amU
Aix Marseille Université

NNT : 2024AIXM0395

PhD Thesis

Defended at AMU — Aix-Marseille University
25th of November 2024 by

David HIRST

Approaches for using prior information to enhance the inference of latent variables from omics data

Discipline
Biology-Health

Specialization
Bioinformatics and Genomics

Doctoral School
ED62 : Life and Health Sciences

Laboratory/Research partners
UMR 1251 - MMG - Marseille Medical Genetics

Jury composition

Anaïs BAUDOT Research director CNRS, MMG	Supervisor
Carl HERRMANN Professor Heidelberg University	Examiner
Andrea RAU Research director INRAE, GABI	Reviewer
Arthur TENENHAUS Professor Paris-Saclay University	Reviewer
Jacques VAN HELDEN Professor Aix-Marseille University	President
Matthieu VIGNES Senior lecturer Massey University	Invited member

Marseille Medical Genetics

NNT : 2024AIXM0395

THÈSE DE DOCTORAT

Soutenue à AMU — Aix-Marseille Université
le 25 novembre 2024 par

David HIRST

Amélioration de l'inférence de variables latentes à partir de données omiques grâce à l'utilisation d'information a priori

Discipline
Biologie-Santé

Spécialité
Bioinformatique et Génomique

École doctorale
ED 62 : Sciences de la Vie et de la Santé

Laboratoire/Partenaires de recherche
UMR 1251 - MMG - Marseille Medical Genetics

Composition du jury

Anaïs BAUDOT Directrice de recherche CNRS, MMG	Directrice de thèse
Carl HERRMANN Professeur Université de Heidelberg	Examinateur
Andrea RAU Directrice de recherche INRAE, GABI	Rapporteure
Arthur TENENHAUS Professeur Université Paris-Saclay	Rapporteur
Jacques VAN HELDEN Professeur Aix-Marseille Université	Président du jury
Matthieu VIGNES Maître de conférences Université Massey	Membre invité

Affidavit

I, undersigned, David Hirst, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific supervision of Anaïs Baudot, in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the french national charter for Research Integrity and the Aix-Marseille University charter on the fight against plagiarism.

This work has not been submitted previously either in this country or in another country in the same or in a similar version to any other examination body.

Marseille, 19/09/2024



This work is licensed under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License

List of publications and participation at conferences

List of publications from work carried out as part of the thesis :

1. **Hirst, D.**, Térézol, M., Cantini, L., Villoutreix, P., Vignes, M., & Baudot, A. (2024) MOTL : Enhancing multi-omics matrix factorization with transfer learning. *bioRxiv* https://doi.org/10.1101/2024.03.22.586210
2. Laberthonnière, C., Delourme, M., Chevalier, R., Dion, C., Ganne, B., **Hirst, D.**., et al. (2023) In skeletal muscle and neural crest cells, SMCHD1 regulates biological pathways relevant for Bosma syndrome and facioscapulohumeral dystrophy phenotype. *Nucleic Acids Research* https://doi.org/10.1093/nar/gkad523

Participation at conferences during the thesis period :

1. Marseille Rare Diseases Institute (MarMaRa) symposium, April 2024, Marseille, France
2. Intelligent Systems for Molecular Biology conference, July 2023, Lyon, France : **Poster**
3. Statistical Methods for Post Genomic Data workshop, February 2023, Ghent, Belgium : **Selected oral presentation**
4. New Zealand Statistical Association UnConference, November 2022, Auckland, New Zealand : **Selected oral presentation**

Abstract and keywords

The activity of biological processes, necessary for the development and survival of living organisms, depends on biological molecules. Therefore, to gain improved understanding of cells, organs, and organisms, it is helpful to analyze omics data, generated by high throughput technologies that profile molecules of a particular type from one or more biological samples.

Omics data is usually high-dimensional, consisting of a large number of observed variables, some of which may be uninformative for understanding biological conditions of interest. Additionally, subsets of the molecules, represented by the observed variables, jointly participate in different biological processes. Thus, a popular approach for the analysis of omics data is unsupervised dimension reduction, in which a smaller number of latent variables are inferred, potentially representing underlying biological processes, and helping with the identification of informative observed variables.

Different types of omics data portray different aspects of the biological processes carried out within and between cells. As such, the analysis of multi-omics data can provide improved understanding of some biological conditions of interest, compared to the analysis of a dataset containing only a single type of omics data. An effective method for inferring latent variables from multi-omics data is joint matrix factorization, an unsupervised dimension reduction method. However, there are cases when, due to the small number of samples in a multi-omics dataset, joint matrix factorization can struggle to separate underlying biological signals into a meaningful lower dimension representation. This is especially a challenge in the study of rare diseases, for which it can be difficult to obtain multi-omics data.

For my main PhD project, I developed MOTL, a transfer learning approach for the application of joint matrix factorization to multi-omics data. MOTL performs joint matrix factorization on a multi-omics target dataset, generated from a small number of samples, by incorporating information inferred from performing joint matrix factorization on a multi-omics learning dataset, generated from a larger number of samples, characterized by various biological conditions. In evaluations of MOTL, I demonstrated that this incorporation of prior information can enhance the ability of joint matrix factorization to discover biological processes associated with a biological condition of interest.

I also explored the use of prior information, to enhance the inference of latent variables from omics data, during one of my other PhD projects. During this project, I implemented, and evaluated, a modification to an existing variational autoencoder architecture, another type of unsupervised dimension reduction. This modification allowed the inclusion of weights that control the relative contribution of observed

variables to the loss function. I observed that the inclusion of weights, based on some prior information, resulted in latent variable values that better differentiated between groups of samples.

Finally, for another project, I was involved in a collaboration that involved the analysis of different types of omics data from patients with two different rare genetic diseases, and healthy controls. In this project, different types of omics data were analyzed separately with differential testing, yielding subsets of observed variables potentially associated with these diseases. With my collaborators, we used prior information, in the form of annotation resources, to convert these lists of relevant observed variables into lists of relevant biological processes.

Keywords: omics data, dimensionality reduction, variational inference, data integration, transfer learning, rare diseases, systems biology

Résumé et mots clés

Une grande variété de processus biologiques sont nécessaires au développement et à la survie des organismes vivants. Ces processus biologiques dépendent d'un grand nombre de molécules biologiques. Les technologies à haut débit permettent d'établir les profils des molécules dans les échantillons, générant ainsi les données dites "omiques" qui permettent de mieux comprendre les cellules, les organes et les organismes.

Les données omiques sont généralement de haute dimension. Elles se composent d'un grand nombre de molécules biologiques, les variables observées, qui peuvent ne pas être informatives pour la compréhension des conditions biologiques d'intérêt. Dans ce contexte, les approches de réduction de dimension non-supervisée sont populaires pour l'analyse des données omiques. Ces approches permettent de déduire un nombre réduit de variables latentes représentant potentiellement les processus biologiques et aidant à l'identification des variables observées informatives.

De plus, différents types de données omiques permettent de charactériser les processus biologiques à différentes échelles à l'intérieur et entre les cellules. Ainsi, il est attendu que l'analyse de données multi-omiques permette de mieux comprendre les systèmes biologiques par rapport à l'analyse de données omiques simples. Une méthode efficace pour déduire les variables latentes des données multi-omiques est la factorisation conjointe de matrices multi-omique, une méthode non supervisée de réduction des dimensions. Cependant, dans certains cas, en raison du petit nombre d'échantillons disponible, les méthodes de factorisation de matrice peuvent avoir du mal à séparer les signaux biologiques latents en une représentation pertinente. Il s'agit d'un défi particulier dans l'étude des maladies rares, pour lesquelles il peut être difficile d'obtenir des données multi-omiques sur une nombre suffisant d'échantillons.

Dans le cadre de mon doctorat, j'ai développé MOTL, une approche d'apprentissage par transfert pour la factorisation de matrices multi-omique. MOTL effectue une factorisation de matrices sur un ensemble de données cible, généré à partir d'un petit nombre d'échantillons, en incorporant des informations déduites de la factorisation multi-omique effectuée sur un ensemble de données d'apprentissage, généré à partir d'un grand nombre d'échantillons caractérisés par diverses conditions biologiques. Dans les évaluations de MOTL, j'ai démontré que l'apprentissage par transfer peut améliorer la capacité de la factorisation de matrices multi-omique à découvrir des processus associés à une condition biologique d'intérêt.

Au cours de l'un de mes autres projets de doctorat, j'ai également exploré l'utilisation d'informations a priori pour améliorer l'inférence de variables latentes à partir de données omiques. J'ai mis en œuvre et évalué une modification d'une architecture d'autoencodeur variationnel, un autre type de méthode de réduction de dimension

non supervisée. Cette modification a permis d'inclure des poids contrôlant la contribution relative des variables observées à la fonction de perte. J'ai observé que l'inclusion de poids basés sur des informations préalables permettait d'obtenir des valeurs de variables latentes qui différenciaient mieux les groupes d'échantillons.

Enfin, j'ai participé à une collaboration portant sur l'analyse de différents types de données omiques provenant de patients atteints de deux maladies génétiques rares différentes et de témoins sains. Dans ce projet, différents types de données omiques ont été analysés séparément avec des tests différentiels, ce qui a permis d'obtenir des sous-ensembles de variables observées potentiellement associées à ces maladies. Avec mes collaborateurs, nous avons utilisé des informations préalables, sous la forme de ressources d'annotation, pour convertir ces listes de variables observées pertinentes en listes de processus biologiques pertinents.

Mots clés: données omiques, reduction de dimension, inférence variationnelle, intégration des données, apprentissage par transfert, maladies rares, biologie des systèmes

Acknowledgments

I wish to first thank my thesis supervisor **Anaïs Baudot**, whose guidance and support has been invaluable for my journey into the sphere of computational biology, my development as a scientist, and for my progress during this PhD. Thank you for your enthusiasm and trust, and for always being available to help and discuss, regardless of circumstances. I have thoroughly enjoyed, and benefited from, both having you as a supervisor, and working within a team with a culture of excellence, mutual respect, and positivity.

I would next like to thank **Matthieu Vignes**, who has contributed significantly as a co-supervisor over the past three years, and whose expertise in statistical theory and applications has been indispensable for my ongoing development as a statistician, and for my progress during this PhD. Thank you for your guidance and support over the years, and consistent willingness to accommodate the challenges of different time zones to be available for valuable discussions.

Thank you **Laura Cantini** and **Paul Villoutreix**, for providing valuable feedback on the implementation and evaluation of MOTL, and to **Morgane Térézol** for so expertly, and enthusiastically, optimizing the MOTL code, and helping make it available for public use.

Thank you to my colleagues in the wider Systems Biomedicine team, as well as to **Élisabeth Remy** and members of the MABioS team. It's been most enjoyable and rewarding to work alongside, and learn from, such a talented and friendly group of people.

Thank you **Carl Herrmann** and **Daria Doncevic** for the guidance and support I received while I was working with OntoVAE, and to the wider team for making me feel so welcome in Heidelberg.

Thank you **Frédérique Magdinier** for involving me in the FSHD and BAMS collaboration, and for explanations of biological concepts, and thank you **Raphaël Chevalier** for assistance throughout this collaboration.

Thank you to **Marc Bartoli**, **Kim-Anh Lê Cao**, and **Lionel Spinelli**, for participating in my thesis committee, your encouragement and feedback was very much appreciated.

Thank you **Andrea Rau** and **Arthur Tenenhaus**, for agreeing to review this thesis, and to the whole jury, including **Jacques van Helden**, for agreeing to participate.

Thank you **Nicolas Terrapon**, **Denis Puthier**, **Christophe Bordi**, **Andreas Zanzoni**, and **Pedro Coutinho**, who I was fortunate to learn from in my role as a teaching assistant.

On a personal note, thank you **Mum** and **Dad** for everything, and finally, thank you **Natália** - I'm extremely grateful for your unwavering support, encouragement, and optimism, over these past three years.

Contents

Affidavit 3

List of publications and participation at conferences 4

Abstract and keywords 5

Résumé et mots clés 7

Acknowledgments 9

Contents 10

List of Figures 13

List of Tables 14

Introduction 15

1 Introduction to Omics Data 17
- 1.1 Introduction . . . 17
- 1.2 Molecular biology basics . . . 17
  - 1.2.1 Cells and biological processes . . . 17
  - 1.2.2 DNA . . . 18
  - 1.2.3 Chromatin . . . 19
  - 1.2.4 RNA . . . 20
  - 1.2.5 Proteins . . . 22
  - 1.2.6 The -omes and biological conditions . . . 23
- 1.3 Omics data . . . 23
  - 1.3.1 Genomic data . . . 24
  - 1.3.2 Epigenomic data . . . 25
  - 1.3.3 Transcriptomic data . . . 27
  - 1.3.4 Multi-omics data . . . 28

2 Omics Data Analysis 30
- 2.1 Introduction . . . 30
- 2.2 Bayesian statistics . . . 31
  - 2.2.1 Empirical Bayes . . . 32
  - 2.2.2 Markov chain Monte Carlo . . . 32
  - 2.2.3 Variational inference . . . 33

2.3 Visual exploration and pre-processing . . . 34
2.3.1 Visual exploration . . . 35
2.3.2 Pre-processing . . . 38
2.4 Differential testing . . . 41
2.4.1 Parametric and nonparametric methods for omics data . . . 41
2.4.2 Differential methylation . . . 42
2.4.3 Differential gene expression . . . 44
2.5 Unsupervised Dimension Reduction . . . 45
2.5.1 Matrix factorization . . . 46
2.5.2 Autoencoders . . . 50
2.5.3 Nearest neighbors approaches . . . 52

3 Data integration 54
3.1 Introduction . . . 54
3.2 Using annotation resources . . . 55
3.2.1 Annotation resources . . . 55
3.2.2 Incorporation after model fitting . . . 57
3.2.3 Incorporation during model fitting . . . 59
3.3 Multi-omics matrix factorization . . . 60
3.3.1 The general joint matrix factorization model . . . 61
3.3.2 Approaches based on distance . . . 62
3.3.3 Approaches based on correlation and covariance . . . 63
3.3.4 Approaches based on distributional assumptions . . . 64
3.3.5 Multi-omics data with few samples . . . 67
3.4 Transfer learning . . . 69
3.4.1 Transfer learning and omics data . . . 69
3.4.2 Transfer learning and matrix factorization . . . 70
3.4.3 Transfer learning based on MOFA . . . 71

4 MOTL: Multi-omics matrix factorization with transfer learning 72
4.1 Introduction . . . 72
4.2 Motivation and objectives . . . 72
4.3 Article . . . 73
4.4 Discussion . . . 104

5 OntoVAE extension 106
5.1 Context . . . 106
5.2 Introduction . . . 107
5.3 The OntoVAE model . . . 107
5.4 OntoVAE with weighted loss . . . 109
5.4.1 Weighted loss statistical implications . . . 110
5.4.2 Weighted loss implementation . . . 112
5.5 Evaluation of OntoVAE with weighted loss . . . 112
5.5.1 Methods and materials . . . 112

5.5.2 Evaluation results 116
5.6 Discussion 121

6 Analysis of FSHD2 and BAMS omics data **124**
6.1 Context 124
6.2 Introduction 124
6.3 Summary of results 125
6.4 My contribution 125
6.5 Article 127

Conclusion **147**

Bibliography **149**

List of Figures

1.1 The structure of DNA . . . 19
1.2 Chromosomes . . . 20
1.3 Transcription and Translation . . . 22

2.1 DNA methylation density plots . . . 36
2.2 Plot of PCA values for RNA-Seq data samples . . . 36
2.3 UMAP values for samples from two different cancer types . . . 37
2.4 Dendrogram from hierarchical clustering of DNA methylation data . . . 38
2.5 Hierarchical clustering combined with heatmap . . . 38
2.6 Matrix factorization overview . . . 47
2.7 Variational autoencoder overview . . . 51

3.1 Muscle tissue development GO term . . . 56
3.2 Reactome graph for the Prostacyclin signalling through prostacyclin receptor pathway . . . 57
3.3 OntoVAE overview . . . 60

5.1 Validation loss curves from fine-tuning OntoVAE with the derived gene weights . . . 115
5.2 Boxplots of AUC values . . . 118
5.3 Violin plots of changes in AUC values . . . 120
5.4 Empirical cumulative distributions of changes in AUC values . . . 121

6.1 Distributions of ChromHMM track annotations . . . 127

List of Tables

5.1 GO terms with the largest positive changes in AUC, achieved from fine-tuning OntoVAE with derived gene weights, by RNA-Seq dataset 122

Introduction

In this thesis, I present the results of the research I carried out towards a PhD in Bioinformatics. My PhD was supervised by Dr. Anaïs Baudot at the Marseille Medical Genetics research center, and also received significant supervisory input from Dr. Matthieu Vignes, based at Massey University, New Zealand.

My research was primarily concerned with methods that, when applied to omics data, infer latent variables that are associated with some biological conditions of interest. Additionally, throughout the course of my PhD, I made use of methods that enable inferences to be made, from omics data, about associations between observed variables and biological conditions. In both cases, my overall focus was on enhancing the ability of these methods to yield understanding of the biological processes that underlie some biological conditions of interest, by using multiple types of omics data and/or prior information.

I first became acquainted with the analysis of omics data, and the field of Bioinformatics, while undertaking an MSc with a specialization in Statistics. Prior to commencing my MSc, I had spent several years working as a data analyst in industries unrelated to biological science, and had minimal knowledge of biological concepts. I was immediately impressed by the interdisciplinary of Bioinformatics, and fascinated by the statistical challenges posed by omics data, as well as by the diversity of solutions proposed for those challenges. In addition, beginning the process of becoming familiar with relevant biological concepts was invigorating and rewarding, as was the promise of being able to contribute to the understanding of human health. This initial exposure provided the motivation for commencing this PhD, from which I hoped to both deepen my existing knowledge of statistical theory and applications, and broaden my scientific horizon with increased familiarity of concepts from molecular biology. I was also enthusiastic about the chance, during the PhD, to work alongside, and collaborate with, scientists from a broad range of backgrounds, as part of a combined effort to better understand the mechanisms of disease. Fortunately these personal objectives have been met, which I have hopefully been able to convey throughout the contents of this thesis.

The topic of my main PhD research project was the application of joint matrix factorization to multi-omics data that has been generated from a small number of samples. This project was part of a larger, interdisciplinary project, called CAMUDI, launched to develop various analysis methods that can accommodate multi-omics data generated from a small number of samples. For this project I developed a transfer learning approach to joint matrix factorization, and evaluated whether it leads to improved lower dimensional representations of such multi-omics datasets.

During my PhD, I successfully applied for a grant which funded a research stay

within Dr. Carl Herrmann's team at Heidelberg University in Germany. While in Heidelberg, I worked on a another project, in which I implemented and evaluated a modification to an existing variational autoencoder. The modification consisted of allowing the inclusion of weights, based on some prior information, that control the relative contribution of observed variables to the loss function used for model fitting.

In addition, I had a third project during my PhD. This project consisted of the continuation of my involvement, began during my MSc, in a collaboration with a team of biologists, led by Dr. Frédérique Magdinier, at Marseille Medical Genetics. I contributed to this collaboration by performing differential methylation analyses to identify, and annotate, differentially methylated positions and regions, for two rare genetic diseases called type 2 facioscapulohumeral muscular dystrophy (FSHD2), and Bosma arhinia and microphthalmia syndrome (BAMS).

This thesis consists of six chapters. In the first three chapters, I introduce the concepts and methods relevant to the research I carried out during my PhD. In chapter 1, I provide an introduction to omics data, beginning with some concepts from molecular biology that motivate omics data analysis, before presenting different types of omics data. In chapter 2, I introduce commonly used omics data analysis concepts and methods that I worked with during the PhD. In chapter 3, I present ways in which omics data analysis can be enhanced, either in the form of multi-omics data analysis, or by the incorporation of prior information into an omics data analysis.

In the last three chapters, I present the results from the work I carried out during my PhD. In chapter 4, I present the results from my main PhD research project, describing the implementation and evaluation of a transfer learning approach to joint matrix factorization. In chapter 5, I present the results from the project that I worked on while in Heidelberg, describing the implementation and evaluation of a modification to an existing variational autoencoder architecture. In chapter 6, I present results from my other project, summarizing the results of the collaboration concerned with the two rare genetic diseases, FSHD2 and BAMS, and describing my contribution.

1 Introduction to Omics Data

Contents

1.1 Introduction . . . 17
1.2 Molecular biology basics . . . 17
1.2.1 Cells and biological processes . . . 17
1.2.2 DNA . . . 18
1.2.3 Chromatin . . . 19
1.2.4 RNA . . . 20
1.2.5 Proteins . . . 22
1.2.6 The -omes and biological conditions . . . 23
1.3 Omics data . . . 23
1.3.1 Genomic data . . . 24
1.3.1.1 NGS platforms . . . 24
1.3.1.2 Variant calling . . . 25
1.3.2 Epigenomic data . . . 25
1.3.2.1 Methylation microarrays . . . 26
1.3.2.2 ChIP-Seq . . . 27
1.3.3 Transcriptomic data . . . 27
1.3.3.1 RNA-Seq . . . 27
1.3.4 Multi-omics data . . . 28

1.1 Introduction

This chapter is an introduction to the different types of omics data I worked with during my PhD. Firstly, section 1.2 summarizes concepts from molecular biology required to understand what omics data is, and why the analysis of omics data is useful. Then, in section 1.3, the different types of omics of omics data I worked with are defined and described.

1.2 Molecular biology basics

1.2.1 Cells and biological processes

All living organisms consist of either one or many cells, which are considered to be their fundamental units [1]. Multicellular organisms begin as a single cell, and

grow and reproduce themselves through processes of cell division and differentiation [2, 3]. It has been estimated that the average human adult is composed of around 30 trillion cells [4], belonging to one of two categories [5]. Germ cells are used to produce gametes for sexual reproduction, and somatic cells are all other cells. At a more granular level, cells can be categorized into one of hundreds of different types, based on morphology and function [4].

In addition to cell division and differentiation, there are many biological processes occurring, within and between cells, that are necessary for the development and survival of living organisms [1, 6]. Some examples of biological processes include cell growth [7], cellular homeostasis [1], immune system responses [8], and the numerous metabolic reactions that provide cells with energy.

Biological processes involve DNA molecules, RNA molecules, proteins, and complexes consisting of these macromolecules, such as chromatin [2, 9]. The composition and function of DNA molecules (section 1.2.2), RNA molecules (section 1.2.4), proteins (section 1.2.5), and chromatin (section 1.2.3), are described in the following sections, along with the biological processes that produce and modify them.

1.2.2 DNA

Living organisms store instructions for biological processes in the form of deoxyribonucleic acid (DNA) molecules. A DNA molecule [9, 10] consists of two strands, each a chain of nucleotides, bound together in the shape of a double helix (Figure 1.1). Each nucleotide is a deoxynucleotide, meaning it consists of a sugar called deoxyribose, a phosphate group, and a base which is either adenine (A), cytosine (C), guanine (G), or thymine (T). The nucleotides of a single chain are linked by bonds between the sugars and phosphates that form a directional backbone with a $3'$ end and a $5'$ end. The two chains are held to each other by bonds formed between base pairings of A and T or C and G. Human DNA is stored in the nucleus (nuclear DNA) and mitochondria (mitochondrial DNA) of cells [11]. Most human cells contain two inherited sets of nuclear DNA, an exception being gametes which contain a single set [12]. Each set of nuclear DNA consists of around 3 billion base pairs located across 23 chromosomes [11] (Figure 1.2).

The sequence of DNA nucleotides in a cell or organism is called a genome [10, 13]. Genomes contain the information required by cells to produce the RNA molecules and proteins that carry out biological processes. The sections of a genome that contain this information are called genes [13, 14]; accordingly, RNA molecules and proteins are called gene products. The process of using the information in a gene to create a gene product is called gene expression [15] (Figure 1.3).

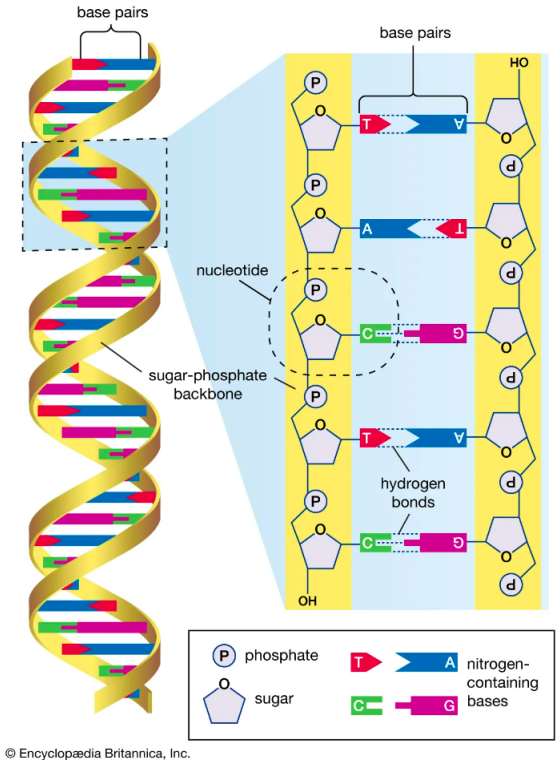


Figure 1.1 – **The structure of DNA**. Image taken from [16]

1.2.3 Chromatin

Chromosomes (Figure 1.2) consist of DNA molecules packaged with proteins in a complex called chromatin [17, 18] . Gene expression is influenced by compounds and proteins that bind to chromatin, modifying its structure and composition while leaving DNA nucleotide sequence unchanged. These modifications are known as epigenetic marks [19], and the set of epigenetic marks in a cell is known as an epigenome [20]. The epigenome is heritable through cell division, and yet varies across the cells and life cycle of an organism [19, 21]. Changes in the epigenome occur through mechanisms such as DNA methylation, histone modification, and chromatin remodeling. These mechanisms play a role in developmental processes, as well as occurring in response to environmental factors.

DNA methylation is the addition of a methyl group to a cytosine (C) base, mostly occurring at CpG sites, where CpG denotes C followed by guanine (G) in the $3'$-$5'$ direction [22]. CpG sites occur with low frequency across human DNA, except for in CpG islands (CGIs), which are characterized by a high frequency of CpG sites and are located in the promoter regions (section 1.2.4) of the majority of genes [19, 23]. There is typically a high prevalence of methylation among CpG sites located outside of CGIs, and a low prevalence of those located within [21]. DNA methylation contributes to gene silencing [19, 24], either by attracting proteins that are involved in the repression of gene expression, or by repelling proteins which bind to DNA in order to initiate gene

expression. DNA methylation patterns are widely erased shortly after fertilization, and then subsequently reestablished and maintained by specific enzymes [19, 25].

Histone modifications and chromatin remodeling cause changes in chromatin packaging and accessibility that influence gene expression. DNA is able to fit in the nucleus thanks to nucleosomes, the basic units of chromatin [17, 21]. Nucleosomes consist of a section of the DNA molecule wrapped around a complex of histone proteins, resulting in a shortening of the DNA molecule. The tightness of this DNA packaging in chromatin is not static, instead it changes depending on requirements for DNA accessibility [18, 19]. In the case of histone modifications, specific enzymes (section 1.2.5) perform reversible modifications of the histones, either impacting their binding with DNA, or the binding of other proteins required for gene expression. On the other hand, chromatin remodeling enzymes move histones in and out of nucleosomes.

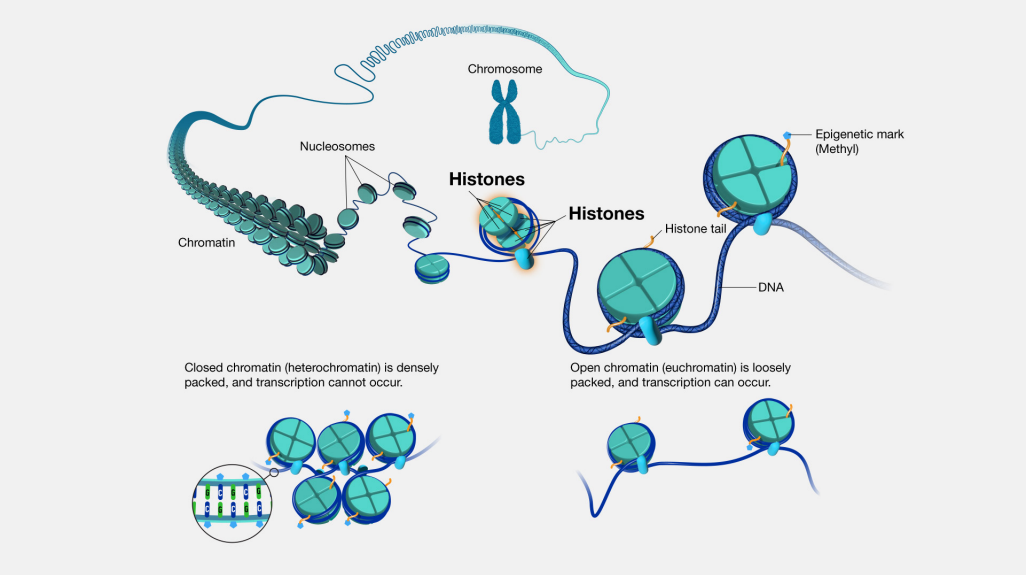


Figure 1.2 – **Chromosomes**. The structure of chromosomes, seen at different scales. Image taken from [13]

1.2.4 RNA

Gene expression begins with the process of transcription [9, 14]. During transcription (Figure 1.3), a section of a gene's DNA is copied to make a ribonucleic acid (RNA) molecule, a single chain of nucleotides also known as an RNA transcript. The nucleotides in an RNA transcript, called ribonucleotides, differ from those in DNA in that they contain uracil (U) bases instead of thymine (T), and ribose instead of deoxyribose.

Some genes are copied to make messenger RNA (mRNA) transcripts, which are subsequently used in the synthesis of proteins (section 1.2.5). For many genes, however, the final product of gene expression is non-coding RNA (ncRNA) [26, 27]. There are several types of ncRNA, and they typically play a regulatory role in gene expression. For

example, small nuclear RNAs (snRNAs) form assemblies, along with protein molecules, that carry out a process, described below, called splicing. microRNA (miRNA) is another type of ncRNA. miRNA molecules bind to target mRNA molecules, preventing protein synthesis.

Transcription is initiated, and carried out, by enzymes called RNA polymerases, with the help of a large number of other proteins such as transcription factors [28]. RNA polymerases and transcription factors bind to a region of DNA called a promoter, whose nucleotide sequence indicates a starting point for transcription. Transcription factors, and other proteins, also bind to a region of DNA called an enhancer, influencing the rate of transcription [29, 30]. When an enhancer comes into contact with a promoter, transcription is activated by the interaction between the proteins at the enhancer, and the proteins at the promoter. After transcription is initiated [28, 31], an RNA polymerase moves along one of the DNA strands, called the template strand, towards the $5'$ end. As the RNA polymerase progresses, it opens the double helix, and synthesizes a growing RNA transcript through complementary base pairing of incoming ribonucleotides with the nucleotides on the template strand. When the RNA polymerase reaches a region of DNA called a terminator, it stops transcription and releases both the template strand and the RNA transcript.

For some types of RNA, including mRNA, the transcript undergoes additional modifications at both ends, as well as a process called splicing. RNA splicing involves the removal of sequences of nucleotides, called introns, leaving a final transcript that consists of nucleotide sequences, called exons, which have been spliced together. For many human genes, there are multiple ways that the raw RNA transcript can be spliced, resulting in a large diversity of possible RNA transcripts and proteins.

The set of all RNA transcripts in a cell is called a transcriptome [32, 33], and the sequence of all exons in a genome is called an exome [13, 34]. In contrast to the genome, the transcriptomes of an organism are highly variable between different cell types, as well as between cells of the same type.

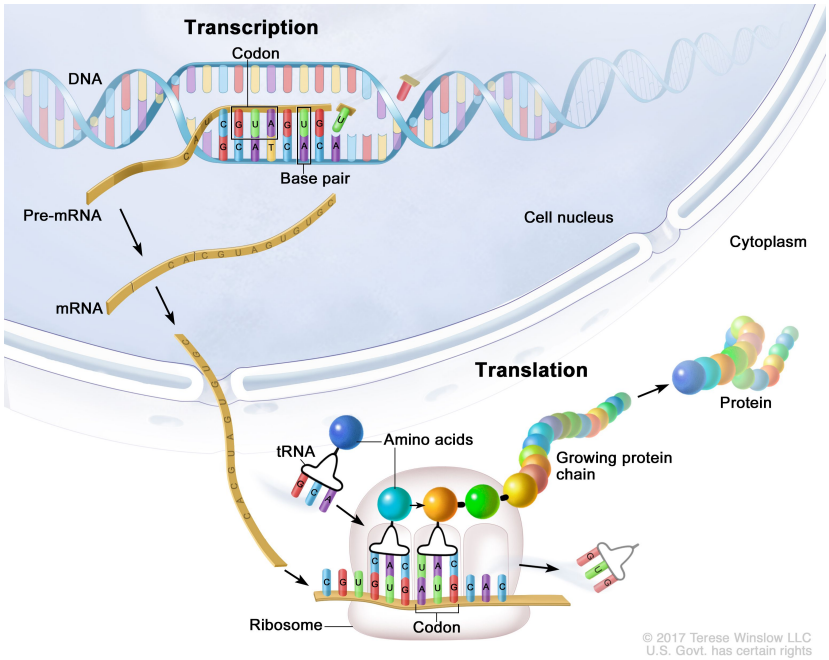


Figure 1.3 – **Transcription and Translation**. During transcription, DNA is copied to make an RNA transcript. During translation, a protein is made based on the information in an mRNA transcript. Image taken from [35]

1.2.5 Proteins

Proteins are formed from polypeptides, chains of amino acids constructed, in a process called translation (Figure 1.3), according to the nucleotide sequences of mRNA transcripts [9, 14, 36]. Translation occurs within an assembly of proteins and ribosomal RNA (rRNA) molecules, called a ribosome, which attaches to an mRNA molecule to initiate the process. Ribosomes contain several rooms and tunnels that facilitate protein assembly. During translation, the mRNA molecule is pulled through a tunnel in the ribosome, while amino acids are brought to an assembly room by transfer RNA (tRNA) molecules and added to a growing polypeptide chain. Each type of tRNA molecule is able to attach to a specific amino acid at one end, while at the other end there is a triplet of nucleotides called an anticodon. Each anticodon corresponds to a codon, a complementary triplet of mRNA nucleotides. When a tRNA molecule arrives at the assembly room in the ribosome, it will bind to the mRNA if its anticodon matches with the mRNA codon currently exposed in the room, allowing the transported amino acid to be added to the polypeptide chain. This process begins at a start codon on the mRNA molecule, and continues until the appearance of a stop codon. Then the peptide chain must be folded into a three-dimensional structure, and undergo some post-translational modification before becoming a functional protein.

There are many different proteins, playing a diverse range of roles in the biological processes required for the development and survival of living organisms [8, 37]. In addition to the proteins which regulate the expression of other gene products, such

as transcription factors, there are enzymes, which are proteins that act as catalysts for chemical reactions, as well as antibodies, which contribute to immune system response by attaching themselves to antigens. Other proteins transport molecules within or between cells, while some proteins play structural roles within cells.

The set of all proteins in a cell, or collection of cells, is called a proteome [32]. The proteome is highly variable across cells [31], due to both variation in gene expression and post-translational modifications.

1.2.6 The -omes and biological conditions

As discussed in the preceding sections, regions of the genome, known as genes, contain instructions for the production of RNA molecules and proteins. These macromolecules, also known as gene products, are the result of a process called gene expression. Gene products are involved in the biological processes carried out within and between cells, including the expression of other genes. The DNA sequence of a gene influences the ability of cells to express the gene, as well as specifying the composition of the resulting gene products. Gene expression is also influenced by the epigenomic modifications that impact the state of the chromatin in which a gene's DNA is packaged, and relies on the products of other genes.

The activity of biological processes, necessary for the development and survival of living organisms, varies across cell types and over time, and is influenced by both an organism's genome and its environmental interactions [1, 38]. These biological processes depend on DNA molecules, chromatin, RNA molecules, and proteins. Therefore, to gain improved understanding of cells, organs, and organisms, it is helpful to identify, and analyze, patterns in profiles of their genomes, epigenomes, transcriptomes, and proteomes.

In particular, many diseases are associated with abnormalities in these molecular patterns. For example, type 2 Facioscapulohumeral muscular dystrophy (FSHD2) (chapter 6), is a disease associated with DNA sequence variants (section 1.3.1.2), as well as an abnormality in chromatin structure. Abnormal patterns in the genome and epigenome are also associated with cancer [18, 24], as are abnormal patterns in gene expression [39, 40]. The analysis of abnormalities in these molecular patterns can provide insight into the biological processes underlying a disease of interest [41], potentially yielding new candidates for treatment targets [42, 43], or biomarkers for diagnosis, prognosis, or response prediction [44].

1.3 Omics data

Our ability to interrogate biological systems has been enhanced in recent years, by high throughput technologies that measure molecules of a particular type from one or more biological samples [45, 46]. These technologies are used to generate high dimensional datasets, called omics data [32], which serve as input for a large number of analysis techniques [47]. Omics data is typically categorized based on the

molecules which have been measured. Examples include genomic, epigenomic, transcriptomic, and proteomic data, containing measurements for genomes, epigenomes, transcriptomes, and proteomes, respectively [32, 46].

Omics data can be generated for each cell in a biological sample (single-cell data), or for the pool of cells as a whole (bulk data). I have worked exclusively with bulk data during my PhD, and the omics data mentioned in this document is bulk data unless specified otherwise. There are many technologies for generating omics data; in the following sections I give an overview of the types of omics data, and the technologies used to generate them, that I have used in the analyses carried out during my PhD.

1.3.1 Genomic data

Genomic data is generated by profiling DNA nucleotide sequences (section 1.2.2) from one or more biological samples [48], and is analyzed to identify variation associated with a biological condition of interest or between specific patients [49]. Identifying biologically relevant variation across genomes has been made possible by the development of high throughput technologies that interrogate large quantities of DNA molecules in parallel. These high throughput technologies include [50, 51] i) microarrays, which use probes that hybridize to DNA fragments to detect variation in predetermined sections of a genome; ii) next generation sequencing (NGS) approaches, which involve recording the sequences of DNA fragments and aligning them to a reference genome, yielding the sequences of whole genomes or exomes. This is the most widely used technology, from which most of the currently accessible data have been produced; iii) third generation sequencing (TGS) approaches, which are based on technologies that are able to read the sequences of unfragmented DNA molecules.

1.3.1.1 NGS platforms

The emergence of NGS technologies since the early 2000's has led to a large increase in genomic data output [48, 49, 51]. NGS technologies can be used to record the sequence of a whole genome, through whole genome sequencing, or of a whole exome, through whole exome sequencing. Commonly used NGS technologies include the Illumina, Ion Torrent and SOLiD platforms. The sequencing workflow begins with fragmentation of the sample DNA, adapter ligation and fragment size selection. For whole exome sequencing, there is an additional step in which the fragments are filtered to produce an exon-enriched sample [34, 52]. The fragments then undergo clonal amplification, resulting in a library consisting of multiple templates per DNA fragment. Then, during sequencing, the nucleotide sequences of the templates are recorded, eventually resulting in a read for each fragment. NGS platforms are able to generate a large number of short reads, 150-600 base pairs long, in 4-56 hours, depending on the platform. Before NGS output can be analyzed with respect to genomic variation, the reads are trimmed and filtered using Phred scores, which measure the quality of nucleotide identification. The reads are then aligned against a reference genome

to obtain a sequence for the whole genome or exome. A reference genome [53] is a representative genome, that has been assembled from the genomic data from multiple organisms from a species.

1.3.1.2 Variant calling

Identifying biologically relevant variation across genomes relies on the detection of variants, such as single nucleotide variants (SNVs), small insertions-deletions (indels), small inversions, and larger structural variants [32, 54]. Structural variants include copy number variants, sequences whose number of repeats vary across genomes of interest [55], along with large indels and inversions.

After reads have been aligned to a reference genome, variants can be identified for a sequenced sample using a variant caller [56]. Variant callers typically employ a statistical model to quantify the uncertainty associated with an identified variant [57–59]. Variants estimated to occur from alignment errors, or that occur with a high frequency within a healthy population, may be subsequently filtered out. Public databases are then used to annotate the remaining variants with respect to their predicted consequences [60, 61]. I used such annotations for the work presented in chapter 4, when pre-processing (section 2.3.2) a dataset containing SNVs and indels identified in samples from cancer patients. I filtered the dataset to only include variants predicted to have an impact on the gene expression process, and the composition of gene products. A variant might, for example, have such an impact if it changes which amino acid is associated with a particular codon during translation (missense variant), disrupts the translation of codons by changing the length of a DNA sequence (frame shift indel), or causes a codon to become a stop codon instead of being associated with an amino acid (nonsense variant) [13].

In the study of some diseases, it is of interest to detect inherited variants. For other diseases, such as cancer, it is informative to detect somatic variants in tumors, which are those that are present in somatic cells but have not been inherited [62, 63]. To detect somatic variants contained in a sample of DNA from a tumor, DNA from a matched normal sample is also sequenced. Variants detected for the normal sample can be used to estimate whether variants detected from the tumor sample are somatic or not.

1.3.2 Epigenomic data

Epigenomic data is generated by profiling a particular aspect of the epigenomes (section 1.2.3) of cells from one or more biological samples, such as DNA methylation, histone modifications and chromatin structure [64, 65]. Epigenomic data can be used to annotate regions of the genome as well as compare epigenomic patterns across samples. There are numerous approaches that are used for generating epigenomic data, depending on the particular aspect of the epigenome being studied. These approaches are typically based on the capture of targeted fragments of DNA, followed by the use of microarray or NGS technology. Below I discuss data generated for the

purposes of studying DNA methylation and histone modifications, as I used such data during my PhD.

1.3.2.1 Methylation microarrays

The most common way to detect DNA methylation is through bisulfite treatment [65–67]. Sample DNA is treated with sodium bisulfite, resulting in the conversion of unmethylated cytosine (C) bases to uracil (U) without altering methylated cytosine bases. Uracil bases are then converted to thymine (T), allowing for quantification of DNA methylation prevalence using a high throughput genomics technology. There are three common ways to quantify DNA methylation after bisulfite treatment: 1) hybridize fragments of the treated DNA to a microarray containing probes targeting genomic locations known to be impacted by methylation; 2) sequence the treated DNA using an NGS platform, in an approach known as whole-genome bisulfite sequencing; 3) enrich the CpG sites in the treated DNA, and then sequence using an NGS platform, in an approach known as reduced representation bisulfite sequencing. While whole-genome bisulfite sequencing offers the most comprehensive overview of DNA methylation, targeted profiling with microarray hybridization is less costly and is the most popular method for the study of DNA methylation across the genome.

Methylation microarrays consist of specific DNA sequences, called probes, attached to a surface. These probes are able to hybridize to fragments of DNA through complementary base pairing. In a Methylation microarray experiment the bisulfite treated DNA is fragmented, and the mixture of fragments is left to hybridize to the probes, resulting in the emission of fluorescent signals. An example of a commonly used methylation microarray technology is the Infinium Bead Chip array by Illumina, a recent version of which is the MethylationEPIC (EPIC) BeadChip [64]. An EPIC BeadChip [68] is divided into 8 sections, each dedicated to a distinct sample. Probes of 50 bases long are located on beads. Each probe consists of a sequence designed to match with one of over 850k sections of the genome that end in a CpG site. There are two types of probes: Type I probes have two versions for a given targeted CpG site, one able to hybridize to a section of DNA ending in a methylated site, the other to an unmethylated site; Type II probes are able to hybridize to DNA containing either an methylated or unmethylated target CpG site, with the methylation status determining the signal color. Multiple copies of the same probe are present in a given BeadChip section. For each sample and CpG site, the total methylated signal intensity, γ_m, unmethylated signal intensity, γ_u, and a small constant, α, are then used to calculate a Beta-value:

$$Beta = \frac{\max(\gamma_m, 0)}{\max(\gamma_m, 0) + \max(\gamma_u, 0) + \alpha} \tag{1.1}$$

A Beta-value represents the percentage of copies of the CpG site that were methylated in the sample [69].

1.3.2.2 ChIP-Seq

The most common way of detecting histone modifications across the genome is chromatin immunoprecipitation followed by sequencing (ChIP-Seq) [65, 70, 71]. I focus on the detection of histone modifications here, but it is worth noting that ChIP-Seq is also used to identify nucleosome and transcription factor binding locations. In a ChIP-Seq experiment for the study of histone modifications, there are a number of steps. The chromatin from a biological sample is fragmented. This is followed by chromatin immunoprecipitation (ChIP), in which antibodies of a particular type are used to target histones that have undergone a modification of interest. The type of antibody used depends on the type of histone modification which is being studied. This results in a ChIP sample, consisting of chromatin fragments that contained the targeted histones. The DNA is then removed from the ChIP sample, and sequenced by an NGS platform. The NGS reads are aligned to a reference genome, and regions of the genome with high read counts, known as peaks, are identified. In order to mitigate the impact of biases in the ChIP-Seq process, the experiment typically includes a sample of Input DNA which has not undergone chromatin immunoprecipitation. The distribution of ChIP sample reads is compared to the distribution of Input DNA sample reads when calling peaks for the ChIP sample. Peaks identified with ChIP-Seq can be used to annotate regions of the genome with respect to the role they play in gene expression [72], or they can be compared across samples from different biological conditions [73].

1.3.3 Transcriptomic data

Transcriptomic data is generated by profiling the transcriptomes (section 1.2.4) of cells from one or more biological samples, for a particular type of RNA. [32, 46, 74]. Transcriptomic data can be used to identify variation in gene expression across genes and samples, revealing insight into relationships between genes, as well as into the processes associated with biological conditions of interest. Some of the same high throughput technologies that are used to generate genomic data, microarrays and NGS platforms, are also used to generate transcriptomic data. Regardless of the technology, the generation of transcriptomic data begins with the extraction of RNA, followed by isolation of the type of RNA which is being studied, such as mRNA or miRNA. The isolated RNA is then fragmented and reverse transcribed to complementary DNA (cDNA), which may or may not be converted to complementary RNA (cRNA), so it can be used as input for a microarray or NGS workflow.

1.3.3.1 RNA-Seq

In an RNA-Seq experiment, cDNA fragments that have been created from sample RNA are sequenced by an NGS platform [74, 75]. RNA-Seq experiment data can be used to quantify gene expression, as well identify novel transcripts and splicing patterns. For the purpose of comparing gene expression across biological conditions, multiple samples can be sequenced together by attaching barcodes to fragments

during library preparation. Variations in the workflow include whether to sequence the fragments in one direction (single-end) or from both ends (paired-end), and whether to prepare the library so that information about the strand is preserved for the corresponding RNA transcript (stranded RNA-seq). Paired-end sequencing and stranded RNA-seq lead to improved alignment, gene annotation and transcript reconstruction. After sequencing and assessment of Phred scores, the trimmed and filtered reads are mapped to a reference genome using a tool, such as STAR [76], that is able to account for RNA splicing. Mapped reads are then used to generate counts for genes, exons or transcripts.

1.3.4 Multi-omics data

My main PhD research project was concerned with the analysis of multi-omics data, a type of data that has become available in recent years [45, 77]. A multi-omics dataset consists of multiple types of omics data that have been generated for the same samples.

The analysis of a multi-omics dataset is potentially more informative than the analysis of a dataset containing only a single type of omics data [78–80]. The analysis of a particular type of omics data can reveal molecular patterns that provide understanding of some biological conditions of interest. However, different types of omics data portray different aspects of the biological processes carried out within and between cells. Thus, a particular type of omics data may offer only a partial view of the underlying biological system, and the information obtained from the analysis of a particular type of omics data can be complementary to that obtained from another type. In addition, the analysis of a single type of omics data may reveal only limited insight into the interdependence between biological molecules of different types. Therefore, the analysis of multi-omics data promises improved understanding of biological conditions, by offering a more comprehensive view of the underlying biological system being studied [81].

In chapter 4, I present the results from my main PhD research project, describing the implementation, and evaluation, of an approach for the analysis of multi-omics data. I evaluated this approach with publicly available multi-omics data from The Cancer Genome Atlas (TCGA) project [82]. TCGA collected genomic, epigenomic, transcriptomic and proteomic data from matched cancer and normal samples, from over 11,000 patients across 33 cancer types. TCGA data is publicly available from an online portal called the GDC Data Portal `https://portal.gdc.cancer.gov/`. I used an R package called TCGAbiolinks [83] to obtain a subset of the available TCGA data. I obtained and used four types of omics data from the cancer samples:

- **Genomic data**: somatic variants (section 1.3.1.2) that had been detected in whole exome sequencing data (1.3.1.1) from cancer samples, based on comparisons with matched normal samples. The somatic variants were SNVs and small indels, which are categorized together as *simple nucleotide variation* on the GDC portal website.
- **Epigenomic data**: DNA methylation (section 1.3.2.1) profiles collected with

HumanMethylation450 BeadChip microarray technology, a predecessor to the MethylationEPIC (EPIC) BeadChip, that has probes targeting 450K CpG sites.

- **Transcriptomic data**: mRNA RNA-Seq counts and miRNA RNA-Seq counts (section 1.3.3.1)

There was data for all four omics for 8,237 cancer samples, across 32 of the cancer types. For the remaining samples, there was data for only one, two, or three of the omics.

2 Omics Data Analysis

Contents

2.1 Introduction . . . 30
2.2 Bayesian statistics . . . 31
2.2.1 Empirical Bayes . . . 32
2.2.2 Markov chain Monte Carlo . . . 32
2.2.3 Variational inference . . . 33
2.3 Visual exploration and pre-processing . . . 34
2.3.1 Visual exploration . . . 35
2.3.2 Pre-processing . . . 38
2.3.2.1 Filtering . . . 39
2.3.2.2 Normalization . . . 39
2.3.2.3 Transformation . . . 40
2.4 Differential testing . . . 41
2.4.1 Parametric and nonparametric methods for omics data . . . 41
2.4.2 Differential methylation . . . 42
2.4.3 Differential gene expression . . . 44
2.5 Unsupervised Dimension Reduction . . . 45
2.5.1 Matrix factorization . . . 46
2.5.1.1 PCA . . . 47
2.5.1.2 ICA . . . 48
2.5.1.3 NMF . . . 49
2.5.1.4 Selecting K . . . 49
2.5.1.5 Interpretation and sparsity . . . 50
2.5.2 Autoencoders . . . 50
2.5.2.1 Variational autoencoders . . . 51
2.5.3 Nearest neighbors approaches . . . 52

2.1 Introduction

My PhD research was concerned with methods for analysis of omics data (section 1.3), generated from multiple samples from different biological conditions. There are numerous methods available for the analysis of such data [47, 78]. I worked with methods that are used to identify observed and/or latent variables associated with biological conditions of interest. In the context of omics data, the *observed variables* correspond to genomic locations, or types of biological molecules, that are profiled

during an omics experiment. Examples of observed variables in omics data include RNA-seq gene counts, and CpG site DNA methylation levels. In contrast, the values of *latent variables* are unobserved, and are inferred from observed data. If we are able to infer latent variables that are associated with underlying biological processes, then their values can provide insight into associations between these processes and biological conditions of interest.

For both my main PhD project, presented in chapter 4, as well as another project, presented in chapter 5, I worked on the implementation of methods that rely on Bayesian approaches to statistical analysis. Accordingly, this chapter starts with a section (2.2) that introduces relevant aspects of Bayesian statistics. This section is intended to serve as reference point for the theoretical aspects of some of the methods described in the remainder of the thesis.

The remainder of this chapter introduces omics data analysis concepts, and methods, that I worked with during the PhD. I worked both with datasets that contained a single type of omics data, as well as with multi-omics datasets. In this chapter the focus is on the analysis of the former, although the methods and concepts can be extended for application to the latter as well. In section 2.3, I describe how omics data is visually explored, and pre-processed, prior to subsequent analysis. Then I introduce differential testing methods (section 2.4), used to identify variables associated with biological conditions of interest, followed by unsupervised dimension reduction (section 2.5), used to infer latent variables.

In chapter 3, I describe how these omics data analysis methods can be extended and/or enhanced, either in the form of multi-omics data analysis, or with the incorporation of prior information into an omics data analysis.

2.2 Bayesian statistics

Bayesian approaches to statistical analysis are widely used for extracting insights from omics data [84, 85]. Suppose an $N \times D$ matrix $\boldsymbol{Y}$ contains data for N observations (samples in the omics data context) and D observed variables. We assume that $\boldsymbol{Y}$ can be modeled with some joint sampling distribution $p(\boldsymbol{Y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes a set of latent values. The latent values in $\boldsymbol{\theta}$ may be parameters that define the distributions of the observed variables for different biological conditions. Alternatively, they may represent some underlying processes that contribute to the values of such parameters. We wish to make some inferences about $\boldsymbol{\theta}$. In a Bayesian approach, we assume, before seeing the observed data, that $\boldsymbol{\theta}$ is a multivariate random variable with joint prior distribution $p(\boldsymbol{\theta})$. The joint prior distribution can be conditional on a set of hyperparameters $\boldsymbol{\eta}$. To make inferences about $\boldsymbol{\theta}$, we estimate a joint posterior distribution, $p(\boldsymbol{\theta}|\boldsymbol{Y})$, based on Bayes' rule:

$$p(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(\boldsymbol{Y}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \tag{2.1}$$

In some situations, it is possible to use a conjugate prior distribution. When a conju-

gate prior is used, the posterior and prior distributions are from the same distribution family, meaning the posterior distribution has an analytical solution [86]. In other situations, the solution for the posterior distribution is either very complicated or intractable, and it can be approximated with a method such as Markov Chain Monte Carlo (section 2.2.2) or Variational Inference (section 2.2.3).

2.2.1 Empirical Bayes

In the analysis of omics data, it is common to conduct a separate statistical test for each observed variable [87]. I refer to this as differential testing (section 2.4) throughout this document. For each observed variable, the test relies on the estimation of values for unobserved parameters, which can be problematic as omics data is often generated from a small number of samples. In differential testing of omics data, the large number of parallel tests enables the use of a statistical technique called empirical Bayes [88]. In empirical Bayes, values are not specified for the hyperparameters $\boldsymbol{\eta}$; instead an estimate, $\hat{\boldsymbol{\eta}}$, is calculated using observed data. The posterior distribution is then estimated using $\hat{\boldsymbol{\eta}}$. In differential testing of omics data, $\hat{\boldsymbol{\eta}}$ might consist of some empirical Bayes values calculated using data from all observed variables, resulting in moderated test statistics that can lead to improved statistical inference when the number of samples is low [89].

2.2.2 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a sampling technique that can be used to estimate a posterior distribution [85]. A Markov chain [86] is a sequence of random vectors $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \boldsymbol{\theta}^3, \ldots$ with the so-called Markov property

$$\Pr\left(\boldsymbol{\theta}^k \in A | \boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{k-1}\right) = \Pr\left(\boldsymbol{\theta}^k \in A | \boldsymbol{\theta}^{k-1}\right) \tag{2.2}$$

A Markov Chain begins with an initial $\boldsymbol{\theta}^1$, and subsequent values are generated using a transition kernel $q\left(\boldsymbol{\theta}^k | \boldsymbol{\theta}^{k-1}\right)$. In an MCMC approach [86, 90] to estimating properties of a posterior distribution of interest, $p(\boldsymbol{\theta}|\boldsymbol{Y})$, a transition kernel is used to generate a Markov chain that eventually converges to a stationary distribution equal to $p(\boldsymbol{\theta}|\boldsymbol{Y})$. The appeal of MCMC is that for a Markov chain with stationary distribution $p(\boldsymbol{\theta}|\boldsymbol{Y})$, and a function $h(\,)$ with finite expectation under the stationary distribution, the *Ergodic Theorem* gives

$$\lim_{s\to\infty} \sum_{k=1}^{s} \frac{h\left(\boldsymbol{\theta}^k\right)}{s} = \int h(\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\boldsymbol{Y})\, d\boldsymbol{\theta} \tag{2.3}$$

If a Markov chain is run for long enough, then it may be considered to satisfactorily approximate the stationary distribution at some point. Typically, after an amount of *burn-in* (or *warm-up*) iterations, subsequent values in the chain can be used to calculate estimates for the posterior distribution [85]. In practice, one or more chains

of $m+n$ values are generated. The first m samples from each chain are treated as the warm-up set and discarded. The remaining n samples from each chain are used for making inferences about the posterior distribution. As n is finite, it is necessary to monitor convergence. A measure called the $\hat{R}$ score can be used to monitor convergence [91]. The $\hat{R}$ score is based on between-chain and within-chain variability, and declines to 1 as $n \to \infty$. Popular methods [85] for generating Markov chains for posterior distribution estimation include: Gibbs sampling [92], the Metropolis-Hastings algorithm [90], and Hamiltonian Monte Carlo [93].

2.2.3 Variational inference

Large datasets and/or complex statistical models can pose challenges for estimating a posterior distribution with MCMC. In such circumstances, a popular alternative is Variational Inference (VI) [85, 94, 95]. Rather than sampling from the posterior distribution, VI is based on optimization and is typically faster than MCMC. In VI, $p(\boldsymbol{\theta}|\boldsymbol{Y})$ is approximated by a simpler distribution $q(\boldsymbol{\theta})$, called a variational distribution. The objective in VI is to find values for $\boldsymbol{\lambda}$, the set of parameters for $q(\boldsymbol{\theta})$, that minimize the Kullback-Leibler (KL) divergence with $p(\boldsymbol{\theta}|\boldsymbol{Y})$. This KL divergence, $\mathrm{KL}\big(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{Y})\big)$, measures the difference between the expected values of $\log q(\boldsymbol{\theta})$ and $\log p(\boldsymbol{\theta}|\boldsymbol{Y})$, where the expectation is calculated with respect to $q(\boldsymbol{\theta})$:

$$\begin{aligned}\mathrm{KL}\big(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{Y})\big) &= \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{\theta}|\boldsymbol{Y})\right] \\ &= \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{\theta},\boldsymbol{Y})\right] + \log p(\boldsymbol{Y})\end{aligned} \tag{2.4}$$

It is difficult to calculate $\mathrm{KL}\big(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{Y})\big)$. Therefore, VI algorithms optimize $q(\boldsymbol{\theta})$ by using a function called the evidence lower bound (ELBO) as the objective function:

$$\begin{aligned}\mathrm{ELBO}(\boldsymbol{\lambda}) &= \mathbb{E}_q\left[\log p(\boldsymbol{\theta},\boldsymbol{Y})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] \\ &= \mathbb{E}_q\left[\log p(\boldsymbol{Y}|\boldsymbol{\theta})\right] + \mathbb{E}_q\left[\log p(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] \\ &= \mathbb{E}_q\left[\log p(\boldsymbol{Y}|\boldsymbol{\theta})\right] - \mathrm{KL}\big(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})\big)\end{aligned} \tag{2.5}$$

Maximizing $\mathrm{ELBO}(\boldsymbol{\lambda})$ is equivalent to minimizing $\mathrm{KL}\big(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{Y})\big)$, as equations 2.4 and 2.5 can be combined to give

$$\mathrm{KL}\big(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{Y})\big) = -\mathrm{ELBO}(\boldsymbol{\lambda}) + \log p(\boldsymbol{Y}) \tag{2.6}$$

It is common to partition $\boldsymbol{\theta}$ into J subsets, each of size one or more. The subsets are assumed to be mutually independent, resulting in a factorized joint variational distribution that takes the form

$$q(\boldsymbol{\theta}) = \prod_{j=1}^{J} q_j(\boldsymbol{\theta}_j) \tag{2.7}$$

This approach, known as mean-field variational inference, results in a simplified optimization problem. It has been shown that mean-field variational inference provides

consistent estimates of posteriors means, yet underestimates posterior variances and covariances.

