Deep Learning Approaches for an Integrated Study of DNA Sequence, Epigenome and Chromatin Architecture

Ron Schwessinger

Lincoln College
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Hilary 2021

Abstract

In the era of genome sequencing, it has become clear that interpreting sequence variation in the non-coding genome represents a critical bottleneck for translating genomic insights to benefits for human health. The non-coding genome harbours the complex sequence patterns that encode regulatory elements and their intricate interplay over large genomic distances, ensuring precise gene regulation throughout time and cell differentiation. This complex sequence grammar is challenging for humans to comprehend and the experiments required to probe it are expensive and time-consuming. Machine learning approaches offer a powerful tool for studying the non-coding genome *in silico*.

In this thesis work, we explored the use of deep neural networks for the purpose of interpreting genomic variations in the non-coding genome. We re-implemented, optimised and evaluated proposed convolutional neural networks models for predicting chromatin features from DNA sequence and utilise them in an integrated pipeline for dissecting non-coding sequence variants identified by Genome-Wide Association Studies (GWAS). We applied this pipeline to variants associated with red blood cell traits, demonstrating how neural networks can facilitate the interpretation of loss and gain of function variants in regulatory elements.

Although chromatin feature networks evaluate variation in larger sequence context than previously possible, they still only offer a highly local perspective ignoring chromatin architecture and the distal interplay of regulatory elements. To address this limitation, we developed a novel neural network framework (deepC) using dilated convolutions and transfer learning to predict chromatin interactions from megabase scale DNA sequence. We validated deepC predictions using chromosome conformation assays at high sensitivity. Finally, we demonstrated how deepC can be used to dissect the sequence determinants of chromatin folding and to predict the impact of structural and single nucleotide variation on genome architecture.

Together this thesis evaluates, utilises and expands the deep learning toolbox for genomics and our capabilities of interpreting non-coding genomic variation.

Deep Learning Approaches for an Integrated Study of DNA Sequence, Epigenome and Chromatin Architecture

Ron Schwessinger

Lincoln College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2021

Acknowledgements

Personal

I like to thank Prof. Jim Hughes for his outstanding supervision and mentorship, for embracing the world of Deep Learning with me and for believing in me from day one of my internship. I thank Prof. Gerton Lunter for excellent supervision and the right ideas to overcome all the bottlenecks along the way. I also thank Prof. Douglas Higgs for his continual support and advice throughout my academic progression and Prof. Yee Whye Teh for his support, feedback and ideas concerning the neural network frameworks. I like to thank all members of the Hughes and Lunter group for providing the stimulating and supportive environment that made my doctoral studies such a pleasant experience. Explicitly, I want to thank Dr. Damien Downes for promoting me from colleague to friend and his pivotal role in translating my predictive models into practice and productive collaborations; and Matthew Gosden for all his help, his friendly ear and telling me when bead pellets looked dry. I thank Ewan Mac Mahon without who's dedication in supporting the GPU computational environment this work would not have been possible. I like to thank my thesis committee, Prof. Thomas Milne and Prof. Aleksandr Sahakyan, for their valuable feedback throughout this thesis work.
I am grateful to my family for all their support over the years and for creating the environment where following scientific curiosity is the most natural thing in the world. I also thank Ron and Lukas for always keeping me grounded and the community of "who cares" for keeping me sane during the national lockdowns.
Most importantly, I want to thank my partner Lea for taking the leap to Oxford with me and for tolerating my bioinformatician's lifestyle.

Institutional

I am grateful for being supported by the Wellcome Trust DPhil studentship in Genomic Medicine and Statistics (203728/Z/16/Z). I also like to thank the Wellcome Centre for Human Genetics and the MRC Weatherall Institute of Molecular Medicine, specifically the MRC Molecular Haematology Unit and the Centre for Computational Biology for providing with the infrastructure and the stimulating environment for conducting this thesis work.

Clarification on Contributions

The developed methods and analysis in this thesis represent my original work. Due to the highly collaborative nature of the overarching work, I chose to write this thesis in "we" form. Contributions of colleagues are outlined explicitly in the "Acknowledgements" sections of each "Results" chapter.

Abstract

In the era of genome sequencing, it has become clear that interpreting sequence variation in the non-coding genome represents a critical bottleneck for translating genomic insights to benefits for human health. The non-coding genome harbours the complex sequence patterns that encode regulatory elements and their intricate interplay over large genomic distances, ensuring precise gene regulation throughout time and cell differentiation. This complex sequence grammar is challenging for humans to comprehend and the experiments required to probe it are expensive and time-consuming. Machine learning approaches offer a powerful tool for studying the non-coding genome *in silico*.

In this thesis work, we explored the use of deep neural networks for the purpose of interpreting genomic variations in the non-coding genome. We re-implemented, optimised and evaluated proposed convolutional neural networks models for predicting chromatin features from DNA sequence and utilise them in an integrated pipeline for dissecting non-coding sequence variants identified by Genome-Wide Association Studies (GWAS). We applied this pipeline to variants associated with red blood cell traits, demonstrating how neural networks can facilitate the interpretation of loss and gain of function variants in regulatory elements.

Although chromatin feature networks evaluate variation in larger sequence context than previously possible, they still only offer a highly local perspective ignoring chromatin architecture and the distal interplay of regulatory elements. To address this limitation, we developed a novel neural network framework (deepC) using dilated convolutions and transfer learning to predict chromatin interactions from megabase scale DNA sequence. We validated deepC predictions using chromosome conformation assays at high sensitivity. Finally, we demonstrated how deepC can be used to dissect the sequence determinants of chromatin folding and to predict the impact of structural and single nucleotide variation on genome architecture.

Together this thesis evaluates, utilises and expands the deep learning toolbox for genomics and our capabilities of interpreting non-coding genomic variation.

Contents

List of Figures **ix**

List of Abbreviations **xii**

1 Introduction **1**
1.1 Transcriptional regulation in the three-dimensional genome 1
1.1.1 Basic building blocks of human transcriptional regulation . . 1
1.1.2 Domains of chromatin interactions 3
1.1.3 Identifying domains and boundary elements 5
1.1.4 Models of chromatin domain formation 8
1.2 Deep learning in genomics and gene regulation 9
1.2.1 Fundamentals of machine learning in genomics 9
1.2.2 Deep neural networks in genomics 13
1.3 Thesis aims 20

2 Deep learning for chromatin feature prediction in the context of GWAS variant interpretation **22**
2.1 Introduction 23
2.2 Results 28
2.2.1 Re-implementing existing deep learning approaches for chromatin feature prediction 28
2.2.1.1 Existing network architecture 28
2.2.1.2 Reimplementation in Tensorflow 29
2.2.1.3 Dilated networks for high resolution context aggregation 33
2.2.2 Chromatin feature networks in a pipeline for dissecting GWAS variants 35
2.2.2.1 An integrated pipeline for dissecting genomic variation 35
2.2.2.2 Predicting the effect of variation with deep learning 37
2.2.2.3 Evaluating deep learning-based chromatin predictions 38
2.2.2.4 Prioritising GWAS SNPs for functional follow-up . 43
2.2.3 Data compendium scale 49

2.3 Discussion . . . 54
2.4 Data & Code Availability . . . 59
2.5 Acknowledgments . . . 60

3 Developing deep learning models for chromatin interaction predictions 61
3.1 Introduction . . . 62
3.2 Results . . . 65
3.2.1 Building deep neural networks for predicting chromatin interactions at mega-base scale . . . 65
3.2.1.1 Design considerations . . . 65
3.2.1.2 Hi-C data encoding . . . 67
3.2.1.3 Model architecture . . . 67
3.2.1.4 Transfer learning . . . 70
3.2.1.5 Regression of distance normalised percentiles . . . 72
3.2.1.6 Predictions in different settings . . . 75
3.2.1.7 Evaluation . . . 76
3.2.1.8 Tissue-specificity . . . 80
3.2.2 Validating computational predictions of chromatin architecture 84
3.2.2.1 Capture-C based methods for validation . . . 85
3.2.2.2 Correlation of virtual4C profiles . . . 88
3.2.2.3 Validating domain boundaries . . . 89
3.2.3 Comparison of sequence-based prediction with chromatin feature based predictions . . . 94
3.3 Discussion . . . 100
3.4 Data & Code Availability . . . 104
3.5 Acknowledgments . . . 104

4 Investigating the regulatory grammar of 3D chromatin interactions 105
4.1 Introduction . . . 105
4.2 Results . . . 108
4.2.1 Mapping sequence determinants at base pair resolution . . . 108
4.2.2 Variant effect prediction . . . 112
4.2.2.1 Predicting the effect of a validated boundary deletion 113
4.2.2.2 Predicting the impact of single base pair variants . 115
4.2.3 Predicting the relative contribution of regulatory element classes to 3D genome structure . . . 120
4.2.4 Dissecting the regulatory grammar of chromatin boundaries 123
4.3 Discussion . . . 131
4.4 Data & Code Availability . . . 136
4.5 Acknowledgments . . . 136

5 Conclusion & Perspectives **137**
5.1 Chromatin features, the non-coding genome and GWAS 138
5.2 3D chromatin architecture predictions 141
5.3 Towards integrated models of transcriptional regulation 142
5.4 Interpretability 143
5.5 Conclusion 146

6 Methods **148**
6.1 Chromatin feature networks 149
6.1.1 Data pre-processing 149
6.1.1.1 Chromatin feature data 149
6.1.1.2 Pre-processing and data encoding 150
6.1.2 Design & architecture 151
6.1.2.1 Chromatin feature prediction 151
6.1.2.2 Re-implementing existing architectures 151
6.1.2.3 Dilated CNNs 151
6.1.2.4 Optimised CNN architecture - deepHaem 152
6.1.3 Training & evaluation 152
6.1.4 Application 153
6.1.4.1 Running window predictions 153
6.1.4.2 SNP effect predictions deepHaem 153
6.1.4.3 Union open chromatin set 153
6.1.4.4 Comparison to Sasquatch 153
6.1.4.5 Read coverage vs. probability scores 154
6.2 Chromatin interaction networks 154
6.2.1 Data pre-processing 154
6.2.1.1 Hi-C data 154
6.2.1.2 Hi-C data encoding and distance normalisation . . 154
6.2.2 Design & architecture 155
6.2.2.1 Predicting chromatin interactions 155
6.2.2.2 Transfer learning and architecture – deepC 155
6.2.3 Training & evaluation 157
6.2.3.1 Training deepC networks 157
6.2.3.2 Distance-stratified correlation 158
6.2.3.3 Comparison against HiC-Reg 158
6.2.3.4 Re-aligning and downsampling Hi-C data 159
6.2.4 Validation 159
6.2.4.1 Capture-C design 159
6.2.4.2 Cell culture and formaldehyde fixation 159

6.2.4.3 3C library preparation . . . 160
6.2.4.4 NG Capture-C . . . 160
6.2.4.5 Tiled-C . . . 160
6.2.4.6 Data analysis . . . 161
6.2.4.7 Distance-normalised NG Capture-C tracks . . . 161
6.2.5 Application . . . 162
6.2.5.1 Predicting changes in chromatin interactions . . . 162
6.2.5.2 Boundary calling . . . 162
6.2.5.3 Distance-normalised NG Capture-C signal over boundaries . . . 163
6.2.5.4 Virtual4C . . . 163
6.2.5.5 Saliency score . . . 163
6.2.5.6 eQTL data analysis . . . 163
6.2.5.7 Deletion screen . . . 164
6.2.5.8 CTCF boundary potential screen . . . 164
6.2.6 Auxiliary . . . 165
6.2.6.1 Additional chromatin feature data . . . 165
6.2.6.2 Chromatin segmentation . . . 165
6.2.6.3 5C . . . 166
6.2.6.4 SNP sampling . . . 166
6.3 Software & Data . . . 166
6.3.1 Neural networks . . . 166
6.3.2 Additional software . . . 167
6.3.2.1 Tools . . . 167
6.3.2.2 Additional R packages . . . 167
6.3.2.3 Additional Python libraries . . . 167
6.3.3 Data availability . . . 168
6.3.4 Code availability . . . 168

References **169**

List of Figures

1.1 Overview of chromatin architecture 6
1.2 Formation of chromatin domains by loop extrusion 9
1.3 Basic units of neural networks and principles of non-linear feature representations 16
1.4 Schematic of a convolutional neural network for predicting chromatin features from DNA sequence 18

2.1 Evaluation of all 936 deepHaem classifiers 31
2.2 Precision recall curve of erythroid related chromatin features of interest 32
2.3 ATAC-seq signal and deepHaem predictions over HBA enhancer region 33
2.4 Training and test loss comparison for reproduction and dilated versions of Basset 36
2.5 Training and test loss of reproduction and dilated deepSEA versions 37
2.6 Pair-wise Pearson correlation between SNP predictions with erythroid specific deepHaem classifiers 39
2.7 Comparison of deep learning based and DNase footprinting based damage predictions 41
2.8 Correlation of deepHaem probability scores with ATAC-seq and ChIP-seq coverage 43
2.9 Enrichment of deepHaem damage scores in open chromatin intersecting SNPs 44
2.10 DeepHaem damage predictions for ~ 26k variants associated with red blood cell phenotypes or in high linkage with lead SNPs 45
2.11 DeepHaem predictions for rs10758656 and the Vanuatu SNV and variants in high respective linkage across classifiers of erythroid differentiation 46
2.12 Intersecting deepHaem prioritised SNPs with open chromatin peaks from three donors 48
2.13 Prioritised regulatory variants in genomic context across three donors 51
2.14 PR curves for erythroid classifiers of interest trained using a ~ 4500 feature dataset 52

2.15 Comparing data selection performance. PRC AUCs were computed for 20 erythroid related classifiers of interest 53

3.1 Encoding Hi-C interactions as for deep learning. Chromatin interactions over a ~ 1 Mb sized windows in bins are encoded a vertical zig-zag pole over the center 68
3.2 Overview of the deepC architecture and training workflow 69
3.3 First deepC attempts encoding Hi-C frequencies as classification problem and predicting class identities without distance normalisation 71
3.4 Distance stratified, pyramid percentile normalisation 73
3.5 Prediction with transfer learning, distance normalisation and regression approach on hold out chromosome 17 in GM12878 cells 74
3.6 Predictions using log observed over expected ratio encoding 75
3.7 Prediction at 10 kb res on six human cell line. Hi-C data with different sequencing depth 77
3.8 DeepC predictions of mouse ES cell Hi-C data on hold out chromosome 11 77
3.9 Cross-validation across all chromosomes in GM12878 and K562 . . . 79
3.10 DeepC 10 kb models GM12878 using downsampled Hi-C data . . . 81
3.11 Separately trained GM12878 and K562 models predict tissue-specific chromatin interactions at the *MEIS1* locus on chr2 82
3.12 Separately trained GM12878 and K562 models predict tissue-specific chromatin interactions at the *CMIP* locus on chr16 83
3.13 A jointly trained GM12878 and K562 model fails to capture tissue-specific patterns 85
3.14 Validating deepC predictions with Capture-C in GM12878 and K562 cells 87
3.15 Virtual 4C profiles from GM12878 Hi-C skeleton and deepC predictions 88
3.16 Pearson correlation of the distance normalised Capture-C profiles with virtual4C profiles from skeleton and prediction 89
3.17 Insulation score-based boundary call in GM12878 91
3.18 Insulation score-based boundary call in K562. 92
3.19 Meta-profiles of distance normalised Capture-C interaction over boundaries called using the insulation score 93
3.20 Validating distance normalised Capture-C interaction over boundaries called with TopDom 94
3.21 Comparing boundary calls on Hi-C data and deepC predictions . . 95
3.22 Comparing deepC and HiC-Reg predictions on hold out chromosome 17 in GM12878 97

3.23 Comparing deepC and HiC-Reg predictions on hold out chromosome 17 in K562 98
3.24 Distance stratified Pearson correlation between HiC-Reg and deepC predictions and the Hi-C data 99

4.1 Comparison of deepC saliency score vs. in silico mutation of variant impact over CTCF site core motif using the GM12878 model 110
4.2 Saliency pattern over the *IKZF2* locus with promoter and CTCF site 111
4.3 Genome wide meta profiles of the saliency score over accessible chromatin and CTCF ChIP-seq peaks relative to the peak centre . . 112
4.4 Saliency score of GM12878 specific eQTLs within accessible chromatin and CTCF sites vs. shuffled background SNPs 113
4.5 Effect of deleting a 30 kb fragment harbouring four CTCF sites in HEK-293T cells 114
4.7 CD4+ T-cell chromatin marks and 4C experiments in allelic contexts of asthma-associated SNP 117
4.8 Nonrisk and risk allele SNP effect predictions on chromatin interactions in GM12878. 118
4.9 Predicting impact of single asthma-associated SNPs (A-C) on chromatin interactions 120
4.10 Predicting the impact of both asthma-associated SNPs in three different tissue models 121
4.11 Exemplary *in silico* CTCF site deletion of a 500 bp fragment on chr17 122
4.12 Boxplot summary of genome-wide deletion screen with chromatin segmentation stratification 123
4.13 CTCF insertion prediction in different loci 126
4.14 Characterising the boundary potential of CTCF sites on chromosome 16 127
4.15 Boundary strength vs. CTCF cell-type-specificity, ChIP-seq peak strength and motifs 129
4.16 Predicting the effect of inserting promoters and enhancers into a locus on chr17 130

6.1 Convolutional network architecture employed 157

List of Abbreviations

AUC . . . . . Area under curve

BE . . . . . Boundary element

CMP . . . . . Common myeloid progenitor

ChIP-seq . . . Chromatin immunoprecipitation and sequencing

ChIA-PET . . Chromatin interaction analysis by paired-end tag sequencing

CTCF . . . . . CCCTC-binding factor

DHS . . . . . DNase hypersensitive site

DNA . . . . . Deoxyribonucleic acid

DNN . . . . . Deep neural network

CNN . . . . . Convolutional neural network

eQTL . . . . . Expression quantitative trait loci

Ery . . . . . Late erythrocyte progenitor

ES . . . . . Embryonic stem cell

FC . . . . . Fully connected (layer)

GWAS . . . . . Genome wide association study

HSC . . . . . Haematopoietic progenitor

H3K27ac . . . Histone modification - acetylation of lysine 27 on Histone 3

H3K4me1/2/3 Histone modification - mono-, di- or trimethylation of lysine 4 on Histone 3

IQR . . . . . Inter quartile range

KS . . . . . Komolgorov-Smirnoff

LAD . . . . . Lamina-associated domain

LD . . . . . Linkage disequilibrium

LEF . . . . . Lop extruding factor

MEP . . . . . Megakaryocyte/erythrocyte progenitors

ML . . . . . Machine learning

MPP . . . . . Multi-potent progenitor

NBD . . . . . Nucleotide binding domain

PIC . . . . . Pre-initiation complex

PR . . . . . Precison-recall

ReLU . . . . . Rectified linear units

RNA . . . . . Ribonucleic acid

ROC . . . . . Receiver operator characteristic

SNP . . . . . Single nucleotide polymorphism

SV . . . . . Structural variant

TAD . . . . . Topolocially associated domains

TF . . . . . Transcription factor

TFBS . . . . . Transcription factor binding site

TSS . . . . . Transcription start site

WIMM . . . . Weatherall institute of molecular medicine

WT . . . . . Wildtype

ZF . . . . . Zinc Finger

1 Introduction

Contents

1.1 Transcriptional regulation in the three-dimensional genome 1
1.1.1 Basic building blocks of human transcriptional regulation 1
1.1.2 Domains of chromatin interactions 3
1.1.3 Identifying domains and boundary elements 5
1.1.4 Models of chromatin domain formation 8
1.2 Deep learning in genomics and gene regulation 9
1.2.1 Fundamentals of machine learning in genomics 9
1.2.2 Deep neural networks in genomics 13
1.3 Thesis aims 20

1.1 Transcriptional regulation in the three-dimensional genome

1.1.1 Basic building blocks of human transcriptional regulation

The human genome contains around 22,000 protein-coding genes (Pertea and Salzberg 2010). House-keeping genes are expressed ubiquitously while the expression of others is tightly regulated across different cell types and along differentiation trajectories. Precise gene regulation is accomplished by a complex framework of transcription factors (TF) that regulate gene expression by modifying the rate

at which protein coding DNA is transcribed into RNA. Protein coding genes are transcribed by RNA polymerase II, which is recruited to the DNA sequence at the transcription start site (TSS) by the pre-initiation complex (PIC). The PIC is a multi-protein complex consisting of the basic transcription factors binding the DNA at the basal promoter. The basal or core promoter is defined as the shortest sequence that allows the recruitment of RNA polymerase II and transcription initiation[1]. The factors assembled at the basal promoter are only able to drive low level transcription without cell type specificity. The rate and specificity of transcription is regulated by TFs binding either directly upstream of the core promoter or at distal cis-regulatory elements. Collectively TFs act by a combination of molecular mechanisms: 1) They open up the chromatin by initiating nucleosomal remodelling, making the DNA accessible to other TFs. 2) By recruiting histone modifying enzymes and 3) by recruiting co-activator complexes such as Mediator, which in turn interact with the basal transcriptional machinery[2, 3]. Transcription factors bind the DNA at specific sites, TF binding sites (TFBS) recognised by their DNA binding domains. These binding preferences are usually summarised in TF binding motifs which encode the frequency with which bases have been observed relative to one another in genome-wide binding sites of a single factor. The exact DNA sequence preferences are characterised for less than 700 of the 2,000 – 3,000 known human TFs[4, 5]. Furthermore, our understanding of TF binding preferences is restricted to a limited number of cell types.

Cis-regulatory elements are primarily classified based on their location relative to the gene they regulate. Promoters are comprised of the core promoter region and usually contain binding sites for additional TFs immediately upstream about 100 – 200 bp from the TSS[6]. Transcription enhancing elements located distal to the TSS on the same chromosome are classified as enhancers. Enhancers can be located megabases away from the gene promoter they regulate[7]. They share many commonalities with promoters such as local structure characterised by accessible chromatin, regulatory activity associated histone marks and TF binding as well

as RNA polymerase II occupancy and bidirectional transcription[6]. Enhancers harbour dense clusters of TFBS many of which are cell-type specific. Genes can be controlled by multiple enhancers and their tissue-specificity enables a complex spatiotemporal regulation of gene expression[3]. Insulators or boundary elements represent the third class of regulatory elements. Rather than directly interacting with the transcriptional machinery, they regulate the three-dimensional chromatin architecture that in turn affects gene expression. A comprehensive overview of gene regulation in the context of chromatin architecture and the role of boundary elements is provided in section 1.1.3. Cis-regulatory elements are associated with characteristic chromatin features. Regions of accessible chromatin are associate with both promoters and enhancer. They are identified genome-wide by an increased sensitivity to DNase I[8] or the transposase TN5[9]. Regulatory elements are also marked by specific histone modifications, identified genome-wide by chromatin immunoprecipitation and sequencing[10, 11]. The promoters of actively transcribed genes and active enhancers are marked by acetylation of lysine 27 on Histone 3 (H3K27ac). Different levels of methylation of Histone 3 lysine 4 are observed at different ratios over promoters and enhancers. Promoters are enriched for di- and trimethylation over monomethylation (H3K4me2, H3K4me3 relative to H3K4me1) and enhancers show the opposite enrichments[12].

1.1.2 Domains of chromatin interactions

The human genome is structured into domains of increased chromatin interactions that restrict and shape the accessible chromatin environment for every genomic element. This context has fundamental implications for transcriptional regulation, by defining which enhancers can physically interact with a gene promoter at any given time. Domains are identified and characterised based on their property of enriched intra- over inter-domain interactions or insulation. The characterisation of chromatin interaction domains was sparked by advances in the resolution, throughput and scale of chromosome confirmation capture assays[13–15]. Insights gained from these cell population-based studies have been validated by means of super

resolution microscopy, single-cell Hi-C and GAM, that also distinguish between probabilistic pairwise contacts vs. multi-contacts in 3D[16–18]. Chromatin domains are hierarchically structured[13, 14, 19–21] and the scales attributed to them depend on the resolution of the assay used. Hi-C studies require binning of the identified chromatin interactions and are usually limited to 5 – 10 kb resolution with only exceptionally deep sequencing being able to push this resolution further down [14, 22]. Hi-C maps reveal that chromosomes are structured into topologically associated domains (TADs) along the diagonal, with high resolution Hi-C estimating them at a scale of 40 kb to 3 Mb[14]. TADs are themselves hierarchically structured into smaller domains, termed sub-TADs or meta-TADs[14, 19]. At very high resolution, Hi-C can also reveal "loops" between genomic bins.

At the scale of the whole genome, TADs are structured into compartments, characterised by increased contact frequencies between TADs in the same compartment. These compartments share epigenetic marks and thus likely similar functionality. Originally, compartments have been divided into A and B compartments, based on the first two principle components of chromatin interactions, or clustering of long-range interactions, and they are associated with repressed hetero- and active euchromatin respectively[13, 23, 24]. Clustering these compartments at higher resolution revealed sub-compartments within A and B, each associated with distinct epigenetic features, chiefly histone modifications, nuclear lamina association and replication timing[14]. Using a more targeted approach, other studies have directly probed chromatin loops between the architectural protein CCCTC-binding factor (CTCF) using Chromatin Interaction Analysis by Paired-End Tag Sequencing (ChIA-PET)[25, 26]. They defined insulated neighbourhoods as delimited by a CTCF homo-dimer and Cohesin bound and probed them at a scale of 25 to 940 kb and containing one to ten genes[25–27]. Because these neighbourhoods en-capsule the vast majority of enhancer-promoter interactions, they have been described as the core functional units of the genome[27]. Recent Capture-C advances, probing the interactome of promoters, enhancers or boundary elements at high

resolution, confirmed the view of insulated regulatory domains[15]. Interestingly, when measured at a population scale and at high resolution the insulation property of boundary elements appear graduate and interactions across. Boundary elements can be observed to varying degrees[15]. Importantly, chromatin structure is dynamic. While TADs are mainly conserved[14, 24], the fine architecture of regulatory domains can be dynamic between cell types[14, 25, 26, 28] and are thus likely to shape promoter enhancer interactions and drive cell type specific gene expression.

An interesting and important relation is the one between chromatin loops and domains. Loops are regarded as specific contacts between distinct elements. This entails enhancer-promoter contacts but also regards to boundary-boundary contacts, while, insulated neighbourhoods restrict the accessible chromatin and create a domain of elevated interaction frequencies. Contacts between distinct elements can be observed on top of there more general interaction domains[15, 29].

In summary, the description of chromatin domains, sub-domains and insulated neighbourhoods is fluid, depending on the scale and resolution of the data and the methodology used.

1.1.3 Identifying domains and boundary elements

Boundaries are characterised by a relative insulation of chromatin interactions, restricting contacts across the boundary and enriching for contacts within the domain between two adjacent boundaries. The regulatory functionality of boundaries derives from their insulation properties. By enriching for intra-domain interactions, boundaries can promote the regulatory interactions between specific enhancers and promoters.

TAD calling depends on the underlying chromatin interaction data. It's all-vs-all design makes Hi-C suitable for mapping chromatin domains and boundaries genome-wide and Hi-C data are frequently used to call compartments, TADs and

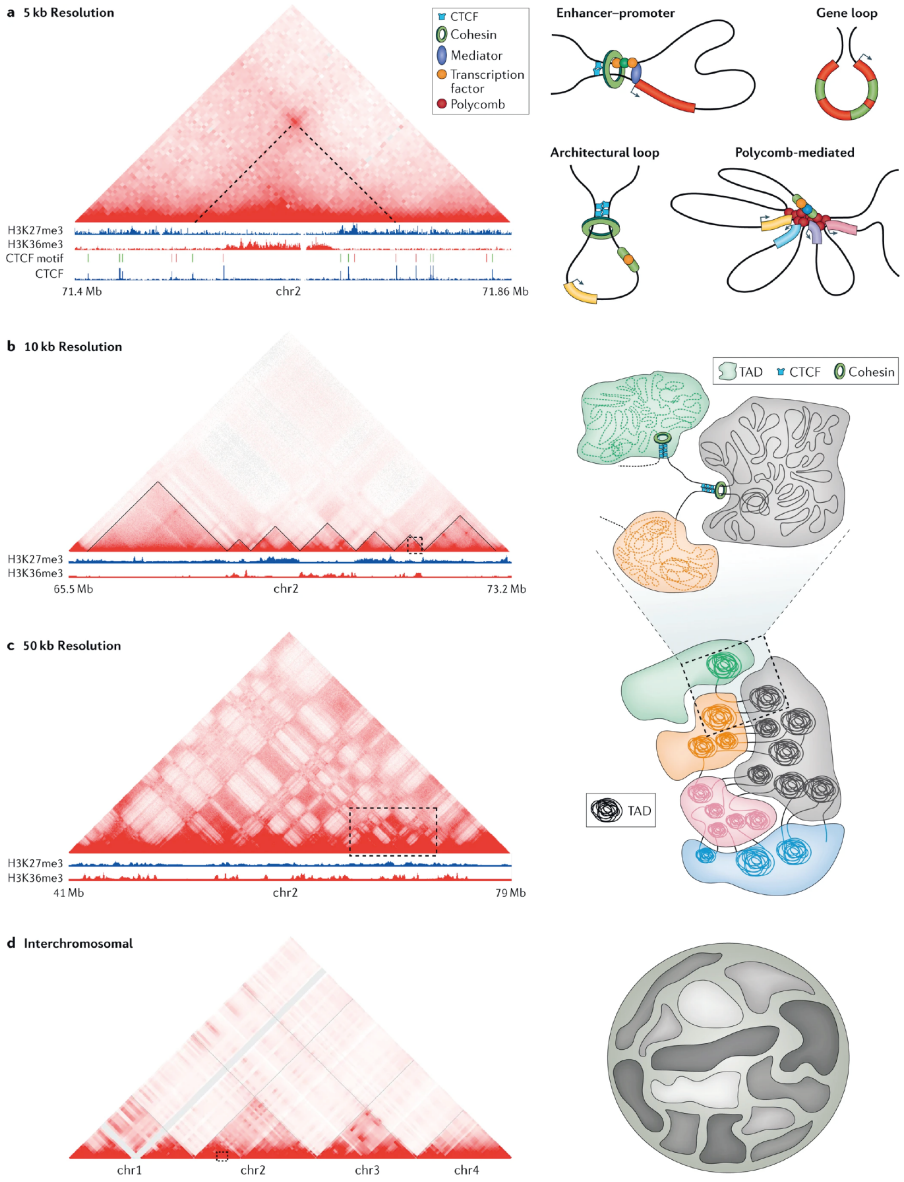

Figure 1.1: Overview of chromatin architecture. Chromatin architecture is organised hierarchically. Shown are high resolution Hi-C interactions in GM12878 cells alongside models displaying the proposed molecular mechanism leading to the observed Hi-C patterns. A) At high resolution individual chromatin contacts or loops can be observed as enriched interaction dots in the Hi-C map. These can for example represent individual enhancer-promoter contacts or may correspond to boundary-to-boundary interactions. B) Each chromosome is structured into domains of increased inter-domain chromatin interactions. These TADs can be observed at different scales and they are hierarchical in nature. C) TADs marked similarly transcriptional activities and consequently similar histone modification profile interact more frequently with each other than with other TADs. This leads to the characteristic chequer-board-pattern on Hi-C maps of whole chromosomes. D) Individual chromosomes occupy distinct territories in the nucleus. Figure adapted from Bonev & Cavalli 2016[21]

subTADs. Using principal component analysis of the measured contacts places genomic regions into the broad compartments A and B. Switching points between compartments[13] are boundaries and usually correspond to TAD boundaries. A plethora of algorithms for calling boundaries and TADs has been proposed, all relying on the same two principles: 1) Directionality or an unbalance of chromatin contacts up- and downstream of a genomic bin[14]. 2) Insulation, formulated as insulation score, which measure the interactions across a genomic bin. Local minima in such an insulation score profile correspond to boundaries[30]. TAD calling algorithms rely heavily on empiric parameter tuning, where the user decides the quality of the call based on visual inspection of the called domains. The performance of TAD callers varies over data resolution and normalisation techniques and no single algorithm performs best in all scenarios[31]. The resolution of Hi-C data presents another major limitation to TAD calling. Usually, boundaries can only be narrowed down to the 5 – 10 kb sized genomic at which Hi-C data can be visualised without being to sparse. Finding the precise location of boundaries elements thus requires fine-mapping using orthologous data, for example by CTCF ChIP-seq (Chromatin Immunoprecipitation and Sequencing). Most TAD callers have been designed for Hi-C data binned at larger intervals and usually perform worse at finer resolution[31].

