Interpretable Machine Learning Methods for Prediction and Analysis of Genome Regulation in 3D

Sarvesh Nikumbh

Dissertation
zur Erlangung des Grades
des Doktors der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

Saarbrücken, Germany
February 2019

Colloquium Date	June 24, 2019
Dean	Prof. Dr. Sebastian Hack

Examination Board

Chairman:	Prof. Dr. Bernt Schiele
First reviewer:	Prof. Dr. Nico Pfeifer
Second Reviewer:	Prof. Dr. Tobias Marschall
Academic Assistant:	Dr. Peter Ebert

Abstract

With the development of chromosome conformation capture-based techniques, we now know that chromatin is packed in three-dimensional (3D) space inside the cell nucleus. Changes in the 3D chromatin architecture have already been implicated in diseases such as cancer. Thus, a better understanding of this 3D conformation is of interest to help enhance our comprehension of the complex, multipronged regulatory mechanisms of the genome. The work described in this dissertation largely focuses on development and application of interpretable machine learning methods for prediction and analysis of long-range genomic interactions output from chromatin interaction experiments.

In the first part, we demonstrate that the genetic sequence information at the genomic loci is predictive of the long-range interactions of a particular locus of interest (LoI). For example, the genetic sequence information at and around enhancers can help predict whether it interacts with a promoter region of interest. This is achieved by building string kernel-based support vector classifiers together with two novel, intuitive visualization methods. These models suggest a potential general role of short tandem repeat motifs in the 3D genome organization. But, the insights gained out of these models are still coarse-grained. To this end, we devised a machine learning method, called *CoMIK* for Conformal Multi-Instance Kernels, capable of providing more fine-grained insights. When comparing sequences of variable length in the supervised learning setting, *CoMIK* can not only identify the features important for classification but also locate them within the sequence. Such precise identification of important segments of the whole sequence can help in gaining *de novo* insights into any role played by the intervening chromatin towards long-range interactions. Although *CoMIK* primarily uses only genetic sequence information, it can also simultaneously utilize other information modalities such as the numerous functional genomics data if available.

The second part describes our pipeline, `pHDee`, for easy manipulation of large amounts of 3D genomics data. We used the pipeline for analyzing HiChIP experimental data for studying the 3D architectural changes in Ewing sarcoma (EWS) which is a rare cancer affecting adolescents. In particular, HiChIP data for two experimental conditions, doxycycline-treated and untreated, and for primary tumor samples is analyzed. We demonstrate that pHDee facilitates processing and easy integration of large amounts of 3D genomics data analysis together with other data-intensive bioinformatics analyses.

Kurzfassung

Mit der Entwicklung von Techniken zur Bestimmung der Chromosomen-Konformation wissen wir jetzt, dass Chromatin in einer dreidimensionalen (3D) Struktur innerhalb des Zellkerns gepackt ist. Änderungen in der 3D-Chromatin-Architektur sind bereits mit Krankheiten wie Krebs in Verbindung gebracht worden. Daher ist ein besseres Verständnis dieser 3D-Konformation von Interesse, um einen tieferen Einblick in die komplexen, vielschichtigen Regulationsmechanismen des Genoms zu ermöglichen. Die in dieser Dissertation beschriebene Arbeit konzentriert sich im Wesentlichen auf die Entwicklung und Anwendung interpretierbarer maschineller Lernmethoden zur Vorhersage und Analyse von weitreichenden genomischen Interaktionen aus Chromatin-Interaktionsexperimenten.

Im ersten Teil zeigen wir, dass die genetische Sequenzinformation an den genomischen Loci prädiktiv für die weitreichenden Interaktionen eines bestimmten Locus von Interesse (LoI) ist. Zum Beispiel kann die genetische Sequenzinformation an und um Enhancer-Elemente helfen, vorherzusagen, ob diese mit einer Promotorregion von Interesse interagieren. Dies wird durch die Erstellung von String-Kernel-basierten Support Vector Klassifikationsmodellen zusammen mit zwei neuen, intuitiven Visualisierungsmethoden erreicht. Diese Modelle deuten auf eine mögliche allgemeine Rolle von kurzen, repetitiven Sequenzmotiven ("tandem repeats") in der dreidimensionalen Genomorganisation hin. Die Erkenntnisse aus diesen Modellen sind jedoch immer noch grobkörnig. Zu diesem Zweck haben wir die maschinelle Lernmethode CoMIK (für Conformal Multi-Instance-Kernel) entwickelt, welche feiner aufgelöste Erkenntnisse liefern kann. Beim Vergleich von Sequenzen mit variabler Länge in überwachten Lernszenarien kann CoMIK nicht nur die für die Klassifizierung wichtigen Merkmale identifizieren, sondern sie auch innerhalb der Sequenz lokalisieren. Diese genaue Identifizierung wichtiger Abschnitte der gesamten Sequenz kann dazu beitragen, de novo Einblick in jede Rolle zu gewinnen, die das dazwischen liegende Chromatin für weitreichende Interaktionen spielt. Obwohl CoMIK hauptsächlich nur genetische Sequenzinformationen verwendet, kann es gleichzeitig auch andere Informationsquellen nutzen, beispielsweise zahlreiche funktionellen Genomdaten sofern verfügbar.

Der zweite Teil beschreibt unsere Pipeline pHDee für die einfache Bearbeitung großer Mengen von 3D-Genomdaten. Wir haben die Pipeline zur Analyse von HiChIP-Experimenten zur Untersuchung von dreidimensionalen Architekturänderungen bei der seltenen Krebsart Ewing-Sarkom (EWS) verwendet, welche Jugendliche betrifft. Insbesondere werden HiChIP-Daten für zwei experimentelle Bedingungen, Doxycyclin-behandelt und unbehandelt, und für primäre Tumorproben analysiert. Wir zeigen, dass pHDee die Verarbeitung und einfache Integration großer Mengen der 3D-Genomik-Datenanalyse zusammen mit anderen datenintensiven Bioinformatik-Analysen erleichtert.

To my parents
To my spiritual guru

Acknowledgments

I thank my advisor and mentor, Nico Pfeifer, for all the things I have learned from him. These range over a wide spectrum, pursuing technical excellence being the foremost. His advise has always helped a great deal in shaping the researcher in me.

I have had the privilege of interacting with Thomas Lengauer. Every encounter with him has been an experience I will cherish throughout my life. Thank you very much Thomas!

I thank the members of the thesis committee for agreeing to serve on the committee and devote their time and efforts for this. Also, special thanks to Peter Ebert for providing with the German translation of the thesis abstract.

I am grateful to the IMPRS-CS and the MPI-INF for providing financial support during this endeavor. I acknowledge the help provided by the administrative staff at our department and thank them for all their support. Life was made a lot more enjoyable by the presence of friends (and friendly colleagues) in and outside university (in no particular order) – Sourav Dutta, Subhabrata Mukherjee (Subho), Kashyap Popat, Dilafruz Amanova, Arunav Mishra, Niket Tandon, Vikram Tankasali, Pratik Jawanpuria, Sairam Gurajada, Pramod Mudrakarta, Adrin Jalali, Nora K. Speicher, Anna Hake (née Feldmann), Matthias Döring, Prabhav Kalaghatgi, Peter Ebert, Tomas Bastys, Sivarajan Karunanithi, Dilip Durai, Florian Schmidt, Lisa Handl. Thanks to colleagues in the department for their interesting group seminars and the feedback.

As this journey comes to an end, marking another beginning, I also fondly remember the following people from my alma mater in India: Sukratu Barve, Abhijat Vichare, Dilip Kanhere and Dr. Jayaraman. Leelavati Narlikar and Mihir Arjunwadkar have been my mentors throughout. All of you played an important role and have been corner stones that have helped nurture the researcher I am today. Thank you!

Finally, I would like to acknowledge the most important people in my life. I thank my parents for being my foremost source of inspiration, support and encouragement. My sister and brother-in-law can't be acknowledged enough for always being there to support me. I acknowledge my parents-in-law for their support and wishes. Special

kudos and thanks to my wife Prachi for always supporting me, especially, during the final phase of my doctoral studies. Thank you Prachi for always inspiring me to be a better person every day.

Contents

Listing of figures

Listing of tables

1

Introduction

Could you have imagined that the size of the wheat genome, i.e. the complete DNA sequence of common bread wheat is larger than that of the human genome (Zimin et al., 2017)? Could you have imagined that Axolotl, a salamander, has a genome $\sim 10\times$ the size of the human genome and that it is able to regenerate its limbs including the bones (Nowoshilow et al., 2018)? Many such astounding facts make the field of genome biology intriguing.

The DNA of eukaryotic organisms (all plants and animals are examples of eukaryotes) is stored inside the nucleus of their cells. Almost all cells of an organism have an identical copy of the complete DNA. This DNA when stretched out can be very long. For example, in humans, the completely stretched out DNA is ~ 2 m long, and the cell nucleus is just 6 μm wide. This storage is achieved by a compact, complex, hierarchical organization in three dimensional (3D) space. Scientists have been studying this structure and organization of the genome of organisms since many years (cite).

Notwithstanding the highly complex organization of the genome, each cell performs its functions based on the instructions encoded into the DNA sequence. The same genetic sequence in each cell can give rise to different functions for different cells. For instance, liver cells perform a different function than brain cells or the skin cells. A particular set of genes can be up-regulated (or down-regulated) in some cells while a completely different set of genes can be up-regulated in another. Instructions for such regulation of genes can be encoded in the DNA genome itself, or, in the epigenome

of a cell[1]. The epigenome comprises of modifications on top of the DNA, and is cell-type- or tissue-specific (Feinberg and Callinan, 2006). Such regulatory instructions could lie either in the vicinity of the gene or even far away on the (epi)genome. It is now known that communication between such distant (regulatory) regions on the genome is possible because of its 3D organization (Bickmore, 2013; Dekker, 2008; Lieberman-Aiden et al., 2009; Rao et al., 2014).

Understanding the mechanisms of 3D genome regulation is important. It is a long-standing interest of the scientific community to understand how living cells function. This knowledge of the fundamental aspects of biology can help in understanding what goes wrong in diseased conditions, or in other words, why certain cells loose their proper functioning capability. To this end, studies have shown that aberrations in the 3D architecture of the chromatin, a complex of DNA and proteins, can lead to disease conditions such as cancer (Zeitz et al., 2013). This, thus, has potential to impact the field of medicine. An improved understanding of these mechanisms can help in better comprehension of various diseases and designing superior treatments and therapies, for example, by identifying more potent drug targets.

Molecular biology techniques that help interrogate and study this 3D organization were invented over the course of the last decade (de Wit and de Laat, 2012). These experiments are performed on many different cell types and conditions to obtain high-resolution (more detailed) genome-wide information about their 3D architecture. Analyzing these data to gain biological insights requires enormous efforts towards developing computational methods.

The need of computational approaches for handling and analyzing such biological data sets has been long recognized. Gauthier et al. (2018) provide a brief overview of the history of bioinformatics. Post 2000, we have witnessed an exponential rise in the amount of data being generated and made available. This is mainly due to advances in technology used for performing molecular biology experiments. Some examples are the many consortium projects such as 'The Human Genome Project' (which culminated in early 2000s) and the human genome sequence (Lander et al., 2001; Venter et al., 2001), 'The 1000 Genomes Project' (1000 Genomes Project Consortium et al., 2010), 1000 Plant Genomes Project[2], and the Precision Medicine Initiative[3] etc. This lead to many academic and industrial labs worldwide performing these experiments on different organisms in diverse conditions with varied objectives. These objectives ranged from studying the basic biology in normal conditions and diseases to using the knowledge gained for identifying biomarkers, drug discoveries, and designing improved therapies. With these developments, the amount and kinds of data available

[1]See definition of 'epigenomics' here: https://www.nature.com/subjects/epigenomics
[2]https://sites.google.com/a/ualberta.ca/onekp/home, Retrieved Jan. 31, 2019
[3]https://ghr.nlm.nih.gov/primer/precisionmedicine/initiative, Retrieved Feb. 4, 2019

soared. It then became infeasible for biologists to scrutinize the data manually. Thus, approaches to automate pre-processing and analysis of such biological data—in part or full—became more and more popular. In addition, many biological phenomena were studied with the help of computer simulations. These involved computer applications of techniques from mathematics and statistics. All of this helped pace multiple facets of science much to the benefit of everyone.

The field of artificial intelligence (AI) also saw much improvement during this time frame. AI studied ways in which one could make automated inferences based on data as evidence. After many algorithmic developments in the initial years of AI, when more and more data started becoming available, performances of the same algorithms improved with larger data. This in turn motivated the generation of even more data and development of infrastructure worldwide for handling such vast data. As AI algorithms got sophisticated during the decades of 1990s and the 2000s, their performances on many benchmark tasks started improving, but their interpretability took a back seat or was slackened.

In many fields, machine learning (ML) algorithms were then used as *black boxes* simply keeping their hefty dividends in mind. For some, this is true even today. But for a field like biology or medicine, the ability to understand the pieces of evidence that the machine used to successfully perform a task such as prediction in large data sets is a lot more important. In domains like linguistics or natural language processing and computer vision, it is a lot easier or cheaper to know the ground truth than in the biological or medical sciences. Also, there is comparatively lesser risk if the ground truth itself is inaccurate. First, consider the task of identifying the subject and object in an English language sentence (domain:linguistics) or identifying cats in images (domain:computer vision). Compare these to the task of identifying the structure of a protein complex. In the latter case, ground truth is only known from experiments, and it can be expensive to obtain it. In general, the pieces of evidence uncovered by computational methods often require going back to the laboratory and designing new experiments or redesigning old ones. This involves monetary costs and/or it can be time-consuming. Second, consider the following example from the field of medicine. Doctors/radiologists routinely perform the task of classifying an image of a potential skin tumor as either benign or malignant. For this task, a machine (or an algorithm) that simply tells whether the tumor is benign or malignant, but nothing more, is usually not very useful (in some exceptional cases where even basic diagnosis is hard to come by, e.g., in underdeveloped or developing countries, this can still be useful). Any information on the piece of evidence from the image that the algorithm used to arrive at a particular answer is very important and holds tremendous value. It can not only help doctors in uncovering something that was not discovered yet—for instance, a complex, non-linear relationship between multiple entities—but can also

help in improved diagnosis as well as prognosis.

Such interpretable computational methods are the prime subject of this thesis.

1.1 Thesis Outline

The work presented in this thesis can be divided into two parts. In the first part, I demonstrate how interpretable ML methods can play an important role towards gaining *de novo* insights into the 3D genome organization. In terms of the biology, we answer the specific question of whether the genetic sequence is predictive of long-range interactions between genomic regions such as enhancers and promoters or others. In terms of methodology, we have developed and applied ML methods that are interpretable and take into account the underlying biology. Our work exemplifies the potential of ML methods with such characteristics in proposing newer hypotheses and refining our understanding of biology itself. In the second part, we focus on analysis pipelines for chromatin conformation data, especially when it is just one part of the complete, global picture involving even larger volumes of data from other experimental assays.

I have organized this thesis as follows. In Chapter 2, I begin by introducing the reader to the basics of genome biology. I then familiarize the reader with the state-of-the-art molecular techniques for interrogation of 3D genome organization. I also shed light on the different ways in which scientists are using data from these experiments to learn more about (a) the basic principles of genome organization, and (b) its role in proper functioning of cells. Next, I introduce the reader to machine learning, specifically focusing on kernel methods for supervised learning and string kernels.

Chapters 3, 4 and 5 describe the main contributions of this thesis. Chapter 3 presents a supervised learning model using only the genetic sequence information for prediction of locus-specific long-range interaction partners. Two novel visualization techniques that we developed to aid in this study are also described here.

Chapter 4 describes our approach for comparison of variable-length sequences in the supervised learning scenario. The scenarios in which the need to compare sequences of arbitrary length arise are discussed in this chapter. Further, we demonstrate the efficacy of the method on a synthetic data set and two real biological data sets. This chapter concludes with a discussion on the benefits of and the challenges in analyzing high-resolution genome-wide chromatin interaction data sets using the method described.

In Chapter 5, I present the pipeline we developed for end-to-end analysis of data from chromatin interaction experiments. We also discuss how the pipeline facilitates analyzing humongous amounts of HiChIP experimental data for two conditions in a Ewing sarcoma cell line to study the changes in the 3D architecture.

Finally, Chapter 6 concludes the thesis and discusses the future directions of this work.

1.1.1 Note on Publications

Parts of the work presented in this thesis have been published at various avenues. In particular,

- work described in Chapter 3 is published in the journal BMC Bioinformatics as:
  Sarvesh Nikumbh and Nico Pfeifer. **Genetic sequence-based prediction of long-range chromatin interactions suggests a potential role of short tandem repeat sequences in genome organization**. *BMC Bioinformatics*, 18(1):218, 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1624-x. URL http://dx.doi.org/10.1186/s12859-017-1624-x.

- the method described in Chapter 4 is published as a conference proceeding in WABI (Workshop on Algorithms in Bioinformatics) 2017:
  Sarvesh Nikumbh, Peter Ebert, and Nico Pfeifer. **All Fingers Are Not the Same: Handling Variable-Length Sequences in a Discriminative Setting Using Conformal Multi-Instance Kernels**. In Russell Schwartz and Knut Reinert, editors, *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*, volume 88 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/LIPIcs.WABI.2017.16.

- the pipeline described in Chapter 5 is available on Github (see next section), and the **manuscript is in preparation**.

At the beginning of every chapter, I report the contributions of all the authors involved in it.

1.1.2 Note on Software

The software arising out of this work is provided for use to the community at large. In Chapter 3, the pipeline for locus-specific analysis of long-range interaction partners is provided in an executable format and is made available at [http://bioinf.mpi-inf.mpg.de/publications/samarth/]. It is made available as free software for academic use, with no warranty or liability.

For *CoMIK* (Conformal Multi-Instance Kernels), its source code is provided via MPI-Github at [https://github.molgen.mpg.de/snikumbh/comik]. It is also provided

in an executable format for non-MATLAB users. *CoMIK* is licensed under the MIT License.

The source code of the pipeline for the analysis of the data from the HiChIP experiments is also provided at [https://github.molgen.mpg.de/snikumbh/pHDee]. `pHDee` pipeline software is provided under MIT License. All software is accompanied by a thorough set of instructions for the benefit of the end user.

The future of research is interdisciplinary, and will quickly take us into areas that today we cannot even foresee, ... This building gives us the space and the flexibility to go where the imagination of our faculty takes us.

Michael Tanner

2 Background

INTERDISCIPLINARY research is research at the intersection of two or more fields. Such research often builds upon ideas, tools and techniques from multiple fields for the progress of science. For example, in the natural sciences such as physics, chemistry and biology, scientists are interested in improving our comprehension of our surroundings. These are often sought with the help of tools and techniques developed in other fields such as mathematics, statistics, and computer science. There are many examples of such synergies leading to ground breaking discoveries. An amalgamation of mathematics, statistics and computer science techniques is now also called as the 'computational science'. However, as has been rightly noted, "If you want to do something successfully, understand the domain first, and the machine learning second..."[1], I begin by giving a basic introduction to molecular biology in the first part of this chapter. This is followed by a section where I introduce machine learning (ML), and popular ML approaches for the field of computational biology.

2.1 Essential Molecular Biology

This section presents a primer on concepts in molecular biology, and is intended to provide the reader with a basis to better understand the work presented in this thesis. To this end, I also present a comprehensive but non-exhaustive introduction to the

[1]Prof. Neil Lawrence summarizing the Talking Machines Podcast episode, Machine Learning in the Field and Bayesian Baked Goods, t=57:50, Retrieved Jan. 23, 2019. Included with permission.

molecular techniques developed for studying the 3D chromatin interaction profiles of organisms. I envisage this to aid in making the journey of the reader through this thesis as smooth as possible.

2.1.1 Genome: The Blueprint of Life

The *cell* is considered the most basic unit of life on earth. A cell is a watery solution of molecules surrounded by a lipid membrane. An individual cell is responsible for functions such as replication, synthesis of proteins, response to external environmental stimuli etc. In order to perform these functions any cell uses up nutrients and can make other newer molecules. Instructions and ingredients for all these responsibilities are typically present in the cell itself. Any living organism has one or more cells. For example, bacteria are unicellular while plants and animals including humans are multicellular.

More specifically, almost all living cells are similar in the following aspects:

- Storage of the hereditary information in a linear chemical code, i.e. the deoxyribonucleic acid (DNA);
- Transcription of portions of this hereditary information to the same intermediary form, i.e. the ribonucleic acid (RNA); and
- Translation of RNA into protein the same way.

That information can be transferred from nucleic acid to nucleic acid or from nucleic acid to protein is termed as the '*central dogma*' of molecular biology. Thus DNA, RNA and proteins are three very critical macromolecules in any cell. In the following I begin with a brief description of DNA, RNA and proteins. Subsequently, I focus the discussion on DNA as it is at the heart of the subject of this thesis.

DNA forms the piece of heritable information stored in a cell. It is essentially the blueprint that provides all instructions necessary for a cell to perform its functions. Both DNA and RNA are macromolecules mainly formed by a chain of nucleotides (nt). A nucleotide is a molecule made up of a nitrogenous *base*, ribose or deoxyribose sugar, and a phosphate group. Each nucleotide is given a name depending on its nitrogenous *base*. For a DNA molecule, the bases are adenine (`A`), guanine (`G`), cytosine (`C`) and thymine (`T`). An RNA molecule has `A`, `G`, `C` and uracil (`U`) instead of `T`. DNA has a double helix structure formed by two chains of nucleotides that cling together.[2] The nucleotide chains are also called *strands*. The two strands of DNA come together such that the base `A` on one strand pairs with the base `T` on the other, and similarly, `G` pairs with `C`. The phosphate of the DNA molecules form the backbone of this double-helix

[2]This double-helical structure of DNA was proposed and discovered in 1953 by Francis Crick and James Watson based on Rosalind Franklin's x-ray crystallography experiment that showed the peculiar diffraction pattern of DNA.

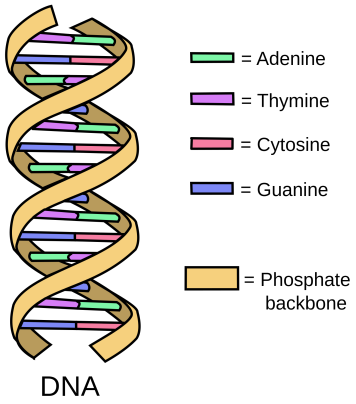


Figure 2.1.1: A schematic showing the DNA structure. Source: Wikimedia Commons, License: Public Domain.

structure. Owing to the base-pairing `A-T` and `G-C`, nucleotides are, interchangeably, also called as '*basepairs*' (bp). One complete turn of the helix encompasses 10 bp. Figure 2.1.1 shows the nucleotides and phosphate group. The phosphate group being common among the nucleotides, a sequence of nucleotides forming the chain can be simply described by the different nitrogenous bases. Thus, a DNA can be described textually with alphabet of four characters `A`, `C`, `G` or `T`. The sequence of nucleotides on one strand of DNA is complementary to those on its other strand. DNA is read from its 5′ end to the 3′ end, i.e. from direction 'upstream' to direction 'downstream'. The sequence of nucleotides on each strand encodes information that describes an organism. The complete DNA in the cell is called the '*genome*' of an organism.

Certain portions of DNA are transcribed into what are known as messenger RNAs or mRNAs. These portions of the DNA are called genes (more on genes below). The mRNAs are then translated into proteins. Twenty different kinds of amino acids are used for protein synthesis. Any amino acid has two chemical groups—the amino group [N] and the carboxyl group [C]—and a third, called the *side chain*. The side chains of the twenty amino acids show different chemical properties. Proteins are the macromolecules responsible for the various tasks performed by cells; they keep the cells up and running. Some important tasks fulfilled are metabolism, transcription and protein synthesis, transportation, and intra- and inter-cellular communication.

2.1.2 Packaging of The Eukaryotic Genome

On the basis of the structure of their cells, organisms can be classified into prokaryotes and eukaryotes. Cells of prokaryotes store the DNA in no distinct compartment, while in eukaryotes the DNA is stored inside a specific intracellular compartment with a surrounding membrane. This compartment is called the nucleus of the cell. Bacteria

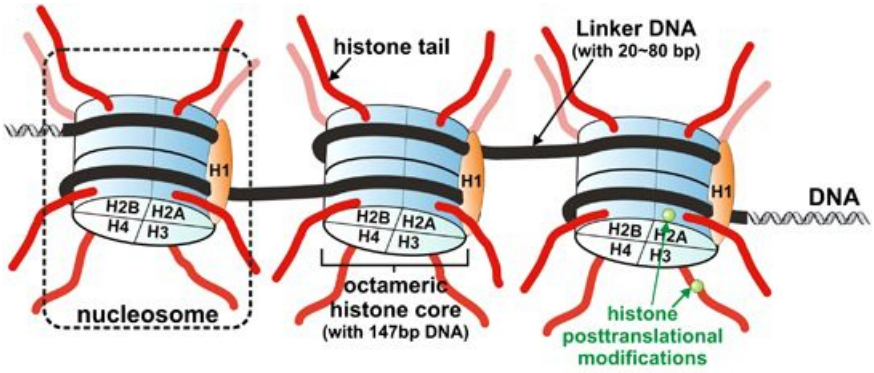


Figure 2.1.2: A schematic showing nucleosomes with histone proteins and DNA coiled around them. Reprinted with modifications by permission from Springer Nature: Nature (License #4487621281224), Füllgrabe et al. (2010), Copyright 2012.

and archaea are examples of prokaryotes. Plants and animals, including humans, are examples of eukaryotes.

For any organism, its complete DNA can be very long when stretched out. For instance, in humans, the DNA (approximately 3×10^9 nucleotides) is about 2 m long, and, yet, it is contained within the cell nucleus which is about 6 μm wide. This is possible due to multiple levels of compaction and organization applied to the DNA. At the most basic level, the double-helical DNA [2 nm] is wound around *histones* which are disc-shaped proteins (see a schematic shown in Figure 2.1.2). Each histone [11 nm] has 1.65 turns, or 147 bp, of DNA tightly wound around it. Eight such histone molecules make a *nucleosome*. These nucleosomes are packed on top of each other to further condense the DNA. The complex of DNA and proteins (histones and non-histones, that bind to the DNA) is called *chromatin* [30 nm]. This 30 nm chromatin fiber forms loops [300 nm]; these are further folded and compressed [700 nm] to form a section of the chromatid of a chromosome. The chromosome itself is 1400 nm wide. Thus, the DNA is packaged in the form of chromosomes. For example, the human genome is divided into 46 chromosomes—22 pairs and two sex chromosomes. A gene is one such peculiar segment of DNA which serves as an instruction to produce a certain protein, thereby making the cell able to fulfill a function. But not all of DNA is genes. Much of the DNA encodes for regulatory instructions. Therefore, one can say that the genome is not just a cookbook filled with recipes, but also includes information on when which recipe is to be used, and where a particular one can be found.

We briefly discuss the gene regulatory mechanisms next.

2.1.3 Gene Regulation

Transcription, Splicing and Translation

Protein synthesis from genes begins by transcription of DNA to mRNA, which then undergoes translation. A gene is transcribed when a host of proteins called transcription factors come together. Transcription factors (TFs) bind to specific DNA sequence motifs called transcription factor binding sites (TFBSs). These sites are usually 5-15 bp long (Bulyk, 2003). The DNA sequence lying upstream to a gene is called the promoter sequence or, simply, promoter. The promoter holds many sequence signals (motifs) that help recruit a general set of TFs. These together then assemble a special protein called RNA polymerase II at a specific position where transcription begins. This position is called the transcription start site (TSS). Upon assembly, the RNA polymerase II moves along the DNA synthesizing copies of pre-mRNAs, the primary RNA transcripts. In addition to acting as activators, TFs can also repress/inhibit gene expression (Latchman, 1997).

The pre-mRNA copies produced from a gene are then spliced to remove selected portions called *introns* and keeping those called *exons* to give the final product. The final product is the (mature) mRNA which is synthesized into proteins by ribosomes. This process is called translation; it takes place outside the cell nucleus.

Role of Chromatin

Gene regulation can also happen at the level of and due to chromatin packing in the cell nucleus (cf. Section 2.1.2). In packaged chromatin, whenever the cell requires access to a certain portion of the DNA, the packaging is temporarily decondensed by various enzymes and proteins. This makes specific regions of the DNA accessible. Only then the basal transcriptional machinery and other TFs are able to do their job. Accessibility of DNA regions can be controlled or facilitated by modifications to DNA itself or the histones. Portions of DNA that are tightly wound around histones remain inaccessible (to be read by proteins/enzymes) while some regions become accessible as the coiled DNA loosens up due to chemical modifications on them. These are termed as epigenetic modifications. Examples are DNA methylation—addition of methyl group directly on the DNA, and histone modifications—chemical modifications attaching to histone tails.

Such gene regulation mechanisms render cells the ability to perform different functions although each cell has an identical copy of the genome. Depending upon the type of the cell (based on the tissue) or cell line, a characteristically different set of genes can be switched on or off.

Finally, we note that any genome has many genes. For example, the human genome has roughly 20,000 genes. While the exact definition of a gene is still debated, it has evolved over time. Salzberg (2018) defines a gene as "any interval along the

chromosomal DNA that is transcribed into a functional RNA molecule or that is transcribed into RNA and then translated into a functional protein". The above definition accounts for genes whose final product is a noncoding RNA molecule, one that does not code for a protein. Finding the number of genes in humans is still an open question (Salzberg, 2018).

2.1.4 The Genome is Now Better Understood in 3D

As discussed, genes are regulated by TFs binding to the promoter sequence. In the early 1980s, studies reported regulation of genes by novel regulatory elements located far away on the linear genome. These elements are usually located in the non-coding portions of the DNA. For example, enhancers are identified as regulatory elements that enhance transcription of a gene. Thus, they are similar to promoters in function but are a lot more distant than the promoter is to the gene (Serfling et al., 1985). These enhancers are, therefore, *long-range* activators of gene transcription.

Scientists first proposed that such regulatory function of an enhancer can be fulfilled by being in spatial proximity or physically interacting with the concerned promoter. Early on many FISH[3]-based studies reported co-localizations of functionally-related elements in the nuclear space. FISH has been used to report examples of gene-rich loci on chromosome 11 localizing outside of its territory, and the role of transcription in it. This suggested formation of open chromatin structure for regions with high gene density (Mahy et al., 2002). Williamson et al. (2012) report the topological co-localization involving the Hoxd13 gene and the global control region located 180 kb away. Consequently, such studies led to the proposition of a 3D conformation of the genome inside the nucleus of the cell. Microscopy-based studies have now combined forces with recently developed molecular biology experiments towards improving our understanding of the mechanisms of regulation of the genome in 3D.

I first introduce the chromosome conformation capture (3C) technology for interrogation of the 3D chromatin interaction landscape in cells. Afterwards, some variants of this technology that were developed to overcome shortcomings or issues with 3C are discussed in brief. In the process, I also shed light on the output of these experiments and the important caveats towards interpretations and conclusions from these experimental data.

Chromosome Conformation Capture-Based Techniques

Chromosome Conformation Capture (3C) is the technology designed to probe interactions between different genomic loci of interest (Dekker et al., 2002). The first few

[3]Fluorescence in situ hybridization (FISH) is a microscopy-based assay used for studying chromosome structure. For more, refer (Volpi and Bridger, 2008)

steps of the protocol for 3C, the maiden one, are common in principle to its derivatives 4C, 5C and Hi-C (these are described subsequent to 3C). The general principle of the 3C techniques follows these steps:

1. **Fixation of the chromatin:** In order to probe the organization of the chromatin later, first its current state is fixated (recorded) using fixating agents such as formaldehyde. This makes chromatin regions that are spatially proximal (thus, also including those that are in direct contact) to cling (cross-link) to each other.

2. **Digestion of the chromatin:** The fixated chromatin is then digested into many small pieces. This is done using either restriction enzymes (REs) or by a process called sonication. A RE, such as HindIII, MboI or DpnII, cuts the DNA at all positions recognized by a specific substring of a certain length, say 6 bp substring '`AAGCTT`' for HindIII or 4 bp substring '`GATC`' for MboI. These positions are referred to as the cut sites of the restriction enzyme. It is easy to observe that a shorter cut site will occur more frequently throughout the genome sequence than the longer one. Thus, a 4 bp cutter such as MboI or DpnII results in shorter restriction fragments (RFs) than a 6 bp cutter such as HindIII (Belaghzal et al., 2017). In comparison to the enzyme-based approach, the process of sonication fragments the DNA at random positions and is unaffected by accessibility of the DNA in the chromatin. Sonication is used in ChIA-PET (Li et al., 2010), which I briefly describe in Section 2.1.4.

3. **Ligation of the digested chromatin:** The cross-linked chromatin fragments are ligated to form DNA molecules which are hybrids of the cross-linked segments. The output of this step is referred to as the contact library.

4. **Reading-out and quantifying the ligation junctions:** The re-ligated DNA is then sheared to fragment the hybrid DNA molecules further. The pieces that have the junctions on them are read out. Each such piece gives information on the pairs of loci that are spatially proximal in the organization. When N such pieces of DNA with junctions are read, we know of N interaction events involving various regions on the genome. This particular step varies depending on the aim and scope of the technique.

From the point of view of the genome itself, what one effectively captures is the frequency of contact between different genomic loci. Thus, one is able to get a picture of the 3D conformation of the chromosomes inside the cell nucleus.

It is important to note the following in the context of chromatin interaction experiments. Here, an 'interaction' or a 'contact' between any two genomic loci could

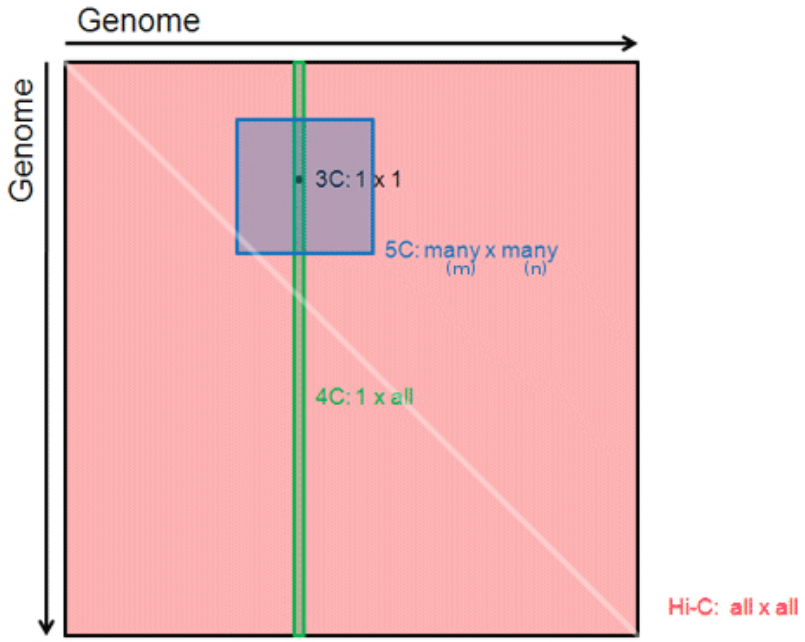


Figure 2.1.3: A schematic depicting the scope of different chromosome conformation capture-based methods, namely 3C, 4C, 5C and Hi-C. Source: Dekker Lab webpage http://my5c.umassmed.edu/about/about.php?tab=welcome&category=cmethods, with modifications. Accessed: December 2018. Included with permission (personal communication).

essentially mean either of two possibilities. First, the two regions are in direct physical contact with each other to perform some cellular function. Any contact is typically also accompanied by a set of other proteins, e.g., transcription factors, (co-)bound to these genomic regions. Second, they are only spatially proximal and are possibly communicating with each other indirectly to fulfill some function, or are close due to other physical constraints. Communication between any two genomic loci may be serving some functional purpose. For instance, a locus that acts as an enhancer or a repressor could be interacting with a promoter region to either enhance or silence the particular gene. In other cases, there is possibility of some physical constraints acting upon the linear DNA polymer. For example, two loci adjacent or close on the linear DNA sequence are bound to be in close spatial proximity.

The schematic shown in Figure 2.1.3 depicts the scope of the various chromosome conformation capture (3C)-based techniques for probing the long-range interactions between different regions of the genome. The difference between these various 3C-based methods—3C, 4C, 5C—lies in the way in which the individual steps are handled to achieve the respective scope with as much improved resolution as possible. I defer the discussion on resolution of the contact matrix to Subsection "Pre-processing of Chromatin Interaction Data".

3C: The maiden technique of the family, which also gives the family its name, 3C (Dekker et al., 2002) can interrogate interactions between specific genomic locus-pairs of interest – hence, this is 1×1. After the contact library generation is completed, the final step of counting the interaction events is performed using semi-quantitative polymerase chain reaction (PCR) amplification[4]. During this step, primers designed for identifying the restriction fragments allow counting the interaction events involving pairs of restriction fragments of interest.

4C: 3C was improved in two different ways, both in 2006 by two separate groups, to probe genome-wide interactions involving a 'bait' locus. In the 3C-on-chip (4C) assay (Simonis et al., 2006), the primary fragmentation step is performed using HindIII. And, after the subsequent ligation step, an additional iteration of fragmentation takes place. This is done using DpnII, a more frequent cutter. The re-ligation step makes the DNA molecules circular in nature. The interaction events involving the bait locus are then counted using inverse PCR with primers specific to this locus (Simonis et al., 2006). The PCR products are finally characterized using dedicated microarrays. In Circular 3C (4C) (Zhao et al., 2006), large concentrations of ligase and incubations longer than a week's time generate circular DNA molecules of protein-DNA complexes. This is followed by nested PCR to enable identification of global interaction partners of the target sequence (Zhao et al., 2006).

Consequently, 4C is a $1 \times \textit{all}$ strategy. This 'bait' locus is also referred to as the 'viewpoint'. 4C has been the preferred technique of choice to study genome-wide interactions of promoters, enhancers and various locus control regions as bait loci (de Wit and de Laat, 2012). .

5C: In Chromosome Conformation Capture Carbon Copy (5C) (Dostie et al., 2006), the aim is to probe interactions between many genomic regions at once. The 5C protocol begins by generating the 3C template. The next step is to multiplex the generated 3C template using 5C oligonucleotides. Then, a collection of forward and reverse primers are used to ligate across the ligation junctions. This identifies the corresponding restriction fragments. Many forward and reverse primers enable analyzing interactions between many restriction fragments. This 5C library is analyzed using microarrays or high-throughput sequencing technology. Thus, this achieves a $\textit{many} \times \textit{many}$ scope (Dostie et al., 2006).

It is instructive to note that the sets of loci studied in a 5C experiment can include regions from any where on the genome. They need not be contiguous and the two sets can have different cardinalities.

[4]PCR: A technique developed for amplification of specific nucleic acid sequences by Kary Mullis, an American biochemist. Mullis received the Nobel Prize in Chemistry 1993 for this invention. Read more about PCR here.

Hi-C: The high-throughput, genome-wide version of 3C is given the name Hi-C. Hi-C maps genome-wide chromatin contacts in an unbiased manner. Before ligation of the digested chromatin, the ends of the cross-linked segments are marked with biotin. Then, the DNA molecules with biotins are pulled down using streptavidin beads. This is followed by shearing and paired-end sequencing (Lieberman-Aiden et al., 2009).

Pre-processing of Chromatin Interaction Data

The raw output from these experiments undergoes some standard pre-processing steps. These are briefly described next. The reader is referred to Ay and Noble (2015) and Belaghzal et al. (2017) for further reading.

1. **Reference alignment of the raw sequencing reads.** The chromatin interaction experiments report chimeric fragments. These are fragments of DNA with ligation junctions on them. On either side of this junction are DNA segments from non-contiguous genomic locations. This is the ideal scenario. Usually, the chimeric fragments are readout using paired-end sequencing. In paired-end sequencing, one sequences both ends of a fragment. Thus, one obtains information about DNA segments forming the chimera. The reads are then aligned to the reference genome. This identifies the genomic locations to which the individual reads correspond.

2. **Assignment of reads to restriction fragments.** Upon mapping, one assigns the individual reads to the RFs. Get all the RFs of the genome using the RE cut-site. Use the distance of the cut-sites from the read locations to assign individual reads to RFs.

3. **Filtration of noise.** Some factors need to be taken into account at the end of the above two stages for filtering of invalid reads. For example, at the individual read level, uniqueness and mapping quality of reads is important. At the read-pair level, one should discard or filter out uninformative reads. Examples of the latter are reads corresponding to dangling ends or self-circles. Dangling ends could be results of unligated fragments, while self-circles are self-ligated fragments.

   The final output from a chromatin interaction experiment is a list of contacts. It is a set of valid interactions between various genomic loci. For 5C and Hi-C, this information is visualized as a two-dimensional matrix.

4. **Binning to build contact maps.** Upon filtering, the final list of contacts can be visualized as 2D matrices. A pair of genomic loci identify each cell in this

matrix. The corresponding cell entry itself is the contact frequency between these loci. Thus, the complete matrix records contact frequencies between the various genomic loci studied in the experiment. The size of these genomic bins determines the resolution of the contact matrix.

With regards to Hi-C, there are two ways for building the contact matrix—one is with RF-based resolution, the other is with resolution in terms of basepairs. With RF-based resolution, each genomic bin along the matrix can represent one or many RFs. If many RFs are combined, the number of RFs combined is uniform throughout the matrix, and, the combined RFs should form a contiguous genomic window. Each cell of the matrix denotes the interaction frequency between the genomic regions characterized by the corresponding RFs. Although the number of RFs per genomic bin along the matrix is fixed (one or many), these regions can be of variable length in terms of basepairs, since the individual RFs are of variable length. Contrastingly, when the resolution is in basepairs, each genomic bin along the matrix is of a fixed number of basepairs (non-overlapping genomic regions). In this case, the following procedure is followed: (a) Bin the complete genome into fixed-size genomic regions; (b) For any given interaction involving a pair of RFs, we note all those genomic bins that have an overlap with the RFs; and, (c) Increment each cell entry corresponding to these genomic bins by 1. For Hi-C, the contact matrix is symmetric in nature.

A 5C contact matrix usually has a RF-based resolution. As described, 5C experiments interrogate interactions between two sets of genomic loci. For example, Sanyal et al. (2012) map the interactions between a set of promoters and a set of enhancers. In this 5C matrix, the promoters are along the rows and the enhancers along the columns. Hence, a 5C matrix is non-symmetric.

Figure 2.1.4 shows an example heatmap visualization of a contact matrix. In the heatmap, the darker the color, higher is the contact frequency. The left panel of the figure shows a genome-wide contact matrix. The genomic bins are arranged in the chromosomal order. In the right panel of the figure, we zoom into the contact matrix, to a specific location on chromosome 4.

The resolution of a contact matrix impacts further downstream analysis steps. It also affects the biological interpretations as I illustrate with examples later. I now discuss the final, important pre-processing step applied to the contact matrix.

5. **Normalizing a contact matrix.** Various kinds of biases affect chromatin interaction experiments. For instance, the GC content of the genomic fragment ends, mappability of reads, and density of restriction sites. One should correct

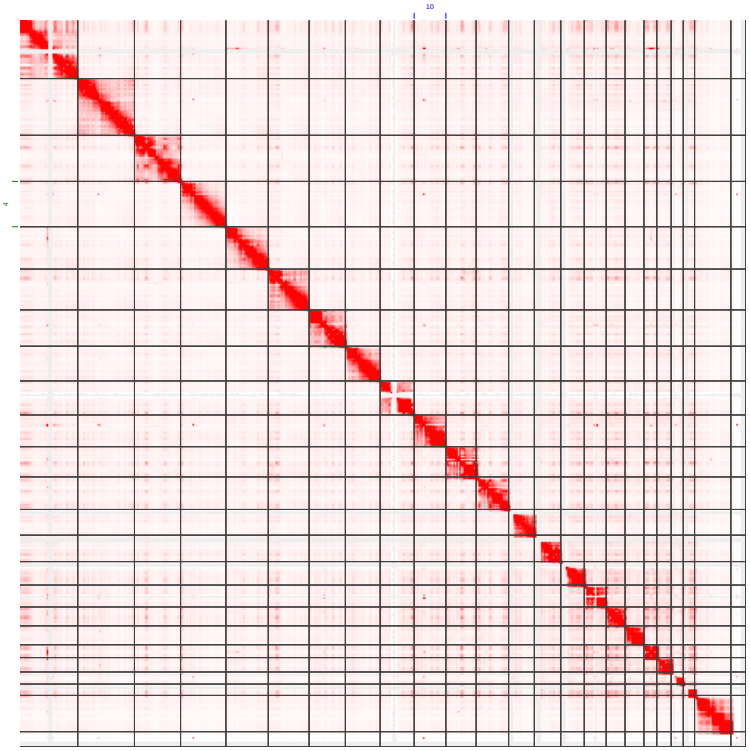


Figure 2.1.4: Example of a genome-wide contact map from a Hi-C experiment in GM12878 (Rao et al., 2014).

for these biases before analysis and interpretation of these data.

