MASTER THESIS

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Medical Engineering & e-Health

Machine learning models for quantifying phenotypic signatures of cancer cells based on transcriptomic and epigenomic data

By: Martin Koban

Student Number: 1910228020

Supervisors: FH-Prof. Dipl. Ing. Dr. Lars Mehnen
Florian Halbritter, PhD

Vienna, September 11, 2020

FH University of Applied Sciences
TECHNIKUM
WIEN

Declaration

“As the author and creator of this work, I confirm with my signature knowledge of relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz / Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of UAS Technikum Wien as amended).

I hereby declare that I have created the presented work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director should there be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of UAS Technikum Wien as amended).

I further declare that, up to this date, I have not published this work nor have I presented it to another examination board in the same or similar form. I affirm that the submitted version of the thesis matches the version in the upload tool.“

Vienna, September 11, 2020 Signature

Abstract

Since the advent of techniques capable of rapid acquisition of genomic data, it is one of the key challenges for researchers to interpret the results of such experiments in meaningful biological terms. In this work, we aim to exploit knowledge hidden in well-characterised transcriptomic and epigenomic data from publicly available sources to aid this interpretation. An integrated resource of chromatin accessibility data (from DNase-seq and ATAC-seq experiments) was created and pre-processed for downstream analyses, complemented by collections of public gene expression (RNA-seq) profiles. These datasets were used for training machine learning classifiers with two primary purposes. Firstly, for augmenting sample annotations by predicting undefined metadata labels in the training datasets. Secondly, for annotation of poorly characterised, unseen data to examine generalisation ability of the constructed models. We demonstrated that biologically relevant information was captured by the trained classifiers while technical artefacts were minimised. Thus, we validated that the proposed supervised machine learning approach can contribute to clarifying contents of cryptic transcriptomic and epigenomic datasets, particularly from the field of cancer research.

Keywords: machine learning, chromatin accessibility, gene expression, metadata, classification, cancer

Rozšírený abstrakt

Rapídny vývoj v oblasti sekvenácie nukleových kyselín umožnil štúdium bunkových procesov na molekulárnej úrovni. Mnohé experimentálne protokoly pre akvizíciu genomických, transkriptomických, epigenomických a ďalších príbuzných dát založené na vysokovýkonnom sekvenovaní sú v súčasnosti vykonávané na rutinnej báze. Veľké množstvo a komplexnosť dát generovaných týmito metódami so sebou prináša výzvy pri interpretácii výsledkov v zmysluplnom biologickom kontexte, pričom bez vhodnej počítačovej analýzy je vyvodenie správnych záverov z experimentálnych údajov väčšinou nemožné. Táto práca sa preto venuje vývoju metód pre charakterizáciu ľudských bunkových vzoriek na základe informácií o ich “molekulárnom programe” so zameraním na identifikáciu rakovinových buniek.

Základným konceptom práce je využitie modelov strojového učenia pre extrakciu štruktúr a súvislostí ukrytých v rozsiahlych súboroch verejne dostupných a dobre anotovaných dát. Informácie získané z týchto trénovacích údajov je následne možné využiť pre popis nových vzoriek, o ktorých vieme z biologického hľadiska veľmi málo. To je častým prípadom práve v onkologickom výskume, kde skúmané bunky podliehajú rozsiahlym morfologickým a funkčným zmenám a ich charakterizácia preto môže byť obzvlášť problematická. Predmetom záujmu sú pritom dva hlavné dátové typy – výsledky transkriptomických experimentov (v rámci ktorých sa kvantifikuje génová expresia) a metód pre hodnotenie chromatínovej dostupnosti, jedného z významných epigenomických faktorov regulujúcich transkripciu génov (jedná sa o mieru kondenzácie molekúl DNA do vyšších štruktúr).

Prvým krokom praktickej časti práce bolo zostavenie trénovacích súborov integrujúcich relevantné dáta z rozličných verejne prístupných zdrojov. Z metód pre kvantifikáciu chromatínovej dostupnosti boli zvolené dve konkrétne techniky – DNase-seq a ATAC-seq, ktoré pre dosiahnutie kvalitatívne porovnateľných výsledkov využívajú odlišný technologický princíp. Pre účely práce bolo potrebné zozbierať profily generované týmito metódami, ktoré kvantifikujú chromatínovú dostupnosť naprieč celým ľudským genómom. Nasledovala agregácia primárnych dát vzhľadom na definovaný súbor regulačných elementov, t.j. nekódujúcich úsekov DNA, ktoré sa zúčastňujú na regulácii génovej expresie (jedná sa napr. o väzobné miesta transkripčných faktorov alebo inhibítorov). Chromatínová dostupnosť regulačných elementov je z biologického hľadiska obzvlášť zaujímavá, pretože vyjadruje mieru ich aktivity, ktorá je výrazne špecifická pre rozličné bunkové typy, vývojové štádiá či fyziologické a patologické stavy. Nakoľko neexistuje jednotný, všeobecne prijatý zoznam regulačných elementov v ľudskom genóme, agregácia bola vykonaná s použitím troch odlišných súborov genomických regiónov s rozličnou mierou komplexnosti. Keďže majú celogenómové profily zvyčajne značný dátový objem, bol vyvinutý nástroj pre ich paralelné stiahnutie z online lokalít a následnú agregáciu, čo prinieslo významné

urýchlenie zberu dát. Čo sa týka kvantifikácie génovej expresie, metódou záujmu bola technika RNA-seq založená na extrakcii a následnej rapídnej sekvenácii molekúl mediátorovej RNA v bunkách. V porovnaní s chromatínovou dostupnosťou sú celogenómové profily génovej expresie dostupnejšie v oveľa väčších množstvách, navyše sú väčšinou aj agregované (v tomto prípade vzhľadom na kódujúce úseky DNA – gény), čo značne uľahčilo zber dát. Okrem použitia zostavených dátových súborov pre účely tejto práce je možné ich priebežné rozširovanie (v prípade dostupnosti nových dát) a využitie pre zodpovedanie ďalších vedeckých otázok – bol tak vytvorený užitočný nástroj pre bioinformatický výskum.

Keďže zozbierané dáta pochádzajú z množstva rozličných zdrojov a boli získané pri rozdielnych experimentálnych podmienkach, prirodzene obsahujú značné množstvo technickej variability, ktorá môže zakryť hľadané biologické rozdiely medzi vzorkami. Tento problém bol adresovaný aplikáciou normalizačných techník pre redukciu technického šumu. Základnou použitou metódou bola "kvantilová normalizácia" (quantile normalisation), ktorá zabezpečí, že všetky numerické vektory pre jednotlivé vzorky majú rovnaké štatistické rozloženie. Ide o štandardnú techniku pre normalizáciu genomických dát. Následná štandardizácia príznakov (t.j. odčítanie priemeru a podelenie hodnôt smerodajnou odchýlkou) je efektívnym spôsobom prevedenia heterogénnych údajov do porovnateľného číselného rozsahu. Pre overenie účinku normalizácie bola potrebná vizualizácia dátových súborov. Zobrazenie mnohorozmerných dát je možné s využitím metód pre redukciu príznakov – okrem analýzy hlavných komponentov, jednej z najstarších a najrozšírenejších techník z tejto kategórie, boli aplikované aj moderné nelineárne prístupy, konkrétne t-SNE (t-distributed stochastic neighbor embedding) a UMAP (uniform manifold approximation and projection). Výstupy týchto metód, ktoré sú schopné premietnuť zložité vysokorozmerné štruktúry do 2-D vizualizácií, potvrdili, že normalizácia priniesla požadovaný efekt zmiernenia technickej variability, zatiaľ čo biologické rozdiely v dátach ostali rozoznateľné.

Okrem primárnych kvantitatívnych dát boli pre potreby trénovania modelov strojového učenia nevyhnutné aj kvalitné metadáta, t.j. anotácie charakterizujúce jednotlivé biologické vzorky (najčastejšie sa jedná o atribúty popisujúce bunkové typy, vývojové štádiá, typ ochorenia, detaily experimentálneho protokolu a pod.). Jednotlivé verejné databázy majú odlišné nároky na štruktúru a jednotnosť anotácií a tak po zozbieraní metadát z mnohých zdrojov bol výsledkom značne heterogénny súbor údajov, veľmi ťažko použiteľný pre automatizovanú počítačovú analýzu. Prvým krokom k riešeniu tejto komplikácie bolo vyvinutie poloautomatického nástroja, ktorého úlohou je zlepšiť konzistenciu anotácií s využitím metód pre spracovanie textu, empiricky definovaných pravidiel a s možnosťou manuálnej korekcie užívateľom. Základnými operáciami v rámci tohto postupu bolo zjednocovanie atribútov s podobným obsahom či náhrada významovo ekvivalentných anotácií jednotnými údajmi pomocou regulárnych výrazov.

Aj napriek popísanému "prečisteniu" anotácií boli dostupné metadáta stále značne nekompletné, t.j. mnohé biologické vzorky boli charakterizované len malým množstvom atribútov. Za účelom zlepšenia tohto stavu bola vyvinutá a implementovaná stratégia pre augmentáciu metadát pomocou modelov strojového učenia s učiteľom. Pre jednotlivé anotácie boli zostavené binárne klasifikátory s cieľom identifikácie vzoriek patriacich do danej kategórie, natrénované a otestované na dostupných dátach. Testovacou stratégiou bola k-násobná krížová validácia, ktorá umožnila kvantitatívne hodnotenie kvality klasifikátorov pomocou výpočtu štandardných metrík pre evaluáciu úspešnosti klasifikácie – presnosti (accuracy), senzitivity (recall) a pozitívnej prediktívnej hodnoty (precision). Na základe extenzívneho testovania bol zvolený vhodný typ klasifikačného modelu – support vector machine s RBF (radial basis function) jadrom – a optimalizované jeho parametre pre danú úlohu. Súhrnné výsledky testovania veľkého množstva binárnych klasifikátorov (vytvorených osobitne pre každú unikátnu anotáciu) umožnili formuláciu pravidiel pre selekciu najkvalitnejších modelov vhodných na samotnú augmentáciu metadát, t.j. predikciu nedefinovaných anotácií.

V poslednej fáze projektu boli klasifikátory natrénované v predchádzajúcom kroku použité pre charakterizáciu nových vzoriek, nepoužitých počas trénovania či testovania. Jednalo sa najmä o biologicky málo popísané dáta, ako napr. výsledky tzv. “single-cell” experimentov, v ktorých je pre meranie génovej expresie či chromatínovej dostupnosti použitý biologický materiál z jedinej bunky. Analýza výstupov týchto metód môže byť obzvlášť problematická, pretože jednotlivé bunky zvyčajne pochádzajú z heterogénneho tkaniva a ich vlastnosti sú tak pred experimentom neznáme. Aplikáciou natrénovaných modelov na profily génovej expresie alebo chromatínovej dostupnosti buniek bola kvantifikovaná ich príslušnosť k jednotlivým triedam (anotáciám), čím bol uskutočnený “preklad” komplexných experimentálnych dát do zrozumiteľných, biologicky informatívnych výrazov.

Pomocou manuálnej revízie a vizualizácie výsledkov klasifikácie nových testovacích dát bolo napokon overené, že natrénované modely poskytujú zmysluplné výsledky a že sú schopné zachytiť biologicky relevantné informácie. Prínos klasifikátorov z hľadiska charakterizácie neznámych bunkových vzoriek je limitovaný najmä obsahom anotácií prislúchajúcich k trénovacím dátam, ktoré definujú výsledné klasifikačné triedy. V súčasnosti je informačný obsah metadát obmedzený predovšetkým na deskriptívne atribúty, ako napr. typy tkanív, buniek či ochorení. Pre získanie hlbšieho náhľadu do bunkových procesov je potrebné rozšírenie existujúcich anotácií o ďalšie biologicky relevantné informácie. Potom je možné využiť postupy vyvinuté a otestované v tejto práci pre lepšie pochopenie zložitých molekulárnych znakov, ktoré sa významne premietajú do funkcie a fenotypu buniek, a tak prispieť k riešeniu jednej zo základných výziev onkologického výskumu – nájdeniu špecificky cielenej biologickej liečby pre čo najviac typov nádorových ochorení.

Acknowledgements

I would like to express my sincere gratitude to Florian Halbritter, who gave me the opportunity to work on this project and continuously supported me with valuable advice and inspiring ideas, in terms of both specific domain knowledge and useful soft skills. A special thanks belongs to my colleagues from Developmental Cancer Genomics group at St. Anna Children's Cancer Research Institute (CCRI) in Vienna – Katja Nettermann and Mohamed Shoeb – for productive collaboration, pleasant working atmosphere and constructive feedback. In particular, I would like to acknowledge Katja Nettermann for providing me with data which played a crucial role in this project. Moreover, it was a great pleasure to work amongst excellent researchers at CCRI and be able to learn from them.

I would also like to thank my supervisor at UAS Technikum Wien, Lars Mehnen, who helped me finalise the thesis through many useful suggestions for improvement. Finally, a special thanks goes to contact persons from UAS Technikum Wien and from Department of Biomedical Engineering at Brno University of Technology, particularly to Marion Mittelmaier and Marina Ronzhina, who provided invaluable support with many administrative issues throughout Double Degree exchange study program and elaboration of this master's thesis.

Contents

1 Introduction

Decoding genetic information and uncovering molecular mechanisms which enable this information to be stored, utilised and passed down to the next generations of cells is undoubtedly one of the greatest feats of molecular biology. These processes represent the very essence of physiological functions in cells and their disruption may lead to the rise of pathological states. Understanding cellular functions on a molecular level has therefore become of utmost importance not only for answering scientific questions but also for developing diagnostic and therapeutic procedures in the clinical environment. However, information obtained when assaying biomolecular activity is often difficult to correctly interpret in meaningful biological terms. This work therefore aspires to contribute towards a better understanding of complex data produced during experiments in the field of molecular biology, with particular focus on transcriptomic and epigenomic data in cancer research.

In this chapter, we first introduce gene expression and its molecular determinants in cells before describing experimental methods to survey transcriptome and epigenome in biological specimens. We continue by providing a short overview of commonly used computational approaches for analysing data generated by these techniques. The chapter is concluded by defining the key aims of the thesis and reviewing related previous works.

1.1 Gene expression

Proteins are the most versatile macromolecules in organisms. They are incorporated in cellular structures as simple building blocks, function as receptors or signalling agents and fulfil indispensable roles in immune response and many other physiological processes. Their structure is fully determined by the order of amino acids in the polypeptide chain, which is in turn encoded in the order of nucleotides in deoxyribonucleic acid (DNA). The information in DNA is first transferred into a complementary molecule of ribonucleic acid (RNA) in the process of transcription. Specific cellular organelles – ribosomes – then ensure translation of RNA into a sequence of amino acids which constitute a protein. This standard flow of information in biological systems was first described and published in 1958 by Francis Crick [1] and is commonly known as the central dogma of molecular biology.

Molecular mechanisms described in the central dogma facilitate gene expression, i.e. transfer of genetic information from the parts of DNA that encode proteins (or other functional molecules, e.g. various types of RNA) and its utilisation for the synthesis of these molecules. Given the importance of gene products for the metabolism of cells, it is unsurprising that precise spatiotemporal regulation of gene expression is necessary to ensure cell viability and function. Moreover, gene expression patterns are strongly reflected in cellular phenotype and therefore facilitate cell differentiation and development in multicellular organisms [2].

In single-celled prokaryotes, which do not have a cell nucleus, DNA floats freely in the cytoplasm. Transcription and translation occur almost simultaneously and the resultant proteins undergo very few (if any) additional modifications. As a result, regulation of transcription is a dominant way of prokaryotic gene regulation. In eukaryotic cells, genes are transcribed in the nucleus (where most of the cellular DNA is stored) and RNA transcripts are subsequently transported through the nuclear membrane into the cytoplasm to be translated at ribosomes (however, not all RNAs undergo translation and may exert function without coding any proteins). This procedure allows for more complex control over gene expression – apart from modification of transcriptional activity, there are other regulatory mechanisms available. For instance, most RNA transcripts in eukaryotes are subject to post-transcriptional modifications, which ensure their chemical stability during transportation and splicing of non-coding sequences (introns). The intensity of translation can be altered as well and the resultant proteins usually undergo post-translational changes to acquire their final, biologically active form [3]. The contents of this thesis, however, concern primarily the processes involved in eukaryotic transcription regulation, which will therefore be covered more thoroughly in the following chapters.

1.2 Regulation of transcription

The central role in the process of transcription is played by an enzyme which catalyses the synthesis of RNA strand based on DNA template – RNA polymerase. More precisely, there are multiple types of RNA polymerase in eukaryotes, each facilitating transcription of specific RNAs. For the sake of simplicity, however, we will refer to all these enzymes collectively in further text. For transcription to start, RNA polymerase must bind to the DNA molecule upstream of the gene to be transcribed. This process is facilitated through specialised proteins – transcription factors – which bind to DNA near the transcription start site, form so-called initiation complex and enable RNA polymerase to commence its activity [3]. While some transcription factors need to be present practically during any transcription (these will be referred to as general transcription factors), specific genes may require recruitment of additional proteins to be successfully transcribed. The following sections are meant to provide a concise overview of transcriptional regulatory mechanisms in eukaryotic cells, focusing on those which are relevant in the context of this work.

1.2.1 Regulatory elements

The term regulatory element (RE) in the context of genetics refers to a genomic region implicated in the regulation of transcription. Two classes of REs may be distinguished: *cis*- and *trans*-REs. *Cis*-REs are portions of non-coding DNA which can influence the transcription of

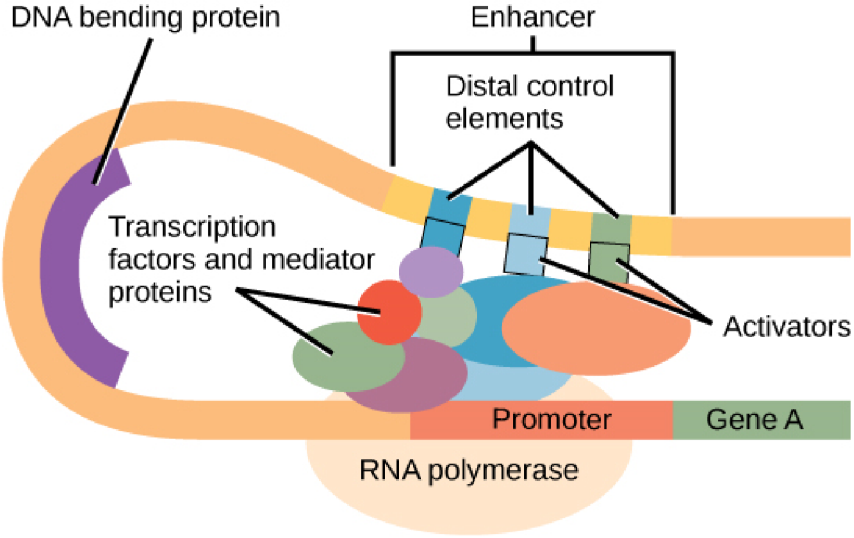


Figure 1: Transcription initiation complex, which consists of RNA polymerase and transcription factors bound to the gene promoter. Its function can be further modulated by the regulatory proteins (activators) associated with distal control elements of the enhancer sequence. Enhancers are brought close to the initiation complex thanks to DNA looping mediated by DNA bending proteins (Source: [3])

genes present in the very same DNA molecule. From the functional point of view, they most commonly serve as binding sites for transcription factors or other regulatory proteins. *Trans*-REs, on the contrary, are coding DNA sequences that encode molecules (proteins or RNA) capable of modulating transcription of genes, located possibly within a different DNA molecule than the one in which the *trans*-RE is present. The most common examples of *trans*-REs are genes for transcription factors, which can then interact with *cis*-REs and mediate intermolecular regulation of transcription [4]. In the remainder of this thesis, the term RE will be used to refer to *cis*-REs only.

A very important and commonly studied class of REs are promoters. These are non-coding sequences located upstream of the corresponding gene, usually directly adjacent or very close to the transcription start site. They contain binding sites for RNA polymerase and other proteins of the transcription initiation complex (see figure 1). Although the length and structure of promoters are gene-specific, several core structural sequences have been identified that can be found in most of the eukaryotic promoters (e.g. so-called TATA box, which acts as a binding site for specific transcription factors) [3].

Formation of the initiation complex and the activity of RNA polymerase can be additionally influenced by REs located further away (sometimes more than 10 kilobase pairs) from the transcription start site. The most prominent of these REs are called enhancers because their activity stimulates (enhances) or even enables the transcription of certain genes. Enhancers share many features with other classes of *cis*-REs, especially promoters, but it is their ability to activate transcription over long genomic distances that sets them apart [5]. Enhancers may be located upstream or downstream of promoters and they are usually composed of multiple distinct functional sequences (called distal control elements), which act as binding sites for regulatory proteins (activators) interacting with molecules of the initiation complex and stimulating the activity of RNA polymerase. These interactions are enabled by DNA looping (often facilitated by yet another group of specialised proteins) that brings the enhancer sequence close to the transcribed gene (see figure 1). [6]

Some REs have the opposite effect on transcription compared to enhancers, i.e. they inhibit (silence) gene expression and are therefore called silencers. However, the mechanism behind the regulatory influence of these REs is often very similar in principle to enhancers. Silencers contain binding sites for gene repressors – regulatory proteins which interfere with transcriptional machinery in various ways (see chapter 1.2.2) and disrupt its function.

Regardless of the type of effect which REs impose on transcription, their coordinated activity is an essential part of regulatory programs that lead to cell development and differentiation. Consequently, disruptions of these processes may often trigger the rise of severe pathology. It has therefore become of great scientific interest to map the activity of REs across the whole genome (particularly in human) in order to understand the molecular basis of gene regulation, cell development and disease. Indeed, vast integrative studies of the human genome focused on characterising REs (such as [7], [8]) have experimentally established that the activity of REs is highly specific for different cell types, developmental stages and physiological or pathological states.

1.2.2 Transcriptional regulatory proteins

The key part of transcription regulation is executed via interactions between molecules of the transcriptional machinery (RNA polymerase, general transcription factors) and specific regulatory proteins. As has been mentioned in chapter 1.2.1, the activity of these proteins is tightly connected with REs in the genome to which they specifically bind. The group of proteins that can stimulate transcription is called transcriptional activators. Their structure is usually modular, composed of DNA-binding and activation domains. The DNA-binding domain binds specifically to regulatory DNA sequences (such as promoters or enhancers) while the activation domain interacts with the components of transcriptional machinery. Interestingly, DNA-binding and activation domains may be interchanged between proteins to ensure a large variety of specific regulatory effects. Moreover, multiple activators may interact simultaneously with different components of the transcriptional machinery to ensure synergistic stimulation. One of the prototypes of eukaryotic activators is factor Sp1 (specificity protein 1), which stimulates expression of genes only in the presence of specific promoters. [6]

Transcriptional repressors, on the contrary, inhibit the process of transcription. In higher organisms (such as mammals), the function of repressors can in principle be described as passive or active. Passive repressors do not have intrinsic inhibitory activity and so their effect is achieved via blocking the function of RNA polymerase, general transcription factors or transcriptional activators. They may compete with these molecules for DNA binding sites or bind directly to the regulatory proteins, rendering them inactive [9]. For instance, some passive repressors contain the same DNA-binding domains as activator proteins but lack activation domains [6]. An example of a passive repressor is protein ICER (inducible cAMP early repressor), which is encoded by one of the splice variants of cAMP-responsive element modulator (CREM) gene [10]. In contrast, the function of active repressors is not dependent on interaction with activators but it is based on mediating chromatin structure alterations. The most prominent members of this group are proteins which recruit histone deacetylases (histone acetylation is necessary to relax tight nucleosome structure, see chapter 1.2.3), their activity being often coupled with DNA methylation [9]. For example, RE-1-silencing transcription factor (REST) functions as an active repressor of neuronal genes in non-neuronal cell types [11].

1.2.3 Chromatin structure and DNA methylation

In all eukaryotic cells, DNA is present in the cellular nucleus not as a naked molecule but associated with specialised proteins in so-called chromatin structures. The basic unit of these structures is the nucleosome, i.e. the complex of double-stranded DNA helix and small, positively charged proteins – histones. A high percentage of positively charged amino acids (arginine, lysine) in histones ensures affinity to negative charges of DNA phosphates [12]. Each nucleosome consists of a DNA strand wrapped around a histone octamer, which contains 2 molecules from each of 4 different histone types (H2A, H2B, H3, and H4). In most nuclei, there is also 1 molecule of histone H1 (linker histone) bound to the DNA as it enters the nucleosome core. This histone promotes the folding of nucleosomes into chromatin fibres, which then form tightly compressed loops and coils and are finally condensed into chromosomes [13]. The whole process is schematically illustrated in figure 2.

The way DNA molecules are packaged into chromatin influences the expression of genes. Association of DNA with histone proteins and its further condensation creates a physical barrier for molecular interactions, i.e. the more condensed DNA is, the less accessible it is to the transcriptional machinery. The positioning of nucleosomes and the level of packaging of DNA into higher-order structures is referred to as chromatin accessibility and represents a very important regulatory mechanism of practically all DNA-dependent processes, including gene transcription [14]. Chromatin accessibility is one of the key concepts in the context of this thesis.

For a gene to be actively transcribed, multiple conditions related to chromatin structure need to be satisfied. The DNA molecule has to be in a decondensed state at the gene locus, corresponding to the 11 nm chromatin fibre in figure 2. However, the DNA is still associated with histones, which poses an obstacle for transcription initiation. This inhibitory effect is eliminated by histone acetylation and the binding of two nonhistone chromosomal proteins (HMG-14 and HMG-17) to nucleosomes of actively transcribed genes. Additional regulatory proteins – nucleosome remodelling factors – may facilitate the binding of transcription factors to DNA through changing nucleosome structure. [6]

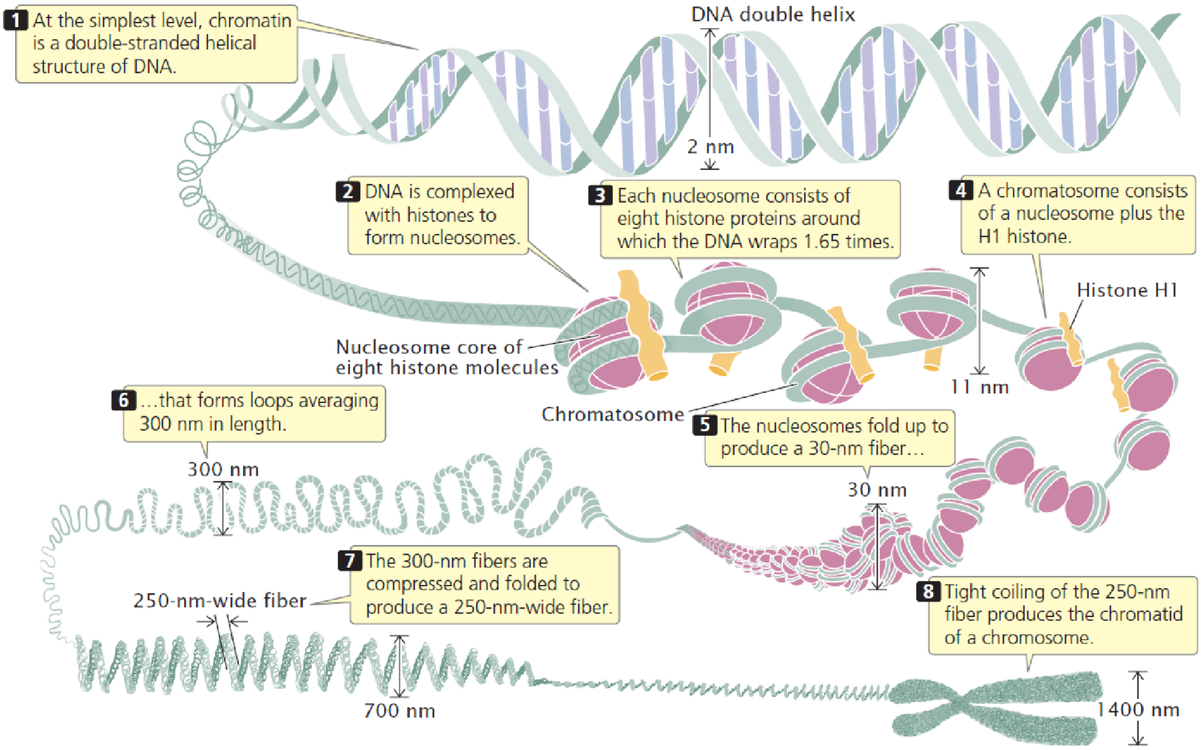

1 At the simplest level, chromatin is a double-stranded helical structure of DNA.
DNA double helix
2 nm
2 DNA is complexed with histones to form nucleosomes.
3 Each nucleosome consists of eight histone proteins around which the DNA wraps 1.65 times.
4 A chromatosome consists of a nucleosome plus the H1 histone.
Histone H1
11 nm
Nucleosome core of eight histone molecules
Chromatosome
6 …that forms loops averaging 300 nm in length.
300 nm
5 The nucleosomes fold up to produce a 30-nm fiber…
30 nm
7 The 300-nm fibers are compressed and folded to produce a 250-nm-wide fiber.
250-nm-wide fiber
700 nm
8 Tight coiling of the 250-nm fiber produces the chromatid of a chromosome.
1400 nm

Figure 2: Chromatin is composed of nucleosomes, i.e complexes of double-stranded DNA with structural histone proteins. Nucleosomes are further organised by folding and looping into more condensed fibres. (Source: [12])

Another general mechanism of transcription regulation in eukaryotes, strongly related to chromatin structure, is DNA methylation. Through the activity of DNA methyltransferase enzyme, a methyl group is added to cytosine bases of DNA strand, which are usually followed by guanine residues (forming so-called CpG dinucleotides). This chemical modification modulates gene expression via regulatory proteins that bind specifically to methylated DNA [6]. Through the silencing of specific genes, DNA methylation is involved in many important cellular processes, such as X-chromosome inactivation or genomic imprinting (the phenomenon of gene expression being dependent on whether it comes from maternal or paternal allele) [15].