Targeted capture such as Capture-C resolves the regulatory domains with higher resolution than all-vs-all approaches. In Capture-C profiles from a single viewpoint, boundaries can be identified as sharp drop-offs of interaction frequency levels[15]. However, the interaction signal rapidly decreases in cis and probing domain boundaries thus requires a large number of viewpoints. Additional layers of data, such as CTCF ChIP-seq are still required to precisely ascertain boundary locations and to determine their underlying nature.

The mapping of proteins associated with boundaries such as CTCF and Cohesin subunits via ChIP-seq, narrows down the search space but has poor specificity given the shear abundance of their binding sites in the genome. Additional information

such as CTCF motif orientation can also help in prioritising functional sites. Directly mapping boundary interactions via ChIA-PET may suffer from bias due to co-enrichment skew[32] acerbated by the large number of CTCF binding sites in the genome. Regardless which assay is used, boundary properties are estimated by imperfect genomic assays. Therefore, calling their locations and inferring their characteristics is restricted by the resolution and the inherent biases of the respective assay used.

1.1.4 Models of chromatin domain formation

The precise mechanisms of chromatin architecture formation are still poorly characterised. The model that recently received the most attention is the loop-extrusion model[33]. According to this model, Cohesin is loaded on to the DNA and extruding the chromatin strands progressively until it is stalled or arrested mainly at boundary elements. CTCF sites are proposed to block loop extrusion in an orientation dependent manner, reflected in the directionality of the CTCF core binding motif. Interestingly it was also proposed that stalling may not be limited to boundary elements but can be caused by transcription and possibly promoter enhancer interactions themselves[34].

Polymer simulations based on the loop extrusion model can replicate patterns seen in Hi-C data and can predict the impact of genome editing experiments targeting boundaries[35, 36]. Loop extrusion can explain several properties of chromatin domains and TADs[36] namely: the hierarchical structure of TADs, subTADs and finer compartments; the preference of convergent CTCF binding sites for loop formation and the absence of interactions between distal and trans located boundary elements. By varying model parameters such as CTCF binding frequency, Cohesin loading frequency and processivity, loop extrusion models can reproduce all of the chromatin interaction patterns observed in Hi-C data, such as domains, loops and stripes[36]. It can also account for imperfect boundaries at a population scale by inter cell variability in CTCF binding or imperfect insulation

properties. *In vitro* studies have since then demonstrated that Cohesin extrudes the human DNA polymer into loops[37].

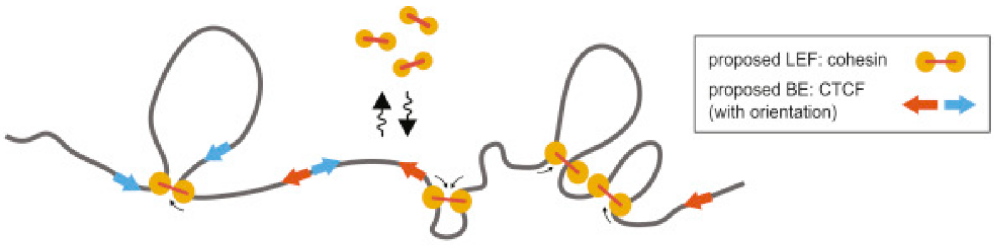

Figure 1.2: Formation of chromatin domains by loop extrusion. Cohesin as the loop extruding factor (LEF) is loaded on to and off the DNA polymer. DNA is extruded through Cohesin to form loops. Chromatin interactions within these loops are enriched over interactions with region outside the loop. The DNA extrusion by CTCF bound sites (boundary elements – BE) in an orientation dependent manner. Figure adapted from Fudenberg et al. 2016[36]

1.2 Deep learning in genomics and gene regulation

1.2.1 Fundamentals of machine learning in genomics

The field of functional genomics studies genomic elements, their function and the molecular mechanisms and sequence determinants that encode them. The common molecular assays: genome and RNA sequencing, epigenetic and chromosome confirmation assays are producing massive amounts of data whose analytical integration is challenging. The DNA sequence is the key template for most genomics research, and the properties or behaviour of the genome detected in genomics assays can be thought of as emergent properties of that sequence. The sequence harbours a complex grammar of protein factor binding motifs and patterns that influence chemical modifications and the physical properties of DNA as a chromatin polymer. Making this grammar comprehensible to researchers is difficult because it is often cell-type specific, convoluted and spans large contexts from thousands to millions of bases. Together, the data quantity and complexity made machine learning (ML) a key tool for functional genomics research. ML encompasses a large field of analytical methods. The key characteristic of ML algorithms is that they

are designed to learn complex patterns and relationships in data to solve defined tasks mainly classification, regression or clustering. Rather than being encoded in a hardwired manner by the researchers, ML methods use a training procedure to adaptively "learn" the patterns and relationships that solve a given task optimally.

ML methods are extensively used in genomics. Examples include but are not limited to:

- Motif discovery to identify preferred binding sites of TF[38]
- Identifying types of genomic elements such as enhancers and promoters and their transcriptional activity status from histone modification profiles[39, 40]
- Predicting gene expression from epigenetic signals[41]
- Predicting the pathogenicity of sequence variation[42]

ML applications in genomics utilise a range of algorithms, from comprehensible linear models to computationally intensive Support Vector Machines[43] and artificial neural networks (NN). The principles of ML in genomics and further applications have been excellently reviewed elsewhere[44]. For the purpose of this work, it is sufficient to revisit the core workflow of traditional machine learning applications.

The key feature of ML algorithms is their ability to learn from data. Data usually refers to a set of observations like images, genes or a set of genomic regions. Each observation is associated with a set of features. Features can be direct properties of the observations, such as pixel colour values or DNA sequence bases. Features may be summary statistics like mean colour value, gene length or GC content. Alternatively they can be derived by other combinations of the direct features such as TF binding motifs. Observations may also be associated with labels. Labels may indicate class membership, for example a dog or a promoter element, or they may represent a score such as the number of cars in an image or the expression value of a gene transcript. At its core, every ML algorithm is a function that takes the features

as input and computes a defined output. This output may be a class assignment for classification, a score for regression or a mapping in 2D space as in clustering. Every ML algorithm has a set of weights, also referred to as parameters. In a manner dependent on the ML algorithm the weights are applied to the features so they can be scaled, selected and summarised. Crucially, the weights are optimised in a training procedure. Given a training dataset the weights are altered to yield the output that optimally solves a defined task, as measured by a performance function on the objective. For a classification task this can be the average accuracy of the class assignment per observation, while for a regression tasks the performance often measures the error between the predicted and the observed score. In clustering tasks labels are usually not available or considered. Rather, the objective is to find structure in the data and the performance may be measured by the degree of variation explained.

The overarching aim of ML approaches is to find models that generalise well, meaning they perform well on unseen data. For example, a trained model might be applied to classify new images, predict the expression of genes or cluster new data based on learned criteria. The traditional ML workflow can be broken down into five steps:

1. **Data collection and curation:** Most effective machine learning methods depend on the availability of large number of observations for training and evaluation. Usually, data pre-processing is required in some form. This may include the assignment of labels and the removal of artefacts.

2. **Algorithm selection and refinement:** This involves precisely formulating the problem that the ML algorithm should try to solve and defining the objective function for optimisation. Next one or more ML algorithms need to be selected. The choice of algorithm will depend on the problem setting and the data available. Furthermore, ML algorithms come at difference capacities. The capacity describes how complex an algorithm is. Capacity is usually summarised by the number of weights in a model that may be

optimised, although other factors such as the encoded architecture contribute to an algorithm's capacity. Algorithms with a higher capacity can learn more complex relationships in data but are also harder to train and are more prone to overfitting and poor generalisation. Finding a suitable algorithm and fine tuning its architecture so as to balance capacity, training time and generalisation is a critical challenge in every ML project.

3. **Feature engineering and selection:** ML methods traditionally do not deal with raw data input. Rather, each observation is encoded as a set of features. These features may be summary statistics of the observation or describe a characteristic detail. ML methods can only be trained well if the extracted features encode the variation in the observation data. For example, although number of wheels, colour and size may be features describing cars, only colour would by useful to distinguish between black cabs and police vehicles. The process of finding representative features of a type of observations that allow the training of effective ML models is called feature engineering. Feature engineering might happen at the stage of data collection, for example housing market data may just be collected as summary statistics such as price and location. Frequently ML researchers are supplied with a number of raw data and pre-extracted features leaving it to them to find the most representative features and feature combinations to solve a task. In genomics, traditional machine learning approaches are poorly equipped to deal with long and complex inputs such as DNA sequence. In consequence, ML approaches to genomics have traditionally relied on feature engineering. Approaches to genomic feature engineering usually involve describing the DNA by general features such GC content or length, as a composition of k-mers[45, 46] or by defining types of binding sites[47].

4. **Training and Evaluation:** The chosen model is trained using observations from the training set. In multiple iterations, the models output on a set of observations is computed, the performance is calculated by comparing

to the known labels and the predicted ones and the weights of the model are updated to yield better predictions. This process continued until the performance plateaus. A crucial challenge in ML is to prevent the algorithm from overfitting to the training data, for example by learning to recognise individual observations and their associated labels from the training data, in other words learning by heart. To monitor overfitting, the training data are usually split into a training, test and validation set, for example by sampling 80 %, 10 % and 10 % of all observations into each respective group. The training set is used to train the model. In regular intervals, the model is evaluated on the test data set to compare the training and test set performance. Similar performance on both sets indicates good generalisation, while poor test set performance points to overfitting. Finally, the validation set is used as an additional test at the very end of the learning process. Since multiple iterations of training and testing under constant alterations to the model architecture are common, the validation set ensures the ML research is not overfitting the model for good performance on the test set.

5. **Application:** Once a model has reached a sufficient performance it can be utilised for is proposed application, such the prediction of classes or regression score of unseen observations. Models may be used to predict the effect of subtle, engineered alterations to observations, for example mutations in the genome. Alternatively, researchers might be more interested in dissecting the trained model itself, seeking to interpret which patterns and relationships a model has learned to infer hypotheses about the underlying data structure. Practically, the ML workflow does not follow a linear trajectory and frequent feedback and refinement of earlier steps is common to improve performance and efficiency.

1.2.2 Deep neural networks in genomics

In traditional ML workflows, feature engineering and selection is a critical bottleneck. Discovering the right combination and weighting of features regularly requires the

most expertise, labour and compute time. The key aim is to find a set of features that represent the observations in a latent space, where the variation in the features allows for optimal classifiers or regressors to be trained or to cluster the data efficiently. Finding effective feature representations for complex data is difficult. In the field of genomics specifically, it has proven challenging to incorporate large DNA sequence contexts and leading to rather crude summary descriptions such as GC-content or k-mer composition[45]. Deep neural networks (DNN), specifically convolutional neural networks (CNN), offer a remedy to this problem. Rather than working with predefined features, they excel at learning to recognise informative patterns in raw input data and to combine them optimally to accurate predictions. Learning the informative features is part of the training process itself leaving classical feature engineering less important or even obsolete. Where before ML researchers were tasked with finding a suitable representation of the observations, it is the algorithm that learns an optimal representation in the deep learning (DL) approach. However, this does not mean that DNNs require less manual fine tuning in the ML workflow, rather the researcher's efforts are shifted to finding a suitable neural network architecture and fine tuning its hyperparameters for optimal performance.

The concept of neural networks has been introduced decades ago[48, 49]. The core unit of neural networks is a single neuron (Figure 1.3 A). It takes multiple inputs and computes their weighted sum, adds a bias term and applies an activation function, usually a non-linear transformation. The weights for the weighted sum and the bias represent the parameters of the neuron that can be trained. Feed forward networks are layers of neurons (Figure 1.3 A). Multiple neurons form a single layer yielding the input for the next layer. Layers between the input and the layer producing the output are usually referred to as hidden layers. Combining neurons in parallel and over multiple layers allows networks to learn more complex relationships, the key being the non-linear transformations. Combining multiple linear functions yields linear output functions, but a combination of non-linear functions can model more complex relationships by transforming the feature representation space (Figure 1.3

B). Early variants, called perceptrons[49] used a step function as non-linearity. Later work by Rumelhart et al. replaced the activation function in multi-layer perceptrons by the sigmoid function. Opposed to the step function, the sigmoid function has a non-zero first order derivative at every point which allows training neural networks by gradient descent using the backpropagation algorithm[50]. For each iteration, a set of training observations is used to compute the models output and its error in a forward pass. Next, moving backwards through each layer, the contribution of each parameter to the error is calculated as the gradient of the error with respect to that parameter in the backwards pass. Thirdly, every parameter is adjusted by small steps along its gradient. This process is repeated for new sets of training observations until the error is sufficiently small. ReLUs (Figure 1 3, B) have been introduced as alternative non-linear activation function in 2011[51]. ReLUs represent a minimalistic non-linearity, outputting 0 for every negative input up to 0 and the inputs identity afterwards. They have the drawback of not being differentiable at 0 and the gradient being 0 for negative inputs. But they are much faster to compute compared to the sigmoid function. ReLUs have proven to work well for training deep neural networks and they are currently used for most practical work, although variations have been proposed that address the limitations of ReLUs[52, 53].
Feed forward networks have been applied in genomics[54] but it was the introduction of convolutional neural networks that sparked their wide-spread use in the field. CNNs have been designed and popularised for image analysis[55–58]. Rather than processing and weighting each input separately CNNs use filters, small matrices or tensors, to apply weights to a local neighbourhood of inputs, for example a number of adjacent pixels, summing them up at each position. The filter is moved over the input for example in 2D over an image, yielding a new 2D representation of the input image. Filters can efficiently detect highly local patterns, yielding a high score for positions in the image where a pattern is prominently present. The weights of these convolutional filters are parameters and are learned in the training procedure. CNNs usually employ a large number of filters over different layers, allowing them to detect a multitude of spatial patterns and combinations of them.

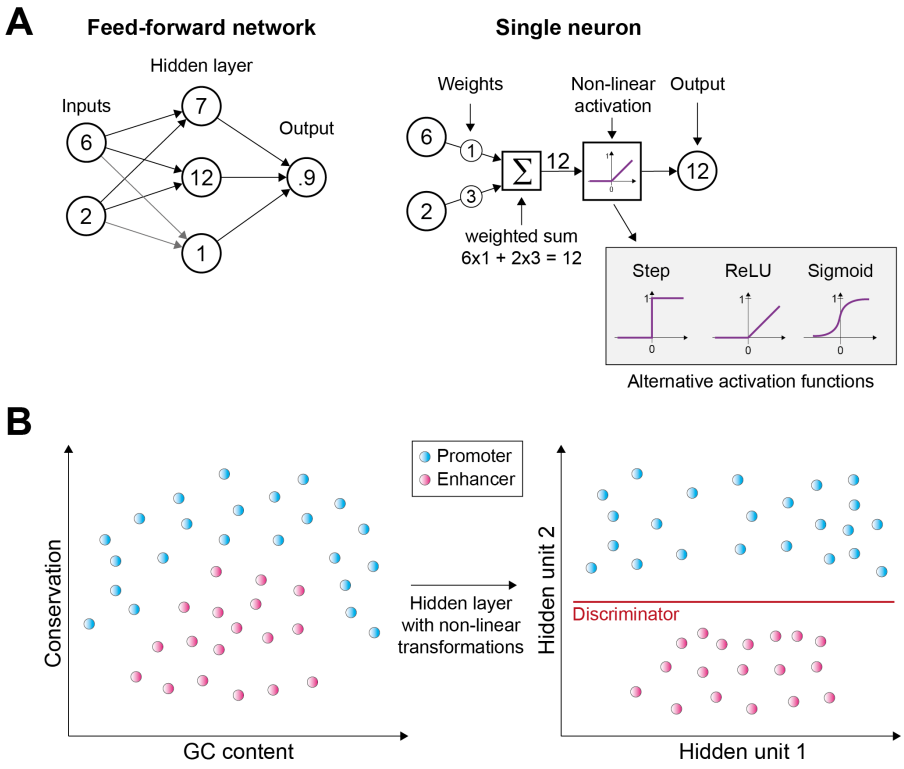

Figure 1.3: Basic units of neural networks and principles of non-linear feature representations. A) Feed-forward networks are a composition of single neurons, collating them into hidden layers that compute intermediate outputs given inputs. The intermediate outputs in turn are used as inputs for subsequent computations, yielding for example a probability of an observation with the given input features belonging to a class. Neurons are the basic building blocks that are combined in a modular fashion. Every single neuron takes a number of inputs, multiplies each input with a weight and calculates the weighted sum. The sum is than passed through a non-linear activation unit, a function that transforms a linear input to a non-linear output. Various options for non-linear activations exists, in NN practise, ReLUs are used as default options. The weights are parameters that are optimised in the training procedure. B) Non-linear activations in the hidden neurons allow for the fitting of linear discrimators or regressors by transforming the feature space to be linearly separable.

Another common architectural feature of CNNs are pooling operations. In a pooling operation a subset of a spatial reach of convolutional filter outputs is selected, this may be the average or the maximum value. Pooling operations reduce the number of inputs and computations needed for subsequent layers. Furthermore, they model positional invariance. For example, a filter may score high when recognising a cat. For classifying the final image, it may not be necessary to know where exactly in 2D space the cat was detected. CNNs have led to the first deep learning breakthroughs in genomics. Alipanahi et al. used CNNs to predict the DNA binding specificities

of DNA and RNA binding proteins, demonstrating how to translate the CNN workflow from image analysis to genomics[59]. Zhou et al.[60] and Kelley et al.[61] subsequently used CNNs to predict chromatin features such as accessible chromatin, TF binding sites and histone modifications from DNA sequence. These works also introduced the concept of using CNNs to predict the effect of sequence variation on chromatin features and to map the importance of sequence determinants through saliency. Saliency measures the gradient of a models output with respect to the DNA sequence input. Usually the dot product of this gradient with the one-hot-encoded DNA sequence is computed (see Methods 6.2.5.5 for more details). An example of how a CNN architecture may be structured to predict chromatin features from DNA sequence is shown in Figure 1.4.

Four key features render CNNs successful in genomics applications:

1. CNNs can work directly on the raw data usually the DNA sequence. CNNs were designed for image analysis using the grey or colour values of pixels as input. This encoding can be readily translated to DNA sequence by one-hot encoding where the DNA is represented as a 4xN matrix with a row for every base stretching over the desired sequence length (Figure 1.4). CNN's have proven capable of effectively learning to detect patterns in the raw data that increase in complexity over subsequent layers in the network. Encoding DNA sequence as a narrow but very long image renders CNNs a natural fit to train on genomic problems. Furthermore, convolutional filters perfectly mimic the way researchers encode TF binding preferences motifs. CNNs allow for the detection of local spatial relationships such as TF binding motifs and to model positional invariance, e.g. via max pooling in the same network architecture.

2. Non-linear activation functions are an essential part of neural networks. Combining a multitude of non-linear activations allows neural networks to learn complex latent space transformations. The power of neural networks stems from being able to learn complex, non-linear representations. This, in turn,

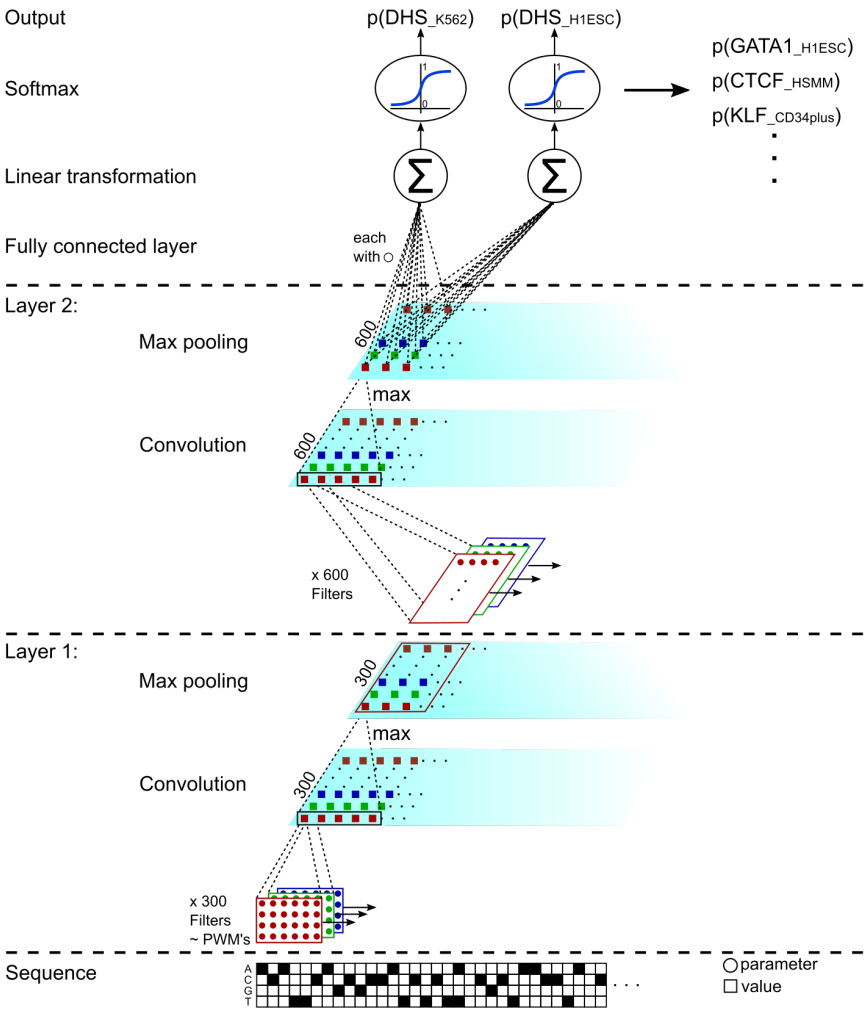

Figure 1.4: Schematic of a convolutional neural network for predicting chromatin features from DNA sequence. DNA sequence is the input for the neural network. The DNA is typically encoded in a one-hot-encoding, where each base is represented as a four-entry vector of all 0's and a 1 for the present base. Convolutional filters are small matrices of weights that are applied to the hot-encoded DNA sequence in a sliding fashion, outputting a weighted sum at each position. This is then followed by non-linear activation function. A number of filters is applied yielding a 2D output for every DNA sequence. The convolutional step is usually followed by a pooling operation, for example max pooling where the maximum output value in a set window range is selected. In the second layers, convolutional filters are then applied to the pooled output of the first layer using non-linear transformations and subsequent pooling operations. After a sufficient number of convolutional layers, the intermediate outputs are summarised in a fully connected layer, performing a weighted sum of all intermediate outputs, effectively performing a linear transformation. This linear output might be used to predict a score and fitted in a regression. However, in a classification task the linear output is transformed using a sigmoid or softmax function, yielding a score between 0 and 1 which can be interpreted as a probability score of the input DNA sequent to belong to a chromatin feature class. Usually, NNs are applied in a multi-label classification setting where multiple fully connected weighted sums are applied, one for each output class, for example a range of chromatin features across cell types.

makes fitting classifiers or regressors easier because non-linear relationships can be detected and exploited more easily.

3. Current DNNs also boast a much higher capacity then traditional ML models[62, 63]. Because it is the learning algorithm that optimises them, they tend to have many more weights, often millions. This allows these models to learn more complex and numerous patterns. Their high capacity also makes DNNs much more prone to overfitting and they require more data and more rigorous regularisation and validation strategies.

4. The combination of raw input processing and high capacity allowed CNNs to work with substantially larger sequence inputs than methods used previously. CNNs for predicting chromatin features use around one kilo base of DNA sequence context which covers the average size of regulatory elements well. Peaks identified from open chromatin assays are typically around 600 bp in size, where enhancers and CTCF site peaks tend to be shorter compared to promoter peaks which often appear as conjunction of multiple peaks. These peaks are comprised of the central nucleosome free region accessible to TFs (100 - 200 bp) flanked by the first evenly spaced nucleosomes to each site. The fragment size of DNase I cut or Tn5 transposed fragments leads to the peak size being larger the nucleosome free region. Histone modifications are linked to the flanking histones. Thus a DNA sequence window of 600 - 1 kb is usually used to capture the full size of accessible chromatin sites and capture histone modifications.

Sparked by advances in natural language processing other types of neutral network architectures have been applied to genomics applications. Noteworthy are recurrent neural networks, designed to capture long range dependencies in spatial data such as text or speech[64]. Recurrent neural networks have been applied to DNA sequence data to predict chromatin features[65] and promoter-enhancer interactions[66]. In practise, training recurrent networks on long spatial data such as DNA sequence has proven slow. Advances in convolutional network architectures, specifically dilated

convolutional neural networks combine the relatively quick computation speed of CNNs with the ability to model long range dependencies[67]. They are based on dilated convolutional filters that are constructed like regular convolutional filters as weight matrices that can be regularly interspersed with 0's. The dilation rate controls how many 0's are interspersed between the weights, the dilation rate minus 1 refers to the number of interspersed 0's. The convolutional filters are expanded accordingly to maintain all weights. For example, a 1D convolutional filter $[1, 2, 3]$ has a dilation rate of 1 meaning no interspersed 0's. With a dilation rate of 2 the filter would look like this: $[1, 0, 2, 0, 3]$; and with a dilation rate of 4 like this: $[1, 0, 0, 0, 2, 0, 0, 0, 3]$. Usually dilated convolutional are stacked with an exponentially increasing dilation rate[67], achieving a rapidly increasing spatial context reach without the need to employ max pooling layers that collapse the data along the pooling dimensions. Convolutional filters can be dilated in all filter dimensions. In genomics they are usually employed as 1D filters dilated only along the sequence dimension. Kelley at el. applied dilated networks to predict chromatin features and gene expression by incorporating DNA sequence at a scale of tens of kilobases [68]. Current advances in DL for genomics are focused on expanding DL to novel genomics problems and to adapt NN architectures specifically for genomic relationships.

1.3 Thesis aims

The overarching aim of the thesis research is to investigate the relationship between DNA sequence, the chromatin and the three-dimensional genome organisation and to develop computational models of gene regulation spanning these domains. As introduced in the above chapter, neural networks, specifically CNN's offer a promising approach to learn complex DNA patterns that are predictive of molecular features. We therefore aim to develop, evaluate and utilise CNN models for functional genomics. Specifically, we are interested in developing computational models allowing us to predict the effect of DNA sequence variation in the regulatory genome. Traditional computational and ML approaches struggle to learn complex sequence determinants and incorporate large DNA sequence contexts. CNNs have

been proposed as effective solutions to address these issues. Therefore, we are specifically interested in developing and evaluating CNN approaches for predicting the effect of DNA sequence variants.

In the first chapter, we explore CNNs for predicting chromatin features such as accessible chromatin and TF binding from DNA sequence. We re-implement published architectures, re-train models from scratch and evaluate their performance. We then integrate these models in a flexible pipeline to predict the effect of non-coding sequence variants and evaluate the model's utility for interpreting such variation. We employ this pipeline to dissect non-coding sequence variants identified by Genome-Wide Association Studies (GWAS) to be associated with red blood cell linked phenotypes.

In the second chapter, we develop a novel CNN model specifically for predicting chromatin interactions from DNA sequence at large scale. We utilise dilated convolutional neural networks to operate on DNA sequence at the megabase scale and learn large spatial relationships. We train these models using publicly available Hi-C data at high resolution. We than use targeted chromosome confirmation capture assays at high sensitivity to validate the chromatin architecture predictions made by our models.