There are approaches that do so in an explicit or implicit fashion. Among those that explicitly correct for these biases are: (a) Yaffe and Tanay (2011)'s probabilistic model for jointly eliminating the biases; (b) HiCNorm, that uses a Poisson or negative binomial regression model based background model (Hu et al., 2012) to model the contact frequencies between loci. Yaffe and Tanay's approach models three known factors explicitly and has many parameters which are estimated using maximum likelihood. This makes it computationally very expensive; even more so with an increasing sequencing depth (Yaffe and Tanay, 2011). In comparison, with HiCNorm, the authors set the mappability feature as a Poisson offset and estimate the effect of GC content and fragment length using a generalized linear regression model. HiCNorm is comparatively faster (Hu et al., 2012). Couple of other approaches that improved on the above two are reviewed in Ay and Noble (2015).

Approaches that correct for the biases in an implicit fashion are based on an important assumption. This assumption is that all regions on the genome should have equal visibility in terms of the technical artifacts of the experiment as well the biological features. The DNA sequencing bias towards different genomic regions is an example of a technical artifact. Examples of biological features

affecting the experiment include GC content of the fragments and fragment length. In 2012, Imakaev et al. adopted an iterative procedure for correction of genome-wide Hi-C contact matrices. The underlying idea here being that the genome-wide contact map can be factorized into the biases and the relative contact map between the genomic loci. Mathematically, this can be represented as

$$\epsilon_{ij} = B_i B_j T_{ij}, \tag{2.1}$$

where B_x is a bias vector denoting the bias associated with any genomic locus x, T is the normalized, relative contact map with every row or column summing up to 1 (or a constant), and ϵ is the expected contact map assuming the biases (the one obtained from the experiments). This procedure is commonly known as matrix balancing, and is based on Sinkhorn and Knopp (1967)'s work on convergence of non-negative square matrices to doubly stochastic matrices (Sinkhorn-Knopp algorithm).

Similarly, Rao et al. (2014) used the Knight and Ruiz (2012)'s algorithm for matrix balancing that is shown to converge faster than the Sinkhorn-Knopp algorithm. Rao et al. used this procedure for normalizing extremely deeply sequenced Hi-C contact maps with up to a billion reads. Matrix balancing procedures are more common these days with genome-wide, deeply sequenced Hi-C data sets becoming increasingly common.

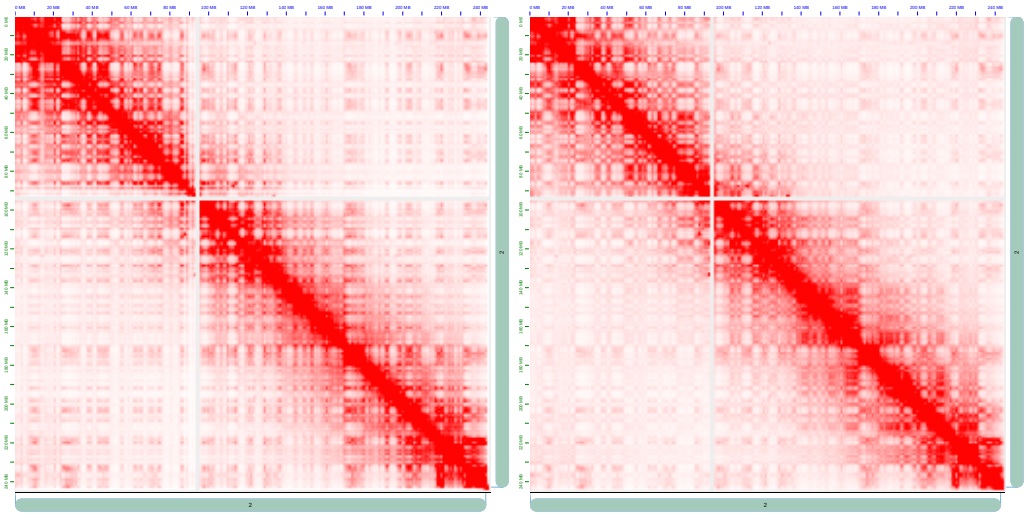


Figure 2.1.5: Exemplar raw and normalized chr2 contact maps from Hi-C experiment in GM12878 (Rao et al., 2014). Resolution used 1M bp.

The normalization procedure leads to smoother contact maps as illustrated in the Figure 2.1.5.

It is often important and useful to visualize the normalized contact matrices as heatmaps and inspect them to identify interesting patterns of long-range interactions. These interaction maps show many key architectural features. I describe three such features next.

A/B compartments: The contact matrix is typically analyzed using Principal Component Analysis. This analysis has revealed an interesting interaction pattern at the megabase scale characterized by the leading eigenvector (Imakaev et al., 2012; Lieberman-Aiden et al., 2009). The genome appears to be divided into two compartments A and B where regions in one compartment show a preference for interactions with other regions in the same compartment but not with those falling in another. These compartments alternate along chromosomes and demarcate regions of open (A) and closed (B) chromatin (Lieberman-Aiden et al., 2009). They correlate with features such as DNA accessibility, transcriptional activity, gene density, GC content and chromatin marks, thus associating compartment A with euchromatic, transcriptionally active regions, and compartment B with closed chromatin (Dekker et al., 2013). Imakaev et al. (2012) have shown that the signal from the leading eigenvector is more continuous in nature than strictly two-phased.

Topologically Associating Domains (TADs): TADs are self-interacting regions seen as triangles in a contact map (Dixon et al., 2012; Nora et al., 2012). When moving along the diagonal of a contact matrix, the effective number of long-range interactions of individual bins show a sudden, drastic change in direction. Consider a collection of genomic bins along the diagonal of the interaction matrix. They show a high number of long-range interactions with loci upstream (downstream) to it. Then, a bin immediately next to this collection shows a sudden shift: it has a high number of interactions with loci downstream (upstream) to it, instead. This observed phenomenon is termed as inversion in the directionality of interactions. It is measured by the 'Directionality Index' (DI) statistic (Dixon et al., 2012). A collection of genomic regions that tend to have more interactions between themselves could result from the topological configuration illustrated in Figure 2.1.6. These are thus called topologically associating domains and are abbreviated as TADs. As can be seen in Figure 2.1.6, TADs are visible as pyramidal structures in the upper or lower triangle of a symmetric contact matrix. Studies have proposed that TADs are hierarchical in nature (Cubeñas-Potts and Corces, 2015; Fraser et al., 2015; Phillips-Cremins et al., 2013; Rao et al., 2014) These lower-scale, domains within domains are called metaTADs (Fraser et al., 2015), subTADs (Cubeñas-Potts and Corces, 2015; Phillips-Cremins et al., 2013) or contact domains formed by DNA loops (Rao et al., 2014).

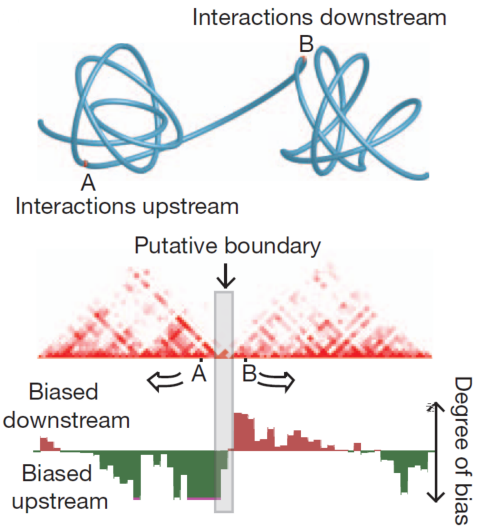


Figure 2.1.6: Illustration of TADs and change in directionality of interactions at the border. Reprinted by permission from Springer Nature: Nature (License #4487630141650), Dixon et al. (2012), Copyright 2012.

TADs are found to be stable across cell-types (Dekker and Heard, 2015), and also conserved across species (Sexton and Cavalli, 2015). Lupiáñez et al. (2015) report that disruption of TADs can lead to changes in the regulatory code. For example, disruption of TADs can lead to gaining newer interactions between enhancers and promoters that were earlier in different TADs. The resultant misregulation caused limb malformations (Lupiáñez et al., 2015). Borders of TADs/loops are often marked by sites bound by CTCF in a convergent orientation and other structural proteins such as the cohesin/SMC (Cubeñas-Potts and Corces, 2015; Rao et al., 2014). Computational studies have analyzed the borders of these domains and have found that short tandem repeats (STRs) are enriched at these borders (Mourad and Cuvier, 2016).

Many tools that attempt to identify TADs by identifying their boundaries exist. Some popular examples include an hidden Markov model-based approach using DI (Dixon et al., 2012), a computational tool called ARMATUS using dynamic programming (Filippova et al., 2014), and Arrowhead (Rao et al., 2014).

Significant interactions: Identifying statistically significant interactions is an important aspect of 3C experiments. Since DNA is a linear polymer, genomic loci on the same chromosome are expected to interact as a function of the genomic distance between them. Shorter the 1D distance between the loci, more frequent their interactions. This can introduce many random looping interactions measured in the chromatin interaction experiment. Thus, one has to consider this factor to identify loci that interact more frequently than they would otherwise, by chance. There is a good possibility that such statistically significant interactions are also functionally meaningful. Examples include long-range interactions between enhancers and

genes/promoter regions. The interaction between the erythroid-specific β-globin gene and its distal enhancer lying 50K bp away, the locus control region, is a well-studied enhancer-gene pair example (Cope et al., 2010).

Different studies estimate this distance-dependent expected interaction profile in different ways. Once the expected contact counts (E) between loci are available, then the observed contact counts (O) between them are normalized w.r.t. E. The O/E ratio is computed and a threshold is applied to determine significant interactions. For example, Lieberman-Aiden et al. (2009) used the average interaction count between genomic regions lying at similar distances to compute E. Sanyal et al. (2012) used a strategy similar to this for identifying significant interactions from their 5C data.

Another approach to calling significant interactions is using a non-parametric approach. For instance, a popular tool, Fit-Hi-C (Ay et al., 2014), iteratively fits smoothing splines to the contact profile of locus pairs arranged in ascending order, i.e. from pairs separated by the least genomic distance to the largest. The first spline fit allows identifying likely non-random interactions (outliers). These outliers are filtered before fitting a second spline. This serves as a refined null model which is then used to assign p-values and q-values to all interactions. Thus, one identifies significant interactions at different false discovery rates (FDRs)[5]. Fit-Hi-C can also incorporate biases per locus computed by any normalization procedure (Ay et al., 2014). There is a high chance that the identified statistically significant interactions also play a functional role, but it may not be the case with all such interactions.

Single cell Hi-C and *in situ* Hi-C: 3C and its derived experimental techniques are performed over a population of cells, typically in the order of millions. Consequently, these experiments characterize an average conformation of the chromosomal structures in all the cells. Nagano et al. (2013) developed single cell Hi-C (scHi-C) for detecting whole genome-wide interactions in a single cell. Here, the contact library is generated by following the same steps as in Lieberman-Aiden et al. (2009)'s dilution Hi-C protocol described above, but, it is performed inside the cell nucleus itself. An individual nucleus is then selected to perform the remaining steps in the protocol, namely, reverse cross-links and readout the biotinylated junctions, further digestion using Alu I, attach adapters for their PCR amplification and lastly, paired-end sequenced (Nagano et al., 2013). The contact maps from scHi-C are sparse (see panel b, Figure 2.1.7 for an example).The authors pooled data for 60 cells and the corresponding contact map showed similar architectural features as Hi-C maps.

In situ Hi-C (Rao et al., 2014) is designed as an improvement over the dilution Hi-C protocol (Lieberman-Aiden et al., 2009). Although inspired from an old nuclear

[5]FDR is the ratio of false positives to the true positives. Thus, controlling the FDR means aiming for a low proportion of false positives results.

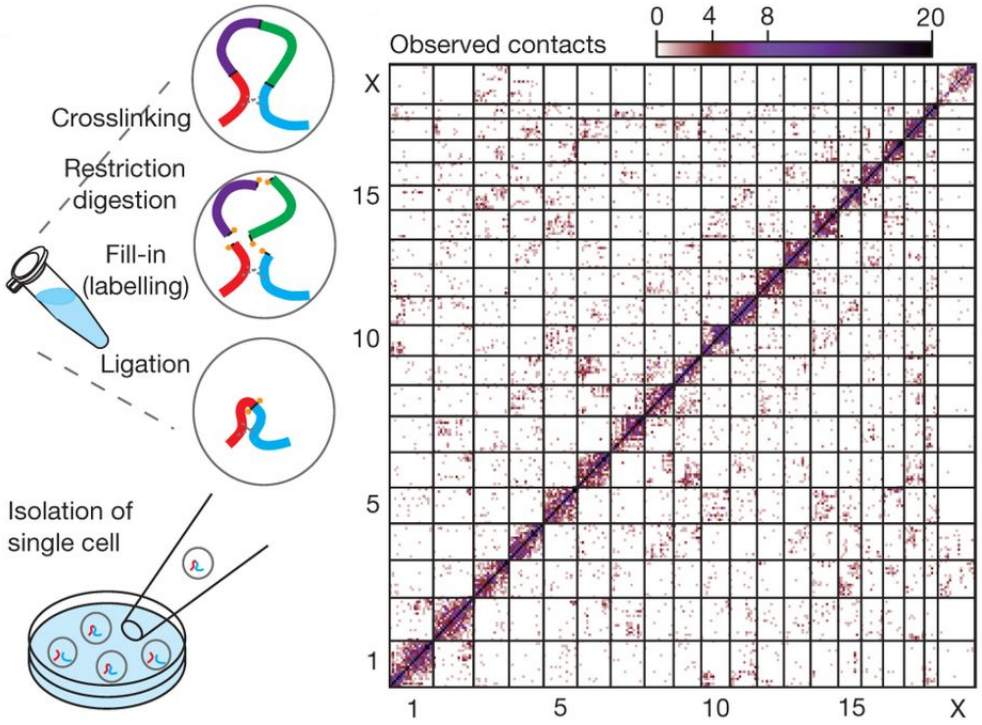


Figure 2.1.7: A contact map generated from a scHi-C experiment is shown. Reprinted by permission from Springer Nature: Nature (License #4483130037612), Nagano et al. (2013), Copyright 2013.

ligation assay (Cullen et al., 1993), *in situ* Hi-C is similar to the scHi-C protocol. In that, it performs the contact generation step inside an intact nucleus. This reduces spurious contact frequency involving mitochondrial DNA and nuclear DNA, which is the case with dilution Hi-C. *In situ* Hi-C uses a 4-cutter restriction enzyme compared to the 6-cutter used in dilution Hi-C. Overall, in situ Hi-C can achieve higher resolution than dilution Hi-C, provided the libraries are sequenced deeply enough (Rao et al., 2014).

Most of the current set of chromatin interaction experiments provide information that is still coarse. Since majority of the (known) regulatory regions are not larger than a few hundred basepairs, experiments that can yield precise information at high-resolutions are preferred. Although, the *in situ* Hi-C assay can provide genome-wide contact maps at kilobase-resolution, it requires extremely deep sequencing. For instance, the 1K bp-contact map for cell line GM12878 required about 1B sequencing reads (Rao et al., 2014). This is still quite expensive.

Factor-Mediated Chromatin Interaction Detection

Hi-C or the other 3C assays explore the long-range interactions landscape in an unbiased manner. Therefore, using these techniques for identifying and studying interactions involving some selected subset of regions, e.g., regulatory regions such as promoters, enhancers etc., requires very deep sequencing. Factor-mediated techniques

help circumvent this requirement. They specifically enrich for interactions mediated by factors identifying such subsets of regions. DNA-binding proteins or other architectural proteins and capture-oligos are examples of such factors.

ChIA-PET: ChIA-PET was the first technique that enabled genome-wide interrogation of chromatin contacts mediated by proteins (Fullwood et al., 2009). It combines 3C with ChIP-seq, the technique for identifying genome-wide binding information (1D information) of proteins such as transcription factors. Thus, ChIA-PET offers a protein-centric view of the complex interaction landscape as against other 3C-based approaches. But a key shortcoming of ChIA-PET is that, it requires a high amount of starting material—in the order of millions of cells—and provides smaller proportion of informative reads at a given sequencing depth.

HiChIP: HiChIP is developed as an improvement over ChIA-PET (Mumbach et al., 2016). The main steps in the HiChIP protocol are as follows. HiChIP adapts the *in situ* Hi-C contact generation procedure as its first step (Rao et al., 2014). Then, those contacts associated with specific proteins of interest are isolated (with ChIP). Finally, as in Hi-C, the biotinylated protein-associated contacts are pulled down to prepare contact libraries. The HiChIP protocol is outlined in Figure 2.1.8. The total time required for performing HiChIP is about 2 days. Another important

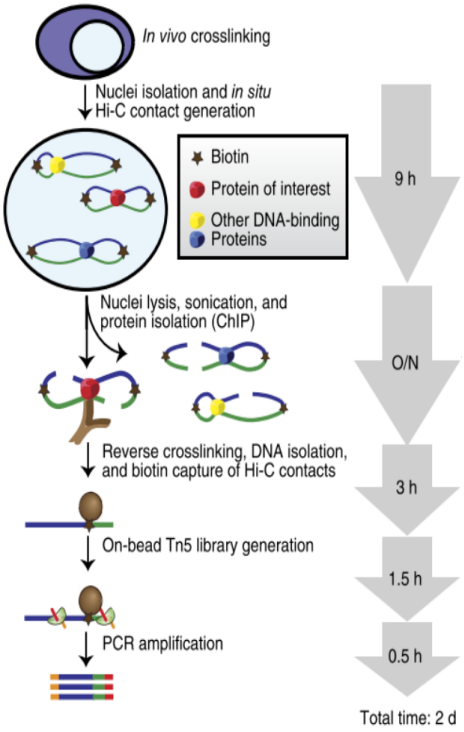


Figure 2.1.8: A schematic of the HiChIP protocol is shown. Reprinted by permission from Springer Nature: Nature (License #4487631458797), Mumbach et al. (2016), Copyright 2016.

advantage of HiChIP over ChIA-PET is that, it requires 100-fold lesser starting ma-

terial, and it can still provide a higher percentage of reads that are informative of the 3D conformation (Mumbach et al., 2016).

Other assays for detection of loci-specific interactions include capture-C (Hughes et al., 2014), capture-Hi-C (Jäger et al., 2015) and HiCap (Sahlén et al., 2015). These techniques enrich for interactions involving a specific set of loci which are identified in different ways. We refer the reader to the respective articles for further reading.

Ligation-Free Chromatin Interaction Detection

The molecular techniques described above profile interactions on the basis of the proximity-ligation principle. Beagrie et al. (2017) developed a ligation-free technique that provides an orthogonal view of the 3D chromatin architecture within cells. Beagrie et al. call this genome-wide chromatin contact detection approach 'Genome Architecture Mapping' (GAM).

GAM mainly employs cryosectioning and laser microdissection techniques from electron microscopy. Structurally preserved, fixated cells are first cryosectioned. Single nuclear profiles at random orientations from these cells are then isolated using laser microdissection. This is followed by extraction of DNA content in each nuclear profile which is then amplified and sequenced. Genomic loci that are proximal in 3D nuclear space are expected to be observed in the same nuclear profile than loci which are distant. Thus, a collection of many such nuclear profiles can together paint a picture of the 3D chromosomal organization inside the cell nucleus. As an inherent advantage of GAM, it can additionally infer the radial positions of the individual loci as well as their relative compaction. GAM can also detect contacts involving triplets of loci efficiently (Beagrie et al., 2017).

While GAM circumvents the biases influencing the digestion and ligation steps of 3C assays, it is affected by other biases such as window detection frequency, GC content and mappability. Beagrie et al. (2017) discuss the steps to estimate and normalize these biases.

2.1.5 Global Initiatives

Striking progress seen in the last decade in developing molecular techniques that yield better insights into the 3D conformation and the long-range interactions, has opened newer opportunities and challenges. Many of these are on the computational front – development of standardized approaches for comparisons of data from different states/conditions such as treated vs. untreated, tumor vs. normal etc. or chromatin interaction data over a series of time points during different cell-cycle stages or single-cell 3D interaction data. Additionally, insights gained from approaches using different types of information need to be put together to understand as complete a picture as

possible. The different types of information include 3D imaging data, such those obtained from FISH, and chromatin interaction data. With these goals in mind, consortia have been established, namely the NIH-4D Nucleome Network (Dekker et al., 2017) in the US, the EU 4DNucleome Initiative (Marti-Renom et al., 2018), LifeTime [https://lifetime-fetflagship.eu/] and MuG (Multiscale Complex Genomics) [https://www.multiscalegenomics.eu/] in EU, the Japan-4DNucleome (Tashiro and Lanctôt, 2015). Here, '4D' stands for four dimensions constituted by the three spatial dimensions and time.

2.2 Ewing Sarcoma

Ewing[6] sarcoma (EWS) is a rare, pediatric cancer. It usually develops in the bones or the soft tissues around them in young adults aged between 10 and 20. When developed in bones, the most common locations for development are the pelvis, legs, rib, arm or the spine, and when developed in a soft tissue, the common locations are the thigh, pelvic region, foot, spine and the chest wall. Although it can occur at any age, it is most frequently observed in adolescents. EWS is not inheritable.

A typical characteristic of EWS is the translocation of genetic material that involves the EWS gene and ETS family transcription factors (Delattre et al., 1992). The reciprocal chromosomal translocation between chromosomes 11 and 22 is typical (about 85% cases[7]) in EWS (Delattre et al., 1992)[8]. The translocation `t(11;22)(q24;q12)` gives rise to a fusion oncogene EWS-FLI1.

EWS has fewer somatic mutations than many other cancers, especially those common in adults, such as breast and colon cancer (Lawrence et al., 2013). Consequently, determining driving factors of EWS is an active area of research with many open questions. This includes designing proper treatment strategies for this malignant cancer. Like other cancers, EWS also has the following stages: localized, metastatic or recurrent. The treatment strategy for EWS depends on its stage. Current adopted treatment strategies for EWS include a combination of chemotherapy and surgery.

2.3 Machine Learning

2.3.1 Learning from Data: The Supervised and Unsupervised Way

I begin by briefly describing the concept of *learning from data* and an example that helps illustrate it. Consider we are given n data points with additional information

[6]Pronounced as: YOO-ing

[7]Different sources report that upto 95% of cases have this translocation

[8]Other translocations are also observed; these are between chromosomes 21 and 22, 7 and 22, and 17 and 22 (Delattre et al., 1992)

describing each of them. These are termed as *features* or *attributes* that describe the data points. Additionally, there could also be information available on the category that each of those data points belongs to. For example, consider a list of products available in a supermarket. The set of features describing them could include ingredients, packaging, make, fragility etc. Based on these features, the inventory supervisor is to make a decision on various aspects. Some examples are: (a) decide whether a certain product requires refrigeration or not; or (b) classify the products based on their shelf-life; or (c) grade the products from low-to-high or on a continuous scale; or (d) categorize the set of products which are seasonal, etc. An automated system that when fed with this products' information (input) can help identify the category (or categories) each product belongs to (outputs). Such a system is said to '*learn from data*'. When the categories assigned to the products are already given, the system uses this information to learn characteristic features distinguishing products of one category from those of the other. It is then tasked with predicting the categories for products that are as yet unseen for the system. This scenario is called supervised learning in which the machine is cognizant of the outputs and can use them to learn common patterns. In supervised learning, the discrete categories that the products need to be categorized into are typically referred to as '*classes*', and their names as '*class labels*'. This scenario is called '*classification*'. Instead of discrete classes as labels, the output labels could be continuous, for example, grading a product's popularity on a continuous scale of $0-5$ (low$-$high). This scenario is called '*regression*'. Before making predictions on the unseen data points, the stage in which the system *learns* from the available data is called the *training* stage. In order to evaluate the predictions made by such systems, the available data is often split into two chunks, one used for *training* and the other that is kept aside to be artificially treated as unseen or *test* data. Contrastingly, in the *unsupervised learning* scenario, no such information about the categories of products is available. In other words, in unsupervised learning, the information on the output label of each data point is missing. The system then simply '*clusters*' all data points into various categories based on their similarities and/or differences. The (dis)similarity is computed using the feature information of the data points. In this case, any number of clusters is plausible; the ideal number of clusters depends on the data and the task. Unsupervised learning is also known as '*clustering*'.

The goal of a supervised learning system is to make the best predictions on the unseen data points, and in doing so it is expected to generalize well. The goal of an unsupervised learning system is to cluster the data into groups such that the data points clustered in the same group are more similar to each other than those clustered into different groups. The 'learning problems' are modeled and solved mathematically. In the following, we take a brief dive into the formal mathematical model for

classification.

A General Mathematical Model of Classification

The learning problem we stated earlier—classification of products in a supermarket—can be more generally but succinctly expressed mathematically as follows. Consider that we are given some input data expressed in pairs: $(\mathrm{x}_i, y_i) \in \mathcal{X} \times \{+1, -1\}$. Here, the x_i are instances of the non-empty set $\mathcal{X}$ representing the observed feature values, and the y_i are the class assignments for each x_i, the class labels being $+1$ and -1. When there are only two possible classes any instance x_i can exclusively belong to, it is called *binary* classification. The two classes are thus typically called the 'positive' and the 'negative' class. This explains the $+$ and $-$ in the labels which is intuitive and also mathematically convenient as we will see later. From a more general point of view, the class labels can also be 0 and 1. If there are more than two possible classes, the prediction scenario is called *multi-class* classification. In binary or multi-class classification, any instance can be assigned only one class label out of those possible. In other words, the different classes are mutually exclusive. However, when they are not mutually exclusive, an instance can be assigned more than one class label at the same time, and it is called *multi-label* classification. We only consider the binary classification scenario in the rest of this section.

The problem of binary classification is essentially inferring a function,

$$f : \mathcal{X} \to \{-1, +1\}. \tag{2.2}$$

Intuitively, the machine is first shown a set of input data and their class assignments (training data). It is then expected to accurately classify any new, as yet unseen, data point(s) (test data). To achieve this, the machine looks for similarities between any new data point and the set of points belonging to the positive class, and the negative class. If this new point is more similar to points in the positive class, the machine classifies it as positive, otherwise negative (assigns the label $+1$ or -1 respectively). This is done for all of the test data points independently.

The notion of similarity: We now look into the reason why considering similarity between data points works. It is assumed that the set of data points we are working with are sampled from a probability distribution $\mathrm{P}(\mathrm{x}, y)$ which is unknown, but is fixed. Additionally, these data points are independent and identically distributed (abbreviated as IID, a notion very common in statistics). Furthermore, we assume that the test data points are also sampled from the same unknown distribution.

Before proceeding, let's consider how we can compute similarities given data points, x_i. These fixed-dimensional vectors can be the features describing the products in our earlier example, but in mathematical abstraction these are simply considered points in

a d-dimensional space. The problem is now geometrical with points in d-dimensional input space. Treating the input data points as vectors, we can perform all linear algebraic operations in this space with geometric interpretations. For example, we can compute dot products (also called *inner products*) between vectors directly in the input space giving a notion of similarity. When the data, $\mathcal{X}$, is not directly available in a vector format, e.g., structured data such as strings, we often construct a *feature map* denoted by Φ that transforms $\mathcal{X}$ in the input space to the feature space which is endowed[9] with dot product.

$$\Phi : \mathcal{X} \to \mathcal{H} \tag{2.3}$$

Firstly, by using a feature map to represent any $x \in \mathcal{X}$, we can vectorize our data points when they are not. Secondly, even when they are already in vector form, choosing a more suitable feature map for a given problem can be beneficial. Some examples of both the cases are presented in the latter sections of this chapter. The machine can now "read" the data and quantitatively understand what is similar and to what extent.

Accurate classification amounts to finding a function f in Eq. (2.2) that generalizes well to the test data also generated from the same probability distribution as the training data (assumption). Alternatively, the function f can be called a *model* or *hypothesis*. We can evaluate the performance of a machine in classifying test data points using mathematical loss functions ($\mathcal{L}$) which quantitatively inform about the correctness of its classifications. Intuitively, a loss function computes the difference between the predicted class label $\hat{y} = f(\mathrm{x})$ and the true class label y for each data point classified. Some popular examples of loss functions used in machine learning applications are the 0–1 loss, hinge loss and squared loss. The 0–1 loss is the simplest one. It just checks if the predicted and the true class label are the same. It is mathematically written as follows.

$$\mathcal{L}(x, y, \hat{y}) = \begin{cases} 0, & y = \hat{y}, \\ 1, & y \neq \hat{y} \end{cases} \tag{2.4}$$

Often, just saying that the predicted class label is not the same as the true class label is not enough. We are more interested in knowing how certain is the classifier of its prediction for a data point. Therefore, $\hat{y} = \mathrm{sgn}(f(\mathrm{x}))$, when $f(\mathrm{x})$ is real-valued, is treated as the class label and the numerical quantity $|f(\mathrm{x})|$ gives the confidence in

[9] Alternatively, also called *equipped*; See: https://math.stackexchange.com/q/961040

the prediction. This notion is used in the hinge loss.

$$\mathcal{L}(\mathrm{x}, y, f(x)) = \begin{cases} 0, & yf(\mathrm{x}) \geq 1, \\ 1 - yf(\mathrm{x}), & \text{otherwise} \end{cases} \tag{2.5}$$

Hinge loss is also known as the *soft margin* loss, as will be clear later (in the Subsection "The Support Vector Machine (SVM) for Classification").

In a real world application, we would like to deploy such a machine in order to automate a certain task, such as the task of product classification in a supermarket. Our expectation would be that it makes as few errors as possible in classifying any future data points which are still unseen. Therefore, for the task, the objective is to find (or choose) an f from set $\mathcal{F}$ of all possible functions, that generalizes well. This means that the chosen f minimizes the overall loss in classifying the test data points. But all that the machine can *see* a priori is the training data, and only the error in classifying the training data can be computed beforehand (Eq. (2.7)). This error is called the *training error* or the *empirical risk*.

$$\mathrm{R}_{Emp}(f) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\mathrm{x}_i, y_i, f(\mathrm{x}_i)) \tag{2.6}$$

$$= \hat{\mathbb{E}}[\mathcal{L}(\mathrm{x}_i, y_i, f(\mathrm{x}_i))] \tag{2.7}$$

What we can certainly do is choose a function that minimizes the *empirical risk*,

$$f^* = \arg\min_{f \in \mathcal{F}} R_{Emp}(f) \tag{2.8}$$

and hope that f^* will also minimize the *true risk* which is the misclassification error on the test data.

$$\arg\min_{f \in \mathcal{F}} R(f) \approx \arg\min_{f \in \mathcal{F}} R_{Emp}(f) \tag{2.9}$$

On choosing f: By the principle of *structural risk minimization* (SRM), one chooses a function that minimizes the empirical risk (*empirical risk minimization*) and is the least complex (Bousquet et al., 2004). Mathematically,

$$f_{\text{SRM}} = \arg\min_{f \in \mathcal{F}} R_{Emp}(f) + penalty(f)_{\text{model complexity}} \tag{2.10}$$

Further to SRM, existing methods use regularized empirical risk minimization (Bousquet et al., 2004).

Finally, there is an important caveat to note here. A small training error does not necessarily guarantee a small test error. The chosen function must simultaneously be restricted as well as rich enough, so that it can predict the non-trivialities well and

also recognize any hidden regularities (or patterns) in the distribution, $\mathrm{P}(\mathrm{x}, y)$. We refer the reader to Bousquet et al. (2004); Cortes and Vapnik (1995); Scholkopf and Smola (2001) for additional in-depth reading.

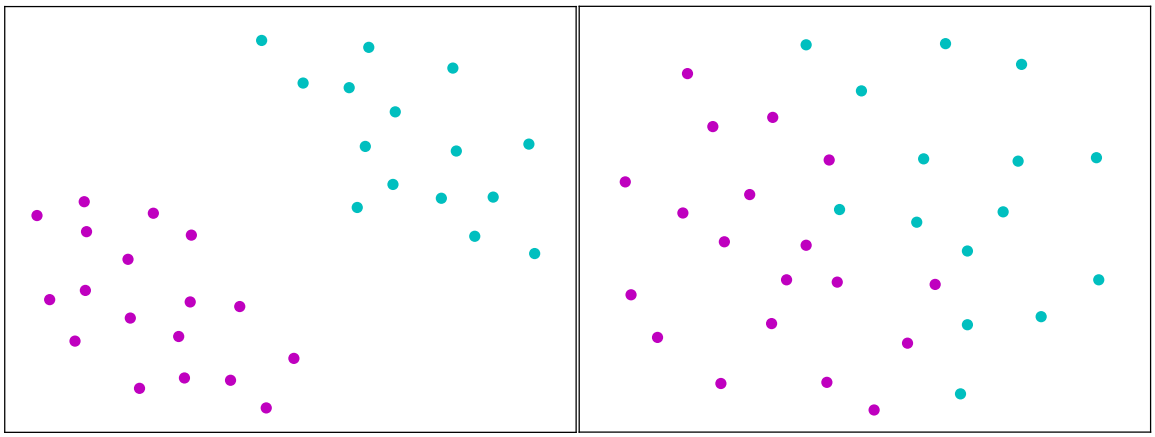


Figure 2.3.1: Linearly separable and non-separable sets of data points are shown in the left and right panels respectively. Points in the positive class are shown in magenta, and those in negative class are shown in cyan.

The Support Vector Machine (SVM) for Classification

In this subsection, we introduce the support vector machine, a very popular machine learning technique for classification. Support vector machines are optimal hyperplane classifiers. I discuss only the binary classification case.

Consider the set of points in 2D shown in Figure 2.3.1, left panel. The magenta colored points belong to the positive class $(+1)$ and others, in cyan, to the negative class (-1). As in the earlier subsection, these points are mathematically represented as $\mathrm{x}_i \in \mathcal{X} | \mathcal{X} = \mathbb{R}^p$. In Figure 2.3.1, $p = 2$. For the case of linearly separable data points, the SVM attempts to learn a linear function f that can separate the two sets of points belonging to the two classes. This function has the form

$$f(\mathrm{x}) = w^T \mathrm{x} + b \tag{2.11}$$

where $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$. This equation represents an hyperplane. For any data point, if $y_i f(\mathrm{x}_i) \geq 0$, the point is correctly classified, otherwise, not. There are many possible hyperplanes that can help achieve the objective of perfectly classifying the given set of points (see left panel, Figure 2.3.2). We described the criterion followed for choosing such an f from $\mathcal{F}$ in the earlier subsection.

When the points are not perfectly separable linearly (right panel, Figure 2.3.1), SVMs allow a small number of misclassifications. This is done by deploying the hinge loss (Eq. (2.5)). As will see later, this is called the *soft margin* case. The right panel

in Figure 2.3.2 shows two possible hyperplanes in this scenario of data points not being linearly separable. Each of them misclassifies an equal number of points, but have different confidences ($|f(\mathrm{x})|$). From among several choices of f, SVMs focus on choosing an f that also maximizes the confidence in its classifications. This helps SVMs choose one among the possible hyperplanes, thus attaining a unique solution in both scenarios.

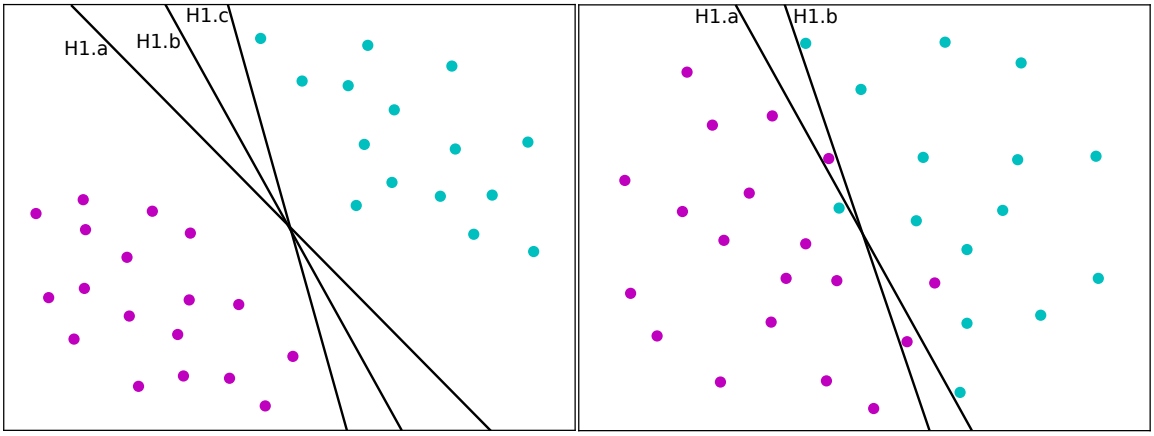


Figure 2.3.2: Multiple possible hyperplanes for the linearly separable and non-separable cases are shown in the left and right panels respectively. In the non-separable case (right panel), both hyperplanes, H1.a and H1.b, misclassify equal number of data points (3), but with different confidences.

The concept of *margin*: Observe that the hyperplane defined by f creates two half-spaces[10] which can be denoted by $h^+ = \{\mathrm{x} : f(\mathrm{x}) \geq 1\}$ (points in positive class), and $h^- = \{\mathrm{x} : f(\mathrm{x}) \leq -1\}$ (points in negative class). The distance between these two half-spaces is given by $2 \times \frac{1}{\|w\|}$ and is called the *margin*. Then, it would be ideal to choose an f that supports a maximal separation between the two half-spaces h^+ and h^-, i.e. a maximum margin. We note that maximizing $\frac{2}{\|w\|}$ is equivalent to minimizing $\frac{1}{2}\|w\|^2$ (squared norm). Mathematically, this translates to the following minimization problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|w\|^2 \\ &\text{subject to} && y_i(w^T \mathrm{x}_i + b) \geq 1 \end{aligned} \tag{2.12}$$

This minimizes the empirical risk. Additionally, as noted, by using the hinge loss function, we can achieve a trade-off between minimizing empirical risk and generalization. In other words, the chosen hyperplane is permitted to make a few classification errors (in the training data) with small confidence. This is useful when the data is not

[10]They are formed by division of an affine space by a hyperplane which, here, is given by f, an affine function.

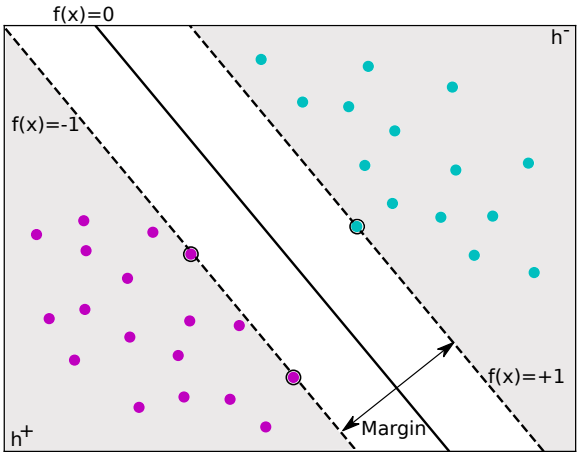


Figure 2.3.3: The hyperplane, f(x), is represented as a solid line, and the boundaries of the two half-spaces, h^+ and h^-, as dashed lines.

completely linearly separable.

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\mathcal{L}(\mathrm{x}_i, y, f) \\ \text{subject to} \quad & y_i(w^T\mathrm{x}_i + b) \geq 1 \\ & C > 0 \end{aligned} \tag{2.13}$$

While a small value of C assigns a relatively higher importance to finding a large margin in comparison to the confidence on the predictions, a large value of C means that the margin could be smaller but the predictions have stronger confidences. This is simply due to the fact that with hinge loss (Eq. (2.5)) and $f(\mathrm{x}) = w^T\mathrm{x} + b$, we are computing the distance of a misclassified x from the correct half-space.

Cortes and Vapnik (1995) introduced the so-called *slack variables*, ξ_i, to replace the hard, hinge loss constraint. This accounts for the data points for which the hinge loss constraint in Eq. (2.13) is not fulfilled—these are points lying in the wrong half-space.

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(w^T\mathrm{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & C > 0 \end{aligned} \tag{2.14}$$

For each x_i, the corresponding slack variable $\xi_i = max(0, 1 - y_i(w^T\mathrm{x}_i + b))$. The sum of ξ_i gives an upper bound on the training error. This formulation is called the *soft margin* SVM, and that in Eq. (2.12), the *hard margin* SVM.

Eq. (2.14) is a constrained, quadratic optimization problem which can be solved by taking its Lagrangian (Boyd and Vandenberghe, 2004). This entails introducing

Lagrangian multipliers, $\boldsymbol{\alpha} = \alpha_i \geq 0$ for each constraints in $y_i(w^T \mathrm{x}_i + b) \geq 1 - \xi_i$, and $\boldsymbol{\beta} = \beta_i \geq 0$ for $\xi_i \geq 0$ as follows.

$$\begin{aligned} L(w, b, \xi, \alpha, \beta) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &- \sum_{i=1}^{n}\alpha_i[\xi_i - 1 + y_i(w^T\mathrm{x}_i + b)] \\ &- \sum_{i=1}^{n}\beta_i\xi_i \end{aligned} \tag{2.15}$$

Solving for the unique saddle point of L will give the maximum w.r.t. the Lagrangian variables $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, and minimum w.r.t. (w, b, ξ). Proceeding as usual for finding the minimum, set the partial derivatives of L w.r.t. (w, b, ξ) to 0.

$$\frac{\partial L}{\partial \mathrm{w}}(\mathrm{w}, b, \xi, \alpha, \beta) = w - \sum_{i=1}^{n}\alpha_i y_i \mathrm{x}_i = 0 \implies w = \sum_{i=1}^{n}\alpha_i y_i \mathrm{x}_i \tag{2.16}$$

$$\frac{\partial L}{\partial b}(\mathrm{w}, b, \xi, \alpha, \beta) = \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{2.17}$$

$$\frac{\partial L}{\partial \xi_i}(\mathrm{w}, b, \xi, \alpha, \beta) = C - \alpha_i - \beta_i = 0 \quad \text{for} \quad \forall i = \{1, \ldots, n\} \tag{2.18}$$

Substituting 2.16 in 2.15, and using 2.17 and 2.18,

$$\begin{aligned} & L(w, b, \xi, \alpha, \beta) \\ &= \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[\xi_i - 1 + y_i(w^T\mathrm{x}_i + b)] - \sum_{i=1}^{n}\beta_i\xi_i \\ &= \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\xi_i + \sum_{i=1}^{n}\alpha_i[1 - y_i(w^T\mathrm{x}_i + b)] - \sum_{i=1}^{n}\beta_i\xi_i \\ &= \frac{1}{2}\langle w, w\rangle + \sum_{i=1}^{n}\xi_i \underbrace{[C - \alpha_i - \beta_i]}_{\text{use 2.18}} + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i y_i(w^T\mathrm{x}_i + b) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathrm{x}_i^{\mathrm{T}}\mathrm{x}_j + \sum_{i=1}^{n}\alpha_i - b\underbrace{\sum_{i=1}^{n}\alpha_i y_i}_{\text{use 2.17}} - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathrm{x}_i\mathrm{x}_j \\ &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathrm{x}_i^{\mathrm{T}}\mathrm{x}_j + \sum_{i=1}^{n}\alpha_i \end{aligned} \tag{2.19}$$

With 2.19, the *dual* of the *primal* problem (2.14) is expressed simply in terms of the

Lagrangian variables $\boldsymbol{\alpha}$.

$$\begin{aligned} \max_{\alpha \in \mathbb{R}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathrm{x}_i^{\mathrm{T}}\mathrm{x}_j + \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \quad \forall i \in \{1,\ldots,n\} \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{2.20}$$

Thus, the SVM decision function for a data point x is given by

$$f(\mathrm{x}) = \mathrm{sgn}(w^T\mathrm{x} + b) = \mathrm{sgn}\big(\sum_{i=1}^{n}\alpha_i y_i \mathrm{x}_i^T\mathrm{x} + b\big). \tag{2.21}$$

The data points x_i only appear as dot products.

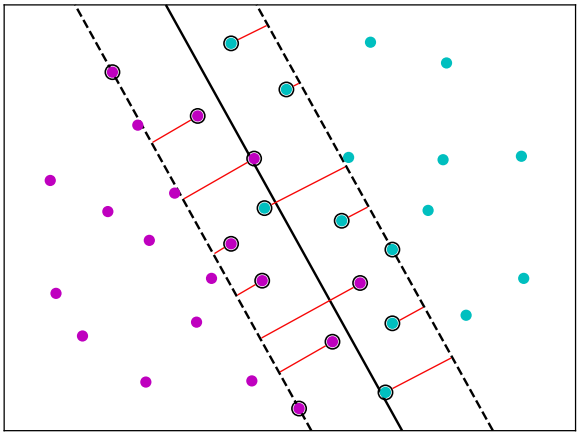


Figure 2.3.4: The hyperplane, f(x), is represented as a solid line. The data points with black borders are support vectors ($\alpha \neq 0$). Shown in red are the penalties (ξ) for points on the wrong side of their corresponding half-space boundary.

