genetic diversity among human populations by SNP association analysis. However, despite the prevalence of CNVs in human genomes and that previous studies showed that the reported CNVs tend to be more common in closely related human populations [14], only limited effort has been made on CNV analysis in the context of human population evolution [15, 30, 31]. A recent study in [15] performed global CNV population stratification on 236 human genomes from seven continental population groups and reported many population-differentiation CNVs by pairwise comparison between the populations.

In this chapter, we propose a tree-guided sparse group selection algorithm (treeSGS) to discover common CNVs from subgroups of individuals across populations organized

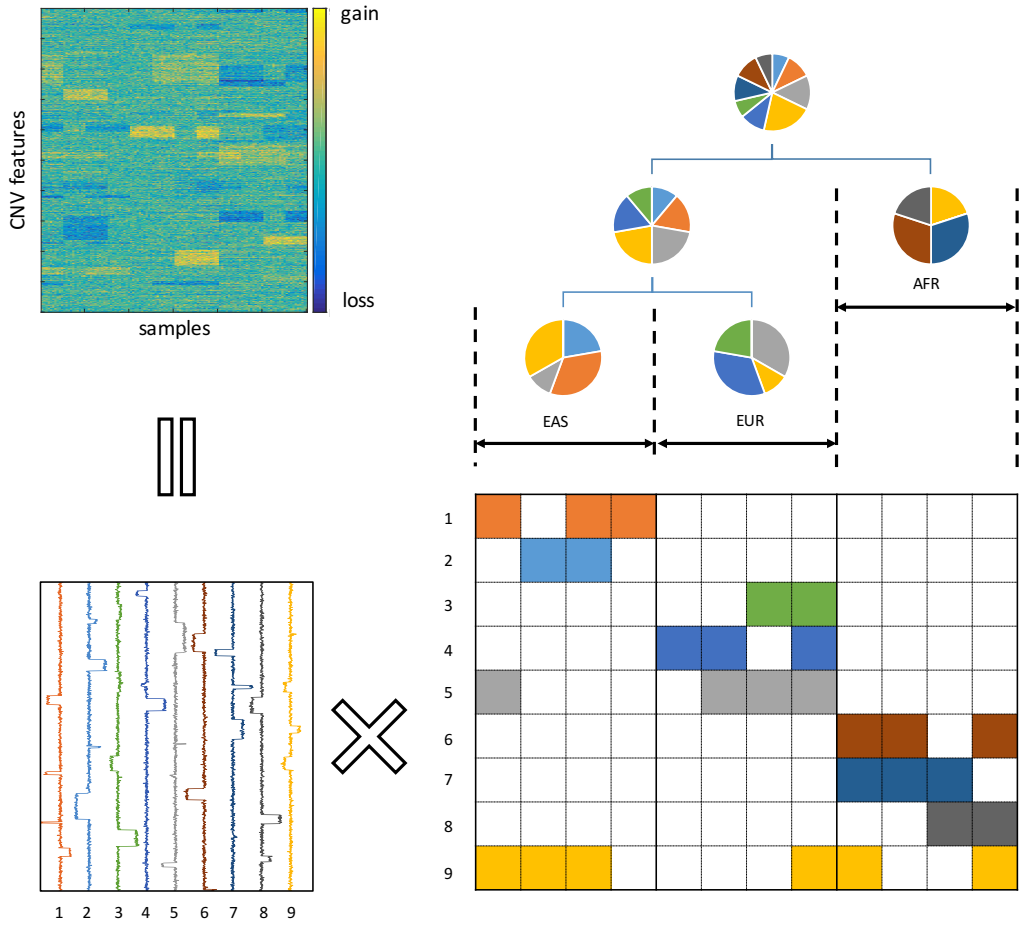


Figure 2.1: **Factorization of CNV genotypes guided by human population tree.** The genotype matrix X (top left) is factorized into latent CNV profiles matrix U (bottom left) and the coefficient matrix V (bottom right). Each column in V indicates a sample. The population tree (top right) shows the hierarchical relation of three super populations, east Asian(EAS), European(EUR) and African(AFR). There are nine latent CNV profiles shown in different colors. Their corresponding coefficients in V are shown in the same color. Each pie chart at a node shows the presence of the CNV profiles under the branch. The coefficients show consistent patterns with the hierarchical structure in the tree.

in the phylogenetic tree of human populations. Based on the human population tree, the focus of the algorithm is to detect CNV profiles representing the collections of CNV

events introduced at each branch of the tree such that there exists a subgroup of individuals in each population below the branch exhibiting the preserved CNV patterns from the ancestral population. By associating CNV signatures with the internal nodes as well as the leaves of the population tree, treeSGS algorithm incorporates the evolutionary relations among the populations to recover the history of CNVs.

Figure 2.1 shows a toy example with 12 samples from east Asian (EAS), European (EUR) and African (AFR) organized in a phylogenetic tree, where EAS and EUR populations more recently differentiated from each other than AFR. The genotype data of the 12 samples can be factorized into CNV profiles and coefficients such that each sample is a linear combination of the latent CNV profiles weighted by the coefficients. The non-zero coefficients shows the selection of a CNV profile in a sample. The light grey profile is selected by some individuals from EAS and EUR and the yellow profile is selected by individual from all three populations while the other seven profiles are specific to one of EAS, EUR and AFR populations. The organization of the coefficients is consistent with the tree since each CNV profile corresponds to population groups organized by the nodes in the tree, e.g. the light grey profile corresponds to the parent node of EAS and EUR and the yellow profile corresponds to the root node. The yellow profile represents the earliest CNV events in this example which thus occurs in all the three populations. Detecting CNVs in the context of a tree among populations is more appropriate setting than pairwise comparison between populations [15].

With the treeSGS algorithm, we studied the 1179 samples from the 11 populations in Hapmap3 CNV genotype data based on the human population tree built with SNPs shown in Figure 2.2. In the experiments, treeSGS more accurately identifies CNV signatures of each population and the collection of populations in each branch of the human population tree than several other methods. We validated each CNV profile and their occurrence in the populations by their consistency among the family trios in Hapmap3 samples and the SNP characterizations of the CNV regions by populations. We also further compared the other population-differentiation CNV signatures reported in other recent studies with the detected CNV signatures by treeSGS to show that the CNV signatures are more accurate annotations describing the differentiation between groups of populations.

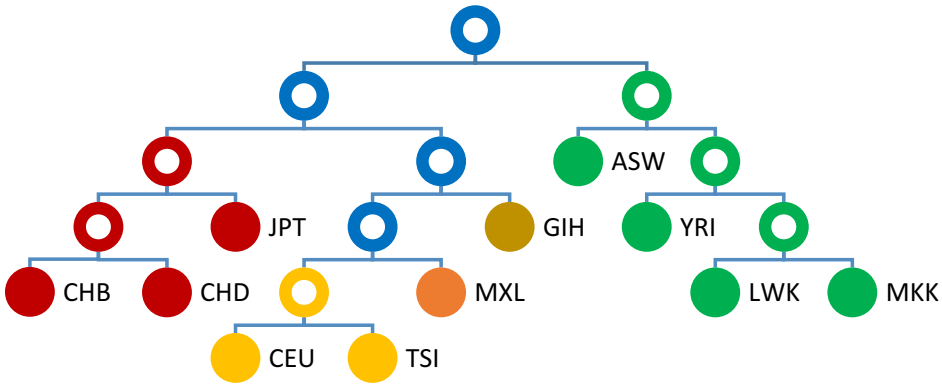


Figure 2.2: **Phylogenetic tree of 11 human populations.** 3 East Asian populations (red), CHB: Han Chinese in Beijing, China; CHD: Chinese in Metropolitan Denver, Colorado; JPT: Japanese in Tokyo, Japan, 4 African populations (green), ASW:African ancestry in Southwest USA; LWK: Luhya in Webuye, Kenya; MKK: Maasai in Kinyawa, Kenya; YRI: Yoruba in Ibadan, Nigeria and 2 European populations (yellow), CEU: Utah residents with Northern and Western European ancestry from the CEPH collection; TSI: Toscani in Italia. MXL:Mexican ancestry in Los Angeles, California and GIH:Gujarati Indians in Houston, Texas. are grouped with the European populations.

2.2 Methods

In this section, we first present the model, and then describe treeSGS algorithm and each major profile of the algorithm.

2.2.1 Tree-guided sparse group selection model

Let $X \in \mathbb{R}^{m \times n}$ be the CNV feature by sample matrix, where m is the number of CNV features and n is the number of samples. Let $U \in \mathbb{R}^{m \times K}$ be the profile matrix and $V \in \mathbb{R}^{K \times n}$ be the coefficient matrix, where UV is a factorization of X and K is the number of latent CNV profiles.

For a given binary population tree $Tree(\mathbf{g})$, let $\mathbf{g} = \{g_1, g_2, \ldots, g_{2L-1}\}$ represent the $2L-1$ nodes in the tree, where $\{g_1, g_2, \ldots, g_L\}$ are the leaf nodes (populations) and $\{g_{L+1}, g_{L+2}, \ldots, g_{2L-1}\}$ are the internal nodes. We define a function $F(g_i)$ to output the set of samples in which a sample belongs to the population g_i when g_i is a leaf otherwise

a population that is a descendant node of g_i,

$$F(g_i) = \begin{cases} \{p|p \in g_i\} & \text{if } i \leq L \\ \cup_j \{F(g_j)|desc(g_i, g_j)\} & \text{otherwise,} \end{cases}$$

where p denotes an individual sample and $desc(x, y)$ denotes y is a descendant node of x. In the first case, g_i is a population and $F(g_i)$ is the set of all the individuals in the population g_i. In the second case, g_i is an internal node and each g_j denotes a leaf descendant of g_i. $F(g_i)$ is the union of all the individuals in each population g_j.

We next define a split of $Tree(\mathbf{g})$ to partition the populations with respect to a certain CNV profiles k as $split(\mathbf{g}, k)$. $split(\mathbf{g}, k)$ is a subset of g denoted as $\{g_{k_1}, ..., ...g_{k_z}\}$ where $z \leq L$ such that the following two conditions are satisfied,

$$\begin{aligned} &(1) F(g_{k_i}) \cap F(g_{k_j}) = \varnothing, \ \forall g_{k_i}, g_{k_j} \in split(\mathbf{g}, k) \\ &(2) \cup_{g_{k_i} \in split(\mathbf{g},k)} F(g_{k_i}) = \cup_{l=1}^{L} F(g_l) \end{aligned} \tag{2.1}$$

The two conditions guarantee that $split(\mathbf{g}, k)$ denotes a set of branches in the tree that exactly partition the n samples by the partition of the leaf nodes (populations) in each branch.

Based on the above definitions, the regularization framework of treeSGS is defined as follows,

$$\begin{aligned} \underset{U,V}{\text{minimize}} \quad & ||X - UV||_F^2 + \lambda \sum_k |U_{\bullet,k}|_1 \\ \text{subject to} \quad & V \succeq 0 \\ & V_{k,\bullet} \times V_{k,\bullet}^T = 1, \quad k = 1, \ldots, K \\ & V_{k,F(g_{k_i})} \times b_{k,F(g_{k_i})} = 0, \\ & \quad \forall g_{k_i} \in split(\mathbf{g}, k) \text{ for } \ k = 1, \ldots, K, \end{aligned} \tag{2.2}$$

where $|U_{\bullet,k}|_1$ is the $L1$ norm on $U_{\bullet,k}$ ($|U_{\bullet,k}|_1 = \sum_i |U_{i,k}|$) for sparse CNV signals; $V_{k,F(g_{k_i})}$ is a sub-vector of $V_{k,\bullet}$ indexed with $F(g_{k_i})$ and $b_{k,F(g_{k_i})}$ is a corresponding binary indicator. If $b_{k,F(g_{k_i})} = 1$, then $V_{k,F(g_{k_i})}$ will be a 0 vector. $b_{k,F(g_{k_i})}$ acts as a selection indicator which 0 means the corresponding group $F(g_{k_i})$ is selected in vector $V_{k,\bullet}$. We will discuss how to choose $b_{k,F(g_{k_i})}$ in the next section. The treeSGS model is the tree-guided version of the sparse group selection model in [60] (formulation given in the supplementary document).

2.2.2 TreeSGL algorithm

The main framework of treeSGS algorithm is shown in Algorithm 5. The algorithm alternatively optimizes the CNV profiles U and the coefficients V until convergence.

Algorithm 1 treeSGS algorithm

1: Input: $X, Tree(\mathbf{g}), \tau, \theta, \lambda, K$
2: U = PCA(X,K)
3: **repeat**
4: **repeat**
5: **for** $k = 1, 2, \ldots, K$ **do**
6: compute $w^{(k)}$ by eqn 2.6
7: $split(\mathbf{g}, k) = EntropyCut(Tree(\mathbf{g}), w^{(k)}, \tau)$
8: $b_{k,\bullet} = SparseGrpSelect(w^{(k)}, split(\mathbf{g}, k), \theta)$
9: Solve $V_{k,\bullet}$ in eqn 4.2
10: **end for**
11: **until** V converge
12: Solve U in eqn 4.3
13: **until** U and V converge
14: **return** U and V

TreeSGL algorithm takes the CNV sample data X, the population tree $Tree(g)$ and four hyper-parameters as inputs. The four hyper-parameters are K: the total number of CNV profiles, λ: the weight on the LASSO regularizer and τ: the cutoff for computing $split(\mathbf{g}, k)$, θ: the weight ratio for group selection, which will be explained in the next sections. At line 2, the CNV profiles U are initialized by the first K principle profiles of X. The repeat-until loop between line 3-13 iteratively solve V or U with the other fixed. The repeat-until loop between line 4-11 iteratively solve V with the sparse group selection computed by the for-end loop between line 5-10.

solve U:

At line 12, when V is fixed to solve U, the subproblem to optimize is given as,

$$\underset{U}{\text{minimize}} \quad ||X - UV||_F^2 + \lambda \sum_k |U_{\bullet,k}|_1, \tag{2.3}$$

In this objective function, $\lambda > 0$ weights the LASSO terms. This function is solved column-wisely on U as the standard $L1$ LASSO linear regression problem [61] by multi-task extension [62] with a fast convergence rate of $O(\frac{1}{\epsilon})$ where ϵ is a desired accuracy,

and per-iteration time complexity $O(K^2m)$. If treeSGS is directly applied to arrayCGH or genotyping array probes, it is also possible to add a fused LASSO penalty as the graph-guided fusion penalty algorithm [62],

$$\lambda_1 \sum_k \sum_{i=1}^{m} |U_{i,k}| + \lambda_2 \sum_k \sum_{i=2}^{m} |U_{i,k} - U_{i-1,k}|, \tag{2.4}$$

where the LASSO and fused LASSO penalties will introduce sparse segmented CNV signals in the profiles.

solve V:

At line 9, when U is fixed, V can be solved column-wisely. For each k, we have the following subproblem,

$$\begin{aligned} \underset{V_{k,\bullet}}{\text{minimize}} \quad & ||\hat{X_k} - U_{\bullet,k}V_{k,\bullet}||_F^2 \\ \text{subject to} \quad & V_{k,\bullet} \succeq 0 \\ & V_{k,\bullet} \times V_{k,\bullet}^T = 1 \\ & V_{k,F(g_{k_i})} \times b_{k,F(g_{k_i})} = 0,\ \forall g_{k_i} \in split(\mathbf{g}, k), \end{aligned} \tag{2.5}$$

where $\hat{X_k} = X - U_{\bullet,\neq k}V_{\neq k,\bullet}$ is the residue matrix of X after removing the contributions from the other profiles. To solve each column of V, we will need to obtain the tree split by grouping the populations and then select the groups with $b_{k,F(g_{k_i})}$, which are described in the following section.

2.2.3 Tree splitting and sparse group selection

The core idea of introducing the population tree is to provide a strategy of grouping the populations intelligently such that the discovered CNV profiles can depict the relations among the populations in the tree. To achieve the goal, we first define vector $w^{(k)}$ for each CNV profile to denote the importance of the CNV profile k to the construction error, in which each element of $w^{(k)}$ corresponds to each sample's contribution.

$$w^{(k)} = \frac{U_{\bullet,k}^T \hat{X_k}}{{U_{\bullet,k}}^T U_{\bullet,k}}. \tag{2.6}$$

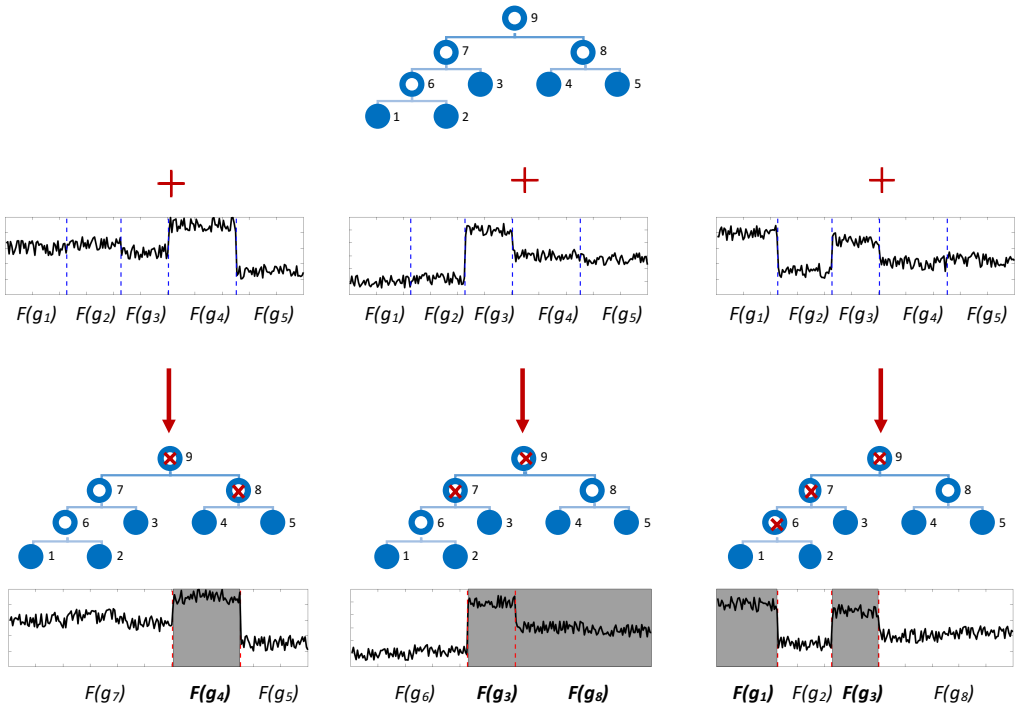


Figure 2.3: **Illustration of tree-based split of populations by CNV profiles.** At the top, a population tree $Tree(\mathbf{g})$ with five populations (solid blue circles) and four ancestral nodes (hollow blue circles) is shown. Below, three series of $w^{(k)}$ denoting the contribution of each sample to the CNV profile are plotted as the black curves. The vertical blue dash line on $w^{(k)}$ plots indicates the separations of the individuals from the five populations at the leaf nodes while the red lines in the bottom plots indicates the groups found by *EntropyCut*. The red "×" marks nodes for the tree split. The grey area are the selection of highly weighted group(s) based on *SparseGrpSelect*.

In the above equation, $U_{\bullet,k}^T \hat{X}_k$ returns the inner product similarity between $U_{\bullet,k}^T$ and each column of $\hat{X}_k$. The higher the similar, the more useful $U_{\bullet,k}$ to the reconstruction of $\hat{X}_k$ $w^{(k)}$ differentiates the populations into related vs non-related groups with respect to the CNV profile k. To illustrate how the populations can be grouped in the true structure, three examples of $w^{(k)}$ are shown in Figure 2.3. In the left example, the five populations can be grouped as (1, 2, 3), (4) and (5), and thus, the tree splitting will be introduced at the internal node 8 and 9; in the middle example, the populations can be grouped as (1,2), (3) and (4,5), and thus, the splitting will be at node 7 and node 9. In the right example, the groups are (1), (2), (3) and (4,5), and thus, the splitting will be

at node 9, node 7 and node 6. After the splitting, $SparseGrpSelect()$ is applied to select the top groups representing at least θ percent of all the contribution to reconstruction. In the three examples, $\{F(g_4)\}$, $\{F(g_3), F(g_8)\}$ and $\{F(g_1), F(g_3)\}$ are selected from left to right. Accordingly, the corresponding $b_{k,F(g_{k_i})}$ are set to be 0 to select the variables for learning.

Algorithm 2 *EntropyCut*

Input: $Tree(\mathbf{g}), w^{(k)}, \tau$
for each parent-children triple $\{p, l, r\}$ in $Tree(\mathbf{g})$ **do**
    use eqn 2.7 or 2.8 for density estimation.
    calculate $entropy(w^{(k)}_{F(g_p)})$, $entropy(w^{(k)}_{F(g_l)})$ and $entropy(w^{(k)}_{F(g_r)})$ by eqn 2.9.
    calculate $InfoGain(p, l, r)$ by eqn 2.10.
    if $InfoGain(\{p, l, r\}) \geq \tau$ **then**
        split_node[p]=true
    else
        split_node[p]=false
    end if
end for
for each $g \in \{g_{L+1} \dots g_{2L-1}\}$ and split_mark[g] **do**
    split_node[$ancestor(g)$]=true
end for
$split(\mathbf{g}, k) = \{g_{root}\}$
for each g in breadth-first traversal of the nodes **do**
    if $g \in \{g_{L+1} \dots g_{2L-1}\}$ and split_node[g] **then**
        $split(\mathbf{g}, k) = split(\mathbf{g}, k)$ - $\{g\}$
        $split(\mathbf{g}, k) = split(\mathbf{g}, k) \cup \{left(g), right(g)\}$
    end if
end for
return $split(\mathbf{g}, k)$

Tree splitting by $EntropyCut()$**:**

At line 7 in Algorithm 5, *EntropyCut* returns the tree partition $split(\mathbf{g}, k)$ for each CNV profile k given the sample reconstruction vector $w^{(k)}$ and the tree structure $Tree(\mathbf{g})$. For every internal node $p \in \{g_{L+1} \dots g_{2L-1}\}$ and the two children node $l = left(p)$ and $r = right(p)$ as a triple $\{p, l, r\}$, we calculate their corresponding entropy $entropy(w^{(k)}_{F(g_p)})$, $entropy(w^{(k)}_{F(g_l)})$ and $entropy(w^{(k)}_{F(g_r)})$. The procedure *EntropyCut* is

described in Algorithm 2. The algorithm applies Information Gain to calculate if splitting the samples into two groups under a particular branch will increase the overall information gain significantly.

