The Partial Order Kernel and its Application to Understanding the Regulatory Grammar of Conserved Non-coding Elements

Maryam Abdollahyan

Submitted in partial fulfilment of the requirements
of the degree of Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary University of London

December 2018

Statement of Originality

I, Maryam Abdollahyan, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature:

Date: December 2018

For details of collaborations and publications, see "Collaboration Details and Publications".

Abstract

Conserved non-coding elements (CNEs) are regions of non-coding DNA which have remained evolutionarily conserved across various species over millions of years and are found to cluster near genes involved in early embryonic development, suggesting that they play an important role as regulatory elements. Indeed, many CNEs have been shown to act as enhancers; however, not all regulatory elements are conserved and in some cases, deletion of CNEs did not result in any notable phenotypes. These opposing findings indicate that the functions of CNEs are still poorly understood and further research on these elements is needed to uncover the reasons for their extreme conservation. The aim of this thesis is to investigate the use and development of algorithms for decoding the regulatory grammar of CNEs. Initially, an assessment of several methods for functional classification of CNEs is provided. The results obtained using these methods are validated by functional assays and their limitations in capturing the grammar of CNEs are discussed. Motivated by these limitations, a partial order graph representation of the sequence of transcription factor binding sites (TFBSs) in a CNE that allows efficient handling of the overlapping sites is introduced. A dynamic programming-based method for aligning two such graphs and identifying regulatory signatures composed of co-occurring TFBSs is proposed and evaluated. The results demonstrate the predictive ability of this method, which can be used to prioritise regions for experimental validation. Building on this method, the partial order kernel (POKer) for comparison of strings containing alternative substrings and represented by partial order graphs is introduced. The POKer is evaluated in different sequence comparison tasks, including visual localisation. An approach using the POKer for functional classification of CNEs is introduced and its effectiveness in capturing the grammar of CNEs is demonstrated. Finally, the implications of the results presented in this work for modelling the evolution of CNEs are discussed.

Acknowledgements

Special thanks to my supervisor Dr Fabrizio Smeraldi for his patient guidance, constant encouragement and invaluable advice throughout this work.

I thank Dr Greg Elgar for giving me the opportunity to work in his laboratory. I also thank the members of the Regulatory Genomics Laboratory at the Francis Crick Institute for their help and friendship.

I am grateful to the members of my progression panel, Dr Thomas Roelleke and Dr William Marsh, for their useful comments and suggestions. I am also grateful to our research student coordinator, Mrs Melissa Yeo, for her ongoing care and assistance.

I dedicate this thesis to my family for their endless love and support.

Contents

List of Figures **8**

List of Tables **10**

Introduction **11**

1 Biological Context and Motivation **16**
1.1 The Building Blocks 16
1.2 From DNA to Protein 17
1.2.1 Transcription 17
1.2.2 Translation 19
1.3 Transcriptional Regulation 20
1.3.1 Cis-regulatory Elements 20
1.3.2 Transcription Factors 21
1.4 Conserved Non-coding Elements (CNEs) 23
1.4.1 Origins 23
1.4.2 General Properties 24
1.4.3 Functions and Unknown Reasons for Conservation 25

2 Computational Background **27**
2.1 Biological Sequence Representation 27
2.2 Sequence Alignment 28
2.2.1 Pairwise Alignment via Dynamic Programming 29
2.3 Identifying Cis-Regulatory Elements and Modules 30
2.4 Kernel Methods 32
2.4.1 Kernels for Biological Sequences 33

3 Comparative Evaluation of Methods for Grouping Functionally Related CNEs **34**
3.1 Related Work 35
3.2 Sequence Similarity Measures 35
3.3 Methods 36

3.3.1 Clustering . . . 36
3.3.2 Dimensionality Reduction . . . 39
3.3.3 Network Topology Visualisation . . . 41
3.4 Experiments . . . 41
3.4.1 Results . . . 42
3.5 Availability . . . 47

4 Identifying Regulatory Signatures in CNEs Using Transcription Factor Binding Site (TFBS) Alignment 48
4.1 Related Work . . . 49
4.2 Graph Representation of CNEs . . . 49
4.3 Partial Order Alignment of CNE Graphs . . . 50
4.4 Measuring the Frequency of Aligned TFBSs . . . 52
4.5 Evaluation . . . 54
4.5.1 Results . . . 55
4.6 Availability . . . 56

5 The Partial Order Kernel (POKer) for Comparing Strings with Alternative Substrings 57
5.1 Related Work . . . 58
5.2 Representing Strings with Alternative Substrings . . . 59
5.3 Partial Order Kernel . . . 60
5.3.1 Computation . . . 61
5.4 Experiments . . . 62
5.4.1 Generalised Spectrum Kernel . . . 63
5.4.2 Data Simulation and Parameters . . . 63
5.4.3 Results . . . 65
5.5 Availability . . . 67

6 Evaluating the POKer: a Computer Vision Application 68
6.1 Related Work . . . 69
6.2 Visual Localisation Using the POKer . . . 70
6.3 Experiments . . . 71
6.3.1 Dataset and Parameters . . . 71
6.3.2 Baseline Methods . . . 73
6.3.3 Results . . . 74
6.4 Availability . . . 77

7 Functional Classification of CNEs Based on Their TFBSs 78
7.1 Related Work . . . 79
7.2 Detecting Regulatory Signatures in CNEs . . . 79

7.3 Experimental Setup 81
7.3.1 Results 82
7.4 Availability 84

Concluding Remarks **85**

References **87**

Appendices **107**

List of Figures

1.1 Central dogma of molecular biology 17
1.2 Transcription 18
1.3 Translation 19
1.4 Models of enhancer activity 22

2.1 Example of an alignment 28
2.2 Example of a score matrix 30

3.1 Hierarchical clustering of CNEs 43
3.2 Spectral clustering of CNEs 44
3.3 t-SNE map of CNEs 45
3.4 BOSAM of the CNE network 45
3.5 Expression driven by the assayed CNEs 46

4.1 Example of a CNE graph 50
4.2 Example of the partial order alignment of two CNE graphs . . . 53

5.1 Relation between alignment-based methods for comparing strings with alternative substrings 58
5.2 Example of the graph representation of a string with alternative substrings 59
5.3 Sample strings from the experiment dataset 64
5.4 ROC curves for the POKer and the generalised spectrum kernel . 66
5.5 Confusion matrices for the POKer and the generalised spectrum kernel 67

6.1 Example of the graph representation of a set of alternative image sequences 71
6.2 Overview of visual localisation using the POKer 72
6.3 Precision-recall curves for the POKer and the baseline methods . 76

7.1 Projection of CNEs using kernel PCA 83

A.1 t-SNE maps of the best performing features 110

B.1 BOSAMs of the user interaction networks 113

List of Tables

3.1 Lance-Williams update formula 37

4.1 List of TF families 54
4.2 Top over-represented co-occurring TFBSs 56

5.1 Mean AUROC values for the POKer and the generalised spectrum kernel 65

6.1 Average recall values for the POKer and the baseline methods . 74

A.1 Accuracy values achieved in the gender recognition task 109

B.1 Details of the message boards 112

Introduction

Gene expression, the process by which the information encoded in genes is used to synthesise functional products such as proteins, is a key process in the development of all organisms. Gene expression consists of several steps, many of which are orchestrated by elements known as regulatory elements. These elements control the timing, location and amount of gene expression, allowing the cell to adapt to the environment and respond to external signals. Understanding the mechanisms of action of regulatory elements is thus crucial to our understanding of evolution and diseases.

An approach to identifying and characterising regulatory elements is comparative sequence analysis, in which genomic sequences from different organisms are compared in order to identify genomic regions that are evolutionarily conserved and analyse them for potential regulatory function. The rationale behind this approach is that functional regions of the genome are generally under selective pressure, and therefore, are more likely to be similar across organisms than other regions [1]. Comparative sequence analysis of vertebrate genomes has led to the discovery of a set of non-coding DNA sequences known as conserved non-coding elements (CNEs). Non-coding DNA – regions of DNA that do not encode proteins – make up over 98% of the human genome [2]. Initially called junk DNA, some non-coding DNA is now known to have important biological functions [3]. What sets CNEs apart from other non-coding DNA is their level of conservation, their distribution throughout the vertebrate lineage and their location within vertebrate genomes [4]: CNEs show exceptionally high levels of conservation [5]; they have remained conserved across evolutionarily distant species for millions of years [6]; and they appear in clusters near key developmental genes [7]. These features make CNEs candidates for regulatory elements. Although the regulatory functions of a number of CNEs have been confirmed [8, 9], the reasons for their extreme conservation are still unknown.

During gene expression, the genetic information stored in DNA is transcribed into RNA molecules. Transcription is partly controlled by proteins known as

transcription factors (TFs). TFs bind, in a cooperative manner, to short DNA motifs called transcription factor binding sites (TFBSs). Thus, the presence of clusters of TFBSs is assumed to be a good indicator of regulatory activity [10]. CNEs are enriched in overlapping TFBSs [11]. Hence, one explanation proposed for the high levels of conservation seen in CNEs is that since mutations within these elements can affect multiple TFBSs, they have come under selective pressure, as mutations in TFBSs may have unexpected effects on gene expression which can be selected against. This, however, does not fully explain the extent of CNEs conservation. According to the 'billboard' model of TF interactions, the exact arrangement of TFBSs is not necessary for regulatory activity [12].

The majority of functionally characterised CNEs act as enhancers, increasing the probability of transcription [13]. This offers another potential explanation for the conservation of CNEs since, as mentioned above, functional sequences tend to be more conserved than those which have been predicted to be non-functional. However, this explanation suffers from the observation that regulatory elements are not necessarily conserved and can diverge [14].

Finally, in some cases, deletion of CNEs resulted in no notable phenotypes, casting doubt on the functional importance of these elements [15]. These opposing findings indicate that the nature of conservation of CNEs is more complex than thought before, and further work is needed in order to understand the roles of CNEs and their mechanisms of action.

Testing whether a CNE drives the expression of a gene is commonly done using *in-vivo* functional assays; however, these methods have a few shortcomings [16], which highlight the need for new methods for assessing the regulatory activity of CNEs. In this work, we aim to

- model CNEs based on their TFBSs,
- develop machine learning algorithms to compare these models, and
- predict novel regulatory elements.

In doing so, we test the following hypotheses:

- regulatory activity of CNEs can be predicted using TFBS-based features (as opposed to their primary sequence),
- CNEs share regulatory sequence signatures that can be detected using machine learning algorithms, and
- these signatures can be used to identify additional regulatory elements.

Structure of the Thesis

Chapter 1 provides an overview of the biological entities and processes that are the subject of our work. We begin with the definitions of information-carrying biopolymers, namely DNA, RNA and proteins, and the central dogma of molecular biology which outlines the flow of genetic information between these biopolymers. Next, we focus on the process of transcription, the first stage in the transfer of information from DNA to protein. We look at how it is regulated and what entities are involved in it. Finally, we review the current research on CNEs, including their origin, their sequence properties and the hypotheses about their roles in transcriptional regulation that have provided the motivation for our work.

Chapter 2 contains the definitions, notations and descriptions for the computational methods that form the foundations of the works presented in subsequent chapters. These include representation of biological sequences, sequence alignment and kernel methods.

In Chapter 3, we consider the applications of a number of existing clustering algorithms and network analysis and dimensionality reduction techniques to group CNEs into clusters of functionally related elements. To measure the similarities between CNEs, we employ different metrics based on the occurrences of TFBSs within the elements. We apply the considered methods in conjunction with these metrics to a set of CNEs and validate the results by performing functional assays. We discuss the limitations of these methods in capturing the regulatory sequence signatures in CNEs and highlight the need for a new method for comparing these elements.

In Chapter 4, we introduce a graph representation of the sequence of TFBSs identified in a CNE that efficiently handles overlapping binding sites. Moreover, we present a dynamic programming algorithm for aligning two such graphs, and show how the frequency of aligned TFBSs is measured in order to detect regulatory sequence signatures composed of co-occurring TFBSs. We use this method to identify the regulatory signatures in elements from a set of functionally validated CNEs. We compare the results with those obtained using functional assays to demonstrate the predictive power of our approach.

In Chapter 5, we introduce a new kernel based on sequence alignment, called the partial order kernel (POKer). The POKer produces a measure of similarity between strings that contain alternative substrings (e.g., the sequence of overlapping TFBSs identified in a CNE) which we represent as partial order graphs. To benchmark the POKer's performance, we extend a state-of-the-art string ker-

nel to handle strings with alternative substrings. We show the effectiveness of the POKer by comparing its performance to that of this kernel in two sets of experiments on different simulated datasets.

In Chapter 6, we further evaluate the POKer in a real-world setting by considering its application to a computer vision problem, namely visual localisation. We introduce a novel sequence-based approach that is robust to changes in the appearance of the environment. That is, we convert the sequences of images of a place taken at different times to strings with alternative substrings represented as graphs, and use the POKer to obtain the similarities between these graphs and match the corresponding locations. We demonstrate the robustness of our approach on a standard dataset and in comparison to the state-of-the-art methods.

In Chapter 7, we propose an approach to classifying CNEs into groups of functionally related elements based on the TFBSs they contain. We use the graph representation introduced in Chapter 4 to model CNEs, and employ the POKer to compare the graphs. To evaluate our approach, we train a classifier on a set of functionally validated CNEs. We then test this classifier on a different set of CNEs whose regulatory activity (or lack of it) has been confirmed, and discuss the biological relevance of the results. Moreover, we show how this approach is used to define a tissue-specific regulatory grammar for CNEs.

The thesis concludes with a discussion of our findings. Appendix A summarises a related work in the field of image processing where we used the dimensionality reduction technique considered in Chapter 3. Appendix B reports a case study on social networks that we carried out to evaluate the network analysis technique used in Chapter 3. Appendix C outlines the proof of the theorem in Chapter 5.

Collaboration Details and Publications

The works presented in Chapters 3 and 4 were done in collaboration with the Regulatory Genomics Laboratory (formerly part of the National Institute Of Medical Research (NIMR)) while visiting the Francis Crick Institute.

The work presented in Chapter 6 was done in collaboration with the Intelligent Systems, Automation and Robotics Laboratory (ISARLab) while visiting the Università degli Studi di Perugia.

Publications:

- Abdollahyan M, Smeraldi F, Noyvert B and Elgar G. Transcription factor binding site-based alignment of conserved non-coding elements. Poster presented at the 15th European Conference on Computational Biology (ECCB); 3–7 September 2016. DOI: 10.7490/f1000research.1113029.1.

- Abdollahyan M, Smeraldi F and Elgar G. Identifying Potential Regulatory Elements by Transcription Factor Binding Site Alignment using Partial Order Graphs. 1st Conference on Mathematical Foundations in Bioinformatics (MatBio) 2016 [Special Issue]. International Journal of Foundations of Computer Science. 2018;29(8):1345–1354.

- Bianconi F, Smeraldi F, Abdollahyan M, Xiao P. On the use of skin texture features for gender recognition: an experimental evaluation. In: Proceedings of the 6th International Conference on Image Processing Theory, Tools and Applications (IPTA); 12–15 December 2016. pp. 1–6.

- Abdollahyan M, Smeraldi F. POKer: a Partial Order Kernel for Comparing Strings with Alternative Substrings. In: Proceedings of the 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN); 26–28 April 2017. pp. 263–268.

- Abdollahyan M, Mondragón R, Bessant C and Smeraldi F. Visualising the Topological Structure of Health-related Message Board User Networks. Proceedings of the 1st International Conference on Applications of Intelligent Systems (APPIS) 2018. Frontiers in Artificial Intelligence and Applications. 2018;310:274–279.

- Abdollahyan M, Cascianelli S, Bellocchio E, Costante G, Ciarfuglia T A, Bianconi F, Smeraldi F and Fravolini M. Visual Localization in the Presence of Appearance Changes Using the Partial Order Kernel. In: Proceedings of the 26th European Signal Processing Conference (EUSIPCO); 3–7 September 2018. pp. 702–706.

Chapter 1

Biological Context and Motivation

1.1 The Building Blocks

The genome is made up of deoxyribonucleic acid (DNA) molecules. DNA consists of two nucleotide strands twisted into a double helix. Each nucleotide is composed of a sugar called deoxyribose, a phosphate group and a nitrogen-containing nucleobase: adenine (A), cytosine (C), guanine (G) or thymine (T). For ease of reference, the five carbon atoms in deoxyribose are assigned a number followed by a prime (1', 2' and so forth). The strands of DNA are joined together by hydrogen bonds between pairs of nucleobases: C pairs with G, and A pairs with T. These strands store the genetic information.

In certain organisms (e.g., some viruses), genetic information is stored in ribonucleic acid (RNA). RNA consists of a single nucleotide strand. Nucleotides in RNA differ from those in DNA in two aspects: the constituent sugar in RNA is ribose and the complementary base to adenine (A) is uracil (U). There exist several types of RNA with different functions. We mention some of them in the following sections.

Another class of biopolymers that carry the genetic information are proteins. Proteins are composed of polypeptides, folded into a 3-dimensional structure. Each polypeptide is a sequence of amino acids linked together by covalent bonds called peptide bonds. Amino acids are organic compounds containing an amino (NH_2) group and a carboxyl (COOH) group. There are 20 amino acids found in proteins.

A family of proteins called histones package DNA into the eukaryotic nucleus,

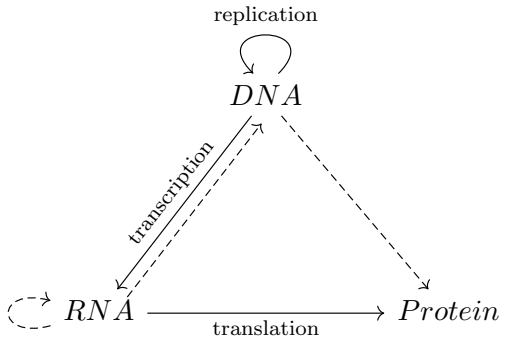


Figure 1.1: Transfer of genetic information between DNA, RNA and protein. Solid arrows are probable general transfers, dotted arrows are possible specific transfers, and absent arrows are unknown transfers which the dogma postulates never occur.

forming units known as nucleosomes. Each nucleosome is composed of approximately 146 base pairs (bp) of DNA, wound 1.65 times around eight histones (two of each of the histones H2A, H2B, H3 and H4). One histone H1 wraps an additional 20 base pairs around this histone core to form a chromatosome. The chain of nucleosomes folds and forms a chromatin fiber, which is further compressed and folded to form one of the two chromatids of a chromosome [17].

1.2 From DNA to Protein

The central dogma of molecular biology [18] outlines the flow of genetic information between DNA, RNA and protein. Figure 1.1 shows the classification of these transfers, as proposed in 1970.

According to the dogma, during a process known as gene expression, genetic information flows from DNA to RNA to protein. Gene Expression occurs in two main steps: transcription and translation. The following two sections describe the processes of transcription and translation in eukaryotes, respectively.

1.2.1 Transcription

During transcription, one strand of DNA, referred to as the template strand, is used for the synthesis of a complementary RNA chain called the primary transcript. The primary transcript is identical to DNA's non-template strand, referred to as the coding strand, with T bases replaced by U bases. Primary transcripts are further modified to yield mature RNAs. DNA is transcribed into different types of RNA. Those that convey the genetic information from DNA to protein are called messenger RNAs (mRNAs), and a primary transcript that

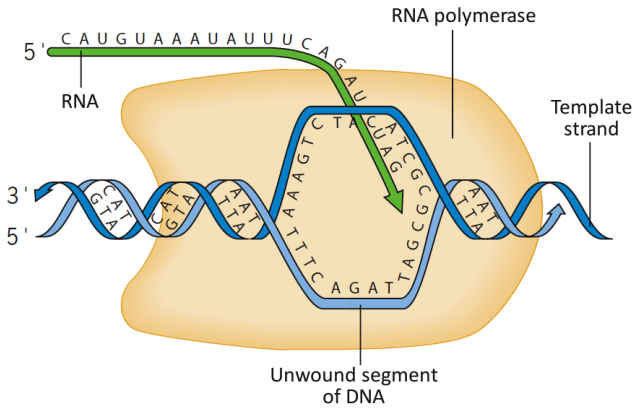


Figure 1.2: RNA synthesis within the transcription bubble. Image source: [20]

becomes an mRNA is called a precursor mRNA (pre-mRNA). RNA synthesis occurs in three stages: initiation, elongation and termination.

In eukaryotes, RNA is synthesised in the nucleus. Synthesis of an RNA chain is initiated by the enzyme RNA polymerase. RNA polymerase, with the assistance of proteins called transcription factors (TFs), binds to a specific region of DNA called the promoter, located near the transcription start site at the 5' end. TFs and promoters are described in Section 1.3. Recall that DNA is tightly packed into the chromatin; therefore, to allow TFs to access it, the chromatin structure must be modified. This process is known as chromatin remodelling. For details of this process, see [19].

During elongation, RNA polymerase moves along the DNA, unwinds it and forms a region called the transcription bubble. In this bubble, RNA polymerase traverses the template strand in the 3' to 5' direction and adds complementary RNA nucleotides to the new chain. DNA that was unwound is rewound after it has been transcribed (Figure 1.2).

Once RNA polymerase passes the end of the gene, a protein complex binds to two locations (a polyadenylation signal sequence and a GU-rich sequence) on the growing transcript and cleaves the primary transcript. The newly synthesised pre-mRNA is released, RNA polymerase dissociates from the template strand and transcription terminates. Pre-mRNAs are further processed before they become mRNAs. This includes the addition of a 5' cap, the addition of a poly(A) tail (polyadenylation) and splicing to remove introns [21]. The resulting mRNA is transported to the cytoplasm where it is translated.

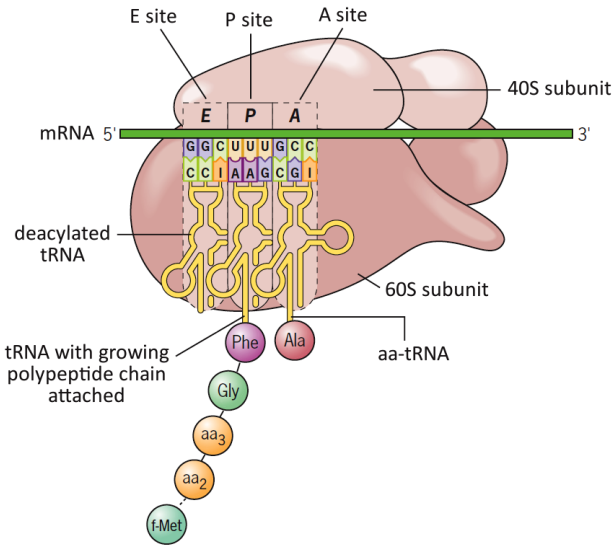


Figure 1.3: Polypeptide chain synthesis in the ribosome. Image source: [20]

1.2.2 Translation

In translation, the genetic information encoded in an mRNA is translated by the ribosome into the chain of amino acids in a polypeptide. The mRNA is read in blocks of nucleotide triplets called codons, each of which specifies an amino acid according to the genetic code. The genetic code is degenerate, meaning that some amino acids are specified by more than one codon. Translation of codons into amino acids requires another class of RNAs called transfer RNAs (tRNAs). Each tRNA carries a nucleotide triplet called an anticodon, which is complementary to a codon in mRNA. tRNAs that are attached to the amino acid which corresponds to their anticodon are called aminoacyl-tRNAs (aa-tRNAs). Similar to transcription, translation occurs in three stages: initiation, elongation and termination.

Ribosomes are made up of ribosomal RNAs (rRNAs) and proteins. Eukaryotic ribosome is composed of two subunits, a small subunit (40S) and a large subunit (60S) [22]. The 40S subunit binds to the mRNA and scans it in the 5' to 3' direction. Protein synthesis is initiated once a start codon (commonly AUG) is encountered and the 40S subunit is joined by the 60S subunit to form the complete ribosome (80S).

There are three binding sites for tRNAs in the ribosome: the aminoacyl (A) site, the peptidyl (P) site and the exit (E) site. During elongation, an aa-tRNA carrying the anticodon that matches the codon in the A site binds to this site. Next, a peptide bond forms between the amino acid of aa-tRNA and the amino acid of the tRNA bound to the P site, which holds the growing polypeptide

chain, and the chain is transferred to aa-tRNA. The tRNAs in the A and P sites are then moved to the P and E sites, respectively. The deacylated tRNA leaves the E site and another aa-tRNA enters the A site (Figure 1.3).

Elongation continues until a stop codon (e.g., UAG) enters the A site. Stop codons are not recognised by any tRNAs. Instead, they are recognised by proteins known as release factors, which trigger the hydrolysis of the bond between the polypeptide chain and the tRNA in the P site. The completed polypeptide is released and the ribosomal subunits dissociate. The resulting polypeptide then folds into a protein and carries out its role in the cell.

1.3 Transcriptional Regulation

Regulation of gene expression is termed gene regulation. Gene regulation occurs at various levels, but the majority of it takes place at the level of transcription. Transcriptional regulation is orchestrated by many entities, notably *cis*-regulatory elements (which regulate the transcription of nearby genes, in contrast to *trans*-regulatory elements) and transcription factors (TFs). The following two sections describe the structures and functions of these regulatory elements.

1.3.1 Cis-regulatory Elements

Promoters and enhancers are the two best-studied *cis*-regulatory elements. Promoters are regions of DNA located upstream of a gene (towards the 5' end of the coding strand) that consist of several elements which can be bound by TFs. The element closest to the transcription start site is the TATA box (named after its sequence which most commonly is TATAAA). Transcription factor IID (TFIID), which contains a TATA box binding protein (TBP) and several TBP-associated factors, is the first TF that binds to the TATA box and forms the transcription preinitiation complex (PIC). It is followed by TFIIA and TFIIB. Next, RNA polymerase and TFIIF join the PIC. Finally, TFIIE and TFIIH bind to the PIC [23]. TFIIH has helicase activity that unwinds DNA and helps create the transcription bubble. Active promoters are marked by trimethylation (methylation is the process by which methyl (CH_3) groups are transferred to lysine or arginine residues in histones; trimethylation refers to the transfer of three methyl groups) of lysine 4 in histone H3 (H3K4) [24]. Promoters vary in length and can be hundreds of base pairs long.

Similar to promoters, enhancers are regions of DNA that contain several TF

binding sites (TFBSs). TFs known as activators bind to enhancers, help recruit the RNA polymerase and increase the transcription rate (a class of regulatory elements called silencers play the opposite role of enhancers; they are bound by TFs known as repressors which prevent transcription). In addition to serving as centres for the assembly of the PIC, enhancers have also been reported to play an important role in transcription elongation [25, 26]. Another way in which enhancers are involved in gene regulation is through a class of non-coding RNAs known as enhancer RNAs (eRNAs), which are transcribed from enhancer sequences. The exact mechanisms by which eRNAs influence gene expression are not known yet; however, a number of them have been shown to have enhancer-like function [27]. Enhancers, similar to promoters, have been characterised by epigenetic features. Active enhancers are marked by monomethylation of H3K4 [24]. Enhancers are typically a few hundred base pairs long.

A hallmark property of enhancers is that they act independently of their location, distance and orientation with respect to their target genes, that is, they may be several hundred kilobases away from their target genes, upstream or downstream of the transcription start site, in forward or reverse direction [28]. Several models of how distal enhancers communicate with their target genes have been proposed [29, 30, 31]. Currently, the favoured model is DNA looping. This model is supported by evidence from chromosome conformation capture (3C) and fluorescence in situ hybridisation (FISH) methods. DNA looping occurs when a protein or a protein complex simultaneously binds to two DNA sites, thereby looping out the intervening DNA [32]. Hence, although an enhancer may be megabases away from its target gene, it is spatially close to the gene and its promoter.

Considering the above, it is not surprising that mutations in *cis*-regulatory elements can result in phenotypic changes and diseases [33].