Support vectors: The data points with a non-zero α value, are called *support vectors*. The points that lie at the half-space boundary or on the wrong side of the corresponding half-space boundary are the support vectors (*soft margin* case). Figure 2.3.4 shows the case where all support vectors either lie at the corresponding half-space boundary or within the margin. As an extreme case, some support vectors can lie even further inside the opposite half-space. The number of points serving as support vectors is often lower than the total number of points, and serves as the upper bound on the error rate of the classifier (Scholkopf and Smola, 2001). From Eq. (2.16), we note that the solution obtained only depends on the support vectors since only the points with a non-zero α contribute to w. In contrast to the *soft margin* case, there are no points within the margin for the *hard margin* case.

Cross-validation (CV): Coming back to *true risk*, R_f in Eq. (2.9), since any real 'future' data is inaccessible, one uses *cross-validation* to evaluate and select a model. Cross-validation mimics the scenario of unseen future test data as follows. The available data is divided into two portions, also called *folds*, one for training our classifier and the other to test its performance. This fold of data kept for testing is artificially treated as unseen data for the machine. Usually, this procedure is repeated k-times to achieve generalization, and is called k-fold cross-validation. In it, per iteration, one of the k-folds is treated as test data, with the remaining $k-1$ folds used for training in that iteration. This is schematically represented in Figure 2.3.5. An extreme case of k-fold cross-validation is leave-one-out cross-validation where $k = \#\text{samples}$.

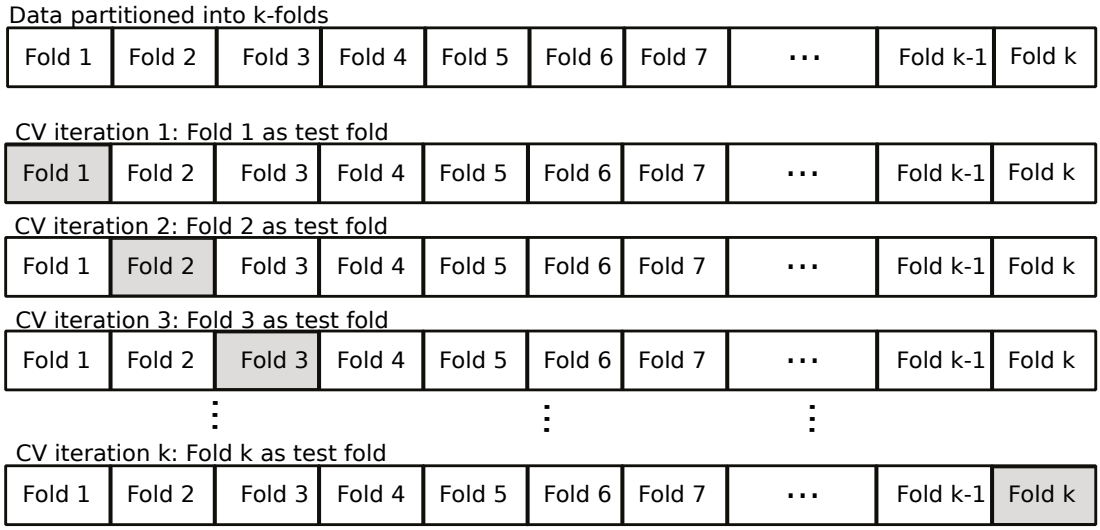


Figure 2.3.5: k-folds cross validation procedure is shown. For each iteration, the shaded fold is used as the test fold.

Learning algorithms have parameters that need to be optimized such that their performance can be maximized. These are called hyperparameters. For example, SVMs have the cost parameter that should be tuned. Kernels, which are introduced in the latter sections, also have parameters that can (and should) be tuned. The CV procedure incorporates tuning as follows. This can be performed using what is known as *nested* cross-validation. In it, data is first partitioned into k outer folds. Then, in each of the k iterations, data in the corresponding $k-1$ training folds is further treated with k- or t-fold cross-validation. Here, one fold (say, the t^{th} fold) is used for tuning the hyperparameters instead of testing.

2.3.2 On Kernels and Their Properties

We now turn our focus to feature spaces, $\mathcal{H}$ in Eq. (2.3), and their properties.

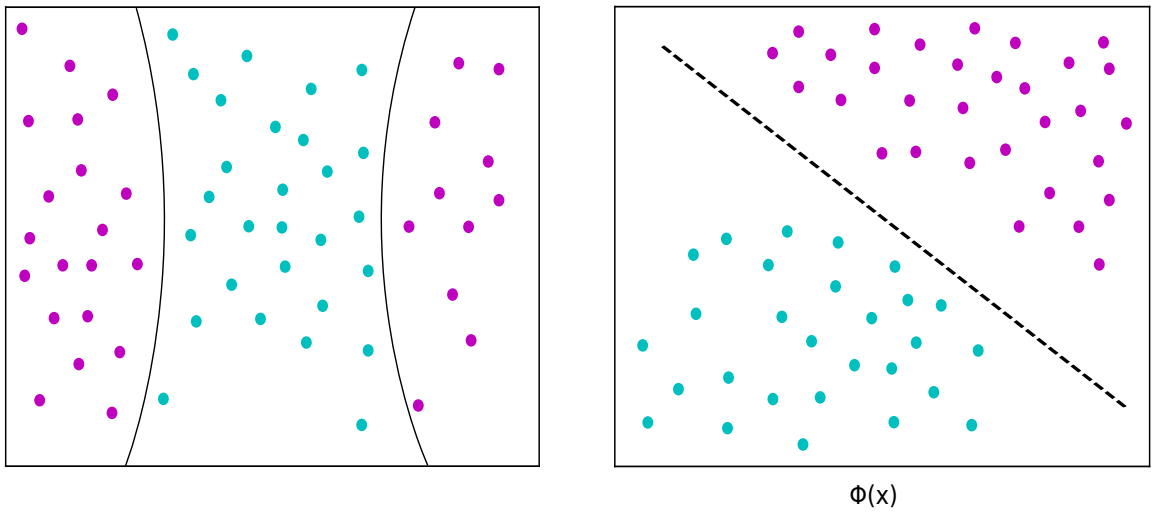


Figure 2.3.6: Left panel: Data points of two classes (shown in cyan and magenta) in the input space are not separable with a linear decision boundary. Right panel: Transforming to the (possibly higher dimensional) feature space using feature map Φ enables separating them well.

There can be scenarios where the data is not well separable using a linear hyperplane in the input space. In such cases, the SVM must be able to realize non-linear decision boundaries in the input space to separate data points belonging to the different classes. Figure 2.3.6 shows such a scenario. It can be observed that no linear decision boundary can separate the magenta points from the cyan ones. Such data can be transformed to a feature space, Φ, where they are well separable, before presenting them to an SVM. Typically, this feature space has higher dimensionality than the input space. Depending on the feature map, computing the explicit feature representation for each data point could be a time-consuming, computationally expensive task. Furthermore, recall that for an SVM we only require information on the dot product between data points instead of their explicit representation. This also follows for the feature space. This is made possible by the so-called *kernels* or *kernel* functions.

Kernels can help project the data points from a lower dimensional input space to a possibly higher- or infinite-dimensional feature space where they are better separable. We call $k(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle$ a *kernel* function. In other words, a *kernel* function is a function that returns the dot product between the feature space representations of any two input data points. A matrix of pairwise kernel values (similarity scores) between all data points $\mathrm{x}_1, \ldots, \mathrm{x}_n$ is a square matrix of size $n \times n$. It is called the *Gram* or the *kernel* matrix. A Gram matrix is a positive definite matrix (see proof below).

Symmetric and Positive Definiteness:

$$\text{Symmetric :} \qquad k(\mathrm{x}_i, \mathrm{x}_j) = k(\mathrm{x}_j, \mathrm{x}_i) \tag{2.22}$$

$$\text{Positive Definite :} \qquad \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \quad n \in \mathbb{N}, c_{1\ldots,n} \in \mathbb{R} \tag{2.23}$$

Note that any symmetric matrix is said to be positive definite if and only if all its eigenvalues are non-negative. In 2.23, equality is attained only when any $c_i = 0$. Positive definiteness of a Gram matrix, $\mathbf{G}$, can be shown as follows. With

$$\mathbf{G}_{ij} = k(\mathrm{x}_i, \mathrm{x}_j) = \langle \Phi(x_i), \Phi(x_j) \rangle, \forall i, j \in [1, \ldots, n],$$

for any vector v,

$$\begin{aligned} v'\mathbf{G}v &= \sum_{i,j=1}^{n} v_i v_j \mathbf{G}_{ij} \\ &= \sum_{i,j=1}^{n} v_i v_j \langle \Phi(\mathrm{x}_i), \Phi(\mathrm{x}_j) \rangle \\ &= \sum_{i=1}^{n} v_i \Phi(\mathrm{x}_i) \sum_{j=1}^{n} v_j \Phi(\mathrm{x}_j) \\ &= \| \sum_{i=1}^{n} v_i \Phi(\mathrm{x}_i) \|^2 \geq 0 \end{aligned} \tag{2.24}$$

It is quite possible that, explicit computation of the feature map Φ may not always be convenient (or computationally cheap). In which case, we can still define a kernel function without explicit construction of the feature space, due to the following theorem that guarantees that for any valid kernel, such a feature space exists.

Theorem 1. *For any kernel k on $\mathcal{X} \times \mathcal{X}$, where $\mathcal{X}$ is a non-empty set, there exists a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that*

$$k(\mathrm{x}_1, \mathrm{x}_2) = \langle \Phi(\mathrm{x}_1), \Phi(\mathrm{x}_2) \rangle, \forall \mathrm{x}_1, \mathrm{x}_2 \in \mathcal{X}, \tag{2.25}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ represents a dot product in the Hilbert space.

Together, we conclude that for a function to be a valid kernel function, it suffices to show that its Gram matrix is positive definite (since every inner product is a positive definite function).

A Kernel and Its Reproducing Kernel Hilbert Space: A reproducing kernel Hilbert space is defined as follows.

Definition 2.3.1. Let X be a non-empty set and $\mathcal{H}$ be a $\mathbb{K}$-Hilbert space over $\mathcal{X}$, i.e. a $\mathbb{K}$-Hilbert space that consists of functions mapping from X into $\mathbb{K}$.

(i) A function $k : \mathcal{X} \times \mathcal{X} \to \mathbf{K}$ is called a reproducing kernel of $\mathcal{H}$ if we have $k(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{X}$ and the reproducing property

$$f(x) = \langle f, k(\cdot, x) \rangle$$

holds $\forall f \in \mathcal{H}$ and all $x \in \mathcal{X}$.

(ii) The Hilbert space $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) over $\mathcal{X}$ if $\forall x \in \mathcal{X}$ the Dirac functional $\delta_x : \mathcal{H} \to \mathbf{K}$ defined by

$$\delta_x(f) := f(x), f \in \mathcal{H}$$

is continuous.

A Note on Normalization

Feature space normalization is an important, recommended step to be performed when training an SVM (Herbrich and Graepel, 2001; Shawe-Taylor and Cristianini, 2004). Feature space normalization was shown to have a large impact on the generalization error of an SVM classifier (Herbrich and Graepel, 2001). For any kernel matrix K, the feature space normalized kernel $\tilde{K}$ is given by

$$\tilde{K_{ij}} = \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}}, \tag{2.26}$$

where K_{ij} is a short-hand for $K(\mathrm{x}_i, \mathrm{x}_j)$. Eq. 2.26 is especially helpful when the feature map used is non-linear and unknown. For a linear feature map such as simple dot products, normalization in the feature space is equivalent to that in the input space performed using *norms*.

$$\tilde{K_{ij}} = \left\langle \frac{\Phi(\mathrm{x_i})}{\|\Phi(\mathrm{x_i})\|}, \frac{\Phi(\mathrm{x_j})}{\|\Phi(\mathrm{x_j})\|} \right\rangle = \frac{\langle \Phi(\mathrm{x_i}), \Phi(\mathrm{x_j}) \rangle}{\|\Phi(\mathrm{x_i})\| \|\Phi(\mathrm{x_j})\|} = \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}} \tag{2.27}$$

Therefore, the kernels we use in this thesis are feature space normalized.

Examples of Kernels

For Real Valued Data: Two popular examples of kernel functions for real-valued data are polynomial kernel and the Gaussian Radial Basis Function kernel.

$$\text{Polynomial kernel :} \quad k(\mathrm{x}_1, \mathrm{x}_2) = (\langle \mathrm{x}_1, \mathrm{x}_2 \rangle + c)^d \tag{2.28}$$

where c is a constant and $d \in \mathbb{N}$.

$$\text{Gaussian RBF kernel :} \quad k(\mathrm{x}_1, \mathrm{x}_2) = \exp\big(-\gamma \|\mathrm{x}_1 - \mathrm{x}_2\|^2 \big) \tag{2.29}$$

One may note that the linear kernel is the special case of the polynomial kernel with $c = 0$ and $d = 1$. Additionally, the Gaussian RBF kernel is a translation invariant kernel. It is the most widely used kernel due to its capacity to generate any complex classification function. This capacity can be controlled by manipulating the parameter γ.

For Structured Data: Data across domains may not always be available in the vector form. Examples of structured data are graphs, text documents, images etc. Computational biology, in particular, has many tasks where data is available in a structured form, e.g., DNA or protein sequence, protein 3D structure representations or even images from microscopy. As we have discussed earlier, given a suitable feature map, one can always define similarity measures or kernels for such data. Since this thesis primarily focuses on DNA sequences (strings), I dedicate Subsection 2.3.3 to a discussion of popular, state-of-the-art string kernels. Finally, I refer the reader to Gärtner (2003)'s survey article on kernels for structured data and Shawe-Taylor and Cristianini (2004) book Kernel Methods for Pattern Analysis for further reading.

2.3.3 String Kernels

I now discuss some popular examples of string kernels that were developed to be suitable for solving problems involving strings in biology.

A note on the nomenclature used in computational biology/bioinformatics, and also in the rest of this thesis. A protein or DNA sequence is simply a string of characters. Here after in this thesis, whenever the object (or data point) is a string, we represent it by s instead of x . The length of any sequence s is the number of characters in it, and is typically represented by $|\mathrm{s}| = L$. The alphabet for proteins has 20 characters for the 20 naturally occurring amino acids[11], while that for DNA has 4 characters {`A,C,G,T`} as seen earlier. The length of the alphabet is represented by $|\Sigma| = l$. Words, which are k-length subsequences (substrings), are also called as k-mers. Bioinformatics also uses the term '*oligomers*' interchangeably with k-mers with any value for k. So does this thesis.

The Spectrum Kernel

The spectrum kernel is one of the simplest string kernels that was designed for the protein sequence classification problem, but it is also generally applicable to any case involving sequences (Leslie et al., 2002).

In general, the spectrum kernel represents each sequence using a feature map which counts the number of times each k-mer occurs in the sequence for all possible k-mers.

[11] A complete list can be looked up at: http://www.virology.wisc.edu/acp/Classes/DropFolders/Drop660_lectures/SingleLetterCode.html

We know that, given the alphabet, the set of all possible k-mers is given by Σ^k. Let $|\Sigma^k| = M$, and the iterator, $m_{i \in [1,M]}$. Thus, for any sequence s, its k-spectrum feature map, $\Phi_k(\mathrm{s})$, is given as

$$\Phi_k(\mathrm{s}) = (\phi_{m_i}(\mathrm{s}))_{m_i \in \Sigma^k} \tag{2.30}$$

where $\phi_{m_i}(s)$ denotes the frequency of occurrence of k-mer m_i in s where $k \geq 1$. Then, the k-spectrum kernel value of sequence pair $(\mathrm{s}_1, \mathrm{s}_2)$ is

$$\mathrm{k}_k(\mathrm{s}_1, \mathrm{s}_2) = \langle \Phi_k(\mathrm{s}_1), \Phi_k(\mathrm{s}_2) \rangle \tag{2.31}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product. A further simpler alternative is when $\phi_{m_i}(\mathrm{s})$ in Eq. (2.30) indicates just the presence or absence of the k-mer m_i in sequence s instead of the frequency.

Intuitively, this feature map captures the profile of a sequence based on its constituent k-mers and their occurrence frequencies. It is the *bag-of-words* kernel (popular in natural language text classification and information retrieval) for biological sequences. The authors used the spectrum kernel in conjunction with the SVM for remote homology detection in protein sequences where it was shown to attain performance comparable to the then state-of-the-art approaches.

String Kernels Considering Occurrence Positions of Features

You may have noted that the spectrum kernel feature map takes the k-mers and their occurrence frequencies into account, however, it is indifferent to their occurrence positions in the sequences. From a biological point of view, the position at which an oligomer occurs in a sequence could have an impact on the underlying biology (cf. section 2.1.3). For example, the position of the oligomer '`TATAAA`', called the `TATA`-box, in promoter sequences plays an important role. Also, several other motifs corresponding to the basal transcription machinery such as TFs IIA, IIB etc., or the intiator are expected to be within a certain distance from the TSS (Butler and Kadonaga, 2002; Juven-Gershon et al., 2008; Smale and Kadonaga, 2003). Other problems involving splice sites or translation initiation sites can also be characterized by the position of the motif in the corresponding sequences. Thus, several kernels have been developed to address this aspect. These kernels specifically consider positions of motifs in sequences when comparing them. We discuss such kernels next.

The Weighted Degree Kernel

The weighted degree kernel (WDK) was proposed to account for the position information of features when computing sequence similarity (Rätsch and Sonnenburg, 2004).

Mathematically, the *WDK of order d* is written as,

$$\mathrm{k}(\mathrm{s}_1, \mathrm{s}_2) = \sum_{k=1}^{d} \beta_k \sum_{p=1}^{L-k+1} \mathrm{I}[u_{k,p}(\mathrm{s}_1) = u_{k,p}(\mathrm{s}_2)] \tag{2.32}$$

where $u_{k,p}(\mathrm{s})$ denotes k-mers at any position p in the sequence s. Here, $k = [1, d]$, and $\mathrm{I}(\cdot)$ denotes the indicator function. In the *WDK of order d*, Rätsch and Sonnenburg proposed weighting the k-mer matches such that longer matches are effectively assigned higher weight. The authors proposed the weighting parameter $\beta_k = 2\frac{d-k+1}{d(d+1)}$, which assigns $\beta_d < \beta_{d-1} < \ldots < \beta_1$. However, every longer k-mer match has many shorter k-mer matches. Thus, any pair of sequences that have many longer k-mer matches score essentially higher than a pair in which only shorter k-mers match. Finally, the two sequences should be of the same length for comparison.

In summary, the *WDK of order d* compares two sequences according to their $[1, d]$-spectrum at each position in the sequence where a k-mer can start (thus, the upper limit $L - k + 1$ on the second summation in Eq. (2.32)). Thus, the WDK feature map is much richer than that of the spectrum kernel. The WDK has been shown to be the state-of-the-art approach for the splice site recognition problem (Rätsch and Sonnenburg, 2004).

From Exact to Inexact Matching For Comparisons

In many real world applications, finding a match is rarely about spotting the exact one. For example, in the problem we discussed earlier, for categorizing food products in a super market, not every banana looks the same or not every yoghurt pack looks exactly the same; they are same or similar, some more than others. Similarly, in biology, protein as well as DNA sequence motifs are mostly degenerate. A DNA binding site recognized by a TF is one such example. While a TF could have a high affinity to a particular sequence of nucleotides for binding, this affinity gradually decreases as the sequence deviates, becoming more and more non-specific. Therefore, it is important that approaches to compare sequences allow facets like gaps, shifts and certain degree of mismatches when comparing sequences. To that end, the spectrum kernel described above has a variant that compares sequences with mismatches (Leslie et al., 2004), and gaps (Leslie and Kuang, 2003) permitted. The weighted degree kernel *with shifts* (WDKS) permits the start positions of the matching k-mers in the two sequences to be slightly shifted, although the shift is penalized (Rätsch et al., 2005).

$$\mathrm{k}(\mathrm{s}_1, \mathrm{s}_2) = \sum_{k=1}^{d} \beta_k \sum_{p=1}^{L-k+1} \sum_{s=0}^{S} \delta_s \mathrm{I}[u_{k,p+s}(\mathrm{s}_1) = u_{k,p}(\mathrm{s}_2)] + \mathrm{I}[u_{k,p}(\mathrm{s}_1) = u_{k,p+s}(\mathrm{s}_2)] \tag{2.33}$$

Here, $\delta_s = \frac{1}{2(s+1)}$ penalizes any shift (s) in the start positions (p) of the k-mers. This tolerance of shift decays with increase in the amount of shift.

I next describe two dot product kernels, the oligo kernel (Meinicke et al., 2004) and the oligomer distance histograms (ODH) kernel (Lingner and Meinicke, 2006). With both of them, one can inherently interpret their rich feature maps and visualize the sequence features. The oligo kernel was the first position-aware approach to also permit positional uncertainty (inexactness of position) in sequence comparison (Meinicke et al., 2004).

The Oligo Kernel

Meinicke et al. (2004) proposed a feature map constructed by defining an *oligo function* for occurrences of all possible k-mers, over a given alphabet, in a sequence. The idea here is to account for the positional as well as the compositional uncertainty.

Every possible k-mer in Σ^k has a corresponding *oligo function* which characterizes their occurrences in a given sequence and the associated positional uncertainty using Gaussians as shown in Eq. (2.34) and 2.35. The feature map for any sequence s is a concatenation of the oligo functions, μ_{m_i}.

$$\mu_{m_i}(t) = \sum_{p \in S_{m_i}} e^{\frac{-(t-p)^2}{2\sigma^2}} \tag{2.34}$$

$$\Phi(\mathrm{s}) = [\mu_{m_i \in \sum^k}]^{\mathrm{T}} \tag{2.35}$$

In Eq. (2.34), t denotes the finite number of discrete positions (in a sequence) considered for representation, σ captures the degree of positional uncertainty, and S_{m_i} represents the positions in sequence s where m_i occurs. $[\cdot]^{\mathrm{T}}$ denotes the transpose in Eq. (2.35). Comparing two sequences then entails computing the inner product of their feature maps.

$$\mathrm{k}(\mathrm{s}_1, \mathrm{s}_2) = \Phi(\mathrm{s}_1)\Phi(\mathrm{s}_2) \tag{2.36}$$

$$= \sqrt{\pi}\sigma \sum_{m_i \in \Sigma^k} \sum_{p \in S_{m_i}} \sum_{q \in S_{m_j}} e^{-\frac{1}{4\sigma^2}(p-q)^2} \tag{2.37}$$

The case $\sigma \to 0$ considers only exact positional matches of k-mers, and $\sigma \to \infty$ allows infinite distance between the starting positions of the k-mers in the two sequences. The latter case makes it equivalent to the spectrum kernel in that the position does not matter at all. The feature map is useful for interpretation of the sequence features deemed important for the problem at hand.

The Oligomer Distance Histograms Kernel

In 2006, Lingner and Meinicke introduced another feature representation focusing on the relative distances between oligomer pairs instead of positions of the individual oligomers for characterizing sequence similarity. The oligomer distance histogram is a fixed-length feature space representation of any arbitrary-length sequence based on histograms of distances between short oligomers as they occur in the sequence. The distance between a pair of k-mers is defined as the difference in their starting positions in the sequence. For any sequence s, let $D = L - k$, the maximum distance between any two k-mers occurring in the sequence. The distance histogram vector of s corresponding to the k-mer pair (i, j) is given by

$$\mathbf{h}_{ij}(\mathrm{s}) = [h^0_{ij}(\mathrm{s}), h^1_{ij}(\mathrm{s}), \ldots, h^D_{ij}(\mathrm{s})]^{\mathrm{T}} \tag{2.38}$$

where T denotes transpose. For all such k-mer pairs over Σ, the corresponding distance histogram vectors are concatenated together, similar to the oligo kernel, giving a complete feature space representation $\Phi(\mathrm{s})$.

$$\Phi(\mathrm{s}) = [\mathbf{h}^{\mathrm{T}}_{11}(\mathrm{s}), \mathbf{h}^{\mathrm{T}}_{12}(\mathrm{s}), \ldots, \mathbf{h}^{\mathrm{T}}_{MM}(\mathrm{s})]^{\mathrm{T}} \tag{2.39}$$

The ODH kernel value for two sequences s_1 and s_2 is given by the dot product.

$$\mathrm{k}(\mathrm{s}_1, \mathrm{s}_2) = \Phi(\mathrm{s}_1)\Phi(\mathrm{s}_2). \tag{2.40}$$

The set of feature vectors for N training samples is: $\mathbf{X} = [\Phi(\mathrm{s}_1), \ldots, \Phi(\mathrm{s}_N)]$ and the $N \times N$ kernel matrix is given by: $\mathbf{K} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$. It was shown that this kernel accurately detects homology in protein sequences and also identifies important oligomer pairs (Lingner and Meinicke, 2006).

With respect to interpretability and visualization, after the oligo and the ODH kernel in 2004 and 2006, the WD kernel (and its *shift* variant) was reinforced in 2008 with *positional oligomer importance matrices* (POIMs) for visualization of important k-mer features (Sonnenburg et al., 2008).

2.3.4 Tricks for Designing Kernels

For many applications, designing a new kernel adapted to the task at hand may often be a good idea. There are two possible ways of doing it. First, using an existing kernel and performing some permissible operation on it to get the final kernel. Second, designing one using some already known domain specific similarity measure. The latter option is useful when the similarity measure does not yield a valid kernel function by itself. I give examples of both below.

Using an existing valid kernel: Let k_1 and k_2 be kernels over $\mathcal{X} \times \mathcal{X}$, $\mathcal{X} \subseteq \mathbb{R}^n$, $\beta \in \mathbb{R}^+$, then

(i) Sum of two kernels is a kernel

$$k(\mathrm{x}_1, \mathrm{x}_2) = k_1(\mathrm{x}_1, \mathrm{x}_2) + k_m(\mathrm{x}_1, \mathrm{x}_2) \tag{2.41}$$

(ii) Constant $\times$ kernel is a kernel.

$$k(\mathrm{x}_1, \mathrm{x}_2) = \beta k_1(\mathrm{x}_1, \mathrm{x}_2) \tag{2.42}$$

(iii) A linear combination of two or more kernels is also a kernel. This is a combination of the above two operations.

$$k(\mathrm{x}_1, \mathrm{x}_2) = \beta_1 k_1(\mathrm{x}_1, \mathrm{x}_2) + \ldots + \beta_m k_m(\mathrm{x}_1, \mathrm{x}_2) \tag{2.43}$$

This is useful when for the same set of objects $x \in \mathcal{X}$, there is multiple modalities of information available. Then one can freely construct a kernel per modality and combine them in this fashion. Assignment of weights (β values) can be resolved using the so-called *multiple kernel learning* problem (Marius Kloft and Zien, 2011) (see Subsection 2.3.5).

(iv) Product of two kernels is also a kernel

$$k(\mathrm{x}_1, \mathrm{x}_2) = k_1(\mathrm{x}_1, \mathrm{x}_2) k_2(\mathrm{x}_1, \mathrm{x}_2) \tag{2.44}$$

We refer to (Shawe-Taylor and Cristianini, 2004) for a more comprehensive reading.

The empirical kernel map: When a certain domain specific similarity measure exists, it can be used to compute pairwise similarity between all data points. But it is not necessary that this similarity measure yields a valid kernel. In this case, one can use the so-called empirical kernel map (Tsuda, 1999). In it, one first chooses a finite subset of the available data points as *templates*, and computes similarities with the *templates* for all the data points. This results in a fixed, finite-dimensional feature vector for each data point. The empirical kernel is then obtained by computing the dot product between the finite-dimensional feature vectors. With r templates, the empirical kernel value between data points x_1 and x_2 is given as

$$\forall \mathrm{x} \in \mathcal{X}, k(\mathrm{x}_1, \mathrm{x}_2) = \langle \Phi(\mathrm{x}_1), \Phi(\mathrm{x}_2) \rangle = \sum_{i=1}^{r} s(\mathrm{x}_1, t_r) s(\mathrm{x}_2, t_r). \tag{2.45}$$

where $s(\cdot, t_r)$ is the similarity of a data point with the templates. Choosing templates can be an overhead though.

Removal of negative eigenvalues: Alternative to the empirical kernel map approach, one can also take a simpler route of making the pairwise similarity matrix positive definite, if it is not, as follows. Constantly shifting a kernel matrix by subtraction of its minimal eigenvalue makes the kernel matrix positive definite (Roth et al., 2003).

$$\tilde{K} = K - \lambda_{min}(K), \tag{2.46}$$

where $\lambda_{min}(\cdot)$ denotes the minimal eigenvalue of the argument matrix.

2.3.5 Learning In View Of The Multiplicities Of The Real World

Talking about multiplicities of the real world (applications), we already mentioned *multi-class* and *multi-label* classification scenarios (see Subsection 2.3.1). In this subsection, we introduce scenarios of learning from: (a) multiple information modalities for objects; (b) multiple (related) tasks; and (c) multiple instances of objects. We briefly discuss them next.

Handling Multiple Information Modalities

Quite frequently, one comes across a scenario where multiple pieces of information are available for the same set of objects in a task. Using all of them can enable learning better models for the task. Consider an example from the biomedical domain. For a prediction task involving patient data, one could have information on their gene expression values, DNA methylation and various histone modifications. Using only one kind of information of these, say the gene expression profiles, restricts what the model can learn about a disease. One would rather prefer using all different kinds of information simultaneously to understand the disease and its mechanisms as much as possible.

This can be done by designing a separate kernel for each kind of information and putting all of them to use (by 2.43). When the kernel function $k(\cdot, \cdot)$ is a combination of multiple kernels, it is called a *combined* kernel.

$$k_{\text{combined}}(\mathrm{s}_1, \mathrm{s}_2) = \sum_{i=1}^{m} \beta_i k(\mathrm{s}_1, \mathrm{s}_2) \tag{2.47}$$

A simple case of a combined kernel is a uniform combination of the individual so-called subkernels, i.e. assigning a weight of 1.0 to each subkernel. This type of kernel is called the *sum* kernel. Alternatively, some subkernel can be more important than others. Then we would be interested in weighting these subkernels appropriately. The problem of learning these subkernel weights from the data itself is termed *multiple kernel learning* (MKL) (Bach et al., 2004).

Convex combinations are the most popular choice for combining the kernel matrices. They allow for assessing the relative importance of the combined kernels. Optimization constraints are based on the ℓ_p-norm on the kernel weights, where p usually has the value one or two. Using the ℓ_1-norm on the kernel coefficients, i.e. $\|\beta\|_1 = \sum_1^m |\beta_i| = 1$ induces sparsity (Ng, 2004). The ℓ_2-norm, i.e. $\|\beta\|_2 = \sqrt{\beta_1^2 + \ldots + \beta_m^2} = 1$ leads to non-sparse coefficient values β_m. ℓ_2-norm MKL has been more popular and successfully used earlier in biological applications, for example, in combining heterogeneous data sources (Tsuda et al., 2004; Yu et al., 2010).

Handling Multiple Tasks

Building a model that is very close to the *truth*—as in it captures the complex relationships well—is feasible when reasonably abundant data is available to train our learning algorithm. But in many real world tasks, only few training examples may be available. Moreover, obtaining additional training data could be very expensive. This is especially true in computational biology or the biomedical domain due to hefty costs of scientific equipments, and time and effort required for an experiment or medical test. The paucity of training data makes the prediction task at hand even more difficult. As a workaround, data available for any other related task can be utilized. In another scenario, imagine that there are several related tasks that need to be learned simultaneously. A supervised learning algorithm that can combine information from these related tasks is expected to build models capable of achieving better accuracies. *Multitask learning* (MTL) attempts to do this—share information across several related tasks—and achieve improved performance on all the tasks. Usually, one uses domain-specific information to measure the task similarity.

(Evgeniou et al., 2005) introduced how multitask learning can be performed with kernel methods. (Jacob and Vert, 2008) provided the following formulation for sharing of information between tasks with a kernel on tasks.

$$
\begin{aligned}
K_{MTL}((s_A, t_A), (s_B, t_B)) &= \langle \Phi(s_A, t_A), \Phi(s_B, t_B) \rangle && (2.48)\\
&= \langle \Phi_T(t_A), \Phi_S(s_A) \rangle \otimes \langle \Phi_T(t_B), \Phi_S(s_B) \rangle && (2.49)\\
&= \langle \Phi_T(t_A), \Phi_T(t_B) \rangle \times \langle \Phi_S(s_A), \Phi_S(s_B) \rangle && (2.50)\\
&= K_T(t_A, t_B) \cdot K_S(s_A, s_B) && (2.51)
\end{aligned}
$$

where t_A, t_B are tasks, and s_A, s_B are examples corresponding to the two tasks, Φ_T and Φ_S are task- and sequence-specific feature maps, and K_{MTL} is the multitask kernel between the two tuples (s_A, t_A) and (s_B, t_B). This formulation for K_{MTL} as a product of a kernel on tasks and a kernel on examples is very convenient. It was used for predicting peptide–MHC-I binding (Jacob and Vert, 2008). We too use

this formulation for a problem we tackle in this thesis (see Chapter 3). Widmer and Rätsch (2012) present an overview of MTL applications for problems in computational biology.

Handling Multiple Instances

Dietterich et al. first described the *multiple instance* learning (MIL) problem in 1997, where the training examples could have many alternative feature vectors describing them.

Consider the following general example involving a lock smith who has to identify the shape of the key that opens the door to a particular room in an office. Most employees have one key in their keychain which unlocks the said door. For this task, The lock smith has access to the keychains (with keys) of all employees without knowing which key in it opens the door, and has no access to the particular door itself. Thus, here, in order to identify the shape of the key that would open the door, the lock smith examines all keys in every keychain as instances with potential features and infers the characteristic shape of the key that would open the said door (Dietterich et al., 1997).

As a second example, consider the drug-activity prediction problem where the task is to infer the observed activity of a drug molecule. Usually, an input drug molecule binds well to a target binding site on some other larger molecule when one out of a set of conformations is adopted by the input molecule. The other conformations in the set can result in binding, but, only weakly. And, any other conformation that is not a member of this set results in no binding. In this case, the different viable conformations are the multiple favorable instances, one out of them leading to a desired result which is a strong binding event. An instance of this example is the task of major histone compatibility (MHC) class II binding peptide prediction by modeling it as a MIL problem (Pfeifer and Kohlbacher, 2008).

Thus, MIL differs from the typical binary classification scenario which has only one feature vector representing each object. MIL describes a binary classification problem for data that consists of pairs (X_i, y_i), where X_i is a *bag* containing so-called instances $x \in X_i$ and y_i is a binary label ($+1$ or -1). The labels of the instances are not known, but each bag X_i with $y_i = -1$ only has negative instances and each bag X_i with $y_i = +1$ has at least one positive instance. Thus, it can also be said that there is ambiguity in the training examples. The goal of MIL is to learn the best classifier to predict y_j given X_j.

The normalized set kernel, also known as the *multi-instance* kernel, introduced by

Gärtner et al. (2002) for MIL is given as follows:

$$k(X, X') := \frac{k_{\text{set}}(X, X')}{f_{\text{norm}}(X) f_{\text{norm}}(X')} \tag{2.52}$$

where $k_{\text{set}}(X, X') := \sum_{x \in X, x' \in X'} k(x, x')$ and $f_{\text{norm}}(X)$ is a suitable normalization function (Gärtner et al., 2002). One could normalize using either averaging ($f_{\text{norm}}(X) := \#X$) or feature space normalization ($f_{\text{norm}}(X) := \sqrt{k_{\text{set}}(X, X)}$).

In this thesis, we use the multiple instance setting to model the problem of comparing variable-length sequences in the classification scenario. In it, we represent any individual sequence (*bag*) as a collection of its segments (*instances*). Chapter 4 is dedicated for a detailed description of this problem.

3

Genetic Sequence-Based Prediction of Long-range Chromatin Interactions

This chapter describes our work on computational prediction of long-range chromatin interactions using the genetic sequence. This work is published as (Nikumbh and Pfeifer, 2017). While the project idea was conceived by Nico Pfeifer, I designed, implemented and performed the computational experiments with Nico's guidance. All model interpretations and analyses were performed by me and supervised by Nico Pfeifer. Large portions of text in this chapter have been adapted from (Nikumbh and Pfeifer, 2017).

3.1 Introduction

As outlined in the biological background (Section 2.1), it is well known that chromatin, a complex of DNA and proteins, is packed in three-dimensional (3D) space inside the nucleus of the cell in a highly regulated fashion. The spatial conformation of chromosomes is governed by certain principles (Bickmore, 2013; Cope et al., 2010; de Wit and de Laat, 2012). The structure of chromatin depends on the functional state of the cell (viz. normal/diseased) and gene activity among other cellular properties. Thus, a better understanding of 3D chromatin structure and the underlying mechanisms determining this structure helps in gaining an enhanced comprehension of many genomic functions. With the advent of chromosome conformation

capture (3C)-based techniques in the last decade (reviewed in detail in Section 2.1.4), genome-wide analysis of the interaction profiles is now possible (Heidari et al., 2014). Studies have revealed a correlation between long-range chromatin interactions and the functional state of the cell (Zeitz et al., 2013), and more generally, cell-type specificity Heidari et al. (2014). These long-range interactions comprise pairs of loci that are close in 3D space, but not necessarily close in sequence. The spatial co-localization of different chromosomal regions—*cis* as well as *trans*—can be due to a mix of factors. For example, two or more loci can co-localize for specific, direct contact between each other, or they are co-localized due to some nonspecific binding as a result of the packing of the chromatin fibre or having the same subnuclear structure. Specific co-localization may signify functional association (Dekker et al., 2013).

Knowing which loci interact over a long-range and evaluating the effect of such interactions can help us further our understanding of genome regulation and organization. Thus, it is of general interest to be able to predict whether a given pair of loci which are distant on the linear chromosome would interact in 3D space. There exist machine learning-based approaches for predicting such long-range interactions between enhancer and promoters using TF binding and epigenetic information (Roy et al., 2015; Whalen et al., 2016; Yang et al., 2017b). These approaches exclude other genomic regions which lack such additional information from their study. A sequence-based model can improve our understanding of chromatin interactions and the principles governing chromosome folding at the most basic level. It can also be useful to study any genomic region excluded from other studies. Such a model has several potential applications. One is to use the predicted label as additional information for the prediction of boundaries of topologically associating domains (TADs) (Dixon et al., 2012). Another is to assist methods that predict the 3D structure of the chromosome from Hi-C data (Varoquaux et al., 2014).

This chapter describes in detail our work on a computational pipeline for prediction of locus-specific long-range chromosomal interactions. We begin by stating the related work followed by our approach in a nutshell vis-à-vis the related work (Sections 3.2 and 3.3). This is followed by the description of the materials in Section 3.4, wherein we describe the experimental data used and pre-processing performed. The subsequent sections of the chapter present our pipeline, the results and discussion (Sections 3.5, 3.6 and 3.7 respectively).

3.2 Related Work

The last few years have seen an increasing interest in prediction of long-range interactions between promoters and enhancers. Thus, there has been a surge of computational studies for predicting and/or understanding interactions involving promoters

and enhancers. Namely (Roy et al., 2015), (Whalen et al., 2016) and (Yang et al., 2017b).

Per cell line, in contrast to our per-locus models, all of these approaches model the '*all* interactions versus *all* non-interactions' scenario. Among all loci, they use only enhancer-promoter (EP) pairs. Roy et al. (2015) and Whalen et al. (2016) use various functional genomics features to represent each EP pair. They include information on different histone modifications and transcription factors available from many experimental assays, such as ChIP-seq, DNase-seq, and RNA-seq, performed in different cell lines. Roy et al. (2015) use an ensemble of Random Forest-based model and a multitask regression model, and Whalen et al. (2016) use gradient boosted trees for classification. In another work, Yang et al. (2017b) use sequence features at the EP pairs for classifying them as interactions and non-interactions in two ways. In one, sequence features are used via information on the TFBSs for all known TF motifs from databases like JASPAR (Sandelin et al., 2004). In another, they represent each EP pair by embedding it in a lower-dimensional space. These are word embeddings obtained using word2vec (Mikolov et al., 2013a,b). In both the variants, the authors use gradient boosted trees for classification (Yang et al., 2017b). In spite of using genetic sequence information, there are limitations to the benefits of these models. Specifically, in the first case using only known TFBS-motifs is similar in principle to using information from TF ChIP-seq data sets, and in the second case, interpreting the word embedding features is difficult. All of these approaches achieve reasonable prediction performances. Also, there is quite an overlap in the set of features reported by these studies as important for EP interactions (EPIs). In particular, all of them report CTCF, cohesin complex (SMC-RAD21) and zinc-finger proteins as important features characterizing EPIs. Contrasting to the other studies, Whalen et al. (2016) also consider information in the intervening chromatin for every EP pair considered. They report many DNA-binding proteins, and histone marks corresponding to activation and elongation in the intervening windows as features important for distal EPIs.

The rest of the chapter presents our approach for prediction of locus-specific long-range chromatin interactions using the genetic sequence.

3.3 Our Approach in a Nutshell

In this study we built a method based on support vector machines (SVMs) (Boser et al., 1992) to predict which genomic loci potentially interact with a given locus under study based on the genetic sequence of the candidate loci. In a nutshell, we do the following. Given a contact matrix delineating interactions between various genomic loci, we build a predictor for a locus of interest (LoI) from the contact matrix. This

predictor learns the characteristics of the genomic loci that happen to significantly interact with the LoI as against the set of loci that do not. Thus, we build a predictor per locus.

We analyzed 5C contact matrices for three *human* cell lines—GM12878, K562 and HeLa-S3. We demonstrate that the genetic sequence is predictive of the long-range interactions. We developed new visualization methods to enable an intuitive visualization of the sequence features that proved useful for discerning the interaction partners of a LoI from those that do not interact with it. This renders our models to be more than black boxes. Since our models are locus-specific, one can compare the important sequence features characterizing (non-)interactors of the same locus in two cell lines. Additionally, we used these locus-specific models trained on 5C data to independently predict potential chromosome-wide interaction partners for the same LoI. This computational validation is done on high-resolution Hi-C data sets from (Rao et al., 2014).

Since the genetic sequence is only the primary level at which genomic function and organization information is encoded, it is apparent that higher levels of modifications will have the final say towards these chromatin interactions. This is especially true for cell line-specificity. In other words, one would not expect a model using sequence information alone to outshine one that (also) utilizes additional information sources in terms of prediction accuracy. But, a sequence-level model has its advantages as already stated. Thus, we would like to stress upon our two-fold aim in performing this study:

1. Answer the question: To what extent can the genetic sequence alone predict these long-range chromosomal interactions? To this end, we performed computational experiments using our genetic sequence-based approach.

2. Understand the characteristic sequence features underlying such long-range interactions. This is achieved with the help of visualization methods we have newly developed in this work. They aid in interpreting the sequence signals that contribute towards predicting locus-specific interaction partners.

In general, we believe that such an approach using sequence-level information could be useful to study sequence peculiarities among the interaction partners of a particular locus.

3.4 Materials

We use the 5C contact matrices from experiments published by (Sanyal et al., 2012). They probed a collection of regions for two tier-I cell lines (GM12878 and K562) and a

tier-II cell line (HeLa-S3) from ENCODE (The ENCODE Project Consortium, 2012).

The data for each cell line was comprised of two biological replicates. For each replicate, Sanyal et al. (2012) performed the following two pre-processing steps. First, filtering of primers. All 5C primers were expected to perform similarly w.r.t. the trans interactions in the experiment. Any variation observed was considered to be due to experimental factors such as differences in primer and ligation efficiencies. On this premise, outlier primers were filtered as follows. Two mean values were computed, the average 5C signal for each restriction fragment in trans, and the global average of all interchromosomal contact frequencies. The global mean was used to obtain a correction factor per restriction fragment that normalizes its trans signal to those of all restriction fragments. If any of these correction factors was too high or too low, specifically beyond mean$\pm$1.654 standard deviations, that restriction fragment was flagged for removal. Second, normalization of the 5C signal per restriction fragment. The above correction procedure was repeated for the remaining set of restriction fragments. The normalized 5C signal between any pair of restriction fragments was obtained by multiplying three quantities, the correction factors for the pair and the corresponding contact frequency. This two-step procedure corrects for detection biases per restriction fragment (Sanyal et al., 2012). The intra-chromosomally interacting restriction fragments are then tested for significance. In the process, the inverse relationship between contact frequencies and the genomic distance between interacting pairs is accounted for, and peaks are called (cf. Subsection 2.1.4). Sanyal et al. apply a conservative FDR cutoff of 1%. Sanyal et al. term the interactions that are called peaks in both replicates as 'TruePeaks' and those not called peaks in either replicate as 'NonPeaks'. Consequently, in our study, positive examples for any classifier are 'TruePeaks' and negative examples, 'NonPeaks'. We considered different FDR cutoff values (1%, 10% and 15%) and selected an FDR cutoff of 10% for the final model (see Subsection 3.4 below). Table 3.4.1 gives information on the number of 'TruePeaks' (#TP) and the number of 'NonPeaks' (#NP) for the genomic regions included in this study.