1.2.4 Regulatory disruptions in cancer

Molecular mechanisms of cellular regulation are inseparably connected with cancer for it is the breakdown of these processes that causes abnormal growth and proliferation of cells. The distinguishing characteristic of cancer cells is their malignancy, i.e. the ability to invade neighbouring tissues and spread throughout the organism (metastasise), as opposed to the locally confined growth of benign tumours [16]. A cell can undergo malignant transformation after acquiring disruptions in regulatory pathways that control important cellular processes. These errors occur as a result of genetic mutations – a change of protein-coding DNA sequence may render the gene or its product inactive/hyperactive, it can result in the production of proteins with aberrant function or it may cause the protein to be expressed in inappropriate amounts. Mutations of certain genes, particularly those involved in the regulation of cell cycle, have a high potential to induce cancer – these genes are referred to as proto-oncogenes. Some proto-oncogenes encode proteins which normally act to push cells through distinct stages of the cell cycle upon receiving appropriate signals, such as cyclin D1 (CCND1) and cyclin E1 (CCNE1). Mutation or inappropriate expression of these genes may result in malignant transformation [17]. Usually, multiple such mutations need to be accumulated by the cell so it can obtain all the necessary traits that lead to malignancy - apart from the unlimited proliferation ability, these include the resistance against signalling from other cells or extracellular matrix, decreased adhesion to surrounding structures or the ability to escape apoptosis and immune system supervision [16].

Based on what has already been said about REs (see chapter 1.2.1), it is clear that not only mutations of genes themselves but also alterations of non-coding regulatory sequences may cause dysregulation of gene expression [18]. Moreover, the expression of genes can be efficiently modulated by epigenetic factors (i.e. without changing the coding DNA sequence), such as methylation or the packaging of DNA into chromatin structures (see chapter 1.2.3). Indeed, it is now firmly established that epigenetic regulation plays an important role in cancer development [19]. Consequently, mapping gene expression and the activity of REs provides a valuable insight into molecular background of physiological as well as pathological cell states, especially in the context of oncological diseases.

1.3 Transcriptomic and epigenomic assays

The study of cellular processes on a molecular level has only been made possible by sophisticated experimental techniques. In genetics, methods for the sequencing of nucleic acids brought a breakthrough as they enabled researchers to uncover how exactly genetic information is stored and expressed. However, the sheer amounts of data to be experimentally obtained and then processed had long posed a severe limitation of these techniques, which urged for the development of more and more efficient sequencing approaches.

A common way of quantifying the expression of genes is to assay cellular transcriptome, i.e. the collection of all RNA transcripts in a cell. For global gene expression profiling, microarrays have been used for more than 20 years [20, 21]. This technique employs short nucleic acid probes, covalently bound to a glass substrate, which are hybridised with fluorescently labelled target sequences. The array is then scanned (e.g. with a laser-scanning microscope) and the intensity of fluorescence corresponds to the amount of hybridised RNA. In recent years, however, microarrays have been gradually replaced by methods utilising next-generation sequencing (NGS) thanks to their greater flexibility and accuracy and eventually lower costs [22]. These high-throughput techniques, which enable parallel sequencing of large amounts of nucleic acid fragments, have revolutionised genomic, transcriptomic and epigenomic studies. The following chapters describe two types of NGS-based methods, which are critical to the topic of this thesis. It is worth noting that the principles to be introduced apply generally to the processing of biological material from a large number of cells (so-called “bulk analysis”). However, recent technological advances enabled all of the described methods to be implemented also in single-cell versions, in which nucleic acids from individual cells are extracted and analysed [23].

1.3.1 RNA-seq

For global transcriptome profiling, RNA-seq is nowadays a standard experimental approach. Thanks to the immense progress in NGS technologies, RNA-seq can provide genome-wide expression profiles with single-base resolution and low levels of background noise, requiring only a small amount of RNA [24]. Moreover, RNA-seq is not dependent on *a priori* knowledge of target sequences (which is needed to design specific probes for microarray hybridisation) and is therefore a suitable method for exploring unannotated transcription regions or novel RNA splice variants [25].

The first step of the experimental protocol in RNA-seq is the isolation of RNA molecules from cells. As the majority (more than 90 %) of cellular RNA consists of ribosomal RNA (rRNA), which is not informative in terms of gene expression, it is usually desirable to filter RNA samples before further processing. Today, there are multiple techniques available either for selective messenger RNA (mRNA) enrichment (usually utilising the presence of polyadenylated tails on most mRNA molecules) or selective depletion of rRNA in samples [26]. The next phase is reverse transcription of RNA needed to create double-stranded molecules of complementary DNA (cDNA), which are suitable for subsequent amplification and sequencing. Unlike small RNAs (such as microRNAs, Piwi-interacting RNAs, small interfering RNAs and others), which can be sequenced as a whole, larger molecules need to be broken into shorter fragments (most commonly via enzymatic digestion or ultrasound application) to be sequenced by NGS methods. After fragmentation, the sample is purified to keep only the fragments of appropriate length (this parameter depends on the specific NGS technique used but is commonly in the order of a few hundreds of base pairs) [24].

Following sample preparation, NGS takes place, simultaneously processing a large number of cDNA fragments. Although the basic principle of most available NGS procedures is somewhat similar, there are substantial differences regarding experimental implementations of individual manufacturers. However, it is out of the scope of this work to discuss these technical nuances.

Once the “laboratory” phase of RNA-seq has been completed and the sequences of cDNA fragments (so-called “reads”) have been obtained, appropriate data processing must be employed in order to transform this raw information into a meaningful and interpretable form [22]. Frequently, the first step is filtering out low-quality reads, for example based on their length or the content of unidentified nucleotides. Next, individual reads need to be matched to the reference genome of the organism from which the processed sample originated (there are also possibilities to assemble transcriptomes *de novo* if the reference genome is unknown, although this is now less common). The sheer volume of data from sequencing methods had long made this step difficult in terms of computational requirements and the task is further complicated by the problematic alignment of repetitive sequences (which constitute almost 50 % of the human genome) [26]. However, the development of effective bioinformatic algorithms, as well as improved availability of computational power in recent years, have helped overcome these obstacles. Finally, raw read counts for individual genome positions are usually transformed (by aggregation with respect to known gene coordinates) and normalised in order to obtain more interpretable quantification of gene expression.

1.3.2 DNase-seq and ATAC-seq

Similarly to transcriptome profiling, NGS has remarkably broadened the possibilities in epigenomic assaying as well. Among these assays belong methods for quantification of chromatin accessibility, which are in principle based either on enzymatic methylation (e.g. NOMe-seq) or cleavage of DNA molecules (e.g. DNase-seq) [27]. In this chapter, two such techniques will be introduced that play a central role in the context of this thesis.

Nucleosome-free genomic regions are more susceptible to enzymatic cleavage by deoxyribonuclease I (DNase) [28]. These DNase hypersensitive sites (DHSs) had been shown to correspond mainly to REs of the genome long before the advent of NGS [29]. However, it was massively parallel sequencing that finally allowed for studying DHSs (and therefore chromatin accessibility) genome-wide [28, 30]. These were some of the first performed DNase-seq experiments, in which NGS was utilised to sequence short DNA fragments produced by cleavage via DNase. DNA is cut predominantly in the regions of open chromatin which can therefore be identified as the positions with increased read counts after mapping sequence reads to the reference genome.

More recently, a technique for quantification of chromatin accessibility called assay for transposase-accessible chromatin using sequencing (ATAC-seq) was introduced in [31]. Its experimental protocol is in a simplified form illustrated in figure 3. Conceptually similar to DNase-seq, ATAC-seq utilises a different molecule for which access to DNA is interrogated – hyperactive Tn5 transposase. This enzyme can cleave double-stranded DNA while simultaneously adding short sequencing adaptors (tags) to the ends of the produced fragments. The process is much more likely to occur in the regions of open (i.e. nucleosome-depleted) chromatin. Specific tags are used for purification and PCR amplification of DNA fragments, which are subsequently sequenced and mapped to the reference genome. Peaks in the resultant profiles denote the regions of accessible chromatin.

Very quick adoption of ATAC-seq in the scientific community was caused primarily by a considerably faster and easier-to-perform experimental protocol compared to DNase-seq, as well as lower requirements for the amount of genetic material [27]. Moreover, ATAC-seq accessibility measurements are highly consistent with the results of DNase-seq experiments, both in terms of data quality and capturing regulatory information [31], [33]. Consequently, chromatin accessibility profiles generated by ATAC-seq and DNase-seq may serve as suitable and mutually comparable proxies for evaluation of the activity of genomic regions, particularly REs.

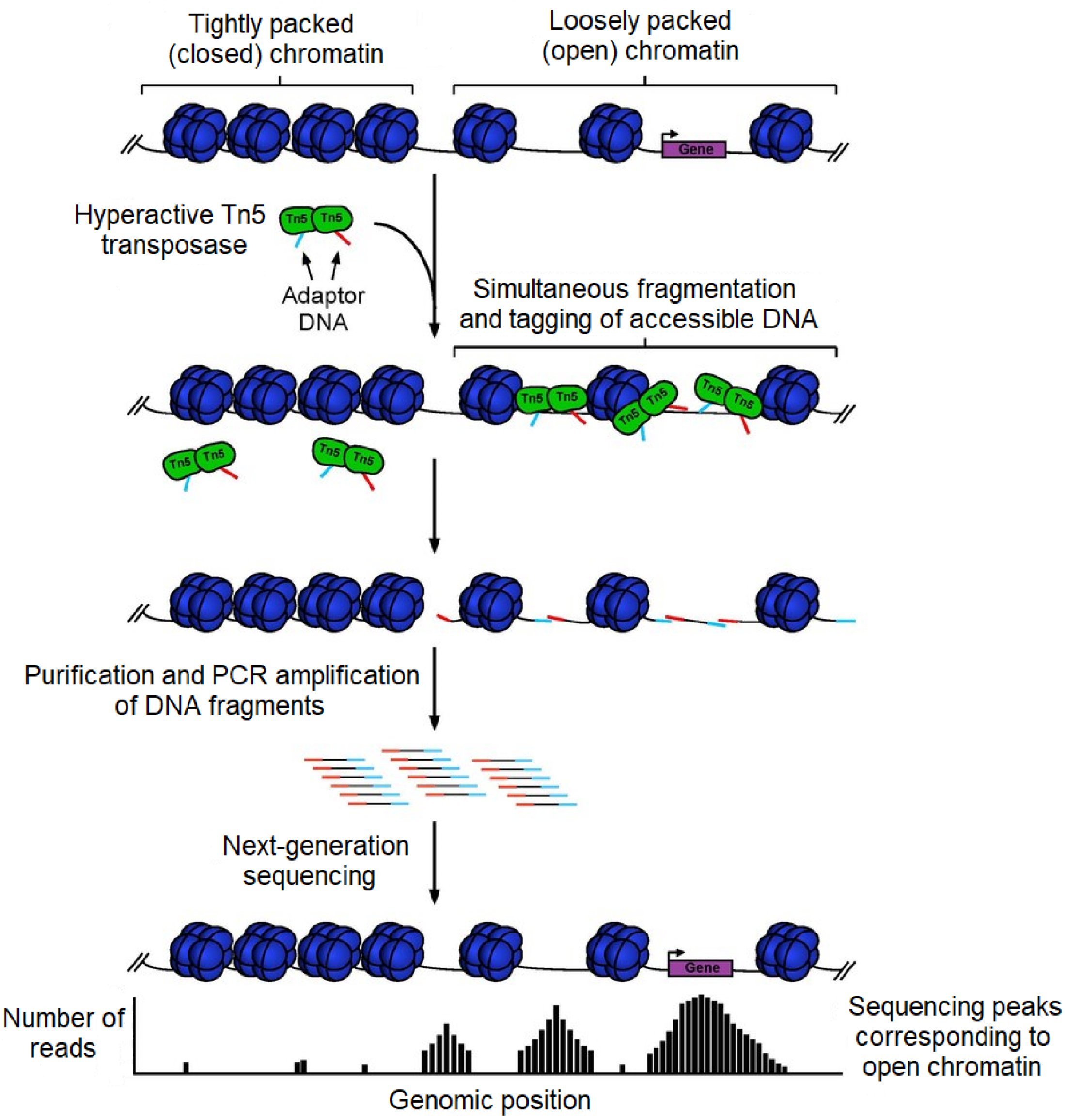

Tightly packed (closed) chromatin
Loosely packed (open) chromatin
Gene
Hyperactive Tn5 transposase
Tn5
Adaptor DNA
Simultaneous fragmentation and tagging of accessible DNA
Purification and PCR amplification of DNA fragments
Next-generation sequencing
Number of reads
Sequencing peaks corresponding to open chromatin
Genomic position

Figure 3: A schematic illustration of ATAC-seq experimental protocol. Hyperactive Tn5 transposase is used to cleave DNA and tag the ends of fragments with sequencing adaptors. After purification and amplification, DNA fragments are rapidly sequenced and mapped to the reference genome. Genomic positions with increased read counts correspond to the regions of accessible chromatin. (Adapted from: [32])

1.4 Analytical methods

As introduced in the previous chapters, NGS technologies produce vast amounts of data. In this section, we review selected general-purpose and bespoke techniques for the analysis of high-dimensional biological datasets.

1.4.1 Standardisation and normalisation

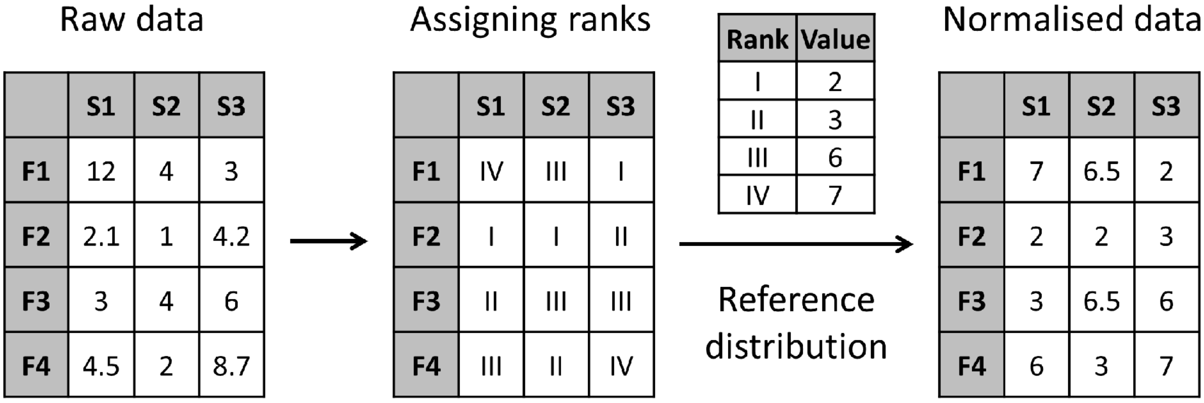
When comparing high-dimensional data, one of the biggest challenges is the elimination of technical variation. This term denotes all the variation in data that is caused by differences in experimental setup and conditions and which can obscure the biological variability of interest. Numerous general-purpose and data-type-specific normalisation techniques have been developed to address this problem [34, 35, 36]. It has been shown that the application of these procedures may critically influence the outcome of downstream analyses of high-throughput data [37, 38] and therefore particular attention should be paid to the choice of suitable normalisation technique.

A common approach to tackle inconsistent range and scaling of data values is to compute the corresponding standard scores (also called z-scores). Each original value x is converted into a standard score z through

$$z = \frac{x - \mu}{\sigma}, \tag{1}$$

based on the mean μ and standard deviation σ of the values to be transformed. For the sake of simplicity, this operation will be hereafter referred to as standardisation (although in statistics, the term standardisation may cover several different scaling methods). Standardisation can be performed on the whole dataset or it can be applied either on individual sample vectors or feature-wise (e.g. separately for each gene in gene expression profiles). Although a very simple technique, standardisation is regularly used as a part of pre-processing pipeline for many advanced analyses (particularly in machine learning).

The specific character of outputs provided by high-throughput genomic, transcriptomic and epigenomic assays has urged for the development of specialised normalisation methods, e.g. quantile normalisation (QN). Although QN was originally introduced for gene expression microarrays [34, 39, 40], it has since been successfully used to remove technical variation in sequencing data as well [41]. The basic principle of QN is illustrated in figure 4. In the first step, original data values for individual samples are sorted and assigned ranks. Subsequently, values from a pre-defined reference distribution are used to replace the original values with the corresponding rank. This operation forces each sample vector to have the same (reference) distribution. Such a strong assumption about statistical properties of processed data may be justified in some biomedical applications, e.g. for gene expression studies where only a minority of targeted genes is expected to be differentially expressed between samples [41].



Raw data

	S1	S2	S3
F1	12	4	3
F2	2.1	1	4.2
F3	3	4	6
F4	4.5	2	8.7

Assigning ranks

	S1	S2	S3
F1	IV	III	I
F2	I	I	II
F3	II	III	III
F4	III	II	IV

Reference distribution

Rank	Value
I	2
II	3
III	6
IV	7

Normalised data

	S1	S2	S3
F1	7	6.5	2
F2	2	2	3
F3	3	6.5	6
F4	6	3	7

Figure 4: A schematic depiction of quantile normalisation principle. Columns of the data matrix (S1–S2) represent samples, rows F1–F4 represent features. First, each sample vector is sorted and based on the order, raw data values are assigned ranks (here, ascending order is used). Normalisation is then performed by replacing the original values with values which have the corresponding rank in the defined reference distribution.

From the practical point of view, the construction of reference distribution for QN is an important issue. It can either be created based on already existing and validated data from similar studies or it may be derived directly from the analysed dataset. In the latter case, reference is usually calculated as the mean of corresponding quantiles across all samples. In figure 4, this would simply require computing the mean of each row of the data matrix after value sorting – resultant numbers would then constitute the reference distribution. Another important technical detail concerns how the normalisation of the same values in sample vectors is treated (particularly for datasets with a high content of identical numbers, usually zeros). An example of such tied ranks can be seen in figure 4, where sample S2 has the same value (and therefore the same rank) defined for features F1 and F3. These values could be both replaced by the 3rd entry in the reference (i.e. 6) or by the mean of the 3rd and the 4th entry (as it is done in the toy example). Other metrics (such as median) come into consideration as well, the final choice should be made based on data specifications.

1.4.2 Dimensionality reduction

The primary output of methods for assaying transcriptome or chromatin structure are numeric vectors for individual samples, where the numbers represent the levels of gene expression or chromatin accessibility, respectively. For genome-scale studies, these levels are usually determined for a large amount of genomic regions (i.e. features), ranging from tens of thousands of genes (e.g. for RNA-seq) up to several millions of REs (for chromatin accessibility assays). The analysis of such high-dimensional data is often problematic as they cannot be visualised straightforwardly and because advanced analyses may become computationally infeasible for a large number of features. Dimensionality reduction techniques offer solutions to tackle these problems.

Principal component analysis

Being one of the first and most commonly employed feature reduction methods, principal component analysis (PCA) owes its success to relatively simple mathematical formulation and interpretation yet high efficiency in many applications. Through PCA, a dataset that contains mutually correlated variables (features) is transformed into a different coordinate system, in which a new set of uncorrelated variables, called principal components (PCs), is defined so that they capture most of the variation present in the original features [42].

Let matrix $\boldsymbol{X}$ of dimensions $M \times N$ (M rows, N columns) represent a dataset with M observations described by N features. To derive PCs, which are in fact linear combinations of the original features, $N \times N$ covariance matrix $\boldsymbol{C}$ first needs to be computed as

$$\boldsymbol{C} = \frac{1}{M}\boldsymbol{Y}^{\intercal}\boldsymbol{Y}, \tag{2}$$

where $\boldsymbol{Y}$ is the matrix of centred data in which the mean of each feature (i.e. column of $\boldsymbol{X}$) was subtracted from all the respective feature values. If the data are additionally scaled by being divided by the standard deviations of features (that is, standardisation is performed feature-wise as described in chapter 1.4.1), the matrix computed according to equation 2 is called the correlation matrix $\boldsymbol{R}$ (which is *de facto* a normalised version of $\boldsymbol{C}$). Both matrices can be used for further computations with slight differences in interpretation of the result, correlation being a common default in many PCA implementations [43]. In the remainder of this chapter, the usage of the correlation matrix will be assumed with practically all the principles being easily transferable to the case of using the covariance matrix.

PCs are found through eigendecomposition of the correlation matrix. This operation is defined for square matrices and results in obtaining eigenvalues and eigenvectors of the matrix. An eigenvector of $N \times N$ matrix $\boldsymbol{R}$ is defined as each non-zero vector $\boldsymbol{v}$ that satisfies the condition

$$\boldsymbol{R}\boldsymbol{v} = \lambda\boldsymbol{v}, \tag{3}$$

where λ is the scalar eigenvalue corresponding to eigenvector $\boldsymbol{v}$. This can be rewritten as

$$(\boldsymbol{R} - \lambda\boldsymbol{I})\boldsymbol{v} = \boldsymbol{0}, \tag{4}$$

with $\boldsymbol{I}$ being $N \times N$ identity matrix and $\boldsymbol{0}$ the zero vector. Equation 4 has a non-zero solution $\boldsymbol{v}$ only if the determinant of matrix $(\boldsymbol{R} - \lambda\boldsymbol{I})$ is zero:

$$|\boldsymbol{R} - \lambda\boldsymbol{I}| = 0. \tag{5}$$

The N roots of this so-called characteristic equation are the eigenvalues λ_n of matrix $\boldsymbol{R}$, each of them associated with eigenvector $\boldsymbol{v}_n$. Strictly speaking, there is an infinite amount of eigenvectors corresponding to each eigenvalue as any scalar multiple of $\boldsymbol{v}_n$ satisfies equation 4 and is therefore an eigenvector as well. However, the requirement of unit length is usually imposed on eigenvectors so that

$$\boldsymbol{v}^{\mathsf{T}}\boldsymbol{v} = 1, \tag{6}$$

which ensures unambiguity. Moreover, all pairs of eigenvectors with different associated eigenvalues are orthogonal (i.e. their dot product is zero). [44]

Special characteristics of correlation matrices (they are symmetric and contain only real positive numbers) ensure that their eigendecomposition always exists with real eigenvectors and real positive eigenvalues [44]. These properties become important when interpreting the meaning of eigenvalues and eigenvectors in the context of PCA. Individual PCs are effectively defined by the corresponding eigenvectors, whose elements represent the coefficients (also called loadings) of the linear combination used to transform original features into PCs. Thus, they also determine the contribution of individual original features to each PC. It is worth noting the mutual orthogonality of eigenvectors ensures that each PC is uncorrelated with all the other PCs.

The sum of all N eigenvalues represents the total variance contained in the transformed data with each eigenvalue expressing the proportion of this total variance captured (explained) by the corresponding PC. Usually, a high percentage of overall variance is covered by the first K PCs (when ordered according to the eigenvalues from the highest to the lowest), where $K << N$. This makes it possible to choose only the K most important PCs to represent the original features, achieving the desired dimensionality reduction [42]. There are various approaches to determining the optimal number of retained PCs, for example keeping the PCs which explain a certain percentage of the total variance.

The last step of PCA is data transformation itself, which is performed as matrix multiplication. The K chosen eigenvectors usually constitute the columns of the transformation matrix $\boldsymbol{V}$ (also called the loading matrix), which is then used to multiply the original data matrix $\boldsymbol{X}$:

$$\boldsymbol{X}\boldsymbol{V} = \boldsymbol{X}'. \tag{7}$$

The transformed data matrix $\boldsymbol{X}'$ contains M observations described by the new set of K variables - the values of these new artificial features are referred to as scores (or factor scores). Geometrically, PCA performs a coordinate system transformation where factor scores may be interpreted as the projections of original data points onto a new set of axes - principal components [43].

t-distributed stochastic neighbor embedding

For complex datasets which contain an underlying non-linear structure in low-dimensional representation, linear feature reduction techniques such as PCA may not be able to faithfully capture the relationships between data points. In these cases, usage of a non-linear transformation may be beneficial for dimensionality reduction and visualisation of data. One of such methods is t-distributed stochastic neighbor embedding (t-SNE) [45], which can be used to visualise high-dimensional data in two- or three-dimensional space by converting the original data points into a matrix of their pairwise similarities.

The central idea behind t-SNE is derived from stochastic neighbor embedding (SNE) algorithm [46], in which similarities between data points are represented by conditional probabilities that two points would be chosen as neighbours, provided that the probability of such selection is proportional to a Gaussian probability density function centred at one of the data points. These probabilities are computed across all pairs of data points as well as the pairs of their respective low-dimensional representations (so-called map points). The model represents high-dimensional relationships correctly if the distributions of conditional probabilities for individual data points are equal to such distributions for the corresponding map points. To measure the similarity between these distributions, Kullback-Leibler divergence [47] is employed as a metric. At the same time, it constitutes the cost function to be optimised (i.e. minimised) during data transformation. One of the critical parameters of the algorithm to be set by users is called perplexity, which may be interpreted as a number of effective nearest neighbours of each data point (i.e. those neighbours which will significantly contribute to the computation of conditional probability distributions) [45].

In t-SNE, multiple alterations are introduced compared to SNE algorithm which address some of its major shortcomings, both in terms of visualisation quality and computational efficiency. The most prominent difference is perhaps the replacement of Gaussian with Student's *t*-distribution to compute similarities between points in low-dimensional space (hence t-distributed SNE). However, for a detailed description of all the implemented changes, which is beyond the scope of this introductory chapter, please refer to the source articles [45] and [46], where precise mathematical formulations of the presented problems can be found. Here, we will conclude by stating that t-SNE has been shown to excel (with appropriate parameter settings) at capturing the local structure of high-dimensional data while also preserving global patterns. Hence, it has become a standard visualisation technique employed with particular success in exploratory, unsupervised analyses of high-throughput sequencing data [48].

Uniform manifold approximation and projection

More recently, uniform manifold approximation and projection (UMAP) was introduced as another non-linear technique for dimensionality reduction [49]. It belongs to the class of manifold learning algorithms, which aim to uncover the intrinsic low-dimensional geometric structure hidden in high-dimensional observations. Thus, the assumption is that these data points lie on or near a low-dimensional manifold, reflecting the process of data generation which often has relatively few degrees of freedom compared to the number of features that describe the data [50].

In UMAP, manifold approximations are utilised to construct topological representations of both the high-dimensional data and their low-dimensional embedding. The layout of the representation in the low-dimensional space is then optimised in order to minimise the cross-entropy between the two topological representations [49]. For detailed theoretical foundations and mathematical formulations of the problem, see [49]. Although UMAP is quite a novel method, its usage in this work is justified by quickly acquired popularity and widespread use within (not only) bioinformatic data analyses, such as [51] or [52]. In these and other similar studies, UMAP has been proven to yield comparable visualisation results as t-SNE, possibly being even superior in preserving global data structure. It also comes with the benefit of faster computations of the outputs than t-SNE and is not restricted in terms of embedding dimensions (t-SNE is only able to provide 2-D or 3-D representations) [49].

1.4.3 Machine learning models for classification

Experiments in molecular biology, particularly in the studies of genetic information, often produce highly complex datasets where relationships between data points or innate data structures are inconceivable by the human mind. In these cases, machine learning (ML) algorithms may come into play as they are capable of uncovering these hidden patterns, providing the human expert with outputs that can be more easily interpreted [53]. The model is said to be trained on the data (so-called training dataset) so that the information extracted during this process can be generalised when making predictions about new (unseen) samples. Importantly, ML algorithms are not explicitly programmed to make decisions according to a fixed set of rules or parameters, they are automatically adjusted based on the contents of the training dataset (i.e. they “learn” from the data).