First at line 3 in the algorithm, we obtain a density estimation of each $w^{(k)}$ with either histogram and Gaussian kernel estimator. Denote the elements in the vector $w^{(k)}_{F(g)} = \{x^t\}_{t=1,2,,\ldots,M}$ assuming IID drawing from $p(x)$. For the histogram of bin size h,

$$\hat{p}_{hist}(x) = \frac{\#\ \{\text{of } x^t \text{ in the same bin as x}}{Mh}\}. \tag{2.7}$$

For Gaussian Kernel estimator with bandwidth $\hat{h}$,

$$\hat{p}(x)_{gk} = \frac{1}{Mh}\sum_{t=1}^{M}\frac{1}{\sqrt{2\pi}}\exp\big[-\frac{(x-x^t)^2}{2\hat{h}^2}\big]. \tag{2.8}$$

Based on the density estimation $\hat{p}(x)$, entropy can be calculated as

$$entropy(X) = -\sum_j \hat{p}(x_j)\log\hat{p}(x_j), \tag{2.9}$$

where x_j is evenly sampled with in the input range. Note that the sampling points in equation 2.7 or 2.8 for calculating entropy of note p, l and r are all within the range of $w^{(k)}_{F(g_p)}$ since $w^{(k)}_{F(g_l)}$ and $w^{(k)}_{F(g_r)}$ are both sub-vectors of $w^{(k)}_{F(g_p)}$.

At line 5, the information gain of splitting at a triple $\{p,l,r\}$ is calculated as

$$\begin{aligned} InfoGain(\{p,l,r\}) = {} & entropy(w^{(k)}_{F(g_p)}) \\ & - \frac{|g_l|}{|g_p|}entropy(w^{(k)}_{F(g_l)}) \\ & - \frac{|g_r|}{|g_p|}entropy(w^{(k)}_{F(g_r)}) \end{aligned} \tag{2.10}$$

At line 6-10, the information gain is compared with the threshold. If $InfoGain(\{p,l,r\}) \geq \tau$, $w^{(k)}_{F(g_l)}$ and $w^{(k)}_{F(g_r)}$ form two distinct distributions and thus $w^{(k)}_{F(g_p)}$ needs to split; otherwise, if $InfoGain(\{p,l,r\}) < \tau$, $w^{(k)}_{F(g_l)}$ and $w^{(k)}_{F(g_r)}$ are similar and there is no need to split.

After all the triples are checked and the internal nodes are marked as split or non-split, we mark all the ancestor nodes of splitting nodes as split (line 12-14 in Algorithm 2) since it is necessary to split all the parent groups before splitting a more specific

group. After this step, a breadth-first traversal of the tree is applied to choose the most specific splitting nodes for generating the partition of the populations at line 15-21. The time complexity of *EntropyCut* is $O(NL + L\log L)$.

Sparse group selection with $SparseGrpSelect()$**:**

Algorithm 3 selects the top groups from $split(\mathbf{g}, k)$. First, in the for-loop between line 2-3, the normalized group weight for each $w^{(k)}$ are calculated as below:

$$h_{k_i} = \frac{||w^{(k)}_{g_{k_i}}||}{\sqrt{|g_{k_i}|}}, \quad g_{k_i} \in split(\mathbf{g}, k).$$

The selection indicator variable $b^{(k)}$=1 for initialization. Next, the normalized group weights are sorted in descending order at line 5. The top groups accounts for at least θ of total group weights are selected and their corresponding binary indicator $b^{(k)}$ is set to be 0 in the repeat-until-loop at line 7-9. The time complexity of the procedure is $O(N + L\log L)$.

Algorithm 3 *SparseGrpSelect*

1: Input: $w^{(k)}, split(\mathbf{g}, k), \theta$
2: **for** every $g_{k_i} \in split(\mathbf{g}, k)$ **do**
3: $\quad h_{k_i} = \frac{||w^{(k)}_{g_{k_i}}||}{\sqrt{|g_{k_i}|}}$, $b_{k,F(g_{k_i})} = 1$
4: **end for**
5: Sort $h_{k_i},\ i = 1, 2, \ldots$ in descending order as $h_{\hat{k_i}},\ i = 1, 2, \ldots$
6: $l = 1$
7: **repeat**
8: $\quad l = l + 1$, $b_{k,F(g_{\hat{k}_l})} = 0$
9: **until** $\frac{\sum_{i=1}^{l} h_{\hat{k_i}}}{\sum_i h_{k_i}} > \theta$
10: **return** $b_{k,\bullet}$

2.2.4 Related work

Tree-SGL is based on the Sparse Group Selection on LASSO (SGL) [60], which don't utilize tree structure but instead only works on non-overlapping groups for cancer CNV data analysis. SGL performs the same group selection without the tree split procedure as

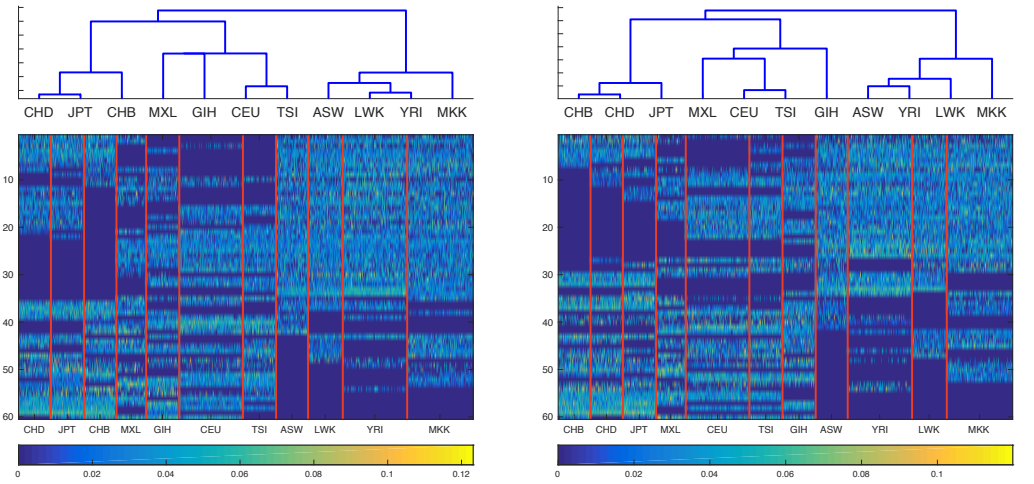
treeSGS. Therefore, SGL will model the 11 population simply as 11 groups for selection ignoring their relations. The complete description of the SGL model can be found in the supplementary document. Another alternative approach to introduce tree structures among variables is tree-guided group LASSO [62]. Tree-guided group LASSO models a path in the tree structure as a group and the coefficients in the same group are smoothed by 2-norm. Since the paths in a tree overlaps, the tree-guided group LASSO problem is more difficult to solve. We adopted the tree-guided group LASSO for our problem by coupling the factorization term with the group LASSO penalty on each path from each leaf to root. The complete formula is presented in the supplementary document and we applied tree_LeastR function in the SLEP package to implement the alternative model (SLEP-GL) [63].

2.3 Experiments

In the experiments, we applied treeSGS method on Hapmap phase 3 CNV genotype data [17,64], which contains 1179 samples and 841 autosome CNVs from 11 populations. The original data are coded with integers from 0 to 5 representing copy numbers and we subtract 2 for the 2 copies for normal. After the transformation, value of -1 represents a heterozygous deletion, and -2 represents a homozygous deletion, and positive integers indicate the additional duplications.

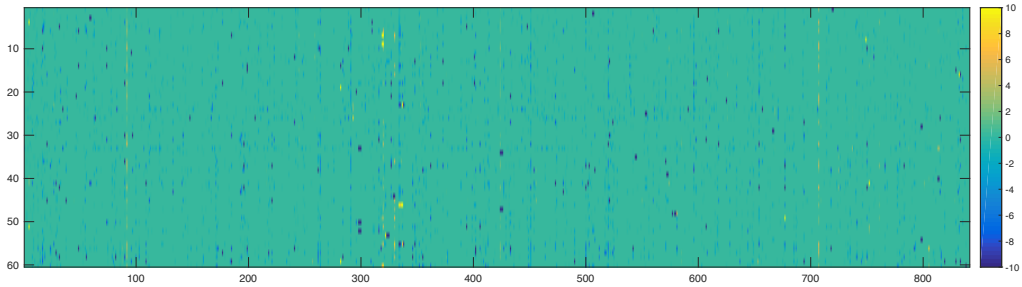
We also obtain all SNP genotypes on the same samples [64] and used SNPphylo tool [65] to generate a phylogenetic tree with sample-wise relationships shown in figure 2.2. The full phylogenetic tree organizing all the samples can be found in the supplementary figures.

TreeSGL is compared with SGL and tree-builded group LASSO methods in the experiments. We evaluated how well each method detect group-specific CNVs that are consistent with the population tree, the SNP data and the family trio annotations in the Hapmap3 samples. In addition, we also collected population-differentiation CNVs detected from different samples from two other studies for further validation of the group-specific CNVs detected by treeSGS [15, 66].



(a) Coefficients (SGL)

(b) Coefficients (treeSGS)



(c) CNV genotype profile (treeSGS)

Figure 2.4: **Visualization of CNV profiles and coefficients by populations.** The coefficient matrices V computed by SGL and treeSGS (Gaussian Kernel Estimator with $\tau = 0.0001$) are shown in (a) and (b), respectively. The CNV profiles learned by treeSGS is shown in (c). Hierarchical clustering is applied to cluster the populations by the mean of V across the samples in each populations.

2.3.1 Interpreting CNV profiles and coefficients

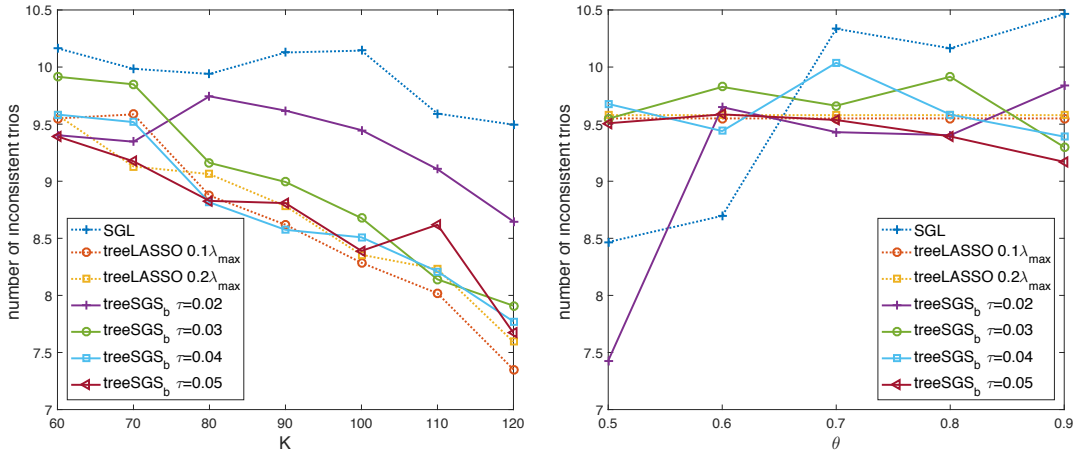
Figure 2.4 shows the visualization of the factorization of Hapmap3 CNV data organized by populations. The original CNV data X (841 by 1179) is factorized into V, the profile matrix of 60 profiles (841 by 60) and U, the coefficient matrix (60 by 1179). After the factorization, we represent each sample by the 60 CNV coefficients as features to cluster the 1179 samples to construct a population tree for comparison with the known population tree. Figure 2.4(a) and (b) show the coefficient matrix by SGL and treeSGS.

$\lambda = 0.8$ was chosen for both treeSGS and SGS for better visualization. Other choices of λ leads to similar patterns. Hierarchical clustering of the coefficient matrix is also shown. Without using the tree structure, SGL generate highly inconsistent population relations compared with the "ground truth" tree structure (Figure 2.2) in clustering the east Asian groups (CHB, CHD, JPT) and the extended European group (CEU, TSI, MXL and GIH). TreeSGS use tree structure as guidance and reproduced the population tree with the sparse coefficients except the African popultions. Note that there is weak consensus on the hierarchical clustering of the African populations due to the individual diversity and longer history of the populations. Figure 2.4(c) shows the 60 profiles. Each non-zero entry represent a deletion/insertion the 841 loci. The CNVs captured in the same profile indicate possible the same origin or similar evolutionary trace since the CNVs co-occur in many samples.

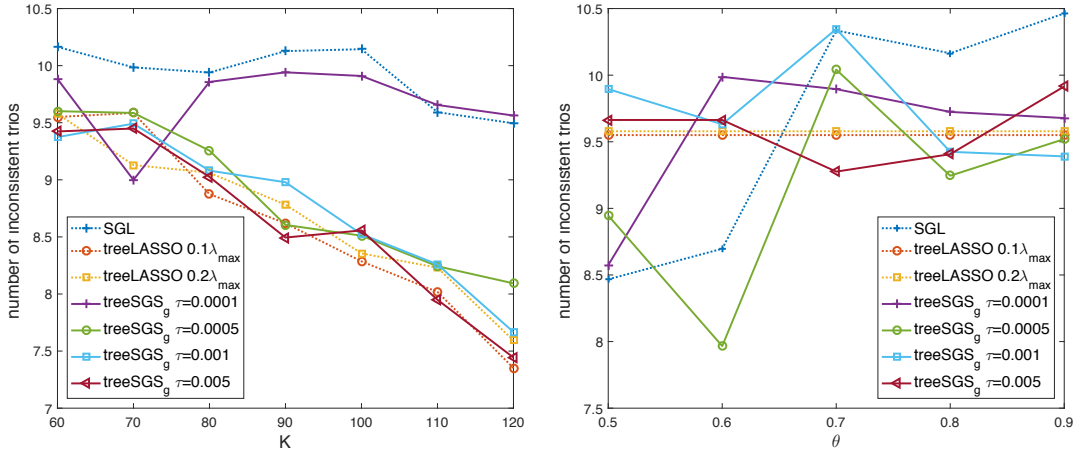
2.3.2 Validating CNV genotypes by family trios

We validated the CNV genotypes by the 155 father-mother-child trios in Hapmap3 data in 5 populations (10 in ASW, 44 in CEU, 28 in MKK, 23 in MXL and 50 in YRI). With each CNV profile representing a CNV genotype, its corresponding coefficients on the samples classify the samples into two groups as with/without the genotype. A trio is considered as likely inconsistent with the CNV genotype if either 1) the child has the genotype but neither of the parents have or 2) the child has not the genotype but both parents have. By dividing the samples in the same population as the trio by the quadratic mean of the coefficients of each CNV genotype, we report the average number of inconsistent trios in the CNV genotypes in Figure 2.5. TreeSGL, SGL and SLEP-GL are tested under different choices of the number of profiles K and other hyper-parameters. TreeSGL was applied with both histogram density estimation (Figure 2.5a) and Gaussian kernel density estimation (Figure 2.5b)

In Figure 2.5, the general trend is that as K increases the number of inconsistent trios gets lower as expected because more CNV profiles could capture more low frequency CNVs that are often more consistent among family. SGL performed worst among all the three methods under a sparse solution with no information from the population tree for grouping. TreeSGL improves the results of SGL by the population tree information. The hyper-parameter θ controls the group selection ratio on SGL and treeSGS methods.



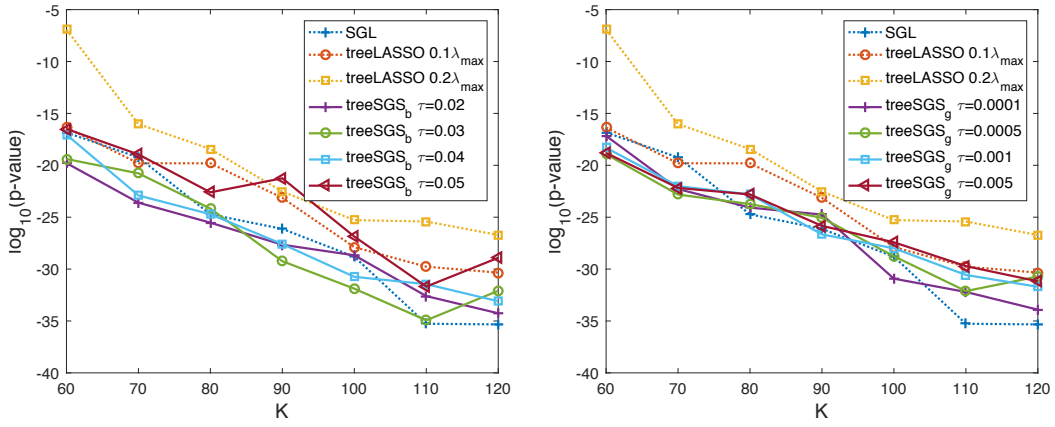
(a) Histogram estimator



(b) Gaussian kernel estimator

Figure 2.5: **Average number of inconsistent trios in CNV genotypes.** treeSGS$_b$ and treeSGS$_g$ denote treeSGS applied with histogram (a) or Gaussian Kernel Estimator for density estimation (b) respectively. The plots on the left show the results of varying the number of CNV profiles K with fixed $\lambda = 0.8$ for SGL and treeSGS. The plots on the right show the results of varying the λ with fixed $K = 60$.

The right plots show that treeSGS is consistently better than SGL method when θ is moderate or large indicating that the reasonably combined groups under tree branches are selected by treeSGS. When the splitting threshold τ gets larger, treeSGS generates denser results and eventually, the performance is comparable or slightly better than SLEP-GL. The results demonstrate that treeSGS provides a combined advantages of



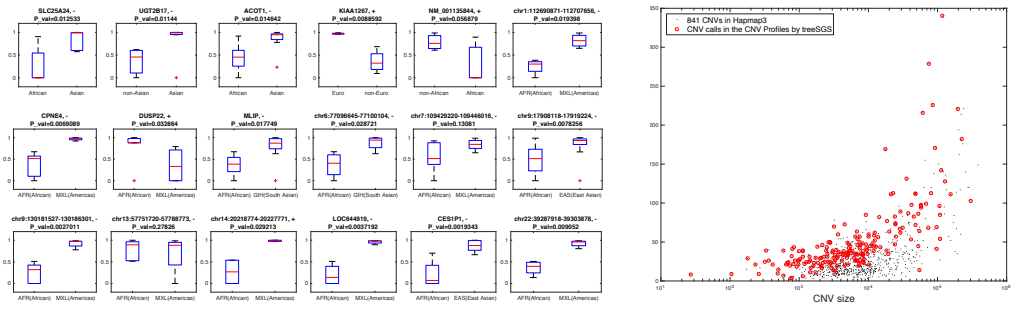
(a) Histogram estimator (b) Gaussian kernel estimator

Figure 2.6: **Comparing CNV events meaningfulness by p-value on classification accuracy of different methods under different K.** log_{10} of p-value is reported. SLEP-GL method used 10% and 20% of maximum penalty parameter. TBL_b and TBL_g indicate TBL method with histogram and Gaussian Kernel Estimator for density estimation respectively. Both SGL and TBL use $\theta = 0.8$.

sparsity and higher sensitivity.

2.3.3 Cooccurrence of CNV and SNP genotypes

To validate whether the CNV genotypes cooccur with some SNP genotypes at the same loci across the samples, we obtained the matched SNP genotype data for the same samples [64] for comparison. For each latent CNV profiles (columns of matrix U) detected by every method, we chose the top 10 most significant CNV regions and obtain the reported SNP that overlapped with the CNV regions. In corresponding sample coefficients (rows of matrix V), samples from the selected populations are divided into the two groups with/without the CNV genotypes. Using the selected SNPs as features, we run leave-one-out cross-validation with polynomial kernel Support Vector Machine classifier to classify the two groups. The experiment is conducted for each of the K CNV genotypes to compute the classification accuracies. To evaluate the classification accuracy, we also repeat the classification using the same number of random consecutive SNPs to obtain the random classification accuracy of the two sample groups with/without the CNV genotype. The SNP features are selected from random regions containing the



(a) Comparison with known population-differentiation CNVs

(b) Comparison with DGV database

Figure 2.7: **Comparison of CNV genotype callings with reported population-differentiation CNVs from literature.** (a) Comparison with 18 known population specific CNV regions from [15,66]. For each gene, bar plot shows the mean and standard deviation of the sample ratios of the over-represented populations across the groups associated with the CNV profiles overlapping the region. Each region is labeled by the gene of interest in the region or the actual genomic coordinates. (b) Comparison with DGV database. Each black dot represents a CNV in Hapmap3 data. The x-axis is the size of the CNVs and y-axis is the number of DGV CNV calls that overlap with the CNV. The red circles denote the CNV calling made by treeSGS for population groups.

same number of SNPs as the background. We repeat the random experiment 100 times to obtain the average accuracy for each of the K CNV genotype. We applied paired t-test between the K accuracies and report the log-p-values in Figure 2.6. SLEP-GL performed worst among the three methods in this measure. This is understandable because SLEP-GL do not encourage population selection such that there is potentially higher false-positive rate in the CNV genotype callings and thus, the groups with/without the CNV genotypes are not supported by the SNPs in the same region. On the contrary, SGL tends to only detect CNV genotypes within a population such that the CNV genotypes callings are better supported by the SNPs by losing sensitivity among the population groups.

In Figure 2.6, varying τ leads to some variation of the classification performance by the treeSGS-detected CNV genotypes. Overall, larger τ leads to worse performance since the density is similar as SLEP-GL's results. For moderate and small τ, treeSGS method performed similarly or better than SGL. Interesting, it is not true that the more sparser the coefficients, the better the classification results suggesting that the treeSGS

make correct CNV callings in the selected population groups. The scatter plot of the actual classification accuracy using the SNP features between treeSGS and the random background is also shown in supplementary document. Generally, the classification accuracy are about 90% on average.