A common approach for solving the mean-field variational inference optimization problem is coordinate ascent variational inference (CAVI) [94, 95]. In the CAVI approach, each $q_j(\boldsymbol{\theta}_j)$ is optimized iteratively while all $q_l(\boldsymbol{\theta}_l), l \neq j$, are held fixed. At each iteration, the algorithm updates a single $q_j(\boldsymbol{\theta}_j)$ by setting

$$q_j(\boldsymbol{\theta}_j) \propto \exp\left\{\mathbb{E}_{q_{-j}}\left[\log p(\boldsymbol{\theta}, \boldsymbol{Y})\right]\right\} \tag{2.8}$$

where $\mathbb{E}_{q_{-j}}$ denotes the expectation with respect to $\prod_{l \neq j} q_l(\boldsymbol{\theta}_l)$. Convergence of the algorithm is assessed based on ELBO(q), for which a local optimum can eventually be found. For my main PhD research project, presented in chapter 4, I implemented an algorithm that uses the CAVI approach to variational inference.

An alternative approach for finding optimal $\boldsymbol{\lambda}$ is stochastic optimization [95–97]. Stochastic VI (SVI) is based on sampling from $\boldsymbol{Y}$ and $q(\boldsymbol{\theta})$ to calculate an estimate of $\nabla_{\boldsymbol{\lambda}} f$ (ELBO), the gradient of some function of ELBO$(\boldsymbol{\lambda})$. At each iteration of an SVI algorithm, $\nabla_{\boldsymbol{\lambda}} f$ (ELBO) is estimated and then used for updating the values of $\boldsymbol{\lambda}$ [98]. For another project, presented in chapter 5, I implemented a modification to an algorithm that uses the SVI approach to variational inference.

2.3 Visual exploration and pre-processing

During my PhD, I worked with differential testing (section 2.4), and unsupervised dimension reduction (section 2.5). These omics data analysis methods, as well as others, rely on the estimates of distributional parameters, test statistics, and latent variable values.

Omics data (section 1.3) may contain values of low quality, due to measurement inaccuracies, or process errors, that can influence these estimates. The calculations of necessary estimates can also be influenced by signals, prevalent in omics data, that are unrelated to the biological conditions being studied [47, 78]. Batch effects [99], for example, are signals that can arise when, during the creation of omics data, samples are processed in different batches. Different batches of samples might be processed at different times, and/or by different handlers, and/or with different materials, and perhaps even at different locations. Batch effects may impact statistical power, reproducibility and generalizability. In addition to batch effects, there are often other unwanted signals in omics data. In DNA methylation microarray data (section 1.3.2.1), values may be influenced by technical biases, such as background signal, and biases resulting from the use of different probe types and probe signal colors [100, 101]. In RNA-seq data, unwanted signals can be present due to differences in total read counts per sample, as well as due to differences in how those counts are proportionally distributed within samples [75, 102].

Visual exploration, and pre-processing, can help identify, and correct for, data quality issues, as well as unwanted and/or unexpected signals, and are therefore

important steps in an omics data analysis [47, 78, 87]. Visual exploration is also helpful for checking, and interpreting, the outputs of omics data analysis methods.

Additionally, omics data may be in a form that poorly matches the assumptions underlying the calculation, or testing, of estimates, while high dimensionality can reduce the power of statistical tests, and large differences in the scales of observed variables can cause subsets of observed variables to have too much influence on estimations. These issues can also be addressed during pre-processing

In the following sections, I discuss how visual exploration (section 2.3.1), and pre-processing (section 2.3.2), can be used for the aforementioned purposes.

2.3.1 Visual exploration

At the beginning of an omics data analysis workflow, it is common to evaluate the data based on quality control metrics, for which visual exploration is helpful. For example, with DNA methylation microarray data (section 1.3.2.1), a detection p-value can be calculated for each CpG site and sample, for which a high value indicates low confidence in the accuracy of the methylation measurement. Plotting the proportion of CpG sites that have a detection p-value over a threshold, by sample, can indicate whether any samples should be omitted from subsequent analysis (section 2.3.2.1).

Visual exploration of the distributions of observed variables can also help with quality control, as widespread abnormalities may indicate issues with overall data quality. Plotting the distributions of observed values, and comparing them across samples, also enables the identification of outliers, samples whose omics profiles are unexpectedly dissimilar to those of samples from the same biological conditions. The identification of such samples may necessitate deeper investigation of their omics profiles, in case there are erroneous values. If there is confidence in the accuracy of the omics profiles of such samples, their existence may necessitate giving consideration to the groupings used in subsequent differential testing. Figure 2.1 shows density plots of DNA methylation microarray Beta-values from type 2 facioscapulohumeral muscular dystrophy (FSHD2) and Bosma arhinia and microphthalmia syndrome (BAMS) patients (chapter 6), which I made prior to differential testing. One of the FSHD2 samples had a slightly different distribution of Beta-values, but, after further inspection (for example Figure 2.4), I did not believe it was necessary to remove the sample from further analysis.

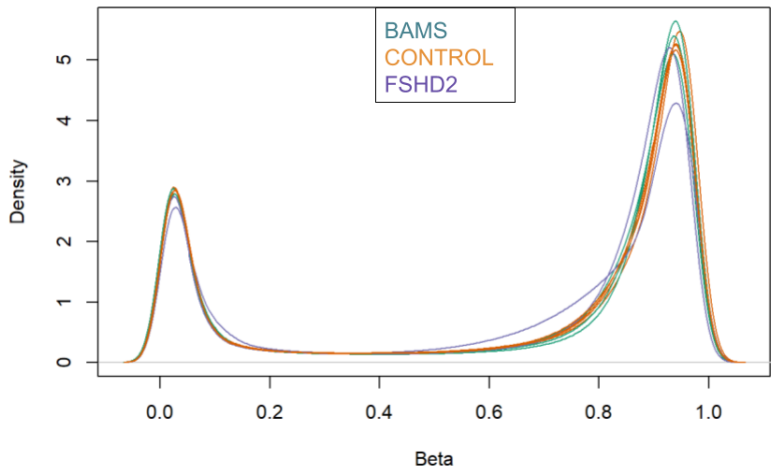


Figure 2.1 – **DNA methylation density plots**. Each line is a density plot of the DNA methylation Beta-values for a particular sample, colored by disease status.

Samples can also be compared by plotting latent variable values, inferred with unsupervised dimension reduction, and checking whether samples from the same known biological condition cluster together. Figure 2.2, for example, shows the values for the first two principal components inferred with PCA (section 2.5.1.1), from RNA-Seq data (section 1.3.3.1) from FSHD2 and BAMS patients (chapter 6). Each plotted data point is a sample, colored by disease status to allow inspection of whether samples with the same status clustered together.

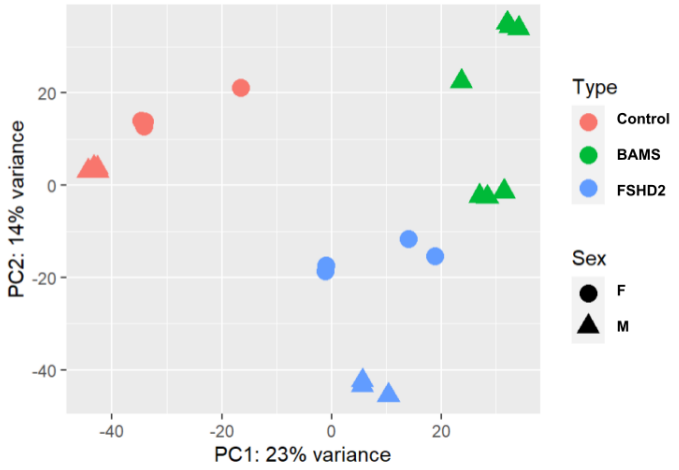


Figure 2.2 – **Plot of PCA values for RNA-Seq data samples**. The values of the first two principal components, inferred from an RNA-Seq dataset, are plotted for each sample. The x-axis is the value for the first principal component (PC1), and y-axis is the value for the second principal component (PC2)

It is also common to plot latent variable values to identify batch effects. If batch

effects are detected, they may be removed during pre-processing (section 2.3.2.2), or accounted for during subsequent analysis. In Figure 2.3, I plotted the values of latent variables inferred with UMAP (section 2.5.3), from the output of a MOFA (section 3.3.4.2) factorization of TCGA multi-omics data (section 1.3.4). Each point is a sample, from one of two cancer types, with the shape of the point representing the cancer type, and the color representing a batch identifier. I made plots like these as part of my work described in chapter 4, in order to check for batch effects in TCGA data.

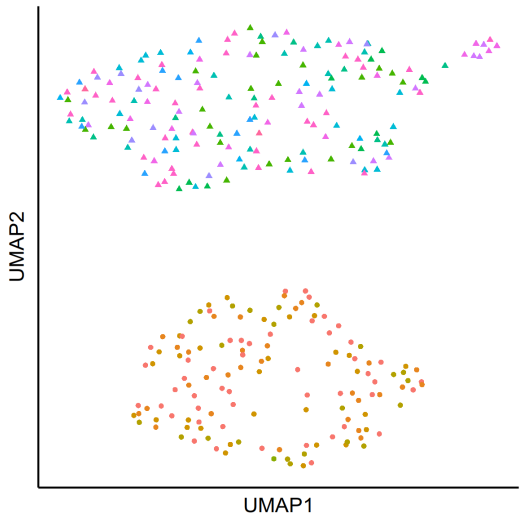


Figure 2.3 – **UMAP values for samples from two different cancer types**. The shape is the cancer type, the color is the batch. UMAP variables were inferred from the output of a MOFA factorization of TCGA multi-omics data.

Another way to compare samples, either before or after an analysis, is to visually explore the outputs of a clustering algorithm. An example of a clustering algorithm is agglomerative hierarchical clustering [103]. In agglomerative hierarchical clustering, each sample is initially a cluster of size 1. At each iteration of the algorithm, pairwise dissimilarities are calculated between clusters, and the two clusters with the lowest pairwise dissimilarity are merged. The algorithm stops when all samples are in a single cluster, resulting in a hierarchy of cluster assignments. Agglomerative hierarchical clustering can be used to assign samples to clusters, without the need to stipulate the number of clusters in advance. Rather, the number of desired clusters can be chosen after the algorithm has run, and the level of the hierarchy with that number of clusters is used for cluster assignment. Alternatively, the entire hierarchy can be visually explored, with the help of a dendrogram representing the hierarchy. For example, I produced the dendrogram in Figure 2.4 to check, prior to differential methylation analysis (section 2.4.2), the hierarchical clustering of DNA methylation microarray data from FSHD1, FSHD2, and BAMS patients (chapter 6), for which the pairwise dissimilarities were calculated using the vectors of Beta-values for each sample.

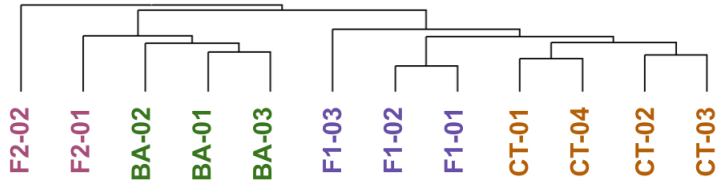


Figure 2.4 – **Dendrogram from hierarchical clustering of DNA methylation data**. Clustering was performed on Beta-values. The sample ids are colored by disease status

The visual exploration, of hierarchical clustering of samples, and observed or latent variables, combined with a heatmap, can be helpful when interpreting the outputs of an omics data analysis method, as well for the detection of anomalies. Figure 2.5 is a panel from a figure from the article presented in chapter 4. I created Figure 2.5 to display the results of hierarchical clustering of samples, using the latent variables inferred with MOTL, the unsupervised dimension reduction method I developed during my main PhD research project. With this Figure 2.5, my intention was to evaluate how well the latent variables separated cancer and normal samples.

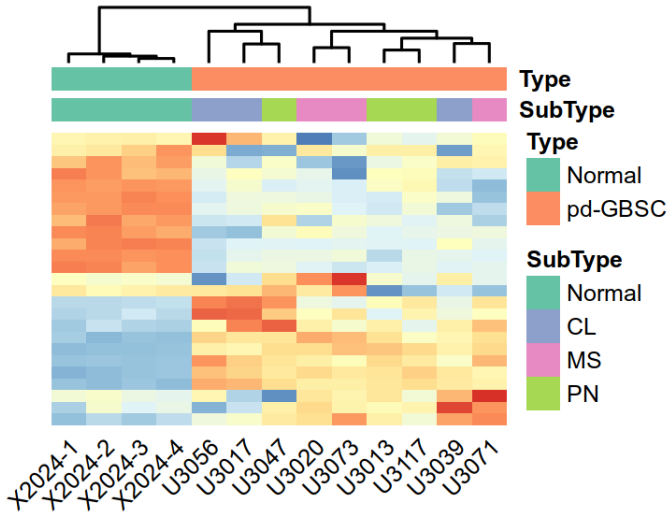


Figure 2.5 – **Hierarchical clustering combined with heatmap**. The clustering and heatmap are based on latent variable values inferred, with MOTL, from glioblastoma multi-omics data. The columns are samples, and the rows are latent variables.

2.3.2 Pre-processing

Common pre-processing steps for omics data [78], discussed in the following sections, include filtering (section 2.3.2.1), normalization (section 2.3.2.2), and transformation (section 2.3.2.3).

2.3.2.1 Filtering

It is common to filter omics data, prior to differential testing, or unsupervised dimension reduction, in order to remove observed variables, or samples, whose inclusion may adversely impact model fitting.

Initial filtering is typically carried out during quality control, whereby observed variables, and/or samples, are assessed based on quality control metrics, and annotations, and those deemed as low quality are excluded. With DNA methylation microarray data (section 1.3.2.1), for example, CpG sites for which there is low confidence in the accuracy of the values are typically removed, as are those which are annotated with characteristics, unrelated to the biological conditions being studied, known to influence methylation values [104, 105]. For the analysis of genomic variants, quality control filtering may involve removing variants whose predicted consequences (section 1.3.1.2) are not considered relevant to the analysis [106, 107]. Filtering during quality control may also involve excluding observed variables, or samples, that have more missing values than a given threshold, or for which visual exploration (section 2.3.1) has revealed abnormalities of concern.

For differential testing, it is common to perform additional filtering, after quality control, in order to optimize model fitting. This may involve removing observed variables with low overall variance, as this may improve statistical power in subsequent hypothesis testing [108]. Filtering out observed variables with low variance is also a common step prior to unsupervised dimension reduction, either to reduce computing time [109], or to address imbalances in dimensionality in multi-omics data [110].

2.3.2.2 Normalization

There are often signals in omics data, unrelated to the biological conditions being studied, that need to be accounted for before the data can be analyzed [47, 78]. Adjusting omics data to remove these signals is called normalization.

DNA methylation microarray data (section 1.3.2.1) is often normalized to account for the variation, between and within samples, resulting from batch effects, and/or technical biases unrelated to sample batch [100, 105]. An example of a normalization approach for DNA methylation microarray data is subset quantile normalization [22, 111], which groups probe signal intensity values into subsets based on genomic location and signal type (methylated or unmethylated). Subset quantile normalization forces the the distributions of type I and type II probe values to be the same within each subset. A reference distribution of probe values is calculated using all samples, then type I and type II probe values for individual samples are adjusted so that their quantiles match those of the reference distribution.

Prior to comparing RNA-Seq data across samples, it is necessary to normalize raw counts due to differences in total read counts per sample, as well as for differences in how those counts are proportionally distributed within samples [75, 102]. For comparing gene expression across samples, two well regarded RNA-seq data normalization methods [112, 113] are the DESeq [114] and Trimmed Mean of M-values (TMM) [115]

approaches. Both methods are based on the assumption that the majority of genes have similar expression levels across samples. They calculate a scaling factor for each sample, representing a relative total read count for these genes. For a given sample, a ratio is calculated for each gene count with respect to the corresponding count from a reference sample. Either the median of the ratios (DESeq), or trimmed mean of log ratios (TMM), are used to calculate the scaling factor for the sample. Scaling factors can be incorporated into subsequent statistical modeling, or used to adjust the total read counts of samples.

If, after the initial application of a normalization method, visual exploration of normalized omics data reveals the existence of batch effects (section 2.3.1), they can be estimated and removed prior to further omics data analysis. For example, ComBat [116] fits a linear model for each observed variable, estimates the additive and multiplicative effects of a batch variable, and then removes these estimated effects from the data. ComBat assumes normally distributed errors, and so is only suitable for continuous data. A potential consequence, of the explicit removal of batch effects, is that, when batch effects are confounded with biological conditions of interest, the removal of batch effects may adversely impact subsequent estimates of the effects of those biological conditions [78, 117]. An alternative way to account for remaining batch effects is to leave the data unadjusted, and incorporate batch identifiers into any models used for data analysis.

2.3.2.3 Transformation

The analysis of omics data often utilizes a statistical model. This may necessitate transforming the data to better match the assumptions underlying the model.

For the analysis of genomic variants, it might be necessary to aggregate the data by gene and sample, and then create binary values indicating the presence or absence of a variant [107].

For DNA methylation analysis based on the assumption of constant variance, it is recommended [69] to convert Beta-values to M-values with a logit transformation. This transformation is also common prior to using an analysis method that is based on the assumption that values are not restricted to being between 0 and 1 [107, 118].

When RNA-seq data is modeled as continuous data, a log transformation can be performed to make the counts approximately continuous [119, 120]. The log transformation also potentially leads to values with more constant variance. Other options for transforming RNA-seq counts to continuous values, with more constant variance, include the regularized log and variance-stabilizing transformations from the DESeq2 R package [121].

Prior to the application of unsupervised dimension reduction, it is common to transform omics data with standardization [78, 122, 123]. Standardization entails centering data by subtracting a constant from each value, and/or scaling the data by dividing each value by a constant. An example is variable-wise standardization, whereby the values for each observed variable are centered and scaled by their empirical means and standard deviations. Applying variable-wise standardization ensures

that observed variables have equal contributions to model fitting. For some joint matrix factorization methods (3.3), it is recommended to scale each omics matrix by a different constant, in order to control their relative contributions to model fitting.

2.4 Differential testing

In the work presented in chapters 5, and 6, I analyzed omics data with *differential testing*. In the analysis of a matrix of omics data, I use the term *differential testing* to refer to the popular analysis method [87] in which a separate statistical test is conducted for each observed variable. Omics data is usually high-dimensional, consisting of a large number of observed variables, some of which may be uninformative for understanding the biological conditions of interest. In this context, differential testing enables the identification of a subset of informative observed variables [47, 78].

Differential testing can also be used to analyze the outputs of unsupervised dimension reduction (section 2.5), whereby a separate statistical test is conducted for each latent variable [124, 125]. In chapter 4, for example, I present an article in which I describe how I used differential testing to identify latent variables that were associated with different cancer types.

The next section describes the basics of parametric, and nonparametric, differential testing methods for the analysis of an omics data dataset (section 2.4.1). Then follows more detail on differential testing with DNA methylation data (section 2.4.2), and differential testing with RNA-Seq data (section 2.4.3). For an example of how differential testing can be applied to the outputs of unsupervised dimension reduction, please see the article in section 4.3.

2.4.1 Parametric and nonparametric methods for omics data

For the analysis of omics data, popular methods for differential testing, such as limma [126] and DESeq2 [121], are parametric approaches. These approaches fit a generalized linear model (GLM) for each observed variable, in order to estimate distributional parameters that are used in hypothesis testing [127]. Suppose some omics data is generated for N samples and D observed variables, resulting in $N \times D$ matrix $\boldsymbol{Y}$. Element y_{nd} is the value in $\boldsymbol{Y}$ for the nth sample and the dth observed variable. The column vector, $\boldsymbol{y}_{:d}$, containing values for the dth observed variable, is modeled as

$$\mathbb{E}\left[\boldsymbol{y}_{:d}|\boldsymbol{X}\right] = g^{-1}\left(\boldsymbol{X}\boldsymbol{\beta}_d\right) \tag{2.9}$$

where $\boldsymbol{X}$ is a $N \times P$ design matrix for P explanatory variables. Element x_{np} is the value in $\boldsymbol{X}$ for the nth sample and the pth explanatory variable. The set of explanatory variables often includes factors indicating the biological conditions that characterize each sample; they may also include other variables such as batch identifiers or continuous covariates. $\boldsymbol{\beta}_d \in \mathbb{R}^{P\times 1}$ is a vector of coefficients to be estimated. $g()$ is the link function, chosen depending on the assumptions underlying the GLM. Depending on the type

of data, $g(\,)$ might simply be the identity function, in which case $\mathbb{E}\left[\boldsymbol{y}_{:d}|\boldsymbol{X}\right]$ is modeled as per equation 2.11.

When differential testing is based on a GLM, the objective is usually to compare coefficients corresponding to different biological conditions. In the case of RNA-seq data, for example, it might be of interest to compare the coefficient corresponding to a disease subtype with that of another disease subtype, and infer whether a gene has different expression levels between the two subtypes. Coefficients can be compared by using a $P \times J$ contrast matrix, $\boldsymbol{C}$, whose columns correspond to J different comparisons known as contrasts. A vector of estimated contrasts is calculated as:

$$\hat{\boldsymbol{\delta}}_d = \boldsymbol{C}^T \hat{\boldsymbol{\beta}}_d \tag{2.10}$$

where $\hat{\boldsymbol{\beta}}_d$ is the vector of estimated coefficients. A contrast estimate, $\hat{\delta}_{dj}$, can then be used to test the null hypothesis $H_0 : \delta_{dj} = 0$, where δ_{dj} is the jth element of $\boldsymbol{\delta}_d = \boldsymbol{C}^T \hat{\boldsymbol{\beta}}_d$. Testing different contrasts can lead to inferences about associations between the dth observed variable and the biological conditions of interest. The values of the test statistics depend on assumptions about the variability of $\boldsymbol{y}_{:d}$, and are moderated using the empirical Bayes technique (section 2.2.1). An important characteristic of differential testing is that each observed variable is tested separately, resulting in a large number of simultaneous hypothesis tests. Due to the large number of tests, the p-values are adjusted to control for the false discovery rate (FDR) [128].

An alternative to GLM based differential testing is to apply a nonparametric test to each observed variable, removing the need to make distributional assumptions about $\boldsymbol{y}_{:d}$ [78, 87]. For example, the Wilcoxian rank sum test [129] can be used to infer an association for the values of an observed variable with two biological conditions. The null hypothesis for the Wilcoxian rank sum test is $H_0 : F_A = F_B$, where F_A is the population distribution of values for some group A, and F_B the distribution for some group B. The observed values from both groups are combined and ranked. The ranks are summed for each group, with extreme values leading to a rejection of H_0 in favor of the alternative hypothesis of a difference in location between F_A and F_B. In the context of omics data analysis, groups A and B would correspond to different biological conditions. Another example is the Kruskal-Wallis test, an extension of the Wilcoxian rank sum test for three or more groups.

2.4.2 Differential methylation

For DNA methylation microarray data (section 1.3.2.1), differential testing involves testing each unique CpG site for differential methylation. If a null hypothesis is rejected for a CpG site, that site is considered to be a differentially methylated position (DMP) for the corresponding biological conditions. Testing for differential methylation is performed on $N \times D$ matrix $\boldsymbol{Y}$, in which element y_{nd} is either a Beta-value or M-value (section 2.3.2) for the nth sample and the dth CpG site. It is assumed that the data has already undergone pre-processing steps such as filtering and normalization. Estimated batch effects may or may not have been removed, depending on necessity

and preference for treatment of batch effects. It is common to use the limma tool for identifying DMPs [104, 105, 130]. The GLM used by limma assumes:

$$\mathbb{E}\left[\boldsymbol{y}_{:d}|\boldsymbol{X}\right] = \boldsymbol{X}\boldsymbol{\beta}_d \tag{2.11}$$

and

$$\text{var}\left[\boldsymbol{y}_{:d}|\boldsymbol{X}\right] = \boldsymbol{W}_d\sigma_d^2 \tag{2.12}$$

By default, $\boldsymbol{W}_d$ is a $N \times N$ identity matrix, although this can be adapted to allow for non constant variance or correlation between samples. $\hat{\boldsymbol{\beta}}_d$ is estimated with least squares, and used for calculating the residual variance, s_d^2, and vector of estimated contrasts, $\hat{\boldsymbol{\delta}}_d$.

Distributional assumptions (section 2.2) for $\hat{\boldsymbol{\delta}}_d$, s_d^2, $\boldsymbol{\beta}_d$, and σ_d^2, lead to the posterior mean:

$$\tilde{s}_d^2 = \mathbb{E}\left[\sigma_d^2|s_d^2\right] = \frac{\kappa_0 s_0^2 + \kappa_d s_d^2}{\kappa_0 + \kappa_d} \tag{2.13}$$

where κ_d is the residual degrees of freedom. κ_0 and s_0^2 are estimated with an empirical Bayes approach (section 2.2.1), using observed data for all CpG sites. The null hypothesis $H_0 : \delta_{dj} = 0$ can then be tested with a moderated t-statistic:

$$\tilde{t}_{dj} = \frac{\hat{\delta}_{dj}}{\tilde{s}_d^2\sqrt{v_{djj}}} \tag{2.14}$$

where v_{djj} is the jth diagonal element of matrix $\boldsymbol{\Upsilon}_d$, derived from $\boldsymbol{X}$, $\boldsymbol{W}_d$, and contrast matrix $\boldsymbol{C}$. After p-values have been calculated for each CpG site, they are adjusted to control for the FDR. CpG sites with FDR adjusted p-values lower than a pre-defined threshold are considered to be DMPs. DMP selection may also depend on the absolute values of estimated contrasts. A DMP can represent either hypermethylation or hypomethylation for a given biological condition, in comparison to another biological condition. If the contrast estimate for a DMP represents increased methylation for condition A, compared to condition B, then that DMP is hypermethylated for condition A, relative to condition B. If the contrast estimate represents decreased methylation, then the DMP is hypomethylated

Identification of DMPs can be followed by identification of differentially methylated regions (DMRs) [105]. The focus here is not on individual CpG sites with methylation levels that differ between biological conditions, but rather regions of the genome. An example of a method that can be used to identify DMRs is the Probe Lasso method [131]. Probe Lasso starts by selecting DMPs and extending a lasso in either direction. The size of the lasso depends on the genomic location of the DMP, due to differences in the density of the CpG sites targeted by the microarray probes. Mini candidate DMRs are created when the number of DMPs contained within a lasso is greater than a threshold. Overlapping or neighboring mini candidate DMRs are merged to create candidate DMRs. A p-value is then calculated for each candidate DMR. This is done by using Stouffer's method to combine the p-values, previously calculated with limma,

of all CpG sites contained within a candidate DMR. The FDR adjusted p-values of candidate DMRs are used to identify DMRs.

2.4.3 Differential gene expression

For RNA-seq data (section 1.3.3.1), differential testing involves testing each observed variable for differential expression. When testing genes for differential expression, a gene is considered to be a differentially expressed gene (DEG), for some biological conditions, if the corresponding null hypothesis is rejected for that gene. There are two commonly used approaches [78]: 1) transform the counts and use a classical tool from microarray data analysis, such as limma [126], that fits a model to continuous data, or 2) model the data as counts with a tool such as DESeq2 [121]. I describe these approaches in the following paragraphs.

In both approaches to testing for differential expression, testing is performed on $N \times D$ matrix $\boldsymbol{Y}$. Element y_{nd}, from $\boldsymbol{Y}$, is an expression value for the nth sample and the dth gene. It is assumed that the data has already undergone pre-processing steps such as filtering, and the calculation of a scaling factor for normalization (section 2.3.2).

When testing for DEGs with limma, each y_{nd} is a log-counts per million (log-cpm) value, calculated as:

$$y_{nd} = \log_2\left(\frac{k_{nd}+0.5}{s_n \sum_d k_{nd}+1.0} \times 10^6\right) \tag{2.15}$$

where k_{nd} is the raw count for the nth sample and the dth gene, and s_n is the scaling factor for the nth sample. Before testing with limma, a tool called voom [119] estimates the mean-variance relationship for expression values. The estimated mean-variance relationship is used to derive values for $\boldsymbol{W}_d$, the weighting matrix from equation 2.12.

When testing for DEGs with DESeq2, each y_{nd} is a raw count modeled with a negative binomial (NB) distribution:

$$p\left(y_{nd}|\boldsymbol{x}_{n:}\right) = \mathrm{NB}\left(\text{mean} = \mu_{dn}, \text{dispersion} = \alpha_d\right) \tag{2.16}$$

$$\mu_{dn} = s_n q_{dn} \tag{2.17}$$

$$\log_2 q_{dn} = \boldsymbol{x}_{n:}\boldsymbol{\beta}_d \tag{2.18}$$

where $\boldsymbol{x}_{n:}$ is the row vector, for the nth sample, from the design matrix, $\boldsymbol{X}$. s_n is the scaling factor for the nth sample, and μ_{dn} is the nth element of mean vector $\boldsymbol{\mu}_d \in \mathbb{R}^{N\times 1}$. Testing with DESeq2 starts by estimating values for the dispersion parameters. It does this by firstly fitting the GLM for each gene to obtain initial estimates of the mean vector, $\hat{\boldsymbol{\mu}}_d$, and the dispersion, $\hat{\alpha}_d$. It then obtains new dispersion estimates, conditional on $\hat{\boldsymbol{\mu}}_d$ values, as

$$\hat{\alpha}_d = \arg\max_{\alpha_d}\left(\sum_n \log p\left(y_{nd}|\boldsymbol{x}_{n:}\right) + \mathrm{CR}\left(\boldsymbol{X}, \hat{\boldsymbol{\mu}}_d, \alpha_d\right) + \Lambda\left(\alpha_d\right)\right) \tag{2.19}$$

where $\mathrm{CR}(\,)$ is an adjustment called the Cox-Reid bias adjustment. $\Lambda(\alpha_d)$ is derived

from the prior distribution assumed for α_d, whose hyperparameters are estimated with an empirical Bayes approach (section 2.2.1) using values from the initial GLM estimates. DESeq2 then estimates a coefficient vector, $\hat{\boldsymbol{\beta}}_d$, for each gene, conditional on the dispersion estimates:

$$\hat{\boldsymbol{\beta}}_d = \arg\max_{\boldsymbol{\beta}_d} \left(\sum_n \log p\left(y_{nd} | \boldsymbol{x}_{n:}\right) + \Lambda\left(\boldsymbol{\beta}_d\right) \right) \tag{2.20}$$

where $\Lambda\left(\boldsymbol{\beta}_d\right)$ is derived from the prior distribution (section 2.2) assumed for $\boldsymbol{\beta}_d$, whose hyperparameters are estimated with an empirical Bayes approach (section 2.2.1) using values obtained from a second GLM fit. Contrasts are estimated as per equation 2.10, and assigned p-values with Wald tests. FDR adjusted p-values are then used to select DEGs.

2.5 Unsupervised Dimension Reduction

Omics data is usually high-dimensional, consisting of a large number of observed variables, some of which may be uninformative. In section 2.4, I described how differential testing can be used to select a subset of informative observed variables. Another characteristic of omics data [132], is that subsets of molecules, represented by the observed variables, jointly participate in different biological processes (section 1.2). An alternative, or complementary, approach to the analysis of high-dimensional omics data, that is able to exploit this characteristic, is dimension reduction [47, 133]. In dimension reduction, the observed variables are represented by a smaller number of latent variables. These latent variables may yield insight into associations between biological processes and the biological conditions of interest [132], as well as help with the identification of informative observed variables [134]. Additionally, the reduction in dimension can help with visual exploration (section 2.3.1), as well as improve computational efficiency [133], and generalizability [80], when the latent variables are used as input for subsequent analysis tasks.

In my main PhD research project (chapter 4), as well as in one of my other projects (chapter 5), I focused on the application of unsupervised dimension reduction methods to omics data. Statistical analysis may be carried out in a supervised or unsupervised manner [103]. In supervised omics data analysis, the objective is to infer relationships between the omics data and a set of response, or classification, variables. In the unsupervised setting, the analysis does not include response variables, and the objective is to infer relationships between the omics data observed variables.

In the following sections, I introduce unsupervised dimension reduction methods in general, assuming they are applied to an $N \times D$ matrix, $\boldsymbol{Y}$, containing data for N observations and D observed variables. Element y_{nd}, in $\boldsymbol{Y}$, is the value for the nth observation and the dth observed variable. In the omics data context, the N observations are the samples, and the D observed variables correspond to the profiled molecules.

In unsupervised dimension reduction, the objective is to infer K latent variables that retain a sufficient proportion of the information contained in the D observed variables, with $K < D$. The resulting lower dimensional representations are used for many purposes including: visual exploration (section 2.3.1) to check for technical variation such as batch effects, as well as for large differences between samples from the same biological condition [78, 122]; using latent variables as inputs for clustering and predictive modeling [81, 133]; assessing latent variables for differences between conditions to yield candidates for differentially active biological processes [124, 135].

Common unsupervised dimension reduction approaches can be placed into one of three categories, based on how the latent variables are inferred: 1) Matrix factorization approaches factorize a matrix of observed data into the product of lower dimensional matrices; 2) Autoencoders employ neural networks to compress the data for each observation into a layer of latent variables; 3) Nearest neighbors approaches infer latent variables that preserve distances between neighboring observations

Prior to applying dimension reduction, $\boldsymbol{Y}$ is typically filtered (section 2.3.2.1), by removing observed variables, and samples, that are deemed to be of low quality, as well as potentially removing observed variables with low variance. For the dimension reduction implementations presented in this chapter, $\boldsymbol{Y}$ cannot contain missing values, meaning observed variables with missing values need to be removed. $\boldsymbol{Y}$ may be normalized (section 2.3.2.2) to remove unwanted signals, and is often transformed (2.3.2.3) by centering and/or scaling. Additionally, transformations yielding continuous, or non-negative, values are applied if necessary.

2.5.1 Matrix factorization

Matrix factorization (Figure 2.6) is an approach to unsupervised dimension reduction that represents $\boldsymbol{Y}$ by the product of two lower dimensional matrices

$$\boldsymbol{Y} \approx \boldsymbol{Z}\boldsymbol{W} = \sum_{k=1}^{K} \boldsymbol{z}_{:k}\boldsymbol{w}_{k:} \tag{2.21}$$

$\boldsymbol{Z}$ is an $N \times K$ matrix whose columns are latent variables [110, 136]. The values for the kth latent variable, across the observations, are in column vector $\boldsymbol{z}_{:k}$. $\boldsymbol{W}$ is a $K \times D$ matrix whose rows correspond to the same latent variables. The values for the kth latent variable, across the observed variables, are in row vector $\boldsymbol{w}_{k:}$.

For omics data matrix $\boldsymbol{Y}$, the latent variables ideally represent underlying biological signals, such as the activity of biological processes [135, 137]. They may also represent technical signals such as batch effects [122]. Matrix factorization assumes each value in $\boldsymbol{Y}$ depends on the sum of these signals. If the kth latent variable represents a subset of biological processes, the value of z_{nk} can be interpreted as the relative activity of these biological processes for the nth sample. The value of w_{kd} indicates how much the molecule corresponding to the dth observed variable contributes to these same biological processes.

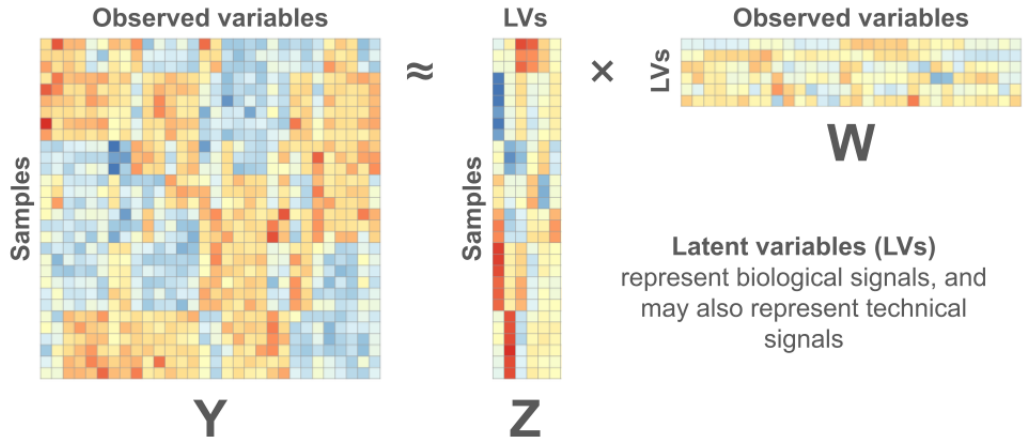


Figure 2.6 – **Matrix factorization overview**. With matrix factorization, omics data matrix $\boldsymbol{Y}$ is represented as the product of lower dimensional matrices $\boldsymbol{Z}$ and $\boldsymbol{W}$

Commonly used matrix factorization approaches [110, 132] for a single omics data matrix include principal components analysis (PCA), independent components analysis (ICA), and non-negative matrix factorization (NMF). These approaches, described in the following sections, differ in the assumptions and constraints they impose when inferring the factorization.

2.5.1.1 PCA

For a given K, matrix factorization with PCA produces $\boldsymbol{ZW}$ that minimizes the distance between $\boldsymbol{ZW}$ and $\boldsymbol{Y}$, when the distance is measured with the squared Frobenius norm [134]

$$||\boldsymbol{Y} - \boldsymbol{ZW}||_F^2 = \sum_{n,d} \left(y_{nd} - \boldsymbol{z}_{n:}\boldsymbol{w}_{:d}\right)^2 \tag{2.22}$$

PCA is commonly computed using singular value decomposition (SVD) [110, 138]. The SVD of $N \times D$ matrix $\boldsymbol{Y}$ with rank R ($R < \min(N, D)$) is

$$\boldsymbol{Y} = \boldsymbol{UDV}^T = \sum_{k=1}^{R} d_{kk}\boldsymbol{u}_{:k}\boldsymbol{v}_{:k}^T \tag{2.23}$$

where $\boldsymbol{D}$ is a diagonal matrix with $d_{11} \geq d_{22} \geq ... \geq d_{RR} \geq 0$, and both $\boldsymbol{U}$ and $\boldsymbol{V}$ consist of orthonormal column vectors corresponding to the diagonal entries in $\boldsymbol{D}$. The columns of $\boldsymbol{UD} = \boldsymbol{YV}$ are called principal components (PCs), and the columns of $\boldsymbol{V}$ are called loading vectors. The kth PC, $d_{kk}\boldsymbol{u}_{:k}$, contains coefficients for projections of the row vectors of $\boldsymbol{Y}$ onto loading vector $\boldsymbol{v}_{k:}$. A lower dimensional representation of $\boldsymbol{Y}$ with PCA is obtained by using $\boldsymbol{Z} = \boldsymbol{U}_K\boldsymbol{D}_K$ and $\boldsymbol{W} = \boldsymbol{V}_K^T$, where $\boldsymbol{U}_K$ contains the first K column vectors of $\boldsymbol{U}$, $\boldsymbol{D}_K$ contains the first K row and column vectors of $\boldsymbol{D}$, and $\boldsymbol{V}_K$ contains the first K column vectors of $\boldsymbol{V}$.

The proportion of the variance of $\boldsymbol{Y}$ explained by the first K PCs is

$$\frac{\text{trace}\left(\boldsymbol{D}_K^2\right)}{\text{trace}\left(\boldsymbol{D}^2\right)} \tag{2.24}$$

This means that latent variables obtained with PCA have decreasing variance and can be added sequentially based on a threshold for the total, or the change in, proportion of variance explained.

There are no explicit distributional assumptions for PCA, however if y_{nd} values are assumed to be conditionally independent with

$$p\left(y_{nd}|\boldsymbol{z}_{n:},\boldsymbol{w}_{:d}\right) = \text{Normal}\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d},\sigma^2\right) \tag{2.25}$$

and z_{nk} values are assumed to be independent with

$$p\left(z_{nk}\right) = \text{Normal}\left(0,1\right) \tag{2.26}$$

then the maximum likelihood estimate [139] of $\boldsymbol{W}$ is

$$\boldsymbol{W}_{ML} = \left(\boldsymbol{V}_K\left(\boldsymbol{D}_K^2 - \sigma^2\boldsymbol{I}_K\right)^{1/2}\right)^T \tag{2.27}$$

where $\boldsymbol{I}$ is the $K \times K$ identity matrix.

An implementation of PCA can be found in the *prcomp* function in the stats R package.

2.5.1.2 ICA

ICA [135, 140] is based on the assumption that the values in $\boldsymbol{Y}$ are the result of mixtures of independent signals

$$\boldsymbol{Y} \approx \boldsymbol{S}\boldsymbol{A} \tag{2.28}$$

where $\boldsymbol{S}$ contains the signals for each observation, and $\boldsymbol{A}$ contains the mixture coefficients for the observed variables. It is assumed that the columns of $\boldsymbol{S}$ are statistically independent latent variables with non-Gaussian distributions. ICA typically begins with the SVD lower dimensional representation $\boldsymbol{Y}_K = \boldsymbol{U}_K\boldsymbol{D}_K\boldsymbol{V}_K^T$. This is followed by the transformation called whitening [138], producing matrix $\tilde{\boldsymbol{U}}_K$ whose orthogonal column vectors each have mean equal to zero and unit variance

$$\tilde{\boldsymbol{U}}_K = \boldsymbol{Y}_K\boldsymbol{V}_K\boldsymbol{D}_K^{-1}\sqrt{N-1} \tag{2.29}$$

A matrix $\boldsymbol{M}$ is estimated to fit $\boldsymbol{S} = \tilde{\boldsymbol{U}}_K\boldsymbol{M}$, by maximizing the non-Gaussianity of the columns of $\boldsymbol{S}$; this is followed by estimation of $\boldsymbol{A}$. A lower dimensional representation of $\boldsymbol{Y}$ with ICA is obtained by using $\boldsymbol{Z} = \boldsymbol{S}$ and $\boldsymbol{W} = \boldsymbol{A}$.

An implementation of ICA can be found in the fastICA R package.

2.5.1.3 NMF

NMF [141, 142] finds a lower dimensional representation $\boldsymbol{ZW}$ that minimizes a distance between $\boldsymbol{ZW}$ and $\boldsymbol{Y}$. The factorization is not constrained by orthogonality or independence requirements, but instead by the requirement of non-negativity for the elements of $\boldsymbol{Z}$ and $\boldsymbol{W}$. Common measures of distance used by NMF [143, 144] include the Frobenius norm, and a generalized KL divergence calculated as

$$\mathrm{KL}\left(\boldsymbol{Y}||\boldsymbol{ZW}\right) = \sum_{n=1}^{N}\sum_{d=1}^{D}\left(y_{nd}\log\frac{y_{nd}}{\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}} - y_{nd} + \boldsymbol{z}_{n:}\boldsymbol{w}_{:d}\right) \tag{2.30}$$

The choice of distance measure implies certain distributional assumptions and influences how the factorization is computed. When using the Frobenius norm, NMF is equivalent to maximum likelihood estimation under the assumption

$$p\left(y_{nd}|\boldsymbol{z}_{n:},\boldsymbol{w}_{:d}\right) = \mathrm{Normal}\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d},\sigma^2\right) \tag{2.31}$$

In contrast, when using generalized KL divergence, NMF is equivalent to maximum likelihood estimation under the assumption

$$p\left(y_{nd}|\boldsymbol{z}_{n:},\boldsymbol{w}_{:d}\right) = \mathrm{Poisson}\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}\right) \tag{2.32}$$

Both of these versions of NMF are implemented in the NMF R package [143].

2.5.1.4 Selecting K

It is necessary to select a value for K when obtaining a lower dimensional representation with matrix factorization, yet the optimal method for selecting K is an open research problem [132]. For PCA, a common method is to plot the proportion of variance explained by each latent variable and select K based on the elbow point of the curve [110]. Another common method for PCA is to factorize matrices derived from permuting $\boldsymbol{Y}$, to create a background distribution for each k, against which the proportion of variance explained by individual latent variables can be compared [135, 145]. It may be preferred to select K based on the factorization as a whole, rather than based on the individual latent variables. In this case, options for selecting K are based on factorizing $\boldsymbol{Y}$ using different values for K, and selecting the best factorization: using a threshold for total proportion of variance explained [110, 133]; using an estimate of the Bayesian Information Criterion for each K [135]; using an estimate of the stability of clusters derived from $\boldsymbol{Z}$ for each K [141]; comparing the distance between $\boldsymbol{ZW}$ and $\boldsymbol{Y}$, for each K, against a background distribution obtained from factorizing matrices derived from permuting $\boldsymbol{Y}$ [146]. For non-PCA methods, K can also be selected by first selecting an optimal K for PCA, and then increasing it by a percentage [136].

2.5.1.5 Interpretation and sparsity

An appeal of matrix factorization is that the values in $\boldsymbol{W}$ offer insight into relationships between observed and latent variables. This potentially results in better interpretability in comparison to other unsupervised dimension reduction approaches [147, 148]. When observed values correspond to genes, for example, each row of $\boldsymbol{W}$ can be assessed to infer associations with sets of genes known to be functionally related due to their joint participation in specific biological processes [132]. For omics data, ICA and NMF are considered to yield factorizations with better interpretability, compared to PCA [110, 135]. This improved interpretability is due to their focus on either the separation of independent signals (ICA), or representing the observed data as the sum of non-negative parts (NMF).

Additionally, NMF factorizations tend to produce sparse matrices, meaning only a small subset of the elements are non-zero [143]. In the case of omics data, this sparsity can help in inferring associations between the rows of $\boldsymbol{W}$ and functionally related sets of molecules. The level of sparsity can be controlled for NMF [149], by computing the factorization with a non-negativity constrained alternating least squares approach that incorporates a regularization penalty. An implementation of NMF, that allows for the level of sparsity to be controlled, can be found in the NMF R package [143]. A desired level of sparsity can also be obtained with PCA, using regularized SVD. This is based on the fact that SVD can be computed with alternating least squares. In regularized SVD, an alternating least squares approach that incorporates a regularization penalty is used to obtain a sparse PCA factorization [134], implemented as sPCA in the mixOmics R package [109].

2.5.2 Autoencoders

An autoencoder (AE) is a model for unsupervised dimension reduction, consisting of two neural networks called an encoder and a decoder [133, 150, 151]. The input for the encoder is $\boldsymbol{y}$, a vector of D observed variables known as the input layer. The output of the encoder is $\boldsymbol{z}$, a vector of K latent variables known as the bottleneck layer. An important characteristic of the bottleneck layer is that $K < D$, meaning $\boldsymbol{z}$ is a lower dimensional representation of $\boldsymbol{y}$.

The encoder typically contains, between $\boldsymbol{y}$ and $\boldsymbol{z}$, consecutive hidden layers of latent variables. The values of latent variables, in each layer, are the outputs of functions whose inputs are values of variables from preceding layers. The decoder takes vector $\boldsymbol{z}$ as input. The output of the decoder is $\hat{\boldsymbol{y}}$, a vector of D elements known as the reconstruction layer, which is a reconstruction of $\boldsymbol{y}$. The decoder typically contains, between $\boldsymbol{z}$ and $\hat{\boldsymbol{y}}$, consecutive hidden layers of latent variables, whose values are outputs of functions of variables from preceding layers.

For an $N \times D$ matrix $\boldsymbol{Y}$, containing data for N observations and D observed variables, the input to an AE, for the nth observation, is $\boldsymbol{y}_{n:}$, the nth row vector of $\boldsymbol{Y}$. For each $\boldsymbol{y}_{n:}$, an AE derives a bottleneck layer vector $\boldsymbol{z}_{n:}$ and reconstruction layer vector $\hat{\boldsymbol{y}}_{n:}$. The AE jointly optimizes the parameters of the encoder functions, $\boldsymbol{\phi}$, and parameters

the decoder functions, $\boldsymbol{\theta}$, based on a distance between each $\boldsymbol{y}_{n:}$ and its reconstruction layer vector $\hat{\boldsymbol{y}}_{n:}$.

2.5.2.1 Variational autoencoders

A part of my PhD research, presented in chapter 5, was concerned with the variational autoencoder (VAE), a recent autoencoder variant. VAE architectures [151, 152] are based on distributional assumptions for observed and latent variables, and use variational inference (section 2.2.3) for optimization. In a VAE (Figure 2.7), the parameters of the encoder functions, $\boldsymbol{\phi}$, determine the parameter values of a variational distribution for the latent variables in the bottleneck layer, $q(\boldsymbol{z}|\boldsymbol{y})$. The parameters of the decoder functions, $\boldsymbol{\theta}$, determine the values of $\hat{\boldsymbol{y}}$, which serve as parameter values of a sampling distribution for the observed variables in the input layer, $p(\boldsymbol{y}|\boldsymbol{z})$.

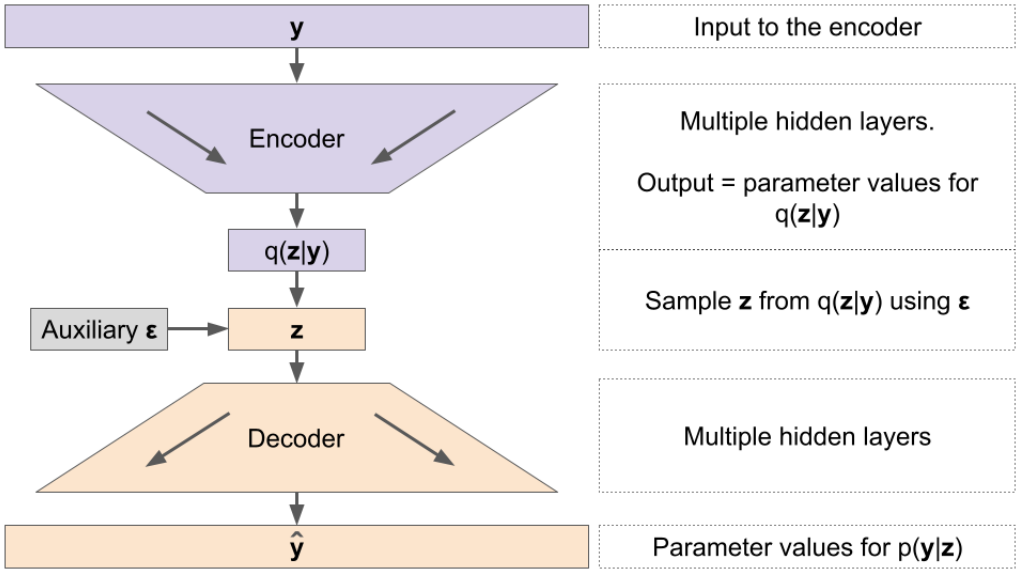


Figure 2.7 – **Variational autoencoder overview**

For matrix $N \times D$ matrix $\boldsymbol{Y}$, containing data for N observations and D observed variables, the objective is to maximize

$$\begin{aligned} \text{ELBO}(\boldsymbol{\phi},\boldsymbol{\theta}) &= \mathbb{E}_q\left[\log p(\boldsymbol{Y}|\boldsymbol{Z})\right] - \text{KL}\left(q(\boldsymbol{Z}|\boldsymbol{Y})||p(\boldsymbol{Z})\right) \\ &= \mathbb{E}_q\left[\log p(\boldsymbol{Z},\boldsymbol{Y})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{Z}|\boldsymbol{Y})\right] \end{aligned} \tag{2.33}$$

where $\boldsymbol{Z}$ is the $N \times K$ matrix whose columns are the latent variables from the bottleneck layer, and $p(\boldsymbol{Z})$ is a prior distribution assumed for $\boldsymbol{Z}$.

An example of a VAE algorithm is the Auto-Encoding Variational Bayes (AEVB) algorithm [153]. The AEVB algorithm is based on stochastic gradient ascent. At each iteration of the algorithm, a random minibatch of M observations, $\mathcal{M}$, is used to

calculate an empirical estimate of ELBO$(\boldsymbol{\phi},\boldsymbol{\theta})$

$$\hat{\mathscr{L}}_{AEVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{M}}\right) = \frac{N}{M}\sum_{n\in\mathscr{M}}\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{2.34}$$

$\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$ is the Stochastic Gradient Variational Bayes (SGVB) estimator for the nth observation, calculated as either:

$$\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) = \frac{1}{L}\sum_{l=1}^{L}\left(\log p\left(\boldsymbol{y}_{n:},\boldsymbol{z}_{n:}^{(l)}\right) - \log q\left(\boldsymbol{z}_{n:}^{(l)}|\boldsymbol{y}_{n:}\right)\right) \tag{2.35}$$

or

$$\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) = -\mathrm{KL}\left(q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right)||p\left(\boldsymbol{z}_{n:}\right)\right) + \frac{1}{L}\sum_{l=1}^{L}\left(\log p\left(\boldsymbol{y}_{n:}|\boldsymbol{z}_{n:}^{(l)}\right)\right) \tag{2.36}$$

where $\boldsymbol{z}_{n:}^{(l)}$ is a value sampled from $q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right)$, and the choice of calculation depends on whether the KL divergence term can be solved analytically.

The algorithm does not sample from $q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right)$ directly, but instead samples an auxiliary variable $\boldsymbol{\epsilon}^{(n,l)} \in \mathbb{R}^K$ from some distribution $p\left(\boldsymbol{\epsilon}\right)$. A differentiable function, whose inputs are $\boldsymbol{\epsilon}^{(n,l)}$ and the outputs of the encoder, is then used to compute a value for $\boldsymbol{z}_{n:}^{(l)}$. This is known as the reparameterization trick, and it enables differentiation of $\hat{\mathscr{L}}_{AEVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{M}}\right)$ with respect to $\boldsymbol{\phi}$. At each iteration, $\hat{\mathscr{L}}_{AEVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{M}}\right)$ is used to estimate $\nabla_{\boldsymbol{\phi},\boldsymbol{\theta}}$ELBO, the gradients of ELBO$(\boldsymbol{\phi},\boldsymbol{\theta})$, and the values of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are updated with gradient ascent.

A trained VAE can be used to derive a $\boldsymbol{Z}$ matrix, containing values for each observation and latent variable. The values in $\boldsymbol{Z}$ can be based on the parameter values of $q\left(\boldsymbol{Z}|\boldsymbol{Y}\right)$, or sampling from $q\left(\boldsymbol{Z}|\boldsymbol{Y}\right)$ can be used to populate $\boldsymbol{Z}$. For omics data, a $\boldsymbol{Z}$ matrix derived with a VAE is potentially more informative for clustering and predictive modeling than $\boldsymbol{Z}$ derived with matrix factorization [133, 151], due to the ability of VAEs to model non-linearity underlying the observed data [154]. However, compared to matrix factorization, VAEs offer less interpretability with respect to the relationships between observed and latent variables [148]. I present a VAE implementation, called OntoVAE [125], in chapter 5.

2.5.3 Nearest neighbors approaches

Nearest neighbors approaches to unsupervised dimension reduction, such as t-SNE [155] and UMAP [147], are popular for transforming an $N \times D$ matrix $\boldsymbol{Y}$ into an $N \times K$ latent variable matrix $\boldsymbol{Z}$, when the intention is to use $\boldsymbol{Z}$ for visualization [78, 122] (section 2.3.1). Both of these algorithms calculate pairwise measures of proximity between observations in the original D-dimensional space. The proximities are weighted, with higher weighting given to neighboring observations than to more distant observations. They then iteratively populate $\boldsymbol{Z}$ with values that retain these proximities, thus retaining local structure as well as non-linearities underlying the

observed data.

For omics data, t-SNE and UMAP produce a $\boldsymbol{Z}$ matrix that is potentially more informative for visualizing heterogeneity between clusters of similar samples than matrix factorization. However, dimension reduction with t-SNE or UMAP does not offer insight into relationships between observed and latent variables. Additionally, these algorithms focus on local proximities, so it is hard to interpret the distances between clusters of samples. For $\boldsymbol{Y}$ with large D, computational efficiency may be improved by firstly applying matrix factorization to $\boldsymbol{Y}$, and then using the resulting $\boldsymbol{Z}$ as input for further dimension reduction with t-SNE or UMAP.

An implementation of UMAP, which I used during my PhD, can be found in the uwot R package.

3 Data integration

Contents

3.1 Introduction . . . 54
3.2 Using annotation resources . . . 55
3.2.1 Annotation resources . . . 55
3.2.2 Incorporation after model fitting . . . 57
3.2.3 Incorporation during model fitting . . . 59
3.3 Multi-omics matrix factorization . . . 60
3.3.1 The general joint matrix factorization model . . . 61
3.3.2 Approaches based on distance . . . 62
3.3.2.1 Extending PCA . . . 62
3.3.2.2 Extending NMF . . . 63
3.3.3 Approaches based on correlation and covariance . . . 63
3.3.4 Approaches based on distributional assumptions . . . 64
3.3.4.1 iClusterBayes . . . 65
3.3.4.2 MOFA . . . 66
3.3.5 Multi-omics data with few samples . . . 67
3.4 Transfer learning . . . 69
3.4.1 Transfer learning and omics data . . . 69
3.4.2 Transfer learning and matrix factorization . . . 70
3.4.3 Transfer learning based on MOFA . . . 71

3.1 Introduction

A central theme of my PhD research was: enhancing omics data analysis by using multiple types of omics data, and/or prior information. The analysis of a particular type of omics data can provide insight into a biological system, by revealing associations between biological molecules and biological conditions of interest. This insight can be improved upon by analyzing multi-omics data, and/or incorporating prior information into an analysis workflow. Here I use the term *prior information* to represent information that has already been obtained, from some analysis, prior to the omics data analysis into which it is incorporated. The use of multiple types of omics data, and/or prior information, can be categorized together within the field of *data integration*, defined by Nature Portfolio [156] as

> the process of combining data generated using a variety of different research methods in order to enable detection of underlying themes and, in computational biology and bioinformatics, biological principles.

As described in the following sections, I worked with three ways of enhancing omics data analysis using data integration: 1) the incorporation of prior information from annotation resources into an analysis workflow (section 3.2); 2) the analysis of multi-omics data with joint matrix factorization (multi-omics matrix factorization) (section 3.3); 3) the application of a transfer learning approach to multi-omics matrix factorization, enabling the incorporation of prior information (section 3.4).

3.2 Using annotation resources

The development and survival of living organisms depends on biological processes, consisting of molecular interactions, carried out within and between cells (section 1.2.1). Therefore an objective of omics data analysis may be to make inferences about which interactions and processes are associated with some biological conditions of interest (section 1.2.6).

The outputs from differential testing (section 2.4), and unsupervised dimension reduction (section 2.5), can be used to select, and/or rank, observed variables, with respect to their association with the biological conditions being studied. However, additional information is required to use these selections, and rankings, to infer associations between biological conditions and processes. This can be achieved by incorporating information from annotation resources into analysis workflows that involve differential testing and/or unsupervised dimension reduction.

The are multiple resources that contain annotations related to gene function. When the observed variables in an omics data analysis are genes or gene products, the information in such annotation resources can be incorporated based on direct annotations of the observed variables.

For other types of observed variables, there are less annotation resources available, but there are some. CpG sites (section 1.2.3), for instance, can be annotated based on the genomic region to which they belong. An alternative and/or complementary strategy, for observed variables that are not genes or gene products, is to firstly annotate the variables with the gene(s) they are associated with, and then use an annotation resource related to gene function.

3.2.1 Annotation resources

The are multiple resources that contain annotations for genes, populated based on prior experimental research. For example, the Gene Ontology (GO) knowledgebase [157] contains three classes of annotations related to gene function: molecular function, cellular component, and biological process. Molecular functions are actions carried out by individual gene products, or complexes of multiple gene products. Cellular components are locations, within cells, where these actions are carried out.

Biological processes relate to objectives accomplished by the actions of multiple gene products (section 1.2). The GO knowledgebase contains annotations for genes from multiple organisms. In the GO knowledgebase, the annotations are called *terms*, and genes are annotated with one or more terms from each class. Each of the three classes of annotations is a GO sub-ontology.

Due to the hierarchical nature of GO terms, each GO sub-ontology is a directed acyclic graph (DAG) (Figure 3.1). In a GO sub-ontology DAG, the terms are nodes in the graph, and the inclusion relationships between those terms are edges in the graph. More generic terms are followed by more specific terms, and eventually by genes as terminal nodes. If a gene is a terminal node for a particular GO term, whether they share an edge, or are connected through other GO terms, then the gene is annotated with that GO term. A GO sub-ontology can also be represented by gene sets, where each gene set contains all genes annotated with a specific GO term.