Formally, ML models try to approximate an unknown transformation function f, which converts input data $\boldsymbol{X}$ (usually, but not necessarily, a vector or matrix of numeric features) into the corresponding output $\boldsymbol{Y}$:

$$\boldsymbol{Y} = f(\boldsymbol{X}). \tag{8}$$

The output $\boldsymbol{Y}$ can be expressed in various forms as well (a matrix, a vector, a single number, alternatively a category label etc.), depending on the task at hand. One of the main challenges in ML is to train models which are not only able to transform the training data correctly but also react adequately when presented with new inputs, not used during the training phase. This generalisation ability is the key characteristic of ML models.

In principle, there are two basic approaches to training models in ML. In supervised learning, each data point from the training dataset has the corresponding desired (real) output defined.

Iteratively, the model is adapted to improve the approximation of transformation function f so that the predicted outputs are getting closer to the desired outputs. This configuration also allows for a relatively straightforward assessment of model's generalisation ability by using so-called testing data. Ideally, the testing dataset is independent of the training data (e.g. obtained from a different source) but more commonly, as such data are often unavailable, it is created as a part of the original input dataset. The testing data are not used during training so they can be presented to the already trained model as new, unseen samples. The quality of the resultant predictions is then evaluated through various performance metrics, chosen according to the output type. Typical applications of supervised ML are classification and regression tasks.

In unsupervised learning, the desired outputs are unknown. Therefore, the model is trained solely on the input dataset in order to extract its intrinsic structures and relationships between data points. Performance evaluation of such models is complicated by the fact that there is no reference with which the obtained results could be compared. Unsupervised learning is commonly used for data clustering. However, the application domain of ML methods has expanded rapidly in recent years with a large amount of novel approaches emerging in the process. Therefore, in the remainder of the chapter, only the techniques utilised in this thesis are introduced in more detail. They belong to the category of supervised ML classifiers.

Logistic regression

Logistic regression is a linear model used primarily for the purpose of binary classification. As stated previously, supervised ML models are tasked with estimating an unknown transformation function f between the input and output variables. The approximation of this function is commonly referred to as the hypothesis and will be denoted h in further text. For linear regression models, the hypothesis is a linear function of the inputs. Formally, if each training example is represented by a vector $\boldsymbol{x}$ of n elements, the hypothesis $h_{\boldsymbol{w}}(\boldsymbol{x})$ has the form

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = w_0 + w_1x_1 + ... + w_nx_n = w_0 + \sum_{i=1}^{n} w_ix_i, \tag{9}$$

where $\boldsymbol{w}$ is the vector of $n+1$ weights and the term w_0 is called the intercept. If an artificial input feature $x_0 = 1$ is defined, the hypothesis can be written simply as the dot product of the weights and the input vector:

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} = \sum_{i=0}^{n} w_ix_i. \tag{10}$$

The equation $h_{\boldsymbol{w}}(\boldsymbol{x}) = 0$ then defines a line in 2-D feature space (i.e. when $n = 2$), a plane in 3-D space and a hyperplane in the general case of $n > 3$ dimensions. It is the aim of binary classification to find such a set of weights $\boldsymbol{w}$ that the corresponding hyperplane constitutes a

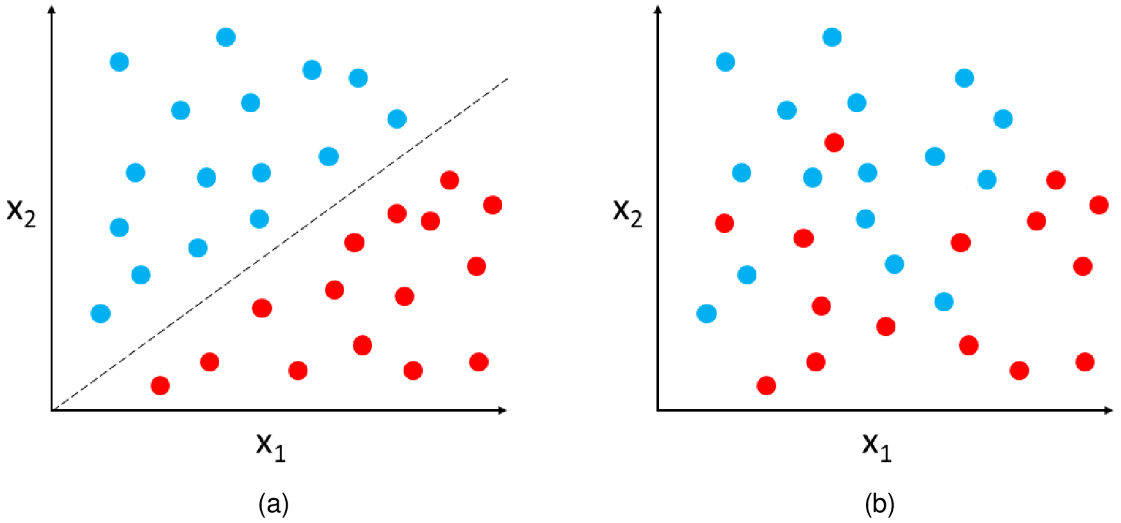


Figure 5: Figure a) shows an example of linearly separable dataset in 2-D space, where each observation is described by features x_1 and x_2. The dataset consists of data points from two classes (blue and red dots) and the dashed line represents one of possible decision boundaries which can separate these classes. For the dataset in figure b), no such linear separator can be found and the data are therefore linearly inseparable.

decision boundary which separates the two classes of data points. If such linear separator exists, the data are referred to as linearly separable [54]. See figure 5 for the illustration of linear separability.

However, the output of the hypothesis defined in equation 10 is not suitable for direct use in classification as it can be any real number. For binary classification tasks, the required output is usually the assignment of label 1 or 0 expressing whether the example belongs or does not belong to the positive (i.e. labelled as 1) class. In the simplest case, this can be achieved by thresholding the hypothesis:

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{w} \cdot \boldsymbol{x} \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Geometrically, the thresholding can be interpreted as assigning a binary label to the example by determining on which side of the decision boundary it lies in the feature space. However, two main problems arise with such an approach. Firstly, the hypothesis becomes a discontinuous (and therefore not differentiable) function, which complicates the learning of the model (i.e. the process of finding optimal weights $\boldsymbol{w}$ based on the training examples). Secondly, the predictions are always completely confident, even for the samples that lie very close to the boundary (consequently, the reliability of predictions cannot be assessed). Logistic regression resolves these shortcomings by “softening” the hard threshold [54]. For this purpose, the logistic function is utilised to construct the hypothesis, which is then defined as

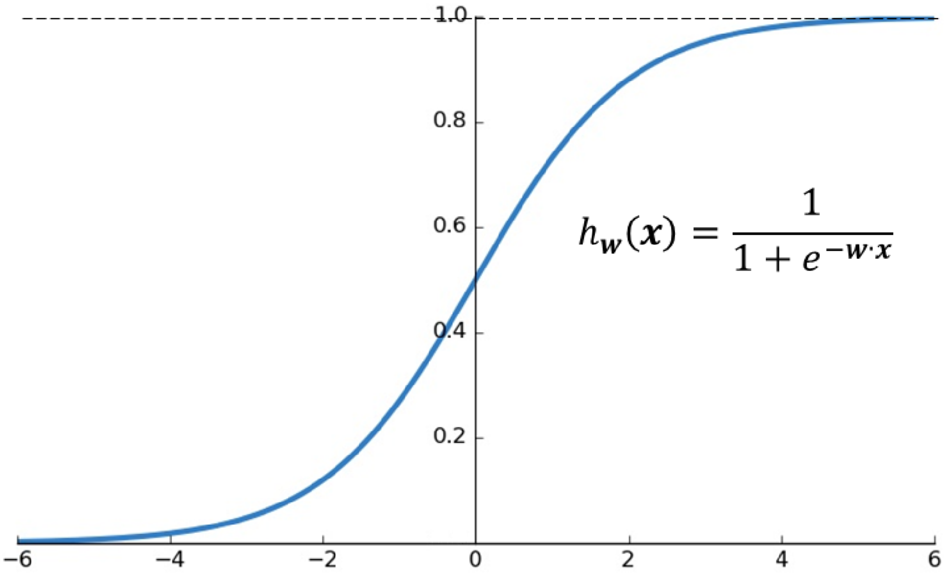


Figure 6: The logistic function is used to define hypothesis $h_{\boldsymbol{w}}(\boldsymbol{x})$ for logistic regression.

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}. \tag{12}$$

As can be seen in figure 6, the logistic function always outputs values in the range from 0 to 1. If the input data point lies exactly on the decision boundary (i.e. when $\boldsymbol{w}\cdot\boldsymbol{x} = 0$), the output is 0.5 and approaches 0 or 1 as $\boldsymbol{w}\cdot\boldsymbol{x}$ decreases or increases when moving away from the boundary. Thanks to these properties, a prediction made by logistic regression may be interpreted as the probability with which the example belongs to the positive class. If desired, the probability estimate may be thresholded to obtain binary output.

Estimating the set of weights for logistic regression based on the training examples is an optimisation task, which is usually solved by using some of the well-established algorithms with efficient implementations, such as the method of least squares, (stochastic) gradient descent or maximum likelihood estimation [55]. Mathematical foundations of these techniques, however, are not of central importance in the context of this thesis.

Support vector machines

The basic concept of classification through the support vector machine (SVM) framework is constructing a maximum margin separator. The idea is illustrated in figure 7 on a toy, linearly separable 2-D dataset. Figure 7a depicts two different decision boundaries, each of them correctly separating all of the training examples. However, one can intuitively assess that boundary 1 is not as good as boundary 2 because it lies too close to some of the training data points. In such case, the new examples may fall on the wrong side of the boundary (i.e. be misclassified)

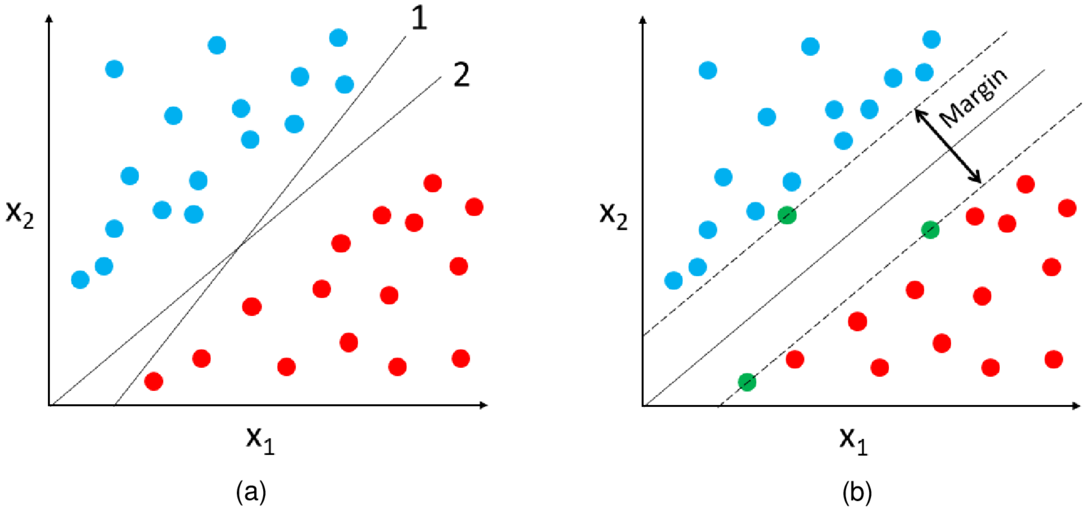


Figure 7: The concept of maximum margin separator. Placing the decision boundary as far away from all the training data points as possible is ensured by maximisation of the margin. Such separator is then expected to have a lower generalisation error when presented with new examples, compared to other possible separators (such as boundary 1 in figure a)). The training data points with the lowest distance to the separating hyperplane are called support vectors and are marked with green colour in figure b).

much more easily than when the separator is as far away from all the training examples as possible. SVMs formalise this intuition by defining the margin, which is twice the distance between the separating hyperplane (in a general case of n-dimensional feature space) and the nearest training example [54]. In figure 7b, the margin is the width of the lane delimited by the two dashed lines with the decision boundary lying in its centre. By maximising the margin, SVMs are able to decrease generalisation error, i.e. improve the accuracy when classifying unseen samples.

Using a similar notation as in the previous section about logistic regression, the separator (i.e. a hyperplane) can be defined as

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0, \tag{13}$$

where $\boldsymbol{x}$ is the feature vector and $\boldsymbol{w}$ is the vector of weights. The only difference compared to the previous notation is that the intercept is now not included in $\boldsymbol{w}$ but stands separately as the term b. If the training data are centred and standardised (see chapter 1.4.1), which is a common pre-processing step before training ML models, the hyperplanes delimiting the margin (see the dashed lines in figure 7b for an example) are defined as

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 1, \boldsymbol{w} \cdot \boldsymbol{x} + b = -1. \tag{14}$$

The distance between these hyperplanes (i.e. the margin) is then $\frac{2}{||\boldsymbol{w}||}$. Thus, it is evident that in order to maximise the margin it is necessary to minimise the magnitude of vector $\boldsymbol{w}$, i.e. the squared sum of the weights. At the same time, additional constraints have to be applied during optimisation so that none of the m training examples is allowed to lie within the margin:

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1, \forall i = 1, 2, ..., m. \tag{15}$$

Here, $\boldsymbol{x}_i$ are the training vectors with the corresponding binary class labels $y_i = 1$ or $y_i = -1$. Due to practical benefits, the stated optimisation problem is usually solved by utilising its dual representation, which will not be introduced here for the sake of brevity. Importantly, the optimised cost function is convex and therefore has a single global optimum, which corresponds to the parameters ($\boldsymbol{w}$, b) of the maximum margin separator. Such separator is fully determined by the support vectors, i.e. the data points closest to the separating hyperplane. [54]

If the data are not linearly separable (see figure 5b for an example of such data), a maximum margin classifier as described above will not work because it does not allow any misclassifications of the training examples. However, as the real data can hardly ever be perfectly separated due to the presence of noise, it is usually beneficial to accept some amount of errors during training to obtain generally more robust predictions. This is achieved by removing the constraint in equation 15 and adding a special term into the cost function that penalises misclassified samples proportionally to their distance from the separating hyperplane. Such a model is then referred to as a soft margin classifier. [55]

Besides, the SVM framework can be adjusted to learn non-linear decision boundaries through the utilisation of so-called kernels. In fact, the aforementioned support vector classifiers are often referred to as SVMs with linear kernel. The idea behind non-linear SVMs is the transformation of data from their original feature domain (where they are linearly inseparable) into a high-dimensional space in which a separating hyperplane can be found. When mapped back to the feature space of a lower dimension, this hyperplane will become a non-linear decision boundary.

The data transformation itself is performed by utilising kernel functions. Each kernel function corresponds to a particular feature space into which it maps the input data. However, not every function can be used as a kernel – the group of acceptable functions is defined by Mercer's theorem [54]. From the computational point of view, it is an important feature of non-linear SVMs that individual training data points do not actually have to be transformed into a high-dimensional domain. This is because the optimisation task is based on computing dot products of the pairs of input vectors and these dot products can be mapped directly to the high-dimensional space without prior transformation of individual vectors. A kernel function is simply evaluated for each pair of input vectors to achieve such mapping. This so-called kernel trick brings a remarkable improvement of computational efficiency for non-linear SVMs [55].

Finally, one of the most useful kernel functions will be introduced which maps the data into a space with the infinite number of dimensions (therefore, it would not even be possible to perform the mapping without the kernel trick). For a pair of input vectors $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and a positive constant γ, the function

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2} \tag{16}$$

is called the radial basis function (RBF) kernel or the Gaussian kernel. It can be seen that if the Euclidean distance between input vectors in the feature space is large, the output of the RBF kernel becomes a very small number and vice versa. This ensures local behaviour of the kernel because the predictions for new samples are primarily influenced by the training examples from a certain neighbourhood of this sample [55]. The size of the neighbourhood is controlled by adjusting parameter γ.

1.5 Aims of the thesis

This work aims to contribute towards better understanding and interpretation of complex data produced by the experimental techniques that query cellular functions on a molecular level. Specifically, two distinct data sources will be considered – gene expression and chromatin accessibility profiling experiments, in which large amounts of data have been generated and made publicly available. The goal is to extract patterns from these data so they can be subsequently applied to provide an insight into the biology (and pathology) of newly examined cells, with particular focus on cancer development.

For chromatin accessibility, a comprehensive data source first needs to be created which integrates information scattered across studies and databases. For gene expression profiles, multiple integrated resources are already available. Due to the inherent heterogeneity of the collected data, it is necessary to employ pre-processing steps aimed at reducing technical variation contained in the datasets. Subsequently, unsupervised analysis comprising mainly feature reduction methods will be performed to visualise the data and assess the effects of normalisation. Moreover, good quality annotations (i.e. labels) need to be defined for individual samples as these are essential for the training of supervised ML models. To this end, a semi-automated framework will be used to refine sample metadata, both in terms of completeness and consistency. The assembly of an integrated and well-annotated chromatin accessibility dataset is important not only for the purposes of this work but should provide a re-usable resource also for further research as, to the very best of our knowledge, there is no comprehensive collection of public chromatin accessibility data available so far.

Once the quantitative data and the corresponding annotations have been prepared, ML models will be trained to extract relationships between samples based on similarities of their gene expression patterns or chromatin accessibility landscapes. Supervised ML classifiers will be used for this task and their performance will be evaluated through standard metrics and testing strategy. Finally, the constructed models can be applied for classification of new (unseen) samples and help to uncover biological relationships which may not have been obvious at first sight. This translation of experimental results into interpretable terms is necessary in every scientific study and becomes especially important when working with data from molecular assays due to their immense complexity. The analysis of single-cell data (produced by methods assaying biological material from individual cells) may be particularly problematic as the examined cells usually come from a heterogeneous tissue and therefore their properties are unknown before the experiment. A tool that helps to characterise such samples would therefore simplify data analysis and improve the usability of experimental results.

1.6 Related work

Applying ML methods to process vast collections of genomic data is an area of active scientific interest. Ellis et al. used RNA-seq profiles from large public repositories – Genotype-Tissue Expression Project, The Cancer Genome Atlas (TCGA) and the Sequence Read Archive – to train supervised linear predictors of chosen technical and biological sample attributes – sex, sample source (cell line or tissue), sequencing strategy (single or paired-end) and tissue type [56]. Linear ML models have also been employed to predict sensitivity and identify genomic markers of anticancer drugs based on genome-wide gene expression profiles complemented by additional information (such as chromosomal copy number measurements or pharmacological profiles of investigated drugs) [57, 58]. In these studies, elastic net regression – a method akin to logistic regression with a modified cost function – was utilised to reveal associations between specific genes and pharmaceutics.

The performance of simple linear predictors and more complex, non-linear ML techniques was compared by Stetson et al., who trained random forest and SVM models on multi-omic data (comprising microarray gene expression profiles, copy number variation and mutational status) to predict anticancer therapeutic response [59]. Tumour drug sensitivity prediction was also the subject of other supervised ML approaches, including deep neural networks trained on gene expression and mutation profiles [60] and multitask learning, in which models for individual drugs are not trained independently but information is shared between tasks during training to achieve improved performance [61].

In recent years, many researchers have employed unsupervised ML methods to process public *omics* data, with deep learning architectures being particularly popular. For instance, biologically informative features were extracted from gene expression profiles of breast cancer cells by denoising autoencoders [62]. Similar models were used to improve the results of gene clustering based on gene expression data [63], with prior biological information incorporated into the clustering process (in the form of a network-based metric) by Cui et al. [64]. Variational autoencoders have been employed to extract a biologically relevant latent space from pan-cancer RNA-seq data publicly available in TCGA [65] and to predict the response of cancer cells to chemotherapeutic drugs based on gene expression profiles [66, 67]. Moreover, to overcome an insufficient number of training examples for unsupervised ML techniques, a transfer learning framework for transcriptomic data was introduced [68]. The authors show that ML models can be trained using large, public gene expression compendia and then transferred to much smaller datasets of rare disease samples.

As can be seen, processing and analysing gene expression data is a frequent subject of scientific research and a plethora of relevant data sources are publicly accessible. On the contrary, the chromatin accessibility landscape of cells is a much less explored area, with a lower amount of experimental data available. A shortage of training examples also limits the choice of ML models suitable for the analysis of these data, with many methods (particularly from the category of deep learning) relying on large training datasets. In this work, we bring chromatin accessibility to the centre of interest, trying to gather as much relevant data as possible and to exploit it using customised technical means.

2 Methods

This chapter contains a description of methodological steps of the project, starting with the assembly and pre-processing of a comprehensive chromatin accessibility dataset. Next, we will describe the metadata augmentation procedure, which results in a collection of ML classifiers subsequently used to classify new samples. We will conclude with information about data and code availability. If not stated otherwise, all methods were implemented in Python[1] programming language, version 3.7.4.

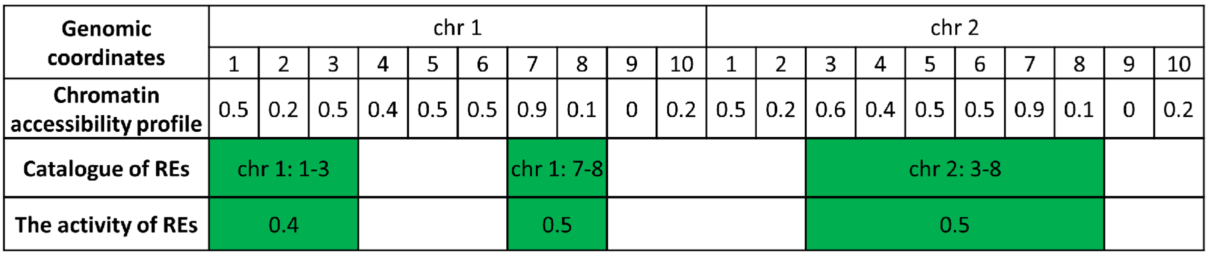
2.1 Integrated chromatin accessibility dataset

As explained in chapter 1.3.2, measuring chromatin accessibility makes it possible to assess the activity of REs. However, the results of such experiments are scattered across many different databases or separate publications, which makes the collection and uniform processing of these data a necessary first step before they can be utilised for further computational analyses. Here, we were looking for data generated by two different methods with comparable outputs – DNase-seq and ATAC-seq (see chapter 1.3.2) – performed on human cell samples. The primary input for creating the integrated dataset were genomic coverage tracks (or profiles), which quantify chromatin accessibility across the entire genome. Due to experimental limitations, however, the whole genome is never fully covered. Nevertheless, coverage tracks can be very large, therefore they are usually stored in binary *bigWig* format in order to reduce the file size.

2.1.1 Data aggregation and filtration

Genome-wide chromatin accessibility profiles do not directly provide quantification of the activity of REs. To achieve such representation, they need to be aggregated with respect to pre-defined catalogues of REs. First, a list of REs in the human genome is established, identifying each element unambiguously by specifying the chromosome number and start/end coordinates of the region. Such a catalogue can be conveniently stored in a *BED* (Browser Extensible Data) file. Based on this set of genomic regions, the process of aggregation can be performed by calculating the mean of numbers in chromatin accessibility coverage tracks over each RE (see figure 8 for illustration). After aggregation, every element is assigned a single numeric value, which expresses its overall accessibility (and therefore activity) in the examined cell sample.

[1] https://www.python.org/ [September 11 2020]



Genomic coordinates	chr 1										chr 2									
	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
Chromatin accessibility profile	0.5	0.2	0.5	0.4	0.5	0.5	0.9	0.1	0	0.2	0.5	0.2	0.6	0.4	0.5	0.5	0.9	0.1	0	0.2
Catalogue of REs	chr 1: 1-3						chr 1: 7-8						chr 2: 3-8							
The activity of REs	0.4						0.5						0.5							

Figure 8: A schematic depiction of the aggregation procedure. The chromatin accessibility profile contains numeric values for (ideally all) genomic positions, which are identified by the chromosome number (e.g. chr1, chr2) and the corresponding index in the DNA sequence. Analogically, a RE is defined as a range of such positions. During aggregation, each RE is assigned the mean of the numbers from the genome-wide profile which correspond to its range. In this toy example, three REs (marked with green colour) are defined within chromosomes 1 and 2.

Defining a catalogue of REs in the human genome is not a trivial task as these sequences are highly dynamic and diverse and there is no single universally accepted set of REs, which could be used as a reference. We have therefore decided to use three publicly available sets of regions identified within different projects/studies to cover a broad range of complexity:

- FANTOM5 enhancers – an atlas of approx. 65,000 active enhancers retrieved by applying Cap Analysis of Gene Expression on samples covering many human tissues and cell types [69]
- The Ensembl Regulatory Build – a set of more than 610,000 genomic segments identified using machine learning and classified according to their function [70]
- Annotated Regulatory Index – a deep reference map of DNase I hypersensitive sites from 733 human cell samples, containing approx. 3.6 million REs [71]

To simplify notation, these catalogues will be hereafter referred to as FANTOM5, ENSEMBL and INDEX references of REs, respectively.

Before performing aggregation, (in)compatibility of reference genome assemblies had to be addressed. Genomic coordinates are defined with respect to a certain version of the genome assembly, e.g. the reference genome used to align raw sequence reads (these are produced directly by NGS during ATAC-seq or DNase-seq). To obtain correct results, both chromatin accessibility coverage tracks and lists of REs have to be transferred into the same coordinate system before aggregation if that is not the case by default. In practice, we encountered two versions of the reference human genome – GRCh37 (hg19) and GRCh38 (hg38). To create a homogeneous dataset, we have decided to convert all collected data with hg19 reference to the latest hg38 assembly. We used *liftOver*[2] to perform this operation, which is a Linux command-line tool developed by the Genomics Institute of University of California Santa Cruz

[2] Available from (for 64-bit Linux system): http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/ [September 11 2020]

(UCSC). After coordinate conversion, every chromatin accessibility track could be aggregated with respect to each of the aforementioned sets of REs. For the aggregation operation itself, we utilised another UCSC's command-line utility: *bigWigAverageOverBed* [2].

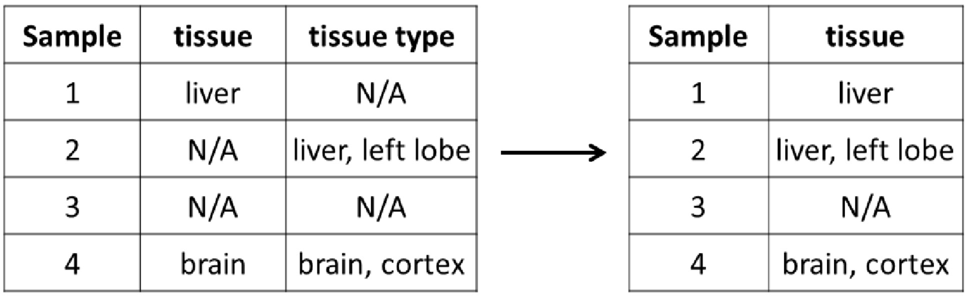
One more technical issue emerged in connection with the aggregation procedure. As opposed to the toy example in figure 8, not each RE has full coverage in the chromatin accessibility profile. In fact, for many samples, the accessibility levels corresponding to some of the REs are fully or partly missing. In the case when only a certain portion of the RE is covered in the genome-wide profile, there are two options on how to calculate the overall accessibility of such region. The sum of defined values from the chromatin accessibility track corresponding to the RE can be divided either by the full length of the region or by the number of genomic positions (nucleotides) covered. The results obtained in the latter case will be referred to as effective mean values of chromatin accessibility. As the effect of this technical difference on further analyses was not clear during dataset assembly, both mean and effective mean values were computed and kept in separate data files until unsupervised analysis was performed.

Finally, after aggregation and before being added to the integrated chromatin accessibility dataset, each sample undergoes a simple "quality control" step: sample vectors containing too many zeros are filtered out. The rejection threshold was defined as a maximum acceptable percentage of zero values for a sample and was set to 90 % (see chapter 3.1.1). This filtration procedure is justified by the assumption that the sparsity of sample vectors is caused primarily by the low coverage of the corresponding genome-wide profiles and that such samples bring little to no information usable for downstream analyses.

2.1.2 Metadata refinement

Apart from the primary quantitative data (i.e. chromatin accessibility levels), high-quality annotations were needed for individual samples. The information contained in these metadata are crucial for training supervised ML models as they determine what the training examples actually are – annotations may comprise descriptions of cell types, developmental stages, diseases, experimental conditions and many other attributes. However, as the chromatin accessibility dataset integrates data from various sources, the metadata suffered from a great level of inconsistency and incompleteness. To remedy this problem, a semi-automatic refinement procedure was developed to make the annotations more usable for automatised computational processing. For the sake of clarity, the terms metadata *attribute* and *value* should be carefully distinguished. An attribute can be perceived as a certain category of annotations (e.g. "tissue type"), whereas an attribute value is a concrete label assigned to a sample in this category (e.g. "kidney").