We selected ten regions per cell line to evaluate the potential of the DNA sequence to serve as the sole information source in predicting the long-range interactions. For each cell line, these are the 10 regions with the most positive examples available. These are the 'model-defining' *regions* for our study. All genomic coordinates are w.r.t. hg19, GRCh37 assembly. The 'model-defining' loci are among the TSS-containing regions (by GENCODE v7 (Harrow et al., 2012)) and the sets of loci in the positive and negative class for the individual classifiers are restriction fragments

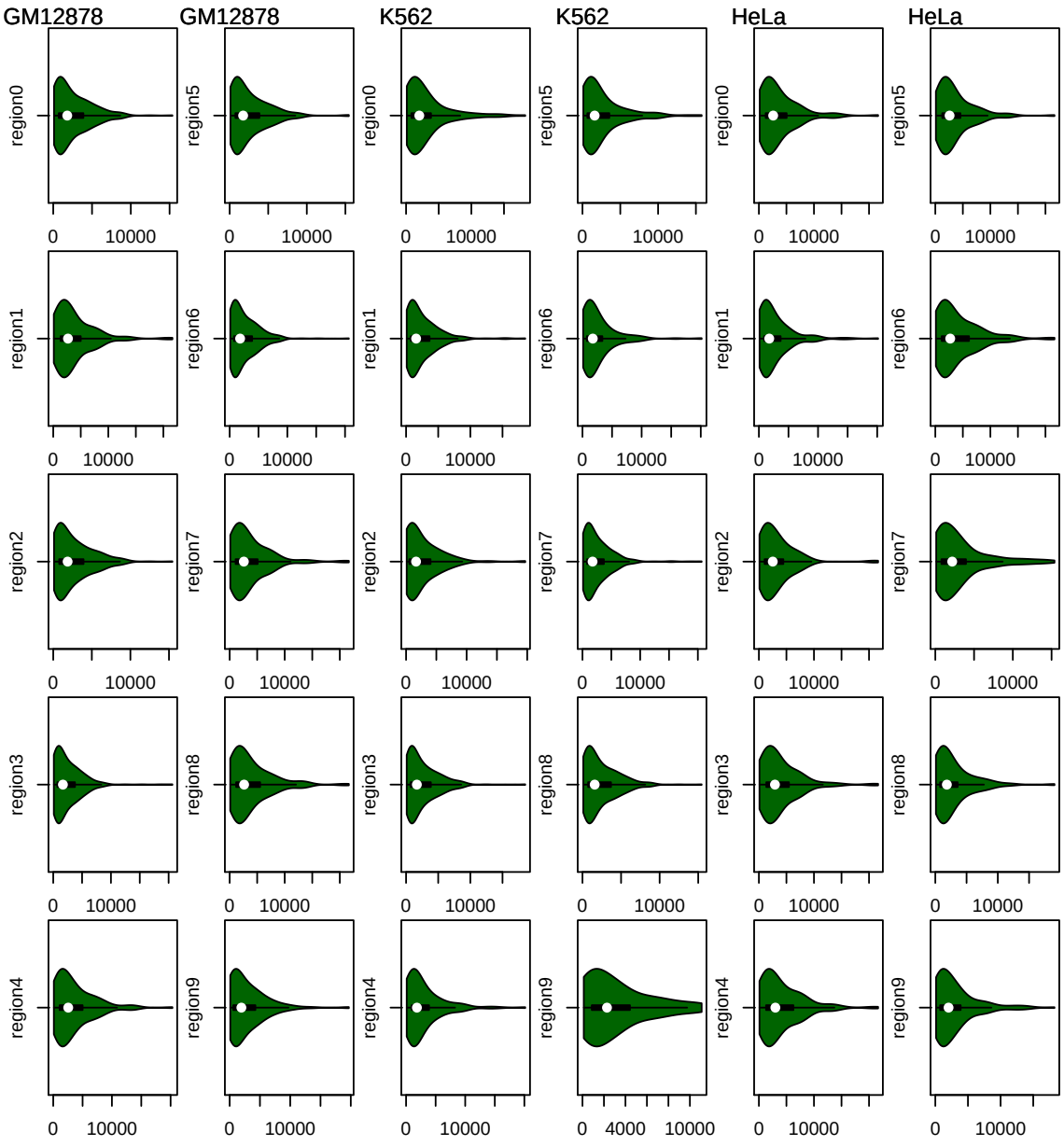


Figure 3.4.1: Lengths of restriction fragments for various *regions* in different cell lines. Their violin plots are arranged in two columns per cell line. Length is measured in terms of the #nucleotides in a sequence.

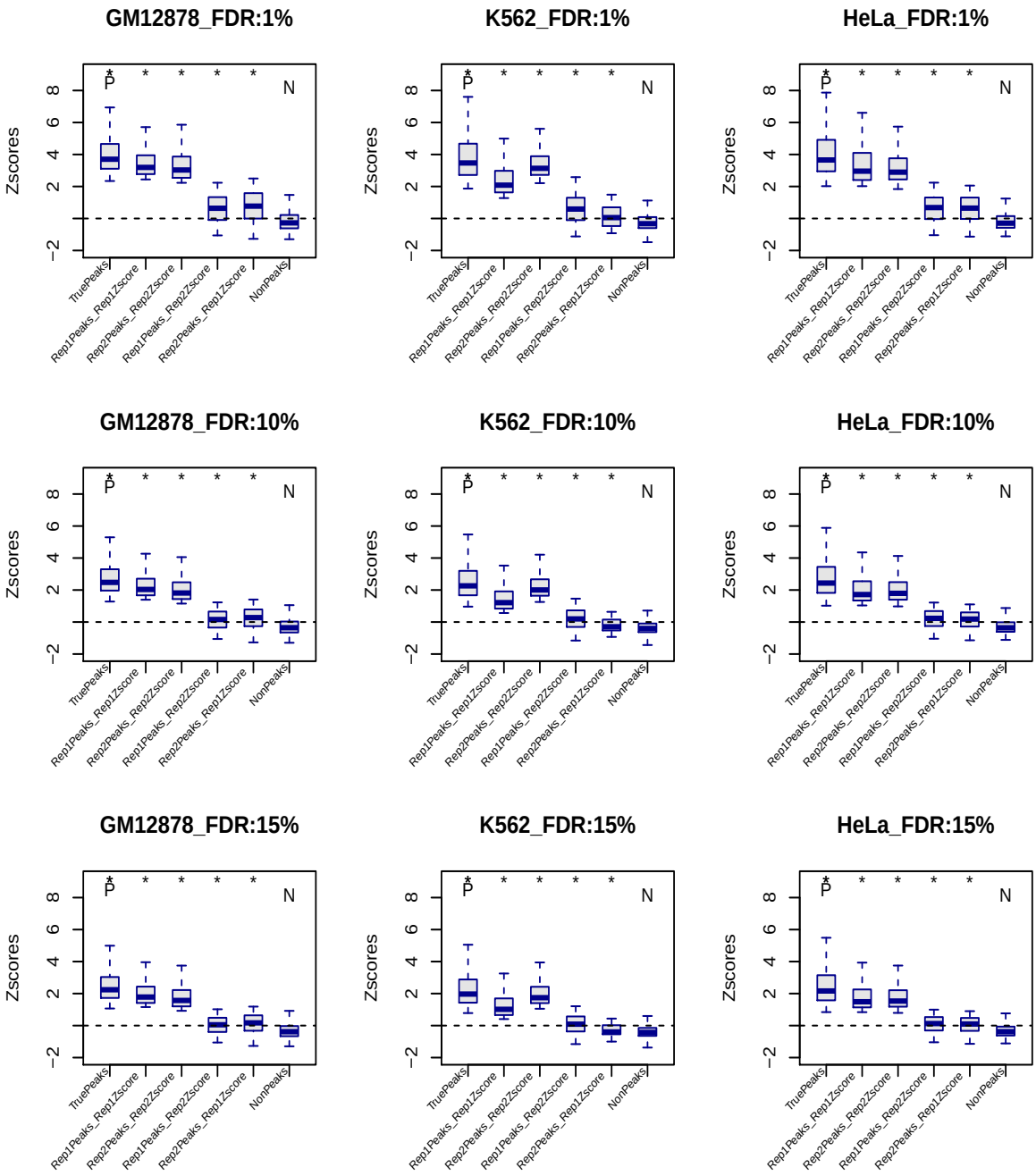


Figure 3.4.2: Z-scores for various cell lines at 1, 10 and 15% FDRs. We have followed the nomenclature from (Sanyal et al., 2012). Rep1Peak_Rep1Zscore: peak in rep1, z-score in rep1 plotted; Rep2Peak_Rep2Zscore: peak in rep2, z-score in rep2 plotted; Rep1Peak_Rep2Zscore: peak in rep1, z-score in rep2 plotted, Rep2Peak_Rep1Zscore: peak in rep2, z-score in rep1 plotted; *TruePeaks*: called peak in both replicates; *NonPeaks*: not called peak in either replicate. We compared each z-score distribution of the different peak classes to the z-score distribution of the NonPeaks with an unpaired Wilcoxon test. Asterisks (*) are shown for significant difference in z-score distribution at significance level 0.05. We did not correct for multiple testing to keep the analysis comparable to (Sanyal et al., 2012). The marks 'P' and 'N' on the box-plots for TruePeaks and NonPeaks denote they constituted the positive- and negative-set of examples respectively in our work.

corresponding to enhancers (also by GENCODE v7 (Harrow et al., 2012)) (Sanyal et al., 2012). All values of #TruePeaks and #NonPeaks in Table 3.4.1 are for FDR 10%.

For the computational validation with high-resolution Hi-C data, we used the data for cell lines GM12878[1] and K562[2] from Rao et al. (2014) deposited at Gene Expression Omnibus (Edgar et al., 2002)

Relaxation of FDR cutoff to enable studying of putative 'bystander' or structural interactions

From a biological point of view, we attempted to take a more broader view and defined an interaction taking into account not just the significant 'looping interactions' but also the possibility of so-called 'bystander' or structural interactions involving the intervening chromatin (Hughes et al., 2014; Sanyal et al., 2012). A conservative FDR cutoff percentage, such as 1%, would include only significant 'looping interactions' as prevalently defined in the community, and a comparatively liberal one would include structural interactions. Thus, in all computational experiments, in order to distinguish significant interactions from non-interactions in the 5C data, we relaxed the FDR cutoff to 10%, instead of 1% as in (Sanyal et al., 2012). In this manner, we traded off between being very conservative and comparatively liberal.

This relaxation still maintained a significantly higher mean z-score of the interactions for TruePeaks in comparison to NonPeaks for all the cell lines, similar to the 1% cutoff case (see Figure 3.4.2). Although 15% FDR also shows a significant difference, it did not provide much benefit in the number of additional TruePeaks per region in comparison to relaxing the FDR from 1% to 10%, consistently across all three cell lines. (i.e., positive examples per classification problem in our study)

3.5 Methods

We use string kernels (introduced in Section 2.3.3) in conjunction with an SVM as a classifier to analyze the genomic loci in this study. Because these loci have highly diverse lengths (see Figure 3.4.1), we could not directly use position-aware string kernels like the oligo kernel (Meinicke et al., 2004) or weighted degree (WD) kernels (Rätsch et al., 2005; Rätsch and Sonnenburg, 2004) for comparing the loci.

[1] GSE63525_GM12878_combined_contact_matrices.tar.gz
[2] GSE63525_K562_intrachromosomal_contact_matrices.tar.gz

Table 3.4.1: Details of the genomic *regions* for the three cell-types (GM12878, K562 and HeLa-S3) for which we built our models. All genomic coordinates are w.r.t. hg19, GRCh37 assembly. These are among the TSS-containing regions (according to the GENCODE v7 (Harrow et al., 2012)) for which reverse 5C primers were designed by (Sanyal et al., 2012). As in (Sanyal et al., 2012), the terms are defined as follows. TruePeaks (TP): called peaks in both replicates; Rep1Peaks (R1P)/Rep2Peaks (R2P): called a peak in either replicate; NonPeaks (NP): not a peak in either replicate. The binary prediction problem for each *region* is TruePeaks vs. NonPeaks (i.e., #TruePeaks = #positive examples and #NonPeaks = #negative examples). All values of #TruePeaks, #Rep1Peaks, #Rep2Peaks and #NonPeaks are for FDR 10%. TCR locus lengths are in base pairs (bp). *R*:regions.

GM12878

R	TCR	length	#TP	#R1P	#R2P	#NP	R	TCR	length	#TP	#R1P	#R2P	#NP
0	chr7:115847372-115857098	9726	63	120	116	226	5	chr7:90224881-90229046	4166	34	51	97	122
1	chr7:115890993-115892266	1273	56	124	97	234	6	chr7:116434729-116454408	19680	33	63	77	292
2	chr7:115861595-115870968	9373	52	88	111	252	7	chr7:90337078-90341001	3924	32	67	43	158
3	chr5:131722317-131724751	2434	39	53	50	91	8	chr22:32162110-32166713	4604	31	74	52	127
4	chr5:131892428-131895867	3439	34	52	57	80	9	chr21:34819525-34821921	2397	30	48	44	201

K562

R	TCR	length	#TP	#R1P	#R2P	#NP	R	TCR	length	#TP	#R1P	#R2P	#NP
0	chr22:32764253-32784733	20480	46	101	62	105	5	chr7:89787744-89795672	7929	35	97	56	118
1	chr22:32920308-32927723	7415	45	101	57	109	6	chrX:153625659-153635385	9727	34	55	43	46
2	chr22:32012966-32043914	30948	42	77	83	104	7	chr22:32170492-32188129	17638	32	83	74	97
3	chr21:35242603-35256847	14244	39	100	52	150	8	chr22:32740683-32750950	10268	32	85	57	112
4	chr7:115847372-115857098	9726	37	125	73	238	9	chr11:5721056-5732713	11658	31	67	40	85

HeLa-S3

R	TCR	length	#TP	#R1P	#R2P	#NP	R	TCR	length	#TP	#R1P	#R2P	#NP
0	chr7:115847372-115857098	9726	98	152	138	207	5	chr7:115861595-115870968	9374	40	77	78	284
1	chr7:116434729-116454408	19679	71	122	137	211	6	chr22:32170492-32188129	17638	40	64	96	102
2	chr22:32920308-32927723	7415	53	72	94	109	7	chr22:32053085-32061138	8054	37	64	80	115
3	chr7:115890993-115892266	1273	50	82	124	243	8	chr22:33262063-33266567	4505	37	87	60	112
4	chr7:89787744-89795672	7928	49	92	85	108	9	chr21:34750664-34761738	11075	37	86	67	147

3.5.1 Pipeline for Predicting Long-range Chromatin Interactions

As discussed in the background chapter, Section 2.1.4, a contact matrix output by any chromatin conformation experiment must be subjected to normalization and extraction of significant contacts. Also, these experiments are usually performed for multiple biological replicates to assess the impact of experimental errors and other variations.

Figure 3.5.1 depicts our approach for predicting long-range chromatin interactions. The normalization and peak-calling procedures that we adopted for analyzing the 5C data used in this study are described in Section 3.4. Once a raw contact matrix has been normalized and the significant interactions have been called, we binarize the contact matrix as follows. Genomic loci (along the rows) not called significant interaction partners of a particular locus (along the columns) in either replicate constitute the negative class (see Figure 3.5.1, cells denoted by filled black boxes). Those called significant in all replicates constitute the positive class (see Figure 3.5.1, cells denoted by filled orange boxes). This leaves a lot of uncalled loci (along the rows). These are denoted by unfilled boxes (Figure 3.5.1). Then, we build classifiers for loci along the column of the matrix (one per locus). We call these loci the 'model-defining' loci. For each individual classifier, the corresponding positive and negative set of examples is built as stated above. Accordingly, from Figure 3.5.1, loci r3, rM and the like are included in the positive class for the classifier corresponding to locus c1. Locus r2 and the like are included in the negative class for it. Note that loci denoted by unfilled boxes, e.g., r1 and r4, are not included in either class and are excluded by the model. Clearly, any locus that belongs to the positive class in one model, may belong to either the positive or negative class in another model or may be even completely excluded.

For each classifier, 80% of the given set of sequences were used for training while 20% were held-out as test sequences. The classifiers are based on an SVM with the ODH kernel (cf. Section 2.3.3). The cost parameter for the SVM, and oligomer length, and maximum distance value for the ODH kernel can be set by the user. Our pipeline also accounts for class-imbalance by proportionately up-weighting the misclassification cost for the minority class (here, positive class) (Elkan, 2001). Recall the misclassification error term $C\sum_{i=1}^{n}\xi_i$ in Eq. (2.14) from Chapter 2. It is replaced by $C^+\sum_{i\in P}\xi_i + C^-\sum_{i\in N}\xi_i$ where C^+ and C^- are costs associated with misclassification errors for examples of the positive (P) and negative (N) class, respectively. Proportionally up-weighting the misclassification cost for the minority class leaves just the cost variable C to be set. Typically, $C^+/C^- = |P|/|N|$ (Ben-Hur and Weston, 2010; Elkan, 2001).

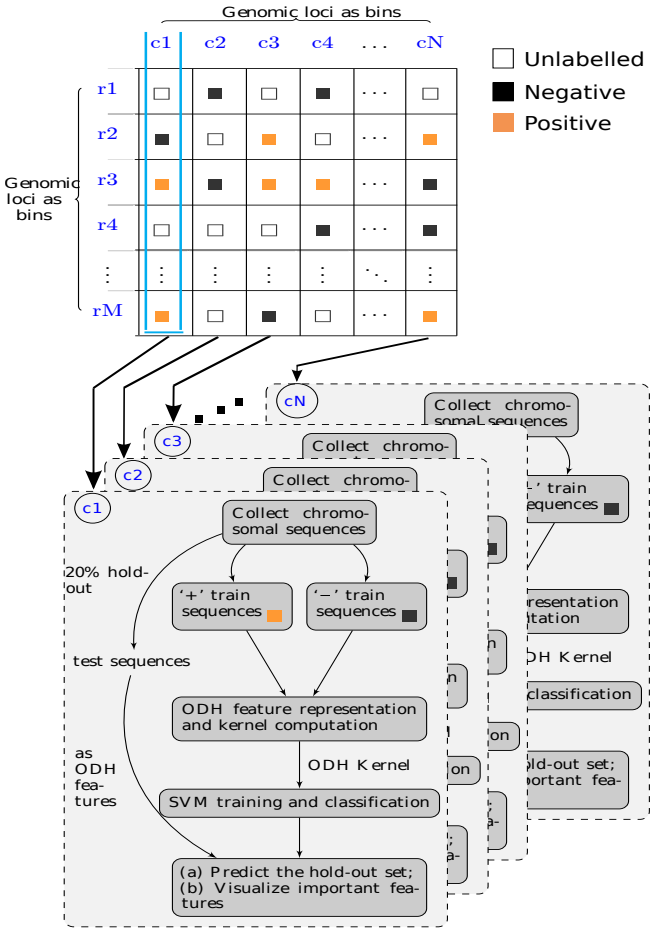


Figure 3.5.1: Pipeline for predicting locus-specific long-range chromatin interactions using the genetic sequence. In the contact matrix, cells denoted by filled orange boxes (■) correspond to loci that are called significantly interacting with the LoI in all replicates of any experiment profiling chromatin interactions. This constitutes the positive set of sequences for the corresponding classifier. Those denoted by filled black boxes (■) correspond to loci that are not called significantly interacting in any of the replicates. This constitutes the negative set of sequences for the corresponding classifier. This leaves those loci which are called significantly interacting in at least one, but not in all of the replicates. They are visualized by unfilled boxes (□) and are not used by the classifier. The genomic loci along the columns of the contact matrix (c1, c2, c3,...,cN) are the LoI for which we build locus-specific classifiers.

Experimental Setup

For each model, the cost parameter for SVM is varied in the range $10^{-3}, 10^{-2}, \ldots, 10^{3}$. We performed experiments with K-mer values 3 and 5 and the maximum distance between K-mers as 100. The ODH kernel has no other hyperparameters to be tuned. Intuitively, a model built with K-mer value 5 encodes more specificity while the K-mer value 3 maintains relative generality. We perform a 5-fold nested cross-validation to select the best performing cost-value for the SVM while the ODH feature representation parameters are fixed.

3.5.2 New Visualization Techniques

We developed two new visualization techniques suitable for our models using the ODH representation. The aim is to enable better interpretation of the sequence signals that contributed towards prediction of locus-specific interaction partners. The first of these two techniques is 'Absolute Max Per Distance' and, the second is 'Position-Wise Weight Matrix (PWWM)-based TopN'.

Absolute Max Per Distance (AMPD) visualizations

We introduced the ODH representation in Section 2.3.3. Recall that the dimensionality of the ODH feature vector for a given alphabet $\sum$ using oligomer length K and distances up to D is $[(|\sum|^K)^2 \times (D+1)]$. For the DNA sequence alphabet, and oligomer length 3 and 5, this gives 413,696 and 105,906,176 dimensions, respectively. The SVM weight vector for a model has the same dimensionality as the feature vector (cf. Subsection 2.3.1). This implies, in this scenario that its dimensionality is the same as the dimensionality of the ODH feature vector. Thus, each entry of the SVM weight vector is the coefficient assigned to a K-mer pair separated by a distance $d \in [0, 1, \ldots, D]$. For each of our locus-specific models, the 5-fold outer cross validation gives 5 different SVM weight vectors. These five individual weight vectors are averaged to obtain one representative weight vector for a per-locus model. From this averaged weight vector, we note two K-mer pairs per distance value, one that was assigned the most positive coefficient and the other, most negative. A positive coefficient means the d-separated K-mer pair is an important feature among the positive sequences, while a negative coefficient means it is an important feature to classify the sequence as negative. All such selected K-mers at the various distance values are visualized to provide a distance-centric view of the important features. Such a visualization for *region* 9 of cell line GM12878 is shown in Figure 3.6.3. We call these visualizations 'Absolute Max Per Distance' (AMPD) visualizations. For better readability, the K-mer pairs at even distance values are arranged in the outer column and those at odd distance values in the inner column. Figures 3.6.3, 3.6.5, and 3.6.8

show examples of 'AMPD' visualization for different regions across the three cell lines GM12878, K562 and HeLa-S3. In particular, these are for *regions* 9, 7 and 6 from among those given in Table 3.4.1.

Position-Wise Weight Matrix (PWWM)-based 'TopN' visualizations

Independently, the entries of the averaged weight vector are sorted in descending order and then thresholded to reveal the top 25 scoring entries. Figure 3.6.4 visualizes only those selected top-25 K-mer pairs. Here, the $(D+1)$ distances are arranged radially. Each spoke gives the magnitude of the highest-scoring K-mer pair at the corresponding distance. If the magnitude does not cross the threshold value, that spoke is plotted in gray. If it does, it is plotted in 'blue' when it has a positive contribution (see Figure 3.6.4), and in 'red' when it has a negative contribution (see Figure 3.6.9). We call these visualizations 'Top25', or more generally, 'TopN' visualizations where one can choose a suitable value for 'N'. Since there can be more than one entry at the same distance d among the top-N, this leads to sequence logo-like representations. At any distance d, all motifs that exceeded the threshold are collected along with their weight magnitudes and stacked one over the other to finally represent them with a consensus motif. This consensus motif is obtained by constructing a 'Position-Wise Weight Matrix' (PWWM) of dimension $(|\sum| \times 2K)$. It represents the nucleotides appearing at each position from 1 to $2K$ along with their relative contribution to the weight vector. A dummy example illustrating this is shown in Table 3.5.1. This PWWM is computed as follows. For position $p \in \{1, \ldots, 2K\}$, the

Table 3.5.1: A dummy PWWM for selected 3-mer pairs at certain distance d. $|w_1|$, $|w_2|$, and $|w_3|$ are magnitudes of the weights for the example 3-mer pairs. `` `A' ``, `` `C' ``, `` `G' `` and `` `T' `` are the rows corresponding to the nucleotides. Position, $p \in \{1, \ldots, 6\}$. Each cell is divided by $W = (|w_1| + |w_2| + |w_3|)$.

	3-mer pairs					
$\|w_1\|$	A A A			G A A		
$\|w_2\|$	G A A			A G A		
$\|w_3\|$	A A G			A A A		
`A`	$\frac{1}{W}(\|w_1\|+\|w_3\|)$	$\frac{1}{W}(\|w_1\|+\|w_2\|+\|w_3\|)$	$\frac{1}{W}(\|w_1\|+\|w_2\|)$	$\frac{1}{W}(\|w_2\|+\|w_3\|)$	$\frac{1}{W}(\|w_1\|+\|w_3\|)$	$\frac{1}{W}(\|w_1\|+\|w_2\|+\|w_3\|)$
`C`	0	0	0	0	0	0
`G`	$\frac{1}{W}(\|w_2\|)$	0	$\frac{1}{W}(\|w_3\|)$	$\frac{1}{W}(\|w_1\|)$	$\frac{1}{W}(\|w_2\|)$	0
`T`	0	0	0	0	0	0
p	1	2	3	4	5	6

matrix cell (`` `A'/`C'/`G'/`T' ``, p) is populated with the sum of the weight contribution of those motifs in which the given nucleotide is present at position p. The matrix is then normalized for the column entries to sum up to 1. The resulting consensus motifs are represented as sequence logos (Schneider and Stephens, 1990). Examples of 'Top25' visualizations are shown in Figures 3.6.4, 3.6.6, 3.6.7, 3.6.9, and 3.6.10.

3.5.3 Implementation and Availability of Software

As compared to the protein sequences used in (Lingner and Meinicke, 2006), the ODH feature vectors for the DNA sequences used in this study are relatively dense. This is because the DNA alphabet is just 4 characters, and many of these sequences are very long (cf. Figure 3.4.1). To tackle this scenario, we adapted the MATLAB[3] code provided by the authors for ODH feature representation and kernel computation (Lingner and Meinicke, 2006). We used LIBSVM's SVM implementation (Chang and Lin, 2011). Our complete pipeline with all the wrappers, and the additional MTL implementation is written in MATLAB. Our pipeline is named 'Samarth', and made available for download at the supplemental website `http://bioinf.mpi-inf.mpg.de/publications/samarth/` as free software for academic use, with no warranty or liability. The 'AMPD' and the 'TopN' visualizations were created using custom MATLAB and R (R Core Team, 2013) scripts respectively.

3.6 Results

We used the pipeline described above for predicting the long-range interaction partners of each of the ten model-defining loci per cell line. In this section we describe the results of these computational experiments.

3.6.1 Prediction of Long-Range Chromatin Interactions is Possible from the Sequence Alone Using Non-Linear SVMs

Table 3.6.1 shows the test AUC (area under the ROC curve) values for all regions in all the three cell lines resulting from our 5-fold nested cross validation. Furthermore, our pipeline is also capable of handling imbalances in the data. For all the model-defining regions in our computational experiments, the positive class is in minority (see columns #TP and #NP reproduced from Table 3.4.1). We report performances with data imbalance handled (see Section 3.5.1). The average test AUC values for the individual tasks are as follows.

Oligomer length 3 {GM12878, K562, HeLa-S3}: {0.7251, 0.7534, 0.6782};

Oligomer length 5 {GM12878, K562, HeLa-S3}: {0.7443, 0.7716, 0.7153}.

Box plots of all the test performances for different regions in all three cell lines are given in Figure 3.6.1, and Figure 3.6.2. Owing to small sample sizes, the model test

[3]MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.

Table 3.6.1: Locus information for regions and prediction performances. #TruePeaks (#TP) and #NonPeaks (#NP) for all the studied genomic regions (column 'R') for the three cell lines (GM12878, K562 and HeLa-S3). Columns marked 'A', 'B', 'C' and 'D' show the mean test AUC values with oligomer length 3 and 5 respectively for two settings: Individual tasks ('A' and 'B') and Multiple tasks ('C' and 'D').

GM12878

R	TCR	#TP	#NP	Test AUC				R	TCR	#TP	#NP	Test AUC			
				A	B	C	D					A	B	C	D
0	chr7:115847372-115857098	63	226	0.7417	0.7538	0.8979	0.9042	5	chr7:90224881-90229046	34	122	0.8078	0.8307	0.9221	0.9118
1	chr7:115890993-115892266	56	234	0.7141	0.7341	0.8876	0.8960	6	chr7:116434729-116454408	33	292	0.7785	0.7787	0.7308	0.7036
2	chr7:115861595-115870968	52	252	0.7346	0.7763	0.9152	0.9376	7	chr7:90337078-90341001	32	158	0.8163	0.8275	0.9286	0.9324
3	chr5:131722317-131724751	39	91	0.6122	0.6547	0.8666	0.8286	8	chr22:32162110-32166713	31	127	0.7779	0.7832	0.7789	0.7738
4	chr5:131892428-131895867	34	80	0.5971	0.6343	0.8889	0.8543	9	chr21:34819525-34821921	30	201	0.6704	0.6694	0.7157	0.6901

K562

R	TCR	#TP	#NP	Test AUC				R	TCR	#TP	#NP	Test AUC			
				A	B	C	D					A	B	C	D
0	chr22:32764253-32784733	46	105	0.8163	0.8121	0.9308	0.9382	5	chr7:89787744-89795672	35	118	0.8546	0.8648	0.8566	0.8727
1	chr22:32920308-32927723	45	109	0.6808	0.7242	0.7744	0.7972	6	chrX:153625659-153635385	34	46	0.8501	0.8495	0.8044	0.8184
2	chr22:32012966-32043914	42	104	0.7145	0.7324	0.8378	0.8599	7	chr22:32170492-32188129	32	97	0.7456	0.7146	0.8003	0.8228
3	chr21:35242603-35256847	39	150	0.7321	0.725	0.7251	0.7407	8	chr22:32740683-32750950	32	112	0.7167	0.7582	0.8836	0.9166
4	chr7:115847372-115857098	37	238	0.7521	0.7756	0.7765	0.7908	9	chr11:5721056-5732713	31	85	0.671	0.76	0.7345	0.7545

HeLa-S3

R	TCR	#TP	#NP	Test AUC				R	TCR	#TP	#NP	Test AUC			
				A	B	C	D					A	B	C	D
0	chr7:115847372-115857098	98	207	0.6914	0.7111	0.8007	0.8228	5	chr7:115861595-115870968	40	284	0.6624	0.732	0.8964	0.9114
1	chr7:116434729-116454408	71	211	0.73	0.7674	0.8573	0.8738	6	chr22:32170492-32188129	40	102	0.677	0.755	0.8245	0.8590
2	chr22:32920308-32927723	53	109	0.644	0.6369	0.7338	0.7091	7	chr22:32053085-32061138	37	115	0.6018	0.6420	0.7886	0.7991
3	chr7:115890993-115892266	50	243	0.6817	0.7225	0.907	0.9162	8	chr22:33262063-33266567	37	112	0.5634	0.6564	0.8449	0.8491
4	chr7:89787744-89795672	49	108	0.8108	0.8007	0.8005	0.8084	9	chr21:34750664-34761738	37	147	0.7194	0.7294	0.7053	0.7273

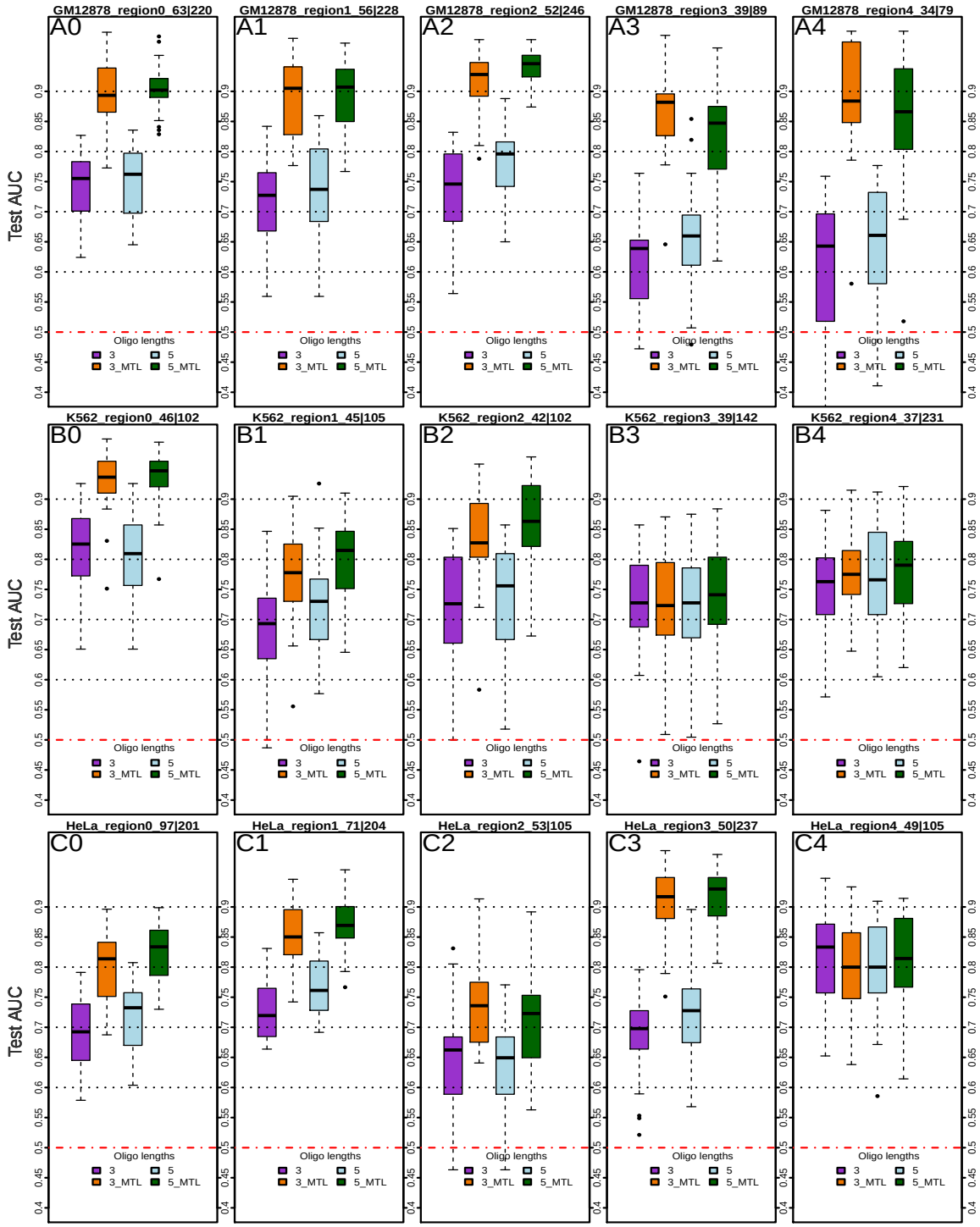


Figure 3.6.1: Box-plots of SVC performances for cell lines GM12878, K562 and Hela-S3. Five *regions* (numbered 'A0-A4', 'B0-B4' and 'C0-C4' for GM12878, K562 and Hela-S3 respectively) out of 10 are shown. Individual tasks setting, oligomer lengths $= \{3, 5\}$ in purple and light blue respectively. MTL with 10 tasks, oligomer lengths $= \{3, 5\}$ in orange and green. Distances between K-mer pairs upto $D = 100$. Box-plots for the other five regions among the 10 are given in Figure 3.6.2.

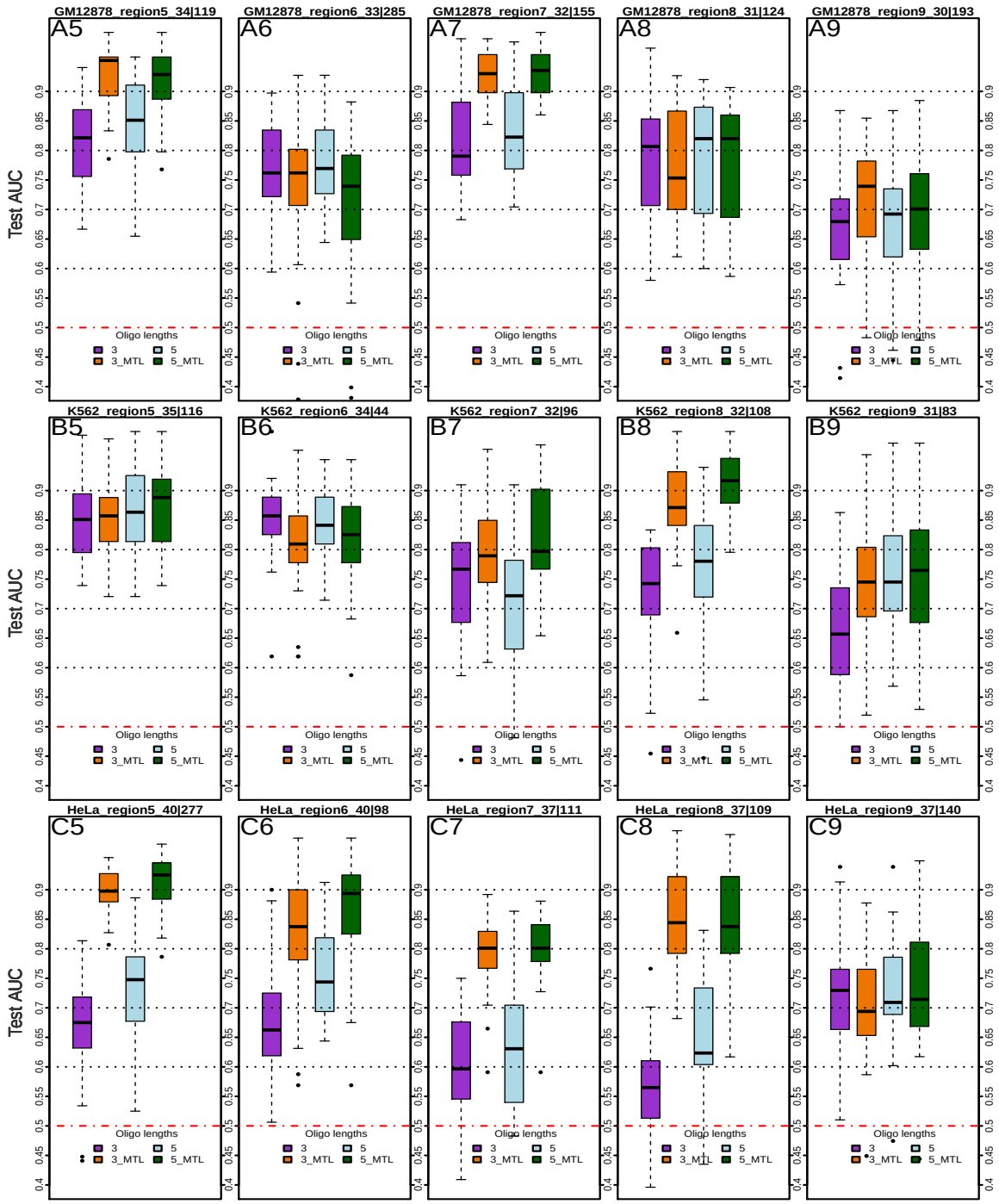


Figure 3.6.2: Box-plots of SVC performances for further five regions in cell lines GM12878, K562 and Hela-S3. Five *regions* (numbered 'A5-A9', 'B5-B9' and 'C5-C9' for GM12878, K562 and Hela-S3 respectively) out of 10 are shown. Individual tasks setting, oligomer lengths $= \{3, 5\}$ in purple and light blue respectively. MTL with 10 tasks, oligomer lengths $= \{3, 5\}$ in orange and green. Distances between K-mer pairs upto $D = 100$. Box-plots for the other five regions among the 10 are given in Figure 3.6.1.

performances mostly show high variance (Figures 3.6.1 and 3.6.2) .

3.6.2 Tandem Repeat Motifs are an Important Feature Distinguishing Interaction Partners

Figure 3.6.3 and Figure 3.6.4 show our new visualizations of the set of K-mer pairs that influenced the prediction the most. Recall that, in both these visualizations, any K-mer pair is represented as an adjoined $\{2K\}$-mer separated by '|', e.g., 3-mer pairs as 6-mers, and we loosely address these K-mer pairs as 'motifs', although they are not contiguous. Figure 3.6.3 shows the 'Absolute Max Per Distance' (AMPD) visualization for *region* 9 in cell line GM12878. The AMPD visualization shows, at each distance value (plotted on vertical axis), the K-mer pair that contributes the most in predicting a locus as positive and negative. The weights of these K-mer pairs (fetched from the SVM weight vector) are plotted on the horizontal axis. Figure 3.6.3 top panel shows 6-mers consisting of the 3-mer pairs separated by '|', and in the bottom panel are the adjoined 5-mers. Owing to the high dimensionality of the 5-mer case, we observe that the magnitudes of the weights quickly shrink in this case. We filter this information further and visualize only the top few high-scoring features in the 'Top25' visualization in Figure 3.6.4.

Across various regions, among many motifs, short tandem repeat sequences, especially di- and trinucleotide repeats, are prominently observed at various distances. Our 'AMPD' visualizations facilitate spotting of patterns spread over distances while the 'TopN' visualizations can help spot possibly hidden shorter K-mer signals. Refer to Figure 3.6.3 for the following discussion. The dinucleotide '`GT`' being repeated is observed to have a maximal contribution for distances up to 26 and 34 in the 3-mers and the 5-mers case. In both the cases, the model identifies it as an important feature towards predicting a locus as a potential interacting partner of *region* 9 in GM12878. Additionally, the 3-mer case shows patterns prominently containing more '`T`'s separated by ∼30-60 bp as a negatively contributing feature. They are absent from the set of positive contributors. Interestingly, we observe various such patterns for different regions across cell lines.

Our literature search revealed some relevant studies on tandem repeat sequences and their potential biological roles. A 1990 review by Vogt provides an extensive account of the potential functions of tandem repeat sequences in the human genome (Vogt, 1990). It includes an exhaustive discussion of the various repeat sequences, viz. mono-, di-, tri-, tetranucleotides and beyond. It also postulates their association with a multitude of nuclear proteins that help them assume specific chromosomal structures. The author terms this ability of the tandem sequence repeat

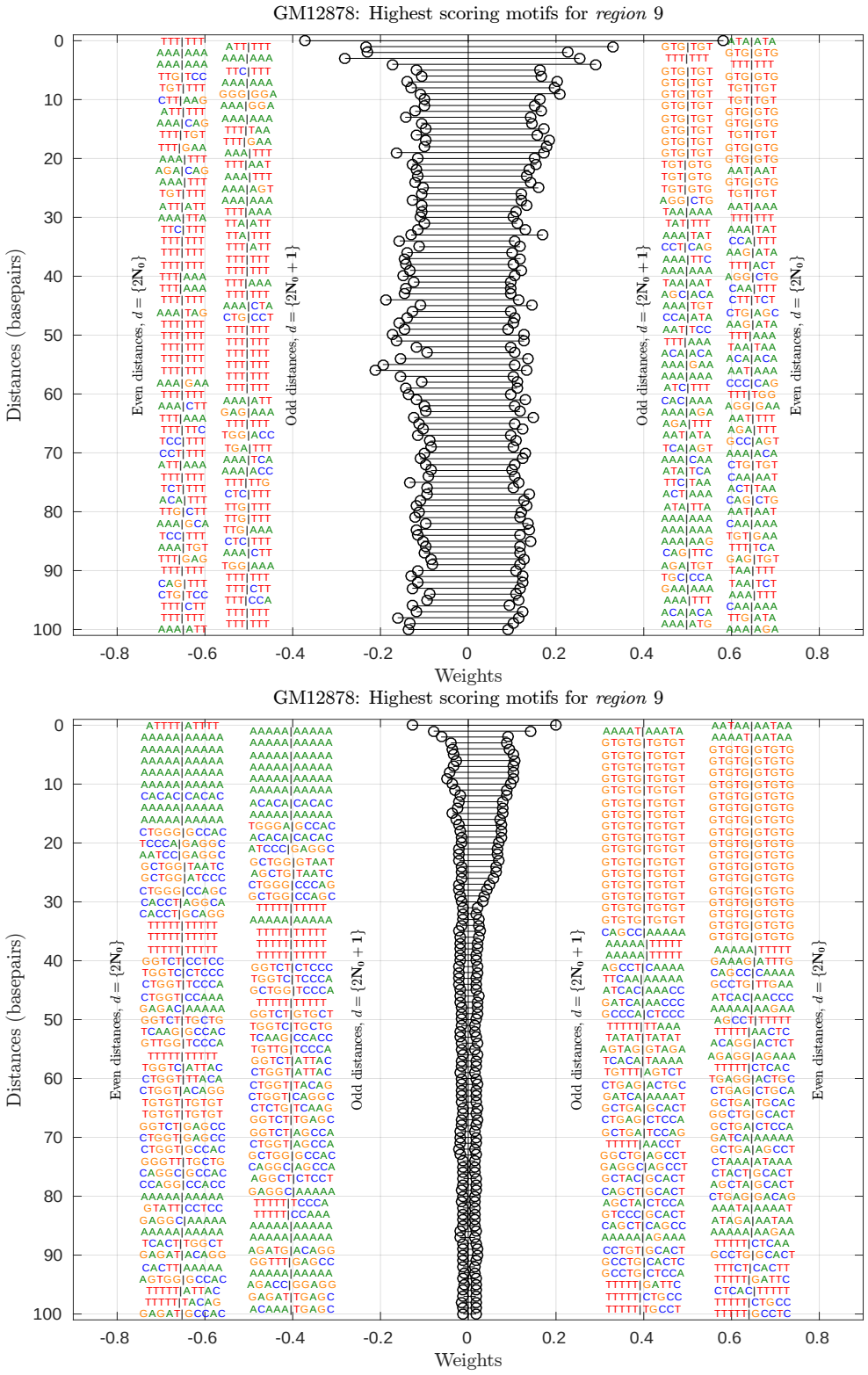


Figure 3.6.3: 'AMPD' visualization of the informative K-mer pairs from the predictor for ***region* 9 in GM12878** (Refer Table 3.6.1 for *region* details). Top: At distances in $\{0, \ldots, 100\}$, the 3-mer pair that maximally contributes towards positive and negative classification of a given locus is shown. Weights are shown on the horizontal axis, distances on the vertical axis. Below: 'AMPD' visualization for the 5-mer case.