2.3.4 Comparison to known population-differentiation CNVs

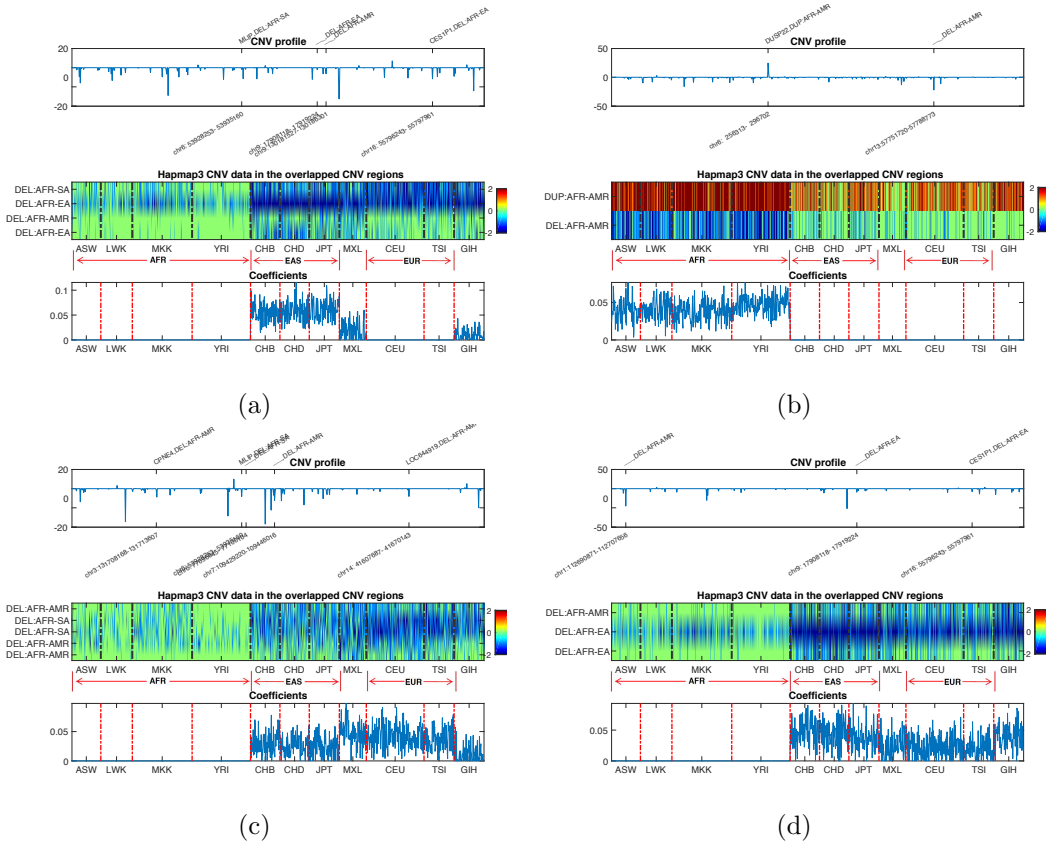


Figure 2.8: **Examples of improved annotation of population-differentiation CNVs.** Four CNV profiles are illustrated. In each example, the top plot shows the overlaps between a CNV profile and the known population-differentiation CNVs with the overlapping regions marked in red. The middle plot shows the original Hapmap3 CNV data in the overlapping regions across all the samples organized by populations. The bottom plot shows the coefficients of the overlapped CNV profile. In all the plots, the populations are separated by the black column bars.

We also compared the detected population(-group)-specific CNVs with two other

studies on cross-population CNV analysis [15, 66]. The study in [66] reported 30 CNV regions (genes) that are population differentiated among CEU, CHB+JPT, MKK and YRI based on the analysis of SNP array data of 487 samples. There are 6 regions overlapped with the Hapmap 3 CNV call regions. The study in [15] performed global CNV population stratification on 236 human genomes from seven continental population groups. Four of the populations are roughly matched with Hapmap3 populations including AFR(African), EAS(East Asian), GIH(South Asian) and MXL(Americas). There are 14 CNV regions overlapped between the reported extreme stratification CNVs ($V_{st} > 0.5$) and Hapmap3 CNVs. In both studies, one of the overlapping regions occurs in very few Hapmap3 samples and thus was removed from the analysis, which leaves 18 population-differentiation regions in total for comparison.

Figure 2.7a shows the comparison of the population specificity of the CNVs reported by treeSGS in the 18 regions. In the comparison again each region, if the CNVs in a CNV profile overlap with the region, the coefficients of the CNV profiles are used to classify the samples into two groups as with/without CNV calls in the region. For each profile, the samples with the CNV calls in the region are further divided by the differentiated populations as suggested in the two studies. Then, across the overlapping profiles, the mean and standard deviation of the ratio of the samples in the overrepresented population are reported. For example, for gene $SLC25A24$, African population shows more copy number than Asian populations; for gene $KIAA1267$, European populations show more copy number than non-European populations. A P_value is then calculated to measure the significance of the enrichment in the suggested population. Overall, the CNV profiles show very consistent population specifically suggested by the two studies while both SGL and SLEP-GL did not provide comparable consistency as shown in the supplementary document.

Database of Genomic Variants (DGV) [67] provide a comprehensive summary of structural variation in the healthy human genome. It reported more than 300,000 CNVs from about 55,000 samples in 72 studies. Figure 2.7b shows that, among all the 841 CNVs in Hapmap3 data, treeSGS detected CNV genotypes that overlaps with the most frequently CNVs in DGV. The agreement supports the CNV are likely true population-level CNV signatures.

The recent study in [15] only reported population-differentiation CNVs from pairwise population comparisons. In contrast, treeSGS can detect CNVs that differentiate between groups of populations based on the population tree. To demonstrate the differences, we show several cases of improved annotation of population-differentiation CNVs by treeSGS in Figure 2.8. In Figure 2.8b one of the CNV profiles (top) identified by treeSGS overlaps with 2 reported AFR-vs-AMR population-differentiation CNV regions, one duplication region and one deletion region. However, the coefficients of the latent CNV profile (bottom) suggest that this profile should be AFR (ASW, LWK, MKK, YRI) population-differentiation vs all other populations. The result is clearly supported by the original CNV data (middle) since the AFR population group show highly frequent duplications and deletions in the two regions. Overall, it is a better generalization that the two regions are significantly differentiated between AFR and non-AFR, instead of only AFR vs AMR, and furthermore, these two regions co-occur as they are in the same CNV profile. Figure 2.8a shows a CNV profile (top) overlapping with 4 reported population-differentiation CNV deletions. Although the 4 CNVs are annotated differently, the coefficients (bottom) suggest that this profile should be considered as differentiated between EAS+MXL+GIH and AFR, again strongly supported by the original data (middle). Note that EUR populations is shown some major deletions in 2 CNV regions but not all of the 4 regions, which explains why corresponding coefficients are not active for the EUR group. Figure 2.8c and 2.8d show two CNV profiles with both coefficients reporting differentiation between non-AFR and AFR matching well with the original data. This is clearly an improvement over the reported AFR vs GIH/MXL/EAS population-differentiation in the original study in [15].

2.3.5 Comparison to 1000 genome project data

1000 Genomes Project creates public catalogue of human variation and genotype data. The phase3 project collects 2504 samples' sequencing data from 26 populations. Similarity, the 26 populations are classified in the same 5 super populations as Hapmap3. 1000 Genome Project phase 3 data also share some samples with Hapmap3 data, with 675 samples are identical from 9 populations (no MKK, CHD populations). The variant calls of 1000 Genomes Project phase 3 data report 2974 CNV regions in autosomes, which overlapped 220 CNV regions in Hapmap3 data (overlap with at least 2k base

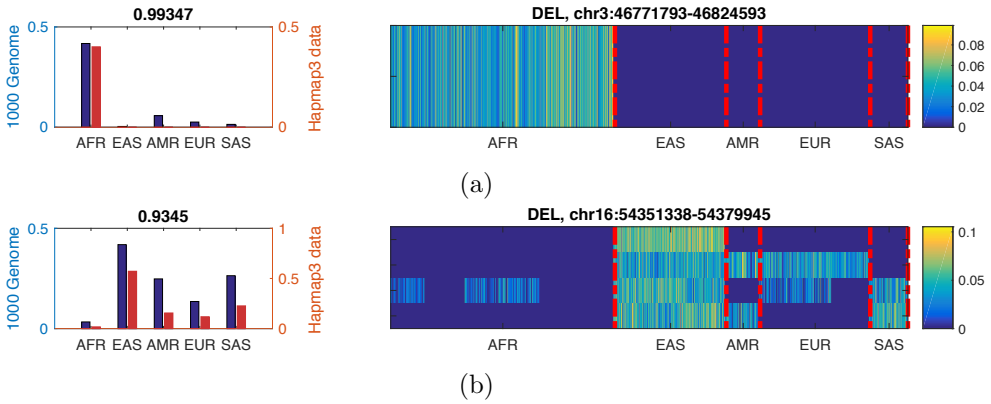


Figure 2.9: **Examples of similarity on super population frequency of 1000 Genome data and treeSGS results on Hapmap3 data.** two CNV regions are shown. Left bar plot shows similarity of 1000 Genomes reported frequency and treeSGS reported frequency on Hapmap3 data, with treeSFS reported coefficients displayed on right plot.

pair). Among these overlapped CNVs, there are 10 CNV regions which are vary significantly in super population level (CNV frequencies of super populations have difference greater than 0.55) and we validate these regions by checking the similarity of sample coefficients from treeSGS results on Hapmap3 and CNV frequency in 1000 Genomes Project phase3 data.

Figure 2.9 shows 2 examples of all 10 results. Others can be find in supplementary material. Each figure represent a comparison on a CNV region that is reported in both 1000 Genomes Project Phase 3 data and Hapmap 3 data. In each sub-figure, the right plot is the treeSGS coefficients of CNV profile(s) on Hapmap3 data that contains this CNV region (marked in title). The corresponding super population ratios are shown in the left plot as red bar. The blue bar shows the sample ratios from 1000 Genomes Project data. The similarity of bars (correlation coefficient) is marked in bar plot title. All 10 CNVs show a average similarity of 0.8884. This high consistency shows that treeSGS detects CNV specificity on super population is accurate with respect to 1000 Genome sequencing data.

2.4 Discussion

In this chapter, we propose a tree-guided group selection method, treeSGS, which using information gain theory to dynamic split tree into groups and then using group sparse selection to identify population-differentiation CNV profiles. Experimental results clearly support that treeSGS accurately identified CNV profiles among population groups. In the comparison with previous studies, treeSGS not only found confirmed population-differentiation CNVs, but also improved CNV annotations with population level differentiation.

Chapter 3

Transfer Learning Across Cancers on DNA Copy Number Variation Analysis

3.1 Introduction

Normally there are two copies of each gene in the human genome located on paired DNAs in a chromosome. Large scale DNA alternations such as insertions or deletions could lead to copy number gain or loss of the genes, which are called DNA copy number variations (CNVs). CNVs have been found extremely common in human cancer genome [18, 19] and it is believed that CNVs play significant roles in cancer [14, 20]. New technologies such as array-based comparative genomic hybridization (arrayCGH) [68, 69] and SNP genotyping arrays, are now available to measure genome-wide CNVs in high resolution at a population scale for characterizing CNV patterns in cancer samples [6]. Identification and systematic analysis of CNVs can provide important insights into the cellular defects that are cancer causative and suggest potential therapeutic strategies.

Most previous computational research work focused on developing models for identifying individual CNV events from CNV samples of a single cancer type. [32] studied 17 cancer types with at least 40 samples in each cancer type and reported that about 80% somatic copy number alternations found in one cancer type can also be found in

pooled analysis excluding that cancer type. The detected regions in the pooled analysis were also found in other cancer types that are better localized. These common and type-specific CNVs can potentially reveal unknown cancer mechanisms in the light of cross-cancer-type analysis. However, currently there is no unified mathematical model to simultaneously detect the CNV events common or specific to multiple cancer types from CNV array datasets.

In this chapter, we propose a Transfer Learning with Fused Lasso model (TLFL) to detect latent CNV components from CNV datasets of multiple cancer types, in which each cancer type can be regarded as one domain in transfer learning. Common latent CNV components are used as a bridge to transfer knowledge among different cancer domains along with the domain-specific components for each cancer type to explain the observed CNV datasets. To represent the pattern of CNV events, fused lasso is applied on each latent CNV component to preserve the sparsity and block structure. By using alternating optimization to solve the TLFL model, common latent features and domain specific features could be detected from multiple domains. Compared with a baseline method without knowledge transfer, TLFL is more robust and identifies more accurate latent CNV components in simulations and experiments on real arrayCGH CNV datasets and SNP genotyping array datasets.

3.2 Related Work

DNA CNVs tend to occur in continuous blocks of various sizes and thus, the adjacent probe features are more likely to be associated in the same CNV event. Previously, several models, such as change-point detection [70, 71], hidden Markov models [72, 73] and Gaussian models [74, 75] have been applied to address the challenge. More recently, fused lasso model [76] which introduces ℓ_1 norm constraint to encourage sparse change points and fused CNV features, has been found to be effective in discovering more interpretable CNV events [77]. A fused lasso latent feature model, FLLat [29] was proposed to take full advantage of any shared information among samples. The model assumes each CNV sample is a linear combination of a few latent CNV components. By factorizing the arrayCGH data matrix into the product of a coefficient matrix and a latent feature matrix, FLLat is able to detect underlying CNV events and discern

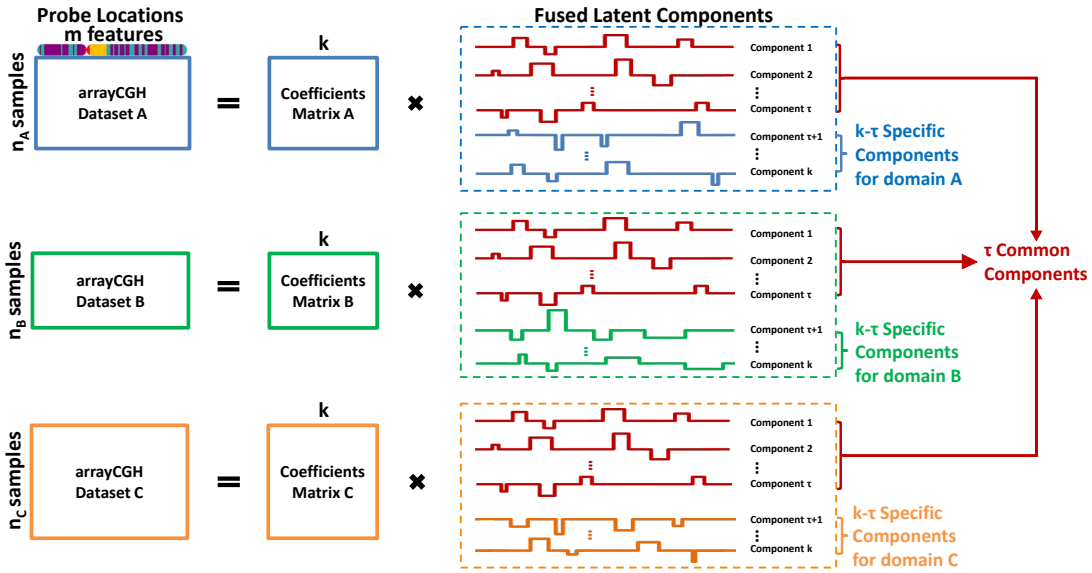


Figure 3.1: Outline of TLFL model. ArrayCGH or SNP genotyping array datasets from three domains are decomposed into coefficient matrices and matrices of k latent components. The probe locations are identical in all three datasets (m features) while the number of samples (n_A, n_B and n_C) can be different. The red latent components are τ common components shared in the three domains, and the remaining components in the same color of each dataset are $k - \tau$ domain specific components. For better visualization, matrices in this figure are transpose from equations.

specific relationships between samples. [60] proposed a latent fused-lasso feature method to use prior knowledge to learn group specific CNVs. Other multiple sample analysis methods which are powerful to identify frequent individual CNVs [78–81], are neither designed to identify CNV components nor capture the heterogeneity of samples. None of the previous methods was specially designed as a unified mathematical formulation to discover CNV events from multiple datasets across different cancer types.

Transfer learning uses common knowledge or structures among different domains to enhance multiple learning tasks [48, 82]. Recently, a lot of research work on transfer learning has been published for various learning problems such as Co-Clustering based Classification [83], Label Propagation [84], Collaborative Dual-PLSA [85] and Matrix Tri-Factorization based Classification [86]. The paradigm of transfer learning also fits

the learning tasks of finding CNV components across cancer types since datasets of the same or similar cancer types presumably bear the same or similar pathogenic cause. However, to the best of our knowledge, no transfer learning method has been designed for latent fused-lasso component discovery.

3.3 Method

Figure 3.1 is an outline of the TLFL model. In the Figure, each of the three cancer CNV datasets is factorized into a product of a coefficient matrix and k latent components. In each set of the k components, τ components are shared across the three datasets and the remain $k - \tau$ components are specific to each dataset. The framework assumes that the CNV features are measured on the same set of probe locations sampled from a chromosome. Each component is learned with fused lasso on the adjacent probe features to enforce a shape of step function to mimic true CNV signals. In the following, we first describe the optimization formulation of the model and then introduce an alternating optimization algorithm to minimize the cost function. Strategies for selecting hyper-parameters and initialization are also suggested for the empirical practice of the algorithm.

Table 3.1: Notations

Notation	Description
δ	# of domains
n_d	# of samples in domain $d \in [1, \delta]$
m	# of CNV features
k	total # of components in one domain
τ	# of common components
X_d	data matrix of domain d, size $m \times n_d$
$\hat{U}$	matrix of common components, size $m \times \tau$
U_d	domain-specific components of domain d, size $m \times (k - \tau)$
V_d	coefficient matrix of domain d, size $k \times n_d$

3.3.1 Transfer Learning Framework

The notations are given in Table 3.1. Given δ datasets measured from the same m probe locations, each dataset X_d contains n_d samples from one cancer domain. The objective is to recover k latent components $[\hat{U}, U_d]$ to reconstruct each dataset X_d with the minimal loss of information, where U_d are $k-\tau$ latent components specific to dataset X_d and $\hat{U}$ are τ common components shared by all the datasets. V_d is the corresponding coefficient matrix of $[\hat{U}, U_d]$ for reconstructing X_d. Specifically, the TLFL model assumes that each sample in X_d can be represented as a linear combination of k latent components as follows,

$$X_d = [\hat{U}, U_d]V_d.$$

To obtain the k latent components $[\hat{U}, U_d]$ and coefficient matrix V_d that best reconstruct X_d , the objective function minimizes the reconstruction error of all the datasets by a sum of the squared loss across the datasets,

$$\sum_{d=1}^{\delta} ||X_d - [\hat{U}, U_d]V_d||_F^2.$$

To capture the spatial relation in the CNV probe features, each latent component (a column in $[\hat{U}, U_d]$) is constrained by a fused lasso. Specifically, the cost function for the common components in $\hat{U}$ is defined as,

$$\begin{aligned} & g(\hat{U}, \lambda_C, \gamma_C) \\ =& \lambda_C \sum_{j=1}^{\tau}\sum_{i=1}^{m} |\hat{U}_{(i,j)}| + \gamma_C \sum_{j=1}^{\tau}\sum_{l=2}^{m} |\hat{U}_{(l,j)} - \hat{U}_{(l-1,j)}|, \end{aligned} \tag{3.1}$$

where λ_C and $\gamma_C \in \mathbb{R}$ are parameters to weight the penalties and the lasso penalty is introduced to obtain sparse CNV events in the components. Similarly, the cost function for each domain-specific latent component is

$$\begin{aligned} & g(U_d, \lambda_d, \gamma_d) \\ =& \lambda_d \sum_{j=1}^{k-\tau}\sum_{i=1}^{m} |U_{d(i,j)}| + \gamma_d \sum_{j=1}^{k-\tau}\sum_{l=2}^{m} |U_{d(l,j)} - U_{d(l-1,j)}|, \end{aligned} \tag{3.2}$$

where λ_d and $\gamma_d \in \mathbb{R}$ are also parameters to weight the penalties. Here, λ_C, γ_C, λ_d and γ_d for $d = 1, 2, \ldots, \delta$ are hyper-parameters to be tuned (see section 3.3.3).

Given all the cost terms introduced above, the complete objective function is defined as

$$\begin{aligned}\mathcal{L} = \sum_{d=1}^{\delta}(\frac{1}{2}||X_d - [\hat{U}, U_d]V_d||_F^2 \\ +g(U_d, \lambda_d, \gamma_d)) + g(\hat{U}, \lambda_C, \gamma_C) \\ s.t. \\ V_d \geq \mathbf{0} \text{ and } V_{d(i,:)}V_{d(i,:)}^T = 1 \text{ for } i = 1, 2, \ldots, k,\end{aligned} \tag{3.3}$$

where $V_d \geq 0$ denotes the condition that each element in V_d is nonnegative and $V_{d(i,:)}$ is the ith row of V_d. This cost function combines the reconstruction errors with the lasso and fused lasso terms weighted by λ_C, γ_C, λ_d and γ_d for $d = 1, 2, \ldots, \delta$. The nonnegative constraints on V_d only allow positive coefficients to combine latent components which might contain both amplification (positive) and deletion (negative) events. Each row in every V_d is also normalized across the samples such that the learned latent components are scaled to be comparable with each other [29]. The normalization also encourages even contributions from every latent component features to prevent being dominated by a few. Those considerations are meant to improve the interpretability of both the coefficients and the components.

3.3.2 Alternating Optimization

The optimization problem in eqn 5.1 can be solved by alternating updates to the variables $\hat{U}$, U_d and V_d iteratively. Specifically, we solve subproblems on only one group of variables by fixing the other two and alternate through the three groups of variables in each iteration. The alternating procedure is repeated until convergence. The detailed TLFL algorithm is described in Algorithm 4. Below we outline the solution to each subproblem to solve for $\hat{U}$, U_d and V_d, respectively.