Metadata refinement was performed in several stages. First, simple adjustments of individual annotations were made to improve their consistency, such as setting all values to lowercase or replacing various equivalents of undefined entries (e.g. "unknown", "not reported" etc.) with a single uniform label. Subsequently, metadata attributes were refined based on their names



Sample	tissue	tissue type
1	liver	N/A
2	N/A	liver, left lobe
3	N/A	N/A
4	brain	brain, cortex

Sample	tissue
1	liver
2	liver, left lobe
3	N/A
4	brain, cortex

Figure 9: A simplified example of attribute merging during metadata refinement. Two attributes with similar contents – “tissue” and “tissue type” – are merged into a single attribute “tissue”. For each sample, the new attribute acquires the defined value from one of the original attributes, if such value is available. If the sample has labels defined for both original attributes (sample 4 in this figure), the more specific annotation is kept.

or contents. This phase includes two main operations – deletion or merging of attributes. For example, an attribute is deleted if it contains no valid entries or has no informative value (e.g. various types of identifiers). Similarly, two attributes are merged if they describe the same or similar sample properties (e.g. “tissue” and “tissue type”). Merging thus results in the creation of a single attribute (see figure 9) which summarises the information from both original attributes. If a value collision occurs (i.e. the sample has a label defined for both attributes to be merged), we keep the value from the attribute which has more unique entries, assuming that such attribute contains more detailed information about the sample. It is important that each refinement operation can be revised by the user before execution and rejected or adjusted if necessary.

The second round of value refinement was implemented through user-defined substitutions based on utilising simple regular expressions. The substitutions are defined in a dedicated tab-separated values (TSV) file as “pattern” / “replacement” pairs in the form of regular expressions. The file is then loaded during the refinement procedure and the substitutions are executed by looking for the patterns in individual metadata entries and replacing any matches with the defined character strings. In addition, more complex substitution rules may be constructed (taking into account for example the number of words in metadata entries) by defining separate Python functions.

Finally, the attributes were refined once more by merging. This time, however, their similarity was assessed not according to their names but by comparing the values they contain. For this purpose, so-called “overlap ratios” were computed for each pair of attributes. We defined this metric as the ratio between the number of unique values which the two attributes have in common and the total number of their unique values. The attributes are then merged if their overlap ratio is above a chosen threshold (set to 25 % in this work). The procedure is performed iteratively until overlap ratios for all pairs of attributes are below the threshold. Again, the user can manually override any proposed change in each iteration.

2.1.3 Data normalisation and standardisation

The next step in the processing of quantitative chromatin accessibility data is their normalisation and standardisation in order to diminish the effects of technical variation. First, QN was performed for each sample as described in chapter 1.4.1. The construction of reference distribution for QN is discussed in the corresponding results section, chapter 3.1.3. QN was implemented using function `normalize.quantiles.target` from *preprocessCore* [3] package for R[4], which by default replaces tied value ranks with the median of respective values from the reference distribution (see chapter 1.4.1). For the data at hand, this behaviour may cause zero values in the original distribution to be replaced by non-zero numbers after normalisation. The unwanted offset is eliminated by subtracting the minimum from all the values in the normalised distribution, therefore keeping original zero entries unchanged.

As a preparation for further analyses (in particular for training ML models), the normalised data were additionally standardised (see chapter 1.4.1). After this step, each feature (i.e. RE) contains values with zero mean and unit standard deviation. Due to the large volumes of processed data, it may not always be possible to load the whole dataset into computer memory at once. Therefore, we implemented standardisation as an "online" computation – data are loaded into memory incrementally in smaller batches and parameters (mean and standard deviation) are updated with each processed batch. In this work, `sklearn.preprocessing.StandardScaler` class of *scikit-learn* [5] package for Python was utilised (with default settings), which provides incremental implementation of the algorithm introduced in [72]. Moreover, this implementation yields a data object which contains the computed parameters and which can be stored in a binary file. When new data are to be added to the chromatin accessibility dataset, the object can be loaded from this file and its parameters are updated after processing the new data. This way, re-processing of the whole dataset is avoided.

2.1.4 Unsupervised analysis

For the purposes of data visualisation, dimensionality reduction was employed (see chapter 1.4.2). First, PCA was performed using the normalised and standardised data. Similarly to standardisation, PCA was implemented in a way which allows to load the data in batches of defined size (i.e. containing a certain number of samples) and to incrementally compute PCA loadings. After the whole dataset is processed, the loadings can be used to transform the original data into a new feature space of PCs (i.e. PCA scores are computed). Here, `sklearn.decomposition.IncrementalPCA` class from *scikit-learn* was utilised to achieve this result. We used the default parameter settings, changing only the number of retained PCs to 100. As the chosen PCA implementation requires that the number of extracted PCs must be equal or lower than the number of samples in each data batch to be processed, we were limited

[3] https://www.rdocumentation.org/packages/preprocessCore/versions/1.34.0 [September 11 2020]

[4] https://www.r-project.org/ [September 11 2020]

[5] https://scikit-learn.org/ [September 11 2020]

while setting this parameter by the maximum size of data batch that could be safely loaded into memory (the limiting factor was the dataset for INDEX reference as it contains the most features and therefore the largest sample vectors). The computed PCA parameters can be stored in a binary file for later use in the same way as described in the previous chapter.

PCA scores may be used not only for visualisation (by plotting chosen pairs of PCs against each other, usually the 1st and the 2nd component) but also as an input for more complex analytical methods for which it would be computationally infeasible to process the original, high-dimensional data. Indeed, obtained PCA scores were fed into other algorithms for dimensionality reduction and visualisation – namely t-SNE and UMAP (see chapter 1.4.2). t-SNE was used to embed PCA scores into 2-D space, utilising `sklearn.manifold.TSNE` class from *scikit-learn* for implementation. The default settings from documentation were kept except for the $perplexity$ parameter, which was set to 50 based on the results of empirical testing. Similarly, two-dimensional UMAP embedding was performed using `UMAP` class from *umap-learn* [6] Python package. Here, testing was done to examine the effects of $n\_neighbors$ and min_dist parameters on visualisation results, leading to the choice of following values: $n\_neighbors$ = 15, min_dist = 0.5. Specifically, we wanted to achieve visible clustering of samples with individual clusters being as far as possible from each other to ensure good separation.

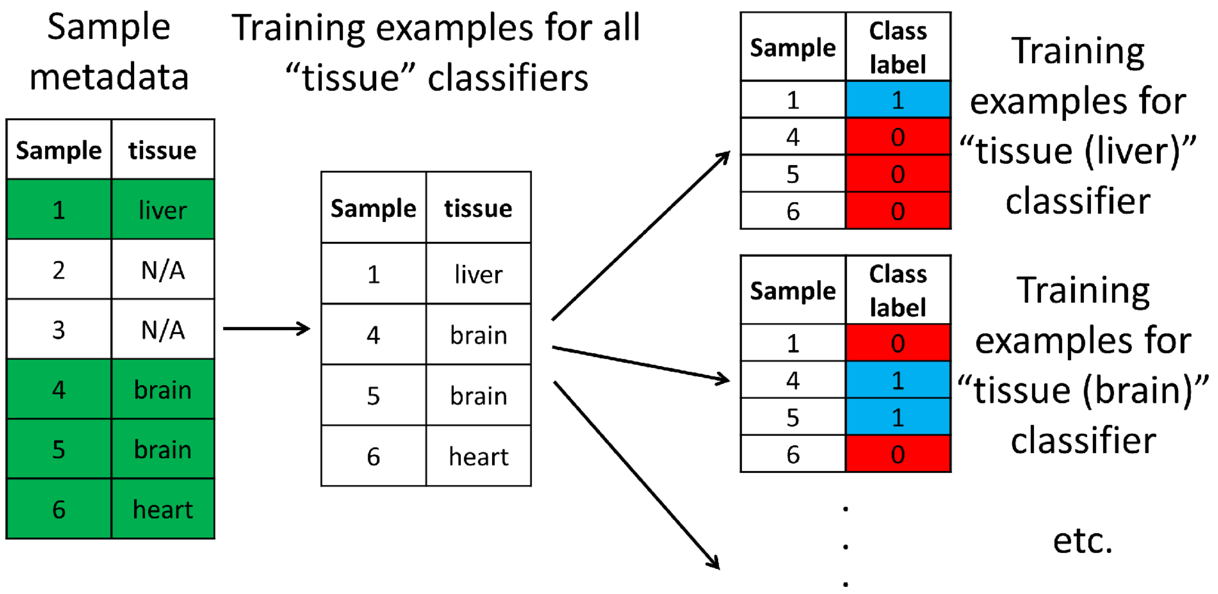
2.2 Metadata augmentation

Although the usability of sample metadata for computational processing was improved through the refinement procedure, the incompleteness of annotations still posed a severe issue. To address this problem, we developed a strategy for metadata augmentation, in which ML models (classifiers) are trained using known annotations and subsequently used to predict undefined metadata labels. Such an approach should not only improve metadata quality for further utilisation but also lead to the construction of ML models capable of distinguishing various cell types and characteristics based on their gene expression profiles or chromatin accessibility landscapes.

2.2.1 Classifier testing and selection

The basic strategy for metadata augmentation was to build a separate binary classifier for each metadata label. This approach is schematically illustrated in figure 10. The training set of each binary classifier comprises all the samples which have the annotations defined for the corresponding metadata attribute. Subsequently, the samples with a particular label (for which the classifier is constructed) represent the positive training examples (i.e. samples belonging to the class in question) and all the other training examples represent the negative class. Each classifier is therefore trained to recognise samples characterised by a specific annotation.

[6] Available from: https://github.com/lmcinnes/umap [September 11 2020]


Sample metadata
Training examples for all "tissue" classifiers
Training examples for "tissue (liver)" classifier
Training examples for "tissue (brain)" classifier
etc.

Sample	tissue
1	liver
2	N/A
3	N/A
4	brain
5	brain
6	heart

Sample	tissue
1	liver
4	brain
5	brain
6	heart

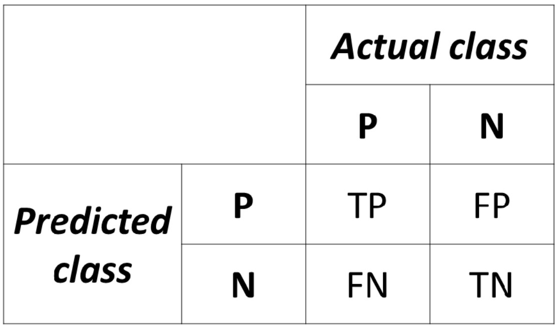
Sample	Class label
1	1
4	0
5	0
6	0

Sample	Class label
1	0
4	1
5	1
6	0

Figure 10: This figure illustrates the strategy used to construct classifiers for metadata augmentation. For each metadata attribute (such as "tissue" in this example), the samples with defined labels (marked with green colour) constitute the training set for all the classifiers corresponding to that attribute. A separate binary classifier is then built for each attribute value (e.g. "liver", "brain" etc.) – the training samples labelled with that value represent the positive class (label 1, marked with blue colour) and all the other training examples constitute the negative class (label 0, marked with red colour).

To evaluate the performance of classifiers, stratified k-fold cross-validation was chosen as the testing strategy. For each classifier, the training set is partitioned into k subsets (folds) of equal size (or almost equal in case the number of training examples is not the multiple of k). The partitioning is random with a single condition imposed – the ratio between the number of positive and negative training examples should be approximately equal in each fold (hence stratified cross-validation). Training of the model is then performed using $k - 1$ folds, the remaining fold serves as the testing set on which classification performance is assessed. The procedure is repeated k times with a different fold being used as the testing set in each round and the evaluation of performance can be averaged across all iterations.

Various types of classifiers were tested for the purpose of metadata augmentation. From the category of linear models, logistic regression or SVM with linear kernel (see chapter 1.4.3) were implemented through *scikit-learn*'s `sklearn.linear_model.SGDClassifier` class, which utilises stochastic gradient descent (SGD) for model training. Perhaps the main advantage of linear classifiers is their simplicity and therefore very high computational efficiency, which makes it possible to use them also for large data volumes. Moreover, the assessment of feature importance is straightforward – the models assign coefficients to individual features (in our case genes or REs) and these are proportional to the contribution of the features to classification results.



		Actual class	
		P	**N**
Predicted class	**P**	TP	FP
	N	FN	TN

Figure 11: General confusion matrix summarising the results of binary classification with positive (P) and negative (N) class. The number of correctly classified samples is the sum of true positive (TP) and true negative (TN) classifications. If the predicted class does not correspond to the actual (real) class, it may be the case of either false positive (FP) or false negative (FN) classification.

As for non-linear classifiers, SVM with RBF kernel was tested using `sklearn.svm.SVC` class from *scikit-learn* for implementation. Due to much higher complexity of computations incorporated in the model, it was not feasible to apply it on the original datasets but only on their PCA-transformed versions, in which the number of features was reduced to 100 PCs. Besides, there is no possibility of directly evaluating feature importances with this model – the decision boundary is non-linear in the original feature space and therefore cannot be easily expressed as a linear combination of features (see chapter 1.4.3).

All implemented ML models can output not only binary class labels (0 or 1) for each classified sample but also the probability (a number between 0 and 1) with which the sample belongs to the positive or negative class. This is an important feature as it enables us to continuously quantify the similarity between samples and classes. By tweaking threshold P ($P > 0.5$), we can also increase the reliability of predictions by accepting the sample into a certain class only if the probability estimate is higher than P. The effects of this probability thresholding on classification performance, as well as the results of testing the classifiers with various parameters and on different datasets, are described in chapter 3.3.1.

Finally, an important question is how to objectively assess the performance of classifiers. For binary classification, the standard metrics are derived from quantities introduced in figure 11. A basic performance score is accuracy, which is the percentage of correctly classified samples from all the samples classified. Using the terms from the confusion matrix in figure 11, it can be calculated as

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{17}$$

However, accuracy may not be very informative in case of imbalanced datasets, in which one of the classes is represented by much more examples than the other. In such a situation, the classifier may label all the samples as belonging to the dominant class, therefore showing good accuracy. However, it failed to recognise those few training examples from the opposite class, which is usually a problem. Thus, to obtain a more complete picture of classifier performance, additional metrics must be employed. In this work, we have decided to use precision (also known as positive predictive value) and recall (also sensitivity or true positive rate), two further scores commonly used for classifier evaluation. They are computed as follows:

$$precision = \frac{TP}{TP + FP} \tag{18}$$

$$recall = \frac{TP}{TP + FN}. \tag{19}$$

Accuracy, precision and recall together provide a complex assessment of classification performance and can therefore be used as metrics for selection of classifiers suitable for final prediction of metadata labels and classification of new samples. The selection itself was performed by imposing conditions (thresholds) on performance scores yielded by classifier testing. The process of determining these conditions is described in chapter 3.3.1.

2.2.2 Iterative training and prediction

Metadata augmentation itself was implemented as an iterative procedure of classifier training, prediction, evaluation and selection. An overview of the whole process can be seen in figure 12. The testing and selection phase described in the previous chapter yields the list of classifiers accepted for the prediction of metadata labels. These classifiers are first re-trained with all available training examples, as opposed to the testing phase, during which only a part of the data was used for training (due to cross-validation).

The unknown annotations may subsequently be predicted by presenting the corresponding sample vectors to the trained classifiers and computing the probabilities with which the samples belong to particular classes. Here, probability thresholding as described in chapter 2.2.1 takes

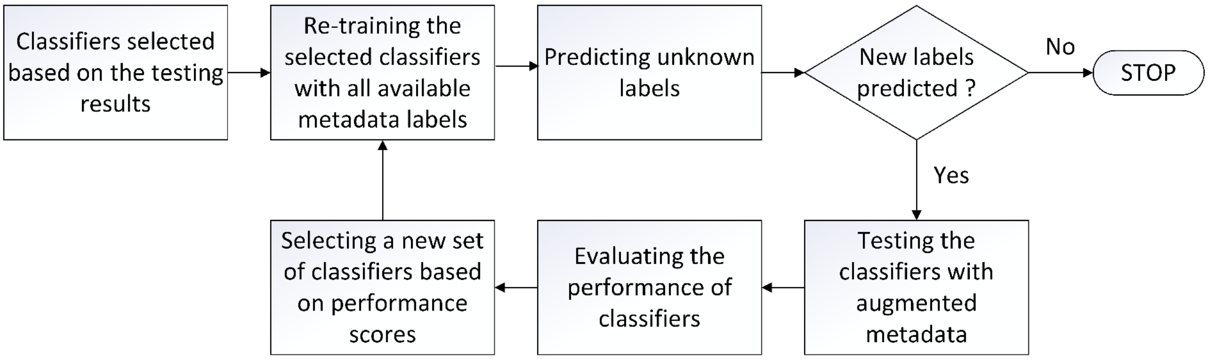

Classifiers selected based on the testing results
Re-training the selected classifiers with all available metadata labels
Predicting unknown labels
New labels predicted ?
No
STOP
Yes
Testing the classifiers with augmented metadata
Evaluating the performance of classifiers
Selecting a new set of classifiers based on performance scores

Figure 12: A flowchart illustrating the iterative procedure of predicting metadata labels.

place – the label is assigned only to the samples for which the probability estimate is higher than the chosen threshold P_p (p stands for "prediction"). It may happen that for a certain sample, two different values of the same metadata attribute are to be assigned (for example, the sample may be classified as positive by both "tissue (liver)" and "tissue (brain)" classifier). In such cases, only one of the labels can be chosen – naturally the one with the higher probability estimate. Moreover, for chromatin accessibility data, there are three training sets available with the same sample metadata, one for each catalogue of REs. Therefore, some metadata labels may have multiple corresponding classifiers trained (provided they were accepted during the selection step). The predictions for these classes can be made more reliable by averaging the probability estimates from all available classifiers.

After prediction of new annotations, the augmented metadata are used for re-evaluation of classifiers. The procedure is the same as described in the previous chapter – stratified k-fold cross-validation is performed, followed by calculation of performance scores (accuracy, precision and recall). Probability thresholding occurs also during this phase to select the samples which should contribute to computing the scores. However, the chosen threshold P_e (e for "evaluation") is practically unrelated to the prediction threshold P_p and may be selected independently. Based on the performance metrics, some of the classifiers may be rejected and then a new set of classifiers enters into the next iteration of training, prediction and evaluation. The whole process is repeated until no new metadata labels are predicted.

All classifiers trained in the course of the described procedure are stored in the form of binary files so they can be later loaded and used for classification of new samples (not included in the training datasets). For the classifiers that were rejected during re-evaluation steps, the last accepted version is kept.

2.3 Classification of new samples

The goal of this work is to construct ML models which could provide biologically meaningful information about cell samples based on their gene expression or chromatin accessibility profiles. This is achieved by using the classifiers trained during metadata augmentation to classify unseen samples. In this chapter, we discuss methodological measures which are necessary for such classification to be successful.

Firstly, the input sample vectors must have the same structure as the training data – only then can they be processed in the same way. For transcriptomic data, this means that each sample vector has to contain expression levels for all of the genes from the reference list described in chapter 3.2. At the same time, each gene must be unambiguously identified by either a gene name according to HUGO Gene Nomenclature Committee (HGNC) or an Ensembl ID (in the format "ENSG" + numeric code). The expression levels for the genes that are included in the reference list but not in the classified sample vector are set to zero. We consider this a reasonable compromise as it can be expected that if a certain gene is not listed in a transcriptomic profile, it was probably not expressed in the sample.

For chromatin accessibility data, an input to the processing pipeline are genome-wide coverage tracks, preferably in *bigWig* format. Each such profile is then aggregated with respect to FANTOM5, ENSEMBL and INDEX reference of REs (see chapter 2.1.1). This ensures that the aggregated sample vector has chromatin accessibility levels defined for the same genomic regions as in the respective training datasets. Once the sample vectors are ready (either for gene expression or chromatin accessibility data), they are fed into the pre-processing pipeline consisting of QN, standardisation and PCA-transformation. The parameters of individual operations (i.e. the reference distribution for QN, feature means and standard deviations, PCA loadings) were obtained during the processing of the training data and are re-used at this point. Finally, the trained ML classifiers are loaded from binary files and upon being presented with the pre-processed sample vectors, they yield the probability with which these samples belong to a particular class.

2.4 Data availability

Source code for all implemented methods is stored on internal servers of St. Anna Children's Cancer Research Institute (CCRI) and in the private GitHub repository at https://github.com/cancerbits. Binary files containing the trained models as well as the integrated gene expression and chromatin accessibility datasets (in the form of compressed text files) are stored solely on the CCRI servers due to their considerable size (more than 30 GB in total). The access permissions may be granted on an individual basis upon request sent to florian.halbritter@ccri.at.

3 Results and Discussion

The outcomes of procedures implemented within individual phases of the project are presented in this section, following a similar structure as in chapter 2. At the end of each major subsection, an additional chapter is included to offer discussion and interpretation of the achieved results.

3.1 Integrated chromatin accessibility dataset

As publicly available data from chromatin accessibility experiments are still relatively scarce (compared to gene expression datasets), an extensive search of suitable data sources was performed. The collected quantitative data were then subjected to a uniform pre-processing pipeline including aggregation, filtration, normalisation and feature reduction, with the effects of these steps being examined by appropriate visualisation techniques. Moreover, sample annotations were computationally collected, summarised and curated to be made usable for further analyses.

3.1.1 Data collection, aggregation and filtration

For the purposes of this work, we were looking for genome-wide chromatin accessibility tracks (in *bigWig* format) from DNase-seq or ATAC-seq experiments performed on human cells. An important requirement was that all the data had to be publicly accessible without any restrictions so that also the integrated dataset could be made available to the public upon completion. The collected samples originate from the following primary data sources:

- ChIP-Atlas[1] [73] – a database of mostly ChIP-seq data acquired from large public repositories and re-processed in a uniform way. For the purposes of this work, we were able to retrieve 1,632 DNase-seq profiles of human cells from ChIP-Atlas.
- Encyclopedia of DNA Elements (ENCODE)[2] [8, 74] – a public repository of data from various molecular assays created by the ENCODE Consortium, which aims to identify functional elements in the human genome. This resource yielded 994 DNase-seq and 63 ATAC-seq samples suitable for our purposes.
- The chromatin accessibility landscape of primary human cancers[3] [75] (for the sake of brevity, this source is referred to as the "Landscape" dataset in further text) – in this study,

[1] https://chip-atlas.org/ [September 11 2020]

[2] https://www.encodeproject.org/ [September 11 2020]

[3] Data available from: https://gdc.cancer.gov/about-data/publications/ATACseq-AWG [September 11 2020]

the chromatin accessibility landscape was determined for 410 tumour samples from 23 cancer types. As technical replicates were available for most of the samples, we could retrieve 796 ATAC-seq profiles in total.

- Gene Expression Omnibus (GEO)[4] [76] – arguably the largest repository of genomic data up to date, GEO was a reasonable place to search for the data not covered by the previous sources. Indeed, through the use of GEO's advanced search functionality combined with a manual revision of pre-selected datasets, we acquired another 981 ATAC-seq samples from 32 different GEO series (commonly, a series contains data from a single study or project).

Due to the amount of retrieved data, it was not feasible to include a complete list of identifiers of downloaded samples/datasets in this work (such catalogue is available upon request, see chapter 2.4). Through the manual revision of sample identifiers, we also checked for duplicity as some samples could potentially be included in multiple data sources. Indeed, it turned out that the majority of DNase-seq experiments from ChIP-Atlas were contained in GEO; there were also some overlaps between GEO and ENCODE databases. All discovered duplicates were removed from the integrated dataset, which therefore contains 4,466 unique samples.

As chromatin accessibility coverage tracks can have a considerable file size (commonly in the order of hundreds of megabytes up to several gigabytes), the collection of large batches of such profiles proved to be a time-consuming procedure. Moreover, the download speed for some of the source databases was very low, probably due to insufficient power or high occupancy of servers. Therefore, we implemented a parallelised framework in which multiple data files could be downloaded and processed simultaneously within separate computational threads. Upon being downloaded from an online source, each coverage track was aggregated with respect to FANTOM5, ENSEMBL and INDEX catalogues of REs. As described in chapter 2.1.1, version of the reference genome assembly had to be taken into account during aggregation as it was not consistent across the collected data. However, rather than converting each coverage track to the latest coordinate system (hg38) individually, we created two corresponding versions of each RE list (for hg19 and hg38 reference), which was a one-time procedure, and used for aggregation always the version which matched the genome reference of the particular sample. This approach resulted in the loss of some regions from RE catalogues (24, 2,991 and 15,335 regions for FANTOM5, ENSEMBL and INDEX references, respectively) but, together with parallel data download and aggregation, greatly reduced (at least 10-fold) time frame in which the whole integrated dataset was assembled.

The filtration step was implemented by rejecting all sample vectors which contained more than 90 % of zero values. Rejection threshold was determined based on visual assessment of distributions of zero-value percentages across all samples. An example of such distribution (for FANTOM5 reference) can be seen in figure 13. It is evident that there is a considerable number of sample vectors with only zero values – such vectors contain no information about chromatin

[4] https://www.ncbi.nlm.nih.gov/geo/ [September 11 2020]

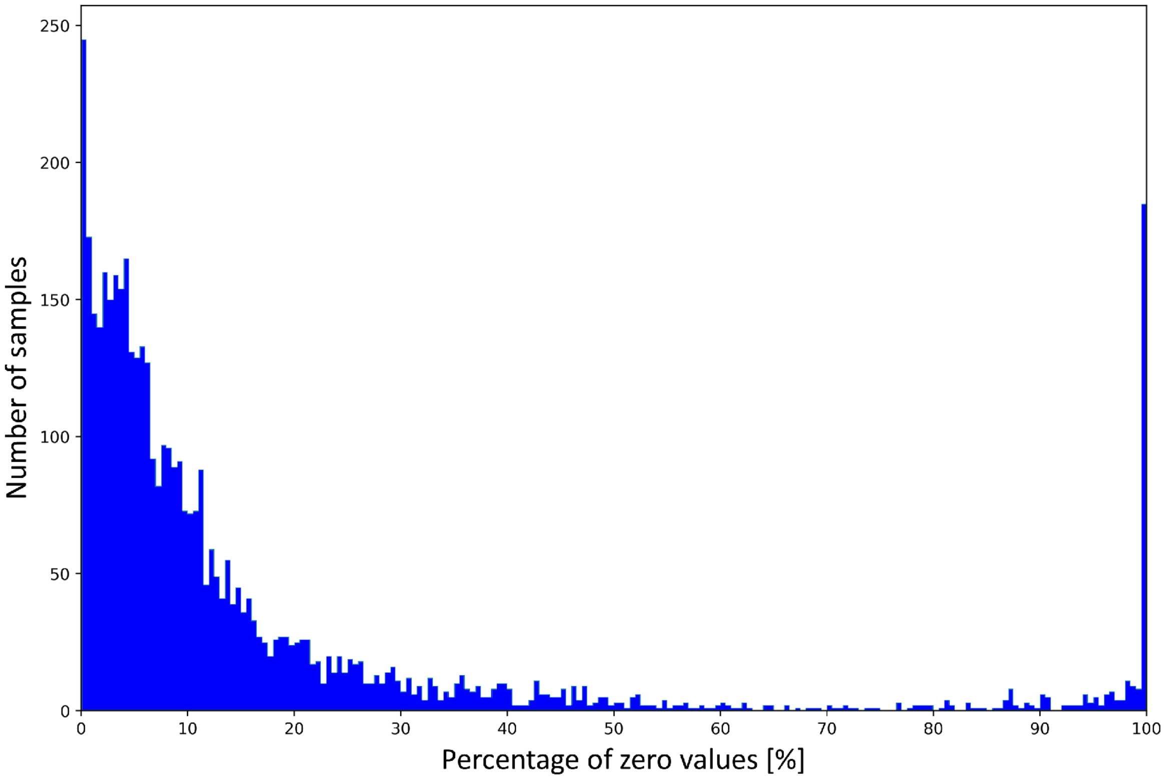


Figure 13: Distribution of percentages of zero values in sample vectors after aggregation with respect to FANTOM5 reference. Bin size is 0.5 %.

accessibility landscape of cells. By looking directly at the original *bigWig* coverage tracks of these samples, we found out that the main reason of such aggregation outcome was very low coverage of the genome-wide profiles – chromatin accessibility levels were simply not defined in the regions corresponding to REs. In summary, the filtration procedure resulted in rejecting 269, 249 and 297 samples for FANTOM5, ENSEMBL and INDEX references, respectively.

It is worth noting that through the aggregation procedure, the original data (i.e. *bigWig* profiles) were "compressed" as the number of genomic regions defined in the catalogues of REs is in general much smaller than in genome-wide coverage tracks. Thus, hundreds of gigabytes of source data were aggregated into compressed text files with sizes of approx. 393 MB, 3.4 GB and 18.4 GB for FANTOM5, ENSEMBL and INDEX references, respectively. We stored the data in plain text to make them both human-readable and compatible with all common data processing frameworks for further analyses.