In the third chapter, we utilise our developed chromatin interaction networks to map the sequence determinants predictive for chromatin interactions at high resolution and large scale to ask questions about which genomic elements help dictate the mapped interaction patterns in the genome. We then evaluate the potential of our networks to predict the impact of DNA sequence variants, both larger structural as well as well as single base pair variants.

2

Deep learning for chromatin feature prediction in the context of GWAS variant interpretation

Contents

2.1 Introduction 23
2.2 Results 28
2.2.1 Re-implementing existing deep learning approaches for chromatin feature prediction 28
2.2.1.1 Existing network architecture 28
2.2.1.2 Reimplementation in Tensorflow 29
2.2.1.3 Dilated networks for high resolution context aggregation 33
2.2.2 Chromatin feature networks in a pipeline for dissecting GWAS variants 35
2.2.2.1 An integrated pipeline for dissecting genomic variation 35
2.2.2.2 Predicting the effect of variation with deep learning 37
2.2.2.3 Evaluating deep learning-based chromatin predictions 38
2.2.2.4 Prioritising GWAS SNPs for functional follow-up 43
2.2.3 Data compendium scale 49
2.3 Discussion 54
2.4 Data & Code Availability 59
2.5 Acknowledgments 60

2.1 Introduction

Over the last decade a surge of Genome Wide Association Studies (GWAS) has identified genetic variation associated with a plethora of human traits, covering basic features as well as disease associated phenotypes. Interpretation and exploitation of this rich genetic information requires researchers to identify the causal variant and to link the genetic signals to causal genes and pathways in the right cell types.

Linkage disequilibrium (LD) frequently causes trait-associated variants to be closely associated with tens or hundreds of other variants. Any of the highly correlated variants may be causal. Approaches for prioritising the most likely causal variants are frequently grouped under the term fine-mapping. Genetics driven approaches for fine mapping either: 1) increase the density of SNPs by target re-sequencing or 2) leverage ethnic LD differences and LD based prioritisation. This can be achieved through applying an empirical threshold to the r^2 (lead SNP - linkage SNP correlation measure) or by narrowing down the likely causal SNPs using the r^2 and estimated effect sizes with penalized regression or Bayesian methods (excellently reviewed in Schaid et al. 2018[69]). Genetic fine mapping alone can impose strong assumptions and frequently does not narrow down the list of candidate SNPs sufficiently to effectively prioritise time consuming functional characterisation experiments. Moreover, it does not inform on the underlying molecular mechanism of action which is critical for understanding the cellular alterations, including whether it is a gain or loss of function which is essential for designing potential therapeutic modalities.

Functional fine mapping seeks to prioritise variant using molecular phenotypes. Such approaches are usually based on measurements of gene expression or on epigenetic features such as regions of accessible chromatin, TF binding or histone modifications. Big consortia efforts, such as the ENCODE[70], Roadmap[71] and GTEx, are producing vast amounts of publicly available epigenetic data across diverse cell types, fuelling functional fine mapping approaches. The vast majority of

identified sequence variants (95 %) lie within the non-coding regions of the genome and are suspected to alter gene activity by perturbing gene regulation[72, 73]. Variants may only exert their effect in specific cell types or at narrow differentiation time points and the dynamics of the regulatory genome thus further complicate variant interpretation[74]. Addressing all three bottlenecks requires systematic functional dissection. The sheer number of associated variants makes computational tools that predict their potential effect crucial for prioritising a tractable number of variants for experimental validation.

Models for predicting the effect of non-coding sequence variation are not designed to operate in a vacuum. Rather they are intended to form part of integrated pipelines for studying genomic variation. In our case, a pipeline to functionally dissect genetic variation associated with red blood cell traits. We therefore describe here the main assumptions and the principal workflow of this pipeline to describe and evaluate the developed machine learning tools in a comprehensive context.

The pipeline was designed with the working assumption that the mechanisms through which single nucleotide variations can affect gene function or expression can be divided into five broader topics:

1. Protein coding: Changes to exonic sequences may cause non-synonymous changes in the encoded amino acid chains or frame shifts that in turn may cause changes in protein function or transcript depletion through non-sense mediated decay triggered by premature stop codons.

2. Splicing: Intronic variation may alter the positions of splice sites and the frequency of their usage. Alterations in the splicing patterns in turn my cause different and even novel isoforms or frame shifts with the potential consequences described above.

3. Loss of cis-regulatory activity: Variation within cis-regulatory elements may impact transcription by altering the cis-regulatory activity. The primary

hypothesised mechanism is the loss or modification of TF binding sites. This may be close to a transcription start site when located within promoters or distal, with variation in enhancers intronic or intergenic enhancers.

4. Gain of cis-regulatory activation: If alterations and loss of cis-regulatory functions are hypothesised as being frequent mechanisms, then gain of cis-regulatory activity seems a plausible yet understudied mechanism. Gain of function may take the form of novel TF binding sites within existing elements or the creation of entire novel elements.

5. Altering the 3D genome: Genomic variation may impact the three-dimensional genomic architecture that in turn impacts the gene expression through altering enhancer promoter contact frequencies. Variants may alter, disrupt or create boundary elements and alter the magnitude of insulation or shape novel interaction domains.

Neither protein coding nor splicing mechanisms were a focus in the work presented in this thesis, but have been taken into account for the full analysis and functional follow up. For the work presented here, we refer to loss and gain of function relative to the allele present in the reference genome.

The work presented in this thesis focuses on methods for predicting the impact of non-coding sequence variation. Traditional computational methods for predicting the impact of non-coding sequence variation are broadly based on the following principles:

1. Linking to known variants with known effects such as clinical consequences[75].

2. Overlap with genomic annotations such as splice sites, regulatory elements or conserved regions across species. These can be direct read outs of epigenetic assays or patterns and annotations derived from the integration of multiple genomics signals, for example chromatin segmentation into promoters, enhancer, repressed regions etc.[39, 40]. Genomic annotations may also include indicators of chromatin interactions.

3. Overlap with variants associated with quantitative changes in gene expression, epigenetic marks or protein binding, across individuals (quantitative trait loci - QTL).

4. Overlap with and disruption of sequence patterns identified to be predictive of regulatory function. Regulatory function is often approximated by epigenetic signals such as accessible chromatin, histone modifications or transcription factor binding or footprints. Methods may search for the disruption of TF binding motifs[76], k-mers and motifs associated with TF footprints[77, 78] or sequences predictive of open chromatin[46].

5. Linking variants with the regulatory activity of the underlying sequence in reporter assays.

6. Linking variants with the observed effects of deleting the underlying sequence in genomic engineering screens at the level of entire elements[79] or at single base pair resolution[80].

Attempts to attribute cis-regulatory potential to genomic variation usually involves intersection with reference maps of regulatory activity. These reference maps provide the necessary context for tools that predict the impact of genetic variation. For example, TF binding motifs are frequent both within and outside of regulatory elements. Predicting the impact of genetic variation on motifs without knowledge of a motif's location may thus lead to wrong conclusions. Open chromatin assays such as DNase-seq or ATAC-seq are commonly used options because they map regulatory potential or protein binding in a relatively unbiased manner. The origin and completeness of these reference maps is crucial for the comprehensiveness of the functional dissection. The reference maps frequently suffer from two limitations. 1) Regulatory activity is highly dynamic across cell and tissue types[70]. Intersection and evaluation of variation thus requires the reference maps to be representative or at least inclusive of the regulatory activity of all cell types in which a variants effect might come into play. 2) Reference maps are frequently constructed from a limited

and often biased number of individuals and therefore represent a very under-powered sampling of the regulatory potential within a population. Limitations in these reference maps also limit their utility for interpreting sequence variation.

The incompleteness of reference maps at the population level renders gain of function variants particularly challenging to identify. For reasonably common haplotypes reference activity maps from a sufficiently large number of individuals may highlight regions prone to gain of function variation, although the generation of such maps across multiple individuals for a given cell type is rare. However, rare variation will likely evade coverage from even such multi-individual reference maps. Functional impact prediction using neural networks offers a potential solution to this problem, requiring only data from a single individual per cell type. Since they are designed to effectively learn the probability of a sequence (typically $\sim$ 0.6 - 1 kb) being associated with a given genomic activity or function, they can assess the likelihood of a given sequence variant changing that probability in the context of the variant's surrounding sequence[60, 61]. These approaches therefore learn the general sequence context associated with an activity, essentially the presence and arrangement of motifs within a given sequence and are therefore able to predict any relative loss or gain of function associated with the inclusion of a variant. As this only requires training on data from a single representative genome for a given cell type, such approaches can assess the effect of any given variant independent of its frequency in the population.

Here, we reproduce existing neural networks for non-coding variant effect prediction and adapt and integrate them into a pipeline for dissecting red blood cell trait associated genetic variation.

2.2 Results

2.2.1 Re-implementing existing deep learning approaches for chromatin feature prediction

At the start of this thesis work, autumn 2016, the first neural network approaches for genomics and the non-coding genome had been published recently[59–61]. The code base for all these applications was written in LUA and the torch deep learning framework. Google's TensorFlow[81] deep learning library implemented in C++ and python was gaining more traction in the scientific community, extending its user base and expanding its functionality. We consequently decided to re-implement the published convolutional neural network approaches in TensorFlow, with the aim of getting familiar with the deep learning framework code structure and functionality, producing a flexible application for our downstream analysis and reproducing the published results independently.

2.2.1.1 Existing network architecture

Two published, genomics deep learning applications were of interest to our work. DeepSEA[60] and Basset[61]. Both are convolutional neural networks that learn to predict a range of chromatin features as a multi-label classification task, predicting a probability score between 0 and 1 for each chromatin feature using the underlying DNA as input. They differ in terms of input data and architectural details (Table 1). DeepSEA uses a compendium of human chromatin features spanning transcription factor binding and histone modification ChIP-seq as well as DNase-seq assays mapping DHS for open chromatin. In contrast, Basset uses exclusively open chromatin data from DNase-seq. Both methods derive their data set from publicly available data repositories chiefly from the ENCODE and Roadmap consortia. Basset focuses on a sequence window of 600 bp, a size reflecting the average open chromatin peak, whereas uses a 1 kb DNA sequence, likely better suited for capturing broader histone marks as well. Basset defines open chromatin sites based on merged union peaks. DeepSEA bins the entire genome into 200 bp bins and classifies a sequence to belong to chromatin class X if the corresponding peak call from X

overlaps a given bin by at least 50 %.

Both methods use three convolutional layers with a varying number of hidden units. DeepSEA follows the convolutional layers with the final fully connected layer, whereas Basset uses two additional linear layers before the final fully connected one. For both methods, the final output neurons, one per chromatin feature class and cell type available in the data compendium, perform a logistic regression with a sigmoid function predicting the probability (0-1) of the sequence to be associated with the respective chromatin feature in the respective cell type. This class affiliation is defined based on the sequence windows overlap with open chromatin or ChIP-seq peaks.

Ultimately, we were more interested in models like since models that are capable of predicting a diverse range of chromatin features offer more utility for interpreting sequence variants. For example, a mixed chromatin feature model could predict a variants impact on open chromatin as a proxy for gene regulatory potential and simultaneously predict the effect on the binding of specific TFs.

Table 2.1: Overview of and Basset data set up and architectures.

	Input Data	Encoding	Architecture
	918 Chromatin Features (Histone, TF, DHS) across 35 cell types	1000 bp DNA sequence centred on 200 bp genomic bins overlapping with chromatin feature peaks	3x Convolution (filters 320 - 960) →fully connected
Basset	DHS from 164 cell types	600 bp DNA sequences centred on union DHS peaks	3x Convolution (filters 200 - 300) →2x Linear Layer (filters 1000) →fully connected

2.2.1.2 Reimplementation in Tensorflow

We reimplemented both architectures in TensorFlow. The Basset reproduction version reached an averaged Area Under Curve (AUC) under the Receiver Operator Characteristic (ROC) curve of 0.84. Applying the deepSEA like architecture to the

same data yielded 0.85 and an architecture with 5 convolutional layers achieved 0.86. For comparison, the published mean AUC for Basset is 0.895. On the deepSEA, mixed chromatin feature data set, our reproduction version of deepSEA reached an averaged AUC under the ROC curve of 0.96, matching the published metric. We also calculated the Precision Recall Curves (PRC) as they are less prone to inflation by the imbalanced numbers of true negatives[65]. The reproduction reached an averaged AUC under the PR curve of 0.46.

To better tailor our neural network for use in erythroid cell traits, we added 18 additional chromatin features from cell types across erythroid differentiation for a total of 936 cell types. This included 15 ATAC-seq, two DNase-seq and one CTCF ChIP-seq experiment sourced from Corces et al.[82]) and in-house[74]. In addition, we sought to reduce the large number of trainable parameters in the deepSEA architecture. The majority of parameters is dedicated to the fully connected layer which requires a connection from every position in the max pooling aggregated sequence dimension with every output class. We shifted parameters away from the fully connected layer by adding convolutional layers instead, increasing the overall sequence aggregation via max pooling and reducing the required reach of the fully connected layer. In addition to reducing the potential number of tune-able parameters the choice of this architecture was also driven by the reports in the field supporting the idea that deeper models also tend to more effectively learn more complex patterns[62], such as those expected in regulatory sequences.

Our final model, termed deepHaem, reached an AUC under the ROC curve of 0.94 and 0.42 under the PR curve (Figure 2.1). The total number of parameters was reduced from ~ 60 M to ~ 17 M although 18 classes have been added. We observed the same wide spread of performance across individual classifiers, largely depending on the data type. Transcription factor binding ChIP-seq based classifiers performed best, likely caused by their relatively narrow nature and the potentially limited sequence patterns required for their distinction, primarily TF binding motifs.

Open chromatin classifiers performance was in the intermediate, likely caused by their relatively narrow nature but more complex regulatory patterns underlying. Histone marks performed worst on average, explainable by their relatively broad nature. In addition, they are more likely to be strongly influenced by sequence patterns outside the 1 kb sequence context models here such as distal regulatory elements. These observations matched with the behaviour reported by Zhou et al.[60]

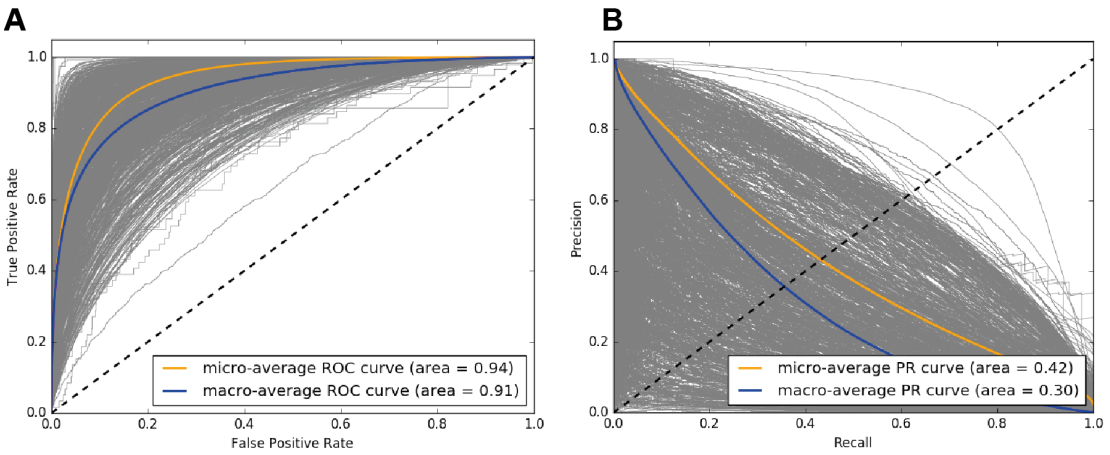

Figure 2.1: Evaluation of all 936 deepHaem classifiers. A) Receiver Operator Characteristic curves. The averaged AUC is 0.94 but is spread widely depending on the underlying data type. Micro and macro averages were calculated separately (see Methods 6.1.3). B) Precision Recall curves for all classifiers.

Because we were particularly interested in harnessing our additional erythroid predictors, we evaluated their precision in more detail by plotting the PR curves for these 18 classifiers separately (Figure 2.2). The erythroid data covers haematopeitic stem cells (HSC), multi-potent progenitors (MPP), common myeloid progenitors (CMP), megakaryocyte/ erythrocyte progenitors (MEPs) and late erythroblasts (Ery) from Corces et al. as well as erythroid time points from four differentiation protocols from single and combined donors covering ATAC-seq, DNase-seq and CTCF ChIP-seq. As observed over all classifiers, the TF classifier, here CTCF, showed the highest PR AUC. Interestingly, the open-chromatin classifiers from later erythroid differentiation stages, primarily from the in-house data performed better than the earlier stages. This effect may be related to quality of the ATAC-seq

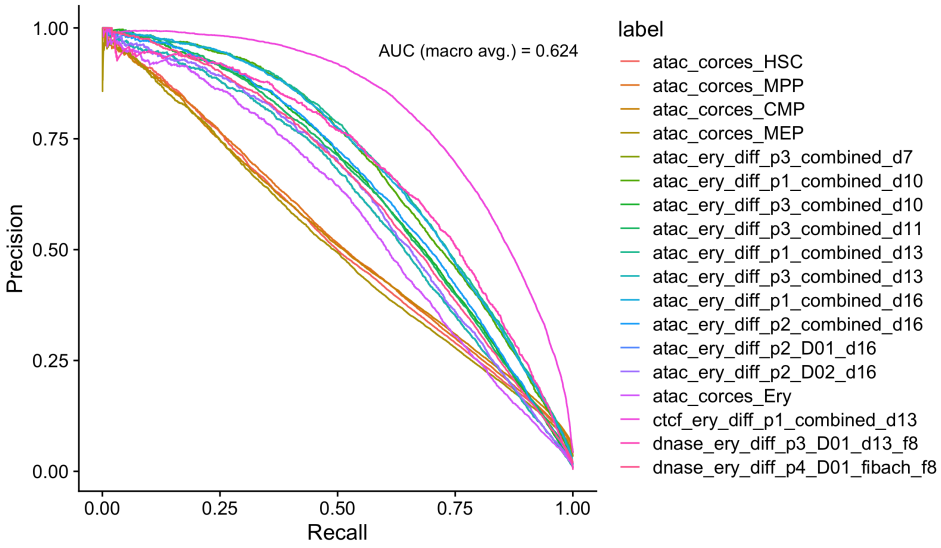

Figure 2.2: Precision recall curve of erythroid related chromatin features of interest. ATAC-seq classifiers are sorted along the erythroid differentiation timeline, ranging from haematopoetic stem cells (HSC) to late erythrocytes pre-enucleation (Ery). "Corces" labelled classifiers are derived from Corces et al. and "ery_diff" labelled ones are derived from different in-house erythroid differentiation protocols (p - refers to the differentiation protocol used, d - denotes the day of differentiation, combined – refers to data combined from 3 individuals, D – refers to single donor derived data). The CTCF ChIP-seq classifier performed best, followed by the intermediate and later differentiation stages. The early differentiation stage ATAC classifiers performed poorest but still above the average of all classifiers.

data and consequently the underlying peak calls. Alternatively, it may reflect that classifiers of chromatin features with more closely related features in the dataset train better, potentially due to more shared DNA sequence patterns of importance.

To visualise the performance of the trained classifiers in the context of a single, well characterised genomic locus, we used deepHaem to predict class probabilities from sequence for 1 kb genomic bins over the HBA locus in 250 bp strides (Figure 2.3) and compared them to the ATAC-seq read coverage of the respective assays. Overall, we observed good agreement between the deepHaem predictions and the ATAC-seq coverage. DeepHaem predicts the open chromatin signal in the HBA enhancer region (intronic region of NPRL3) well but also shows regions where the predictions pre-seed the ATAC-seq signal in terms of differentiation time-point (see green and yellow regions) or are not backed up by detectable ATAC-seq signal

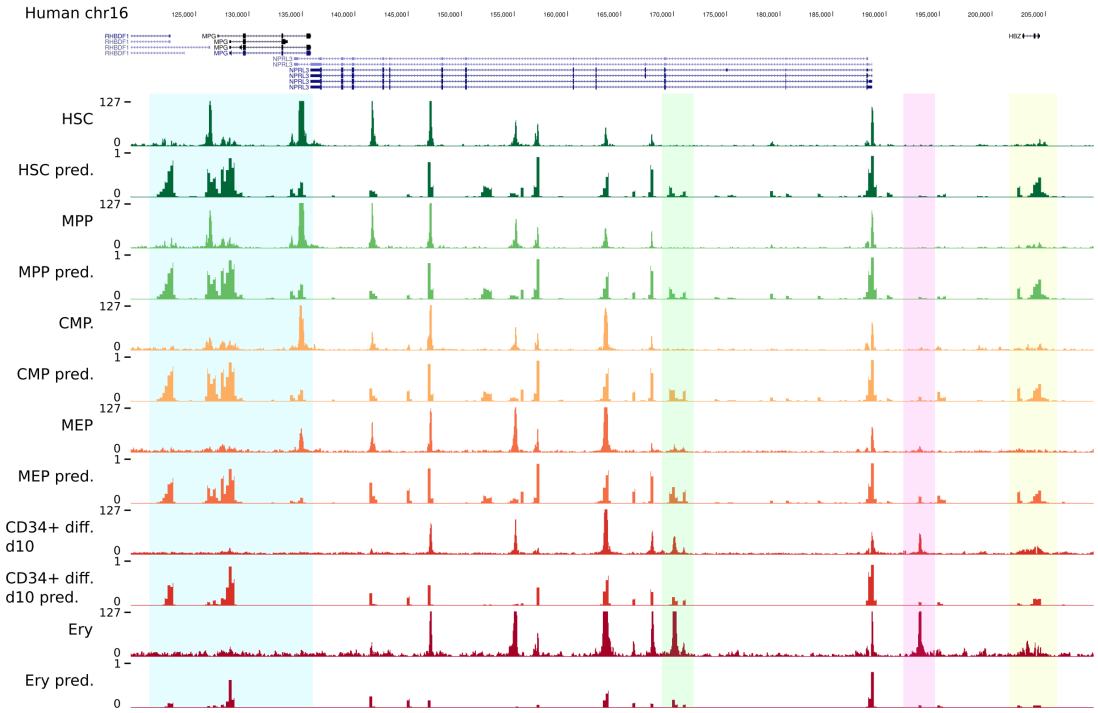

Figure 2.3: ATAC-seq signal and deepHaem predictions over HBA enhancer region. Predictions were made for 1000 bp windows in 250 bp strides. True ATAC-seq coverage and deepHaem probability score are paired for each differentiation time-point. Note that the height of the probability score does not necessarily have to reflect the ATAC-seq coverage height as it is trained on presence in a classification setting not as a regression problem. DeepHaem predicts the overall open chromatin structure well considering that the entire 100 kb sequence is processed. (Blue) Not all predictions are fully supported by ATAC-seq coverage or (Green) predictions are made to early in the differentiation. (Red) DeepHaem predicts an open chromatin site specific to later differentiation stages that are not supported by ATAC-seq coverage. (Yellow) Interestingly, deepHaem predicts the *HBZ* promoter region to be open. *HBZ* is known to be expressed in fetal but not in adult erythroid cells. Interestingly, the *HBZ* promoter is regulated by the same TFs as *HBA* but is repressed in the adult due to an unknown process.

(see red highlighted region).

2.2.1.3 Dilated networks for high resolution context aggregation

An important architectural feature of the convolutional neural networks employed so far, are the max-pooling layers. The max pooling layers summarise intermediate values along the sequence dimension. They are crucial for increasing the spatial reach of deeper layers without having to use increasingly large filter widths. They also reduce the number of computations required in deeper layers, making networks

with larger spatial reach tractable. The major drawback is that deeper layers effectively loose resolution along the sequence dimension. This can be desirable to capture positional invariance but can fail if the exact distance between patterns is crucial. For example, for detecting TF motifs with large gaps of variant bases or the relative position of co-factors.

While max pooling works well for the chromatin feature nets used, we hypothesized that the resolution loss would become a limiting factor when constructing networks with larger spatial reach. Dilated convolutional networks offer a potential solution[67]. Dilated networks employ convolutional filters with gaps (see Methods). The gaps are designed to increase exponentially with every additional layer. This enables large spatial ranges without massively increasing the number of required computations and without the need for pooling and the associated resolution loss. Dilated networks have been successfully applied in image analysis[67] and audio wave analysis[83]. We therefore experimented with dilated convolutional networks architectures for the chromatin feature nets. Subsequently, within the time frame of this thesis, dilated networks have also been used successfully in genomics[68].

We implemented dilated convolutional networks for predicting chromatin features from DNA sequence. As general design, we opted for a single convolutional filter with max pooling and a filter size of 20 followed by a stack of dilated convolutional layers with a fixed filter size of 3 with varying numbers of layers and hidden units. The first layer was designed to be able to capture large TF binding motifs, such as CTCF, and to allow for minor positional invariance. We adapted two architectural features that have been used successfully in image and sound wave analysis[83, 84]:

1. Residual connections allow for the propagation of identified patterns through multiple routes either through continuous convolutional operations or through linear shortcuts. For example, a single motif match could be already identified in the first layer. If the presence of this motif on its own independent of

co-motifs is informative for the prediction, residual connections would allow the information to flow to the final layers without having to maintain the information through additional layers of convolutional operations.

2. Gated convolutions (see Methods 6.1.2.3 & Figure 6.1) can model more complex relationships by combining standard convolution with hyperbolic tangent activation with a convolution followed by sigmoid activation as gate[83, 84].

To evaluate the performance of the dilated networks we designed and trained dilated versions of Basset and deepSea and compared their performance relative to their original architectures (Figure 2.4 & 2.5)).

All dilated versions of Basset achieved lower training and test losses as the reproduction CNN version (Figure 2.4). In contrast, the dilated versions of deepSEA did not reach the same loss levels as our reproduction version (Figure 2.5). Interestingly, neither of them reached the loss scores reported for deepSEA and DanQ, a recurrent network implementation of deepSEA, by the DanQ authors[65]. In our work, we were only able to overcome this gap in loss values by changing the test and validation sets to entire hold-out chromosomes rather than sampling them randomly as proposed by the deepSEA authors. For all subsequent work on chromatin feature networks, we did not further consider dilated neural networks. We instead used a convolutional architecture with max-pooling but with 5 convolutional layers and a single fully connected layer as described above.