Algorithm 4 TLFL

Input: $\{X_d\}_{d=1}^{\delta}$, k, τ, $\{\gamma_d\}_{d=1}^{\delta}$,$\{\lambda_d\}_{d=1}^{\delta}$, γ_C, λ_C
Output: $\hat{U}$, $\{U_d\}_{d=1}^{\delta}$, $\{V_d\}_{d=1}^{\delta}$

initialize $\hat{U}$, $\{U_d\}_{d=1}^{\delta}$
repeat
  for $d = 1, \ldots, \delta$ **do**
    for $j = 1, \ldots, n_d$ **do**
      solve $\arg\min_{V_{d(:,j)}} ||X_{d(:,j)} - [\hat{U}, U_d]V_{d(:,j)}||_F^2$
      s.t. $V_{d(:,j)} \geq \mathbf{0}$ (eqn 3.5)
    end for
    normalize V_d s.t. $V_{d(i,:)} \times V_{d(i,:)}^T = 1$ for $i = 1, 2, \ldots, k$
    $\dot{X}_d = X_d - \hat{U}V_{d(1:\tau,:)}$
    solve $\arg\min_{U_d}(\frac{1}{2}||\dot{X}_d - U_d V_{d(\tau+1:k,:)}||_F^2 + g(U_d, \gamma_d, \lambda_d))$ (eqn 3.6)
  end for
  for $d = 1, \ldots, \delta$ **do**
    $\ddot{X}_d = X_d - U_d V_{d(\tau+1:k,:)}$
  end for
  $X_{all} = [\ddot{X}_1, \ddot{X}_2, \ldots, \ddot{X}_\delta]$
  $V_{all} = [V_{1(1:\tau,:)}, V_{2(1:\tau,:)}, \ldots, V_{\delta(1:\tau,:)}]$
  solve $\arg\min_{\hat{U}}(\frac{1}{2}||X_{all} - \hat{U}V_{all}||_F^2 + g(\hat{U}, \gamma_C, \lambda_C))$ (eqn 3.7)
until $\hat{U}$, $\{U_d\}_{d=1}^{\delta}$, $\{V_d\}_{d=1}^{\delta}$ converge

Updating coefficient matrix V_d

When $\hat{U}$ and U_d are fixed, eqn 5.1 is only a function on V_d simplified as

$$\begin{aligned} &\arg\min_{V_d} ||X_d - [\hat{U}, U_d]V_d||_F^2 \\ &s.t. \\ &V_d \geq \mathbf{0} \text{ and } V_{d(i,:)}V_{d(i,:)}^T = 1 \text{ for } i = 1, 2, \ldots, k. \end{aligned} \tag{3.4}$$

For each column $X_{d(:,j)}$, we can solve a nonnegative least-square problem to obtain a solution for $V_{d(:,j)}$.

$$\begin{aligned} &\arg\min_{V_{d(:,j)}} ||X_{d(:,j)} - [\hat{U}, U_d]V_{d(:,j)}||_F^2 \\ &s.t. \\ &V_{d(:,j)} \geq \mathbf{0}. \end{aligned} \tag{3.5}$$

Then V_d can be normalized as $V_{d(i,:)}V_{d(i,:)}^T = 1$ for $i = 1, 2, \ldots, k$.

Updating domain-specific components U_d

When $\hat{U}$ and V_d are fixed, eqn 5.1 is only a function on U_d simplified as

$$\begin{aligned} &\frac{1}{2}||X_d - [\hat{U}, U_d]V_d||_F^2 + g(U_d, \gamma_d, \lambda_d) \\ =&\frac{1}{2}||\dot{X}_d - U_d V_{d(\tau+1:k,:)}||_F^2 + g(U_d, \gamma_d, \lambda_d), \end{aligned} \tag{3.6}$$

where residue $\dot{X}_d$ is defined as

$$\dot{X}_d \equiv X_d - \hat{U}V_{d(1:\tau,:)}.$$

This problem is equivalent to the general fused lasso problem, which can be solved by the SLEP package [87].

Updating common components $\hat{U}$

When U_d and V_d are fixed, eqn 5.1 is only a function on $\hat{U}$ simplified as

$$\begin{aligned} &\sum_{d=1}^{\delta}(\frac{1}{2}||X_d - [\hat{U}, U_d]V_d||_F^2) + g(\hat{U}, \gamma_C, \lambda_C) \\ =&\frac{1}{2}||X_{all} - \hat{U}V_{all}||_F^2 + g(\hat{U}, \gamma_C, \lambda_C), \end{aligned} \tag{3.7}$$

where we define

$$\begin{aligned} \ddot{X}_d &\equiv X_d - U_d V_{d(\tau+1:k,:)}, \\ X_{all} &\equiv [\ddot{X}_1, \ddot{X}_2, \ldots, \ddot{X}_\delta], \\ V_{all} &\equiv [V_{1(1:\tau,:)}, V_{2(1:\tau,:)}, \ldots, V_{\delta(1:\tau,:)}]. \end{aligned}$$

Similarly, this problem is also equivalent to the general fused lasso problem, which can be solved by the SLEP package.

3.3.3 Initialization and Hyper-parameter Selection

Since eqn 5.1 is not convex, alternating updates in TLFL do not guarantee a global optimal solution. The local optimal solution heavily relies on proper initialization of $\hat{U}$ and U_d. We adopt a simple strategy to choose the initialization. We use Principle

Component Analysis (PCA) on pooled data $[X_1, X_2, \ldots, X_\delta]$ to select top τ components as the initialization of common components $\hat{U}$. For domain specific components, PCA is applied on each domain data separately to select the top k components for each domain. Then, the top τ components of the k components of each domain that are most similar to the initialization of $\hat{U}$ are removed. The similarity is measured by the absolute correlation coefficients. For each domain, the remaining $k - \tau$ components are used as the initialization of domain specific components U_d.

The number of latent component k was chosen as the number of principle components that can explain $\alpha \in [0, 1]$ variation of the arrayCGH or SNP genotyping array datasets. For multiple domains, the calculated k could vary among the datasets. We simply choose the maximal as a global k to explain at least α variance in each dataset. A user also needs to select a parameter $\beta \equiv \tau/k$ to control the ratio between common component number τ and total component number k. For similar datasets such as datasets of the same or closely related cancer types, β should be chosen larger while for datasets from different cancer types, β should be chosen smaller. Presumably, β could be determined by a user's perception of the similarity across the domains.

Parameters λ_C, γ_C, λ_d and γ_d are chosen by the same Bayesian Information Criterion (BIC) introduced in [29]. BIC controls both model complexity and training error to avoid overfitting. For each domain, λ_d and γ_d are selected with dataset X_d and k components. λ_C and γ_C are selected with the combined dataset $[X_1, X_2, \ldots, X_\delta]$ and $\tau + \delta * (k - \tau)$ components. Note that we could apply BIC to the complete model in eqn 5.1 to jointly select λ_C, γ_C, λ_d and γ_d. However, jointly choosing four parameters is not scalable even on datasets of moderate size. Thus, we divided the estimation into smaller BIC problems as described above.

3.4 Simulation

In the section, we generated artificial datasets to test TLFL model in three measurements: 1) performance of recovering latent components; 2) performance of detecting hidden sample group structures in coefficient matrix for classification and clustering; and 3) convergence and robustness under different noise levels and ratios between common and domain-specific components. The synthetic datasets are constructed as

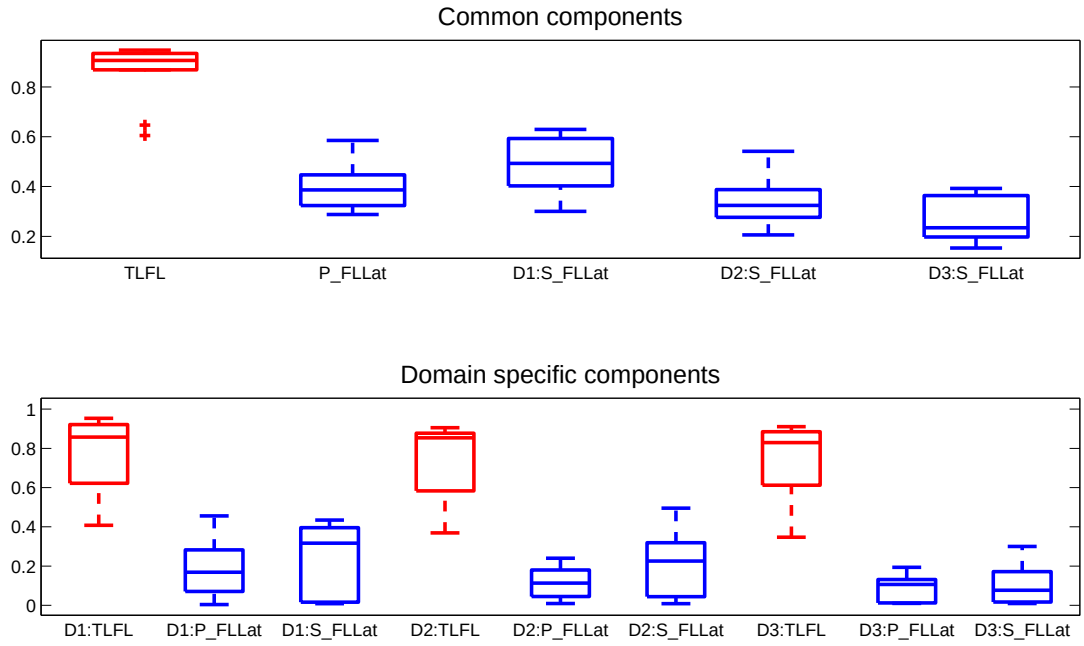


Figure 3.2: Performance of latent component detection by TLFL, pool FLLat (P_FLLat) and split FLLat (S_FLLat) section 3.4.1. The box-plots are computed from 10 random experiments. D1, D2 and D3 denote the three domains.

$X_d = [\hat{U}, U_d] * V_d + \Xi$, where latent component matrix $[\hat{U}, U_d]$ and coefficient matrix V_d are either predefined or randomly generated, and the entries in Ξ are IID gaussian noises. In all simulations, the hyper-parameters λ and γ are selected as described in section 3.3.3, and k and τ are assumed known. In each component in $[\hat{U}, U_d]$, 4 independent copy number gain or loss events were assumed and randomly located with magnitudes in $[-1, 1]$ over 2000 probe features. The components are not strictly orthogonal but the correlation between any two components is required to be smaller than 0.3. The entries in V_d are random nonnegative values in $[0, 1]$ and normalized as $V_{d(i,:)} V_{d(i,:)}^T = 1, i = 1, 2, \ldots, k$. We compared TLFL with FLLat [29] to show the advantage of transfer learning and discrimination of common and domain specific components. In each experiment, TLFL is applied jointly on three datasets. FLLat was applied on 1) a pooled dataset of all the domain datasets (pool FLLat) and 2) each domain dataset individually (split FLLat).

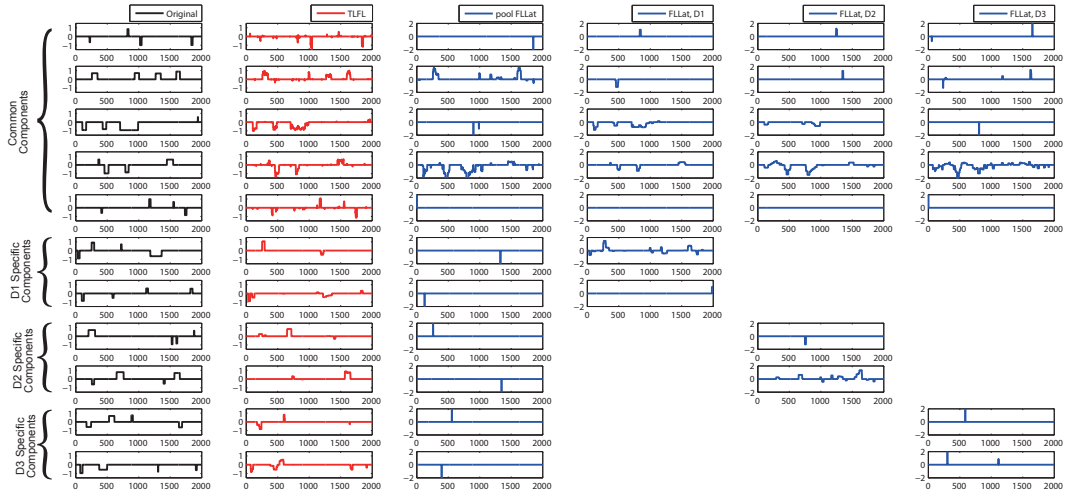


Figure 3.3: Latent components detected by TLFL and FLLat are compared with the known true components. The rows represent the common components and the components specific to the domains (D1, D2 and D3). The columns from left to right represent true components, components detected by TLFL, components detected by pool FLLat, and components detected by split FLLat for D1, D2 and D3 with one column for each domain.

3.4.1 Recovering Latent CNV Components

Three synthetic datasets of sample size 300, 420 and 510 respectively were generated. In all the datasets, there are 7 latent components, 5 of which are common components $\hat{U}$ and 2 are domain-specific components U_d for each dataset, Note that no structure is assumed in the coefficient matrices V_d in this simulation. Gaussian noises $\Xi \sim (\mu = 0, \sigma = 0.3)$ were added. In this simulation, we focused on recovering the known latent components used to generate the synthetic datasets with added noise. The performance is measured by the average Pearson correlation coefficients of each estimated latent component with its corresponding known component. Since FLLat allows negative coefficients, some latent components were negated to obtain the best correlation coefficients with the known components. With the components were fixed, randomized coefficient matrices and noise were generated for 10 trials.

The performance of TLFL, split FLLat and pool FLLat for recovering the known components is shown in Figure 3.2. TLFL outperformed both split and pool FLLat

in each domain under the comparison across either common components or domain-specific components. Interestingly, TLFL tends to identify more consistent common components than the FLLat models in the 10 repeats with smaller variance. Paired-sample t-test of the component correlations by TLFL and FLLat for common components, domain-specific components and all components are all significant with the largest $p-value = 4.46E-04$, which indicates that TLFL significantly outperforms both split and pool FLLat in detecting the known latent CNV components. To illustrate the detect components, Figure 3.3 shows the side-by-side comparison of each component detected by FLTL, split FLLat or pool FLLat with the known component from one trial. In this example, pool FLLat failed to detect the third common component and split FLLat detected no signal correlating with the second common component in all three domains while TLFL captured all the true events accurately. In the fifth common component, both FLLat methods failed to separate the signal from the other components. Similar advantages by TLFL are also seen in the comparison of domain-specific components.

3.4.2 Sample Classification by Coefficient Matrices

Under the assumption that the latent components are underlying features describing tumor characteristics, the coefficient matrices are presumably informative for patient classification or clustering. For example, some latent features might represent CNV aberrations disrupting a gene pathway in a certain tumor stage, and thus samples with a large coefficient on the latent features are more likely to be associated with that particular tumor stage. Therefore, in this simulation we focused on using the learned coefficient matrices for sample classification and clustering.

Similarly, three synthetic datasets of sample size 300, 420 and 510 respectively were generated with 5 common latent components and 2 domain specific components in each domain. To create patient classes (clusters), we designed coefficient matrices representing patterns of three classes (patient subgroups) in each domain as shown in Figure 3.4. The true coefficient matrices shown at row 2 in Figure 3.4 are constructed by adding gaussian noise on the structured seed matrices at row 1. The coefficient matrices were then multiplied with components similarly generated as in section 3.4.1 and added with gaussian noises $\Xi \sim (\mu = 0, \sigma = 0.3)$ to get the synthetic datasets. With the latent components and structure seeds fixed, we repeated the simulation procedure 10 times

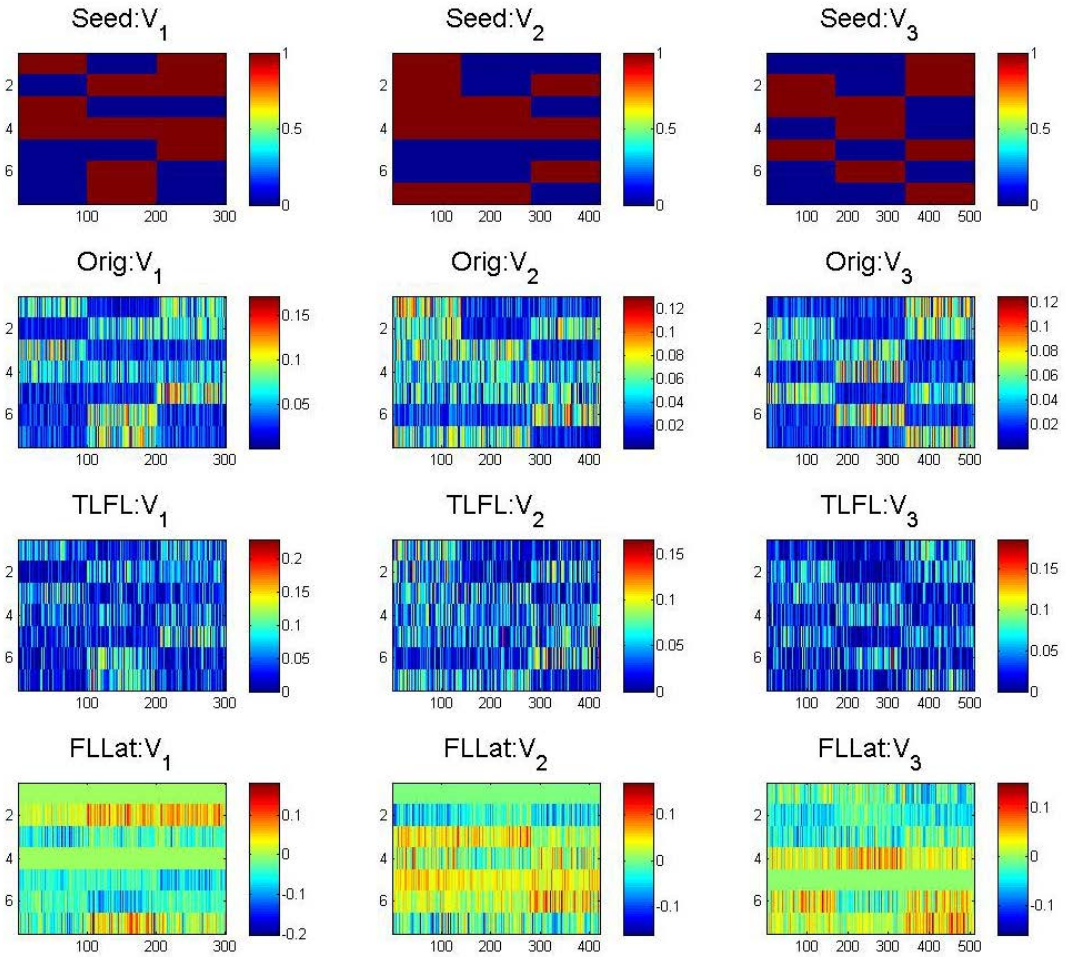
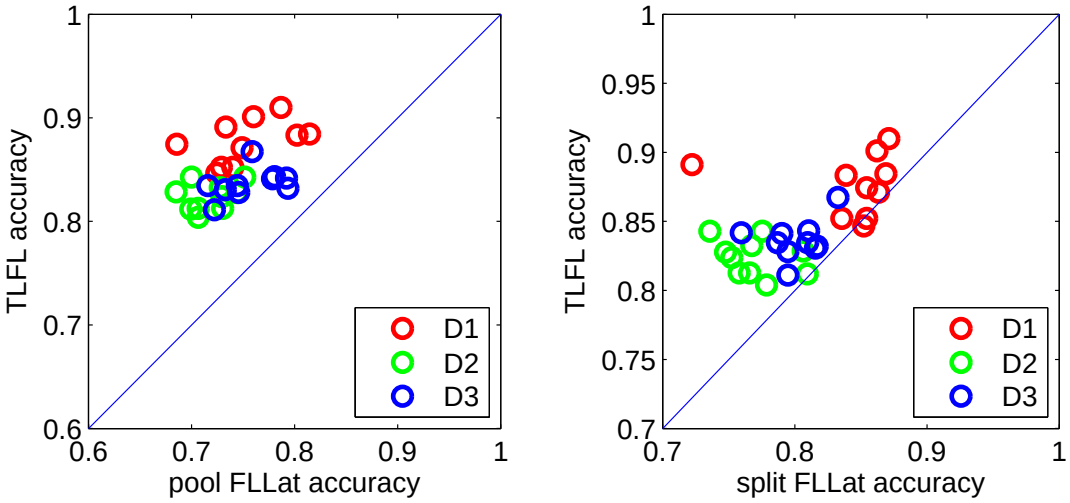


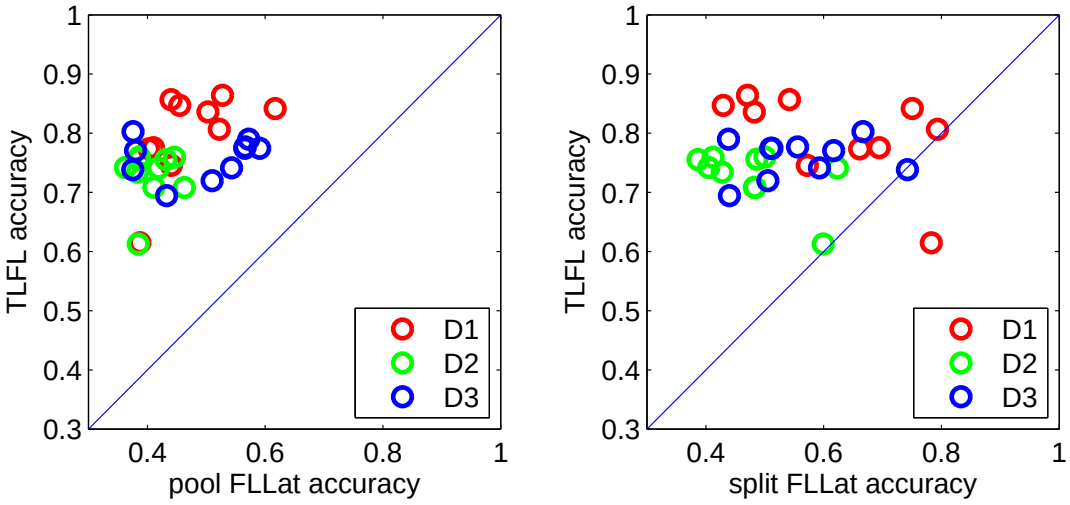
Figure 3.4: Comparison of learned coefficient matrices (components by samples). The plots are shown for row 1: structured seed matrices; row 2: true coefficient matrices constructed by adding noise to the structured seed matrices; row 3: coefficient matrices learned by TLFL; and row 4: coefficient matrices learned by split FLLat. Three classes of equal sizes are assumed in each domain.

under the gaussian noises.

The last two rows of matrices in Figure 3.4 show the coefficient matrices learned by TLFL and split FLLat in one trial. In this visualization, it is clear that split FLLat made mistakes in several places such as zero coefficient of the first component in domain



(a) Leave-one-out classification



(b) K-means clustering

Figure 3.5: Classification and clustering performance on coefficient matrices learned by TLFL, pool FLLat and split FLLat. The comparisons are between the methods on the three domains (D1, D2 and D3) in 10 random trials.