3.1.2 Metadata collection and refinement

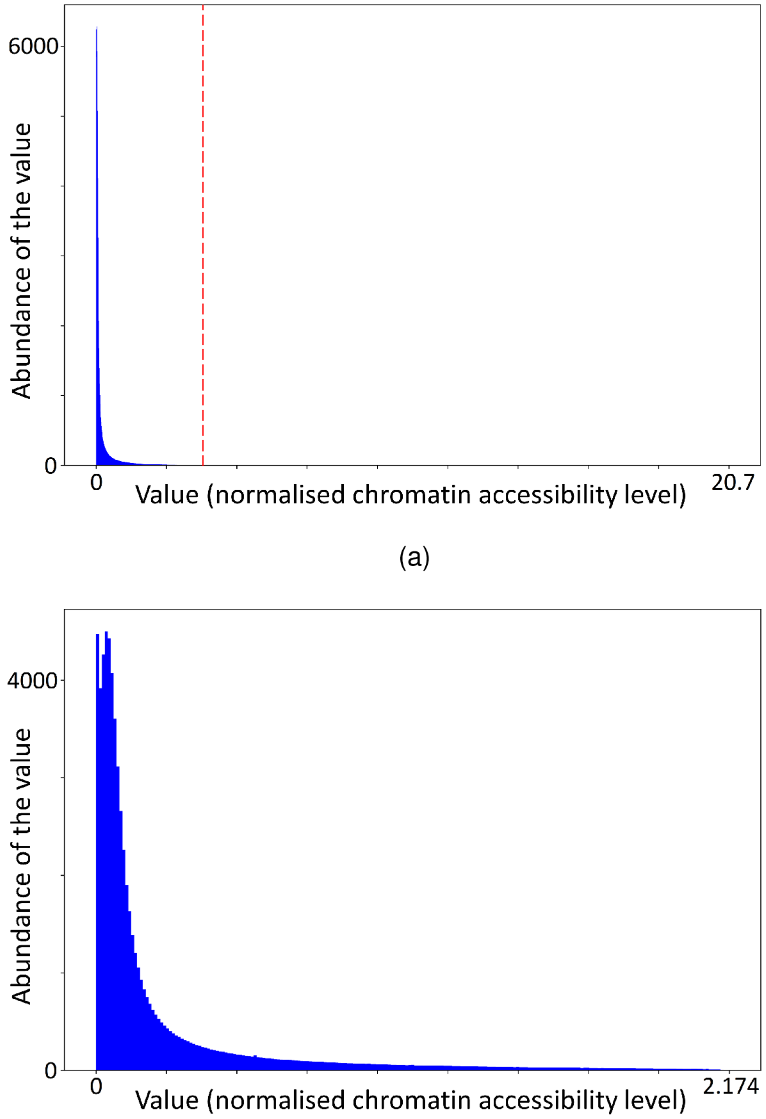
In addition to chromatin accessibility profiles, the corresponding sample metadata had to be retrieved from data sources listed in the previous chapter. For samples from ChIP-Atlas, ENCODE and "Landscape", this was a straightforward procedure as annotations could be downloaded collectively for all chosen samples in a single text, TSV or Microsoft® Excel® file. The retrieved files were then manually merged into a joint metadata table, removing some uninformative or redundant attributes and merging attributes with identical contents in the process.

In GEO, however, metadata are stored separately for individual GEO series (labelled by GSExxx accession numbers) in Simple Omnibus Format in Text (SOFT) family files. These structured text files contain (among others) annotations to all samples from a particular series. As the collected samples were scattered across multiple series, it was necessary to access GEO programmatically to retrieve the metadata. We utilised FTP directory structure of the database and developed a simple framework which, given the list of GSE accessions, automatically downloads the corresponding SOFT family files, parses them to extract metadata and adds new annotations to an already existing file. A basic curation of specific metadata attributes is performed at this step as well. Moreover, as most of the samples from ChIP-Atlas are stored also in GEO, we decided to retrieve additional annotations for these samples. To do so, we first had to match sample identifiers used in ChIP-Atlas with the corresponding GEO IDs, utilising GEO's E-Util programs (specifically, eSearch and eSummary). Subsequently, the metadata could be retrieved in the same way as described above.

The collection step resulted in creating a joint metadata table which contained 288 different attributes describing all 4,466 samples. However, from the total number of potential metadata labels only around 12 % were defined. Annotations then underwent the refinement procedure described in chapter 2.1.2. The first round of attribute refinement reduced the number of attributes to 73 and then automated curation of metadata values was performed (based on user-defined regular expression patterns) to improve the consistency of labels. Finally, 19 pairs of attributes were merged in an iterative process due to the similarity of their values, leaving the final number of metadata categories after refinement at 54. The threshold imposed on overlap ratios during iterative merging was set to 25 %. Note that the refinement procedure can be controlled by the user at most of the steps – in this case, several manual corrections were necessary to avoid undesirable changes (for example, merging of some attribute pairs was prohibited despite the high overlap of their values – this concerned mainly attributes with numeric values, such as “age” and “replicate number”).

3.1.3 Data normalisation and standardisation

Collected chromatin accessibility profiles were quantile normalised in order to reduce technical variation in the data. The reference distribution for QN was computed as the mean of approximately 950 DNase-seq samples from ENCODE database (the exact number of sample vectors was slightly different for each set of REs due to the filtration step in which different samples could be rejected). We chose “read-depth normalised signals” (as named in ENCODE) to calculate the reference distribution because they were the results of a uniform processing pipeline developed by ENCODE for DNase-seq experiments. Moreover, these samples cover a wide range of cell types, developmental stages, treatments etc. and therefore their mean can be expected to provide a good representation of a standard chromatin accessibility profile.



(a)

(b)

Figure 14: Reference value distributions for QN. In figure a), the distribution is computed by simply averaging values of the selected samples from ENCODE database. In figure b), individual sample vectors were “capped” (using the threshold indicated by a red dashed line in a)) and logarithmically transformed in order to reduce range and skewness of the distribution (note the change in the scale of x-axis). The presented data were aggregated with respect to FANTOM5 reference, bin size is 0.01.

Before averaging, individual sample vectors were “capped” at threshold T, i.e. all values higher than this threshold were replaced by T. The purpose of this operation was to eliminate large values which are usually present with low frequencies and can be assumed to be the artefacts of experimental procedure or data processing and which could damage the averaged reference distribution in its upper portion. Threshold T was computed as the 99$^{\text{th}}$ percentile of all values in the chosen ENCODE samples: $T_{FANTOM5} = 3.547$, $T_{ENSEMBL} = 0.997$, $T_{INDEX} = 1.441$. In addition, each value x of individual distributions was transformed by computing $\log_2(x+1)$ to counteract skewness of the distribution and limit its range. Addition of 1 is necessary to handle zero values (which stay unchanged after this transformation) and to avoid negative values in the resulting distribution. A comparison between the reference distribution computed from the original values retrieved from ENCODE and after applying the presented adjustments can be seen in figure 14. After QN, the data were standardised feature-wise – for each RE, the mean of values across all samples became 0 and standard deviation became 1.

3.1.4 Unsupervised analysis

The first step towards the analysis of the chromatin accessibility dataset was performing PCA for data aggregated with each of the lists of REs and for both mean and effective mean values of chromatin accessibility levels (see chapter 2.1.1). We extracted the first 100 PCs and processed the data in batches of 4,500, 500 and 100 samples for FANTOM5, ENSEMBL and INDEX references, respectively (note that for FANTOM5, the whole dataset could be transformed at once). By plotting the first two PCs against each other we could now visualise the data in

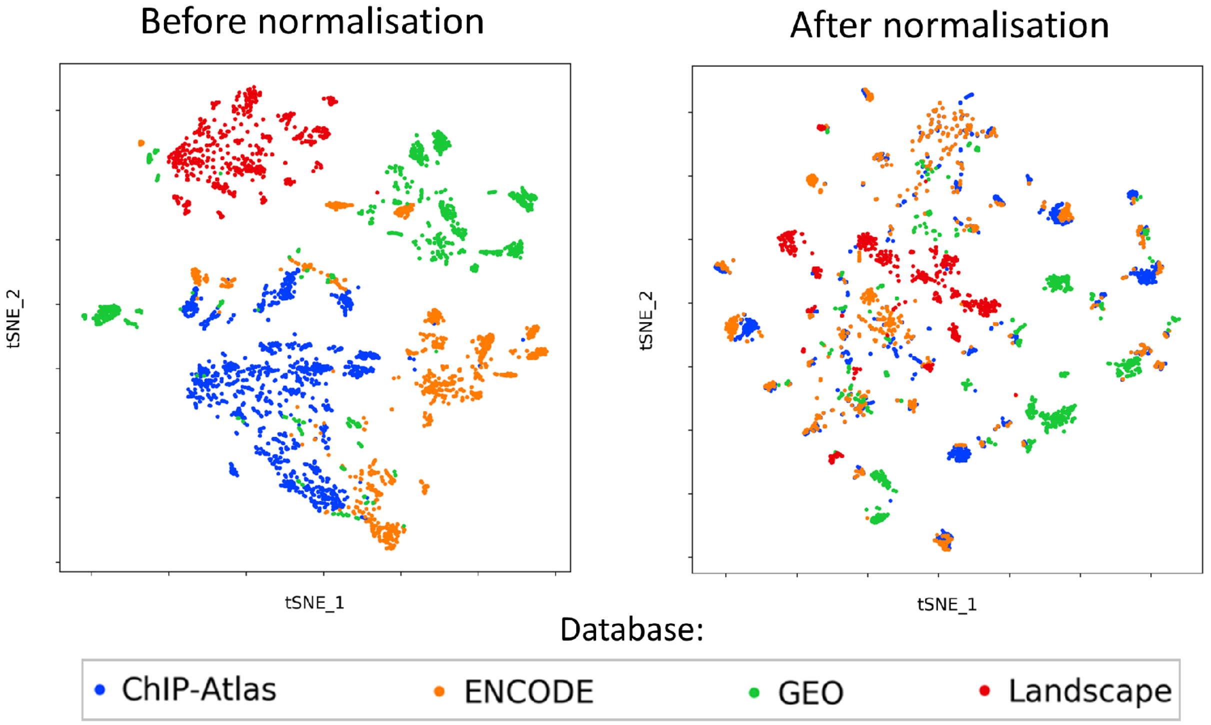


Figure 15: A comparison between the visualisation of the chromatin accessibility dataset before and after QN using t-SNE. Samples are visibly grouped according to a technical attribute (source database in this example) before normalisation whereas such grouping is much less pronounced after the data were normalised. This indicates that technical variation in the dataset was reduced by QN. The data were aggregated with respect to FANTOM5 reference.

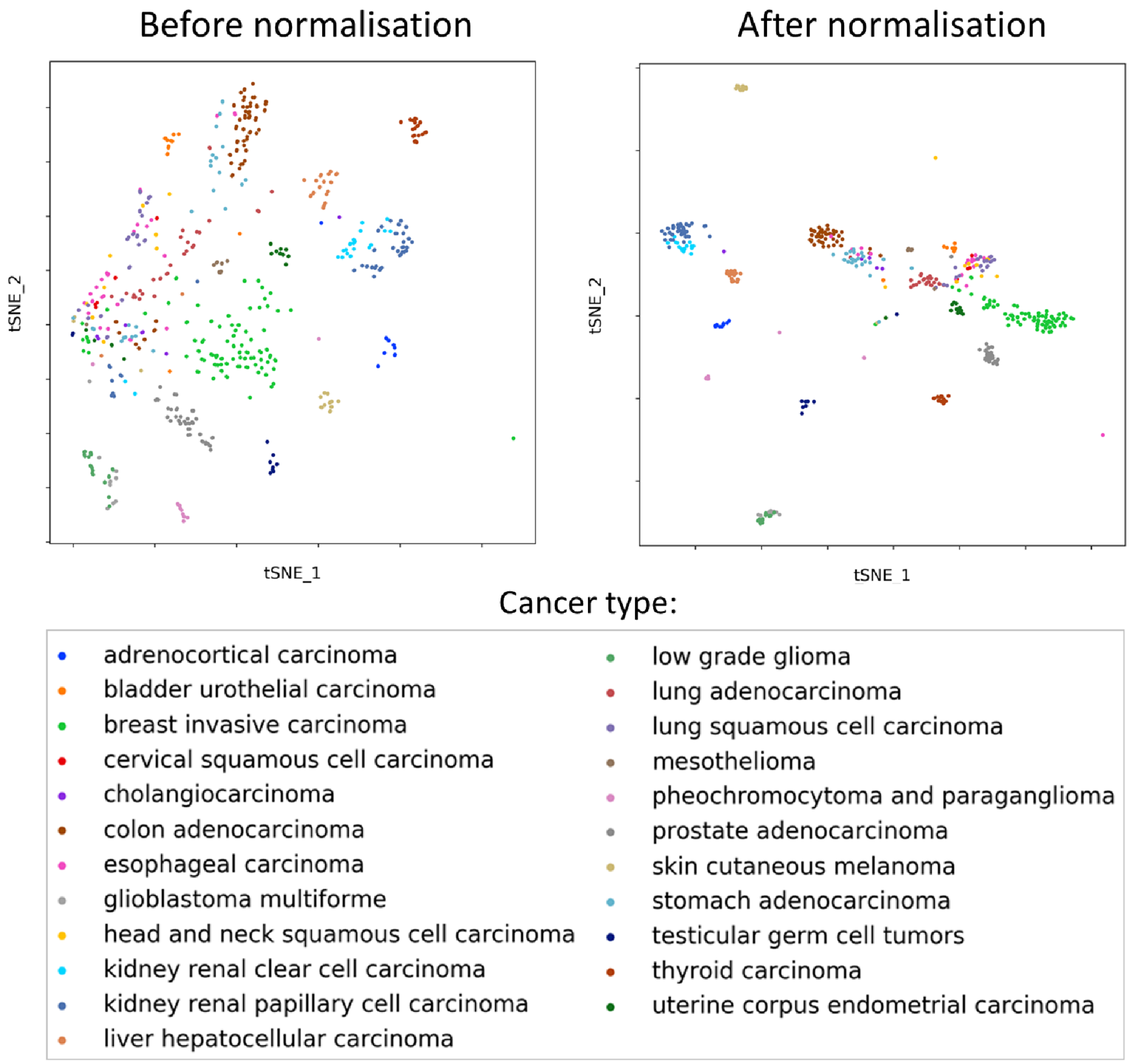


Figure 16: A comparison between the visualisation of the chromatin accessibility dataset before and after QN using t-SNE. Before QN, the samples are visibly grouped according to a biological attribute (cancer type in this example). Although the size, shape and orientation of clusters are different after QN (which is caused partly by t-SNE itself as it is a stochastic algorithm), the grouping of biologically similar samples is still present. This indicates that biological variation in the dataset is retained. The data were aggregated with respect to FANTOM5 reference, note that only a subset from the whole dataset is visualised here (for the remaining samples, cancer type was not defined).

2-D space. We first wanted to examine the effect of using mean or effective mean values for the computations. By visually comparing the corresponding pairs of PCA plots for each RE reference we came to the conclusion that there is little to no difference in the overall data structure between the two cases. Plots that support this claim can be found in figure 23 in the appendix. We therefore decided to use only effective mean values for all further analyses as computing effective mean during aggregation does not distort chromatin accessibility levels for regions with low coverage.

The next goal was to confirm that QN had the desired effect, i.e. reduced technical variation in the data. To capture finer data structures, we utilised t-SNE and UMAP for visualisation purposes and compared the plots of embeddings of the data before and after normalisation. Figures 15 and 16 containing t-SNE plots demonstrate that while technical variation was successfully diminished by QN, biological differences were preserved. This is supported not only by the visualisations presented here but we observed similar results using a different dimensionality reduction technique (i.e. UMAP), each of the RE references and multiple technical and biological metadata attributes. It was not feasible to include all results in this work but several additional visualisations can be found in the appendix (figures 24 and 25).

3.1.5 Discussion

We assembled a comprehensive resource of chromatin accessibility data from publicly available repositories. The heterogeneity that was inherently present in the created dataset was addressed on two levels. Firstly, sample metadata were curated by automatic, text-processing-based operations combined with manual interventions to improve the consistency of annotations. Secondly, primary quantitative data (i.e. chromatin accessibility levels) were aggregated with respect to predefined sets of REs and made more comparable through normalisation and standardisation procedures. Unsupervised analysis aimed at dimensionality reduction and subsequent visualisation revealed that these pre-processing steps succeeded in reducing technical variation in the data while keeping biologically relevant features distinguishable. This is an important finding which opens the way for downstream analyses performed on the dataset.

A simple storage format (plain or compressed text files) makes it possible to process the collected data with practically any programming language or framework and supports re-usability for other purposes. Moreover, the developed pipelines for automatic data download, aggregation and pre-processing allow for a straightforward extension of the dataset in case new data become available. Considering that no immediately re-usable chromatin accessibility dataset of comparable proportions currently exists, the assembled collection has sufficient quality and comprehensiveness to be a valuable tool for bioinformatic research.

3.2 Gene expression datasets

In this chapter, the assembly of comprehensive datasets containing gene expression profiles generated by RNA-seq is covered briefly. The primary contribution towards this task was delivered by Katja Nettermann, a member of the research group of Developmental Cancer Genomics at CCRI. The results of her work were re-used for the purposes of this thesis. Specifically, two distinct gene expression datasets were utilised:

- Tissue and Cancer Dataset (TCD) – contains expression levels of 52,685 genes in 29,618 human cell samples. It comprises healthy tissue samples retrieved from GTEx Portal[5] and samples from various types of cancers, made publicly available by the Treehouse Childhood Cancer Initiative[6].
- Stem Cell Differentiation Dataset (SCDD) – contains expression levels of 42,653 genes in 49,072 samples, which are mostly human stem cells in various differentiation stages. The data were acquired from refine.bio[7], a public repository of transcriptomic data.

As can be seen, each of the datasets contains a different set of features (genes). However, for the purposes of training ML classifiers, it was necessary to create a unified feature set, which would then have to be defined also for each new sample to be classified. This was achieved by determining an intersection between gene lists for TCD and SCDD. Some complications during the procedure arose from the fact that genes in each dataset are labelled with different types of identifiers. For TCD, gene names according to HGNC are used whereas in SCDD, Ensembl IDs are defined. The conversion between these two systems is possible but the correspondence is not one-to-one, i.e. multiple gene names may be assigned to a single ID or vice versa. In our case, among all the genes common between the two datasets, there were 19 HGNC gene names which had 2 different Ensembl IDs assigned. To avoid ambiguity, we decided to remove all these genes from the final reference gene list. This left us with 21,741 genes identified uniquely by both HGNC names and Ensembl IDs. It is worth noting that the majority of removed genes belong to classes that are of little interest for our purposes (e.g. genes encoding rRNAs or small regulatory RNAs) while most protein-coding genes were retained.

Keeping only genes in the reference list, pre-processing of gene expression datasets was performed similarly as for the chromatin accessibility data. Sample vectors were quantile normalised, standardised and PCA scores were computed, keeping the first 100 PCs. The reference distribution for QN was calculated as the mean of 17,382 sample vectors from GTEx Portal and therefore summarises gene expression profiles of cells from healthy tissues. Before averaging, individual sample vectors were capped (see chapter 3.1.3) at the 99th percentile of all values in GTEx samples, i.e. at $T = 11.526$. However, the data had already been logarithmically transformed by default so this operation could be skipped.

[5] https://www.gtexportal.org/ [September 11 2020]

[6] https://treehousegenomics.soe.ucsc.edu/ [September 11 2020]

[7] https://www.refine.bio/ [September 11 2020]

In addition, sample metadata were curated, separately for TCD and SCDD. The refinement procedure was analogical to the one used for the chromatin accessibility dataset and described in chapter 2.1.2, with additional manual pre-selection of metadata attributes aimed primarily at eliminating large amounts of uninformative annotations (such as redundant identifiers, empty attributes, dates etc.). After refinement, 27 metadata attributes were left for TCD and 17 attributes for SCDD.

3.3 Metadata augmentation

As mentioned in chapter 3.1.2, only over 12 % of all annotations for the chromatin accessibility dataset were defined directly after the collection of metadata from source databases. This improved to 27.8 % after the refinement procedure as several very specific and therefore sparsely defined attributes were eliminated. For gene expression datasets, the situation was somewhat better with 40.3 % of defined labels for TCD and 42.8 % for SCDD as the data came from fewer larger sources. Such high level of incompleteness motivated us to attempt metadata augmentation through ML approaches.

3.3.1 Classifier testing and selection

We tested various types of ML classifiers using all available chromatin accessibility datasets (for FANTOM5, ENSEMBL and INDEX references of REs) and their PCA-transformed versions, trying to find optimal parameters of these models as well as a suitable probability threshold P (see chapter 2.2.1). We always changed a single parameter at a time when comparing the models so we could assess its impact on classification performance. This strategy does not necessarily lead to finding optimal parameters as specific combinations of settings may yield improved results. However, due to a considerable amount of variables that could influence classification, it was not feasible to exhaustively search the entire parameter space.

Classification results were summarised per metadata attributes – for each attribute (e.g. “tissue”), a separate results table was generated containing the performance scores of all corresponding classifiers (i.e. binary classifiers for all attribute values, such as “liver” or “brain”). Due to the considerable extent of these tables, only an example was included in this work and can be found in the appendix (table 5). As k-fold stratified cross-validation was employed as the testing strategy, all performance scores – accuracy, recall, precision – were calculated as the means of k values obtained from individual testing rounds, with range of the computed metrics being included in results tables as well. This information was further complemented by processing time needed to train the classifiers and generate performance scores to examine how certain ML models or settings influence computational demands. A classifier was constructed and tested for all metadata values with at least $k = 5$ entries available because to evaluate performance, both positive and negative class must be represented in each of the cross-validation folds.

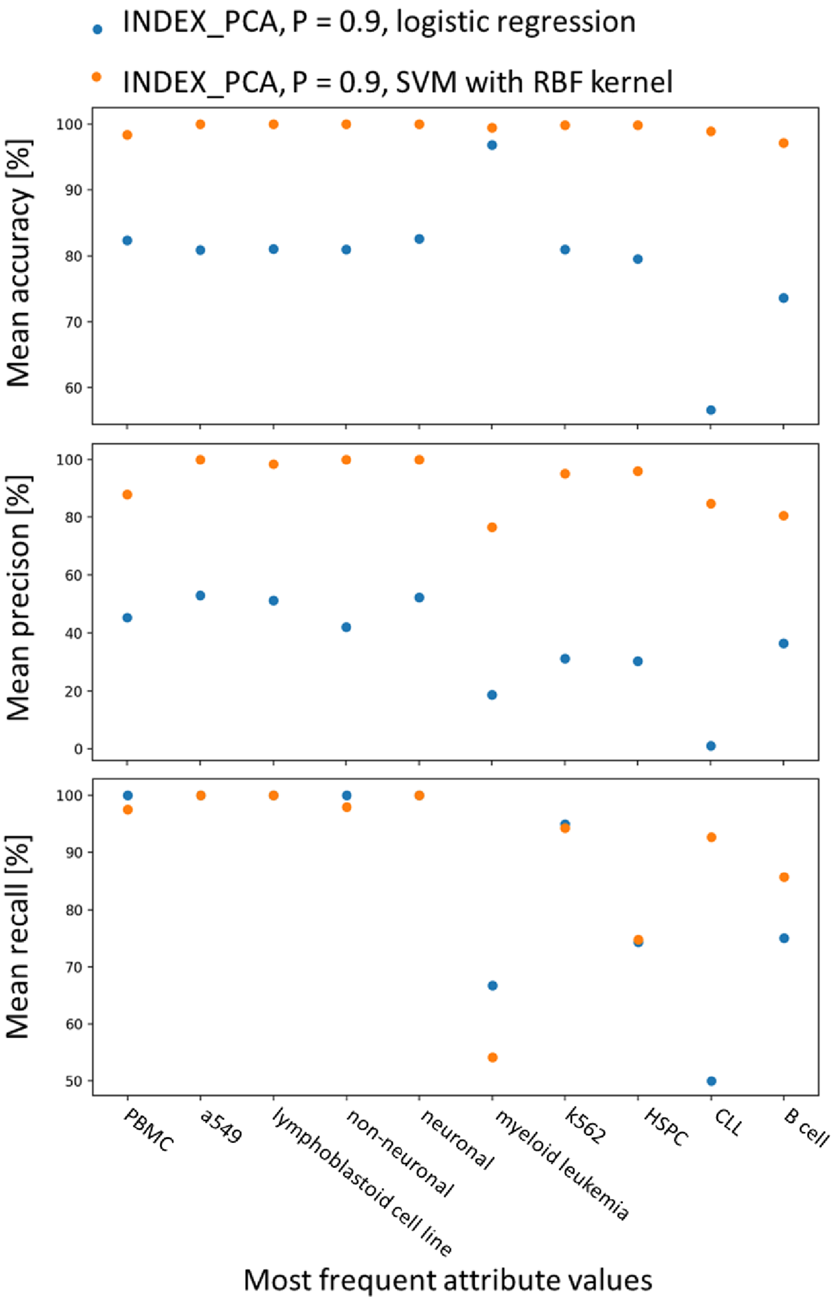


Figure 17: Summary plots showing the comparison between the performance of two different ML classifier types – logistic regression and SVM with RBF kernel – applied on the same training data (PCA-transformed chromatin accessibility dataset aggregated with respect to INDEX reference of REs) and with the same probability threshold ($P = 0.9$). Mean performance scores are plotted on the y-axes, labels of x-axes are the 10 most frequent values of "biosample_term_name" attribute (a binary classifier was constructed for each of these metadata labels). It can be seen that for this particular attribute, SVM yields consistently higher accuracy and precision than logistic regression, while recall is comparable for most of the binary classifiers.

To compare classification performance between different model types, datasets and parameter settings, we summarised generated results tables into plots for better readability. An example of such visualisation can be seen in figure 17, where the performance of two ML models is compared. We assessed classifier results per individual metadata attributes (such as "biosample_term_name" in figure 17), each time plotting the scores of at most 10 binary classifiers corresponding to attribute values with the highest frequency. This restriction was imposed for the sake of readability – for attributes with many unique values (sometimes in the order of hundreds), it would be infeasible to include the scores of all classifiers into a single plot in a sensible manner. The restriction is also justified by the fact that for the majority of attributes, only their most frequent values are of interest for training ML models as other values do not have enough positive training examples available. By visually comparing accuracy, precision and recall of individual binary classifiers, we could decide whether a certain combination of ML model, chromatin accessibility dataset and specific settings provided superior results.

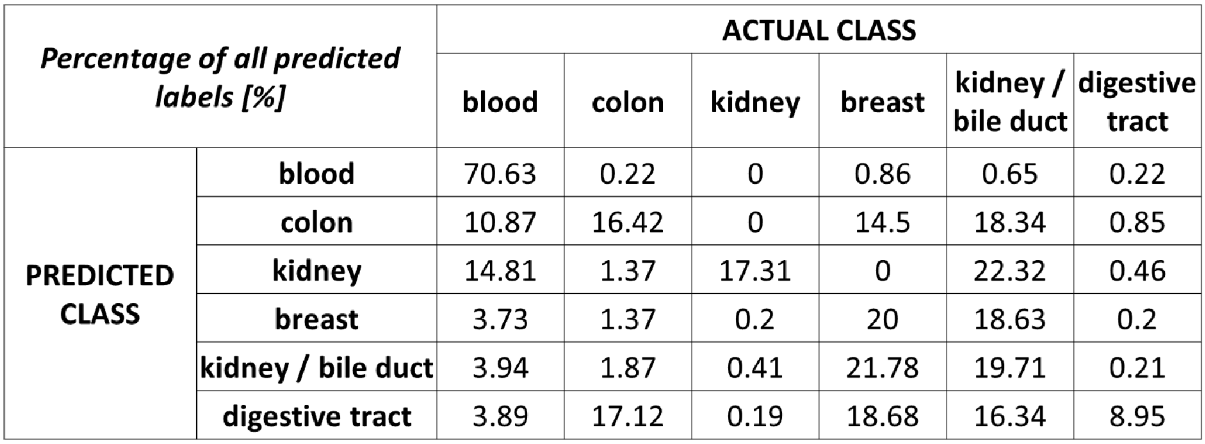
For the example illustrated in figure 17, it is relatively straightforward to conclude that one set of classifier settings outperforms the other as it yields higher performance scores for the vast majority of trained classifiers. However, the situation was often much less clear – comparing a pair of parameter sets, it could happen that the number of classifiers with better performance was comparable between the two cases. Sometimes, it was even problematic to assess which of the two compared binary classifiers performs better, for example if one yielded higher precision and the other one higher recall etc. In these situations, we turned to a more detailed way of classifier evaluation – confusion matrices. The structure of these matrices with the explanation of their contents can be found in table 1. In some cases, the confusion matrix may reveal that a classifier, although having poor performance scores, in fact confuses closely related classes and its real performance is therefore better than original assessment indicated (classifier for "digestive tract" in table 1b is an example of such case). Also, a "cleaner" confusion matrix (i.e. with mostly zero values outside the main diagonal) indicates a good separation between classes as confusions occur infrequently.