2.2.2 Chromatin feature networks in a pipeline for dissecting GWAS variants

2.2.2.1 An integrated pipeline for dissecting genomic variation

We designed a pipeline for the systematic, functional dissection of trait associated genetic variation as identified by GWAS[74]. Here we applied this pipeline to a set

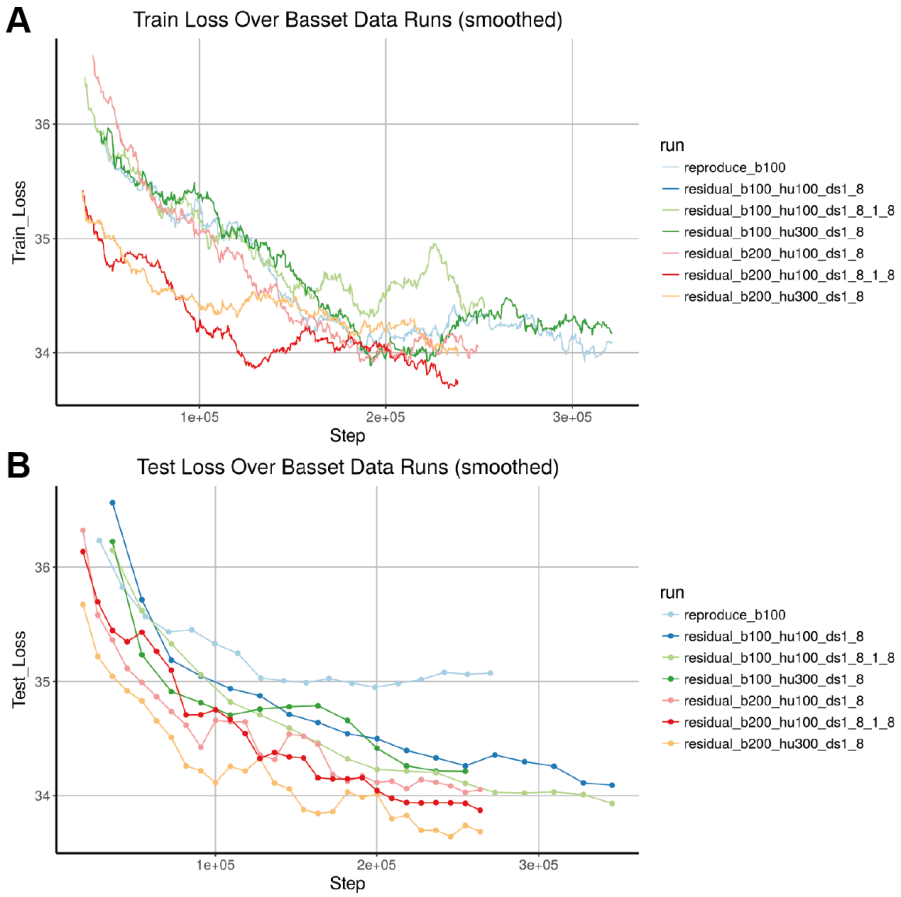

Figure 2.4: Training and test loss comparison for reproduction and dilated versions of Basset. Training and test loss comparison for reproduction and dilated versions of Basset. Shown are training and test loss over training steps. The reproduction version of Basset is shown in light blue, the other colours correspond to different versions of dilated networks, varying in terms of number of hidden units and dilated stacks. Reproduce indicates the reproduction versions, residual indicated the dilated networks with residual connections; b - refers to the number of convolutional filters in the first layer that is followed by max pooling. hu - refers to the number of dilated convolutional filters in each dilated layer; ds - indicates the dilated stack set up with exponentially increasing dilation rates. E.g. 1_8 refers four dilated convolutional layers with dilation rates of: 1,2,4,8.

of ~ 26 k unique SNPs associated with seven red blood cell traits[85]. These SNPs include the identified lead variants as well as all SNPs in tight association with the lead ($r^2 > 0.8$). To form a reference map for SNP intersection, we combined ATAC-seq, DNase-seq and CTCF ChIP-seq from three individuals across timepoints of erythroid differentiation as well as publicly available ATAC-seq experiments from earlier erythroid progenitors. The union of separate peaks called was used as reference map for SNP intersection. Gene intersection and ANNOVAR[86] were

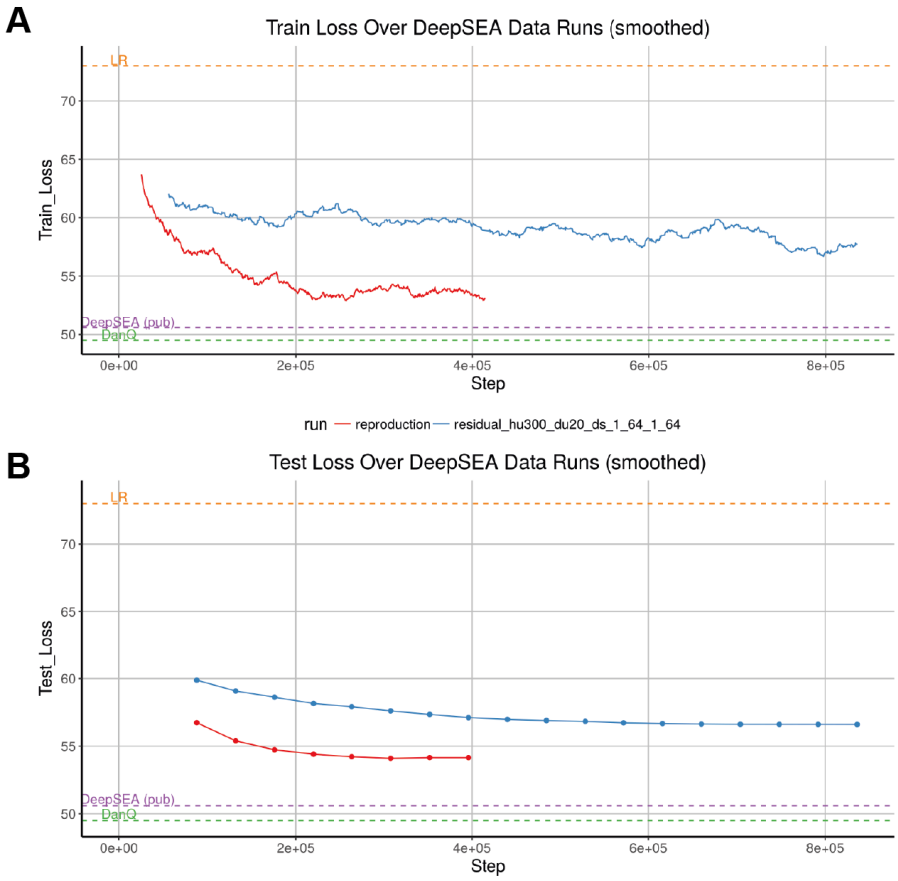

Figure 2.5: Training and test loss of reproduction and dilated deepSEA versions. The reproduction version of deepSEA is shown in red and the best performing dilated deepSEA version is shown in blue. Purple and green dashed lines represent the loss values for deepSEA and DanQ (a recurrent network version) reported in the DanQ publication. Yellow dashed line represents the background performance of a logistic regression classifier based on k-mers.

used for SNP annotation (coding and non-coding) and coding effect prediction. Non-coding effects were predicted based on DNase footprints[78], and deepHaem and were substantiated using allelic imbalance analysis. Splicing effects were predicted using SpliceAI[87]. The combined computational effect predictions were then used to prioritise likely regulatory causal variants for functional follow up via CRISPR/Cas9 genome editing[88].

2.2.2.2 Predicting the effect of variation with deep learning

We employed deepHaem under the hypothesis that a computational model sophisticated enough to predict cell type specific genomic features from DNA sequence can

be used to predict the effect of genomic variation on altering them. Open chromatin as well as TF binding were used as proxies for regulatory function.

We predicted the effect of all SNPs using the erythroid classifiers we introduced into the deepHaem model, mainly ATAC-seq across progenitors and erythroid differentiation timepoints. Each classifier predicts a probability score for a 1 kb++ input sequence to be associated with the respective chromatin feature. To predict the effect of a variant, we predict the score of the 1 kb DNA sequence centred on the variant for the reference and the alternative allele. We then calculate a damage score as the difference between the reference and variant prediction. Positive damage scores reflect a reduced probability of belonging to the respective feature class, while a negative score indicates an increased probability. The damage score can thus flag potential loss as well of gain of function variants.

2.2.2.3 Evaluating deep learning-based chromatin predictions

Although neural network-based approaches have been suggested for predicting the effect of non-coding variation, they had not been comprehensively evaluated for use in dissecting genetic variants at scale. We calculated the Pearson correlation between SNP effect predictions across 16 of the 18 erythroid specific classifiers (Figure 2.6). The ATAC-seq classifiers across erythroid differentiation protocols and time points are highly correlated with all pairwise Pearson correlations > 0.7. While the earlier progenitors (HSC, MPP, CMP) are more highly correlated with each other (pairwise Pearson correlations > 0.7 but < 0.7 with other classifiers), which is expected since they are the are highly related cell types. The CTCF ChIP-seq classifier shows markedly lower correlation with the ATAC-seq datasets than they have amongst each other. This difference is expected given the different targets of the assays. Taken together, the correlation patterns reaffirm the tissue-specificity of deepHaem. To compare deepHaem predictions to an established method in detail, we compared deepHaem SNP effect predictions to the DNase footprinting based predictions at scale across a well characterised enhancer of the alpha-globin genes. We employed

39

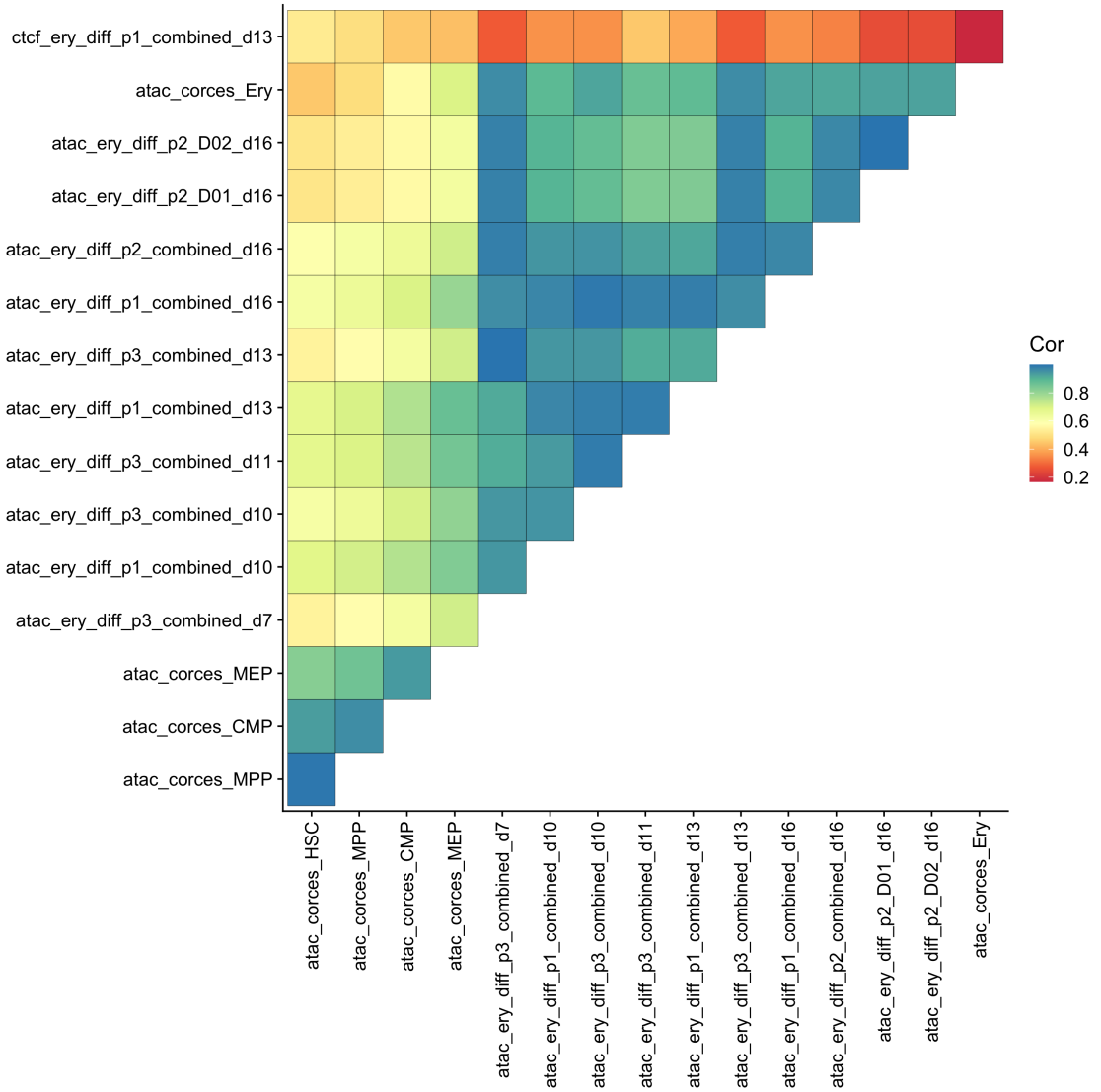

Figure 2.6: Pair-wise Pearson correlation between SNP predictions with erythroid specific deepHaem classifiers. Shown are ATAC-seq from progenitors and timepoints throughout erythroid differentiation. (p - refers to the differentiation protocol used, d - denotes the day of differentiation, combined – refers to data combined from 3 individuals, D – refers to single donor derived data).

Sasquatch, a DNase footprinting-based tool developed in earlier work[78]. In brief, Sasquatch utilises DNase-seq data to derive k-mer centered, meta footprints from genome-wide open chromatin to predict a variants impact on TF binding based on likely changes in the footprint profile. Sasquatch predicts if a variant is likely to alter TF binding, whereas deepHaem predicts if a variant is likely to alter a chromatin feature such as an open chromatin site. For every position across the enhancer, we predicted the effect of every possible single base pair change using Sasquatch and deepHaem (Figure 2.7). Both Sasquatch and deepHaem show high damaging scores at the prominent TF binding sites as indicated by clear troughs in the DNase footprint profile. The remaining peaks in predicted damage overlap relatively well. Importantly, deepHaem predictions decrease towards the flanks of the enhancer, while Sasquatch predictions remain on the same magnitude. This is an effect of the different prediction premises. Sasquatch assumes open chromatin content and is thus prone to false positives outside such regions. In contrast, deepHaem learns to predict the open chromatin context itself, making it more applicable for predicting variant effect outside of open chromatin sites.

We trained and employed deepHaem as a classifier, using the output probability scores associated with chromatin feature presence, as defined by peaks, and changes therein as predictors for the effect of a genetic variant. This mode has been suggested in the original neural network genomics applications[59–61]. Alternatively, one may treat chromatin features as quantitative entities by using, for example, the read coverage of the respective sequencing assay. Neural networks can then be trained to predict the coverage using a regression approach. Kelley and colleagues demonstrated in 2018 that such a neural network can predict chromatin feature coverage effectively[68]. Although related, classification and regression design ask fundamentally different questions. A classifier tries to predict the presence of a feature as yes/no question and predicts how much a sequence variant alters the overall likeliness of a sequence to be associated with that feature. In contrast, a regression seeks to predict coverage and therefore predicts the difference in coverage

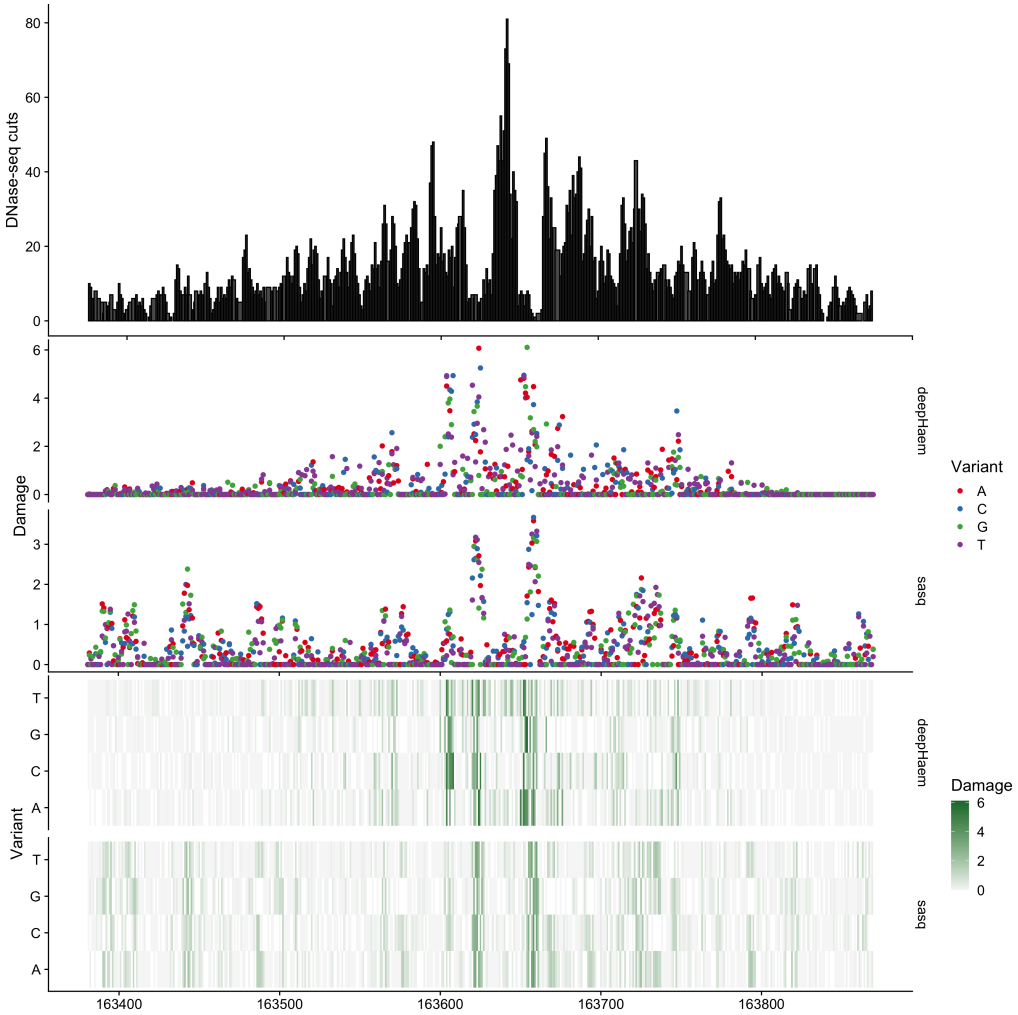

Figure 2.7: Comparison of deep learning based and DNase footprinting based damage predictions. Top) DNase-footprinting track. Middle) deepHaem and Sasquatch damage scores colour-coded per respective variant base. Bottom) deepHaem and Sasquatch damage scores visualised as matrix with variant bases split into rows. Only positive damage scores were kept for visualisation. Predictions were standardised (z-score) respectively.

caused by a sequence variant. It is unclear, how exactly these two measures are related and how good a classification probability score reflects coverage and vice versa. To assess this in our context, we calculated the read coverage of erythroid chromatin features over the union open chromatin regions and correlated them with the respective tissue-specific deepHaem probability scores predicted from the underlying sequence (Figure 2.8). We observed Pearson correlations between 0.39 and 0.71 depending on the chromatin feature. CTCF ChIP-seq showed the highest score, likely reflecting the less complex nature of CTCF bound site relative to

other classes of open chromatin sites. The scatter plots show a sigmoid like shape matching to the sigmoid activation function employed in the classification task. Taken together, this suggests that the classification probability score is not a perfect but useful proxy for read coverage. Moreover, the read coverage may reflect batch affects or other experimental rather then biological variation. Read depth based design demands proper read-depth normalisation for batch effects across experiments and for sequence effects such as GC content across regions. Encoding class affiliation in a binary manner based on overlap with peak calls alleviates this problem. For the work presented here, we decided to focus on the classification-based design, because we primarily used open chromatin as proxies for regulatory potential. We hypothesised that predicting SNP damage based on the likelihood of disrupting the entire entity would be more useful than predicting the change in coverage.

Intersecting SNPs with open chromatin elements is a common approach for providing genomic context to variant effect prediction tools. DeepHaem allows us to evaluate variants without the need to supplement genomic context. Nevertheless, intersection with open chromatin is part of our integrated pipeline, to provide genomic context to other tools and to separate potential loss from gain of function effects.

To this end, we intersected all ~ 26 k variant with a union set of open chromatin regions across differentiation time points and individuals as well as CTCF sites. This intersection in turn provided us with an opportunity to evaluate deepHaem's performance on a large SNP set. We hypothesised that SNPs intersecting with open chromatin sites should be more likely to affect TF binding sites. We thus expect enrichment of deepHaem damage scores in the open chromatin intersecting SNP set relative to the non-intersecting set. We plotted the empirical cumulative probability curves for the two sets observing an enrichment of positive and negative damage scores in the intersection set (Figure 2.9). Testing with a two-sample Komolgorov-Smirnoff (KS) test confirmed a significant enrichment (p-value $<$ 2.2e-16).

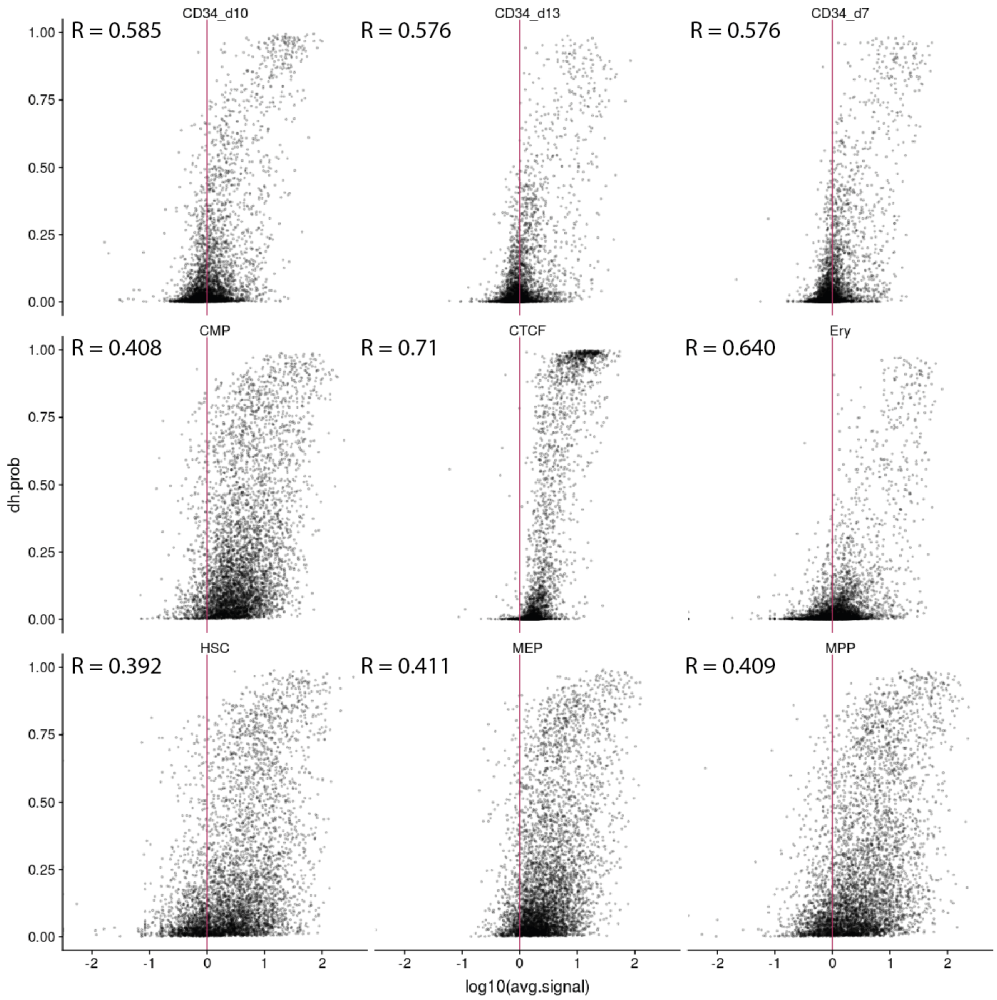

Figure 2.8: Correlation of deepHaem probability scores with ATAC-seq and ChIP-seq coverage. Nine classifier prediction vs. coverage scatter plots are shown. The read coverage was log_{10} transformed.

Taken together, we observed strong indications that deepHaem could be a powerful tool for dissecting non-coding genomic variation. However, although correlations and enrichments can bolster the confidence in predictions, ultimately, building trust in predictions requires full functional follow-up of variants predicted to have strong effects on gene regulation.

2.2.2.4 Prioritising GWAS SNPs for functional follow-up

We applied the deepHaem classifiers to the ~ 26 k Astle variants and ranked the effect predictions (Figure 2.10). We observed positive and negative predicted damages as potential gain and loss of function variants. In absence of gold standard sets for

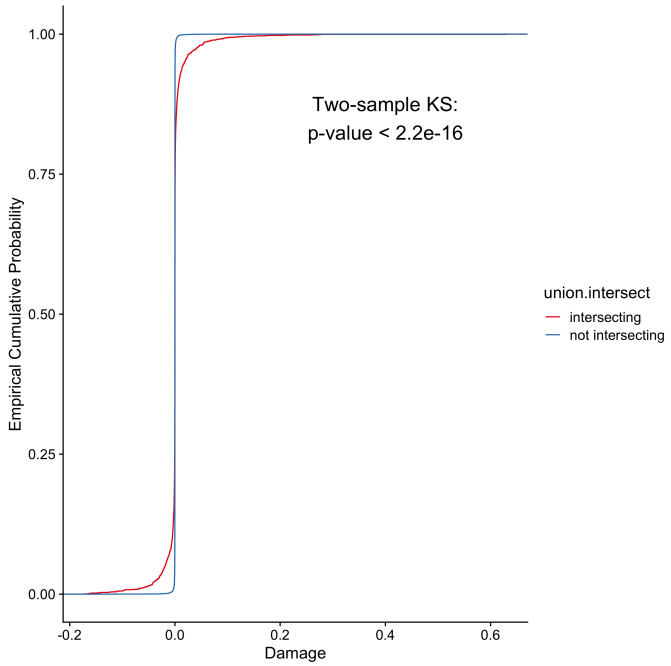

Figure 2.9: Enrichment of deepHaem damage scores in open chromatin intersecting SNPs. Shown are empirical cumulative probability curves of deepHaem damage for the union open chromatin intersecting and not intersecting SNP sets. The shown curves combine the predictions from all 16 erythroid specific classifiers, showing a significant enrichment (two sample KS test). KS tests showed significant enrichment for every individual classifier as well (p-value < 0.01, individual curves not shown).

calibrating the damage score, we used a rough damage threshold cut off at +/- 0.5 for prioritising variants.

DeepHaem allows us to simultaneously address two bottlenecks in GWAS variant prioritisation: LD blocks and tissue specificity. Using deepHaem we can predict the effect of all variants on different chromatin features across cell types and differentiation timepoints (Figure 2.11). Visualising the predicted impacts for all variants in high linkage allows us to prioritise the likely causal variant, while predictions across cell types allows us to narrow the cell type and differentiation time point. Two variants that were functionally validated using our integrated pipeline and in earlier work substantiated our confidence in the deepHaem predictions.

The single base pair substitution rs10758656 (A to G) showed a strong predicted impact on open chromatin features (Figure 2.10, blue dot) using deepHaem as

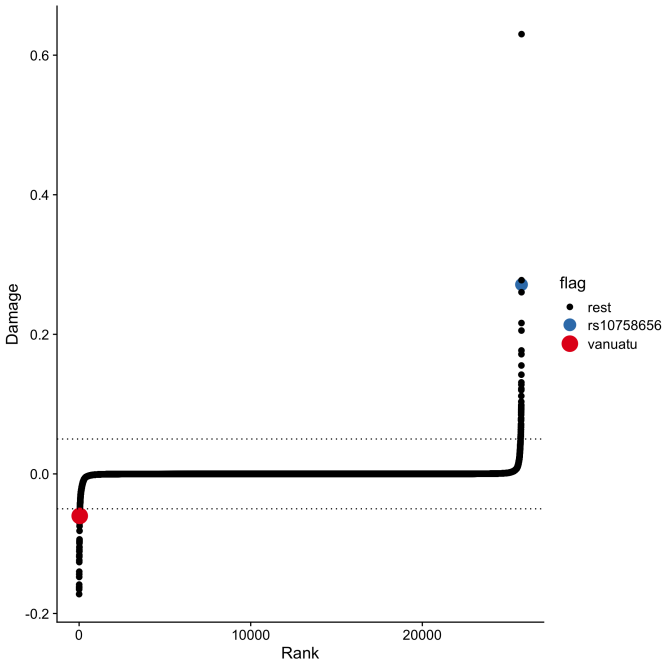

Figure 2.10: DeepHaem damage predictions for ~ 26k variants associated with red blood cell phenotypes or in high linkage with lead SNPs. Shown are the results for a single classifier (erythroid differentiation day 13). Dotted lines represent the empirical damage thresholds used for filtering the variants. The blue dot represents a variant is the *JAK2* locus that was prioritised from our integrated pipeline for full functional follow up. The red dot represents the spiked in Vanuatu SNV known to have a gain of function regulatory mechanism (also see Figure 2.11).

well as predicted DNase footprint changes from Sasquatch analysis. The variant was predicted to change a GATA binding motif within an open chromatin site in erythroid cells, located near the *JAK2* locus. Visualising the predicted effect of all variants in high linkage with rs10758656 (Figure 2.11 A) singled out rs10758656 as the likely causal variant. Predictions across timepoints of erythroid differentiation suggested an erythroid specific effect. Next, we checked for allelic bias in the open chromatin assays from our three donors. One donor was heterozygous showing a significant imbalance in ATAC-seq read coverage (A:131 vs. G:7 $p < 1e\text{-}30$ using a two-sided binomial test). Considering all evidence and computational predictions, the variant was prioritised for functional follow up. Experimental characterisation using Capture-C revealed chromatin interactions of the SNP region with the JAK2 promoter. Single base pair resolution genome editing in HUDEP cells revealed the G allele to lose GATA binding in an allele specific manner. *JAK2* expression was

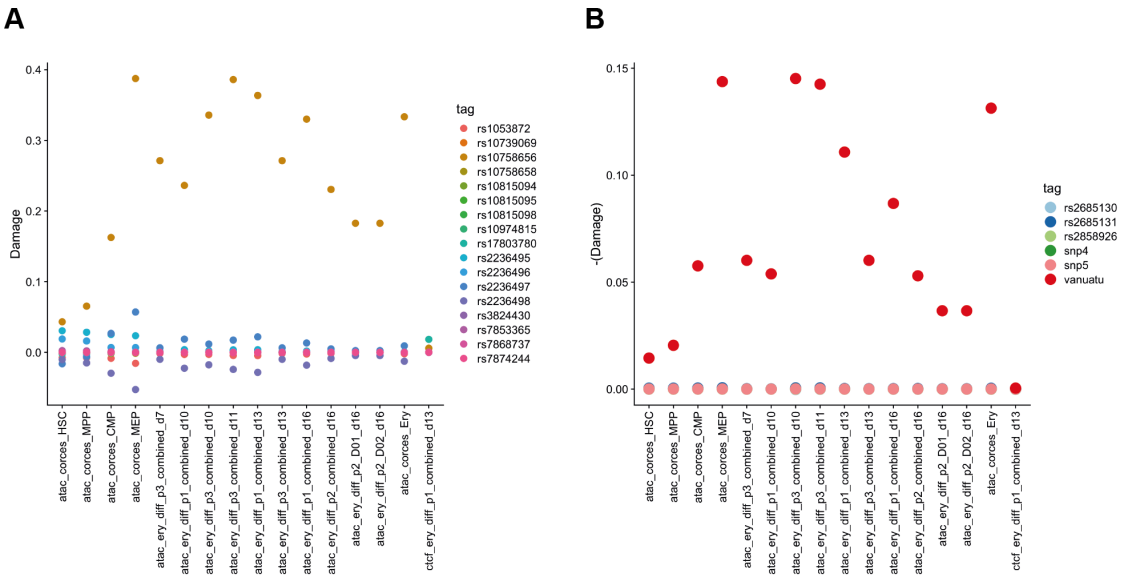

Figure 2.11: DeepHaem predictions for rs10758656 and the Vanuatu SNV and variants in high respective linkage across classifiers of erythroid differentiation. Both lead variants show stronger predicted effects than their linked variants. A) rs10758656 showing strong positive damage – decreased open chromatin probability and B) the Vanuatu SNV showing a strong positive damage – an increase of the open chromatin probability score.

significantly decreased in the homozygous G edit[74].