1 and domain 2, and the fifth component on domain 3. The overall structure of the coefficient matrices in not as distinguishable as those detected by TLFL. Since pool FLLat learned a different number features (number of rows in V_d), it is not directly

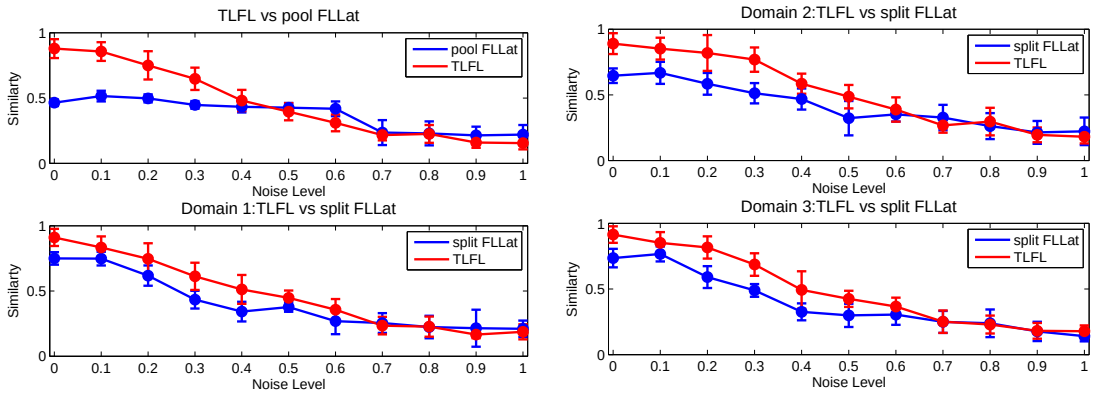


Figure 3.6: Components detection performance comparison between TLFL and pool/split FLLat under different noise levels.

comparable in Figure 3.4 .

To better measure the accuracy of the coefficients, classification and clustering of samples were performed on the learned coefficient matrices. The leave-one-out cross-validation with linear SVM classifier was performed for classification of the samples. K-means clustering (K=3) was applied to cluster the samples. For K-means clustering, the averages of 100 runs are reported for each domain in each trial. Figure 3.5 shows the comparison of the classification and clustering results by TLFL and FLLat (pool and split) by scatter plots. In both classification and clustering comparisons, almost all the cases are well above the diagonal line, i.e. TLFL performed better than FLLat by a large margin. In addition, TLFL also detected better components in this simulation (results not shown).

3.4.3 Robustness and Convergence

To understand the robustness of TLFL and FLLat under the presence of different noise level, we tested datasets with varying amount of added noise in this simulation. Three domain datasets of sizes 60, 75 and 90 respectively were generated with 5 common components and 2 domain specific components in each domain. The gaussian noises were drew from $(\mu = 0, \sigma)$ with σ ranging from 0 to 1 with 0.1 step. To test each noise level, the simulations were repeated 10 times. Figure 3.6 shows that the performance

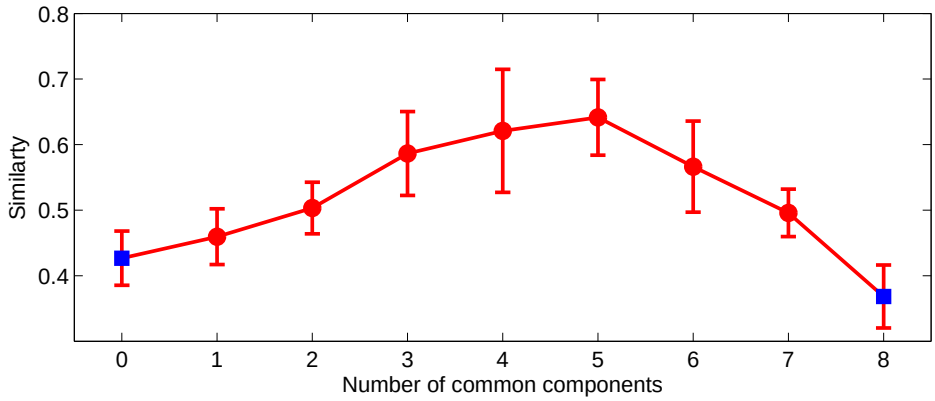


Figure 3.7: Effect of varying the number of common components. The errorbars show TLFL performance under different τ with fixed datasets. Note that when $\tau = 0$, TLFL is equivalent to split FLLat and when $\tau = 8$, TLFL is equivalent to pool FLLat.

of component detection drops as the noise level increases for both TLFL and FLLat. TLFL performs consistently better than both pool FLLat and split FLLat when the noise level is reasonable (≤ 0.5) with the benefit of transfer learning. TLFL and FLLat performs similarly due to the extremely high noise level that almost completely blurred the original signals. And at this noise level the accuracy of the learn components is very low.

In most of the real cases, the best ratio of τ and k is unknown. It is thus interesting to understand the performance of TLFL when τ varies. Intuitively, τ is directly related to how much knowledge to transfer across the different domains. The more similar the domains, the larger τ desired. In the two extremes, when $\tau = 0$ TLFL is equivalent to split FLLat, and when $\tau = k$ TLFL is equivalent to pool FLLat. We generated synthetic datasets of sample size 150, 180 and 210, each with 600 features and 8 latent components in each domain, 4 of which are common components. Similarly, we fixed the components and generated coefficient matrices randomly with gaussian noises $\Xi \sim (\mu = 0, \sigma = 0.3)$ added in 10 trials for each choice of $\tau \in [1, 2, \ldots, 7]$. The results of 10 trials is shown in Figure 3.7. It is clear that when $\tau = 4$ or 5, which is close to the true τ, TLFL performs the best.

Figure 3.8 shows one example of convergency in running the TLFL algorithm. TLFL

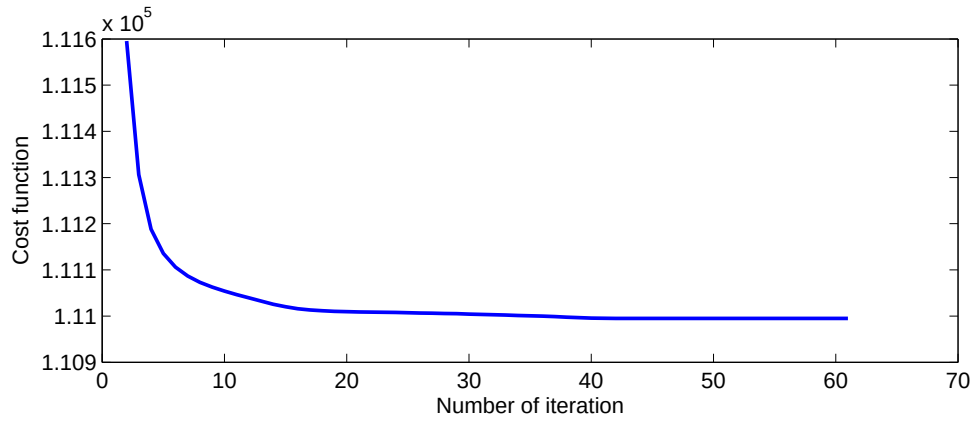


Figure 3.8: Convergence of TLFL for one run from section 3.4.1. After around 60 iteration, components and coefficient matrices are converged.

convergences fast within lower tens of iterations. Most of the simulations aforementioned converged less within 100 times regardless of the sample sizes.

3.5 Experiments on Cancer Datasets

We performed two experiments on real cancer CNV datasets. The first experiment is a cross-dataset analysis on bladder cancer to show that TLFL can utilize information from other similar datasets to improve classification. The second experiment is a cross-domain analysis on breast cancer and ovarian cancer.

3.5.1 Analysis Across Bladder Cancer Datasets

TLFL, split FLLat and pool FLLat were tested on two bladder cancer arrayCGH datasets: Blaveri05 [88] and Stansky06 [89]. Both datasets contain urothelial carcinomas with whole-genome tiling resolution arrayCGH and high density expression profiling. There are 98 samples in Blaveri05 dataset and 57 in Stansky06. Since the two datasets were not measured by the same resolution, we interpolated the datasets in whole genome to obtain CNV readings at the same probe positions with a resolution of 500k bps per probe. All the samples from the two arrayCGH datasets are provided with information on tumor stage. In Blaveri05 dataset, the stages are Ta, T1, T2, T3

and T4, and in Stransky06 dataset, the stages are Ta, T1a, T1b, T2, T3a, T3b, T4a and T4b. We relabel the stages into 3 classes for each dataset: Blaveri05 with stages ({Ta}, {T1, T2}, {T3, T4}) and Stransky05 with stages ({Ta},{T1a, T1b, T2},{T3a, T3b, T4a, T4b}), ordered from less severe stage to more advanced stage.

For each chromosome in the two datasets, the number of latent components was chosen as the number of principle components that could explain at least 80% variance of the data. The parameter k for a certain chromosome was then set as the larger number of principle components of the two datasets. Since both datasets are on similar bladder carcinomas, we assume a large fraction of common components. For each chromosome, we took the ratio of τ/k as 70%. Parameters $\lambda_1, \gamma_1, \lambda_2, \gamma_2, \lambda_C$ and γ_C were calculated by BIC as described in section 3.3.3. Table 3.2 reports the leave-one-out SVM classification results of the three classes using the coefficient matrices learned by TLFL, split FLLat and pool FLLat. Among the tests on all 22 chromosomes, 11 tests of Stransky06 and 10 tests of Blaveri05 present the best classification results by TLFL than both FLLat methods (numbers with color red) while on two chromosomes of Stransky06 dataset and 7 of Blaveri05 dataset, TLFL performed worse classification than both FLLat methods (numbers with color blue). Overall improvement is observed on both datasets for the average classification results of the 22 chromosomes.

Table 3.2: Classification of bladder cancer datasets.

Chr	Stransky06			Blaveri05			Average		
	TLFL	pool FLLat	split FLLat	TLFL	pool FLLat	split FLLat	TLFL	pool FLLat	split FLLat
1	0.4795	0.4444	0.4386	0.5748	0.5714	0.5782	0.5272	0.5079	0.5084
2	0.4912	0.4737	0.4737	0.6361	0.6224	0.6361	0.5636	0.5481	0.5549
3	0.5906	0.5614	0.5029	0.6429	0.6429	0.6599	0.6168	0.6021	0.5814
4	0.6608	0.5848	0.6082	0.5544	0.5578	0.5544	0.6076	0.5713	0.5813
5	0.5439	0.5088	0.5263	0.6565	0.6565	0.6429	0.6002	0.5826	0.5846
6	0.5731	0.5556	0.5556	0.5884	0.6190	0.5918	0.5808	0.5873	0.5737
7	0.5906	0.6667	0.6374	0.6633	0.6395	0.6361	0.6270	0.6531	0.6367
8	0.6199	0.6316	0.6140	0.5952	0.5986	0.5714	0.6076	0.6151	0.5927
9	0.6140	0.6082	0.5146	0.6020	0.6224	0.6156	0.6080	0.6153	0.5651
10	0.6023	0.6316	0.5322	0.5850	0.5748	0.5748	0.5937	0.6032	0.5535
11	0.6140	0.6082	0.6023	0.6088	0.6395	0.6361	0.6114	0.6238	0.6192
12	0.5848	0.5556	0.5380	0.6020	0.5748	0.5748	0.5934	0.5652	0.5564
13	0.5439	0.5205	0.5673	0.5952	0.5816	0.5952	0.5695	0.5511	0.5812
14	0.5848	0.6433	0.5789	0.5680	0.5816	0.5918	0.5764	0.6125	0.5854
15	0.4737	0.4444	0.4795	0.6293	0.6190	0.5918	0.5515	0.5317	0.5357
16	0.6433	0.6491	0.6316	0.5782	0.6122	0.5884	0.6108	0.6307	0.6100
17	0.5205	0.6257	0.5322	0.5000	0.5034	0.5646	0.5102	0.5646	0.5484
18	0.5380	0.5322	0.4971	0.6224	0.6122	0.6054	0.5802	0.5722	0.5513
19	0.5322	0.5146	0.5789	0.5850	0.6122	0.6054	0.5586	0.5634	0.5922
20	0.6550	0.6667	0.6491	0.5986	0.6020	0.5918	0.6268	0.6344	0.6205
21	0.4561	0.4795	0.4561	0.5374	0.5136	0.5238	0.4968	0.4966	0.4900
22	0.5673	0.5380	0.4678	0.5782	0.5136	0.5340	0.5727	0.5258	0.5009
ave	0.5673	0.5657	0.5447	0.5955	0.5942	0.5938	0.5814	0.5799	0.5693

3.5.2 Analysis Across Cancer Domains

We applied TLFL method on two related cancer types, breast cancer and ovarian cancer, to detect common CNV patterns. The two CNV datasets were downloaded from TCGA data-portal[1] SNP level 2 tangent data, generated from Affymetrix Genome-Wide Human SNP Array 6.0 platform. To label the patients for survival prediction, we chose breast cancer patient samples that had a survival time less than 5 years as the positive group and longer than 8 years as the negative group. Similarly, we chose the ovarian cancer patients with survival time less 1 year as positive samples and longer than 5 years as negative samples. With this criteria, 103 breast cancer samples (56 positive and 47 negative) and 124 ovarian cancer samples (46 positive and 78 negative) were selected. To reduce the computational load, we sampled data with 150k bp per probe resolution. Based on the genetic relevance of breast cancer and ovarian cancer described in OMIM, we focused on chromosomes 3, 8, 10, 13 and 17 in this analysis. The number of components were chosen to explain between 60%-75% of variance in each chromosome respectively. Since these are two different but related cancer types, we took a smaller ratio of τ/k as 60%.

Similarly, leave-one-out classification was performed on the coefficient matrices learned by TLFL, pool FLLat and split FLLat. The results are shown in Table 3.3. TLFL performed similar classification to FLLat on chromosome 3 and 8 but better on the other chromosomes and overall average of both the breast cancer and ovarian cancer datasets.

To detect more focal CNV events (short CNV regions), we increased the hyper-parameter of common components γ_C and λ_C by multiplying a factor 2.5 and reran TLFL on both datasets. The common CNVs between breast cancer and ovarian cancer detected by TLFL are shown in Figure 3.9. Eighteen known cancer genes locate in these very focal CNV regions. thirteen among the eighteen genes (except CCDC6, FAM22A, ZMYM2 and SRSF2. GATA3 is found only related with breast cancer) were reported to play a role in both breast cancer and ovarian cancer as reported by details in Table 3.4. For example, deletion or hyper-methylation of tumor suppressor FHIT leads to high proliferation of both breast cancer and ovarian cancer [90–93]; and BRIP1 interacts with BRCA1 and its variants are candidates of breast and ovarian cancer susceptibility [94].

[1] `https://tcga-data.nci.nih.gov/`.

The extensive literature supports that those common CNVs might play an important role in both breast and ovarian cancer.

Table 3.3: Classification of breast and ovarian cancer datasets.

Chr	Breast cancer			Ovarian cancer			Average		
	TLFL	pool FLLat	split FLLat	TLFL	pool FLLat	split FLLat	TLFL	pool FLLat	split FLLat
3	0.5777	0.5922	0.5971	0.5363	0.6048	0.5040	0.5570	0.5985	0.5506
8	0.4466	0.4223	0.4612	0.4234	0.4153	0.5081	0.4350	0.4188	0.4846
10	0.6553	0.5194	0.5922	0.4758	0.3992	0.4516	0.5656	0.4593	0.5219
13	0.5194	0.4951	0.4612	0.5887	0.5887	0.5847	0.5541	0.5419	0.5229
17	0.5291	0.5049	0.5194	0.5766	0.5645	0.5323	0.5529	0.5347	0.5258
ave	0.5456	0.5068	0.5262	0.5202	0.5145	0.5161	0.5329	0.5107	0.5212

3.6 Conclusions

Application of transfer learning to CNV analysis across multiple cancer types is promising since CNVs are a hallmark of cancer genomes. To the best of our knowledge, TLFL is the first transfer learning method to utilize multiple cancer domains for detecting common and domain-specific CNVs as fused latent components. The transfer learning enables sharing information in datasets of different cancer domains to discover latent CNV features that can explain common and domain-specific cancer characteristics and better classify patient samples as shown in the experiments. In the recent TCGA (The Cancer Genome Atlas) initiative, more and more CNV datasets are becoming available for 21 types of cancer. It is expected that transfer learning will play an important role in the comparative analysis of the large patient cohorts to improve the current knowledge of cancer development and progression in the light of both common and specific cancer CNVs.

Table 3.4: Cancer genes in common components

Gene	Association with breast cancer and ovarian cancer	Hyperlink to reference
MLH1	Loss of MLH1 plays a role in drug resistance in breast cancer; methylation of the hMLH1 promoter is possibly related to cisplatin-resistance in ovarian cancer.	Mackay, H. J., et al. Samimi, Goli, et al. Strathdee, G., et al.
FHIT	Deletion or hyper-methylation of tumor suppressor FHIT leads to high proliferation of both breast cancer and ovarian cancer.	Fullwood, P., et al. Dhillon, V.S., et al. Campiglio, M., et al. Zochbauer-Muller, S., et al.
TFRC	TFRC together with ACTB are used for breast cancer quantification; TFRC expresses differently between normal and poorly differentiated serous papillary adenocarcinoma (PD-SPA) of the ovary.	Majidzadeh-A, K., et al. Martoglio, A. M., et al.
BMPR1A	BMPR1A highly expresses in breast cancer and ovarian cancer.	Alarmo, E. L., et al. Shepherd, T. G., et al. Bowen, N. J., et al.
CCDC6	Lack of evidence	
FAM22A	Lack of evidence	
FGFR2	Four SNPs of FGFR2 are confirmed highly associated with breast cancer and FGFR2 expresses increasingly in the rare homozygotes; combining FGFR2 inhibitors with platinum-containing cytotoxic agents for the treatment of epithelial ovarian cancer may yield increased anti-tumor activity.	Hunter, D. J., et al. Meyer, K. B., et al. Cole, C., et al.
GATA3	Low GATA3 expression is associated with higher histologic grade and short survival time in breast cancer; No direct evidence to show relation between GATA3 with ovarian cancer.	Mehra, R., et al. Hoch, R. V., et al.
MYST4	MYST4 is up-regulated in ER-positive breast cancer cells and ovarian cancer cells.	Kok, M., et al. Vignati, S., et al.
PTEN	PTEN may suppress tumor cell growth and regulate tumor cell invasion and metastasis through interactions at focal adhesions in breast cancer; PTEN mutations are frequent in endometrioid ovarian tumors.	Li, J., et al. Obata, K., et al.
FAS	FAS is a reliable prognostic marker to predict DFS and OS in patients with early breast cancer; Decreased sensitivity to Fas-mediated apoptosis could contribute to ovarian tumorigenesis and may play a role in ovarian tumorigenesis.	Alo, P. L., et al. Baldwin, R. L., et al. Meinhold-Heerlein, I., et al.
RB1	RB1 is most likely involved in the development of breast cancer; Two SNPs of RB1 showed significant association with ovarian cancer risk.	Spandidos, D. A., et al. Song, H., et al.
ZMYM2	Lack of evidence	
BRCA1	The 17q-linked BRCA1 gene is identified to have influences susceptibility to breast and ovarian cancer.	Ford, D., et al. Miki, Y., et al.
BRIP1	BRIP1 interacts with BRCA1 and its variants are candidates of breast and ovarian cancer susceptibility.	Song, H., et al.
SEPT9	Increased SEPT9_v1 expression contributes to the malignant pathogenesis of some breast tumors; Experiment shows consistent and specific overexpression of both SEPT9_v1 and SEPT9_v4 transcripts in the epithelial component of ovarian tumors.	Gonzalez, M. E., et al. Scott, M., et al.
SRSF2	Lack of evidence	
YWHAE	Expression level upregulated gene YWHAE together with other 5 genes show a significant association to both disease-free and overall survival in breast cancer; YWHAE is identified from the TOV-112D ovarian cancer cell line.	Cimino, D., et al. Gagné J. P., et al.

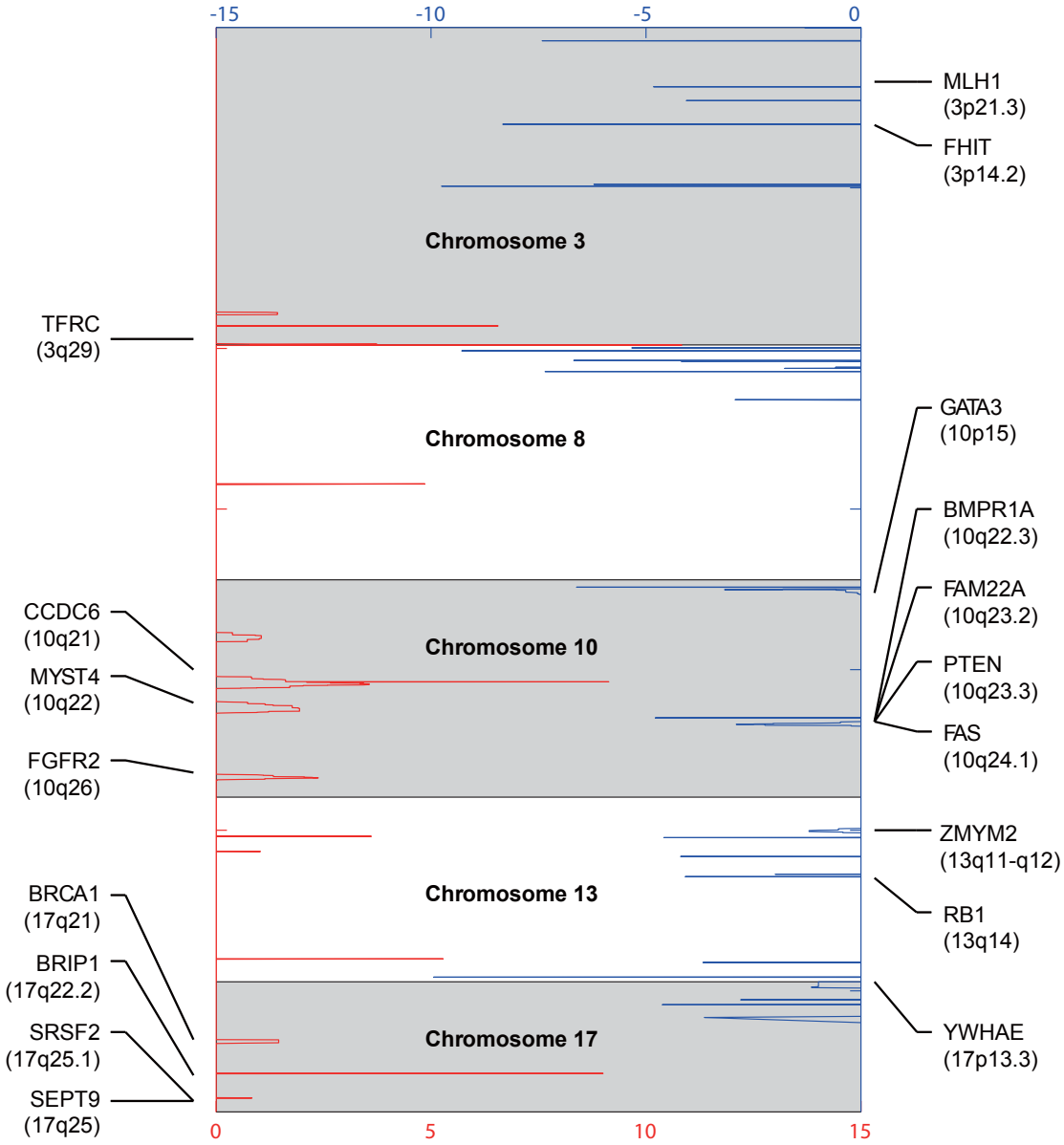


Figure 3.9: Common CNV events in breast cancer and ovarian cancer with co-located cancer genes annotated. Amplification (red) and deletion (blue) CNV events are plot along the selected chromosomes.