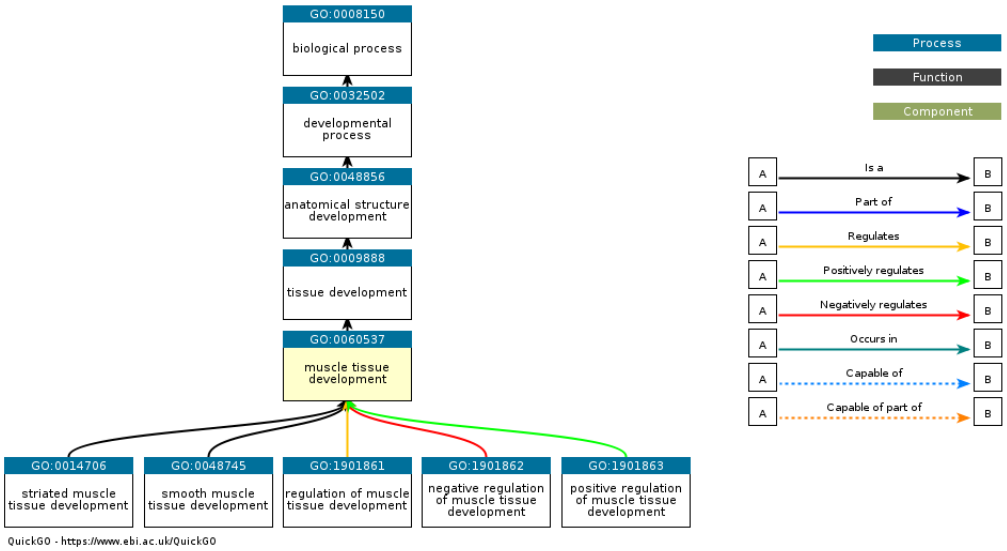


Figure 3.1 – **Muscle tissue development GO term**. Subgraph from the directed acyclic graph of GO biological process terms. This subgraph shows terms related to the *muscle tissue development* term. The nodes above the muscle tissue development node are related terms that are more generic. The nodes below are its child terms, more specific terms with which it shares an edge. Image taken from [158]

Most annotations in the GO knowledgebase are curated manually, while others are inferred computationally without being subject to manual curation.

Other sources of gene annotations include Reactome [159], and KEGG [160]. They contain gene annotations based on the participation of gene products in certain biological processes, called biological pathways, that consist of *ordered* molecular interactions. Both Reactome and KEGG represent biological pathways as networks of interactions (Figure 3.2), which are subject to manual curation. Reactome and KEGG pathways can also be represented as gene sets, containing the participating genes.

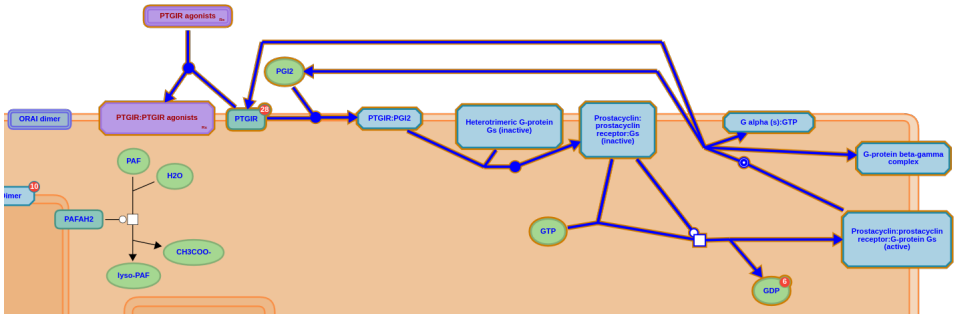


Figure 3.2 – **Reactome graph for the *Prostacyclin signalling through prostacyclin receptor* pathway**. This pathway is a component of the larger *Hemostasis* pathway for Homo Sapiens. Image taken from [161]

For observed variables that are neither genes nor gene products, annotations can be based on the roles played by their genomic locations in gene expression. For example, Broad ChromHMM tracks [72] segment the genome, according to chromatin state, for nine different human cell types. The segmentation is based on ChIP-Seq experiments that profiled different types of histone modifications associated with different roles played in gene expression. The data from these experiments was summarized by cell type, yielding a map of typical chromatin states for each cell type. Each resulting Broad ChromHMM track is a set of genome annotations for a different cell type. These annotations, based on chromatin state, correspond to the role the location is predicted to play in gene expression. For observed variables that are neither genes nor gene products, they can also be annotated with gene associations. An observed variable, such as a CpG site, can be associated with a gene if the genomic location it represents is close to the genomic location of that gene. The information about the gene(s) an observed variable is associated with, combined with information about the roles played by its genomic location in gene expression, can be used to further annotate the observed variable with a gene annotation resource such as the GO knowledgebase, KEGG or Reactome.

3.2.2 Incorporation after model fitting

The outputs of differential testing (section 2.4) typically include contrast estimates, and corresponding p-values, for the observed variables that were tested. For unsupervised dimension reduction (section 2.5), the outputs may include latent variable values associated with each observed variable. The $K \times D$ matrix $\boldsymbol{W}$ inferred with matrix factorization (section 2.5.1), for example, contains K latent variable values for each of D observed variables. It is assumed that the values of the observed variables can be represented as linear combinations of the values in $\boldsymbol{W}$. These outputs of differential testing and unsupervised dimension reduction can be used to select, and/or rank, observed variables (e.g., genes, genomic locations), with respect to their

association with the biological conditions being studied. However, these selections and rankings are not necessarily informative on their own. Therefore a common step after differential testing [102, 162], and dimension reduction [81, 132], is to combine the outputs with information from annotation resources.

For a selection of genes, or other genomic locations, over-representation analysis (ORA) can be used to incorporate information from an annotation resource [163, 164]. For example, when differential expression analysis (section 2.4.3) yields a list of DEGs, the list can be compared to the genes annotated with a specific GO biological process term. The genes from that GO term can be tested for over-representation in the DEG list, compared to the list of all genes that were tested for differential expression, using a method based on the hypergeometric distribution. Over-representation indicates that the biological processes may be associated with the biological conditions that were the object of the differential expression analysis. Usually, multiple gene sets, corresponding to different annotations, are tested for over-representation, and FDR adjusted p-values are used for assessing statistical significance.

This same procedure can be carried out for a list of DMPs found with differential methylation analysis (section 2.4.2). The missMethyl R package[162, 165], for example, includes functionality to use ORA to identify over-represented biological processes from a list of DMPs. The CpG sites tested for differential methylation can be annotated based on prior knowledge of which genes they are associated with. In the missMethyl approach, genes are given a value of 1 if they are associated with a DMP. Genes that are associated with a CpG, but not with any DMP, are given a value of 0. ORA is then performed using Wallenius' noncentral hypergeometric distribution, taking into account the fact that different genes are associated with different numbers of CpGs.

ORA can also be applied to the outputs of matrix factorization. For matrix factorization methods that impose sparsity on $\boldsymbol{W}$, each sparse latent variable vector, $\boldsymbol{w}_{k:}$, can be used for over-representation testing [149]. For a particular $\boldsymbol{w}_{k:}$, over-representation of a gene set is tested based on which observed variables have non-zero values in $\boldsymbol{w}_{k:}$, and which observed variables have values equal to zero.

In ORA, it is necessary to use a threshold to select a list of genes from a background set. Over-representation testing is then performed on the selected list. An alternative to ORA is a method known as gene set enrichment analysis (GSEA) [166, 167], in which it is not necessary to first select a list of genes from the background set. Instead, in GSEA, testing is applied to a ranked list of all the genes from the background set. For differential expression the rankings might be based on a function of the estimated contrasts and/or p-values [102, 164]. For matrix factorization methods that do not impose sparsity on $\boldsymbol{W}$, GSEA can be applied to each latent variable vector, $\boldsymbol{w}_{k:}$, with the ranking based on the values in $\boldsymbol{w}_{k:}$ [132, 135]. For a gene set characterized by a given annotation, such as a GO term, GSEA scans the ranked list to test whether the gene set is over-represented at the top or bottom of the list. For a collection of gene sets, permutations are used to calculate a normalized enrichment score (NES) and FDR q-value [168] for each gene set. A positive NES combined with a significant q-value indicates the gene set is over-represented at the top of the ranked list, while a statistically significant negative NES indicates it is over-represented at the bottom of

the list.

ORA and GSEA are popular methods for using the outputs of differential testing, and unsupervised dimension reduction, to infer associations between biological processes and some biological conditions of interest. In chapter 4, I present an article in which GSEA to the outputs of unsupervised dimension reduction, and in chapter 6, I present an article in which ORA was applied to the outputs of differential testing. It is worth mentioning that there are several other methods available for this purpose [169, 170]. One example is EnrichNet [171], which uses the genes, from a list of DEGs, as seeds for a random walk algorithm applied to a gene interaction network. EnrichNet uses the random walk algorithm to calculate distances, between the DEG list and gene sets representing biological processes, resulting in a ranked list of biological processes. Another example is sigPathway [172], which takes as input a vector of scores for genes. The scores may be the values calculated for a test statistic during differential testing, or the values in a $\boldsymbol{w}_{k:}$ vector. sigPathway then calculates a weighted average score value for each gene set, normalized for size, and tests each gene set for statistical significance using a permutation generated null distribution.

3.2.3 Incorporation during model fitting

For unsupervised dimension reduction, the information in an annotation resource can be incorporated during model fitting. This constrains the optimization with the objective of improving the interpretability of the outputs.

For example, Pathway-Level Information ExtractoR (PLIER) [136] is a matrix factorization algorithm (section 2.5.1) capable of incorporating such prior information. PLIER is designed to factorize $N \times D$ matrix $\boldsymbol{Y}$, containing gene expression data for N samples and D genes, by minimizing

$$\begin{aligned} &||\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{W}||_F^2 + \lambda_1||\boldsymbol{W} - \boldsymbol{U}\boldsymbol{C}||_F^2 + \lambda_2||\boldsymbol{Z}||_F^2 + \lambda_3||\boldsymbol{U}||_1 \\ &\text{s.t. } \boldsymbol{U} \geq 0,\ \boldsymbol{W} \geq 0 \end{aligned} \tag{3.1}$$

where $\boldsymbol{C}$ is an $M \times D$ binary matrix whose rows represent gene sets corresponding to annotations. If the dth gene is annotated as a member of the mth gene set, then $c_{md} = 1$. $\boldsymbol{U}$ is a non-negative $K \times M$ matrix that constrains the rows of $\boldsymbol{W}$ to be close to linear combinations of the rows of $\boldsymbol{C}$. Sparsity is imposed on $\boldsymbol{U}$ by the term $||\boldsymbol{U}||_1 = \sum_{k,m} |u_{km}|$.

Another example is OntoVAE [125], a variational autoencoder (VAE) (section 2.5.2.1) whose architecture is constrained by the incorporation of a directed acyclic graph (DAG). I worked with OntoVAE for my research project presented in chapter 5. In OntoVAE, the latent and observed variables in the bottleneck, decoder, and reconstruction layers, represent nodes in the incorporated DAG. Connections between latent variables in different layers, as well as between latent variables and observed variables in the reconstruction layer, are only possible if there is a corresponding edge in the DAG.

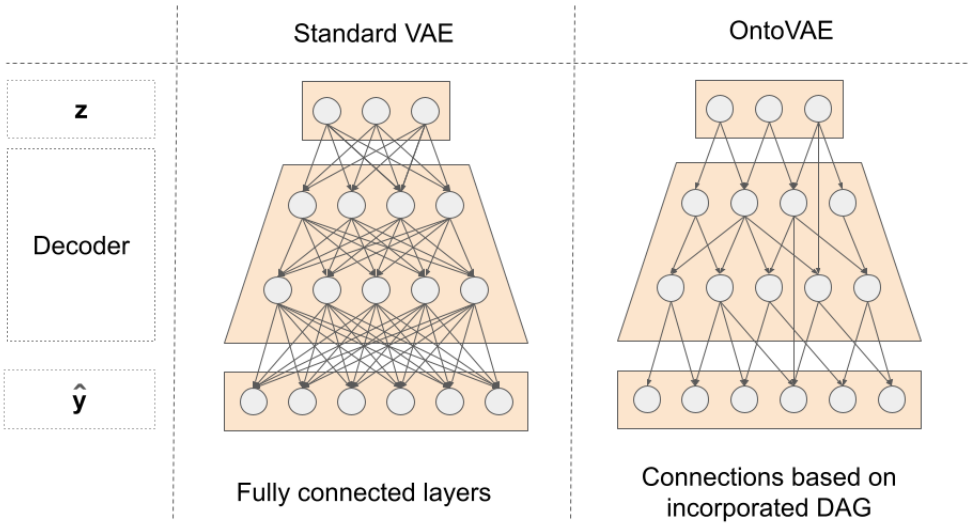


Figure 3.3 – **OntoVAE overview**. The latent and observed variables in the bottleneck (z), decoder, and reconstruction ($\hat{y}$) layers, represent nodes in the incorporated directed acyclic graph (DAG), and the connections between them depend on the edges in the DAG.

A GO sub-ontology DAG (section 3.2.1), for example, can be incorporated into OntoVAE. In a GO sub-ontology DAG, the nodes are GO terms, and the edges represent the relationships between the terms; more generic terms are followed by more specific terms, and eventually by genes as terminal nodes. In an OntoVAE model based on a GO DAG of biological process terms, the latent variables in the bottleneck and decoder layers represent different biological processes. When the model is trained on a matrix of gene expression data, the values of those latent variables represent the activity of corresponding biological processes underlying the observed gene expression profiles. For a given latent variable, the values may be compared across samples from different biological conditions, allowing inference as to whether the corresponding biological process differs across those conditions. With standard VAEs it can be difficult to interpret the relationships between observed and latent variables; therefore the interpretability offered by OntoVAE is an appealing characteristic.

3.3 Multi-omics matrix factorization

The analysis of multi-omics data (section 1.3.4) can provide improved understanding of some biological conditions of interest, compared to the analysis of a dataset containing only a single type of omics data. Multi-omics data consists of multiple types of omics data generated for the same samples; its recent availability means the development of multi-omics data analysis tools is an active area of research [77, 79, 80]. Multi-omics data analysis is faced with several challenges: uneven dimensionality due to different numbers of observed variables across different omics; the presence

of multiple data types, such as continuous, binary and counts data; diverse sources of technological noise; diverse ranges of variability; samples that are only partially matched, meaning some of the types of omics data have not been generated for them. A category of analysis tools that has proven effective at navigating these challenges is joint matrix factorization [81, 173–176].

I refer to the application of joint matrix factorization to multi-omics data as *multi-omics matrix factorization*. My main PhD research project (chapter 4) was concerned with the application of joint matrix factorization to multi-omics data that has been generated from a small number of samples. In particular, I was interested in whether a transfer learning approach to multi-omics matrix factorization (section 3.4) would lead to improved lower dimensional representations of such data. In order to introduce this topic, the following sections provide an overview of multi-omics matrix factorization.

3.3.1 The general joint matrix factorization model

Consider a dataset $\boldsymbol{Y}$, consisting of matrices $\boldsymbol{Y}^{(m)}$, $m = 1, ..., M$. Each $N \times D_m$ matrix $\boldsymbol{Y}^{(m)}$ contains data for the same N observations, but for a different set of D_m observed variables. When $\boldsymbol{Y}$ is a multi-omics dataset, each $\boldsymbol{Y}^{(m)}$ is a single omics data matrix, containing a different type of omics data for the same N samples. The type of data in a particular $\boldsymbol{Y}^{(m)}$ might include, but is not limited to: RNA-seq counts by gene; DNA methylation levels by CPG site; binary values indicating the presence or absence of a DNA variant at different genomic locations. Some samples may be only partially matched, meaning some of the types of omics data have not been generated for them. If, for partially matched sample n, there is no data for omics m, then row vector $\boldsymbol{y}^{(m)}_{n:}$ will contain missing values.

In general, joint matrix factorization extends matrix factorization (section 2.5.1) to factorize each $\boldsymbol{Y}^{(m)}$ into a lower dimensional representation

$$\boldsymbol{Y}^{(m)} \approx \boldsymbol{J}^{(m)} + \boldsymbol{S}^{(m)} \tag{3.2}$$

where $\boldsymbol{J}^{(m)} = \boldsymbol{Z}\boldsymbol{W}^{(m)}$ and $\boldsymbol{S}^{(m)} = \boldsymbol{Z}^{(m)}\boldsymbol{U}^{(m)}$. Element $y^{(m)}_{nd}$ from $\boldsymbol{Y}^{(m)}$ is assumed to be the result of the sum of joint signals, contained in element $j^{(m)}_{nd}$ from $\boldsymbol{J}^{(m)}$, and the sum of individual signals, contained in element $s^{(m)}_{nd}$ from $\boldsymbol{S}^{(m)}$.

The columns of $N \times K_W$ matrix $\boldsymbol{Z}$ are latent variables shared by all M matrices. The rows of $K_W \times D_m$ matrix $\boldsymbol{W}^{(m)}$ correspond to these same latent variables, the columns correspond to the observed variables from $\boldsymbol{Y}^{(m)}$. In contrast, the columns of $N \times K_U$ matrix $\boldsymbol{Z}^{(m)}$ are latent variables for $\boldsymbol{Y}^{(m)}$ only. The rows of $K_U \times D_m$ matrix $\boldsymbol{U}^{(m)}$ correspond to these same latent variables, the columns correspond to the observed variables from $\boldsymbol{Y}^{(m)}$. The latent variables in $\boldsymbol{Z}$ are considered to represent joint signals, while the latent variables in $\boldsymbol{Z}^{(m)}$ are considered to represent individual signals. The total number of latent variables is $K = K_W + K_U$.

When $\boldsymbol{Y}$ is a multi-omics dataset, joint signals might be the activity of biological processes associated with all M types of omics profiles, while individual signals are biological processes only associated with a specific omics data matrix.

The latent variables, inferred with multi-omics matrix factorization, can be used for: assigning samples to clusters [81, 176]; input to subsequent model fitting, such as for survival analysis [81, 173]; testing for associations between biological processes and biological conditions [81, 110].

There are numerous joint matrix factorization approaches available for multi-omics matrix factorization. As with the matrix factorization approaches discussed in section 2.5.1, joint matrix factorization approaches differ in their assumptions and constraints, including whether to infer both joint and individual signals, or joint signals only. These approaches also differ in how they require observed data to be pre-processed in order to overcome the challenges posed by different numbers of observed variables, data types, and ranges of variability. I categorize joint matrix factorization approaches based on their optimization objectives, leading to three categories: 1) approaches based on distance 2) approaches based on correlation and covariance; 3) approaches based on explicit distributional assumptions. Examples of approaches commonly used for multi-omics matrix factorization, from each category, are described in the following sections.

3.3.2 Approaches based on distance

Extensions of both PCA (section 2.5.1.1) and NMF (2.5.1.3) can be used for multi-omics matrix factorization.

3.3.2.1 Extending PCA

An example of an extension of PCA, for joint matrix factorization, is the joint and individual variation explained (JIVE) approach [177], implemented in the r.jive R package [178]. JIVE factorizes each $\boldsymbol{Y}^{(m)}$ as per equation 3.2, using iterative SVD. For each given $\boldsymbol{J}^{(m)}$, JIVE finds $\boldsymbol{S}^{(m)}$ that minimizes the squared Frobenius norm

$$||\boldsymbol{Y}^{(m)} - \boldsymbol{J}^{(m)} - \boldsymbol{S}^{(m)}||_F^2 = \sum_{n,d} \left(y_{nd}^{(m)} - j_{nd}^{(m)} - s_{nd}^{(m)} \right)^2 \tag{3.3}$$

Then for jointly given $\boldsymbol{S}^{(1)}, ..., \boldsymbol{S}^{(M)}$, JIVE jointly finds $\boldsymbol{J}^{(1)}, ..., \boldsymbol{J}^{(M)}$ that minimizes

$$\sum_m ||\boldsymbol{Y}^{(m)} - \boldsymbol{J}^{(m)} - \boldsymbol{S}^{(m)}||_F^2 \tag{3.4}$$

JIVE iterates between these steps until convergence. JIVE constrains the factorization by forcing the columns of each $\boldsymbol{J}^{(M)}$ to be orthogonal to the columns of each $\boldsymbol{S}^{(m)}$, assuring that the latent variables are uniquely determined. Prior to factorization, the authors of JIVE recommend performing variable-wise centering so that each column of each $\boldsymbol{Y}^{(m)}$ has a mean value of 0. To account for differences in the number and variability of observed variables, they also recommended scaling each $\boldsymbol{Y}^{(m)}$ by its total variation. JIVE can handle the presence of missing values in $\boldsymbol{Y}^{(m)}$ matrices, but not partially matched samples.

3.3.2.2 Extending NMF

An example of an extension of NMF, for joint matrix factorization, is IntNMF [179], implemented in the IntNMF R package. IntNMF only attempts to infer latent variables corresponding to joint signals, and so factorizes each $\boldsymbol{Y}^{(m)}$ as

$$\boldsymbol{Y}^{(m)} \approx \boldsymbol{Z}\boldsymbol{W}^{(m)} \tag{3.5}$$

The IntNMF algorithm begins with initialization of $\boldsymbol{Z}$ with non-negative values. Then, for given $\boldsymbol{Z}$, it uses non-negativity constrained alternating least squares to find, for each m, $\boldsymbol{W}^{(m)}$ that minimizes

$$\begin{aligned} &||\boldsymbol{Y}^{(m)} - \boldsymbol{Z}\boldsymbol{W}^{(m)}||_F^2 \\ &\text{s.t. } \boldsymbol{W}^{(m)} \geq 0 \end{aligned} \tag{3.6}$$

Then, for given $\boldsymbol{W}^{(1)}, ..., \boldsymbol{W}^{(M)}$, it uses non-negativity constrained alternating least squares to find $\boldsymbol{Z}$ that minimizes

$$\begin{aligned} &\sum_m \theta_m ||\boldsymbol{Y}^{(m)} - \boldsymbol{Z}\boldsymbol{W}^{(m)}||_F^2 \\ &\text{s.t. } \boldsymbol{Z} \geq 0 \end{aligned} \tag{3.7}$$

IntNMF iterates between these steps until convergence. θ_m is a user-defined weight controlling the relative contribution of $\boldsymbol{Y}^{(m)}$ to the factorization, recommended by the authors to be based on the relative size of $\sum_{n,d}\left(y_{nd}^{(m)}\right)^2$. Their recommended pre-processing step is to scale the values in each $\boldsymbol{Y}^{(m)}$ to be between 0 and 1. IntNMF cannot handle the presence of missing values in $\boldsymbol{Y}^{(m)}$ matrices.

3.3.3 Approaches based on correlation and covariance

There are several joint matrix factorization methods that infer latent variables, specific to each omics data matrix, based on correlation and/or covariance. The objective of these methods is to infer latent variables that both represent the variance within each $\boldsymbol{Y}^{(m)}$, as well as the relationships across $\boldsymbol{Y}^{(1)}, ..., \boldsymbol{Y}^{(M)}$. For example, regularized generalized canonical correlation analysis (RGCCA) [180], implemented in the RGCCA R package, factorizes each $\boldsymbol{Y}^{(m)}$ as

$$\boldsymbol{Y}^{(m)} \approx \boldsymbol{Z}^{(m)}\boldsymbol{U}^{(m)} \tag{3.8}$$

RGCCA begins by inferring values for a single latent variable for each $\boldsymbol{Y}^{(m)}$ by maximizing

$$\begin{aligned} &\sum_{l,m=1;l\neq m}^{M} c_{lm} g\left(\text{Cov}\left(\boldsymbol{Y}^{(l)}\boldsymbol{a}_1^{(l)}, \boldsymbol{Y}^{(m)}\boldsymbol{a}_1^{(m)}\right)\right) \\ &\text{s.t. } (1-\tau_m)\text{Var}\left(\boldsymbol{Y}^{(m)}\boldsymbol{a}_1^{(m)}\right) + \tau_m ||\boldsymbol{a}_1^{(m)}||_2^2 = 1 \end{aligned} \tag{3.9}$$

Each c_{lm} is either 0 or 1, indicating whether the relationship between $\boldsymbol{Y}^{(l)}$ and $\boldsymbol{Y}^{(m)}$ should be taken into account. The function $g(\,)$ can be defined as either $g(x) = x$, $g(x) = |x|$, or $g(x) = x^2$. If $g(x) = x$, then negative correlations between latent variables are penalized, which is not the case for either $g(x) = |x|$, or $g(x) = x^2$ [181]. The parameters $\tau_1, ..., \tau_M$ can take values ranging from 0 to 1, controlling whether the optimization is based on correlation, covariance, or a mixture of both.

Each $\boldsymbol{a}_1^{(m)} \in \mathbb{R}^{D_m \times 1}$ is used to derive vector $\boldsymbol{x}_1^{(m)} = \boldsymbol{Y}^{(m)}\boldsymbol{a}_1^{(m)}$. This results in an initial factorization with $\boldsymbol{z}_{:1}^{(m)} = \boldsymbol{x}_1^{(m)}$ and $\boldsymbol{u}_{1:}^{(m)} = \left(\boldsymbol{x}_1^{(m)^T}\boldsymbol{x}_1^{(m)}\right)^{-1}\boldsymbol{x}_1^{(m)^T}\boldsymbol{Y}^{(m)}$, where $\boldsymbol{z}_{:1}^{(m)}$ and $\boldsymbol{u}_{1:}^{(m)}$ are the first column and row vectors, respectively, of $\boldsymbol{Z}^{(m)}$ and $\boldsymbol{U}^{(m)}$ from equation 3.8. Alternatively the factorization can be formulated as $\boldsymbol{u}_{1:}^{(m)} = \boldsymbol{a}_1^{(m)^T}$, with $\boldsymbol{z}_{:1}^{(m)} = \boldsymbol{Y}^{(m)}\boldsymbol{a}_1^{(m)}\left(\boldsymbol{a}_1^{(m)^T}\boldsymbol{a}_1^{(m)}\right)^{-1}$.

Subsequent latent variables are inferred with a deflation procedure [181]. The deflation procedure begins by deriving $\boldsymbol{Y}_1^{(m)} = \boldsymbol{Y}^{(m)} - \boldsymbol{z}_{:1}^{(m)}\boldsymbol{u}_{1:}^{(m)}$ for $m = 1, ..., M$. Then $\boldsymbol{a}_2^{(1)}, ..., \boldsymbol{a}_2^{(M)}$ are inferred for $\boldsymbol{Y}_1^{(1)}, ..., \boldsymbol{Y}_1^{(M)}$ based on the maximization described in equation 3.9. Vectors $\boldsymbol{a}_2^{(1)}, ..., \boldsymbol{a}_2^{(M)}$ are then used to derive $\boldsymbol{Y}_2^{(m)} = \boldsymbol{Y}^{(m)} - \boldsymbol{z}_{:2}^{(m)}\boldsymbol{u}_{2:}^{(m)}$ for $m = 1, ..., M$, for which $\boldsymbol{a}_3^{(1)}, ..., \boldsymbol{a}_3^{(M)}$ are subsequently inferred. The deflation procedure continues until the desired number of latent variables have been inferred.

Recommended pre-processing for RGCCA factorization [182] is variable-wise standardization, followed by dividing each $\boldsymbol{Y}^{(m)}$ by $\sqrt{D_m}$. RGGCA can handle the presence of missing values in $\boldsymbol{Y}^{(m)}$ matrices, as well as partially matched samples.

It is possible to impose sparsity on the factorization by using a variant of RGCCA called sparse generalized canonical correlation analysis (SGCCA). In SGCCA, which is also implemented in the RGCCA R package, equation 3.9 is replaced by

$$\begin{aligned} &\sum_{l,m=1;l\neq m}^{M} c_{lm} g\left(\text{Cov}\left(\boldsymbol{Y}^{(l)}\boldsymbol{a}_1^{(l)}, \boldsymbol{Y}^{(m)}\boldsymbol{a}_1^{(m)}\right)\right) \\ &\text{s.t. } ||\boldsymbol{a}_1^{(m)}||_2 = 1, \text{ and } ||\boldsymbol{a}_1^{(m)}||_1 \leq s_m \end{aligned} \tag{3.10}$$

where $s_1, ..., s_M$ are user defined parameters for controlling the sparsity.

3.3.4 Approaches based on distributional assumptions

Joint matrix factorization approaches based on explicit distributional assumptions are popular for the analysis of multi-omics data. Examples include iClusterBayes

[107] and multi-omics factor analysis (MOFA) [118, 183]. Both of these methods are Bayesian approaches (section 2.2) that allow the assumption of different sampling distributions to account for data type heterogeneity.

3.3.4.1 iClusterBayes

iClusterBayes [107], implemented in the iClusterPlus R package [184], is a Bayesian approach that assumes each $\boldsymbol{Y}^{(m)}$ can be represented as

$$\boldsymbol{Y}^{(m)} \approx \mathbf{1}_N^T \boldsymbol{a}^{(m)} + \boldsymbol{Z}\boldsymbol{W}^{(m)} \tag{3.11}$$

where $\mathbf{1}_N = (1,...,1) \in \mathbb{R}^{1\times N}$, and $\boldsymbol{a}^{(m)} \in \mathbb{R}^{1\times D_m}$ is a vector of intercepts whose dth element, $a_d^{(m)}$, is the intercept for the dth observed variable from $\boldsymbol{Y}^{(m)}$.

iClusterBayes factorization is based on assumed sampling and prior distributions (section 2.2). Each observed value, $y_{nd}^{(m)}$, is modeled with a sampling distribution conditional on $\boldsymbol{z}_{n:}$ and $\boldsymbol{w}_{:d}^{(m)} = \gamma_d^{(m)} \hat{\boldsymbol{w}}_d^{(m)}$, where $\boldsymbol{z}_{n:}$ is the nth row vector of $\boldsymbol{Z}$, and $\boldsymbol{w}_{:d}^{(m)}$ is the dth column vector of $\boldsymbol{W}^{(m)}$. The value of binary variable $\gamma_d^{(m)}$ indicates whether or not all the values in $\hat{\boldsymbol{w}}_d^{(m)} \in \mathbb{R}^{K\times 1}$ are associated with $y_{nd}^{(m)}$.

For continuous data, the assumed sampling distribution is

$$p\left(y_{nd}^{(m)} | a_d^{(m)}, \boldsymbol{z}_{n:}, \boldsymbol{w}_{:d}^{(m)}\right) = \text{Normal}\left(a_d^{(m)} + \boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}, \sigma_{dm}^2\right) \tag{3.12}$$

If $y_{nd}^{(m)}$ is binary, the assumed sampling distribution is

$$p\left(y_{nd}^{(m)} | a_d^{(m)}, \boldsymbol{z}_{n:}, \boldsymbol{w}_{:d}^{(m)}\right) = \text{Bernoulli}\left(\left(1 + \exp\left(-a_d^{(m)} - \boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right)\right)^{-1}\right) \tag{3.13}$$

If $y_{nd}^{(m)}$ is a count, the assumed sampling distribution is

$$p\left(y_{nd}^{(m)} | a_d^{(m)}, \boldsymbol{z}_{n:}, \boldsymbol{w}_{:d}^{(m)}\right) = \text{Poisson}\left(\exp\left(a_d^{(m)} + \boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right)\right) \tag{3.14}$$

iClusterBayes assumes the prior distributions

$$\begin{aligned}
p(z_{nk}) &= \text{Normal}(0,1) \\
p\left(a_d^{(m)}\right) &= \text{Normal}\left(0, \sigma_w^2\right) \\
p\left(\hat{w}_{kd}^{(m)}\right) &= \text{Normal}\left(0, \sigma_w^2\right) \\
p\left(\sigma_{dm}^2\right) &= \text{Inv-Gamma}(1,1) \\
p\left(\gamma_d^{(m)}\right) &= \text{Bernoulli}\left(q_m\right)
\end{aligned}$$

The user is required to select values for σ_w^2 and $q_1,...,q_M$

iClusterBayes uses Markov chain Monte Carlo (MCMC) (section 2.2.2) to sample from the posterior distributions of these latent variables, employing Gibbs sampling

for some, and the Metropolis-Hastings algorithm for others. The estimated posterior means are then used to populate the values in the lower dimensional matrices from equation 3.11. iClusterBayes returns a Bayesian Information Criterion value to help with selecting K.

iClusterBayes cannot handle the presence of missing values in $\boldsymbol{Y}^{(m)}$ matrices.

3.3.4.2 MOFA

An alternative to iClusterBayes is multi-omics factor analysis (MOFA) [118], implemented in the MOFA2 R package [183]. MOFA makes similar distributional assumptions to iClusterBayes, but estimates posterior distributions with variational inference (VI) (section 2.2.3), which is faster than MCMC.

MOFA assumes each $\boldsymbol{Y}^{(m)}$ can be represented as

$$\boldsymbol{Y}^{(m)} \approx \boldsymbol{Z}\boldsymbol{W}^{(m)} \tag{3.15}$$

and models observed value $y_{nd}^{(m)}$ with a sampling distribution conditional on $\boldsymbol{z}_{n:}$ and $\boldsymbol{w}_{:d}^{(m)}$, where $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} s_{kd}^{(m)}$. Similar to iClusterBayes, $s_{kd}^{(m)}$ is a binary variable indicating whether or not the kth element of $\hat{\boldsymbol{w}}_{d}^{(m)} \in \mathbb{R}^{K\times 1}$ is associated with $y_{nd}^{(m)}$.

MOFA also allows the assumption of different sampling distributions. For continuous data, the assumed sampling distribution is

$$p\left(y_{nd}^{(m)} | \boldsymbol{z}_{n:}, \boldsymbol{w}_{:d}^{(m)}\right) = \text{Normal}\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}, 1/\tau_{d}^{(m)}\right) \tag{3.16}$$

If $y_{nd}^{(m)}$ is binary, the assumed sampling distribution is

$$p\left(y_{nd}^{(m)} | \boldsymbol{z}_{n:}, \boldsymbol{w}_{:d}^{(m)}\right) = \text{Bernoulli}\left(\left(1+\exp\left(-\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right)\right)^{-1}\right) \tag{3.17}$$

If $y_{nd}^{(m)}$ is a count, the assumed sampling distribution is

$$p\left(y_{nd}^{(m)} | \boldsymbol{z}_{n:}, \boldsymbol{w}_{:d}^{(m)}\right) = \text{Poisson}\left(\log\left(1+\exp\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right)\right)\right) \tag{3.18}$$

MOFA assumes the prior distributions

$$\begin{aligned}
p\left(z_{nk}\right) &= \text{Normal}\left(0,1\right) \\
p\left(\hat{w}_{kd}^{(m)}\right) &= \text{Normal}\left(0, 1/\alpha_{k}^{(m)}\right) \\
p\left(\tau_{d}^{(m)}\right) &= \text{Gamma}\left(1e^{-14}, 1e^{-14}\right) \\
p\left(s_{kd}^{(m)}\right) &= \text{Bernoulli}\left(\theta_{k}^{(m)}\right)
\end{aligned}$$

MOFA also assumes $p\left(\theta_{k}^{(m)}\right) = \text{Beta}\left(1,1\right)$ and $p\left(\alpha_{k}^{(m)}\right) = \text{Gamma}\left(1e^{-14}, 1e^{-14}\right)$, meaning the user does not need to select values for hyperparameters.

Differences between MOFA and iClusterBayes: There are some differences between iClusterBayes and MOFA that potentially make MOFA more appealing as a tool for inferring meaningful latent variables. For example, the binary variable $s_{kd}^{(m)}$ used by MOFA is assumed to vary across k, m, and d. This ultimately has the effect that, for given m and d, some estimated values in the $\boldsymbol{w}_{:d}^{(m)}$ vector are shrunk towards zero (section 2.5.1.5). In contrast, the binary $\gamma_d^{(m)}$ variables used by iClusterBayes potentially force all estimated values in certain $\boldsymbol{w}_{:d}^{(m)}$ vectors to be shrunk towards zero. Another example is that $1/\alpha_k^{(m)}$, the variance that MOFA uses for $\hat{w}_{kd}^{(m)}$, is assumed to have different values across k and m. If, for a given k and m, $\alpha_k^{(m)}$ is inferred to have a very high value, then all the estimated values in vector $\boldsymbol{w}_{k:}^{(m)}$ will be shrunk towards zero. This potentially results in a factorization that represents both joint and individual signals. In contrast, iClusterBayes uses the same variance, σ_w^2, for all $\hat{w}_{kd}^{(m)}$ across different k, d, and m.

MOFA estimates posterior distributions with mean-field variational inference (section 2.2.3), which is faster than the iClusterBayes approach of MCMC sampling. MOFA uses the CAVI algorithm to estimate parameters for the factorized joint variational distribution

$$\prod_{n=1}^{N}\prod_{k=1}^{K} q(z_{nk}) \prod_{m=1}^{M}\prod_{k=1}^{K} q\left(\alpha_k^{(m)}\right) q\left(\theta_k^{(m)}\right) \prod_{m=1}^{M}\prod_{d=1}^{D_m} q\left(\tau_d^{(m)}\right) \prod_{m=1}^{M}\prod_{d=1}^{D_m}\prod_{k=1}^{K} q\left(\hat{w}_{kd}^{(m)}, s_{kd}^{(m)}\right) \tag{3.19}$$

The estimated posterior means are then used to populate the values in the lower dimensional matrices from equation 3.15.

For $\boldsymbol{Y}^{(m)}$ matrices whose values are binary or counts, MOFA transforms them while iterating through the CAVI algorithm. Transformed $y_{nd}^{(m)}$ values are then modeled as continuous data for posterior distribution inference. Prior to inference with the CAVI algorithm, MOFA performs variable-wise centering of $\boldsymbol{Y}^{(m)}$ matrices that contain continuous data. For $\boldsymbol{Y}^{(m)}$ matrices whose values are binary or counts, the centering is performed after the transformation to continuous data.

The authors of MOFA recommend filtering observed variables so that the dimensions of the $\boldsymbol{Y}^{(m)}$ matrices are within the same order of magnitude. MOFA can handle the presence of missing values in $\boldsymbol{Y}^{(m)}$ matrices, as well as partially matched samples. It is not necessary to select a value for K for the final factorization estimated by MOFA. Instead the user selects an initial value for K, and latent variables are sequentially dropped from the factorization based on a threshold for variance explained.

3.3.5 Multi-omics data with few samples

When a multi-omics dataset is generated from a small number of samples, joint matrix factorization may struggle to separate biological signals into a meaningful lower dimension representation [124, 185]. For example, the authors of MOFA suggest at least 15 samples are required for a meaningful representation. However, there are

cases where omics data are generated from only a small number of samples, due to the rareness of the biological conditions being studied, or cost constraints [124, 185].

An example, of where it is challenging to obtain enough samples for effective multi-omics matrix factorization, is the study of rare disease. In Europe, a disease is considered to be rare if it has prevalence of less than 50:100,000 [186]. It is estimated that, in Europe, around 30 million people suffer from a rare disease, of which there are over 6,000. So while few people suffer from any particular rare disease, making it difficult to obtain multi-omics data for that disease, overall there are many rare disease patients.

In chapter 6, I present two rare diseases: type 2 facioscapulohumeral muscular dystrophy (FSHD2), and Bosma arhinia and microphthalmia syndrome (BAMS). In the work presented in chapter 6, differential testing (section 2.4) was performed on different types of omics data. We would eventually like to analyze this data with multi-omics matrix factorization, but there are less than 15 samples with multi-omics data.

The objective of my main PhD project, presented in chapter 4, was to implement a transfer learning (section 3.4) approach to multi-omics matrix factorization, and evaluate whether such an approach improved the factorization of a multi-omics dataset generated from a small number of samples.

I chose to investigate a transfer learning approach based on the MOFA algorithm. (section 3.3.4.2) This choice was informed by results from benchmarking studies comparing the performance of algorithms for the analysis of multi-omics data. These studies assessed approaches based on using their lower dimensional representations for: assigning samples to clusters [81, 173–176]; performing survival analysis [81, 173]; testing for associations between latent variables, or sample clusters, and clinical annotations [81, 173]; testing for associations between latent variables and biological processes [81]; identifying observed variables that vary the most across clusters of samples [176].

No single approach, or category of approaches, was the best performer across these studies. However, a common conclusion was that imposing sparsity, and inferring latent variables associated with individual signals, are beneficial. MOFA does not force values in the estimated lower dimensional representation to be exactly equal to zero. However, the statistical model used by MOFA assumes sparse $\boldsymbol{w}_{k:}^{(m)}$ vectors, resulting in some estimated values being close to zero. In addition, by shrinking all the estimated values for certain $\boldsymbol{w}_{k:}^{(m)}$ vectors towards zero, MOFA is able to infer latent variables associated with individual signal. Another appealing characteristic of MOFA is that it does not require parameter tuning, making it a fast and easy approach to use. Finally, the MOFA algorithm can handle the existence of partially matched samples, which I believed would be a beneficial characteristic of a transfer learning approach developed for the factorization of multi-omics datasets generated from small numbers of samples.

3.4 Transfer learning

When a multi-omics dataset is generated from a small number of samples, joint matrix factorization may struggle to separate biological signals into a meaningful lower dimension representation (section 3.3.5). For a multi-omics dataset generated from a small number of samples, a transfer learning approach to joint matrix factorization potentially produces an improved lower dimensional representation.

In transfer learning, information extracted from fitting a model for a learning dataset is incorporated into the fitting of a model for a target dataset, from a different, yet related, domain [185]. Transfer learning can be used when fitting a supervised model, such as regression or classification, for the target dataset, or when modeling the target dataset with an unsupervised approach, such as unsupervised dimension reduction [187]. For example, Shin et al. [188] trained an image classification model on a dataset of more than 1.2 million categorized natural images, from around 1,000 categories. They then *fine-tuned* this pre-trained model on smaller datasets of medical images. The results from fine-tuning compared favorably to training the same model on the medical images without fine-tuning. Fine-tuning [189] is a transfer learning approach, in which the parameters (or a subset of them) of a pre-trained model are updated by retraining the model on a target dataset. When training the model on the target dataset, the parameters are initialized using the values from the pre-trained model. Then, during fine-tuning, the model is trained on the target dataset, updating the model parameters that are not held fixed.

Transfer learning is considered to be valuable when the target dataset contains an insufficient number of observations for effective model fitting [190], yet the learning dataset contains a large number of observations characterized by a heterogeneous set of conditions [124]. If the two domains share an overlapping latent space, then information relating to this shared latent space can be inferred from the learning dataset [191, 192]. When this information is incorporated into fitting a model for the target dataset, it may yield a better model than would otherwise be obtained [185, 193].

The effectiveness of transfer learning has been demonstrated for various tasks [193], including text classification [191], image classification [188, 194], indoor WiFi localization [191], signal classification [195], and in building recommender systems [196].

3.4.1 Transfer learning and omics data

In the context of omics data, the learning and target datasets might, for example, contain the same type of omics data, but for samples from different organisms, tissue types, or sets of diseases. In this context, it is expected that the learning and target dataset domains will share an overlapping latent space associated with shared biological processes [192, 197, 198]. If the learning dataset was generated from a large number of samples, characterized by a heterogeneous set of biological conditions, it should be possible to infer information about these shared biological processes from

it.

3.4.2 Transfer learning and matrix factorization

The usefulness of transfer learning approaches to matrix factorization has been demonstrated in contexts where both the target and learning datasets were single matrices of transcriptomic data. Consider a target dataset, $\boldsymbol{T}$, that is an $N_T \times D$ matrix containing omics data for N_T samples and D observed variables. The objective is to use matrix factorization (section 2.5.1) to infer a lower dimensional representation of $\boldsymbol{T}$.

If N_T is small, matrix factorization may struggle to separate underlying biological signals, and instead combine them into a limited number of latent variables [124, 185]. To mitigate this problem, a transfer learning approach to matrix factorization incorporates information inferred from the prior factorization of a learning dataset. The learning dataset, $\boldsymbol{L}$, is an $N_L \times D$ matrix containing omics data for a different set of $N_L > N_T$ samples [199], and the same D observed variables. A prior factorization of $\boldsymbol{L}$ gives the lower dimensional representation

$$\boldsymbol{L} \approx \boldsymbol{Z}_L \boldsymbol{W} \tag{3.20}$$

It is assumed that if the samples in $\boldsymbol{L}$ are characterized by a heterogeneous set of biological conditions, then the row vectors of $\boldsymbol{W}$ will represent well separated biological processes, and that some of these are associated with observed variation in $\boldsymbol{T}$ [124, 192, 197]. In a transfer learning approach to matrix factorization, $\boldsymbol{T}$ can be factorized as

$$\boldsymbol{T} \approx \boldsymbol{Z}\boldsymbol{W} \tag{3.21}$$

where $\boldsymbol{W}$ comes from the factorization of $\boldsymbol{L}$ (equation 3.20). $\boldsymbol{Z}$ is inferred for the samples from $\boldsymbol{T}$, with respect to $\boldsymbol{W}$.

In one study [192], the authors used a matrix of mouse single-cell RNA-Seq data as $\boldsymbol{L}$, and factorized it with non-negative matrix factorization (NMF). Then, for $\boldsymbol{T}$ consisting of a matrix of human time course bulk RNA-Seq data, they used the tool projectR [198] to infer $\boldsymbol{Z}$. The resulting latent variables were associated with known spatiotemporal differences across the samples. For $\boldsymbol{t}_{n:}$, the row vector in $\boldsymbol{T}$ for the nth sample, projectR infers $\boldsymbol{z}_{n:}$, the vector of latent variable values for that sample, as the least squares solution to

$$\boldsymbol{t}_{n:}^T \approx \boldsymbol{W}^T \boldsymbol{z}_{n:}^T \tag{3.22}$$

Likewise, Davis-Marcisak et al. [200] factorized a matrix of mouse single-cell RNA-Seq data with NMF, and then used projectR to infer $\boldsymbol{Z}$ for bulk RNA-Seq data from human cancer samples. They observed an association between a particular latent variable, inferred with projectR, and outcomes in metastatic melanoma. Taroni et al. developed a transfer learning framework called MultiPLIER [124]. For $\boldsymbol{L}$ they used a subset of Recount2 [201], a compendium that contains bulk RNA-seq data for more than 70,000 human samples from more than 2,000 studies. They factorized $\boldsymbol{L}$ with the

NMF algorithm PLIER (section 3.2.3). In MultiPLIER, $\boldsymbol{Z}$ is inferred for $\boldsymbol{T}$ as

$$\boldsymbol{Z}^T = \left(\boldsymbol{W}\boldsymbol{W}^T + \lambda_2 \boldsymbol{I}\right)^{-1} \boldsymbol{W}\boldsymbol{T}^T \tag{3.23}$$

where λ_2 is from equation 3.1 and $\boldsymbol{I}$ is the $K \times K$ identity matrix. In one application of MultiPLIER, they used a blood cell compendium of microarray gene expression data for $\boldsymbol{T}$. In another, they used microarray gene expression data for 79 samples from a rare disease group called antineutrophil cytoplasmic autoantibody associated vasculitis. In both cases, the MultiPLIER approach inferred biologically relevant latent variables.

3.4.3 Transfer learning based on MOFA

The application of matrix factorization to a large, heterogeneous learning dataset can yield latent variables that are biologically relevant for a target dataset characterized by a different set of biological conditions. When I began my main PhD project, existing transfer learning approaches to matrix factorization had been designed for, and demonstrated on, datasets comprised of a single matrix of omics data only. This meant that transfer learning approaches to multi-omics matrix factorization were lacking. Additionally, for a target dataset containing omics data for a small number of samples, there was a need for more quantitative evaluation of whether transfer learning produces an improved lower dimensional representation, when compared to matrix factorization without transfer learning.

To address these needs, my main PhD project had two objectives: 1) design and implement a transfer learning approach for multi-omics matrix factorization; 2) design and implement an evaluation protocol to assess whether multi-omics matrix factorization is improved with a transfer learning approach. I designed a transfer learning approach for multi-omics matrix factorization based on the MOFA algorithm (section 3.3.4.2). A description of this transfer learning approach, called MOTL [202], is in chapter 4, along with results from the implementation of the evaluation protocol.

4 MOTL: Multi-omics matrix factorization with transfer learning

Contents

4.1 Introduction 72
4.2 Motivation and objectives 72
4.3 Article 73
4.4 Discussion 104

4.1 Introduction

In this chapter, I present the results from the work I carried out for my main PhD research project. This project was part of a larger, interdisciplinary project, called CAMUDI (https://anr.fr/Project-ANR-21-CE45-0001). The CAMUDI project, involving researchers from multiple research laboratories, was launched to develop analysis methods that can accommodate multi-omics data generated from a small number of samples. The analysis methods investigated for the CAMUDI project include matrix factorization, network exploration, and autoencoders. My focus was the design, and implementation, of a transfer learning approach for multi-omics matrix factorization

In section 4.2, I summarize the background, already described in more detail in preceding chapters, that motivated the objectives for my research project. Then, in section 4.3, I present the article that resulted from this project.

4.2 Motivation and objectives

Omics data analysis (section 2) can provide insight into a biological system, by revealing associations between biological molecules and biological conditions of interest. Prior information, contained in annotation resources (section 3.2), can be used to extend this insight, allowing researchers to infer associations between biological processes and conditions.

Including multiple types of omics data (section 1.3.4) in an analysis can further improve understanding of a biological system, compared to that obtained from the

analysis of a single type of omics data. Multi-omics matrix factorization (section 3.3), the application of joint matrix factorization to multi-omics data, has proven to be an effective approach for the analysis of multiple types of omics data. Multi-omics matrix factorization yields insight into associations between biological processes and conditions, by inferring a lower dimensional representation that consists of latent variables representing underlying biological signals.

There are cases where multi-omics data are generated from only a small number of samples, hindering the ability of joint matrix factorization to infer a meaningful representation. For omics data generated from a small number of samples, transfer learning (section 3.4) has been proposed as a potential solution to the limited effectiveness of matrix factorization. The usefulness of transfer learning approaches to matrix factorization has been demonstrated for single matrices of omics data (section 3.4.2).

When I began my main PhD project, it appeared that 1) transfer learning approaches to multi-omics matrix factorization were lacking; 2) there was a need for more quantitative evaluation of whether transfer learning enhances matrix factorization. To address the first need, I developed MOTL (Multi-Omics Transfer Learning), a transfer learning algorithm for multi-omics matrix factorization. MOTL is based on the variational inference (section 2.2.3) methodology used by the multi-omics matrix factorization approach MOFA (section 3.3.4.2). MOTL factorizes a multi-omics target dataset by incorporating values inferred by a prior MOFA factorization of a multi-omics learning dataset. To address the second need, I designed two protocols to evaluate the performance of MOTL, against multi-omics matrix factorization without transfer learning

4.3 Article

This section contains the scientific article that resulted from the development and evaluation of MOTL. In this article, I firstly describe the statistical framework and implementation of MOTL. I then present results from applying the two protocols that I designed to evaluate the performance of MOTL. Finally, I present the application of MOTL to a multi-omics dataset comprised of a small number of samples.

In the protocols I designed to evaluate MOTL, I used both simulated (first protocol) and real (second protocol) multi-omics data, and assessed performance based on the inference of latent variables associated with differences in biological conditions. The real multi-omics data I used was from The Cancer Genome Atlas (TCGA) project (section 1.3.4). Additionally, after evaluating MOTL with the two protocols, I applied MOTL to another multi-omics dataset, comprised of a small number of glioblastoma samples, as well as healthy controls. I also applied MOFA to this same multi-omics dataset, and compared the inferred latent variables. From these analyses I observed that, for a target multi-omics dataset comprised of a small number of samples, MOTL was more effective in uncovering informative latent variables than joint matrix factorization without transfer learning.

This article is currently under review with a journal. A preprint [202] is available on bioRxiv (https://doi.org/10.1101/2024.03.22.586210), where I uploaded it prior to journal submission, in order to make it available to the research community and obtain feedback. An open source R implementation of MOTL, with accompanying documentation, as well as code and instructions for reproducing the analyses in the article, are available at https://github.com/david-hirst/MOTL.

Before presenting the MOTL article, I wish to bring the reader's attention to some differences in terminology from the article, compared to terminology used throughout this thesis:

1. I have used *observed variable* in this thesis, whereas I used *feature* in the article
2. I have used *latent variable* in this thesis, whereas I used *factor* in the article
3. I have used the abbreviation *SNV*, standing for *single nucleotide variant*, in this thesis (section 1.3.1.2). In the article, the abbreviation *SNV* stands for *simple nucleotide variation*, which includes *single nucleotide variants* and *small indels*.

MOTL: enhancing multi-omics matrix factorization with transfer learning

David Hirst[1,*], Morgane Térézol[1], Laura Cantini[2], Paul Villoutreix[1], Matthieu Vignes[3], Anaïs Baudot[1, 4, 5,*]

1 Aix Marseille Univ, INSERM, MMG, Marseille, France
2 Institut Pasteur, Université Paris Cité, CNRS UMR 3738, Machine Learning for Integrative Genomics Group, F-75015 Paris, France
3 School of Mathematical and Computational Sciences, College of Science, Massey University, Palmerston North, New Zealand
4 CNRS, Marseille, France
5 Barcelona Supercomputing Center, Spain

* Correspondence:
David Hirst, david.hirst@univ-amu.fr; Anaïs Baudot, anais.baudot@univ-amu.fr

Abstract

Joint matrix factorization is a popular method for extracting lower dimensional representations of multi-omics data. It disentangles underlying mixtures of biological signals, facilitating efficient sample clustering, disease subtyping, or biomarker identification, for instance. However, when a multi-omics dataset is generated from only a limited number of samples, the effectiveness of matrix factorization is reduced. Addressing this limitation, we introduce MOTL (Multi-Omics Transfer Learning), a novel framework for multi-omics matrix factorization with transfer learning based on MOFA (Multi-Omics Factor Analysis). MOTL infers latent factors for a small multi-omics dataset, with respect to those inferred from a large heterogeneous learning dataset. We designed two protocols to evaluate transfer learning approaches, based on simulated and real multi-omics data. Using these protocols, we observed that MOTL improves the factorization of multi-omics datasets, comprised of a limited number of samples, when compared to factorization without transfer learning. We showcase the usefulness of MOTL on a glioblastoma dataset comprised of a small number of samples, revealing an enhanced delineation of cancer status and subtype thanks to transfer learning.

1 Introduction

Omics data have transformed the study of biology and medicine by enabling high-throughput measurements of the activity and abundance of biological molecules and processes (Conesa and Beck, 2019; Dermitzakis, 2008; Manzoni et al., 2018) In recent years, the fields of biology and medicine have been revolutionized by the increased availability of multi-omics datasets (Conesa and Beck, 2019; Subramanian et al., 2020; Huang et al., 2017). A multi-omics dataset is comprised of multiple data matrices, each containing a different type of

omics data (e.g., mRNA transcript counts, genomic mutations, DNA methylation prevalence). The integrative analysis of multi-omics data can provide a better understanding of a biological system than that obtained from the analysis of a single omics data matrix (Rohart et al., 2017; Huang et al., 2017; Rappoport and Shamir, 2018; Subramanian et al., 2020; Chauvel et al., 2020; Pierre-Jean et al., 2020; Cantini et al., 2021). Indeed, different omics contain complementary information that contribute to a more comprehensive overview of the underlying biological system. Additionally, using multiple omics can reveal insights into relationships between the different biological layers they represent. Combining omics is also expected to reduce the impact of noise. However, multi-omics data poses further analysis challenges beyond those encountered in single omics data analysis. These challenges include increased dimensionality, the presence of multiple data types, diverse sources of technological noise, and diverse ranges of variability. In this context, there has been an increased need for methods able to carry out integrative analysis of multiple omics.

The development of multi-omics analysis tools is an active area of research. A category of multi-omics analysis tools that has proven effective is matrix factorization (Rappoport and Shamir, 2018; Tini et al., 2019; Chauvel et al., 2020; Pierre-Jean et al., 2020; Cantini et al., 2021). Matrix factorization infers a lower dimensional representation of the observed data, in which a sufficiently informative proportion of the original signal is retained (Stein-O'Brien et al., 2018). Most classical matrix factorization approaches were designed for the analysis of a single data matrix. Applying matrix factorization to a single omics matrix produces a score matrix and a weight matrix, both of which contain values for latent factors that are potentially associated with different sources of underlying biological signal. The values in the weight matrix ideally represent signal across the assayed biological features, and the values in the score matrix represent the signal across the samples. For a multi-omics dataset, one of the strategies is to jointly factorize multiple omics data matrices. Various methods are now available for this purpose (Cantini et al., 2021). Multi-omics joint matrix factorization methods typically produce a weight matrix for each omics, and either a shared score matrix or a combination of shared and omics specific score matrices. Many multi-omics matrix factorization methods are extensions of classical methods designed for single omics. For example, intNMF (Chalise and Fridley, 2017) extends non-negative matrix factorization to the multi-omics setting, and allows the user to determine the relative contribution of each omics to the extraction of joint signal. JIVE (Lock et al., 2013) extends principal component analysis to model both joint and omics specific signal. MOFA (Argelaguet et al., 2018), which is an extension of Factor Analysis, uses a Bayesian framework to account for the presence of multiple data types and to distinguish between joint and omics specific signal. Overall, the factors inferred by multi-omics matrix factorization can be used for clustering samples to reveal disease sub-types, for identifying molecular profiles and biomarkers associated with diseases, as well as for prediction of outcomes such as drug response and survival (Rappoport and Shamir, 2018; Taroni et al., 2019; Pierre-Jean et al., 2020; Cantini et al., 2021; Banerjee et al., 2023). A challenge for matrix factorization is that it requires a large amount of observed data to produce a meaningful representation. However, there are cases where omics are measured from only a small number of samples, due to the rareness or cost of obtaining the data, and so there is a need for methods which help mitigate this challenge (Weiss et al., 2016; Stein-O'Brien et al., 2018; Banerjee et al., 2023).

For a dataset generated from a small number of samples, transfer learning is a potential

solution to the limited effectiveness of matrix factorization. Transfer learning is a machine learning approach in which information extracted from a large learning domain is used to improve the performance of a task applied to a smaller target domain (Weiss et al., 2016; Stein-O'Brien et al., 2019; Taroni et al., 2019; Banerjee et al., 2023). It is assumed that the two domains share an overlapping latent space, allowing knowledge from the learning domain to be transferred to the application of the task to the target domain. Transfer learning has been successfully used in various machine learning applications, including image classification, text sentiment classification and recommendation systems (Weiss et al., 2016; Veeramachaneni et al., 2019; Dong et al., 2021; Banerjee et al., 2023). In a transfer learning approach to omics matrix factorization, information inferred from the prior factorization of a learning dataset, comprised of a large number of samples from a heterogeneous set of biological conditions, is incorporated into the factorization of a small target dataset (Peng et al., 2021; Banerjee et al., 2023). It is assumed that if the latent factors inferred from the learning dataset represent common underlying biological processes, they should help improve the factorization of the target dataset (Stein-O'Brien et al., 2019).

The usefulness of transfer learning approaches to matrix factorization, for omics data analysis, has been demonstrated in contexts in which both the target and learning datasets were comprised of single matrices of omics data. In these cases, transfer learning was used to infer a score matrix for the target dataset by projecting it onto a weight matrix inferred from a learning dataset. In one study, Stein-O'Brien et al. factorized a mouse single cell RNA-seq learning dataset with the Bayesian non-negative matrix factorization algorithm CoGAPS (Fertig et al., 2010). Then, they used the transfer learning tool projectR (Sharma et al., 2020) to infer a score matrix for a human time course bulk RNA-seq dataset. The resulting factors were associated with known spatiotemporal differences across the samples. In another example, Davis-Marcisak et al. factorized a mouse single cell RNA-seq learning dataset with CoGAPS, and then used projectR to infer a score matrix for bulk RNA-seq data from human cancer samples. They observed an association between a particular projectR factor and outcomes in metastatic melanoma. Taroni et al. developed MultiPLIER, a transfer learning framework, which they demonstrated by firstly applying the non-negative matrix factorization algorithm PLIER (Mao et al., 2019) to a subset of Recount2 to infer a weight matrix. Recount2 is a compendium of RNA-seq data obtained from 70,000+ human samples taken across more than 2,000 studies (Collado-Torres et al., 2017). Taroni et al. then used a blood cell compendium of microarray gene expression data as the target dataset, for which they inferred a score matrix with MultiPLIER, as well as factorizing the target dataset directly with PLIER. For counts of a cell type of interest, MultiPLIER inferred a more highly correlated factor than was inferred by direct factorization of the target dataset. They also used microarray gene expression data for 79 samples from a rare disease group called antineutrophil cytoplasmic autoantibody associated vasculitis (AAV) as a target dataset. There are no AAV samples in the Recount2 compendium, yet the MultiPLIER factors were positively correlated with their best match from factors inferred by direct factorization of the target dataset.