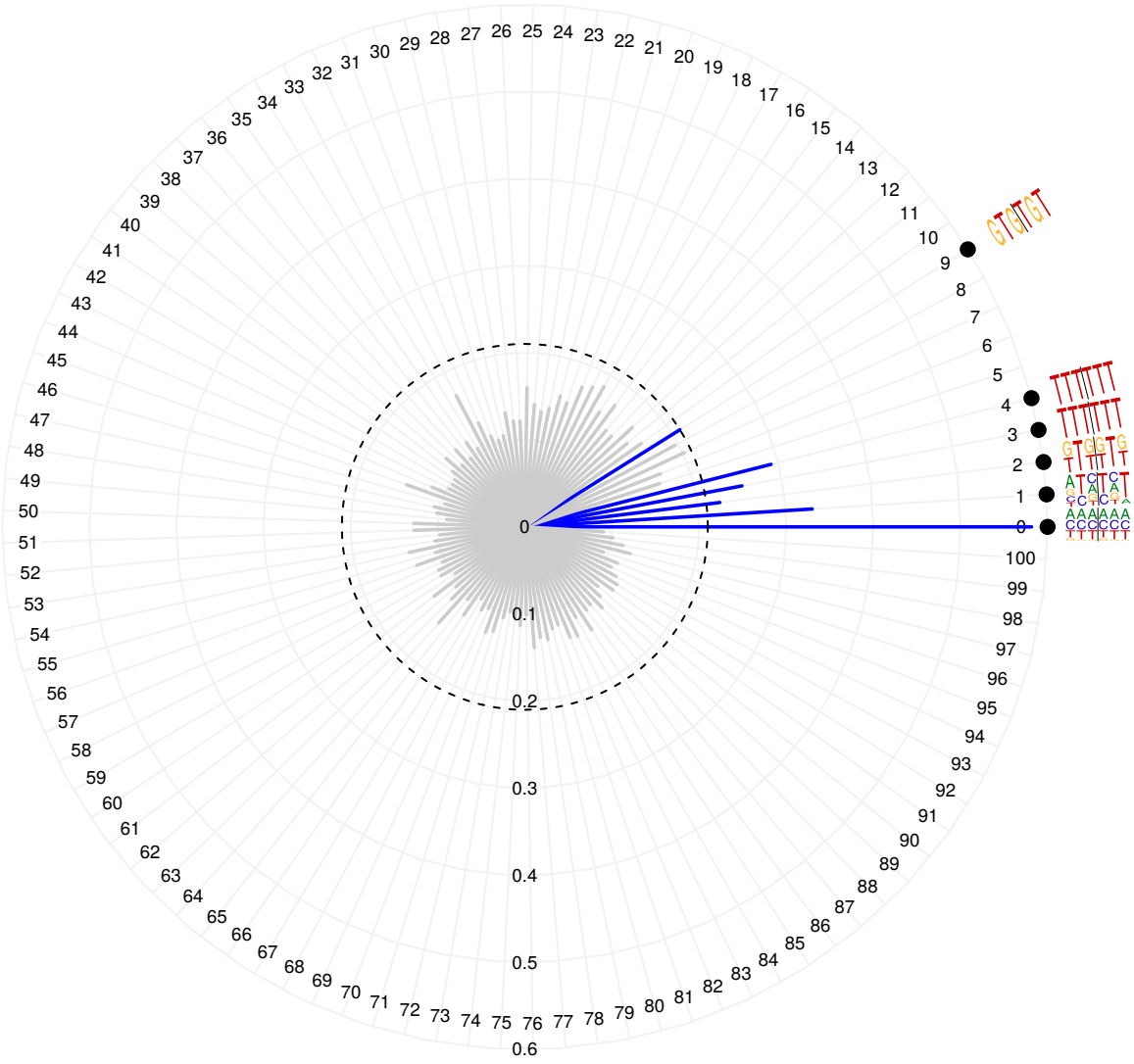


Figure 3.6.4: 'Top25' visualization of the informative 3-mer pairs separated by various distances and their magnitudes from the predictor for ***region* 7 in GM12878** (Refer Table 3.4.1 for *region* details). Top-25 3-mer pairs, with weight magnitudes higher than the threshold (dashed inner circle), for the positive class (blue). The dashed inner circle is the threshold to select the top-25 entries of the averaged SVM weight vector.

blocks to render locus-specific higher order structure, and, to play a role in organization as the 'chromatin folding code' (Vogt, 1990). In the review (Vogt, 1990), the author also points to a specific case of the dinucleotide `TG' as a simple repeating block, which has already been shown to have an enhancer function *in vitro* in as early as 1984 (Hamada et al., 1984). More recently, a 2014 study by Yáñez-Cuna et al. identified dinucleotide repeat motifs (DRMs) as general features that can render a nonfunctional sequence into an active enhancer element. Another comprehensive study of 2014 suggests a potential role of simple sequence repeats (SSRs), including their repeat lengths, in genome regulation and organization (Ramamoorthy et al., 2014). These sequence repeats are broadly termed as variable number tandem repeats (VNTRs). VNTRs have already been implicated in many complex neurological disorders (e.g., Huntington disease (Malaspina et al., 2001)), and are generally known to be polymorphic (Brookes, 2013).

There are more studies that bolster this hypothesis concerning the general role of short tandem repeats. For example, (Gymrek et al., 2016) finds significant expression simple tandem repeats (eSTRs) to be enriched in clinically relevant phenotypes, and contributing to the variations in gene expression. Specific to the three dimensional architecture of the chromatin, (Mourad and Cuvier, 2016) suggests that the repeat regions also play a role at the borders of TADs. X-chromosome inactivation (XCI), the process of inactivating a copy of the X-chromosome in a female mammal, has been of particular interest to the community. It has been studied with regards to the three-dimensional organization. The work by Rao et al. (2014) that produced kilobase-resolution Hi-C data already highlights a specific case of the inactive X (Xi) chromosome containing large loops anchored at CTCF-binding repeats. Studies also report on the role of the macrosatellite repeat DXZ4 in Xi chromosome using Hi-C (Darrow et al., 2016; Giorgetti et al., 2016). Darrow et al. (2016) report that in the Xi chromosome many superloops[4] are often anchored at the DXZ4 repeats, and that there are two superdomains[5] formed whose separating boundary lies at the macrosatellite DXZ4[6]. The authors specifically perform deletion of DXZ4 and observe that this leads to disruption of the superloops and superdomains, thus rendering the macrosatellite DXZ4 essential for XCI. The work of Giorgetti et al. (2016) that studied the role of Xist in Xi chromosome organization also similarly reports loss of superdomains (or mega-domains, as they termed it) upon deletion of DXZ4.

We wish to note that among recent work discussed above, studies from 2016 were published while our manuscript, (Nikumbh and Pfeifer, 2017), was in review.

[4]superloops: extremely large loops within superdomains
[5]superdomains: contact domains unusually larger than TADs
[6]superloops and superdomains, both span several dozen megabases (Darrow et al., 2016)

3.6.3 Identifying Cell-Line Specific Characteristic Signals

An advantage of studying locus-specific interactions at the sequence-level is realized when our models can reveal the characteristic signals among interaction partners of the same locus in two different cell lines. Consider the locus `chr22:32170492-32188129` which is, both, *region* 6 and *region* 7 among our models for HeLa-S3 and K562 respectively (see Table 3.6.1). Refer to their 'AMPD' visualizations with 3-mers and 5-mers in Figures 3.6.8 and 3.6.5 respectively. For K562, the '`CA`' dinucleotide repeat sequence stretch of length ∼20 markedly denotes a non-interacting partner while this same repeat sequence seems to be interrupted with a short stretch of '`T`'s in HeLa-S3. Also, another repeat sequence, '`AGA`', is notable beyond distance values 50 among the non-interacting partners for this locus in K562 as compared to HeLa-S3, where it is only intermittently observed. Similarly, these signals are also picked up by our 5-mer models. The corresponding 'Top25' visualizations for these regions are given in Figures 3.6.6, 3.6.7, 3.6.9, and 3.6.10.

3.6.4 Multitask Learning (MTL) Helps Mitigate Issue of Having Too Few Interacting Partners per Locus

Recall from Section 2.3.5 that any individual learning problem can be termed as a 'task', or in other words, is a single task. Thus, each locus-specific prediction problem in our scenario is termed as a single task. We also stated in Section 3.4 that the choice of the ten regions for this study was made based on the number of positive samples available for each task. The number of positive samples decreases from *region* 0 through 9 in each cell line (Table 3.6.1). Such small samples sizes affect the learning ability of machine learning methods including the SVMs, and often lead to a loss of generalizability. These small sample sizes in the single-task setting can be mitigated with the help of the so-called 'multitask' setting (see Section 2.3.5 for an introduction). In order to evaluate the efficacy of MTL for this problem, we used the available 10 individual tasks. Here, to compute the task similarity, we used the 'model-defining' locus (the LoI) information, i.e., the genetic sequence at this locus. The locus sequence of every 'model-defining' region was also represented as an ODH feature vector using the K-mer values 3 and 5, separately, and maximum distance 100. The similarities between these regions (in turn, the *tasks*) were given by dot products (ODH kernel). For single-task models that used oligomer length 3 and 5 representations for the input sample sequences, we used the corresponding

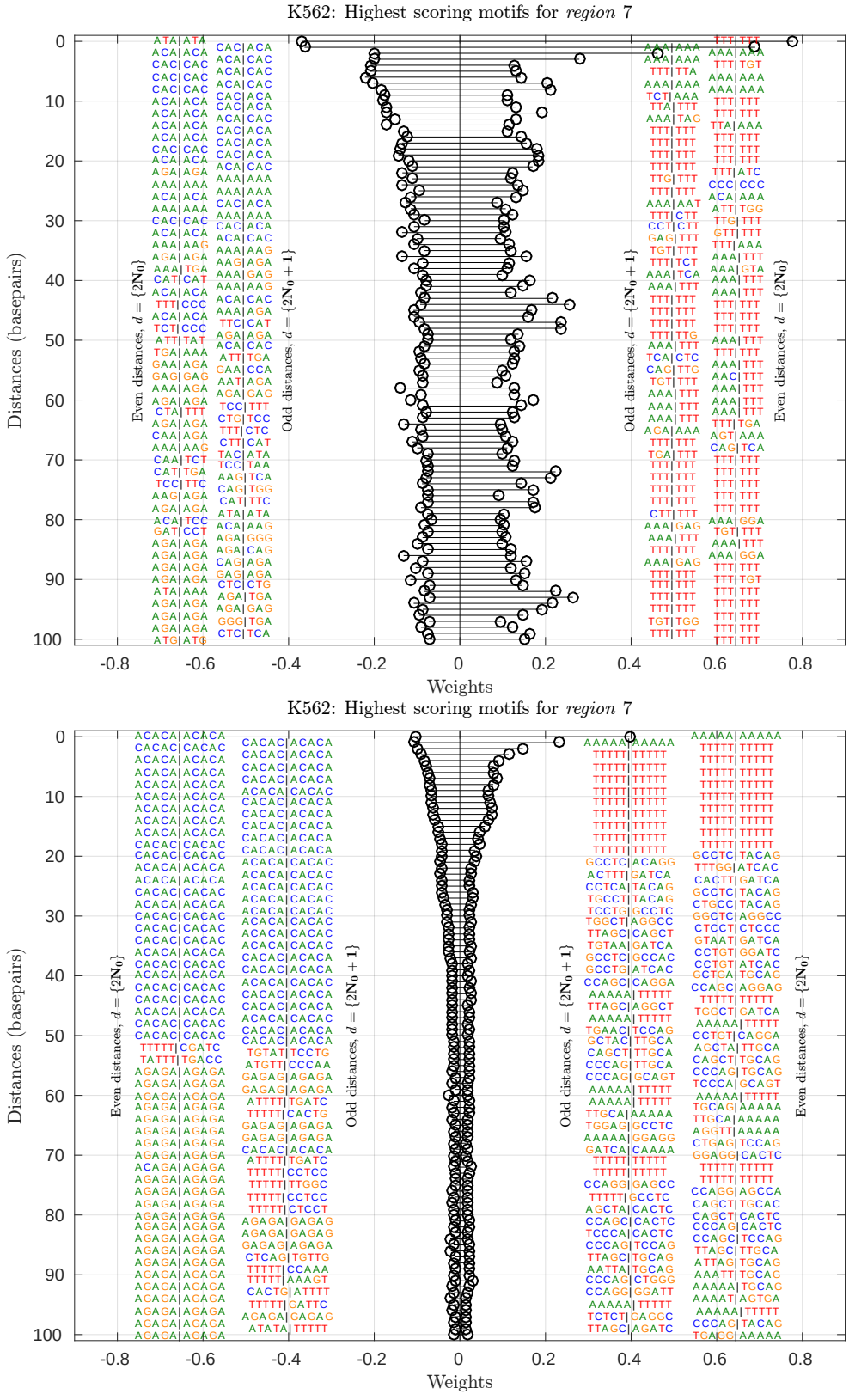


Figure 3.6.5: 'AMPD' visualization of the informative K-mer pairs from the classifier for ***region* 7 in K562** (Refer Table 3.4.1 for *region* details). Top panel: At distances in (0-100), the 3-mer pair that maximally contributes towards positive and negative classification of a given locus is shown. Weights are shown on the horizontal axis, distances on the vertical axis. Bottom panel: Visualization of the 5-mer case.

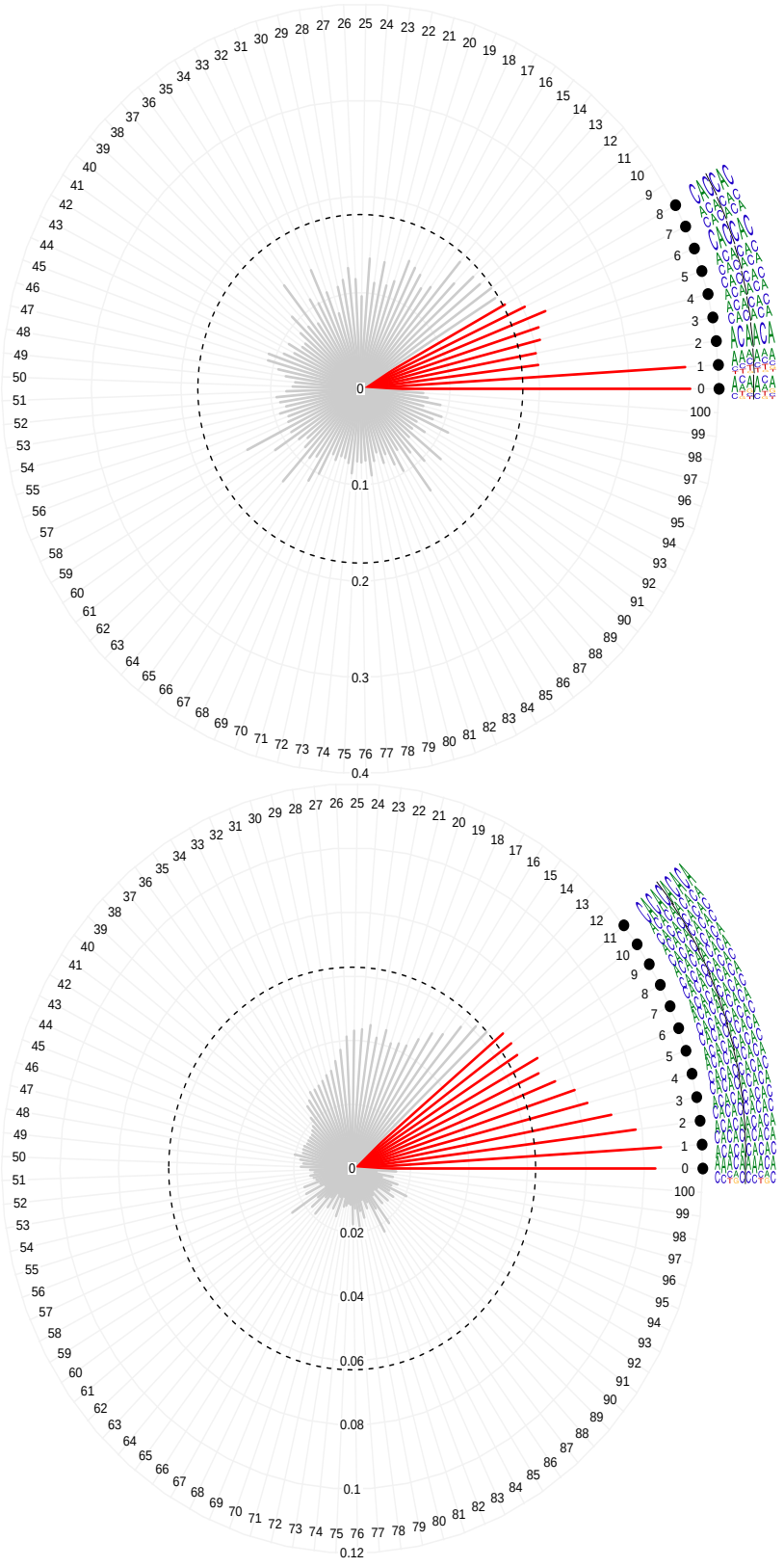


Figure 3.6.6: 'Top25' visualization of the informative 3-mer pairs separated by various distances and their magnitudes from the classifier for ***region* 7 in K562** (Refer Table 3.4.1 for *region* details). Top panel: Top-25 3-mer pairs contributing to predicting a locus as belonging to the negative class (red); Bottom: Top-25 5-mer pairs. Dashed inner circle is the threshold to select the top-25 dimensions of the SVM weight vector.

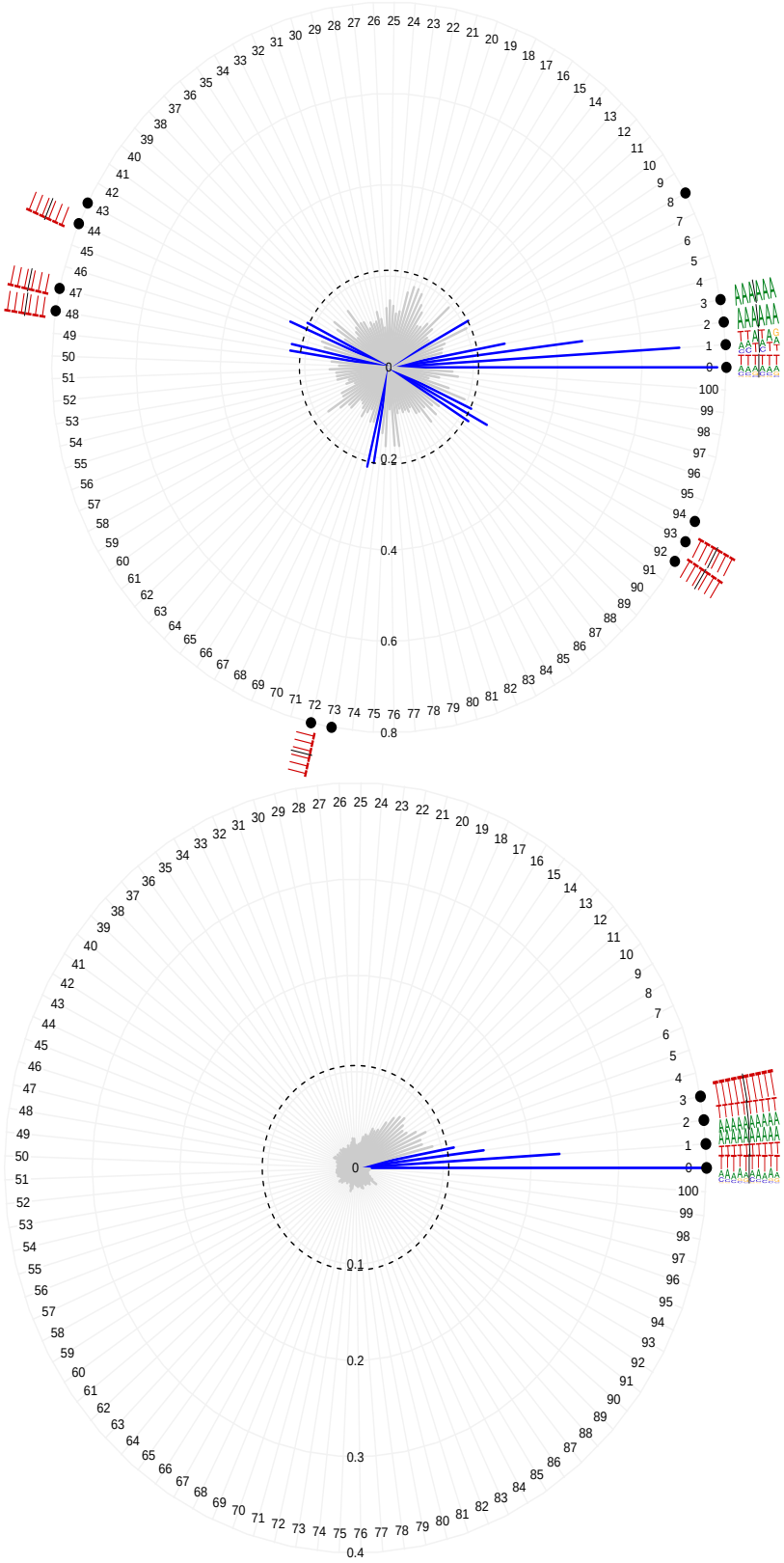


Figure 3.6.7: 'Top25' visualization of the informative 3-mer pairs separated by various distances and their magnitudes from the classifier for ***region* 7 in K562** (Refer Table 3.4.1 for *region* details). Top panel: Top-25 3-mer pairs contributing to predicting a locus as belonging to the positive class (blue); Bottom: Top-25 5-mer pairs. Dashed inner circle is the threshold to select the top-25 dimensions of the SVM weight vector.

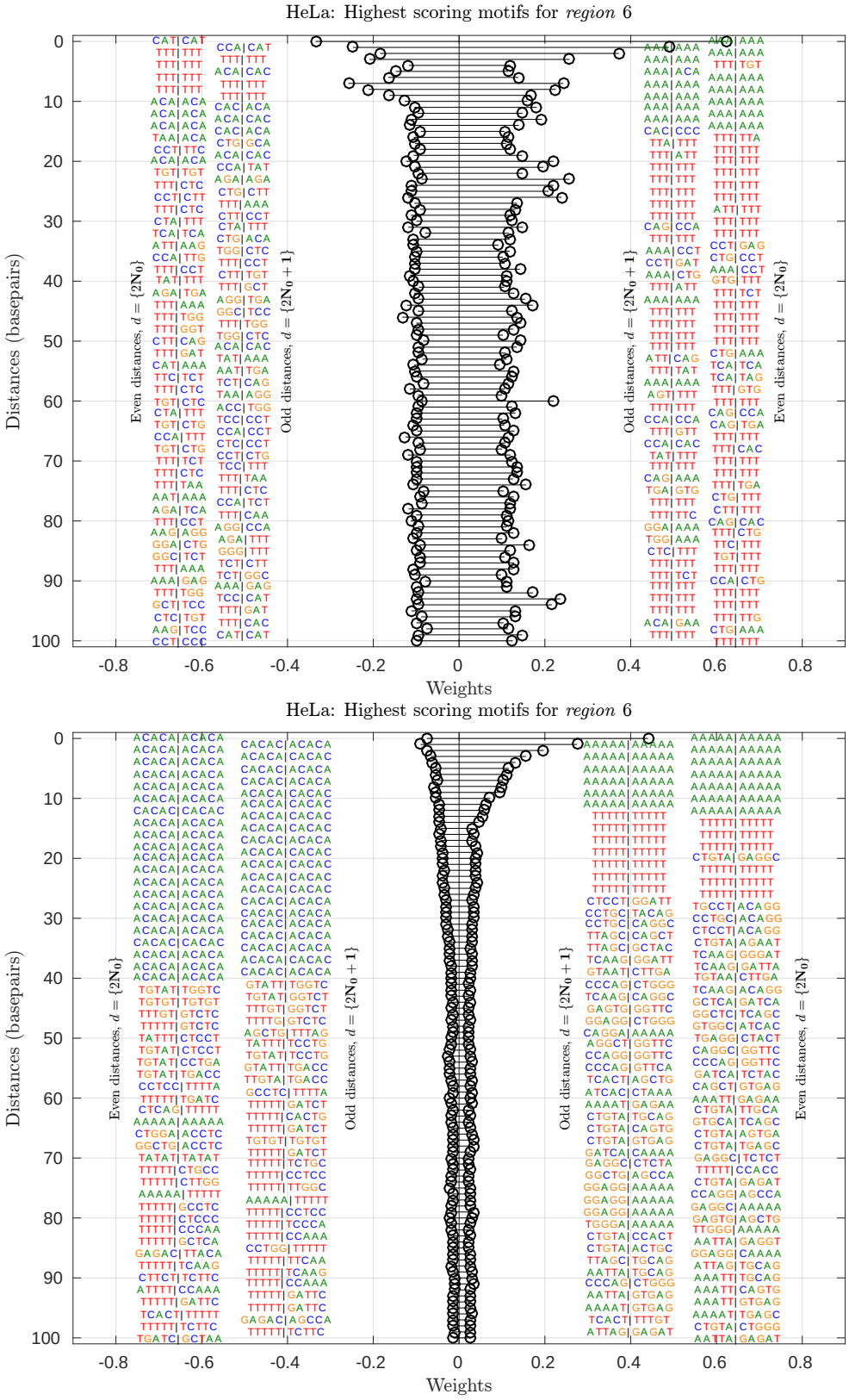


Figure 3.6.8: 'AMPD' visualization of the informative K-mer pairs from the classifier for ***region* 6 in HeLa-S3** (Refer Table 3.4.1 for *region* details). Top panel: At distances in (0-100), the 3-mer pair that maximally contributes towards positive and negative classification of a given locus is shown. Weights are shown on the horizontal axis, distances on the vertical axis. Bottom: Visualization of the 5-mer case.

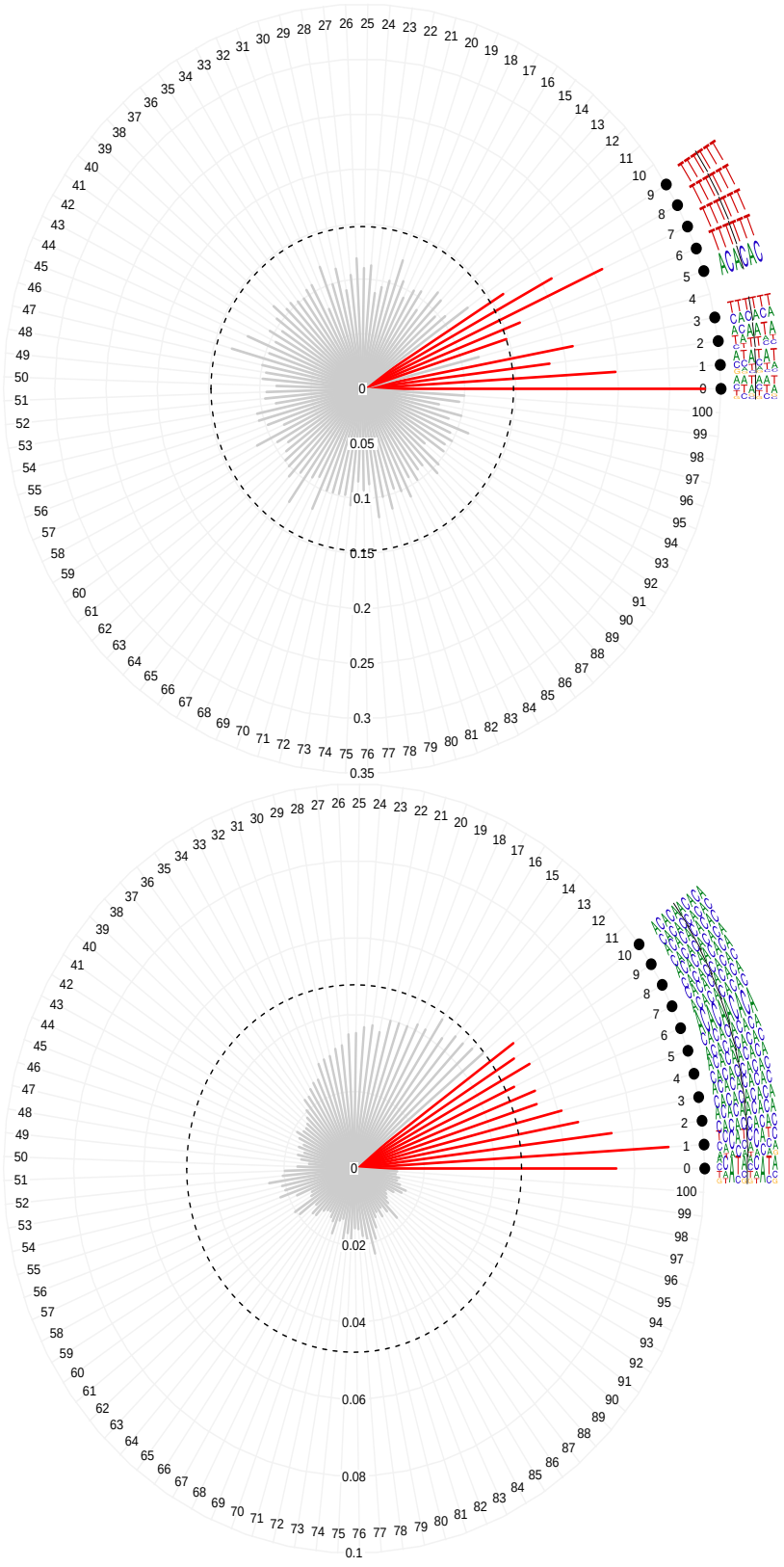


Figure 3.6.9: 'Top25' visualization of the informative 3-mer pairs separated by various distances and their magnitudes from the classifier for ***region* 6 in HeLa-S3** (Refer Table 3.4.1 for *region* details). Top panel: Top-25 3-mer pairs contributing to predicting a locus as belonging to the negative class (red); Bottom: Top-25 5-mer pairs. Dashed inner circle is the threshold to select the top-25 dimensions of the SVM weight vector.

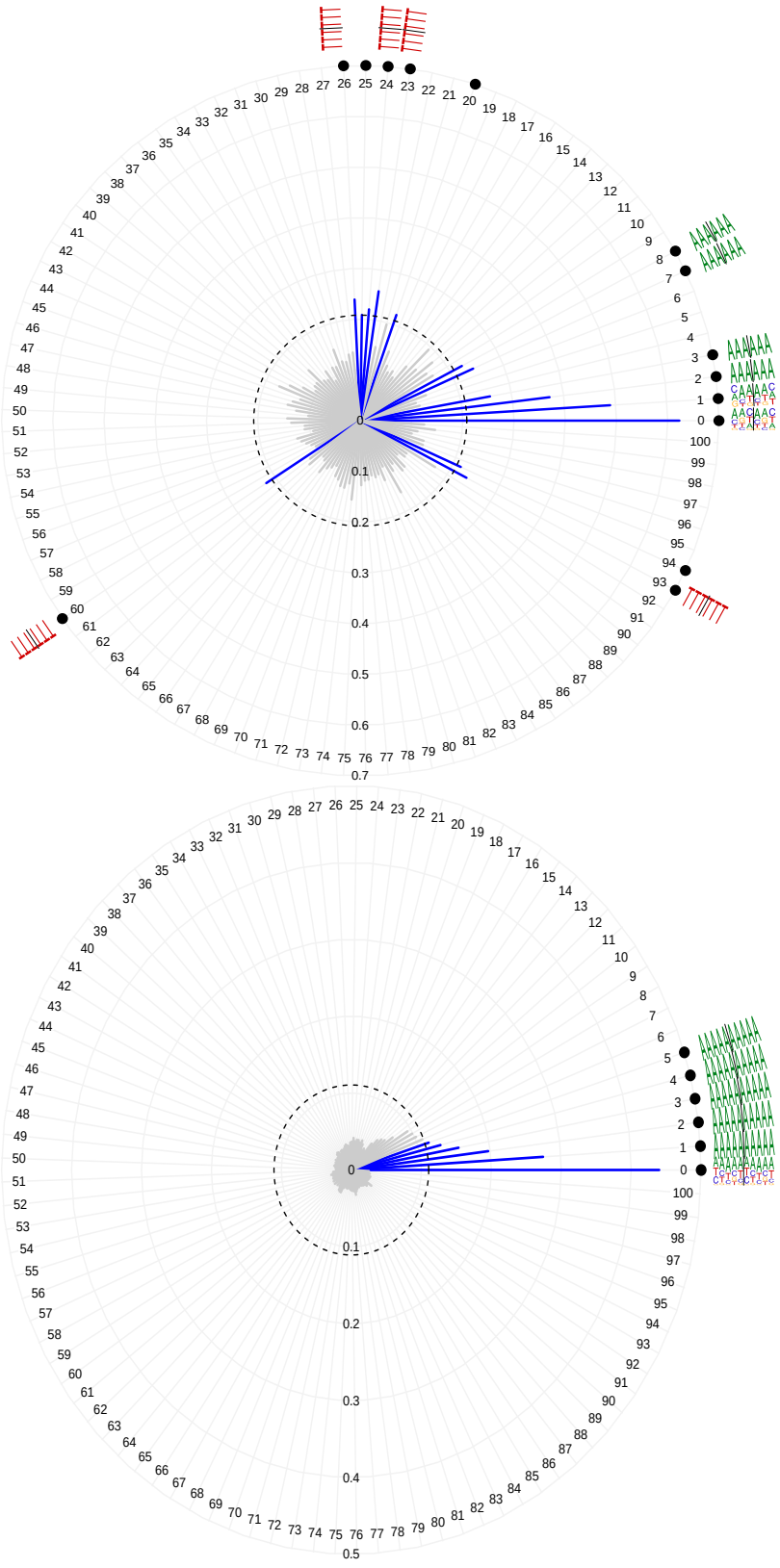


Figure 3.6.10: 'Top25' visualization of the informative 3-mer pairs separated by various distances and their magnitudes from the classifier for ***region* 6 in HeLa** (Refer Table 3.4.1 for *region* details). Top panel: Top-25 3-mer pairs contributing to predicting a locus as belonging to the positive class (blue); Bottom: Top-25 5-mer pairs. Dashed inner circle is the threshold to select the top-25 dimensions of the SVM weight vector.

task similarities also with oligomer length 3 and 5, respectively. The mean test AUC values for the multitask setting with 10 tasks are shown in columns marked 'C' and 'D' (oligomer length 3 and 5, respectively) of Table 3.6.1. Mean performance increase across all regions are:

Oligomer length 3 {GM12878, K562, HeLa-S3}: {0.13, 0.06, 0.13}

Oligomer length 5 {GM12878, K562, HeLa-S3}: {0.09, 0.06, 0.11}

Their box plots are shown in Figures 3.6.1 and 3.6.2. Performances in the MTL setting mostly show reduced variance as compared to the single-task performances.

Thus, our pipeline in the MTL setting can mitigate the issue of having too few interacting partners per locus. In the extreme case when a locus is not profiled, it can identify putative interaction partners of the locus, provided that at least some regions from the same cell line have been profiled with 4C or 5C .

3.6.5 Computational Validation with High-Resolution Hi-C

We attempted to test the ability of our locus-specific models trained on the restriction fragment-resolution 5C data from (Sanyal et al., 2012) to predict interaction partners of the same loci on high-resolution Hi-C data from (Rao et al., 2014). This section describes this experiment in detail.

Preparation of Validation Data

(Rao et al., 2014) performed Hi-C experiments resulting in contact matrices at very high-resolution for various cell lines including GM12878 and K562 (Rao et al., 2014). Contact matrices are available for resolutions 1K, 5K, 10K, 25K bp, etc. For both cell lines GM12878 and K562, we used information from *cis*-contact matrices at 5, 10 and 25K resolutions and considered 5K as our base resolution. That means the final set of sequences that we used for validation are 5K bp long.

These Hi-C *cis*-contact matrices were normalized using Knight and Ruiz (KR) normalization procedure (Knight and Ruiz, 2012). After normalizing, in order to identify significantly interacting partners of a locus, we computed the observed/expected (O/E) values for each pair of loci. Following Lieberman-Aiden et al. (2009), we used an ad-hoc cutoff of 2.5 to call an interaction significant. In other words, per contact matrix, a locus with a normalized O/E value ≥ 2.5 was considered significantly interacting with the LoI. The final set of loci significantly interacting with the LoI is obtained by the stringent criterion described below. We performed the same procedure as above for contact matrices at resolutions 5, 10 and 25K. Corresponding to our LoI, we marked those columns from the *cis*-contact matrix (of the relevant chromosome) which have an overlap with the LoI. For example, if the LoI was 12,000

bp long and matrix resolution 5K, we would mark three contiguous column bins in the matrix. These contiguous columns are together considered to correspond to the LoI in the 5C data. Then, we collected those bins along the rows of the contact matrix which have a non-zero KR-normalized interaction frequency with the LoI. From among these, bins which have a significant interaction with the LoI are considered as positive samples at resolution 5K. The above procedure of calling significant interaction partners is repeated for resolutions 10K and 25K. The final set of loci that are considered significantly interacting with any particular LoI includes only those that are significant at 5K resolution and also at 10K or 25K resolutions. In other words, if a locus was deemed significant only at 5K resolution but not at 10K or 25K, then we did not consider it a true positive.

These *cis*-interacting genomic loci from the high-resolution contact maps are treated as unseen test sequences for the classifiers built for each *region* using the 5C data. Each of these unseen test sequences is 5K bps long. In the pipeline, these are thus treated similar to the 20% hold-out set. The ODH feature representations of the unseen test sequences are fed to the classifier to predict their labels. We performed this experiment for cell lines GM12878 and K562.

Table 3.6.2: Computational validation with high-resolution Hi-C data. Reported values are mean±s.d. (s.d.: standard deviation)

Chromosome-wide interaction partners		
Cell-type	**Oligomer length 3**	**Oligomer length 5**
GM12878 (regions 0-4)	0.5552 ± 0.009	0.5503 ± 0.006
GM12878 (regions 0-9)	0.5358 ± 0.025	0.5279 ± 0.028
K562 (regions 0-4)	0.5508 ± 0.091	0.5650 ± 0.088
K562 (regions 0-9)	0.5122 ± 0.084	0.5239 ± 0.081
Interaction partners beyond 1M bp		
GM12878 (regions 0-4)	0.5468 ± 0.005	0.5419 ± 0.007
GM12878 (regions 0-9)	0.5327 ± 0.019	0.5220 ± 0.026
K562 (regions 0-4)	0.5593 ± 0.062	0.5646 ± 0.064
K562 (regions 0-9)	0.5304 ± 0.058	0.5294 ± 0.064

Validation in the 5C→Hi-C Setting

When evaluating performances of our models for predictions on unseen loci from Hi-C data, we did so for two scenarios. One, where all chromosome-wide loci are considered; and the other, only those lying beyond 1M bp from the 'model-defining' locus are considered. Using the stringent criterion described above, the mean AUC values and their standard deviations are as follows. For prediction with models using oligomer length 3:

(a) chromosome-wide partners: {GM12878, K562} : {0.5358±0.025, 0.5122±0.084}

(b) partners beyond 1M bps: {GM12878, K562} : {0.5327 ± 0.019, 0.5304 ± 0.057}

And, with models using oligomer length 5:

(a) chromosome-wide partners: {GM12878, K562} : {0.5278±0.028, 0.5238±0.081}

(b) partners beyond 1M bps: {GM12878, K562} : {0.5220 ± 0.026, 0.5294 ± 0.064}

For both cell lines, when considering only the first five regions, the average performance was ∼0.55 test AUC (see Table 3.6.2). Models for K562 show higher variance than models for GM12878. We observe that performances of models for some LoI are comparatively poorer than those of other models.

Training on contact information from 5C and predicting contacts chromosome-wide is a hard problem. We envisage there are two reasons contributing to this hardness. Having few negative samples to learn from is one, and the other is, having a rather long model defining locus (cf. Table 3.4.1). We hypothesize that this is due to the following reasons. First, the 3C assays give no information on the potential causal portion(s)—causal for the said interaction—along the complete restriction fragment. In this case, the genomic regions that are part of the reported restriction fragment but play no role towards the interaction, simply pose as noise. These regions cannot be easily weeded out. Second, the interacting as well as non-interacting partners of a rather long 'model-defining' locus can have many different contributing characteristics in them. Examples of this include the many transcription factor binding sites or genetic elements which impose important architectural restrictions. These may not be comprehensively captured by the few available samples in the 5C data. This especially affects a model that learns from 5C data and predicts on high-resolution Hi-C as is the case here. Third, these 5C experiments are performed on selected promoter regions and distal enhancers (Sanyal et al., 2012). We make the models trained on such restricted 5C data to predict a potential interaction partner anywhere on the genome not just promoter or distal enhancer regions. Fourth, contacts over different distance ranges are a result of different genomic backgrounds. This is completely violated in the 5C→Hi-C setting and evaluating the model's prediction performance on regions beyond promoters or enhancers exacerbates the issue.

3.7 Discussion

From the point of view of understanding chromatin interactions at the sequence level, ours is the first approach to study these interactions in a locus-specific manner. In this study, we hypothesized that the genetic sequence at the loci that significantly

interact with a LoI is informative in discerning them from loci that do not interact with this LoI. Studying chromatin interactions in a locus-specific manner gives novel insights into potentially important sequence-level mechanisms for three-dimensional organization of chromosomes.

As already noted, our aim in performing this study was two-fold. First, to establish if the genetic sequence alone can predict the long-range chromosomal interactions. Second, to understand the characteristic sequence features underlying these interactions. We achieved these by performing computational experiments on 10 regions each in three different cell lines. The motivation for deciding upon such an aim for our study was as follows. The existing approaches for analyzing and predicting long-range interactions focus at the level of *all* interactions vs. *all* non-interactions from a contact matrix (cf. Section 3.2). These approaches give insights into the general genome-wide, cell-type specific interactions. We envisage this genome-wide versus per-locus relationship on the computational side is analogous to the Hi-C versus 4C relationship on the experimental side. There are similar comparative advantages of the per-locus approach over the 'all versus all' prediction models and vice versa.

In comparison to the literature for prediction of enhancer-promoter interactions, we have used the term long-range chromatin interactions in a broader sense. These include possible interactions between intervening chromatin regions in addition to the enhancer-promoter interactions. We also hypothesized that the intervening chromatin could play an important role in maintaining a favorable landscape for the loci to interact. For example, it could provide the necessary structural or organizational backbone required for chromosome folding inside the nucleus. This view was largely motivated based on the hypothesis by Bickmore (2013). Thus, we use the complete restriction fragments instead of the shorter promoter or enhancer loci. Such broader set of interactions that can involve the intervening chromatin are termed *bystander* interactions (Sanyal et al., 2012). Hughes et al. (2014) observed such interactions in capture-C experiment data, and called them *structural* interactions. These are possibly weaker interactions due to putative low-affinity binding sites. Low-affinity binding sites have been largely unexplored as yet, even in general (Tanay, 2006).

Our classifiers trained on data from 5C experiments that probed selected TCRs and distal enhancers in three cell lines GM12878, K562 and HeLa-S3 (Sanyal et al., 2012). The classifiers attained an average test performance of $\sim$0.75 in the single-task setting. We developed two new, intuitive visualization methods that are suited for our problem scenario involving variable-length sequences and an appropriately chosen ODH feature representation. Aided by these visualizations, our per-locus models shed light on the potential sequence signals that can characterize the interactors versus the non-interactors of a LoI. Analysis of the various sequence signals from our models suggests a possible functional and organizational role for short tandem repeat

sequences in the genome potentially more than previously thought. We cited various recent studies corroborating this in Section 3.6.2. Furthermore, our approach can also identify cell line-specific sequence features characterizing the (non-)interactors of the same genomic locus ini two cell lines (see Section 3.6.3).