Chapter 4

Multitask Clustering of scRNA-seq Data

4.1 Introduction

Single-cell RNA sequencing (scRNA-seq) technology has emerged as a promising genome-wide mRNA expression quantification method in individual cells. Traditional bulk RNA-seq ignores the cell differences in a cell population and treats all cells as homogeneous. Furthermore, genes with low expression values may be undetectable in bulk RNA-seq since they may only be expressed in a small number of uncommon or transient cell types. To overcome these limitations, scRNA-seq identifies cell types by sub-populations of single cells to characterize sub-population structure and to understand disease progression and mechanisms of transcription regulation [25].

As a new technology, there are unique challenges in scRNA-seq experiments and data analysis. A typical scRNA-seq protocol is as follows: isolation of single cells and RNA, reverse transcription, amplification, library generation, and sequencing. In each step, technical noise and biases are introduced [11]. In addition to the noise and bias that also exist in bulk RNA-seq experiments, issues unique to scRNA-seq include those from biological sources, such as cell-cycle stage or cell size, as well as from technical/systematic sources, such as capture inefficiency, material degradation, sample contamination, amplification biases, GC content, and sequencing depth. For example, due to the tiny amount of starting material [26], heavy PCR amplification is needed

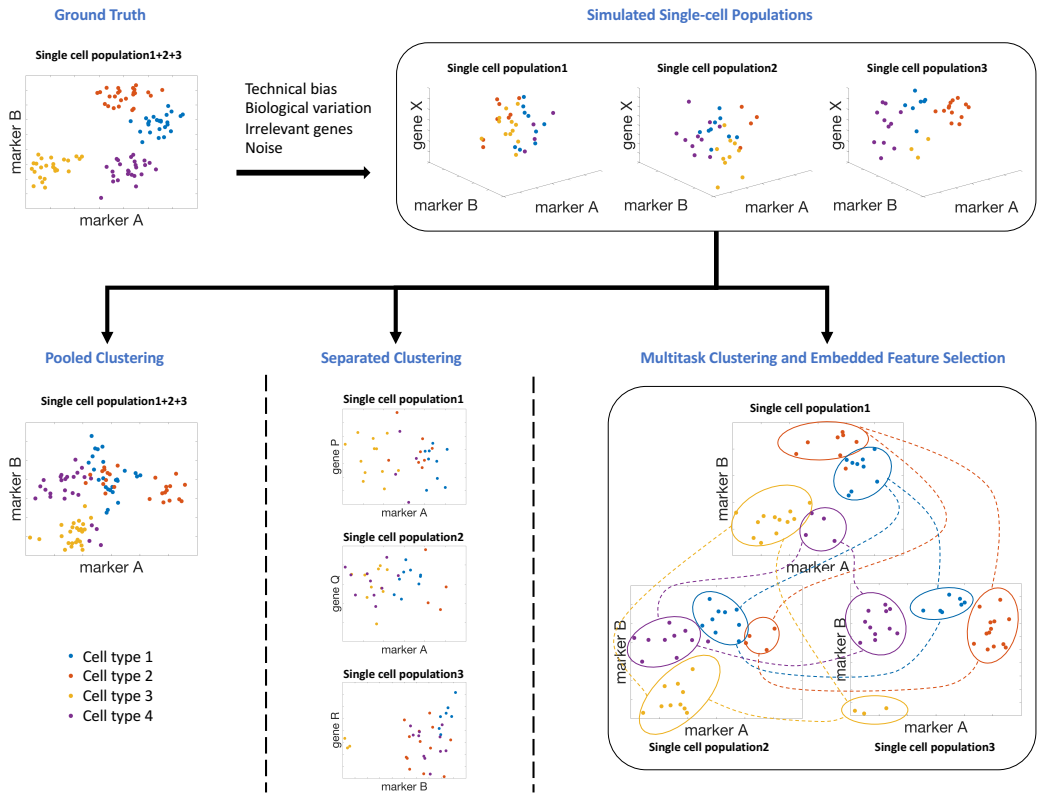


Figure 4.1: **Strategies of clustering multiple single-cell populations.**. In the example, four cell types are shown in four different colors. **Ground Truth**: 2D plot of a pool of single cells by the true marker genes A and B combined from 3 single-cell populations of identical distributions. **Simulated Single-cell Populations**: 3D plots of the three single-cell populations separated by marker genes A, B and non-marker gene X. The simulation data are generated from the ground truth data with rotation and scaling to represent technical biases and biological variation with an additional 997 random genes (1000 genes in total) added to each experiment. Additional noise is also introduced. Three different clustering strategies are shown below. **Pooled Clustering**: 2D plot on the true marker genes A and B on pooled data that simply combines 3 single-cell populations together for clustering. Even with the correct marker selection, cells from different types are still mixed because of the rotations and scaling. **Separated Clustering**: 2D plot on each individual cell population. With the limited single-cell sample size and skewed cell-type distribution, incorrect marker genes may be selected, shown as genes P, Q and R. **Multitask Clustering and Embedded Feature Selection**: Our proposed method can identify both the true marker genes and cluster the cell types in each population with a multitask learning strategy. The clustering of each dataset is reinforced from the results in the other two datasets shown as the connected clusters across the three experiments.

before sequencing. Biases introduced by PCR are exponentially amplified. Alternative amplification techniques, such as *in vitro* transcription also suffer from transcription inefficiency and sequence drop-out. These biases and noise cause uneven coverage of the entire transcriptome and result in an abundance of zero-coverage regions [28].

In this chapter, we introduce a multitask learning method with an embedded feature selection to capture most the differentially expressed genes among cell clusters across all cell populations to achieve better single-cell clustering simultaneously. The key to doing this is the use of multiple single-cell populations available from biological replicates or related samples with significant biological variances such as samples cultured independently or obtained from different patients. To illustrate the objective, Figure 4.1 shows an example of scRNA-seq data of 100 single cells from three cell populations (n = 33, 33 and 34) with 1000 genes expressed. Of the 1000 genes, genes A and gene B are the hidden markers that are differentially expressed across the four cell types (indicated by the four different colors). In the ideal scenario, there is no technical bias and the marker genes are known as shown in the "Ground Truth" in Figure 4.1. "Simulated Single-cell Populations" in Figure 4.1 shows the single-cell datasets after biological variation, technical biases, and noise are introduced, The data distribution is very different across the three cell populations after the rotation, re-scaling and addition of noise. This makes it challenging to identify the true marker genes with a limited number of samples in each population. Simply pooling the single-cell data from the three populations together will confuse the clustering, even with the correct marker genes identified in "Pooled Clustering"; separated clustering on each single-cell population suffers more from the biological variation as the number of single cells are not sufficient in each individual analysis to identify the true maker genes in "Separated Clustering". We propose a variance-driven multitask clustering of single-cell RNA-seq data (scVDMC) algorithm, as shown in "Multitask Clustering and Embedded Feature Selection" in Figure 4.1, that utilizes expression patterns of different single-cell populations with shared cell-type markers for better integration.

4.2 Method

In this section, we first introduce the model and the algorithm of variance-driven multitask clustering of single cells (scVDMC) and then discuss the parameter selection for scVDMC and related work in single-cell RNA-seq clustering.

4.2.1 A multitask clustering and feature selection model

Assume a total of D domains with each domain representing a single-cell population for clustering. Let matrix $X^{(d)} \in \mathbb{R}^{m \times n^{(d)}}$ denote RNA-seq gene expression values from domain d, where m is the number of features (genes) and $n^{(d)}$ is the single-cell sample size of domain d, $d = 1, 2, \ldots, D$. Let $U^{(d)} \in \mathbb{R}^{m \times k}$ denote the cell-type cluster centers and the binary matrix $V^{(d)} \in \mathbb{Z}_2^{n^{(d)} \times k}$ denote the assignments of each single-cell to the clusters, where k is the number of cell types (clusters) and $\mathbb{Z}_2 = \{0, 1\}$. With the binary vector $B \in \mathbb{Z}_2^{m \times 1}$ denoting the indicators of feature selection (1: selected and 0: not selected) and D_B denoting the diagonal matrix with B on the diagonal, scVDMC model is defined as:

$$
\begin{aligned}
\underset{U^{(d)}, V^{(d)}, B}{\text{minimize}} \quad & \frac{1}{2} \sum_{d=1}^{D} ||D_B(X^{(d)} - U^{(d)} {V^{(d)}}^T)||_F^2 \\
& -w \sum_{d=1}^{D} B^T \mathrm{Var}(U^{(d)}) \\
\text{subject to} \quad & \sum B = \lambda, \\
& \sum_j V_{i,j}^{(d)} = 1, \\
& \forall i = 1, 2, \ldots, n^{(d)}, \quad \forall d = 1, 2, \ldots, D
\end{aligned}
\tag{4.1}
$$

where $w > 0$ is a hyper-parameter to balance the two error terms, the reconstruction error, and the cluster center separation, and $\lambda \in \mathbb{Z}^+$ is the predefined number of features to be selected.

In the model in equation (4.1), $||X^{(d)} - U^{(d)} {V^{(d)}}^T||_F^2$ denotes the reconstruction error of the classic k-means clustering as matrix factorization. Since only a small number of genes are expected as the markers differentiating the cell types, the model restricts the reconstruction error as $||D_B(X^{(d)} - U^{(d)} {V^{(d)}}^T)||_F^2$, where D_B only selects the errors on the selected markers by B. The second term $B^T \mathrm{Var}(U^{(d)})$ is introduced to maximize

the separation of the cluster centers, where $\mathrm{Var}(U^{(d)})$ is defined as a vector where each element is the variance of the vector $U^{(d)}_{i,:} \in \mathbb{R}^{k\times 1}$ [95]. Note that the reconstruction error encourages selection of low expression genes since the errors are smaller on smaller numbers and the second variance terms encourages selection of high expression genes since the variances are larger on larger numbers. Together as the sum over all the domains, the cost function provides a balanced error on the compactness and separation of the clusters of the cell types tuned by feature selection across all the domains. The unique cluster centers in each domain preserves the unique expression patterns while the features are selected as common marker genes for different cell types. For the two hyper-parameters in equation (4.1), λ (the number of marker genes) is typically a small number based on prior knowledge of the cell types, and the selection of balancing weight w is discussed in section 4.2.3.

4.2.2 Alternating updating algorithm

Algorithm 5 scVDMC algorithm

Input: $X^{(d)}, k, w, \lambda, d = 1, 2, \ldots, D$
output: $U^{(d)}, V^{(d)}, B$
Initialize $U^{(d)}$ and $V^{(d)}$.
repeat
    compute B with linear programming in equation (4.7)
    for $d = 1, 2, \ldots, D$ **do**
        solve $V^{(d)}$ by equation (4.2)
        repeat: split the largest cluster if there is an empty cluster
        solve $U^{(d)}$ by (4.6)
    end for
until $U^{(d)}$, $V^{(d)}$ and B converge
return $U^{(d)}$, $V^{(d)}$ and B

The goal is to minimize the cost function in equation (4.1) to obtain the optimal $U^{(d)}$, $V^{(d)}$ and B. We employ an alternating update strategy to solve the optimization problem. First, we fix the feature selection B, all the cluster centers $U^{(d)}$ and all other

$V^{(d)}$ to obtain a certain $V^{(d)}$.

$$\begin{aligned} \underset{V^{(d)}}{\text{minimize}} \quad & \frac{1}{2}||D_B(X^{(d)} - U^{(d)}{V^{(d)}}^T)||_F^2 \\ \text{subject to} \quad & \sum_j V_{i,j}^{(d)} = 1, \quad \forall i = 1, 2, \ldots, n^{(d)}. \end{aligned} \tag{4.2}$$

This is equivalent to assigning samples to the nearest centers $U^{(d)}$ by the Euclidean distance in the features selected by B, where each column of $D_B X^{(d)}$ is a sample and each column of $D_B U^{(d)}$ is a center. Then the distance of a sample to every center is calculated and the nearest center is chosen to assign 1 to the corresponding $V^{(d)}$.

Next, we fix the feature selection B, all clustering assignments $V^{(d)}$, and all other $U^{(d)}$ to solve a certain $U^{(d)}$, rewritten as:

$$\underset{U^{(d)}}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{m} B_i||(X_{i,:}^{(d)} - U_{i,:}^{(d)}{V^{(d)}}^T)||_2^2 - w\sum_{i=1}^{m} B_i \text{Var}(U_{i,:}^{(d)}), \tag{4.3}$$

where $\text{Var}(U_{i,:}^{(d)})$ is the variance of vector $U_{i,:}^{(d)}$ which is defined as

$$\begin{aligned} \text{Var}(U_{i,:}^{(d)}) &= \frac{1}{k}(U_{i,:}^{(d)} - \frac{U_{i,:}^{(d)}\mathbf{1}\mathbf{1}^T}{k})(U_{i,:}^{(d)} - \frac{U_{i,:}^{(d)}\mathbf{1}\mathbf{1}^T}{k})^T \\ &= \frac{1}{k}U_{i,:}^{(d)}(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{k})(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{k})^T{U_{i,:}^{(d)}}^T \\ &= \frac{1}{k}U_{i,:}^{(d)}(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{k}){U_{i,:}^{(d)}}^T, \end{aligned} \tag{4.4}$$

where $\mathbf{I}$ denotes the identity matrix and $\mathbf{1}$ is a column vector of all ones. Let $M \equiv \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{k}$. Then equation (4.3) can be rewritten as:

$$\begin{aligned} \underset{U^{(d)}}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{m} B_i {U_{i,:}^{(d)}}^T({V^{(d)}}^T V^{(d)} - \frac{2wM}{k})U_{i,:}^{(d)} \\ - \quad & \sum_{i=1}^{m} B_i X_{i,:}^{(d)} V^{(d)} {U_{i,:}^{(d)}}^T + \frac{1}{2}\sum_{i=1}^{m} B_i X_{i,:}^{(d)} {X_{i,:}^{(d)}}^T. \end{aligned} \tag{4.5}$$

When w is properly chosen (see section 4.2.3), equation (4.5) is convex and the closed-form solution is

$${U_{i,:}^{(d)}}^T = ({V^{(d)}}^T V^{(d)} - \frac{2wM}{k})^{-1}{V^{(d)}}^T {X_{i,:}^{(d)}}^T. \tag{4.6}$$

Finally, to update binary vector B, we fix all $U^{(d)}$ and $V^{(d)}$ to optimize

$$\begin{aligned} \underset{B}{\text{minimize}} \quad & \sum_{i=1}^{m} B_i \sum_{d=1}^{D} (\frac{1}{2}||(X_{i,:}^{(d)} - U_{i,:}^{(d)} {V^{(d)}}^T)||_2^2 - w\text{Var}(U_{i,:}^{(d)})) \\ \text{subject to} \quad & \sum B = \lambda, \end{aligned} \tag{4.7}$$

which is a standard constrained linear programming problem.

When an empty cluster is created, the calculation of cluster center variance will be invalid. To avoid this, we use a simple splitting procedure to handle empty clusters. Specifically, if there is an empty cluster in $V^{(d)}$ (i.e. the whole column is 0) we randomly split the largest cluster into two clusters. This procedure is repeated until there are exactly k clusters. The full scVDMC algorithm is shown in Algorithm 5.

4.2.3 Upper bound of parameter w

Equation (4.5) is a sum of a few quadratic terms of variable $U_{i,:}^{(d)}$. The global minimum of $U_{i,:}^{(d)}$ can be solved in closed-form if the Hessian below is positive semi-definite,

$$H = {V^{(d)}}^T V^{(d)} - \frac{2wM}{k}. \tag{4.8}$$

In the following, we show that an upper bound on w will guarantee that H is positive semi-definite. By Gershgorin circle theorem[1] , the sufficient condition of $H \succeq 0$ is $H_{ii} - \sum_{j \neq i} |H_{ij}| \geq 0$ for $\forall i$. This is equivalent to stating that H is diagonally dominant and only has non-negative diagonal entries. H can be rewritten as follows,

$$\begin{aligned} H_{ii} &= c_i + \frac{2w(1-k)}{k^2}, \ \forall i = 1, ..., k \\ H_{ij} &= \frac{2w}{k^2}, \ \forall i \neq j, \end{aligned}$$

where c_i is the i^{th} diagonal entry of matrix ${V^{(d)}}^T V^{(d)}$, i.e., the size of cluster i. Then we have

$$c_i + \frac{2w(1-k)}{k^2} \geq \frac{2w(k-1)}{k^2}$$

[1] For any eigenvalue δ of matrix H, $|\delta - H_{ii}| \leq \sum_{j \neq i} |H_{ij}|$ for $\forall i \iff H_{ii} - \sum_{j \neq i} |H_{ij}| \leq \delta \leq H_{ii} + \sum_{j \neq i} |H_{ij}|$.

and thus,

$$w \leq \frac{k^2 c_i}{4(k-1)} \leq \frac{k^2 c_{min}}{4(k-1)},$$

where c_{min} as the minimum of c_i, $\forall i = 1, ..., k$. Since $c_{min} \geq 1$, we obtain a loose upper bound of $w = \frac{k^2}{4(k-1)}$. In all the experiments, we set w to be smaller than the upper bound for feasible implementation.

4.2.4 Related work

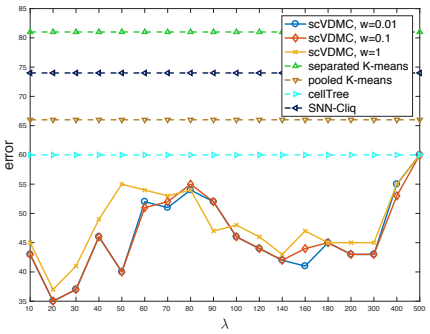
Most existing methods focus only on sub-population clustering and differential gene expression detection among the learned cell clusters with one (pooled) cell population. Some of these methods were directly adopted from traditional bulk RNA-seq analysis and/or classical dimension reduction algorithms such as Principal Component Analysis [33–35], hierarchical clustering [36], t-SNE [37–39], Independent Component Analysis [40] and Multi-dimensional Scaling [41]. Other methods focus on special properties of scRNA-seq data, such as high variance and uneven expressions. For example, SNN-Cliq [42] uses a ranking measurement to get reliable results on high dimensional data; [43] proposed a special dimension reduction method to handle the large amount of zeros in scRNA-seq; [44] proposed a Latent Dirichlet Allocation model with latent gene groups to measure cell-to-cell distance.

Mixed multiple batch strategy has been proposed [36, 45] to reduce the technical variance, which does not directly improve clustering. To the best of our knowledge, multitask clustering with an embedded feature selection has not been previously applied to scRNA-seq data analysis.

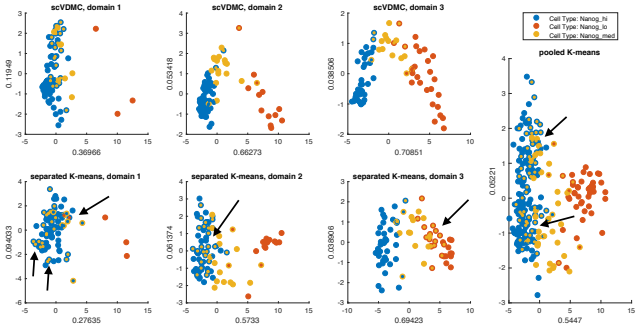
4.3 Experiments

We applied scVDMC to two existing scRNA-seq datasets and compared the clustering results with four baseline methods: (1) k-means clustering on each domain separately, (2) pooling all domains and applying k-means clustering, (3) SNN-Cliq [42], and (4) CellTree [44]. Pooled k-means (2) was used to obtain the initialization for scVDMC.

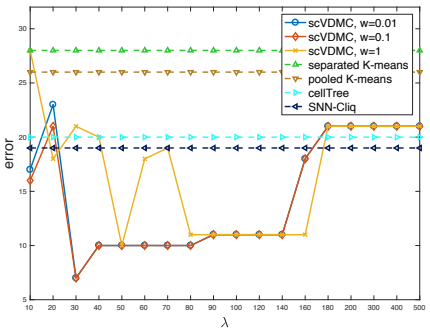
To apply the SNN-Cliq method in (3), we used the provided MATLAB code to transform the data into the SNN graph, then used the Python code to produce the



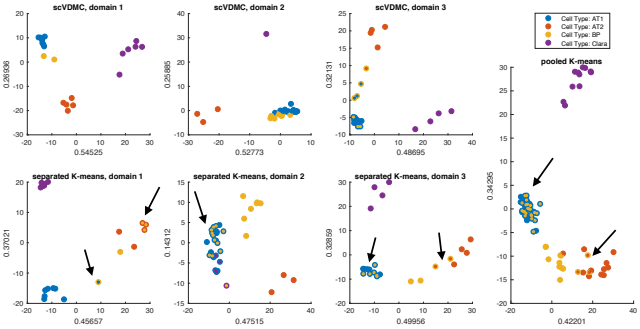
(A) Clustering on mESC dataset



(B) Clustering errors on mESC dataset



(C) Clustering on lung dataset



(D) clustering errors on lung dataset

Figure 4.2: **Clustering performance on mESC and lung epithelial scRNA-seq datasets**. (A) & (C) show the clustering results of the scVDMC algorithm with varying numbers of selected marker genes compared with the four baseline methods. (B) & (D) show the PCA of scVDMC, pooled k-means, and separated k-means results on the selected top 20 marker genes. For each dot, the layer (outer) color indicates the true cell type, while the inner color indicates the predicted cell type. The hyper-parameters for scVDMC are $\lambda = 20, w = 0.1$ on the mESC dataset and $\lambda = 30, w = 0.1$ on the lung dataset.

clustering result by ranking measurement. There are three hyper-parameters: k (size of the nearest neighbor list), r (parameter for quasi-clique finding, range (0,1]), and m (parameter for cluster merging range (0,1]). We tested multiple combinations of the three hyper-parameters using $k = 3, 5, 7$, $r = 0.1, 0.2, \ldots, 0.9$ and $m = 0.1, 0.2, \ldots, 0.9$. We also required the program to annotate all the data instead of leaving singletons unlabeled ($-n$). Since SNN-Cliq identifies the number of clusters automatically, we only reported the results with the correct number of clusters.