It has thus been demonstrated that the application of matrix factorization to a large, heterogeneous learning dataset can yield factors containing transferable information, that are biologically relevant to target datasets from different organisms, diseases, cell types and omics platforms. However, existing transfer learning approaches to matrix factorization have been designed for, and demonstrated on, datasets comprised of single omics

data only. To the best of our knowledge, transfer learning approaches to joint multi-omics matrix factorization are currently lacking.

We here introduce MOTL (Multi-Omics Transfer Learning), a novel Bayesian transfer learning algorithm for multi-omics matrix factorization. MOTL is based on MOFA, a popular tool for integrative multi-omics analysis (Argelaguet et al., 2018). We first present the statistical framework and implementation of MOTL. Next, we propose two protocols, that we designed based on simulated and real multi-omics datasets, for evaluating the performance of transfer learning approaches. We used these protocols to evaluate MOTL, and observed that, for a target multi-omics dataset comprised of a small number of samples, our transfer learning approach to matrix factorization is more effective than matrix factorization without transfer learning. Lastly, we showcase a practical use case of MOTL on a limited glioblastoma sample set, revealing an enhanced delineation of cancer status and subtype thanks to transfer learning.

2 Results

2.1 MOTL: A new transfer learning framework for multi-omics matrix factorization

We propose MOTL, a transfer learning approach to multi-omics matrix factorization. MOTL is based on MOFA (Argelaguet et al., 2018), which uses variational Bayesian inference (Blei et al., 2017). Consider a multi-omics target dataset, $\boldsymbol{T}$, consisting of omics matrices, $\boldsymbol{T}^{(m)}$, $m = 1, ..., M$. Each $\boldsymbol{T}^{(m)} = \left[t_{nd}^{(m)}\right] \in \mathbb{R}^{N_t \times D_m}$ contains data for N_t samples (rows) and D_m features (columns), where $t_{nd}^{(m)}$ is the value for the nth sample and the dth feature from the mth matrix. The features depend on which molecules were assayed to generate a given omics matrix; for example the features for mRNA counts are genes, while those for DNA methylation are CpG sites.

We wish to jointly factorize $\boldsymbol{T}^{(m)}$ into a matrix of sample scores, $\boldsymbol{Z} = [z_{nk}] \in \mathbb{R}^{N_t \times K}$, and an omics specific matrix of feature weights, $\boldsymbol{W}^{(m)} = \left[w_{kd}^{(m)}\right] \in \mathbb{R}^{K \times D_m}$. The resulting lower dimensional representation is based on K factors, which ideally represent underlying biological signals associated with some biological condition(s) of interest. z_{nk} is the score for the nth sample and the kth factor, while $w_{kd}^{(m)}$ is the weight for the kth factor and the dth feature from the mth matrix. The kth column vector of $\boldsymbol{Z}$, denoted by $\boldsymbol{z}_{:k}$, contains scores for factor k, while the nth row vector, $\boldsymbol{z}_{n:}$, contains scores for sample n. The kth row vector of $\boldsymbol{W}^{(m)}$, denoted by $\boldsymbol{w}_{k:}^{(m)}$, contains weights for factor k, for the mth matrix, while the dth column vector, $\boldsymbol{w}_{:d}^{(m)}$, contains weights for feature d from the mth matrix.

We are concerned with the situation in which N_t is small, exacerbating the curse of dimensionality, and therefore we expect to improve the factorization of $\boldsymbol{T}$ by employing a transfer learning approach (see Figure 1). We do this transfer learning by incorporating values that have already been inferred from the prior factorization of a learning dataset, $\boldsymbol{L}$, and we assume that the Bayesian matrix factorization algorithm MOFA (Argelaguet et al., 2018) was used for factorizing $\boldsymbol{L}$.

The learning dataset consists of omics matrices $\boldsymbol{L}^{(m)}$, $m = 1, ..., M$. Each $\boldsymbol{L}^{(m)} = \left[l_{nd}^{(m)}\right] \in \mathbb{R}^{N_l \times D_m}$ contains data for the same D_m features as $\boldsymbol{T}^{(m)}$, but for a different set of $N_l > N_t$

samples. We hypothesise that if $\boldsymbol{L}$ is comprised of samples from a heterogeneous set of biological conditions, then the factorization of $\boldsymbol{L}$ will yield information that is relevant for the factorization of $\boldsymbol{T}$.

MOTL is based on the variational Bayesian inference methodology used by MOFA (Methods 4.2). We have modified the MOFA algorithm to enable us to supplement the factorization of $\boldsymbol{T}$ by incorporating values already inferred from the prior factorization of $\boldsymbol{L}$. For MOTL, we assume that each observed $t_{nd}^{(m)}$ is a random variable, with a likelihood that is conditional on vectors $\boldsymbol{z}_{n:}$ and $\boldsymbol{w}_{:d}^{(m)}$. We model continuous, counts and binary data with the same likelihoods and link functions that MOFA uses. For observed continuous data we thus assume a Gaussian likelihood, into which we include a feature-wise precision parameter, $\tau_d^{(m)}$, for each feature d from matrix m. For observed binary data we assume a Bernoulli likelihood, and for observed counts data we assume a Poisson likelihood. In contrast to MOFA, MOTL doesn't center the input data during factorization fitting, as we want to incorporate an intercept that is compatible with the factorization of $\boldsymbol{L}$. We therefore replace $\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}$ with $a_d^{(m)} + \boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}$ in the likelihood, where $a_d^{(m)}$ is the feature-wise intercept for feature d, from matrix m. We infer $a_d^{(m)}$ values based on the MOFA factorization of $\boldsymbol{L}$ (Methods 4.7). MOTL accepts missing $t_{nd}^{(m)}$ values; therefore it is not necessary to remove features with missing values, or perform imputation, before using MOTL.

In order to carry out a transfer learning approach to matrix factorization, MOTL uses the matrix of feature weights, $\boldsymbol{W}^{(m)}$, vector of feature-wise intercepts, $\boldsymbol{a}^{(m)} = \left[a_d^{(m)}\right] \in \mathbb{R}^{D_m}$, and vector of feature-wise precision parameter values, $\boldsymbol{\tau}^{(m)} = \left[\tau_d^{(m)}\right] \in \mathbb{R}^{D_m}$, inferred for each $\boldsymbol{L}^{(m)}$ with a prior MOFA factorization of $\boldsymbol{L}$. Instead of modelling these as random variables, we treat them as constants. We aim to obtain point estimates of z_{nk} values, for which we assume the same joint prior distribution as MOFA does,

$$p(\boldsymbol{Z}) = \prod_{n=1}^{N_t}\prod_{k=1}^{K} \text{Normal}(z_{nk}|0,1) \tag{1}$$

MOTL obtains point estimates of z_{nk} values by approximating the joint posterior distribution $p(\boldsymbol{Z}|\boldsymbol{T})$ with a variational distribution:

$$q(\boldsymbol{Z}) = \prod_{n=1}^{N_t}\prod_{k=1}^{K} q(z_{nk}) = \prod_{n=1}^{N_t}\prod_{k=1}^{K} \text{Normal}(z_{nk}|\mu_{nk},\sigma_{nk}) \tag{2}$$

MOTL infers $q(\boldsymbol{Z})$ iteratively. At each iteration, the value of each parameter is updated while all other parameter values are held fixed. MOTL optimizes the joint variational distribution by iterating until convergence. For each z_{nk}, the expected value, $\mathbb{E}_q[z_{nk}] = \mu_{nk}$, is used as the point estimate throughout and after model fitting. MOTL uses the same update equations for the parameters of $q(z_{nk})$ as MOFA, but with the inclusion of intercepts:

$$\sigma_{nk}^2 = \left(\sum_{m=1}^{M}\sum_{d=1}^{D_m}\tau_{nd}^{(m)}\left(w_{kd}^{(m)}\right)^2 + 1\right)^{-1} \tag{3}$$

$$\mu_{nk} = \sigma_{nk}^2\sum_{m=1}^{M}\sum_{d=1}^{D_m}\tau_{nd}^{(m)}w_{kd}^{(m)}\left(\hat{t}_{nd}^{(m)} - a_d^{(m)} - \sum_{j\neq k} z_{nj}w_{jd}^{(m)}\right) \tag{4}$$

where $\tau_{nd}^{(m)}$ is the precision for the nth sample and dth feature from the mth matrix, and $\hat{t}_{nd}^{(m)}$ denotes a (possibly) transformed observed data point (Methods 4.2). For observed data with a Gaussian assumed likelihood, a feature-wise precision, $\tau_d^{(m)}$, is used instead of $\tau_{nd}^{(m)}$, and there is no transformation, meaning $\hat{t}_{nd}^{(m)} = t_{nd}^{(m)}$. For observed data with a non-Gaussian assumed likelihood, MOTL transforms the data to yield Gaussian pseudo-data values, which it does not center. The transformation to Gaussian pseudo-data allows updates of $q(\boldsymbol{Z})$ to be based on the assumption of Gaussian observed data. When MOFA transforms observed data with a Bernoulli assumed likelihood, it derives and uses a precision parameter, $\tau_{nd}^{(m)}$, for each sample and feature. For observed data with a Poisson assumed likelihood, it derives and uses a feature-wise precision, $\tau_d^{(m)}$. Thus for Bernoulli observed data, MOTL initializes $\tau_{nd}^{(m)}$ values with $\tau_d^{(m)}$ values, which are averages of the $\tau_{nd}^{(m)}$ values returned by the factorization of $\boldsymbol{L}$, and these are subsequently updated at each iteration of the algorithm. For Poisson observed data, MOTL uses the $\tau_d^{(m)}$ values obtained from the prior factorization of $\boldsymbol{L}$, and holds them fixed.

To monitor convergence we calculate the evidence lower bound (ELBO), which can be used to evaluate how well a variational distribution approximates a posterior distribution of interest. We calculate the ELBO with respect to $\boldsymbol{Z}$:

$$\text{ELBO}(\boldsymbol{Z}) = \mathbb{E}_q\left[\log\, p(\boldsymbol{T}|\boldsymbol{Z})\right] + \mathbb{E}_q\left[\log\, p(\boldsymbol{Z})\right] - \mathbb{E}_q\left[\log\, q(\boldsymbol{Z})\right] \tag{5}$$

For $\boldsymbol{T}^{(m)}$ with a non-Gaussian assumed likelihood, we use the same lower bound for $\log\, p\left(t_{nd}^{(m)}|\boldsymbol{Z}\right)$ as MOFA does. Maximizing this lower bound, coupled with the use of $\hat{t}_{nd}^{(m)}$ values, allows updates of $q(\boldsymbol{Z})$ based on the assumption of Gaussian observed data (Jaakkola and Jordan, 2000; Seeger and Bouchard, 2012). We calculate the ELBO at regular intervals, and the number of iterations between each calculation is a user defined parameter. We check for convergence based on the absolute change in the ELBO (from the previous check) as a percentage of the initial ELBO. The algorithm is deemed to have converged when a specified number of changes in the ELBO are consecutively below a threshold. Both the threshold, and the required number of consecutive changes falling below this threshold, are user defined parameters.

We allow factors to be dropped during training, based on the fraction of variance explained:

$$R_{mk}^2 = 1 - \frac{\sum_{n=1}^{N_t}\sum_{d=1}^{D_m}\left(\hat{t}_{nd}^{(m)} - a_d^{(m)} - z_{nk}w_{kd}^{(m)}\right)^2}{\sum_{n=1}^{N_t}\sum_{d=1}^{D_m}\left(\hat{t}_{nd}^{(m)} - a_d^{(m)}\right)^2} \tag{6}$$

We drop the factor with the lowest R_{mk}^2 that does not have any R_{mk}^2 above the threshold. We assess factors in this way after each round of updates. After convergence the algorithm returns $\boldsymbol{Z}$ and $\boldsymbol{W}^{(m)}$ matrices for the factors that have not been dropped.

MOTL is available as an open source R implementation (Methods 4.9)

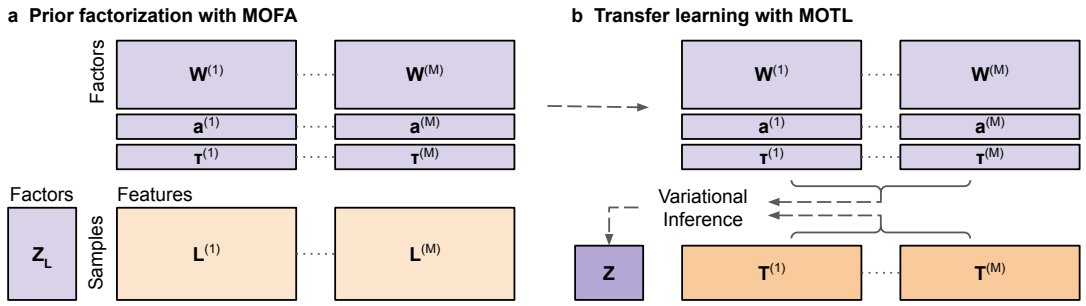


Figure 1: Overview of MOTL, our transfer learning approach to joint multi-omics matrix factorization based on variational Bayesian inference. **a** A multi-omics learning dataset, $\boldsymbol{L}$, consisting of M omics matrices, $\boldsymbol{L}^{(m)}$, $m = 1, ..., M$, is factorized with MOFA to infer a matrix of feature weights, $\boldsymbol{W}^{(m)}$, vector of feature-wise intercepts, $\boldsymbol{a}^{(m)}$, and a vector of feature-wise precision parameter values, $\boldsymbol{\tau}^{(m)}$, for each $\boldsymbol{L}^{(m)}$. **b** The feature weight, intercept, and precision parameter values, inferred from the factorization of $\boldsymbol{L}$, are incorporated into the factorization of a multi-omics target dataset, $\boldsymbol{T}$, for which MOTL infers a matrix of sample scores, $\boldsymbol{Z}$, with variational inference.

2.2 Evaluation protocol using simulated multi-omics data

We first designed and implemented a transfer learning evaluation protocol based on simulated multi-omics datasets, which we generated from groundtruth factors (Methods 4.3). In brief, in each simulation instance, we generated a multi-omics dataset $\boldsymbol{Y}$, which we subsequently split into a target dataset, $\boldsymbol{T}$, and a learning dataset, $\boldsymbol{L}$. $\boldsymbol{Y}$ consisted of matrices of counts, continuous, and binary data. We generated each matrix, $\boldsymbol{Y}^{(m)}$, from a statistical distribution conditional on random matrices $\boldsymbol{Z}$ and $\boldsymbol{W}^{(m)}$, which each contained values for K groundtruth factors. The kth column vector of $\boldsymbol{Z}$ contained sample scores for the kth groundtruth factor. The kth row vector of $\boldsymbol{W}^{(m)}$ contained feature weights for that same factor. We varied the number of groundtruth factors across configurations, using $K \in \{20, 30\}$. We generated $\boldsymbol{Z}$ based on the group membership of samples. In each instance, we created two groups of five samples for the target dataset. The learning dataset samples belonged to either 20 or 40 differently sized groups of randomly selected sizes. For each groundtruth factor and group, the sample scores were generated using a mean parameter value that was common to all samples in the group. We induced heterogeneity by allowing the means to vary across groups and factors, randomly selecting each group mean, for a given groundtruth factor, from a pool of three possible values. We split each $\boldsymbol{Y}^{(m)}$ into $\boldsymbol{T}^{(m)}$ and $\boldsymbol{L}^{(m)}$, based on the sample groups used to generate $\boldsymbol{Z}$. In each instance $\boldsymbol{T}$ contained data for 10 samples, while the expected number of samples for $\boldsymbol{L}$ was $\in \{400, 1000\}$.

For each simulation instance, we factorized $\boldsymbol{L}$ with MOFA (Methods 4.6). We then factorized $\boldsymbol{T}$ with our transfer learning method MOTL (Methods 4.7), incorporating output from the factorization of $\boldsymbol{L}$. To benchmark the performance of MOTL, we also performed direct MOFA factorizations (i.e., factorization without transfer learning) of $\boldsymbol{T}$ datasets. We evaluated both the MOTL and direct MOFA factorization of each $\boldsymbol{T}$, and compared the overall performance of each approach. We evaluated factorizations of each $\boldsymbol{T}$ by calculating an F1 score (Methods 4.8), to measure how well the factorization allowed us to uncover differentially active groundtruth factors underlying $\boldsymbol{T}$. The kth groundtruth factor was differentially active for $\boldsymbol{T}$ if the mean parameter values used to simulate the sample scores, for that factor, differed between the two groups of target dataset sam-

ples. Factorization with MOTL led to higher F1 scores than direct MOFA factorization, indicating that the MOTL factorizations were more effective in uncovering differentially active latent signal from $\boldsymbol{T}$ datasets (Figure 2). This was observed across all simulation configurations, and the overall uplift in mean F1 score for MOTL, when compared to direct MOFA factorization, was 0.21 (p-value < 0.01, Methods 4.8). We thus observed that transfer learning with MOTL was more effective in uncovering differentially active latent signal, when compared to direct MOFA factorization (without transfer learning) of $\boldsymbol{T}$.

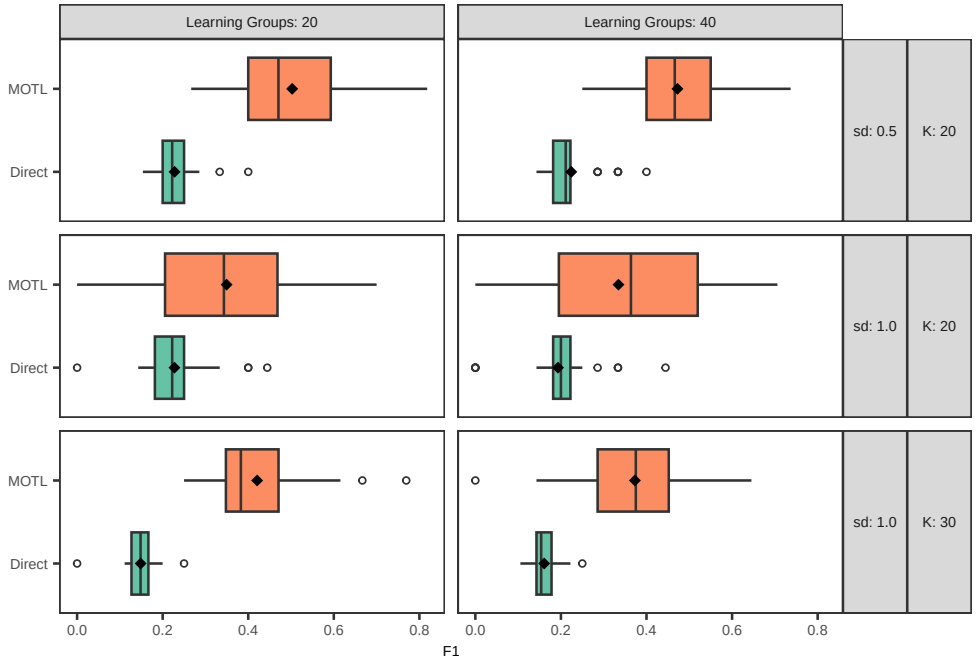


Figure 2: Evaluation of factorizations of small simulated multi-omics target datasets with and without MOTL transfer learning. The boxplots represent the F1 scores obtained for different factorization approaches and simulation configuration settings. Simulation configurations varied in the number of groups of samples used for the learning dataset (*Learning Groups*), the number of groundtruth factors (K), or the standard deviation used to simulate z_{nk} values (sd). F1 scores take a value between 0 and 1, and higher values indicate better factorizations. Each boxplot is based on 30 F1 scores. The hinges of the boxes are the 25th and 75th percentiles, the middle lines are medians, the diamonds are the mean values, and the whiskers are either extreme values or extend 1.5 times the inter-quartile range from the hinge.

2.3 Evaluation protocol using TCGA multi-omics data

We next designed, and implemented, a second transfer learning evaluation protocol, based on TCGA multi-omics data (Methods 4.4). We used four types of omics data: log2 transformed mRNA counts, log2 transformed miRNA counts, DNA methylation M-values, and simple nucleotide variation (SNV) binary data, which we obtained for 32 different cancer types. We created target datasets using data from three cancer types; acute myeloid leukemia (LAML), pancreatic adenocarcinoma (PAAD) and skin cutaneous melanoma (SKCM). We created these target datasets by firstly creating four reference datasets. Each reference dataset, $\boldsymbol{R}$, contained multi-omics data for all samples from either two, or all three of the cancer types. We then randomly split every $\boldsymbol{R}$ into non-overlapping target

datasets which each contained only five samples per cancer type (Figure 3**a**). We merged data from the remaining 29 cancer types into a learning dataset, $\boldsymbol{L}$, which contained multi-omics data for 7,217 samples.

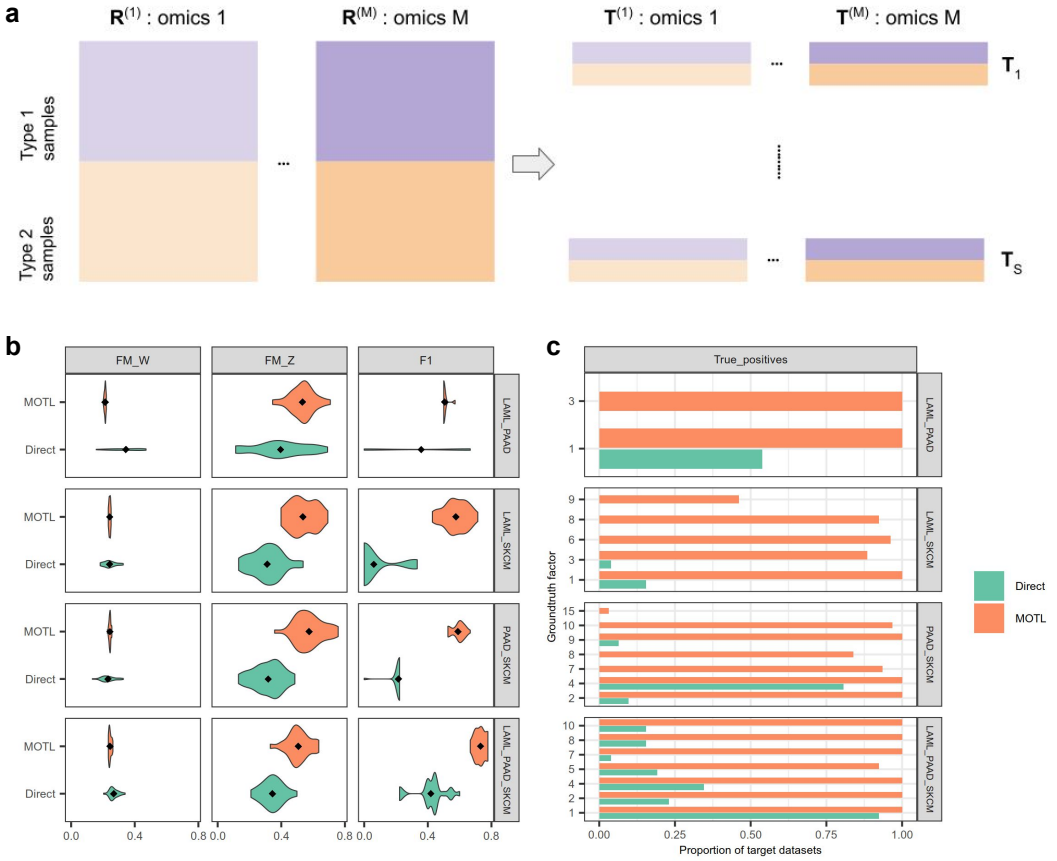


Figure 3: Evaluations using TCGA multi-omics data. **a** TCGA target datasets are created from two or three cancer types: We created a reference dataset, $\boldsymbol{R}$, containing multi-omics data for all samples from the selected cancer types. We then randomly split $\boldsymbol{R}$ into non-overlapping target, $\boldsymbol{T}$, datasets, containing multi-omics data for five samples per cancer type. We did this for subsets of the set of cancer types {LAML, PAAD, SKCM}; in total we created, and split, four reference datasets, each of which contained multi-omics data for all samples from either two (LAML and PAAD, LAML and SKCM, PAAD and SKCM), or three (LAML, PAAD and SKCM) cancer types. **b** Comparison of factorization approaches applied to TCGA multi-omics datasets. Violin plots of F-measure values for weight matrix factors (FM_W), F-measure values for score matrix factors (FM_Z), and F1 scores ($F1$). For each evaluation score, higher values indicate better factorizations. Scores are plotted by factorization method and by the cancer types characterizing the target dataset samples. **c** Frequency with which differentially active groundtruth TCGA factors were true positives. A differentially active groundtruth factor was a true positive if it was predicted as being differentially active based on a factorization of a target dataset. Each bar represents the proportion of target datasets, for which the factorization led to the differentially active groundtruth factor being a true positive. Proportions are plotted by factorization method, and by the cancer types characterizing the reference and target dataset samples.

We factorized $\boldsymbol{L}$ with MOFA (Methods 4.6), based on which we used MOTL to factorize each $\boldsymbol{T}$ (Methods 4.7). To benchmark the performance of MOTL, we also performed direct MOFA factorizations (without transfer learning) of $\boldsymbol{T}$ datasets. In order to evaluate the factorizations of $\boldsymbol{T}$ datasets, we factorized $\boldsymbol{R}$ datasets with MOFA and treated the

resulting score, $\boldsymbol{Z}$, and weight, $\boldsymbol{W}^{(m)}$, matrices as groundtruth factor matrices (Methods 4.8). We were interested in how well the factorizations of $\boldsymbol{T}$ datasets uncovered the groundtruth, and we used F-measure values and F1 scores to evaluate this.

We calculated F-measure values to assess the correlation between factors inferred from each $\boldsymbol{T}$, and the groundtruth factors obtained from the factorization of the corresponding $\boldsymbol{R}$ dataset (Methods 4.8). We calculated F-measure values for weight matrices (FM_W), as well as for score matrices (FM_Z) . The overall mean FM_W for MOTL was slightly lower (0.03 reduction, p-value < 0.01) than for direct MOFA factorizations of $\boldsymbol{T}$ datasets (Figure 3**b**, column 1), which is the result of lower average relevance counterbalancing higher average recovery. We concluded from this that groundtruth $\boldsymbol{W}^{(m)}$ factors were more easily uncovered with those transferred from $\boldsymbol{L}$ than with direct MOFA factorization. However, despite factor trimming during MOTL factorization, some remaining transferred factors were less associated with groundtruth factors than those obtained with direct MOFA factorization. It is of note that the difference in average FM_W is attributable to the datasets containing LAML and PAAD samples only. If we exclude these, there is no difference in FM_W (p-value 0.59). The overall mean FM_Z for MOTL was 0.20 higher (p-value < 0.01, Methods 4.8) than for direct MOFA factorizations (Figure 3**b**, column 2). We thus observed that the $\boldsymbol{Z}$ factors obtained with MOTL, from $\boldsymbol{T}$ datasets, were more correlated with groundtruth factors, overall, than those obtained with direct MOFA factorization.

We also calculated F1 scores to measure how well the factorizations of $\boldsymbol{T}$ datasets uncovered differentially active groundtruth factors (Methods 4.8). For each $\boldsymbol{T}$ dataset, the groundtruth factors were the factors obtained from factorization of the corresponding $\boldsymbol{R}$ dataset. We considered the kth groundtruth factor to be differentially active if the distribution of scores in the kth column vector, of groundtruth $\boldsymbol{Z}$, differed between the cancer types. We can simultaneously evaluate the $\boldsymbol{Z}$ and $\boldsymbol{W}^{(m)}$ factors, and assess the overall quality of factorizations, by checking for an uplift in F1 scores (Figure 3**b**, column 3). MOTL (0.34 uplift, p-value < 0.01) yielded higher F1 scores than direct MOFA factorization, meaning it was more effective in uncovering latent activity that varied across cancer types.

We next examined differentially active groundtruth factors, with an initial focus on the frequency with which these factors were true positives (Figure 3**c**). For each factorization of a $\boldsymbol{T}$ dataset, a differentially active groundtruth factor was a true positive if it was predicted as being differentially active based on the factorization of $\boldsymbol{T}$. The unique count of true positives was a component of each F1 score value (Methods 4.8). We further performed a gene set enrichment analysis to identify the pathways and processes associated with differentially active groundtruth factors that were true positives (Methods 4.8, Supplementary Table 1), and that explained at least 1% of the mRNA variance in $\boldsymbol{R}$ (Supplementary Table 2).

The factorization of the $\boldsymbol{R}$ dataset containing all LAML and PAAD samples yielded six groundtruth factors, of which two were differentially active; Factor 1 and Factor 3. Both of these factors were true positives for 100% of MOTL factorizations of $\boldsymbol{T}$ datasets containing subsets of five LAML and PAAD samples (Figure 3**c**). In contrast, only one of these factors was a true positive for direct MOFA factorizations, and for just over half of the same $\boldsymbol{T}$ datasets (Figure 3**c**). Factor 1 is significantly associated with developmental processes, cell communication and immunity signaling. Factor 3 displays similar enrichments, with

an additional specific enrichment related to the regulation of gene expression in beta cells (Supplementary Table 1).

The factorization of the $\boldsymbol{R}$ dataset containing all LAML and SKCM samples yielded 12 groundtruth factors, of which five were differentially active. Four out these five factors were true positives for MOTL factorizations for more than 80% of the $\boldsymbol{T}$ datasets; one factor was a true positive for just under half of the $\boldsymbol{T}$ datasets. Importantly, only two of these five groundtruth factors were true positives for direct MOFA factorizations, and only for a small proportion of the $\boldsymbol{T}$ datasets. Four out of these five differentially active groundtruth factors, that were true positives, explained at least 1% of the mRNA variance in $\boldsymbol{R}$: Factor 1, Factor 3, Factor 8, and Factor 9. We performed gene set enrichment analysis on these four factors. Factor 1 is significantly associated with extracellular organisation, developmental processes, cell communication signaling, and Fc Receptor mediated immune processes (Supplementary Table 1). Factor 3 is significantly associated with hematopoeitic cell lineage, Pi3K/AKT and G protein-coupled signaling, and chemokine, interleukin, interferon signaling. Both Factors 1 and 3 are associated with keratinisation and formation of the cornified envelope. Factors 8 and 9 do not present significant pathway enrichments beyond keratinisation and processes already associated with the first two factors.

The factorization of the $\boldsymbol{R}$ dataset containing all PAAD and SKCM samples yielded 17 groundtruth factors, of which eight were differentially active. Seven of these eight factors were true positives for MOTL factorizations, six with high frequency (i.e., identified for more than 80% of the $\boldsymbol{T}$ target datasets). Only one differentially active groundtruth factor is frequently a true positive for direct MOFA factorization of the $\boldsymbol{T}$ target datasets. Factor 2 is related to B cell receptor signaling and Fc Receptor mediated immune processes, drug metabolism by cytochrome p450 and other metabolism-related processes. This Factor 2 is rarely uncovered by direct MOFA factorization. Contrarily, Factor 4 is a true positive for MOTL for 100% of the target datasets and for direct MOFA factorizations for more than 75% of the target datasets. This factor is associated with developmental processes and cytokine-cytokine receptor interactions. Factor 7 is associated with keratinisation and formation of the cornified envelope, Factor 9 with cell adhesion and migration, and Factor 10 with cytokine and chemokine signaling.

The factorization of the $\boldsymbol{R}$ dataset containing all LAML, PAAD and SKCM samples yielded 13 groundtruth factors, of which seven were differentially active. All seven of these differentially active groundtruth factors were true positives for MOTL factorization with high frequency, compared to one factor for direct MOFA factorization. These groundtruth factors, differentially active between all three cancer types, are associated with the same cellular processes and pathways identified when comparing the cancer types pairwise (Supplementary Table 1).

Overall, the factors that were differentially active when comparing two or the three cancer types reflect the different embryonic origins of the cancerous tissues, and highlight the importance of immunity and microenviroment in cancer pathophysiology and response to treatments. In conclusion, matrix factorization with transfer learning using MOTL better uncovers differentially active groundtruth factors from target datasets containing only a small number of samples.

2.4 Application of MOTL to glioblastoma

Glioblastoma is a rare, heterogeneous, and aggressive cancer type. Multi-omics datasets offer an important opportunity to better characterize glioblastoma subtypes, identify biomarkers, and propose novel therapeutic options (Santamarina-Ojeda et al., 2023). However, large collections of glioblastoma tissue samples are difficult to obtain due to the relative scarcity of the disease and the challenges involved in acquiring samples via invasive biopsies.

In Santamarina-Ojeda et al. (2023), the authors conducted a multi-omics profiling (mRNA expression, DNA methylation) for four normal brain samples and nine patient-derived glioblastoma stem cell (pd-GBSC) cultures. The nine cancer samples had been previously classified into three subtypes thanks to transcriptome-based signatures: classical (CL), proneural (PN), and mesenchymal (MS). Given the small number of samples, the authors devised a strategy based on analyzing this dataset in parallel with datasets gathered from the literature. We illustrate here how MOTL could help in analysing such a dataset comprised of a limited number of samples.

We first applied a direct MOFA factorization (Methods 4.6) to a target dataset comprised of the four normal and nine pd-GBSC samples (Methods 4.5), revealing eight factors. Heatmap clustering of the samples, based on these factors, does not demonstrate clear grouping with respect to either cancer status or subtype (Figure 4a). Next, we applied MOTL to the same target dataset (Methods 4.7). In this case, we first created a TCGA learning dataset containing mRNA expression, miRNA expression, DNA methylation, and SNV data for samples from all 32 cancer types (Methods 4.4). It is noteworthy that this learning dataset did not contain data for glioblastoma, as there were no TCGA glioblastoma samples fulfilling our selection criteria (i.e., with complete 4-layer multi-omics profiles). We factorized this learning dataset with MOFA (Methods 4.6), based on which we applied MOTL to the target dataset. MOTL transfer learning factorization revealed 25 factors. In this case, the heatmap clustering of the MOTL factors separates cancer and normal samples and also displays subgroups partially matching subtypes previously defined based on transcriptome signatures (Figure 4b).

Further statistical tests revealed that only one of the eight factors identified by direct MOFA factorization was differentially active between normal and cancer samples, whereas 19 of the 25 factors obtained by MOTL transfer learning factorization were differentially active (Methods 4.8). Gene set enrichment analysis, focusing on differentially active factors that explained at least 1% of mRNA variance, revealed 715 processes and pathways associated with the direct MOFA factor, and 1061 processes and pathways associated with the 12 MOTL factors. The overlap between the two sets of associated processes and pathways was 318, which is statistically significant (hypergeometric test, one-sided p-value < 0.01, Supplementary Table 3).

We also applied both direct MOFA, and MOTL factorizations, to target datasets comprised of normal samples and samples from just a single cancer subtype. In all cases, the direct MOFA factorization yielded only one differentially active factor, whereas MOTL yielded 11 (CL vs normal), 16 (MS vs normal), and 12 (PN vs normal) differentially active factors. Focusing on the subset of differentially active factors that explained at least 1% of mRNA variance, we performed gene set enrichment analyses. We identified processes and pathways that were associated with only a single cancer subtype, such as fatty acid metabolism enrichment for the MS subtype and clathrin-mediated endocytosis for the PN

subtype (Supplementary Table 4).

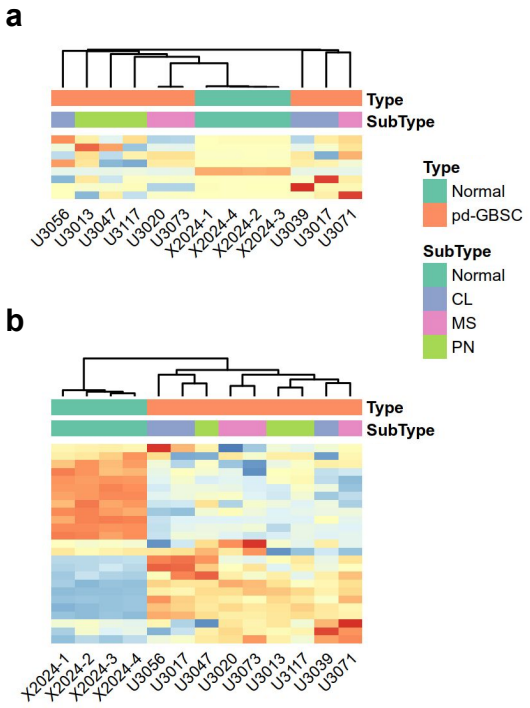


Figure 4: Heatmaps of factorizations of glioblastoma data: Each heatmap is based on the score matrix, $\boldsymbol{Z}$, inferred with a factorization of the target dataset comprised of multi-omics data (mRNA expression, DNA methylation) for all normal and patient-derived glioblastoma stem cell (pd-GBSC) samples. The multi-omics data was obtained from Santamarina-Ojeda et al., who also provided subtypes for the cancer samples, previously defined from transcriptomic signatures: CN (classical), PN (proneural), MS (mesenchymal). The rows of each heatmap are the factors, the columns are the samples, and the cells are the row-wise centered and scaled factor values. The rows and columns have been order with hierarchical clustering (complete-linkage). **a** Heatmap of $\boldsymbol{Z}$ matrix inferred with Direct MOFA factorization of the target dataset. **b** Heatmap of $\boldsymbol{Z}$ matrix inferred with MOTL factorization of the target dataset.

3 Discussion

We presented MOTL, an approach for multi-omics matrix factorization with transfer learning, which infers latent factor values for a multi-omics target dataset comprised of a small number of samples. MOTL factorizes the target dataset by incorporating latent factor values already inferred from the factorization of a learning dataset. We designed two protocols, based on simulated and real multi-omics datasets, for evaluating the performance of multi-omics matrix factorization with transfer learning. We implemented these protocols to evaluate MOTL, and observed that MOTL was more effective in uncovering differentially active groundtruth latent factors than direct matrix factorization without transfer learning. In addition, in the evaluation protocol based on TCGA data, the factors identified by MOTL are associated with biological processes and pathways consistent with the disease and tissue types represented by the target datasets. Finally, the application of MOTL to a glioblastoma dataset, comprised of a small number of samples, revealed

an enhanced delineation of cancer status and subtype thanks to transfer learning. We thus demonstrated, in the case of a multi-omics dataset comprised of a small number of samples, that MOTL can enhance the discovery of biological processes and pathways associated with a biological condition of interest.

In the work presented here, we are concerned with the situation in which we wish to analyse a target dataset comprised of a limited number of samples. In our evaluations, the target datasets never contained data for more than 15 samples, as our primary concern was with target datasets considered too small for useful factor analysis. It would be insightful to extend the evaluations by using a larger range of sizes for target datasets, in order to identify a crossover point at which transfer learning no longer enhances matrix factorization. In addition, with target datasets containing few samples, the learning dataset is likely to represent a set of biological conditions that does not fully overlap with the set of biological conditions represented by the target dataset. It would be relevant to investigate how similar the learning and target datasets need to be in order for shared factors to exist. For instance, it would be beneficial to use a measure of similarity, between a given learning and target dataset, which would predict the effectiveness of using a transfer learning approach to apply matrix factorization to the target dataset. It could also be interesting to quantify how heterogeneous (i.e., representing a large diversity of tissues, diseases, experimental conditions ...) a learning dataset needs to be, in order to yield factors which can be relevant for a given target dataset.

To evaluate MOTL, we designed two evaluation protocols, based on simulated and real data. Importantly, these protocols can be reused to evaluate other transfer learning strategies for multi-omics data integration. For the evaluation protocol based on simulated datasets, we created groups of samples, and distinguished between groups by assigning different mean parameters when simulating sample scores. We intended to simulate the situation in which the factors are biological processes whose activity varies across different biological conditions. We used the same set of feature weights for all groups of samples, meaning they shared the same underlying latent factors, albeit with varying levels of activity. A future extension to this would be to make some factors completely inactive for the learning dataset. We could then evaluate how well MOTL is able to uncover groundtruth latent structure, for the target dataset, when not all latent factors are shared by both datasets. For the evaluation protocol based on real data, we used TCGA, as it is a public repository of multi-omics data with a large number of samples, representing various cancer and tissue types. We selected three different cancer types as references from which to build target datasets, and did not include samples from these cancer types in the learning dataset. In this setting, it was unknown, prior to evaluation, whether there were latent factors common to the learning and target datasets. Yet MOTL was effective in uncovering differentially active latent factors, demonstrating that latent factors can be shared across different cancer types. We envisage MOTL as being a helpful tool in the study of rare diseases in general. Therefore a future extension of our work would be to evaluate the application of MOTL to a target dataset with non-cancer rare disease samples, using factors inferred from the TCGA learning dataset.

With MOTL we have designed a transfer learning framework that is compatible with a prior learning dataset factorization carried out with the MOFA Bayesian approach (Argelaguet et al., 2018). The appeal of a Bayesian framework is the flexibility with regard to the incorporation of prior information, and variational inference serves as a fast alternative to sampling methods. In the future, MOTL could be extended to allow

information to be incorporated at other levels of the assumed hierarchy of latent variables. For example, instead of fixing the feature weight values, they could be treated as random variables by MOTL, with priors informed by the factorization of the learning dataset.

In addition to MOFA, there are numerous methods available for multi-omics matrix factorization (Chalise and Fridley, 2017; Lock et al., 2013; Rohart et al., 2017; Mo et al., 2018), meaning a future extension of our work could be a transfer learning framework matched to a different multi-omics matrix factorization method. Similarly, we foresee value in extending an existing transfer learning method that has been designed for single omics data (Sharma et al., 2020; Taroni et al., 2019), so it can be applied in the multi-omics context. The evaluation of such a method, based on the factorization of a learning dataset with a variety of matrix factorization methods, would be informative.

Finally, a limitation with MOTL is that we are restricted to a factorization based on features that were retained for factorization of the learning dataset. A consequence is that some features which are highly variable in the target dataset may not contribute to the MOTL factorization. Therefore a future extension could be to add flexibility into the MOTL workflow, so that all features that are highly variable in the target dataset contribute to the factorization, even if they were not retained for the factorization of the learning dataset.

4 Methods

4.1 Mathematical Notation

- We denote matrices and datasets with bold capital letters: $\boldsymbol{Y}$
- If $\boldsymbol{Y}$ denotes a matrix, we introduce it as $\boldsymbol{Y} = [y_{nd}] \in \mathbb{R}^{N \times D}$ for which:
  - there are N rows and D columns
  - y_{nd} denotes the value in the nth row and the dth column
  - $\boldsymbol{y}_{n:}$ denotes the nth row vector, and $\boldsymbol{y}_{:d}$ denotes the dth column vector
- If $\boldsymbol{Y}$ denotes a dataset comprised of multiple matrices, we specify this, and denote each of the matrices as $\boldsymbol{Y}^{(m)} = \left[y_{nd}^{(m)}\right] \in \mathbb{R}^{N \times D_m}$
- We denote parameters for statistical distributions as non-bold, lower case letters. If the parameter is for a random variable stored in a matrix, we add indices. For example, $\tau_{nd}^{(m)}$ is a parameter for a random variable in the nth row and dth column of the mth matrix of some dataset. If $\tau_{d}^{(m)}$ is a parameter for the same matrix, then it is used for all values in the dth column.

4.2 The MOFA model

Consider a multi-omics dataset $\boldsymbol{Y}$ consisting of omics matrices $\boldsymbol{Y}^{(m)}$, $m = 1, ..., M$. Each $\boldsymbol{Y}^{(m)} = \left[y_{nd}^{(m)}\right] \in \mathbb{R}^{N \times D_m}$ contains data for N samples (rows) and D_m features (columns), where $y_{nd}^{(m)}$ is the value for the nth sample and the dth feature from the mth matrix. MOFA (Argelaguet et al., 2020), assumes the existence of latent factors, and jointly factorizes each $\boldsymbol{Y}^{(m)}$ into a shared matrix of sample scores $\boldsymbol{Z} = [z_{nk}] \in \mathbb{R}^{N \times K}$, and an omics specific

matrix of feature weights $\boldsymbol{W}^{(m)} = \left[w_{kd}^{(m)}\right] \in \mathbb{R}^{K \times D_m}$. The kth column of $\boldsymbol{Z}$ contains scores for the kth factor, and the kth row of $\boldsymbol{W}^{(m)}$ contains corresponding weights for that factor.

MOFA assumes that each observed $y_{nd}^{(m)}$ is a random variable, characterised by a probability distribution conditional on a set of latent random variables $\boldsymbol{\beta}$. It is assumed that the joint likelihood $p(\boldsymbol{Y}|\boldsymbol{\beta})$ is equal to $\prod_{m=1}^{M}\prod_{n=1}^{N}\prod_{d=1}^{D_m} p(y_{nd}^{(m)} \mid \boldsymbol{\beta})$, and the choice of probability distribution depends on the type of observed data. A Gaussian likelihood is assumed for observed continuous data:

$$p(y_{nd}^{(m)} \mid \boldsymbol{\beta}) = \text{Normal}\left(y_{nd}^{(m)} \mid \boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}, 1/\tau_d^{(m)}\right) \tag{7}$$

where $\boldsymbol{z}_{n:}$ is the vector of scores for the nth sample, $\boldsymbol{w}_{:d}^{(m)}$ is the vector of weights the dth feature from the mth matrix, and $\tau_d^{(m)}$ is the precision for that feature. A Bernoulli likelihood is assumed for observed binary data and the logistic link function $\pi(x) = (1+e^{-x})^{-1}$ is used:

$$p(y_{nd}^{(m)} \mid \boldsymbol{\beta}) = \text{Bernoulli}\left(y_{nd}^{(m)} \mid \pi\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right)\right) \tag{8}$$

A Poisson likelihood is assumed for observed counts data and the link function $\lambda(x) = \log(1+e^{x})$ is used:

$$p(y_{nd}^{(m)} \mid \boldsymbol{\beta}) = \text{Poisson}\left(y_{nd}^{(m)} \mid \lambda\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right)\right) \tag{9}$$

The assumed joint prior distribution, $p(\boldsymbol{\beta})$, is comprised of independent priors: $z_{nk} \sim \text{Normal}(0,\ 1)$, $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} s_{kd}^{(m)}$, $\hat{w}_{kd}^{(m)} \sim \text{Normal}(0,\ 1/\alpha_k^{(m)})$, $\alpha_k^{(m)} \sim \text{Gamma}(1e^{-14},\ 1e^{-14})$, $s_{kd}^{(m)} \sim \text{Bernoulli}(\theta_k^{(m)})$, $\theta_k^{(m)} \sim \text{Beta}(1,\ 1)$, $\tau_d^{(m)} \sim \text{Gamma}(1e^{-14},\ 1e^{-14})$.

MOFA uses mean-field variational inference (Grimmer, 2011; Fox and Roberts, 2012; Blei et al., 2017) to approximate the joint posterior distribution, $p(\boldsymbol{\beta}|\boldsymbol{Y})$, with a joint variational distribution factorized over J disjoint groups of variables:

$$\begin{aligned} q(\boldsymbol{\beta}) &= \prod_{j=1}^{J} q(\boldsymbol{\beta}_j) \\ &= \prod_{n=1}^{N}\prod_{k=1}^{K} q(z_{nk}) \prod_{m=1}^{M}\prod_{k=1}^{K} q(\alpha_k^{(m)})\ q(\theta_k^{(m)}) \prod_{m=1}^{M}\prod_{d=1}^{D_m} q(\tau_d^{(m)}) \prod_{m=1}^{M}\prod_{d=1}^{D_m}\prod_{k=1}^{K} q(\hat{w}_{kd}^{(m)}, s_{kd}^{(m)}) \end{aligned} \tag{10}$$

MOFA infers $q(\boldsymbol{\beta})$ iteratively until convergence. At each iteration, each $q(\boldsymbol{\beta}_j)$ is updated as

$$q(\boldsymbol{\beta}_j) \propto exp\{\mathbb{E}_{q_{-j}}[\log\ p(\boldsymbol{\beta}, \hat{\boldsymbol{Y}})]\} \tag{11}$$

where $\mathbb{E}_{q_{-j}}$ denotes an expectation with respect to the joint variational distribution, after removing $q(\boldsymbol{\beta}_j)$. The dataset $\hat{\boldsymbol{Y}}$ is derived by transformation of $\boldsymbol{Y}$. Observed data with a Gaussian assumed likelihood are transformed with feature-wise centering, which avoids

the need to estimate intercepts. Observed data with a non-Gaussian assumed likelihood are transformed to derive Gaussian pseudo-data. The derivation of Gaussian pseudo-data occurs at each iteration, and is based on a new parameter, $\zeta_{nd}^{(m)}$, which is derived for each sample n and feature d from matrix m. For observed data with a Bernoulli assumed likelihood, a precision parameter, $\tau_{nd}^{(m)}$, is introduced for each sample and feature as part of the transformation:

$$\hat{y}_{nd}^{(m)} = \frac{2y_{nd}^{(m)} - 1}{2\tau_{nd}^{(m)}} \tag{12}$$

$$\tau_{nd}^{(m)} = 2\lambda\left(\zeta_{nd}^{(m)}\right) \tag{13}$$

$$\left(\zeta_{nd}^{(m)}\right)^2 = \mathbb{E}_q\left[\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right)^2\right] \tag{14}$$

For observed data with a Poisson assumed likelihood, a precision parameter, $\tau_d^{(m)}$, is introduced for each feature as part of the transformation:

$$\hat{y}_{nd}^{(m)} = \zeta_{nd}^{(m)} - \frac{\pi\left(\zeta_{nd}^{(m)}\right)\left(1 - y_{nd}^{(m)}/\lambda\left(\zeta_{nd}^{(m)}\right)\right)}{\tau_d^{(m)}} \tag{15}$$

$$\zeta_{nd}^{(m)} = \mathbb{E}_q\left[\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(m)}\right] \tag{16}$$

$$\tau_d^{(m)} = 0.25 + 0.17 \times max\left(\boldsymbol{y}_{:d}^{(m)}\right) \tag{17}$$

where $\boldsymbol{y}_{:d}^{(m)}$ is the vector of observed values for the dth feature from the mth matrix. For both Bernoulli and Poisson observed data, the $\hat{y}_{nd}^{(m)}$ values are centered at each iteration, and $\zeta_{nd}^{(m)}$ values are derived using the factorization fit from the preceding iteration. MOFA monitors convergence with the evidence lower bound (ELBO), which is used to evaluate how well the variational distribution approximates the posterior distribution. The ELBO is calculated as:

$$\text{ELBO}(\boldsymbol{\beta}) = \mathbb{E}_q\left[\log p(\boldsymbol{Y}|\boldsymbol{\beta})\right] + \mathbb{E}_q\left[\log p(\boldsymbol{\beta})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\beta})\right] \tag{18}$$

For $\boldsymbol{Y}^{(m)}$ with non-Gaussian assumed likelihood, MOFA uses a lower bound for each $\log p\left(y_{nd}^{(m)}|\boldsymbol{\beta}\right)$. Maximizing this lower bound, coupled with the use of $\hat{y}_{nd}^{(m)}$ values, allows updates of $q(\boldsymbol{\beta})$ based on the assumption of Gaussian observed data (Jaakkola and Jordan, 2000; Seeger and Bouchard, 2012). MOFA assesses convergence at regular intervals, based on the percentage change in ELBO after. MOFA allows factors to be dropped during training, based on the fraction of variance explained for each matrix. After each iteration, MOFA identifies factors that do not explain a fraction of variance, for any omics matrix, over a threshold. MOFA then drops one of the identified factors.

4.3 Multi-omics data simulated with groundtruth factors

We simulated multi-omics datasets, from groundtruth factors, with various configurations. For each simulation configuration we generated 30 instances of a multi-omics dataset, $\boldsymbol{Y}$, consisting of matrices, $\boldsymbol{Y}^{(m)}, m = 1, 2, 3$. We split each $\boldsymbol{Y}$ into a target dataset, $\boldsymbol{T}$, and a learning dataset, $\boldsymbol{L}$. Each $\boldsymbol{Y}^{(m)} = \left[y_{nd}^{(m)}\right] \in \mathbb{R}^{N \times D_m}$ contained data for $N = N_t + N_l$ samples (rows) and $D_m = 2000$ features (columns), where $y_{nd}^{(m)}$ is the value for the nth sample and the dth feature from the mth matrix. N_t is the number of samples subsequently belonging to $\boldsymbol{T}$, and N_l is the number of samples belonging to $\boldsymbol{L}$. We generated each $\boldsymbol{Y}^{(m)}$ from a different statistical distribution, conditional on a random matrix of sample scores, $\boldsymbol{Z} = [z_{nk}] \in \mathbb{R}^{N \times K}$, and a random matrix of feature weights, $\boldsymbol{W}^{(m)} = \left[w_{kd}^{(m)}\right] \in \mathbb{R}^{K \times D_m}$. The kth column vector of $\boldsymbol{Z}$ contained sample scores for the kth groundtruth factor. The kth row vector of $\boldsymbol{W}^{(m)}$ contained feature weights for that same factor. We varied the number of groundtruth factors across configurations, using $K \in \{20, 30\}$.

We generated $\boldsymbol{Z}$ based on each sample being a member of a group. In each instance we created two groups of five samples belonging to $\boldsymbol{T}$, meaning N_t was always equal to 10 samples. We allowed N_l to vary across instances, with samples belonging to $\boldsymbol{L}$ being in differently sized groups of randomly selected sizes. We used either 20 learning groups of size $\in \{10, 20, 30\}$, or 40 groups of size $\in \{10, 25, 40\}$. For the nth sample, and kth groundtruth factor, we generated the score as $z_{nk} \sim \text{Normal}(\mu_{g(n)k},\ \sigma_z)$, where $\mu_{g(n)k}$ is the mean parameter for groundtruth factor k, for the group that sample n belonged to, $g(n)$. In each instance we selected $\mu_{g(n)k}$ randomly for each group and groundtruth factor, with probabilities $Pr(3) = 1/8$, $Pr(5) = 3/4$, $Pr(7) = 1/8$. The kth groundtruth factor was differentially active for $\boldsymbol{T}$ if $\mu_{g(n)k}$ differed between the two target dataset groups. For all instances of a given simulation configuration, the same standard deviation parameter, σ_z, was shared by all groups and groundtruth factors. We varied the latent noise-to-signal ratio across our simulation configurations by using $\sigma_z \in \{0.5, 1.0\}$

For the kth groundtruth factor, and the dth feature from the mth matrix, we generated the weight as $w_{kd}^{(m)} = \hat{w}_{kd}^{(m)} s_{kd}^{(m)}$. As such, each $w_{kd}^{(m)}$ was the product of a continuous random variable, $\hat{w}_{kd}^{(m)} \sim \text{Normal}(\mu^{(m)},\ \sigma_k^{(m)})$, and a binary random variable, $s_{kd}^{(m)} \sim \text{Bernoulli}(\theta_k^{(m)})$. We specified $\mu^{(m)}$, the mean parameter for the mth matrix, with $\mu^{(1)} = 5$; $\mu^{(2)} = 0$; $\mu^{(3)} = 0$. We generated , $\sigma_k^{(m)}$, the standard deviation parameter for the kth groundtruth factor and the mth matrix, with $\sigma_k^{(1)} \sim \text{Uniform}(0.5, 1.5)$; $\sigma_k^{(2)} \sim \text{Uniform}(0.5, 1.5)$; $\sigma_k^{(3)} \sim \text{Uniform}(0.1, 0.2)$. We generated the sparsity for the kth groundtruth factor and the mth matrix, $1 - \theta_k^{(m)}$, with $\theta_k^{(m)} \sim \text{Uniform}(0.15, 0.25)$.

We generated the values in each $\boldsymbol{Y}^{(m)}$ as:

$$
\begin{aligned}
y_{nd}^{(1)} &\sim \text{Poisson}\left(\log\left(1 + \exp\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(1)}\right)\right)\right) \\
y_{nd}^{(2)} &\sim \text{Normal}\left(\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(2)},\ \sigma_d \sim \text{Uniform}\left(0.25, 0.75\right)\right) \\
y_{nd}^{(3)} &\sim \text{Bernoulli}\left(1/\left(1 + \exp\left(-\boldsymbol{z}_{n:}\boldsymbol{w}_{:d}^{(3)}\right)\right)\right)
\end{aligned}
\tag{19}
$$

where $\boldsymbol{z}_{n:}$ is the vector of scores for the nth sample, $\boldsymbol{w}_{:d}^{(m)}$ is the vector of weights the dth feature from the mth matrix, and σ_d is the standard deviation for the dth feature.

We split each $\boldsymbol{Y}^{(m)}$ into $\boldsymbol{T}^{(m)} = \left[t_{nd}^{(m)}\right] \in \mathbb{R}^{N_t \times D_m}$, which contained values for the target group samples, and $\boldsymbol{L}^{(m)} = \left[l_{nd}^{(m)}\right] \in \mathbb{R}^{N_l \times D_m}$, which contained values for the learning group samples.

Before direct factorization with MOFA we pre-processed simulated $\boldsymbol{T}$ and $\boldsymbol{L}$ datasets by removing features with 0 variance across samples. Before factorization with transfer learning with MOTL, we pre-processed simulated $\boldsymbol{T}$ datasets by removing features that had 0 variance across samples or that had been removed from the corresponding $\boldsymbol{L}$ datasets.

4.4 TCGA multi-omics data acquisition and pre-processing

We used the R packages *TCGAbiolinks* (v.2.25.3) and *SummarizedExperiment* (v.1.28.0) to download and save TCGA mRNA expression, miRNA expression, DNA methylation, and simple nucleotide variation (SNV) data (Silva et al., 2016; Hutter and Zenklusen, 2018; Mounir et al., 2019). The mRNA and miRNA expression data consisted of raw counts. The DNA methylation data consisted of CpG site β-values, which had been derived from HM450 array intensities with R package *SeSAMe* (v.1.16.0) (Zhou et al., 2018). The SNV data consisted of masked somatic mutation files.

We created four reference datasets, using data from three cancer types; acute myeloid leukemia (LAML), pancreatic adenocarcinoma (PAAD) and skin cutaneous melanoma (SKCM). Each reference dataset, $\boldsymbol{R}$, contained multi-omics data for all samples from either two, or all three of the cancer types. We did not include SNV data in $\boldsymbol{R}$ datasets containing LAML samples, due to the sparsity of SNV data for LAML. We only used samples that had data for all omics of interest, and only included one sample per study participant. We thus had multi-omics data for 134 LAML samples, 157 PAAD samples and 435 SKCM samples. We then randomly split each $\boldsymbol{R}$ into non-overlapping target datasets. Each resulting target dataset, $\boldsymbol{T}$, contained multi-omics data for five samples per cancer type.

For the evaluation protocol based on TCGA multi-omics data, we merged data from the remaining 29 cancer types into a learning dataset, $\boldsymbol{L}$. For this $\boldsymbol{L}$ we only used samples that had data for all four omics, and only included one sample per study participant. This $\boldsymbol{L}$ contained multi-omics data (mRNA, miRNA, DNA methylation and SNV) for 7,217 samples.

For the application of MOTL to the pd-GBSC target datasets, we created a new learning dataset by merging data from all 32 cancer types. This new learning dataset contained multi-omics data (mRNA, miRNA, DNA methylation and SNV) for 7,866 samples.

Before direct factorization with MOFA, we pre-processed $\boldsymbol{R}$, $\boldsymbol{T}$ and $\boldsymbol{L}$ datasets in the same way. For mRNA data we removed genes that map to the Y chromosome. For both mRNA and miRNA we removed genes if they had a count of zero in $\geq 90\%$ of samples, or had zero variance across samples. We normalized mRNA and miRNA counts with the *DESeq2* (v.1.38.0) R package (Love et al., 2014), and $\log_2(x+1)$ transformed the normalized counts. For DNA methylation data, we removed CpG sites that map to the X or Y chromosome, were masked during SeSAMe quality control, had missing values in $\geq 20\%$ of samples, or had zero variance across samples. We converted DNA methylation β-values to M-values (Du et al., 2010). We included SNV records whose variant classification was either *Frame_Shift_Del*, *Frame_Shift_Ins*, *In_Frame_Del*, *In_Frame_Ins*, *Missense_Mutation*,

Nonsense_Mutation, *Nonstop_Mutation*, *Splice_Site* or *Translation_Start_Site*. We then created binary SNV matrices aggregated by gene and sample. We removed genes from SNV matrices if the mutation rate across samples was $\leq 1\%$. We filtered all omics to include only the 5,000 most variable features. We did not perform any batch effect correction on $\boldsymbol{L}$ datasets in order to preserve biological signal (Lee et al., 2020). We checked each $\boldsymbol{R}$ for batch effects with visualizations of UMAP co-ordinates (McInnes et al., 2020). We used the R package *uwot* (v.0.1.14) to derive UMAP coordinates from MOFA factorizations, and we did not observe the need to correct $\boldsymbol{R}$ datasets for batch effects.