1.3.2 Transcription Factors

Transcription factors (TFs) are proteins that bind to the regulatory regions of DNA and control the gene expression. The two main types of TF are general (or basal) TFs and specific TFs. General TFs bind to promoters and are involved in the recruitment of RNA polymerase [34, 35]. TFIIH, a general TF, also participates in DNA repair [36]. Specific TFs, namely activators and repressors, bind to enhancers and silencers, respectively. Activators facilitate the binding of general TFs, while repressors inhibit the transcription process using various mechanisms. These include competing with an activator for a common binding site, interfering with an activator bound nearby (quenching) and interfering with

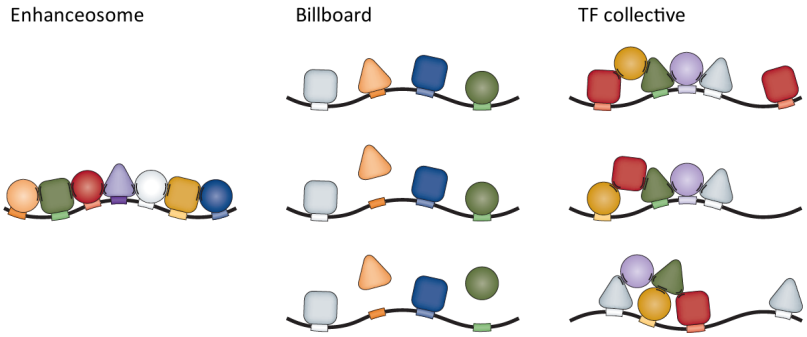


Figure 1.4: Models of TF interactions. The enhanceosome model represents fixed TFBSs composition and positioning, where presence of all TFs is required for enhancer activation. In the billboard model, TFBSs composition is fixed while their positioning is flexible; only a subset of TFs must be bound for enhancer activation. The TF collective model features diverse TFBSs composition and positioning. Image source: [43]

the function of general TFs [37].

TFs have a modular structure where distinct regions are responsible for different functions such as DNA binding, activation and repression of transcription [38]. TFs are categorised into four families, based on the motifs that constitute their binding domains: helix-turn-helix (HTH) (including homeodomain proteins), zinc finger, leucine zipper and helix-loop-helix (HLH). For details of each family, see [39]. TFs are typically 10 base pairs long [40].

TFs do not act alone; they interact with components of the transcription machinery including other TFs [41]. A DNA sequence containing a cluster of TFBSs for multiple interacting TFs is called a *cis*-regulatory module (CRM). TF interactions may impose constraints on the composition of regulatory sequences (e.g., the number, location, orientation and order of TFBSs within the sequence), which define the 'grammar' of regulatory elements. To what extent the composition of regulatory sequences is determinant of their function is still an open question [42] – one that is investigated in this thesis.

Three models of TF interactions have been proposed: enhanceosome, billboard and TF collective. Examples of each model are shown in Figure 1.4. In the enhanceosome model, TFs interact in a highly cooperative manner. This model represents a situation where a strict TFBSs arrangement is essential for enhancer activity and changes in individual TFs affect the outcome [44]. The interferon-β enhancer is an example of an enhancer that follows the enhanceosome model. In contrast, the billboard model offers a more flexible arrangement of TFBSs. In this model, TFs interact in a relatively independent manner. The billboard model suggests that enhancers act as information displays (hence the term billboard), transmitting signals that are then decoded by the basal ma-

chinery and used to turn genes on or off [12]. Examples of enhancers following the billboard model are the even-skipped (eve) enhancers. The two models mentioned are the extreme ends of the spectrum. In a third model, called the TF collective model, similar to the enhanceosome model, TFs bind to enhancers in a cooperative manner; however, their cooperation does not require a specific arrangement, similar to the billboard model [45]. Ultimately, TF interactions fall on different points along this spectrum.

1.4 Conserved Non-coding Elements (CNEs)

Over 98% of the human genome does not encode proteins [2]. Non-coding DNA, once dismissed as 'junk' [46], has important functions. For instance, some non-coding DNA sequences are transcribed into non-coding RNAs, many of which are involved in translation. There exist regions of non-coding DNA that are under strong purifying selection. Properties of these conserved non-coding elements (CNEs) suggest that they may be involved in gene regulation; however, while the functions of a number of CNEs have been identified, the role of many CNEs and hence the reason for their extreme conservation remains a mystery. The following three sections provide a review of what is currently known about these elements.

1.4.1 Origins

CNEs are identified through comparative genomic analysis, whereby genomic sequences from two or more species are compared in order to identify orthologous regions that are evolutionarily conserved across the species being analysed. The idea behind this approach is that functional regions are under purifying selection since changes in these regions may reduce the organism's fitness; therefore, such sequences are expected to be more similar across species than other regions [47, 1]. For details of this approach, see Section 2.3.

Comparative sequence analysis of vertebrate genomes has led to the discovery of thousands of CNEs. Various criteria are used to define CNEs, including a minimum sequence similarity and a minimum sequence length. For example, ultraconserved elements (UCEs) are sequences of length 200bp or more that are 100% identical between the orthologous regions of the human, mouse and rat genomes [5]; or, long conserved non-coding sequences (LCNSs) are sequences longer than 500bp with at least 95% sequence identity [48]. Throughout this thesis, we refer to all of these sequences as CNEs.

CNEs were thought to be mutational cold spots; however, analysis of allele (variants of a gene) frequencies showed that new alleles of single nucleotide polymorphisms (SNPs) within CNEs are rarer than those within other regions, indicating that CNEs are under purifying selection [49, 50]. Despite being under strong purifying selection, CNE losses do occur [51] and may be accompanied by phenotypic changes [52].

CNEs have been found to originate from diverse sources, including transposable elements (DNA sequences that can jump from one location in the genome to another) [53] and ancient repeats [54].

Most CNEs have been identified in vertebrates. A small number of these elements are conserved across evolutionarily distant species. One example is the set of CNEs that are conserved between human and sea lamprey (*Petromyzon marinus*), a jawless fish which belongs to a group of vertebrates that separated from the jawed lineage around 600 million years ago [55]. Another example is the set of CNEs found in amphioxi (also known as lancelets), which last shared an ancestor with vertebrates over 520 million years ago [56]. Although primarily identified in vertebrates, CNEs have also been identified in invertebrates such as worms and insects [57, 58]. Moreover, hundreds of CNEs have been identified in plants [59, 60]. Although CNEs identified in different animal groups share little sequence similarity, the sets of developmental genes that they are associated with overlap [61].

The extreme conservation of CNEs over large evolutionary distances implies that these elements are likely to play an essential role during evolution.

1.4.2 General Properties

Analysis of the nucleotide composition of CNEs revealed a sharp drop in A+T frequency beyond the boundaries of these elements, meaning that CNEs are more rich in A+T content than their flanking sequences [62]. CNEs have also been shown to be enriched in motif TAATTA, which contains the core recognition motif TAAT for homeodomain proteins [63].

CNEs often appear in clusters near key developmental genes [7] and similar to TFs, function in a cooperative manner [64]. These clusters, referred to as genomic regulatory blocks (GRBs), can span up to several hundred kilobases around their target genes. Many GRBs contain gene deserts, i.e., regions of DNA that are devoid of genes. In addition to target genes, GRBs also contain other genes called bystander genes, whose regulation and functions are unrelated to those of the target genes. There exists evidence that there is evolutionary

pressure to keep GRBs intact [65, 66], suggesting that CNEs may act as long-range regulatory elements which are required to remain in *cis* with their target genes [67, 68, 69]. An example of such CNEs is the elements around the Sonic Hedgehog (SHH) gene [70].

More recently, GRBs were found to coincide with topologically associating domains (TADs) [71]. TADs are regions of the genome that prefer to interact with themselves rather than with regions outside the TAD [72]. TADs that overlap GRBs, termed GRB-TADs, have distinct features: compared to non-GRB-TADs, they are larger, gene-sparse, exhibit higher levels of self-interactions and are more insulated from neighbouring regions [71]. While still preliminary, this observation is consistent with the presence of selective pressure against disruption of GRBs and further supports the potential role of CNEs as long-range regulatory elements.

CNEs are rich in overlapping TFBSs [11], a feature of regulatory elements such as enhancers. In some cases, the position and order of CNEs within GRBs have remained intact [6], while in others, they have undergone shuffling [73]. The former observation is in agreement with the enhanceosome model which proposes that a strict arrangement of TFBSs within CNEs is required for their regulatory function, while the latter observation is consistent with the billboard model which suggests that the exact arrangement of TFBSs within CNEs is not necessary for their regulatory function.

Overall, the above features make CNEs good candidates for regulatory elements.

1.4.3 Functions and Unknown Reasons for Conservation

The majority of CNEs tested using functional assays and chromatin immunoprecipitation combined with massively parallel DNA sequencing (ChIP-seq) have been shown to act as enhancers, in that they drive tissue-specific gene expression *in vivo*, most commonly in mice [74, 9, 75] and zebrafish [8]. The VISTA enhancer browser [76] contains hundreds of these validated CNEs with enhancer activity.

A number of diseases have been linked to mutations in CNEs, including point mutations [77, 78, 79], deletions [80] and duplications [81, 82]. These diseases include malformations (for a list of examples, see [83]) and behavioural disorders [84, 85].

The above, however, do not fully explain the high levels of conservation

seen in CNEs. Sequence conservation does not necessarily imply regulatory function [86, 87], and not all regulatory elements are highly conserved [88, 89]. Moreover, deletion of some CNEs yielded viable mice with no significant phenotypic changes [90, 15], although it was recently demonstrated that deletion of these elements causes phenotypes, including reduced growth and neurological abnormalities, that may have been too subtle to be detected in a laboratory setting [91]. In addition, it is possible that CNEs which act as enhancers and are associated with the same gene have redundant activity in order to provide robustness to the loss of function that can result from their loss, and therefore, deletion of one CNE results in no or subtle phenotypes [92]. Nevertheless, such cases raise doubts about the functional importance of CNEs.

These opposing findings pose interesting questions regarding the functions of CNEs and the reasons for their extreme conservation, and indicate that further research on CNEs is required to understand their mechanisms of action.

Chapter 2

Computational Background

2.1 Biological Sequence Representation

DNA, RNA and protein sequences are essentially strings over a finite alphabet of symbols, where each symbol represents a structural unit of that biopolymer. Thus, one approach to representing these sequences is to use natural language processing (NLP) techniques such as word embedding [93]. For example, in [94], DNA sequences are segmented into variable-length k-mers and a distributed representation of these k-mers is computed.

Another approach to representing biological sequences is to use a graphical representation. 2D and 3D graphical representations of DNA sequences [95, 96, 97] and proteins [98] are derived from the mathematical denotation of the sequences and provide a simple way to analyse these sequences.

One data structure that has been used to represent biological sequences is a graph. For example, in [99], nucleotide sequences are represented by directed acyclic graphs (DAGs) called sequence graphs, in which nodes are labelled with nucleobase symbols and alternative paths in the graph correspond to different DNA sequences that encode the same amino acid sequence. In another example [100], nucleotide sequences are represented by graphs similar to, but more condensed than, sequence graphs called back-translation graphs. In back-translation graphs, IUPAC codes are used to join multiple nodes, each labelled with a nucleobase symbol, together into a single node labelled by an ambiguity character.

In Chapter 4, we introduce a new graph representation for CNEs.

```
GCATG-CU
G-ATTACA
```

Figure 2.1: Example of a pairwise alignment. Sequences are written in rows, creating a correspondence between the symbols in the same column.

2.2 Sequence Alignment

Sequence alignment is the most widely used method for comparing two (pairwise sequence alignment) or more (multiple sequence alignment) biological sequences and forms the core of many other methods in computational biology, including those developed to predict regulatory elements.

Definition 1. Given two sequences $S = s_1 s_2 \dots s_n$ and $T = t_1 t_2 \dots t_m$ over an alphabet A, an alignment between S and T is a pair of sequences $S' = {s'}_1 {s'}_2 \dots {s'}_l$ and $T' = {t'}_1 {t'}_2 \dots {t'}_l$ over the alphabet $A' = A \cup \{-\}$ with the condition that $\nexists k \in \{1, \dots, l\} : {s'}_k = {t'}_k = -$. If ${s'}_i, {t'}_i \in A$, then the aligned pair $({s'}_i, {t'}_i)$ corresponds to a match when ${s'}_i = {t'}_i$, and a mismatch otherwise. If one of the symbols is a gap (denoted by $-$), then the aligned pair corresponds to an indel (insertion or deletion).

An example of a pairwise alignment is shown in Figure 2.1. A score is assigned to each aligned pair; usually, matches are rewarded, while mismatches and gaps are penalised. Match and mismatch scores are often given by a scoring matrix (e.g., PAM [101] and BLOSUM [102] substitution matrices). There are various types of gap penalty; the simplest are the constant, linear and affine gap penalties. A constant gap penalty assigns a fixed cost e to each gap, regardless of its length and position in the alignment. A linear gap penalty takes into account the length of the gap (denoted by g) and has the form ge. An affine gap penalty combines the above two types of penalty and has the form $o + (g - 1)e$, where o and e are the gap opening penalty and the gap extension penalty (cost of extending the length of a gap by 1), respectively [103].

The score of an alignment is the sum of the scores of all the aligned pairs, and is computed using one of the following approaches: global, semi-global and local. Global alignment is an alignment over the entire length of the sequences. Local alignment aligns similar regions within the sequences. In semi-global alignment, sequences are globally aligned, but leading and trailing gaps are ignored. Once the desired approach is chosen, an optimal alignment is an alignment with the highest score among all the possible alignments.

2.2.1 Pairwise Alignment via Dynamic Programming

Pairwise alignment algorithms are divided into two main categories: dynamic programming-based methods and heuristic methods. Following is an overview of the classic alignment algorithms that belong to the first category and form the basis for the work presented in Chapters 4 and 5. They are the Needleman-Wunsch global alignment algorithm [104] and the Smith-Waterman local alignment algorithm [105].

The first step in dynamic programming-based methods is the construction and initialisation of a score matrix whose axes represent the sequences to be aligned, i.e., cell (i, j) corresponds to position i in the first sequence and position j in the second sequence (Figure 2.2). Next, this matrix is filled, from top to bottom and from left to right, by computing the score $M(i, j)$ for each cell as follows:

$$M(i,j) = max \begin{cases} M(i-1, j-1) + s(i,j) \\ M(i, j-1) + g \\ M(i-1, j) + g \end{cases} \tag{2.1}$$

where $s(i, j)$ is the match (or mismatch) score and g is the gap penalty. A diagonal move (first term in Equation 2.1) corresponds to aligning two symbols, while horizontal and vertical moves (second and third terms in Equation 2.1, respectively) correspond to inserting a gap in the first and second sequences, respectively.

The last step is to trace back the path that leads to an optimal alignment. This is done by starting from a cell and then backtracking, i.e., selecting the predecessors with the optimal score iteratively, until another cell is reached. The choice of which cell to start from and end at depends on the type of the alignment: in the Needleman-Wunsch algorithm, backtracking starts from the lower right corner of the matrix and ends at the top left corner of the matrix. In the Smith-Waterman algorithm, the first row and the first column of the score matrix are filled with zeros, and if $M(i, j)$ becomes negative, it is reset to zero. In this algorithm, backtracking starts at the cell with the highest score in the matrix and ends when a cell with a score of zero is reached.

Heuristic methods (e.g., FASTA [106] and BLAST [107]) follow the seed-and-extend paradigm: given a query sequence, they list words of length k from the query sequence, and search the database sequences for matching words called seeds. The matches are then extended using dynamic programming, and those whose score (computed using a scoring matrix) is greater than a given cut-off score are listed and their statistical significance is determined. Finally, matches

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Figure 2.2: Score matrix showing the paths corresponding to the optimal global alignments between `GCATGCU` and `GATTACA`. The path coloured in blue corresponds to the alignment shown in Figure 2.1. The match score, mismatch score and gap penalty are equal to 1, -1 and -1, respectively.

with expectation values smaller than a given threshold are returned. In contrast to dynamic programming-based methods, heuristic methods do not guarantee to find an optimal alignment between the sequences.

2.3 Identifying Cis-Regulatory Elements and Modules

Following is a survey of the computational methods for identifying *cis*-regulatory elements and modules. Today, many tools combine two or more of the following approaches.

One approach to identifying *cis*-regulatory elements is cross-species sequence comparison. The key assumption in comparative sequence analysis methods such as phylogenetic footprinting [108] is that sequences regulating the expression of orthologous genes (genes that are derived from a common ancestral gene and usually have the same function) are conserved across different species [109].

The first step in comparative sequence analysis is selecting the sequences that are to be compared from species with appropriate phylogenetic distances. Comparisons at long phylogenetic distances (e.g., >one billion years) allow one to distinguish between coding and non-coding sequences. An example of analysis at this distance is the comparison between the genomes of flies, worms and yeast [110]. Comparisons at moderate phylogenetic distances (e.g., 100 million years) separate functional from non-functional sequences. Examples of anal-

yses at this distance are the comparisons among several yeast species [111]. Comparisons at short phylogenetic distances (e.g., 5 million years) reveal the sequence differences that are responsible for differences between the compared species [112]. An example of analysis at this distance is the comparison between the genomes of human and chimpanzee [113].

The second step in comparative sequence analysis is annotating the reference sequences for known features such as repetitive DNA. For many species, this information is available in genome browsers (e.g., Ensembl [114]). After known features are located, the remaining regions are candidates for regulatory elements. The last step in comparative sequence analysis is aligning the sequences and visualising the alignments. Examples of tools commonly used in this step are VISTA [115] and PipMaker [116], which produce global and local alignments, respectively. The plots returned by these tools are examined to find regulatory elements.

Another approach to identifying *cis*-regulatory elements is searching for statistically over-represented motifs in sequences. Word counting methods fall in this category. Word counting methods consider sequences as text and count the number of occurrences of all nucleotide words (or oligonucleotides) of a defined length k. They then compare the observed frequency of each word with its expected frequency in a background model (e.g., a set of randomly generated sequences, a Markov chain model [117] or a lexicon [118]). In order to determine whether a word is over-represented, the significance of its observed versus expected frequency is evaluated using some criteria (e.g., Z-score).

A third approach to identifying *cis*-regulatory elements is using probabilistic models. In this approach, motifs are modelled as position probability matrices (PPMs) hidden in a noisy background sequence. The parameters of the model are found using algorithms such as expectation maximisation and Gibbs sampling. Examples of regulatory elements identified using this approach are human heart enhancers [119].

A fourth category of methods for identifying *cis*-regulatory elements rely on chromatin state features of regulatory elements (unlike the previous categories, which use sequence-based features). These methods use hidden Markov models (HMMs) [120], neural networks [121] and random forests [122]. A comparison between different approaches using various features showed that regulatory elements, in particular enhancers, can be accurately predicted using only sequence-based features (specifically, the TFBS occurrence-based features), and the prediction accuracy is further improved by adding epigenetic features [123].

Methods for identifying *cis*-regulatory modules (CRMs) rely on the spa-

tiotemporal relationships among TFBSs. A number of these methods employ HMMs to model CRMs as sequences generated from a set of TFBSs, where each state represents a motif [124]. Another group of these methods apply Bayesian inference to locate CRMs [125, 126].

2.4 Kernel Methods

In computational biology, kernel methods have been successfully employed to discover patterns of regulatory activity. The two components of a kernel method are: a kernel function and a learning algorithm [127, 128].

Definition 2. Given a non-empty set of objects $\mathcal{X}$, a kernel function (or kernel in short) $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a similarity measure for which there exists a feature map $\Phi : \mathcal{X} \to \mathcal{H}$ that maps objects into a dot product space (also called a feature space) $\mathcal{H}$ satisfying

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \tag{2.2}$$

for all $x, x' \in \mathcal{X}$. For a kernel to be valid, it must be positive semi-definite.

Definition 3. Given a kernel k on $\mathcal{X} \times \mathcal{X}$ and its Gram matrix $K_{ij} := (k(x_i, x_j))$, k is positive semi-definite if

$$\sum_{i,j} c_i c_j K_{ij} \geq 0 \tag{2.3}$$

for all $x_i, x_j \in \mathcal{X}$ and $c_i, c_j \in \mathbb{R}$. Throughout this thesis, we refer to positive semi-definite kernels simply as kernels.

The set of kernels is a closed convex cone, i.e., it is closed under addition, multiplication by a positive constant and pointwise limits. It is also closed under product. These properties make it possible to build new kernels from existing ones. One such family of kernels is known as convolution kernels.

Definition 4. Suppose that $x \in X$ is a composite structure (e.g., a string) and $\bar{x} = x_1, x_2, \ldots, x_D$ ($x_d \in X_d, 1 \leq d \leq D$) are its parts. Let relation R_x over $X_1 \times X_2 \times \ldots \times X_D$, where $\bar{x} \in R_x$, denote that $\bar{x}$ are the parts of x. Suppose that $x' \in X$ is another composite structure and kernel $K_d(x_d, x'_d)$ measures the similarity between x_d and x'_d. If $K_1, K_2, \ldots, K_D$ are kernels on $X_1 \times X_1, X_2 \times X_2, \ldots, X_D \times X_D$, respectively, then the convolution

$$K_1 \star K_2 \star \ldots \star K_D := \sum_{\substack{\bar{x} \in R_x \\ \bar{x}' \in R_{x'}}} \prod_{d=1}^{D} K_d(x_d, x'_d) \tag{2.4}$$

is a kernel on $X \times X$ [129].

Many learning algorithms are used in combination with kernels. For details of these methods, see [127].

2.4.1 Kernels for Biological Sequences

Kernels have been widely used, often combined with support vector machines (SVMs), to solve computational biology problems such as protein homology detection. A kernel can be explicitly built by extracting features from the sequences and forming the kernel as the dot product of the feature vectors.

Kernels based on probabilistic models such as the marginalised kernels [130] assume that sequences are generated by a latent variable model (e.g., a HMM). The idea behind these approaches is to first, build a joint kernel over both visible and latent variables (which are assumed to be available), and then obtain a kernel by taking the expectation of the joint kernel with respect to the latent variables (marginalising). A special case of marginalised kernels, called the Fisher kernel [131], uses the gradient of the log-likelihood of the sequence with respect to the probabilistic model with a given set of parameters as features.

String kernels such as the spectrum kernel [132] and its variants [133] use counts of the number of occurrences of k-mers in the sequences as features. The oligo kernel [134] and the weighted degree (WD) kernel [135] and its variant [136] are similar to the spectrum kernel, but they also use positional information, i.e., given a sequence of length l, they consider all the k-mers starting at positions $i = 1, 2, \ldots, l$. Another kernel, called the motif kernel [137], similarly uses counts of the number of occurrences of motifs from a database as features.

The two frameworks of sequence alignment and kernel methods were unified in [138], where the feature vector corresponding to a sequence consists of the E-values of the Smith-Waterman alignments between that sequence and all the sequences in the training set (the E-value or the expectation value of an alignment measures its significance. For details of how E-value is calculated, see [139]). A similar kernel, called the local alignment (LA) kernel [140, 141], computes an exponentially weighted sum of the scores of all the possible local alignments between the sequences.

The convolution kernels and the string kernels described above are all special cases of the rational kernels [142]. In Chapter 5, we introduce a new convolution kernel for comparison of strings.

Chapter 3

Comparative Evaluation of Methods for Grouping Functionally Related CNEs

In this chapter, we investigate the use of a number of existing methods for grouping CNEs into clusters of functionally related elements based on the TFBSs they contain. The assumption here is that CNEs having the same combination of TFBSs are likely to be involved in regulating genes with similar expression profiles, as explained in Section 1.3.

We convert each CNE into a feature vector, where the features are based on the occurrences of TFBSs within that CNE. We then use different metrics to measure the similarities (or distances) between these vectors. Next, we group CNEs based on the similarity of their corresponding feature vectors. Finally, we check whether the CNEs that belong to the same cluster are similar in terms of their regulatory function. We test this approach on a set of CNEs as follows: to group CNEs, we consider five clustering algorithms (four hierarchical clustering algorithms and a spectral clustering algorithm), a dimensionality reduction technique (t-SNE [143]) and a network topology visualisation tool (BOSAM [144]). These methods are widely used in practice and form the basis of numerous other methods. Among the clusters returned by each method, we select those that are compact. We then obtain the gene expression data for CNEs in the selected clusters by performing functional assays in zebrafish, and compare these data to the clustering results. We discuss the shortcomings of the above approach and highlight the need for a new method for comparing CNEs.

3.1 Related Work

Alignment-free methods for sequence comparison model sequences based on the presence and frequency of subsequences (also called words) that they contain, and use a similarity measure (or a distance function) to compare the models [145].

These methods commonly represent sequences by frequency vectors, where each entry corresponds to the number of occurrences of a matched word. Some methods take only exact word matches [146, 147] into account, while others allow approximate matches with a bounded number of mismatches [148]. Another category of methods use patterns called spaced words. Positions in a spaced word are either a 'match' or a 'don't care' position; two words are considered a match if they contain matching symbols in the match positions [149]. A third category of these methods measure the similarity between a pair of sequences based on the lengths of their common subsequences [150, 151], i.e., for each position in one sequence, the length of the longest subsequence starting at that position and matching some subsequence starting at any position in the other sequence is considered. Some methods, instead, consider the length of the shortest subsequence [152].

Various similarity measures and distance functions are used to compare frequency vectors. They range from the simple Euclidean distance and its variants (e.g., D^2 score [153]), statistical measures based on correlation and covariance (e.g., the Pearson correlation coefficient [154] and the Mahalanobis distance [155]) to cosine similarity and information theoretic measures (e.g., the Kullback-Leibler divergence [156]).

3.2 Sequence Similarity Measures

We measure the similarity between CNEs in a way that takes the presence and number of TFBSs into account, and is compatible with the billboard model of TF interactions (for descriptions of different models of TF interactions, see Section 1.3.2). Given a set of TFBSs $T = \{t_1, t_2, \ldots, t_n\}$, we represent a CNE S by a vector $V_S = \langle v_1, v_2, \ldots, v_n \rangle$, where v_i is the number of times TFBS t_i has been identified in S. For each pair of vectors, we compute two types of distance: the Euclidean distance and the Jensen-Shannon distance, which is defined as the square root of the Jensen-Shannon divergence (JSD).

In detail, let X and Y be two CNEs, represented by vectors V_X and V_Y, respectively. The TFBS-occurrence vectors V_X and V_Y are normalised to form

probability distributions P and Q, respectively. This cannot be done if $V_X = \vec{0}$; in that case, we consider $P(i) = 0\ \forall i$. Similarly, if $V_Y = \vec{0}$, then we consider $Q(i) = 0\ \forall i$. We compute an unnormalised Jensen-Shannon divergence using

$$UJSD(P\|Q) = \frac{1}{2}UKLD(P\|M) + \frac{1}{2}UKLD(Q\|M) \tag{3.1}$$

where $M = \frac{1}{2}(P + Q)$ and $UKLD(P\|M)$ is the unnormalised Kullback-Leibler divergence [157], defined as

$$UKLD(P\|M) = \sum_i P(i) \ln \frac{P(i)}{M(i)} + \sum_i (M(i) - P(i)) \tag{3.2}$$

$UKLD(Q\|M)$ is defined similarly. If $M(i)$ is zero for some i, i.e., when both $P(i)$ and $Q(i)$ are zero or, in other words, TFBS t_i has not been identified in either X or Y, then we consider $P(i) \ln \frac{P(i)}{M(i)} = Q(i) \ln \frac{Q(i)}{M(i)} = 0 \ln \frac{0}{0} = 0$. Note that when $V_X \neq \vec{0}$ and $V_Y \neq \vec{0}$, i.e., P and Q are probability distributions, $UJSD$ becomes the classic Jensen-Shannon divergence as $UKLD$ becomes the classic Kullback-Leibler divergence (see Equation 3.10).

Another way to compare two CNEs based on the TFBSs they contain is to identify the sequence of TFBSs in each CNE, align these sequences and then compare the CNEs according to the score of an optimal alignment between their corresponding TFBS sequences. Such an alignment-based similarity measure takes the presence, number and relative position of TFBSs into account, and is consistent with the enhanceosome model of TF interactions. In Chapter 4, we present an algorithm for aligning a pair of TFBS sequences.

3.3 Methods

3.3.1 Clustering

In computational biology, clustering algorithms have traditionally been used to partition genes on the basis of their expression profiles, and then identify regulatory elements in the obtained clusters [158, 159]. Our approach follows the opposite direction, i.e., we use clustering algorithms to group CNEs based on their TFBSs composition, and then validate the expression profiles of elements in the obtained clusters. For a similar approach applied to genes, see [160]. We consider two types of clustering algorithms: hierarchical clustering and spectral clustering. Following is an overview of these algorithms.