While the loss of function mechanism is being studied extensively, current literature lacks appreciation for potential gain of function variants. To test our hypothesis that deepHaem can identify potential gain of function SNPs, we turned to a rare, well characterised gain of regulatory function associated with a single SNV. De Gobbi and colleagues[89] have identified a single base pair change, termed the Vanuatu SNP, that is associated with the gain of a cryptic regulatory element. The cryptic element in term reduces *HBA* expression. Competition for interactions with the *HBA* locus enhancer elements is the suspected mechanisms (Bozhilov personal communication). De Gobbi and colleagues have identified five SNVs in the same locus on the Vanuatu allele. We used deepHaem to predict the effect of the Vanuatu and five associated SNVs across cell types over erythroid differentiation (Figure 2.11 B). The Vanuatu SNV is the only one with high predicted negative damage. The negative damage indicates a gain of function mechanism, fitting to the observed

gain of a novel GATA binding site[89, 90]. Single base pair genome editing by Krasimirov Bozhilov has confirmed that the Vanuatu SNV is both necessary and sufficient to cause the phenotype[90].

In addition to the examples discussed above, we sought to utilise the deepHaem predictions to identify additional loss and gain of function predictions that we could substantiate with the chromatin data available from three donors. To this end, we set the threshold for deepHaem predictions for an absolute predicted damage larger than 0.05 in any of the erythroid differentiation classifier (in-house differentiation day 7 - day 16). We then intersected the identified variants with the open chromatin (ATAC-seq) peak calls from the single time points from individual donors. We plotted the intersection in a binary heatmap (Figure 2.12) to quickly identify SNPs that showed differential support across the donors (Downes et al. personal communication). We identified and functionally characterised five SNPs within three loci where at least one of the SNPs intersected with peaks in two donors and not the third or vice versa.

Two SNPs are located in the *PPP1R3B* locus (Figure 2.13 A). A predicted loss of function SNP, rs330946, overlaps with ATAC-seq peaks in all three donors. One donor was homozygote for the reference allele and two donors who were heterozygote showed allelic imbalance in the ATAC-seq reads ([A:47|C:4], [A:59|C:5], both p-values $<$ 1e-9 using a binomial test). A predicted gain of function SNP, rs3748136, is located in the same locus and overlaps an ATAC-seq peak in two donors, both heterozygote for the SNP (Figure 2.13 A). The alternative allele has significantly more ATAC-seq read coverage ([G:1|A:16], [G:1|A:25], both p-values $<$ 0.002 using a binomial test).

Two identified SNPs are located in the *BZW2/ANKMY2* locus. For the A $>$ T change rs706027, two donors were homozygote for the reference A allele and one donor homozygote for the alternative allele. The reference allele coincided with ATAC-seq peaks (Figure 2.13 B), while the variant allele does not and no ATAC-seq

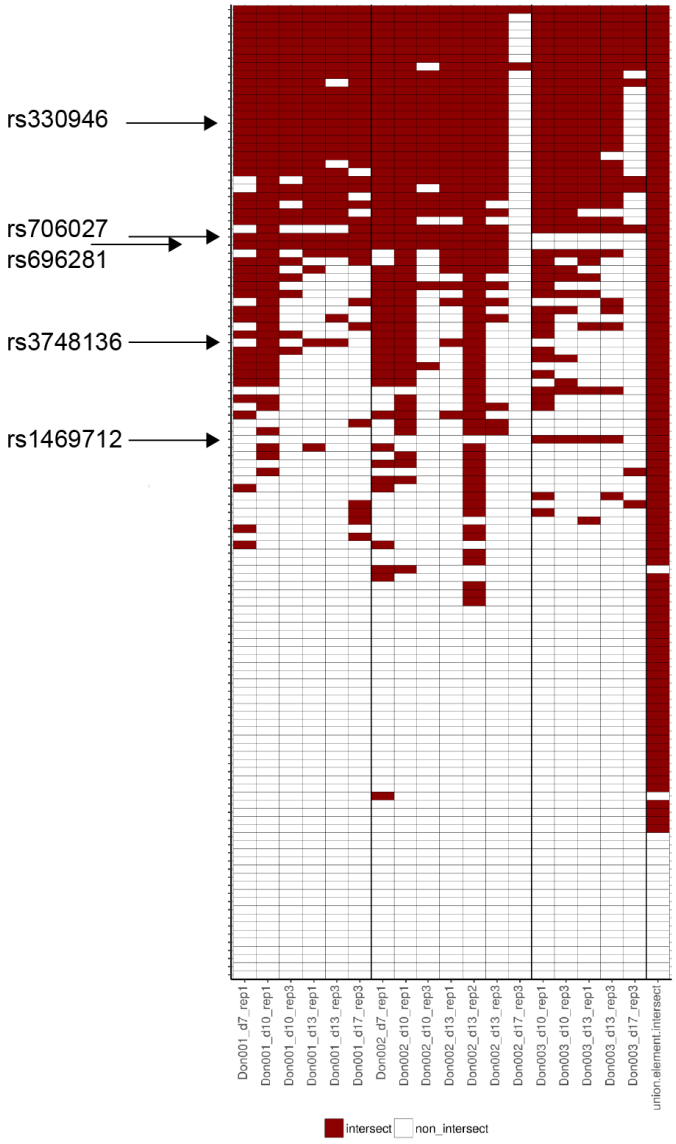

Figure 2.12: Intersecting deepHaem prioritised SNPs with open chromatin peaks from three donors. Multiple time-points through erythroid differentiation were selected as well as the union of all open chromatin and CTCF peaks across donors and time-points. SNPs with differential intersection profile across the three donors were prioritised for allelic imbalance analysis and functional follow up. (d – donates day of differentiation, Don - indicates the donor, rep - the replicate)

reads with the variant allele could be observed. For the predicted gain of function G > T transition one donor was homozygote for the alternative allele and one donor heterozygote with ATAC-seq reads showing a significant shift towards the alternative allele [G:3|T:106] ($p < 1e\text{-}27$ using a two-sided binomial test). The alternative allele coincided with ATAC-seq peaks in the respective donors while no peak could be

observed over in the third donors (Figure 2.13 B). The functional follow up of both loci with two identified SNPs demonstrated that the change in open chromatin was accompanied by changes in H3K4me1, H3K27ac, 3C interactions and gene expression (Downes et al. personal communication). Finally, a predicted gain of function SNP (C > A) overlapped with an ATAC-seq peak only in one donor, homozygote for the alternative allele (Figure 2.12 C), while the other donors were homozygote for the reference allele and showed no ATAC-seq coverage.

Taken together, several variants predicted by deepHaem to have strong effects on chromatin features could be substantiated with evidence from the three available donors. Functional follow up in our integrated pipeline demonstrated the causal role of the identified SNPs for gene expression. DeepHaem proved to be a valuable tool for prioritising GWAS identified SNPs for loss and importantly gain of function effect on gene regulation.

2.2.3 Data compendium scale

During the time of this thesis work the number of publicly available chromatin data increased significantly through efforts such as ENCODE, covering a broader range of targets in a more diverse set of primary tissues and cell types. Models trained on significantly more data than used here have been proposed[68]. Deep learning approaches usually benefit from larger and more diverse data sets. We therefore sought to apply deepHaem to an expanded data set of chromatin features. A larger data compendium allows for the training of more independent chromatin feature classifiers. This has the obvious benefit of making predictions possible across a larger range of new cell types and chromatin features. What is less clear, is how much a larger data compendium benefits the performance of a selected number of chromatin features of interest. In other words, if we are interested in open chromatin through erythroid differentiation, how much do we benefit from training on a more diverse data set of other chromatin features spanning both different assays and targets and a larger variety of cell types. While this question is of particular interest

for experimental design it has only been started to be addressed[91].

To this end, we downloaded and processed ~ 4500 chromatin feature data from a range of sources[68, 70, 71] and combined those data with 24 erythroid differentiation

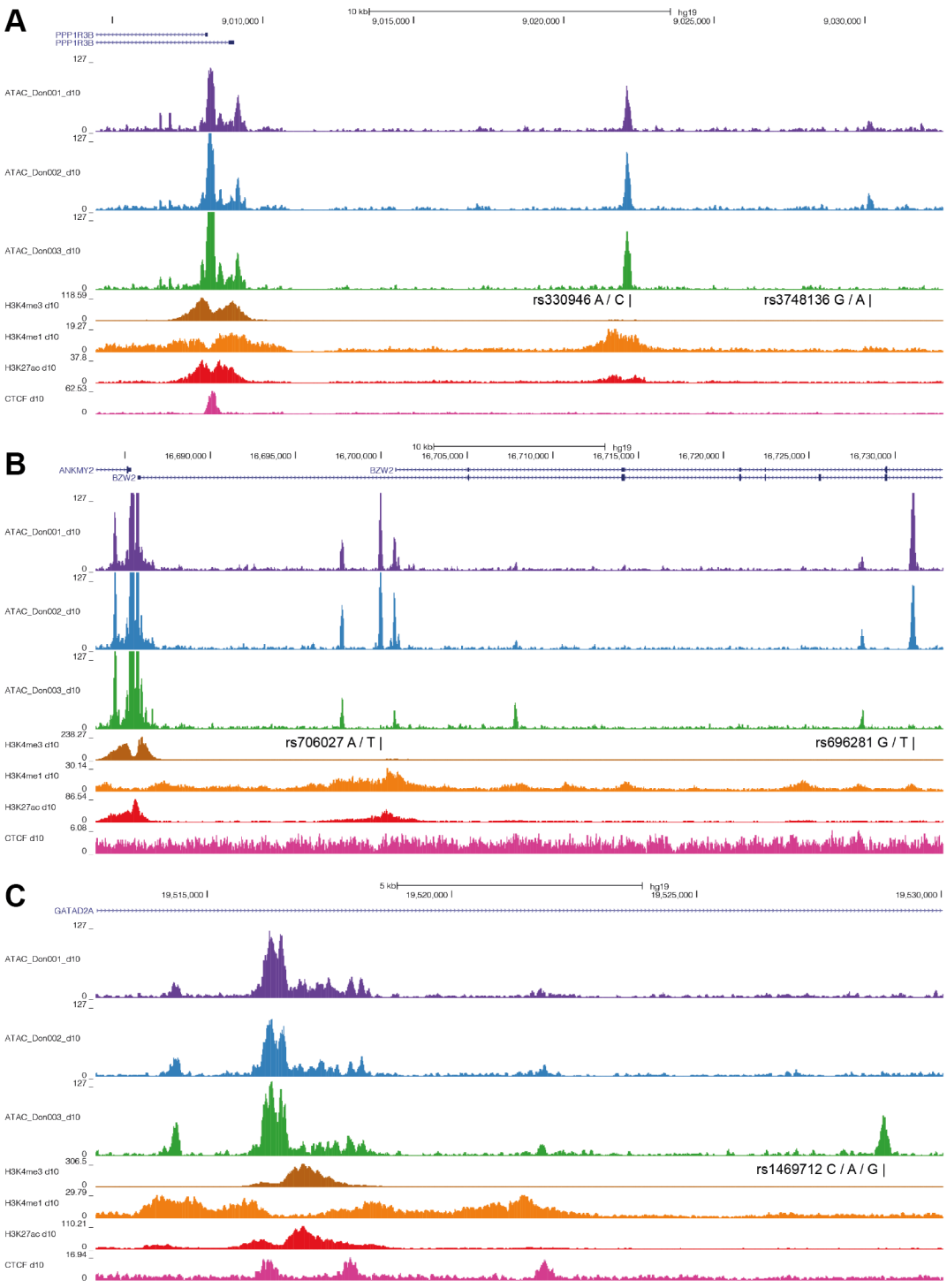

Figure 2.13: Prioritised regulatory variants in genomic context across three donors. Shown are ATAC-seq tracks of three donors overlapped with histone modifications and CTCF ChIP-seq coverage combined from three donors. Prioritised SNPs are highlighted. A) Predicted loss of function variant rs330946 overlaps within open chromatin peak in all three donors with two of them being heterozygote at the site and showing a significant allelic imbalance with more reference reads. The predicted gain of function rs3748136 overlaps a peak in two donors, both heterozygote for the SNP, showing significantly more ATAC-seq reads with the alternative base. B) rs706027 overlapped with a peak in donor one and two, homozygote for the reference allele. Donors three, homozygote for the alternative allele showed no peak and no ATAC-seq reads covering the variant. For the predicted date gain of function SNP rs696281, donor one was homozygote for the alternative allele, donor two heterozygote with a significant shift towards alternative allele reads and donor three homozygote for the reference. Only the alternative allele showed an open chromatin peak. C) Donor three is homozygote for the rs1469712 alternative allele (A) on showed the only present open chromatin peak at this site. No ATAC-seq reads covering the reference allele could be observed in any of the donors (Downes et al. personal communication).

related chromatin features[74, 82, 92]. From this data compendium we constructed four data sets:

1. deepsea - Using all data used in deepSEA and the erythroid data (#942)
2. dhs - using only open chromatin data (DNase-seq and ATAC-seq) (#694)
3. ery using only the erythroid data (#24)
4. chrom - using the full set (#4384)

We then trained deepHaem in two configurations, with the network described above (hidden units between 300 and 900) and with an increased network size with 1000 hidden units per layer. We calculated the PR curves and AUCs for the 24 erythroid related classifiers of interest. We first compared the performance of the different network sizes on the full chromatin dataset. With the inclusion of more erythroid related data, we observed a drop in the average AUC as compared to (Figure 2.14 A and 2.2 for comparison). This was likely caused by worse performance on some of the new data types. Two additional reasons may decrease the PR curve AUC: 1). Having more closely related data in a data compendium may drive down individual classifier performance and 2) training on a $\sim$ 4.5-fold bigger data set may require a

more complex model with more parameters. Our model with an increased number of hidden units showed higher PR AUCs (Figure 2.14 B). In a model with 2,000 hidden units per layer, we observed a further performance increase (data not shown). This suggests that an increased model complexity could further boost performance at the cost of significantly increased run time. We selected the 1,000 hidden units per layer model as an acceptable trade-off.

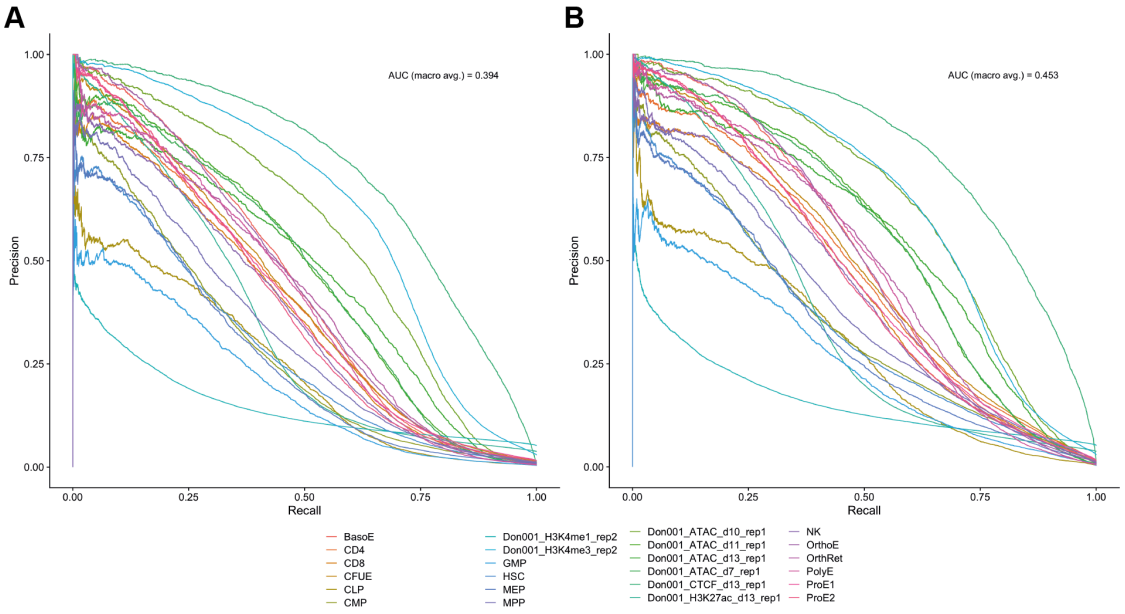

Figure 2.14: PR curves for erythroid classifiers of interest trained using a ~ 4500 feature dataset. Compared are A) the deepHaem architecture used earlier and B) an architecture with 1000 hidden unit per layer.

Next, we trained the 1,000 hidden unit per layer model on all four data set selections (Figure 2.15). We trained for 30 epochs, which was enough to plateau the training and validation loss for all four models. To make a fair comparison across the data set designs, we calculated the PR AUCs for the 20 erythroid related open chromatin classifiers of interest while excluding all genomic bins that did not overlap with at least one of these chromatin features. The deepSEA design performed worse. The full chromatin model performed best across the classifiers of interest, followed by the dhs model. Interestingly, the erythroid data only set, using only 24 chromatin

53

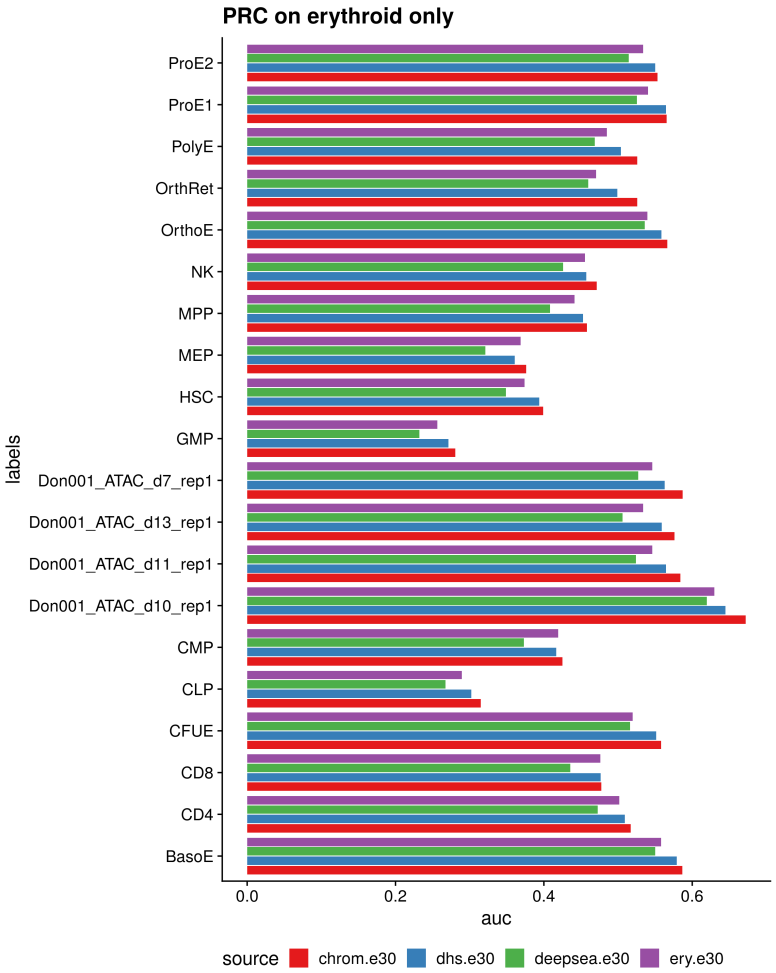

Figure 2.15: Comparing data selection performance. PRC AUCs were computed for 20 erythroid related classifiers of interest. Only genomic bins intersecting with at least of the erythroid related chromatin features were considered for evaluating the performance. All models were trained for 30 epochs at which point they showed saturation. (chrom. – full ~ 4500 data set; dhs - only open chromatin features ~ 700; data as used in deepSEA ~ 932; ery - only erythroid data of interest ~ 23)

features, performed overall better than the deepSEA set.

Because of training time constraints, we only trained one iteration of each model for the comparison of all four data sets. We next trained five iterations with random start points using the same architecture for the erythroid only and the full chromatin data. The mean macro PRC AUC (see Methods 6.1.3) for the erythroid-only models was 0.467 with a standard deviation of 0.00705 and for the full chromatin models 0.511 and 0.00543 respectively. Taken together, these results

suggest that training on larger and more diverse datasets benefits the prediction of a selection of chromatin feature classifiers of interest. This comes at the expense of model complexity and computation time. However, the margin between the performances is not large and training neural networks with a limited number of chromatin features still gives reasonable performance. This is an important finding for the potential application of these approaches where the number of chromatin features available is limited, for example for work in non-model organisms.

2.3 Discussion

GWAS have identified an abundance of genetic variation associated with a plethora of disease related phenotypes. To fully utilise this information for improving human health, we need to reliably pinpoint the biological pathways affected. However, linkage disequilibrium combined with the lack of a cell type context for such GWAS signals renders identifying the causal variants and the underlying mechanisms as a great challenge. Computational models are key for prioritisation a tractable number of candidate causal variants. While a plethora of computational approaches has been proposed, all come with specific biases and limitations and none has solved the prioritisation problem sufficiently.

We chose to apply DNNs and evaluate their performance for dissecting GWAS identified variants because DNNs offer promising advantage over existing approaches. Traditional methods, such as motif-[76] or footprinting-based models[77, 78] require chromatin context to interpret the findings and to reduce false positive and performed poorly for the interpretation of gain of function variants. Approaches that try to predict chromatin features and thus the context itself can circumvent this problem. However, traditional ML approaches proposed for this task usually work with very limited sequence context frequently relying on k-mers[46]. The limited sequence context does not enable these models to capture motif – motif interaction and prevents them from integrating all sequence patterns associated with a single

element. Convolutional neural networks can operate on a much wider sequence context that can capture entire elements (usually ~ 1 kb) and integrate the patterns within. Conversely, DNNs require large amount of labelled training data to learn complex patterns and relationships. In order to supply this data, neural network approaches rely on sampling the entire genome and examine and learn from the sequences, usually from a single reference, underlying epigenetic features genome-wide. Therefore, these approaches are effective even when data for a given cell type is derived from a single individual, which is generally the case. This is in contrast to QTL approaches, where differences at a single locus across individuals are taken into consideration and so requires deep sampling of a cell type across a population of individuals.

The potential to use DNNs for predicting the impact of non-coding sequence variation has been suggested in pioneering studies[59]. However, they have not been employed to interpret GWAS signals systematically. In this chapter, we have demonstrated how to use DNNs for dissecting GWAS variants and prioritise candidate causal variants for functional follow up. We and others have since advocated for the use of DNNs for these tasks[74, 91]. To employ DNNs for GWAS dissection we first reproduced two published models in an updated and flexible framework in TensorFlow. When training the reproduction networks on the same data we reached the reported loss values and accuracy for the Basset model but fell short of the reported performance of the deepSea model. A possible explanation is that published models can be trained for a long time and with only the best performing iteration being reported in the end. After switching to whole chromosome hold-out test sets, rather than the reported sampling, we reached published accuracy and loss levels. We chose a whole chromosome hold out dataset construction to ensure roughly equal representation of all types of DNA elements in the training, test and validation sets. Randomly sampled sets harbour the danger of not sampling certain element types, whereas whole chromosomes are considered as functionally intact

and comprehensive data sources.

Next, we added erythroid specific data of interest to tailor our data compendium to the dissection task, a GWAS of blood cell associated phenotypes. We hypothesised that having representative epigenetic data in the model is crucial for interpreting variation in the appropriate cell context. We chose to add this specific data to the pre-existing data compendium. We also introduce modifications to the network architecture to reduce the model size while maintaining complexity. We added convolution and max pooling layers, making the model deeper, but in turn reducing the complexity of the fully connected layers. These considerations are in line with advancements in the field of image processing, where deeper models with less complexity in each respective layer have gained traction[62]. We experimented with dilated convolutional neural networks for the chromatin feature prediction task but could not improve on the performance of the convolutional network with max pooling architecture.

Next, we integrated our optimised model, termed deepHaem, into an integrated analysis pipeline for dissecting sequence variations identified in GWAS. We applied deepHaem to a set of ~ 26 k SNPs including lead SNPs and SNPs in high LD, associated with red blood cell traits[85]. For each SNP we predicted the effect on each of the 18 erythroid-specific chromatin features. The SNP effect predictions proved highly correlated with open chromatin features of cell types during late erythroid differentiation identifying the likely target call types (Figure 2.6). We compared deepHaem predictions against a DNase footprinting-based method, Sasquatch, for predicting the effect of SNPs on TF binding (Figure 2.7). We observed good agreement between the methods with deepHaem showing the added benefit of demarcating regions of open chromatin intrinsically, while Sasquatch assumes open chromatin context, potentially leading to false positives if not properly filtered.

The first convolutional neural network approaches for genomics frame the task as a classification problem. The probability of a sequence belonging to a distinct chromatin class is predicted, scaled between 0 and 1. In this work, we reproduced DNNs as classifiers. Since the pioneering studies of DNNs in genomics[59–61], others have approached similar problems as a regression task, where the models are trained to predict a continuous signal, for example an assay's derived read coverage over a genomic element[68]. The relationship between regression and classification problem in the context of genomic DL has been poorly addressed. Here, we correlated the predicted chromatin feature probabilities of 9 classifiers with the average read coverage over the respective elements. Depending on the chromatin feature these correlations ranged from 0.39 to 0.7, being stronger for the chromatin features that also achieved higher accuracy (Figure 2.2). Our analysis suggests that classifier models can be employed to infer the read coverage, particularly when they perform well on the initial classification task.

The predicted effects of the ~ 26 k SNPs studied revealed both potential loss and gain of function variants. In this context, loss and gain of function are defined relative to the reference genome to which the variants are mapped. A gain of function mutation might thus still signify the disruption of a regulatory element within the human population but where the reference genome itself is an example of the allele lacking the molecular phenotype. Because, dissecting non-coding variation relies heavily on intersection with reference maps of regulatory elements, such variants nevertheless pose a challenge. In addition, examples of rare gain of function variants creating a cryptic regulatory element have been described[89]. These likely depend on local chromatin context[90]. Regardless of the evolutionary mechanisms underlying them, gain of function variants pose a challenging and underappreciated mechanism in GWAS follow up studies. Applying deepHaem allowed us to successfully prioritise variants from neighbouring and genetically linked SNPs, both in known examples from the literature and novel GWAS loci (Figure 2.11). We then substantiated our prioritisation with allelic imbalance analysis. Experimental follow up conducted in

the lab involved chromatin interaction capture, chromatin characterisation via ChIP-seq and ATAC-seq in healthy donors, as well as base pair resolution CRISPR/Cas9 genome editing and characterisation of the edited clones. Taken together, this work demonstrated that deepHaem can successfully prioritise loss of function (rs10758656, see Figure 2 11) as well as likely gain of function variants (Figure 2.12 & 2.13).

Training complex DNNs requires large amounts of diverse training data. DNN for genomics have usually been trained with a large compendium of chromatin data covering a broad range of assays and nuclear targets in a variety of cell and tissue types. The impact of the scale of the chromatin feature compendium is poorly studied. Sequence patterns can be shared between classifiers and their joint learning can thus improve power. We hypothesised that having well represented epigenetic data for the cell type of interest could help in learning predictive cell type specific patterns. To interpret sequence variation in erythroid cell contexts, we thus chose to add erythroid-specific data to the pre-existing data compendium. Others have argued that training on tissue-specific data alone gives better results than networks trained on a broad compendium[91]. However, in their study they did not explicitly test the benefit of combining a broad data compendium supplemented by designated tissue-specific data. We addressed this question by training individual networks on four datasets compositions (see 2.2.3): one containing only the erythroid chromatin data of interest, one combining the erythroid data with 918 chromatin features used in the original deepSea publication, one set combining ~ 4500 chromatin feature data with the erythroid set and one using only the open chromatin assays from the former set. Evaluating their respective performance on the erythroid classifiers, we observed that training the large compendium adds benefit over only training on data of interest. This comes at the expense of having to train bigger, more complex networks, but when this trade-off is manageable, we recommend large compendiums for future work.

In summary, we demonstrated that DNNs trained to predict chromatin features are valuable tools for dissecting non-coding sequence variation. Nevertheless, they do not represent a stand-alone solution for dissecting GWAS results but need integration into a comprehensive pipeline for prioritising variants. While CNNs are able to integrate a larger sequence context than previously possible, they still evaluate variants in a highly local context. CNNs can predict the likely effect of a variant on a regulatory element. However, to understand the consequent effect on gene expression, it is equally important to map the three-dimensional interactions of the element that shape its regulatory activity. Enhancers can exert their transcriptional enhancing effects over megabases and active promoters may interact with other promoters and influence their expression levels making the 3D genome crucial to understanding disease-associated gene expression[93].

Chromosome conformation capture techniques such as Hi-C[13] or Capture-C[15] allow us to map chromatin interactions at varying degrees of sensitivity and genomic scale. However, these assays are expensive and time consuming, especially when they need to be performed in the correct cell types and genomic backgrounds to understand sequence variants. A machine learning driven solution to predicting chromatin architecture could thus be a valuable asset for dissecting non-coding variant impact. We developed such a model in the following chapter.

2.4 Data & Code Availability

Majority of the chromatin feature data is available through ENCODE and Roadmap data portals. Chromatin feature compendium compositions were adapted from deepSea[60] and Basenji[68] respectively, supplemented additional erythroid data from Corces et al.[82] (GSE74912), Downes et al.[74] (GSE125926, GSE136976, GSE137982, GSE137983) and Schwessinger et al.[78] (GSE86393). All datasets used are listed in Appendix A (https://doi.org/10.5281/zenodo.4751244). The deepHaem code is available via github: (https://github.com/rschwess/deepHaem)

2.5 Acknowledgments

Guidance on implementing and evaluation of the DNNs was provided by Yee Whye Teh and Gerton Lunter. The integrated pipeline for dissecting GWAS variants was a central project of the entire Hughes lab. The design was driven by Jim Hughes and Damien Downes with contributions from the entire lab. Experimental chromatin data from healthy donors and edited clones were generated by Damien Downes, Matthew Gosden, Lea Nussbaum, Stephanie Hill and Priscilla Hirschfeld. Genome editing experiments were performed by Lea Nussbaum, Stephanie Hill and Priscilla Hirschfeld. Evaluation and prioritisation of variants for functional follow up was led by Damien Downes. Protein-coding effect predictions were conducted by Simon McGowan and splicing effect predictions were conducted by Richard Brown and Damien Downes.