Before factorization with transfer learning with MOTL, we pre-processed $\boldsymbol{T}$ datasets by removing all omics features that had zero variance across samples, or that had been removed from $\boldsymbol{L}$ during pre-processing. We used *DESeq2* to normalize mRNA and miRNA counts with the geometric means from $\boldsymbol{L}$, and then $\log_2(x+1)$ transformed the normalized counts. We converted DNA methlyation β-values to M-values, and converted SNV data to binary matrices after filtering on variant classification, as described previously.

4.5 Glioblastoma target dataset acquisition and pre-processing

We created four pd-GBSC target datasets, based on multi-omics profiling conducted by Santamarina-Ojeda et al. for four normal brain samples and nine patient-derived glioblastoma stem cell (pd-GBSC) cultures. The nine cancer samples had been previously classified into three subtypes thanks to transcriptome-based signatures: classical (CL), proneural (PN), and mesenchymal (MS). Each pd-GBSC target dataset contained mRNA expression and DNA methylation data for the four normal brain cortex samples, as well as either all nine cancer samples or just the samples from a subtype.

For factorization with transfer learning with MOTL, the pd-GBSC target datasets initially consisted of mRNA expression raw counts and DNA methylation β-values. Before factorization with MOTL, we pre-processed a pd-GBSC target dataset by removing all omics features that had zero variance across samples, or that had been removed from $\boldsymbol{L}$ during pre-processing. We used *DESeq2* to normalize mRNA counts with the geometric means from $\boldsymbol{L}$, and then $\log_2(x+1)$ transformed the normalized counts. We converted DNA methlyation β-values to M-values.

For direct MOFA factorization, without transfer learning, the pd-GBSC target datasets initially consisted of the same mRNA expression data, but already normalized and transformed by Santamarina-Ojeda et al., and DNA methylation β-values. Before direct MOFA factorization, we pre-processed mRNA data by removing genes that map to the Y chromosome, if they had a count of zero in $\geq 90\%$ of samples, or had zero variance across samples. We pre-processed DNA methylation data by removing CpG sites that had missing values in $\geq 20\%$ of samples, or had zero variance across samples. We converted DNA methylation β-values to M-values. We filtered both omics to include only the 5,000 most variable features.

4.6 Application of MOFA to simulated, TCGA and glioblastoma multi-omics datasets

We factorized simulated target, $\boldsymbol{T}$, and learning, $\boldsymbol{L}$, datasets with the MOFA Python implementation *mofapy2* (v.0.6.4). The number of factors we used for each MOFA factorization was equal to the lesser of the number of samples and the number of groundtruth

factors that were differentially active when simulating the dataset. The kth groundtruth factor was differentially active for a dataset if the mean parameter, $\mu_{g(n)k}$, for the sample scores for that factor, was not the same for all groups of samples in the dataset. We specified observed data likelihoods corresponding to those used for simulating $\boldsymbol{Y}^{(m)}$ matrices. We set the maximum number of iterations to 10,000 to ensure convergence. For the remaining settings we used the *mofapy2* defaults, meaning that all datasets were feature-wise centered during factorization fitting.

We factorized pre-processed TCGA reference, $\boldsymbol{R}$, target, $\boldsymbol{T}$, and learning, $\boldsymbol{L}$, datasets with the MOFA Python implementation *mofapy2* (v.0.7.0). We specified Gaussian as the observed data likelihood for mRNA, miRNA and DNA methylation data, and specified Bernoulli as the likelihood for SNV data. For the $\boldsymbol{L}$ datasets, we started the factorization with 100 factors and allowed factors to be dropped based on the fraction of variance explained, for which we set the threshold to 0.001. We set the threshold so low in order to retain factors that explained little of the variance in $\boldsymbol{L}$, yet could be potentially relevant for transfer learning. For $\boldsymbol{R}$ datasets we also started with 100 factors. For $\boldsymbol{T}$ datasets, we started with the maximum number of factors allowed by MOFA, which was either 10 factors (two cancer types) or 15 factors (three cancer types). For $\boldsymbol{R}$ and $\boldsymbol{T}$ datasets we dropped factors based on a threshold of 0.01, in order to only retain relevant factors. For all TCGA datasets, we set the maximum number of iterations to 10,000, to ensure convergence, and the frequency of convergence checking to five, to ensure that the algorithm had stopped dropping factors before converging.

When saving the factorizations of simulated and TCGA $\boldsymbol{L}$ datasets, we set the expectations argument to *all*. We did this to ensure that the point estimate for each precision parameter was saved in addition to those that are saved by default.

We factorized the pre-processed pd-GBSC target datasets with the MOFA Python implementation *mofapy2* (v.0.7.0). We specified Gaussian as the observed data likelihood for the mRNA and the DNA methylation data. We started with the maximum allowable number of factors and dropped factors based on a threshold of 0.01.

4.7 Application of MOTL to simulated, TCGA and glioblastoma multi-omics datasets

We applied MOTL to simulated, TCGA and pd-GBSC multi-omics target datasets. For each target dataset, we used point estimates of feature weight and precision values saved from the MOFA factorization of the corresponding learning, $\boldsymbol{L}$, dataset. For observed data with a Gaussian or Poisson assumed likelihood, the transferred value of the precision for each feature, $\tau_d^{(m)}$, was held fixed throughout iterations of MOTL updates. For observed data with a Bernoulli assumed likelihood, we initialized the value of the precision for each sample and feature, $\tau_{nd}^{(m)}$, with a feature-wise average, $\tau_d^{(m)}$, of the $\tau_{nd}^{(m)}$ values from the factorization of $\boldsymbol{L}$. The precisions for Bernoulli observed data were then iteratively updated by MOTL. We estimated intercepts using likelihoods assumed for $\boldsymbol{L}$, combined with outputs from the MOFA factorization of $\boldsymbol{L}$. For Gaussian observed data we calculated the intercept for the dth feature, from the mth matrix, as $a_d^{(m)} = \frac{1}{N_l}\sum_{n=1}^{N_l} l_{nd}^{(m)}$, where $l_{nd}^{(m)}$ denotes an uncentered learning dataset value after pre-processing. For Poisson and Bernoulli observed data we obtained maximum likelihood estimates of $a_d^{(m)}$ values, for which we used the *mle* function from the R package *stats4* (v.4.2.0). For Poisson observed data we

initialized each estimate with $a_d^{(m)} = \log\left(-1 + \exp\left(\frac{1}{N_l}\sum_{n=1}^{N_l} l_{nd}^{(m)}\right)\right)$, and for Bernoulli observed data we initialized it with $a_d^{(m)} = \log\left(\left(\frac{1}{N_l}\sum_{n=1}^{N_l} l_{nd}^{(m)}\right)\left(1 - \frac{1}{N_l}\sum_{n=1}^{N_l} l_{nd}^{(m)}\right)^{-1}\right)$.

When checking the ELBO for convergence, we used 0.0005% as the threshold, which is the default for MOFA. The algorithm was stopped when the absolute change in ELBO was under this threshold for two consecutive checks, and we set the maximum number of iterations to 10,000 to be consistent with our application of MOFA. For TCGA and pd-GBSC target datasets, we allowed factors to be dropped based on a threshold of 0.01 for the fraction of variance explained. We checked the ELBO after every five iterations, to ensure that the algorithm had stopped dropping factors before converging.

4.8 Evaluation methods

Groundtruth factors: For each simulated $\boldsymbol{T}$ (Methods 4.3), the groundtruth factor values were contained in the corresponding simulated $\boldsymbol{Z}$ and $\boldsymbol{W}^{(m)}$ matrices. The sample scores for the kth groundtruth factor were contained in $\boldsymbol{z}_{:k}$, the kth column vector of simulated $\boldsymbol{Z}$. The feature weights for the mth matrix, for that same groundtruth factor, were contained in $\boldsymbol{w}_{k:}^{(m)}$, the kth row vector of simulated $\boldsymbol{W}^{(m)}$. For each TCGA $\boldsymbol{T}$ (Methods 4.4), the groundtruth factors were based on the $\boldsymbol{R}$ dataset which we had split to create $\boldsymbol{T}$. We factorized each $\boldsymbol{R}$ with MOFA, and treated the inferred $\boldsymbol{z}_{:k}$ and $\boldsymbol{w}_{k:}^{(m)}$ vectors as groundtruth factors for each $\boldsymbol{T}$ that had been created by splitting $\boldsymbol{R}$.

Differentially active groundtruth factors: For each simulated and TCGA $\boldsymbol{T}$, groundtruth factor k was differentially active if the group means for groundtruth $\boldsymbol{z}_{:k}$ differed between the target dataset groups. For each simulated $\boldsymbol{T}$, this was the group mean, $\mu_{g(n)k}$, used to simulate groundtruth $\boldsymbol{z}_{:k}$. For each TCGA $\boldsymbol{T}$, the factorization of corresponding $\boldsymbol{R}$ provided groundtruth $\boldsymbol{z}_{:k}$ and $\boldsymbol{w}_{k:}^{(m)}$ factor vectors. We performed either the Wilcoxian Rank Sum test (two cancer types), or the Kruskal-Wallis test (three cancer types), on each groundtruth $\boldsymbol{z}_{:k}$ to determine if there was a statistically significant difference between the cancer types. We classed a groundtruth factor as differentially active if its BH-adjusted p-value was below 0.05.

Post-processing: We post-processed inferred and groundtruth $\boldsymbol{W}^{(m)}$ matrices before evaluation. We scaled each feature vector, $\boldsymbol{w}_{:d}^{(m)}$, by its Frobenius norm. We then centered each factor vector, $\boldsymbol{w}_{k:}^{(m)}$, of scaled values separately for each m. We then concatenated $\boldsymbol{w}_{k:}^{(m)}$ vectors to produce a single vector, $\boldsymbol{w}_{k:}$, of centered and scaled feature weights for each factor k.

Best hits: For each factorization of each simulated and TCGA $\boldsymbol{T}$, we identified the best hits between the factor vectors inferred with the factorization of $\boldsymbol{T}$, and the groundtruth factor vectors. For two sets of vectors $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_{K_v}\}$ and $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{K_x}\}$ we define the best hit for vector $\boldsymbol{v}_{k_v}$ as

$$\text{BestHit}\left(\boldsymbol{v}_{k_v}\right) = \underset{\boldsymbol{x}_{k_x}}{\arg\max}\ \text{cor}\left(\boldsymbol{v}_{k_v}, \boldsymbol{x}_{k_x}\right) \tag{20}$$

where $\text{cor}\left(\boldsymbol{v}, \boldsymbol{x}\right)$ is the Pearson correlation coefficient between vectors $\boldsymbol{v}$ and $\boldsymbol{x}$. We define the best hit for vector $\boldsymbol{x}_{k_x}$ as

$$\text{BestHit}\left(\boldsymbol{x}_{k_x}\right) = \underset{\boldsymbol{v}_{k_v}}{\arg\max}\ \text{cor}\left(\boldsymbol{x}_{k_x}, \boldsymbol{v}_{k_v}\right) \tag{21}$$

For each simulated $\boldsymbol{T}$, we identified best hits between inferred and groundtruth $\boldsymbol{w}_{k:}$ vectors. For each TCGA $\boldsymbol{T}$, we identified best hits between inferred and groundtruth $\boldsymbol{w}_{k:}$ vectors, as well as between inferred and groundtruth $\boldsymbol{z}_{:k}$ vectors. We used shared features when calculating correlations for $\boldsymbol{w}_{k:}$ vectors, and we used shared samples for $\boldsymbol{z}_{:k}$ vectors. We calculated p-values for the correlations, and only considered correlations with a p-value < 0.05 (two-sided alternative hypothesis) when identifying best hits.

F-measure values: For each factorization of each TCGA $\boldsymbol{T}$, we calculated an F-measure value to assess the overall correlation between factor vectors inferred with the factorization of $\boldsymbol{T}$, and groundtruth factor vectors. We based this on the F-measure presented by Saelens et al., which we adapted in order to assess correlations. For a given set of inferred factor vectors, $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_{K_v}\}$, and a set of groundtruth factor vectors, $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{K_x}\}$, we calculated the F-measure as

$$FM = 2/((1/Relevance) + (1/Recovery)) \tag{22}$$

where

$$Relevance = \frac{1}{K_v}\sum_{k_v=1}^{K_v} \text{cor}\left(\boldsymbol{v}_{k_v}, \text{BestHit}\left(\boldsymbol{v}_{k_v}\right)\right) \tag{23}$$

and

$$Recovery = \frac{1}{K_x}\sum_{k_x=1}^{K_x} \text{cor}\left(\boldsymbol{x}_{K_x}, \text{BestHit}\left(\boldsymbol{x}_{K_x}\right)\right) \tag{24}$$

Here $\text{cor}\left(\boldsymbol{v}, \boldsymbol{x}\right)$ is the Pearson correlation coefficient between vectors $\boldsymbol{v}$ and $\boldsymbol{x}$. We calculated F-measure values for sets of inferred and groundtruth $\boldsymbol{w}_{k:}$ vectors, as well as for $\boldsymbol{z}_{:k}$ vectors.

F1 scores: We calculated F1 scores to evaluate the factorizations of simulated and TCGA $\boldsymbol{T}$ datasets:

$$\begin{aligned} &F1 = (2 \times Precision \times Recall)\ /\ (Precision + Recall) \\ &Precision = True\ Positives\ /\ Predicted\ Positives \\ &Recall = True\ Positives\ /\ Actual\ Positives \end{aligned} \tag{25}$$

Actual Positives were the groundtruth factors of $\boldsymbol{T}$, that were differentially active.

Predicted Positives were the groundtruth factors that were predicted as being differentially active, based on the factorization of $\boldsymbol{T}$. We firstly performed either the Wilcoxian Rank Sum test (two groups), or the Kruskal-Wallis test (three groups), on each $\boldsymbol{z}_{:k}$ vector inferred with the factorization of $\boldsymbol{T}$, and classed factors with a p-value < 0.05 as differentially active. For inferred factors classed as differentially active, we identified the best hits for their corresponding inferred $\boldsymbol{w}_{k:}$ vectors. We selected these best hits from the

set of groundtruth $\boldsymbol{w}_{k:}$ vectors for $\boldsymbol{T}$. If groundtruth $\boldsymbol{w}_{k:}$ was selected as a best hit for a differentially active inferred factor, then groundtruth factor k was predicted as being differentially active.

True Positives were the differentially active groundtruth factors that were predicted as being differentially active, based on the factorization of $\boldsymbol{T}$.

Statistical testing of differences between factorization methods: We calculated the differences in evaluation measures between factorization methods, and tested the statistical significance of these differences. To do this we fit generalized least squares regressions with the R package *nlme* (v.3.1.157) (Pinheiro and Bates, 2000). We fit a single regression to model the F1 scores for simulated data. For TCGA data we fit a separate regression for each evaluation measure. For each regression we modelled $y_i = \beta_0 + \boldsymbol{d}_i\boldsymbol{\beta}_d + \boldsymbol{f}_i\boldsymbol{\beta}_f + \epsilon_i$. The vector $\boldsymbol{d}_i = (d_{i1}, ..., d_{iT})$ indicates the simulation configuration, or cancer type, that y_i relates to, and vector $\boldsymbol{f}_i = (f_{i1}, ..., f_{iM})$ indicates the factorization method. Vectors $\boldsymbol{\beta}_d = (\beta_{d1}, ..., \beta_{dT})^\top$ and $\boldsymbol{\beta}_f = (\beta_{f1}, ..., \beta_{fM})^\top$ are estimated fixed effects and ϵ_i is the residual. We incorporated correlations between residuals from the same target dataset using the compound symmetry structure method. We calculated contrasts for the factorization method effects in $\boldsymbol{\beta}_f$ using the R package *emmeans* (v.1.8.7) (Searle et al., 1980), and used Tukey-adjusted p-values for assessing statistical significance.

Differentially active factors from glioblastoma target datasets: We identified differentially active factors from the MOTL factorization of each pd-GBSC target dataset, as well as from the direct MOFA factorization (without transfer learning), of each pd-GBSCs target dataset. We performed the Wilcoxian Rank Sum test on each $\boldsymbol{z}_{:k}$ vector inferred with the factorization of a pd-GBSC target dataset. We classed factors with a BH-adjusted p-value < 0.05 as differentially active between the normal samples and the cancer samples.

Gene set enrichment analysis: We used R package *fgsea* (v.1.24.0) (Sergushichev, 2016) to perform gene set enrichment analysis on differentially active groundtruth factors that were true positives for factorizations of TCGA $\boldsymbol{T}$ datasets. For each differentially active groundtruth TCGA factor k, we analysed vector $\boldsymbol{w}_k^{(m)}$ if the fraction of mRNA variance explained by k was > 0.01, and where m corresponded to the mRNA matrix. We tested KEGG, REACTOME, GO:BP, and GO:MF gene sets that have a size of between 15 and 500 genes, obtained using the R package *msigdbr* (v.7.5.1). We used an BH-adjusted p-value cutoff of 0.01 for selecting enriched gene sets. We also performed gene set enrichment analysis on differentially active factors from the pd-GBSC target datasets, and used the same criteria as outlined above for differentially active groundtruth TCGA factors.

4.9 Implementation

An open source R implementation of MOTL, as well as code for reproducing these analyses, is available at `https://github.com/david-hirst/MOTL`. The factorization fit of the full TCGA learning dataset, used for the application of MOTL to the pd-GBSC target dataset, is available at `https://zenodo.org/doi/10.5281/zenodo.10847986`.

We used a Dell computer with 20 cores at 3GHz, and 64 GB of RAM, to perform factorizations. To pre-process and factorize the $\boldsymbol{L}$ used in the TCGA evaluation protocol, it took 26,405 seconds (over seven hours). Hence, we have made the factorization of a large

TCGA $\boldsymbol{L}$ dataset publicly available for transfer learning. It took an average of 37 seconds to pre-process a $\boldsymbol{T}$ dataset, comprised of four omics, and factorize it directly with MOFA. The average time increased to 134 seconds for MOTL.

5 Supplementary Information

Supplementary Table 1 contains gene sets associated with differentially active groundtruth TCGA factors.

Supplementary Table 2 contains the percentage of variance explained, by groundtruth TCGA factors, for each omics (MOFA factorizations of TCGA reference datasets).

Supplementary Table 3 contains gene sets associated with factors differentially active between all pd-GBSC samples and normal samples.

Supplementary Table 4 contains gene sets associated with factors differentially active between all pd-GBSC samples and normal samples, as well as those associated with factors differentially active between pd-GBSC subtype samples (CL, MS and PN) and normal samples.

6 Acknowledgements

We would like to thank Carl Herrmann, Céline Chevalier, Kim-Anh Lê Cao, Lionel Spinelli and Olivia Angelin-Bonnet for helpful feedback and discussions.

7 Funding

This research was supported by l'Agence Nationale de la Recherche (ANR), project ANR-21-CE45-0001-01, by France 2030 PEPR Digital Health managed by ANR, project ANR-22-PESN-0013, by the Royal Society of New Zealand, Catalyst project CSG-MAU1902 and by the Association Française contre les Myopathies (AFM).

References

Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J. C., and Stegle, O. (2020). MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology*, 21(1):111.

Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J. C., Buettner, F., Huber, W., and Stegle, O. (2018). Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, 14(6):e8124. Publisher: John Wiley & Sons, Ltd.

Banerjee, J., Taroni, J. N., Allaway, R. J., Prasad, D. V., Guinney, J., and Greene, C. (2023). Machine learning in rare disease. *Nature Methods*, pages 1–12. Publisher: Nature Publishing Group.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877. arXiv: 1601.00670.

Cantini, L., Zakeri, P., Hernandez, C., Naldi, A., Thieffry, D., Remy, E., and Baudot, A. (2021). Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. *Nature Communications*, 12(1):124. Number: 1 Publisher: Nature Publishing Group.

Chalise, P. and Fridley, B. L. (2017). Integrative clustering of multi-level 'omic data based on non-negative matrix factorization algorithm. *PLOS ONE*, 12(5):e0176278. Publisher: Public Library of Science.

Chauvel, C., Novoloaca, A., Veyre, P., Reynier, F., and Becker, J. (2020). Evaluation of integrative clustering methods for the analysis of multi-omics data. *Briefings in Bioinformatics*, 21(2):541–552. Publisher: Oxford University Press / USA.

Collado-Torres, L., Nellore, A., Kammers, K., Ellis, S. E., Taub, M. A., Hansen, K. D., Jaffe, A. E., Langmead, B., and Leek, J. T. (2017). Reproducible RNA-seq analysis using recount2. *Nature Biotechnology*, 35(4):319–321. Number: 4 Publisher: Nature Publishing Group.

Conesa, A. and Beck, S. (2019). Making multi-omics data accessible to researchers. *Scientific Data*, 6(1):251. Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Comments & Opinion Publisher: Nature Publishing Group Subject_term: Data integration;Genome Subject_term_id: data-integration;genome.

Davis-Marcisak, E. F., Fitzgerald, A. A., Kessler, M. D., Danilova, L., Jaffee, E. M., Zaidi, N., Weiner, L. M., and Fertig, E. J. (2021). Transfer learning between preclinical models and human tumors identifies a conserved NK cell activation signature in anti-CTLA-4 responsive tumors. *Genome Medicine*, 13(1).

Dermitzakis, E. T. (2008). From gene expression to disease risk. *Nature Genetics*, 40(5):492–493. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 5 Primary_atype: News & Views Publisher: Nature Publishing Group.

Dong, A., Li, Z., and Zheng, Q. (2021). Transferred Subspace Learning Based on Non-negative Matrix Factorization for EEG Signal Classification. *Frontiers in Neuroscience*, 15.

Du, P., Zhang, X., Huang, C.-C., Jafari, N., Kibbe, W. A., Hou, L., and Lin, S. M. (2010). Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis. *BMC Bioinformatics*, 11(1):587.

Fertig, E. J., Ding, J., Favorov, A. V., Parmigiani, G., and Ochs, M. F. (2010). CoGAPS: an R/C++ package to identify patterns and biological process activity in transcriptomic data. *Bioinformatics*, 26(21):2792–2793.

Fox, C. W. and Roberts, S. J. (2012). A tutorial on variational Bayesian inference. *Artificial Intelligence Review*, 38(2):85–95.

Grimmer, J. (2011). An Introduction to Bayesian Inference via Variational Approximations. *Political Analysis*, 19(1):32–47.

Huang, S., Chaudhary, K., and Garmire, L. X. (2017). More Is Better: Recent Progress in Multi-Omics Data Integration Methods. *Frontiers in Genetics*, 8:84.

Hutter, C. and Zenklusen, J. C. (2018). The Cancer Genome Atlas: Creating Lasting Value beyond Its Data. *Cell*, 173(2):283–285.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37.

Lee, A. J., Park, Y., Doing, G., Hogan, D. A., and Greene, C. S. (2020). Correcting for experiment-specific variability in expression compendia can remove underlying signals. *GigaScience*, 9(11).

Lock, E. F., Hoadley, K. A., Marron, J., and Nobel, A. B. (2013). JOINT AND INDIVIDUAL VARIATION EXPLAINED (JIVE) FOR INTEGRATED ANALYSIS OF MULTIPLE DATA TYPES. *The annals of applied statistics*, 7(1):523–542.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550.

Manzoni, C., Kia, D. A., Vandrovcova, J., Hardy, J., Wood, N. W., Lewis, P. A., and Ferrari, R. (2018). Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences. *Briefings in Bioinformatics*, 19(2):286–302.

Mao, W., Zaslavsky, E., Hartmann, B. M., Sealfon, S. C., and Chikina, M. (2019). Pathway-level information extractor (PLIER) for gene expression data. *Nature Methods*, 16(7):607–610.

McInnes, L., Healy, J., and Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [cs, stat].

Mo, Q., Shen, R., Guo, C., Vannucci, M., Chan, K. S., and Hilsenbeck, S. G. (2018). A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. *Biostatistics (Oxford, England)*, 19(1):71–86.

Mounir, M., Lucchetta, M., Silva, T. C., Olsen, C., Bontempi, G., Chen, X., Noushmehr, H., Colaprico, A., and Papaleo, E. (2019). New functionalities in the TCGAbiolinks package for the study and integration of cancer data from GDC and GTEx. *PLOS Computational Biology*, 15(3):e1006701. Publisher: Public Library of Science.

Peng, M., Li, Y., Wamsley, B., Wei, Y., and Roeder, K. (2021). Integration and transfer learning of single-cell transcriptomes via cFIT. *Proceedings of the National Academy of Sciences*, 118(10):e2024383118.

Pierre-Jean, M., Deleuze, J.-F., Le Floch, E., and Mauger, F. (2020). Clustering and variable selection evaluation of 13 unsupervised methods for multi-omics data integration. *Briefings in Bioinformatics*, 21(6):2011–2030.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-PLUS*. Statistics and Computing. Springer, New York.

Rappoport, N. and Shamir, R. (2018). Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Research*, 46(20):10546–10562.

Rohart, F., Gautier, B., Singh, A., and Cao, K.-A. L. (2017). mixOmics: An R package for 'omics feature selection and multiple data integration. *PLOS Computational Biology*, 13(11):e1005752. Publisher: Public Library of Science.

Saelens, W., Cannoodt, R., and Saeys, Y. (2018). A comprehensive evaluation of module detection methods for gene expression data. *Nature Communications*, 9(1):1090.

Santamarina-Ojeda, P., Tejedor, J. R., Pérez, R. F., López, V., Roberti, A., Mangas, C., Fernández, A. F., and Fraga, M. F. (2023). Multi-omic integration of DNA methylation and gene expression data reveals molecular vulnerabilities in glioblastoma. *Molecular Oncology*, 17(9):1726–1743. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/1878-0261.13479.

Searle, S. R., Speed, F. M., and Milliken, G. A. (1980). Population Marginal Means in the Linear Model: An Alternative to Least Squares Means. *The American Statistician*, 34(4):216–221. Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/00031305.1980.10483031.

Seeger, M. and Bouchard, G. (2012). Fast Variational Bayesian Inference for Non-Conjugate Matrix Factorization Models. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 1012–1018. PMLR. ISSN: 1938-7228.

Sergushichev, A. A. (2016). An algorithm for fast preranked gene set enrichment analysis using cumulative statistic calculation. *bioRxiv*, page 060012. Publisher: Cold Spring Harbor Laboratory Section: New Results.

Sharma, G., Colantuoni, C., Goff, L. A., Fertig, E. J., and Stein-O'Brien, G. (2020). projectR: an R/Bioconductor package for transfer learning via PCA, NMF, correlation and clustering. *Bioinformatics*, 36(11):3592–3593.

Silva, T. C., Colaprico, A., Olsen, C., D'Angelo, F., Bontempi, G., Ceccarelli, M., and Noushmehr, H. (2016). TCGA Workflow: Analyze cancer genomics and epigenomics data using Bioconductor packages. *F1000Research*, 5:1542.

Stein-O'Brien, G. L., Arora, R., Culhane, A. C., Favorov, A. V., Garmire, L. X., Greene, C. S., Goff, L. A., Li, Y., Ngom, A., Ochs, M. F., Xu, Y., and Fertig, E. J. (2018). Enter the Matrix: Factorization Uncovers Knowledge from Omics. *Trends in Genetics*, 34(10):790–805.

Stein-O'Brien, G. L., Clark, B. S., Sherman, T., Zibetti, C., Hu, Q., Sealfon, R., Liu, S., Qian, J., Colantuoni, C., Blackshaw, S., Goff, L. A., and Fertig, E. J. (2019). Decomposing Cell Identity for Transfer Learning across Cellular Measurements, Platforms, Tissues, and Species. *Cell Systems*, 8(5):395–411.e8.

Subramanian, I., Verma, S., Kumar, S., Jere, A., and Anamika, K. (2020). Multi-omics Data Integration, Interpretation, and Its Application. *Bioinformatics and Biology Insights*, 14:1177932219899051. Publisher: SAGE Publications Ltd STM.

Taroni, J. N., Grayson, P. C., Hu, Q., Eddy, S., Kretzler, M., Merkel, P. A., and Greene, C. S. (2019). MultiPLIER: A Transfer Learning Framework for Transcriptomics Reveals Systemic Features of Rare Disease. *Cell Systems*, 8(5):380–394.e4.

Tini, G., Marchetti, L., Priami, C., and Scott-Boyer, M.-P. (2019). Multi-omics integration—a comparison of unsupervised clustering methodologies. *Briefings in Bioinformatics*, 20(4):1269–1279.

Veeramachaneni, S. D., Pujari, A. K., Padmanabhan, V., and Kumar, V. (2019). A Maximum Margin Matrix Factorization based Transfer Learning Approach for Cross-Domain Recommendation. *Applied Soft Computing*, 85:105751.

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

Zhou, W., Triche, Jr, T. J., Laird, P. W., and Shen, H. (2018). SeSAMe: reducing artifactual detection of DNA methylation by Infinium BeadChips in genomic deletions. *Nucleic Acids Research*, 46(20):e123.

4.4 Discussion

The results of my main PhD research project are discussed in the article presented in section 4.3. Here, I would like to reiterate the main conclusions, and add some further discussion.

I developed MOTL, an approach for multi-omics matrix factorization with transfer learning. MOTL infers latent variable values for a multi-omics target dataset, comprised of a small number of samples. MOTL factorizes the target dataset by incorporating latent variable values already inferred from the factorization of a learning dataset. I developed, and implemented, two protocols to evaluate MOTL, and also applied MOTL to a dataset comprised of a small number of cancer and control samples. With this, I demonstrated that MOTL can enhance the discovery of biological processes and pathways associated with a biological condition of interest.

A key assumption, made when applying a transfer learning approach to matrix factorization, is that the learning and target datasets share an overlapping latent space associated with shared biological processes. This seemed to be valid for the learning and target datasets used in the evaluation of MOTL, which consisted of cancer samples, albeit cancers of different types. However, a motivation for developing MOTL was to enhance the ability of multi-omics matrix factorization to facilitate the study of rare diseases. For a target dataset, with samples from a rare disease, the latent space may share less in common with that of the learning dataset, than was the case in the MOTL evaluations. Therefore, as an important next step, it would be beneficial to evaluate MOTL using the same learning dataset, but with target datasets that contain samples from diseases that are not cancer. In such an evaluation, it would be interesting to calculate a measure of similarity, between the target and learning datasets, based on the biological conditions that characterize the samples. Perhaps such a measure could be calculated via the exploration of a disease-disease network, in which the nodes represent diseases, and edges represent associations. Ideally this measure of difference would correlate with the metrics used in the evaluation. Then researchers, wishing to apply MOTL, could calculate the similarity between their target dataset, and a learning dataset, to estimate if they are sufficiently similar for transfer learning to be effective.

In addition to the degree of similarity between the learning and target datasets, it would beneficial to explore both how the number of samples in the learning dataset, and the heterogeneity of their biological conditions, impact the effectiveness of the transfer learning approach. This would require evaluating MOTL using different learning datasets, and perhaps assigning each learning dataset with a heterogeneity score. Again, a disease-disease network, or some other network in which biological conditions are nodes, could be used to estimate the heterogeneity of the learning dataset.

There are likely to be situations, when there are biological processes that are associated with the conditions characterizing a target dataset, but not represented by the latent variables inferred from the learning dataset. Therefore, it would be beneficial to add some flexibility into the MOTL model, so that the latent variables inferred for

the target dataset represent a union of those inferred with transfer learning, and those inferred with direct factorization of the target dataset.

A final aspect of this work, that I'd like to discuss, is the methodology I used for evaluating MOTL. A major challenge I encountered, both during this project, and the project outlined in chapter 5, was the difficulty in evaluating unsupervised dimension reduction methods. When using real data, it is difficult find a *groundtruth* against which to compare the lower dimensional representations. In the evaluation of MOTL, I factorized larger datasets (called *reference* datasets in the article) and treated the resulting latent variables as the groundtruth. In particular, I was interested in the groundtruth latent variables that were *differentially active* between cancer types, and whether MOTL was able to infer differentially active latent variables that resembled these. This was motivated by observations made by other researchers (section 3.3.5) that, when a multi-omics dataset is generated from a small number of samples, joint matrix factorization may struggle to separate the underlying biological signals. In the event of a future evaluation, I would be interested in evaluating MOTL based on using the inferred latent variables for sample clustering. Additionally, multi-omics matrix factorization can also be evaluated based on using the inferred latent variables in survival analysis. I was interested in evaluating MOTL in this way, but was concerned that the small number of samples would make survival analysis infeasible. Therefore, an extension of the evaluation protocol could be based on combining MOTL outputs from different target datasets, and performing survival analysis using the latent variable values for the combined set of samples.

5 OntoVAE extension

Contents

5.1 Context . . . 106
5.2 Introduction . . . 107
5.3 The OntoVAE model . . . 107
5.4 OntoVAE with weighted loss . . . 109
5.4.1 Weighted loss statistical implications . . . 110
5.4.2 Weighted loss implementation . . . 112
5.5 Evaluation of OntoVAE with weighted loss . . . 112
5.5.1 Methods and materials . . . 112
5.5.1.1 OntoVAE versions . . . 112
5.5.1.2 Pre-trained OntoVAE model . . . 113
5.5.1.3 RNA-Seq datasets . . . 113
5.5.1.4 Fine-tuning with versions of OntoVAE . . . 114
5.5.1.5 Calculation of weights . . . 115
5.5.1.6 AUC values for latent variables . . . 116
5.5.2 Evaluation results . . . 116
5.6 Discussion . . . 121

5.1 Context

During my PhD, I successfully applied for a DAAD scholarship, a research grant awarded by the German Academic Exchange Service. I used the DAAD scholarship to fund a research stay within Dr. Carl Herrmann's team (https://www.hdsu.org/) at the Institute for Pharmacy and Molecular Biotechnology, Heidelberg University, Germany. In this chapter I present the work I carried out, in collaboration with Dr. Herrmann's team, while I was in Heidelberg.

I spent six weeks in Heidelberg, during November and December 2023, conducting research into variational autoencoders (VAEs) (section 2.5.2.1). I was interested in learning about the VAE approach to unsupervised dimension reduction, as it is an alternative to matrix factorization (section 2.5.1) that falls into the broader machine learning category of deep learning. These characteristics meant that learning about VAEs promised to expand my knowledge of unsupervised dimension reduction, while allowing me to gain some experience in the field of deep learning. Additionally, VAEs employ variational inference (section 2.2.3), meaning I could build upon the

theoretical and algorithmic expertise I developed in variational inference during my main PhD research project (chapter 4).

Dr. Herrmann's team is engaged in the development and application of both matrix factorization and VAE algorithms, for the analysis of omics data. Therefore, I was able to benefit from their expertise in these areas, both from their collaboration in the research I conducted while in Heidelberg, as well as from discussions and group seminars focused on team members' different research areas. Additionally, while in Heidelberg, I was able to present results from my main PhD research project to Dr. Herrmann's team, as well as to another team, obtaining valuable feedback.

5.2 Introduction

The objective of this work was to implement, and evaluate, a modification to OntoVAE [125] (section 3.2.3), a VAE previously developed by members of Dr. Herrmann's team. OntoVAE takes a matrix of transcriptomic data as input, and has an architecture constrained by the incorporation of a directed acyclic graph (DAG) (Figure 3.3). In OntoVAE, latent and observed variables in the bottleneck, decoder, and reconstruction layers, represent nodes in the incorporated DAG, allowing for easier interpretation of the resulting lower dimensional representation.

My work was motivated by the hypothesis that weighting the contribution of each gene to the loss function, based on some prior information, would further improve the meaningfulness of the latent variables inferred with OntoVAE. I implemented a modification to OntoVAE, allowing the optional inclusion of a vector of weights, representing some prior information. These weights allow the user to control the relative contribution of genes to the loss function used for model fitting. Then, using real RNA-Seq datasets, I derived gene weights and evaluated the modification, comparing the lower dimensional representations to those obtained with OntoVAE without using the weights.

This chapter begins with a presentation of theoretical and algorithmic characteristics of the OntoVAE model (section 5.3). The implemented modification to OntoVAE is presented in section 5.4, followed by a description of the evaluation methods and results in section 5.5.

5.3 The OntoVAE model

OntoVAE consists of two neural networks, an encoder and a decoder (section 2.5.2.1). The encoder is parameterized by $\boldsymbol{\phi}$, and the decoder is parameterized by $\boldsymbol{\theta}$. We consider an $N \times D$ matrix $\boldsymbol{Y}$, containing RNA-seq data for N samples and D genes. Vector $\boldsymbol{y}_{n:}$ contains data for the nth sample, and element y_{nd} is the expression value for the nth sample and dth gene.

The input to the encoder, for the nth sample, is $\boldsymbol{y}_{n:}$. The outputs of the encoder are vectors $\boldsymbol{\mu}_{n:}$ and $\boldsymbol{\sigma}^2_{n:}$. These vectors contain the parameters of the joint variational

distribution (section 2.2.3) for $\boldsymbol{z}_{n:} \in \mathbb{R}^{1\times K}$, the bottleneck layer vector of latent variables. The variational distribution is

$$q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right) = \prod_{k=1}^{K} q_k\left(z_{nk}|\boldsymbol{y}_{n:}\right) \tag{5.1}$$

$$q_k\left(z_{nk}|\boldsymbol{y}_{n:}\right) = \text{Normal}\left(\mu_{nk}, \sigma^2_{nk}\right) \tag{5.2}$$

where z_{nk} is the kth latent variable in the bottleneck layer vector, μ_{nk} is the kth element of $\boldsymbol{\mu}_{n:}$, and σ^2_{nk} is the kth element of $\boldsymbol{\sigma}^2_{n:}$. Both $\boldsymbol{\mu}_{n:}$ and $\boldsymbol{\sigma}^2_{n:}$ are functions of $\boldsymbol{y}_{n:}$ and $\boldsymbol{\phi}$.

The input to the decoder, for the nth sample, is $\boldsymbol{z}_{n:}$. The decoder then consists of consecutive hidden layers of latent variables, whose values are outputs of functions of the values of variables from preceding layers. The final output of the decoder is the reconstruction layer, $\hat{\boldsymbol{y}}_{n:} \in \mathbb{R}^{1\times D}$, in which each element, $\hat{y}_{nd}$, is ultimately a function of $\boldsymbol{z}_{n:}$ and $\boldsymbol{\theta}$.

In OntoVAE, the latent variables in the bottleneck and hidden decoder layers are constrained to represent nodes in an incorporated DAG. For example, if the DAG representing GO biological process terms (section 3.2.1) is used to constrain the model, each latent variable represents a GO biological process term. The value of a latent variable is only an input to the function producing the value of another latent variable, if there is a corresponding directional edge in the DAG. Additionally, for the dth gene, the value of a latent variable is only an input to the function producing $\hat{y}_{nd}$, if there is an edge between its GO term and the dth gene. Before incorporating a DAG into OntoVAE, it can be trimmed by removing nodes if they annotate too many or too few genes, based on user defined thresholds. The DAG can also be trimmed by only including edges of a certain type.

For the $\boldsymbol{Y}$ matrix, OntoVAE is trained with the objective of minimizing the loss function

$$\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}\right) = \frac{1}{N}\sum_{n} \hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.3}$$

where

$$\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) = -\frac{\lambda}{2}\sum_{k}\left(1+\log\left(\sigma^2_{nk}\right)-\mu^2_{nk}-\sigma^2_{nk}\right)+\sum_{d}\left(y_{nd}-\hat{y}_{nd}\right)^2 \tag{5.4}$$

is the loss for the nth sample. Here the user defined parameter λ, called the *KL coefficient*, controls the contribution of $\text{KL}\left(q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right)||p\left(\boldsymbol{z}_{n:}\right)\right)$ to the loss (see equation 5.16). Prior to training, the samples are randomly partitioned into a training set, $\mathscr{T}$, and a validation set, $\mathscr{V}$. Minibatches of samples are randomly created for the training set, as well as for the validation set. Then, the model is trained over a number of epochs. In each training epoch, the algorithm iterates over the training set minibatches, updating $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ at each iteration.

For the bth training set minibatch, $\mathscr{T}_b$, of T_b samples, $\boldsymbol{z}_{n:}$ is simulated for $n \in \mathscr{T}_b$ using the reparameterization trick. More specifically, the reparameterization trick

proceeds as follows: auxiliary variables ϵ_{nk} are simulated from

$$p(\epsilon_{nk}) = \text{Normal}(0,1) \tag{5.5}$$

and each z_{nk} is derived as

$$z_{nk} = \mu_{nk} + \epsilon_{nk}\sigma_{nk} \tag{5.6}$$

Then, an empirical risk is calculated

$$\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{T}_b}\right) = \frac{1}{T_b}\sum_{n\in\mathscr{T}_b}\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.7}$$

The gradients of $\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{T}_b}\right)$ are calculated with respect to the parameters in $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. This allows the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ to be updated with gradient descent [98].

At the end of each epoch, the model generates $\boldsymbol{z}_{n:}$ and $\hat{\boldsymbol{y}}_{n:}$ for samples in the validation set. Convergence is assessed based on the change in the validation loss

$$\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{V}}\right) = \frac{1}{B_{\mathscr{V}}}\sum_{b}\frac{1}{V_b}\sum_{n\in\mathscr{V}_b}\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.8}$$

where $B_{\mathscr{V}}$ is the number of validation set minibatches, and $\mathscr{V}_b$ is the bth validation set minibatch of V_b samples.

Once an OntoVAE model has been trained, it can be used to generate latent variable values from $\boldsymbol{y}_{n:}$ vectors. The values of the latent variables can be interpreted as representing the activity of the corresponding nodes, such as GO biological process terms, from the DAG that was used to constrain the model. The values of the latent variables can be compared across samples to identify nodes of interest. For example, GO biological process terms corresponding to latent variables that vary significantly between groups of samples, may be associated with the biological conditions that characterize those groups.

5.4 OntoVAE with weighted loss

We hypothesized that the meaningfulness of the latent variable values inferred with OntoVAE could be improved by weighting the contribution of each gene to the loss function. In this modified OntoVAE model, we replace $\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$ with

$$\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) = -\frac{\lambda}{2}\sum_{k}\left(1+\log\left(\sigma_{nk}^2\right)-\mu_{nk}^2-\sigma_{nk}^2\right)+\sum_{d}w_d\left(y_{nd}-\hat{y}_{nd}\right)^2 \tag{5.9}$$

where w_d is a weight, controlling the contribution of the dth gene to the loss function. The value of w_d is based on some prior information about the extent to which the dth gene is associated with the biological conditions that characterize $\boldsymbol{Y}$. The empirical

risk, used for calculating gradients for updating parameter values, now becomes

$$\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{T}_b}\right)=\frac{1}{T_b}\sum_{n\in\mathscr{T}_b}\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.10}$$

The validation loss, used for monitoring convergence, now becomes

$$\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{V}}\right)=\frac{1}{B_{\mathscr{V}}}\sum_{b}\frac{1}{V_b}\sum_{n\in\mathscr{V}_b}\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.11}$$

5.4.1 Weighted loss statistical implications

Before implementing the optional weighted loss modification to OntoVAE, I checked how this would impact the statistical assumptions underlying OntoVAE. I found that training OntoVAE with $\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$, instead of $\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$, is equivalent to forcing an assumption of heterogeneous variance on the sampling distribution. I illustrate this equivalence by comparing the OntoVAE algorithm to the Auto-Encoding Variational Bayes (AEVB) algorithm [153] (section 2.5.2.1).

Recall, from section 2.5.2.1, that a potential Stochastic Gradient Variational Bayes (SGVB) estimator for the AEVB algorithm is

$$\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)=-\mathrm{KL}\left(q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right)||p\left(\boldsymbol{z}_{n:}\right)\right)+\log p\left(\boldsymbol{y}_{n:}|\boldsymbol{z}_{n:}\right) \tag{5.12}$$

At each iteration of the AEVB algorithm, a random minibatch of M samples, $\mathscr{M}$, is used to calculate

$$\hat{\mathscr{L}}_{AEVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{M}}\right)=\frac{N}{M}\sum_{n\in\mathscr{M}}\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.13}$$

The gradients of $\hat{\mathscr{L}}_{AEVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{M}}\right)$ are calculated, and $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are updated with gradient *ascent*. If we use the variational distribution, $q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right)$, described by equations 5.1 and 5.2, and make prior distributional assumptions

$$p(\boldsymbol{z}_{n:})=\prod_{k=1}^{K}p\left(z_{nk}\right) \tag{5.14}$$

$$p\left(z_{nk}\right)=\mathrm{Normal}\left(0,1\right) \tag{5.15}$$

then [153]

$$-\mathrm{KL}\left(q\left(\boldsymbol{z}_{n:}|\boldsymbol{y}_{n:}\right)||p\left(\boldsymbol{z}_{n:}\right)\right)=\frac{1}{2}\sum_{k}\left(1+\log\left(\sigma_{nk}^2\right)-\mu_{nk}^2-\sigma_{nk}^2\right) \tag{5.16}$$

If y_{nd} are assumed to be conditionally independent with sampling distribution

$$p\left(\boldsymbol{y}_{n:}|\boldsymbol{z}_{n:}\right)=\prod_{d=1}^{D}p\left(y_{nd}|\boldsymbol{z}_{n:}\right) \tag{5.17}$$

$$p\left(y_{nd}|\boldsymbol{z}_{n:}\right) = \text{Normal}\left(\hat{y}_{nd}, \sigma^2\right) \tag{5.18}$$

then

$$\log p\left(\boldsymbol{y}_{n:}|\boldsymbol{z}_{n:}\right) = -\frac{1}{2}\sum_d \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_d \left(y_{nd} - \hat{y}_{nd}\right)^2 \tag{5.19}$$

and $\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$ becomes

$$\frac{1}{2}\left(\sum_k \left(1 + \log\left(\sigma_{nk}^2\right) - \mu_{nk}^2 - \sigma_{nk}^2\right) - \sum_d \log\left(2\pi\sigma^2\right) - \frac{1}{\sigma^2}\sum_d \left(y_{nd} - \hat{y}_{nd}\right)^2\right) \tag{5.20}$$

If we denote the constant $C_1 = \sigma^2 \sum_d \log\left(2\pi\sigma^2\right)$, then we can derive

$$2\sigma^2 \hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) + C_1 = \sigma^2 \sum_k \left(1 + \log\left(\sigma_{nk}^2\right) - \mu_{nk}^2 - \sigma_{nk}^2\right) - \sum_d \left(y_{nd} - \hat{y}_{nd}\right)^2 \tag{5.21}$$

Combining equations 5.4 and 5.21, and setting $\lambda = 2\sigma^2$ in equation 5.4, gives

$$2\sigma^2 \hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) + C_1 = -\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.22}$$

Minimization based on $\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{T}_b}\right)$ (equation 5.7) is therefore equivalent to maximization based on $\hat{\mathscr{L}}_{AEVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathscr{M}}\right)$ (equation 5.13) when the sampling variance is assumed to be homogeneous (equation 5.18).

If, instead of assuming homogeneous sampling variance, we assume heterogeneous sampling variance

$$p\left(y_{nd}|\boldsymbol{z}_{n:}\right) = \text{Normal}\left(\hat{y}_{nd}, \frac{\sigma^2}{w_d}\right) \tag{5.23}$$

then now

$$\log p\left(\boldsymbol{y}_{n:}|\boldsymbol{z}_{n:}\right) = -\frac{1}{2}\sum_d \log\left(\frac{2\pi\sigma^2}{w_d}\right) - \frac{1}{2\sigma^2}\sum_d w_d \left(y_{nd} - \hat{y}_{nd}\right)^2 \tag{5.24}$$

and $\hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$ becomes

$$\frac{1}{2}\left(\sum_k \left(1 + \log\left(\sigma_{nk}^2\right) - \mu_{nk}^2 - \sigma_{nk}^2\right) - \sum_d \log\left(\frac{2\pi\sigma^2}{w_d}\right) - \frac{1}{\sigma^2}\sum_d w_d \left(y_{nd} - \hat{y}_{nd}\right)^2\right) \tag{5.25}$$

If we now denote constant $C_2 = \sigma^2 \frac{1}{2}\sum_d \log\left(\frac{2\pi\sigma^2}{w_d}\right)$, then we can now derive

$$2\sigma^2 \hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) + C_2 = \sigma^2 \sum_k \left(1 + \log\left(\sigma_{nk}^2\right) - \mu_{nk}^2 - \sigma_{nk}^2\right) - \sum_d w_d \left(y_{nd} - \hat{y}_{nd}\right)^2 \tag{5.26}$$

Combining equations 5.9 and 5.26, and setting $\lambda = 2\sigma^2$ in equation 5.9, gives

$$2\sigma^2 \hat{\mathscr{L}}_{SGVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) + C_2 = -\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right) \tag{5.27}$$

Minimization based on $\hat{\mathcal{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathcal{T}_b}\right)$ (equation 5.10) is therefore equivalent to maximization based on $\hat{\mathcal{L}}_{AEVB}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathcal{M}}\right)$ (equation 5.13) when the sampling variance is assumed to be heterogeneous (equation 5.23) instead of homogeneous.

5.4.2 Weighted loss implementation

OntoVAE is a Python package consisting of several modules. To implement and evaluate the proposed modification to OntoVAE, I created a forked OntoVAE GitHub repository, `https://github.com/david-hirst/onto-vae`. I implemented the weighted loss option in the `onto_vae/vae_model.py` module in this repository. The implementation includes a new argument in the function called to train OntoVAE

```
def train_model(..., rec_loss_wts=None)
```

To train an OntoVAE model with the weighted loss, the user specifies a path to a `csv` file containing the weights. For example

```
ontovae_model.train_model(....,
rec_loss_wts = os.path.join(os.getcwd(),'gene_weights.csv')
)
```

The first column of the `csv` file should contain gene symbols consistent with those used in the loaded ontology object. The second column should contain a weight for each gene. The vector of raw weights from the csv file, $\boldsymbol{r} \in \mathbb{R}^{D\times 1}$, is scaled to give

$$w_d = \frac{r_d \times D}{\sum_d r_d} \tag{5.28}$$

where r_d is the raw weight for the dth gene, and w_d is the weight used for that gene when training OntoVAE with $\hat{\mathcal{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$ (equation 5.9).

5.5 Evaluation of OntoVAE with weighted loss

After implementing the weighted loss option for OntoVAE, I used four different RNA-Seq datasets (section 5.5.1.3) to evaluate the implementation. For each RNA-Seq dataset, I inferred a lower dimensional representation with each of three versions of OntoVAE, called Unweighted, Derived, and Random (section 5.5.1.1). I then compared the versions of OntoVAE by assessing how well their inferred latent variables enabled sample classification. The following sections describe the evaluation methods (section 5.5.1) and results (section 5.5.2)

5.5.1 Methods and materials

5.5.1.1 OntoVAE versions

I compared latent variables inferred with three different versions of OntoVAE:

- **Unweighted**: OntoVAE with the original unweighted loss, $\hat{\mathscr{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$ (equation 5.4)
- **Derived**: OntoVAE with the weighted loss, $\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$ (equation 5.9), using derived gene weights
- **Random**: OntoVAE with the weighted loss, $\hat{\mathscr{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{y}_{n:}\right)$, using randomly generated gene weights

5.5.1.2 Pre-trained OntoVAE model

For each RNA-Seq dataset, and version of OntoVAE, I inferred a lower dimensional representation by fine-tuning a pre-trained OntoVAE model (section 5.5.1.4). Fine-tuning [189] is a transfer learning approach to model fitting (section 3.4), in which the parameters (or a subset of them) of a pre-trained model are updated by retraining the model on a new training dataset. The model parameters are initialized using the values from the pre-trained model, after which the model is trained on the new dataset, updating the parameters that are not held fixed.

I chose to fine-tune a pre-trained OntoVAE model, for each RNA-Seq dataset, and version of OntoVAE, instead training models from scratch, for two reasons: 1) Fine-tuning allowed for less training epochs to be used, and removed the need to perform extensive parameter tuning, both of which were appealing given my limited time in Heidelberg; 2) With the team in Heidelberg, we felt that the RNA-Seq datasets used in the evaluations did not have enough samples for training an OntoVAE model effectively from scratch.

I used a pre-trained OntoVAE model, called recount3_GTEx_GO_best_model.pt, available from the *OntoVAE (Ontology guided VAE) manuscript* project on `https://figshare.com`. This model had been trained [125], with the original unweighted loss (section 5.3), on GTEx v8 data obtained from the recount3 database [203]. The GTEx v8 release contains bulk human RNA-Seq data for 19,214 samples, from 972 individuals and 32 tissue types.

The model was constrained by a trimmed DAG of 3,083 GO biological process terms and 19,387 annotated genes. The model had been trained for 300 epochs, with KL coefficient $\lambda = 1e-4$.

5.5.1.3 RNA-Seq datasets

I used four different RNA-Seq datasets, for fine-tuning the pre-trained model with the different versions of OntoVAE. Each RNA-Seq dataset, $\boldsymbol{Y}^{(s)}$, was a matrix of GTEx RNA-Seq data for a different subset, $\mathscr{S}_s$, of the GTEx dataset samples. Each subset contained samples from two tissue types, with only one sample selected, per individual, per tissue type. This gave the following number of samples for each subset:

- $\mathscr{S}_1$ contained 241 liver and 252 spleen samples
- $\mathscr{S}_2$ contained 396 brain and 349 pancreas samples
- $\mathscr{S}_3$ contained 422 adipose tissue and 422 breast samples
- $\mathscr{S}_4$ contained 582 heart and 842 muscle samples

For $\mathcal{S}_3$, the 422 samples from each tissue type were from individuals for whom there was a sample for both tissue types. I added this extra condition for $\mathcal{S}_3$ because I assumed that breast samples would be from females only, but adipose tissue samples would be from both males and females. I derived the gene weights with differential expression analysis (section 5.5.1.5), and wanted the weights to reflect differences between tissues, not sex.

5.5.1.4 Fine-tuning with versions of OntoVAE

For each RNA-Seq dataset, I fine-tuned the pre-trained model with the Unweighted, Derived, and Random versions of OntoVAE (section 5.5.1.1). I performed each fine-tuning independently, initializing the model parameters, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, with their values from the original pre-trained model. Then, during fine-tuning, the initialized parameters were updated based on the RNA-Seq dataset, and OntoVAE version, used for the fine-tuning. Each fine-tuning was carried out for a training set, $\mathcal{T}_s$, of samples randomly selected from $\mathcal{S}_s$, and was run for 100 epochs.

In each epoch, the parameters of the model were updated based on the empirical risk function used for the fine-tuning. For the Unweighted version, the empirical risk function was $\hat{\mathcal{L}}_{OV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathcal{T}_b}\right)$ (equation 5.7). For the Derived version, the empirical risk function was $\hat{\mathcal{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathcal{T}_b}\right)$ (equation 5.10), with derived gene weights. For the Random version, the empirical risk was also $\hat{\mathcal{L}}_{wOV}\left(\boldsymbol{\phi},\boldsymbol{\theta}|\boldsymbol{Y}_{\mathcal{T}_b}\right)$, but with randomly generated gene weights.

For each fine-tuning, I used a validation set, $\mathcal{V}_s$, consisting of the samples not selected for $\mathcal{T}_s$. I assessed the convergence of the fine-tuning based on the validation loss (equation 5.8 or 5.11 depending on the OntoVAE version), calculated for the samples in $\mathcal{V}_s$. For each fine-tuning, I checked convergence by plotting the value of the validation loss for each epoch. I was fairly satisfied with the shape of the curves after 100 epochs, as they had largely flattened out, indicating the models were close to convergence. For an example of the validation loss curves, see Figure 5.1, which shows the curves for fine-tuning with derived gene weights in the loss function.

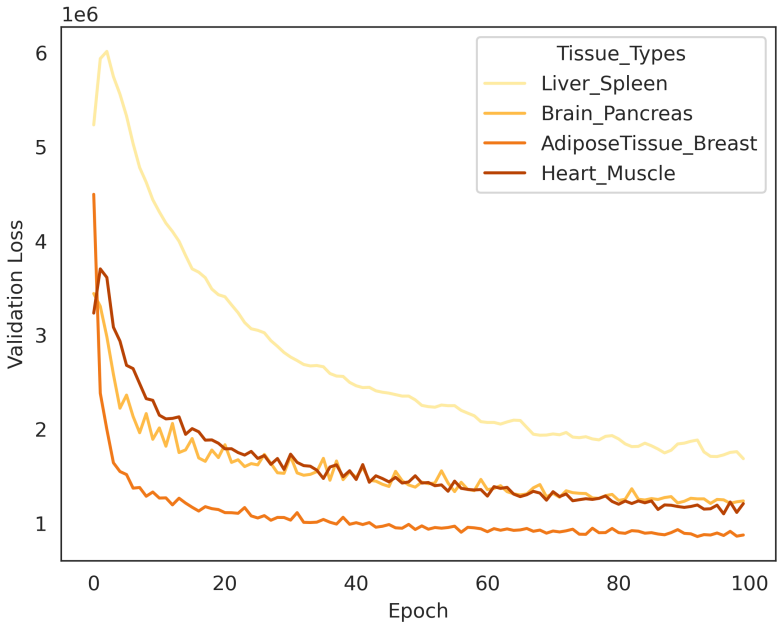


Figure 5.1 – **Validation loss curves from fine-tuning OntoVAE with the derived gene weights**. Each line shows the validation loss for a fine tuning with the Derived version of OntoVAE (weighted loss with derived gene weights). Each line is for a fine-tuning with a different RNA-Seq dataset, in which each sample is from one of two tissue types. The x-axis is the epoch, and the y-axis is the value of the validation loss for that epoch

5.5.1.5 Calculation of weights

For each RNA-Seq dataset (section 5.5.1.3), I derived gene weights by performing a differential expression analysis (section 2.4.3) with DESeq2 [121]. I compared the gene expression values for samples from one tissue type, to those for samples from the other tissue type. This yielded a contrast estimate for each gene. For RNA-Seq dataset $\boldsymbol{Y}^{(s)}$, the raw derived weight for the dth gene was

$$r_{sd} = |\hat{\delta}_{sd}| \tag{5.29}$$

where $\hat{\delta}_{sd}$ was the contrast estimate for the dth gene, for $\boldsymbol{Y}^{(s)}$. I hypothesized that using these weights, based on differential expression analysis, when carrying out fine-tuning, might force more importance to be given to latent variables representing GO terms associated with genes that vary the most across the two tissue types.

For fine-tuning with randomly generated weights, I generated the raw random weight, for the dth gene, for $\boldsymbol{Y}^{(s)}$, as

$$r_{sd} \sim \text{Uniform}\,(0,1) \tag{5.30}$$

5.5.1.6 AUC values for latent variables

For each RNA-Seq dataset (section 5.5.1.3), $\boldsymbol{Y}^{(s)}$, corresponding to sample subset $\mathscr{S}_s$, I evaluated each fine-tuned model (section 5.5.1.4) by firstly using it to generate latent variable values from $\boldsymbol{y}_{n:}^{(s)}$ vectors, for $n \in \mathscr{S}_s$. For each fine-tuned model and latent variable, I then trained a Gaussian naive Bayes classifier [204] on the latent variable values generated for samples in $\mathscr{S}_s$. For each RNA-Seq dataset, fine-tuned model, and latent variable, I then calculated an area under the ROC curve (AUC) value [205], and used these AUC values for evaluating the fine-tuned models.

For a continuous variable x, and a class variable c, a Gaussian naive Bayes classifier is trained to assign a class, $\hat{c}$, to a new observation of x, based on Bayes' rule (section 2.2)

$$\hat{c} = \arg\max_{c} p(c)\, p(x|c) \tag{5.31}$$

The prior distribution, $p(c)$, and sampling distributions, $p(x|c) = \text{Normal}\left(\mu_c, \sigma_c^2\right)$, are inferred from a training dataset.