Algorithm	α_i	β	γ
Single linkage	0.5	0	-0.5
Average linkage	$\frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}$	0	0
Complete linkage	0.5	0	0.5
Ward's	$\frac{\lvert i\rvert+\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$	$-\frac{\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$	0

Table 3.1: Lance-Williams update formula parameters for the three linkage methods and the Ward's method

Hierarchical Clustering

For hierarchical clustering, we choose three linkage methods (single-linkage, average-linkage and complete-linkage) and the Ward's method [161]. All four algorithms can be described by the Lance-Williams update formula [162]. Let i and j be two clusters (including singletons) that have been agglomerated into cluster $i \cup j$. The Lance-Williams update formula for computing the distance between this cluster and other clusters is defined as

$$d(i \cup j, k) = \alpha_i d(i, k) + \alpha_j d(j, k) + \beta d(i, j) + \gamma |d(i, k) - d(j, k)| \tag{3.3}$$

where α_i, α_j, β and γ are parameters. Table 3.1 lists the values of these parameters for each algorithm.

For the linkage methods, we use the Jensen-Shannon distances. In the Ward's method, the initial distances are the squared Euclidean distances between feature vectors.

Spectral Clustering

The three main variants of spectral clustering are unnormalised, symmetric normalised [163] and asymmetric normalised [164]. Here, we use the asymmetric normalised variant. The reasons for our choice are as follows: first, normalised spectral clustering implements both clustering objectives, namely maximising intra-cluster similarity and minimising inter-cluster similarity, while unnormalised spectral clustering implements only the latter. Second, normalised spectral clustering is consistent, while unnormalised spectral clustering may fail to converge. Lastly, in symmetric normalised spectral clustering, eigenvectors of the Laplacian are multiplied by an additional factor which can lead to undesired effects [165]. Below is the description of asymmetric normalised spectral clustering from the graph cut point of view (the others being the random walk and

perturbation theory points of view).

Given a set of objects and their pairwise similarities (we convert the Jensen-Shannon distances to similarities using a Gaussian kernel and use them as the initial similarities), a similarity graph G is constructed. In this graph, each node represents an object. Two nodes are connected if the similarity between their corresponding objects is positive or above a certain threshold, and the edge connecting them is weighted by that similarity. The objective is to find a partition of G such that edges within the partition have high weights (high intra-cluster similarity) and edges between the partitions have low weights (low inter-cluster similarity); in other words, to find partitions $A_1, \ldots, A_k$ that minimise

$$cut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} S(A_i, \bar{A}_i) \tag{3.4}$$

where $\bar{A}$ is the complement of A, and $S(A, B)$ is the sum of weights of edges with one node in A and one node in B. To achieve balanced clusters, the objective function is defined as

$$Ncut(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \tag{3.5}$$

where $vol(A)$ is the sum of weights of all edges connected to nodes in A.

The minimisation of $Ncut(A_1, \ldots, A_k)$ is an NP-hard problem; however, by relaxing the discrete constraint on the indicator functions of the partitions, i.e., allowing them to take real values, this problem reduces to minimisation of the Laplacian of G.

The first step in asymmetric normalised spectral clustering is computing the Laplacian of the graph. Let W and D be the weighted adjacency matrix and the degree matrix of G, respectively. The Laplacian is defined as $L = D - W$. The next step is finding the first k eigenvectors $u_1, \ldots, u_k$ of $Lu = \lambda Du$. Eigenvalues are sorted in the ascending order while respecting their multiplicity, and therefore, the eigenvectors corresponding to the k smallest values are selected. Let U be the matrix containing these vectors as columns and let y_i be the i-th row of U.

The final step is grouping $(y_i)_{i=1,\ldots,n}$ into clusters $C_1, \ldots, C_k$ using the k-means algorithm. We use the eigengap heuristic technique for determining the number of clusters k for the k-means algorithm. Given eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{n-1}$, k is chosen such that eigengaps $|\lambda_i - \lambda_{i-1}|$ $(i \leq k)$ are small and eigengap $|\lambda_{k+1} - \lambda_k|$ is large. The output is clusters $A_1, \ldots, A_k$ with $A_i =$

$\{j|y_j \in C_i\}$.

Cluster Validation

To evaluate the quality of a clustering, we use the silhouette coefficient. The silhouette coefficient is an internal clustering evaluation measure which considers both cohesion and separation of clusters. Given a data point i, its silhouette coefficient is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3.6}$$

where $a(i)$ is the average distance between i and data points in the same cluster, and $b(i)$ is the minimum of the average distances between i and data points in other clusters. The silhouette coefficient of a cluster is equal to the average $s(i)$ over all data points i in that cluster. Values close to 1 indicate good clustering, whereas values close to -1 indicate poor clustering [166].

In the case of hierarchical clustering algorithms, we cut each dendrogram at different levels to obtain different partitions (with up to 100 clusters), and choose the clustering that yields the maximum mean silhouette coefficient.

To compare two partitions, we use the adjusted Rand index (ARI). Given a set of n objects S, suppose that $U = u_1, \ldots, u_N$ and $V = v_1, \ldots, v_M$ are two different partitions of S such that $\cup_{i=1}^{N} u_i = \cup_{j=1}^{M} v_j = S$ and $u_i \cap u_j = v_i \cap v_j = \emptyset$. Let n_i and n_j be the number of objects in clusters u_i and v_j, respectively, and let n_{ij} be the number of objects that belong to both clusters. The ARI is defined as

$$ARI = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}} \tag{3.7}$$

The expected and maximum values of the ARI are 0 and 1 (when the two partitions are exactly the same), respectively [167].

3.3.2 Dimensionality Reduction

We use the t-distributed stochastic neighbour embedding (t-SNE) [143] to visualise how the vector representations of CNEs are distributed in the feature space. t-SNE is a dimensionality reduction technique for visualisation of high-dimensional data. For example, it has been employed to analyse transcriptome data [168, 169, 170]. The t-SNE algorithm aims to reveal the global structure

of the data in a low-dimensional space (also called a map), while preserving its local structure. In addition, in contrast to the clustering algorithms reviewed in the previous section, using t-SNE we do not need to specify the number of clusters beforehand. We tested t-SNE in a computer vision task, namely gender recognition using skin texture features, where it successfully displayed individual subjects as separate points, while grouping samples from the same subject together. For a summary of this work, see Appendix A. Below is a summary of the t-SNE algorithm.

Let x_i and x_j be two data points in the high-dimensional input space. The similarity between them is given by the joint probability distribution P with values

$$p_{ij} = \frac{\exp(-d^2(x_i, x_j)/\sigma^2)}{\sum_k \sum_{l \neq k} \exp(-d^2(x_k, x_l)/\sigma^2)} \qquad \forall i, j : i \neq j \tag{3.8}$$

where $d^2(x_i, x_j)$ is the distance between x_i and x_j (normally the Euclidean distance). Here, we use the Jensen-Shannon distances as the initial distances. The similarity between the counterparts of x_i and x_j in the low-dimensional target space, denoted by y_i and y_j, respectively, is computed using a heavy-tailed Student t-distribution Q (with one degree of freedom).

$$q_{ij} = \frac{(1 + d^2(y_i, y_j))^{-1}}{\sum_k \sum_{l \neq k} (1 + d^2(y_k, y_l))^{-1}} \qquad \forall i, j : i \neq j \tag{3.9}$$

To find data points y that reflect the similarities p, t-SNE minimises the Kullback-Leibler divergence of P and Q:

$$KLD(P \| Q) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.10}$$

Tuneable parameters of t-SNE are the dimension of the embedding space, perplexity and the number of iterations. The so-called perplexity is the number of neighbours each data point in the high-dimensional space is considered to have, and determines the balance between the local and global structures of the data that will be preserved. The t-SNE algorithm is robust to the choice of the perplexity value [143]. We set the dimension, perplexity and the number of iterations to 3, 30 and 1000, respectively.

3.3.3 Network Topology Visualisation

Network analysis methods have been commonly used to study gene regulatory networks, mostly in *Escherichia coli* (E. coli) and yeast. For a survey of these methods, see [171]. Here, we use a simple and effective network visualisation tool called the bitmap of sorted adjacency matrix (BOSAM) [144], to visualise the network of CNEs. BOSAM reveals the topological structure of a network based on its adjacency matrix, and has been employed to characterise various types of networks, including protein-protein interaction networks [144]. We used BOSAM to perform an analysis of the structure of user interaction networks on health-related message boards, where the BOSAM of each network closely correlated with its characteristics. For a summary of this work, see Appendix B. Below is a summary of how the BOSAM of a network is generated.

Given an undirected network containing n nodes with indices $1, 2, \ldots, n$, let A be the network's adjacency matrix, where entry a_{ij} is 1 if there is an edge connecting nodes i and j, and 0 otherwise. Matrix A can be represented as a black and white bitmap where a pixel at coordinate (i, j) is black if $a_{ij} = 1$, and white otherwise. When the indices of nodes are arbitrary, this bitmap looks like a collection of random points; however, nodes can be sorted in a way that their indices correspond to their connectivity. To achieve this, nodes are sorted in the ascending order of their degrees, i.e., the number of edges they are connected to. If two nodes have the same degree, then they are ranked in the ascending order of the largest degree of their neighbours; ties are broken by looking at the neighbour with the next higher degree. Nodes are then re-indexed accordingly. The bitmap of the reordered adjacency matrix is the network's BOSAM. The BOSAM of a network often exhibits recurring fractal patterns, the components of which are related to the statistical properties of the network [172].

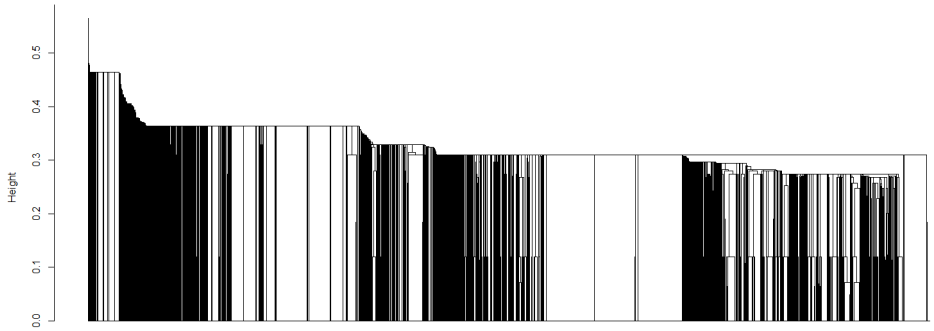
In the CNE network that we created, each CNE is represented by a node, and two nodes are connected if their corresponding CNEs have at least one TFBS in common.

3.4 Experiments

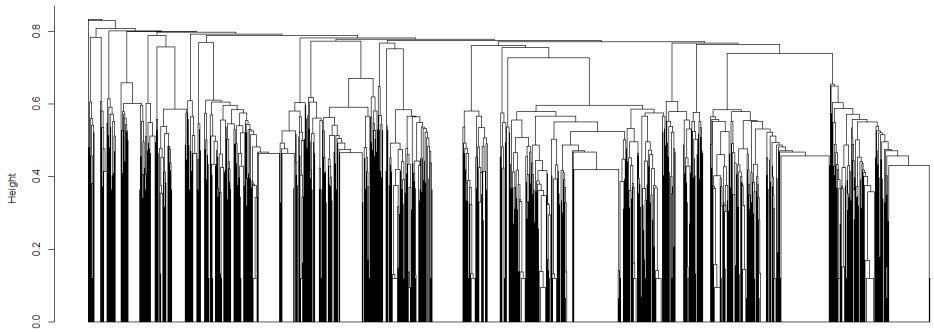
We applied the methods reviewed in Section 3.3 to a set of 5138 candidate human CNEs, acquired from the UCSC Genome Browser [173]. For the binding sites, we considered a set of 31 TFBSs for TFs associated with developmental patterning. For details of these TFBSs, see Section 4.5. The locations of the TFBSs within the CNEs were identified using FIMO [174] (p-value $\leq$ 1E-5).

3.4.1 Results

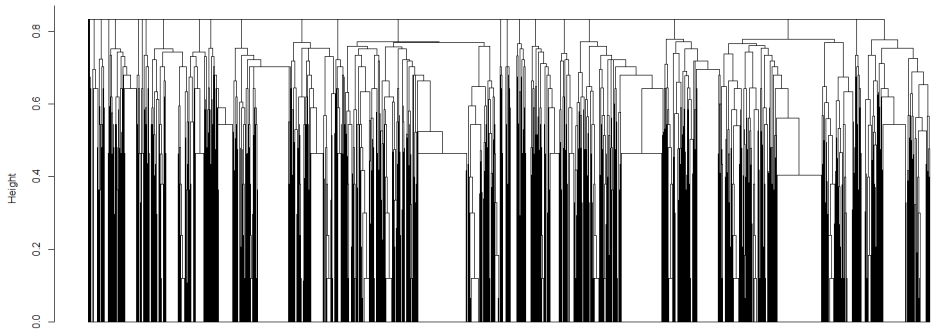
The results of clustering CNEs using the hierarchical clustering algorithms are shown in Figure 3.1. The maximum mean silhouette coefficients obtained for the single-linkage, average-linkage and complete-linkage methods and the Ward's method are 0.10, 0.30, 0.38 and 0.30, respectively. The number of clusters in these partitions is 2 (the lowest number of clusters considered) for the single-linkage method and 100 (the highest number of clusters considered) for the other methods. According to the silhouette coefficients, these hierarchical clustering algorithms did not produce a good clustering.



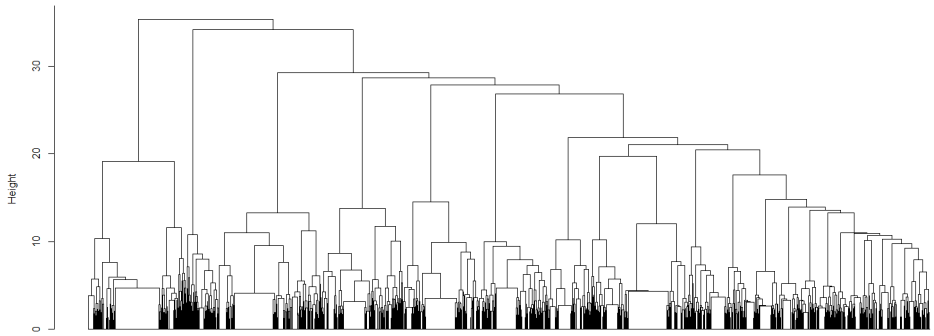
(a)



(b)



(c)



(d)

Figure 3.1: Results of clustering CNEs using the (a) single-linkage method, (b) average-linkage method, (c) complete-linkage method and (d) Ward's method

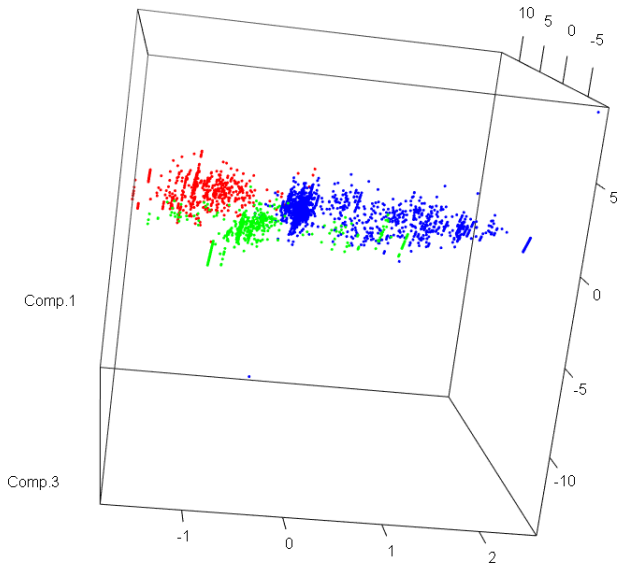


Figure 3.2: Spectral clustering of CNEs. Points are coloured according to the result of the k-means algorithm.

The result of clustering CNEs using spectral clustering is shown in Figure 3.2. For spectral clustering, the optimal number of clusters estimated using the eigen-gap heuristic technique is k=3, and the mean silhouette coefficient obtained for three clusters is 0.47. The low silhouette coefficient suggests that asymmetric normalised spectral clustering did not produce a good clustering either.

The result of visualising CNE feature vectors using the t-SNE algorithm and the BOSAM of CNE network are shown in Figures 3.3 and 3.4, respectively. In both figures, several groups of CNEs appear to form clusters. We selected the six largest clusters observed in the t-SNE map and compared them to the six largest clusters displayed in the BOSAM using the ARI. The ARI for these two partitions is 0.98; hence, these clusters are nearly identical.

Validation by Functional Assays

We randomly chose one of the six clusters and from this cluster, we randomly picked 10 CNEs for functional validation. The enhancer activities of the selected CNEs were detected using functional assays in zebrafish as follows: we amplified the orthologues of these CNEs from the zebrafish genomic DNA by polymerase chain reaction (PCR). The PCR products were cloned into plasmids using a TA cloning strategy (TOPO TA Cloning Kit, Invitrogen, USA) and transferred to an expression vector containing the green fluorescent protein (GFP) by LR recombination reaction. The plasmids were isolated using the QIAprep Spin Miniprep Kit (QIAGEN, Germany) and verified by Sanger sequencing. The

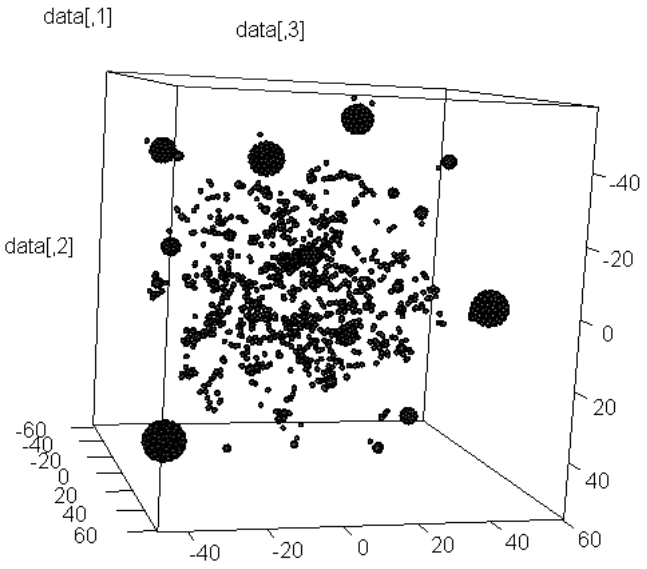


Figure 3.3: Result of visualising CNE feature vectors using t-SNE

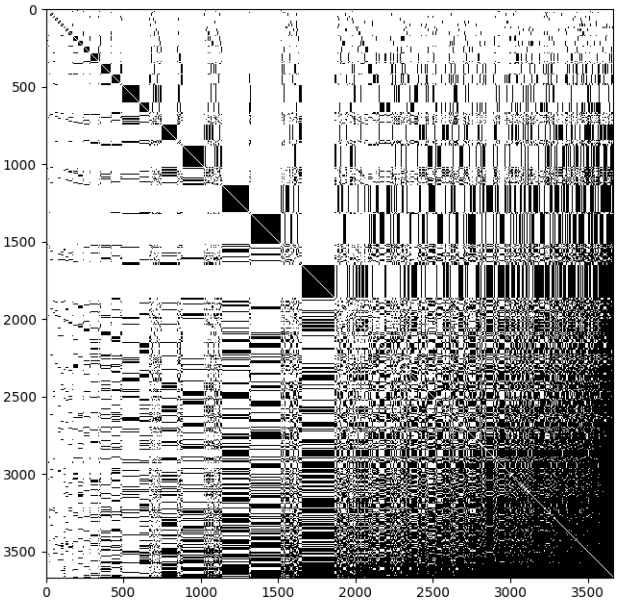
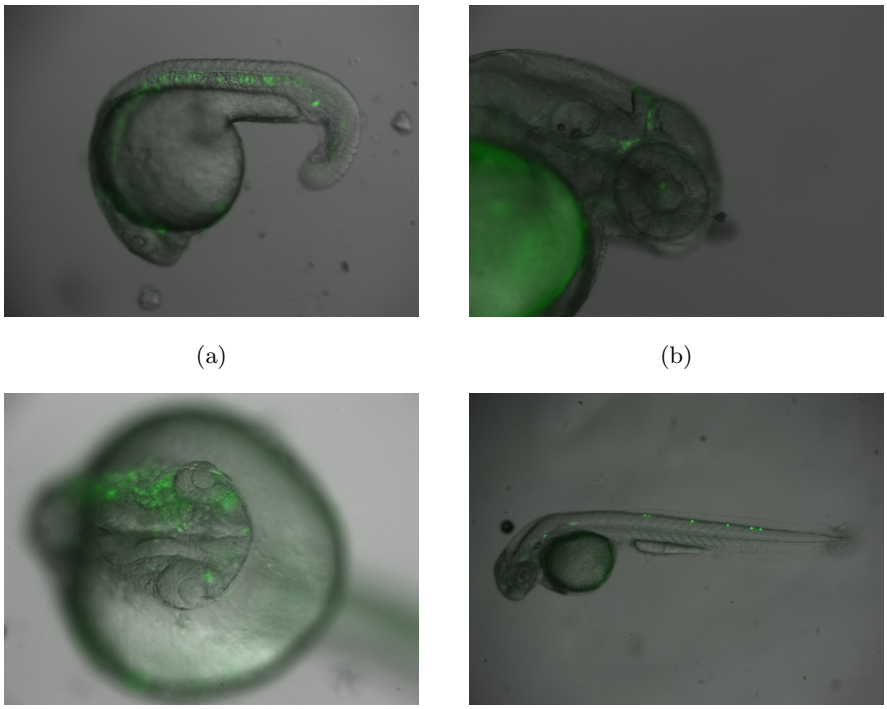


Figure 3.4: BOSAM of the CNE network



(a) (b)

(c) (d)

Figure 3.5: Expression driven by the assayed CNEs (a) CRCNE00008385, (b) CRCNE00006282 and (c)(d) CRCNE00003046 in different tissues of zebrafish (the elements are named after their accession numbers in the CONDOR database [175])

expression constructs were injected, together with Tol2 transposase mRNA and phenol red, into zebrafish embryos. The embryos were incubated at 28°C and screened for GFP expression at 24 and 48 hours post fertilisation (hpf) using fluorescence microscopy. We considered a CNE to be an enhancer if at least 20% of the embryos were GFP-positive. The detailed protocol used for the assays is provided in the supplementary materials.

The results show expression in very different tissues, including the neural crest, notochord and brain (Figure 3.5), and hence, do not suggest a similarity between the validated CNEs in terms of their regulatory function. Therefore, t-SNE and BOSAM also did not yield groups of functionally related CNEs. Furthermore, while the partitions obtained using t-SNE and BOSAM agree with each other, CNEs in each of the six clusters contain only one TFBS, indicating that the similarity measures used here do not capture the combinatorial nature of TF interactions.

Overall, the results show that the regulatory grammar of CNEs is complex, and a metric defined based on the presence and number of TFBSs alone does not provide a good measure of similarity between CNEs. This highlights the need for a metric which incorporates other factors such as the order of TFBSs and

the distances between them.

3.5 Availability

The following are included in the supplementary materials: the set of CNEs and the positions of identified TFBSs within each CNE, the matrices containing the pairwise distances (or similarities) between the CNEs, the scripts for running all the methods, and the results of functional assays. Related works using the methods employed in this chapter on other types of data have appeared in

- Bianconi F, Smeraldi F, Abdollahyan M, Xiao P. On the use of skin texture features for gender recognition: an experimental evaluation. In: Proceedings of the 6th International Conference on Image Processing Theory, Tools and Applications (IPTA); 12–15 December 2016. pp. 1–6.
- Abdollahyan M, Mondragón R, Bessant C and Smeraldi F. Visualising the Topological Structure of Health-related Message Board User Networks. Proceedings of the 1st International Conference on Applications of Intelligent Systems (APPIS) 2018. Frontiers in Artificial Intelligence and Applications. 2018;310:274–279.

Chapter 4

Identifying Regulatory Signatures in CNEs Using Transcription Factor Binding Site (TFBS) Alignment

In Sections 1.3 and 2.3, we explained how gene regulation is mediated by TFs that bind to DNA in a cooperative manner, and therefore, searching for multiple TFBSs located in close proximity can lead to the discovery of regulatory elements. In this chapter, we introduce an approach to identifying regulatory sequence signatures composed of over-represented co-occurring TFBSs in CNEs, which can be used to prioritise elements for functional assays.

We do not directly use the DNA sequence of CNEs; instead, we consider the sequence of TFBSs in a CNE. The idea is to align such TFBS sequences and find the short subsequences that are frequently matched in the alignments, i.e., co-occurring TFBSs. Analysis of the co-occurrence of TFBSs is complicated by the fact that binding sites may overlap. This rules out the use of classic alignment algorithms [104, 105], which cannot handle overlapping subsequences, and k-mer-based methods, which count the occurrences of subsequences and would enumerate overlapping subsequences indiscriminately. To address this problem, we use partial order graphs to handle overlapping TFBSs and represent the sequence of TFBSs in a CNE as a directed acyclic graph (DAG). We then find the optimal alignment between two sequences of TFBSs by aligning their corresponding DAGs using a dynamic programming-based alignment algorithm, originally developed in the context of multiple sequence alignment. This reduces the effect of spurious matches that are unlikely to occur in the same order in

multiple sequences, while taking into account the spatial order of TFBSs. To identify over-represented co-occurring TFBSs, we measure the relative frequency of aligned TFBSs with respect to a background model. We evaluate our approach on a set of functionally validated CNEs.

4.1 Related Work

The assumption that TFBSs which appear in clusters are more likely to act as regulatory elements than solitary binding sites is the rationale behind numerous methods for identifying *cis*-regulatory elements and modules, as covered in Section 2.3.

Methods such as Ahab [176] and MSCAN [177] use a sliding window to scan the sequences for putative TFBSs. Each window that contains multiple hits is assigned a score. Windows containing statistically significant hits (compared to windows in a background model) are potential regulatory regions. For details of how the scores are calculated and their significance is determined, see [176, 177].

Other tools such as Cluster-Buster [178], MCAST [179] and MORPH [180], use probabilistic models. They estimate a log-likelihood ratio for each subsequence in the sequence, i.e., the likelihood that the subsequence was generated by a cluster model, rather than a background model. The cluster model consists of a distribution of motifs based on position weight matrices (PWMs). A PWM specifies the probability distribution of nucleotides at each position in the motif. Subsequences with log-likelihood ratios higher than a specific threshold are candidates for regulatory elements.

Our approach differs from the above in that CNEs are not directly aligned; instead, we align the sequences of TFBSs identified in the CNEs, ignoring the differences in the distances between adjacent binding sites. Another advantage of this approach is that, in comparison, it has fewer parameters. A similar approach was presented in [181]. The difference between our approach and theirs is that we handle the uncertainty (due to overlapping TFBSs) in the sequences, while they do not handle this.