We also demonstrated how knowledge of individual models could be transferred to those of other regions (those having too few examples to learn from) via multitask learning. Mean performance for the multitask setting is 0.83. This is an average of the performances of all models for oligomer length 3 and 5 combined together.

We made our models trained on 5C data predict interactions between 5K bp long loci from the recent high-resolution Hi-C data (Rao et al., 2014) for cell lines where the Hi-C data was available. In this case, the prediction performance of our models was only slightly better than random. Another study to perform a similar 5C→Hi-C validation is by Roy et al. (2015). Following are the aspects pertaining to the 5C→Hi-C setting, where our approach is different in comparison to (Roy et al., 2015). First, we used a stringent criterion to identify true positives in the high-resolution Hi-C data. For this task, Roy et al. (2015) consider an interaction to be true positive if it is called a peak in any one of the three resolutions 5K, 10K and 25K bp. The ultimate resolution of the genomic loci considered is 5K bp (Roy et al., 2015). Second, our models are based on sequence information. Recall that for each locus, Roy et al. (2015) use a simple feature vector corresponding to the functional genomics information. Consequently, the segments on the restriction fragment that supposedly pose as noise affect our models adversely, more than theirs. Third, the additional layers of chemical modifications on the genetic sequence make a sequence-based model less authoritative towards determining whether a pair of loci interact. We already stated this in Section 3.3. Having said that, Roy et al. too achieved relatively modest performances in comparison to those in other settings viz. 5C→5C and Hi-C→Hi-C. They report performances in terms of the area under the precision-recall curve (auPRC). Specifically, their model achieved an auPRC of 0.643 in K562 and 0.687 in GM12878 for the 5C→Hi-C setting. Note that the genomic regions for which data are not available are left out all together from these state-of-the-art studies. Our sequence-based approach can be still be helpful in such scenarios. Moreover, we expect that our models can be further strengthened or supported by utilizing the additional regulatory (epi)genomic information wherever available.

Finally, an important point to note here is that our models do not require any locus to be either a TCR or an enhancer region per se. In principle, it can be seamlessly applied to contact matrices output by any 5C-based or even high resolution Hi-C-based experiments (as training data). At places, we have used the terms TCR and enhancers for the interacting regions because the contact matrices we use in this study come from 5C experiments involving these loci. So, when given a Hi-C contact

matrix, any locus therein could be used to learn corresponding models in a similar fashion, and it need not necessarily be an enhancer or a promoter region. It is for this reason, we preferred to call these genomic loci as simply *regions* in this study.

4

Comparison of Variable-Length DNA Sequences Using Conformal Multi-Instance Kernels

This chapter describes our work on comparing variable-length DNA sequences in the discriminative setting without much of the positional constraints of the existing string kernels. While this work was motivated by my earlier project (described in Chapter 3), it got fast-tracked as my immediate next project after a discussion with Prof. Ana Pombo at the EMBL conference on Transcription and Chromatin in 2016[1]*. Work described in this chapter was presented at WABI 2017 as (Nikumbh et al., 2017b). An earlier version is also available as a preprint, (Nikumbh et al., 2017a). Consequently, large portions of text in this chapter have been adapted from Nikumbh et al. (2017a) and (Nikumbh et al., 2017b). Author contributions are as follows: I conceived and designed the project with Nico's guidance. I implemented CoMIK. Peter Ebert's group seminar in the department motivated CoMIK's applicability for a more general problem of different promoter definitions. While I designed the synthetic data set, Peter contributed with ideas for suitable biological data sets for showing the general efficacy of CoMIK, helped with comments in improving*

[1] https://www.embl.de/training/events/2016/TRM16-01/

CoMIK's code-base in terms usability.

4.1 Introduction and Motivation

Until now we saw that all chromatin interaction data is pre-processed to output contact maps using either of the following strategies: (a) they perform uniform binning and produce a contact matrix with fixed-size genomic bins; or, (b) produce a contact matrix at restriction fragment-resolution wherein individual bins of the matrix are of non-uniform size in terms of the genomic regions but uniform in the number of restriction fragments per bin. This introduces a coarseness that affects the understanding and interpretation of chromatin interactions. For example, across studies, two arbitrarily long genomic loci that respectively contain a promoter and an enhancer are simply considered as an enhancer-promoter (EP) interacting pair. Any possible role played by the intervening or the flanking regions is ignored. This is illustrated in Figure 4.1.1.

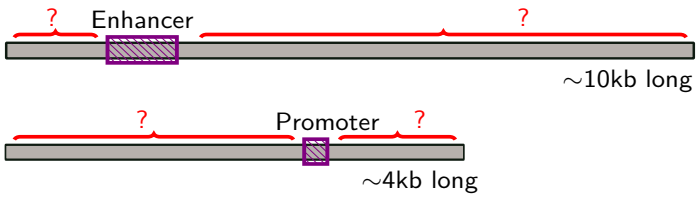


Figure 4.1.1: Illustration of an enhancer–promoter interaction (EPI). The enhancer and the promoter are shown as located somwhere on the restriction fragment shown in grey. Contact between such restriction fragments is simply considered as an EPI neglecting any possible role of the flanking or intervening chromatin.

In various studies since the elucidation of the human genome, many different definitions of promoters have been used in different studies. For example, Butler and Kadonaga defined a core promoter as a minimal stretch of contiguous DNA sequence (~40 nucleotides (nt)) that contains the TSS and is sufficient for accurate transcription initiation (Butler and Kadonaga, 2002; Juven-Gershon et al., 2008). A proximal promoter is a region in the immediate vicinity of the TSS, roughly 250 bp upstream and downstream (Butler and Kadonaga, 2002; Juven-Gershon et al., 2008). There are examples of many studies that consider using either an arbitrary-sized window around the TSS (albeit fixed for the study) or only the region upstream of it as promoter sequences. Some examples are shown in Figure 4.1.2. This gives rise to a conundrum about the choice of an appropriate size of a promoter in any new study or one that unifies promoter sequences from prior works. Both the scenarios described above warrant development of methods that can compare sequences of variable lengths.

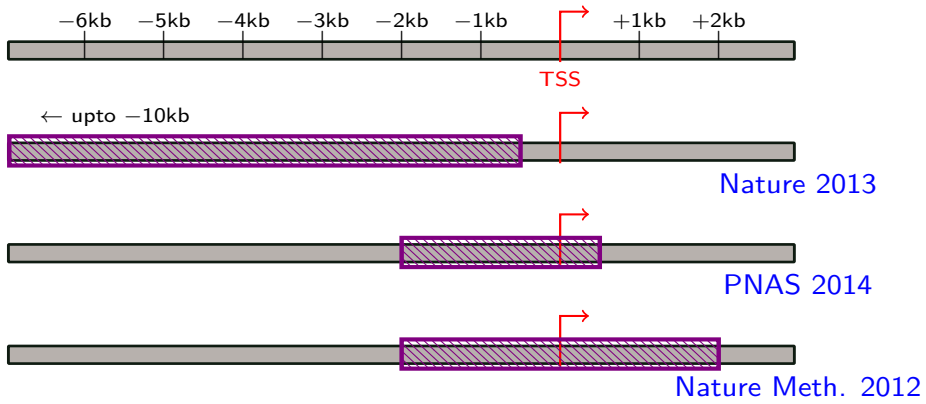


Figure 4.1.2: Some examples from literature using different promoters definitions are shown. From top to bottom, the three examples are Chen et al. (2013), He et al. (2014) and Ernst and Kellis (2012) respectively.

Discriminative machine learning methods like SVMs (Boser et al., 1992) together with string kernels have achieved state-of-the-art performance on many relevant problems in computational biology (e.g., splice site prediction (Rätsch et al., 2005)). The earliest kernel-based approaches for computing similarities between biological sequences, e.g., spectrum (Leslie et al., 2002) and mismatch kernel (Leslie et al., 2004), allow comparing sequences of different length. But they do not encode any positional information. Later approaches, for example, the weighted degree kernel (Rätsch and Sonnenburg, 2004) and the oligo kernel (Meinicke et al., 2004), do consider positional information in the corresponding sequences, some even with a certain amount of positional uncertainty (Rätsch et al., 2005). Additionally, alignment-based sequence comparison also provides a position-dependent similarity score albeit with a gap penalty (Saigo et al., 2004). Thus, these approaches do allow deviations from exact matches but they are penalized. The oligomer distance histograms (ODH) kernel (Lingner and Meinicke, 2006) allows comparing sequences of different length by representing a sequence with a fixed-length feature vector, but it ignores information about the position of such oligomer pairs within the sequence. See Chapter 2 for a brief overview of the above kernels.

Figure 4.1.3 outlines the above mentioned scenarios. Any position-aware kernel that also allows shifts can detect the signal in case (a), but not in case (b), where the signal is very far apart. Even if it does, it would penalize this deviation. Case (c) represents how ODH would detect this signal and thus consider the two sequences to be similar, but information on the position of this signal in the individual sequences is lost. The work presented in this chapter is a step towards filling this gap. We want to compare sequences allowing reasonable degree of positional freedom without

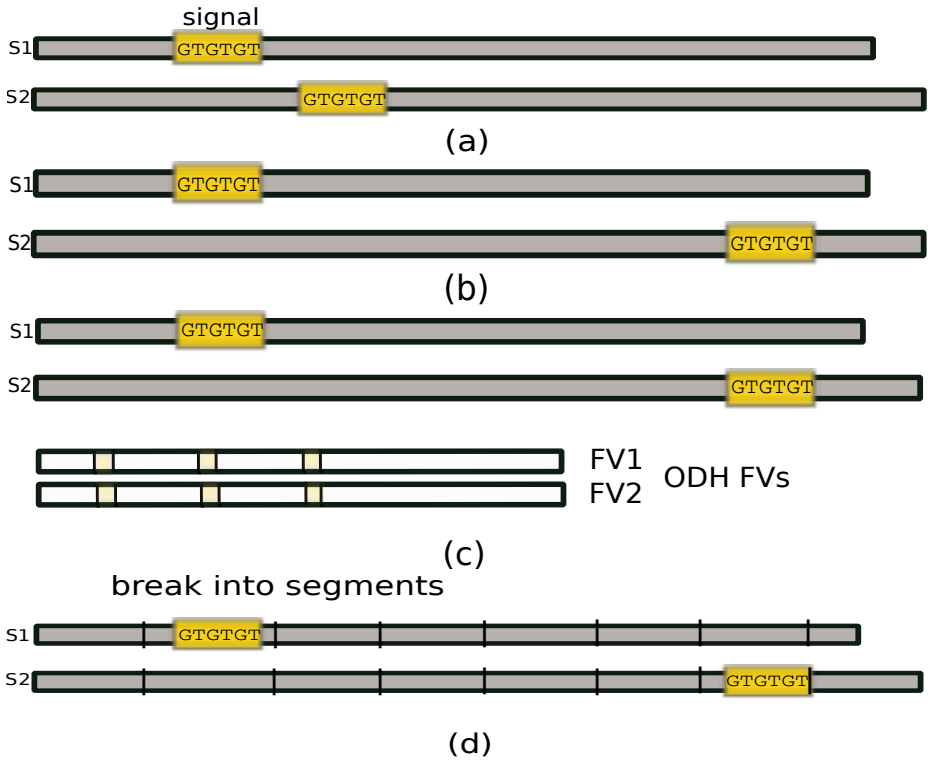


Figure 4.1.3: Various scenarios comparing sequences of different lengths using existing approaches.

simultaneously penalizing this deviation, or, better yet, keep it problem-dependent. This scenario, as we have seen, can arise in case of chromatin conformation data. Here, the pairs of loci interacting over a long-range are variable-length restriction fragments reported from the experiments. The putative causal signal in the two interacting loci does not have any positional restriction unlike the transcription start site in the promoter sequences.

We approach this problem of handling comparison of variable-length sequences in a discriminative setting. Such a comparison also warrants allowing positional freedom as motivated above. We cast the typical binary classification problem involving pair-wise sequence comparisons into a multiple instance learning (MIL) problem (Dietterich et al., 1997) (see Section 2.3.5 for a general introduction to MIL). Briefly, in this MIL setting, each sequence is broken into segments (Figure 4.1.3 (d)). Any segment can hold predictive features important for the classification problem. Then, in order to compare two sequences, all segments of one sequence are compared to those of the other. Such a bipartite comparison of all segments of any two sequences can point to the importance of the individual segments of the sequences towards their similarity. We employ conformal multi-instance kernels (Blaschko and Hofmann, 2006) to obtain the importance for segments of each sequence for the classification problem. Thus, casting into an MIL problem enables our two-fold objective—handling variable-length sequence comparison and allowing positional freedom in doing so.

This renders the capability to identify segments of a sequence informative for the classification problem. We call our approach *CoMIK* for 'Conformal Multi-Instance Kernels'.

In the following, we begin with a detailed description of *CoMIK* in Section 4.2. We first describe our complementary segmentation procedure (Section 4.2.1). Further, we show how we exploit this design with the help of conformal transformations to the multi-instance kernel (Blaschko and Hofmann, 2006) to identify important segments of a sequence towards its classification with SVMs (Section 4.2.2). Subsequently, we discuss efficient retrieval of the SVM weight vector for the complex setting of multiple conformal multi-instance kernels in Section 4.2.4. Then, we demonstrate how to interpret the nonlinear classifiers by adopting visualization techniques introduced in the previous chapter. Results of the computational experiments follow.

4.2 Methods

4.2.1 Segment Instantiation with Complementary Views

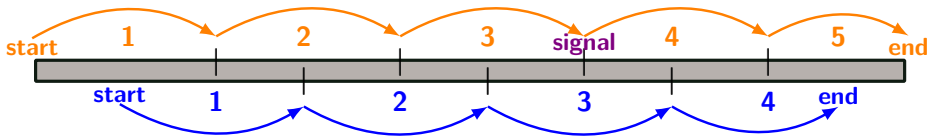


Figure 4.2.1: Illustration of complementary segmentation procedure. The sequence is shown in gray. *Non-shifted* segmentation is shown on top, in orange color, and *shifted* segmentation with blue. Segmentation begins at the position marked 'start', and ends at the position marked as 'end'.

Non-shifted Segment Instantiation

Given any arbitrary length sequence, we propose representing it by its segments where a segment is defined as a smaller part of the whole sequence. Beginning right at the start of the sequence, we create segments of a predetermined size along the sequence until it ends. The last segment is allowed to have a different size, either smaller or larger than the other segments (elaborated below). This accommodates any remainder portion of the sequence in case the sequence length is not an exact multiple of the segment-size. If the final segment is as short as half of the predetermined segment-size or shorter, we concatenate it to the penultimate segment, making the eventual final segment longer than the other segments. In other cases, it is maintained as is, leading to a final segment shorter than the other segments. This segmentation provides a non-shifted view of the whole sequence as the first segment starts at the

beginning of the sequence and, together, the segments span the entire sequence. We call this instantiation the *non-shifted* segment instantiation. Figure 4.2.1 illustrates the *non-shifted* segmentation procedure in orange color.

Shifted Segment Instantiation

There may still be signals at the boundaries of any two *non-shifted* segments (see Figure 4.2.1) which may get overlooked when comparing sequences using just *non-shifted* segments. To cover for this scenario, we introduce an alternate instantiation called *shifted* segmentation whereby the boundaries due to *non-shifted* segmentation of the sequence end up in the same segment in this representation. In this case, segmentation begins from the mid-point of the first *non-shifted* segment, and proceeds to create further segments along the sequence essentially covering the boundaries of the *non-shifted* segments. The portions of the sequence before start and after end can be omitted since they are already covered (in the *non-shifted* view). *Shifted* segments can either be of the same size as the *non-shifted* segments or different. Thus, *shifted* segmentation provides a complementary view of the same sequence covering the portions which get overlooked by *non-shifted* segmentation. A simple

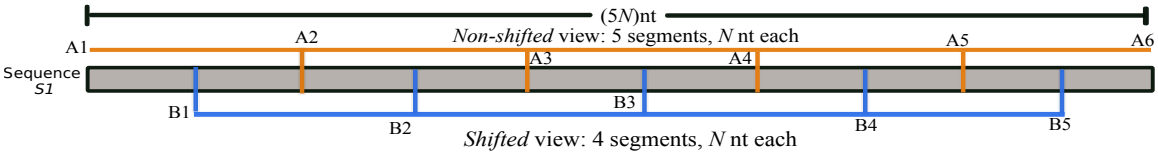


Figure 4.2.2: Complementary segmentation is illustrated on a sequence. In this dummy example case, the sequence is (5N)nt long and the individual segments are N nt long. The ends of shifted segments are marked by points B1-B5, and those of non-shifted segments, by points A1-A6. A1 and A6 coincide with the beginning and end of the sequence itself. And, B1-B5 are mid-points of the 5 *non-shifted* segments. In *shifted* segmentation, sequence portion before B1 and after B5 can be ignored. See text for details on handling the scenario when the sequence-length is not a multiple of the segment-size.

case of *non-shifted* and *shifted* segmentation is depicted in Figure 4.2.2. The segment-size is chosen *a priori* by the user as is suitable for the problem at hand. Refer to Section 4.2.3 for a discussion on choosing an appropriate segment-size and its overall influence on the algorithm.

4.2.2 Conformal Multi-Instance Kernels for Complimentary Set of Segments

Once segmented, we cast this problem into a multiple instance learning (MIL) problem (Dietterich et al., 1997). Recall from the background chapter, Section 2.3.5, in MIL, each sample (X, y) represents a set X of instances x ($x \in X$) and a label y

for X. The sets of instances are also called bags. One or more instances from a bag could be responsible for the bag to be classified as positive or negative due to the presence or absence of class-specific features. A bag can have any number of instances. In *CoMIK*, each sequence is treated as a bag and all its segments—*non-shifted* and *shifted*—as instances in the bag. Therefore, *CoMIK* can inherently handle sequences of arbitrary lengths.

Multi-Instance Kernels

Gärtner et al. proposed the normalized set kernel (*a.k.a.* multi-instance kernel) for the multiple instance problem (Gärtner et al., 2002) (cf. Section 2.3.5). Briefly, for each sample represented as a bag of instances, the kernel value between any two bags X and X', $k(X, X')$, is given as in Eq. (4.1).

$$k(X, X') := \frac{k_{\text{set}}(X, X')}{f_{\text{norm}}(X) f_{\text{norm}}(X')} \tag{4.1}$$

where $k_{\text{set}}(X, X') := \sum_{x \in X, x' \in X'} k(x, x')$. Here $f_{\text{norm}}(X)$ is a suitable normalization function. One could normalize using either averaging ($f_{\text{norm}}(X) := \#X$, where $\#X$ denotes the number of instances in bag X) or feature space normalization ($f_{\text{norm}}(X) := \sqrt{k_{\text{set}}(X, X)}$). In this work, we used feature space normalization.

While the multi-instance kernel can successfully handle comparison between bags by comparing their individual instances, it has the issue that, in averaging, it looses any information related to the contributions of the individual instances. In other words, it treats all the instances in a bag equally. And, it is usually desirable to not only obtain a solution to a problem, but also to identify (a) the features that contribute to that specific solution, and (b) the parts which contain these features. Here, (b) amounts to knowing which instance(s) in a bag have features that helped determining the correct class label of the bag (positive or negative class). To this end, we propose using conformal multi-instance kernels (Blaschko and Hofmann, 2006) that allow us to obtain an instance weighting based on the contribution of these instances to learning the discriminant function.

Conformal Multi-Instance Kernels

Blaschko and Hofmann proposed the conformal multi-instance kernel as a modification to the normalized set kernel (Blaschko and Hofmann, 2006). This modification is a conformal transformation parameterized by θ, $t_\theta > 0$, applied to the kernel function. The conformal transformation preserves the angle between vectors in the mapped space. The idea is to magnify those regions in the feature space which are discriminative and shrinking those which are not discriminative. Selection of these

candidate regions in the feature space is done by clustering. All input instances are clustered using any clustering algorithm and the corresponding cluster centres are chosen as candidate regions or expansion points. The decision of whether the region characterized by any cluster centre is discriminative or not is made by solving the multiple kernel learning problem as explained further.

Blaschko and Hofmann proposed: (a) the conformal transformation $t_\theta(x)$ to be of the form given in Eq. (4.2).

$$t_\theta(x) = \sum_{e=1}^{E} \theta_e \tilde{\kappa}(x, c_e) \tag{4.2}$$

$$\tilde{\kappa}(x, c_e) = \exp\left(- \frac{\|x - c_e\|^2}{2\sigma^2} \right) \tag{4.3}$$

Here, c_e's denote the cluster centres indexed by $e \in \{1, \ldots, E\}$ for a total of E expansion points; and (b) $\tilde{\kappa}$ to be a Gaussian (Eq. (4.3)) whose bandwidth (σ) can be adjusted. The parameter θ_e in Eq. (4.2) tells how discriminative the region around a certain cluster centre is. A large value of θ_e denotes that the neighborhood of the corresponding expansion point c_e is a discriminative region. As mentioned, the θ_e values are learned via multiple kernel learning (see below and Eq. (4.5)). Thus, replacing $k(x, x')$ by its conformal transformation $t_\theta(x)t_\theta(x')k(x, x')$

$$k(X, X') = \frac{1}{f_{\text{norm}}(X) \cdot f_{\text{norm}}(X')} \sum_{x \in X} \sum_{x' \in X'} t_\theta(x) t_\theta(x') \underbrace{k(x, x')}_{\text{base kernel}} \tag{4.4}$$

Identifying expansion points. Following Blaschko and Hofmann, we use k-means clustering to identify clusters. The corresponding cluster centres, c_e's, are then treated as expansion points ($E = k$). Here, the individual instances are represented by their ODH feature vectors as discussed in Section 4.2.2. Too many instances can create a bottleneck for clustering. Blaschko and Hofmann suggest using the buckshot clustering heuristic (Cutting et al., 1992) in this scenario. By this heuristic, to identify E clusters from n instances, one can perform k-means on randomly sampled $\sqrt{En}$ instances (Blaschko and Hofmann, 2006). This has been shown to identify qualitatively similar clusters and being highly scalable at the same time (Blaschko and Hofmann, 2006).

Resultant conformal multi-instance kernel. Upon substituting Eq. (4.2) in Eq. (4.4), and simplification (see (Blaschko and Hofmann, 2006) for more details), the conformal multi-instance kernel is given by

$$k(X, X') \approx \sum_{e=1}^{E} \theta_e^2 \left(\frac{1}{f_{\text{norm}}(X) \cdot f_{\text{norm}}(X')} \sum_{x \in X} \sum_{x' \in X'} \tilde{\kappa}(x, c_e) \tilde{\kappa}(x', c_e) \underbrace{k(x, x')}_{\text{base kernel}} \right) \tag{4.5}$$

Eq. (4.5) is then posed as a multiple kernel learning (MKL) (Bach et al., 2004) problem (linear in $\beta_e \equiv \theta_e^2$) to simultaneously learn the θ_e's and the SVM parameters α, also called λ in part of the literature.

Obtaining individual instance weights. Upon solving the MKL problem, once the sub-kernel weights (θ_e's) are obtained we can directly obtain $t_\theta(x)$ for any instance x of a bag X using Eq. (4.2).

Figure 4.2.3 shows the resultant kernel matrix.

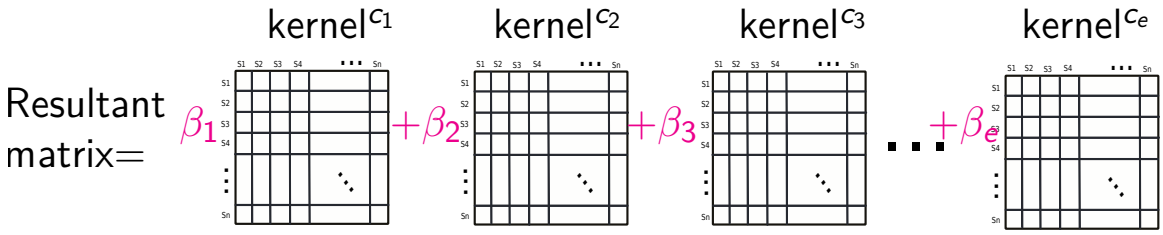


Figure 4.2.3: Resultant kernel matrix as a weighted sum of conformally transformed kernels corresponding to cluster centres $c_1, \ldots, c_e$. $\beta_{1,\ldots,e}$ are the weights assigned to the different sub-kernels upon solving the MKL problem.

Oligomer Distance Histograms (ODH) Kernel as Base Kernel

The choice of the base kernel to compare the individual instances depends on the problem. Here, we propose representing the individual segments of a sequence by its ODH representation and using the ODH kernel (Lingner and Meinicke, 2006) to compute similarities between segments. Using the ODH representation enables comparing variable-length segments to one another. Furthermore, it allows predictive motifs to occur anywhere within the segment.

We only reproduce the mathematical form of the ODH representation of a sequence 's' (Eq. (4.6)) and the ODH kernel (Eq. (4.7)) here. Refer to Section 2.3.3 for more details on ODH representations.

$$\Phi(\mathrm{s}) = [\mathbf{h}_{11}^{\mathrm{T}}(\mathrm{s}), \mathbf{h}_{12}^{\mathrm{T}}(\mathrm{s}), \ldots, \mathbf{h}_{MM}^{\mathrm{T}}(\mathrm{s})]^{\mathrm{T}} \tag{4.6}$$

The N training samples are given as: $\mathbf{X} = [\Phi(\mathrm{s}_1), \ldots, \Phi(\mathrm{s}_N)]$ and the $N \times N$ kernel matrix is given by

$$\mathbf{K} = \mathbf{X}^{\mathrm{T}}\mathbf{X} \tag{4.7}$$

4.2.3 Choosing an Appropriate Segment-Size

While the user could choose a segment-size that is appropriate for a problem, there is a four-way trade-off one should consider. This four-way trade-off involves the following

factors: the segment-size itself, the resulting number of segments, the computation time, and the prediction performance. The ODH kernel computation involving dot products between very high-dimensional feature vectors benefits from the sparsity of these feature vectors. But, with just 4 characters in the DNA alphabet, representation of a very long segment may not be sparse enough to reap the benefits of sparse computations. Therefore, shorter segments are preferred in this case. But, a small segment-size could result in a large number of segments if the sequences in the study are rather long. And, having too many segments influences the computation time spent performing clustering and subsequently applying the transformation per segment. Also, prediction performance-wise, if the segments are too small they may not cover the predictive motifs.

In general, many long segments in total from all the sequences could lead to a longer computation time for the instance-wise base kernel at the training stage. But this is a one-time computation needed to be performed in the beginning.

4.2.4 Interpretation and Visualization of Features

In the following, we discuss how one can interpret and visualize the sequence features deemed important by *CoMIK* for a prediction problem.

Obtaining the SVM Weight Vector for *CoMIK*

In the MKL problem (Bach et al., 2004), the weight vector corresponding to a given sub-kernel K_e is given as in Eq. (4.8).

$$\mathrm{w}_e = \beta_e \sum_{i=1}^{N} \alpha_i y_i \Phi_e(X_i) \tag{4.8}$$

$$\Phi_e(X) = \frac{1}{B} \sum_{x \in X} \tilde{\kappa}(x, c_e) \phi_e(x) \tag{4.9}$$

Here β_e is the sub-kernel weight learned by solving the MKL problem and each $\Phi_e(X_i)$ is the feature space representation of sequence X_i corresponding to sub-kernel K_e. For the conformally transformed multi-instance setting, this means $\Phi_e(X)$ is the bag-level ODH representation of the sequence upon transformation w.r.t. to the cluster centre characterizing the sub-kernel K_e. Thus, $\Phi_m^e(X)$ can be represented mathematically as in Eq. (4.9), where $\phi_e(x)$ is the ODH representation of segment x (Eq. (4.6)) belonging to bag X, $\tilde{\kappa}(x, c_e)$ is the Gaussian transformation (Eq. (4.3)) and B is the feature space normalization factor. Following Shawe-Taylor and Cristianini (2004), B can either be $\sqrt{k(X, X)}$ or $\| \sum_{x \in X} \tilde{\kappa}(x, c_e) \phi_e(x) \|_2$ since our base kernel, the ODH kernel, is a dot product kernel (refer to Section 4.2.2). Thus, we have a bag-level representation

of a sequence corresponding to all cluster centres which allows us to compute all the relevant weight vectors. These individual weight vectors can also be used to make fast predictions on test examples. For this, we only need the transformed ODH representations of the test examples corresponding to each kernel in the collection.

Visualizing Features from the *CoMIK* Weight Vector

Figure 4.5.1 shows visualizations of the features deemed important by *CoMIK* in discerning the positive set of sequences from the negative set. The top panel in Figure 4.5.1 shows the 'Absolute Max Per Distance' (AMPD) visualization (Nikumbh and Pfeifer, 2017) that provides a distance-centric view of features (cf. Chapter 3, Section 3.5.2). Recall that the AMPD visualization shows the K-mer pairs assigned the most positive and most negative coefficient in the discriminant at all distances considered. The bottom panel shows the K-mer-centric view which was introduced by Lingner and Meinicke (2006). It shows the importance of each K-mer pair towards prediction. Simply stated, the K-mer-centric view of the discriminant is a matrix which is obtained by taking an ℓ_2-norm of the weight vector with itself. A K-mer pair which holds high importance will have a high absolute value in the matrix.

4.2.5 Implementation and Availability of Software

MATLAB implementation of *CoMIK* is made available on Github at: `https://github.molgen.mpg.de/snikumbh/comik`. For non-MATLAB users, we provide an executable version of *CoMIK* which can be run together with MATLAB Runtime[2]. *CoMIK* takes as input a positive and negative set of sequences as separate FASTA files and performs complementary segmentation. Further, it uses Shogun's (Sonnenburg et al., 2010) MKL solver to obtain the sub-kernel weights. *CoMIK* is licensed under the MIT License.

4.3 Data Sets

To establish the general efficacy of *CoMIK*, we performed experiments on a simulated data set consisting of variable-length sequences and a yeast data set with the typical fixed-length sequences scenario. Both of these are described next. Additionally, we briefly describe the long-range enhancer-promoter interaction data used by (Whalen et al., 2016).

[2]MATLAB Runtime available at: `https://mathworks.com/products/compiler/mcr.html`

Simulated Data Set

We prepared a simulated data set of 1000 arbitrary-length sequences with a mix of many coupled and non-coupled motifs as explained below. Of these 1000 were three kinds of positive sequences totaling 500; the rest 500 comprised of two kinds of negative sequences. Refer to Table 4.3.1 for the following: (a) 300 of the 500 positive

Table 4.3.1: Motif sets planted in the simulated data set. The differences between the positive and the negative variants are underlined (e.g., 4P and 4N). '–' denotes a gap. Columns marked '+' and '−' give the number of positives and negatives respectively containing the corresponding set of motifs. Columns 'P' and 'N' give the #segment (non-shifted) in which the motif could lie (start positions).

Set	Motifs	+	−	P	N
A	1.`GAGTTATACATGGTATAGACCACACTATTA'	300	300	{1,2}	{2,3}
	2.`AACATGGTCTAGACCATTTT'			{3}	{1}
	3.`CTAAACAGGGTCTATACCACACTATTA'			{5}	{5}
	4P.`AGGATATATATGTGCTCTTCAGATTTTCACCCTTAGCAAGAGCGAGG'			{6}	-
	4N.`ACCATATACATGTGCAGATCAGATTTTCACCCCGAGCAAGAGCGAGG'			-	{6}
	5P.`ACACAGCTACTACCACAGGGACAGACAGACAG'			{4}	-
	5N.`ATAGCGCTACTACCACACCCACAGACAGACAG'			-	{1}
B	1.`ACCATATACATGTGCAGATCAGATTTTCACCCCGAGCAAGAGCGAGG'	100	200	{3}	{2,3}
	2.`ATAGCGCTACTACCACACCCACAGACAGACAG'			{2}	{1}
	3P.`GACACATGTGCACATATGGTTTTCACCCCGATACATAGTGAGG'			{4}	-
	3N.`GACACATGTGCACATATG-TAGCGAGG'			-	{3,4}
C	`GA' repeated at every 10 nt in the sequence	100	-	-	-

sequences had motifs from set A planted in them (column marked '+'), all except those marked with N (e.g., 4N and 5N which are negative variants of the positive motifs 4P and 5P, respectively). (b) Another 100 positive sequences had motifs from set B planted in them; 3P and 3N denoting variants as in (a). (c) Additional 100 positive sequences had the dinucleotide `GA' repeated at every 10nt throughout the sequences. For the 500 negative sequences, 300 contained all motifs from set A (1, 2, 3 and the negative variants) and the remaining 200, similarly, with motifs from set B. In all the sequences, each motif was planted at a randomly chosen start position inside a respective window. For *CoMIK*, it was then possible to determine the segment in which the different motifs could lie. Since we later discuss results with segment-size 70 nt, columns 'P' and 'N' already give the segment numbers (for non-shifted segments) where each motif could lie. Length of sequences of type (a) and (b), either positive or negative, was in the range [300,500] nt, and [500,600] nt for type (c). All sequences were generated with uniform probabilities for A, C, G and T and the motifs had a 0.1 mutation probability. Maintaining equal proportions of the different kinds of positives and negatives, we held out 200 sequences as unseen test examples (100 positives and 100 negatives) and used the remaining 800 sequences for training.

Yeast Data Set

Lubliner et al. (2015) studied yeast core promoter sequences analyzing the effect of sequence variation in different core promoter regions. Among other things, the authors showed that location, orientation, and flanking bases are important for `TATA` element function. We obtained a total of 316 core promoter sequences, each 118 bp long. The core promoter activity measurements for each of these sequences are provided. We followed the procedure in Figure 5 in (Lubliner et al., 2015) to classify them into two classes, sequences showing either low or high activity (expression). This resulted in 28 positive and 288 negative sequences.

Table 4.3.2: Number of positive and negative sequences in the 5C data set.

	GM12878	K562	HeLa-S3
#Positives	63	46	98
#Negatives	226	105	207

5C Data Set

This is the data set used in our earlier work (cf. Chapter3, Section 3.4). In order to demonstrate the efficacy of *CoMIK* compared to our earlier approach, we performed experiments on one region from each cell line. We fetched the positive and negative set of sequences for *region* 0 in K562, GM12878 and HeLa. The number of positive and negative sequences for each of these are given in Table 4.3.2.

4.4 Experimental Setup

Table 4.4.1: Parameters and the range of values tested for the simulated, 5C and the yeast data set.

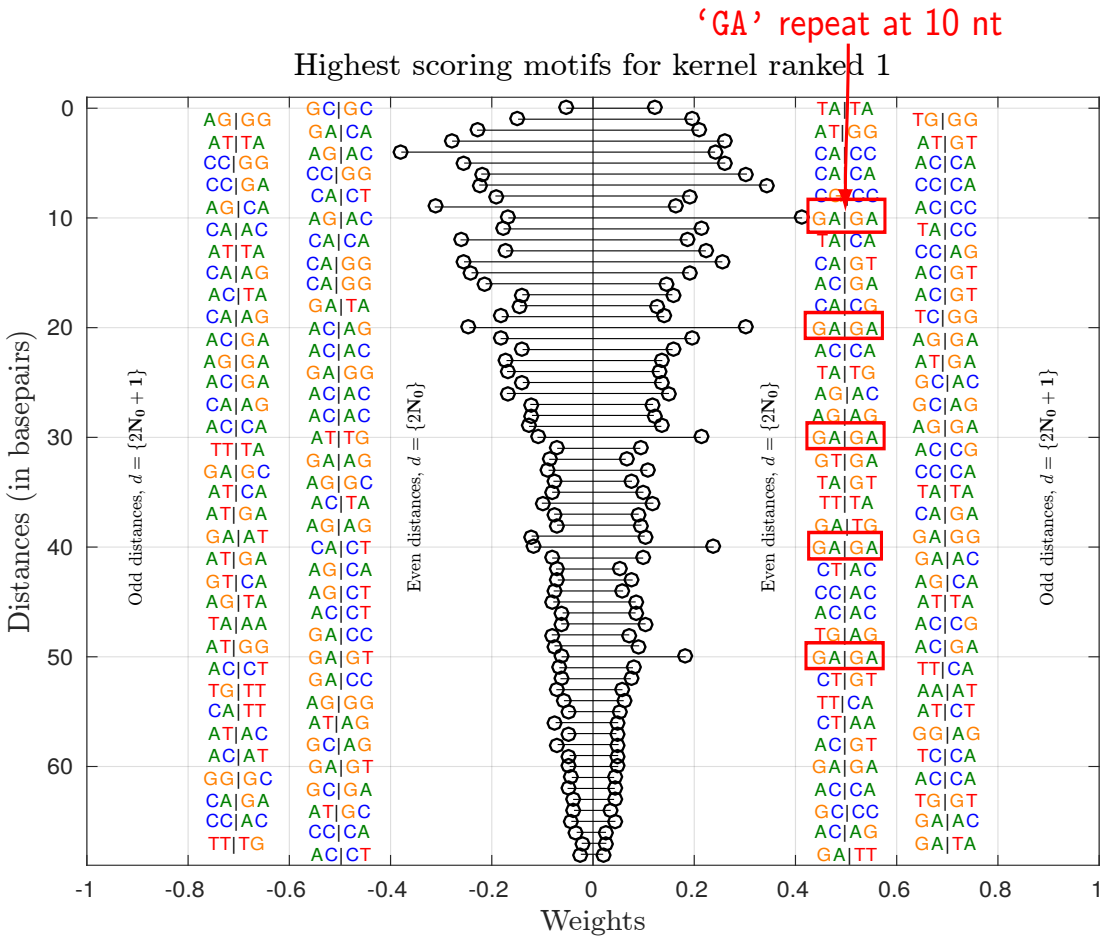
Parameters	Simulated data set	5C	Yeast
#Clusters	$\{2, 5, 7\}$	$\{5, 7, 10\}$	$\{2, 5, 7\}$
Segment-size	$\{50, 70\}$	50	10
Sigma (σ) for Gaussian transformation	$10^{\{-1,\ldots,2\}}$	$10^{\{2,4,6\}}$	$10^{\{-1,\ldots,2\}}$
Oligomer length	$\{2, 3\}$	$\{2, 3\}$	$\{2, 3\}$
Maximum distance	$\{50, 70\}$	50	10
SVM-cost	$10^{\{-3,\ldots,3\}}$	$10^{\{-3,\ldots,6\}}$	$10^{\{-3,\ldots,3\}}$

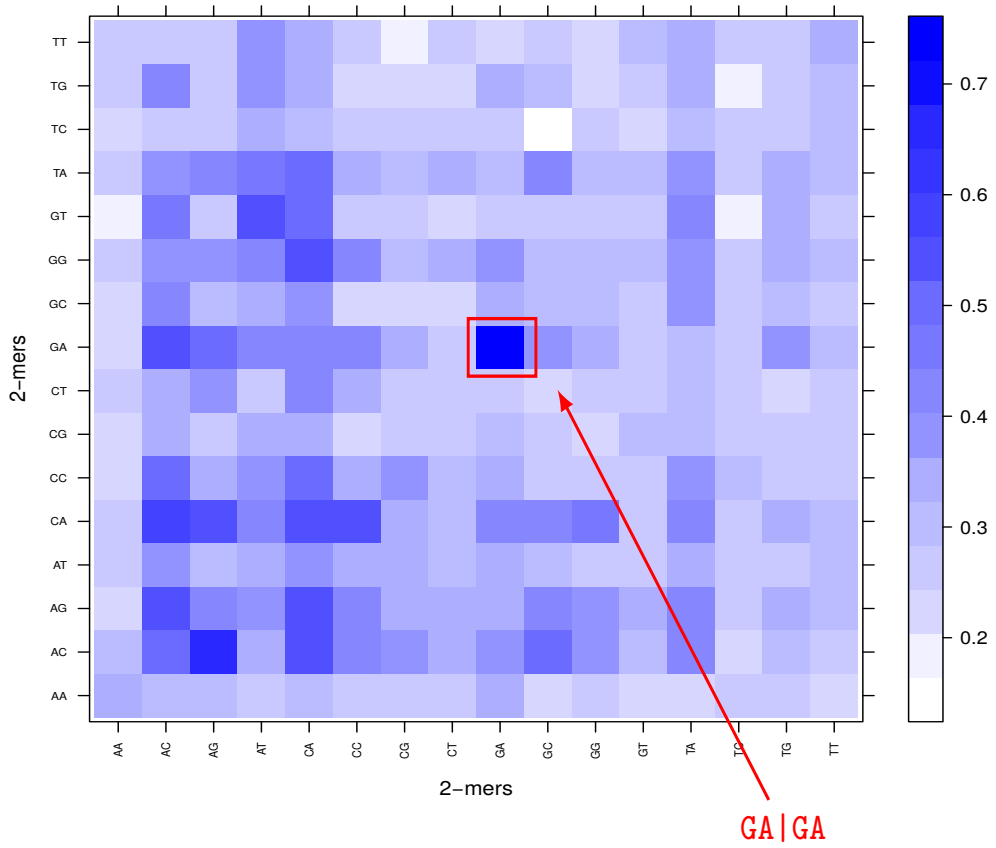
For each data set, we performed 5-fold nested cross-validation (CV) by splitting the data into 80%:20% for training and test, respectively. For each outer-fold, model selection was performed with a 5-fold inner CV loop on the training set with ℓ_1-

and ℓ_2-norm MKL. *CoMIK* accounts for any class imbalance by proportionately up-weighting the misclassification cost for the minority class as proposed in (Elkan, 2001). All parameters and the range of values tested for them are given in Table 4.4.1. Of these, #Clusters, σ and SVM-cost are optimized by cross-validation while other parameters, namely segment-size, oligomer length and maximum distance, are assigned fixed values for each individual run. We used the same segment-size for the *shifted* and the *non-shifted* cases. The best performing set of parameter values obtained from the inner CV-folds was used to re-train the model using the complete training data and make predictions on the unseen test set of examples per outer CV-fold. We report the area under the receiver operating characteristic (ROC) curve (AUC) for predictions on this held-out test set averaged over the five outer folds.

We compare our performance on the simulated data set to that of KIRMES (Kernel-based Identification of Regulatory Modules in Euchromatic Sequences) (Schultheiss et al., 2009). The approach taken by KIRMES is as follows. Consider a set of motifs representing TFBSs, and a positive and negative set of genomic sequences (corresponding to some task) given. Then, a motif finding step can be performed a *priori* to obtain the match-positions of each motif in all the sequences. Each of the positive and negative sequences is then represented instead by a collection of subsequences. These subsequences are fixed size windows extracted around the best match-positions of the given motifs in the individual sequences. Subsequences corresponding to a particular motif from all sequences are compared to each other using a variant of WDKS. This variant supplements the WDKS kernel (cf. Section 2.3.3) with conservation information for the sequences (Schultheiss et al., 2009). This procedure results in as many kernels as the number of motifs. This ensemble of kernels is then used for classification. The remaining parts of the sequences—those not extracted—are neglected. KIRMES was shown to perform well on gene sets derived from microarray experiments for identifying loss or gain of gene function (Schultheiss et al., 2009).

Figure 4.5.1 ***(following page)*****:** Distance-centric and K-mer-centric visualizations of features for the simulated data set. The distance-centric visualization in the top-panel shows 2-mer pairs that were assigned the highest positive and negative weights at each distance value corresponding to a sub-kernel that was assigned the highest weight upon MKL. For easy viewing, the K-mer-pairs at odd distances are placed on the outside and the even distances, inside. Horizontal axis: weights, vertical axis: distances between 2-mer pairs (#basepairs). The bottom-panel shows the K-mer-centric visualization. Refer to Section 4.2.4 for details on the K-mer centric visualization.


'GA' repeat at 10 nt
Highest scoring motifs for kernel ranked 1
Distances (in basepairs)
Weights
Odd distances, $d = \{2N_0 + 1\}$
Even distances, $d = \{2N_0\}$
Even distances, $d = \{2N_0\}$
Odd distances, $d = \{2N_0 + 1\}$


2-mers
2-mers
GA|GA

4.5 Results

We present results of computational experiments on simulated DNA sequences, yeast core promoters and 5C data. We also demonstrate how *CoMIK* can be used to not just determine features important for classification, but also delineate them at the segment level. This can be done for all candidate sequences including test sequences.

Simulated data set

For this data set, while KIRMES achieves an AUC of 0.9432, *CoMIK* attains near-perfect classification, AUC 0.9960 ± 0.003. We surmise that the superior performance of *CoMIK* is due to the sequences containing the dinucleotide repeat motif '`GA`' (see Table 4.3.1) This motif may not be captured well at the motif-finding stage and thus affects KIRMES' prediction performance.