The classes I used when training the classifier were the two tissue types. I used 10-fold cross validation when training the classifier. In K-fold cross validation [206], a dataset is randomly split into K subsets, called folds. For the kth fold, the model is trained on the other folds, and the kth fold is used as a validation set. For each of the 10 folds, I computed the AUC obtained from applying the classifier to the fold. Higher AUC values indicate the trained classifier is more able to correctly classify new samples. I then computed the median AUC for the 10 folds. This gave a median AUC value for each latent variable, from each fine-tuned model and RNA-Seq dataset. I used these median AUC values as the AUC values when evaluating fine-tuned models.

5.5.2 Evaluation results

I evaluated the weighted loss implementation of OntoVAE by comparing the latent variables for three different versions of OntoVAE (section 5.5.1.1): OntoVAE with the original unweighted loss (Unweighted); OntoVAE with the weighted loss using derived gene weights (Derived); OntoVAE with the weighted loss using randomly generated gene weights (Random). For each version, I generated latent variable values after fine-tuning, with that version, a pre-trained OntoVAE model (section 5.5.1.2). I performed fine-tuning (section 5.5.1.4), and generated latent variable values, for each of four RNA-Seq datasets (section 5.5.1.3).

For each RNA-Seq dataset, I firstly compared the fine-tuned models based on the distributions of the AUC values calculated for their latent variables (section 5.5.1.6). For all RNA-Seq datasets, there was an upwards shift in the distribution of AUC values for the model fine-tuned with derived gene weights (Derived), compared to the model fine-tuned with no weights (Unweighted) (Figure 5.2). This indicated that weighting the contribution of each gene to the loss function, based on some prior information, yielded latent variables that better differentiated between groups of samples.

For liver and spleen, the latent variables from the unweighted model classified the

samples well, and using the derived weights led to only a very small improvement in the distribution of AUC values. However, for the other tissue type pairs, using the derived weights led to larger visible improvements in the distributions of AUC values. The distributions of AUC values were the same for models fine-tuned with random weights (Random) and models fine-tuned with no weights, indicating that the uplift from the derived weights was not simply due to the inclusion of weights, but depended on the information contained in those weights.

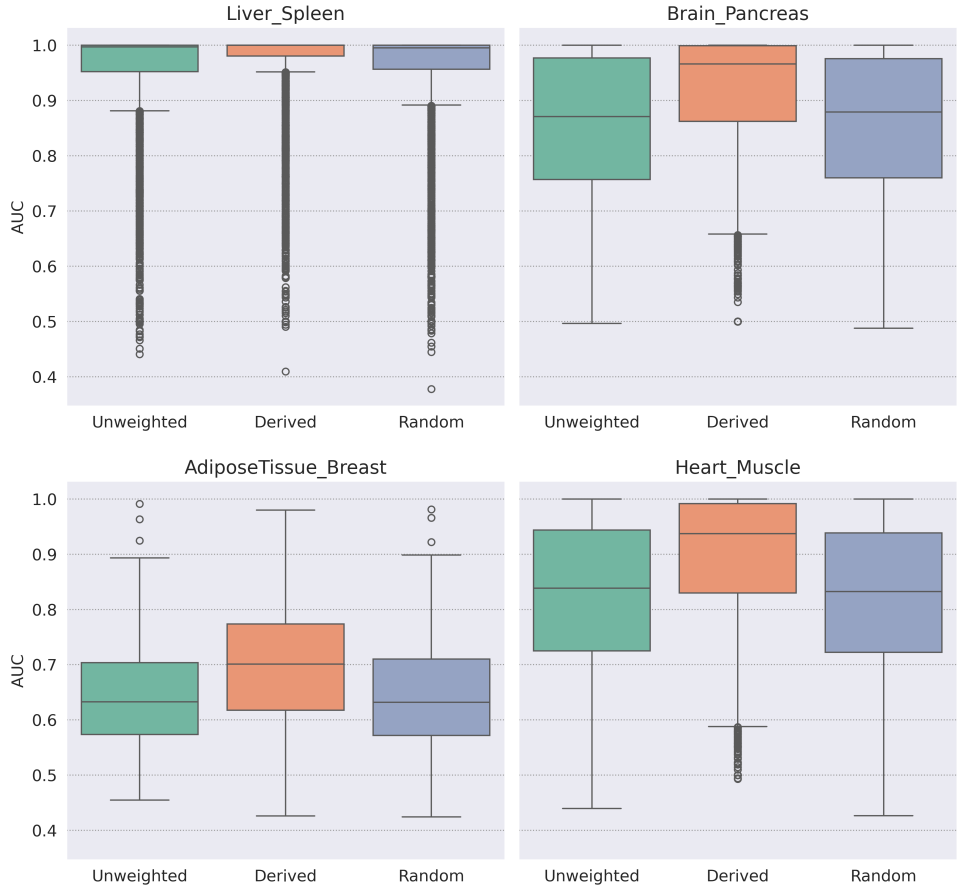


Figure 5.2 – **Boxplots of AUC values**. The boxplots represent the AUC values obtained for latent variables representing GO biological process terms. Each plot is for a different RNA-Seq dataset, in which each sample is from one of two tissue types. Distributions of AUC values for the three different fine-tuned models are plotted separately. Models were fine-tuned without weights (Unweighted), with derived weights (Derived), and with randomly generated weights (Random). For each fine-tuned model, and latent variable, the AUC value is the median area under the ROC curve value obtained by training a classifier with 10-fold cross validation. The higher the AUC value is, the better a latent variable served for classification of samples with respect to tissue type

For each RNA-Seq dataset and latent variable, I then calculated a change in AUC. I calculated the change as: the AUC obtained from the model fine-tuned with derived gene weights (Derived), less the AUC obtained from the model fine-tuned with no weights (Unweighted). Figure 5.3 contains violin plots of the changes in AUC, for each RNA-Seq dataset. For all RNA-Seq datasets, there were more latent variables whose AUC value was higher for the model fine-tuned with derived weights, than

for the model fine-tuned with no weights, indicating an improvement from using the derived weights. For the liver and spleen RNA-Seq data, due to a large number of latent variables with unchanged AUC, the proportion of positive changes is more easily seen in Figure 5.4, which contains empirical cumulative distribution plots for the changes in AUC.

For liver and spleen samples, 23% of latent variables had a negative change and 42% of latent variables had a positive change, while for many latent variables (35%) there was no change. For brain and pancreas samples, 18% of latent variables had a negative change and 73% of latent variables had a positive change. For adipose and breast samples, 25% of latent variables had a negative change and 74% of latent variables had a positive change. For heart and muscle samples, 18% of latent variables had a negative change and 78% of latent variables had a positive change. For each RNA-Seq dataset, I performed the Wilcoxon signed-rank test [207] to test whether the distribution of changes in AUC was symmetric about zero. The p-value was < 0.01 for each RNA-Seq dataset, indicating that the ratio of positive to negative changes was statistically significant for each subset.

We can conclude that, weighting the contribution of each gene to the loss function, based on some prior information, resulted in more latent variables that better differentiated between groups of samples, than latent variables for which there was a decline in this ability.

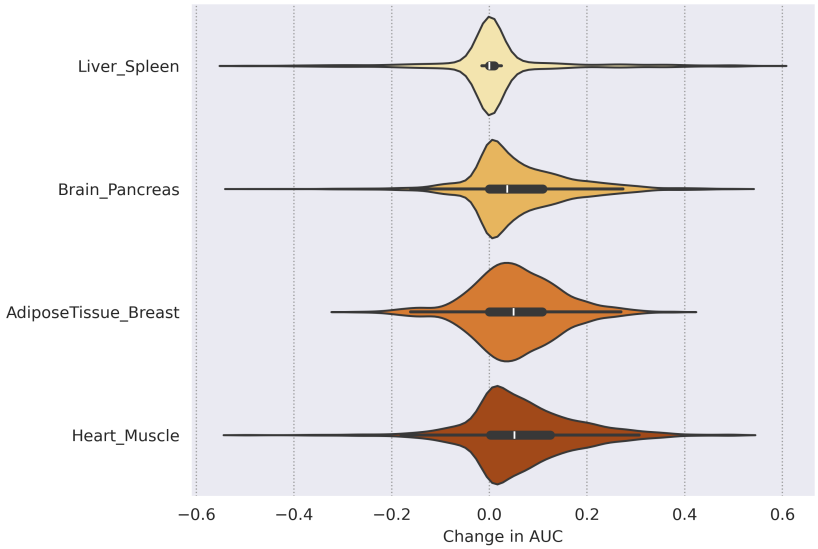


Figure 5.3 – **Violin plots of changes in AUC values**. The violin plots represent the changes in AUC values, for latent variables representing GO biological process terms. Each plot is for a different RNA-Seq dataset, in which each sample is from one of two tissue types. For each RNA-Seq dataset and latent variable, the change in AUC was calculated as: the AUC for the model fine-tuned with derived weights less the AUC for the model fine-tuned with no weights. For a given latent variable, a positive change in AUC indicated that the inclusion of the derived weights led to improved classification of samples with respect to tissue type

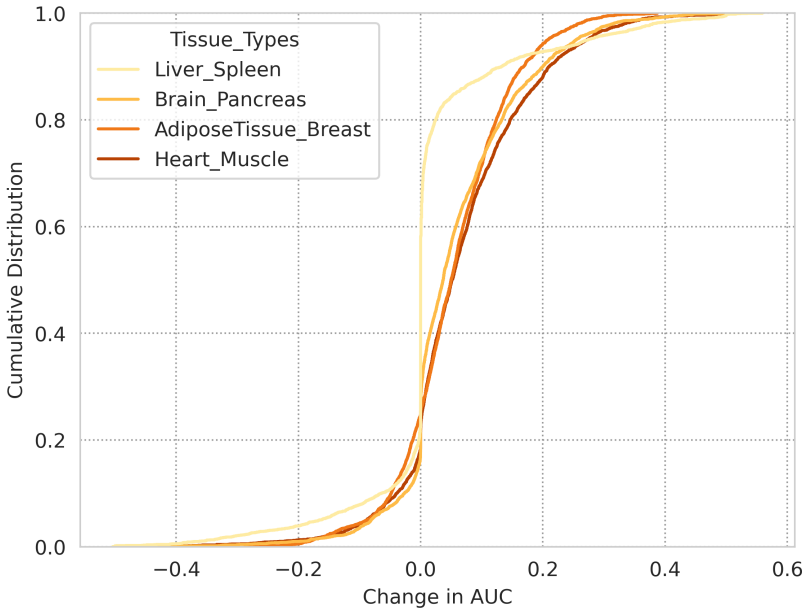


Figure 5.4 – **Empirical cumulative distributions of changes in AUC values**. Each line represents the empirical cumulative distribution of changes in AUC values, for latent variables representing GO biological process terms. Each line is for a different RNA-Seq dataset, in which each sample is from one of two tissue types. For each RNA-Seq dataset and latent variable, the change in AUC was calculated as: the AUC for the model fine-tuned with derived weights less the AUC for the model fine-tuned with no weights. For a given latent variable, a positive change in AUC indicated that the inclusion of the derived weights led to improved classification of samples with respect to tissue type

I then looked at the GO terms associated with the latent variables, to determine which terms had the largest positive changes in AUC (Derived vs Unweighted). Table 5.1 shows the five terms with the largest positive changes in AUC, by RNA-Seq dataset tissue type pair.

5.6 Discussion

I implemented a modification to OntoVAE, allowing the inclusion of weights that control the relative contribution of genes to the loss function. I evaluated the implementation by fine-tuning a pre-trained OntoVAE model, using different subsets of the original training data for each fine-tuning. I observed that weighting the contributions of genes to the loss, based on prior differential expression analysis, resulted in latent variable values that better differentiated between groups of samples, when used to train a classifier. This initial, proof-of-concept, analysis suggests it would be

Tissue types	**GO biological process term**
Liver, Spleen	regulation of neuroinflammatory response positive regulation of vasoconstriction striated muscle tissue development connective tissue development cellular senescence
Brain, Pancreas	positive regulation of reproductive process mesonephric epithelium development kidney epithelium development cell cycle G2/M phase transition response to reactive oxygen species
Adipose, Breast	Wnt signaling pathway, calcium modulating pathway Wnt signaling pathway maintenance of cell number cell surface receptor signaling pathway involved in cell-cell signaling canonical Wnt signaling pathway
Heart, Muscle	skeletal muscle tissue development muscle tissue development striated muscle tissue development cellular response to acid chemical positive regulation of leukocyte mediated immunity

Table 5.1 – GO terms with the largest positive changes in AUC, achieved from fine-tuning OntoVAE with derived gene weights, by RNA-Seq dataset

worthwhile to further evaluate training and/or fine-tuning OntoVAE models with the weighted loss function.

A limitation of this work, is that I used the same data for deriving the weights as for fine-tuning the model, potentially biasing the evaluations to favor the weighted loss version of OntoVAE. Therefore, a potential next step would be to incorporate weights that are based on prior information that is derived independently of the input RNA-Seq dataset. For example, this could be based on prior differential expression analysis that has been performed on a different RNA-Seq dataset, characterized by the same biological conditions. In the case of an input RNA-Seq dataset with disease and control samples, the weights could be based on prior, independent, knowledge of genomic variants that are expected to be associated with the disease.

Another limitation relates to the methodology I used for comparing the different versions of OntoVAE. I made the assumption that, if the weighted loss function can improve the ability of OntoVAE to reveal biological signals, underlying some biological conditions of interest, then this would result in latent variables that better differentiate between samples from those biological conditions. However, with latent variables representing GO biological process terms, it is likely that not all latent variables should better differentiate between the biological conditions being studied. Therefore, these evaluations would benefit from more in-depth analysis of which GO biological process terms were represented by the latent variables that, when generated using the weighted loss version of OntoVAE, better differentiated between samples from different tissue types.

For the evaluations, I fine-tuned the models using data that was included in the dataset used for training the pre-trained model. This strategy was chosen due to time constraints, as well as the number of samples in the RNA-Seq datasets used in the evaluations. It was assumed that the pre-trained model parameters would already be able to generate latent variable values relevant for the tissue types in the evaluation datasets, but that these could be improved by further training on data from those tissue types only. In future evaluations, it would be of interest to use new data to either fine-tune a pre-trained model, or train a new model from scratch, and evaluate the impact of the weighted loss function.

Another potential extension of this study is related to the statistical implications of the weighted loss modification to OntoVAE. As discussed in section 5.4.1, the modification is equivalent to forcing an assumption of heterogeneous variance on the sampling distribution. The results of this study suggest that it could be of interest to investigate a weighted least squares approach to OntoVAE. In such an approach, the weights could be allowed to change after each epoch, based on the observed variance between the input and reconstruction layers.

Finally, future work for extending OntoVAE could involve introducing flexibility, by allowing the removal and addition of connections between the latent variables and the values in the reconstruction layer. As discussed in [125], these connections are currently hard-coded according to the incorporated DAG, preventing OntoVAE from learning new connections. In the initial stages of planning my stay in Heidelberg, we had discussed a project in which such flexibility could be introduced, via incorporating an additional term into the loss function, penalizing dissimilarity between the connections in the model and the connections in the DAG. We eventually decided that the six weeks was not long enough for this modification, but it would be interesting to revisit this challenge.

6 Analysis of FSHD2 and BAMS omics data

Contents

6.1 Context 124
6.2 Introduction 124
6.3 Summary of results 125
6.4 My contribution 125
6.5 Article 127

6.1 Context

During both my MSc and PhD, I participated in a collaborative work led by Dr. Frédérique Magdinier's team (https://www.marseille-medical-genetics.org/en/f-magdinier/) at the Marseille Medical Genetics research center. Dr. Magdinier is a biologist, whose team's area of expertise includes the study of epigenomic patterns in certain rare diseases. The specialization of my MSc was in Statistics, and I had previously had only limited experience with concepts, research methodologies, and data from the field of biology. Participation in this work represented my first interdisciplinary collaboration with biologists, and first in-depth exposure to the opportunities and challenges presented by omics data analysis.

During this collaboration we analyzed transcriptomic (section 1.3.3) and epigenomic (section 1.3.2) data from patients with type 2 facioscapulohumeral muscular dystrophy (FSHD2), patients with Bosma arhinia and microphthalmia syndrome (BAMS), and healthy controls. In this chapter, I summarize the background (section 6.2) and results (section 6.3) of this work, highlight my contribution (section 6.4), and present the resulting published article [208] (section 6.5).

6.2 Introduction

Facioscapulohumeral muscular dystrophy (FSHD) [209, 210] is a rare genetic disorder characterized by muscle weakness and atrophy in the face, shoulders, and upper arms, with an estimated prevalence of 5:100,000 to 12:100,000. FSHD is associated with a modification in chromatin structure (section 1.2.3) in the D4Z4 region of chromosome 4. In 95% of FSHD cases, known as type 1 FSHD, the modified chromatin

structure is associated with a contraction of the D4Z4 region. Most of the remaining cases are FSHD2, in which the modified chromatin structure is associated with DNA sequence variants (section 1.3.1.2) in the SMCHD1 gene. It is expected that in both cases, genomic or epigenomic changes observed in FSHD1 and FSHD2 result in the ectopic expression of the DUX4 gene [209]. Another genetic disorder associated with SMCHD1 variants, but in different locations of the gene, is BAMS [210, 211]. BAMS is a rare disorder with less than 100 reported cases, characterized by absence of the nose, and eye defects.

SMCHD1 variants are associated with two different disorders, FSHD2 and BAMS, yet the impacts of SMCHD1 variants are not well understood. With this work, we therefore sought to better understand the function of SMCHD1 in biological processes, and how this function is impacted by DNA sequence variants in the SMCHD1 gene.

6.3 Summary of results

We analyzed RNA-Seq (section 1.3.3.1), DNA Methylation microarray (section 1.3.2.1), and ChIP-Seq (section 1.3.2.2) data. The omics datasets were generated from samples of primary fibroblast cells (skin-derived), induced pluripotent stem cells, muscle cells and neural crest stem cells, derived from FSHD2 patients, BAMS patients, and healthy controls.

We performed both differential methylation analysis (section 2.4.2), and differential expression analysis (section 2.4.3), comparing FSHD2 and BAMS samples to the control group samples. We identified differentially methylated positions (DMPs), differentially methylated regions (DMRs), and differentially expressed genes (DEGs), at various locations across the genome. We also compared the distributions of chromatin binding peaks (section 1.3.2.2) for FSHD2 and BAMS samples, against the distributions found in the control group samples. We observed a redistribution of peaks for proteins involved in histone modification and chromatin remodeling mechanisms.

We considered the DEGs, as well as the genes associated with the DMRs, and the genes associated with chromatin binding peaks specific to FSHD2 and BAMS samples, to be genes of interest. We used these genes of interest to test for associations between GO biological processes (section 3.2.2), and FSHD2 and BAMS. We identified a number of biological processes, related to development and differentiation, that are associated with either FSHD2 or BAMS, or both, and that thus appear to depend on SMCHD1. More details on these genes of interest, DMRs, binding peaks, and biological processes, can be found in the article (section 6.5).

6.4 My contribution

I contributed to this collaboration by performing differential methylation analyses (section 2.4.2). I performed four different methylation analyses, each consisting of a comparison of methylation levels, for either FSHD2 or BAMS samples, to control group samples, for a particular cell type. This yielded four sets of differentially methylated

positions (DMPs) and differentially methylated regions (DMRs): 1) FSHD2 vs controls, fibroblast cells; 2) BAMS vs controls, fibroblast cells; 3) FSHD2 vs controls, induced pluripotent stem cells; 4) BAMS vs controls, induced pluripotent stem cells.

I also contributed by revisiting annotations of DMRs, in response to a concern that was raised by a peer reviewer during the article review process. This concern, and our subsequent followup, is outlined in the following paragraphs.

In addition to differential methylation testing based on groups of samples, we also tested for differential methylation by separately comparing each FSHD2 and BAMS patient to the control group. The resulting fibroblast cell DMRs were then annotated by comparing their genomic coordinates to the Broad ChromHMM track (section 3.2.1) for epidermal keratinocyte (NHEK) cells. Some DMRs were annotated as occurring in the enhancer regions (section 1.2.4) of genes, and those genes were used for subsequent testing of associations with biological processes. The concern, raised by the peer reviewer, was that the ChromHMM track we used was based on a cell type that was not relevant to DMRs inferred from fibroblast cells.

The original rationale for using the ChromHMM track based on NHEK cells, was that the fibroblast cells we had used for identifying DMRs were skin-derived. There is a ChromHMM track for fibroblast cells, but it is for lung fibroblast (NHLF) cells. To illustrate that the NHEK track was as relevant as the NHLF track, I annotated the DMRs with the NHLF track, using the bedtools intersect tool (`https://bedtools.readthedocs.io/en/latest/content/tools/intersect.html`) on the Galaxy server (`https://usegalaxy.fr/`). We compared the distributions of DMRs annotations (figure 6.1), and observed little difference between the distribution based on the NHEK track, and the distribution based on the NHLF track. Based on this comparison, the published article subsequently retained those results and conclusions that relied on the original NHEK track annotations.

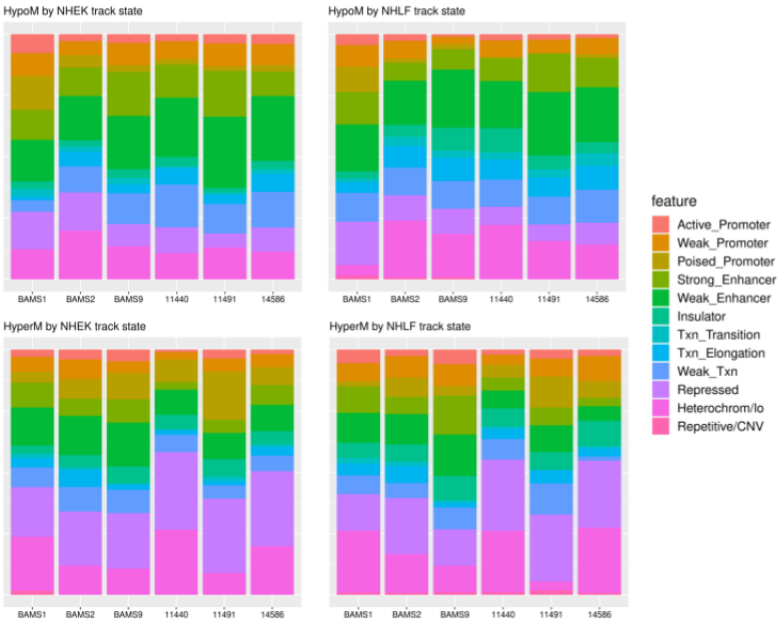


Figure 6.1 – **Distributions of ChromHMM track annotations**. Each bar is the distribution of ChromHMM annotations (feature) for the differentially methylated regions of a particular FSHD2 or BAMS sample. Each plot is for either hypomethylated (HypoM), or hypermethylated (HyperM) differentially methylated regions, annotated with either the NHEK or NHLF ChromHMM track

6.5 Article

Published online 19 June 2023

Nucleic Acids Research, 2023, Vol. 51, No. 14 7269–7287
https://doi.org/10.1093/nar/gkad523

In skeletal muscle and neural crest cells, SMCHD1 regulates biological pathways relevant for Bosma syndrome and facioscapulohumeral dystrophy phenotype

Camille Laberthonnière[1], Mégane Delourme[1], Raphaël Chevalier[1], Camille Dion[1], Benjamin Ganne[1], David Hirst[1], Leslie Caron[1], Pierre Perrin[1], José Adélaïde[2], Max Chaffanet[2], Shifeng Xue[3,4], Karine Nguyen[1,5], Bruno Reversade[4,6,7,8], Jérôme Déjardin[9], Anaïs Baudot[1], Jérôme D. Robin[1] and Frédérique Magdinier[1,*]

[1]Aix Marseille Univ, INSERM, Marseille Medical Genetics, Marseille 13005, France, [2]Aix Marseille Univ, INSERM, CNRS, Institut Paoli Calmette, Centre de Recherche en Cancérologie de Marseille, Laboratory of predictive Oncology, Marseille 13009, France, [3]Department of Biological Sciences, National University of Singapore, Singapore 117558, Singapore, [4]Genome Institute of Singapore, A*STAR, Singapore, Singapore, [5]Département de Génétique Médicale, AP-HM, Hôpital d'enfants de la Timone, Marseille 13005, France, [6]Department of Medical Genetics, Koç University, School of Medicine, Istanbul, Turkey, [7]Department of Physiology, Cardiovascular Disease Translational Research Programme, Yong Loo Lin School of Medicine, National University of Singapore, Singapore, [8]Laboratory of Human Genetics & Therapeutics, Smart-Health Initiative, BESE, KAUST, Thuwal, Saudi Arabia and [9]Institut de Génétique Humaine, UMR 9002, CNRS–Université de Montpellier, Montpellier 34000, France

Received March 24, 2023; Revised May 15, 2023; Editorial Decision May 31, 2023; Accepted June 05, 2023

ABSTRACT

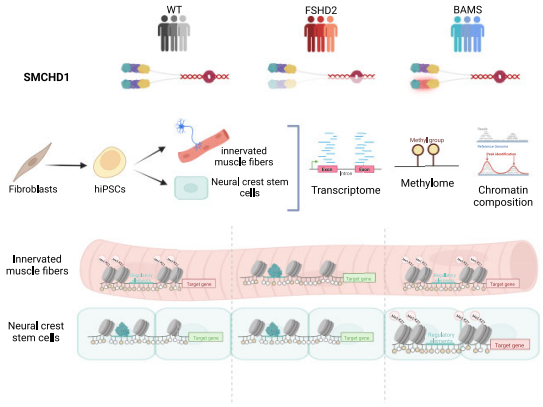
Many genetic syndromes are linked to mutations in genes encoding factors that guide chromatin organization. Among them, several distinct rare genetic diseases are linked to mutations in *SMCHD1* that encodes the structural maintenance of chromosomes flexible hinge domain containing 1 chromatin-associated factor. In humans, its function as well as the impact of its mutations remains poorly defined. To fill this gap, we determined the episignature associated with heterozygous *SMCHD1* variants in primary cells and cell lineages derived from induced pluripotent stem cells for Bosma arhinia and microphthalmia syndrome (BAMS) and type 2 facioscapulohumeral dystrophy (FSHD2). In human tissues, SMCHD1 regulates the distribution of methylated CpGs, H3K27 trimethylation and CTCF at repressed chromatin but also at euchromatin. Based on the exploration of tissues affected either in FSHD or in BAMS, i.e. skeletal muscle fibers and neural crest stem cells, respectively, our results emphasize multiple functions for SMCHD1, in chromatin compaction, chromatin insulation and gene regulation with variable targets or phenotypical outcomes. We concluded that in rare genetic diseases, *SMCHD1* variants impact gene expression in two ways: (i) by changing the chromatin context at a number of euchromatin loci or (ii) by directly regulating some loci encoding master transcription factors required for cell fate determination and tissue differentiation.

*To whom correspondence should be addressed. Tel: +33 4 91 32 49 08; Email: frederique.magdinier@univ-amu.fr

© The Author(s) 2023. Published by Oxford University Press on behalf of Nucleic Acids Research.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

7270 Nucleic Acids Research, 2023, Vol. 51, No. 14

GRAPHICAL ABSTRACT



INTRODUCTION

Based on the presence of an SMC hinge domain, the SMCHD1 (structural maintenance of chromosomes flexible hinge domain containing 1) chromatin-associated factor belongs to the SMC family of chromosomal proteins (1). However, although it forms homodimers by interacting through its SMC hinge domains, SMCHD1 does not participate in the tripartite ring complex formed by other cohesins (2). In mice, Smchd1 is mainly implicated in X inactivation and its depletion causes embryonic lethality with a derepression of genes on the inactive X (Xi) chromosome (3–5). The protein is also involved in the topological conformation of the Xi (6–8) and A/B mega-compartments (6,9–11). Smchd1 plays a role in the regulation of heterochromatin, repetitive DNA sequences or clustered imprinted genes and monoallelically expressed protocadherins (1,3,5,7,12–17). In addition, SMCHD1 binds to telomeres with a direct correlation between telomere length and SMCHD1 enrichment (18,19). However, its precise mechanism of action at telomeres remains partially understood.

Heterozygous germline *SMCHD1* mutations have been reported in at least three distinct rare human genetic diseases, type 2 facioscapulohumeral muscular dystrophy (FSHD2, OMIM #158901) (20–22), Bosma arhinia and microphtalmia syndrome (BAMS, OMIM #603457) (23,24) and isolated hypogonadotrophic hypogonadism with combined pituitary hormone deficiency and septo-optic dysplasia (25). In addition, this gene is often deleted in patients carrying an 18p deletion and presenting symptoms that differ from the three above-mentioned diseases (26,27) or some displaying a typical FSHD phenotype (28). In BAMS, mutations are clustered within exons 3–13, spanning the GHKL-type ATPase domain and an associated region immediately C-terminal to it, while in FSHD2 mutations are dispersed across the whole coding sequence (23,24). In BAMS, many mutations have been associated with a gain of the SMCHD1 ATPase activity (gain-of-function mutation) (23,24,29), while in FSHD2 missense, splice and truncating mutations lead to a loss of function or haploinsufficiency (21,23,29,30).

FSHD is linked in 95% of cases (FSHD1, OMIM #158900) to the shortening of an array of GC-rich 3.3-kb macrosatellite elements (D4Z4) in the 4q arm subtelomeric region (31,32). A smaller proportion of patients (2–3%) carry a mutation in *SMCHD1* (FSHD2) (21). D4Z4 DNA hypomethylation is a cardinal feature of FSHD (20,33–37), but the macrosatellite is also hypomethylated to the same level in BAMS, as well as in 18p11.32 hemizygous patients (23,24,30). Thus, *SMCHD1* haploinsufficiency and loss- or gain-of-function mutations are all associated with D4Z4 hypomethylation with various phenotypical outcomes.

In animal models, *Smchd1* invalidation does not recapitulate the BAMS or FSHD phenotype (23,24). In human tissues, *SMCHD1* heterozygous mutations are not associated with the massive defects in X inactivation observed in mice (30). Thus, how the different heterozygous mutations lead to these diseases affecting the muscle (FSHD) or formation of the nasal placode, migration of surrounding neural crest cells and projection of the gonadotrophin-releasing hormone neurons (BAMS) must be established together with the identification of biological processes (BPs) that depend on SMCHD1.

To tackle this question of the role of SMCHD1 in human tissues, we took advantage of primary cells and induced pluripotent stem cells (hiPSCs) derived from patients affected either with BAMS or with FSHD2. We showed that *SMCHD1* heterozygous mutations impact two major epigenetic determinants, DNA methylation (DNAme) and chromatin architecture, with distinct phenotypical outcomes depending on the tissue. Given SMCHD1 potential implication in the regulation of genes entailed for development and tissue differentiation, we further propose that variegated gene expression triggered by *SMCHD1* variants might contribute to the clinical spectrum of 'SMCHDnopathies'.

MATERIALS AND METHODS

Samples

All individuals had provided written informed consent. The study was done in accordance with the Declaration of Helsinki. Controls were neither carrier of any genetic mutation nor affected by any constitutive pathology and were randomly chosen in the age range and sex representation of patients. All iPSC clones were derived from primary fibroblasts (Supplementary Table S1) (38,39). hiPSCs were treated as described for neuromuscular (40,41) or neural crest stem cell (NCSC) (42) differentiation (Supplementary Table S1).

Cell culture

The human embryonic kidney 293 (HEK293) cell line (CRL-1573) was obtained from ATCC. In HEK293 cells, *SMCHD1* expression was invalidated by transfection of zinc finger nucleases. Cells were provided by J. Déjardin. Primary fibroblasts and HEK293 cells were grown in Dulbecco's modified Eagle's medium (DMEM) with L-alanyl-L-glutamine (GlutaMAX™-I), D-glucose and sodium pyruvate (Life Technologies). Media were supplemented with 10% fetal bovine serum (FBS, Gibco) and 1% penicillin/streptomycin. Cells were grown at 37°C, 5% CO_2,

Nucleic Acids Research, 2023, Vol. 51, No. 14 7271

in a humidified atmosphere. The hiPSCs were differentiated into functional muscles according to the protocol described in (40,41). For chromatin immunoprecipitation sequencing (ChIP-seq) and RNA sequencing (RNA-seq) experiments, cells were collected 30 days post-differentiation. Neural crest differentiation was described in (42). Cells were collected at day 11.

Transfections

Transfections were performed with 250 mM $CaCl_2$ and 2× BBS (50 mM BES, 1.5 mM Na_2HPO_4, 280 mM NaCl). Cells were collected 48–72 h post-transfection. For *SMCHD1* knockdown, 1.5×10^5 cells were cultivated in six-well plates. At 70–80% confluence, we replaced the medium with 500 μl of DMEM + 10% FBS + 0.3 μl of polybrene (6 μg/ml, final concentration). Cells were transduced with 6 μl of pLKO.1-puro-CMV-tGFP (Sigma–Aldrich, St Louis, MO; ref: TRCN0000253776) and 6 μl of pLKO.1-puro-CMV-tGFP (Sigma–Aldrich, St Louis, MO; ref: TRCN0000253778). As scramble, we used 12 μl of MISSION® TRC2 Control Transduction Particle puro TurboGFP (Sigma–Aldrich, St Louis, MO; ref: TRCN0000253776) at a multiplicity of infection of 1. Puromycin (Sigma–Aldrich, St Louis, MO; ref: P8833-25MG) was added 24 h after transduction at a final concentration of 1 μg/ml. Clones were harvested after 15 days of culture for RNA, DNA and protein extraction.

RNA extraction, quality control and library preparation

For high-throughput RNA-seq, total RNA was extracted using the RNeasy kit (Qiagen) following manufacturer's instructions. Quality, quantification and sizing of total RNA were evaluated using the RNA 6000 Pico assay (Agilent Technologies; ref: 5067-1513) on an Agilent 2100 Bioanalyzer system. The RNA integrity number (RIN) was calculated for each sample and only samples with a RIN > 9 were used. RNA-seq libraries were generated from 600 ng of total RNA using TruSeq Stranded mRNA Library Prep Kit and TruSeq RNA Single Indexes kits A and B (Illumina, San Diego, CA), according to manufacturer's instructions. The final cDNA libraries were checked for quality and quantified using capillary electrophoresis. Libraries were sequenced at the IGBMC GenomeEast facility using an Illumina HiSeq 4000 1 × 50 bp. For quantitative polymerase chain reaction (PCR) analysis, total RNA was extracted using TRIzol (Thermo Fisher).

Quantitative reverse transcriptase PCR

Reverse transcription of 1 μg of total RNA was performed using the Superscript IV First-Strand cDNA Synthesis Kit (Life Technologies) using a mix of oligo dT and random hexamers following manufacturer's instructions. Primers were designed using Primer Blast (Supplementary Table S2). Real-time PCR amplification was performed on a QuantStudio 5 Real-Time PCR System (Thermo Fisher) using the SYBR Green Master Mix (Roche). All PCRs were performed using a standardized protocol and data were analyzed with the QuantStudio 5 Real-Time PCR System (Thermo Fisher). For each sample, fold change (FC) was obtained by comparative quantification and normalization to expression of the *GAPDHD*, *HPRT* and *PPIA* housekeeping genes used as standard. Data are expressed as mean ± standard deviation (SD).

DNA extraction and sodium bisulfite sequencing

DNA was extracted from the different types of samples using the NucleoSpin Tissue (Macherey-Nagel) according to manufacturer's instructions.

For sodium bisulfite sequencing, 1 μg of genomic DNA was denatured for 30 min at 37°C in NaOH 0.4N and incubated overnight in a solution of 3 M sodium bisulfite (pH 5) and 10 mM hydroquinone as previously described (43). Primers were designed in order to amplify methylated and unmethylated DNA with the same efficiency using the MethPrimer software avoiding the presence of CpGs in the primer sequence (Supplementary Table S3). After sequencing, sense and antisense sequences were assembled in a single sequence and BAM file converted to fastq file. After trimming of each BSP primer, data were aligned using the BiQ Analyzer HiMod software and processed in R as described in (44). Three methylation scores were calculated: (i) the CpG methylation score of each CpG, (ii) the sequence methylation score, corresponding to the average methylation level of each sequence, and (iii) the global methylation score that corresponds to the global level of methylation for each biological sample calculated as the ratio of methylated CpG with the number of aligned CpGs for all sequences and CpG.

Luciferase assays

The different experimental DNA fragments were synthetized by GenScript or after PCR amplification and inserted into pGL3 luciferase reporter vectors (pGL3 enhancer and pGL3 promoter, Promega) at the polylinker NheI (NEB) restriction site. Extraction and purification were achieved using NucleoBond® Xtra Midi/Maxi Kit (Macherey-Nagel). Orientation and integrity of the insert were verified by Sanger sequencing. For each experiment, pGL3 firefly luciferase experimental vector has been co-transfected with the pGL4.74[hRluc/TK] Renilla luciferase plasmid used as transfection control. Plasmids were mixed in a 10:1 proportion. For transfection, cells were plated 24 h before transfection. Luminescence was measured using the Dual-Glo® Luciferase Assay System (Promega) kit in a GloMax® Explorer Multimode Microplate Reader. Solutions were dispensed through automatic injectors with 2 s delay and 10 s measurement for each reporter. Experiments were realized in triplicates and measures were performed in triplicates for each construct ($n = 9$). Values were normalized to Renilla luciferase levels. Positions of constructs are described in Supplementary Table S4.

Position effect variegation assays

The pCMV-derived plasmid is described in (45). DNA inserts were obtained from GenScript and inserted at the AscI restriction site downstream of the eGFP reporter gene in

7272 Nucleic Acids Research, 2023, Vol. 51, No. 14

pCMV vectors. Details are available upon request. Transfection of the linearized vectors was performed using a modified calcium phosphate method optimized in order to obtain a single integration per cell (46). Three days post-transfection, the hygromycin B selection antibiotic was added to the culture medium (Life Technologies) at a final concentration of 400 μg/ml. Cells were kept under permanent selection for several passages. At different time points, eGFP expression was analyzed using an Accuri flow cytometer and processed using the FlowJo software (Becton-Dickinson). The percentage of eGFP-positive cells was determined using the corresponding nontransfected cells as the baseline for autofluorescence. Mean values were used to compare fluorescence in the different samples.

RNA-seq data processing and differential expression analysis

RNA-seq libraries were generated from 600 ng of total RNA using TruSeq Stranded mRNA Library Prep Kit and TruSeq RNA Single Indexes kits A and B (Illumina, San Diego, CA) from four different FSHD2 samples, three BAMS samples and two controls, according to manufacturer's instructions. Fastq sequence data quality was assessed at the IGBMC facility using FastQC v0.11.5 and reads were trimmed to remove adapter sequences and low-quality bases using DimerRemover v0.9.2. Trimmed single-end reads were aligned using STAR v2.5.3a (6) to the hg38 human genome release. BAM files were sorted and indexed using Sambamba (v0.6.6). Stringtie (v1.3.1c) was used to quantify aligned reads with GENCODE. Differentially expressed genes (DEGs) were identified using the DESeq2 (v1.18.1) R package. Heatmaps were built using the R package pheatmap (v1.0.12) and TPM counts. Overrepresentation enrichment analyses were conducted using the clusterProfiler (v3.14.3) R package or Panther, using DEGs with a false discovery rate (FDR)-adjusted P-value <0.05 and abs($\log_2$FC) > 2 as input.

Infinium MethylationEPIC array

Genome-wide DNAme analysis was performed by Infinium MethylationEPIC array through Diagenode services. Genomic DNA was extracted using the NucleoSpin Tissue Kit (Macherey-Nagel) from two different cell pellets for each sample (BAMS, $n = 3$; FSHD2, $n = 3$; controls, $n = 2$). The analysis was mainly carried out using the ChAMP R package (47). Probes were filtered out if they had missing values, had a detection P-value >0.01 in at least one sample or had a bead count <3 in at least 5% of samples. Samples with $>10\%$ of probes with a detection P-value >0.01 were filtered out. An annotation file that contains information about the location of probes (chromosome, position and nearby genes) (48) was loaded. Probes targeting CpG sites near single-nucleotide polymorphisms, that belong to X/Y chromosomes or that align with multiple locations were filtered out. For each probe, the methylation level of CpGs covered by the array was determined with 99% confidence by calculating the median DNAme β-values and SD within sample groups. Beta values indicate the percentage of CpG sites from a given sample that are methylated (49). Beta values are determined for each CpG location as the relative intensity of methylated and unmethylated signals (50). Probes with an absolute difference (mean $\Delta\beta$-values) of 0.2 and an adjusted P-value <0.05 between sample groups were considered as differentially methylated (50). The ChAMP function for identifying differentially methylated probe (DMPs) uses the limma package (51). An ordinary least squares linear model is fit to the methylation values of each probe to estimate the effect of belonging to a phenotype versus control group. Moderated t-statistics are used to calculate P-values, and the P-values are adjusted with the Benjamini–Hochberg method. The Probe Lasso method was used to identify differentially methylated regions (DMRs) using the champ.DMR function. Probe Lasso selects DMPs (default cutoff P-value $= 0.05$) and extends a lasso in either direction. The size of the lasso depends on the gene feature [e.g. body, intergenomic region (IGR)] and CGI relation (e.g. island, shore) of probes. Mini-DMRs are created when the number of DMPs contained within a lasso is greater than or equal to a minimum value. Overlapping or neighboring mini-DMRs are merged until the distance between merged DMRs is at least 1000 bp. A P-value is calculated for each DMR, using Stouffer's weighted method to combine the P-values of the probes contained in the DMR. The default cutoff P-value for DMRs to be selected is 0.05, and they must contain at least seven probes and exceed 50 bp in width. The Combat method was used to correct for batch effects (52). Heatmaps were built using the pheatmap (v1.0.12) R package using the β-value matrix for methylation. Violin plots were realized by converting β-values to M-value (for better visualization) using the lumi (v2.42.0) R package and then by plotting values using the ggplot2 (v3.3.3) R package. DMPs with an FDR-adjusted P-value <0.05 and abs($\Delta\beta$) > 0.2 were represented as bar plots using the ggplot2 (v3.3.3) R package for genomic features and CGI status according to Illumina Epic array annotation file. Karyoplots of DMPs with an FDR-adjusted P-value <0.05 and abs($\Delta\beta$) > 0.2 were realized using the KaryoplteR (v1.16.0) R package. DMPs were represented as a density measured in bins. For karyoplots, DMPs with FDR < 0.05 and abs($\Delta\beta$) > 0.5 were plotted as dot over chromosomes with the y-axis ranging from 0 to 1.

ChIP and ChIP-seq data processing

Cross-linking of 10–20 $\times$ 10^6 cells was performed as described (38). We used the iDeal ChIP-seq Kit for Transcription Factors and iDeal ChIP-seq Kit for Histones (Diagenode s.a., Seraing, Belgium), as recommended by the manufacturer in hiPSC-derived muscle fibers (MFs) and NCSCs for two FSHD2, two BAMS and controls. For ChIP-qPCR, 5 μg of antibodies against SMCHD1 (Abcam ab31865 and Sigma HPA039441) were mixed. Primers are described in Supplementary Table S5. ChIP-grade antibodies against CTCF, H3K4me3, H3K27Ac or H3K27me3 were used as recommended (Diagenode). ChIP samples were purified using Agencourt AMPure XP beads (Beckman Coulter) and quantified with the Qubit (Invitrogen). ChIP-seq libraries were prepared from 10 ng of double-stranded purified DNA using the MicroPlex Library Preparation Kit v2 (C05010014, Diagenode), according to manufacturer's instructions. Illumina compatible indexes were

Nucleic Acids Research, 2023, Vol. 51, No. 14 7273

added through a PCR amplification (seven cycles). Amplified libraries were purified and size selected using Agencourt AMPure XP beads (Beckman Coulter) to remove unincorporated primers and other reagents. Libraries were sequenced at the IGBMC GenomEast facility using an Illumina HiSeq 4000 1 × 50 bp. Sequence reads were mapped to reference genome hg38 using Bowtie 1.0.0 with the following parameters: -m 1 –strata –best -y -S -l 40 -p 2 and delivered as BAM files along with Wig files generated using an in-house script at IGBMC (variable step, span = 25, reads were elongated to 200 bp). BAM files were analyzed using MACS2 (v2.2.6) using input DNA as control. Peaks were identified using broad peak calling for H3K4me3, H3K27Ac and H3K27me3 and sharp peak calling for CTCF, both with a q-value threshold <0.05. For samples with lower chip signals, we used no model parameter with extsize = 200 bp. Peaks' correspondence between replicates was assessed using MSPC to obtain consensus peaks. Using –B parameter, bedGraph files were generated and whenever possible the replicates were merged and then converted to bigWig files for data visualization. Karyoplots of bigWig files were realized using the KaryoplteR (v1.16.0) R package. Tracks are scaled to the highest value found in the window for each mark. Overrepresentation was determined using the clusterProfiler (v3.14.3) R package. Genes overlapping identified peaks with at least a q-value <0.05 in one replicate were used as input. Identified Gene Ontology (GO) terms in biological process (BP) ontology were selected based on an FDR-adjusted P-value <0.05. Intersection of genomic coordinates from ChIP-seq peaks was realized using Intervene (18).

We also analyzed a publicly available data set for SMCHD1 distribution in HCT116 cells (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1130654). Sites with SMCHD1 enrichment were determined by peak calling and comparison between HCT116 cells and knockout (KO) cells. A .bed genomic coordinate file containing the coordinates of 526 peaks of SMCHD1 from a MACS2 peak calling analysis was used. From this coordinate file, a fasta .fa sequence file using bedtools getfasta was created.

RESULTS

SMCHD1 deficiency results in both increased and decreased DNAme at the genome-wide level

Considering the role of SMCHD1 in DNAme (21,23,24,30), we first asked whether genome-wide methylation profiles might distinguish BAMS from FSHD2 and control cells. To address this question, we performed DNAme profiling in primary fibroblasts at early passages (<10 passages; BAMS, $n = 3$; FSHD2, $n = 3$; controls, $n = 2$; Supplementary Table S1) after sodium bisulfite conversion using the Illumina Infinium HumanMethylation 850 BeadChip (53). Density plot analyses and unsupervised hierarchical clustering indicated an overall overlapping of profiles between samples (Supplementary Figure S1A–D). We identified 29 528 DMPs in BAMS (3% of the probes) and 53 709 DMPs in FSHD2 (6% of the probes), including 12 054 DMPs shared between the two diseases (Supplementary Table S6 and Supplementary File S1). Compared to controls, methylation is globally increased in BAMS cells but decreased in FSHD2 (20% of disease-associated DMPs; Supplementary Figure S1E). Changes occur across all autosomes (Supplementary Figure S2). However, the differences between the two diseases might be explained in part by the discrepancy in the proportion of hypermethylated probes in the different BAMS samples, with a higher proportion of hypermethylated probes in BAMS1, compared to the balanced proportion of hypermethylated and hypomethylated probes in all FSHD2 samples (Figure 1A and B, Supplementary Figure S1E and Supplementary Table S6).

In the two diseases, the distribution of DMPs by CpG content categorized using the Illumina annotations is equivalent. The majority of DMPs correspond to open seas (CpGs outside CpG islands) (Figure 1C and D, Supplementary Figure S1F and Supplementary Table S6), gene bodies and IGRs (Figure 1E and F, Supplementary Figure S1G and Supplementary Table S6). A large proportion of DMPs map to the 6p21.32–6p22.1 locus encompassing the major histocompatibility complex locus and genes encoding chromatin factors and microRNAs (Supplementary Figure S3A).

Next, we searched for DMRs by using the Probe Lasso method (48). When compared to controls, we found between 70 and 263 DMRs in BAMS samples and between 49 and 110 DMRs in FSHD2 samples, distributed across all autosomes (Supplementary Figures S2A and B and S3B–E, and Supplementary File S1). We then selected DMRs shared between BAMS and FSHD2 and sorted hypermethylated and hypomethylated DMRs or DMRs with an opposite methylation status (Supplementary Tables S7 and S8, and Supplementary Figure S3B–E). For prediction of functional elements showing DNAme changes, we analyzed them using the ChromHMM track from NHEK (normal epidermal keratinocytes, Broad Institute; Encode). NHEKs were used as the closest tissue type relative to the patient's arm dermal primary fibroblasts used in our analyses and the tissue with the closest anatomical location. The largest proportion of DMRs corresponds to repressed chromatin/heterochromatin or repressed and poised promoters (Figure 1G–I, and Supplementary Tables S7 and S8). Interestingly, we also noticed a marked hypomethylation at enhancers in patient's cells with 29.23–39.64% of hypomethylated DMRs at enhancers in BAMS and 36.42–47.66% in FSHD2 (Figure 1J, and Supplementary Tables S7 and S8). GO BPs associated with genes in the vicinity of these hypomethylated enhancers encode factors involved in developmental processes (Figure 1K and Supplementary File S1), including genes related to skeletal muscle system development and cell differentiation. Comparisons were also made with adult primary lung fibroblasts and yield comparable results (not shown).

Overall, the high proportion of hypomethylated probes in cells carrying a variant in *SMCHD1* confirms a role for this factor in directing DNAme. On the other hand, increased DNAme at heterochromatin might also suggest that shielding against deposition or spreading of this epigenetic mark might be impaired in patient's cells. Changes of methylation at enhancers in *SMCHD1*-mutated cells further suggest a

7274 Nucleic Acids Research, 2023, Vol. 51, No. 14

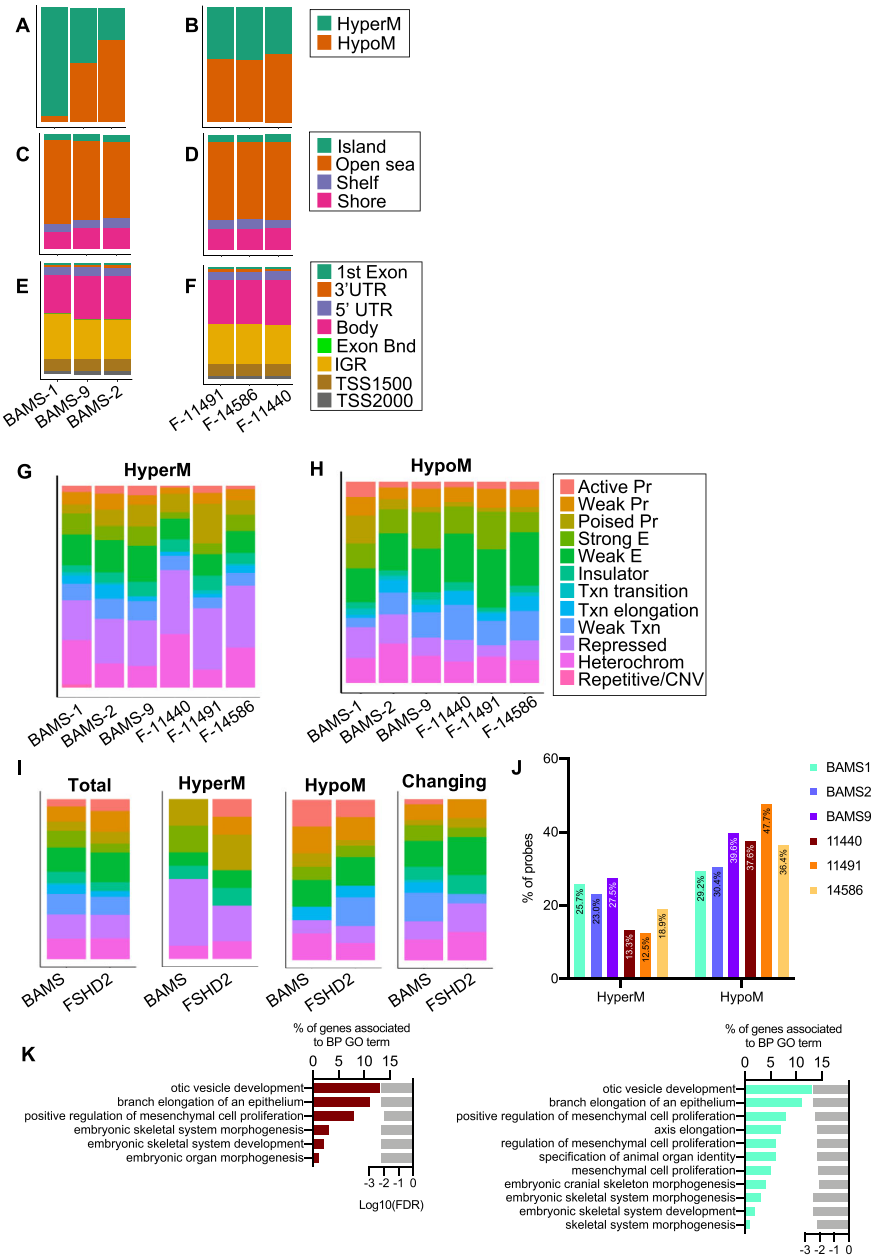


Figure 1. Profiling of DNAme in SMCHD1-deficient cells. One hundred percent stacked bar graphs of the distribution of hypomethylated and hypermethylated probes in individual BAMS samples (**A**) or individual FSHD2 samples (**B**). One hundred percent stacked bar graphs for DMPs by CpG content relative to CpG islands, shores (2 kb flanking CpG islands), shelves (2 kb extending from shores) or open seas (isolated CpGs in the rest of the genome) in individual BAMS samples (**C**) or individual FSHD2 samples (**D**). One hundred percent stacked bar graphs for DMPs by features corresponding to genes' first exon, 3′ UTR, 5′ UTR, gene bodies, exon boundaries, internal genomic regions and probes located 1500 bp from transcription start sites (TSS1500) or 2000 bp from transcription start sites (TSS200) in individual BAMS samples (**E**) or individual FSHD2 samples (**F**). DMRs with an FDR-adjusted *P*-value <0.05 were analyzed for ChromHMM features using NHEK cell annotations and plotted using the ggplot2 (v3.3.3) R package. One hundred percent stacked bar graphs for hypermethylated (**G**) or hypomethylated (**H**) probes are presented. (**I**) Left to right: bar plots of DMRs shared between patients when compared to controls (total)**,** shared hypermethylated DMRs (HyperM), shared hypomethylated DMRs (HypoM) and DMRs with a different methylation profile between patients (changing). (**J**) Graph displaying the percentage of hypermethylated or hypomethylated probes at enhancers in the different BAMS and FSHD2 samples. (**K**) BPs overrepresented for BAMS (cyan, right) or FSHD2 (red, left) DMRs. Bar plots on the left represent the percentage of genes and associated with GO terms listed in the right column. Light gray bars on the right represent the enrichment score ($\log_{10}$FDR) for each GO term.

Nucleic Acids Research, 2023, Vol. 51, No. 14 7275

role for SMCHD1 in the regulation of *cis*-regulatory elements in human tissues.

In pluripotent cells, *SMCHD1* variants impact methylation of a small subset of DNA sequences

After cell reprogramming, methylation profiles overlap between controls and SMCHD1-deficient cells (Supplementary Figure S4A and B, and Supplementary Table S9). Compared to fibroblasts, we observed an opposite distribution in DNAme between the two diseases, with a majority of hypomethylated probes in BAMS but a majority of hypermethylated probes in FSHD2 (Supplementary Figure S4C). As in fibroblasts, most DMPs are located in gene bodies and IGRs (Supplementary Figure S4E), with only 211 DMPs shared between the two diseases (Supplementary Figure S4F). Besides gene bodies and IGRs, we also observed a higher proportion of DMPs in CpG islands in BAMS and a higher proportion in open seas and shelves for FSHD2 hiPSCs (Supplementary Figure S4D). These results indicate a massive reset of the DNAme profile after cell reprogramming with distinct genome-wide impacts depending on the type of *SMCHD1* variants.

Mutations in *SMCHD1* are associated with changes in the expression of a small subset of protein-coding genes

In order to test whether SMCHD1 variants alter gene expression, we searched for genes dysregulated in primary fibroblasts from the same three BAMS and three FSHD2 patients carrying variants in *SMCHD1* (Figure 2A). In comparison to healthy donors (controls), we identified 1134 DEGs in FSHD2 cells and 1507 in BAMS cells, with an overlap of 626 DEGs between the two diseases (FC $\geq$ 2; *P*-value <0.05; Figure 2B, Supplementary Figure S5A and B, and Supplementary File S2). Only a few genes show an opposite trend of expression (up in BAMS and down in FSHD2: *KRT7*, *HAPLN1* or vice versa, *EMB*, *MEOX2* and *LCNL1*). The analysis of BPs for BAMS and FSHD2 DEGs revealed that 11 out of the 15 most significant processes are related to development and body patterning and shared between the two diseases (Figure 2C).

In hiPSCs, we identified a smaller number of DEGs (117 in FSHD2 and 174 in BAMS), with an overlap of 46 DEGs between the two diseases (FC $\geq$ 2, *P*-value <0.05; Figure 2D, Supplementary Figure S5C and D, and Supplementary File S2). After reprogramming, the expression of ∼90% of the DEGs found in fibroblasts is thus reset with only 17 DEGs shared between primary fibroblasts and hiPSCs for FSHD2 (Figure 2E) and 33 genes in BAMS (Figure 2F). Associated BPs correspond to various cellular functions or developmental processes with 3 out of 15 of the most significant pathways common between the two diseases (Figure 2G).

SMCHD1 loss of function, D4Z4 chromatin relaxation and subsequent *DUX4* activation are proposed as FSHD driver mechanisms. In fibroblasts, *DUX4* is undetectable by RNA-seq due to its low expression level. Only a small number of the 422 DUX4 target genes (54) are among DEGs, i.e. 19 genes in BAMS and 11 in FSHD2, including 5 common to both diseases (Supplementary Figure S5E). In hiPSCs, only one DUX4 target (*OAS1*) was identified among DEGs in BAMS (Supplementary Figure S5F).

As described previously, cells carrying *SMCHD1* variants (BAMS or FSHD2) display differential expression of several genes associated with development and cell differentiation but are not distinguishable based on the expression of *DUX4* or of its target genes. As observed for DNAme, gene expression profiles are reset upon reprogramming of primary cells carrying a mutation in *SMCHD1*. Based on this characterization, hiPSCs appear a suitable model for investigating how *SMCHD1* variants might later impact specific tissues during differentiation.

Absence of correlation between methylation changes and *HOX* gene expression

As a comparison to genes regulated by Smchd1 in KO mice (7), we characterized *HOX* loci more carefully. We analyzed changes in DNAme at the four human *HOX* gene clusters and compared their methylation level to *HOX* gene expression. Methylation profiles are comparable in BAMS and FSHD2 cells. We identified several DMPs across all *HOX* loci (Supplementary Figures S6 and S7, and Supplementary File S1) with a majority of hypermethylated probes for *HOXA*, *HOXB* and *HOXD* clusters and a higher proportion of hypomethylated probes for the *HOXC* locus. Twenty-two *HOX* genes are differentially expressed in BAMS and FSHD2 fibroblasts, but we did not observe any obvious correlation between gene expression and methylation changes (Supplementary Figure S6E and F, and Supplementary Files S1 and S2).