4.2 Graph Representation of CNEs

Given a conserved non-coding sequence $S = s_1 s_2 \dots s_n$ over the alphabet $N = \{A, T, C, G\}$, its graph representation is constructed as follows: first, we assign

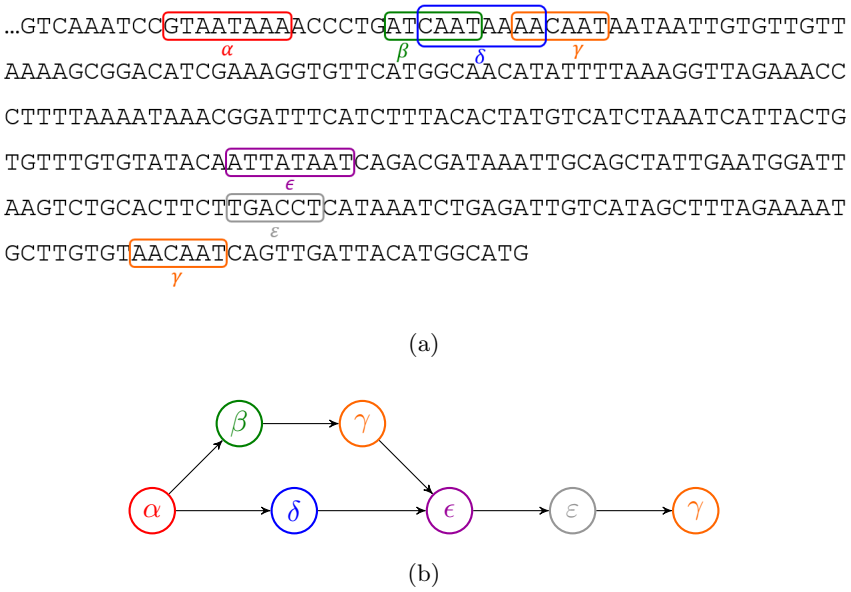


Figure 4.1: (a) An example CNE with seven example TFBSs. TFBSs β and γ overlap with TFBS δ. (b) Graph representation of the CNE shown in (a). Nodes have the same colour as their corresponding binding sites.

a symbol to each TFBS identified in S to obtain the partially ordered multiset $T = \{t_1, t_2, \ldots, t_m\}$. This partial order reflects the relative positions of the TFBSs in S, i.e., $t_i \prec t_j$ $(i \neq j)$ if and only if in S, every nucleotide in t_i comes before every nucleotide in t_j. Next, we transform this set into a DAG G. For each symbol in T, we create a node and label it with that symbol. In cases where the same TF binds to overlapping sites, only a single node is created. We add an edge between two nodes if their corresponding symbols are consecutive in T, i.e., $t_i \prec t_j$ and $\nexists k \in \{1, \ldots, m\} : t_i \prec t_k \prec t_j$. In this graph, each path from a source to a sink node (there may exist multiple source/sink nodes since G can start/end with overlapping nodes) corresponds to a sequence of non-overlapping TFBSs that were identified in S. Figure 4.1 shows an example of the graph representation of a CNE. A more formal definition of this representation is provided in Section 5.2.

Note that in contrast to usual graph representations of biological sequences such as those presented in [99] and [100], the alternative paths in a CNE graph are not necessarily of equal length.

4.3 Partial Order Alignment of CNE Graphs

We use the partial order-partial order (PO-PO) alignment algorithm [182] to find the optimal alignment between a pair of DAGs, each representing a CNE.

The PO-PO algorithm is a generalisation of the partial order alignment (POA) algorithm [183], which was proposed as an approach to multiple sequence alignment (MSA). In [183], linear representation of an MSA was replaced by a DAG called a partial order MSA (PO-MSA), and classic dynamic programming-based alignment algorithms [104, 105] were modified to find the optimal alignment between a sequence and a PO-MSA. This involved adding the branches of the PO-MSA as additional 'surfaces' to the score matrix (for an example of a score matrix, see Figure 2.2). The set of possible moves at each position in the matrix was extended accordingly to allow moves to any surface at junctions between the surfaces. The PO-PO algorithm generalised the above approach to align two PO-MSAs. Here, we reformulate this algorithm in a graph framework as a dynamic programming approach to finding an optimal path (corresponding to an optimal alignment) in the strong product graph of two DAGs.

We denote the node set and the edge set of a DAG G by $V(G)$ and $E(G)$, respectively. A directed edge from node u to node v is written as uv. Given two DAGs G_1 and G_2, their strong product graph $G_1 \boxtimes G_2$ has node set $V(G_1) \times V(G_2)$, where nodes (v_1, v_2) and (u_1, u_2) are connected if and only if for $k \in \{1, 2\}$ either $v_k = u_k$ or $v_k u_k \in E(G_k)$ (e.g., in Figure 4.2c, nodes (α, ε) and (α, β) are connected since they share node α and $\varepsilon\beta$ is an edge in the DAG shown in Figure 4.2b, while nodes (α, ε) and (γ, β) are connected since $\alpha\gamma$ and $\varepsilon\beta$ are edges in the DAGs shown in Figures 4.2a and 4.2b, respectively). In this graph, which generalises the score matrix, each path corresponds to an alignment of a path in G_1 against a path in G_2. The objective is to find the path with the optimal alignment score in the set of all paths in $G_1 \boxtimes G_2$. This requires finding the move (incoming edge) with the optimal score at every node (m, n) in $G_1 \boxtimes G_2$. Possible moves are aligning two symbols with substitution score $s(m, n)$ and indels with gap penalty g. Nodes in G_1 and G_2 may have multiple predecessors, and hence, when computing score $S(m, n)$ of a node (m, n), all possible incoming edges must be considered:

$$S(m,n) = \max \begin{cases} S(p,q) + s(m,n) & pm \in E(G_1) \text{ and } qn \in E(G_2) \\ S(m,q) + g & qn \in E(G_2) \\ S(p,n) + g & pm \in E(G_1) \end{cases} \tag{4.1}$$

In the case of sequences that do not contain overlapping TFBSs, the corresponding DAGs do not branch and m and n can be thought of as simply positions in the sequences, with the product graph reverting to a score matrix. Tracing the path that leads to the optimal alignment is done in the same way as in classic alignment algorithms: for global alignments, back-tracking starts

from node (m, n), where m and n are sink nodes in G_1 and G_2, respectively; for semi-global alignments, back-tracking starts from the highest scoring node (m, n), where m or n is a sink node. Starting from the chosen start node, the optimal alignment is traced back along the product graph to node (s_1, s_2), where s_i is a source node in G_i for at least one $i \in \{1, 2\}$ (semi-global alignment) or both (global alignment). Figure 4.2 shows an example of the PO-PO alignment between two DAGs.

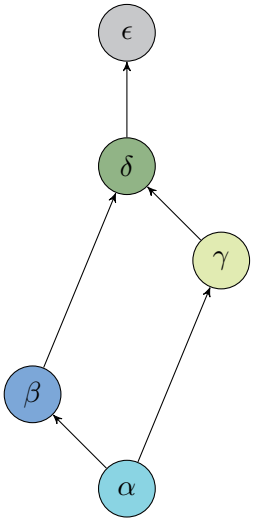
Note that using the Needleman-Wunsch algorithm [104], finding the optimal alignment between two sequences with overlapping subsequences requires aligning all possible pairs of sequences (without overlapping subsequences) corresponding to all choices of alternative paths in their respective DAGs. This results in an exponential time complexity as the number of overlaps increases. In contrast, the above algorithm finds the optimal alignment between two DAGs efficiently, with a complexity that is quadratic with respect to the number of nodes.

We find the partial order alignments between all pairs of CNEs in the main dataset. For each pair of CNEs, we obtain two alignments, one for each of the two possible relative orientations of the sequences. The alignment with the highest score is chosen as the optimal alignment.

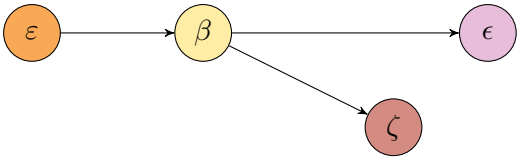
4.4 Measuring the Frequency of Aligned TFBSs

We search the alignments for words composed of up to four aligned symbols, i.e., co-occurring TFBSs. We then compute the relative frequency of each word with respect to a background model as follows: let C be the set of words that appear in the alignments of sequences from the main dataset, and let B be the set of words that appear in the alignments of sequences from the background model, obtained using the same type of alignment (global, semi-global or local. For a definition of each type of alignment, see Section 2.2). Let $n_C(w)$ be the number of occurrences of word w in C, and let $n_B(w)$ be the number of occurrences of w in B. We denote the length of w by $|w|$. Not all words that are present in C are present in B, and vice versa, i.e., $n_B(w) = 0$ or $n_C(w) = 0$ for some w. To account for unseen words, we apply Laplace smoothing by adding the constant λ to all counts of w. The probability of occurrence of w in the alignments of main sequences is computed as follows:

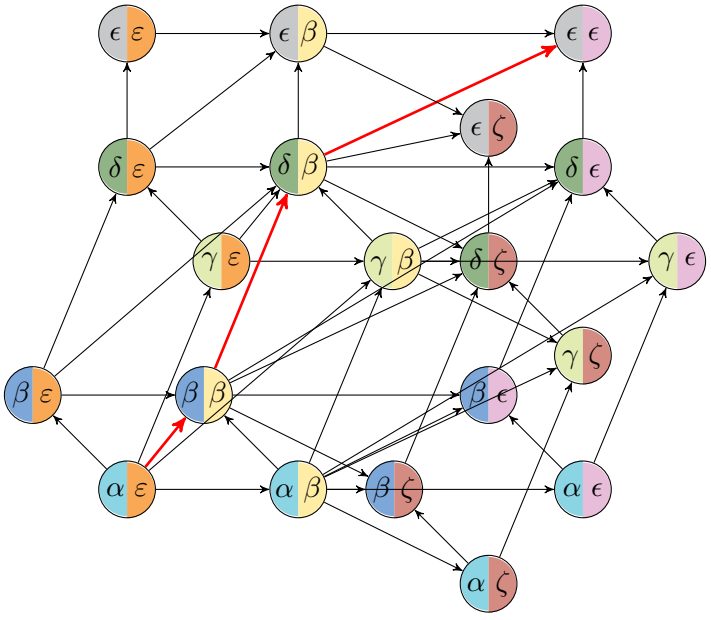
$$P_C(w) = \frac{n_C(w) + \lambda}{\sum\limits_{\substack{w' \in C \cup B \\ |w'| = |w|}} (n_C(w') + \lambda)} \tag{4.2}$$



(a)



(b)



(c)

$$\alpha\,\beta\,\delta\,\epsilon$$
$$\varepsilon\,\beta{-}\epsilon$$

(d)

Figure 4.2: (c) Strong product graph of DAGs shown in (a) and (b). The path corresponding to the optimal global alignment shown in (d) is coloured in red. (d) An optimal global alignment between the two sequences represented by DAGs shown in (a) and (b)

TF Family Names		
CDX	MEIS	RFX
ETS	NKX	RUNX
FOX	NRF	SIX
GATA	PAX	SOX
HMX	PBX	TCF
HOX	PITX	TFAP
IRX	POU	ZIC
MAF	RA	

Table 4.1: Names of TF families from which the representative TFs were chosen

Note that in computing $P_C(w)$, only words of the same length as w are considered. The probability of occurrence of w in the alignments of background sequences, $P_B(w)$, is computed in the same way. The relative frequency of w is given by

$$R_{CB}(w) = \frac{P_C(w)}{P_B(w)} \tag{4.3}$$

4.5 Evaluation

We evaluated our approach to identifying over-represented motifs on a set of CNEs from the CONDOR database [175]. The main dataset consists of 426 sequences in four orthologous sets of CNEs from human, mouse, rat/dog and pufferfish. We generated the background model as follows: each background sequence was generated by randomly shuffling a CNE from the main dataset using the MEME Suit toolkit [184]. We repeated this process ten times to generate ten different sets of shuffled sequences, i.e., 4260 sequences.

We chose a set of 31 TFBSs for representative family members of TFs known to be involved in developmental patterning, and retrieved their binding preferences from the UniPROBE [185] and JASPAR [186] databases (Table 4.1). We scanned the sequences in both the main dataset and the background model for the occurrences of these TFBSs (specified by their consensus sequences using IUPAC codes) using FIMO [174] (p-value ≤ 0.001).

We obtained both the global and semi-global partial order alignments of CNEs in the main dataset, which we refer to as the Global and Semi-global sets, respectively. The alignment parameters were set as defined in the following section. We computed the relative frequencies of words of lengths two, three and four in each of the Global and Semi-global sets with regards to its respective background model, i.e., words in the global and semi-global alignments of se-

quences from the background model, respectively. The number of occurrences of each word from the background model was averaged over the ten shuffled sets. The Laplace smoothing constant (λ) was set to 1.

Scoring the TFBS Alignments

We defined the scores for aligned TFBSs in a way that takes the number of each TFBS in the main dataset into consideration, i.e., the more rare the TFBSs, the higher their matching score. Let $n(t)$ be the number of times that TFBS t has been identified in CNEs from the main dataset and N be the total number of TFBSs in the same dataset. The matching score $s(t, r)$ is defined as

$$s(t,r) = \begin{cases} \log \dfrac{1}{P^2(t)} & \text{if } r = t \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

where $P(t) = n(t)/N$. The linear gap penalty was set to -1 (comparable to the obtained matching scores).

4.5.1 Results

The Global and Semi-global sets contain 229 and 270 words, respectively; 207 words appear in both sets. The number of words of lengths three and four is low and collectively, they constitute less than 14% of the words in the two sets. In over 99% of cases, words from the Global set that are over-represented are also over-represented in the Semi-global set, and vice versa. Hence, the results are stable irrespective of the type of alignment. The top five words of length two with the highest relative frequency are listed in Table 4.2.

The words '$\delta\epsilon$' and '$\gamma\epsilon$' are of note since ZIC has been shown to regulate retinoic acid (RA) signalling during early embryogenesis, which affects the expression levels of HOX and MEIS during hindbrain patterning [187]. Moreover, both MEIS and ZIC are involved in the patterning of the brain and spinal cord, and as such, are likely to be co-expressed spatially and temporally in the embryo [188, 189]. The word '$\beta\gamma$' represents the known interaction of MEIS with HOX [190].

The highest-ranked word '$\gamma\delta$' represents the previously reported interaction of MEIS with the PBX-HOX complex [191, 192]. The regulatory activity of this word has been functionally validated in elements from our dataset [193]. In [193], this syntax was identified in a set of conserved vertebrate hindbrain enhancers.

Word	TFs	Relative Frequency
$\gamma\delta$	MEIS, PBX-HOX	20.3
$\delta\epsilon$	PBX-HOX, ZIC	5.4
$\gamma\epsilon$	MEIS, ZIC	2.3
$\alpha\gamma$	CDX2, MEIS	1.8
$\beta\gamma$	HOXD10-HOXD13, MEIS	1.5

Table 4.2: Top five over-represented words of length two from the Global set and their relative frequency (symbols used to represent TFBSs are arbitrary and are only included to allow a rapid assessment of the similarities between the words)

It was shown that MEIS motifs are frequently proximal (within 100bp) to PBX-HOX motifs, and that both are required for hindbrain enhancer function. This syntax was then used to predict hindbrain enhancers in this dataset with an accuracy of 89%. Furthermore, this syntax was refined and used to predict over 3,000 hindbrain enhancers across the human genome.

The results demonstrate the predictive power of our approach, which can be used as a fast alternative to wet-lab methods for the analysis of a large set of CNEs in order to prioritise the elements for functional assays.

4.6 Availability

The scripts for aligning CNEs and retrieving words from the alignments are available at https://bitbucket.org/mabdollahyan/cnealign. The following are provided in the supplementary materials: the main dataset, the background model, the TFBS sequences and their matching scores, the FIMO output files, the script for computing the frequencies of words, and the complete list of words along with their frequencies. The work presented in this chapter has appeared in

- Abdollahyan M, Smeraldi F, Noyvert B and Elgar G. Transcription factor binding site-based alignment of conserved non-coding elements. Poster presented at the 15th European Conference on Computational Biology (ECCB); 3–7 September 2016. DOI: 10.7490/f1000research.1113029.1.
- Abdollahyan M, Smeraldi F and Elgar G. Identifying Potential Regulatory Elements by Transcription Factor Binding Site Alignment using Partial Order Graphs. 1st Conference on Mathematical Foundations in Bioinformatics (MatBio) 2016 [Special Issue]. International Journal of Foundations of Computer Science. 2018;29(8):1345–1354.

Chapter 5

The Partial Order Kernel (POKer) for Comparing Strings with Alternative Substrings

In Chapter 3, we showed that existing metrics are unable to capture the regulatory sequence signatures in CNEs. And in Chapter 4, we presented a method for identifying the regulatory signatures in CNEs by aligning the sequences of TFBSs identified in the elements, and showed that it successfully captures the regulatory grammar of CNEs in terms of co-regulatory TFs. However, the score of such an alignment is not necessarily metric and therefore impractical for use in numerous learning algorithms. In this chapter, we introduce a kernel for comparison of strings that contain alternative substrings, i.e., substrings that can be substituted with each other. The sequence of TFBSs identified in a CNE is an example of this type of strings, where overlapping TFBSs constitute the alternative substrings. Our kernel, named the partial order kernel (POKer), provides (via the induced norm) a metric interpretation of the score of the alignment method presented in Chapter 4, and can be used in combination with other algorithms beyond kernel methods.

The POKer is a convolution kernel defined over the product of two DAGs, each representing a partial order over the characters of a string with alternative substrings, and is computed using dynamic programming. We evaluate the POKer on artificial data generated by detecting motifs in simulated DNA sequences. We use the POKer in conjunction with SVMs to classify these strings. In order to have a benchmark against which to compare the POKer's perfor-

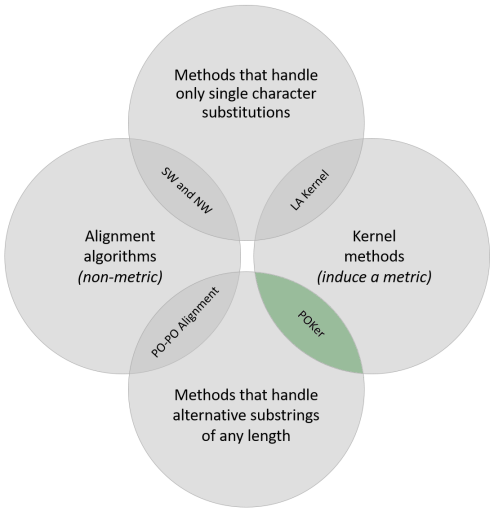


Figure 5.1: Relation between alignment algorithms and kernel methods for comparing strings with alternative substrings

mance, we extend the state-of-the-art spectrum kernel [132] to handle strings with alternative substrings.

5.1 Related Work

Classic sequence alignment algorithms based on dynamic programming, such as the Needleman-Wunsch (NW) [104] and the Smith-Waterman (SW) [105] alignment algorithms (see Section 2.2.1), can handle single character substitutions. Variations of these algorithms can, in addition, deal with alternative substrings. The alignment scores returned by these methods, however, are not necessarily metric since they do not always satisfy all the properties of a metric, i.e., identity of indiscernibles, non-negativity, symmetry and triangle inequality. For example, the score of a partial order-partial order (PO-PO) alignment [182] (see Section 4.3) can violate the non-negativity and triangle inequality conditions. Specifically, given two strings X and Y, the score of their PO-PO alignment, depending on the choice of parameters, may lack the non-negativity property. Moreover, this score corresponds to a minimum of the weighted Levenshtein distances between all pairs of strings in $Alt_X \times Alt_Y$, where Alt_X is the set of strings with no alternative substrings extracted from X, and thus can violate the triangle inequality condition. This makes the PO-PO alignment scores impractical for use with methods that require a metric dissimilarity. String kernels (see Section 2.4.1), on the other hand, produce a measure of similarity between strings that is metric (via the induced norm). They, however, cannot handle strings with alternative substrings. Specifically, the local alignment (LA) kernel [140],

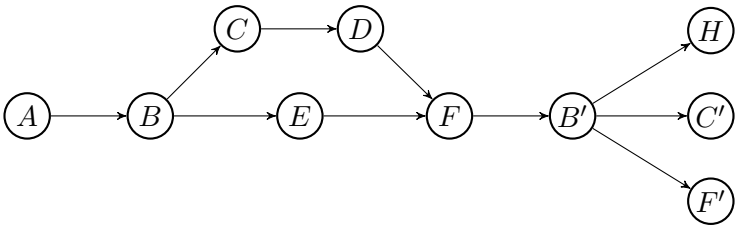


Figure 5.2: DAG representation of $AB[CD|E]FB'[H|C'|F']$

which computes an exponentially weighted sum of the scores of all the possible local alignments between two strings as given by the SW algorithm, cannot handle alternative substrings. This motivated us to develop a novel kernel for the comparison of this type of strings (Figure 5.1).

The POKer offers a metric interpretation of the sum of the scores of PO-PO alignments between two strings. The LA kernel is a special case of the POKer, i.e., when the two strings contain no alternative substrings, the value of the POKer is equal to that of the LA kernel.

5.2 Representing Strings with Alternative Substrings

Before introducing the POKer, we formally define the graph representation of strings with alternative substrings, which was described specifically for CNEs in Section 4.2. Let x be a string with alternative substrings. For example, consider the string $x = AB[CD|E]FB'[H|C'|F']$ (Figure 5.2), where the brackets contain the alternative substrings (e.g., $[CD|E]$ indicates that substring CD appears in x as an alternative to character E). Here, we use the prime symbol for convenience to distinguish between multiple occurrences of the same character. We denote the multiset of characters in x by $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ and the set of standard strings, generated by iterating over all possible choices of alternative substrings in x, by Alt_x (e.g., $ABEFB'C' \in Alt_x$). Note that in $\mathcal{X}$, repeated characters count as distinct elements. For any two characters $x_i, x_j \in \mathcal{X}$, we write $x_i \prec x_j$ if x_i precedes x_j in some string in Alt_x. The relation $\prec$ is a partial order on $\mathcal{X}$ since, for instance, $A \prec E$ and $C \prec D$ but neither $H \preceq C'$ nor $C' \preceq H$.

We represent $(\mathcal{X}, \prec)$ as a directed acyclic graph (DAG) G_x, where $V(G_x) = \mathcal{X}$ is the set of nodes and an edge exists between x_i and x_j if and only if they are consecutive in $\mathcal{X}$. Each path in G_x, from a source to a sink node, corresponds to a string in Alt_x. There may exist multiple source/sink nodes since x can start/end with alternative substrings.

5.3 Partial Order Kernel

Let x and y be two strings with alternative substrings, represented by DAGs G_x and G_y, respectively. The partial order kernel (POKer) $K(x, y)$ is defined as a convolution of (several instances of) two kernels K_a and K_g with respect to relations R_x^m and R_y^m over the nodes of paths in G_x and G_y, respectively. We define R_x^m as the set of lists $(X_1, X_2, \ldots, X_{2m-1})$, $X_i \subseteq V(G_x)$ with the conditions that

1. there exists a path π_x in G_x such that $X_i \subseteq V(\pi_x) \, \forall i$ and $\cup_i X_i = V(\pi_x)$, and
2. the X_i are consecutive, i.e., for all $v \in X_i$ and $u \in X_{i+1}$ we have $v \prec u$.

Note that the X_i are disjoint, and some of them may be empty. R_y^m is defined in a similar way. Intuitively, R_x^m and R_y^m represent a local alignment of m characters along π_x, possibly separated by gaps, with m characters along a path π_y in G_y. This intuition guides the definition of the two kernels to be convolved, namely the substitution kernel K_a and the gap kernel K_g. Let $X \subseteq V(\pi_x)$ and $Y \subseteq V(\pi_y)$. Similarity of aligned characters is measured by the substitution kernel

$$K_a{}^{(\beta)}(X, Y) = \begin{cases} exp(\beta s(X, Y)) & \text{if } |X| = 1 \text{ and } |Y| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where $\beta \geq 0$ is a parameter and $s(X, Y)$ is the substitution score for the labels of nodes in X and Y specified by, for instance, a scoring matrix. Valid values for β are those for which the kernel remains positive semi-definite. Penalty for gaps is quantified by the gap kernel

$$K_g{}^{(\beta)}(X, Y) = \exp\left(\beta g \left(|X| + |Y|\right)\right) \tag{5.2}$$

where g is a linear gap penalty. We convolve the above kernels to construct the kernel K_m.

$$K_m(x, y) = (K_a * K_g)^{m-1} * K_a = \sum_{X \in R_x^m, Y \in R_y^m} \left[\prod_{k=1}^{m-1} K_a{}^{(\beta)}(X_{2k-1}, Y_{2k-1}) K_g{}^{(\beta)}(X_{2k}, Y_{2k}) \right] K_a{}^{(\beta)}(X_{2m-1}, Y_{2m-1}) \tag{5.3}$$

It is clear from the definition of K_a that the terms in the above sum are zero

unless all the X_i and Y_i with odd indices are singletons, and that

$$K_m(x, y) = \sum_{X \in R_x^m, Y \in R_y^m} \exp(\beta S(X, Y)) \tag{5.4}$$

where $S(X, Y)$ is the score of the local alignment of m characters along π_x with m characters along π_y specified by the two lists X and Y. Note that K_m is zero when m is larger than the length of the longest path in G_x or G_y. With the above definitions, we now define the POKer as

$$K(x, y) = \sum_{m \geq 0} K_m(x, y) = \sum_{m \geq 0} \sum_{X \in R_x^m, Y \in R_y^m} \exp(\beta S(X, Y)) \tag{5.5}$$

The POKer is equal to an exponentially weighted sum of the scores of all the local alignments between any number of characters in x and the same number of characters in y, selected from any paths in G_x and G_y, that is, from any choice of alternative substrings. The importance of the contributions of non-optimal alignments to the kernel value is controlled by parameter β; for $\beta \to \infty$, only the best alignments are taken into account. In the next section, we show how this finite sum is computed efficiently using dynamic programming.

Note that simply computing the local alignment score for each pair of strings in $Alt_x \times Alt_y$ using the LA kernel, and then summing these scores for all possible pairs does not yield the same measure of similarity as the one produced by the POKer. This approach considers the contributions of those substrings that are common to all the strings (e.g., AB in $ABCDFB'H$ and $ABEFB'H$ shown in Figure 5.2) more than once. Moreover, it results in a time complexity that is exponential in the number of alternative substrings. In contrast, the POKer takes the contributions of such substrings into account only once and, as we show in the next section, its value is computed with quadratic complexity.

5.3.1 Computation

The POKer is computed efficiently using dynamic programming over the strong product graph $G_{xy} = G_x \boxtimes G_y$, with a time complexity that is linear in the number of nodes of G_{xy} (here, each node is denoted simply as (i, j)).

Theorem 1. We assume, without loss of generality, that both x and y begin with a start character $x_0 = y_0 = \phi$, that is, G_x and G_y each have a single source node labelled ϕ, and $s(\phi, \phi) = s(x_i, \phi) = s(\phi, y_j) = 0$ for all i and j. We then

have

$$K(x,y) = 1 + \sum_{(i,j)\in V(G_{xy})} M(i,j) \tag{5.6}$$

where $M(i,j)$ is computed recursively as follows:

$$\begin{cases} M(i,\phi) = M(\phi,j) = 0 \\ N(i,\phi) = N(\phi,j) = 0 \end{cases} \tag{5.7}$$

and

$$\begin{cases} M(i,j) = & \exp(\beta s(i,j)) \left[1 + \sum\limits_{\substack{m,n \\ mi\in E(G_x), nj\in E(G_y)}} N(m,n) \right] \\ N(i,j) = & \exp(\beta g) \sum\limits_{\substack{m \\ mi\in E(G_x)}} N(m,j) \\ & + \exp(\beta g) \sum\limits_{\substack{n \\ nj\in E(G_y)}} N(i,n) \\ & - \exp(2\beta g) \sum\limits_{\substack{m,n \\ mi\in E(G_x), nj\in E(G_y)}} N(m,n) \\ & + M(i,j). \end{cases} \tag{5.8}$$

where $\beta \geq 0$ is a parameter and g is the gap penalty.

The proof of this theorem is provided in Appendix C.

Each local alignment corresponds to a path in G_{xy}. The POKer (Equation 5.6) is a sum over the exponentiated scores of all the local alignments, including the empty alignment, ending at each node (i,j) in G_{xy}. The contributions of all the local alignments ending at (i,j), including those with the labels of i and j being the only aligned characters are accounted for by $M(i,j)$. Penalties for inserting gaps in x or y at the end of a partial alignment are included in $N(i,j)$. A gap in x followed by a gap in y, and a gap in y followed by a gap in x, are equivalent in terms of aligned characters; this is accounted for by the negative term in Equation 5.8.

5.4 Experiments

We tested the POKer in conjunction with SVMs in two multi-class classification scenarios using artificial data and compared its performance to that of a generalised spectrum kernel introduced in the following section. The aim of the first set of experiments is to assess the classification accuracy of the POKer, while the aim of the second set of experiments is to assess the ability of the POKer to

capture both the global and local structures of the data.