3

Developing deep learning models for chromatin interaction predictions

Contents

3.1 Introduction 62
3.2 Results 65
3.2.1 Building deep neural networks for predicting chromatin interactions at mega-base scale 65
3.2.1.1 Design considerations 65
3.2.1.2 Hi-C data encoding 67
3.2.1.3 Model architecture 67
3.2.1.4 Transfer learning 70
3.2.1.5 Regression of distance normalised percentiles . 72
3.2.1.6 Predictions in different settings 75
3.2.1.7 Evaluation 76
3.2.1.8 Tissue-specificity 80
3.2.2 Validating computational predictions of chromatin architecture 84
3.2.2.1 Capture-C based methods for validation 85
3.2.2.2 Correlation of virtual4C profiles 88
3.2.2.3 Validating domain boundaries 89
3.2.3 Comparison of sequence-based prediction with chromatin feature based predictions 94
3.3 Discussion 100
3.4 Data & Code Availability 104
3.5 Acknowledgments 104

3.1 Introduction

The mammalian genome is organised in an intricate, hierarchical, three-dimensional structure that promotes or blocks interactions between functional genomic elements. This can have profound implications for gene expression, by determining which enhancers contact which promoters in an often highly tissue-specific context. Alterations in chromatin architecture can drive changes in gene expression and lead to disease or susceptibility for it[94, 95]. Interpretation of regulatory DNA variants thus requires us to assess variants within their respective, cell-type-specific chromatin architecture to map variant effects to effector genes. This is pivotal for understanding mutational mechanisms, predicting alterations in transcription and to understand downstream effects.

Conducting assays for mapping chromatin interactions, such as chromatin conformation capture[96] are expensive and time consuming, especially when trying to achieve genome-wide, all-vs-all contact maps at high resolution[13, 14]. In addition, to investigate the potential effect of a regulatory sequence variant on chromatin architecture, an assay has to be conducted in the right cell type in wild-type and alternative genomic context. This bottleneck has spurred the development of computational tools, seeking to predict chromatin interactions *in silico*. Usually these approaches use Hi-C data as the chromatin interaction baseline because of its genome-wide, all-vs-all measurements. Increasingly, more high-resolution Hi-C data across various cell types is becoming available publicly, providing a valuable resource for tool development. Simplistic approaches to predicting a variants effect on chromatin architecture rely on intersection with boundary elements and/or CTCF motifs. More sophisticated methods have been developed based on machine learning or polymer physics. They can be broadly grouped into four categories:

1. Polymer models: Physical models seek to predict the 3D structure of the genome[97–99]. They employ approximations of the DNA as polymer, for

example beads on a string[98]. Using Hi-C data of a given chromosome as constraints, these models learn the optimal polymer structure that best explains the observed Hi-C interactions. On top of the linear proximity signal, factors connecting polymer segments may be inferred from the data and then related back to enriched chromatin features. Fitted models can then be used to predict the impact of structural variation on the 3D structure and derived interaction maps. All polymer models segregate the DNA into coarse segments, losing the resolution and context of the DNA sequence itself. This limits the interpretation of the sequence determinants of genome folding. Furthermore, predicting the impact of smaller DNA sequence variants requires intermediate models that predict the impact of the variant on its binding properties. In addition, polymer models can only be fitted from observed interaction data of the corresponding chromosome. This prevents polymer models from generalising to unseen chromosomes.

2. Window-to-window based predictions: These models work under the hypothesis that pair-wise interactions of two genomic regions can be accurately predicted from the DNA sequence[66] and/or chromatin features, such as TF binding, accessible chromatin or histone marks, associated with each individual region[100, 101]. These methods therefore intrinsically neglect the wider chromatin context. Chromatin interactions are shaped by insulated domains, demarcated by boundary elements. To effectively predict enhancer-promoter interactions it thus appears crucial to incorporate information of domain boundaries and potential, interjacent boundary elements. Furthermore, if trained primarily on active regulatory element interactions, such models may struggle to predict interactions of regions not overlapping active regulatory elements. Consequently, such studies have refrained from predicting comprehensive maps of chromatin interactions and focused on pairwise predictions of promoter-promoter and enhancer-promoter pairs.

3. Window-to-window with interjacent window features: To overcome the limited scope of purely window-to-window based approaches, other models have been developed that take into account the interjacent window between two elements. This has been implemented by averaging the chromatin features in the interjacent window[102, 103] or by aggregating defined features in that region, such as CTCF motif orientation[47]. These approaches dissect the interjacent window by varying degrees of coarse segmentation, with averaging the entire window being the crudest. None, however, integrates a wider genomic context when learning from the DNA sequence itself. Thus, predicting the impact of sequence variation requires intermediate models that predict the impact of a variation on the important chromatin features. While the principal tools for building such a framework of sequential models exist, no such approaches have been proposed so far and extensive research is needed to find solutions and evaluate their accuracy. Moreover, window-to-window based approaches neglect the potential influence of genomic elements outside of the region spanned by the pair of elements in question. For example, information about boundary elements outside of the pair might still be crucial to predict the size of a domain.

4. DL approaches for increasing Hi-C resolution: Several DL approaches have been proposed with the aim of increasing Hi-C data resolution. Hi-C's high sequencing costs often lead to sparse contact maps, especially at high resolution. Tools based on Autoencoders[104] or Generative Adversarial Networks (GANs)[105, 106] learn to project low resolution Hi-C data to high resolution data, yielding smooth, higher resolution contact maps. How well these methods cope with tissue-specificity especially in novel cell types without high resolution Hi-C data for training, is unclear. It needs to be characterised further how much general architectural features these methods capture and how well they can predict fine details in genomic architecture not visible in the input training data. Furthermore, the question these models pose is fundamentally different from learning the sequence determinants of chromatin

architecture and they do not allow for predicting the impact of sequence variation.

In this chapter, we describe our efforts in designing such a tool, deepC[107]. We describe our design, demonstrate the predictive power and validate the computational predictions with high resolution and highly sensitive chromosome confirmation capture assays.

3.2 Results

3.2.1 Building deep neural networks for predicting chromatin interactions at mega-base scale

3.2.1.1 Design considerations

Neural networks have been applied to predict chromatin features from DNA sequence[60, 61] & see Chapter 2). The success of these approaches paired with innovations in deep learning encouraged us to design neural networks to predict chromatin interactions while incorporating a large sequence context. Chromatin feature networks operate with DNA sequence input of ~ 1 kb. Window-to-window based chromatin feature predictors have also adapted this window size for the processing of the sequence pairs.

The size and intricate nature of chromatin architecture motivated the design of models that can incorporate a larger sequence context, while maintaining high resolution. TADs are frequently observed at megabase size. We wanted our model to be able to predict large scale chromatin domains and integrate the contributions of TAD boundaries on chromatin interactions at scale. We therefore sought to use at least a megabase of DNA sequence as input. This scale makes the spatial range of the NNs a crucial requirement. CNNs for chromatin feature prediction integrate information over an increasing sequence context by pooling layers, usually max pooling. The max pooling operation selects the best scoring match to a convolutional filter over a hyperparameter defined range along the sequence, collapsing along

the sequence dimension. Subsequent pooling layers increase the spatial range, significantly reducing the number of computations a model needs to perform which otherwise quickly become limiting. Collapsing along the sequence dimension forces models to encode position invariance, which is appealing for models of regulatory biology. For example, a TF binding site might have the same regulatory effect if it were shifted to either site. However, there might be cases where the exact position of patterns relative to each other is crucial. This may be sub motifs, co-factors or the positioning of binding sites or entire regulatory elements relative to each other. This proves a major drawback of pooling operations for large scale context aggregation for chromatin interactions. Predicting chromatin architecture requires context aggregation without losing information of motifs and elements relative positions. As an example, convergent CTCF sites with motifs of 19 bp are known to be enriched in domain boundaries[14] and are likely important patterns to recognise and identify their relative positioning.

Dilated convolutions provide a potential solution for the context aggregation problem. They have been developed for image analysis[67, 84] and since been applied successfully for modelling audio signals[83]. Dilated convolutions perform convolutional operations with gaps (see Methods). Exponentially increasing the gap sizes, the dilation rate, with successive layers leads to a rapid growing receptive field and reduced computations, while maintaining full, high resolution coverage of the input. One key drawback of dilated convolutions is the lack of positional invariance as encoded by pooling layers.

Here, we sought to utilise dilated convolutional layers as a key feature for our chromatin interaction networks. During the work on this model, Kelley et al. demonstrated that dilated convolutions can be used to predict chromatin features while taking into account wider DNA sequence contexts then before (32 kb widows)[68]. In practise, a mixture of pooling layers and dilated convolutions can be used simultaneously to achieve positional invariance and relative pattern positioning

at large scale[68] (see below).

Here we developed deepC. DeepC is described in detail in the following and has been published under Schwessinger et al. 2020[107]. All results shown here hence relate to this manuscript.

3.2.1.2 Hi-C data encoding

High resolution Hi-C data[14, 22] offers a comprehensive and detailed chromatin interaction map at bin sizes up to 5 kb. Although they can be visualised at lower resolutions, these maps are usually sparse and are therefore prone to having empty bins. Hence, we chose 5 kb as our base resolution for predicting high resolution interaction maps as a good trade-off between detailed maps and sufficient per bin coverage. We chose a DNA sequence window of 1 Mb plus one times the respective genomic bin size as model input. A deepC model trained on 5 kb binned Hi-C data uses an input of 1 Mb plus 5 kb. This sequence window is divided into bins of equal size (Figure 3.1). We use ICE[108] normalised Hi-C interaction counts as interaction frequencies. We encode Hi-C data by extracting the interaction frequencies of the triangular Hi-C map, centred on the sequence window. The target vector contains the pairwise interaction frequencies of genomic bins walking outside from the window centre. This yields a vertical zig-zag pole of interactions (Figure 3.1). We slide this sequence window over a chromosome in bin sized steps to recover the intra-chromosomal Hi-C map of all interactions up to 1 Mb distance. Longer ranged interactions as well as the inter-chromosomal context are ignored in our model. Further developments in model efficiency and encoding may provide us with the opportunity to extend deepC's spatial reach. The task of the neural network is to predict the target vector of interaction frequencies given the associated ~ 1 Mb DNA sequence window. We experimented with different frequency encodings (see 3.2.1.5).

3.2.1.3 Model architecture

The model architecture was inspired by previous chromatin features CNNs[60, 61]. We used our adapted version, deepHaem (see Chapter 2), as reference. We

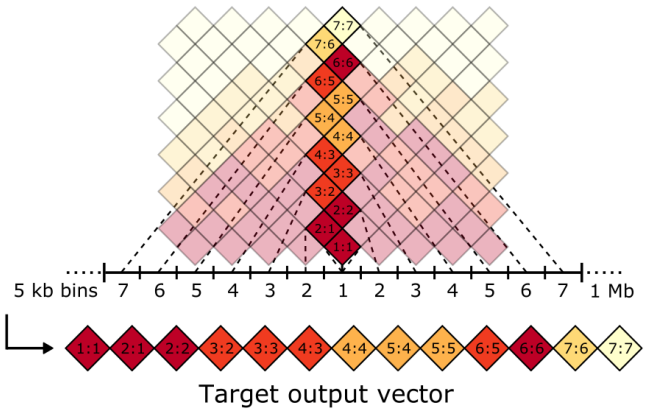

Figure 3.1: Encoding Hi-C interactions as for deep learning. Chromatin interactions over a ~ 1 Mb sized windows in bins are encoded a vertical zig-zag pole over the center. The pole forms the target vector where each entry corresponds to the pair-wise interaction frequency of two genomic bins walking outwards from the center, so that the last entry captures the interactions over 1 Mb distance. Sliding this window encoding over a chromosome in bin sized increments recovers the chromatin interaction map up to 1 Mb distance.

used five convolutional layers with ReLU activations and max pooling designed to aggregate a total of 1 kb along the sequence dimension (Figure 3.2). We then followed this max pooling module with a module of dilated convolutional layers with exponentially increasing dilation rates. We adapted the gated, residual variant of dilated convolutional layers that have been used successfully in image and audio processing[83, 84]. The gated connections allow for learning more complex pattern relationships, while the residual connections allow flexible information propagation[63]. Finally, a single fully connected (FC) layer connects the hidden units to the target vectors outputs (see Methods 6.2.2.2 for a detailed architecture description). We experimented with encoding the Hi-C data as classification and regressions problems (see Section 3.2.1.5). For the classification versions the FC layers use soft-max activations, optimising the categorical cross-entropy between class probabilities and one-hot-encoded class labels. The regression models use the linear FC output to minimise the sum of squared error between predicted and observed interaction frequency.

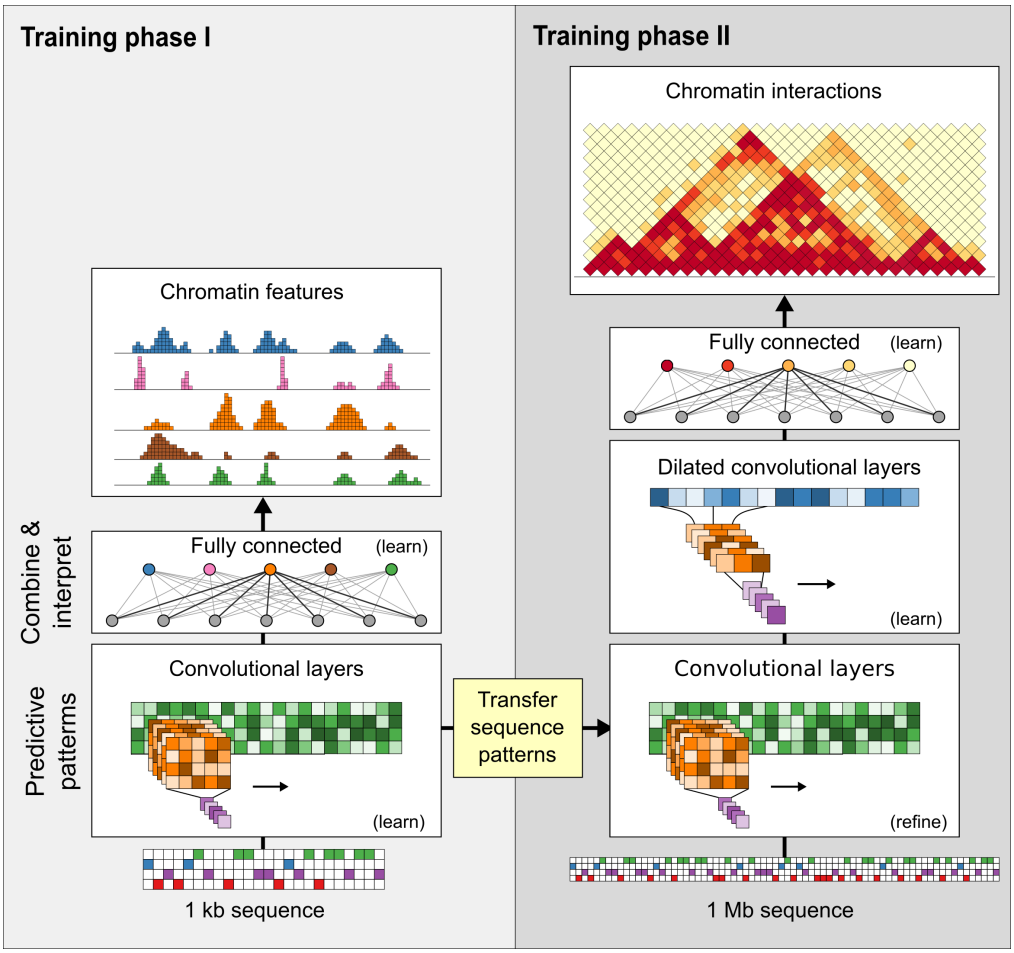

Figure 3.2: Overview of the deepC architecture and training workflow. In the first training phase a network is trained to predict presence of chromatin features (e.g. open chromatin, TF binding) on a compendium of assays and data types given 1 kb of DNA sequence as input. This network employs several convolutional layers with ReLU activation and max pooling, followed by a single fully connected layer with sigmoid activation. In the second training phase a network is trained to predict chromatin interactions on a single cell type of tissue-specific Hi-C using as input ~ 1 Mb of DNA sequence. The initial module equals the convolutional layers of the chromatin feature network. This is followed by a dilated convolutional module, using gated dilated convolutions with residual connections. The two modules are connected by a 1x1 convolution to match hidden dimensions. Finally, a fully connected layer with linear activation predicts the chromatin interactions of the target vector central over the input DNA sequence. The pre-trained weights of the convolutional layers are transferred from the chromatin feature network to the chromatin interaction network, where they are refined during the training. The remaining layers are trained *ab initio*.

3.2.1.4 Transfer learning

In our first attempts at training deepC networks we initialised all filter weights randomly using a Glorot uniform initialiser[51]. This produced coarse interaction maps where insulated domains could only be faintly noticed (Figure 3.3). The predicted interaction maps were dominated by the linear proximity gradient inherent to Hi-C data (see 3.2.1.5 for detailed discussion).
These observations suggested that the model did not capture the underlying sequence patterns sufficiently. We identified three potential bottlenecks that may prevent successful training *ab initio*: 1) The deepC's model size requires to train ~ 30 M parameters (compared ~ 10 M parameters in deepHaem). The number of layers and parameters significantly dilutes the gradients slowing down parameter updates. 2) The combined model and input size led to large requirements of GPU memory (~ 10 GB) which in turn limited us to using a batch size of 1. Larger batch sizes would provide more robust parameter updates. 3) Even though we use the entire genome, except for hold out chromosomes, for training, the number of training examples is limited. Splitting the hg19 reference genome into ~ 1 Mb windows with 5 kb increments yields ~ 550 k examples from which test and validation hold out chromosomes need to be split (compared to ~ 4.5 M training examples for deepHaem).

Looking for potential approaches for a faster and more robust training procedure we turned to transfer learning. Transfer learning, developed for image recognition[109] seeks to re-use basic patterns learned on a different, perhaps easier task and refine them for a novel task. For example, a CNN image classifier trained to discriminate between dogs and cars may be re-purposed to classify boats and cats. Recognising the basic image patterns the CNN has already learned, such as straight lines, curves or textures can be useful for the new task. The new task may require different weighting of the learned patterns, which can be achieved in the refinement training. This usually involves training a new FC layer while using the same convolutional

layers with the pre-trained weights as in the original network.

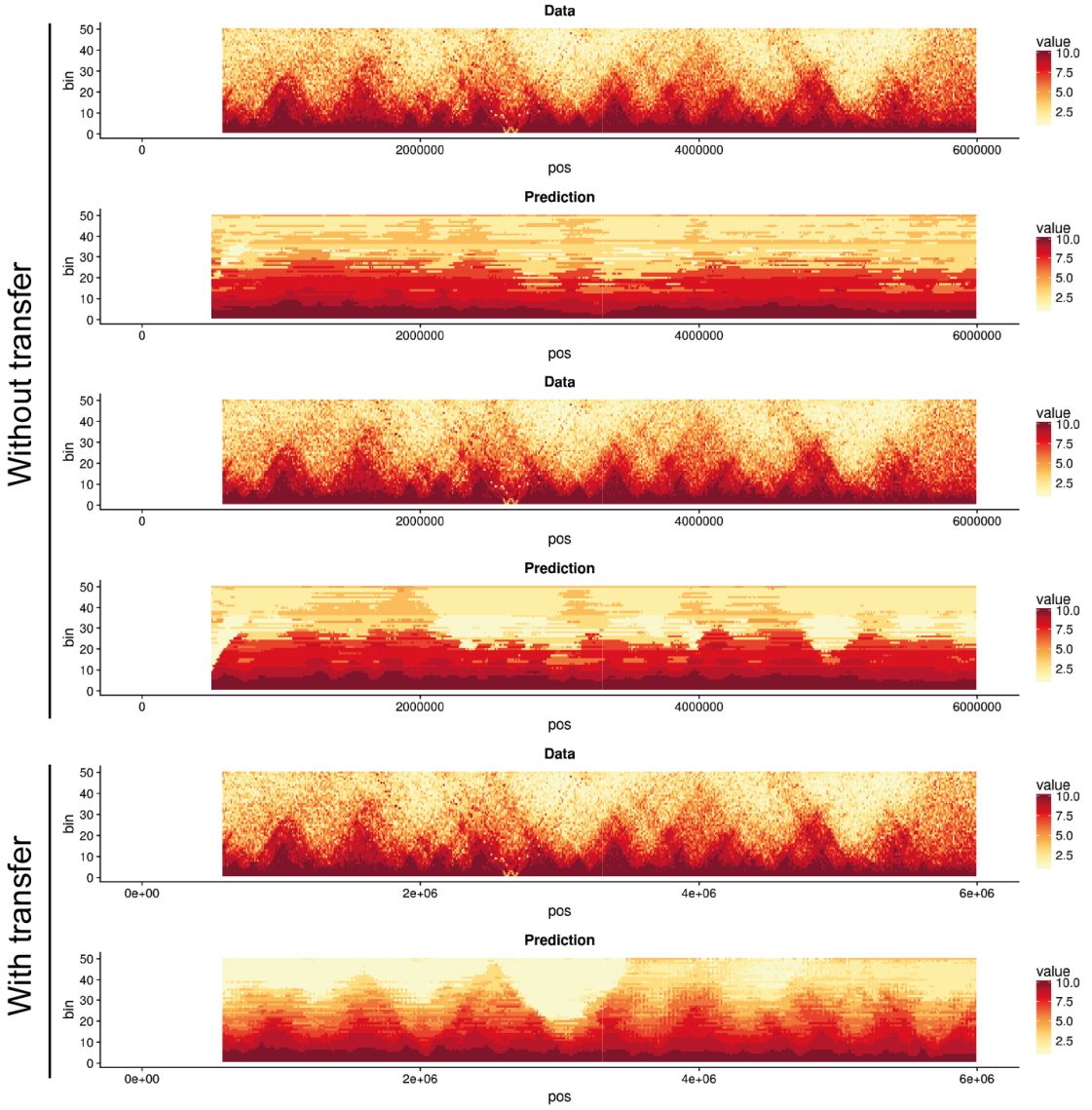

Figure 3.3: First deepC attempts encoding Hi-C frequencies as classification problem and predicting class identities without distance normalisation. Shown are the same Hi-C data for reference paired with and predictions on a hold out chromosome. Using transfer learning improves predictions. Overall, predicted maps are coarse and dominated by the linear distance signal. Differences in the examples without transfer learning are variations of hyperparameters such as learning and dropout rate.

We adapted transfer learning to genomic CNNs. We and others successfully trained CNNs for identifying chromatin features such as accessible chromatin and TF binding, which in turn are associated with regulatory elements, such as

promoters, enhancers and boundary element. We hypothesised that the DNA sequence patterns useful for identifying chromatin features, would also be useful for predicting chromatin interactions. Rather than only replacing the FC layer we also added an intermediary dilated convolutional module for aggregating more complex patterns over a larger sequence context. For employing transfer learning, we matched the architecture of the initial convolutional module with max pooling of the chromatin feature neural network (deepHaem), allowing one-to-one transfer for filter weights and biases. First attempts at using transfer learning yielded smoother interaction predictions with insulated domain structures emerging more prominently (Figure 3.3). However, the proximity gradient was still dominating the predicted interaction maps.

3.2.1.5 Regression of distance normalised percentiles

After our initial experiments and the adaption of transfer learning, two issues remained prominent in the predicted interaction maps: 1) The predictions were still fairly coarse grained and structured domains were only faintly defined. 2) The interaction gradient derived from the linear genomic distance was still the most prominent feature learned. Independent of higher order structures, genomic segments on a chromosome interact with each other with a frequency that decreases exponentially with increasing linear distance on the DNA strand. This is an emerging property of the DNA polymer. The linear distance gradient is apparent in all Hi-C data. Distance normalisation techniques such as the observed-vs-expected ratio have been developed, for which the observed interaction frequency is divided by the frequency expected from DNA polymer. The expected frequency is usually approximated by the mean chromosome- or genome-wide interaction frequency per linear distance in equal bins[13].

To address both problems, we reworked the prediction task and developed a distance normalisation strategy for encoding the Hi-C interaction frequencies for deep learning. Initially, we treated the interaction frequency prediction as a classification problem.

To do this, we binned the frequencies in percentiles and then predicted the frequencies from predicted class identities. But interaction class identity encoding yielded coarse and discrete predictions. The models were only able to predict integer values, limiting their resolution. Furthermore, the model cannot easily learn relations between frequency bins, for example that percentile 10 has a higher interaction frequency than 9. We thus switched to predicting the bin number (from 1 to 10) as a regression problem, optimising the sum of squared error between the fully connected, linear output and the bin number. This approach still discretises the observed interaction frequencies but allows the model to learn the relationship between interaction strengths and to predict real valued interaction strengths.

Next, we addressed the distance dependence signal by normalising the interaction frequencies for linear distance. We binned the interaction frequency data in a distance stratified manner (Figure 3.4 & see Methods 6.2.1.2). To emphasise differences between high interaction frequencies, as observed within domains and at domain boundaries, we designed the percentiles at uneven size. Using 20 % percentiles at the lower end and 5 % percentiles at the upper end of observed interaction frequencies (Figure 3.4). This normalisation, referred to as skeleton transformation, effectively removes the distance dependence gradient and highlights the domain boundaries (Figure 3.5).

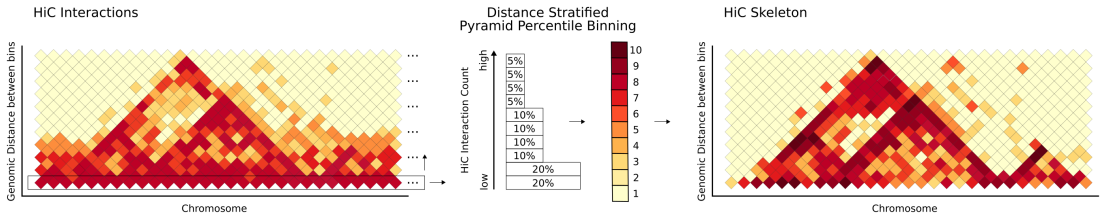

Figure 3.4: Distance stratified, pyramid percentile normalisation. Hi-C frequencies are binned into uneven percentiles, with a decreasing size towards higher interaction frequencies. The binning is done separately for every linear distance across entire chromosomes. The resulting skeleton highlights interaction domains boundaries and has a higher resolution at the higher interaction frequency spectrum. Bins are assigned numbers from 1 to 10 which serve as targets for the prediction task.

The combination of transfer learning, skeleton transformation yielded smooth and intricate predictions (Figure 3.5). Predictions with this set-up truthfully identify hierarchical chromatin structures at high resolution. Transfer learning was still crucial in this setting. Models trained without transfer learning predict intricate and smooth chromatin structures but fail to replicate the observed structures and their precise locations.

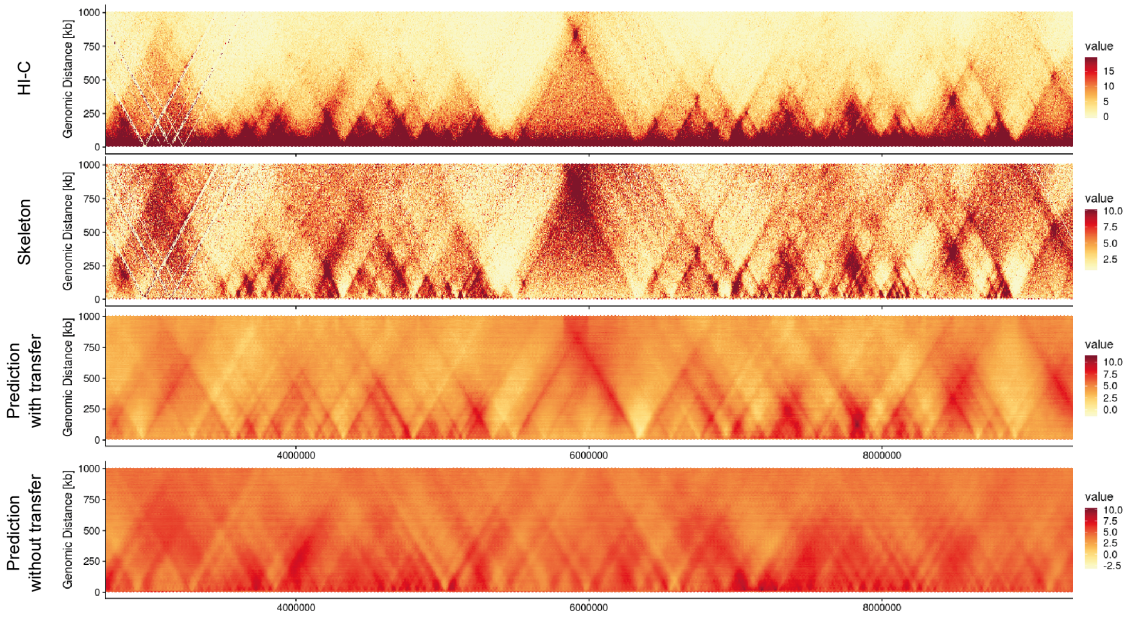

Figure 3.5: Prediction with transfer learning, distance normalisation and regression approach on hold out chromosome 17 in GM12878 cells. The regression approach leads to smoother and fine-grained predictions. The distance normalisation removes the linear distance gradient. With transfer learning the model accurately predicts larger and hierarchically folded genome structure. Without transfer learning the model predicts domain structures inaccurately.