We provide visualizations of features from the run that achieved the best performance with oligomer-length 2, segment-size 70nt, ℓ_1-norm MKL in Figures 4.5.1. In the top panel, is the 'AMPD' visualization of the SVM weight vector and the bottom panel, the K-mer-centric visualization. While the K-mer-centric view clearly indicates `GA`'s important role, the distance-centric visualization shows that it could be periodic. Experiments using different segment-sizes can easily uncover the fact that they are spread throughout the sequences. Figure 4.5.2 visualizes the 70nt-long

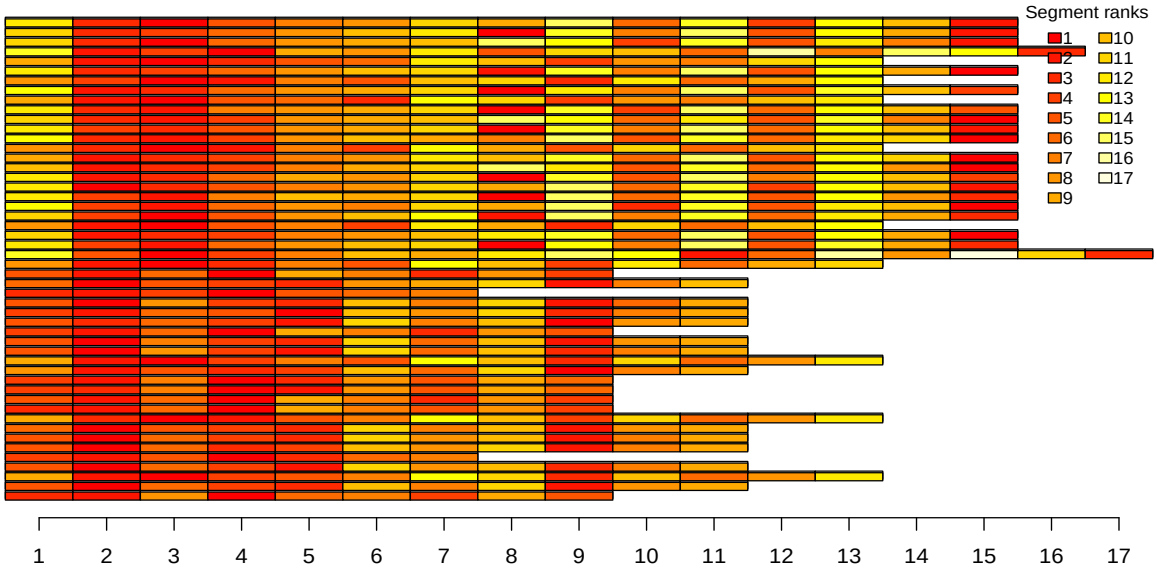


Figure 4.5.2: Bar plot with intensities to visualize importance of segments of 50 test sequences from the simulated data set. The sequences are arranged along the vertical axis, and the segments along the horizontal axis (Numbers represent segment-IDs). Among these 50 sequences, the longest sequence has a total of 17 segments (9 *non-shifted* and 8 *shifted* segments). For every sequence, *non-shifted* segments are shown first, followed by its *shifted* segments.

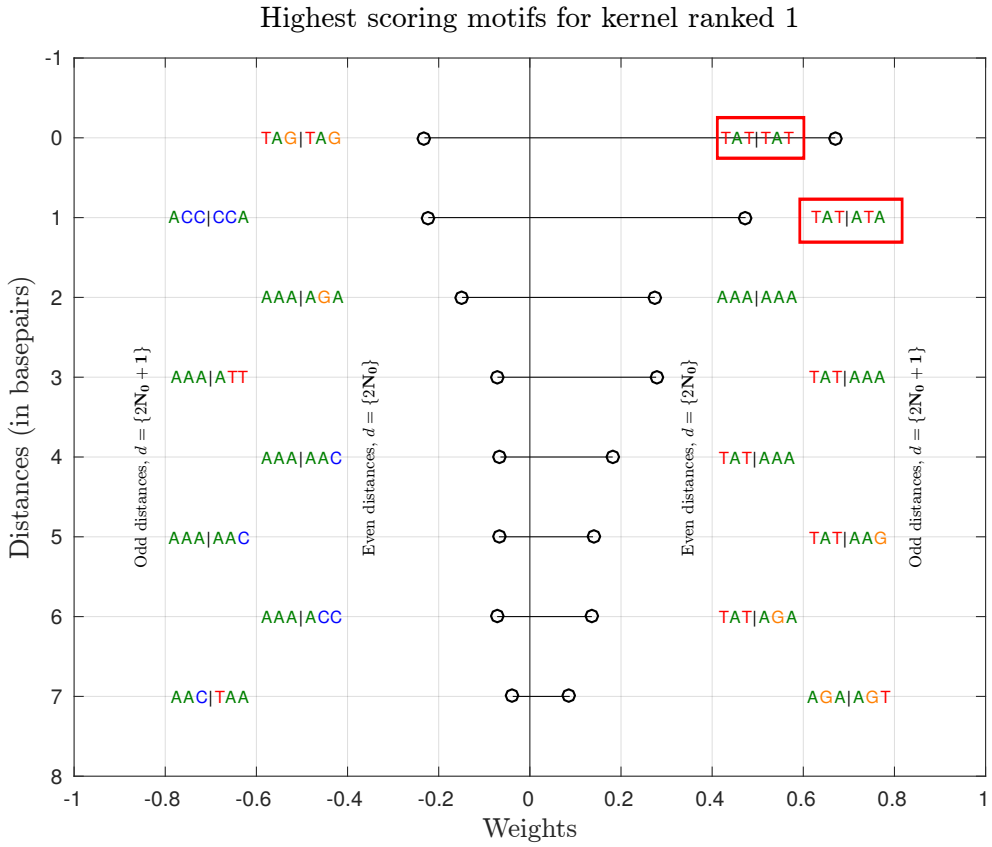
segments of 50 out of the 200 test sequences horizontally. For each sequence, the *non-*

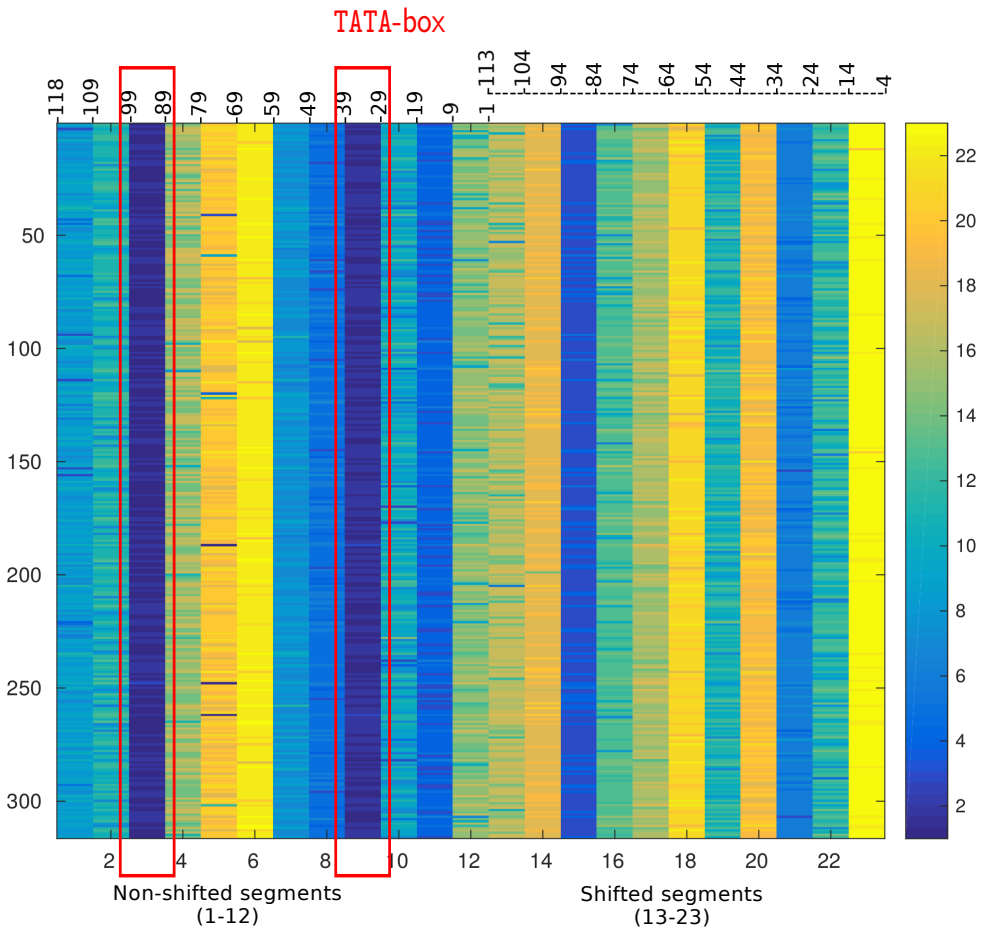
shifted segments are followed by its shifted segments. Per sequence, the higher-ranked segments would be the ones where the features are located.

Yeast

CoMIK achieved an AUC of 0.9459 ± 0.029 on this data set with segment-size 10nt, oligomer-length 3 and ℓ_1-norm MKL. Furthermore, the most important features represent motifs known as important for classification. We visualize the 3-mer pairs deemed important by *CoMIK* for this classification in Figure 4.5.3, left panel. The right panel here visualizes the sequences and their ranked segments as a heatmap. The 316 sequences are arranged vertically from top to bottom, and their segments horizontally. For the 118nt-long sequences in this data set, the segment-size of 10nt lead to 12 *non-shifted* and 11 *shifted* segments, and are arranged in that order. Thus, the coordinates for the *non-shifted* and *shifted* segments in the sequence are as marked on the top of the heatmap. We observe that segments 3 and 9, i.e. regions $[-98,-89]$ and $[-38,-29]$ happen to be ranked first consistently. Segments 15 and 21 are the best-ranked shifted segments also corresponding to the same genomic window. And, indeed, Lubliner et al. report that the main TSS lay at position -30 and that the regions $[-118,-99]$ and $[-98,-69]$ hold important features which upon mutations greatly reduced expression (Lubliner et al., 2015). In the left panel, the top-ranked kernel shows `TATA`-like elements to be important for classification. Furthermore, among the features reported by other kernels in the collection (not shown), *CoMIK* rightly identifies `T/C`-rich K-mers to be enriched among the positive sequences as against `G/C`-rich K-mers which are also reported in Supplementary Figure 4 in (Lubliner et al., 2015).

Figure 4.5.3 ***(following page)*:** Distance-centric visualization of features (top) and visualization of weights assigned to segments per sequence for the yeast data set (bottom). As in Figure 4.5.1, the top panel shows 3-mer pairs that were assigned the highest positive and negative weights at each distance value corresponding to the sub-kernel with the highest weight among all sub-kernels in the collection. For legibility, the K-mer-pairs at odd distances are placed on the outside and the even distances, inside. Horizontal axis: weights, vertical axis: distances between 3-mer pairs (#basepairs). The bottom panel shows all sequences in the data set (training as well as test) as segments: Segment rankings based on the weights assigned to the various segments are visualized as a heatmap. The rank-to-color mapping is as shown in the colorbar on the extreme right. Since all sequences in this data set are 118nt-long, we have 12 *non-shifted* segments and 11 *shifted* segments. Per sequence, non-shifted segments are shown first, then the shifted segments.


Highest scoring motifs for kernel ranked 1
Distances (in basepairs)
Weights
Odd distances, $d = \{2N_0 + 1\}$
Even distances, $d = \{2N_0\}$
Even distances, $d = \{2N_0\}$
Odd distances, $d = \{2N_0 + 1\}$
TAG|TAG
ACC|CCA
AAA|AGA
AAA|ATT
AAA|AAC
AAA|AAC
AAA|ACC
AAC|TAA
TAT|TAT
TAT|ATA
AAA|AAA
TAT|AAA
TAT|AAA
TAT|AAG
TAT|AGA
AGA|AGT


TATA-box
Non-shifted segments
(1-12)
Shifted segments
(13-23)

5C data set

Performances of *CoMIK* on the three cell lines are given in Table 4.5.1. For comparison with our approach described in the earlier chapter, we report the performances with oligomer length 3 from Table 3.6.1 in Chapter 3. We observe that in experiments using 3-mers and with segment-size 50, *CoMIK* already achieves better or *at par* performance. Furthermore, *CoMIK*'s additional ability to identify important portions in the individual sequences can give novel insights.

Table 4.5.1: Performance of *CoMIK* on 5C data set: Test AUC values (mean±s.d.) for *region* 0 in three cell lines. The approach presented in Chapter 3 is referred to as (Nikumbh and Pfeifer, 2017).

Method↓/Cell lines→	GM12878	K562	HeLa-S3
(Nikumbh and Pfeifer, 2017)	0.7417 ± 0.059	0.8163 ± 0.071	0.6914 ± 0.058
CoMIK	0.7829 ± 0.063	0.7920 ± 0.084	0.6993 ± 0.012

4.6 Discussion

This chapter presented *CoMIK*, a method for comparing variable-length sequences in a discriminative setting using conformal multi-instance kernels. We assessed the performance of *CoMIK* on three classification problems, namely, a simulated data set of variable-length sequences and two real biological data sets involving DNA sequences. This includes the 5C data set used in the earlier chapter. Together with the visualizations, we demonstrated the efficacy of *CoMIK* on all these problems.

We compared *CoMIK* to KIRMES, another approach that can handle comparison of variable-length sequences. Section 4.4 briefly notes various shortcomings of KIRMES. Performance of KIRMES heavily relies on the motif-finding step (see Section 4.4). The influence of selecting matches other than the best one is not clear. Also, choosing only one when multiple matches have (nearly) the same score seems rather arbitrary. This risks neglecting putative low-affinity or weak binding sites or *de novo* features in the sequences. In principle, although one could use the complete sequence with KIRMES by way of having the motifs spread through-out the sequence, this is again controlled by the motif-finding step. These shortcomings render KIRMES unsuitable for problems such as comparison of genomic regions from chromatin interaction experiments. In contrast, *CoMIK* uses the complete sequence by design. This helps is capturing complex relationships between features lying anywhere in the whole sequence as demonstrated by *CoMIK*'s superior performance on the simulated data set.

For the 5C data set, our earlier approach, described in Chapter 3, does not give any information on the location of the features in the restriction fragments. In comparison, for the 3 genomic regions on which we tested, *CoMIK* not only maintains the quantitative prediction performance, but also locates important features and segments within each sequence. This qualitative gain enables the possibility to learn more fine-grained insights and makes *CoMIK* relatively more advantageous. Thus, *CoMIK*'s ability to locate the segment with signal can be useful in studying the so-called structural interactions between the intervening chromatin (Sanyal et al., 2012) of the long-range interacting loci. *CoMIK*'s high prediction performance and accurate feature identification on the yeast data set demonstrates that *CoMIK* is also useful in the typical scenario involving fixed-length sequences.

CoMIK's computation time is largely governed by the clustering step and the subsequent transformation of the segments. Both of these are performed at every CV iteration, and are influenced by the choice of the segment-size. Our implementation exploits the sparsity of the ODH features for short individual segments by making use of sparse representations and computations. In general, the segment-size only affects *CoMIK's* running time. However, for scenarios like the discussed yeast problem, shorter segments may be preferred. In the clustering step, the buckshot heuristic is oblivious to the imbalance in the data. This could be replaced with stratified sampling for buckshot clustering. For scenarios wherein positional information is more important, kernels like the WDS (Rätsch et al., 2005) or the oligo kernel (Meinicke et al., 2004) may be more suitable as base kernels.

5

Pipeline for End-to-End Analysis of Chromatin Interaction Data

This chapter describes the pipeline I developed as part of work done towards analyzing HiChIP data for different conditions and tumor samples in Ewing sarcoma (EWS) cells. This project is done in collaboration with Eleni Tomazou, Andre Rendeiro, Nico Pfeifer and Christoph Bock. Eleni was responsible for performing all the biological experiments, while I and Andre shared the joint responsibility of performing the computational analyses with guidance from other members. In particular, my prime responsibility was development of this pipeline, while downstream bioinformatics analysis being Andre's responsibility. All authors took part in the interpretation of the results. The project manuscript is under preparation, and the pipeline is under active development.

DESIGNING bioinformatics workflows is a craft that requires putting together many tools to solve different intermediate computational problems. These tools have various parameters that need to be set. Choosing parameter values based on misunderstandings can result in downstream bioinformatics analyses being misconstrued leading to flawed conclusions. An example that was recently reported is the misunderstanding of the parameter '`-max_target_seqs`' for NCBI BLAST (Altschul et al., 1990; Shah et al., 2018). Another example is the commonly misread *perplexity* parameter of t-SNE (t-Distributed Stochastic Neighborhood Embedding), a dimensionality

reduction and visualization technique, that is difficult to interpret (van der Maaten and Hinton, 2008; Wattenberg et al., 2016).

5.1 pHDee: Processing HiChIP/Hi-C Data From End-to-End

As discussed in Chapter 2, chromatin interaction experiment data can be noisy. Using data from these experiments to understand the 3D organizational principles of chromosomes requires data manipulation at different levels such as significant point-to-point interactions, loops, TADs, and A/B compartments. Obtaining each of these from raw experimental data involves solving many computational problems on the way. These are active research problems, and have multiple choices of tools/approaches available for solving them. As a result, when analyzing chromatin interaction data, one should have a good understanding of all the intermediate tools, their parameters and the consequences when they take different values. Ability to sift through all the tools and their parameters at once can facilitate focusing on the downstream analyses.

We describe here such a pipeline we implemented for end-to-end analysis of chromatin interaction data. We call the pipeline, '`pHDee`'–processing HiChIP Data from end-to-end. It can also be used for analyzing Hi-C data. The pipeline uses different tools as its constituent modules to perform the various functions. The workflow of the pipeline is illustrated in Figure 5.1.1. We describe the workflow next.

1. **Raw data to valid interactions.** We split paired-BAM files into FASTQ files corresponding to the two mates in each paired-end (PE) read. These are then fed into `HiC-Pro` (Servant et al., 2015) which is an independent pipeline by itself. As part of this module, `HiC-Pro` performs the following tasks: (a) genome mapping, (b) fragment assignment, (c) identification of valid ligation products (interaction pairs).(cf. pre-processing step 3, Subsection 2.1.4, Chapter 2)

2. **Normalization of interaction frequencies.** `HiC-Pro` provides a fast implementation of iterative correction and eigenvalue decomposition (ICED) for normalization of chromatin interaction frequencies (Imakaev et al., 2012) (cf. Chapter 2, Section 2.1.4 for more details).

3. **Visualizing valid interactions as a 2D matrix.** The valid interactions between genomic regions are visualized as a 2D matrix, also called interaction or contact matrix (cf. Chapter 2, section 2.1.4). Using Juicebox, one can visualize not just the raw or the normalized interaction matrix, but also (a) the

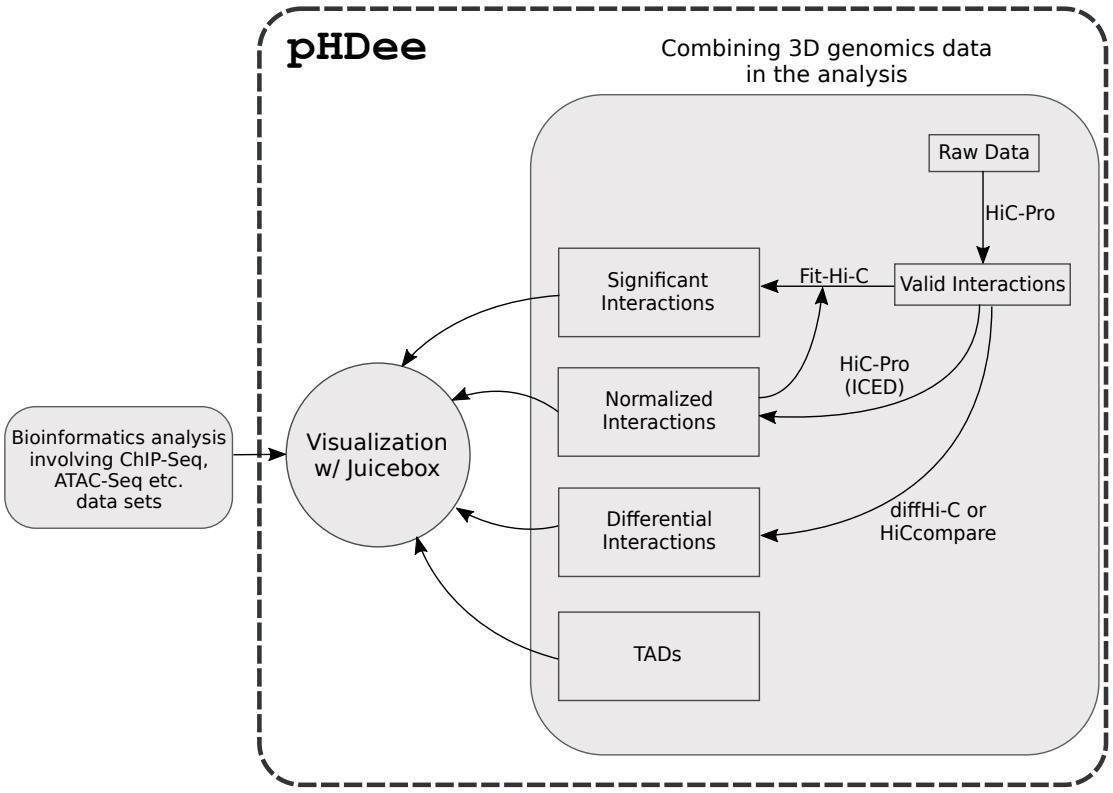


Figure 5.1.1: Different tasks of the workflow of pHDee are shown.

difference of two interaction matrices, and (b) superimposed tracks of additional features along the genome. These include 1D (e.g., ChIP-seq data) and 2D features (e.g., loops, TADs). With `Juicebox`, similar to the zooming-in and -out facility of Google Maps, one can zoom into different regions of a single contact matrix using the mouse-wheel. This enables easy exploration of the contact matrix at different resolutions. By coupling one contact matrix with another, this feature can facilitate seamless comparisons of contact matrices at different resolutions. `Juicebox` is available as a standalone Java application, or also via the web-browser with `Juicebox.js` (Robinson et al., 2018). Therefore, visualizing chromatin interaction data using `Juicebox` is very easy.

4. **Valid interactions to significant point-to-point interactions.** We use another popular tool `Fit-Hi-C` (Ay et al., 2014) to identify statistically significant point-to-point interactions. We refer to Section 2.1.4, Chapter 2 for more details about `Fit-Hi-C`.

5. **Calling significant differential interactions among two conditions.** It is of interest to compare the 3D architectures of cells in two biological conditions, e.g., tumor vs. normal cells or doxycycline-treated vs. untreated cells. Currently, there exist three tools for calling differential interactions: `diffHiC` (Lun and Smyth, 2015), `HiCcompare` (Stansfield et al., 2018) and FIND (Djekidel

et al., 2018). We make use of `HiCcompare` and `diffHiC`.

`HiCcompare` jointly normalizes the Hi-C data sets to be compared. Taking a distance-centric approach for comparison, `HiCcompare` jointly represents the interaction frequencies in two Hi-C data sets as a function of the distances between the interacting regions. The authors call this the MD plot where D=distance (measured in units of resolution), and M=minus (representing the difference between the interaction frequencies at a given distance in the two matrices). The data sets are then jointly normalized using loess regression. The normalized contact counts are then used for comparing the two Hi-C data sets. Finally, `HiCcompare` outputs the set of differential interactions after performing multiple testing correction.

To obtain a robust set of differential interactions, we added another complementary module to identify differential interactions with `diffHiC` (Lun and Smyth, 2015). `diffHiC` uses statistical approaches from the edgeR package (McCarthy et al., 2012) for eliminating biases between data sets and comparing them to identify differential interactions. The set of differential interactions are assigned p-values using the Benjamini-Hochberg method (Benjamini and Hochberg, 1995) for multiple testing correction. While `diffHiC` can detect differential interactions complementary to those detected by `HiCcompare`, `HiCcompare` is reported to handle the between-data set biases better than `diffHiC` (Stansfield et al., 2018).

Additional features of `pHDee`. With `pHDee`, the user just needs to edit a single umbrella *config* file to setup the tools that should be used and their parameters. Different *runs* of `pHDee` are saved separately, clearly noting the choices of tools and the parameter values for the *run*. A log file per *run* of `pHDee` provides information on the timestamped progress of the run. Since, the output from `pHDee` will be used for further bioinformatics analysis, observing how different parameter values affect the result can be a lot easier.

Admittedly, the 3D-genomics field is still in its nascent stage. There is a lack of an 'agreed upon' standard file format for representation and a tool/approach for different tasks. Therefore, there may arise a need to add additional modules to the above workflow in the future. The modular framework of `pHDee` allows for an easy integration of additional modules.

Implementation Details of `pHDee`

`pHDee` itself is implemented in Python. It is essentially a wrapper that puts together using different software tools in one place. These different softwares need not

necessarily be written in Python. This is especially helpful in the current scenario wherein a plethora of newly developed alternative solutions exist for performing each 3D-genomics task noted in Figure 5.1.1. For example, Zufferey et al. (2018) review and benchmark 22 different software tools for identifying TADs. `pHDee` uses a single umbrella configuration file to setup all the tools and their parameters for a single *run*.

While `pHDee` adapts the scripts already provided by the tools for converting between different file formats, when not available, it provides them. These include conversion from one file format to another (e.g., `HiC-Pro` to `Fit-Hi-C`, `HiC-Pro` to `diffHiC`/`HiCcompare`), and some other miscellaneous tasks such as combining contact matrices before visualization. With regards to visualization with the tool `Juicebox` (Durand et al., 2016), scripts are provided for converting the list of valid interactions to the `.hic` format usable with `Juicebox`.

Suitability of a tool/approach for a task depends on various factors. Some of these stem from the underlying biology while others solely from the modeling approach used by the tool. Thus, taking these factors into account, a bioinformatician should be able to select a suitable tool for the task at hand. To this end, one can add a new module to `pHDee` to include the selected tool in the workflow. At present, adding a new module corresponding to a tool amounts to maintaining proper file formats and writing a wrapper function for it.

Comparison With Other Tools

HiCExplorer is a recently developed Python-based pipeline that enables certain ways of processing and analyzing chromatin interaction data (Ramírez et al., 2018). HiCExplorer implements functions for pre-processing raw interaction data to obtain contact matrices, compare different matrices (by a simple difference operation) or perform a correlation analysis of contact matrices. Additionally, it can plot static contact matrices, and TADs with 1D feature tracks. For dynamic visualization, it can convert the files to the *cooler* format (Abdennur et al., 2018) that can be used with HiGlass (Kerpedjiev et al., 2018), another dynamic contact map visualization tool[1]. There is also a web-server/web-based version of HiCExplorer[2] made available via Galaxy (Wolff et al., 2018). This facilitates usage of HiCExlorer by biologists and biomedical researchers who are often non-programmers.

Thus, while `pHDee` has a similar objective—providing an end-to-end solution—in contrast to HiCExplorer, it aims to provide the bioinformatician with two important advantages. First, freedom to use tools/approaches not implemented in Python, and, second, seamless self-addition of a module to incorporate any new tool/approach as desired. This makes integration of newer tools with `pHDee` very easy. For example, re-

[1] http://higlass.io
[2] https://hicexplorer.usegalaxy.eu/

cently developed approaches for comparing contact matrices of replicate experiments or different cell lines include `HiCRep` (Yang et al., 2017a) and `GenomeDISCO` (Ursu et al., 2018). These can be easily integrated into `pHDee` by the end user. In fact, `pHDee` can already incorporate individual functionalities from HiCExplorer as modules. Also, currently, HiCExplorer does not support any functionality w.r.t. identifying differential interactions.

We implemented and used `pHDee` for analyzing data from HiChIP experiments performed on EWS cells. We briefly discuss HiChIP and the role of `pHDee` in analyzing HiChIP data in EWS cells in the following sections of this chapter.

5.2 Analysis of Genome Architecture Changes in EWS Cells Using HiChIP

Since few somatic mutations are observed in EWS as compared to other cancers (Lawrence et al., 2013), researchers hypothesized that tumor heterogeneity could be explainable by the epigenetic heterogeneity. Indeed, studies have shown that the fusion oncoprotein EWS-FLI1 dynamically reprograms the epigenome. Specifically: (a) Riggi et al. (2014) show that divergent chromatin remodeling mechanisms of EWS-FLI1 can activate or repress enhancers in Ewing sarcoma; (b) Tomazou et al. (2015) have shown that EWS-FLI1 reprograms the epigenome at promoters and (super) enhancers, especially the H3K27 acetylation mark; and, (c) Sheffield et al. (2017) showed that heterogeneity in DNA methylation profiles in tumor samples can explain the tumor heterogeneity and the clinical diversity observed. In particular, Tomazou et al. (2015) have mapped the epigenome of EWS cells using the cell line A673 which is the standard model of system biology in Ewing sarcoma cells. Specifically, DNA methylation (using Whole Genome Bisulfite Sequencing (WGBS) and Reduced Representation Bisulfite Sequencing (RRBS)), seven histone modifications (using ChIP-seq), RNA levels (using RNA-seq) and open chromatin states (using ATAC-seq) have been mapped in cells corresponding to two conditions. These are up- and (doxycycline-induced) down-regulation of EWS-FLI1 in A673 cells.

As a next step, we hypothesized that analyzing the architectural changes in Ewing sarcoma cells after down-regulation of EWS-FLI1, and, in tumor samples can help characterize the epigenetic reprogramming better. It could possibly also lead to insights of therapeutic value. We hypothesized using a protein-mediated chromatin interaction detection technique, such as HiChIP, would be ideal in this scenario. Specifically, HiChIP was performed on A673 cells before and after treating them with doxycycline. As noted above, this treatment knocks out the fusion oncogene EWS-FLI1. Two replicate HiChIP experiments are performed with five IPs, namely,

H3K4me2, H3K4me3, CTCF, SMC1 and H3K27ac. Additionally, HiChIP was also performed on two primary tumor samples. Two IPs were used in this case: H3K27ac and CTCF.

At the initial stage, `pHDee` was run on HiChIP data of the individual IPs separately. This helped analyze the quality of the contact libraries per IP and also estimate the appropriate bin size of the contact matrices. For this, we used the criterion proposed by Rao et al. (2014) which suggests that bin size as appropriate which results in 80% bins of the contact matrix to have at least 1000 interactions. The libraries were sequenced to obtain about 200M reads for each IP (counting both replicates). Additionally, we looked at the following for the contact libraries generated: (a) genome coverage of different IPs individually and their combinations (see Figure 5.2.1); and, (b) per IP, the global percentage of interactions overlapping with enhancers or promoters or other regions of the genome (Figure 5.2.2 and 5.2.3).

We obtained (a) as follows. We used `genomecov` from bedtools to obtain the genome-wide coverage for each IP individually and in different subsets. Initially, the IP H3K27ac had two versions determined by its suppliers, Abcam and Diagenode. This made the total IP count to six. Thus, there are 2^6 possible subsets. 63 of these (excluding the empty set) have been arranged along the x-axis. The y-axis shows the percentage of genome covered by each subset with at least 1, 5 or 10 reads. These are separated into three panels. The numbers on top of the bars denote the rank assigned to each subset based on the percentage of genome covered. Specifically, all combinations of two IPs (IDs 7-21) and five IPs (IDs 57-62) are marked with a square. This facilitated choosing the IP-pair CTCF + H3K27ac_Abcam (ID: 11) as a good combination that captured a large fraction of the observed set of interactions at the given sequencing depth.

In case of (b), enhancers (E) and promoters (P) were determined as follows. Stringent criterion: From Tomazou et al. (2015)'s ChIP-seq data, all H3K4me peaks that are present in at least two of the four biological replicates were merged into one set of regions and only those that overlapped or lay within 1kb of the nearest Ensembl-annotated TSS were defined as promoters. This criterion was made lenient by combining H3K4me peaks present in at least one of the biological replicates and enforcing no further restrictions. Similar strategy was used for identifying a stringent and lenient set of enhancers, using H3K27ac sites instead of H3K4me. Figure 5.2.3 and 5.2.2 show percentages of interactions that are EP, PP, EE, ED, PD, and DD interactions. Here, D = distant denoted any other region apart from enhancers and promoters.

Further tasks included, for bin sizes 1M, 500K, 250K, 150K, 100K, 50K, 25K obtaining (a) the significant interactions per IP and all IPs pooled; (b) differential

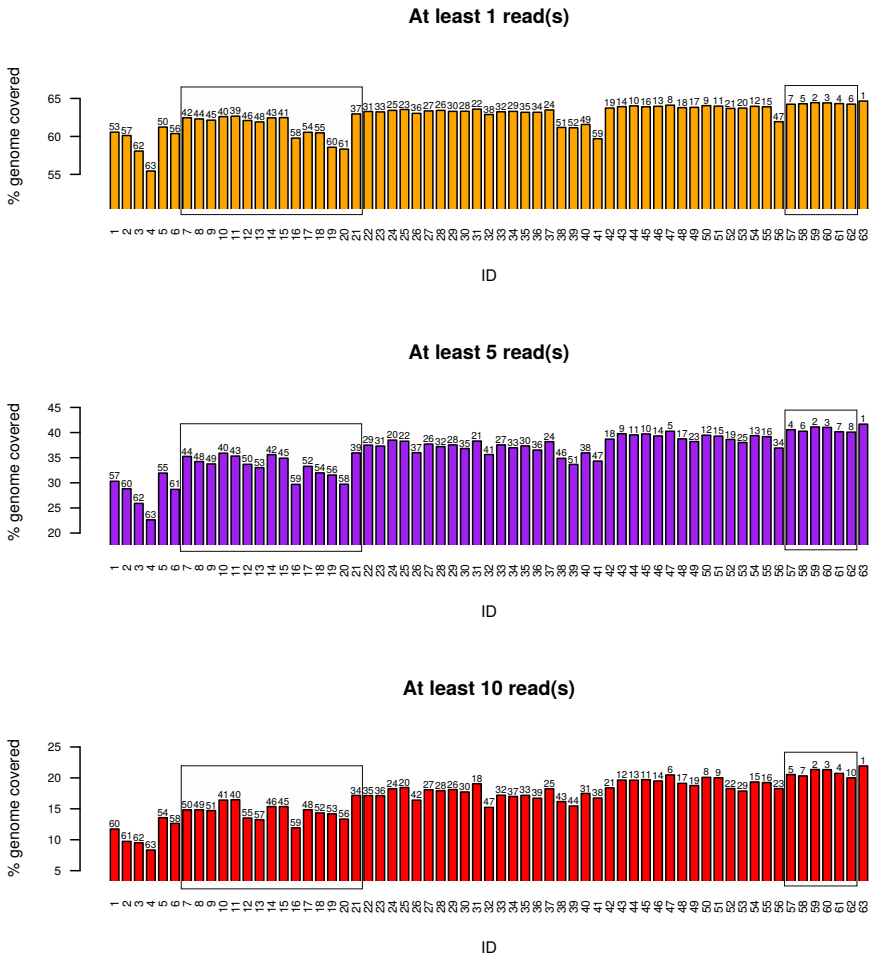


Figure 5.2.1: Figure shows the percentage genomic coverage of different IPs individually and in combinations.

interactions per IP and all IPs pooled using `diffHiC` and `HiCcompare`. This was also performed for the HiChIP data for two primary tumor samples.

Thus `pHDee` helped facilitating a seamless data-intensive downstream bioinformatics analysis.

5.3 Discussion

We developed `pHDee`, an easy-to-use, plug-and-play pipeline for HiChIP or Hi-C data analysis. The pipeline collates handling of different tools for different tasks at one place. This enables the user to concentrate on the downstream bioinformatics analysis. Adding new modules to pHDee is easy. The complete source code is made available at https://github.molgen.mpg.de/snikumbh/pHDee.

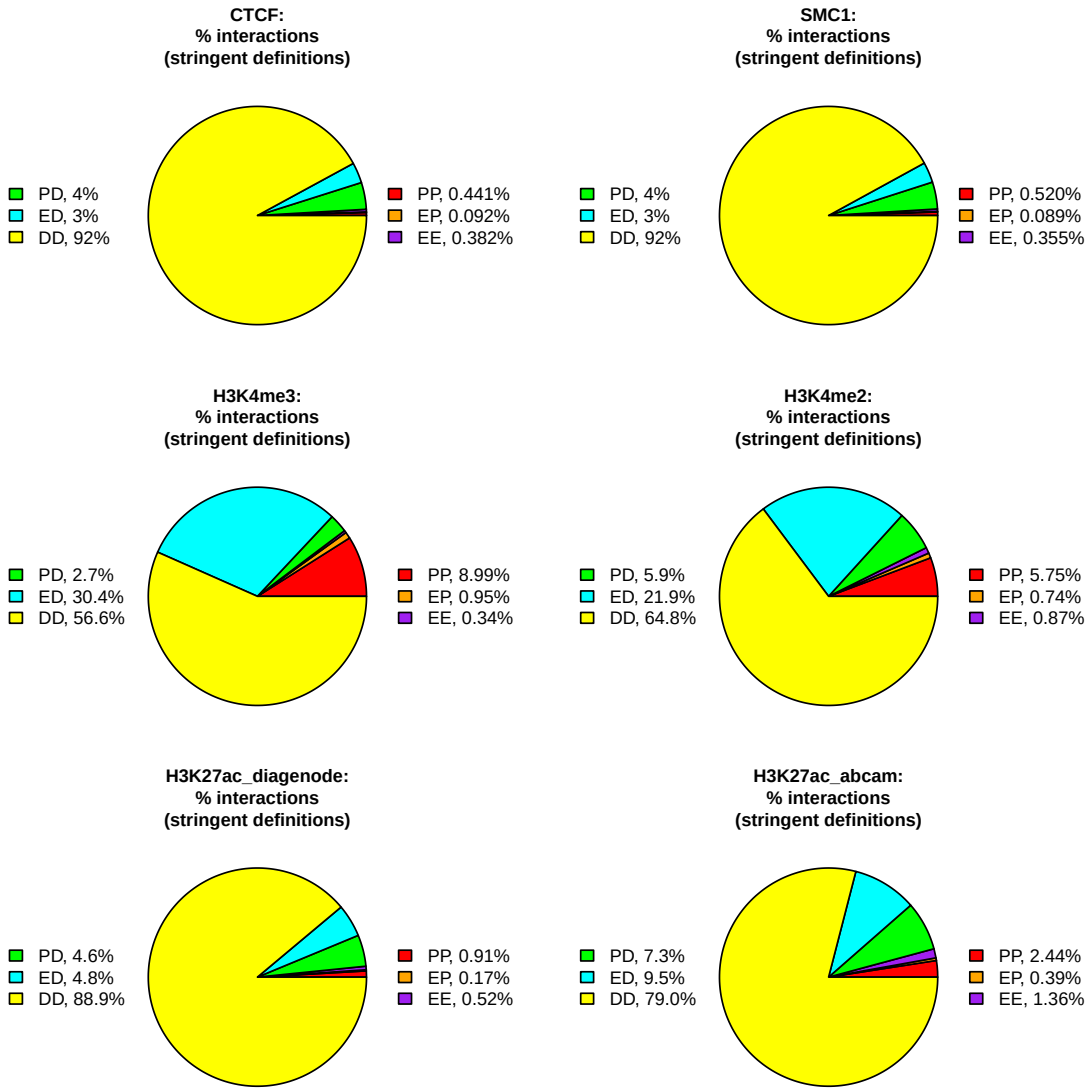


Figure 5.2.2: Figure shows the global percentage of interactions involving loci overlapping an E or P or otherwise. Overlaps are defined comparatively stringently (see text).

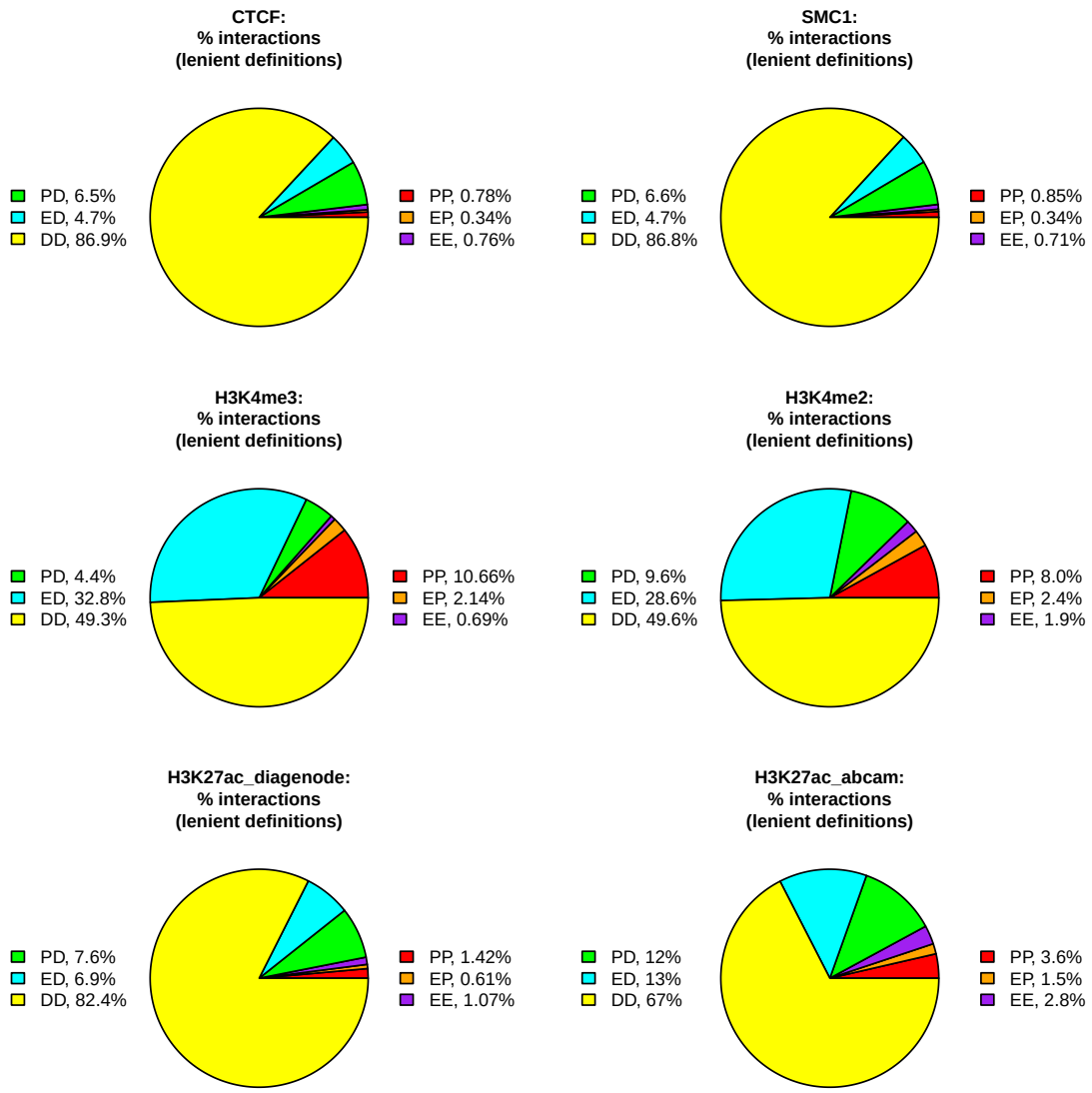


Figure 5.2.3: Lenient definitions used for promoters and enhancers. Figure shows the global percentage of interactions involving loci overlapping an E or P or otherwise.

6

Perspective

It has long been known that regulatory elements like enhancers in the genome are able to regulate genes that are located distantly on the linear DNA sequence. Such long-range interactions between regions in the genome are made possible due to the compact 3D organization of chromatin inside the nucleus. Molecular biology techniques, such as 3C and its derivatives, that can probe this 3D organization were invented during the last decade. In parallel, computational approaches are being developed that help study the role of different features contributing to this 3D organization. This thesis outlines our efforts in developing interpretable machine learning-based approaches for this task. This chapter serves to conclude this thesis. I first summarize the contributions made in this thesis—both biological and methodological. Finally, I lead the discussion into the broader perspectives and possible future directions.

6.1 Conclusions

Chapter 3 outlines our SVM-based approach for prediction of long-range chromatin interactions using only sequence features. In it, we specifically sought to establish the extent to which the genetic sequence at a genomic locus played a role in identifying potential long-range interaction partners of a particular genomic locus of interest.We built locus-specific models using string kernels to compare genomic fragments, and coupled it together with support vector machines for classification. We were able to

show that the genetic sequence information at a genomic locus can help discern if it interacts with the locus of interest or not. Our computational experiments were one of the first to suggest a potential general role for short tandem repeat (STR) sequences in genome organization (Nikumbh and Pfeifer, 2017). We also developed novel ways for visualizing the important sequence features for this prediction problem that aided in establishing this hypothesis. While such tandem repeats have been implicated in neurological diseases like the Huntington disease since the last decade, several recent studies have either proposed or shown evidence of general regulatory roles for STRs (Bagshaw, 2017), including in cancer (Gymrek, 2017; Gymrek et al., 2016), at borders of large chromatin domains such as TADs (Darrow et al., 2016; Mourad and Cuvier, 2016).