5.4.1 Generalised Spectrum Kernel

To the best of our knowledge, no other kernels have been specifically designed for strings with alternative substrings of variable length. However, the popular spectrum kernel [132] can be extended in order to deal with such strings and provide a baseline for comparison. We define this generalised kernel as follows: let x be a string with alternative substrings over an alphabet $\mathcal{A}$ and G_x be its DAG representation. We denote the set of all paths of length k in G_x, i.e., k-mers in x, by $\Pi_{(k,x)}$. We define the feature map indexed by the set of all k-length substrings α from $\mathcal{A}^k$ as

$$\Phi_k(x) = \left(\left|\{\pi \in \Pi_{(k,x)} | (V(\pi)) = \alpha\}\right|\right)_{\alpha \in \mathcal{A}^k} \tag{5.9}$$

where $(V(\pi))$ is the sequence of labels of nodes in π. The generalised spectrum kernel is then defined as

$$K_k(x, y) = \langle \Phi_k(x), \Phi_k(y) \rangle \tag{5.10}$$

Note that this is not equivalent to accumulating the occurrences of each k-mer over all the strings in Alt_x, since then substrings common to multiple strings in Alt_x would be counted more than once.

5.4.2 Data Simulation and Parameters

The dataset for the first set of experiments was created as follows: we used rMotifGen [194], a tool for generating random DNA sequences containing short motifs, to create 40 classes of strings by following these steps: first, we produced a dictionary consisting of 50 randomly generated motifs, where each motif represents a TFBS. We assigned a unique character to each motif. To generate the strings in each class, we randomly chose 10 motifs from the dictionary. We then generated 200 random DNA sequences containing these motifs. We set the parameters so that each motif (out of 10) appears in 70% of the sequences, and used the default values for the rest of the parameters. Finally, we scanned each sequence for the occurrences of all 50 motifs (as motifs others than those explicitly inserted in the sequence can appear in it by chance) in order to obtain a string of characters corresponding to the detected motifs. Since two or more motifs can overlap, but only one TF can bind to overlapping TFBSs at a given time, such a string is a string with alternative substrings. We repeated

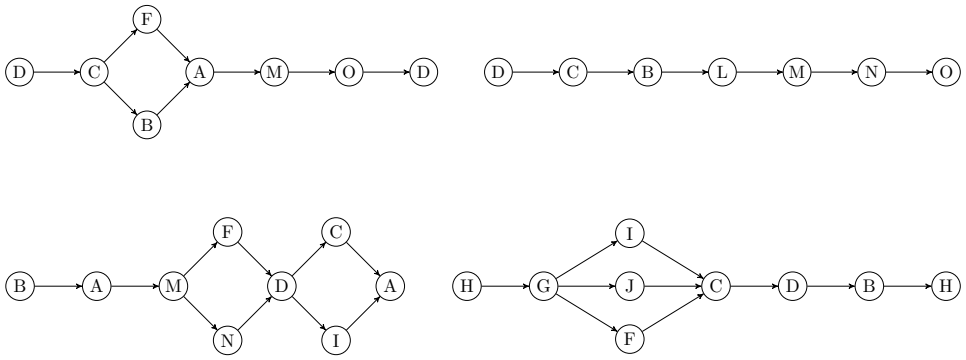


Figure 5.3: Sample strings from the dataset for the second set of experiments. Strings in the top row are both from class seven. The string in the lower left is from class eight, generated from a prototype sequence seeded with a permutation of the motifs found in class seven. The string in the lower right is from class four (unrelated).

this procedure 40 times to generate 40 classes of 200 strings, i.e., a total of 8000 strings.

For each of the two kernels, we built 40 SVM classifiers, one for each class, as follows: we trained each classifier on 160 sequences from one class (using $39 \times 160 = 6240$ training sequences from other classes as negative examples) and tested it on the remaining 40 sequences from that class (using the remaining 1560 test sequences from other classes as negative examples) in a one-versus-all strategy. We performed a 10-fold cross-validation on the training set for selecting the parameters. The match score, mismatch score and gap penalty were set to 4, 0 and -2, respectively. We ran the POKer with several β values, ranging from 0 to ∞, and chose the one that yielded the best performance in the cross-validation (β=0.1). Similarly, we ran the generalised spectrum kernel with several k values $k \in \{2, 3, 4, 5\}$ and chose the one for which the kernel performed best in the cross-validation (k=3).

The dataset for the second set of experiments was created as follows: we considered an alphabet $\mathcal{A}$ of 26 characters, represented by uppercase letters (A-Z). To each character, we associated a randomly generated motif of 6 letters in $\{a, t, c, g\}$ (the four nucleobases in DNA). Starting from a prototype random nucleotide sequence, we generated a further 199 nucleotide sequences by introducing random mutations (substitutions with 0.1 probability) in the prototype sequence. We then identified all characters from $\mathcal{A}$ in each sequence by matching the corresponding motifs. This yielded a class of strings with alternative substrings. We repeated this procedure 5 times to generate 5 classes of 200 strings. Next, we took a single string from each of these 5 classes, permuted its characters and used it to seed another prototype nucleotide sequence by expanding its motifs into nucleobases. We repeated the above procedure of introducing random mutations in these new prototype sequences. Motif detection then yielded

	Mean AUROC	
Number of classes	***POKer***	***Generalised spectrum kernel***
5	0.984	0.934
10	0.981	0.901
20	0.978	0.864
40	0.964	0.82

Table 5.1: Mean AUROC values obtained by the POKer and the generalised spectrum kernel for different number of classes in the first set of experiments

a further 5 classes of 200 strings with alternative substrings, where strings in each class consist of noisy permutations of motifs present in the strings in their respective seed class.

For each kernel, we built 10 SVM classifiers, one for each class, as follows: we trained each classifier on 160 sequences from one class (using $9 \times 160 = 1440$ training sequences from other classes as negative examples) and tested it on the remaining 40 sequences from that class (using the remaining 360 test sequences from other classes as negative examples) in a one-versus-all strategy. We performed a 10-fold cross-validation on the training set for selecting the parameters. The match score, mismatch score and gap penalty were set to 4, 0 and -2, respectively. Similar to the first set of experiments, we ran the POKer with a range of β values and chose the one that yielded the best performance in the cross-validation (β=0.01). And we ran the generalised spectrum kernel with different k values and chose the one for which the kernel performed best in the cross-validation (k=3).

5.4.3 Results

We use the value of the area under the receiver operating characteristic (ROC) curve (AUROC), averaged over the classes, to compare the performances of the two kernels.

In the first set of experiments, as shown in Table 5.1, the POKer outperforms the generalised spectrum kernel, with its performance decreasing only marginally as the number of classes increases. The POKer appears to be robust to the choice of β. For instance, in the case of 40 classes, β=0.1 yields the mean AUROC value of 0.96, while β=0.01 and β=1 (that is, varying β by a factor of 10 in either direction) yield 0.96 and 0.91, respectively.

Figure 5.4 displays the individual ROC curves for all classes, produced by each kernel in the second set of experiments. The POKer achieves a higher

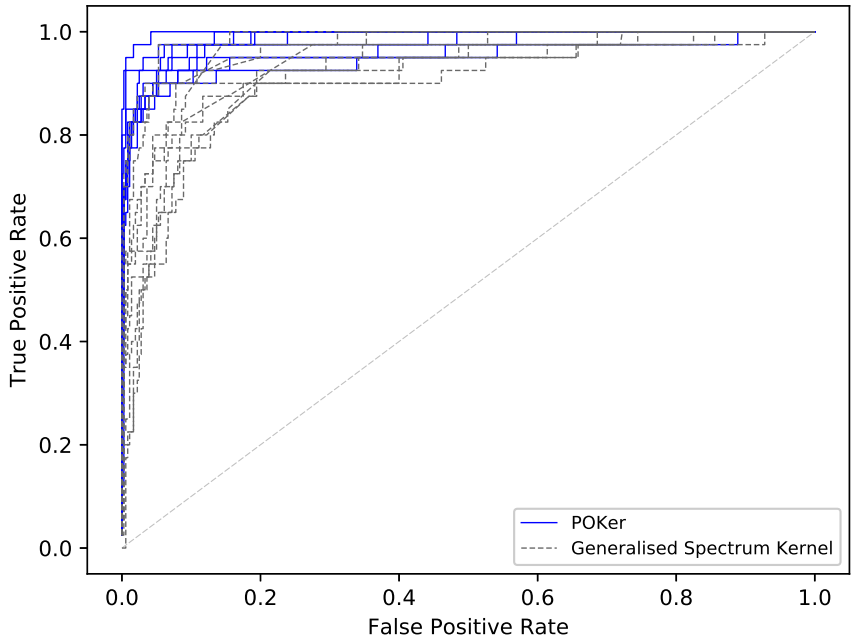


Figure 5.4: ROC curves obtained by the POKer and the generalised spectrum kernel in the second set of experiments

AUROC in all cases, with a mean AUROC value of 0.98 against 0.94 achieved by the generalised spectrum kernel, and an average interpolated equal error rate (EER) of 0.05 against 0.16 for the generalised spectrum kernel. The POKer is fairly robust to the choice of β, its performance being consistent in the range of β=0.01 (yielding the mean AUROC value of 0.98) to β=0.6 (yielding a mean AUROC value of 0.95), and decreasing only slightly to a mean AUROC value of 0.90 when β=1.

By assigning each string to the class with the highest score, the confusion matrices for the POKer and the generalised spectrum kernel are obtained. As shown in Figure 5.5, the POKer outperforms the generalised spectrum kernel in all cases. Notably, the confusion matrix for the latter is roughly block diagonal, as the generalised spectrum kernel does not capture the global order of motifs (being insensitive to where each k-mer occurs within the strings). Hence, it discriminates poorly between the classes generated from series of motifs that are permutations of each other (shown as contiguous in the matrix). These mainly contain rearrangements of the same characters (e.g., $DC[F|B]AMOD$ and $BAM[F|N]D[C|I]A$ in Figure 5.3). On the contrary, the POKer handles these cases almost as well as the easier case of strings that differ substantially in the motifs which they contain (e.g., $DC[F|B]AMOD$ and $HG[I|J|F]CDBH$ in Figure 5.3).

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
c1	33	4	1	1	0	0	0	1	0	0
c2	1	34	0	2	2	0	0	0	1	0
c3	1	1	34	3	0	0	0	1	0	0
c4	0	0	3	35	0	0	0	0	2	0
c5	0	0	0	0	33	2	3	1	1	0
c6	0	0	0	0	1	39	0	0	0	0
c7	0	1	0	1	0	0	38	0	0	0
c8	0	0	0	0	1	1	0	36	2	0
c9	0	0	1	3	1	0	0	2	29	4
c10	0	1	1	1	0	1	0	1	1	34

(a)

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
c1	30	4	1	0	2	1	1	1	0	0
c2	5	28	3	3	0	0	1	0	0	0
c3	2	4	21	12	0	1	0	0	0	0
c4	2	1	7	26	0	0	1	0	3	0
c5	0	2	2	0	23	13	0	0	0	0
c6	0	2	4	0	11	23	0	0	0	0
c7	0	0	0	0	0	1	32	7	0	0
c8	1	0	3	0	0	3	7	26	0	0
c9	0	0	1	2	1	0	0	0	27	9
c10	2	2	3	1	0	0	1	1	2	28

(b)

Figure 5.5: Confusion matrices for (a) the POKer and (b) the generalised spectrum kernel

Overall, the results demonstrate the effectiveness of the POKer in discriminating between classes of strings with alternative substrings. With richer mathematical properties than non-metric alignment scores and efficient computation, the POKer can be a powerful tool in the analysis of this type of strings.

5.5 Availability

The scripts for the POKer are available at https://bitbucket.org/mabdollahyan/poker. The following are provided in the supplementary materials: the two artificial datasets, including the sequences and motifs, the script for generating the dataset for the second set of experiments, the script for detecting motifs, the script for the generalised spectrum kernel, the Gram matrices produced by each kernel, and the script for running the SVM classifiers and obtaining the ROC curves and the mean AUROC values. This work utilised the Apocrita HPC facility (see https://docs.hpc.qmul.ac.uk), supported by QMUL Research-IT [195]. Part of the work presented in this chapter has appeared in

- Abdollahyan M, Smeraldi F. POKer: a Partial Order Kernel for Comparing Strings with Alternative Substrings. In: Proceedings of the 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN); 26–28 April 2017. pp. 263–268.

Chapter 6

Evaluating the POKer: a Computer Vision Application

In Chapter 5, we presented the partial order kernel (POKer) for comparison of strings that contain alternative substrings, and demonstrated its classification effectiveness using artificial data. In this chapter, we evaluate the POKer in a real-world setting. Specifically, we employ the POKer in an approach that addresses a problem in computer vision, namely visual localisation in the presence of changes in the appearance of the environment.

A visual localisation system aims to answer the question of whether an image is of a place it has seen before, and, if so, which one? This is an important problem in robotics and autonomous systems. For a comprehensive survey of visual localisation, see [196]. Real-world scenarios pose many challenges for autonomous navigation systems. One such challenge is the presence of mismatches between images of the same place which occur due to changes in the appearance of the environment. Appearance changes are caused by a number of factors, including illumination variations, different weather conditions and seasonal changes.

We propose a sequence-based visual localisation approach that consists of two steps: in the first step, we build the graph representations of database image sequences, obtained during the exploration phase, using the partial order alignment (POA) algorithm [183]. In Chapter 4, we employed such a representation to model the sequence of (possibly overlapping) TFBSs detected in a CNE. Here, we consider the same representation to model alternative sequences

of images, i.e., sequences of images of the same place that differ in appearance. Using this representation not only allows us to model the temporal sequential nature of images, but also efficiently models the alternative image sequences in the form of alternative paths in a partial order graph. Moreover, it does not require the alternative paths in the graph to be of equal length, and therefore, is robust to differences in the traversal speed. In the second phase, we compare these graphs to query image sequences, obtained during the localisation phase and represented as DAGs without alternative paths, using the POKer. We test our approach on a dataset which consists of image sequences of a train journey collected across four different seasons. The sequences from three seasons constitute the training dataset, while the sequence from the remaining season is used for testing, in a cross-validation fashion. We compare the performance of the POKer to those of two state-of-the-art localisation methods.

6.1 Related Work

Various approaches have been proposed to address the problem of appearance changes in visual localisation. In [197], a probability distribution is learnt in order to model the illumination variations in images. In [198], to reduce illumination variations, images are transformed into an illumination-invariant colour space. A number of approaches exploit image descriptors such as SIFT and SURF to handle appearance changes (e.g., see the approach proposed in [199]). More recently, the use of convolutional neural networks (ConvNets) to extract descriptors that are robust to appearance changes has gained a lot of attention. In [200], a neural network is trained to learn illumination-invariant descriptors that map the image patches into a low-dimensional space where non-matching images are easily separable. Incorporating features learnt using ConvNets has been shown to improve the performance of place recognition systems, as these features are more robust to appearance changes [201, 202, 203]. Here, we employ the recently released ConvNet VGG-Places365 [204] to extract the descriptors. This ConvNet was trained on a dataset of images from diverse types of environments. ConvNets specifically trained for place recognition have been shown to outperform networks trained using generic data [205, 206].

A number of approaches, relying on the fact that some appearance changes such as seasonal changes are cyclic and therefore predictable, learn a transformation between the images [207, 208]. In [208], a superpixel vocabulary for each season and a dictionary to translate the words from one season to their matches in another are generated. This, however, requires the pairs of training images to be perfectly aligned. In contrast, our approach does not make any assumptions

on the nature of appearance changes or pixel alignment of images.

Another category of approaches leverage the sequential nature of images to handle appearance changes. The state-of-the-art method SeqSLAM [209] considers sequences of images instead of single images. Given an image, it finds the local best match within every short image sequence; localisation is then done by searching the image similarity matrix for sequences of local best matches. SeqSLAM assumes constant speed during the traversals. A modified version of SeqSLAM that is invariant to speed variations was introduced in [210]. In our approach, we represent the multiple sequences of images of the same place collected at different times and possibly at different speeds as a partial order graph. As shown in Section 5.2, in this representation, the sequences can diverge from one another to form alternative paths in the graph. These paths may be of different lengths. This allows us to deal with mismatches between the images that are due to speed variations. In [211], a hidden Markov model (HMM) is used to compute the most likely path through the image similarity matrix. While the method, similar to ours, uses dynamic programming (the Viterbi algorithm) to align the sequences, transitions between states are probabilistic. By contrast, our proposed graph representation specifies exactly which transitions are possible at each point. In [212], a modified version of the Smith-Waterman alignment algorithm [105] was used to find matching subsequences within the image sequences in order to detect intersections between maps. The POA algorithm used in our approach is also an extended version of the Smith-Waterman alignment algorithm, however, it works with partial order graphs instead of standard sequences.

The methods described in [213] and [214] build a directed acyclic data association graph to model the matching between an image sequence and a database. The localisation task then becomes a minimum-cost flow problem, i.e., computing a shortest path in this graph. Our approach differs in that the graphs are constructed from database image sequences only and are later compared to query image sequences using the POKer.

6.2 Visual Localisation Using the POKer

In Section 5.2, we showed how a string with alternative substrings can be represented as a directed acyclic graph (DAG). Here, we convert each set of alternative image sequences from the database to a string with alternative substrings, represented as a DAG. The graph representations of the database and query image sequences are built as follows: we represent an image sequence as a simple di-

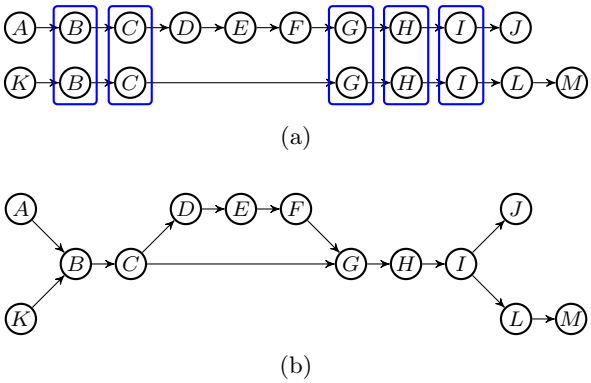


Figure 6.1: (a) DAG representation of two image sequences. Each letter denotes a single image (b) DAG representation of the MSA obtained by aligning the image sequences shown in (a) using the POA algorithm

rected graph, where nodes represent images and there exist edges between nodes whose corresponding images are consecutive in the sequence (Figure 6.1a).

Given a set of alternative database image sequences, we align the sequences using the partial order alignment (POA) algorithm [183]. For the score of an aligned pair of images we use the cosine similarity between their descriptors, and for the gap penalty we use a linear gap model. The output is an MSA in the form of a DAG (Figure 6.1b). This process of building database image sequence graphs is reminiscent of dynamic time warping (DTW) [215], an algorithm for aligning temporal sequences that vary in speed (e.g., sequences of images of the same place taken at different speeds). In fact, DTW is closely related to sequence alignment (for instance, see [216]).

Given a query image sequence, we represent it simply as a DAG without any alternative paths (similar to one of the sequences in Figure 6.1a). We compute the similarity between each pair of database and query graphs using the POKer, and choose the most similar database graph to the query graph as the matching database image sequences to the query image sequence.

6.3 Experiments

6.3.1 Dataset and Parameters

We chose the standard Nordland dataset[1] for evaluating our approach. The dataset consists of video footage of a 728km-long train journey between two cities in Norway, recorded from the perspective of the train driver. The journey was recorded once in every season. We subsampled each video at 0.5fps, which

[1] https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/

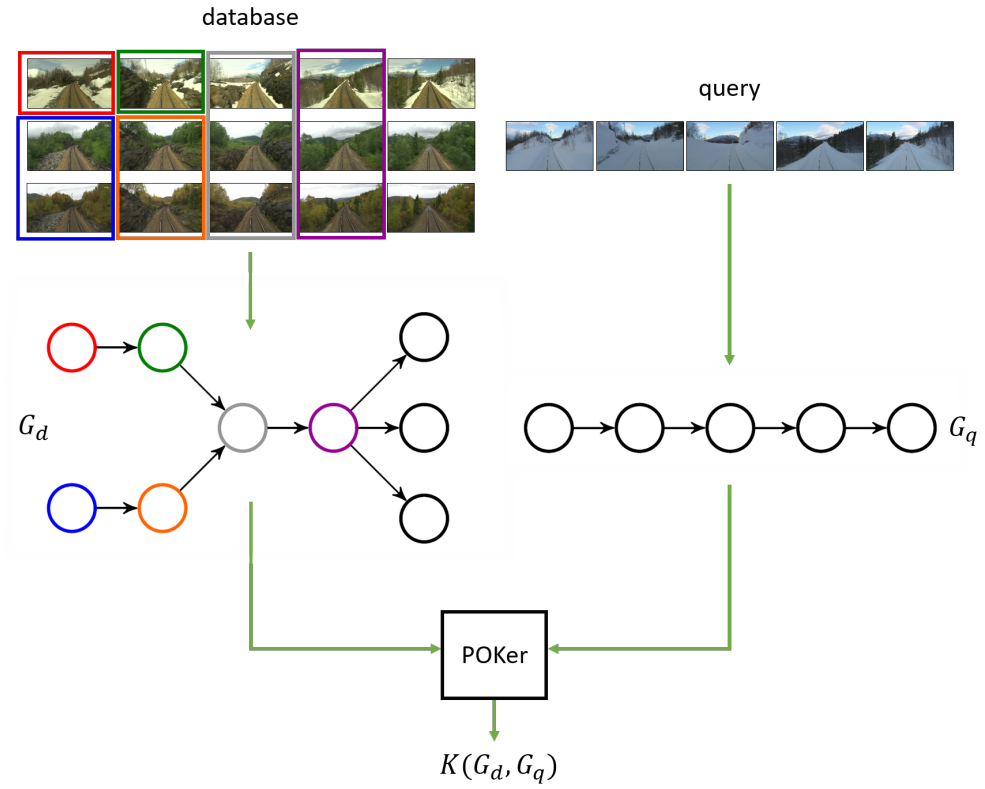


Figure 6.2: Overview of our method applied to the Nordland dataset. On the left: three database image sequences of a place in spring, summer and autumn, respectively. A DAG representation of these alternative sequences is built using the POA algorithm. On the right: a query image sequence of the same place in winter, represented as a DAG. Query and database graphs are compared using the POKer which produces a measure of similarity between the two graphs.

yielded a total of four image sequences. We refer to these image sequences as the Spring, Summer, Autumn and Winter sequences. Note that all sequences are of equal length and that images with the same index are from the same place (this serves as the ground truth). The dataset features severe appearance changes due to different weather conditions and seasonal changes. The train occasionally goes through tunnels and stops at stations. As customary for this dataset [206, 201], we removed all the images taken inside the tunnels and at the stops.

The descriptors were extracted from the fifth layer of the VGG-Places365 ConvNet [204], and we applied locality-sensitive hashing (LSH) [217] to reduce their dimensionality from 100,352 to 4,096.

We performed four sets of experiments, each time using the image sequence belonging to a different season for generating the query image sequences. The data for each set of experiments was generated as follows: during the exploration phase, we consider three of the image sequences in the dataset, i.e., three seasons. We cut each sequence into subsequences of length 15. As a result, for

each location, there exist three alternative image sequences in the database (one per season). We generate triplet image sequences by selecting the three image sequences of the same place in different seasons, according to the ground truth. For each triplet, we align the image sequences in that triplet and build its graph representation, as explained in Section 6.2 (e.g., left column in Figure 6.2).

During the localisation phase, we consider the remaining image sequence in the dataset, i.e., the fourth season. We generate the query image sequences by cutting this sequence into subsequences of length 15. We convert each of these to a DAG without alternative paths, as explained in Section 6.2 (e.g., right column in Figure 6.2). We then compare them to the database triplets using the POKer.

In both the database and query graphs, each node is labelled with the index of its corresponding image. For the alignment parameters, we used the Hamming distance between the descriptors as scores, since after applying LSH, the cosine similarity between the original high-dimensional data is approximated by the Hamming distance between the low-dimensional data. The gap penalty was set to -1. For the POKer, we used β=1.

6.3.2 Baseline Methods

We used two state-of-the-art localisation methods as baselines: the algorithm presented in [213] using network flows, and SeqSLAM utilising ConvNet features. We refer to these methods as NetFlow and CNN+SeqSLAM, respectively. For both baselines, we used the same features as those used for our method, i.e., descriptors extracted by the pre-trained VGG-Places365 ConvNet.

Note that these methods match an image to another, not an image sequence to multiple image sequences (here, a triplet). Therefore, to obtain a measure of similarity between a query image sequence and a triplet of database image sequences, we proceeded as follows: we compared the query image sequence to each database image sequence in the triplet separately. The results are three matrices, where each matrix stores the similarity scores between all pairs of images from the query sequence and from one of the database sequences. We then fused the three matrices by choosing the maximum score for each pair of images as their final similarity score (we considered both average and maximum of scores and chose the maximum as it yielded a better performance). The similarity between the query sequence and the database triplet is the average of entries in this matrix. Parameters for both baselines were set to those that performed best for the most challenging image sequences in the dataset, i.e.,

Precision (%)	**Recall (%)**		
	Our approach	***NetFlow***	***CNN+SeqSLAM***
100	90.7	37.0	75.5
95	99.9	99.2	92.6
90	99.9	99.6	95.1

Table 6.1: Comparison of the average recall values obtained by our method and the baselines

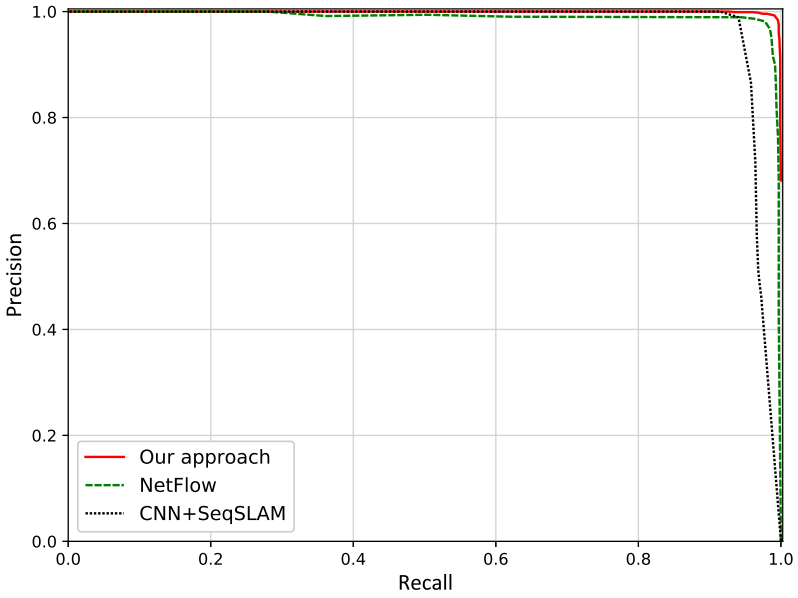
Summer vs Winter.

6.3.3 Results

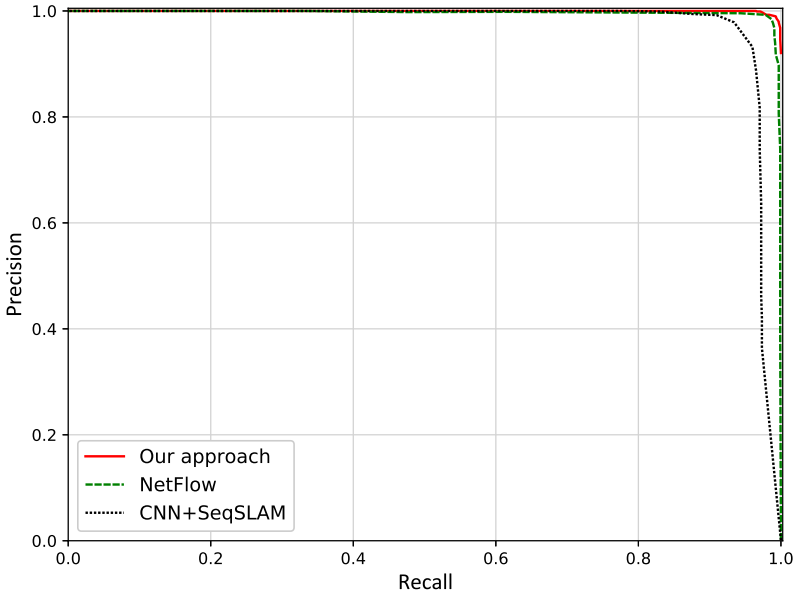
We compare the performances of the POKer and the baseline methods based on the highest precision and recall values achieved by each method.