Using the introduced evaluation tools, we tried to select the best-performing ML model and the corresponding settings, both in terms of classification quality and computational efficiency. Moreover, it was assessed how the choice of testing dataset influences the results. In general, we observed that for all models, the classification performance is comparable when the classifier is tested with different versions of the chromatin accessibility dataset (i.e. aggregated with respect to FANTOM5, ENSEMBL and INDEX references of REs) – each of the datasets is therefore suitable for further use. For linear models, it was computationally feasible to process the aggregated data directly so we could compare performance with the same classifiers applied on PCA-transformed data (see figure 26 in the appendix for an example of such comparison). As there was no discernible difference in classification quality, we concluded that the first 100 extracted PCs cover a sufficient portion of data variation and can therefore be used for further analyses instead of the original data, which greatly reduces computational demands.

As for the choice of probability threshold P, various values were tested ranging from $P = 0.5$

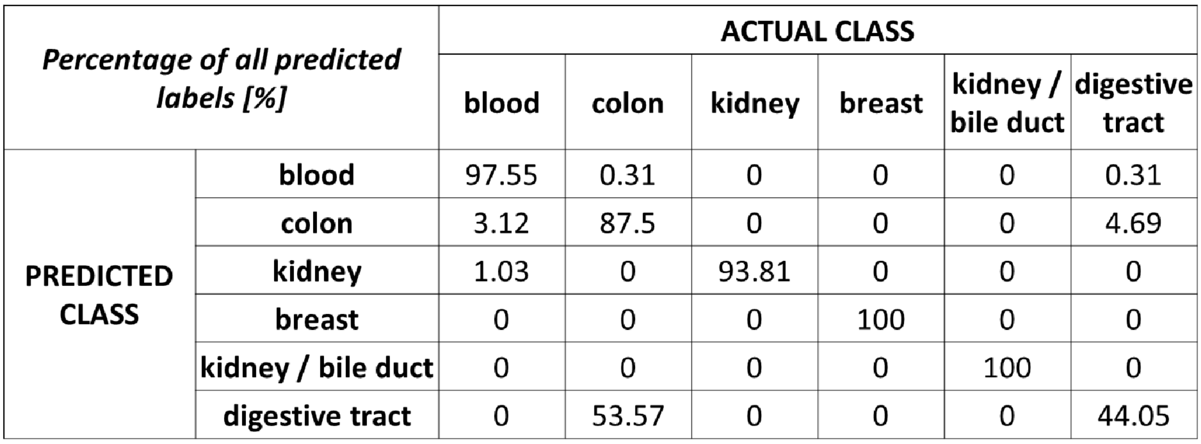
Table 1: Excerpts from confusion matrices generated for classifiers predicting “tissue” attribute, tested on PCA-transformed chromatin accessibility dataset aggregated with respect to ENSEMBL reference of REs with probability threshold $P = 0.9$. Individual cells of the matrix contain percentages expressing how many of the test samples classified as belonging to a particular class (i.e. predicted class, stated in rows of the matrix) are labelled with individual annotations taken from the metadata (i.e. actual class, corresponding to the matrix columns). For example, in matrix a), 70.63 % of samples classified as “blood” were really blood samples, 0.22 % were samples from colon, 0.86 % from breast etc. Therefore, percentages on the main diagonal represent precisions of individual classifiers, numbers in each row sum up to 100 % (not necessarily in these examples as the matrices are not complete). It can be seen that for logistic regression (table a)), there is a considerable confusion between classes with precisions of some classifiers being very low. On the contrary, SVM with RBF kernel is able to distinguish classes more clearly (table b)) with significant misclassifications occurring mainly for biologically similar or overlapping classes (notice confusion rates between “colon” and “digestive tract”).

(a) ENSEMBL_PCA, P = 0.9, logistic regression



Percentage of all predicted labels [%]		**ACTUAL CLASS**					
		blood	**colon**	**kidney**	**breast**	**kidney / bile duct**	**digestive tract**
PREDICTED CLASS	**blood**	70.63	0.22	0	0.86	0.65	0.22
	colon	10.87	16.42	0	14.5	18.34	0.85
	kidney	14.81	1.37	17.31	0	22.32	0.46
	breast	3.73	1.37	0.2	20	18.63	0.2
	kidney / bile duct	3.94	1.87	0.41	21.78	19.71	0.21
	digestive tract	3.89	17.12	0.19	18.68	16.34	8.95

(b) ENSEMBL_PCA, P = 0.9, SVM with RBF kernel



Percentage of all predicted labels [%]		**ACTUAL CLASS**					
		blood	**colon**	**kidney**	**breast**	**kidney / bile duct**	**digestive tract**
PREDICTED CLASS	**blood**	97.55	0.31	0	0	0	0.31
	colon	3.12	87.5	0	0	0	4.69
	kidney	1.03	0	93.81	0	0	0
	breast	0	0	0	100	0	0
	kidney / bile duct	0	0	0	0	100	0
	digestive tract	0	53.57	0	0	0	44.05

(i.e. no thresholding) up to $P = 0.9$. Generally, a slight decrease in overall classifier performance was observed with decreasing threshold. This is demonstrated in figure 27 in the appendix for the chosen model type and metadata attribute. We arrived at the final threshold $P = 0.8$ as a compromise between improving classification performance and increasing the number of predicted labels (as with larger P, fewer predictions are accepted and the computed scores are therefore less reliable).

In the category of linear classifiers, logistic regression ended up to be the best-performing model, slightly outperforming linear SVM (usually in the order of percents for individual performance scores). Processing time was comparable for both models, ranging usually from 5 to 7 seconds for constructing a single binary classifier. However, applying SVM with RBF kernel (a non-linear model) on the PCA-transformed datasets brought an overall classification improvement, yielding higher performance scores for the majority of binary classifiers – across all attributes, more than 70 % of computed performance scores were equal or higher for the non-linear SVM than for logistic regression. In particular, there was an increase in prediction precision for most models while recall was often comparable or slightly lower. This outcome was further supported by manually examining the corresponding confusion matrices, which showed a more favourable structure for the non-linear SVM (e.g. as demonstrated in table 1). Moreover, computation time was slightly reduced compared to logistic regression (usually up to 1 second per classifier).

Based on these results, we chose SVM with RBF kernel as the ML model best suitable for our purposes, despite the general shortcomings of non-linear classifiers described in chapter 2.2.1. After further parameter optimisation, we arrived at the following configuration of the implemented model to be used for subsequent analyses:

```
sklearn.svm.SVC(C = 1000, probability = True, cache_size = 1000,
class_weight = 'balanced')
```

All the other settings of `sklearn.svm.SVC` class were kept at their default values as per *scikit-learn*'s documentation[8], the probability threshold was set to $P = 0.8$ as mentioned previously. The key parameter of SVMs to be determined is positive constant C, which weights the regularisation term of the model. In the utilised implementation, lower C corresponds to stronger regularisation (i.e. the model has higher bias and lower variance and is therefore less prone to overfitting) and vice versa. We assessed classification performance for various values of C, ranging from numbers nearing zero up to tens of thousands (i.e. practically no regularisation). We observed improvement in performance with increasing value up to approximately $C = 1000$, above which the effect on classification results was no longer discernible (besides, high C values had a negative impact on computational time). Although classifier testing was performed only with the chromatin accessibility datasets, this configuration was re-used also for the gene expression data. Naturally, a more suitable set of parameters (or even a different ML model) could probably be found to better address the characteristics of a different data type. However,

[8] Available from: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC [September 11 2020]

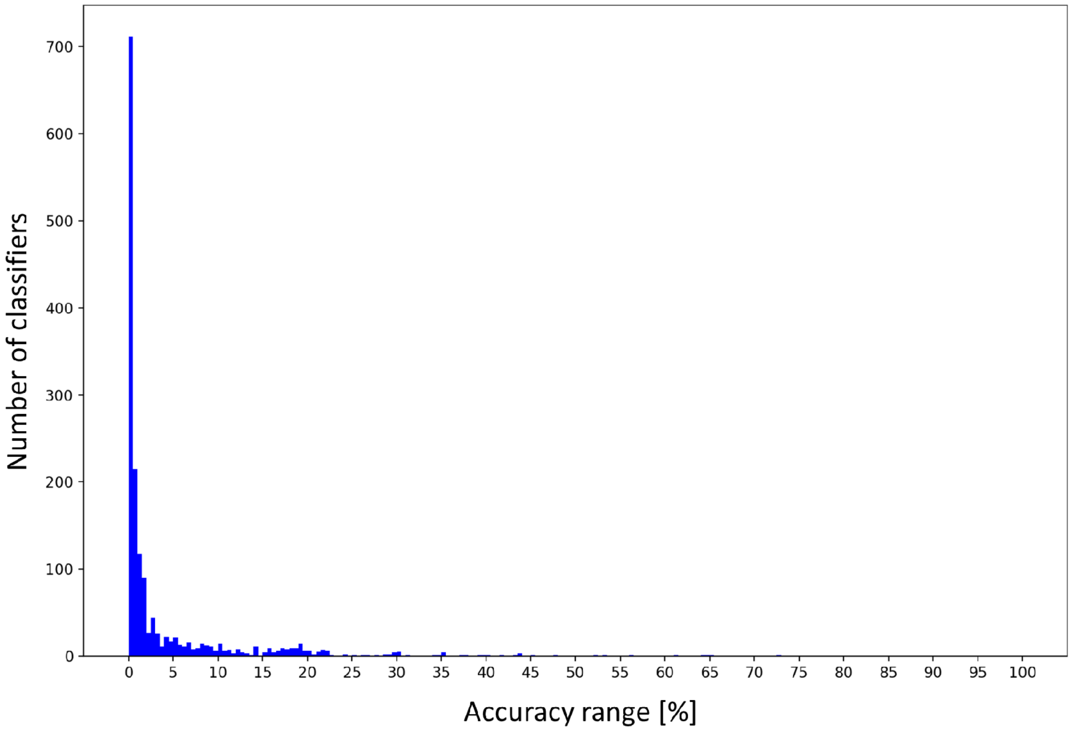


Figure 18: Distribution of the range of accuracy as yielded by 5-fold stratified cross-validation of all classifiers tested on the chromatin accessibility datasets. In total, there were 1611 binary classifiers tested using chromatin accessibility data aggregated with respect to FANTOM5, ENSEMBL and INDEX references of REs. Note that some of these classifiers were constructed for identical metadata labels but tested with different training data. Bin size is 0.5 %.

the limited time frame and scope of the thesis did not allow for extensive testing in case of the gene expression datasets.

With the classification model established, the next step was to select binary classifiers that performed “sufficiently well” to be used for metadata augmentation. This was not a straightforward task as it can be difficult to determine which classifiers are good enough for further use. Nonetheless, using the computed performance scores, we needed to define unambiguous rules based on which such selection could be made. First, we imposed thresholds on the range of performance scores yielded during individual cross-validation rounds. This initial step excludes classifiers which performed inconsistently during testing and ensures that the mean scores are not biased by potential outliers. To set appropriate values of thresholds for individual performance scores, we plotted distributions of scores’ range across all tested classifiers. An example of such distribution for accuracy range can be seen in figure 18. Based on these visualisations, we could make data-driven decisions regarding the choice of filtering parameters.

Having examined the corresponding distributions, we set rejection thresholds for the range of individual performance scores as follows:

1. accuracy: $R_a = 15$ %
2. precision: $R_p = 30$ %
3. recall: $R_r = 35$ %

These are maximum allowed ranges of performance scores – for a classifier to be accepted, all of the conditions must be satisfied. In the next filtration round, we imposed thresholds on the means of performance metrics as the minimum values required to accept a classifier:

1. accuracy: $M_a = 95$ %
2. precision: $M_p = 90$ %
3. recall: $M_r = 25$ %

Again, the thresholds were determined based on inspection of corresponding distributions of mean performance scores across all tested classifiers. All mean scores must be equal or higher than these thresholds for a classifier to be accepted. Note that the threshold for mean recall is much lower than for the other two metrics – this is because we can accept for our purposes a classifier with relatively poor recall provided it has good precision (such classifier would probably miss many samples that actually belong to the predicted class but it can be expected that the samples it detects would be classified correctly, which is preferred to making no predictions at all). All thresholds are identical also for the gene expression datasets except for R_a, which was set to 5 % for TCD and 10 % for SCDD.

The last filtering condition is based on the composition of training sets for the classifiers. A classifier was rejected if the proportion of positive training examples in its training set was higher than P. This step is supposed to eliminate models which could be biased because of imbalanced training data. In such a case, classifiers tend to assign the dominant positive class (mostly incorrectly) to the majority of newly classified samples. Finally, for the classifiers that were rejected *only* due to low mean precision, the confusion matrices were manually checked to find out whether this was not caused by the confusion of similar classes. In such cases, the classifier could be additionally accepted. Assessment of class similarity was performed manually, based on the limited knowledge of biology, and therefore we accepted the classifier only when the relationship between confused classes was very clear.

Summary results of the described selection procedure are given in table 2, where numbers of accepted and rejected classifiers (after imposing all aforementioned conditions) for individual training datasets are stated. Note that for most of the metadata labels it was not possible to test the classifier due to the low amount of training examples. Specifically, the model was not constructed if there were not enough samples to represent both the positive and the negative class in each of the 5 cross-validation folds. This happened most commonly when there were less than 5 samples labelled with a particular attribute value present in a dataset.

Table 2: The summary of classifier selection step. FANTOM5, ENSEMBL and INDEX labels refer to the chromatin accessibility dataset aggregated with respective catalogues of REs. Classifiers with insufficient amount of training examples were not tested.

Dataset	***number of classifiers***		
	accepted	**rejected**	**not tested**
FANTOM5	171	363	1849
ENSEMBL	182	359	1867
INDEX	175	361	1801
TCD	150	337	222
SCDD	73	893	1287

3.3.2 Iterative training and prediction

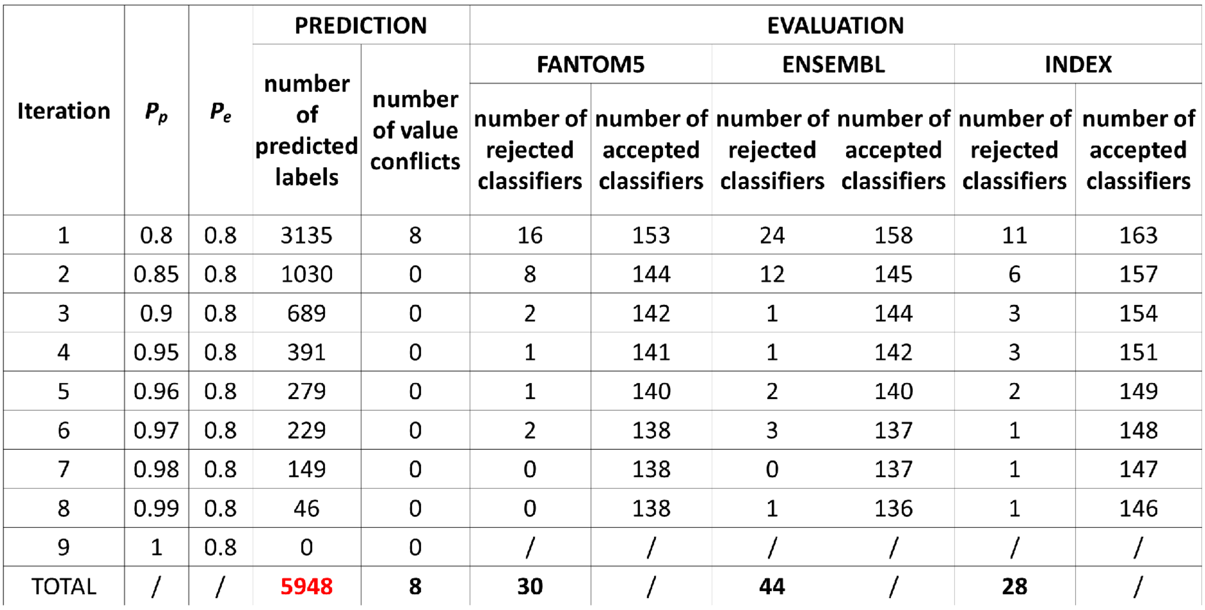
Taking the set of ML classifiers selected in the previous step, we could proceed to the prediction of undefined metadata labels through the iterative workflow described in chapter 2.2.2. The prediction was performed separately for the chromatin accessibility dataset and the two gene expression datasets, using the classifiers trained on the corresponding data. The results of metadata augmentation for the chromatin accessibility dataset are summarised in table 3. For each iteration, the following information is given: probability thresholds for prediction P_p and evaluation P_e (their meaning is explained in chapter 2.2.2), the number of newly predicted metadata labels, the number of value conflicts (occurring when a single sample is assigned more values of a particular attribute, see chapter 2.2.2), the number of classifiers rejected after being tested with augmented metadata and the number of remaining accepted classifiers. Note that while P_e is kept constant throughout all iterations, the probability threshold for prediction is incrementally increased to ensure convergence of the whole procedure.

Results tables for the gene expression datasets are included in the appendix (table 6 for TCD and table 7 for SCDD). In total, more than 22,000 new metadata labels were predicted across all datasets with only a small number of conflicting predictions.

3.3.3 Discussion

We developed a strategy for improving the completeness of sample metadata corresponding to the assembled integrated datasets, utilising ML classifiers trained with existing annotations to predict undefined metadata entries. Based on an extensive testing phase, we identified models best suitable for our purposes and determined appropriate parameter settings. Using standard metrics for assessment of classification performance, we empirically defined rules for selecting best-performing classifiers to be used for prediction of annotations. The implemented iterative procedure of metadata augmentation helped to increase the number of newly predicted labels and fully exploit the potential of constructed classifiers. The cumulation of prediction errors

Table 3: Summary results of metadata augmentation for the chromatin accessibility dataset. For each version of the dataset (aggregated with FANTOM5, ENSEMBL and INDEX references of REs), a separate set of classifiers was tested and selected. Re-evaluation of these classifiers during metadata augmentation was performed separately as well and therefore in each iteration, a different group of classifiers could be rejected for individual dataset versions. Prediction of metadata labels was stopped in the 9th iteration as no new annotations were predicted.



Iteration	***P_p***	***P_e***	**PREDICTION**		**EVALUATION**					
			number of predicted labels	**number of value conflicts**	**FANTOM5**		**ENSEMBL**		**INDEX**	
					number of rejected classifiers	**number of accepted classifiers**	**number of rejected classifiers**	**number of accepted classifiers**	**number of rejected classifiers**	**number of accepted classifiers**
1	0.8	0.8	3135	8	16	153	24	158	11	163
2	0.85	0.8	1030	0	8	144	12	145	6	157
3	0.9	0.8	689	0	2	142	1	144	3	154
4	0.95	0.8	391	0	1	141	1	142	3	151
5	0.96	0.8	279	0	1	140	2	140	2	149
6	0.97	0.8	229	0	2	138	3	137	1	148
7	0.98	0.8	149	0	0	138	0	137	1	147
8	0.99	0.8	46	0	0	138	1	136	1	146
9	1	0.8	0	0	/	/	/	/	/	/
TOTAL	/	/	**5948**	**8**	**30**	/	**44**	/	**28**	/

during the procedure was counteracted by increasing reliability of predictions in each iteration. Moreover, the developed method is a generally applicable strategy for improving quality of metadata needed for any supervised ML approach.

Apart from enhancing the quality of annotations, a valuable outcome of metadata augmentation step are the trained ML models. These were stored (in separate binary files) to be later used for classification of new samples. In the next chapter, it will be discussed how they can be exploited to capture biologically relevant information and aid interpretation of cryptic biomedical datasets.

3.4 Classification of new samples

In the final phase of the project, we used ML models trained during metadata augmentation to classify cell samples that were not included in the training data. We collected several datasets of gene expression and chromatin accessibility profiles to which the classifiers were applied. An overview of these testing datasets is given in table 4. It can be seen that the data comprise bulk and single-cell RNA-seq and single-cell ATAC-seq experiments from various studies, from both publicly available sources and internal data generated at CCRI. To comprehensively validate the classifiers, we tried to cover in these datasets a variety of cell types in terms of biological properties, such as tissue, disease or developmental stage.

Since it would be out of the scope of this work to comprehensively describe classification results achieved for each of the testing datasets, in the following sections only representative examples are introduced to demonstrate the validity of our approach. The most straightfor-

Table 4: An overview of testing datasets on which the trained ML classifiers were applied. COPD = Chronic Obstructive Pulmonary Disease, BCP-ALL = B Cell Precursor Acute Lymphoblastic Leukemia, MNC = Mono-Nuclear Cell, LCH = Langerhans Cell Histiocytosis.

GENE EXPRESSION	**bulk RNA-seq**	public source	Fagerberg et al. (2014) [77]	samples from 27 different tissues of healthy donors
			Kim et al. (2015) [78]	lung tissue from COPD subjects and healthy controls
			Varley et al. (2014) [79]	breast cancer cell lines and primary tumours
		CCRI	Strehl lab	leukemic cells from various types of BCP-ALL
			Taschner-Mandl lab	sorted neuroblastoma cells, bone marrow infiltrates, MNCs and healthy controls
	single-cell RNA-seq	public source	Hay, Ferchen et al. (2018) [80]	bone marrow samples from the Human Cell Atlas
			Pellin et al. (2019) [81]	human hematopoietic progenitors
			Olsen et al. (2020) [82]	neuroblastoma biopsies
		CCRI	Hutter lab	LCH biopsies
CHROMATIN ACCESSIBILITY	**single-cell ATAC-seq**	public source	Sathpaty et al. (2019) [83]	peripheral blood and bone marrow cells of healthy donors and a carcinoma patient

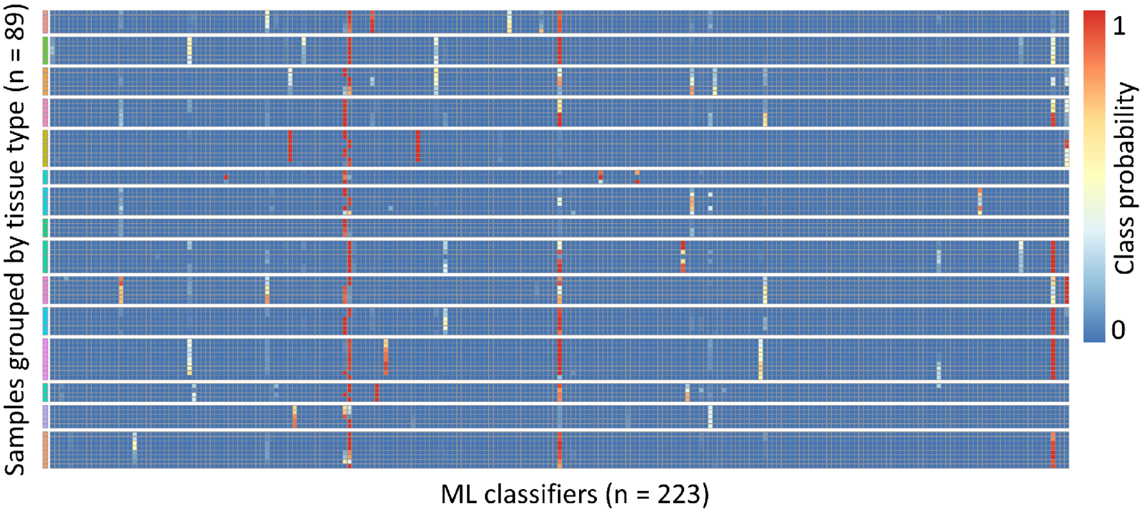


Figure 19: A visualisation of classification results for bulk RNA-seq data of healthy tissue samples from [77]. Columns of the heatmap represent the applied ML classifiers (i.e. the classes for which classification of samples was performed), rows correspond to individual cell samples, grouped by tissue type (each horizontal lane of the heatmap, delimited by empty rows, contains samples from the same tissue) and colour coding reflects the computed probability estimates. The labels of samples and classifiers were not included here for better readability. Note that some classifiers produce high estimates across most of the samples (regardless of tissue type) whereas others yield high probabilities only for a particular sample group. As shown later, these patterns reflect biologically relevant information.

ward way to visualise the outcomes of the classification procedure is through a heatmap. An example of such visualisation for bulk RNA-seq dataset comprising cell samples from various healthy tissues (data from [77]) can be seen in figure 19. General patterns that we observed in the heatmap indicated that the classifiers captured biologically relevant information which was available for the samples (i.e. tissue type in this case). However, to validate this claim, we needed to examine the classification results in more detail.

In figure 20, a detailed excerpt taken from the previous heatmap is shown. It serves as a demonstration that individual ML models we trained produce meaningful outcomes when compared to the limited amount of metadata available for the testing datasets. Moreover, these results provide a validation that the constructed classifiers are able to restore biologically relevant information not only in the training data but also in unseen samples – the models show good generalisation ability.

Moreover, the performed classification may be perceived as a mapping of the original feature space (with features being genes for gene expression data and REs for chromatin accessibility data) to a new domain, in which each feature corresponds to a particular predicted class and the probability estimates are new feature values. Hereafter, we will refer to this domain as the “functional space” as it helps us describe cell samples in comprehensible biological terms. Importantly, we can visualise datasets in the “functional space” similarly as in the original domain, i.e. using already established techniques such as t-SNE or UMAP.

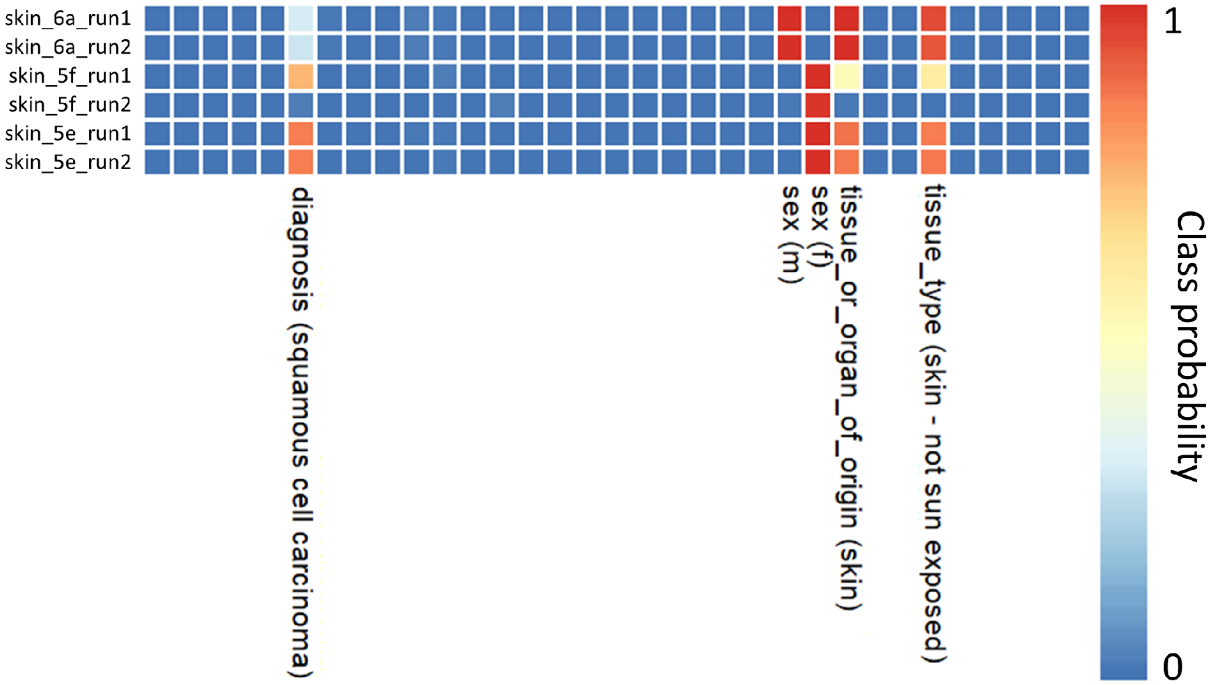
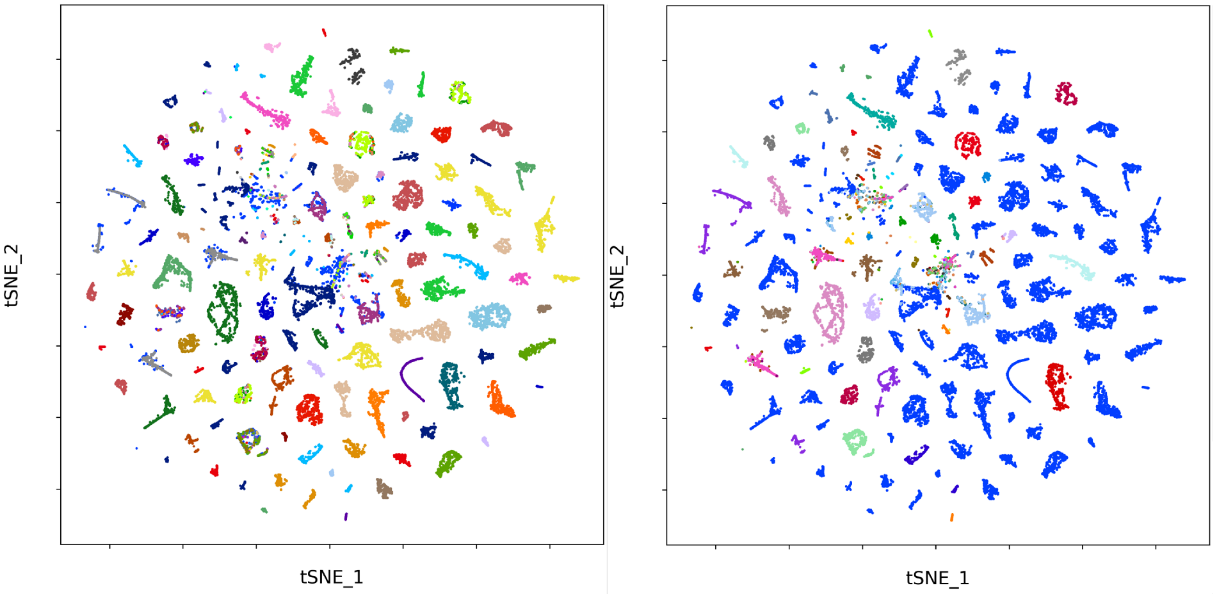


Figure 20: An excerpt from the heatmap in figure 19 showing probability estimates computed for a group of 6 skin samples. It can be seen that higher probabilities are given by the classifiers related to skin cells – either directly predicting tissue type (e.g. "tissue_or_organ_of_origin (skin)") or a disease ("diagnosis (squamous cell carcinoma)"). Note that there are 3 different biological samples in the heatmap, each with 2 technical replicates ("run1" and "run2"). For the first and the last biological sample, the probability estimates are practically identical for the corresponding technical replicates, as would be expected. The second replicate of sample "skin_5f", however, is not detected by the relevant classifiers and has different probabilities computed than the first replicate – this may be indicative of a low-quality or damaged sample (however, we do not have the necessary metadata available to support this hypothesis).