The skeleton normalisation has the drawback of losing information when attempting to fully reconstruct the equivalent Hi-C maps due to the removal of the proximity signal. Alternative distance normalisation methods such as the observed-vs-expected ratio have minimal reconstruction losses and so perform better in this aspect. Indeed, in parallel work, Fudenberg et al. trained a CNN on $log_2(observed/expected)$ ratios[110]. We attempted to use the same metric to train deepC models. However, in our framework, models trained on this metric did not learn to predict chromatin structures, but only learned to predict chromosome-wide mean interaction values

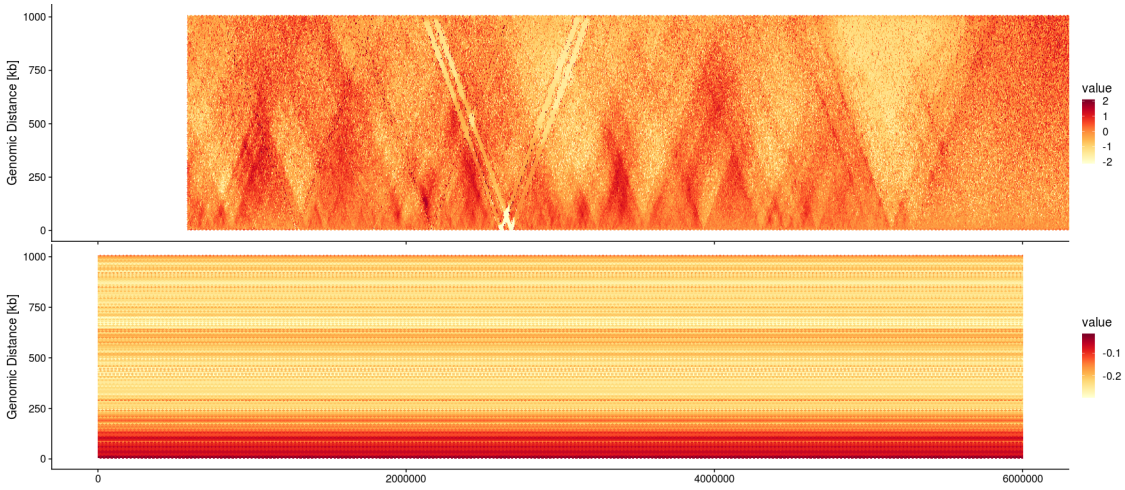

Figure 3.6: Predictions using log observed over expected ratio encoding. The model predicts a single interaction value per linear distance. The scale is small, likely reflecting residual frequency variation from the distance normalisation.

apparent as horizontal stripes in the predicted map (Figure 3.6). The predicted mean interactions are much smaller than the linear proximity gradient in the Hi-C data before normalisation. This may point to a residual distance dependent signal still in the $log_2(observed/expected)$ ratio map. The distance normalisation is employed to remove the distance dependence signal in the Hi-C data to prevent the NN training from getting stuck in local minima that correspond to the chromosome-wide mean interaction values per linear distance. If the variation in the remaining mean interactions is large relative to the variation observed in the normalised interaction map, the networks training is more likely to get stuck in such a local minimum.

Given the promising initial results we chose to use the regression of uneven percentile bins as the default set-up for deepC and our subsequent work. Future work on consolidating alternative architectures and data encodings may be beneficial for improving chromatin interaction predictions.

3.2.1.6 Predictions in different settings

We trained deepC models for seven different human cell lines[14] (Figure 3.7) and three mouse cell lines[22] (Figure 3.8). We observed convergence and detailed predictions on hold out chromosomes for each model. For a detailed discussion of

tissue-specificity please refer to section 3.2.1.8. We trained deepC models at different Hi-C data resolutions: 1 kb (not shown), 5 kb (see Figure 3.5 & subsequent plots) and 10 kb (see Figure 3.7). Visualising even the deeply sequenced GM12878 Rao et al. Hi-C data at 1 kb resolution yields many empty genomic bins that render the Hi-C maps difficult to assess visually and algorithmically for example with domain callers. As a good trade-off between resolution and consistent coverage, we conducted our primary work on 5 kb models. For detailed downstream analysis, we focused on the human GM12878 and K562 models, because they were both trained on deeply sequenced Hi-C data, GM12878 (~ 3.6 B reads) and K562 (~ 1.3 B reads), allowing us to compare deepC to Hi-C data with state-of-art depth. Furthermore, the K562 cell type was of particular interest to ultimately integrate deepC into our pipeline for interpreting GWAS variants and apply to associations in the erythroid lineage. To compare predicted Hi-C data from the additional cell types that were less deeply sequenced we fell back to 10 kb resolution (Figure 3.7). After reassuring ourselves that deepC models generalise to hold out chromosomes, we performed all down stream analysis genome-wide including trainig, test and validation chromosomes.

3.2.1.7 Evaluation

Our final deepC architecture appears to faithfully predict detailed chromatin interaction patterns from DNA sequence. To quantify their accuracy and generalisability we evaluated the correlation between predictions and observed, skeleton transformed Hi-C data on the chromosomes hold out from training. Calculating the plain Pearson correlation between interaction frequencies from Hi-C data has been shown to lead to inflated estimates. The linear distance driven interaction gradient yields larger numbers at smaller linear distances. Correlation measures between raw Hi-C maps thus yield large correlation coefficients, even if the higher order chromatin structure is captured poorly[111].

To circumvent this problem one may apply distance normalisation methods and calculate correlations on the transformed data. Alternative, correlations can be

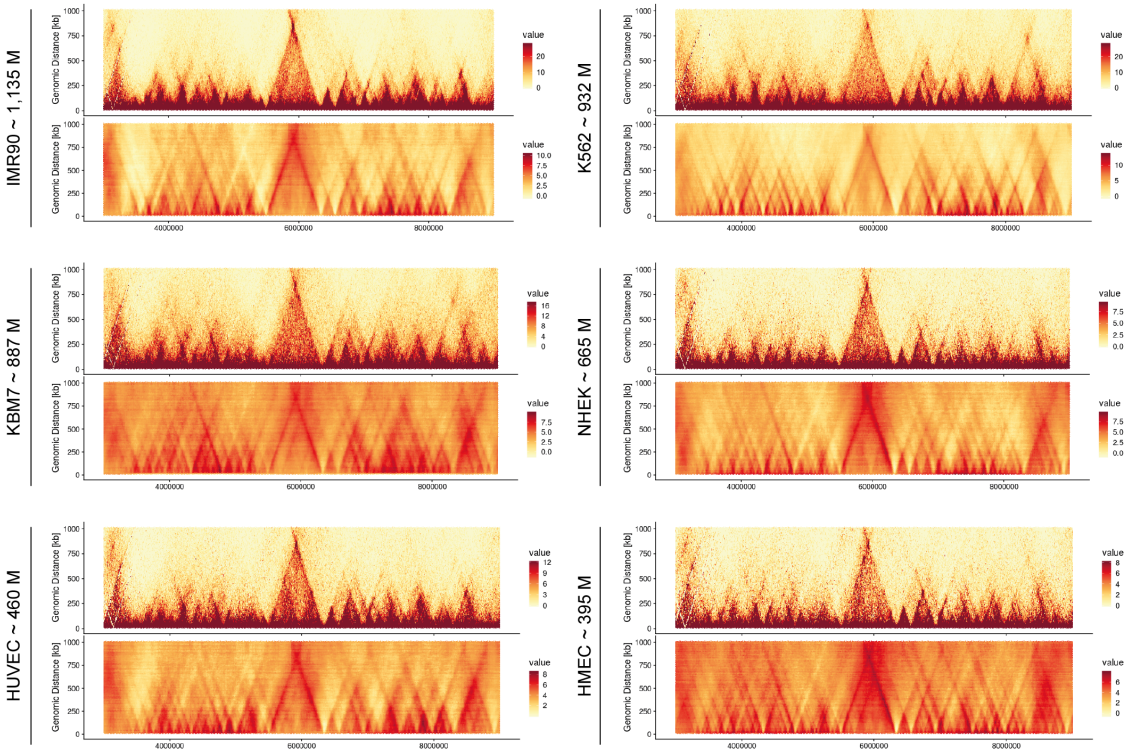

Figure 3.7: Prediction at 10 kb res on six human cell line. Hi-C data with different sequencing depth. Shown are pairs of Hi-C data and deepC predictions on hold out chromosome 17. Listed with each cell type are the number of valid Hi-C interactions associated with each respective Hi-C data set. The number of valid Hi-C interactions depends foremost on the sequencing depth. DeepC accurately predicts chromatin interactions across a range of sequencing depths.

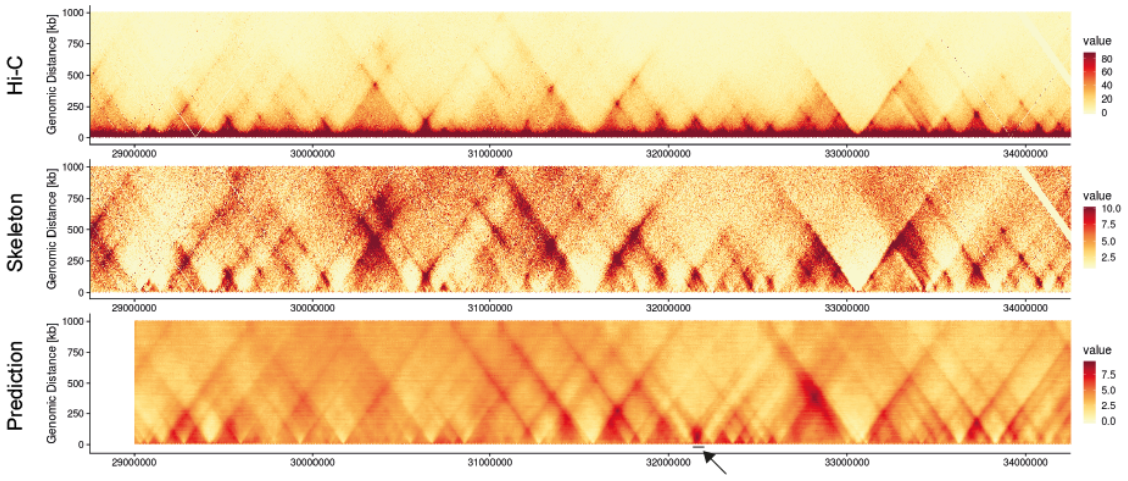

Figure 3.8: DeepC predictions of mouse ES cell Hi-C data on hold out chromosome 11. Shown are 5 kb resolution predictions. The black line indicates the *HBA* domain.

calculated in a distance-stratified manner[103, 111]. In our approach, the employed skeleton transformation removes the distance dependent signal. However, we hypothesised that chromatin structure might still be learned at different efficiency at different linear distances. When we calculate distance stratified Pearson correlations between the predictions and the Hi-C skeleton we observe that indeed the correlation value drops with increasing linear distance (Figure 3.9). This may be explained by the relative sparsity of chromatin architectural features at larger linear distances. At smaller and intermediate distances, domain boundaries and nested domains are more frequent, providing ample training examples for the model to learn. At larger linear distances only the pinnacles of larger TADs emerge as prominent features. Therefore, the sparsity of structural features likely makes learning at larger linear distances more difficult. Furthermore, background noise becomes more prominent over greater genomic distance which also drives down correlations. These nuances suggest that distance stratified correlations are crucial for evaluating chromatin interaction predictions, even if the data are distance normalised.

We evaluated the GM12878 and K562 models in a cross-validation scheme. Evaluating the distance-stratified Pearson correlation between predictions and skeleton on separate training runs in which the respective chromosome has not been used for training and testing (Figure 3.9). Because the Hi-C skeleton data at 5 kb resolution contain a lot of experimental noise, we additionally calculated the correlations using raw skeleton data and after applying a 5x5 mean smoothing filter to the skeleton map. We observed a mean Pearson correlation of 0.36 for the raw and 0.57 for the mean smoothed skeleton in GM12878 (0.28 and 0.51 in K562). These correlation values are comparable to alternative approaches[103] that predict chromatin interactions at 5 kb from chromatin features directly, this is noteworthy as prediction from sequence is a much harder task. The correlations varied significantly by chromosome, an effect observed by others in independent work[47, 103]. We could not connect this observation to any obvious chromosomal features such as percentage of laminate associated domains (LADs). Other work suggested that the performance on each

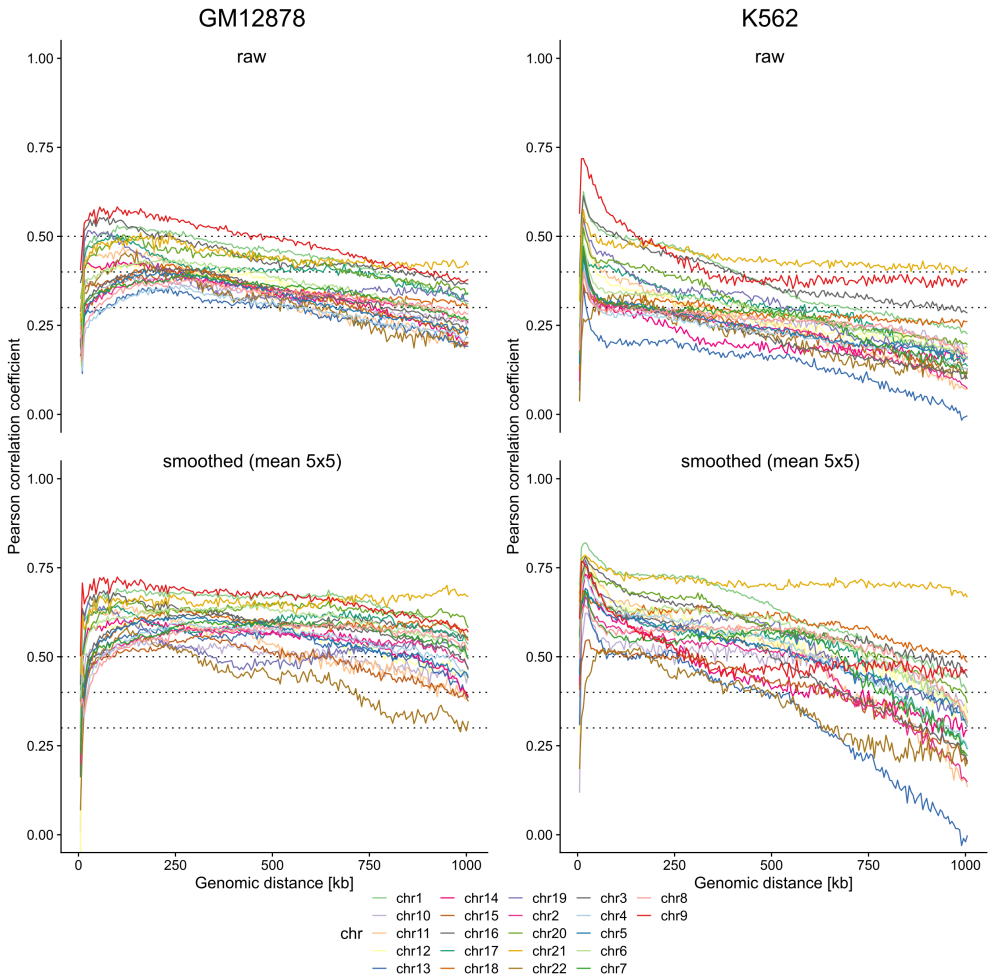

Figure 3.9: Cross-validation across all chromosomes in GM12878 and K562. Models were trained in two separate runs holding out half of the chromosomes for validation in turn. Shown are distance stratified Pearson correlation values between the predicted interactions and the Hi-C skeleton. The skeleton was used raw and after a applying a 5x5 mean filter to smooth the noisy maps.

chromosome depends on the chromosomes used for training[103]. This may point to specific similarities in chromatin architecture between chromosomes that are learned, however, detailed follow-up is needed to draw any firm conclusions.

The Rao et al. and Bonev et al. Hi-C data used in this study are sequenced deeply, achieving up to 6.9 B reads and 4.5 B valid Hi-C contacts. We trained deepC models successfully across the sequencing depths covered in Rao et al. DeepC predicts smooth and detailed chromatin interaction maps in comparison to the source Hi-C data and Hi-C skeleton, which are inherently noisy. This is likely

because even at extremely high sequencing depth, Hi-C data is under sampled compared to the true distribution of interaction frequencies, which deepC tries to predict. However, achieving comparable sequence coverage can be prohibitively expensive. We hypothesised that deepC could be trained on less deeply sequences Hi-C data. To test this, we used raw GM12878 Hi-C data from Rao et al., realigned them, and subsampled the valid Hi-C interactions to sets of 1 B, 100 M and 10 M interactions respectively. We trained 10 kb deepC models for each set (Figure 3.10. We observed that deepC is able to learn to predict chromatin architecture on hold out chromosomes for the 1 B and 100 M contact set, while the 10 M contact set appears too sparse to learn generalisable patterns. In the 100 M contact model in particular, the predictions recover more architecture features than are visible in the Hi-C skeleton. This suggests that deepC may be also used to fine map chromatin architecture not evident in Hi-C of standard depth.

3.2.1.8 Tissue-specificity

Next, we were interested in how well deepC can capture tissue-specific chromatin architecture. Successfully learning to predict tissue-specific chromatin interactions is likely to be a challenging task. The tissue-specificity of chromatin architecture itself is limited, large structures are largely consistent across tissues and variations are largely constraint to fine-grained domains or extensions of larger domains[24, 112]. We trained deepC networks for one cell type at a time using tissue-specific Hi-C data as input. We observed that these models can pick up tissue-specific chromatin architecture. Tissue-specific chromatin architecture was identified by manual comparison of the Hi-C interaction maps and predictions. Here, we show two exemplary regions that show differential chromatin architecture between GM12878 and K562 cells:

1) At the *MEIS1* locus on chromosome 2 deepC predicts a strong boundary to upstream genomic interactions forming a distinct domain in K562 (Figure 3.11),

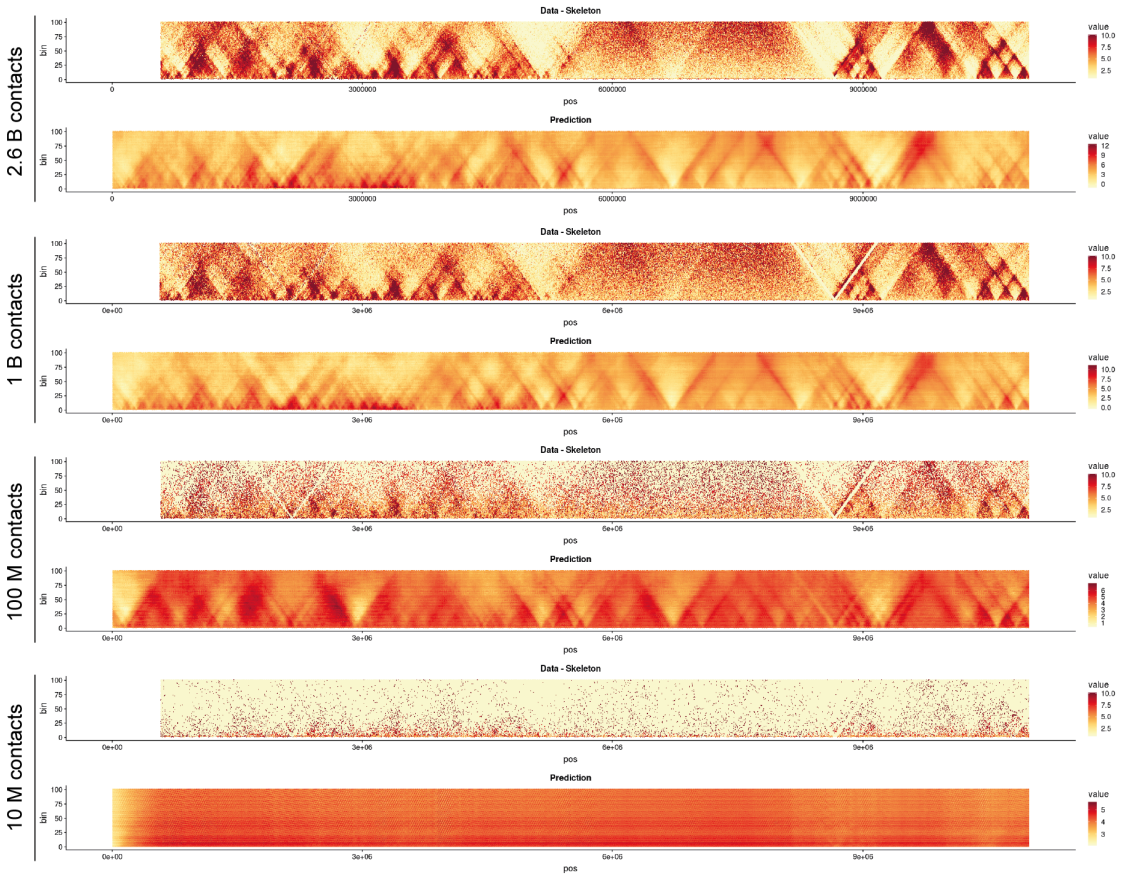

Figure 3.10: DeepC 10 kb models GM12878 using downsampled Hi-C data. Shown are pairs of Hi-C skeleton on hold out chr16. GM12878 was used raw with ~ 2.6 B valid Hi-C interactions and downsampled to 1 B, 100 M and 10 M valid interactions respectively.

in agreement with the Hi-C data. This boundary coincides with a tissue-specific CTCF site observed in K562 cells. In GM12878 cells only a very faint domain is visible both in the Hi-C data as well as the deepC predictions. Interestingly, this may point to a tissue-specific domain formed due to the tissue-specific binding of CTCF. Towards the left of the same domain, overlapping a predicted LAD, we observed that deepC predicts chromatin structures not visible in Hi-C nor Tiled-C. We observed the same artefacts in various loci with predicted LAD overlap. LADs are regions of the chromatin that associate with the nuclear lamina. LADs comprise megabase scale, dynamic regions of repressive heterochromatin that can be marked by H3K9me3 modifications and are characterised by a repressive but dynamic chromatin environment[113]. We observed that deepC frequently predicts smaller

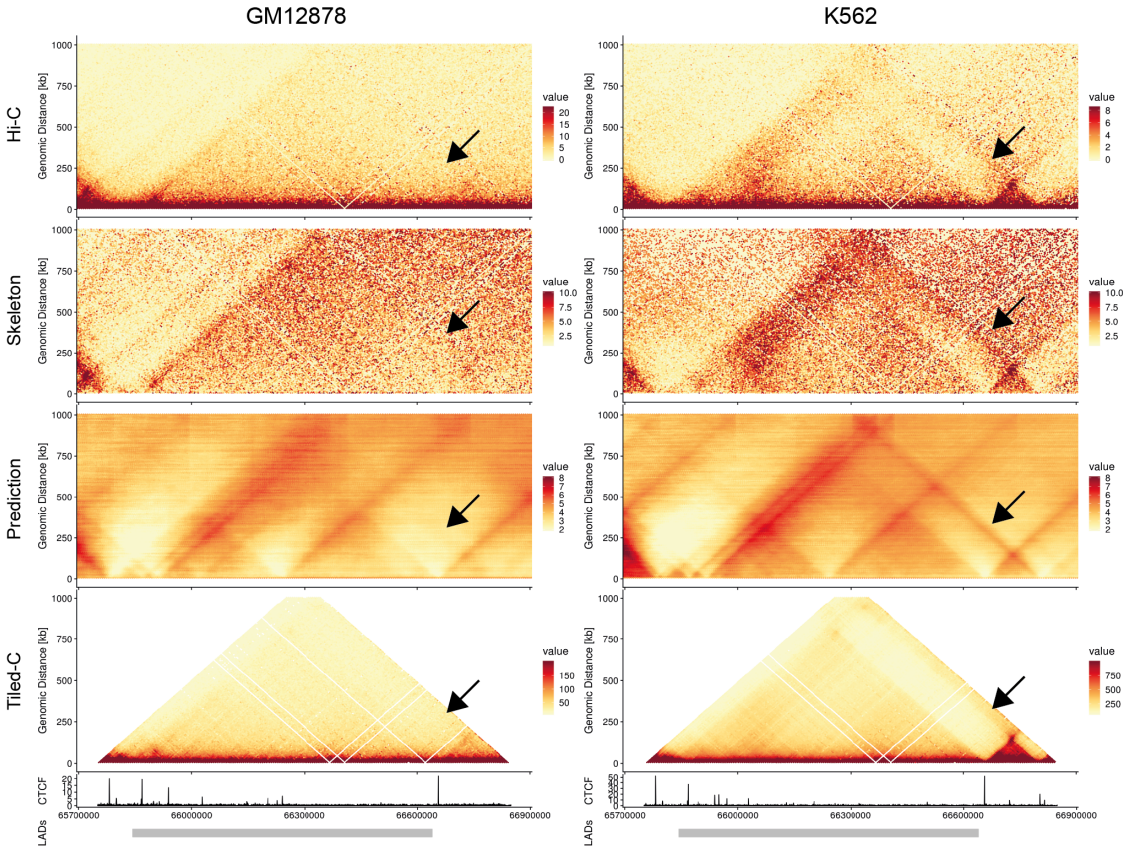

Figure 3.11: Separately trained GM12878 and K562 models predict tissue-specific chromatin interactions at the *MEIS1* locus on chr2. Shown are Hi-C data, skeleton and predictions, Tiled-C validation and CTCF ChIP-seq and fibroblast predicted LADs (gray bars). Models trained individually on the respective Hi-C data predict a stronger insulated domain in K562 (right) matching a tissue-specific CTCF binding site. The specific site leads to a boundary for upstream interactions (black arrow) to the left clearly visible in Hi-C data, skeleton, prediction and Tiled-C. Over the predicted LAD domain, deepC predicts chromatin structure that could not be validated using Tiled-C.

chromatin domains, matching the size of active compartments, underneath predicted LADs. Likely, deepC is unable to recognise LADs because the predictive DNA sequence patterns for LAD associated chromatin features are underrepresented in the chromatin feature compendium for the transfer learning step.

2) At the *CMIP* locus on chromosome 16 the K562 model predicted a distinct domain (Figure 3.12 black arrow) that was not apparent from the Hi-C data. However, targeted validation at the locus using Tiled-C (see 3.2.2 for details) confirmed the presence of a distinct domain, that is likely too subtle to be identified from the

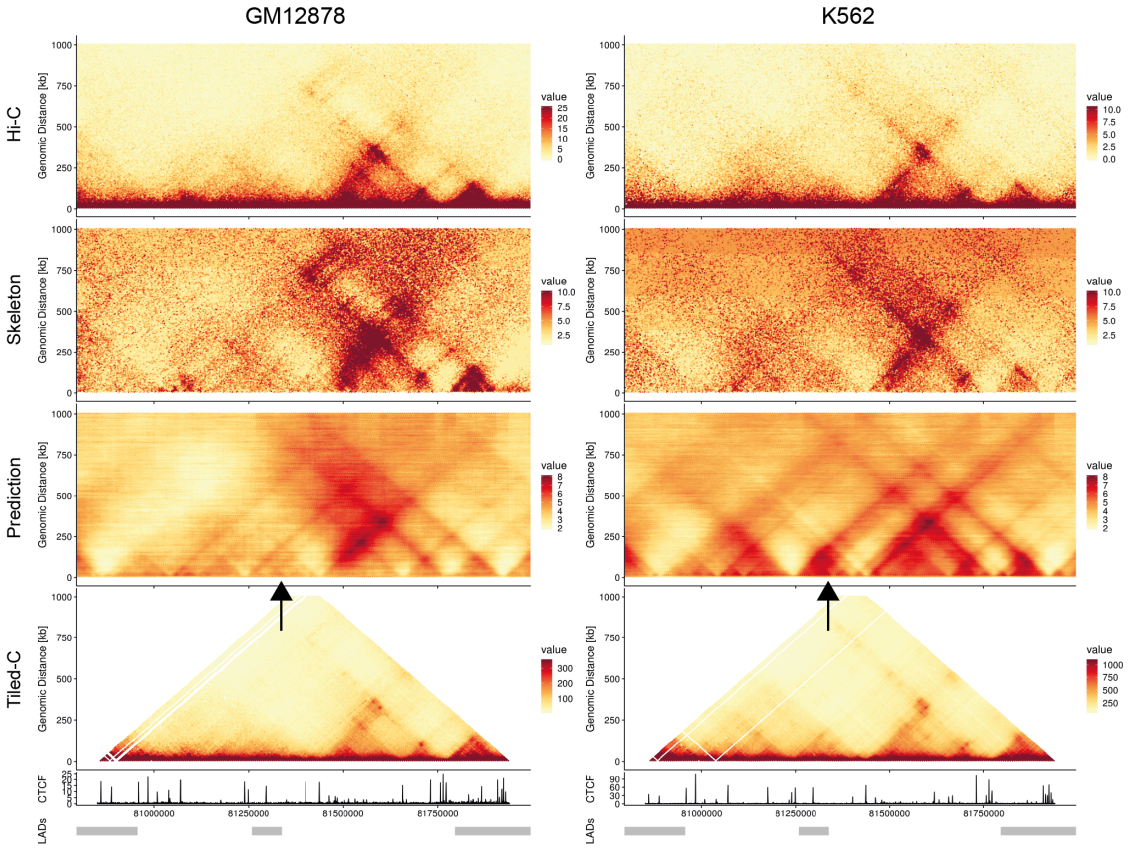

Figure 3.12: Separately trained GM12878 and K562 models predict tissue-specific chromatin interactions at the *CMIP* locus on chr16. Shown are Hi-C data, skeleton and predictions, Tiled-C validation and CTCF ChIP-seq and fibroblast predicted LADs (grey bars). Models trained individually on the respective Hi-C data predict a stronger insulated domain in K562 (black arrow) compared to GM12878. The domain is only faintly present in Hi-C data but distinctly recovered in the higher sensitivity Tiled-C validation map.

Hi-C data. However, neither the GM12878 deepC model, nor the corresponding Hi-C data show any corresponding domain at this genomic location, although the GM12878 Hi-C is sequenced ~ 2-fold deeper. These examples support both the capacity of deepC models to capture tissue-specificity as well as their capabilities to fine map chromatin architecture where Hi-C's sensitivity may be insufficient.