The precision-recall curves for the four sets of experiments are shown in Figure 6.3. In each case, our method either matches or outperforms the baselines in all parts of the curve. Table 6.1 reports the recall values obtained by each method for three precision values of practical interest, averaged over the four experiments. Our approach achieves a high level of recall (>90%) with 100% precision, and by sacrificing 5% precision, almost 100% recall is achieved. In comparison, both NetFlow and CNN+SeqSLAM achieve lower recall values, with their performances being significantly low at 100% precision.

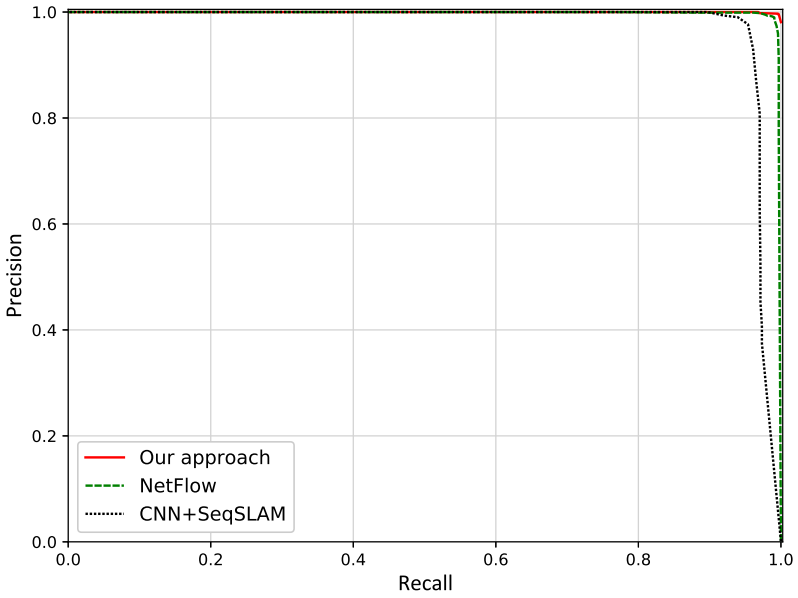
The results show that the POKer accurately computes the similarities between image sequence graphs, in an approach that is robust to appearance changes and outperforms two state-of-the-art methods, and demonstrate the classification effectiveness of the POKer in a real-world setting.



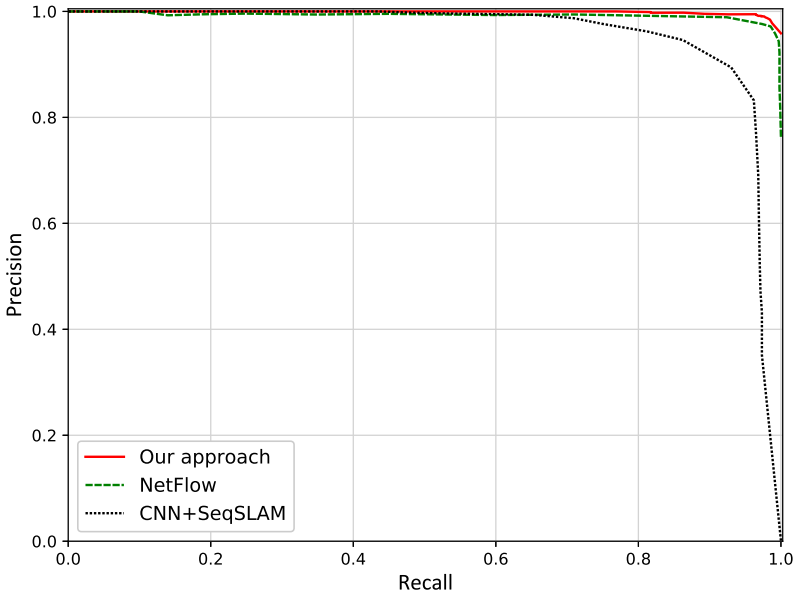
(a) Spring



(b) Summer



(c) Autumn



(d) Winter

Figure 6.3: Precision-recall curves obtained for the four sets of experiments using the (a) Spring, (b) Summer, (c) Autumn and (d) Winter sequences as the query image sequence, respectively

6.4 Availability

The pre-processed Nordland sequences and the similarity matrices produced by all the methods are included in the supplementary materials. For the POA algorithm, we used the implementation at https://github.com/ljdursi/poapy. The work presented in this chapter has appeared in

- Abdollahyan M, Cascianelli S, Bellocchio E, Costante G, Ciarfuglia T A, Bianconi F, Smeraldi F and Fravolini M. Visual Localization in the Presence of Appearance Changes Using the Partial Order Kernel. In: Proceedings of the 26th European Signal Processing Conference (EUSIPCO); 3–7 September 2018. pp. 702–706.

Chapter 7

Functional Classification of CNEs Based on Their TFBSs

The sequence of TFBSs identified in a CNE belongs to the class of strings with alternative substrings, as the binding sites may overlap but only one of them can be bound by a TF at any given time. In Chapter 4, we showed how these sequences can be efficiently represented as partial order graphs. And in Chapter 5, we introduced the partial order kernel (POKer) for the comparison of two such graphs. We demonstrated the effectiveness of the POKer in the classification of real-world data in Chapter 6. In this chapter, we present an approach that employs the POKer to classify CNEs into groups of functionally related elements based on their TFBSs composition.

We train an SVM classifier with the POKer as its kernel on a set of CNEs that have been functionally validated. We then use this classifier to predict the regulatory activity of elements in another set of CNEs that have been shown to act as enhancers in different tissues. To evaluate our approach, we compare the results to those obtained by functional assays. Moreover, we apply kernel PCA to the output of the POKer in order to reduce the dimensionality of the feature space and extract features which can be used to distinguish regulatory elements from non-regulatory ones. Next, we select the top ranked features and analyse the sequence properties of the elements grouped based on these features. According to our findings, we define a regulatory grammar for CNEs that can be used to predict their regulatory activity in a specific tissue. We discuss the biological relevance of these findings and compare them to the evidence available from the existing literature.

7.1 Related Work

The two main approaches to understanding the grammar of regulatory elements are functional assays and methods for identifying *cis*-regulatory elements (see Section 2.3). In order to predict regulatory activity from sequence, these approaches try to infer the rules of transcriptional grammar, including the activity and specificity of TFs, and the number, nucleosome positioning, orientation and order of TFBSs, as well as their co-association [218].

In reporter assays, the sequence of interest is fused to a reporter gene (a gene whose expression is easy to observe, e.g., the green fluorescent protein) and the expression driven by it is measured [219]. However, since each sequence is tested individually, this approach is time-consuming and labour-intensive and therefore low-throughput. In contrast, in high-throughput assays, variants of the sequence of interest are synthesised and each of them is inserted into a plasmid containing a unique 'barcode'. All plasmids are then simultaneously transfected into cells. The regulatory activity of each variant is identified by its barcode using next-generation sequencing. This approach, however, also has a number of limitations [16]. To address these problems, functional assays are commonly combined with computational methods such as phylogenetic footprinting [108] and composite motif discovery methods [220]. As described in Section 2.3, these methods rely on sequence conservation and the combinatorial nature of TF interactions, respectively. Computational methods that do not exploit this information instead often rely on ChIP data [221, 222]. There exist methods that do not use the above information and focus on sequence features alone. An example is the k-mer-based method presented in [223] and its gapped variant [224]. These methods use various k-mer-based kernels (e.g., the spectrum kernel [132]) to compare the sequences. Our approach is similar, with the difference that it considers the sequence of TFBFs identified in a CNE, rather than its nucleotide sequence, and uses the POKer to compare the sequences.

7.2 Detecting Regulatory Signatures in CNEs

We apply kernel principal component analysis (kernel PCA) [225] for feature extraction to detect the regulatory sequence signatures in CNEs.

Principal component analysis (PCA) is a technique which, given N variables, aims to find a set of M ($M \ll N$) uncorrelated variables, called principal components and defined as linear combinations of the initial variables, that would explain most of the variance in the data [226]. Hence, PCA is commonly used for

two purposes: to reduce the dimensionality of the data and to extract features (local and global). An example of the use of PCA for these purposes is the popular eigenfaces approach to face recognition [227]. In this approach, PCA is used to find the principal components (the eigenvectors of the covariance matrix) of a set of vectors representing face images. When visualised, each of these eigenvectors resembles a ghostly face called an eigenface. From an information theoretic point of view, eigenfaces capture the information content in a face image. The k most informative eigenfaces, i.e., the eigenvectors corresponding to the k largest eigenvalues, define the face space. A face image can be projected onto this space and be represented as a linear combination of these eigenfaces, where the weights of the linear combination determine the identity of the person.

While PCA is successful in removing second-order dependencies in the data, it does not deal with higher-order dependencies, i.e., the relationships among three or more variables. In some cases, removing second-order dependencies is not sufficient and important information is contained in higher-order dependencies. Kernel PCA allows extracting such non-linear features. For instance, the kernel eigenfaces approach has been shown to outperform the classic eigenfaces approach [228]. Following is a description of kernel PCA.

Given data points $x_1, x_2, \ldots, x_n$ ($x_i \in \mathbb{R}^N, 1 \leq i \leq n$) and the mapped data $\phi(x_1), \phi(x_2), \ldots, \phi(x_n)$ (see Section 2.4), the covariance matrix in the feature space $\mathcal{F}$ is given by

$$C = \frac{1}{n} \sum_{j=1}^{n} \phi(x_j)\phi(x_j)^T \tag{7.1}$$

The aim is to find eigenvalues $\lambda \geq 0$ and eigenvectors $v \in \mathcal{F} \backslash \{0\}$ that satisfy

$$\lambda v = Cv \tag{7.2}$$

Substituting Equation 7.1 in Equation 7.2 shows that all solutions v lie in the span of $\phi(x_1), \phi(x_2), \ldots, \phi(x_n)$, that is, there exist coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ which satisfy

$$v = \sum_{k=1}^{n} \alpha_k \phi(x_k) \tag{7.3}$$

Furthermore, from Equation 7.2 we have

$$\lambda(\phi(x_i).v) = (\phi(x_i).Cv) \qquad \forall i = 1, \ldots, n \tag{7.4}$$

Substituting Equations 7.1 and 7.3 into Equation 7.4 yields

$$n\lambda K\alpha = K^2\alpha \tag{7.5}$$

where $K_{ij} := (\phi(x_i).\phi(x_j))$ is an $n \times n$ matrix and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$. For non-zero eigenvalues, the solution to Equation 7.5 is equivalent to the solution to

$$n\lambda\alpha = K\alpha \tag{7.6}$$

Solving Equation 7.6, solutions $\alpha^m, \ldots, \alpha^n$ are obtained, where λ_m is the first non-zero eigenvalue (eigenvalues are sorted in the ascending order). The solutions are normalised by requiring that $(v^i.v^i) = \lambda_i(\alpha^i.\alpha^i) = 1$ $(m \leq i \leq n)$. To extract the principal components, a test data point x is projected onto the eigenvectors v^i according to

$$(v^i.\phi(x)) = \sum_{k=1}^{n} \alpha_k^i(\phi(x_k).\phi(x)) = \sum_{k=1}^{n} \alpha_k^i \mathcal{K}(x_k, x) \tag{7.7}$$

where $\phi(x_k).\phi(x)$ is computed using the kernel.

Note that we assumed that the mapped data is centred, i.e., $\sum_{i=1}^{n} \phi(x_i) = 0$; otherwise, the centred data points are given by $\phi_c(x_i) = \phi(x_i) - \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)$, and the centred K is given by $K_c = K - 1_n K - K1_n + 1_n K 1_n$, where $(1_n)_{ij} := \frac{1}{n}$.

In an approach similar to the kernel eigenfaces, we use kernel PCA to capture higher-order correlations in the data in terms of TFBSs composition, and define a regulatory sequence signature for hindbrain enhancers. Note that in kernel PCA, unlike in the classic PCA, reconstruction of the data from the principal components in the original input space is not straightforward; one can only find an approximate reconstruction using a suitable method (e.g., a regression method) depending on the choice of kernel [229]. Therefore, we analyse the data by looking directly at the projection coefficients from Equation 7.7.

7.3 Experimental Setup

To begin with, we built a binary SVM classifier using the POKer to predict whether a CNE drives the expression of genes in the hindbrain.

In Chapter 4, we identified the shared sequence signatures made up of co-occurring motifs in a set of 426 CNEs from the CONDOR database [175], a subset of which (103 sequences) has been functionally validated for enhancer

activity in the hindbrain [193]. Here, we used these 103 CNEs for training our classifier. We labelled the elements that tested positive for hindbrain enhancer activity in at least 20% of the expressing embryos within 48hpf as hindbrain positive (hb+) and the remaining elements as hindbrain negative (hb−). To test our classifier, we used a set of 56 CNEs chosen as follows: we searched the CONDOR database for functionally annotated CNEs that were shown to drive the expression of genes in the hindbrain and other tissues in at least 20% of the expressing embryos within 48hpf. From the results, we removed those elements that appear in the set used for training the classifier. The sequences in both sets were scanned by FIMO [174] (p-value ≤ 0.001) to find the occurrences of TFBSs for 31 TFs involved in developmental patterning. For details of these TFBSs, see Section 4.5.

Before training the classifier, we performed a 5-fold cross-validation on the first set of CNEs in order to obtain the mean accuracy of the classifier on this dataset. We compared this value to the accuracy of signature search-based predictions, where a shared sequence signature composed of MEIS and PBX-HOX motifs co-occurring within 100bp of one another was identified and used to predict hb+ elements in this dataset. An element was considered to be a hindbrain enhancer if it contained this signature [193].

We then trained the classifier on the 103 CNEs from the first set, using the hb+ elements as positive examples and the hb− elements as negative examples, and tested it on the 56 CNEs from the second set. In all cases, the match and mismatch scores for the POKer were set as described in Section 4.5, and the gap penalty was set to -1. We ran the POKer with several β values, ranging from 0 to ∞, and chose the one that yielded the best performance in the cross-validation (β=0.5).

Finally, we applied kernel PCA to the second set of CNEs and visualised the result.

7.3.1 Results

The mean accuracy achieved by our classifier in the cross-validation on the set of 103 CNEs is 0.73. In comparison, searching for the presence of the MEIS plus PBX-HOX signature yields an accuracy of 0.81. Hence, the classifier achieves a comparable accuracy to the signature search-based approach in this case. Note that, contrary to the signature search-based approach, our classifier does not make use of any information on the known interactions between TFs or their function; it only scores the alignments between TFBSs based on their frequency

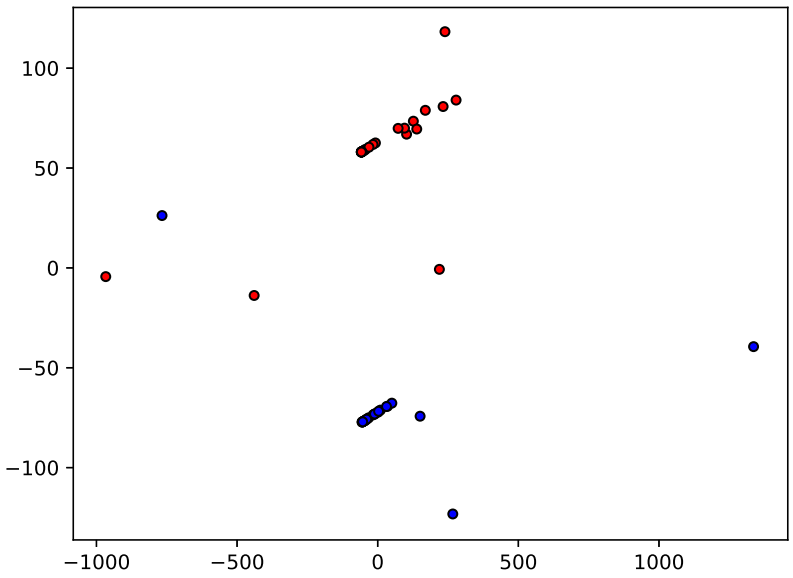


Figure 7.1: Projection of CNEs onto the third and fourth principal components. Points corresponding to the hb+ and hb− elements are coloured in blue and red, respectively.

in the dataset.

We evaluated the performance of the POKer on the set of 56 test CNEs using the AUROC value obtained by the classifier. The classifier achieves an AUROC value of 0.76, indicating a good performance.

Figure 7.1 shows the projection coefficients of the test CNEs obtained using kernel PCA. The principal components 3 to 5 separate the two clusters. We refer to the clusters overlapping with the hb+ and hb− sets of test CNEs as the hb+ and hb− clusters, respectively. For each cluster, we examined the distribution of the considered TFBSs in its elements. Specifically, we analysed the enrichment of each TFBS using AME [230], available in the MEME Suit toolkit [184].

Among the TFBSs enriched in CNEs from the hb+ cluster are motifs from the TFAP, SOX, POU, PBX-HOX, ZIC and MEIS families. TFAP is essential for the development of the hindbrain [231], both SOX and POU have been shown to be involved in the development of the central nervous system (CNS) [232], and the interactions between PBX-HOX, ZIC and MEIS were discussed in Section 4.5.1. The contributions of these TFBSs to identifying hindbrain enhancers have also been reported in other studies. For an example study, see [233]. In comparison, CNEs from the hb− cluster are enriched for motifs from the POU and HMX families. While POU is enriched in CNEs from both clusters, its enrichment in the hb+ cluster is slightly stronger (p-value=3.65E-04 vs p-value=4.82E-04).

Additionally, we looked at the orientation and spacing (up to 100bp) of these TFBSs. We observed no preferences for the orientation of these motifs in CNEs from either cluster. In a number of CNEs from the hb+ cluster, PBX-HOX and MEIS co-occur multiple times within a short window of 50bp. In comparison, CNEs from the hb− cluster contain fewer co-occurrences of these two motifs at larger average inter-motif distances (44.7bp vs 21.8bp). Proximity of PBX-HOX motifs to MEIS motifs has also been observed in the dataset used for training the classifier [193]. In Chapter 4, we detected this syntax in CNEs from the same dataset using an alignment-based approach, suggesting that it may be important for hindbrain enhancer function. According to these findings, a shared sequence signature consisting of enriched TFAP, SOX, POU and ZIC TFBSs plus PBX-HOX and MEIS TFBSs in close proximity characterises the hindbrain enhancers in the dataset on which the classifier was tested.

The kernel PCA results show that the POKer captures the regulatory signatures in CNEs. Hence, as shown by the classification results, our approach using the POKer successfully groups CNEs into functionally related elements, and can be employed to define a tissue-specific regulatory grammar which can be used to predict the enhancer activity of additional CNEs.

7.4 Availability

The following are provided in the supplementary materials: the training and test datasets, the FIMO output files, the results of functional assays, the CONDOR database search results, the Gram matrix produced by the POKer, the script for running cross-validation, the script for running kernel PCA and visualising the projected data, and the AME output.

Concluding Remarks

In this thesis, we investigated the use of machine learning algorithms for predicting the regulatory functions of CNEs in order to gain new insights into the nature of their extreme conservation. To do so, we required, first, a model of CNEs that encapsulates the regulatory sequence signatures present in the elements; and second, a measure of similarity between the modelled CNEs that incorporates their regulatory grammar.

In modelling CNEs, we took a different approach from the current methods by representing the elements based on the TFBSs they contain, instead of their primary sequence. In doing so, we asked the question "which model of TF interactions is the regulatory grammar of CNEs consistent with?". We began by considering metrics that take only the number of TFBSs into account, and used them with several existing algorithms which we had previously successfully employed in other works (Appendices A and B) in an attempt to group functionally related CNEs together. We validated the results of applying this approach to a set of CNEs by functional assays, which suggested that such metrics do not fully capture the regulatory signatures in CNEs (Chapter 3). To obtain a better TFBS-based representation of CNEs, we modelled the elements as partial order graphs, and used a dynamic programming algorithm to align the graphs and identify the regulatory signatures composed of over-represented co-occurring TFBSs that are indicators of potential regulatory activity. The results of testing this approach on a set of CNEs showed that our proposed model, which accounts for the number of TFBSs as well as their relative position, better captures the regulatory signatures in CNEs (Chapter 4).

With an efficient model of CNEs at hand, we searched for a way to measure the similarity between two CNEs. The score of the graph alignment method that we presented in Chapter 4 is not metric; hence, we developed the partial order kernel (POKer) for comparison of partial order graphs, which provides a metric interpretation of this score. To overcome the lack of a suitable benchmark for assessing the POKer's performance, we extended a popular string kernel.

In a series of experiments on artificial data, we demonstrated the effectiveness of our kernel in classifying strings with alternative substrings in comparison to this kernel (Chapter 5). We further evaluated the POKer in a computer vision task, where we represented image sequences as partial order graphs, and used the POKer to compare and match the graphs. The results of experiments on a standard dataset demonstrated the robustness of our proposed visual localisation method as compared to the state-of-the-art methods (Chapter 6). Finally, we employed the POKer in an approach to classifying CNEs, modelled as partial order graphs, into groups of functionally related elements. We tested this approach on a set of CNEs and showed that it achieves a comparable accuracy to predictions made based on the presence of known regulatory signatures. In addition, we applied kernel PCA using the POKer to detect the regulatory signatures in this set of CNEs, and discussed the biological relevance of the results, supported by findings reported in the literature (Chapter 7).

In summary, we introduced a new representation of CNEs and novel methods for both identifying the regulatory signatures in CNEs and predicting the regulatory functions of these elements based on the detected signatures. This was made possible by the introduction of the POKer, a new graph-based kernel of general applicability. Our approach, compared to the existing methods (e.g., gene knockdown experiments), is fast and does not require any prior knowledge of the sequences, although such information can be incorporated into our methods in the future, for instance through the choice of scoring matrices. Another possible future direction is to find an approximate reconstruction of prototypical sequences from the non-linear principal components extracted by the kernel PCA. These could be used to infer a tissue-specific regulatory grammar for CNEs. The problem of reconstruncting data from non-linear principal components has been considered for marginalised kernels and undirected graphs [234].

Overall, the results presented in this thesis further confirm the relationship between the syntax of conserved non-coding sequences and the expression patterns driven by these elements. The methods used in this thesis can be applied to characterise such relationships and learn tissue-specific regulatory grammars for CNEs, which, in turn, allow us to uncover the functions that contribute to the conservation of CNEs.

References

[1] Margulies EH, Birney E. Approaches to comparative sequence analysis: towards a functional view of vertebrate genomes. Nature Reviews Genetics. 2008;9(4):303–313.

[2] Elgar G, Vavouri T. Tuning in to the signals: noncoding sequence conservation in vertebrate genomes. Trends in Genetics. 2008;24(7):344–352.

[3] Palazzo AF, Gregory TR. The case for junk DNA. PLoS Genetics. 2014;10(5):e1004351.

[4] Harmston N, Barešić A, Lenhard B. The mystery of extreme non-coding conservation. Philosophical Transactions of the Royal Society B: Biological Sciences. 2013;368(1632):20130021.

[5] Bejerano G, Pheasant M, Makunin I, Stephen S, Kent WJ, Mattick JS, et al. Ultraconserved elements in the human genome. Science. 2004;304(5675):1321–1325.

[6] Woolfe A, Goodson M, Goode DK, Snell P, McEwen GK, Vavouri T, et al. Highly conserved non-coding sequences are associated with vertebrate development. PLoS Biology. 2004;3(1):e7.

[7] Sandelin A, Bailey P, Bruce S, Engström PG, Klos JM, Wasserman WW, et al. Arrays of ultraconserved non-coding regions span the loci of key developmental genes in vertebrate genomes. BMC Genomics. 2004;5(1):99.

[8] Shin JT, Priest JR, Ovcharenko I, Ronco A, Moore RK, Burns CG, et al. Human-zebrafish non-coding conserved elements act in vivo to regulate transcription. Nucleic Acids Research. 2005;33(17):5437–5445.

[9] Pennacchio LA, Ahituv N, Moses AM, Prabhakar S, Nobrega MA, Shoukry M, et al. In vivo enhancer analysis of human conserved non-coding sequences. Nature. 2006;444(7118):499.

[10] Berman BP, Nibu Y, Pfeiffer BD, Tomancak P, Celniker SE, Levine M, et al. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. Proceedings of the National Academy of Sciences. 2002;99(2):757–762.

[11] Levy S, Hannenhalli S, Workman C. Enrichment of regulatory signals in conserved non-coding genomic sequence. Bioinformatics. 2001;17(10):871–877.

[12] Kulkarni MM, Arnosti DN. Information display by transcriptional enhancers. Development. 2003;130(26):6569–6575.

[13] Walters MC, Fiering S, Eidemiller J, Magis W, Groudine M, Martin DI. Enhancers increase the probability but not the level of gene expression. Proceedings of the National Academy of Sciences. 1995;92(15):7125–7129.

[14] Wittkopp PJ, Kalay G. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. Nature Reviews Genetics. 2012;13(1):59.

[15] Ahituv N, Zhu Y, Visel A, Holt A, Afzal V, Pennacchio LA, et al. Deletion of ultraconserved elements yields viable mice. PLoS Biology. 2007;5(9):e234.

[16] Dailey L. High throughput technologies for the functional discovery of mammalian enhancers: new approaches for understanding transcriptional regulatory network dynamics. Genomics. 2015;106(3):151–158.

[17] Annunziato A. DNA packaging: nucleosomes and chromatin. Nature Education. 2008;1(1):26.

[18] Crick F. Central dogma of molecular biology. Nature. 1970;227(5258):561.

[19] Li B, Carey M, Workman JL. The role of chromatin during transcription. Cell. 2007;128(4):707–719.

[20] Snustad DP, Simmons MJ. Principles of genetics. Wiley New Jersey; 2010.

[21] Griffiths AJF, Miller JH, Suzuki DT, Lewontin RC, Gelbart WM. An introduction to genetic analysis. New York: W. H. Freeman; 2000.

[22] Wilson DN, Cate JHD. The structure and function of the eukaryotic ribosome. Cold Spring Harbor Perspectives in Biology. 2012;4(5):a011536.

[23] Burley SK. The TATA box binding protein. Current Opinion in Structural Biology. 1996;6(1):69–75.

[24] Heintzman ND, Stuart RK, Hon G, Fu Y, Ching CW, Hawkins RD, et al. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. Nature Genetics. 2007;39(3):311.

[25] Zippo A, Serafini R, Rocchigiani M, Pennacchini S, Krepelova A, Oliviero S. Histone crosstalk between H3S10ph and H4K16ac generates a histone code that mediates transcription elongation. Cell. 2009;138(6):1122–1136.

[26] Liu W, Ma Q, Wong K, Li W, Ohgi K, Zhang J, et al. Brd4 and JMJD6-associated anti-pause enhancers in regulation of transcriptional pause release. Cell. 2013;155(7):1581–1595.

[27] Ørom UA, Derrien T, Beringer M, Gumireddy K, Gardini A, Bussotti G, et al. Long noncoding RNAs with enhancer-like function in human cells. Cell. 2010;143(1):46–58.

[28] Shlyueva D, Stampfel G, Stark A. Transcriptional enhancers: from properties to genome-wide predictions. Nature Reviews Genetics. 2014;15(4):272–286.

[29] Blackwood EM, Kadonaga JT. Going the distance: a current view of enhancer action. Science. 1998;281(5373):60–63.

[30] Bulger M, Groudine M. Looping versus linking: toward a model for long-distance gene activation. Genes & Development. 1999;13(19):2465–2477.

[31] Engel JD, Tanimoto K. Looping, linking, and chromatin activity: new insights into beta-globin locus regulation. Cell. 2000;100(5):499–502.

[32] Kadauke S, Blobel GA. Chromatin loops in gene regulation. Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms. 2009;1789(1):17–25.

[33] Epstein DJ. Cis-regulatory mutations in human disease. Briefings in Functional Genomics & Proteomics. 2009;8(4):310–316.