First, we wanted to validate that the collection of trained ML classifiers recapitulates biological differences between samples in the training data. Therefore, we transformed probability estimates calculated for the training datasets into 2-dimensional space using t-SNE and plotted the obtained embeddings – the results of this procedure for one of the gene expression training datasets are presented in figure 21. The clustering of samples visible in the plots corresponds to their labelling based on biological metadata attributes, which supports our hypothesis that biologically relevant information in the training data is restored by the trained classification models.

The next step was to create similar visualisations for the testing datasets. With these, the interpretation and assessment of classification results may be more complicated as the data are not well-characterised (from a biological perspective) and the amount of corresponding metadata is limited. Single-cell datasets are particularly challenging in this respect – samples are extracted from heterogeneous tissues and therefore the type and function of individual cells



(a) Grouping attribute: “tissue or organ of origin”

(b) Grouping attribute: “disease”

Figure 21: A visualisation of Tissue and Cancer Dataset in the “functional space” using t-SNE. The samples are colour-coded according to two different biological attributes – tissue (figure a)) and disease (figure b)). The labels of individual groups were not included due to their considerable amount and because they are not necessary to demonstrate that biological differences were restored in the “functional space” as can be deduced from the visible clustering of identically labelled samples.

are usually unknown. Here, we decided to present results for the single-cell RNA-seq dataset from [82], which contains gene expression profiles of more than 60,000 cells from 17 different neuroblastoma biopsies. Figure 22 contains UMAP plots which show how these cells cluster based on their single-cell gene expression profiles and how the clustering is restored (at least partially) in the “functional space”.

3.4.1 Discussion

We applied ML models constructed during augmentation of metadata for the training gene expression and chromatin accessibility datasets to classify new samples which were not used in the training phase. Before this classification could be performed, several technical issues had to be addressed concerning the format of input data – we needed to achieve identical structure and apply the same pre-processing steps as for the training datasets (see chapter 2.3). Subsequently, we could proceed to classify samples from a varied collection of testing datasets gathered from publicly available and internal sources.

Through visualisation, we demonstrated that the set of trained ML classifiers can recapitulate the overall data structure in both training and testing datasets. In particular, biologically relevant information was captured by the models which is of utmost importance if these are to be used

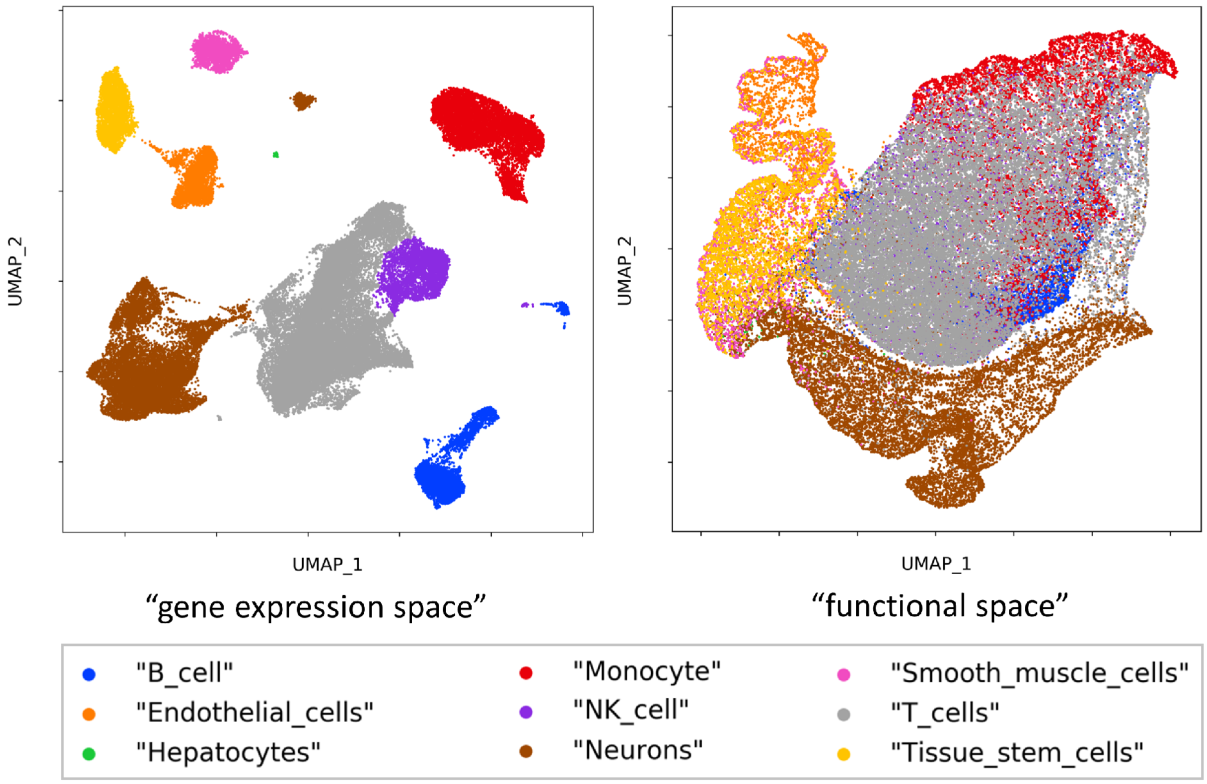


Figure 22: A comparison between UMAP visualisation of the single-cell RNA-seq dataset from [82] in the original (gene expression) domain and in the "functional space". Note that the sample labels are not real annotations (as such metadata were not available) but predicted cell types generated in a previous analysis conducted at CCRI, based on the clustering of cells according to their RNA-seq profiles. It can be seen that this clustering is to some extent restored in the "functional space" as well. However, to aid further interpretation of such data, there would need to be a better separation of sample groups visible.

for providing interpretations of poorly annotated data. By carefully examining the outputs of individual classifiers and comparing them with metadata available for the testing datasets, we verified that the majority of trained models produce meaningful predictions and are therefore able to generalise patterns learned from the training data.

The reliability of predictions for single-cell data, however, was less convincing. Most of the classifiers yielded very low probability estimates for the majority of classified profiles, showing also a systematic bias towards the prediction of certain classes (i.e. some models produced consistently high probabilities for practically all single-cell samples across different datasets). See figure 28 in the appendix for the example of visualisations that support this finding. In a way, such an outcome is not surprising considering that exclusively bulk data were used for the training of classifiers. Single-cell profiles are commonly very sparse (they contain zero values for most genes and other genomic regions) and therefore it can be expected that models trained on bulk data are not able to distinguish single-cell samples well based on relatively subtle changes of gene expression/chromatin accessibility between individual cells.

Finally, the extent of biological insight provided by the classifiers is limited by the contents of annotations for the training data. This is a property inherent to the supervised approach we employed – the models can predict only those classes that are defined in the metadata (and have enough training examples to represent them). Metadata quality is in turn determined by the submission requirements of individual data sources, which are in general very inconsistent. Although we attempted to address this issue (through the developed refinement and augmentation procedures), more work will need to be invested into improving the informational value of sample annotations beyond descriptive biological properties, for example through exploiting ontologies of biomedical terms (see chapter 4).

4 Conclusion

In this work, we aimed to exploit well-characterised public data yielded by transcriptomic and epigenomic assays to aid interpretation of newly generated datasets in comprehensible biological terms. To do so, we first assembled an integrated resource of chromatin accessibility data from publicly available sources, collecting results of DNase-seq and ATAC-seq experiments performed on human cells. To this end, we developed an automated framework for parallel downloading and aggregation of genome-wide chromatin accessibility profiles. Through aggregation, we achieved that chromatin accessibility was evaluated for the same set of genomic regions in all collected samples – this unification was necessary for further computational processing and also significantly reduced volume of the original data.

To diminish technical variation inherently present in the chromatin accessibility dataset due to heterogeneity of the collected samples, we employed normalisation and standardisation methods commonly used for processing of genomic data. By doing so, we were able to make quantitative data more comparable and therefore suitable for downstream analyses. The desired effect of these pre-processing steps was verified by numerous visualisations of the dataset, enabled by advanced dimensionality reduction techniques.

Apart from primary quantitative data, we programmatically collected and unified the annotations corresponding to the gathered cell samples. We again faced the problem of heterogeneity in the metadata, which we addressed by developing a semi-automated procedure for the refinement of annotations. Using mainly text processing, we were able to improve the consistency of metadata to make them more usable for computational analysis. Storing all the collected data in standard, easily accessible formats and creating a framework for straightforward addition of new data, we aimed to support re-usability of the dataset. Such resource has therefore potential to be exploited not only for the purposes of this work but also for further bioinformatic studies, considering increasing popularity and importance of assaying chromatin accessibility in molecular biology research.

As the sample metadata were still vastly incomplete after collection and refinement, we developed an augmentation strategy in which we used supervised ML classifiers trained on existing annotations to predict undefined metadata labels. Based on extensive testing, we identified models and parameter settings suitable for this purpose. Subsequently, standard performance metrics were utilised to assess classification quality and choose the best-performing models for metadata augmentation. This approach was applied also on gene expression datasets of RNA-seq profiles, assembled previously within a different research project at CCRI. Although the number of added annotations was relatively low compared to the total amount of undefined metadata entries, we argue that the informational value of metadata was improved by predicting biologically relevant labels.

In the last phase of the project, trained ML models constructed during metadata augmentation were utilised to classify new, poorly annotated samples from RNA-seq and ATAC-seq experiments. Using these testing datasets, we verified the generalisation ability of classifiers – they were able to extract biologically relevant information from the training data and transfer the learnt patterns during the classification of unseen samples. While the models performed reasonably well for bulk transcriptomic data, the reliability of results for single-cell samples will have to be further investigated. As the training data consisted of exclusively bulk samples, it is also possible that the classifiers are not at all suitable for processing single-cell profiles. Moreover, the choice of analytical methods and the corresponding parameters was conditioned by optimising performance on the specific collection of training data – more suitable settings could probably be found for the classification of each newly analysed dataset.

Novel biological insight brought by the classification procedure, which could aid the interpretation of cryptic biomedical datasets, is currently limited by the contents of sample annotations available for the training data. Therefore, the classifiers presented in this work are able to predict mostly descriptive biological attributes (such as tissue or disease types), which are usually known for the examined samples. One way to overcome this limitation would be to turn to an unsupervised ML approach, which does not rely on the labelling of training examples but rather extracts structures inherently present in the processed dataset. Subsequently, further analyses would have to be performed to match such patterns to biologically meaningful interpretations (e.g. through examining the expression of genes or the activity of REs associated with certain cellular processes or regulatory pathways).

Another possible direction for future work is further enrichment of metadata with more intricate information, for example by using publicly available ontologies which associate standardised biomedical terms with various regulatory pathways or the expression of certain genes. If the contents of annotations were enhanced in such a way, our supervised strategy could yield more informative results. Taken together, this thesis provides evidence that the developed approach is able to transfer knowledge hidden in large collections of public data to poorly characterised datasets. Therefore, we believe that, with additional work, it holds promise in helping to understand complex data produced during transcriptomic and epigenomic studies, particularly in the area of cancer research.

Bibliography

[1] F. H. Crick, “On protein synthesis,” *Symposia of the Society for Experimental Biology*, vol. 12, pp. 138–163, 1958.

[2] T. Phillips, “Regulation of transcription and gene expression in eukaryotes,” *Nature Education*, vol. 1, p. 199, 2008.

[3] C. Rye, *Biology*. Houston, Texas: OpenStax College, Rice University, 2017.

[4] P. J. Wittkopp and G. Kalay, “Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence,” *Nature Reviews Genetics*, vol. 13, no. 1, pp. 59–69, dec 2012.

[5] H. K. Long, S. L. Prescott, and J. Wysocka, “Ever-changing landscapes: Transcriptional enhancers in development and evolution,” *Cell*, vol. 167, no. 5, pp. 1170–1187, nov 2016.

[6] G. M. Cooper, “Regulation of transcription in eukaryotes,” in *The Cell: A Molecular Approach*, 2nd ed. Sunderland (MA): Sinauer Associates, 2000.

[7] A. Kundaje *et al.*, “Integrative analysis of 111 reference human epigenomes,” *Nature*, vol. 518, no. 7539, pp. 317–330, feb 2015.

[8] I. Dunham *et al.*, “An integrated encyclopedia of DNA elements in the human genome,” *Nature*, vol. 489, no. 7414, pp. 57–74, sep 2012.

[9] G. Thiel, M. Lietz, and M. Hohl, “How mammalian transcriptional repressors work,” *European Journal of Biochemistry*, vol. 271, no. 14, pp. 2855–2862, jun 2004.

[10] C. A. Molina, N. S. Foulkes, E. Lalli, and P. Sassone-Corsi, “Inducibility and negative autoregulation of CREM: An alternative promoter directs the expression of ICER, an early response repressor,” *Cell*, vol. 75, no. 5, pp. 875–886, dec 1993.

[11] G. Thiel, M. Lietz, and M. Cramer, “Biological activity and modular structure of RE-1-silencing transcription factor (REST), a repressor of neuronal genes,” *Journal of Biological Chemistry*, vol. 273, no. 41, pp. 26 891–26 899, oct 1998.

[12] B. Pierce, *Genetics : a conceptual approach*. New York: W.H. Freeman, 2012.

[13] C. L. Woodcock and R. P. Ghosh, “Chromatin higher-order structure and dynamics,” *Cold Spring Harbor Perspectives in Biology*, vol. 2, no. 5, p. a000596, apr 2010.

[14] M. Radman-Livaja and O. J. Rando, “Nucleosome positioning: How is it established, and why does it matter?” *Developmental Biology*, vol. 339, no. 2, pp. 258–266, mar 2010.

[15] T. Phillips, “The role of methylation in gene expression,” *Nature Education*, vol. 1, p. 116, 2008.

[16] G. M. Cooper, “The development and causes of cancer,” in *The Cell: A Molecular Approach*, 2nd ed. Sunderland (MA): Sinauer Associates, 2000.

[17] H. Chial, “Proto-oncogenes to oncogenes to cancer,” *Nature Education*, vol. 1, p. 33, 2008.

[18] S. Zhou *et al.*, “Noncoding mutations target cis-regulatory elements of the FOXA1 plexus in prostate cancer,” *Nature Communications*, vol. 11, no. 1, jan 2020.

[19] P. A. Jones and S. B. Baylin, “The epigenomics of cancer,” *Cell*, vol. 128, no. 4, pp. 683–692, feb 2007.

[20] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, “Quantitative monitoring of gene expression patterns with a complementary DNA microarray,” *Science*, vol. 270, no. 5235, pp. 467–470, oct 1995.

[21] D. Shalon, S. J. Smith, and P. O. Brown, “A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization.” *Genome Research*, vol. 6, no. 7, pp. 639–645, jul 1996.

[22] R. Stark, M. Grzelak, and J. Hadfield, “RNA sequencing: the teenage years,” *Nature Reviews Genetics*, vol. 20, no. 11, pp. 631–656, jul 2019.

[23] T. Baslan and J. Hicks, “Unravelling biology and shifting paradigms in cancer with single-cell sequencing,” *Nature Reviews Cancer*, vol. 17, no. 9, pp. 557–569, aug 2017.

[24] Z. Wang, M. Gerstein, and M. Snyder, “RNA-seq: a revolutionary tool for transcriptomics,” *Nature Reviews Genetics*, vol. 10, no. 1, pp. 57–63, jan 2009.

[25] G. B. Stefano *et al.*, “Comparing bioinformatic gene expression profiling methods: Microarray and RNA-seq,” *Medical Science Monitor Basic Research*, vol. 20, pp. 138–141, 2014.

[26] B. T. Wilhelm and J.-R. Landry, “RNA-seq – quantitative measurement of expression through massively parallel RNA-sequencing,” *Methods*, vol. 48, no. 3, pp. 249–257, jul 2009.

[27] S. L. Klemm, Z. Shipony, and W. J. Greenleaf, “Chromatin accessibility and the regulatory epigenome,” *Nature Reviews Genetics*, vol. 20, no. 4, pp. 207–220, jan 2019.

[28] A. P. Boyle *et al.*, “High-resolution mapping and characterization of open chromatin across the genome,” *Cell*, vol. 132, no. 2, pp. 311–322, jan 2008.

[29] D. S. Gross and W. T. Garrard, "Nuclease hypersensitive sites in chromatin," *Annual Review of Biochemistry*, vol. 57, no. 1, pp. 159–197, jun 1988.

[30] J. R. Hesselberth *et al.*, "Global mapping of protein-DNA interactions in vivo by digital genomic footprinting," *Nature Methods*, vol. 6, no. 4, pp. 283–289, mar 2009.

[31] J. D. Buenrostro, P. G. Giresi, L. C. Zaba, H. Y. Chang, and W. J. Greenleaf, "Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position," *Nature Methods*, vol. 10, no. 12, pp. 1213–1218, oct 2013.

[32] S. Davis. AtacSeqWorkshop 0.1.1. [Online]. Available: https://seandavi.github.io/AtacSeqWorkshop/articles/Workflow.html#background

[33] J. D. Buenrostro, B. Wu, H. Y. Chang, and W. J. Greenleaf, "ATAC-seq: A method for assaying chromatin accessibility genome-wide," *Current Protocols in Molecular Biology*, vol. 109, no. 1, jan 2015.

[34] B. Bolstad, R. Irizarry, M. Astrand, and T. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, vol. 19, no. 2, pp. 185–193, jan 2003.

[35] B. Durbin, J. Hardin, D. Hawkins, and D. Rocke, "A variance-stabilizing transformation for gene-expression microarray data," *Bioinformatics*, vol. 18, no. Suppl 1, pp. S105–S110, jul 2002.

[36] C. Workman *et al.*, "A new non-linear normalization method for reducing variability in DNA microarray experiments," *Genome Biology*, vol. 3, no. 9, 2002.

[37] M.-A. Dillies *et al.*, "A comprehensive evaluation of normalization methods for illumina high-throughput RNA sequencing data analysis," *Briefings in Bioinformatics*, vol. 14, no. 6, pp. 671–683, sep 2012.

[38] J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit, "Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments," *BMC Bioinformatics*, vol. 11, no. 1, feb 2010.

[39] D. Amaratunga and J. Cabrera, "Analysis of data from viral DNA microchips," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1161–1170, dec 2001.

[40] R. A. Irizarry *et al.*, "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, no. 2, pp. 249–264, apr 2003.

[41] S. C. Hicks and R. A. Irizarry, "*quantro*: a data-driven approach to guide the choice of an appropriate normalization method," *Genome Biology*, vol. 16, no. 1, jun 2015.

[42] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 2002.

[43] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, jun 2010.

[44] H. Abdi, “The eigen-decomposition:eigenvalues and eigenvectors,” in *Encyclopedia of Measurement and Statistics*, N. J. Salkind, Ed. Thousand Oaks (CA): Sage, 2007.

[45] L. van der Maaten and G. Hinton, “Visualizing high-dimensional data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, nov 2008.

[46] G. E. Hinton and S. T. Roweis, “Stochastic neighbor embedding,” in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2003, pp. 857–864.

[47] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, mar 1951.

[48] D. Kobak and P. Berens, “The art of using t-SNE for single-cell transcriptomics,” *Nature Communications*, vol. 10, no. 1, nov 2019.

[49] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” *ArXiv e-prints*, no. 1802.03426, dec 2018.

[50] N. Zheng and J. Xue, “Manifold learning,” in *Statistical Learning and Pattern Analysis for Image and Video Processing*. Springer London, 2009, pp. 87–119.

[51] E. Becht *et al.*, “Dimensionality reduction for visualizing single-cell data using UMAP,” *Nature Biotechnology*, vol. 37, no. 1, pp. 38–44, dec 2018.

[52] A. Diaz-Papkovich, L. Anderson-Trocmé, C. Ben-Eghan, and S. Gravel, “UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts,” *PLOS Genetics*, vol. 15, no. 11, p. e1008432, nov 2019.

[53] C. Xu and S. A. Jackson, “Machine learning and complex biological data,” *Genome Biology*, vol. 20, no. 1, apr 2019.

[54] S. Russell and P. Norvig, *Artificial intelligence : a modern approach*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2010.

[55] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Springer New York, 2013.

[56] S. E. Ellis, L. Collado-Torres, A. Jaffe, and J. T. Leek, “Improving the value of public RNA-seq expression data by phenotype prediction,” *Nucleic Acids Research*, vol. 46, no. 9, p. e54, mar 2018.

[57] J. Barretina *et al.*, “The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity,” *Nature*, vol. 483, pp. 603–607, mar 2012.

[58] M. J. Garnett *et al.*, “Systematic identification of genomic markers of drug sensitivity in cancer cells,” *Nature*, vol. 483, pp. 570–575, mar 2012.

[59] L. C. Stetson, T. Pearl, Y. Chen, and J. S. Barnholtz-Sloan, “Computational identification of multi-omic correlates of anticancer therapeutic response,” *BMC Genomics*, vol. 15, no. S2, oct 2014.

[60] Y.-C. Chiu *et al.*, “Predicting drug response of tumors from integrated genomic profiles by deep neural networks,” *BMC Medical Genomics*, vol. 12, no. 18, jan 2019.

[61] H. Yuan, I. Paskov, H. Paskov, A. J. González, and C. S. Leslie, “Multitask learning improves prediction of cancer drug sensitivity,” *Scientific Reports*, vol. 6, no. 31619, aug 2016.

[62] J. Tan, M. Ung, C. Cheng, and C. S. Greene, “Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders,” in *Pacific Symposium on Biocomputing 2015*. World Scientific, jan 2015, pp. 132–143.

[63] A. Gupta, H. Wang, and M. Ganapathiraju, “Learning structure in gene expression data using deep architectures, with an application to gene clustering,” in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, nov 2015.

[64] H. Cui *et al.*, “Boosting gene expression clustering with system-wide biological information: a robust autoencoder approach,” *International Journal of Computational Biology and Drug Design*, vol. 13, no. 1, pp. 98–123, 2020.

[65] G. P. Way and C. S. Greene, “Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders,” in *Pacific Symposium on Biocomputing 2018*. World Scientific, jan 2018, pp. 80–91.

[66] A. B. Dincer, S. Celik, N. Hiranuma, and S.-I. Lee, “DeepProfile: Deep learning of cancer molecular profiles for precision medicine,” *bioRxiv*, may 2018.

[67] L. Rampášek, D. Hidru, P. Smirnov, B. Haibe-Kains, and A. Goldenberg, “Dr.VAE: improving drug response prediction via modeling of drug perturbation effects,” *Bioinformatics*, vol. 35, no. 19, pp. 3743–3751, mar 2019.

[68] J. N. Taroni *et al.*, “MultiPLIER: A transfer learning framework for transcriptomics reveals systemic features of rare disease,” *Cell Systems*, vol. 8, no. 5, pp. 380–394.e4, may 2019.

[69] R. Andersson *et al.*, “An atlas of active enhancers across human cell types and tissues,” *Nature*, vol. 507, no. 7493, pp. 455–461, mar 2014.

[70] D. R. Zerbino, S. P. Wilder, N. Johnson, T. Juettemann, and P. R. Flicek, “The ensembl regulatory build,” *Genome Biology*, vol. 16, no. 1, mar 2015.

[71] W. Meuleman *et al.*, “Index and biological spectrum of accessible DNA elements in the human genome,” *bioRxiv*, 2019.

[72] T. F. Chan, G. H. Golub, and R. J. Leveque, “Algorithms for computing the sample variance: Analysis and recommendations,” *The American Statistician*, vol. 37, no. 3, pp. 242–247, aug 1983.

[73] S. Oki *et al.*, “ChIP -atlas: a data-mining suite powered by full integration of public ChIP -seq data,” *EMBO reports*, vol. 19, no. 12, nov 2018.

[74] C. A. Davis *et al.*, “The encyclopedia of DNA elements (ENCODE): data portal update,” *Nucleic Acids Research*, vol. 46, no. D1, pp. D794–D801, jan 2018.

[75] M. R. Corces *et al.*, “The chromatin accessibility landscape of primary human cancers,” *Science*, vol. 362, no. 6413, 2018.

[76] R. Edgar, M. Domrachev, and A. E. Lash, “Gene expression omnibus: NCBI gene expression and hybridization array data repository,” *Nucleic Acids Research*, vol. 30, no. 1, pp. 207–210, jan 2002.

[77] L. Fagerberg *et al.*, “Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics,” *Molecular & Cellular Proteomics*, vol. 13, no. 2, pp. 397–406, feb 2014.

[78] W. J. Kim *et al.*, “Comprehensive analysis of transcriptome sequencing data in the lung tissues of COPD subjects,” *International Journal of Genomics*, vol. 2015, pp. 1–9, 2015.

[79] K. E. Varley *et al.*, “Recurrent read-through fusion transcripts in breast cancer,” *Breast Cancer Research and Treatment*, vol. 146, no. 2, pp. 287–297, jun 2014.

[80] S. B. Hay, K. Ferchen, K. Chetal, H. L. Grimes, and N. Salomonis, “The human cell atlas bone marrow single-cell interactive web portal,” *Experimental Hematology*, vol. 68, pp. 51–61, dec 2018.

[81] D. Pellin *et al.*, “A comprehensive single cell transcriptional landscape of human hematopoietic progenitors,” *Nature Communications*, vol. 10, no. 1, jun 2019.

[82] T. K. Olsen *et al.*, “Malignant schwann cell precursors mediate intratumoral plasticity in human neuroblastoma,” *bioRxiv*, may 2020.

[83] A. T. Satpathy *et al.*, “Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral T cell exhaustion,” *Nature Biotechnology*, vol. 37, no. 8, pp. 925–936, aug 2019.