To apply the CellTree method in (4), we used the provided R package to first fit a Latent Dirichlet Allocation (LDA) model with the default method (joint MAP estimation) to choose the number of topics followed by learning a pair-wise distance for all cells. Then we ran hierarchical clustering with four different methods for computing cluster distance ('ward', 'complete', 'single', 'average') and selected the best clustering results.

For baselines (1) and (2), we followed a similar idea to choose marker genes. After clustering, we chose the genes with large variance among the clusters as markers. Since (3) and (4) use a different strategy for clustering and do not provide marker-gene selection, we only focused on the clustering result for these two baselines.

4.3.1 Mouse embryonic stem cell (mESC) dataset

We downloaded the single-cell expression data for 250 mESCs [96] from the European Bioinformatics Institute's (EBI) ESpresso database. These 250 mESCs were cultured in serum conditions and were captured using the Fluidigm C1 on three different days from three different passages (biological replicates, n = 81, 90, and 79). After removing genes expressed uniformly within a single replicate, 12,114 genes remained. For the SNN-Cliq method, we further removed genes with an average expression less than 20 and log-transformed the data, as recommended in [42].

Figure 4.2(A) shows the clustering results. Compared with the four baselines, scVDMC shows a consistently lower error with different choices of λs. Within a reasonable range of λ, such as from 20 to 300, scVDMC shows significant improvement compared with the other baseline methods. When λ is too small, such as 10 genes selected, there are not enough markers to capture the difference among the cell types so the error is larger. When λ is too big, scVDMC will consider almost all the genes and the variance selection will not play a role. As such, scVDMC will eventually degrade into separated k-means and the error will also increase. It is worth noting that the results are not sensitive to the parameter w, for which the upper bound for w is $\frac{9}{8}$ in this case. It is also interesting that the CellTree method performed better than pooled and separated k-means, while SNN-Cliq performed better than separated k-means but worse than pooled k-means. Figure 4.2(B) shows the detailed clustering results by scVDMC, pooled k-means and separated k-means. Compared with the pooled k-means

and separated k-means, scVDMC captures relatively high variance in the leading principle components and achieves improved clustering in every domain (fewer mixed-color dots).

Analysis of the mESC transcriptome data using scVDMC yielded comparable results on marker gene selection to the hierarchical clustering in the original paper as well as pooled and separated k-means. Both analyses were able to detect and highly rank the known markers for differentiation Krt8, Krt18, Anxa1, Anxa2, Anxa3, Acta1, and Acta2. Further, scVDMC detected several additional genes that pooled k-means, separated k-means and the original paper did not. These included Dppa5a, a core pluripotency gene for mESCs [97] and Igf2, a growth factor that promotes endothelial differentiation in embryonic stem cells [98].

4.3.2 Experiment on lung epithelial single-cell data

We downloaded the single-cell expression data for 80 embryonic mouse lung epithelial cells [99]. These 80 single-cell samples were taken from three different mice (biological replicates, n = 20, 34, and 23) and contained five cell types: ciliated, Clara, AT1, and AT2 cells, as well as a bi-potential progenitor (BP). Since only one replicate contained ciliated cells, we removed these from the analysis, leaving 77 single-cell samples. After removing genes expressed uniformly within a single replicate, 7,357 genes remained. For the SNN-Cliq method, we further removed genes with log-transformed average expression less than 2.

With the limited number of single-cell samples in this dataset, scVDMC still improved clustering over the baselines in the range of $\lambda \in [30, 100]$ shown in figure 4.2(C). In Figure 4.2(D), PCA plots of the top 30 genes show a trend similar to the ESC dataset, where scVDMC's top genes capture more variance and show less clustering error. Both SNN-Cliq and CellTree performed better than pooled k-means and separated k-means, with SNN-Cliq leading CellTree by a very small margin.

Analysis of the mouse lung epithelial transcriptome data using scVDMC yielded comparable results to the hierarchical clustering in the original paper as well as pooled and separated k-means. Both analyses were able to detect and highly rank the known marker genes of the different cell types: Clara (Scgb1a1), AT1 (Pdpn, Ager), and AT2 (Sftpc, Sftpb). Further, scVDMC detected several additional genes that pooled k-means,

separated k-means and the original paper did not. These included several components of the Notch signaling pathway (Notch1, Jag1, and Nrarp) previously shown to be critical for development of lung alveolar spaces, with AT2 cells being major sites of Notch activation [100].

4.4 Analysis of RDEB scRNA-seq data

Recessive Dystrophic Epidermolysis Bullosa (RDEB) is an inherited blistering disorder caused by loss-of-function mutations in the *COL7A1* gene that codes for type VII collagen (C7) [101]. C7 forms the anchoring fibrils that attach the epidermis to the dermis [102]. When C7 is missing, the skin becomes extremely fragile, eroding at the slightest touch. From birth, patients with this disease must undergo intensive bandaging and daily wound care. They are also susceptible to a highly aggressive form of squamous cell carcinoma. [103–106]. It has been shown that allogeneic hematopoeitic cell transplant (HCT) can partially rescue the RDEB phenotype. Cells from the bone marrow home to the skin and deposit C7 at the dermal-epidermal junction, greatly improving skin integrity in a subset of patients [107]. However, the molecular mechanism by which this occurs remains unknown.

To identify sub-populations producing homing signals that could attract bone marrow-derived cells to injured skin, we captured single dermal fibroblasts from patients with severe generalized RDEB and their HLA-matched healthy siblings using the Fluidigm C1 system. In total, 295 patient cells and 248 sibling cells were captured and sequenced. Paired-end 75bp reads were mapped to the UCSC human transcriptome (hg19) using Bowtie 2 (version 2.2.4) and Tophat (version 2.0.9). Gene expression levels were calculated using Cuffquant (Cufflinks version 2.2.1 with parameters -u -max-bundle-frags 10000000) and Cuffnorm (Cufflinks version 2.2.1). FPKM values as estimated by Cufflinks were added a value of 1 (to avoid zeros) and log-transformed. We excluded low-expressed genes (average log2 (FPKM) < 1.5) from further analysis. All of our samples met the requirement of expressing at least 2,000 of these remaining 5,196 genes.

Applying scVDMC to our RDEB single-cell dataset identified several top 100 genes previously known to be involved in RDEB (Figure 3). These included *CXCL12/SDF1*,

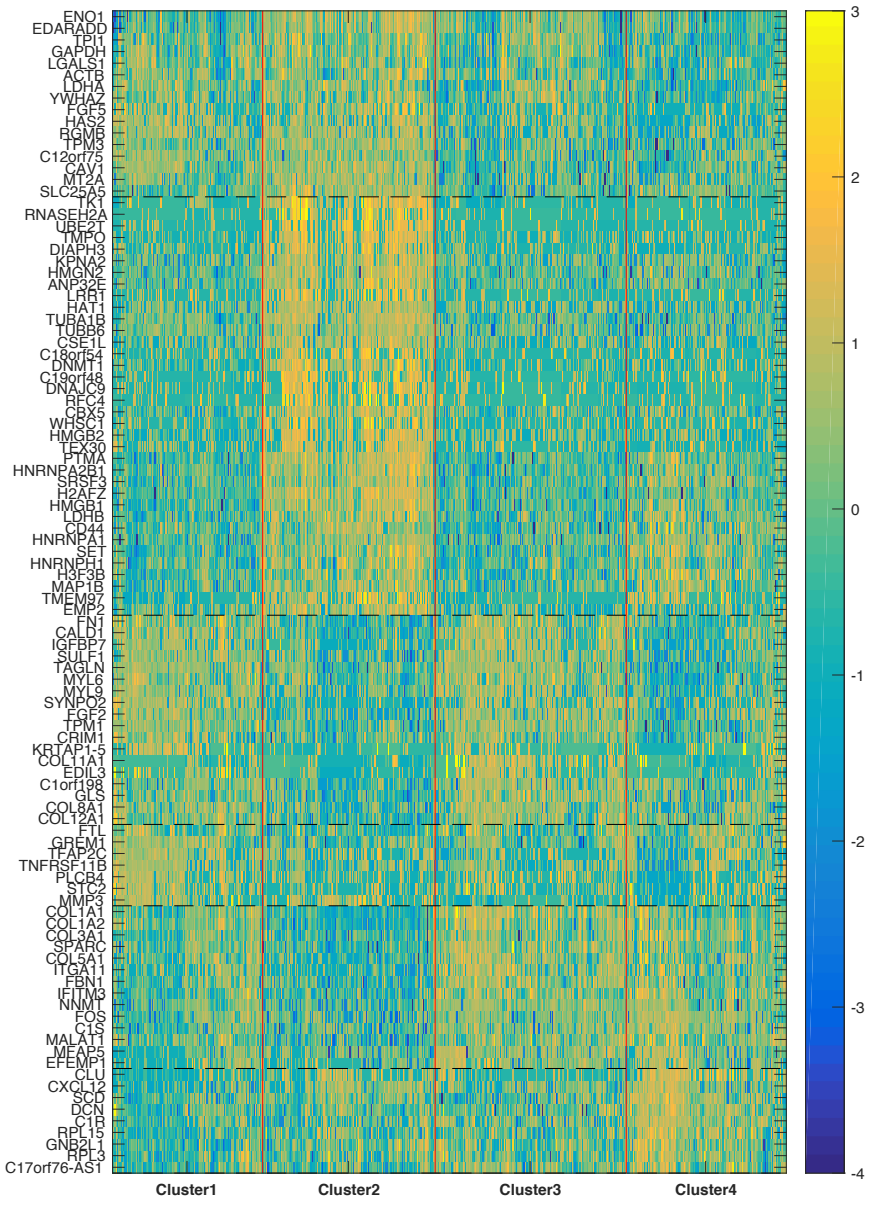


Figure 4.3: **Single-cell sample clustering by 100 markers genes on the RDEB data with scVDMC.** The solid vertical red lines separate the cell clusters and the black dashed horizontal lines indicate marker gene clusters derived by hierarchical clustering.

the ligand for *CXCR4*, which directs cells of the bone marrow to damaged tissue including skin [108] and *HMGB1*, which has shown to be positively correlated with RDEB severity [109] and also mediates recruitment of bone marrow-derived cells to injured tissue [110]. Note that we empirically removed confounding cell cycle genes from the top 100 predicted markers and repeated scVDMC until there were no selected cell cycle genes.

We also identified several genes as markers not previously associated with RDEB. These included *COL11A1*, a minor fibrillar collagen shown to mark activated cancer-associated fibroblasts (CAFs) that is not typically expressed in fibroblasts associated with inflammation and fibrosis [111]. scVDMC also revealed *GREM1*, a BMP antagonist associated with renal and pancreatic fibrosis [112, 113] and *MFAP5*, which promotes attachment of cells to micro-fibrils of the extracellular matrix and interacts with TGBβ growth factors [114]. We performed flow cytometry on the same RDEB patient and matched sibling fibroblasts to validate the expression levels of these genes at the single-cell level and found the results similar to our RNA expression data (Figure 4). As top hits, these genes potentially mark sub-populations of stromal cells that contribute to the transformation of the overlying epithelium and the development of squamous cell carcinoma in RDEB patients.

It is also possible to apply other multitask learning or transfer learning methods [48] for the clustering tasks. scVDMC is a multitask clustering method specifically designed for scRNA-seq data for selection of a smaller set of cell-type markers and allows large variability in gene expression across the cell populations. Other methods are often built using different assumptions of the data that might not be applicable to the characteristics of scRNA-seq populations [49, 56, 57].

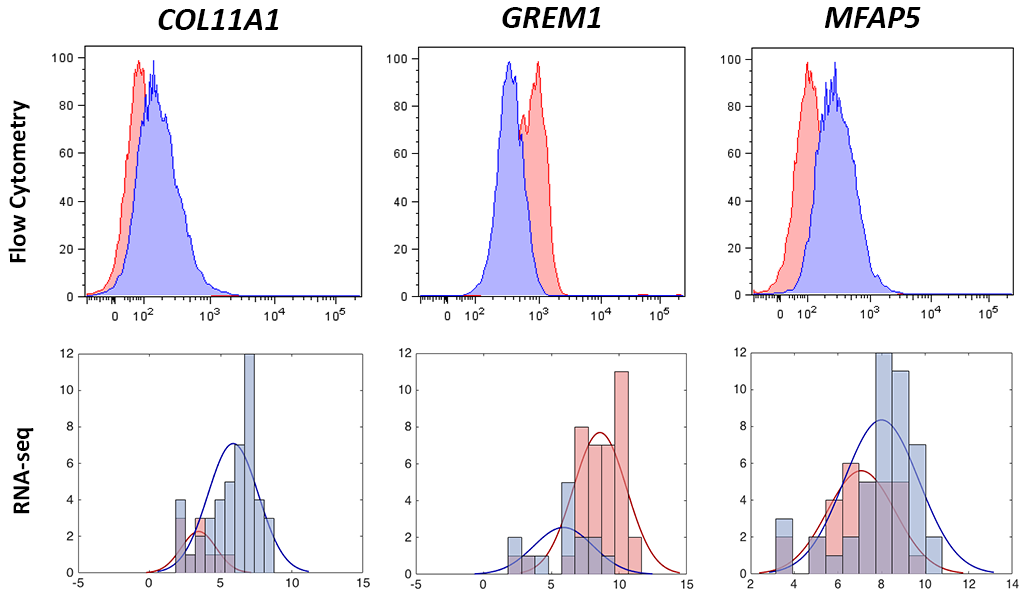


Figure 4.4: **Validation of the novel markers by flow cytometry.** The distribution of expressions for novel genes was similar between flow cytometry experiments (top) and the single-cell RNA-seq data (bottom) for the genes *COL11A1, GREM1,* and *MFAP5.* RDEB patient single-cells are shown in red; matched sibling single-cells are shown in blue. Flow cytometry data are measured as percent of max; RNA-seq data measured in FPKMs.

Chapter 5

Conclusion and Discussion

5.1 Conclusion

Biomarker detection is important as it provides better prognosis and diagnosis for disease and further understanding on population genetics. Learning biomarkers among subgroups is not only more accurate and robust as false positive signal are reduced by only considering similar samples/patients/cells, but it also has advanced biological meaning: for population study, accurate detection of population-differentiation CNV can depict human evolutionary relations; for cancer research, causative and potential therapeutic strategies can be systematic learned from tumor stage specific genes; on single-cell level, understanding cell type specific biomakers provides chance to characterize subpopulation structure, understand disease progression and mechanisms of transcription regulation. However, there are still challenges, which described in introduction section. In this thesis, we have presented several structured latent feature based and multitask learning based methods to address these challenges and achieved improvement on biomarker detection among subgroups for both CNV data and scRNA-seq data.

Our previous work *SubPatCNV* [16] demonstrates that population specific CNV patterns widely exist. Therefore, we developed a structured latent feature based method for high accurate CNV pattern detection for population studies. This tree-guided machine learning algorithm *treeSGS* is able to detect population-differentiation CNVs among population subgroups and the population subgroups are organized by a phylogenetic tree of human populations. This algorithm dynamically splits populations into

groups and sparsely selects subgroup-specific latent CNV features. Those learned latent CNV features exhibit the preserved CNV patterns from the ancestral population. Experiments on Hapmap3 dataset with 11 populations show that, comparing with method that not using hierarchical tree structure and method without sparse selection on sample subgroups, our proposed $treeSGS$ model achieves high accuracy on detecting a list of candidate AIM CNV markers that not only are population-differentiation but also depict the evolutionary relations among the populations.

Then, to study CNVs across different cancer domains, we proposed $TLFL$ algorithm, which uses latent features model combined with transfer learning technic. In this model, each cancer type can be regarded as one domain in transfer learning and common latent CNV features are used as a bridge to transfer knowledge among different cancer domains while domain-specific components are preserved for each cancer type to explain the heterogeneity of each cancer. Fused lasso is also applied on each latent CNV features to preserve the sparsity and block structure of CNV patterns. Experiments on cross cancer type study show that $TLFL$ is more accurate on detect cancer related CNVs comparing with non-transfer-learning model.

Finally, we proposed a multitask learning method with an embedded feature selection ($scVDMC$) on single-cell RNA sequencing data. This algorithm was specifically designed for scRNA-seq data with multiple samples or multiple experiments by capturing the most differentially expressed genes among cell type clusters across all the domains. Since sample heterogeneity and experiment bias play a big role in the divergence of RNA expression on single cells, $scVDMC$ utilizes a variance-driven multitask clustering method to capture shared cell-type bio-markers. The experiments on two real single-cell RNA-seq datasets with several replicates show that $scVDMC$ detected more accurate cell populations and known cell markers than pooled clustering and several other recently proposed scRNA-seq clustering methods. Experiment on in-house Recessive Dystrophic Epidermolysis Bullosa (RDEB) scRNA-seq data also revealed several interesting cell types and markers that were previously unknown.

In summary, all models proposed in this thesis showed promising results in both simulations and experiments on CNV and scRNA-seq data. The proposed algorithms are useful computational tools for population research and disease studies.

5.2 further work

In this section, we will discuss some limitation of our current methods and propose several further work and directions.

5.2.1 treeSGS

Human phylogenetic tree depicts the relations among populations in a hierarchical way. Our proposed *treeSGS* method utilizes this structure as prior information and successfully learns tree-constraint-subgroup specific CNV patterns.

One potential direction to expend this idea is to cooperate more detailed sample relation as prior knowledge to guide CNV learning, such as the father-mother-child trio information, which is available in Hapmap data and 1000 Genome Project. These additional information could either be combined with phylogenetic tree to build more complicated tree structure, or separately applied as additional constraint on coefficient matrix. Keep in mind that if the tree structure is getting complicated, the tree split hyper-parameter would play a more important role in identifying subgroups and therefore fine tuning would be needed to get a balance between easy explanation and reserving low level sample relations.

Even through this method is developed for population genetic study, it is also possible to be used for pan-cancer CNV research. Currently, the major obstacle for utilizing *treeSGS* on pan-cancer study is that reliable cancer type relationship is not available. It is not a trivial problem to get pan-cancer relationship as the knowledge of functional impacts of CNVs on cancers is quite fragmented and pathophysiological role are not fully understand. To overcome this, One solution is to use partially known pan-cancer relations but leave the unknown or uncertain cancer relation unconstraint by always splitting among them in *treeSGS*. Modified algorithm will be needed to address this situation.

5.2.2 TLFL

Application of transfer learning on CNV analysis across multiple cancer types is promising since CNVs are a hallmark of cancer genomes. $TLFL$ enables sharing information in datasets of different cancer domains to discover latent CNV features that can explain

common and domain-specific cancer characteristics and better classify patient samples. It is expected that transfer learning will play an important role in the comparative analysis of the large patient cohorts to improve the current knowledge of cancer development and progression in the light of both common and specific cancer CNVs.

However, there is limitation of $TLFL$ model on handling large number of cancer types. This is because the definition of "common" latent CNV features in $TLFL$ is very strict: they have to be shown in all the cancer type domains. As the number of cancer types to be learned in the model increasing, the number of "common" latent features will decrease. For example, on a study with 10 different cancer types, there may be no latent CNV features across all of them, but there are CNV regions shared between breast cancer and ovarian cancer, such as regions related with gene $BRCA1$. Without utilizing these "partially common" relationship will decrease the power of accurate CNV learning. To solve this problem, the aforementioned $treeSGS$ model could be a solution as long as cancer type relations are depicted. Another possible strategy is applying $TLFL$ multiple times, with each time solving a few related cancer types. The overall optimization problem is the summation of each $TLFL$ subproblem.

Combining prior-known clinical group information could be another direction to improve $TLFL$ method for accurate biomarker detection. Currently $TLFL$ method doesn't assume any sample groups within each cancer domain. With additional sample information in each domain available, such as tumor stage or tumor grade, latent CNV features could be identified not only domain specific or common among domains, but also assigned to specific group(s) within certain domain. Below is a possible framework,

which combines $TLFL$ with $SGS - FL$ [60]:

$$\begin{aligned}
&\min \sum_{d=1}^{\delta} (\frac{1}{2}||X_d - [\hat{U}, U_d]V_d||_F^2 \\
&+\lambda_d \sum_{j=1}^{k-\tau}\sum_{i=1}^{m} |U_{d(i,j)}| + \gamma_d \sum_{j=1}^{k-\tau}\sum_{l=2}^{m} |U_{d(l,j)} - U_{d(l-1,j)}|) \\
&+\lambda_C \sum_{j=1}^{\tau}\sum_{i=1}^{m} |\hat{U}_{(i,j)}| + \gamma_C \sum_{j=1}^{\tau}\sum_{l=2}^{m} |\hat{U}_{(l,j)} - \hat{U}_{(l-1,j)}|, \\
&s.t. \\
&V_d \geq \mathbf{0} \text{ and } V_{d(i,:)}V_{d(i,:)}^T = 1 \text{ for } i = 1, 2, \ldots, k, \\
&V_{d(i,g_l^d)} \times b_{d(i,g_l^d)} = 0 \text{ for } i = 1, 2, \ldots, k,
\end{aligned} \tag{5.1}$$

where g_l^d is the selected samples from group l on domain d.

5.2.3 scVDMC

There are limitations in the $scVDMC$ method. In multitask clustering, assuming a global k as the number of clusters in each cell population dataset does not always hold true as for some rare cell types, the corresponding cells may only be present in some populations. $scVDMC$ might incorrectly split a cluster of one cell type because no empty cluster is allowed. One possible improvement is to model each domain with an individual $k^{(d)}$ with a more adaptive strategy for choosing $k^{(d)}$. In this case, the overall balance between within-cluster distance and the variance will need to be more carefully weighted. In addition, cell cycle-related genes could be a large source of confounders. Unless the stages of cell cycle are the biological signal under study, cell cycle-related variation could obscure biological signals of interest. It is possible to model the confounders directly in the $scVDMC$ method with complex modeling. Alternatively, we could pre-process the scRNA-seq data to remove the cell cycle signals. For example, a Gaussian processes-based latent-variable model [115] was used to account for confounding variations due to the cell cycle in scRNA-seq data sets and then linear regression was applied to remove them. In this approach, a clearly defined cell cycle gene set is necessary to avoid removing true signals unexpectedly.