[34] Orphanides G, Lagrange T, Reinberg D. The general transcription factors of RNA polymerase II. Genes & Development. 1996;10(21):2657–2683.

[35] Reese JC. Basal transcription factors. Current Opinion in Genetics & Development. 2003;13(2):114–118.

[36] Compe E, Egly JM. TFIIH: when transcription met DNA repair. Nature Reviews Molecular Cell Biology. 2012;13(6):343–354.

[37] Cohen P, Foulkes JG. The hormonal control of gene transcription. vol. 6. Elsevier; 2012.

[38] Latchman DS. Transcription factors: an overview. The International Journal of Biochemistry & Cell Biology. 1997;29(12):1305–1312.

[39] Yang VW. Eukaryotic transcription factors: identification, characterization and functions. The Journal of Nutrition. 1998;128(11):2045–2051.

[40] Stewart AJ, Plotkin JB. Why transcription factor binding sites are ten nucleotides long. Genetics. 2012;192(3):973—-985.

[41] Ravasi T, Suzuki H, Cannistraci CV, Katayama S, Bajic VB, Tan K, et al. An atlas of combinatorial transcriptional regulation in mouse and man. Cell. 2010;140(5):744–752.

[42] Levo M, Segal E. In pursuit of design principles of regulatory sequences. Nature Reviews Genetics. 2014;15(7):453–468.

[43] Spitz F, Furlong EEM. Transcription factors: from enhancer binding to developmental control. Nature Reviews Genetics. 2012;13(9):613.

[44] Arnosti DN, Kulkarni MM. Transcriptional enhancers: Intelligent enhanceosomes or flexible billboards? Journal of Cellular Biochemistry. 2005;94(5):890–898.

[45] Junion G, Spivakov M, Girardot C, Braun M, Gustafson EH, Birney E, et al. A transcription factor collective defines cardiac cell fate and reflects lineage history. Cell. 2012;148(3):473–486.

[46] Ohno S. So much" junk" DNA in our genome. In: Brookhaven Symposium in Biology. vol. 23; 1972. p. 366–370.

[47] Ureta-Vidal A, Ettwiller L, Birney E. Comparative genomics: genome-wide analysis in metazoan eukaryotes. Nature Reviews Genetics. 2003;4(4):251–262.

[48] Sakuraba Y, Kimura T, Masuya H, Noguchi H, Sezutsu H, Takahasi KR, et al. Identification and characterization of new long conserved noncoding sequences in vertebrates. Mammalian Genome. 2008;19(10-12):703–712.

[49] Drake JA, Bird C, Nemesh J, Thomas DJ, Newton-Cheh C, Reymond A, et al. Conserved noncoding sequences are selectively constrained and not mutation cold spots. Nature Genetics. 2006;38(2):223.

[50] Katzman S, Kern AD, Bejerano G, Fewell G, Fulton L, Wilson RK, et al. Human genome ultraconserved elements are ultraselected. Science. 2007;317(5840):915.

[51] Hiller M, Schaar BT, Bejerano G. Hundreds of conserved non-coding genomic regions are independently lost in mammals. Nucleic Acids Research. 2012;40(22):11463–11476.

[52] McLean CY, Reno PL, Pollen AA, Bassan AI, Capellini TD, Guenther C, et al. Human-specific loss of regulatory DNA and the evolution of human-specific traits. Nature. 2011;471(7337):216–219.

[53] Lowe CB, Bejerano G, Haussler D. Thousands of human mobile element fragments undergo strong purifying selection near developmental genes. Proceedings of the National Academy of Sciences. 2007;104(19):8005–8010.

[54] Kamal M, Xie X, Lander ES. A large family of ancient repeat elements in the human genome is under strong selection. Proceedings of the National Academy of Sciences of the United States of America. 2006;103(8):2740–2745.

[55] McEwen GK, Goode DK, Parker HJ, Woolfe A, Callaway H, Elgar G. Early evolution of conserved regulatory sequences associated with development in vertebrates. PLoS Genetics. 2009;5(12):e1000762.

[56] Holland LZ, Albalat R, Azumi K, Benito-Gutiérrez È, Blow MJ, Bronner-Fraser M, et al. The amphioxus genome illuminates vertebrate origins and cephalochordate biology. Genome Research. 2008;18(7):1100–1111.

[57] Siepel A, Bejerano G, Pedersen JS, Hinrichs AS, Hou M, Rosenbloom K, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. Genome Research. 2005;15(8):1034–1050.

[58] Vavouri T, Walter K, Gilks WR, Lehner B, Elgar G. Parallel evolution of conserved non-coding elements that target a common set of

developmental regulatory genes from worms to humans. Genome Biology. 2007;8(2):R15.

[59] Baxter L, Jironkin A, Hickman R, Moore J, Barrington C, Krusche P, et al. Conserved noncoding sequences highlight shared components of regulatory networks in dicotyledonous plants. The Plant Cell. 2012;24(10):3949–3965.

[60] Burgess D, Freeling M. The most deeply conserved noncoding sequences in plants serve similar functions to those in vertebrates despite large differences in evolutionary rates. The Plant Cell. 2014;26(3):946–961.

[61] Vavouri T, Lehner B. Conserved noncoding elements and the evolution of animal body plans. Bioessays. 2009;31(7):727–735.

[62] Walter K, Abnizova I, Elgar G, Gilks WR. Striking nucleotide frequency pattern at the borders of highly conserved vertebrate non-coding sequences. Trends in Genetics. 2005;21(8):436–440.

[63] Chiang CWK, Derti A, Schwartz D, Chou MF, Hirschhorn JN, Others. Ultraconserved elements: analyses of dosage sensitivity, motifs and boundaries. Genetics. 2008;180(4):2277–2293.

[64] Dimitrieva S, Bucher P. Genomic context analysis reveals dense interaction network between vertebrate ultraconserved non-coding elements. Bioinformatics. 2012;28(18):i395–i401.

[65] Kikuta H, Laplante M, Navratilova P, Komisarczuk AZ, Engström PG, Fredman D, et al. Genomic regulatory blocks encompass multiple neighboring genes and maintain conserved synteny in vertebrates. Genome Research. 2007;17(5):545–555.

[66] Engström PG, Sui SJH, Drivenes Ø, Becker TS, Lenhard B. Genomic regulatory blocks underlie extensive microsynteny conservation in insects. Genome Research. 2007;17(12):1898–1908.

[67] Lettice LA, Heaney SJH, Purdie LA, Li L, de Beer P, Oostra BA, et al. A long-range Shh enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. Human Molecular Genetics. 2003;12(14):1725–1735.

[68] Ahituv N, Prabhakar S, Poulin F, Rubin EM, Couronne O. Mapping cis-regulatory domains in the human genome using multi-species conservation of synteny. Human Molecular Genetics. 2005;14(20):3057–3063.

[69] Irimia M, Tena JJ, Alexis MS, Fernandez-Miñan A, Maeso I, Bogdanović O, et al. Extensive conservation of ancient microsynteny across metazoans due to cis-regulatory constraints. Genome Research. 2012;22(12):2356–2367.

[70] Goode DK, Snell P, Smith SF, Cooke JE, Elgar G. Highly conserved regulatory elements around the SHH gene may contribute to the maintenance of conserved synteny across human chromosome 7q36. 3. Genomics. 2005;86(2):172–181.

[71] Harmston N, Ing-Simmons E, Tan G, Perry M, Merkenschlager M, Lenhard B. Topologically associating domains are ancient features that coincide with Metazoan clusters of extreme noncoding conservation. Nature Communications. 2017;8(1):441.

[72] Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature. 2012;485(7398):376.

[73] Sanges R, Kalmar E, Claudiani P, D'Amato M, Muller F, Stupka E. Shuffling of cis-regulatory elements is a pervasive feature of the vertebrate lineage. Genome Biology. 2006;7(7):R56.

[74] Nobrega MA, Ovcharenko I, Afzal V, Rubin EM. Scanning human gene deserts for long-range enhancers. Science. 2003;302(5644):413.

[75] Visel A, Blow MJ, Li Z, Zhang T, Akiyama JA, Holt A, et al. ChIP-seq accurately predicts tissue-specific activity of enhancers. Nature. 2009;457(7231):854–858.

[76] Visel A, Minovitsky S, Dubchak I, Pennacchio LA. VISTA Enhancer Browsera database of tissue-specific human enhancers. Nucleic Acids Research. 2007;35:D88–D92.

[77] Lettice LA, Horikoshi T, Heaney SJH, van Baren MJ, van der Linde HC, Breedveld GJ, et al. Disruption of a long-range cis-acting regulator for Shh causes preaxial polydactyly. Proceedings of the National Academy of Sciences. 2002;99(11):7548–7553.

[78] Benko S, Fantes JA, Amiel J, Kleinjan DJ, Thomas S, Ramsay J, et al. Highly conserved non-coding elements on either side of SOX9 associated with Pierre Robin sequence. Nature Genetics. 2009;41(3):359.

[79] Roessler E, Hu P, Hong SK, Srivastava K, Carrington B, Sood R, et al. Unique alterations of an ultraconserved non-coding element in the 3 UTR of ZIC2 in holoprosencephaly. PLoS One. 2012;7(7):e39026.

[80] Bondurand N, Fouquet V, Baral V, Lecerf L, Loundon N, Goossens M, et al. Alu-mediated deletion of SOX10 regulatory elements in Waardenburg syndrome type 4. European Journal of Human Genetics. 2012;20(9):990.

[81] Dathe K, Kjaer KW, Brehm A, Meinecke P, Nürnberg P, Neto JC, et al. Duplications involving a conserved regulatory element downstream of BMP2 are associated with brachydactyly type A2. The American Journal of Human Genetics. 2009;84(4):483–492.

[82] Kurth I, Klopocki E, Stricker S, van Oosterwijk J, Vanek S, Altmann J, et al. Duplications of noncoding elements 5 of SOX9 are associated with brachydactyly-anonychia. Nature Genetics. 2009;41(8):862.

[83] Amiel J, Benko S, Gordon CT, Lyonnet S. Disruption of long-distance highly conserved noncoding elements in neurocristopathies. Annals of the New York Academy of Sciences. 2010;1214(1):34–46.

[84] Spieler D, Kaffe M, Knauf F, Bessa J, Tena JJ, Giesert F, et al. Restless legs syndrome-associated intronic common variant in Meis1 alters enhancer function in the developing telencephalon. Genome Research. 2014;24(4):592–603.

[85] Martinez AF, Abe Y, Hong S, Molyneux K, Yarnell D, Löhr H, et al. An ultraconserved brain-specific enhancer within ADGRL3 (LPHN3) underpins attention-deficit/hyperactivity disorder susceptibility. Biological Psychiatry. 2016;80(12):943–954.

[86] Birney E, Stamatoyannopoulos JA, Dutta A, Guigó R, Gingeras TR, Margulies EH, et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature. 2007;447(7146):799–816.

[87] Weirauch MT, Hughes TR. Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same. Trends in Genetics. 2010;26(2):66–74.

[88] Hare EE, Peterson BK, Iyer VN, Meier R, Eisen MB. Sepsid even-skipped enhancers are functionally conserved in Drosophila despite lack of sequence conservation. PLoS Genetics. 2008;4(6):e1000106.

[89] Blow MJ, McCulley DJ, Li Z, Zhang T, Akiyama JA, Holt A, et al. ChIP-Seq identification of weakly conserved heart enhancers. Nature Genetics. 2010;42(9):806.

[90] Nobrega MA, Zhu Y, Plajzer-Frick I, Afzal V, Rubin EM. Megabase deletions of gene deserts result in viable mice. Nature. 2004;431(7011):988–993.

[91] Dickel DE, Ypsilanti AR, Pla R, Zhu Y, Barozzi I, Mannion BJ, et al. Ultraconserved Enhancers Are Required for Normal Development. Cell. 2018;.

[92] Osterwalder M, Barozzi I, Tissières V, Fukuda-Yuzawa Y, Mannion BJ, Afzal SY, et al. Enhancer redundancy provides phenotypic robustness in mammalian development. Nature. 2018;.

[93] Asgari E, Mofrad MRK. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS One. 2015;10(11):e0141287.

[94] Ng P. dna2vec: Consistent vector representations of variable-length k-mers. arXiv preprint arXiv:170106279. 2017;.

[95] Liao B, Wang TM. New 2D graphical representation of DNA sequences. Journal of Computational Chemistry. 2004;25(11):1364–1368.

[96] Yao Yh, Wang Tm. A class of new 2-D graphical representation of DNA sequences and their application. Chemical Physics Letters. 2004;398(4):318–323.

[97] Liao B, Ding K. A 3D graphical representation of DNA sequences and its application. Theoretical Computer Science. 2006;358(1):56–64.

[98] Randić M, Butina D, Zupan J. Novel 2-D graphical representation of proteins. Chemical Physics Letters. 2006;419(4):528–532.

[99] Arvestad L. Algorithms for biological sequence alignment. Royal Institute of Technology; 1999.

[100] Gîrdea M, Noé L, Kucherov G. Back-translation for discovering distant protein homologies in the presence of frameshift mutations. Algorithms for Molecular Biology. 2010;5(1):6.

[101] Dayhoff MO, Schwartz RM, Orcutt BC. A model of evolutionary change in proteins. In: Atlas of Protein Sequence and Structure. vol. 5. National Biomedical Research Foundation Silver Spring, MD; 1978. p. 345–352.

[102] Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences. 1992;89(22):10915–10919.

[103] Zvelebil M, Baum J. Understanding bioinformatics. Garland Science; 2007.

[104] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology. 1970;48(3):443–453.

[105] Smith TF, Waterman MS. Identification of common molecular subsequences. Journal of Molecular Biology. 1981;147(1):195–197.

[106] Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proceedings of the National Academy of Sciences. 1988;85(8):2444–2448.

[107] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of Molecular Biology. 1990;215(3):403–410.

[108] Tagle DA, Koop BF, Goodman M, Slightom JL, Hess DL, Jones RT. Embryonic epsilon and gamma globin genes of a prosimian primate (Galago crassicaudatus): Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. Journal of Molecular Biology. 1988;203(2):439–455.

[109] Hardison RC. Comparative genomics. PLoS Biology. 2003;1(2):e58.

[110] Rubin GM, Yandell MD, Wortman JR, Gabor GL, Nelson CR, Hariharan IK, et al. Comparative genomics of the eukaryotes. Science. 2000;287(5461):2204–2215.

[111] Cliften PF, Hillier LW, Fulton L, Graves T, Miner T, Gish WR, et al. Surveying Saccharomyces genomes to identify functional elements by comparative DNA sequence analysis. Genome Research. 2001;11(7):1175–1186.

[112] Frazer KA, Elnitski L, Church DM, Dubchak I, Hardison RC. Cross-species sequence comparisons: a review of methods and available resources. Genome Research. 2003;13(1):1–12.

[113] Waterson RH, Lander ES, Wilson RK, Others. Initial sequence of the chimpanzee genome and comparison with the human genome. Nature. 2005;437(7055):69.

[114] Hubbard T, Barker D, Birney E, Cameron G, Chen Y, Clark L, et al. The Ensembl genome database project. Nucleic Acids Research. 2002;30(1):38–41.

[115] Mayor C, Brudno M, Schwartz JR, Poliakov A, Rubin EM, Frazer KA, et al. VISTA: visualizing global DNA sequence alignments of arbitrary length. Bioinformatics. 2000;16(11):1046–1047.

[116] Schwartz S, Zhang Z, Frazer KA, Smit A, Riemer C, Bouck J, et al. PipMakera web server for aligning two genomic DNA sequences. Genome Research. 2000;10(4):577–586.

[117] van Helden J, l\'\i del Olmo M, Pérez-Ort\'\in JE. Statistical analysis of yeast genomic downstream sequences reveals putative polyadenylation signals. Nucleic Acids Research. 2000;28(4):1000–1010.

[118] Bussemaker HJ, Li H, Siggia ED. Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis. Proceedings of the National Academy of Sciences. 2000;97(18):10096–10100.

[119] Narlikar L, Sakabe NJ, Blanski AA, Arimura FE, Westlund JM, Nobrega MA, et al. Genome-wide discovery of human heart enhancers. Genome Research. 2010;20(3):381–392.

[120] Won KJ, Chepelev I, Ren B, Wang W. Prediction of regulatory elements in mammalian genomes using chromatin signatures. BMC Bioinformatics. 2008;9(1):547.

[121] Firpi HA, Ucar D, Tan K. Discover regulatory DNA elements using chromatin signatures and artificial neural network. Bioinformatics. 2010;26(13):1579–1586.

[122] Rajagopal N, Xie W, Li Y, Wagner U, Wang W, Stamatoyannopoulos J, et al. RFECS: a random-forest based algorithm for enhancer identification from chromatin state. PLoS Computational Biology. 2013;9(3):e1002968.

[123] Fang Y, Wang Y, Zhu Q, Wang J, Li G. In silico identification of enhancers on the basis of a combination of transcription factor binding motif occurrences. Scientific Reports. 2016;6:32476.

[124] Sinha S, Van Nimwegen E, Siggia ED. A probabilistic method to detect regulatory modules. Bioinformatics. 2003;19(suppl_1):i292–i301.

[125] Crowley EM, Roeder K, Bina M. A Statistical Model for Locating Regulatory Regions in Genomic DNA. Journal of Molecular Biology. 1997;268:8–14.

[126] Zhou Q, Wong WH. CisModule: de novo discovery of cis-regulatory modules by hierarchical mixture modeling. Proceedings of the National Academy of Sciences of the United States of America. 2004;101(33):12114–12119.

[127] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. Cambridge University Press; 2004.

[128] Hofmann T, Schölkopf B, Smola AJ. Kernel methods in machine learning. The Annals of Statistics. 2008;p. 1171–1220.

[129] Haussler D. Convolution kernels on discrete structures; 1999.

[130] Tsuda K, Kin T, Asai K. Marginalized kernels for biological sequences. Bioinformatics. 2002;18(suppl_1):S268–S275.

[131] Jaakkola TS, Diekhans M, Haussler D. Using the Fisher kernel method to detect remote protein homologies. In: Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB). vol. 99; 1999. p. 149–158.

[132] Leslie C, Eskin E, Noble WS. The spectrum kernel: A string kernel for SVM protein classification. In: Biocomputing 2002. World Scientific; 2001. p. 564–575.

[133] Leslie C, Kuang R. Fast string kernels using inexact matching for protein sequences. Journal of Machine Learning Research. 2004;5(Nov):1435–1455.

[134] Meinicke P, Tech M, Morgenstern B, Merkl R. Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. BMC Bioinformatics. 2004;5(1):169.

[135] Rätsch G, Sonnenburg S. Accurate splice site detection for Caenorhabditis elegans. Kernel Methods in Computational Biology. 2004;p. 277–298.

[136] Rätsch G, Sonnenburg S, Schölkopf B. RASE: recognition of alternatively spliced exons in C. elegans. Bioinformatics. 2005;21(suppl_1):i369–i377.

[137] Ben-Hur A, Brutlag D. Remote homology detection: a motif based approach. Bioinformatics. 2003;19(suppl_1):i26–i33.

[138] Liao L, Noble WS. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. Journal of Computational Biology. 2003;10(6):857–868.

[139] Karlin S, Altschul SF. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proceedings of the National Academy of Sciences. 1990;87(6):2264–2268.

[140] Vert JP, Saigo H, Akutsu T. Convolution and local alignment kernels. Kernel Methods in Computational Biology. 2004;p. 131–154.

[141] Saigo H, Vert JP, Ueda N, Akutsu T. Protein homology detection using string alignment kernels. Bioinformatics. 2004;20(11):1682–1689.

[142] Cortes C, Haffner P, Mohri M. Rational kernels: theory and algorithms. Journal of Machine Learning Research. 2004;5(Aug):1035–1062.

[143] van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research. 2008;9(Nov):2579–2605.

[144] Guo Y, Chen C, Zhou S. Topology visualisation tool for large-scale communications networks. Electronics Letters. 2007;43(10):597–598.

[145] Vinga S, Almeida J. Alignment-free sequence comparisona review. Bioinformatics. 2003;19(4):513–523.

[146] Blaisdell BE. A measure of the similarity of sets of sequences not requiring sequence alignment. Proceedings of the National Academy of Sciences. 1986;83(14):5155–5159.

[147] Sims GE, Jun SR, Wu GA, Kim SH. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. Proceedings of the National Academy of Sciences. 2009;106(8):2677–2682.

[148] Apostolico A, Guerra C, Pizzi C. Alignment free sequence similarity with bounded hamming distance. In: Data Compression Conference (DCC), 2014. IEEE; 2014. p. 183–192.

[149] Leimeister CA, Boden M, Horwege S, Lindner S, Morgenstern B. Fast alignment-free sequence comparison using spaced-word frequencies. Bioinformatics. 2014;30(14):1991–1999.

[150] Ulitsky I, Burstein D, Tuller T, Chor B. The average common substring approach to phylogenomic reconstruction. Journal of Computational Biology. 2006;13(2):336–350.

[151] Leimeister CA, Morgenstern B. Kmacs: the k-mismatch average common substring approach to alignment-free sequence comparison. Bioinformatics. 2014;30(14):2000–2008.

[152] Haubold B, Pierstorff N, Möller F, Wiehe T. Genome comparison without alignment using shortest unique substrings. BMC Bioinformatics. 2005;6(1):123.

[153] Torney DC, Burks C, Davison D, Sirotkin KM. Computation of d2: a measure of sequence dissimilarity. In: Bell GI, Marr TG, editors. Computers and DNA: the Proceedings of the Interface between Computation Science and Nucleic Acid Sequencing Workshop, held December 12 to 16, 1988 in Santa Fe, New Mexico. Addison-Wesley Publishing Co.; 1990. .

[154] Pearson K. Note on regression and inheritance in the case of two parents. Proceedings of the Royal Society of London. 1895;58:240–242.

[155] Mahalanobis PC. On the generalised distance in statistics. Proceedings of the National Institute of Sciences of India. 1936;p. 49–55.

[156] Kullback S, Leibler RA. On information and sufficiency. The Annals of Mathematical Statistics. 1951;22(1):79–86.

[157] Collins M, Schapire RE, Singer Y. Logistic regression, AdaBoost and Bregman distances. Machine Learning. 2002;48(1-3):253–285.

[158] Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences. 1998;95(25):14863–14868.

[159] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. Nature Genetics. 1999;22(3):281.

[160] Park PJ, Butte AJ, Kohane IS. Comparing expression profiles of genes with similar promoter regions. Bioinformatics. 2002;18(12):1576–1584.

[161] Ward Jr JH. Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association. 1963;58(301):236–244.

[162] Murtagh F, Contreras P. Algorithms for hierarchical clustering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2012;2(1):86–97.

[163] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems; 2002. p. 849–856.

[164] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000;22(8):888–905.

[165] Von Luxburg U. A tutorial on spectral clustering. Statistics and Computing. 2007;17(4):395–416.

[166] Zaki MJ, Meira Jr W, Meira W. Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press; 2014.

[167] Yeung KY, Ruzzo WL. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics. 2001;17(9):763–774.

[168] Bushati N, Smith J, Briscoe J, Watkins C. An intuitive graphical visualization technique for the interrogation of transcriptome data. Nucleic Acids Research. 2011;39(17):7380–7389.

[169] Mahfouz A, van de Giessen M, van der Maaten L, Huisman S, Reinders M, Hawrylycz MJ, et al. Visualizing the spatial gene expression organization in the brain through non-linear similarity embeddings. Methods. 2015;73:79–89.

[170] Taskesen E, Reinders MJT. 2D representation of transcriptomes by t-SNE exposes relatedness between human tissues. PLoS One. 2016;11(2):e0149853.

[171] Babu MM, Luscombe NM, Aravind L, Gerstein M, Teichmann SA. Structure and evolution of transcriptional regulatory networks. Current Opinion in Structural Biology. 2004;14(3):283–291.

[172] Mondragón RJ. Topological modelling of large networks. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. 2008;366(1872):1931–1940.

[173] Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, et al. The human genome browser at UCSC. Genome Research. 2002;12(6):996–1006.

[174] Grant CE, Bailey TL, Noble WS. FIMO: scanning for occurrences of a given motif. Bioinformatics. 2011;27(7):1017–1018.

[175] Woolfe A, Goode DK, Cooke J, Callaway H, Smith S, Snell P, et al. CONDOR: a database resource of developmentally associated conserved non-coding elements. BMC Developmental Biology. 2007;7(1):100.

[176] Rajewsky N, Vergassola M, Gaul U, Siggia ED. Computational detection of genomic cis-regulatory modules applied to body patterning in the early Drosophila embryo. BMC Bioinformatics. 2002;3(1):30.

[177] Johansson Ö, Alkema W, Wasserman WW, Lagergren J. Identification of functional clusters of transcription factor binding motifs in genome sequences: the MSCAN algorithm. Bioinformatics. 2003;19(suppl_1):i169–i176.

[178] Frith MC, Li MC, Weng Z. Cluster-Buster: Finding dense clusters of motifs in DNA sequences. Nucleic Acids Research. 2003;31(13):3666–3668.

[179] Bailey TL, Noble WS. Searching for statistically significant regulatory modules. Bioinformatics. 2003;19(suppl_2):ii16–ii25.

[180] Sinha S, He X. MORPH: probabilistic alignment combined with hidden Markov models of cis-regulatory modules. PLoS Computational Biology. 2007;3(11):e216.

[181] Hallikas O, Palin K, Sinjushina N, Rautiainen R, Partanen J, Ukkonen E, et al. Genome-wide prediction of mammalian enhancers based on analysis of transcription-factor binding affinity. Cell. 2006;124(1):47–59.

[182] Grasso C, Lee C. Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems. Bioinformatics. 2004;20(10):1546–1556.

[183] Lee C, Grasso C, Sharlow MF. Multiple sequence alignment using partial order graphs. Bioinformatics. 2002;18(3):452–464.

[184] Bailey TL, Boden M, Buske FA, Frith M, Grant CE, Clementi L, et al. MEME SUITE: tools for motif discovery and searching. Nucleic Acids Research. 2009;37(suppl_2):W202–W208.

[185] Robasky K, Bulyk ML. UniPROBE, update 2011: expanded content and search tools in the online database of protein-binding microarray data on protein–DNA interactions. Nucleic Acids Research. 2010;39(suppl_1):D124–D128.

[186] Mathelier A, Zhao X, Zhang AW, Parcy F, Worsley-Hunt R, Arenillas DJ, et al. JASPAR 2014: an extensively expanded and updated

open-access database of transcription factor binding profiles. Nucleic Acids Research. 2013;42(D1):D142–D147.

[187] Drummond DL, Cheng CS, Selland LG, Hocking JC, Prichard LB, Waskiewicz AJ. The role of Zic transcription factors in regulating hindbrain retinoic acid signaling. BMC Developmental Biology. 2013;13(1):31.

[188] Biemar F, Devos N, Martial JA, Driever W, Peers B. Cloning and expression of the TALE superclass homeobox Meis2 gene during zebrafish embryonic development. Mechanisms of Development. 2001;109(2):427–431.

[189] Nagai T, Aruga J, Takada S, Günther T, Spörle R, Schughart K, et al. The Expression of the MouseZic1, Zic2, andZic3Gene Suggests an Essential Role forZicGenes in Body Pattern Formation. Developmental Biology. 1997;182(2):299–313.

[190] Jacobs Y, Schnabel CA, Cleary ML. Trimeric association of Hox and TALE homeodomain proteins mediates Hoxb2 hindbrain enhancer activity. Molecular and Cellular Biology. 1999;19(7):5134–5142.

[191] Chang CP, Jacobs Y, Nakamura T, Jenkins NA, Copeland NG, Cleary ML. Meis proteins are major in vivo DNA binding partners for wild-type but not chimeric Pbx proteins. Molecular and cellular biology. 1997;17(10):5679–5687.

[192] Waskiewicz AJ, Rikhof HA, Hernandez RE, Moens CB. Zebrafish Meis functions to stabilize Pbx proteins and regulate hindbrain patterning. Development. 2001;128(21):4139–4151.

[193] Grice J, Noyvert B, Doglio L, Elgar G. A simple predictive enhancer syntax for hindbrain patterning is conserved in vertebrate genomes. PLoS One. 2015;10(7):e0130413.