For *HOX* genes, we selected four main DMRs with features of polycomb repressed chromatin (*HOXA13/HOTTIP*), poised promoter (*HOXB2*, *HOXB6*) or enhancer (*HOXC4*, *HOXC5*, *HOXC6*) (Supplementary File S1). As not all CGs are covered by the 850k Epic array, we then performed in-depth methylation profiling of these four DMRs by sodium bisulfite sequencing in fibroblasts and hiPSCs. We noted variable DNAme profiles in fibroblasts and hiPSCs. *HOXA13* methylation level is identical between control fibroblasts or hiPSCs and between samples (Supplementary Figure S8A). This DMR is markedly hypomethylated in one of the FSHD2 samples compared to the other FSHD2 and BAMS cells with no apparent correlation with mutations altering SMCHD1 ATPase activity (F-14586, loss of ATPase activity; BAMS1 and BAMS2, gain of ATPase activity). In hiPSCs, the opposite trend is observed, with an increased methylation in cells derived from hypomethylated fibroblasts (F-14586) and an absence of methylation when the DMR is methylated in fibroblasts (F-120521C, BAMS1–BAMS9; Supplementary Figure S8A). The *HOXB2* DMR is unmethylated and remains unmethylated upon reprogramming in the majority of samples (Supplementary Figure S8B). *HOXB6* methylation is low except in BAMS2 fibroblasts [gain of SMCHD1 ATPase activity (29)] but increases after reprogramming in one of the FSHD2 samples [F-14586; loss of ATPase activity (30)] (Supplementary Figure S8C). Interestingly, for *HOXC4*, we observed an absence of methylation in samples carrying a mutation affecting the ATPase activity (F-14586, BAMS1, BAMS2) and an increased

7276 Nucleic Acids Research, 2023, Vol. 51, No. 14

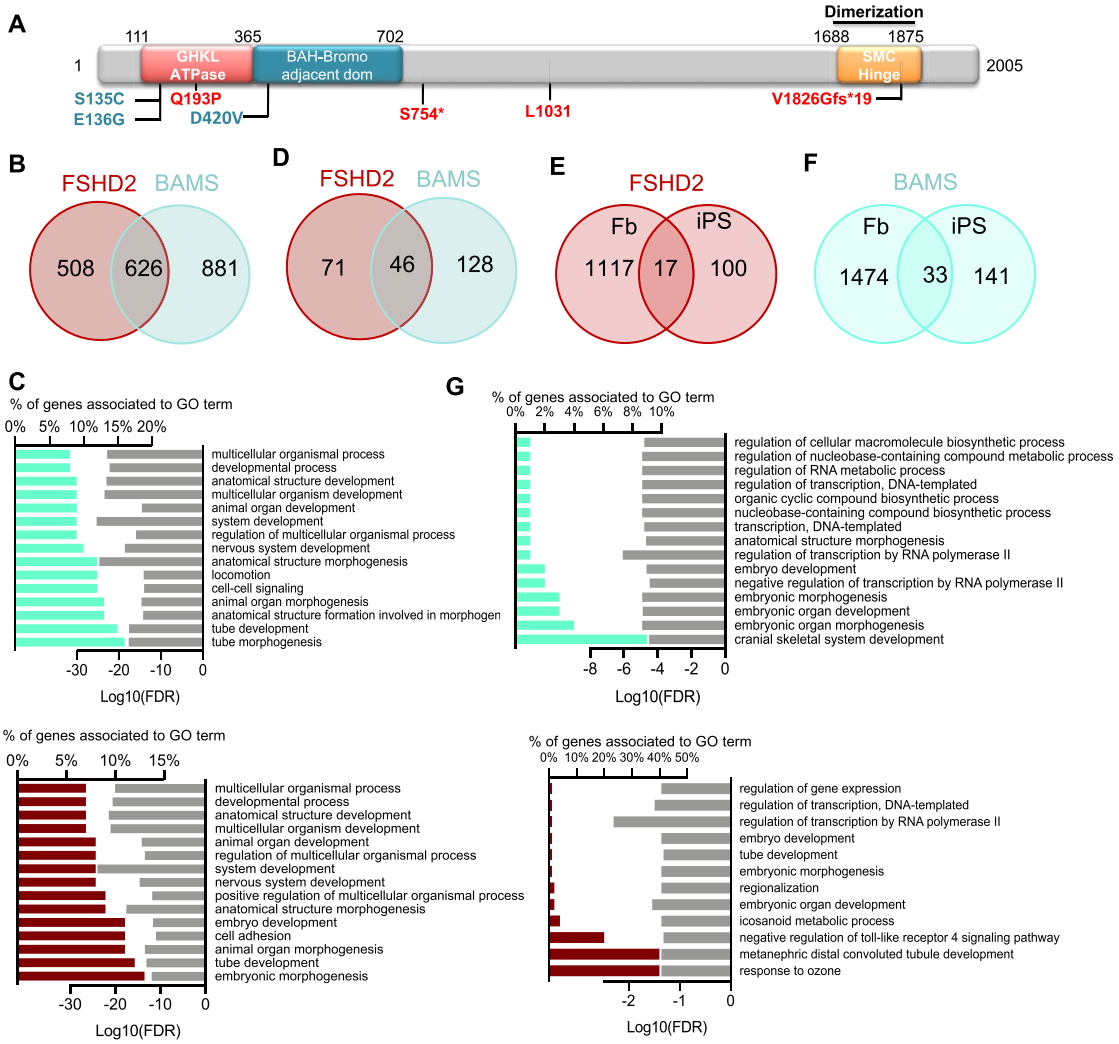


Figure 2. Expression profiling in fibroblasts and iPSCs from patients affected with BAMS or FSHD2. (**A**) Schematic representation of the SMCHD1 protein with position of the different mutations in BAMS (cyan) or FSHD2 (red) patients. (**B**) Venn diagrams for comparison of genes that are differentially expressed in FSHD2 and BAMS primary fibroblasts compared to controls with $-2 > FC > 2$ and FDR < 0.05. (**C**) GO for BPs corresponding to enrichment analysis of DEGs in FSHD2 (red, upper panel) or BAMS (cyan, lower panel) versus control fibroblasts filtered on $-2 > FC > 2$ and FDR < 0.05. Bar plots on the left represent the percentage of genes that are differentially expressed and associated with a GO term shown in the right column. Light gray bars on the right represent the enrichment score ($\log_{10}$FDR) for each BP. (**D**) Venn diagrams for comparison of DEGs in FSHD2 and BAMS hiPSCs compared to controls. (**E**) Venn diagrams for comparison of DEGs between FSHD2 fibroblasts and hiPSCs. (**F**) Venn diagrams for comparison of DEGs between BAMS fibroblasts and hiPSCs. (**G**) GO for BPs corresponding to enrichment analysis of DEGs in FSHD2 (red, upper panel) or BAMS (cyan, lower panel) versus control hiPSCs filtered on $-2 > FC > 2$ and FDR < 0.05. Bar plots on the left represent the percentage of DEGs and associated with GO terms in the right column. Light gray bars on the right represent the enrichment score ($\log_{10}$FDR) for each GO term.

methylation in other patient's fibroblasts (F-120521C, BAMS9). Demethylation of this DMR occurs in all samples after reprogramming (Supplementary Figure S8C). Overall, we observed an absence of correlation between DNAme and gene expression. As in mice, SMCHD1 contributes to the regulation of DNAme profile at autosomal loci in human cells. Nonetheless, as observed for D4Z4 (30), *SMCHD1* germline mutations have different impact on the distribution of methylated CGs.

In vitro or *in vivo* somatic *SMCHD1* invalidation does not modify DNAme

In patient's cells, SMCHD1 binding to the D4Z4 macrosatellite is DNAme independent (30). D4Z4 hypomethylation is a hallmark of patients carrying a germline *SMCHD1* mutation or heterozygosity of the corresponding locus (23,24,30,36). However, as shown in cancer cell lines (30), somatic *SMCHD1* invalidation in human

Nucleic Acids Research, 2023, Vol. 51, No. 14 7277

primary fibroblasts does not cause D4Z4 hypomethylation (Supplementary Figure S9A). To address whether these observations also apply to other genomic sequences, we analyzed DNAme of the four selected *HOX* DMRs (*HOXA13*, *HOXB2*, *HOXB6*, *HOXC4*) in HEK and HEK *SMCHD1* KO cells by sodium bisulfite sequencing. In HEK *SMCHD1* KO cells, somatic *SMCHD1* invalidation and subsequent absence of SMCHD1 binding (Supplementary Figure S9B) does not cause any change in DNAme of *HOX* DMRs (Supplementary Figure S9C). In the absence of SMCHD1, the expression of *HOXB* and *HOXD* genes is variable, while expression of genes in the vicinity of the *HOXA13* and *HOXC4/5/6* DMRs (Supplementary Figure S9D) is increased. As indicated by our analyses in patient's primary cells, this further highlights a possible role for SMCHD1 in the regulation of *HOX* gene expression, independently of DNA hypomethylation, but also shows that *SMCHD1* KO in primary cells does not directly impact *HOX* gene expression, as observed before (55). As shown here and previously *in vitro* and confirmed here *in vivo* in cancer specimens with a somatic loss of heterozygosity of the 18p locus encompassing *SMCHD1* (Supplementary Figure S10), DNAme changes are only triggered by germline *SMCHD1* mutations. This further confirms a role for SMCHD1 in *de novo* DNAme but not in its maintenance in somatic cells.

SMCHD1 variants impact the distribution of H3K27me3 during differentiation

Despite the lack of correlation observed between gene expression and DNAme changes, differentially methylated sites are associated with genes that belong to BPs relevant to either BAMS or FSHD2 respective phenotype and associated with development or differentiation.

After characterization of primary cells and hiPSCs from patients carrying a mutation in *SMCHD1*, we then asked how SMCHD1 controls chromatin organization during differentiation. To this aim, we performed chromatin profiling in hiPSC-derived skeletal MFs and NCSCs (42,56) derived from patient's cells. We mapped the activating H3K4me3, H3K27Ac and repressive H3K27me3 histone marks together with distribution of the CTCF insulating factor by ChIP-seq. Enrichment peaks are distributed along the 22 autosomes for the two diseases in the two cell types (Supplementary Figure S11A and B, and Supplementary Files S3 and S4). Contrarily to DNAme, we did not observe disproportional enrichments at chromosome 6 nor at any other chromosome.

Relative to transcription start sites (TSSs), CTCF, H3K4me3 and H3K27Ac peaks are evenly distributed in diseased MFs or NCSCs compared to controls (Figure 3A and Supplementary Figure S12A). H3K27me3 distribution is more variable (Figure 3A). The majority of H3K27me3 peaks are located at TSSs (<0.1 kb) or at a short distance (0.1–5 kb) from TSSs. In BAMS and FSHD2 cells, most changes in H3K27me3 enrichment occur at TSSs (0.1 and 0.1–1 kb) but also at a distance of 10–50 kb. In patient's samples, we observed an opposite trend between these two regions: an increased H3K27me3 enrichment at TSSs (<0.1 kb) corresponds to a decrease at a distance of 10–50 kb from the TSSs and vice versa, suggesting a redistribution of this chromatin mark in patient's cells.

We then determined chromatin enrichment relative to chromatin features in the different samples using the ChromHMM track for human skeletal muscle myoblast (HSMM) annotation. Consistent with previous analyses, we did not observe major changes for activating chromatin marks (H3K4me3, H3K27Ac) between conditions (Figure 3B, Supplementary Figure S12B, and Supplementary Files S3 and S4). We noticed an increased proportion of CTCF peaks at repressed chromatin and heterochromatin but a decrease at active promoters and insulators. This suggests that SMCHD1 deficiency might be associated with a redistribution of CTCF at heterochromatin to protect against heterochromatin spreading or silencing.

As for DNAme, the H3K27me3 silencing mark is increased at repressed chromatin and poised promoters in several of patient's samples with differences between proliferative (NCSCs) and post-mitotic differentiated cells (MFs) and type of mutation (gain or loss of the ATPase activity) (Figure 3A). We overall concluded that SMCHD1 deficiency does not lead to a massive loss of repressive chromatin marks (H3K27me3, DNAme), as illustrated at *HOX* loci (Figure 4), but rather to a local redistribution of repressive marks, at promoters, repressed chromatin and heterochromatin. Moreover, based on changes in CTCF distribution, we hypothesized that SMCHD1 might shield against the deposition of repressive chromatin marks and that this shielding might be impaired in SMCHD1-deficient cells, with different repercussions compared to Smchd1 KO mouse cells (7).

CTCF binding is increased in the total absence of SMCHD1 but not in SMCHD1-deficient cell lineages

As we observed that *SMCHD1* mutations in human cells have a different impact on chromatin features compared to its absence in mouse tissues regarding CTCF binding, we also analyzed publicly available ChIP-seq data (Gene Expression Omnibus, GSM1130654) obtained in a human SMCHD1-null cell line (HCT116). In these cells, *SMCHD1* has been somatically invalidated using a zinc finger nuclease. After filtering for repetitive DNA sequences, we retrieved 526 SMCHD1-enriched peaks mostly located at a distance of 50–500 kb from TSSs. By artificial extension of these SMCHD1 peak coordinates (±10 kb from the peak), we sorted a list of genes in their vicinity and compared this list to the list of fibroblast DEGs (Supplementary File S1). We identified 24 DEGs with 14/24 harboring a CTCF site in their vicinity. To assess whether SMCHD1 might interfere with CTCF enrichment, we selected three loci harboring insulator features and corresponding to three DEGs (*BET1L*, *NCAM2* and *SEMA5A*) together with one gene close to a putative SMCHD1 peak that does not overlap with CTCF (*WASH7PL*; no associated chromatin feature) (Supplementary Table S4). BET1L and WASH7P are located in heterochromatin regions. Sites located close to the *NCAM2* and *SEMA5A* genes correspond to distal enhancers (>50 kb from TSSs). We analyzed CTCF and SMCHD1 enrichment by ChIP-qPCR at all four sites. In HCT116 KO cells compared to HCT116 cells, the absence of SMCHD1 is

7278 Nucleic Acids Research, 2023, Vol. 51, No. 14

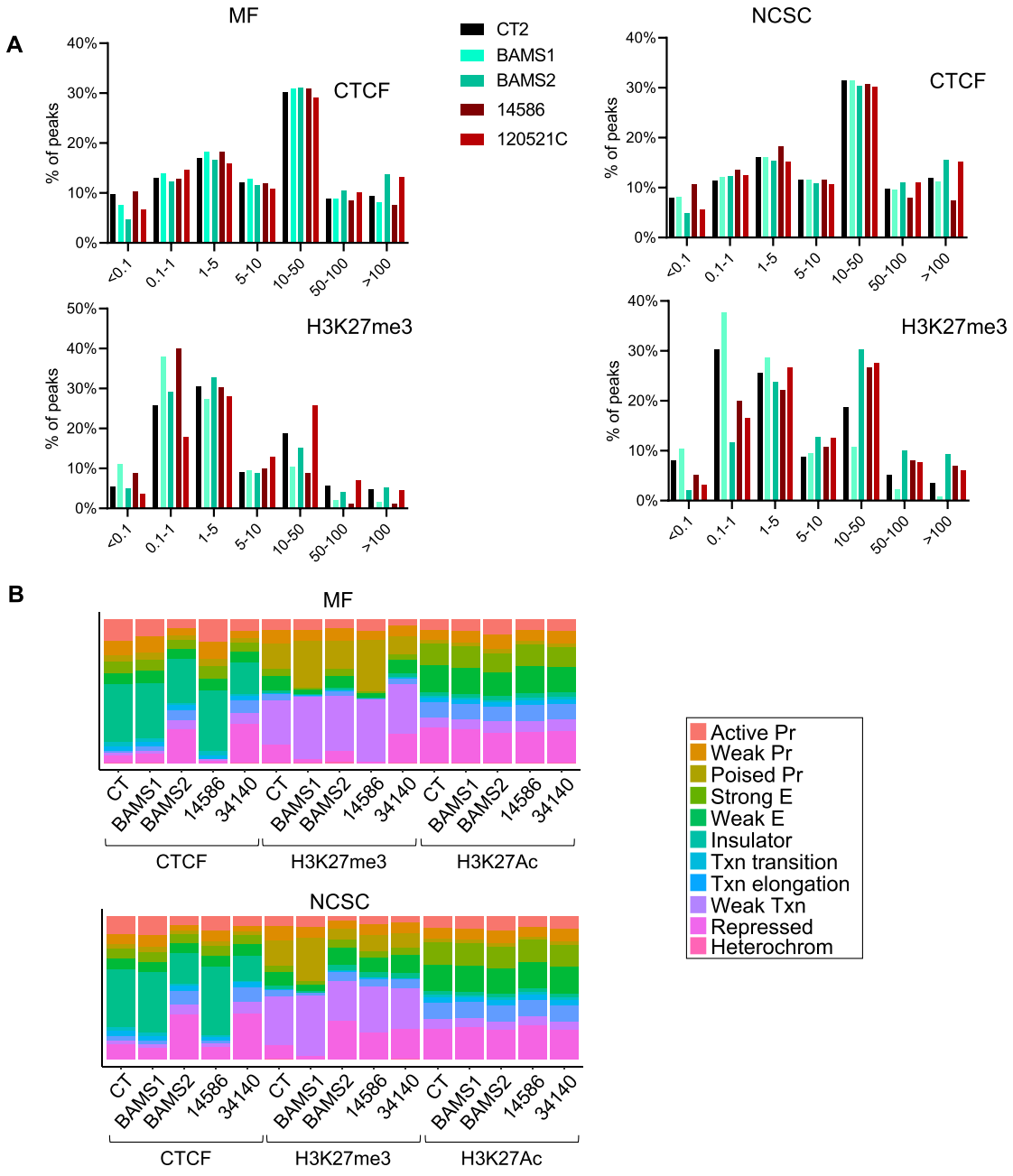


Figure 3. Chromatin profiling of hiPSC-derived MFs and NCSCs from SMCHD1-deficient patients. (**A**) Distribution of distances from TSSs for peaks enriched in CTCF or H3K27me3 in controls, BAMS or FSHD2 MFs or NCSCs. Peaks' distribution relative to TSSs was assessed using the chipenrich (v2.14.0) R package. (**B**) Distribution of peaks enriched in CTCF, H3K27me3 or H3K27Ac relative to chromatin features determined using the ChromHMM track in MFs or NCSCs derived from control, BAMS or FSHD hiPSCs. Peaks with at least a q-value <0.05 in one replicate were analyzed for ChromHMM features using HSMM cell annotations and represented as bar plots using the R ggplot2 (v3.3.3) package.

Nucleic Acids Research, 2023, Vol. 51, No. 14 7279

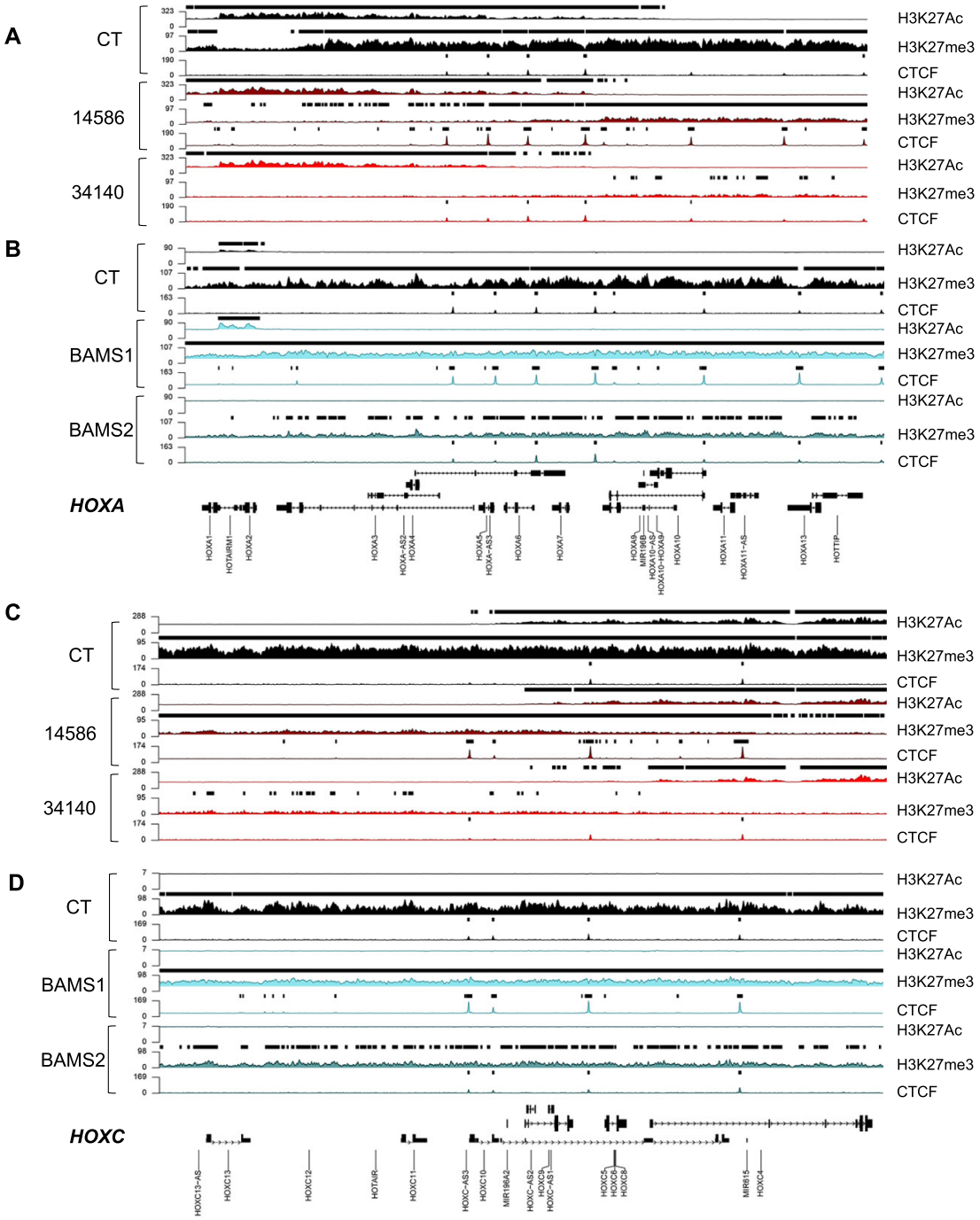


Figure 4. Chromatin profiling of *HOXA* and *HOXC* genes in *SMCHD1*-deficient cells. (**A**) Profiling of the *HOXA* gene locus (chr7:27090339–27211142, 120 803 bp) in controls (black) or FSHD2 (red) MFs for H3K27Ac, H3K27me3 or CTCF enrichment. (**B**) Profiling of the *HOXA* gene locus (chr7:27090339–27211142, 120 803 bp) in controls (black) or BAMS (cyan) NCSCs for H3K27Ac, H3K27me3 or CTCF enrichment. (**C**) Profiling of the *HOXC* gene locus (chr12:53930562–54058000; 127 438 bp) in controls (black) or FSHD2 (red) MFs for H3K27Ac, H3K27me3 or CTCF enrichment. (**D**) Profiling of the *HOXC* gene locus (chr12:53930562–54058000; 127 438 bp) in controls (black) or BAMS (cyan) NCSCs for H3K27Ac, H3K27me3 or CTCF enrichment.

7280 Nucleic Acids Research, 2023, Vol. 51, No. 14

associated with a significant increase in CTCF occupancy (Supplementary Figure S13, *P*-value <0.00001). We concluded that, as in the mouse (7), the total absence of SMCHD1 impacts CTCF distribution in human cells. Based on these results and those obtained in *SMCHD1*-mutated cells (BAMS and FSHD2 cells), we also concluded that depending on the genomic functional features, SMCHD1 contributes to CTCF distribution in human cells.

SMCHD1 regulates gene expression *in cis*

Having evidenced a role for SMCHD1 in the distribution of CTCF or H3K27me3, we then tested functionally whether different sequences might regulate gene expression and insulation in an SMCHD1-dependent manner. To this aim, we performed transient and stable reporter gene expression assays in HEK and HEK *SMCHD1* KO cells. Transient reporter assays were used to evaluate enhancer or promoter activity. Stable insertions of single-copy transgenes are used to reveal protection against position effects and insulation (45).

For D4Z4, we selected subregions that are bound by SMCHD1 (30) and that display a differential methylation profile in patients (37,44) on transient luciferase expression (Supplementary Figure S14A). In a vector lacking the SV40 enhancer (pGL3 promoter), the proximal part of D4Z4 differentially methylated in *SMCHD1* KO cells (DR1 and DR1 subfragments, up to CG 21) activates luciferase expression in an SMCHD1-dependent manner (Figure 5A and B). This sequence has no activity in a promoterless reporter (Supplementary Figure S14B). This suggests that the DR1 fragment acts as an SMCHD1-dependent *cis*-activator. For the different *HOX* DMRs bound by SMCHD1 in HEK cells, we did not observe any enhancer activity (Figure 5C). However, in the absence of SMCHD1, luciferase expression increases for *HOXB2*, *HOXB6* and *HOXC4* DMRs (Figure 5C). This shows that SMCHD1 negatively regulates gene expression via these DMRs. We made the same observation for the region near *SEMA5A* identified as a potential SMCHD1 binding site but not for *NCAM2* or *BET1L* (Figure 5D). Using these transient transfection assays, we concluded that SMCHD1 might contribute to both transcriptional activation and silencing depending on its target sequence.

SMCHD1 protects against position effect variegation in human tissues

Smchd1 has been initially identified as a suppressor of position effect variegation (PEV) (1). PEV is defined as the variable expression of genes due to the stochastic establishment of an epigenetic state modifying their regulatory region. We tested the impact of SMCHD1 on PEV in human cells by monitoring reporter gene expression after stable integration of a single copy of our reporter system (45).

The D4Z4 insulator sequence was first used as a reference (Figure 5E). As indicated by the increased proportion of eGFP-positive cells in D4Z4-containing constructs (Figure 5E, blue shading), D4Z4 protects against PEV in HEK cells. This D4Z4-dependent protection against PEV strongly increases when *SMCHD*1 is re-expressed in HEK KO cells (Figure 5E, gray shading). This shows that SMCHD1 contributes to the protection against PEV mediated by D4Z4. Regarding D4Z4 subfragments, DR1 harbors a moderate anti-PEV that is not improved by *SMCHD1* overexpression. The 5P region located downstream of DR1 or the DR1–5P combination increases the percentage of cells expressing the eGFP reporter. This reveals that protection against PEV is mediated by the proximal part of the D4Z4 repeat and depends on SMCHD1. These results confirm that the D4Z4 proximal part that is hypomethylated in patients displays an SMCHD1-dependent enhancer–insulator activity.

Regarding other loci, the *HOXA13* DMR has no effect on the eGFP reporter expression as in luciferase assays (Figure 5F). By increasing the percentage of eGFP-positive cells, *HOXB2* DMR acts as an activating element partially dependent on SMCHD1 (KO cells, orange shading, Supplementary Figure S14D). The insertion of *HOXB6* (Supplementary Figure S14D) and *HOXC4/C5/C6* (Figure 5F) DMRs slightly increases the percentage of eGFP-positive cells. This percentage is augmented in the absence of SMCHD1 indicating that SMCHD1 negatively regulates *HOXB6* and *HOXC4/C5/C6* DMRs as observed in transient luciferase assays. Regarding the sites identified in HCT116/HCT116 KO cells, similar results were obtained for *NCAM2*, *BET1L* and *WASH7PL* with a marked increase in eGFP-positive cells in HEK KO cells. This marked increase is counterbalanced upon *SMCHD1* re-expression (Supplementary Figure S14D). We concluded that these regions repress gene expression in an SMCHD1-dependent manner, while the effect on gene expression of the site located at the *SEMA5A* locus is not mediated by SMCHD1.

Overall, these results highlight versatile functions for SMCHD1 as a *cis*-activator (D4Z4–DR1, *SEMA5A*), or as a moderate (*HOXB2*, *NCAM2*, *WASH7PL*) to strong (*HOXA13*, *HOXB6*, *HOXC4/C5/C6*, *BET1L*) repressor of gene expression.

SMCHD1 regulates BPs relevant to BAMS and FSHD2 phenotype in patient's derived cells

Given the different roles attributable to SMCHD1 and the identification of changes in the expression of genes associated with BPs relevant to BAMS or FSHD2 respective phenotype in hiPSC-derived NCSCs and MFs (42,56), we then analyzed the top 15 BPs associated with peaks found in patient's cells for the different chromatin marks (Figure 6). We observed similar BPs for H3K4me3 in BAMS and FSHD2 MFs or NCSCs and H3K27me3 peaks in NCSCs. Most BPs are associated with RNA processing and telomere or chromatin regulation (Supplementary Figure S15A–D). Compared to NCSCs, the top 15 BPs are different in BAMS and FSHD2 MFs with many BPs related to development and differentiation (Supplementary Figure S15A and B). In BAMS MFs or NCSCs, CTCF peaks mostly correspond to BPs related to cell migration and motility or tissue differentiation (Figure 6A–D). In FSHD2 MFs and NCSCs, CTCF peaks are largely associated with BPs related to neuromuscular processes (striated muscle contraction or structure, actin filament processes) (Figure 6A–D). H3K27me3 peaks in BAMS and FSHD2 MFs and NCSCs correspond

Nucleic Acids Research, 2023, Vol. 51, No. 14 7281

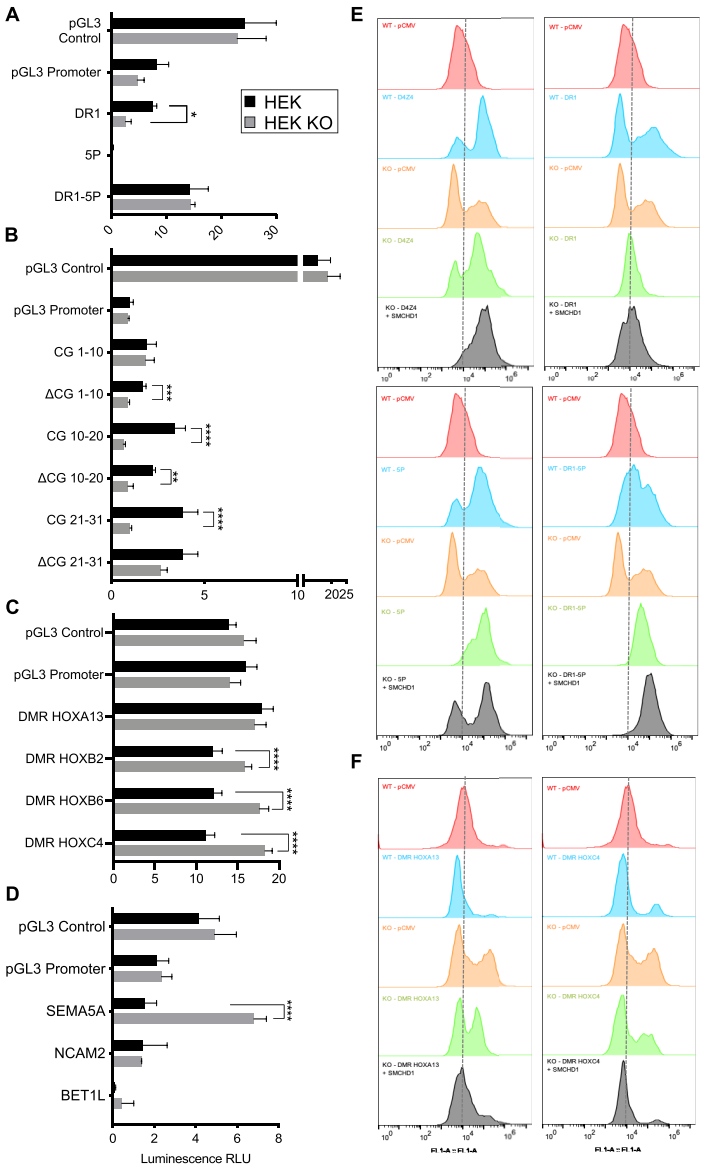


Figure 5. Depending on the genomic context, SMCHD1 contributes to gene silencing or protects against position effects. For the different pGL3 constructs, fragments corresponding to the regions that are differentially methylated (**A**–**C**) or corresponding to SMCHD1 binding sites (**D**) were cloned downstream of the luciferase reporter gene in vectors lacking an enhancer (pGL3 promoter). Firefly luciferase expression was determined 48 h post-transfection of the different constructs in HEK293 and HEK *SMCHD1* KO cells. The pGL3 control vector was used as a transfection control. Firefly luciferase levels were normalized to expression of the Renilla luciferase used as transfection control. Values corresponding to the normalized luciferase activity (expressed in relative luminescence units, RLUs) are the average of three independent assays, each realized as technical triplicates ($n = 9$). Error bars represent standard error. Statistical significance was determined using a Mann–Whitney test (*****P*-value <0.00001, ****P*-value <0.0001, ***P*-value <0.001, **P*-value = 0.01). (A) For the D4Z4 macrosatellite, regions that are differentially methylated in patients carrying a mutation in *SMCHD1* were tested (DR1, 5P; a scheme of the D4Z4 repeat is presented in Supplementary Figure S13). (B) The DR1 sequence contains 31 CG sites. Fragments corresponding to CG1–10, CG10–20 and CG21–31 or deleted of 10 of these CGs (ΔCG1–10, ΔCG10–20 and ΔCG21–31) were tested. (C) H*OX* gene DMRs. (D) Putative SMCHD1 binding sites overlapping or not with CTCF binding sites. (**E**, **F**) For evaluation of protection against position effect, we used a vector carrying a hygromycin resistance gene and an *eGFP* reporter gene. Sequences to be tested are cloned downstream of the *eGFP* gene and transfected into HEK or HEK KO cells. Stable eGFP expression was measured by flow cytometry (FACS) for an extended period of time in cells grown in the presence of hygromycin B. Representative spectra of the % of eGFP-positive cells are presented. For each condition, eGFP expression was compared to values obtained in cells transfected with the empty vector (pCMV). In HEK *SMCHD1* KO cells, eGFP expression was also measured 72 h after transfection of an *SMCHD1* expression vector (gray curves). (E) D4Z4 (left upper panel), DR1–5P (right upper panel), DR1 (left lower panel) and 5P (right lower panel). (F) Results obtained for the HOXA13 (left) and HOXC4/5/6 (right) DMRs.

7282 Nucleic Acids Research, 2023, Vol. 51, No. 14

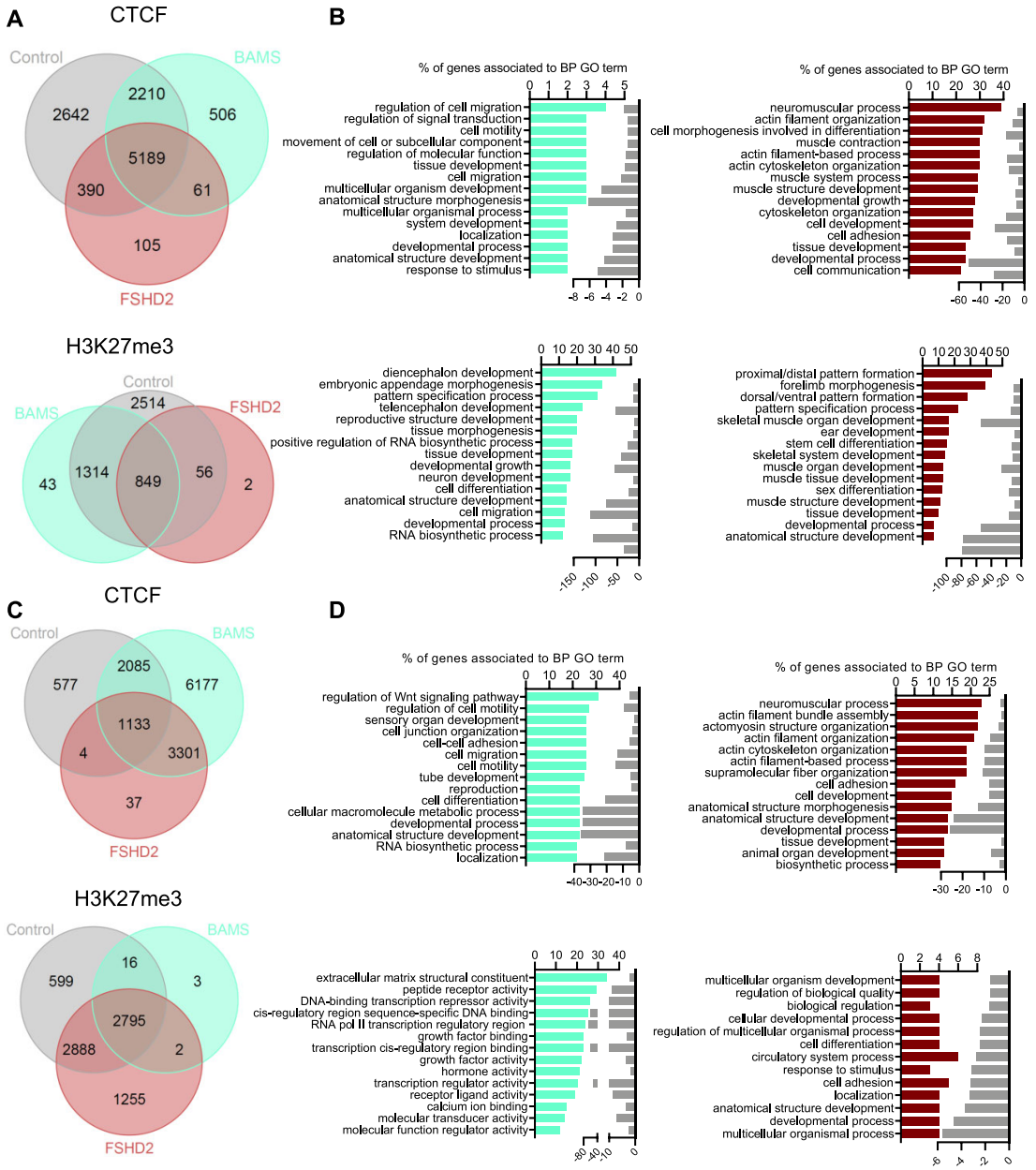


Figure 6. Chromatin profiling in SMCHD1-deficient cells. (**A**) Venn diagrams for comparison of peaks enriched in controls, BAMS and FSHD2 hiPSC-derived MFs for CTCF or H3K27me3. (**B**) GO terms for BPs corresponding to peaks enriched in CTCF or H3K27me3 in BAMS cells (cyan) or FSHD2 (red) MFs. Bars correspond to the number of genes corresponding to the different BPs. Light gray bars on the right represent the enrichment score ($\log_{10}$FDR) for each GO term. (**C**) Venn diagrams for comparison of peaks enriched in controls, BAMS and FSHD2 hiPSC-derived NCSCs for CTCF or H3K27me3. (**D**) GO terms for BPs corresponding to peaks enriched in CTCF or H3K27me3 in NCSCs in BAMS (cyan) or FSHD2 (red) cells. Bars correspond to the number of genes corresponding to the different BPs. Light gray bars on the right represent the enrichment score ($\log_{10}$FDR) for each GO term.

Nucleic Acids Research, 2023, Vol. 51, No. 14 7283

to loci encoding genes involved in developmental processes, tissue morphogenesis, pattern specification and cell migration (Figure 6A–D).

Noticeably, in FSHD2 MFs, the top 15 H3K27me3 and CTCF peaks are found at loci encoding genes involved in skeletal muscle or neuromuscular function, muscle contraction and actin filament-based processes, all consistent with the disease's clinical manifestations.

SMCHD1 regulates tissue-specific differentiation pathways

Giving the striking overlap between BPs associated with DEGs (42,56) and chromatin profiling in hiPSC-derived cells, we analyzed more carefully the overlap between CTCF or H3K27me3 peaks and DEGs. We specifically focused on MFs in FSHD2 cells and NCSCs in BAMS cells (Figure 7 and Supplementary File S5).

In BAMS NCSCs, BPs corresponding to the overlap between DEGs and CTCF- or H3K27me3-enriched peaks are consistent with those identified by RNA-seq in this tissue (42). They include developmental processes and embryonic development, head, eye and sensory organ development, or extracellular matrix organization that are relevant to the clinical manifestations that characterize this developmental syndrome (Figure 7A). Among H3K27me3 peaks that are specific to BAMS NCSCs, we noticed peaks corresponding to *HOX* genes (*HOXB1*, *HOXB2*, *HOXC4*, *HOXA3*), all associated with pattern specialization processes and development.

In FSHD2 MFs, BPs correspond to genes associated with developmental and differentiation processes. Among them, several BPs include transcription factors involved in differentiation pathways. In FSHD2 MFs, several DEGs associated with changes in H3K27me3 peaks are also associated with skeletal muscle tissue development, muscle cell fate commitment and striated muscle differentiation (Figure 7B). Furthermore, by considering GO, KEGG or Reactome, we noticed a significant overrepresentation of genes related to calcium (calcium signaling, calcium ion binding, calcium ion transmembrane transport) in cells from FSHD2 patients carrying a mutation in *SMCHD1*, compared to controls.

DISCUSSION

Originally identified as a regulator of variegation in the mouse (1), Smchd1 is required for proper X inactivation and regulates large gene clusters such as *Hox* or *Pcdh* loci and imprinted gene chromatin in this species (3,4,6–8,13–16). At the two-cell stage, Smchd1 shields against active Tet-dependent DNA demethylation (57) and plays a predominant role in autosomal imprinting through a maternal effect (13,58). However, the role of this protein in human tissues remains only partially understood despite its implication in several distinct rare genetic diseases (21,23–25). In mice, Smchd1 KO is fully penetrant. In BAMS patients, the majority of *SMCHD1* mutations cause a gain of function of the ATPase activity (29). Mutations in FSHD2 act on protein dosage or cause a loss of function (29). Importantly, key differences between mouse and human are the absence of BAMS and FSHD2-like phenotype in mouse models and the lack of a clear evidence for X inactivation defect in BAMS and FSHD2 patients (30), as both sexes are equally affected.

To define SMCHD1 role in the epigenetic regulation of the human genome and its implication in FSHD and BAMS, we performed genome-wide DNAme and chromatin profiling in cells from patients carrying a heterozygous mutation in this gene. By focusing on the role of this protein in the regulation of autosomes, we show here that SMCHD1 shapes chromatin at different levels depending on functional genomic features. As in the mouse, SMCHD1 contributes to *de novo* DNAme but is dispensable for DNAme maintenance in human cells. Its occupancy is not correlated with the level of DNAme (8,30,57). In human cells, SMCHD1 is involved in the distribution of repressive marks (DNAme, K3K27me3) at heterochromatin. We further show that SMCHD1 might also shield against the deposition of H3K27me3 suggesting a role in chromatin insulation. Importantly, SMCHD1 also regulates DNAme and H3K27me3 deposition at euchromatin loci, orchestrating the expression of genes involved in development and differentiation circuits.

Our results further suggest that SMCHD1 might impact gene expression in two ways: (i) by acting on CTCF and H3K27me3 enrichment, changing the chromatin context at a number of loci, or (ii) by regulating loci encoding master transcription factors required for cell fate determination and differentiation. In addition, SMCHD1 contributes to chromatin insulation by protecting against repressive chromatin and contributes to D4Z4 boundary activity (45,59). We further uncovered here that in human cells, multiple roles are attributable to SMCHD1, either as a *cis*-activator (D4Z4–DR1, *SEMA5A*) or as a moderate (*HOXB2*, *NCAM2*, *WASH7PL*) to strong (*HOXA13*, *HOXB6*, *HOXC4/C5/C6*, *BET1L*) repressor of gene expression. As Smchd1 protects against Tet-dependent demethylation (57), it remains to be determined whether the increased proportion of hypomethylated CpGs in BAMS and FSHD2 cells is due to SMCHD1 inability to counteract active demethylation or to target *de novo* methylation during cell fate transition and cell lineage specification.

Autosomal dominant *SMCHD1* mutations impact stem cell properties as well as postnatal health (60). The diverse phenotypes of *SMCHD1*-mutated patients and animal models might be the consequence of pleiotropic effects of the protein. These pleiotropic effects might be triggered by dosage-dependent or functional chromatin state changes at regulatory regions with cell-to-cell differences. In human cells, these stochastic changes might contribute to the variegated expression of gene required for development and differentiation in space and time. This variegation in expression might, in particular, concern master transcription factors required for cell differentiation as evidenced by comparing H3K27me3 and CTCF profiles. Changes in the epigenetic signature of several predicted SMCHD1-dependent targets, including gene promoters and enhancers, might induce promiscuous or precocious activation/repression of genes controlled or located at distance from these regulatory elements.

In patients carrying a heterozygous *SMCHD1* mutation, SMCHD1 deficiency is less drastic than in KO mice that

7284 Nucleic Acids Research, 2023, Vol. 51, No. 14

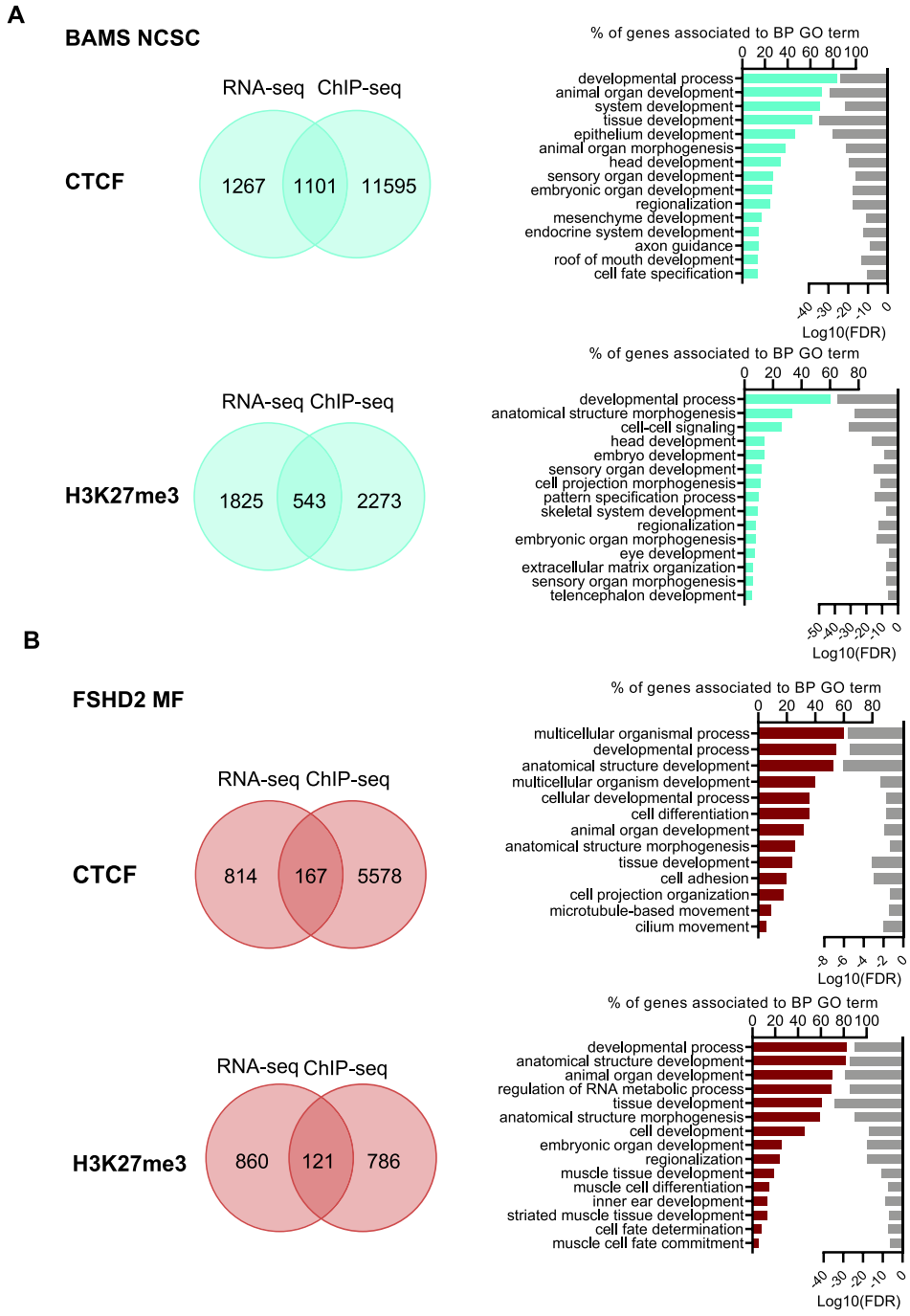


Figure 7. Comparison between chromatin peaks and DEGs in NCSCs and MFs from BAMS and FSHD2 patients. (**A**) Venn diagrams (left) and GO terms for BPs (right) of peaks specifically enriched in CTCF or H3K27me3 in BAMS NCSCs (cyan). (**B**) Venn diagrams (left) and GO terms for BPs (right) of peaks specifically enriched in CTCF or H3K27me3 in FSHD2 MFs (red). For GO terms, bars correspond to the number of genes corresponding to the different BPs. Light gray bars on the right represent the enrichment score ($\log_{10}$FDR) for each GO term.

Nucleic Acids Research, 2023, Vol. 51, No. 14 7285

die *in utero*. However, *SMCHD1* mutations impact NCSCs and skeletal muscle differentiation. Based on our results from patient's cells, we hypothesize that titration of functional SMCHD1 homodimers alters DNA or H3K27 methylation and CTCF-dependent chromatin organization at various loci. Several of these loci are associated with striated muscle and NCSC differentiation or function. These changes might in turn modify the expression of genes implicated in development, cell fate transition and tissue differentiation in rare genetic diseases linked to heterozygous *SMCHD1* mutations. The specific molecular mechanisms by which DNAme or H3K27me3 regulates the selection of CTCF sites involved in cell differentiation, among the thousands present, are still not fully understood (61,62). We now need to understand how SMCHD1 regulates DNAme or H3K27me3 deposition and how SMCHD1 and CTCF might cooperate in guiding the organization and topology of the human genome.

Our study, along with previous research in animal models (6,55), confirms the critical role of SMCHD1 in shaping the mammalian genome and sheds light on the consequences of *SMCHD1* mutations in rare genetic diseases. Our findings demonstrate the link between SMCHD1 function and cell differentiation and development, providing new avenues for investigating diseases associated with mutations in this gene. Our results emphasize the importance of characterizing SMCHD1's functional domains to better understand the effects of germline mutations on genetic diagnostics and therapeutic interventions. Additionally, our gene expression and chromatin profiling data consistently demonstrate that hiPSC-derived innervated fibers from patient cells provide an accurate model for studying functional muscle defects in cells from FSHD patients.

DATA AVAILABILITY

Raw data were deposited at the NCBI Gene Expression Omnibus (https://www.ncbi.nlm.nih.gov/geo/) with the following accession numbers.

Raw RNA-seq data and raw count matrix: GSE174604.

DNAme data: GSE175527.

Raw ChIP-seq data in MFs and raw count matrix record: GSE174769.

Raw ChIP-seq data in NCSCs and raw count matrix record: GSE179986.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We are indebted and thank all patients for participating in this study. We are thankful to Drs Claire Francastel and Stéphane Zaffran for critical comments on the work and manuscript. We acknowledge the GBiM genomics and bioinformatics core facility for RNA-seq and the Marseille Stem Cells core facility for stem cells. The graphical abstract was made using BioRender (agreement number K125AM4BGM). B.R. is an investigator of the National Research Foundation (Singapore) and Branco Weiss Foundation (Switzerland) and an EMBO Young Investigator.

Authors' contributions: C.L. conducted most of the experiments, analyzed the data and contributed to the writing of the manuscript. M.D., C.D., B.G., P.P. and L.C. conducted part of the experiments. R.C., D.H. and A.B. performed or supervised omics analysis. J.D. provided key reagents and edited the manuscript. J.A., M.C., K.N., S.X. and B.R. provided patients' samples and clinical data and edited the manuscript. A.B. supervised RNA-seq and DNAme data analysis and edited the manuscript. J.R.D. provided intellectual input and edited the manuscript. F.M. designed the study, obtained funding, analyzed the data, supervised the project, and wrote and edited the manuscript.

FUNDING

Association Française contre les Myopathies; Fondation Maladies Rares; Agence Nationale pour la Recherche [ANR-21-CE45-0001-01]; Ministry of Education, France (to C.D., C.L. and M.D.); FSH Society (to C.D); Aix-Marseille University [AMX-19-IET-007 to A.B.]; Agency for Science, Technology and Research (to B.R.); Ministry of Education, Singapore [MOE-T2EP30122-0005 to S.X.]; National University of Singapore (to S.X.). Funding for open access charge: Aix-Marseille University.

Conflict of interest statement. None declared.

REFERENCES

1. Blewitt,M.E., Vickaryous,N.K., Hemley,S.J., Ashe,A., Bruxner,T.J., Preis,J.I., Arkell,R. and Whitelaw,E. (2005) An *N*-ethyl-*N*-nitrosourea screen for genes involved in variegation in the mouse. *Proc. Natl Acad. Sci. U.S.A.*, **102**, 7629–7634.
2. Chen,K., Czabotar,P.E., Blewitt,M.E. and Murphy,J.M. (2016) The hinge domain of the epigenetic repressor Smchd1 adopts an unconventional homodimeric configuration. *Biochem. J.*, **473**, 733–742.
3. Blewitt,M.E., Gendrel,A.V., Pang,Z., Sparrow,D.B., Whitelaw,N., Craig,J.M., Apedaile,A., Hilton,D.J., Dunwoodie,S.L., Brockdorff,N. *et al.* (2008) SmcHD1, containing a structural-maintenance-of-chromosomes hinge domain, has a critical role in X inactivation. *Nat. Genet.*, **40**, 663–669.
4. Gendrel,A.V., Apedaile,A., Coker,H., Termanis,A., Zvetkova,I., Godwin,J., Tang,Y.A., Huntley,D., Montana,G., Taylor,S. *et al.* (2012) Smchd1-dependent and -independent pathways determine developmental dynamics of CpG island methylation on the inactive X chromosome. *Dev. Cell*, **23**, 265–279.
5. Gendrel,A.V., Tang,Y.A., Suzuki,M., Godwin,J., Nesterova,T.B., Greally,J.M., Heard,E. and Brockdorff,N. (2013) Epigenetic functions of Smchd1 repress gene clusters on the inactive X chromosome and on autosomes. *Mol. Cell. Biol.*, **33**, 3150–3165.
6. Wang,C.Y., Jegu,T., Chu,H.P., Oh,H.J. and Lee,J.T. (2018) SMCHD1 merges chromosome compartments and assists formation of super-structures on the inactive X. *Cell*, **174**, 406–421.
7. Jansz,N., Keniry,A., Trussart,M., Bildsoe,H., Beck,T., Tonks,I.D., Mould,A.W., Hickey,P., Breslin,K., Iminitoff,M. *et al.* (2018) Smchd1 regulates long-range chromatin interactions on the inactive X chromosome and at Hox clusters. *Nat. Struct. Mol. Biol.*, **25**, 766–777.
8. Gdula,M.R., Nesterova,T.B., Pintacuda,G., Godwin,J., Zhan,Y., Ozadam,H., McClellan,M., Moralli,D., Krueger,F., Green,C.M. *et al.* (2019) The non-canonical SMC protein SmcHD1 antagonises TAD formation and compartmentalisation on the inactive X chromosome. *Nat. Commun.*, **10**, 30.
9. Giorgetti,L., Lajoie,B.R., Carter,A.C., Attia,M., Zhan,Y., Xu,J., Chen,C.J., Kaplan,N., Chang,H.Y., Heard,E. *et al.* (2016) Structural organization of the inactive X chromosome in the mouse. *Nature*, **535**, 575–579.