Although the ODH feature representation allowed us to compare the variable-length restriction fragments, the sequence features identified as important for classification cannot be traced back to their location in the individual sequences. We addressed this limitation next.

Chapter 4 presents *CoMIK*, a string kernel-based method we developed to enable comparison of variable-length sequences in a supervised ML scenario. This comparison can be free of using *absolute* positional information which is a characteristic of the existing methods. *CoMIK* achieves this by modeling the problem in a way that is borrowed from the computer vision or natural language processing domain, but so far novel in computational biology. For example, analogous to a document (paragraph) being composed of paragraphs (sentences), *CoMIK* treats every sequence as a set of its segments. It then characterizes a sequence by the features in all its segments and uses this information for its classification, i.e., assigning a class label to the sequence. We demonstrated that *CoMIK* can accurately classify sequences using this approach. It also helps in assessing the contributions of the segments of each sequence towards classification (Nikumbh et al., 2017b). This, in a way, combines the best of both worlds—identification of important features and locating segments within the whole sequence that hold these important features.

An overarching, long-term goal here is to analyze long-range interactions using *CoMIK*. Compared to other approaches (Roy et al., 2015; Whalen et al., 2016; Yang et al., 2017b), when comparing genomic fragments reported from chromatin interaction experiments, this method can provide *de novo*, fine-grained insights into the sequence drivers of such long-range interactions. This can enable studying the role of the flanking regions or the intervening chromatin in the so-called 'structural' interactions (Hughes et al., 2014; Sanyal et al., 2012).

This approach can have some limitations. If there are just too many or too long sequences, breaking them down into shorter segments can pose a computationally intensive problem (cf. Section 4.2.3)

In Chapter 5, the focus is shifted from developing and applying interpretable machine learning methods to enabling easy manipulation of chromatin conformation data in order to facilitate their analysis together with other facets of cellular biology. To this end, Chapter 5 delineates our pipeline, `pHDee`, for end-to-end processing of HiChIP/Hi-C data. Bioinformatics analyses often involve manipulating and analyzing large amounts of data from different molecular biology experiments. For example, when studying differences in cell-types or tissues, in order to get a more complete view, analyses involve data from several experiments. This entails ploughing through data described using different formats by various tools. In this scenario, `pHDee` enables focusing on the downstream, data-intensive bioinformatics analyses. We have used the pipeline for processing data from two replicates of HiChIP experiments profiling long-range interactions mediated by selected histone modifications and architectural proteins in a Ewing sarcoma cell line and primary EWS tumor samples. Processing this huge volume of chromatin conformation data using `pHDee` facilitates performing downstream analysis. This downstream analysis additionally involved handling of genome-wide ChIP-seq data of different histone modifications, open chromatin regions (ATAC-seq) and RNA-seq data in the same cell line. As research in 3D-genomics progresses, it is expected that approaches for tasks and the different file formats will be more standardized reducing the burden of managing many of these as is the case today.

6.2 Future Directions

On the methodology side, we already noted that kernels should be positive definite. In *CoMIK*, the conformal transformation applied to each multi-instance kernel is not guaranteed to yield a positive definite kernel matrix due to the Gaussian nature of the transformation. This is an open problem (Feragen and Hauberg, 2016). While we did not encounter such a scenario in our experiments on data sets reported in Chapter 4, as a workaround, we propose removing the negative eigenvalues and making the kernel matrices positive definite (cf. Subsection 2.3.4). Furthermore, from the point of view of practical applicability of *CoMIK* for large-scale data sets such as the genome-wide conformation data, one will have to devise ways to handle the computation of the intermediate large similarity matrix.

From the perspective of biology, as already noted in Chapter 4, development of *CoMIK* was mainly motivated by two aspects. First, our work in Chapter 3 which we already discussed in the earlier section. Second, the prevalence of different definitions of promoters used across studies that yield promoter sequences of different lengths, and also the notion of focused and dispersed core promoters (Butler and Kadonaga, 2002; Juven-Gershon et al., 2008). It will be interesting to process promoter sequences

using *CoMIK* to gain insights into the core promoter sequences and their transcription initiation modes.

Recall the 'Bickmore hypothesis' we discussed in Chapter 3, Section 3.7. Briefly, it states that there can be different mechanisms that help establish interactions between distant genomic regions, for instance, physical contact via single large loop or many mini loops, or interaction via diffusion but no physical contact. Intriguingly, there is a recent line of work suggesting liquid-liquid phase separation as an emerging model that can explain these mechanisms (Hnisz et al., 2017).

Finally, I would like to conclude by making the following point. In terms of interpretability of computational models, there are always important caveats that ought to be taken into account. For instance, when predicting long-range interactions between genomic regions, the available (or used) ground truth for this task clearly defines what the machine actually learns. To this end, it is very important to distinguish between (a) models predicting point-to-point and looping interactions; (b) interactions between known or annotated regulatory regions and those involving other genomic regions, etc. Often, this distinction can be hazy due to the experimental design itself or due to noise in the measurements. We note the well-known aphorism by the British Statistician George E. P. Box that addresses this: "*All models are wrong, some are useful*".

Bibliography

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

Nezar Abdennur, Anton Goloborodko, Maxim Imakaev, and Leonid Mirny. mirnylab/cooler: v0.7.10, May 2018. URL https://doi.org/10.5281/zenodo.1243296.

Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215 (3):403–410, 1990.

Ferhat Ay and William Noble. Analysis methods for studying the 3D architecture of the genome. *Genome Biology*, 16(1):183, 2015. ISSN 1474-760X. doi: 10.1186/s13059-015-0745-7. URL http://genomebiology.com/2015/16/1/183.

Ferhat Ay, Timothy L Bailey, and William Stafford Noble. Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. *Genome research*, 2014.

Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 6–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015424. URL http://doi.acm.org/10.1145/1015330.1015424.

Andrew T.M. Bagshaw. Functional mechanisms of microsatellite DNA in eukaryotic genomes. *Genome Biology and Evolution*, 9(9):2428–2443, 2017. doi: 10.1093/gbe/evx164. URL http://dx.doi.org/10.1093/gbe/evx164.

Robert A Beagrie, Antonio Scialdone, Markus Schueler, Dorothee CA Kraemer, Mita Chotalia, Sheila Q Xie, Mariano Barbieri, Inês de Santiago, Liron-Mark Lavitas, Miguel R Branco, et al. Complex multi-enhancer contacts captured by genome architecture mapping. *Nature*, 543(7646):519, 2017.

Houda Belaghzal, Job Dekker, and Johan H Gibcus. Hi-C 2.0: An optimized Hi-C procedure for high-resolution genome-wide mapping of chromosome conformation. *Methods*, 123:56–65, 2017.

Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

Wendy A. Bickmore. The spatial organization of the human genome. *Annual Review of Genomics and Human Genetics*, 14(1):67–84, 2013. doi: 10.1146/annurev-genom-091212-153515. URL http://dx.doi.org/10.1146/annurev-genom-091212-153515. PMID: 23875797.

Matthew B Blaschko and Thomas Hofmann. Conformal multi-instance kernels. In *NIPS 2006 Workshop on Learning to Compare Examples*, 2006.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130401. URL http://doi.acm.org/10.1145/130385.130401.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced lectures on machine learning*, pages 169–207. Springer, 2004.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

K.J. Brookes. The VNTR in complex disorders: The forgotten polymorphisms? a functional way forward? *Genomics*, 101(5):273 – 281, 2013. ISSN 0888-7543. doi: http://dx.doi.org/10.1016/j.ygeno.2013.03.003. URL http://www.sciencedirect.com/science/article/pii/S0888754313000451.

Martha L Bulyk. Computational prediction of transcription-factor binding site locations. *Genome biology*, 5(1):201, 2003.

Jennifer E.F. Butler and James T. Kadonaga. The RNA polymerase II core promoter: a key component in the regulation of gene expression. *Genes & Development*, 16 (20):2583–2592, 2002. doi: 10.1101/gad.1026202. URL http://genesdev.cshlp.org/content/16/20/2583.short.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm.

Qiang Chen, Yibin Chen, Chunjing Bian, Ryoji Fujiki, and Xiaochun Yu. TET2 promotes histone o-glcnacylation during gene transcription. *Nature*, 493(7433): 561–564, 2013.

Nathan Cope, Peter Fraser, and Christopher Eskiw. The yin and yang of chromatin spatial organization. *Genome Biology*, 11(3):204, 2010. ISSN 1465-6906. doi: 10.1186/gb-2010-11-3-204. URL http://genomebiology.com/2010/11/3/204.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 0885-6125. doi: 10.1007/BF00994018. URL http://dx.doi.org/10.1007/BF00994018.

Caelin Cubeñas-Potts and Victor G. Corces. Architectural proteins, transcription, and the three-dimensional organization of the genome. *FEBS Letters*, 589(20, Part A):2923 – 2930, 2015. ISSN 0014-5793. doi: https://doi.org/10.1016/j.febslet.2015.05.025. URL http://www.sciencedirect.com/science/article/pii/S0014579315004019. 3D Genome structure.

Katherine E Cullen, Michael P Kladde, and Mark A Seyfred. Interaction between transcription regulatory regions of prolactin chromatin. *Science*, 261(5118):203–206, 1993.

Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 318–329, New York, NY, USA, 1992. ACM. ISBN 0-89791-523-2. doi: 10.1145/133160.133214. URL http://doi.acm.org/10.1145/133160.133214.

Emily M. Darrow, Miriam H. Huntley, Olga Dudchenko, Elena K. Stamenova, Neva C. Durand, Zhuo Sun, Su-Chen Huang, Adrian L. Sanborn, Ido Machol, Muhammad Shamim, Andrew P. Seberg, Eric S. Lander, Brian P. Chadwick, and Erez Lieberman Aiden. Deletion of DXZ4 on the human inactive X chromosome alters higher-order genome architecture. *Proceedings of the National Academy of Sciences*, 113(31):E4504–E4512, 2016. doi: 10.1073/pnas.1609643113. URL http://www.pnas.org/content/113/31/E4504.abstract.

Elzo de Wit and Wouter de Laat. A decade of 3C technologies: insights into nuclear organization. *Genes & Development*, 26(1):11–24, January 2012. ISSN 1549-5477. doi: 10.1101/gad.179804.111. URL http://dx.doi.org/10.1101/gad.179804.111.

Job Dekker. Gene regulation in the third dimension. *Science*, 319(5871):1793–1794, 2008. ISSN 0036-8075. doi: 10.1126/science.1152850. URL http://science.sciencemag.org/content/319/5871/1793.

Job Dekker and Edith Heard. Structural and functional diversity of topologically associating domains. *FEBS letters*, 589(20PartA):2877–2884, 2015.

Job Dekker, Karsten Rippe, Martijn Dekker, and Nancy Kleckner. Capturing chromosome conformation. *science*, 295(5558):1306–1311, 2002.

Job Dekker, Marc A. Marti-Renom, and Leonid A. Mirny. Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. *Nat Rev Genet*, 14(6):390–403, Jun 2013. ISSN 1471-0056. URL http://dx.doi.org/10.1038/nrg3454. Review.

Job Dekker, Andrew S Belmont, Mitchell Guttman, Victor O Leshyk, John T Lis, Stavros Lomvardas, Leonid A Mirny, Clodagh C O'shea, Peter J Park, Bing Ren, et al. The 4d nucleome project. *Nature*, 549(7671):219, 2017.

Olivier Delattre, Jessica Zucman, Béatrice Plougastel, Chantal Desmaze, Thomas Melot, Martine Peter, Heinrich Kovar, Isabelle Joubert, Pieter de Jong, Guy

Rouleau, et al. Gene fusion with an ETS DNA-binding domain caused by chromosome translocation in human tumours. *Nature*, 359(6391):162, 1992.

Thomas G. Dietterich, Richard H. Lathrop, Tomas Lozano-Perez, and Arris Pharmaceutical. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, May 2012. ISSN 0028-0836. doi: 10.1038/nature11082. URL http://dx.doi.org/10.1038/nature11082.

Mohamed Nadhir Djekidel, Yang Chen, and Michael Q Zhang. FIND: differential chromatin interactions detection using a spatial poisson process. *Genome research*, 2018.

Josèe Dostie, Todd A. Richmond, Ramy A. Arnaout, Rebecca R. Selzer, William L. Lee, Tracey A. Honan, Eric D. Rubio, Anton Krumm, Justin Lamb, Chad Nusbaum, Roland D. Green, and Job Dekker. Chromosome conformation capture carbon copy (5C): A massively parallel solution for mapping interactions between genomic elements. *Genome Research*, 16(10):1299–1309, 2006. doi: 10.1101/gr.5571506. URL http://genome.cshlp.org/content/16/10/1299.abstract.

Neva C Durand, James T Robinson, Muhammad S Shamim, Ido Machol, Jill P Mesirov, Eric S Lander, and Erez Lieberman Aiden. Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom. *Cell systems*, 3(1):99–101, 2016.

Ron Edgar, Michael Domrachev, and Alex E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002. doi: 10.1093/nar/30.1.207. URL http://nar.oxfordjournals.org/content/30/1/207.abstract.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pages 973–978, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-812-5, 978-1-558-60812-2. URL http://dl.acm.org/citation.cfm?id=1642194.1642224.

Jason Ernst and Manolis Kellis. ChromHMM: Automating chromatin-state discovery and characterization. *Nat Meth*, 9(3):215–216, March 2012. ISSN 15487091. doi: 10.1038/nmeth.1906.

Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, December 2005. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1046920.1088693.

Andrew P. Feinberg and Pauline A. Callinan. The emerging science of epigenomics. *Human Molecular Genetics*, 15(suppl_1):R95–R101, 04 2006. ISSN 0964-6906. doi: 10.1093/hmg/ddl095. URL https://dx.doi.org/10.1093/hmg/ddl095.

Aasa Feragen and Søren Hauberg. Open problem: Kernel methods on manifolds and metric spaces. what is the probability of a positive definite geodesic exponential kernel? In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1647–1650, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL http://proceedings.mlr.press/v49/feragen16.html.

Darya Filippova, Rob Patro, Geet Duggal, and Carl Kingsford. Identification of alternative topological domains in chromatin. *Algorithms for Molecular Biology*, 9 (1):14, 2014.

James Fraser, Carmelo Ferrai, Andrea M Chiariello, Markus Schueler, Tiago Rito, Giovanni Laudanno, Mariano Barbieri, Benjamin L Moore, Dorothee CA Kraemer, Stuart Aitken, et al. Hierarchical folding and reorganization of chromosomes are linked to transcriptional changes in cellular differentiation. *Molecular systems biology*, 11(12):852, 2015.

J Füllgrabe, N Hajji, and B Joseph. Cracking the death code: apoptosis-related histone modifications. *Cell death and differentiation*, 17(8):1238, 2010.

Melissa J. Fullwood, Mei Hui Liu, You Fu Pan, Jun Liu, Han Xu, Yusoff Bin Mohamed, Yuriy L. Orlov, Stoyan Velkov, Andrea Ho, Poh Huay Mei, Elaine G. Y. Chew, Phillips Yao Hui Huang, Willem-Jan Welboren, Yuyuan Han, Hong Sain Ooi, Pramila N. Ariyaratne, Vinsensius B. Vega, Yanquan Luo, Peck Yean Tan, Pei Ye Choy, K. D. Senali Abayratna Wansa, Bing Zhao, Kar Sian Lim, Shi Chi Leow, Jit Sin Yow, Roy Joseph, Haixia Li, Kartiki V. Desai, Jane S. Thomsen, Yew Kok Lee, R. Krishna Murthy Karuturi, Thoreau Herve, Guillaume Bourque, Hendrik G. Stunnenberg, Xiaoan Ruan, Valere Cacheux-Rataboul, Wing-Kin Sung, Edison T. Liu, Chia-Lin Wei, Edwin Cheung, and Yijun Ruan. An oestrogen-receptor-[agr]-bound human chromatin interactome. *Nature*, 462(7269): 58–64, Nov 2009. ISSN 0028-0836. doi: 10.1038/nature08497. URL http://dx.doi.org/10.1038/nature08497.

Thomas Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003.

Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. Multi-Instance kernels. In *Proc. 19th International Conf. on Machine Learning*, pages 179–186, Massachusetts, 2002. Morgan Kaufmann.

Jeff Gauthier, Antony T Vincent, Steve J Charette, and Nicolas Derome. A brief history of bioinformatics. *Briefings in Bioinformatics*, page bby063, 2018. doi: 10.1093/bib/bby063. URL http://dx.doi.org/10.1093/bib/bby063.

Luca Giorgetti, Bryan R Lajoie, Ava C Carter, Mikael Attia, Ye Zhan, Jin Xu, Chong Jian Chen, Noam Kaplan, Howard Y Chang, Edith Heard, et al. Structural organization of the inactive x chromosome in the mouse. *Nature*, 2016.

Melissa Gymrek. A genomic view of short tandem repeats. *Current opinion in genetics & development*, 44:9–16, 2017.

Melissa Gymrek, Thomas Willems, Audrey Guilmatre, Haoyang Zeng, Barak Markus, Stoyan Georgiev, Mark J. Daly, Alkes L. Price, Jonathan K. Pritchard, Andrew J. Sharp, and Yaniv Erlich. Abundant contribution of short tandem repeats to gene expression variation in humans. *Nature Genetics*, 48:22, 2016. URL http://dx.doi.org/10.1038/ng.3461.

H. Hamada et al. Characterization of genomic poly(dT-dG).poly(dC-dA) sequences: structure, organization, and conformation. *Mol Cell Biol*, 4(12):2610–2621, Dec 1984. ISSN 0270-7306. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC369266/. 6098814[pmid].

Jennifer Harrow, Adam Frankish, Jose M. Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L. Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, If Barnes, Alexandra Bignell, Veronika Boychenko, Toby Hunt, Mike Kay, Gaurab Mukherjee, Jeena Rajan, Gloria Despacio-Reyes, Gary Saunders, Charles Steward, Rachel Harte, Michael Lin, Códric Howald, Andrea Tanzer, Thomas Derrien, Jacqueline Chrast, Nathalie Walters, Suganthi Balasubramanian, Baikang Pei, Michael Tress, Jose Manuel Rodriguez, Iakes Ezkurdia, Jeltje van Baren, Michael Brent, David Haussler, Manolis Kellis, Alfonso Valencia, Alexandre Reymond, Mark Gerstein, Roderic Guigó, and Tim J. Hubbard. GENCODE: The reference human genome annotation for the ENCODE project. *Genome Research*, 22(9):1760–1774, 2012. doi: 10.1101/gr.135350.111. URL http://genome.cshlp.org/content/22/9/1760.abstract.

Bing He, Changya Chen, Li Teng, and Kai Tan. Global view of enhancer–promoter interactome in human cells. *Proceedings of the National Academy of Sciences*, 111 (21):E2191–E2199, 2014. doi: 10.1073/pnas.1320308111. URL http://www.pnas.org/content/111/21/E2191.abstract.

Nastaran Heidari, Douglas H. Phanstiel, Chao He, Fabian Grubert, Fereshteh Jahanbani, Maya Kasowski, Michael Q. Zhang, and Michael P. Snyder. Genome-wide map of regulatory interactions in the human genome. *Genome Research*, 24(12): 1905–1917, 2014. doi: 10.1101/gr.176586.114. URL http://genome.cshlp.org/content/24/12/1905.abstract.

Ralf Herbrich and Thore Graepel. A pac-bayesian margin bound for linear classifiers: Why svms work. In *Advances in neural information processing systems*, pages 224–230, 2001.

Denes Hnisz, Krishna Shrinivas, Richard A Young, Arup K Chakraborty, and Phillip A Sharp. A phase separation model for transcriptional control. *Cell*, 169 (1):13–23, 2017.

Ming Hu, Ke Deng, Siddarth Selvaraj, Zhaohui Qin, Bing Ren, and Jun S. Liu. HiCNorm: removing biases in Hi-C data via Poisson regression. *Bioinformatics*, 28 (23):3131–3133, 2012. doi: 10.1093/bioinformatics/bts570. URL http://dx.doi.org/10.1093/bioinformatics/bts570.

Jim R. Hughes, Nigel Roberts, Simon McGowan, Deborah Hay, Eleni Giannoulatou, Magnus Lynch, Marco De Gobbi, Stephen Taylor, Richard Gibbons, and Douglas R. Higgs. Analysis of hundreds of cis-regulatory landscapes at high resolution in a

single, high-throughput experiment. *Nat Genet*, 46(2):205–212, Feb 2014. ISSN 1061-4036. URL http://dx.doi.org/10.1038/ng.2871. Technical Report.

Maxim Imakaev, Geoffrey Fudenberg, Rachel Patton McCord, Natalia Naumova, Anton Goloborodko, Bryan R Lajoie, Job Dekker, and Leonid A Mirny. Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nature methods*, 9(10):999, 2012.

Laurent Jacob and Jean-Philippe Vert. Efficient peptide–MHC-I binding prediction for alleles with few known binders. *Bioinformatics*, 24(3):358–366, 2008. doi: 10.1093/bioinformatics/btm611. URL http://bioinformatics.oxfordjournals.org/content/24/3/358.abstract.

Roland Jäger, Gabriele Migliorini, Marc Henrion, Radhika Kandaswamy, Helen E Speedy, Andreas Heindl, Nicola Whiffin, Maria J Carnicer, Laura Broome, Nicola Dryden, et al. Capture Hi-C identifies the chromatin interactome of colorectal cancer risk loci. *Nature communications*, 6:6178, 2015.

Tamar Juven-Gershon, Jer-Yuan Hsu, Joshua WM Theisen, and James T Kadonaga. The RNA polymerase II core promoter—the gateway to transcription. *Current opinion in cell biology*, 20(3):253–259, 2008.

Peter Kerpedjiev, Nezar Abdennur, Fritz Lekschas, Chuck McCallum, Kasper Dinkla, Hendrik Strobelt, Jacob M. Luber, Scott B. Ouellette, Alaleh Azhir, Nikhil Kumar, Jeewon Hwang, Soohyun Lee, Burak H. Alver, Hanspeter Pfister, Leonid A. Mirny, Peter J. Park, and Nils Gehlenborg. HiGlass: web-based visual exploration and analysis of genome interaction maps. *Genome Biology*, 19(1):125, Aug 2018. doi: 10.1186/s13059-018-1486-1. URL https://doi.org/10.1186/s13059-018-1486-1.

Philip A. Knight and Daniel Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 2012. doi: 10.1093/imanum/drs019. URL http://imajna.oxfordjournals.org/content/early/2012/10/26/imanum.drs019.abstract.

Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–922, 2001.

David S. Latchman. Transcription factors: An overview. *The International Journal of Biochemistry & Cell Biology*, 29(12):1305 – 1312, 1997. ISSN 1357-2725. doi: https://doi.org/10.1016/S1357-2725(97)00085-X. URL http://www.sciencedirect.com/science/article/pii/S135727259700085X.

Michael S Lawrence, Petar Stojanov, Paz Polak, Gregory V Kryukov, Kristian Cibulskis, Andrey Sivachenko, Scott L Carter, Chip Stewart, Craig H Mermel, Steven A Roberts, et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, 499(7457):214, 2013.

C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 7, pages 566–575, 2002.

Christina Leslie and Rui Kuang. Fast kernels for inexact string matching. In *Learning Theory and Kernel Machines*, pages 114–128. Springer, 2003.

Christina S. Leslie, Eleazar Eskin, Adiel Cohen, Jason Weston, and William Stafford Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004. doi: 10.1093/bioinformatics/btg431. URL http://bioinformatics.oxfordjournals.org/content/20/4/467.abstract.

Guoliang Li, Melissa J. Fullwood, Han Xu, Fabianus Hendriyan Mulawadi, Stoyan Velkov, Vinsensius Vega, Pramila Nuwantha Ariyaratne, Yusoff Bin Mohamed, Hong-Sain Ooi, Chandana Tennakoon, Chia-Lin Wei, Yijun Ruan, and Wing-Kin Sung. Chia-pet tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. *Genome Biology*, 11(2):R22, Feb 2010. doi: 10.1186/gb-2010-11-2-r22. URL https://doi.org/10.1186/gb-2010-11-2-r22.

Erez Lieberman-Aiden, Nynke L. van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, Bradley Bernstein, M. A. Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A. Mirny, Eric S. Lander, and Job Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, 2009. doi: 10.1126/science.1181369. URL http://www.sciencemag.org/content/326/5950/289.abstract.

Thomas Lingner and Peter Meinicke. Remote homology detection based on oligomer distances. *Bioinformatics (Oxford, England)*, 22(18):2224–2231, September 2006. ISSN 1367-4811. doi: 10.1093/bioinformatics/btl376. URL http://www.ncbi.nlm.nih.gov/pubmed/16837522.

Shai Lubliner, Ifat Regev, Maya Lotan-Pompan, Sarit Edelheit, Adina Weinberger, and Eran Segal. Core promoter sequence in yeast is a major determinant of expression level. *Genome research*, 25(7):1008–1017, 2015.

Aaron TL Lun and Gordon K Smyth. diffhic: a bioconductor package to detect differential genomic interactions in Hi-C data. *BMC bioinformatics*, 16(1):258, 2015.

Darío G Lupiáñez, Katerina Kraft, Verena Heinrich, Peter Krawitz, Francesco Brancati, Eva Klopocki, Denise Horn, Hülya Kayserili, John M Opitz, Renata Laxova, et al. Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell*, 161(5):1012–1025, 2015.

Nicola L. Mahy, Paul E. Perry, and Wendy A. Bickmore. Gene density and transcription influence the localization of chromatin outside of chromosome territories detectable by FISH. *The Journal of Cell Biology*, 159(5):753–763, 2002. doi: 10.1083/jcb.200207115. URL http://jcb.rupress.org/content/159/5/753.abstract.

Andrea Malaspina et al. A survey of trinucleotide/tandem repeat-containing transcripts (TNRTs) isolated from human spinal cord to identify genes containing unstable DNA regions as candidates for disorders of motor function. *Brain Research Bulletin*, 56(3-4):299 – 306, 2001. ISSN 0361-9230. doi: http://dx.doi.org/10.

1016/S0361-9230(01)00597-4. URL http://www.sciencedirect.com/science/article/pii/S0361923001005974. Triplet Repeat Diseases.

Sören Sonnenburg Marius Kloft, Ulf Brefeld and Alexander Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.

Marc A Marti-Renom, Genevieve Almouzni, Wendy A Bickmore, Kerstin Bystricky, Giacomo Cavalli, Peter Fraser, Susan M Gasser, Luca Giorgetti, Edith Heard, Mario Nicodemi, et al. Challenges and guidelines toward 4d nucleome data and model standards. *Nature genetics*, page 1, 2018.

Davis J McCarthy, Yunshun Chen, and Gordon K Smyth. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic acids research*, 40(10):4288–4297, 2012.

Peter Meinicke, Maike Tech, Burkhard Morgenstern, and Rainer Merkl. Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. *BMC Bioinformatics*, 5(1):169, 2004. ISSN 1471-2105. doi: 10.1186/1471-2105-5-169. URL http://www.biomedcentral.com/1471-2105/5/169.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a. URL http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

Raphaël Mourad and Olivier Cuvier. Computational identification of genomic features that influence 3D chromatin domain formation. *PLOS Computational Biology*, 12 (5):1–24, 05 2016. doi: 10.1371/journal.pcbi.1004908. URL https://doi.org/10.1371/journal.pcbi.1004908.

Maxwell R Mumbach, Adam J Rubin, Ryan A Flynn, Chao Dai, Paul A Khavari, William J Greenleaf, and Howard Y Chang. HiChIP: efficient and sensitive analysis of protein-directed genome architecture. *Nature methods*, 13(11):919, 2016.

Takashi Nagano, Yaniv Lubling, Tim J Stevens, Stefan Schoenfelder, Eitan Yaffe, Wendy Dean, Ernest D Laue, Amos Tanay, and Peter Fraser. Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. *Nature*, 502(7469):59, 2013.

Andrew Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 78–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015435. URL http://doi.acm.org/10.1145/1015330.1015435.

Sarvesh Nikumbh and Nico Pfeifer. Genetic sequence-based prediction of long-range chromatin interactions suggests a potential role of short tandem repeat sequences in genome organization. *BMC Bioinformatics*, 18(1):218, 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1624-x. URL http://dx.doi.org/10.1186/s12859-017-1624-x.

Sarvesh Nikumbh, Peter Ebert, and Nico Pfeifer. All fingers are not the same: Handling variable-length sequences in a discriminative setting using conformal multi-instance kernels. *bioRxiv*, 2017a. doi: 10.1101/139618. URL https://www.biorxiv.org/content/early/2017/05/18/139618.

Sarvesh Nikumbh, Peter Ebert, and Nico Pfeifer. All Fingers Are Not the Same: Handling Variable-Length Sequences in a Discriminative Setting Using Conformal Multi-Instance Kernels. In Russell Schwartz and Knut Reinert, editors, *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*, volume 88 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:14, Dagstuhl, Germany, 2017b. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/LIPIcs.WABI.2017.16. URL http://drops.dagstuhl.de/opus/volltexte/2017/7645.

Elphège P Nora, Bryan R Lajoie, Edda G Schulz, Luca Giorgetti, Ikuhiro Okamoto, Nicolas Servant, Tristan Piolot, Nynke L van Berkum, Johannes Meisig, John Sedat, et al. Spatial partitioning of the regulatory landscape of the X-inactivation centre. *Nature*, 485(7398):381, 2012.

Sergej Nowoshilow, Siegfried Schloissnig, Ji-Feng Fei, Andreas Dahl, Andy WC Pang, Martin Pippel, Sylke Winkler, Alex R Hastie, George Young, Juliana G Roscito, et al. The axolotl genome and the evolution of key tissue formation regulators. *Nature*, 554(7690):50, 2018.

Nico Pfeifer and Oliver Kohlbacher. Multiple instance learning allows MHC class II epitope predictions across alleles. In Keith A. Crandall and Jens Lagergren, editors, *Algorithms in Bioinformatics*, pages 210–221, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-87361-7.

Jennifer E Phillips-Cremins, Michael EG Sauria, Amartya Sanyal, Tatiana I Gerasimova, Bryan R Lajoie, Joshua SK Bell, Chin-Tong Ong, Tracy A Hookway, Changying Guo, Yuhua Sun, et al. Architectural protein subclasses shape 3d organization of genomes during lineage commitment. *Cell*, 153(6):1281–1295, 2013.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL http://www.R-project.org/.

Senthilkumar Ramamoorthy, Hita Sony Garapati, and Rakesh Kumar Mishra. Length and sequence dependent accumulation of simple sequence repeats in vertebrates: Potential role in genome organization and regulation. *Gene*, 551(2):167 – 175, 2014. ISSN 0378-1119. doi: https://doi.org/10.1016/j.gene.2014.08.052. URL http://www.sciencedirect.com/science/article/pii/S0378111914009913.

Fidel Ramírez, Vivek Bhardwaj, Laura Arrigoni, Kin Chung Lam, Björn A Grüning, José Villaveces, Bianca Habermann, Asifa Akhtar, and Thomas Manke. High-resolution TADs reveal DNA sequences underlying genome organization in flies. *Nature communications*, 9(1):189, 2018.

Suhas S P. Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer,

Eric S Lander, and Erez Lieberman Aiden. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014. doi: 10.1016/j.cell.2014.11.021. URL http://www.cell.com/cell/abstract/S0092-8674(14)01497-4.

G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: recognition of alternatively spliced exons in C.elegans. *Bioinformatics*, 21(suppl 1):i369–i377, 2005. doi: 10.1093/bioinformatics/bti1053. URL http://bioinformatics.oxfordjournals.org/content/21/suppl_1/i369.abstract.

Gunnar Rätsch and Sören Sonnenburg. Accurate splice site prediction for caenorhabditis elegans. In *Kernel Methods in Computational Biology*, MIT Press series on Computational Molecular Biology, pages 277–298. MIT Press, Cambridge, MA., 2004.

Nicolò Riggi, Birgit Knoechel, Shawn M Gillespie, Esther Rheinbay, Gaylor Boulay, Mario L Suvà, Nikki E Rossetti, Wannaporn E Boonseng, Ozgur Oksuz, Edward B Cook, et al. EWS-FLI1 utilizes divergent chromatin remodeling mechanisms to directly activate or repress enhancer elements in Ewing sarcoma. *Cancer cell*, 26 (5):668–681, 2014.

James T. Robinson, Douglass Turner, Neva C. Durand, Helga Thorvaldsdóttir, Jill P. Mesirov, and Erez Lieberman Aiden. Juicebox.js provides a cloud-based visualization system for Hi-C data. *Cell Systems*, 6(2):256 – 258.e1, 2018. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2018.01.001. URL http://www.sciencedirect.com/science/article/pii/S2405471218300012.

Volker Roth, Julian Laub, Klaus-Robert Müller, and Joachim M Buhmann. Going metric: Denoising pairwise data. In *Advances in Neural Information Processing Systems*, pages 841–848, 2003.

Sushmita Roy, Alireza Fotuhi Siahpirani, Deborah Chasman, Sara Knaack, Ferhat Ay, Ron Stewart, Michael Wilson, and Rupa Sridharan. A predictive modeling approach for cell line-specific long-range regulatory interactions. *Nucleic Acids Research*, 2015. doi: 10.1093/nar/gkv865. URL http://nar.oxfordjournals.org/content/early/2015/09/03/nar.gkv865.abstract.

Pelin Sahlén, Ilgar Abdullayev, Daniel Ramsköld, Liudmila Matskova, Nemanja Rilakovic, Britta Lötstedt, Thomas J. Albert, Joakim Lundeberg, and Rickard Sandberg. Genome-wide mapping of promoter-anchored interactions with close to single-enhancer resolution. *Genome Biology*, 16(1):156, Aug 2015. doi: 10.1186/s13059-015-0727-9. URL https://doi.org/10.1186/s13059-015-0727-9.

Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda, and Tatsuya Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, July 2004. ISSN 1367-4803. doi: 10.1093/bioinformatics/bth141. URL http://dx.doi.org/10.1093/bioinformatics/bth141.

Steven L Salzberg. Open questions: How many genes do we have? *BMC biology*, 16 (1):94, 2018.

Albin Sandelin, Wynand Alkema, Pär Engström, Wyeth W Wasserman, and Boris Lenhard. Jaspar: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic acids research*, 32(suppl_1):D91–D94, 2004.

Amartya Sanyal, Bryan R. Lajoie, Gaurav Jain, and Job Dekker. The long-range interaction landscape of gene promoters. *Nature*, 489(7414):109–113, Sep 2012. ISSN 0028-0836. doi: 10.1038/nature11279. URL http://dx.doi.org/10.1038/nature11279.

T. D. Schneider and R. M. Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*, 18:6097–6100, 1990.

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Sebastian J. Schultheiss, Wolfgang Busch, Jan U. Lohmann, Oliver Kohlbacher, and Gunnar Rätsch. KIRMES: kernel-based identification of regulatory modules in euchromatic sequences. *Bioinformatics*, 25(16):2126–2133, 2009. doi: 10.1093/bioinformatics/btp278. URL http://bioinformatics.oxfordjournals.org/content/25/16/2126.abstract.

Edgar Serfling, Maria Jasin, and Walter Schaffner. Enhancers and eukaryotic gene transcription. *Trends in Genetics*, 1:224 – 230, 1985. ISSN 0168-9525. doi: https://doi.org/10.1016/0168-9525(85)90088-5. URL http://www.sciencedirect.com/science/article/pii/0168952585900885.

Nicolas Servant, Nelle Varoquaux, Bryan R Lajoie, Eric Viara, Chong-Jian Chen, Jean-Philippe Vert, Edith Heard, Job Dekker, and Emmanuel Barillot. HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. *Genome biology*, 16 (1):259, 2015.

Tom Sexton and Giacomo Cavalli. The role of chromosome domains in shaping the functional genome. *Cell*, 160(6):1049 – 1059, 2015. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2015.02.040. URL http://www.sciencedirect.com/science/article/pii/S009286741500241X.

Nidhi Shah, Michael G Nute, Tandy Warnow, and Mihai Pop. Misunderstood parameter of ncbi blast impacts the correctness of bioinformatics workflows. *Bioinformatics*, page bty833, 2018. doi: 10.1093/bioinformatics/bty833. URL http://dx.doi.org/10.1093/bioinformatics/bty833.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521813972.

Nathan C Sheffield, Gaelle Pierron, Johanna Klughammer, Paul Datlinger, Andreas Schönegger, Michael Schuster, Johanna Hadler, Didier Surdez, Delphine Guillemot, Eve Lapouble, et al. DNA methylation heterogeneity defines a disease spectrum in Ewing sarcoma. *Nature medicine*, 23(3):386, 2017.

Marieke Simonis, Petra Klous, Erik Splinter, Yuri Moshkin, Rob Willemsen, Elzo De Wit, Bas Van Steensel, and Wouter De Laat. Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture–on-chip (4C). *Nature genetics*, 38(11):1348, 2006.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Stephen T. Smale and James T. Kadonaga. The RNA polymerase II core promoter. *Annual Review of Biochemistry*, 72(1):449–479, 2003. doi: 10.1146/annurev.biochem.72.121801.161520. URL https://doi.org/10.1146/annurev.biochem.72.121801.161520. PMID: 12651739.

Sören Sonnenburg, Alexander Zien, Petra Philips, and Gunnar Rätsch. POIMs: positional oligomer importance matrices — understanding support vector machine based signal detectors. *Bioinformatics*, July 2008. URL http://bioinformatics.oxfordjournals.org/cgi/content/full/24/13/i6.

Sören Sonnenburg, Gunnar Rätsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtěch Franc. The SHOGUN machine learning toolbox. *J. Mach. Learn. Res.*, 11: 1799–1802, August 2010. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1756006.1859911.

John C Stansfield, Kellen G Cresswell, Vladimir I Vladimirov, and Mikhail G Dozmorov. Hi-Ccompare: an R-package for joint normalization and comparison of Hi-C datasets. *BMC bioinformatics*, 19(1):279, 2018.

Amos Tanay. Extensive low-affinity transcriptional interactions in the yeast genome. *Genome Research*, 16(8):962–972, 2006. doi: 10.1101/gr.5113606. URL http://genome.cshlp.org/content/16/8/962.abstract.

Satoshi Tashiro and Christian Lanctôt. The international nucleome consortium. *Nucleus*, 6(2):89–92, 2015. doi: 10.1080/19491034.2015.1022703. URL https://doi.org/10.1080/19491034.2015.1022703. PMID: 25738524.

The ENCODE Project Consortium. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, Sep 2012. ISSN 0028-0836. doi: 10.1038/nature11247. URL http://dx.doi.org/10.1038/nature11247.

Eleni M Tomazou, Nathan C Sheffield, Christian Schmidl, Michael Schuster, Andreas Schönegger, Paul Datlinger, Stefan Kubicek, Christoph Bock, and Heinrich Kovar. Epigenome mapping reveals distinct modes of gene regulation and widespread enhancer reprogramming by the oncogenic fusion protein ews-fli1. *Cell reports*, 10(7): 1082–1095, 2015.

K. Tsuda, S. Uda, T. Kin, and K. Asai. Minimizing the cross validation error to mix kernel matrices of heterogeneous biological data. *Neural Processing Letters*, 19:63–72, 2004.

Koji Tsuda. Support vector classifier with asymmetric kernel functions. In *in European Symposium on Artificial Neural Networks (ESANN*. Citeseer, 1999.

Oana Ursu, Nathan Boley, Maryna Taranova, YX Rachel Wang, Galip Gurkan Yardimci, William Stafford Noble, Anshul Kundaje, and Inanc Birol. Genomedisco: A concordance score for chromosome conformation capture experiments using random walks on contact map graphs. *Bioinformatics*, 1:7, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Nelle Varoquaux, Ferhat Ay, William Stafford Noble, and Jean-Philippe Vert. A statistical approach for inferring the 3D structure of the genome. *Bioinformatics (Oxford, England)*, 30(12):i26–33, June 2014. ISSN 1367-4811. doi: 10.1093/bioinformatics/btu268. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4229903&tool=pmcentrez&rendertype=abstract.

J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

P Vogt. Potential genetic functions of tandem repeated dna sequence blocks in the human genome are based on a highly conserved "chromatin folding code". *Human genetics*, 84(4):301—336, March 1990. ISSN 0340-6717. doi: 10.1007/bf00196228. URL http://dx.doi.org/10.1007/BF00196228.

Emanuela V Volpi and Joanna M Bridger. FISH glossary: an overview of the fluorescence in situ hybridization technique. *Biotechniques*, 45(4):385–409, 2008.

Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016. doi: 10.23915/distill.00002. URL http://distill.pub/2016/misread-tsne.

Sean Whalen, Rebecca M Truty, and Katherine S Pollard. Enhancer-promoter interactions are encoded by complex genomic signatures on looping chromatin. *Nature Genetics*, 48(5):488–496, 2016.

Christian Widmer and Gunnar Rätsch. Multitask learning in computational biology. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 207–216, 2012.

Iain Williamson, Ragnhild Eskeland, Laura A Lettice, Alison E Hill, Shelagh Boyle, Graeme R Grimes, Robert E Hill, and Wendy A Bickmore. Anterior-posterior differences in HoxD chromatin topology in limb development. *Development*, 139 (17):3157–3167, 2012.

Joachim Wolff, Vivek Bhardwaj, Stephan Nothjunge, Gautier Richard, Gina Renschler, Ralf Gilsbach, Thomas Manke, Rolf Backofen, Fidel Ramírez, and Björn A Grüning. Galaxy hicexplorer: a web server for reproducible Hi-C data analysis, quality control and visualization. *Nucleic Acids Research*, 46(W1):W11–W16, 2018. doi: 10.1093/nar/gky504. URL http://dx.doi.org/10.1093/nar/gky504.

Eitan Yaffe and Amos Tanay. Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nat Genet*, 43 (11):1059–1065, Nov 2011. ISSN 1061-4036. doi: 10.1038/ng.947. URL http://dx.doi.org/10.1038/ng.947.

J. Omar Yáñez-Cuna et al. Dissection of thousands of cell type-specific enhancers identifies dinucleotide repeat motifs as general enhancer features. *Genome Research*, 24(7):1147–1156, 2014. doi: 10.1101/gr.169243.113. URL http://genome.cshlp.org/content/24/7/1147.abstract.

Tao Yang, Feipeng Zhang, Galip Gurkan Yardimci, Fan Song, Ross C Hardison, William Stafford Noble, Feng Yue, and Qunhua Li. HiCRep: assessing the reproducibility of Hi-C data using a stratum-adjusted correlation coefficient. *Genome research*, pages gr–220640, 2017a.

Yang Yang, Ruochi Zhang, Shashank Singh, and Jian Ma. Exploiting sequence-based features for predicting enhancer–promoter interactions. *Bioinformatics*, 33(14): i252–i260, 2017b. doi: 10.1093/bioinformatics/btx257. URL http://dx.doi.org/10.1093/bioinformatics/btx257.

Shi Yu, Tillmann Falck, Anneleen Daemen, Leon-Charles Tranchevent, Johan Suykens, Bart De Moor, and Yves Moreau. L2-norm multiple kernel learning and its application to biomedical data fusion. *BMC Bioinformatics*, 11(1):309, 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-309. URL http://www.biomedcentral.com/1471-2105/11/309.

Michael J. Zeitz, Ferhat Ay, Julia D. Heidmann, Paula L. Lerner, William S. Noble, Brandon N. Steelman, and Andrew R. Hoffman. Genomic interaction profiles in breast cancer reveal altered chromatin architecture. *PLoS ONE*, 8(9):e73974, 09 2013. doi: 10.1371/journal.pone.0073974. URL http://dx.doi.org/10.1371%2Fjournal.pone.0073974.

Zhihu Zhao, Gholamreza Tavoosidana, Mikael Sjölinder, Anita Göndör, Piero Mariano, Sha Wang, Chandrasekhar Kanduri, Magda Lezcano, Kuljeet Singh Sandhu, Umashankar Singh, et al. Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra-and interchromosomal interactions. *Nature genetics*, 38(11):1341, 2006.

Aleksey V Zimin, Daniela Puiu, Richard Hall, Sarah Kingan, Bernardo J Clavijo, and Steven L Salzberg. The first near-complete assembly of the hexaploid bread wheat genome, Triticum aestivum. *GigaScience*, 6(11):1–7, 2017. doi: 10.1093/gigascience/gix097. URL http://dx.doi.org/10.1093/gigascience/gix097.

Marie Zufferey, Daniele Tavernari, Elisa Oricchio, and Giovanni Ciriello. Comparison of computational methods for the identification of topologically associating domains. *Genome Biology*, 19(1):217, Dec 2018. ISSN 1474-760X. doi: 10.1186/s13059-018-1596-9. URL https://doi.org/10.1186/s13059-018-1596-9.