[194] Rouchka EC, Hardin CT. rMotifGen: random motif generator for DNA and protein sequences. BMC Bioinformatics. 2007;8(1):292.

[195] King T, Butcher S, Zalewski L. Apocrita - High Performance Computing Cluster for Queen Mary University of London; 2017.

[196] Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, Corke P, et al. Visual Place Recognition: A Survey. IEEE Transactions on Robotics. 2016;32(1):1–19.

[197] Ranganathan A, Matsumoto S, Ilstrup D. Towards illumination invariance for visual localization. In: 2013 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2013. p. 3791–3798.

[198] Maddern W, Stewart A, McManus C, Upcroft B, Churchill W, Newman P. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In: Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China. vol. 2; 2014. p. 3.

[199] Valgren C, Lilienthal AJ. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. Robotics and Autonomous Systems. 2010;58(2):149–156.

[200] Carlevaris-Bianco N, Eustice RM. Learning visual feature descriptors for dynamic lighting conditions. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2014. p. 2769–2776.

[201] Gomez-Ojeda R, Lopez-Antequera M, Petkov N, Gonzalez-Jimenez J. Training a convolutional neural network for appearance-invariant place recognition. arXiv preprint arXiv:150507428. 2015;.

[202] Sünderhauf N, Shirazi S, Jacobson A, Dayoub F, Pepperell E, Upcroft B, et al. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. Proceedings of Robotics: Science and Systems XII. 2015;.

[203] Cascianelli S, Costante G, Bellocchio E, Valigi P, Fravolini ML, Ciarfuglia TA. Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features. Robotics and Autonomous Systems. 2017;92:53–65.

[204] Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A. Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;.

[205] Chen Z, Lam O, Jacobson A, Milford M. Convolutional neural network-based place recognition. arXiv preprint arXiv:14111509. 2014;.

[206] Sünderhauf N, Shirazi S, Dayoub F, Upcroft B, Milford M. On the performance of convnet features for place recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2015. p. 4297–4304.

[207] Lowry SM, Milford MJ, Wyeth GF. Transforming morning to afternoon using linear regression techniques. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2014. p. 3950–3955.

[208] Neubert P, Sünderhauf N, Protzel P. Superpixel-based appearance change prediction for long-term navigation across seasons. Robotics and Autonomous Systems. 2015;69:15–27.

[209] Milford MJ, Wyeth GF. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In: 2012 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2012. p. 1643–1649.

[210] Pepperell E, Corke PI, Milford MJ. All-environment visual place recognition with SMART. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2014. p. 1612–1618.

[211] Hansen P, Browning B. Visual place recognition using HMM sequence matching. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2014. p. 4549–4555.

[212] Ho K, Newman P. Multiple map intersection detection using visual appearance. In: International Conference on Computational Intelligence, Robotics and Autonomous Systems; 2005. .

[213] Naseer T, Spinello L, Burgard W, Stachniss C. Robust Visual Robot Localization Across Seasons Using Network Flows. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence; 2014. p. 2564–2570.

[214] Vysotska O, Stachniss C. Lazy data association for image sequences matching under substantial appearance changes. IEEE Robotics and Automation Letters. 2016;1(1):213–220.

[215] Senin P. Dynamic time warping algorithm review. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA. 2008;855:1–23.

[216] Arribas-Gil A, Matias C. A time warping approach to multiple sequence alignment. Statistical Applications in Genetics and Molecular Biology. 2017;16(2):133–144.

[217] Gionis A, Indyk P, Motwani R, Others. Similarity search in high dimensions via hashing. In: Proceedings of the 25th International Conference on Very Large Data Bases (VLDB). vol. 99; 1999. p. 518–529.

[218] Weingarten-Gabbay S, Segal E. The grammar of transcriptional regulation. Human Genetics. 2014;133(6):701–711.

[219] Sittampalam GS, Coussens NP, Brimacombe K, Grossman A, Arkin M. Assay guidance manual; 2004. Available from: https://www.ncbi.nlm.nih.gov/books/NBK53196/.

[220] Sandve GK, Drabløs F. A survey of motif discovery methods in an integrated framework. Biology Direct. 2006;1(1):11.

[221] Zinzen RP, Girardot C, Gagneur J, Braun M, Furlong EEM. Combinatorial binding predicts spatio-temporal cis-regulatory activity. Nature. 2009;462(7269):65.

[222] Wilczynski B, Liu YH, Yeo ZX, Furlong EEM. Predicting spatial and temporal gene expression using an integrative model of transcription factor occupancy and chromatin state. PLoS Computational Biology. 2012;8(12):e1002798.

[223] Lee D, Karchin R, Beer MA. Discriminative prediction of mammalian enhancers from DNA sequence. Genome Research. 2011;.

[224] Ghandi M, Lee D, Mohammad-Noori M, Beer MA. Enhanced regulatory sequence prediction using gapped k-mer features. PLoS Computational Biology. 2014;10(7):e1003711.

[225] Schölkopf B, Smola A, Müller KR. Kernel principal component analysis. In: International Conference on Artificial Neural Networks. Springer; 1997. p. 583–588.

[226] Shlens J. A tutorial on principal component analysis. arXiv preprint arXiv:14041100. 2014;.

[227] Turk MA, Pentland AP. Face recognition using eigenfaces. In: Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 1991. p. 586–591.

[228] Yang MH, Ahuja N, Kriegman D. Face recognition using kernel eigenfaces. In: Proceedings of the 2000 International Conference on Image Processing. vol. 1. IEEE; 2000. p. 37–40.

[229] Schölkopf B, Mika S, Smola A, Rätsch G, Müller KR. Kernel PCA pattern reconstruction via approximate pre-images. In: Perspectives in Neural Computing: Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN). Springer; 1998. p. 147–152.

[230] McLeay RC, Bailey TL. Motif Enrichment Analysis: a unified framework and an evaluation on ChIP data. BMC bioinformatics. 2010;11(1):165.

[231] Holzschuh J, Barrallo-Gimeno A, Ettl AK, Dürr K, Knapik EW, Driever W. Noradrenergic neurons in the zebrafish hindbrain are induced by retinoic acid and require tfap2a for expression of the neurotransmitter phenotype. Development. 2003;130(23):5741–5754.

[232] Kiyota T, Kato A, Altmann CR, Kato Y. The POU homeobox protein Oct-1 regulates radial glia formation downstream of Notch signaling. Developmental Biology. 2008;315(2):579–592.

[233] Burzynski GM, Reed X, Taher L, Stine ZE, Matsui T, Ovcharenko I, et al. Systematic elucidation and in vivo validation of sequences enriched in hindbrain transcriptional control. Genome Research. 2012;.

[234] Bakır GH, Zien A, Tsuda K. Learning to find graph pre-images. In: Joint Pattern Recognition Symposium. Springer; 2004. p. 253–261.

[235] Rubin DM. A simple autocorrelation algorithm for determining grain size from digital images of sediment. Journal of Sedimentary Research. 2004;74(1):160–165.

[236] Turner MR. Texture discrimination by Gabor functions. Biological Cybernetics. 1986;55(2):71–82.

[237] Haralick RM, Shanmugam K, Others. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics. 1973;3(6):610–621.

[238] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002;24(7):971–987.

[239] Bianconi F, Di Maria F, Micale C, Fernández A, Harvey RW. Grain-size assessment of fine and coarse aggregates through bipolar area morphology. Machine Vision and Applications. 2015;26(6):775–789.

[240] Hanbury A, Kandaswamy U, Adjeroh DA. Illumination-invariant morphological texture classification. In: Mathematical Morphology: 40 Years On. Springer; 2005. p. 377–386.

Appendix A

What Skin Texture Tells Us About Gender

As a case study on the t-SNE dimensionality reduction technique [143] (see Section 3.3.2), we investigated the use of skin microtexture for gender recognition. We considered a variety of approaches for feature extraction and applied them to a set of images acquired by two different imaging modalities, namely digital dermoscopy and capacitive imaging using a fingerprint sensor. We then classified the feature vectors using two different methods. Moreover, we performed dimensionality reduction on the features using t-SNE. Statistical analysis of the significance of classification results and the maps obtained using t-SNE both indicate that while skin texture contains useful information for person identification, little can be inferred from it about gender.

A.1 Data Acquisition, Feature Extraction and Classification

Images in our dataset were acquired from 43 subjects (24 males and 19 females) with average age 32.1±14.2 and of various ethnic backgrounds. From each of the back of the hand (BH), forearm (FR) and palm (PL) of the upper left limb of each participant, four images were taken using both a digital dermoscope (ProScope HR2, Bodelin Technologies, USA) and a fingerprint sensor (Epsilon, Biox Systems Ltd, UK). Various features were extracted from each image using the following methods: autocorrelation [235], Gabor filters [236], grey-level co-occurrence matrices (GLCM) [237], local binary patterns (LBP) [238], granulometry [239] and semi-variogram [240]. We considered the feature vectors

Classifier	Image descriptor	Epsilon			ProScope			Epsilon + ProScope		
		BH	***FR***	***PL***	***BH***	***FR***	***PL***	***BH***	***FR***	***PL***
1-NN	**Autocorrelation**	49.42	44.19	50.00	50.58	49.42	47.09	49.42	44.19	50.00
	Gabor filters	50.00	**61.05**	51.74	45.93	53.49	48.84	50.00	**61.05**	51.74
	GLCM	47.67	58.72	49.42	44.77	44.77	48.84	47.67	58.72	49.42
	Granulometry	51.74	54.07	44.77	51.16	51.74	51.16	51.74	54.07	44.77
	LBP	45.93	40.70	52.91	46.51	54.07	44.77	45.93	40.70	52.91
	Semi-variogram	52.33	51.16	51.74	52.33	47.67	52.33	52.33	51.16	51.74
SVMs	**Autocorrelation**	48.26	52.33	49.42	51.16	**63.37**	55.23	49.42	51.74	52.33
	Gabor filters	43.60	53.49	53.49	53.49	58.72	57.56	43.02	54.65	53.49
	GLCM	53.49	53.49	53.49	53.49	53.49	52.91	53.49	53.49	53.49
	Granulometry	55.81	51.74	52.33	59.88	53.49	53.49	49.42	51.74	51.16
	LBP	45.35	52.91	53.49	43.60	51.16	55.23	50.00	56.98	51.74
	Semi-variogram	47.09	51.16	53.49	52.91	54.07	49.42	55.81	51.16	53.49

Table A.1: Overall accuracy values achieved by each feature and classifier combination in the gender recognition task

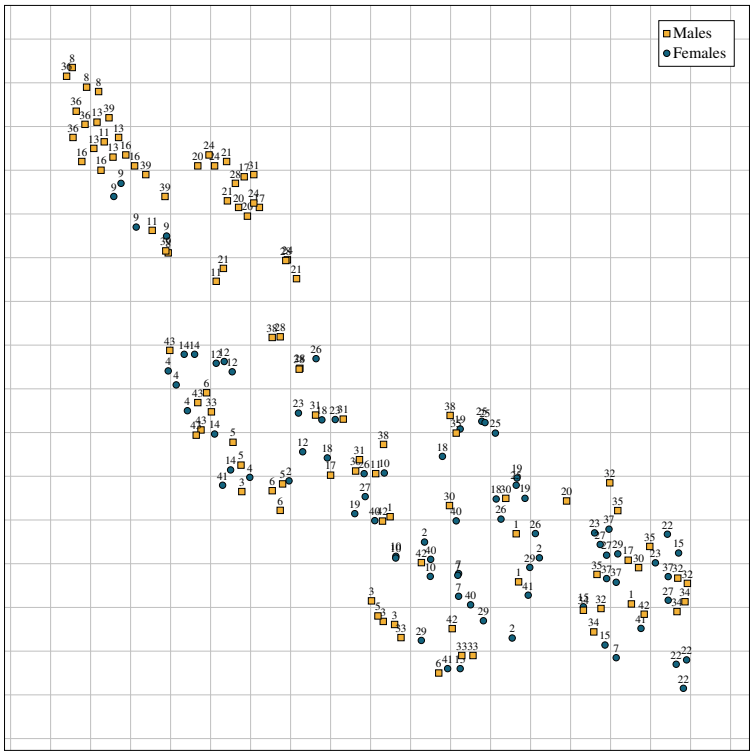
obtained by the Epsilon and those obtained by the ProScope separately, as well as the concatenation of the two vectors (denoted by Epsilon + ProScope).

We classified the images into two classes (male or female) using two different methods: 1-NN classifiers with Euclidean distances (L2) and SVMs with radial basis kernel. In each case, accuracy was computed using a leave-one-out cross-validation strategy, i.e., for each acquisition zone (BH, FR and PL), all four images of each subject in turn were removed from the dataset and images of the remaining subjects were used for training the classifier. The classifier was then tested on the images which had been removed. This way we controlled for the effect of identity-specific information. To further investigate the gender information in these features, we visualised the features that yielded the highest classification accuracy values (autocorrelation/FR/ProScope and Gabor filters/FR/Epsilon + ProScope) using t-SNE.

A.2 Results and Conclusion

As shown in Table A.1, the accuracy values achieved by all classifier and feature combinations are poor. Furthermore, 1-tailed Fisher's exact test to assess the statistical significance of the results (where the null hypothesis is that the estimated and true genders are uncorrelated) yielded a p-value of 0.1, indicating a weak correlation between skin texture and gender. In contrast, consistently good accuracy values were achieved when the same features were used for person identification. For instance, using the Gabor filters/FR/Epsilon + ProScope combination, an accuracy of over 92% was achieved. The results are further confirmed by the t-SNE maps shown in Figure A.1. In these maps, individual subjects are fairly separable, however, their gender does not appear to be. Overall, although far from exhaustive, these results suggest that skin texture carries little information on gender.

Autocorrelation (forearm, Proscope)



Gabor filters (forearm, Epsilon + Proscope)

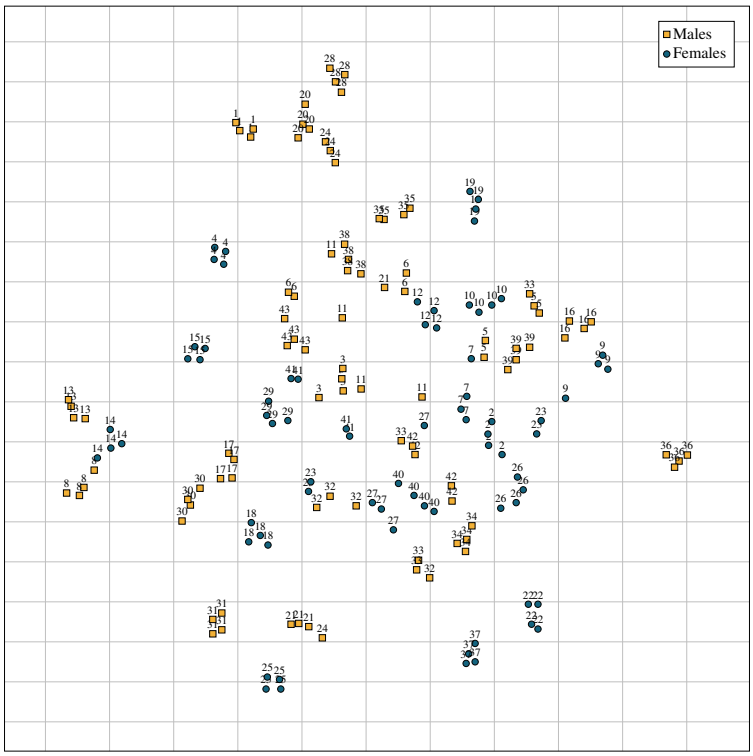


Figure A.1: t-SNE maps displaying the distribution of features used by the two best performing classifiers. Numbers denote identities (there exist four points for each subject), while colours denote gender.

Appendix B

Visualising the Topology of Message Board User Networks

As a case study on the BOSAM algorithm [144], a simple yet effective network visualisation tool (see Section 3.3.3), we present an analysis of the topological structure of user interaction networks on several health-related message boards. We used BOSAM to visualise the network of interactions among users and between users and site administrators on six forums of different scopes and sizes. The results reveal major differences between the user interaction networks of these forums, and show that the BOSAM of each network closely correlates with the characteristics of its respective message board, in terms of coverage (single-topic or multi-topic), presence of user communities, nature of the forum (commercial or voluntary) and administration style.

B.1 Data

We acquired data from six health-related forums: Crohn's Forum (CROHN)[1], HealthBoards (HBOARDS)[2], Huntington's Disease Association (HDA)[3], Inspire (INSPIRE)[4], Patient Info (PINFO)[5] and Psoriasis Association (PSORIASIS)[6] (Table B.1). To model the user interaction network for a forum, we created a

[1] www.crohnsforum.com
[2] www.healthboards.com
[3] www.hdmessageboard.com
[4] www.inspire.com
[5] patient.info
[6] www.psoriasis-association.org.uk

Name	No. of users	No. of posts	Multi-topic	Commercial
CROHN	21703	673162	No	No
HBOARDS	403682	4929836	Yes	Yes
HDA	1245	34173	No	No
INSPIRE	221432	3725555	Yes	Yes
PINFO	139567	1699131	Yes	Yes
PSORIASIS	1816	4402	No	No

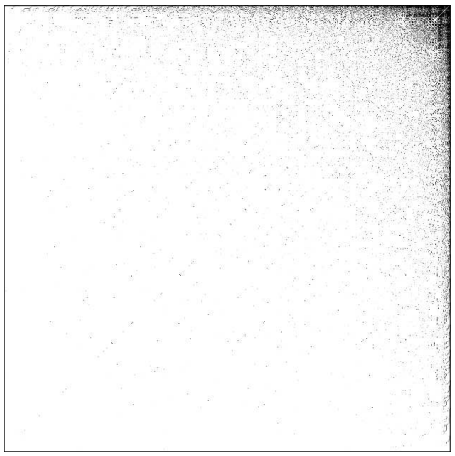
Table B.1: Details of the modelled networks

node for each user and linked two nodes if their corresponding users had posted to the same conversation thread.

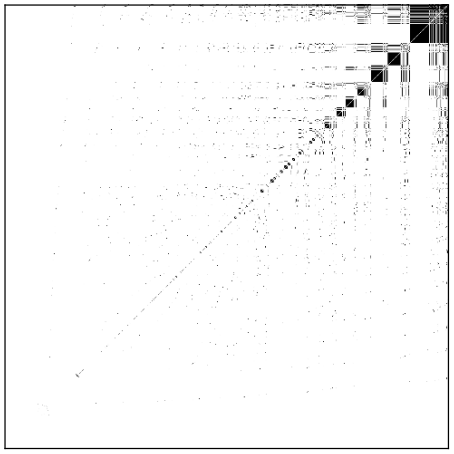
B.2 Results and Discussion

As shown in Figure B.1, in the HDA network connectivity is dominated by the nodes with high degrees, represented by pixels that are densely distributed along the upper right corner of its BOSAM, meaning that low degree users tend to interact with high degree users. Hence, the HDA forum appears as a single-topic forum that patients use to ask questions from expert users. In contrast, in the PSORIASIS network, users tend to interact with other users of similar degrees, indicated by the 'squares' near the diagonal of its BOSAM. These show the formation of communities of nodes of similar degrees, with noticeably larger communities for nodes of a high degree. These node communities correspond to the patient communities on the PSORIASIS forum, where patients in each community have been diagnosed with a different type of psoriasis. The CROHN forum, also a single-topic forum being run by a community, shows a mixture of the above two characteristics.

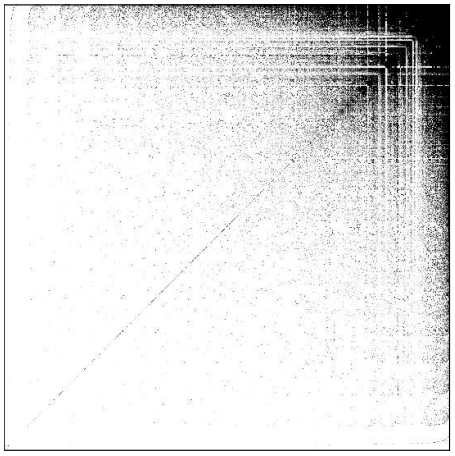
The HBOARDS, INSPIRE and PINFO forums, being large commercial forums, have a much richer structure (Figure B.1). The lines radiating from the upper right corners of their bitmaps are a feature specific to social networks [172]. The HBOARDS and INSPIRE bitmaps exhibit regions of dense shading which correspond to posts from the site administrators. This is notable since all of these forums are multi-topic, and therefore, would normally divide into communities with little interaction between them. However, particularly in the case of INSPIRE, administrators routinely cross-post across the threads and also, to some extent, allow cross-posting by users in the form of journals. This tends to obliterate the communities structure. Part of such structure can still be observed in the BOSAM of the PINFO network, which suggests that, in comparison, this forum is more lightly managed.



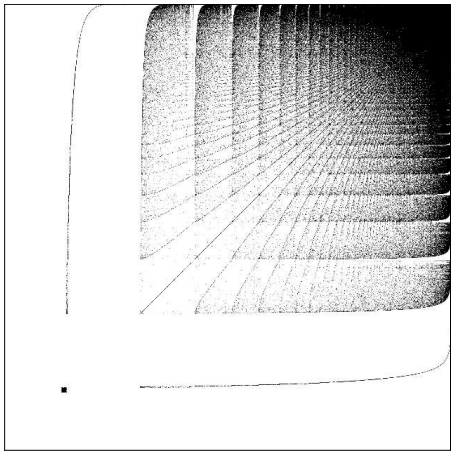
(a) HDA



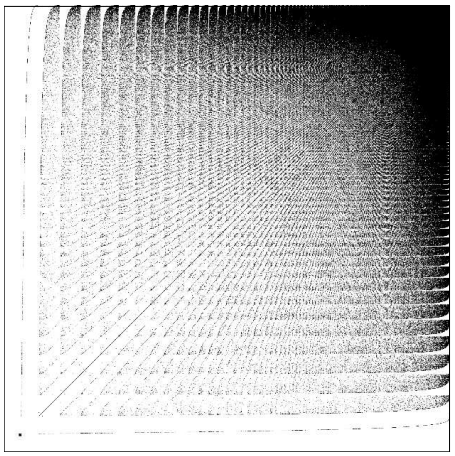
(b) PSORIASIS



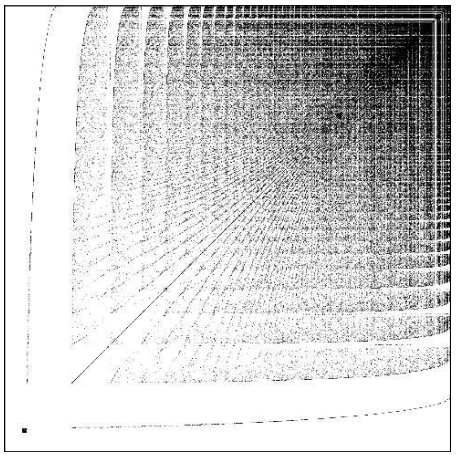
(c) CROHN



(d) HBOARDS



(e) INSPIRE



(f) PINFO

Figure B.1: BOSAMs of the modelled networks

Appendix C

Proof of Theorem 1 in Section 5.3.1

Proof. Let x and y be two strings with alternative substrings, represented by graphs G_x and G_y, respectively. We denote the set of all non-empty local alignments, i.e., paths, ending at node $(i, j) \in G_{xy}$, with the labels of nodes i and j being the last aligned characters, by $A_{i,j}$, where G_{xy} is the strong product graph of G_x and G_y:

$$A_{i,j} = \bigcup_{n \geq 1} \{(X, Y) \in R_x^n \times R_y^n | V(X_{2n-1}) = \{i\} \text{ and } V(Y_{2n-1}) = \{j\}\} \tag{C.1}$$

For $i = \phi$ or $j = \phi$, let

$$\begin{aligned} M(i, \phi) = M(\phi, j) = 0 \\ N(i, \phi) = N(\phi, j) = 0 \end{aligned} \tag{C.2}$$

where ϕ is the start character. First, we show that the following holds.

$$\begin{aligned} M(i, j) &= \sum_{a \in A_{i,j}} \exp(\beta S(a)) \\ N(i, j) &= \sum_{m \preceq i, n \preceq j, a \in A_{m,n}} \exp(\beta S(a) + \beta g(|mi|) + \beta g(|nj|)) \end{aligned} \tag{C.3}$$

where $|mi|$ denotes the length of the path from m to i and $\preceq$ is the partial order specified by G_x or G_y, as appropriate. We prove the above by induction on nodes

(i,j) of G_{xy}. For $M(i,j)$, with $(i \neq \phi, j \neq \phi)$, we have

$$\begin{aligned}
&\sum_{a \in A_{i,j}} \exp(\beta S(a)) = \exp(\beta s(i,j)) \\
&+ \exp(\beta s(i,j)) \left(\sum_{\substack{m \preceq i', n \preceq j', a \in A_{m,n} \\ i'i \in E(G_x), j'j \in E(G_y)}} \exp(\beta S(a) + \beta g(|mi'|) + \beta g(|nj'|)) \right) \\
&= \exp(\beta s(i,j)) \left[1 + \sum_{\substack{m,n \\ mi \in E(G_x), nj \in E(G_y)}} N(m,n) \right] \\
&= M(i,j)
\end{aligned} \tag{C.4}$$

where the first equality is obtained using the definition of local alignment score, the second equality holds for nodes before (i,j) by induction and the last equality is the definition of $M(i,j)$ in Equation 5.8. Similarly, for $N(i,j)$, with $(i \neq \phi, j \neq \phi)$, we have

$$\begin{aligned}
&\sum_{m \preceq i, n \preceq j, a \in A_{m,n}} \exp(\beta S(a) + \beta g(|mi|) + \beta g(|nj|)) = \\
&\sum_{\substack{m \preceq i, n \preceq j', a \in A_{m,n} \\ j'j \in E(G_y)}} \exp(\beta S(a) + \beta g(|mi|) + \beta g(|nj'|) + \beta g) \\
&+ \sum_{\substack{m \preceq i', n \preceq j, a \in A_{m,n} \\ i'i \in E(G_x)}} \exp(\beta S(a) + \beta g(|mi'|) + \beta g(|nj|) + \beta g) \\
&- \sum_{\substack{m \preceq i', n \preceq j', a \in A_{m,n} \\ i'i \in E(G_x), j'j \in E(G_y)}} \exp(\beta S(a) + \beta g(|mi'|) + \beta g(|nj'|) + 2\beta g) \\
&+ \sum_{a \in A_{i,j}} \exp(\beta S(a)) \\
&= \exp(\beta g) \sum_{\substack{m \\ mi \in E(G_x)}} N(m,j) + \exp(\beta g) \sum_{\substack{n \\ nj \in E(G_y)}} N(i,n) \\
&- \exp(2\beta g) \sum_{\substack{m,n \\ mi \in E(G_x), nj \in E(G_y)}} N(m,n) + M(i,j) \\
&= N(i,j)
\end{aligned} \tag{C.5}$$

where the second equality is obtained as follow: since in the ground case, values of both N and M are 0, it is possible to calculate the value of N for each node (m,n) before (i,j), where either $m \preceq i$ and $n \prec j$ or $m \prec i$ and $n \preceq j$. This gives us the first and second terms. The third term holds for nodes before (i,j) by induction, and the fourth term is the definition of $M(i,j)$ in Equation C.3

proven above. The last equality is the definition of $N(i,j)$ in Equation 5.8. Now, in order to prove the theorem, we have

$$\begin{aligned}\sum_{n\geq 0}\sum_{X\in R_x^n, Y\in R_y^n}\exp(\beta S(X,Y)) &= 1+\sum_{n\geq 1}\sum_{X\in R_x^n, Y\in R_y^n}\exp(\beta S(X,Y)) \\ &= 1+\sum_{(i,j)\in V(G_{xy})}\sum_{a\in A_{i,j}}\exp(\beta S(a)) \\ &= 1+\sum_{(i,j)\in V(G_{xy})} M(i,j)\end{aligned} \tag{C.6}$$

where 1 is for the empty alignment. The second equality is obtained using the definition of $A_{i,j}$ and the last equality is obtained using the definition of $M(i,j)$ in Equation C.3. ■