List of Figures

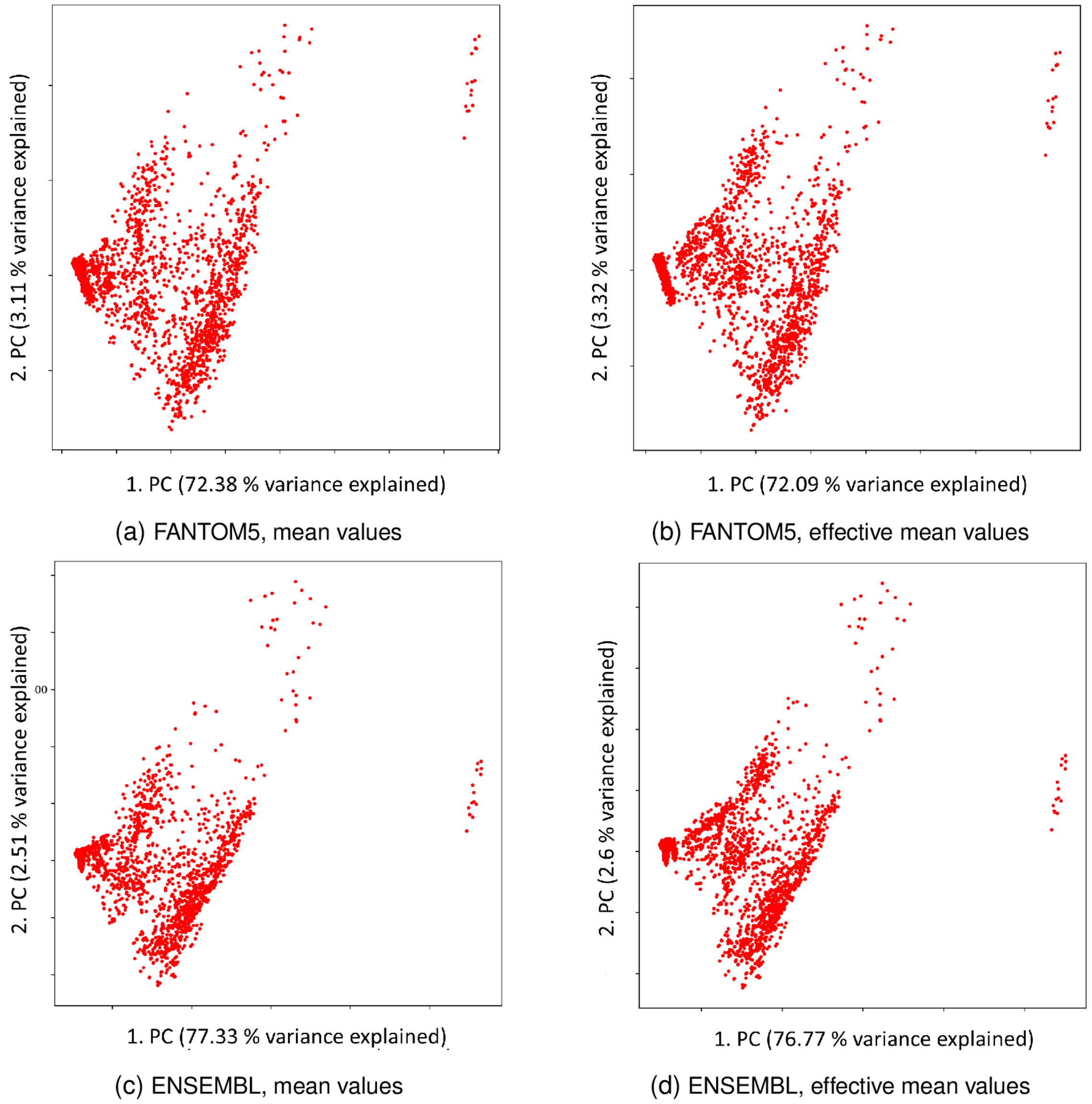
List of Tables

List of Abbreviations

DNA	deoxyribonucleic acid
RNA	ribonucleic acid
RE	regulatory element
NGS	next-generation sequencing
rRNA	ribosomal RNA
mRNA	messenger RNA
cDNA	complementary DNA
DNase	deoxyribonuclease I
DHS	DNase hypersensitive site
ATAC-seq	assay for transposase-accessible chromatin using sequencing
QN	quantile normalisation
PCA	principal component analysis
PC	principal component
t-SNE	t-distributed stochastic neighbor embedding
SNE	stochastic neighbor embedding
UMAP	uniform manifold approximation and projection
ML	machine learning
SVM	support vector machine
RBF	radial basis function
TCGA	The Cancer Genome Atlas
UCSC	University of California Santa Cruz
TSV	tab-separated values

SGD	stochastic gradient descent
HGNC	HUGO Gene Nomenclature Committee
CCRI	St. Anna Children’s Cancer Research Institute
ENCODE	Encyclopedia of DNA Elements
GEO	Gene Expression Omnibus
SOFT	Simple Omnibus Format in Text
TCD	Tissue and Cancer Dataset
SCDD	Stem Cell Differentiation Dataset

A Appendix: Supplementary figures



(a) FANTOM5, mean values

(b) FANTOM5, effective mean values

(c) ENSEMBL, mean values

(d) ENSEMBL, effective mean values

Figure 23: Plots of the PCA-transformed chromatin accessibility dataset – each figure shows the second PC plotted against the first one. Figures in the upper row represent data aggregated with respect to FANTOM5 reference – in figure a), mean values were computed during aggregation, effective mean values were used for figure b). As can be seen, there are very little changes in the overall data structure as well as in the variance explained by PCs between the two cases. A similar conclusion can be drawn for data aggregated with ENSEMBL reference (figures c) and d)) and INDEX reference (plots for INDEX were not included for the sake of brevity).

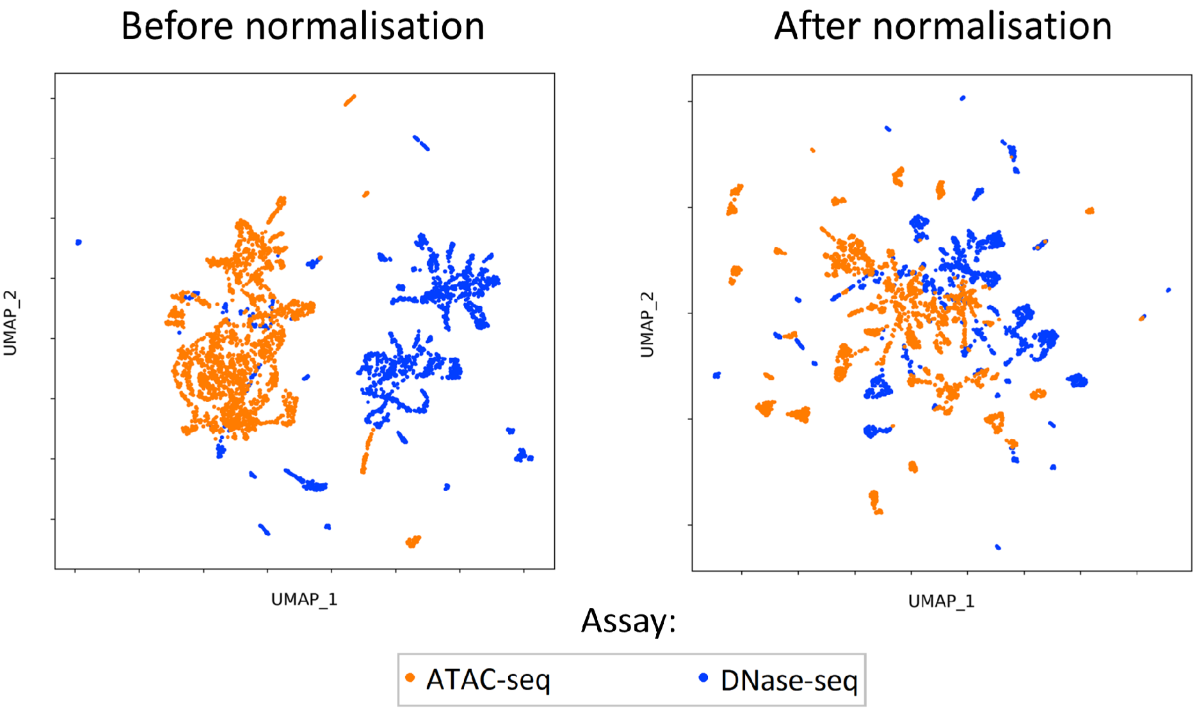


Figure 24: A comparison between visualisation of the chromatin accessibility dataset before and after QN using UMAP. QN normalisation reduced the clustering of samples according to a technical parameter – assay used to measure chromatin accessibility. The data were aggregated with respect to ENSEMBL reference.

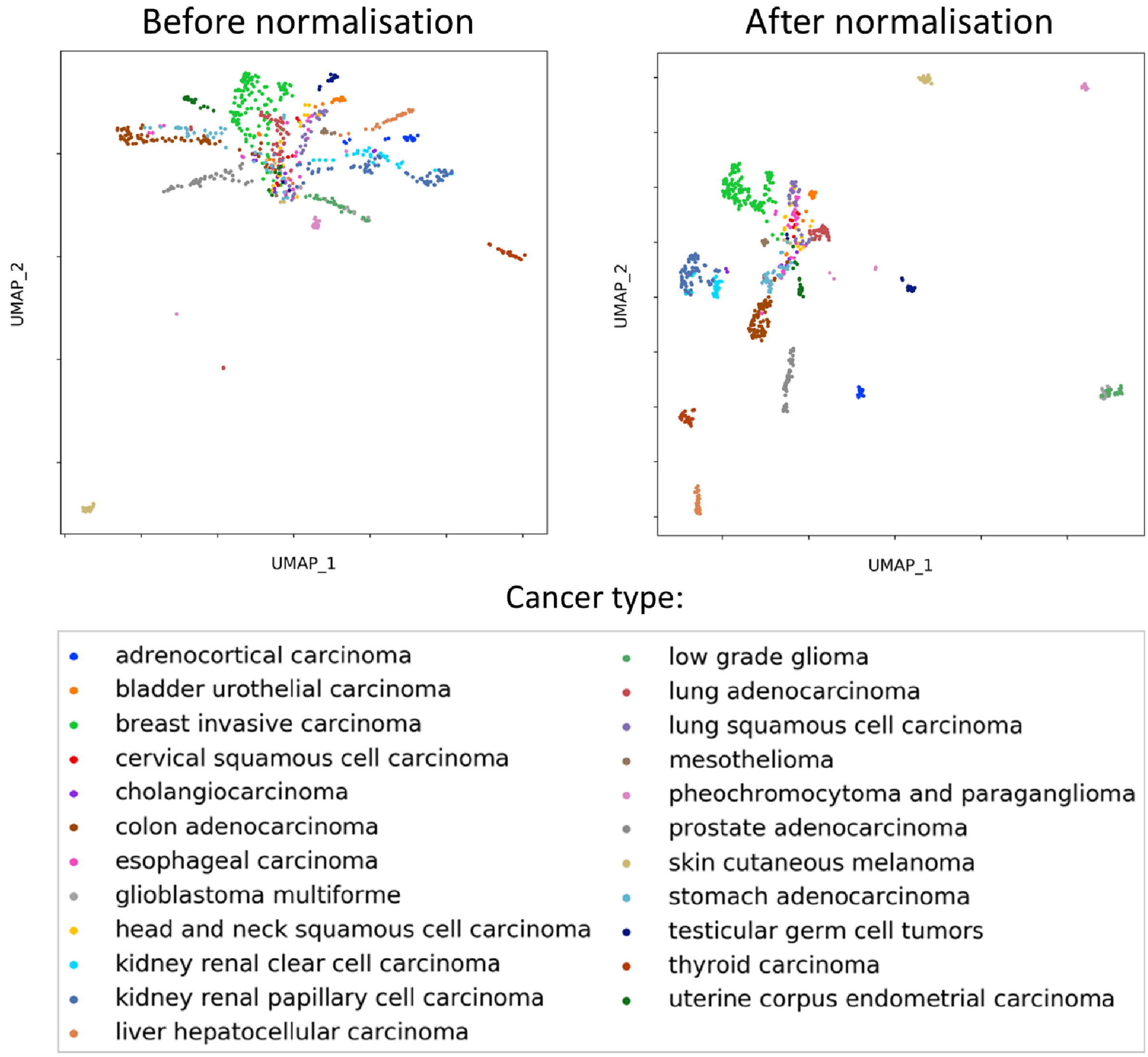


Figure 25: A comparison between visualisation of the chromatin accessibility dataset before and after QN using UMAP. The clustering of samples according to their biological similarity – cancer type – remains visible after normalisation. The data were aggregated with respect to INDEX reference, only a subset of the whole dataset (for which cancer type was defined) is visualised.

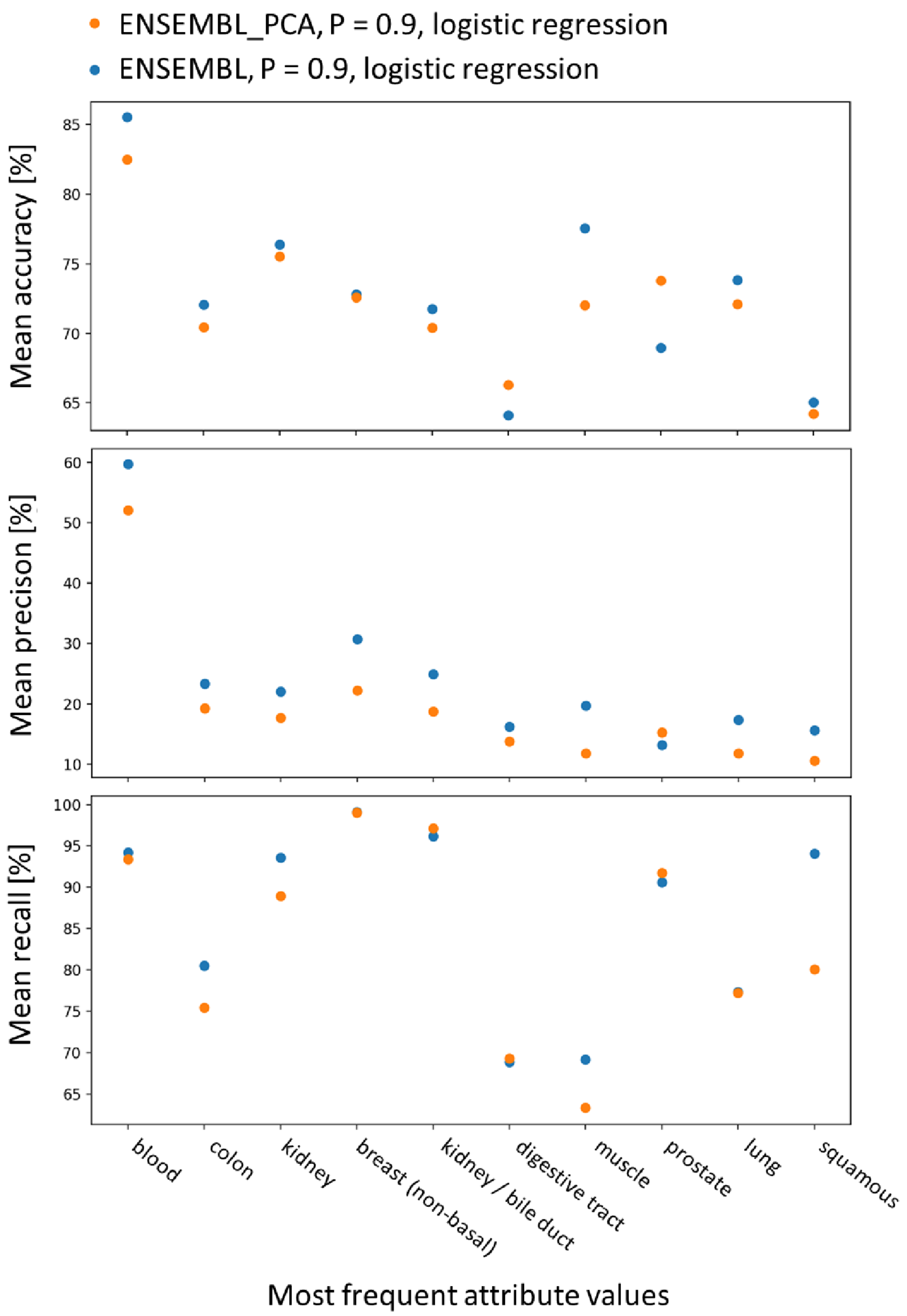


Figure 26: Summary plots showing the comparison between performance of logistic regression classifier applied on the aggregated and PCA-transformed chromatin accessibility dataset with the uniform probability threshold $P = 0.9$. The data were aggregated with respect to ENSEMBL reference of REs, the first 100 PCs were extracted during PCA. Mean performance scores are plotted on the y-axes, labels of x-axes are the 10 most frequent values of “tissue” attribute. It can be seen that the performance of linear classifiers is comparable when applied to the original and PCA-transformed data (i.e. neither of the classifiers outperforms the other consistently). Similar results were obtained also for other metadata attributes.

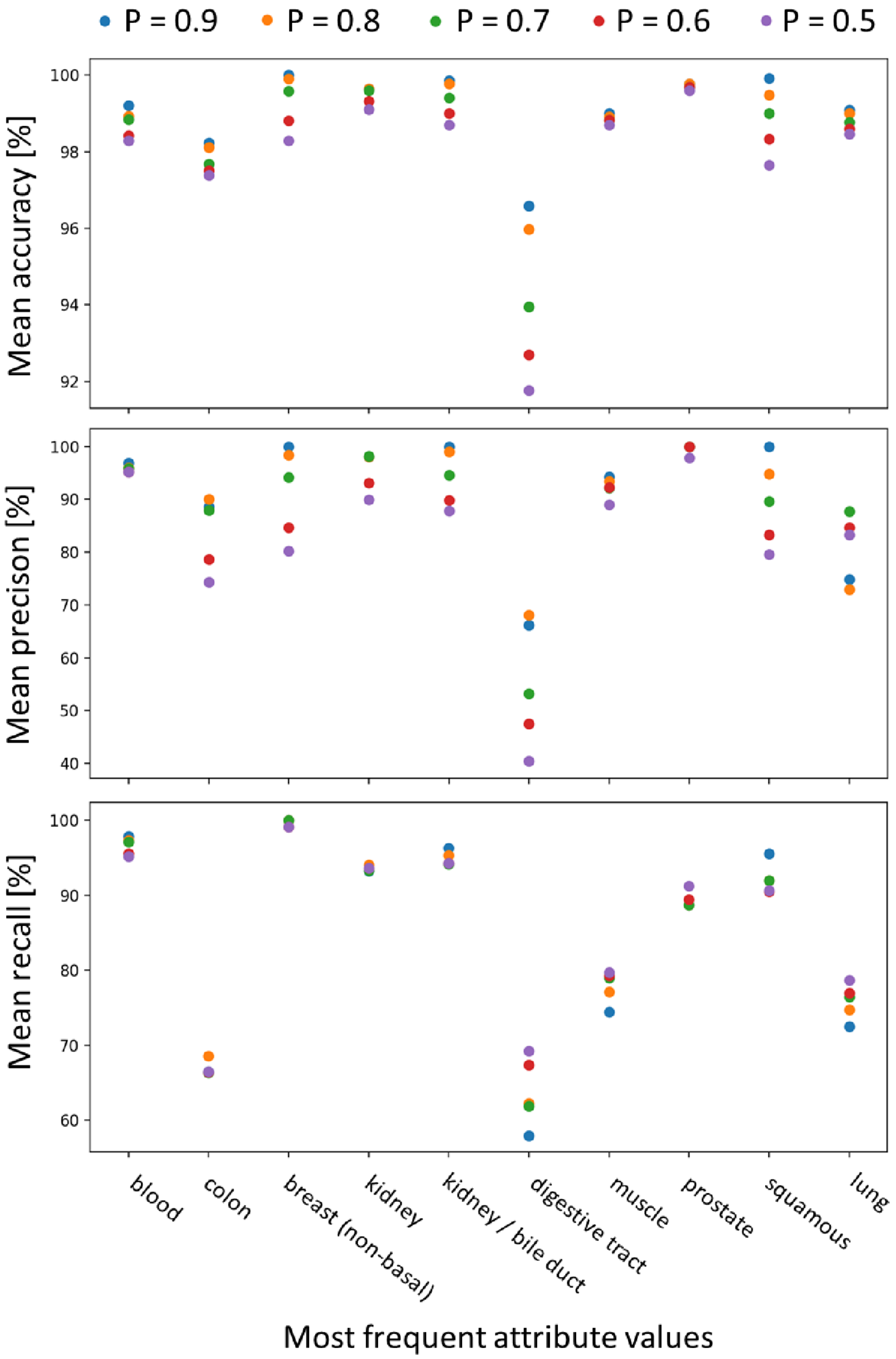
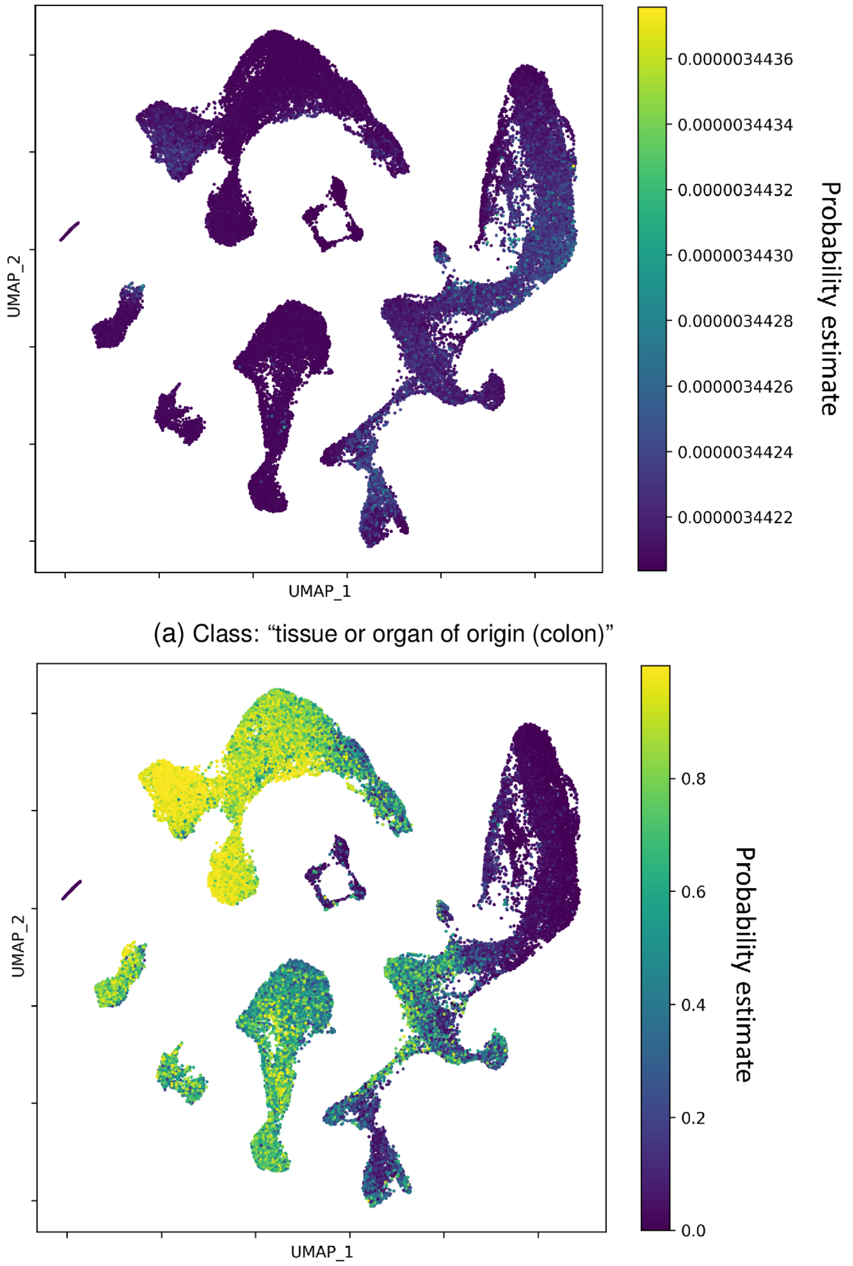


Figure 27: Summary plots showing the comparison between performance of SVMs with RBF kernel applied on the PCA-transformed chromatin accessibility dataset aggregated with respect to INDEX reference of REs with different values of probability threshold P used. Mean performance scores are plotted on the y-axes, labels of x-axes are the 10 most frequent values of “tissue” attribute. It can be seen that with increasing threshold P, there is an improvement in accuracy and precision for the majority of classifiers while recall is mostly comparable.



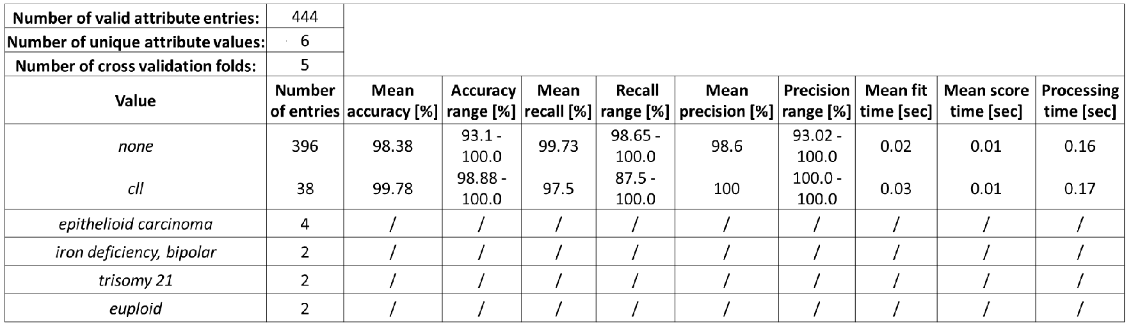
(a) Class: “tissue or organ of origin (colon)”

(b) Class: “refinebio cell line (hues2)”

Figure 28: Visualisation of classification results for single-cell RNA-seq data from [80]. UMAP embeddings of gene expression profiles are colour-coded according to the probability estimates assigned to each sample by a particular classifier. In a), classification results for “tissue or organ of origin (colon)” classifier are shown. Note that all samples were assigned very low probabilities – a similar situation was observed for most other classifiers as well. On the contrary, certain models yielded high probability estimates for some of the samples. In b), outputs of one of such classifiers (“refinebio cell line (hues2)”) are visualised. To a certain extent, classification results correspond to the clustering of samples in UMAP coordinates. However, similar patterns were observed for this particular classifier also when applied to other (unrelated) datasets – it is therefore probable that the classifier does not produce reliable predictions.

B Appendix: Supplementary tables

Table 5: An example of the results table from the testing of ML classifiers. A similar table was created for each metadata attribute, the presented results are for the attribute “disease”. Each attribute value corresponds to a single binary classifier whose performance was evaluated. As the testing was performed through k-fold cross validation (here, $k = 5$), performance scores were averaged across all testing rounds with the mean and range of the values stated in the table. Apart from classification performance, computational times for model training (fitting) and evaluation are given. Note that if there were less than k positive training examples available for a metadata value, the classifier was not constructed and the testing could not be performed.



Number of valid attribute entries:	444								
Number of unique attribute values:	6								
Number of cross validation folds:	5								
Value	**Number of entries**	**Mean accuracy [%]**	**Accuracy range [%]**	**Mean recall [%]**	**Recall range [%]**	**Mean precision [%]**	**Precision range [%]**	**Mean fit time [sec]**	**Mean score time [sec]**	**Processing time [sec]**
none	396	98.38	93.1 - 100.0	99.73	98.65 - 100.0	98.6	93.02 - 100.0	0.02	0.01	0.16
cll	38	99.78	98.88 - 100.0	97.5	87.5 - 100.0	100	100.0 - 100.0	0.03	0.01	0.17
epithelioid carcinoma	4	/	/	/	/	/	/	/	/	/
iron deficiency, bipolar	2	/	/	/	/	/	/	/	/	/
trisomy 21	2	/	/	/	/	/	/	/	/	/
euploid	2	/	/	/	/	/	/	/	/	/

Table 6: Summary results of metadata augmentation for Tissue and Cancer Dataset.

Iteration	P_p	P_e	**PREDICTION**		**EVALUATION**	
			number of predicted labels	**number of value conflicts**	**number of rejected classifiers**	**number of accepted classifiers**
1	0.9	0.8	2951	5	15	135
2	0.92	0.8	2220	0	3	132
3	0.94	0.8	266	0	0	132
4	0.95	0.8	325	0	1	131
5	0.96	0.8	389	0	0	131
6	0.97	0.8	209	0	0	131
7	0.98	0.8	64	0	0	131
8	0.99	0.8	0	0	/	/
TOTAL	/	/	**6424**	**5**	**19**	/

Table 7: Summary results of metadata augmentation for Stem Cell Differentiation Dataset.

Iteration	P_p	P_e	**PREDICTION**		**EVALUATION**	
			number of predicted labels	**number of value conflicts**	**number of rejected classifiers**	**number of accepted classifiers**
1	0.8	0.8	2854	5	12	61
2	0.85	0.8	2186	0	4	57
3	0.9	0.8	1627	0	0	57
4	0.95	0.8	1120	0	0	57
5	0.96	0.8	841	0	0	57
6	0.97	0.8	844	0	1	56
7	0.98	0.8	669	0	3	53
8	0.99	0.8	21	0	0	53
9	1	0.8	0	0	/	/
TOTAL	/	/	**10162**	**5**	**20**	/