References

[1] 1000 Genomes Project Consortium et al. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073, 2010.

[2] Laurent Excoffier, Guillaume Laval, and Stefan Schneider. Arlequin (version 3.0): an integrated software package for population genetics data analysis. *Evolutionary bioinformatics*, 1, 2005.

[3] John W Davey, Paul A Hohenlohe, Paul D Etter, Jason Q Boone, Julian M Catchen, and Mark L Blaxter. Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nature Reviews Genetics*, 12(7):499–510, 2011.

[4] Barbara Weir, Xiaojun Zhao, and Matthew Meyerson. Somatic alterations in the human cancer genome. *Cancer cell*, 6(5):433–438, 2004.

[5] Lynda Chin and Joe W Gray. Translating insights from the cancer genome into clinical practice. *Nature*, 452(7187):553–563, 2008.

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

[7] Christos Sotiriou, Soek-Ying Neo, Lisa M McShane, Edward L Korn, Philip M Long, Amir Jazaeri, Philippe Martiat, Steve B Fox, Adrian L Harris, and Edison T Liu. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences*, 100(18):10393–10398, 2003.

[8] Markus Heidenblad et al. Tiling resolution array CGH and high density expression profiling of urothelial carcinomas delineate genomic amplicons and candidate target genes specific for advanced tumors. *BMC Medical Genomics*, 1, JAN 31 2008.

[9] Torsten O Nielsen, Forrest D Hsu, Kristin Jensen, Maggie Cheang, Gamze Karaca, Zhiyuan Hu, Tina Hernandez-Boussard, Chad Livasy, Dave Cowan, Lynn Dressler, et al. Immunohistochemical and clinical characterization of the basal-like subtype of invasive breast carcinoma. *Clinical cancer research*, 10(16):5367–5374, 2004.

[10] Felix Schmidt and Thomas Efferth. Tumor heterogeneity, single-cell sequencing, and drug resistance. *Pharmaceuticals*, 9(2):33, 2016.

[11] Daniel Hebenstreit. Methods, challenges and potentials of single cell rna-seq. *Biology*, 1(3):658–667, 2012.

[12] Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Naama Elefant, Franziska Paul, Irina Zaretsky, Alexander Mildner, Nadav Cohen, Steffen Jung, Amos Tanay, et al. Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. *Science*, 343(6172):776–779, 2014.

[13] Serena Liu and Cole Trapnell. Single-cell transcriptome sequencing: recent advances and remaining challenges. *F1000Research*, 5, 2016.

[14]

[15] Peter H Sudmant, Swapan Mallick, Bradley J Nelson, Fereydoun Hormozdiari, Niklas Krumm, John Huddleston, Bradley P Coe, Carl Baker, Susanne Nordenfelt, Michael Bamshad, et al. Global diversity, population stratification, and selection of human copy-number variation. *Science*, 349(6253):aab3761, 2015.

[16] Nicholas Johnson, Huanan Zhang, Gang Fang, Vipin Kumar, and Rui Kuang. Subpatcnv: approximate subspace pattern mining for mapping copy-number variations. *BMC Bioinformatics*, 16(1), 2014.

[17] Richard A Gibbs, John W Belmont, Paul Hardenbol, Thomas D Willis, Fuli Yu, Huanming Yang, Lan-Yang Ch'ang, Wei Huang, Bin Liu, Yan Shen, et al. The international hapmap project. *Nature*, 426(6968):789–796, 2003.

[18] M. Baudis. Genomic imbalances in 5918 malignant epithelial tumors: an explorative meta-analysis of chromosomal CGH data. *BMC cancer*, 7(1):226, 2007.

[19] F. Mitelman, B. Johansson, and F. Mertens. *Mitelman database of chromosome aberrations in cancer*. Cancer Genome Anatomy Project., 2007.

[20] Lars Feuk, Andrew R Carson, and Stephen W Scherer. Structural variation in the human genome. *Nature Reviews Genetics*, 7(2):85–97, 2006.

[21] Andrea Sottoriva, Inmaculada Spiteri, Sara GM Piccirillo, Anestis Touloumis, V Peter Collins, John C Marioni, Christina Curtis, Colin Watts, and Simon Tavaré. Intratumor heterogeneity in human glioblastoma reflects cancer evolutionary dynamics. *Proceedings of the National Academy of Sciences*, 110(10):4009–4014, 2013.

[22] R Fisher, L Pusztai, and C Swanton. Cancer heterogeneity: implications for targeted therapeutics. *British journal of cancer*, 108(3):479–485, 2013.

[23] Ehud Shapiro, Tamir Biezuner, and Sten Linnarsson. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics*, 14(9):618–630, 2013.

[24] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann. The technology and biology of single-cell rna sequencing. *Molecular cell*, 58(4):610–620, 2015.

[25] Brian Munsky, Gregor Neuert, and Alexander van Oudenaarden. Using gene expression noise to understand gene regulation. *Science*, 336(6078):183–187, 2012.

[26] Hirofumi Shintaku, Hidekazu Nishikii, Lewis A Marshall, Hidetoshi Kotera, and Juan G Santiago. On-chip separation and analysis of rna and dna from single cells. *Analytical chemistry*, 86(4):1953–1957, 2014.

[27] James Eberwine, Jai-Yoon Sul, Tamas Bartfai, and Junhyong Kim. The promise of single-cell sequencing. *Nature methods*, 11(1):25–27, 2014.

[28] Rhonda Bacher and Christina Kendziorski. Design and computational analysis of single-cell rna-sequencing experiments. *Genome biology*, 17(1):63, 2016.

[29] G. Nowak, T. Hastie, J.R. Pollack, and R. Tibshirani. A fused lasso latent feature model for analyzing multi-sample aCGH data. *Biostatistics*, 12(4):776–791, 2011.

[30] Feng Zhang, Wenli Gu, Matthew E Hurles, and James R Lupski. Copy number variation in human health, disease, and evolution. *Annual review of genomics and human genetics*, 10:451, 2009.

[31] Mehdi Zarrei, Jeffrey R MacDonald, Daniele Merico, and Stephen W Scherer. A copy number variation map of the human genome. *Nature Reviews Genetics*, 16(3):172–183, 2015.

[32] Rameen Beroukhim, Craig H Mermel, Dale Porter, Guo Wei, Soumya Raychaudhuri, Jerry Donovan, Jordi Barretina, Jesse S Boehm, Jennifer Dobson, Mitsuyoshi Urashima, et al. The landscape of somatic copy-number alteration across human cancers. *Nature*, 463(7283):899–905, 2010.

[33] Saiful Islam, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnerberg, and Sten Linnarsson. Characterization of the single-cell transcriptional landscape by highly multiplex rna-seq. *Genome research*, 21(7):1160–1167, 2011.

[34] Alex K Shalek, Rahul Satija, Joe Shuga, John J Trombetta, Dave Gennert, Diana Lu, Peilin Chen, Rona S Gertner, Jellert T Gaublomme, Nir Yosef, et al. Single-cell rna-seq reveals dynamic paracrine control of cellular variation. *Nature*, 510(7505):363–369, 2014.

[35] Alex K Shalek, Rahul Satija, Xian Adiconis, Rona S Gertner, Jellert T Gaublomme, Raktima Raychowdhury, Schraga Schwartz, Nir Yosef, Christine Malboeuf, Diana Lu, et al. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453):236–240, 2013.

[36] Liying Yan, Mingyu Yang, Hongshan Guo, Lu Yang, Jun Wu, Rong Li, Ping Liu, Ying Lian, Xiaoying Zheng, Jie Yan, et al. Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature structural & molecular biology*, 20(9):1131–1139, 2013.

[37] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.

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

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

[40] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381–386, 2014.

[41] Tsukasa Kouno, Michiel de Hoon, Jessica C Mar, Yasuhiro Tomaru, Mitsuoki Kawano, Piero Carninci, Harukazu Suzuki, Yoshihide Hayashizaki, and Jay W Shin. Temporal dynamics and transcriptional control using single-cell gene expression analysis. *Genome biology*, 14(10):R118, 2013.

[42] Chen Xu and Zhengchang Su. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, page btv088, 2015.

[43] Emma Pierson and Christopher Yau. Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology*, 16(1):241, 2015.

[44] Sohiya Yotsukura, Seitaro Nomura, Hiroyuki Aburatani, Koji Tsuda, et al. Celltree: an r/bioconductor package to infer the hierarchical structure of cell populations from single-cell rna-seq data. *BMC bioinformatics*, 17(1):363, 2016.

[45] Stephanie C Hicks, Mingxiang Teng, and Rafael A Irizarry. On the widespread and critical impact of systematic bias and batch effects in single-cell rna-seq data. *bioRxiv*, page 025528, 2015.

[46] Jean-Philippe Vert and Kevin Bleakley. Fast detection of multiple change-points shared by many signals using group lars. In *Advances in Neural Information Processing Systems*, pages 2343–2351, 2010.

[47] Kevin Bleakley and Jean-Philippe Vert. The group fused lasso for multiple change-point detection. *arXiv preprint arXiv:1106.4199*, 2011.

[48] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[49] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. In *Proceedings of the 25th international conference on Machine learning*, pages 200–207. ACM, 2008.

[50] Andreas Argyriou, Charles A Micchelli, Massimiliano Pontil, and Yiming Ying. A spectral regularization framework for multi-task structure learning, nips 20. *Journal Publications on Mathematics (Harmonic Analysis)*, 2008.

[51] Su-In Lee, Vassil Chatalbashev, David Vickrey, and Daphne Koller. Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of the 24th international conference on Machine learning*, pages 489–496. ACM, 2007.

[52] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. *Advances in neural information processing systems*, 19:41, 2007.

[53] Tony Jebara. Multi-task feature and kernel selection for svms. In *Proceedings of the twenty-first international conference on Machine learning*, page 55. ACM, 2004.

[54] Ulrich Rückert and Stefan Kramer. Kernel-based inductive transfer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 220–233. Springer, 2008.

[55] Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th international conference on Machine learning*, pages 1120–1127. ACM, 2008.

[56] Zheng Wang, Yangqiu Song, and Changshui Zhang. Transferred dimensionality reduction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 550–565. Springer, 2008.

[57] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[58] Mattias Jakobsson, Sonja W Scholz, Paul Scheet, J Raphael Gibbs, Jenna M VanLiere, Hon-Chung Fung, Zachary A Szpiech, James H Degnan, Kai Wang, Rita Guerreiro, et al. Genotype, haplotype and copy-number variation in worldwide human populations. *Nature*, 451(7181):998–1003, 2008.

[59] Ravi Sachidanandam, David Weissman, Steven C Schmidt, Jerzy M Kakol, Lincoln D Stein, Gabor Marth, Steve Sherry, James C Mullikin, Beverley J Mortimore, David L Willey, et al. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409(6822):928–933, 2001.

[60]

[61] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[62] Xi Chen, Qihang Lin, Seyoung Kim, Jaime G Carbonell, Eric P Xing, et al. Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, 6(2):719–752, 2012.

[63] Jun Liu, Shuiwang Ji, Jieping Ye, et al. Slep: Sparse learning with efficient projections. *Arizona State University*, 6:491, 2009.

[64] International HapMap 3 Consortium et al. Integrating common and rare genetic variation in diverse human populations. *Nature*, 467(7311):52–58, 2010.

[65] Tae-Ho Lee, Hui Guo, Xiyin Wang, Changsoo Kim, and Andrew H Paterson. Snphylo: a pipeline to construct a phylogenetic tree from huge snp data. *BMC genomics*, 15(1):162, 2014.

[66] Catarina D Campbell, Nick Sampas, Anya Tsalenko, Peter H Sudmant, Jeffrey M Kidd, Maika Malig, Tiffany H Vu, Laura Vives, Peter Tsang, Laurakay Bruhn, et al. Population-genetic properties of differentiated human copy-number polymorphisms. *The American Journal of Human Genetics*, 88(3):317–332, 2011.

[67] Jeffrey R MacDonald, Robert Ziman, Ryan KC Yuen, Lars Feuk, and Stephen W Scherer. The database of genomic variants: a curated collection of structural variation in the human genome. *Nucleic acids research*, page gkt958, 2013.

[68] D. Pinkel et al. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature genetics*, 20(2):207–211, 1998.

[69] D. Pinkel and D.G. Albertson. Array comparative genomic hybridization and its applications in cancer. *Nature genetics*, 37:S11–S17, 2005.

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

[71] ES Venkatraman and A.B. Olshen. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, 23(6):657–663, 2007.

[72] J. Fridlyand, A.M. Snijders, D. Pinkel, D.G. Albertson, and A.N. Jain. Hidden markov models approach to the analysis of array CGH data. *Journal of multivariate analysis*, 90(1):132–153, 2004.

[73] S. Stjernqvist, T. Rydén, M. Sköld, and J. Staaf. Continuous-index hidden markov modelling of array CGH copy number data. *Bioinformatics*, 23(8):1006–1014, 2007.

[74] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.J. Daudin. A statistical approach for array CGH data analysis. *BMC bioinformatics*, 6(1):27, 2005.

[75] P. Hupé, N. Stransky, J.P. Thiery, F. Radvanyi, and E. Barillot. Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20(18):3413–3422, 2004.

[76] R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18–29, 2008.

[77] F Rapaport, E Barillot, and JP Vert. Classification of arrayCGH data using fused svm. *Bioinformatics*, 24(13):i375–82, 2008.

[78] S.J. Diskin, T. Eck, J. Greshock, Y.P. Mosse, T. Naylor, C.J. Stoeckert, B.L. Weber, J.M. Maris, and G.R. Grant. Stac: A method for testing the significance of DNA copy number aberrations across multiple array-CGH experiments. *Genome research*, 16(9):1149–1158, 2006.

[79] M. Guttman and others. Assessing the significance of conserved genomic aberrations using high resolution genomic microarrays. *PLoS Genetics*, 3(8):e143, 2007.

[80] R. Beroukhim et al. Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. *Proceedings of the National Academy of Sciences*, 104(50):20007–20012, 2007.

[81] D.M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[82] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang. Dual transfer learning. In *Proceedings of the 12th SIAM International Conference on Data Mining*, 2012.

[83] W. Dai, G.R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13 th ACM SIGKDD international conference on Knowledge discovery and data mining*, volume 12, pages 210–219, 2007.

[84] Z. Wang, Y. Song, and C. Zhang. Knowledge transfer on hybrid graph. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1291–1296. Morgan Kaufmann Publishers Inc., 2009.

[85] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong. Collaborative dual-plsa: mining distinction and commonality across multiple domains for text classification. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 359–368. ACM, 2010.

[86] F. Zhuang, P. Luo, H. Xiong, Q. He, Y. Xiong, and Z. Shi. Exploiting associations between word clusters and document classes for cross-domain text categorization? *Statistical Analysis and Data Mining*, 4(1):100–114, 2011.

[87] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

[88] E. Blaveri et al. Bladder cancer stage and outcome by array-based comparative genomic hybridization. *Clinical cancer research*, 11(19):7012–7022, 2005.

[89] N. Stransky et al. Regional copy number–independent deregulation of transcription in cancer. *Nature genetics*, 38(12):1386–1396, 2006.

[90] P Fullwood and others. Detailed genetic and physical mapping of tumor suppressor loci on chromosome 3p in ovarian cancer. *Cancer Res*, 59(18):4662–4667, 1999.

[91] M Campiglio, Y Pekarsky, S Menard, E Tagliabue, S Pilotti, and CM Croce. FHIT loss of function in human primary breast cancer correlates with advanced stage of the disease. *Cancer Res*, 59(16):3866–3869, 1999.

[92] VS Dhillon, M Shahid, and SA Husain. Cpg methylation of the FHIT, FANCF, cyclin-D2, BRCA2 and RUNX3 genes in granulosa cell tumors (gcts) of ovarian origin. *Mol Cancer*, 3:33, 2004.

[93] S Zochbauer-Muller, KM Fong, A Maitra, S Lam, J Geradts, R Ashfaq, AK Virmani, S Milchgrub, AF Gazdar, and JD Minna. 5' cpg island methylation of the fhit gene is correlated with loss of gene expression in lung and breast cancer. *Cancer Res*, 61(9):3581–3585, 2001.

[94] H Song and others. Tagging single nucleotide polymorphisms in the BRIP1 gene and susceptibility to breast and ovarian cancer. *PLoS One*, 2(3):e268, 2007.

[95] De Wang, Feiping Nie, and Heng Huang. Unsupervised feature selection via unified trace ratio formulation and k-means clustering (track). In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 306–321. Springer, 2014.

[96] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Jason CH Tsang, Tomislav Ilicic, Johan Henriksson, Kedar N Natarajan, Alex C Tuck, Xuefei Gao, Marc Bühler, Pentao Liu, et al. Single cell rna-sequencing of pluripotent states unlocks modular transcriptional variation. *Cell stem cell*, 17(4):471–485, 2015.

[97] E. Y. Kim, K. Jeon, H. Y. Park, Y. J. Han, B. C. Yang, S. B. Park, H. M. Chung, and S. P. Park. Differences between cellular and molecular profiles of induced pluripotent stem cells generated from mouse embryonic fibroblasts. *Cell Reprogram*, 12(6):627–639, Dec 2010.

[98] S. M. Piecewicz, A. Pandey, B. Roy, S. H. Xiang, B. R. Zetter, and S. Sengupta. Insulin-like growth factors promote vasculogenesis in embryonic stem cells. *PLoS ONE*, 7(2):e32191, 2012.

[99] Barbara Treutlein, Doug G Brownfield, Angela R Wu, Norma F Neff, Gary L Mantalas, F Hernan Espinoza, Tushar J Desai, Mark A Krasnow, and Stephen R Quake. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq. *Nature*, 509(7500):371–375, 2014.

[100] P. N. Tsao, C. Matsuoka, S. C. Wei, A. Sato, S. Sato, K. Hasegawa, H. K. Chen, T. Y. Ling, M. Mori, W. V. Cardoso, and M. Morimoto. Epithelial Notch signaling regulates lung alveolar morphogenesis and airway epithelial integrity. *Proc. Natl. Acad. Sci. U.S.A.*, 113(29):8242–8247, Jul 2016.

[101] A. Hovnanian, P. Duquesnoy, C. Blanchet-Bardon, R. G. Knowlton, S. Amselem, M. Lathrop, L. Dubertret, J. Uitto, and M. Goossens. Genetic linkage of recessive dystrophic epidermolysis bullosa to the type VII collagen gene. *J. Clin. Invest.*, 90(3):1032–1036, Sep 1992.

[102] D. R. Keene, L. Y. Sakai, G. P. Lunstrum, N. P. Morris, and R. E. Burgeson. Type VII collagen forms an extended network of anchoring fibrils. *J. Cell Biol.*, 104(3):611–621, Mar 1987.

[103] B. R. Webber and J. Tolar. From marrow to matrix: novel gene and cell therapies for epidermolysis bullosa. *Mol. Ther.*, 23(6):987–992, Jun 2015.

[104] H. M. Horn and M. J. Tidman. Quality of life in epidermolysis bullosa. *Clin. Exp. Dermatol.*, 27(8):707–710, Nov 2002.

[105] L. Bruckner-Tuderman. Dystrophic epidermolysis bullosa: pathogenesis and clinical features. *Dermatol Clin*, 28(1):107–114, Jan 2010.

[106] A. P. South and E. A. O'Toole. Understanding the pathogenesis of recessive dystrophic epidermolysis bullosa squamous cell carcinoma. *Dermatol Clin*, 28(1):171–178, Jan 2010.

[107] J. E. Wagner, A. Ishida-Yamamoto, J. A. McGrath, M. Hordinsky, D. R. Keene, D. T. Woodley, M. Chen, M. J. Riddle, M. J. Osborn, T. Lund, M. Dolan, B. R. Blazar, and J. Tolar. Bone marrow transplantation for recessive dystrophic epidermolysis bullosa. *N. Engl. J. Med.*, 363(7):629–639, Aug 2010.

[108] S. Iinuma, E. Aikawa, K. Tamai, R. Fujita, Y. Kikuchi, T. Chino, J. Kikuta, J. A. McGrath, J. Uitto, M. Ishii, H. Iizuka, and Y. Kaneda. Transplanted bone marrow-derived circulating PDGFR+ cells restore type VII collagen in recessive dystrophic epidermolysis bullosa mouse skin graft. *J. Immunol.*, 194(4):1996–2003, Feb 2015.

[109] G. Petrof, A. Abdul-Wahab, L. Proudfoot, R. Pramanik, J. E. Mellerio, and J. A. McGrath. Serum levels of high mobility group box 1 correlate with disease severity in recessive dystrophic epidermolysis bullosa. *Exp. Dermatol.*, 22(6):433–435, Jun 2013.

[110] K. Tamai, T. Yamazaki, T. Chino, M. Ishii, S. Otsuru, Y. Kikuchi, S. Iinuma, K. Saga, K. Nimura, T. Shimbo, N. Umegaki, I. Katayama, J. Miyazaki, J. Takeda,

J. A. McGrath, J. Uitto, and Y. Kaneda. PDGFRalpha-positive cells in bone marrow are mobilized by high mobility group box 1 (HMGB1) to regenerate injured epithelia. *Proc. Natl. Acad. Sci. U.S.A.*, 108(16):6609–6614, Apr 2011.

[111] D. Jia, Z. Liu, N. Deng, T. Z. Tan, R. Y. Huang, B. Taylor-Harding, D. J. Cheon, K. Lawrenson, W. R. Wiedemeyer, A. E. Walts, B. Y. Karlan, and S. Orsulic. A COL11A1-correlated pan-cancer gene signature of activated fibroblasts for the prioritization of therapeutic targets. *Cancer Lett.*, 382(2):203–214, Nov 2016.

[112] D. Staloch, X. Gao, K. Liu, M. Xu, X. Feng, J. F. Aronson, M. Falzon, G. H. Greeley, C. Rastellini, C. Chao, M. R. Hellmich, Y. Cao, and T. C. Ko. Gremlin is a key pro-fibrogenic factor in chronic pancreatitis. *J. Mol. Med.*, 93(10):1085–1093, Oct 2015.

[113] R. H. Church, I. Ali, M. Tate, D. Lavin, A. Krishnakumar, H. M. Kok, R. Goldschmeding, F. Martin, and D. Brazil. Gremlin1 plays a key role in kidney development and renal fibrosis. *Am. J. Physiol. Renal Physiol.*, page ajprenal.00344.2016, Jan 2017.

[114] R. P. Mecham and M. A. Gibson. The microfibril-associated glycoproteins (MAGPs) and the microfibrillar niche. *Matrix Biol.*, 47:13–33, Sep 2015.

[115] Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145, 2015.