10. Deng,X., Ma,W., Ramani,V., Hill,A., Yang,F., Ay,F., Berletch,J.B., Blau,C.A., Shendure,J., Duan,Z. *et al.* (2015) Bipartite structure of the inactive mouse X chromosome. *Genome Biol.*, **16**, 152.
11. Darrow,E.M., Huntley,M.H., Dudchenko,O., Stamenova,E.K., Durand,N.C., Sun,Z., Huang,S.C., Sanborn,A.L., Machol,I., Shamim,M. *et al.* (2016) Deletion of DXZ4 on the human inactive X chromosome alters higher-order genome architecture. *Proc. Natl Acad. Sci. U.S.A.*, **113**, E4504–E4512.
12. Mould,A.W., Pang,Z., Pakusch,M., Tonks,I.D., Stark,M., Carrie,D., Mukhopadhyay,P., Seidel,A., Ellis,J.J., Deakin,J. *et al.* (2013) Smchd1 regulates a subset of autosomal genes subject to monoallelic expression in addition to being critical for X inactivation. *Epigenetics Chromatin*, **6**, 19.
13. Wanigasuriya,I., Gouil,Q., Kinkel,S.A., Tapia Del Fierro,A., Beck,T., Roper,E.A., Breslin,K., Stringer,J., Hutt,K., Lee,H.J. *et al.* (2020) Smchd1 is a maternal effect gene required for genomic imprinting. *eLife*, **9**, e55529.
14. Mason,A.G., Slieker,R.C., Balog,J., Lemmers,R., Wong,C.J., Yao,Z., Lim,J.W., Filippova,G.N., Ne,E., Tawil,R. *et al.* (2017) SMCHD1 regulates a limited set of gene clusters on autosomal chromosomes. *Skelet. Muscle*, **7**, 12.
15. Brideau,N.J., Coker,H., Gendrel,A.V., Siebert,C.A., Bezstarosti,K., Demmers,J., Poot,R.A., Nesterova,T.B. and Brockdorff,N. (2015) Independent mechanisms target SMCHD1 to trimethylated histone H3 lysine 9-modified chromatin and the inactive X chromosome. *Mol. Cell. Biol.*, **35**, 4053–4068.
16. Massah,S., Hollebakken,R., Labrecque,M.P., Kolybaba,A.M., Beischlag,T.V. and Prefontaine,G.G. (2014) Epigenetic characterization of the growth hormone gene identifies SmcHD1 as a regulator of autosomal gene clusters. *PLoS One*, **9**, e97535.
17. Nozawa,R.S., Nagao,K., Igami,K.T., Shibata,S., Shirai,N., Nozaki,N., Sado,T., Kimura,H. and Obuse,C. (2013) Human inactive X chromosome is compacted through a PRC2-independent SMCHD1-HBiX1 pathway. *Nat. Struct. Mol. Biol.*, **20**, 566–573.
18. Dejardin,J. and Kingston,R.E. (2009) Purification of proteins associated with specific genomic loci. *Cell*, **136**, 175–186.
19. Grolimund,L., Aeby,E., Hamelin,R., Armand,F., Chiappe,D., Moniatte,M. and Lingner,J. (2013) A quantitative telomeric chromatin isolation protocol identifies different telomeric states. *Nat. Commun.*, **4**, 2848.
20. Sacconi,S., Camano,P., de Greef,J.C., Lemmers,R.J., Salviati,L., Boileau,P., Lopez de Munain Arregui,A., van der Maarel,S.M. and Desnuelle,C. (2012) Patients with a phenotype consistent with facioscapulohumeral muscular dystrophy display genetic and epigenetic heterogeneity. *J. Med. Genet.*, **49**, 41–46.
21. Lemmers,R.J., Tawil,R., Petek,L.M., Balog,J., Block,G.J., Santen,G.W., Amell,A.M., van der Vliet,P.J., Almomani,R., Straasheijm,K.R. *et al.* (2012) Digenic inheritance of an SMCHD1 mutation and an FSHD-permissive D4Z4 allele causes facioscapulohumeral muscular dystrophy type 2. *Nat. Genet.*, **44**, 1370–1374.
22. Gaillard,M.C., Puppo,F., Roche,S., Dion,C., Campana,E.S., Mariot,V., Chaix,C., Vovan,C., Mazaleyrat,K., Tasmadjian,A. *et al.* (2016) Segregation between SMCHD1 mutation, D4Z4 hypomethylation and facio-scapulo-humeral dystrophy: a case report. *BMC Med. Genet.*, **17**, 66.
23. Gordon,C.T., Xue,S., Yigit,G., Filali,H., Chen,K., Rosin,N., Yoshiura,K.I., Oufadem,M., Beck,T.J., McGowan,R. *et al.* (2017) *De novo* mutations in SMCHD1 cause Bosma arhinia microphthalmia syndrome and abrogate nasal development. *Nat. Genet.*, **49**, 249–255.
24. Shaw,N.D., Brand,H., Kupchinsky,Z.A., Bengani,H., Plummer,L., Jones,T.I., Erdin,S., Williamson,K.A., Rainger,J., Stortchevoi,A. *et al.* (2017) SMCHD1 mutations associated with a rare muscular dystrophy can also cause isolated arhinia and Bosma arhinia microphthalmia syndrome. *Nat. Genet.*, **49**, 238–248.
25. Kinjo,K., Nagasaki,K., Muroya,K., Suzuki,E., Ishiwata,K., Nakabayashi,K., Hattori,A., Nagao,K., Nozawa,R.S., Obuse,C. *et al.* (2020) Rare variant of the epigenetic regulator SMCHD1 in a patient with pituitary hormone deficiency. *Sci. Rep.*, **10**, 10985.
26. Hasi-Zogaj,M., Sebold,C., Heard,P., Carter,E., Soileau,B., Hill,A., Rupert,D., Perry,B., Atkinson,S., O'Donnell,L. *et al.* (2015) A review of 18p deletions. *Am. J. Med. Genet. C Semin. Med. Genet.*, **169**, 251–264.
27. Cody,J.D., Sebold,C., Heard,P., Carter,E., Soileau,B., Hasi-Zogaj,M., Hill,A., Rupert,D., Perry,B., O'Donnell,L. *et al.* (2015) Consequences of chromosome 18q deletions. *Am. J. Med. Genet. C Semin. Med. Genet.*, **169**, 265–280.
28. Lemmers,R.J., van den Boogaard,M.L., van der Vliet,P.J., Donlin-Smith,C.M., Nations,S.P., Ruivenkamp,C.A., Heard,P., Bakker,B., Tapscott,S., Cody,J.D. *et al.* (2015) Hemizygosity for SMCHD1 in facioscapulohumeral muscular dystrophy type 2: consequences for 18p deletion syndrome. *Hum. Mutat.*, **36**, 679–683.
29. Gurzau,A.D., Chen,K., Xue,S., Dai,W., Lucet,I.S., Ly,T.T.N., Reversade,B., Blewitt,M.E. and Murphy,J.M. (2018) FSHD2- and BAMS-associated mutations confer opposing effects on SMCHD1 function. *J. Biol. Chem.*, **293**, 9841–9853.
30. Dion,C., Roche,S., Laberthonniere,C., Broucqsault,N., Mariot,V., Xue,S., Gurzau,A.D., Nowak,A., Gordon,C.T., Gaillard,M.C. *et al.* (2019) SMCHD1 is involved in *de novo* methylation of the DUX4-encoding D4Z4 macrosatellite. *Nucleic Acids Res.*, **47**, 2822–2839.
31. Wijmenga,C., Sandkuijl,L.A., Moerer,P., van der Boorn,N., Bodrug,S.E., Ray,P.N., Brouwer,O.F., Murray,J.C., van Ommen,G.J., Padberg,G.W. *et al.* (1992) Genetic linkage map of facioscapulohumeral muscular dystrophy and five polymorphic loci on chromosome 4q35-qter. *Am. J. Hum. Genet.*, **51**, 411–415.
32. Lyle,R., Wright,T.J., Clark,L.N. and Hewitt,J.E. (1995) The FSHD-associated repeat, D4Z4, is a member of a dispersed family of homeobox-containing repeats, subsets of which are clustered on the short arms of the acrocentric chromosomes. *Genomics*, **28**, 389–397.
33. van Overveld,P.G., Enthoven,L., Ricci,E., Rossi,M., Felicetti,L., Jeanpierre,M., Winokur,S.T., Frants,R.R., Padberg,G.W. and van der Maarel,S.M. (2005) Variable hypomethylation of D4Z4 in facioscapulohumeral muscular dystrophy. *Ann. Neurol.*, **58**, 569–576.
34. van Overveld,P.G., Lemmers,R.J., Sandkuijl,L.A., Enthoven,L., Winokur,S.T., Bakels,F., Padberg,G.W., van Ommen,G.J., Frants,R.R. and van der Maarel,S.M. (2003) Hypomethylation of D4Z4 in 4q-linked and non-4q-linked facioscapulohumeral muscular dystrophy. *Nat. Genet.*, **35**, 315–317.
35. de Greef,J.C., Lemmers,R.J., van Engelen,B.G., Sacconi,S., Venance,S.L., Frants,R.R., Tawil,R. and van der Maarel,S.M. (2009) Common epigenetic changes of D4Z4 in contraction-dependent and contraction-independent FSHD. *Hum. Mutat.*, **30**, 1449–1459.
36. Hartweck,L.M., Anderson,L.J., Lemmers,R.J., Dandapat,A., Toso,E.A., Dalton,J.C., Tawil,R., Day,J.W., van der Maarel,S.M. and Kyba,M. (2013) A focal domain of extreme demethylation within D4Z4 in FSHD2. *Neurology*, **80**, 392–399.
37. Gaillard,M.C., Roche,S., Dion,C., Tasmadjian,A., Bouget,G., Salort-Campana,E., Vovan,C., Chaix,C., Broucqsault,N., Morere,J. *et al.* (2014) Differential DNA methylation of the D4Z4 repeat in patients with FSHD and asymptomatic carriers. *Neurology*, **83**, 733–742.
38. Dion,C., Roche,S., Laberthonniere,C., Broucqsault,N., Mariot,V., Xue,S., Gurzau,A.D., Nowak,A., Gordon,C.T., Gaillard,M.C. *et al.* (2019) SMCHD1 is involved in *de novo* methylation of the DUX4-encoding D4Z4 macrosatellite. *Nucleic Acids Res.*, **47**, 2822–2839.
39. Badja,C., Maleeva,G., El-Yazidi,C., Barruet,E., Lasserre,M., Tropel,P., Binetruy,B., Bregestovski,P. and Magdinier,F. (2014) Efficient and cost-effective generation of mature neurons from human induced pluripotent stem cells. *Stem Cells Transl. Med.*, **3**, 1467–1472.
40. Mazaleyrat,K., Badja,C., Broucqsault,N., Chevalier,R., Laberthonniere,C., Dion,C., Baldasseroni,L., El-Yazidi,C., Thomas,M., Bachelier,R. *et al.* (2020) Multilineage differentiation for formation of innervated skeletal muscle fibers from healthy and diseased human pluripotent stem cells. *Cells*, **9**, 1531.
41. Delourme,M., Broucqsault,N., Mazaleyrat,K. and Magdinier,F. (2020) Production of innervated skeletal muscle fibers using human induced pluripotent stem cells. *Methods Mol. Biol.*, **2454**, 231–239.
42. Laberthonniere,C., Novoa-Del-Toro,E.M., Chevalier,R., Broucqsault,N., Rao,V.V., Trani,J.P., Nguyen,K., Xue,S., Reversade,B., Robin,J.D. *et al.* (2021) AKT signaling modifies the balance between cell proliferation and migration in neural crest cells from patients affected with Bosma arhinia and microphthalmia syndrome. *Biomedicines*, **9**, 751.
43. Magdinier,F., Billard,L.M., Wittmann,G., Frappart,L., Benchaib,M., Lenoir,G.M., Guerin,J.F. and Dante,R. (2000) Regional methylation

of the 5′ end CpG island of BRCA1 is associated with reduced gene expression in human somatic cells. *FASEB J.*, **14**, 1585–1594.
44. Roche,S., Dion,C., Broucqsault,N., Laberthonniere,C., Gaillard,M.C., Robin,J.D., Lagarde,A., Puppo,F., Vovan,C., Chaix,C. *et al.* (2019) Methylation hotspots evidenced by deep sequencing in patients with facioscapulohumeral dystrophy and mosaicism. *Neurol. Genet.*, **5**, e372.
45. Ottaviani,A., Rival-Gervier,S., Boussouar,A., Foerster,A.M., Rondier,D., Sacconi,S., Desnuelle,C., Gilson,E. and Magdinier,F. (2009) The D4Z4 macrosatellite repeat acts as a CTCF and A-type lamins-dependent insulator in facio-scapulo-humeral dystrophy. *PLoS Genet.*, **5**, e1000394.
46. Koering,C.E., Pollice,A., Zibella,M.P., Bauwens,S., Puisieux,A., Brunori,M., Brun,C., Martins,L., Sabatier,L., Pulitzer,J.F. *et al.* (2002) Human telomeric position effect is determined by chromosomal context and telomeric chromatin integrity. *EMBO Rep.*, **3**, 1055–1061.
47. Morris,T.J., Butcher,L.M., Feber,A., Teschendorff,A.E., Chakravarthy,A.R., Wojdacz,T.K. and Beck,S. (2014) ChAMP: 450k chip analysis methylation pipeline. *Bioinformatics*, **30**, 428–430.
48. Butcher,L.M. and Beck,S. (2015) Probe Lasso: a novel method to rope in differentially methylated regions with 450K DNA methylation data. *Methods*, **72**, 21–28.
49. Du,P., Zhang,X., Huang,C.C., Jafari,N., Kibbe,W.A., Hou,L. and Lin,S.M. (2010) Comparison of Beta-value and *M*-value methods for quantifying methylation levels by microarray analysis. *BMC Bioinformatics*, **11**, 587.
50. Bibikova,M., Barnes,B., Tsan,C., Ho,V., Klotzle,B., Le,J.M., Delano,D., Zhang,L., Schroth,G.P., Gunderson,K.L. *et al.* (2011) High density DNA methylation array with single CpG site resolution. *Genomics*, **98**, 288–295.
51. Ritchie,M.E., Phipson,B., Wu,D., Hu,Y., Law,C.W., Shi,W. and Smyth,G.K. (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.*, **43**, e47.
52. Johnson,W.E., Li,C. and Rabinovic,A. (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, **8**, 118–127.
53. Moran,S., Arribas,C. and Esteller,M. (2016) Validation of a DNA methylation microarray for 850,000 CpG sites of the human genome enriched in enhancer sequences. *Epigenomics*, **8**, 389–399.
54. Geng,L.N., Yao,Z., Snider,L., Fong,A.P., Cech,J.N., Young,J.M., van der Maarel,S.M., Ruzzo,W.L., Gentleman,R.C., Tawil,R. *et al.* (2012) DUX4 activates germline genes, retroelements, and immune mediators: implications for facioscapulohumeral dystrophy. *Dev. Cell*, **22**, 38–51.
55. Xue,S., Ly,T.T.N., Vijayakar,R.S., Chen,J., Ng,J., Mathuru,A.S., Magdinier,F. and Reversade,B. (2022) HOX epimutations driven by maternal SMCHD1/LRIF1 haploinsufficiency trigger homeotic transformations in genetically wildtype offspring. *Nat. Commun.*, **13**, 3583.
56. Laberthonniere,C., Novoa-Del-Toro,E.M., Delourme,M., Chevalier,R., Broucqsault,N., Mazaleyrat,K., Streichenberger,N., Manel,V., Bernard,R., Salort Campana,E. *et al.* (2022) Facioscapulohumeral dystrophy weakened sarcomeric contractility is mimicked in induced pluripotent stem cells-derived innervated muscle fibres. *J. Cachexia Sarcopenia Muscle*, **13**, 621–635.
57. Huang,Z., Yu,J., Cui,W., Johnson,B.K., Kim,K. and Pfeifer,G.P. (2021) The chromosomal protein SMCHD1 regulates DNA methylation and the 2c-like state of embryonic stem cells by antagonizing TET proteins. *Sci. Adv.*, **7**, eabb9149.
58. Ruebel,M.L., Vincent,K.A., Schall,P.Z., Wang,K. and Latham,K.E. (2019) SMCHD1 terminates the first embryonic genome activation event in mouse two-cell embryos and contributes to a transcriptionally repressive state. *Am. J. Physiol. Cell Physiol.*, **317**, C655–C664.
59. Ottaviani,A., Schluth-Bolard,C., Gilson,E. and Magdinier,F. (2010) D4Z4 as a prototype of CTCF and lamins-dependent insulator in human cells. *Nucleus*, **1**, 30–36.
60. Schall,P.Z., Ruebel,M.L. and Latham,K.E. (2019) A new role for SMCHD1 in life's master switch and beyond. *Trends Genet.*, **35**, 948–955.
61. Weth,O., Paprotka,C., Gunther,K., Schulte,A., Baierl,M., Leers,J., Galjart,N. and Renkawitz,R. (2014) CTCF induces histone variant incorporation, erases the H3K27me3 histone mark and opens chromatin. *Nucleic Acids Res.*, **42**, 11941–11951.
62. Xu,M., Zhao,G.N., Lv,X., Liu,G., Wang,L.Y., Hao,D.L., Wang,J., Liu,D.P. and Liang,C.C. (2014) CTCF controls HOXA cluster silencing and mediates PRC2-repressive higher-order chromatin structure in NT2/D1 cells. *Mol. Cell. Biol.*, **34**, 3867–3879.

© The Author(s) 2023. Published by Oxford University Press on behalf of Nucleic Acids Research.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Conclusion

In this thesis, I presented the results of the research I carried out, across three different projects, towards a PhD in Bioinformatics. The primary theme of this research, central to two of my PhD projects, was the exploration of approaches for using prior information to enhance the inference of latent variables from omics data.

For my main project (chapter 4), I developed and implemented MOTL, a transfer learning approach to multi-omics matrix factorization. MOTL was developed in response to the need for analysis methods that can accommodate multi-omics data generated from a small number of samples. I designed two protocols, based on simulated and real multi-omics data, to evaluate the performance of MOTL. With these protocols, as well as an additional application to a small cancer dataset, I observed that using prior information, via transfer learning, enhanced the discovery of biological processes and pathways associated with different cancer types. With this evaluation, I was able to demonstrate the effectiveness of MOTL, as well as address the need for protocols that enable the quantitative evaluation of transfer learning approaches to matrix factorization.

MOTL has the potential to become a widely used, advantageous tool, and is especially likely to be of use to researchers studying rare diseases, for whom it can be difficult to obtain multi-omics datasets generated from more than a small number of samples. As discussed in chapter 4, more evaluation should be carried out to give guidance to rare disease researchers on the use of MOTL. In particular, it would be good to be able to give guidance on: the required level of similarity between target and learning datasets; the required size and level of heterogeneity of the learning datasets. Additionally, MOTL might benefit from the flexibility of being able to infer latent variables, for a target dataset, that were not transferred from the factorization of a learning dataset.

A proposed future application, for MOTL, is the analysis of multi-omics data from the diseases type 2 facioscapulohumeral muscular dystrophy (FSHD2), and Bosma arhinia and microphthalmia syndrome (BAMS). In chapter 6, I presented one of my other projects, which involved the analysis of omics data from FSHD2 and BAMS samples. In this project, different types of omics data were analyzed separately with differential testing, yielding subsets of observed variables potentially associated with these diseases. With my collaborators, we used prior information, in the form of annotation resources, to convert these lists of relevant observed variables into lists of relevant biological processes. As a natural extension to these analyses, we intend to apply MOTL to available mutli-omics data, from these samples, with the hope of gaining further insight into the biological processes associated with these diseases.

I also explored the use of prior information, to enhance the inference of latent

variables from omics data, as part of another project carried out during a six week research stay at the Institute for Pharmacy and Molecular Biotechnology, Heidelberg University (chapter 5). More specifically, while in Heidelberg I worked with OntoVAE, a variational autoencoder that already incorporated prior information, in the form of a directed acyclic graph, in order to improve interpretability. I implemented, and evaluated, a modification to OntoVAE, allowing the inclusion of weights that control the relative contribution of genes to the loss function. I observed that the inclusion of gene weights, based on prior information, resulted in latent variable values that better differentiated between groups of samples. As discussed in chapter 5, this proof-of-concept evaluation could be improved by trying different sources of information for deriving gene weights, different datasets for the evaluation, and conducting more in-depth analysis of the biological relevance of the inferred latent variables. Additionally, other possible extensions of OntoVAE include allowing the gene weights to change during training, or allowing the removal and addition of connections between the latent variables and the values in the reconstruction layer.

Overall, with the work carried out during my PhD, I have made contributions to the field of data integration for the analysis of omics data. My work was primarily focused on inferring latent variables with unsupervised dimension reduction, however there are other categories of methods routinely employed for omics data analysis. A possible extension of the work outlined in this thesis is to further explore data integration through the combination of dimension reduction with other analysis methods. In addition to the annotation resources mentioned in this thesis, there are numerous sources of prior information in the form of networks of molecular interactions, and network analysis is an active area of research in the field of bioinformatics. As such, I would be interested in pursuing future research into novel ways in which dimension reduction can be enhanced, by combining it with network analysis techniques such as module identification, and random walks, to integrate prior information. In particular, I have in mind the use of multi-layer networks, to either constrain the fitting of a dimension reduction model for multi-omics data, or to help with interpretation of the resulting latent variables. Alternatively, inferring multi-layer networks based on the outputs of dimension reduction could be an avenue for future exploration.

Bibliography

[1] Aviv Regev, Sarah A Teichmann, Eric S Lander, et al. “The Human Cell Atlas”. In: *eLife* 6 (2017). Ed. by Thomas R Gingeras, e27041 (cit. on pp. 17, 18, 23).

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, et al. “The Universal Features of Cells on Earth”. In: *Molecular Biology of the Cell*. 2002 (cit. on p. 18).

[3] Suzan Ruijtenberg and Sander van den Heuvel. “Coordinating cell proliferation and differentiation: Antagonism between cell cycle regulators and cell type-specific gene expression”. In: *Cell Cycle* 15.2 (2016), pp. 196–212 (cit. on p. 18).

[4] Ian A. Hatton, Eric D. Galbraith, Nono S. C. Merleau, et al. “The human cell count and size distribution”. In: *Proceedings of the National Academy of Sciences* 120.39 (2023), e2303077120 (cit. on p. 18).

[5] Herman Denis and Jean-Claude Lacroix. “The dichotomy between germ line and somatic line, and the origin of cell mortality”. In: *Trends in Genetics* 9.1 (1993), pp. 7–11 (cit. on p. 18).

[6] Bruce Alberts, Alexander Johnson, Julian Lewis, et al. “Catalysis and the Use of Energy by Cells”. In: *Molecular Biology of the Cell*. 2002 (cit. on p. 18).

[7] Michael Ashburner, Catherine A. Ball, Judith A. Blake, et al. “Gene Ontology: tool for the unification of biology”. In: *Nature Genetics* 25.1 (2000), p. 25 (cit. on p. 18).

[8] Bruce Alberts, Alexander Johnson, Julian Lewis, et al. “Protein Function”. In: *Molecular Biology of the Cell*. 2002 (cit. on pp. 18, 22).

[9] Takashi Ohtsuki and Masahiko Sisido. “The Central Dogma: From DNA to RNA, and to Protein”. In: *Automation in Proteomics and Genomics*. 2009 (cit. on pp. 18, 20, 22).

[10] Bruce Alberts, Alexander Johnson, Julian Lewis, et al. “The Structure and Function of DNA”. In: *Molecular Biology of the Cell*. 2002 (cit. on p. 18).

[11] Allison Piovesan, Maria Chiara Pelleri, Francesca Antonaros, et al. “On the length, weight and GC content of the human genome”. In: *BMC Research Notes* 12.1 (2019), p. 106 (cit. on p. 18).

[12] Andrea S. Foulkes. *Applied Statistical Genetics with R*. 2009 (cit. on p. 18).

[13] *Talking Glossary of Genetic Terms | NHGRI*. 2024. URL: https://www.genome.gov/genetics-glossary (cit. on pp. 18, 20, 21, 25).

[14] Patricia J Pukkila. “Molecular Biology: The Central Dogma”. In: *Encyclopedia of Life Sciences*. 2001 (cit. on pp. 18, 20, 22).

[15] Thanasis Mitsis, Aspasia Efthimiadou, Flora Bacopoulou, et al. "Transcription factors and evolution: An integral part of gene expression (Review)". In: *World Academy of Sciences Journal* 2.1 (2020), pp. 3–8 (cit. on p. 18).

[16] Judith L Fridovich-Keil. *Human genome*. 2024. URL: https://www.britannica.com/science/human-genome (cit. on p. 19).

[17] Bruce Alberts, Alexander Johnson, Julian Lewis, et al. "Chromosomal DNA and Its Packaging in the Chromatin Fiber". In: *Molecular Biology of the Cell*. 2002 (cit. on pp. 19, 20).

[18] Pingyu Zhang, Keila Torres, Xiuping Liu, et al. "An Overview of Chromatin-Regulating Proteins in Cells". In: *Current Protein & Peptide Science* 17.5 (2016), pp. 401–410 (cit. on pp. 19, 20, 23).

[19] Estela G. Toraño, María G. García, Juan Luis Fernández-Morera, et al. "The Impact of External Factors on the Epigenome: In Utero and over Lifetime". In: *BioMed Research International* 2016 (2016), p. 2568635 (cit. on pp. 19, 20).

[20] *Epigenomics Fact Sheet*. 2020. URL: https://www.genome.gov/about-genomics/fact-sheets/Epigenomics-Fact-Sheet (cit. on p. 19).

[21] E. R. Gibney and C. M. Nolan. "Epigenetics and gene expression". In: *Heredity* 105.1 (2010), pp. 4–13 (cit. on pp. 19, 20).

[22] Martin J. Aryee, Andrew E. Jaffe, Hector Corrada-Bravo, et al. "Minfi: a flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays". In: *Bioinformatics* 30.10 (2014), pp. 1363–1369 (cit. on pp. 19, 39).

[23] Serge Saxonov, Paul Berg, and Douglas L. Brutlag. "A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters". In: *Proceedings of the National Academy of Sciences of the United States of America* 103.5 (2006), pp. 1412–1417 (cit. on p. 19).

[24] En Li and Yi Zhang. "DNA Methylation in Mammals". In: *Cold Spring Harbor Perspectives in Biology* 6.5 (2014), a019133 (cit. on pp. 19, 23).

[25] Hiroaki Okae, Hatsune Chiba, Hitoshi Hiura, et al. "Genome-Wide Analysis of DNA Methylation Dynamics during Early Human Development". In: *PLOS Genetics* 10.12 (2014), e1004868 (cit. on p. 20).

[26] Elena López-Jiménez and Eduardo Andrés-León. "The Implications of ncRNAs in the Development of Human Diseases". In: *Non-Coding RNA* 7.1 (2021), p. 17 (cit. on p. 20).

[27] Kannan Ranganathan and Vaishnavi Sivasankar. "MicroRNAs - Biology and clinical applications". In: *Journal of Oral and Maxillofacial Pathology : JOMFP* 18.2 (2014), pp. 229–234 (cit. on p. 20).

[28] Bruce Alberts, Alexander Johnson, Julian Lewis, et al. "From DNA to RNA". In: *Molecular Biology of the Cell*. 2002 (cit. on p. 21).

[29] Len A. Pennacchio, Wendy Bickmore, Ann Dean, et al. “Enhancers: five essential questions”. In: *Nature reviews. Genetics* 14.4 (2013), pp. 288–295 (cit. on p. 21).

[30] Anil Panigrahi and Bert W. O’Malley. “Mechanisms of enhancer action: the known and the unknown”. In: *Genome Biology* 22.1 (2021), p. 108 (cit. on p. 21).

[31] George Orphanides and Danny Reinberg. “A Unified Theory of Gene Expression”. In: *Cell* 108.4 (2002), pp. 439–451 (cit. on pp. 21, 23).

[32] Claudia Manzoni, Demis A Kia, Jana Vandrovcova, et al. “Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences”. In: *Briefings in Bioinformatics* 19.2 (2018), pp. 286–302 (cit. on pp. 21, 23–25, 27).

[33] *Transcriptome Fact Sheet*. 2020. URL: https://www.genome.gov/about-genomics/fact-sheets/Transcriptome-Fact-Sheet (cit. on p. 21).

[34] Jamie K. Teer and James C. Mullikin. “Exome sequencing: the sweet spot before whole genomes”. In: *Human Molecular Genetics* 19.R2 (2010), R145–R151 (cit. on pp. 21, 24).

[35] *NCI Dictionary of Genetics Terms - NCI*. 2024. URL: https://www.cancer.gov/publications/dictionaries/genetics-dictionary/ (cit. on p. 22).

[36] Bruce Alberts, Alexander Johnson, Julian Lewis, et al. “From RNA to Protein”. In: *Molecular Biology of the Cell*. 2002 (cit. on p. 22).

[37] Rhiannon Morris, Katrina A. Black, and Elliott J. Stollar. “Uncovering protein function: from classification to complexes”. In: *Essays in Biochemistry* 66.3 (2022), pp. 255–285 (cit. on p. 22).

[38] W. Thomas Boyce, Marla B. Sokolowski, and Gene E. Robinson. “Genes and environments, development and time”. In: *Proceedings of the National Academy of Sciences* 117.38 (2020), pp. 23235–23241 (cit. on p. 23).

[39] Laura J. van ’t Veer, Hongyue Dai, Marc J. van de Vijver, et al. “Gene expression profiling predicts clinical outcome of breast cancer”. In: *Nature* 415.6871 (2002), pp. 530–536 (cit. on p. 23).

[40] Peng Liang and Arthur B. Pardee. “Analysing differential gene expression in cancer”. In: *Nature Reviews Cancer* 3.11 (2003), pp. 869–876 (cit. on p. 23).

[41] Christine M. Micheel, Sharly J. Nass, Gilbert S. Omenn, et al. “Omics-Based Clinical Discovery: Science, Technology, and Applications”. In: *Evolution of Translational Omics: Lessons Learned and the Path Forward*. 2012 (cit. on p. 23).

[42] Yehudit Hasin, Marcus Seldin, and Aldons Lusis. “Multi-omics approaches to disease”. In: *Genome Biology* 18.1 (2017), p. 83 (cit. on p. 23).

[43] Biswapriya B. Misra, Carl D. Langefeld, Michael Olivier, et al. “Integrated Omics: Tools, Advances, and Future Approaches”. In: *Journal of Molecular Endocrinology* (2018), JME–18–0055 (cit. on p. 23).

[44] Héctor Quezada, Ana Laura Guzmán-Ortiz, Hugo Díaz-Sánchez, et al. "Omics-based biomarkers: current status and potential use in the clinic". In: *Boletín Médico Del Hospital Infantil de México (English Edition)* 74.3 (2017), pp. 219–226 (cit. on p. 23).

[45] Ana Conesa and Stephan Beck. "Making multi-omics data accessible to researchers". In: *Scientific Data* 6.1 (2019), p. 251 (cit. on pp. 23, 28).

[46] Xiaofeng Dai and Li Shen. "Advances and Trends in Omics Technology Development". In: *Frontiers in Medicine* 9 (2022), p. 911861 (cit. on pp. 23, 24, 27).

[47] Ryo Yamada, Daigo Okada, Juan Wang, et al. "Interpretation of omics data analyses". In: *Journal of Human Genetics* 66.1 (2021), pp. 93–102 (cit. on pp. 23, 30, 34, 35, 39, 41, 45).

[48] Alice Maria Giani, Guido Roberto Gallo, Luca Gianfranceschi, et al. "Long walk to genomics: History and current approaches to genome sequencing and assembly". In: *Computational and Structural Biotechnology Journal* 18 (2020), pp. 9–19 (cit. on p. 24).

[49] Jian Li, Aarif Mohamed Nazeer Batcha, Björn Grüning, et al. "An NGS Workflow Blueprint for DNA Sequencing Data and Its Application in Individualized Molecular Oncology". In: *Cancer Informatics* 14.Suppl 5 (2016), pp. 87–107 (cit. on p. 24).

[50] David Gresham, Maitreya J. Dunham, and David Botstein. "Comparing whole genomes using DNA microarrays". In: *Nature Reviews Genetics* 9.4 (2008), pp. 291–302 (cit. on p. 24).

[51] Taishan Hu, Nilesh Chitnis, Dimitri Monos, et al. "Next-generation sequencing technologies: An overview". In: *Human Immunology* 82.11 (2021), pp. 801–811 (cit. on p. 24).

[52] Eleanor G. Seaby, Reuben J. Pengelly, and Sarah Ennis. "Exome sequencing explained: a practical guide to its clinical application". In: *Briefings in Functional Genomics* 15.5 (2016), pp. 374–384 (cit. on p. 24).

[53] *Human Genome Reference Sequence*. 2024. URL: https://www.genome.gov/genetics-glossary/Human-Genome-Reference-Sequence (cit. on p. 25).

[54] Aziz Belkadi, Alexandre Bolze, Yuval Itan, et al. "Whole-genome sequencing is more powerful than whole-exome sequencing for detecting exome variants". In: *Proceedings of the National Academy of Sciences* 112.17 (2015), pp. 5473–5478 (cit. on p. 25).

[55] Ondrej Pös, Jan Radvanszky, Gergely Buglyó, et al. "DNA copy number variation: Main characteristics, evolutionary significance, and pathological aspects". In: *Biomedical Journal* 44.5 (2021), p. 548 (cit. on p. 25).

[56] Daniel C. Koboldt. "Best practices for variant calling in clinical sequencing". In: *Genome Medicine* 12.1 (2020), p. 91 (cit. on p. 25).

[57] Heng Li. “A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data”. In: *Bioinformatics* 27.21 (2011), pp. 2987–2993 (cit. on p. 25).

[58] Daniel C. Koboldt, Qunyuan Zhang, David E. Larson, et al. “VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing”. In: *Genome Research* 22.3 (2012), pp. 568–576 (cit. on p. 25).

[59] David Benjamin, Takuto Sato, Kristian Cibulskis, et al. “Calling Somatic SNVs and Indels with Mutect2”. In: *bioRxiv* (2019) (cit. on p. 25).

[60] William McLaren, Bethan Pritchard, Daniel Rios, et al. “Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor”. In: *Bioinformatics* 26.16 (2010), pp. 2069–2070 (cit. on p. 25).

[61] Kai Wang, Mingyao Li, and Hakon Hakonarson. “ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data”. In: *Nucleic Acids Research* 38.16 (2010), e164 (cit. on p. 25).

[62] David E. Larson, Christopher C. Harris, Ken Chen, et al. “SomaticSniper: identification of somatic point mutations in whole genome sequencing data”. In: *Bioinformatics* 28.3 (2012), pp. 311–317 (cit. on p. 25).

[63] Daniel C. Koboldt, Karyn Meltz Steinberg, David E. Larson, et al. “The Next-Generation Sequencing Revolution and Its Impact on Genomics”. In: *Cell* 155.1 (2013), pp. 27–38 (cit. on p. 25).

[64] Mahya Mehrmohamadi, Mohammad Hossein Sepehri, Naghme Nazer, et al. “A Comparative Overview of Epigenomic Profiling Methods”. In: *Frontiers in Cell and Developmental Biology* 9 (2021), p. 714687 (cit. on pp. 25, 26).

[65] Yuanyuan Li. “Modern Epigenetics Methods in Biological Research”. In: *Methods* 187 (2021), pp. 104–113 (cit. on pp. 25–27).

[66] Sergey Kurdyukov and Martyn Bullock. “DNA Methylation Analysis: Choosing the Right Method”. In: *Biology* 5.1 (2016), p. 3 (cit. on p. 26).

[67] Ehsan Khodadadi, Leila Fahmideh, Ehsaneh Khodadadi, et al. “Current Advances in DNA Methylation Analysis Methods”. In: *BioMed Research International* 2021 (2021), p. 8827516 (cit. on p. 26).

[68] Ruth Pidsley, Elena Zotenko, Timothy J. Peters, et al. “Critical evaluation of the Illumina MethylationEPIC BeadChip microarray for whole-genome DNA methylation profiling”. In: *Genome Biology* 17.1 (2016), p. 208 (cit. on p. 26).

[69] Pan Du, Xiao Zhang, Chiang-Ching Huang, et al. “Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis”. In: *BMC Bioinformatics* 11.1 (2010), p. 587 (cit. on pp. 26, 40).

[70] Peter J. Park. “ChIP-seq: advantages and challenges of a maturing technology”. In: *NATURE REVIEWS GENETICS* 10.10 (2009), pp. 669–680 (cit. on p. 27).

[71] Terrence S. Furey. "ChIP-seq and beyond: new and improved methodologies to detect and characterize protein-DNA interactions". In: *Nature Reviews Genetics* 13.12 (2012), pp. 840–852 (cit. on p. 27).

[72] Jason Ernst, Pouya Kheradpour, Tarjei S. Mikkelsen, et al. "Mapping and analysis of chromatin state dynamics in nine human cell types". In: *Nature* 473.7345 (2011), pp. 43–49 (cit. on pp. 27, 57).

[73] Li Chen, Chi Wang, Zhaohui S. Qin, et al. "A novel statistical method for quantitative comparison of multiple ChIP-seq datasets". In: *Bioinformatics* 31.12 (2015), p. 1889 (cit. on p. 27).

[74] Rohan Lowe, Neil Shirley, Mark Bleackley, et al. "Transcriptomics technologies". In: *PLoS computational biology* 13.5 (2017), e1005457 (cit. on p. 27).

[75] Kimberly R. Kukurba and Stephen B. Montgomery. "RNA Sequencing and Analysis". In: *Cold Spring Harbor protocols* 2015.11 (2015), pp. 951–969 (cit. on pp. 27, 34, 39).

[76] Alexander Dobin, Carrie A. Davis, Felix Schlesinger, et al. "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1 (2013), pp. 15–21 (cit. on p. 28).

[77] Sijia Huang, Kumardeep Chaudhary, and Lana X. Garmire. "More Is Better: Recent Progress in Multi-Omics Data Integration Methods". In: *Frontiers in Genetics* 8 (2017), p. 84 (cit. on pp. 28, 60).

[78] Jörg Rahnenführer, Riccardo De Bin, Axel Benner, et al. "Statistical analysis of high-dimensional biomedical data: a gentle introduction to analytical goals, common approaches and challenges". In: *BMC Medicine* 21.1 (2023), p. 182 (cit. on pp. 28, 30, 34, 35, 38–42, 44, 46, 52).

[79] Indhupriya Subramanian, Srikant Verma, Shiva Kumar, et al. "Multi-omics Data Integration, Interpretation, and Its Application". In: *Bioinformatics and Biology Insights* 14 (2020) (cit. on pp. 28, 60).

[80] Milan Picard, Marie-Pier Scott-Boyer, Antoine Bodein, et al. "Integration strategies of multi-omics data for machine learning analysis". In: *Computational and Structural Biotechnology Journal* 19 (2021), pp. 3735–3746 (cit. on pp. 28, 45, 60).

[81] Laura Cantini, Pooya Zakeri, Celine Hernandez, et al. "Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer". In: *Nature Communications* 12.1 (2021), p. 124 (cit. on pp. 28, 46, 58, 61, 62, 68).

[82] Carolyn Hutter and Jean Claude Zenklusen. "The Cancer Genome Atlas: Creating Lasting Value beyond Its Data". In: *Cell* 173.2 (2018), pp. 283–285 (cit. on p. 28).

[83] Mohamed Mounir, Marta Lucchetta, Tiago C. Silva, et al. "New functionalities in the TCGAbiolinks package for the study and integration of cancer data from GDC and GTEx". In: *PLOS Computational Biology* 15.3 (2019). Publisher: Public Library of Science, e1006701 (cit. on p. 28).

[84] Darren J. Wilkinson. "Bayesian methods in bioinformatics and computational systems biology". In: *Briefings in Bioinformatics* 8.2 (2007), pp. 109–116 (cit. on p. 31).

[85] Rens van de Schoot, Sarah Depaoli, Ruth King, et al. "Bayesian statistics and modelling". In: *Nature Reviews Methods Primers* 1.1 (2021), pp. 1–26 (cit. on pp. 31–33).

[86] Ronald Christensen, Wesley Johnson, Adam Branscum, et al. *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. 2010 (cit. on p. 32).

[87] Daniela Dunkler, Fátima Sánchez-Cabo, and Georg Heinze. "Statistical Analysis Principles for Omics Data". In: *Bioinformatics for Omics Data: Methods and Protocols*. Ed. by Bernd Mayer. 2011 (cit. on pp. 32, 35, 41, 42).

[88] Bradley Efron. "Bayes, Oracle Bayes and Empirical Bayes". In: *Statistical Science* 34.2 (2019), pp. 177–201 (cit. on p. 32).

[89] Belinda Phipson, Stanley Lee, Ian J. Majewski, et al. "ROBUST HYPERPARAMETER ESTIMATION PROTECTS AGAINST HYPERVARIABLE GENES AND IMPROVES POWER TO DETECT DIFFERENTIAL EXPRESSION". In: *The annals of applied statistics* 10.2 (2016), pp. 946–963 (cit. on p. 32).

[90] Siddhartha Chib and Edward Greenberg. "Understanding the Metropolis-Hastings Algorithm". In: *The American Statistician* 49.4 (1995), pp. 327–335 (cit. on pp. 32, 33).

[91] Andrew Gelman and Donald B. Rubin. "Inference from Iterative Simulation Using Multiple Sequences". In: *Statistical Science* 7.4 (1992), pp. 457–472 (cit. on p. 33).

[92] George Casella and Edward I. George. "Explaining the Gibbs Sampler". In: *The American Statistician* 46.3 (1992), pp. 167–174 (cit. on p. 33).

[93] Radford M. Neal. *MCMC using Hamiltonian dynamics*. 2011 (cit. on p. 33).

[94] Justin Grimmer. "An Introduction to Bayesian Inference via Variational Approximations". In: *Political Analysis* 19.1 (2011), pp. 32–47 (cit. on pp. 33, 34).

[95] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *arXiv* (2017) (cit. on pp. 33, 34).

[96] Rajesh Ranganath, Sean Gerrish, and David M. Blei. "Black Box Variational Inference". In: *arXiv* (2013) (cit. on p. 34).

[97] Alp Kucukelbir. "Automatic Differentiation Variational Inference". In: *Journal of Machine Learning Research* 18 (2017), pp. 1–45 (cit. on p. 34).

[98] Léon Bottou. "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *Statistical Learning and Data Science*. Ed. by Mireille Gettler Summa, Bernard Goldfarb, Fionn Murtagh, et al. 2011 (cit. on pp. 34, 109).

[99] Wilson Wen Bin Goh, Wei Wang, and Limsoon Wong. "Why Batch Effects Matter in Omics Data, and How to Avoid Them". In: *Trends in Biotechnology* 35.6 (2017), pp. 498–507 (cit. on p. 34).

[100] Wanding Zhou, Timothy J Triche Jr, Peter W Laird, et al. "SeSAMe: reducing artifactual detection of DNA methylation by Infinium BeadChips in genomic deletions". In: *Nucleic Acids Research* 46.20 (2018), e123 (cit. on pp. 34, 39).

[101] H. Welsh, C. M. P. F. Batalha, W. Li, et al. "A systematic evaluation of normalization methods and probe replicability using infinium EPIC methylation data". In: *Clinical Epigenetics* 15.1 (2023), p. 41 (cit. on p. 34).

[102] Ana Conesa, Pedro Madrigal, Sonia Tarazona, et al. "A survey of best practices for RNA-seq data analysis". In: *Genome Biology* 17.1 (2016), p. 13 (cit. on pp. 34, 39, 58).

[103] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2009 (cit. on pp. 37, 45).

[104] Tiffany J. Morris, Lee M. Butcher, Andrew Feber, et al. "ChAMP: 450k Chip Analysis Methylation Pipeline". In: *Bioinformatics* 30.3 (2014), pp. 428–430 (cit. on pp. 39, 43).

[105] Jovana Maksimovic, Belinda Phipson, and Alicia Oshlack. "A cross-package Bioconductor workflow for analysing methylation array data". In: *F1000Research* 5 (2017), p. 1281 (cit. on pp. 39, 43).

[106] Tiago C. Silva, Antonio Colaprico, Catharina Olsen, et al. "TCGA Workflow: Analyze cancer genomics and epigenomics data using Bioconductor packages". In: *F1000Research* 5 (2016), p. 1542 (cit. on p. 39).

[107] Qianxing Mo, Ronglai Shen, Cui Guo, et al. "A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data". In: *Biostatistics (Oxford, England)* 19.1 (2018), pp. 71–86 (cit. on pp. 39, 40, 65).

[108] R. Bourgon, R. Gentleman, and W. Huber. "Independent filtering increases detection power for high-throughput experiments". In: *Proceedings of the National Academy of Sciences* 107.21 (2010), pp. 9546–9551 (cit. on p. 39).

[109] Florian Rohart, Benoît Gautier, Amrit Singh, et al. "mixOmics: An R package for 'omics feature selection and multiple data integration". In: *PLOS Computational Biology* 13.11 (2017), e1005752 (cit. on pp. 39, 50).

[110] Chen Meng, Oana A. Zeleznik, Gerhard G. Thallinger, et al. "Dimension reduction techniques for the integrative analysis of multi-omics data". In: *Briefings in Bioinformatics* 17.4 (2016), pp. 628–641 (cit. on pp. 39, 46, 47, 49, 50, 62).

[111] Nizar Touleimat and Jörg Tost. "Complete pipeline for Infinium(®) Human Methylation 450K BeadChip data processing using subset quantile normalization for accurate DNA methylation estimation". In: *Epigenomics* 4.3 (2012), pp. 325–341 (cit. on p. 39).

[112] Marie-Agnès Dillies, Andrea Rau, Julie Aubert, et al. "A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis". In: *Briefings in Bioinformatics* 14.6 (2013), pp. 671–683 (cit. on p. 39).

[113] Ciaran Evans, Johanna Hardin, and Daniel M Stoebel. "Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions". In: *Briefings in Bioinformatics* 19.5 (2018), pp. 776–792 (cit. on p. 39).

[114] Simon Anders and Wolfgang Huber. "Differential expression analysis for sequence count data". In: *Genome Biology* 11.10 (2010), R106 (cit. on p. 39).

[115] Mark D. Robinson and Alicia Oshlack. "A scaling normalization method for differential expression analysis of RNA-seq data". In: *Genome Biology* 11.3 (2010), R25 (cit. on p. 39).

[116] W. Evan Johnson, Cheng Li, and Ariel Rabinovic. "Adjusting batch effects in microarray expression data using empirical Bayes methods". In: *Biostatistics* 8.1 (2007), pp. 118–127 (cit. on p. 40).

[117] Vegard Nygaard, Einar Andreas Rødland, and Eivind Hovig. "Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses". In: *Biostatistics* 17.1 (2016), pp. 29–39 (cit. on p. 40).

[118] Ricard Argelaguet, Britta Velten, Damien Arnol, et al. "Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets". In: *Molecular Systems Biology* 14.6 (2018), e8124 (cit. on pp. 40, 65, 66).

[119] Charity W. Law, Yunshun Chen, Wei Shi, et al. "voom: precision weights unlock linear model analysis tools for RNA-seq read counts". In: *Genome Biology* 15.2 (2014), R29 (cit. on pp. 40, 44).

[120] Amrit Singh, Casey P. Shannon, Benoît Gautier, et al. "DIABLO: an integrative approach for identifying key molecular drivers from multi-omics assays". In: *Bioinformatics (Oxford, England)* 35.17 (2019), pp. 3055–3062 (cit. on p. 40).

[121] Michael I. Love, Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome Biology* 15.12 (2014), p. 550 (cit. on pp. 40, 41, 44, 115).

[122] Yang Yang, Hongjian Sun, Yu Zhang, et al. "Dimensionality reduction by UMAP reinforces sample heterogeneity analysis in bulk transcriptomic data". In: *Cell Reports* 36.4 (2021), p. 109442 (cit. on pp. 40, 46, 52).

[123] Isabella Zwiener, Barbara Frisch, and Harald Binder. "Transforming RNA-Seq Data to Improve the Performance of Prognostic Gene Signatures". In: *PLoS ONE* 9.1 (2014) (cit. on p. 40).

[124] Jaclyn N. Taroni, Peter C. Grayson, Qiwen Hu, et al. "MultiPLIER: A Transfer Learning Framework for Transcriptomics Reveals Systemic Features of Rare Disease". In: *Cell Systems* 8.5 (2019), 380–394.e4 (cit. on pp. 41, 46, 67–70).

[125] Daria Doncevic and Carl Herrmann. "Biologically informed variational autoencoders allow predictive modeling of genetic and drug-induced perturbations". In: *Bioinformatics* 39.6 (2023), btad387 (cit. on pp. 41, 52, 59, 107, 113, 123).

[126] Matthew E. Ritchie, Belinda Phipson, Di Wu, et al. "limma powers differential expression analyses for RNA-sequencing and microarray studies". In: *Nucleic Acids Research* 43.7 (2015), e47–e47 (cit. on pp. 41, 44).

[127] Charity W. Law, Kathleen Zeglinski, Xueyi Dong, et al. "A guide to creating design matrices for gene expression experiments". In: *F1000Research* 9 (2020), p. 1444 (cit. on p. 41).

[128] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995), pp. 289–300 (cit. on p. 42).

[129] Lee J. Bain and Max Engelhardt. *Introduction to probability and mathematical statistics*. 1992 (cit. on p. 42).

[130] Gordon K. Smyth. "Linear models and empirical bayes methods for assessing differential expression in microarray experiments". In: *Statistical Applications in Genetics and Molecular Biology* 3 (2004), Article3 (cit. on p. 43).

[131] Lee M. Butcher and Stephan Beck. "Probe Lasso: A novel method to rope in differentially methylated regions with 450K DNA methylation data". In: *Methods* 72 (2015), pp. 21–28 (cit. on p. 43).

[132] Genevieve L. Stein-O'Brien, Raman Arora, Aedin C. Culhane, et al. "Enter the Matrix: Factorization Uncovers Knowledge from Omics". In: *Trends in Genetics* 34.10 (2018), pp. 790–805 (cit. on pp. 45, 47, 49, 50, 58).

[133] Md Faisal Kabir, Tianjie Chen, and Simone A. Ludwig. "A performance analysis of dimensionality reduction algorithms in machine learning models for cancer prediction". In: *Healthcare Analytics* 3 (2023), p. 100125 (cit. on pp. 45, 46, 49, 50, 52).

[134] Haipeng Shen and Jianhua Z. Huang. "Sparse principal component analysis via regularized low rank matrix approximation". In: *Journal of Multivariate Analysis* 99.6 (2008), pp. 1015–1034 (cit. on pp. 45, 47, 50).

[135] Nicolas Sompairac, Petr V. Nazarov, Urszula Czerwinska, et al. "Independent Component Analysis for Unraveling the Complexity of Cancer Omics Datasets". In: *International Journal of Molecular Sciences* 20.18 (2019), p. 4414 (cit. on pp. 46, 48–50, 58).

[136] Weiguang Mao, Elena Zaslavsky, Boris M. Hartmann, et al. "Pathway-level information extractor (PLIER) for gene expression data". In: *Nature Methods* 16.7 (2019), pp. 607–610 (cit. on pp. 46, 49, 59).

[137] Elana J. Fertig, Jie Ding, Alexander V. Favorov, et al. "CoGAPS: an R/C++ package to identify patterns and biological process activity in transcriptomic data". In: *Bioinformatics* 26.21 (2010), pp. 2792–2793 (cit. on p. 46).

[138] Fangzhou Yao, Jeff Coquery, and Kim-Anh Lê Cao. "Independent Principal Component Analysis for biologically meaningful dimension reduction of large biological data sets". In: *BMC Bioinformatics* 13.1 (2012), p. 24 (cit. on pp. 47, 48).

[139] Michael E. Tipping and Christopher M. Bishop. "Probabilistic Principal Component Analysis". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622 (cit. on p. 48).

[140] A. Hyvärinen and E. Oja. "Independent component analysis: algorithms and applications". In: *Neural Networks* 13.4-5 (2000), pp. 411–430 (cit. on p. 48).

[141] Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, et al. "Metagenes and molecular pattern discovery using matrix factorization". In: *Proceedings of the National Academy of Sciences* 101.12 (2004), pp. 4164–4169 (cit. on p. 49).

[142] Daniel D Lee and H Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791 (cit. on p. 49).

[143] Renaud Gaujoux and Cathal Seoighe. "A flexible R package for nonnegative matrix factorization". In: *BMC Bioinformatics* 11.1 (2010), p. 367 (cit. on pp. 49, 50).

[144] Cedric Fevotte and A Taylan Cemgil. "NONNEGATIVE MATRIX FACTORIZATIONS AS PROBABILISTIC INFERENCE IN COMPOSITE MODELS". In: *17th European Signal Processing Conference* (2009), pp. 1913–1917 (cit. on p. 49).

[145] John L. Horn. "A rationale and test for the number of factors in factor analysis". In: *Psychometrika* 30.2 (1965), pp. 179–185 (cit. on p. 49).

[146] Attila Frigyesi and Mattias Höglund. "Non-Negative Matrix Factorization for the Analysis of Complex Gene Expression Data: Identification of Clinically Relevant Tumor Subtypes". In: *Cancer Informatics* 6 (2008), CIN.S606 (cit. on p. 49).

[147] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv* (2020) (cit. on pp. 50, 52).

[148] Lucas Seninge, Ioannis Anastopoulos, Hongxu Ding, et al. "VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics". In: *Nature Communications* 12 (2021), p. 5684 (cit. on pp. 50, 52).

[149] Hyunsoo Kim and Haesun Park. "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis". In: *Bioinformatics* 23.12 (2007), pp. 1495–1502 (cit. on pp. 50, 58).

[150] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006), pp. 504–507 (cit. on p. 50).

[151] David Pratella, Samira Ait-El-Mkadem Saadi, Sylvie Bannwarth, et al. "A Survey of Autoencoder Algorithms to Pave the Diagnosis of Rare Diseases". In: *International Journal of Molecular Sciences* 22.19 (2021), p. 10891 (cit. on pp. 50–52).

[152] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392 (cit. on p. 51).

[153] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *arXiv* (2014) (cit. on pp. 51, 110).

[154] Christopher M. Bishop. *Pattern recognition and machine learning*. 2006 (cit. on p. 52).

[155] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605 (cit. on p. 52).

[156] *Data integration - Latest research and news | Nature*. 2024. URL: https://www.nature.com/subjects/data-integration (cit. on p. 54).

[157] The Gene Ontology Consortium, Suzi A Aleksander, James Balhoff, et al. "The Gene Ontology knowledgebase in 2023". In: *Genetics* 224.1 (2023) (cit. on p. 55).

[158] *QuickGO*. 2024. URL: https://www.ebi.ac.uk/QuickGO/ (cit. on p. 56).

[159] Antonio Fabregat, Steven Jupe, Lisa Matthews, et al. "The Reactome Pathway Knowledgebase". In: *Nucleic Acids Research* 46.D1 (2018), pp. D649–D655 (cit. on p. 56).

[160] Minoru Kanehisa and Susumu Goto. "KEGG: Kyoto Encyclopedia of Genes and Genomes". In: *Nucleic Acids Research* 28.1 (2000), pp. 27–30 (cit. on p. 56).

[161] *Reactome | Pathway Browser*. 2024. URL: https://reactome.org/PathwayBrowser/ (cit. on p. 57).

[162] Jovana Maksimovic, Alicia Oshlack, and Belinda Phipson. "Gene set enrichment analysis for genome-wide DNA methylation data". In: *Genome Biology* 22.1 (2021), p. 173 (cit. on p. 58).

[163] Guangchuang Yu, Li-Gen Wang, Yanyan Han, et al. "clusterProfiler: an R package for comparing biological themes among gene clusters". In: *Omics: A Journal of Integrative Biology* 16.5 (2012), pp. 284–287 (cit. on p. 58).

[164] Jüri Reimand, Ruth Isser, Veronique Voisin, et al. "Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap". In: *Nature protocols* 14.2 (2019), pp. 482–517 (cit. on p. 58).

[165] Belinda Phipson, Jovana Maksimovic, and Alicia Oshlack. "missMethyl: an R package for analyzing data from Illumina's HumanMethylation450 platform". In: *Bioinformatics* 32.2 (2016), pp. 286–288 (cit. on p. 58).

[166] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, et al. "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545–15550 (cit. on p. 58).

[167] Alexey A. Sergushichev. "An algorithm for fast preranked gene set enrichment analysis using cumulative statistic calculation". In: *bioRxiv* (2016) (cit. on p. 58).

[168] J. D. Storey and R. Tibshirani. "Statistical significance for genomewide studies". In: *Proceedings of the National Academy of Sciences* 100.16 (2003), pp. 9440–9445 (cit. on p. 58).

[169] Purvesh Khatri, Marina Sirota, and Atul J. Butte. "Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges". In: *PLoS Computational Biology* 8.2 (2012), e1002375 (cit. on p. 59).

[170] Ravi Mathur, Daniel Rotroff, Jun Ma, et al. "Gene set analysis methods: a systematic comparison". In: *BioData Mining* 11.1 (2018), p. 8 (cit. on p. 59).

[171] Enrico Glaab, Anaïs Baudot, Natalio Krasnogor, et al. "EnrichNet: network-based gene set enrichment analysis". In: *Bioinformatics* 28.18 (2012), pp. i451–i457 (cit. on p. 59).

[172] Lu Tian, Steven A. Greenberg, Sek Won Kong, et al. "Discovering statistically significant pathways in expression profiling studies". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.38 (2005), p. 13544 (cit. on p. 59).

[173] Nimrod Rappoport and Ron Shamir. "Multi-omic and multi-view clustering algorithms: review and cancer benchmark". In: *Nucleic Acids Research* 46.20 (2018), pp. 10546–10562 (cit. on pp. 61, 62, 68).

[174] Giulia Tini, Luca Marchetti, Corrado Priami, et al. "Multi-omics integration—a comparison of unsupervised clustering methodologies". In: *Briefings in Bioinformatics* 20.4 (2019), pp. 1269–1279 (cit. on pp. 61, 68).

[175] Cécile Chauvel, Alexei Novoloaca, Pierre Veyre, et al. "Evaluation of integrative clustering methods for the analysis of multi-omics data". In: *Briefings in Bioinformatics* 21.2 (2020), pp. 541–552 (cit. on pp. 61, 68).

[176] Morgane Pierre-Jean, Jean-François Deleuze, Edith Le Floch, et al. "Clustering and variable selection evaluation of 13 unsupervised methods for multi-omics data integration". In: *Briefings in Bioinformatics* 21.6 (2020), pp. 2011–2030 (cit. on pp. 61, 62, 68).

[177] Eric F. Lock, Katherine A. Hoadley, J.S. Marron, et al. "JOINT AND INDIVIDUAL VARIATION EXPLAINED (JIVE) FOR INTEGRATED ANALYSIS OF MULTIPLE DATA TYPES". In: *The annals of applied statistics* 7.1 (2013), pp. 523–542 (cit. on p. 62).

[178] Michael J. O'Connell and Eric F. Lock. "R.JIVE for exploration of multi-source molecular data". In: *Bioinformatics* 32.18 (2016), pp. 2877–2879 (cit. on p. 62).

[179] Prabhakar Chalise and Brooke L. Fridley. "Integrative clustering of multi-level 'omic data based on non-negative matrix factorization algorithm". In: *PLOS ONE* 12.5 (2017), e0176278 (cit. on p. 63).

[180] Arthur Tenenhaus and Michel Tenenhaus. "Regularized Generalized Canonical Correlation Analysis". In: *Psychometrika* 76.2 (2011), pp. 257–284 (cit. on p. 63).

[181] Fabien Girka, Etienne Camenen, Caroline Peltier, et al. "Multiblock data analysis with the RGCCA package". In: *Journal of Statistical Software* (2023) (cit. on p. 64).

[182] Imene Garali, Isaac M Adanyeguh, Farid Ichou, et al. "A strategy for multimodal data integration: application to biomarkers identification in spinocerebellar ataxia". In: *Briefings in Bioinformatics* 19.6 (2018), pp. 1356–1369 (cit. on p. 64).

[183] Ricard Argelaguet, Damien Arnol, Danila Bredikhin, et al. "MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data". In: *Genome Biology* 21.1 (2020), p. 111 (cit. on pp. 65, 66).

[184] Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, et al. "Pattern discovery and cancer gene identification in integrated cancer genomic data". In: *Proceedings of the National Academy of Sciences* 110.11 (2013), pp. 4245–4250 (cit. on p. 65).

[185] Jineta Banerjee, Jaclyn N. Taroni, Robert J. Allaway, et al. "Machine learning in rare disease". In: *Nature Methods* (2023), pp. 1–12 (cit. on pp. 67–70).

[186] Melissa Haendel, Nicole Vasilevsky, Deepak Unni, et al. "How many rare diseases are there?" In: *Nature reviews. Drug discovery* 19.2 (2020), pp. 77–78 (cit. on p. 68).

[187] Asmaul Hosna, Ethel Merry, Jigmey Gyalmo, et al. "Transfer learning: a friendly introduction". In: *Journal of Big Data* 9.1 (2022), p. 102 (cit. on p. 69).

[188] Hoo-Chang Shin, Holger R. Roth, Mingchen Gao, et al. "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning". In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1285–1298 (cit. on p. 69).

[189] Dave Bergmann. *What is Fine-Tuning? | IBM*. 2024. URL: https://www.ibm.com/topics/fine-tuning (cit. on pp. 69, 113).

[190] Minshi Peng, Yue Li, Brie Wamsley, et al. "Integration and transfer learning of single-cell transcriptomes via cFIT". In: *Proceedings of the National Academy of Sciences* 118.10 (2021), e2024383118 (cit. on p. 69).

[191] Sinno Jialin Pan, James T Kwok, and Qiang Yang. "Transfer Learning via Dimensionality Reduction". en. In: (2008), p. 6 (cit. on p. 69).

[192] Genevieve L. Stein-O'Brien, Brian S. Clark, Thomas Sherman, et al. "Decomposing Cell Identity for Transfer Learning across Cellular Measurements, Platforms, Tissues, and Species". In: *Cell Systems* 8.5 (2019), 395–411.e8 (cit. on pp. 69, 70).

[193] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big Data* 3.1 (2016), p. 9 (cit. on p. 69).

[194] Yin Zhu, Yuqiang Chen, Zhongqi Lu, et al. "Heterogeneous Transfer Learning for Image Classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 25.1 (2011), pp. 1304–1309 (cit. on p. 69).

[195] Aimei Dong, Zhigang Li, and Qiuyu Zheng. "Transferred Subspace Learning Based on Non-negative Matrix Factorization for EEG Signal Classification". In: *Frontiers in Neuroscience* 15 (2021) (cit. on p. 69).

[196] Sowmini Devi Veeramachaneni, Arun K Pujari, Vineet Padmanabhan, et al. "A Maximum Margin Matrix Factorization based Transfer Learning Approach for Cross-Domain Recommendation". In: *Applied Soft Computing* 85 (2019), p. 105751 (cit. on p. 69).

[197] Rossin Erbe, Michael D Kessler, Alexander V Favorov, et al. "Matrix factorization and transfer learning uncover regulatory biology across multiple single-cell ATAC-seq data sets". In: *Nucleic Acids Research* 48.12 (2020), e68 (cit. on pp. 69, 70).

[198] Gaurav Sharma, Carlo Colantuoni, Loyal A Goff, et al. "projectR: an R/Bioconductor package for transfer learning via PCA, NMF, correlation and clustering". In: *Bioinformatics* 36.11 (2020), pp. 3592–3593 (cit. on pp. 69, 70).

[199] Nicolas Verzelen. "Minimax risks for sparse regressions: Ultra-high dimensional phenomenons". In: *Electronic Journal of Statistics* 6.none (2012), pp. 38–90 (cit. on p. 70).

[200] Emily F. Davis-Marcisak, Allison A. Fitzgerald, Michael D. Kessler, et al. "Transfer learning between preclinical models and human tumors identifies a conserved NK cell activation signature in anti-CTLA-4 responsive tumors". In: *Genome Medicine* 13.1 (2021) (cit. on p. 70).

[201] Leonardo Collado-Torres, Abhinav Nellore, Kai Kammers, et al. "Reproducible RNA-seq analysis using recount2". In: *Nature Biotechnology* 35.4 (2017), pp. 319–321 (cit. on p. 70).

[202] David Hirst, Morgane Térézol, Laura Cantini, et al. "MOTL: enhancing multi-omics matrix factorization with transfer learning". In: *bioRxiv* (2024) (cit. on pp. 71, 74).

[203] Christopher Wilks, Shijie C. Zheng, Feng Yong Chen, et al. "recount3: summaries and queries for large-scale RNA-seq expression and splicing". In: *Genome Biology* 22.1 (2021), p. 323 (cit. on p. 113).

[204] *scikit-learn Naive Bayes*. 2024. URL: https://scikit-learn.org/stable/modules/naive_bayes.html (cit. on p. 116).

[205] *scikit-learn roc_auc_score*. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html (cit. on p. 116).

[206] *scikit-learn KFold*. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html (cit. on p. 116).

[207] *SciPy Wilcoxian*. 2024. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html (cit. on p. 119).

[208] Camille Laberthonnière, Mégane Delourme, Raphaël Chevalier, et al. "In skeletal muscle and neural crest cells, SMCHD1 regulates biological pathways relevant for Bosma syndrome and facioscapulohumeral dystrophy phenotype". In: *Nucleic Acids Research* (2023) (cit. on p. 124).

[209] Mara S. Tihaya, Karlien Mul, Judit Balog, et al. "Facioscapulohumeral muscular dystrophy: the road to targeted therapies". In: *Nature Reviews Neurology* (2023) (cit. on pp. 124, 125).

[210] Alexandra D. Gurzau, Kelan Chen, Shifeng Xue, et al. "FSHD2- and BAMS-associated mutations confer opposing effects on SMCHD1 function". In: *The Journal of Biological Chemistry* 293.25 (2018) (cit. on pp. 124, 125).

[211] Christopher T. Gordon, Shifeng Xue, Gökhan Yigit, et al. "De novo mutations in SMCHD1 cause Bosma arhinia microphthalmia syndrome and abrogate nasal development". In: *Nature Genetics* 49.2 (2017) (cit. on p. 125).