However, such observations of tissue-specific differences are from independently trained models for each cell type. Since the models start with randomly initialised values for the network weights that are not transferred from the initial training phase, the training processes are independent and extremely likely to reach independent

local optima. This makes quantitative comparison across tissues less reliable without developing a bespoke normalisation procedure. This caveat means that we have so far restricted ourselves to qualitative observations. Training models synchronously in a joint training procedure is a potential solution to this limitation. For example, Fudenberg and colleagues have used their model, Akita[110], to successfully train five cell types jointly. We therefore tested a model jointly trained on GM12878 and K562 data at 5 kb resolution. This joint model trained smoothly and generalised to hold out chromosomes comparably well to single tissue models. However, when revisiting the tissue-specific patterns learned in the single tissue models, we observed that the joint tissue model did not learn to predict those patterns (Figure 3.13). Rather it predicted a highly similar chromatin architecture for both cell types. This output may suggest that such joint models have only learn the tissue-invariant part of the chromatin architecture, ignoring much of the tissue-specific aspects. In agreement with this observation, Fudenberg and colleagues showed that their joint tissue model also struggled to learn tissue-specific chromatin architecture. The failure of joint models to dissect tissue-specificity may be due to the way the Hi-C data of different cell types are encoded in the deep learning models. Currently, both Fudenberg et al. and we encode the Hi-C data from different cell types as independent output vectors, or maps. While convolutional filters are shared between cell type outputs the final fully connected layers have independent weights from each other. Thus, the final interpretation of the detected sequence patterns has to be learned independently for every cell type, meaning that network has to learn to interpret shared patterns multiple times. Future advancements addressing these bottlenecks may enable us to achieve tissue-specificity in the desired joint training mode.

3.2.2 Validating computational predictions of chromatin architecture

For an initial validation of the deepC models we evaluated their accuracy on hold out chromosomes. As stated, these results suggested that deepC generalises well to unseen chromosomes and may be able to predict finer grained chromatin structure

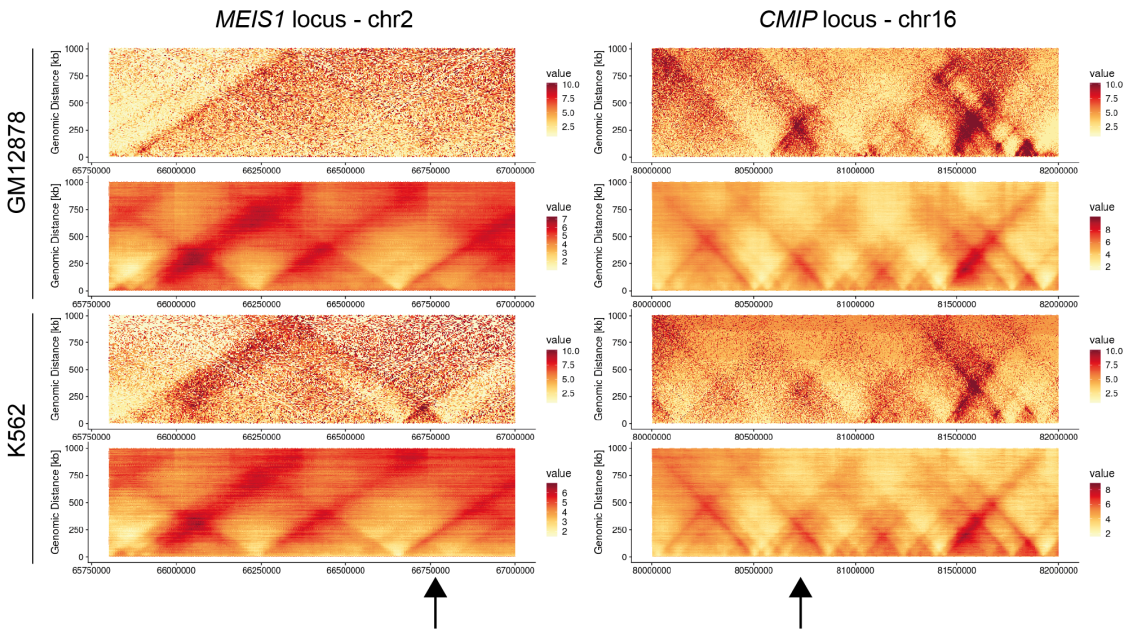

Figure 3.13: A jointly trained GM12878 and K562 model fails to capture tissue-specific patterns. Tissue-specific patterns accurately predicted by single cell type models (black arrows) are not predicted accurately by the joint model. The interaction predictions between the cell types in the joint model differ only marginally and rather represent an average structure.

than observable in the input Hi-C data. To substantiate these claims we progressed to experimental validation using focused chromosome confirmation capture techniques that resolve chromatin interactions at higher sensitivity than the used Hi-C data.

3.2.2.1 Capture-C based methods for validation

We were particularly interested in validating two aspects of deepC models: 1) How well can deepC maps predict the interactions of a genomic element or region of interest and 2) how well does deepC predict fine-grained details of chromatin architecture below the resolution of the input Hi-C maps. To address point 1) we used Capture-C from ~ 200 viewpoints to compare the experimentally determined interactions at high sensitivity to the corresponding predictions. Of these targets 50 % were designed on the hold out chromosomes. To address point 2) we performed Tiled-C[112] on six loci to detect all-vs-all chromatin interactions across specific genomic windows, resolving the chromatin architecture at higher sensitivity then achieved by Hi-C.

For the Capture-C experimental design, we targeted ~ 200 genomic regions. Targets were designed to either probe regions where deepC predicted chromatin structure at higher sensitivity than the Hi-C data or where predictions and Hi-C differed. The targets were split into two categories designed to probe different general aspects of chromatin architecture:

1. 80 regions overlapped with CTCF binding sites, as determined by CTCF ChIP-seq, that were located at chromatin domain boundaries. With the "CTCF" target we aimed to quantify how well deepC learns CTCF site anchored interactions. CTCF anchored interactions frequently manifest as relatively punctuate interactions on top of an enriched domain signal[112].

2. 139 regions were chosen to lie within insulated domains but avoiding any overlap with CTCF binding sites or accessible chromatin regions. These "intra domain" targets were designed to probe the general domain structure, showing chromatin interactions from the viewpoint of the inside of domains, rather than discrete interactions between regulatory elements.

We performed Capture-C in GM12878 and K562 cell lines to match the deepC training data. The 3C libraries were sequenced deeply to capture fine-grained interactions and to be able to resolve minute interaction frequency changes at domain boundaries from the intra domain viewpoints. We achieved on average 29 k cis interactions per viewpoint fragment in GM12878 and 148 k in K562. We distance-normalised the Capture-C profiles to make them comparable to the Hi-C skeleton and deepC predictions. When visualising the Capture-C profiles with their corresponding Hi-C data and deepC predictions we observe good agreement (Figure 3.14). In the Capture-C data CTCF viewpoints show sharp interaction peaks with other genomic regions bound by CTCF. In contrast, in the intra domain viewpoint profiles we observe more subtle interaction level drops and peaks which are generally coincident with predicted boundaries.

87

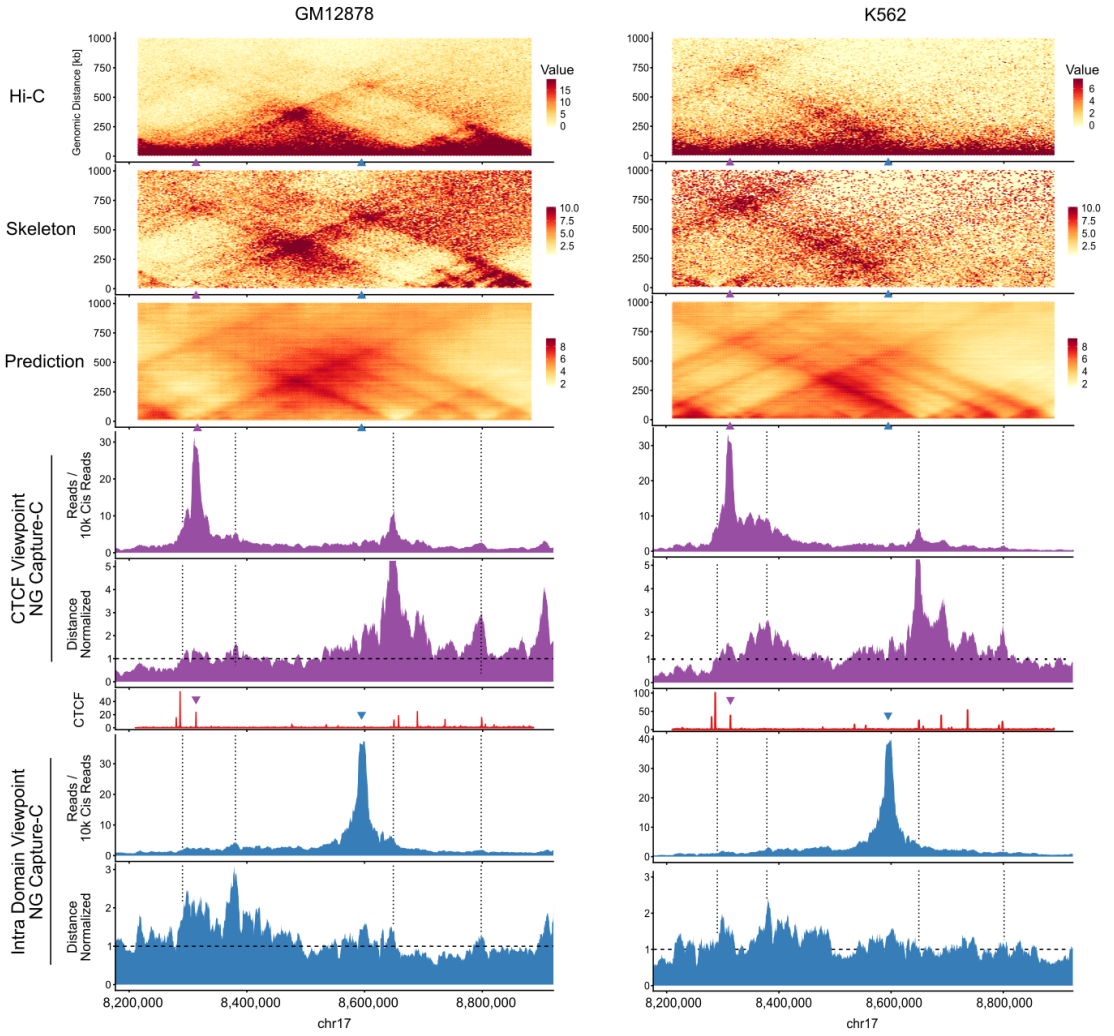

Figure 3.14: Validating deepC predictions with Capture-C in GM12878 and K562 cells. Shown is an exemplary region on hold out chromosome 17 with Hi-C data, skeleton and prediction. Capture-C was performed from a CTCF bound viewpoint (purple) and from an intra domain viewpoint not overlapping any potential active regulatory elements (blue). Capture-C interaction profiles are visualised before and after applying a distance normalisation. CTCF ChIP-seq of GM12878 and K562 cells are visualised for orientation (red). Black dotted lines highlight domain boundaries predicted from deepC at high resolution and validated by Capture-C.

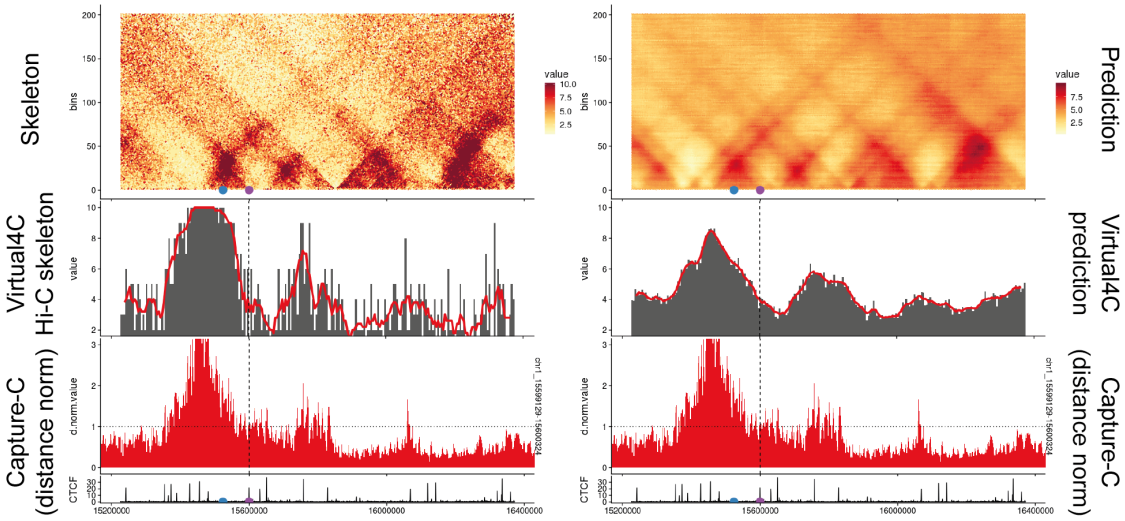

Figure 3.15: Virtual 4C profiles from GM12878 Hi-C skeleton and deepC predictions. Profiles were derived from the Capture-C probes viewpoints and correlated to the distance normalised Capture-C profile. The skeleton is discrete and noisy, and a smoothing filter was applied (red line on left). For the deepC predictions smoothing was applied for visualisation here but not for calculating the Pearson correlations.

3.2.2.2 Correlation of virtual4C profiles

To quantify the predictions agreement with Capture-C we derived virtual4C profiles from the Hi-C skeleton and the deepC predictions (Figure 3 15). Because the Hi-C skeleton is discretised from the percentile binning, we applied a small smoothing window (see Methods). The deepC predictions are already continuous and do not require this procedure.

We then correlated the virtual4C profiles with the distance-normalised Capture-C profiles and visualise them as box and violin plots for comparison (Figure 3.16). For CTCF sites the Hi-C skeleton profiles outperform the deepC predictions (GM12878: 0.59 vs. 0.30; K562: 0.55 vs. 0.37). This matches the observation that deepC tends to predict more prominent stripes at domain boundaries, rather than discrete and punctuate interactions. Moreover, deepC maps appear less highly resolved at the pinnacle of domains as compared to the map at close linear distance (Figure 3.5, 3.7, 3.8). At intra domain sites deepC performed comparable to the Hi-C skeleton (GM12878: 0.30 vs 0.36; K562: 0.46 vs 0.42). Overall, deepC predictions were closer or even better than the Hi-C skeleton in K562 cells (~ 1.4 B reads)

which is less deeply sequences than the GM12878 (~ 3.6 B reads). It is noteworthy that the sequencing depth used to generate the GM12878 data is prohibitively expensive for the majority of the field and that the K562 results demonstrate the value of using deep learning models in conjunction with standard depth Hi-C analysis. Furthermore, deepC makes these predictions from DNA sequence rather than from chromatin features and so can predict the effect of sequence changes without experimental data from the genotype under investigation.

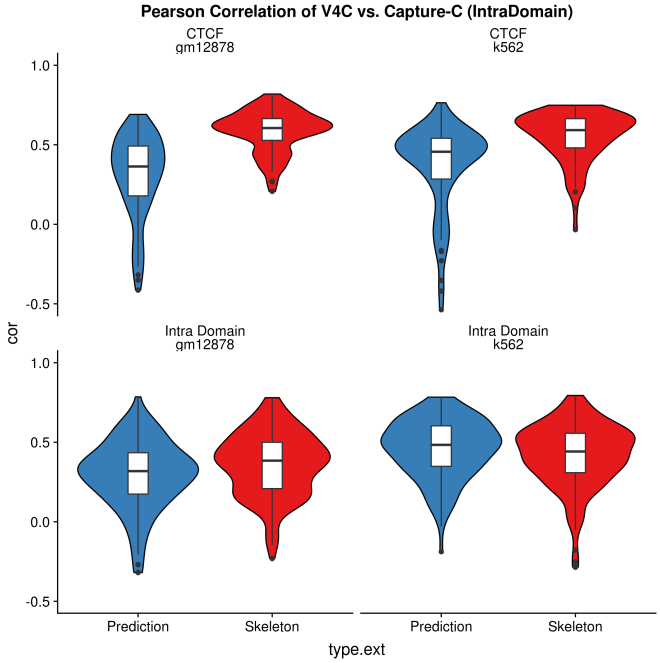

Figure 3.16: Pearson correlation of the distance normalised Capture-C profiles with virtual4C profiles from skeleton and prediction. Shown are violin plots and box plots of the separately for CTCF and intra domain viewpoints and GM12878 and K562 respectively. Box plots: median middle thick line, 25th and 75th percentile left and right hinge respectively, whiskers stretch up 1.5 times the IQR (inter quartile range).

3.2.2.3 Validating domain boundaries

Next, we sought to validate how well deepC predicts domain boundaries as these represent the key features for the general demarcation of the genome into regulatory domains. To achieve this, we first called domain boundaries from the Hi-C data, the skeleton and the predictions respectively. We adapted the established insulation score-based approached[30], which was designed for lower resolution calls, for calling

domains at higher resolution such as those identified by deepC predictions. In both GM12878 and K562 we performed insulation score-based boundary calling using parameters for high resolution calls on Hi-C data, the Hi-C skeleton and deepC predictions. as well as low resolution calling on Hi-C data for scale. A first visual assessment of the calls at exemplar regions suggested that prediction-based calls tend to capture real boundaries better, especially in the less deeply sequenced K562 data (Figure 3.17 & 3.18). To further substantiate this observation, we employed TopDom[114], a TAD caller that recently ranked highest in a review of boundary calling algorithms[31].

To quantify how well the called domains represents true biophysical boundaries to chromatin interactions in the genome, we calculated the enrichment of distance-normalised Capture-C interactions relative to called boundaries. For sensitivity we restricted this analysis to interactions within 1 Mb of their respective viewpoints. Although Capture-C interactions can be detected at much larger distances than 1 Mb, the distinction between the interacting fragment and the immediate vicinity becomes less clear, distorting the high-resolution enrichment analysis especially for the less sensitive intra domain capture. We used the high-resolution calls from the insulation score-based method and stratified the enrichment metaplots based on the boundary call source data and the sharpness of the predicted boundary (Figure 3.19). For TopDom we used Hi-C and prediction boundary calls with two window parameter settings, a higher resolution (5) and a lower resolution (20) version (Figure 3.20). In both cases we distinguished between CTCF and intra domain viewpoints. Both boundaries called based on Hi-C data and from deepC's predictions showed clear enrichment for Capture-C interactions over the called boundaries relative to the immediate neighbourhood (Figure 3.19 & Figure 3.20). Overall, boundaries called from the Hi-C skeleton performed the poorest. This is likely due to noisy nature of the unsmoothed skeleton leading to a lot of false positive calls, which in turn decreases the relative enrichment. The enrichment was stronger for sharper boundaries. Interestingly, the prediction-based boundaries

91

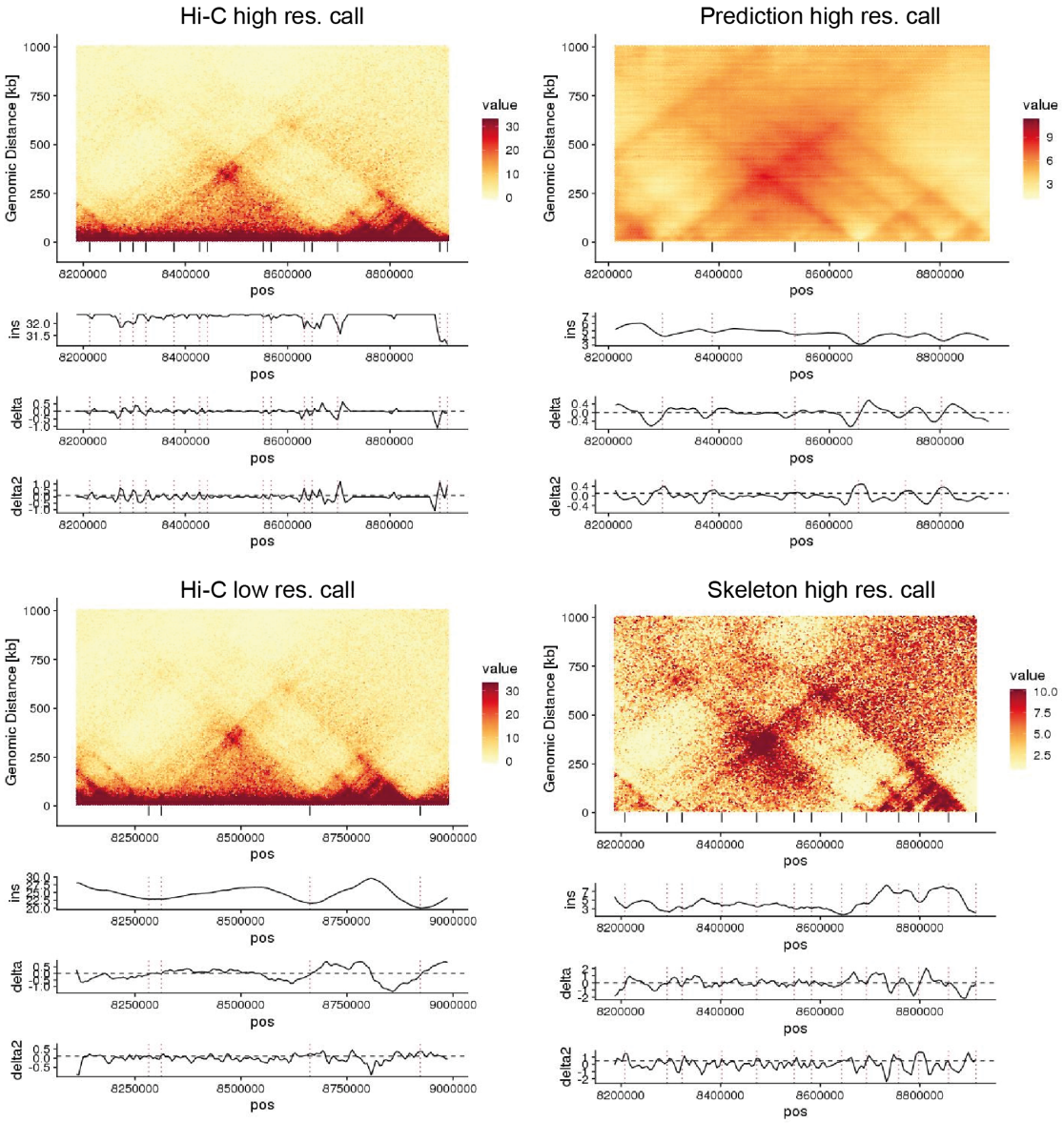

Figure 3.17: Insulation score-based boundary call in GM12878. Shown are Hi-C data, skeleton and prediction. Aligned underneath each are the respective insulation score profile and the approximated first and second order derivative (delta and delta2). Local minima in the insulation score are identified as 0 crossings in the delta profile, filtering out maxima using the neighbourhood. The sharpness of a boundary is defined by its delta2 profile which is used to threshold and stratify called boundaries, A high resolution and a low-resolution threshold was used.

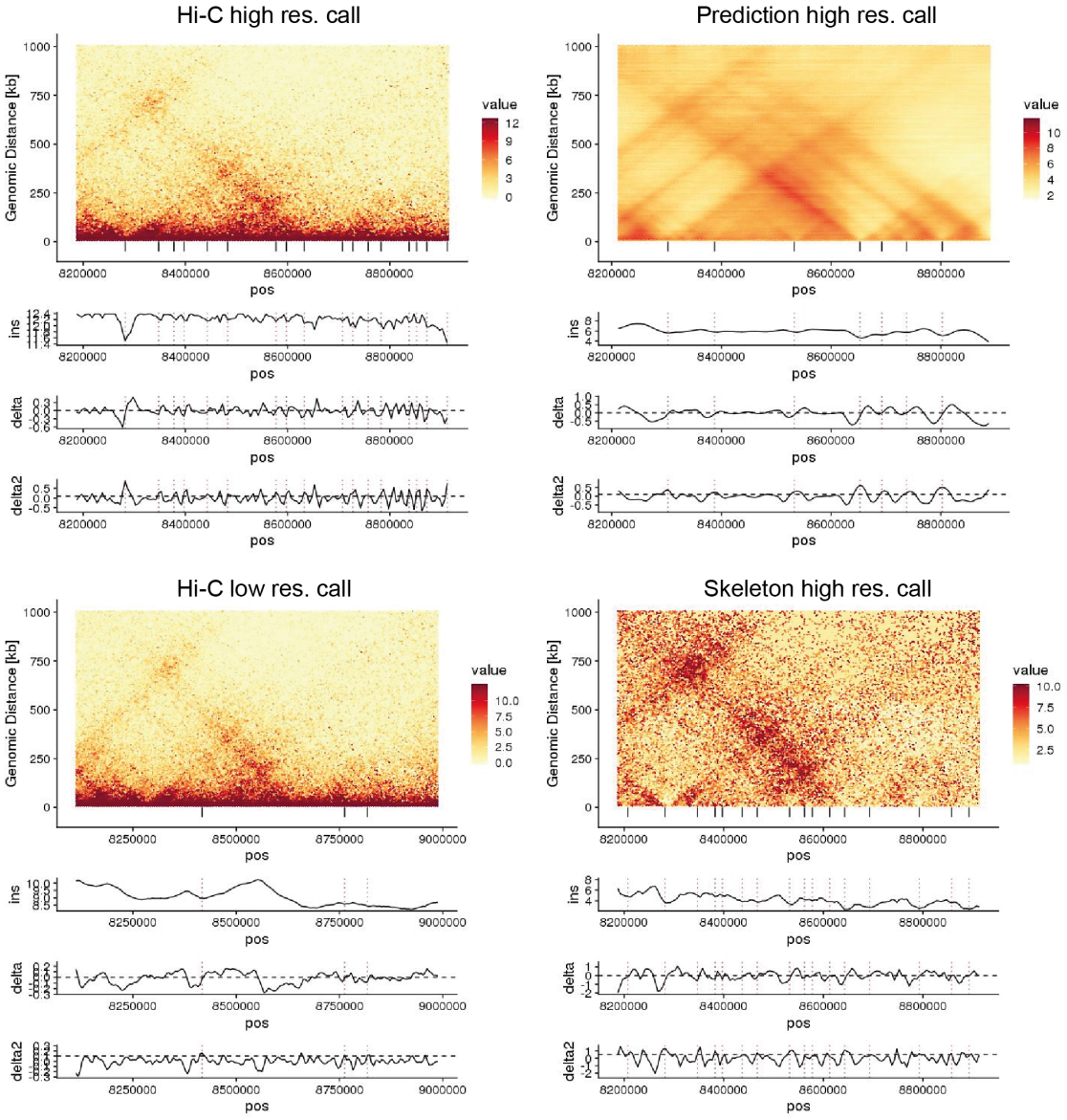

Figure 3.18: Insulation score-based boundary call in K562. Visualisations as described in Figure 3.17. The K562 data is less deeply sequenced and high-resolution boundary calling overcalls boundaries on the Hi-C data. Calling from the smooth predictions appears more accurate.

93

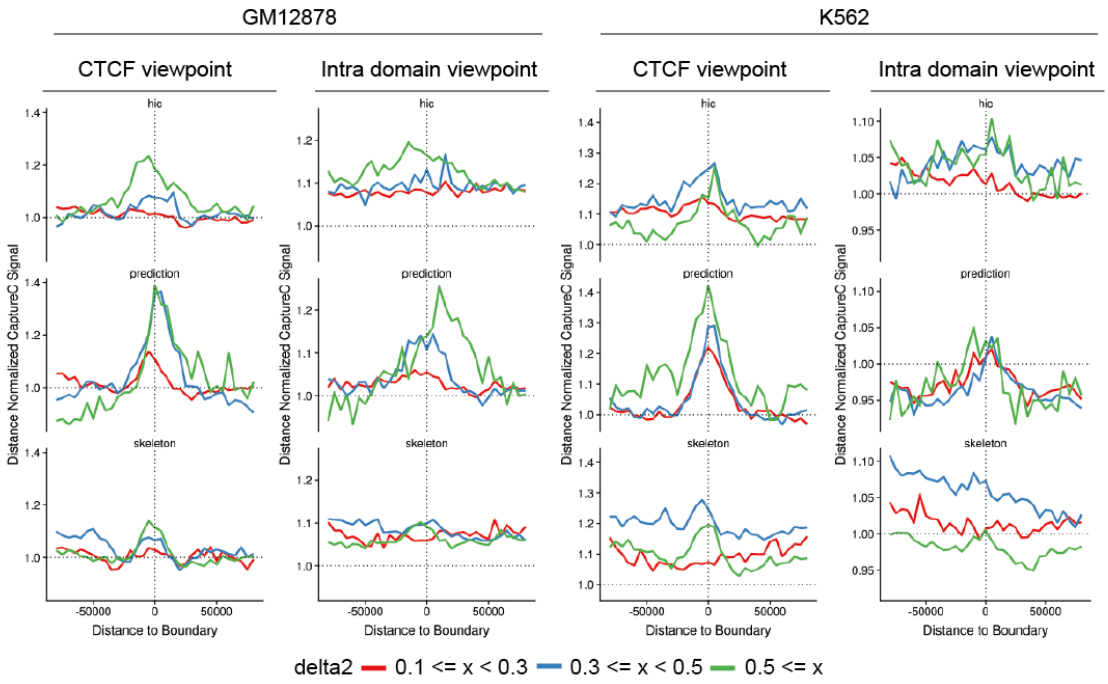

Figure 3.19: Meta-profiles of distance normalised Capture-C interaction over boundaries called using the insulation score. Capture-C interactions within 1 Mb of a Capture-C viewpoint are counted relative to the respective boundary center. Profiles are stratified based on cell type and Capture-C viewpoint point type as well as on the boundary sharpness (delta2) increasing from red over blue to green.

showed a consistently stronger enrichment centred on the boundary relative to the surrounding window as compared to the Hi-C based calls (Figure 3.19). This analysis suggests that deepC predictions can be used to call bio-physically meaningful boundaries at high resolution, which are frequently not evident in standard Hi-C data used to train the network.

To evaluate if deepC can indeed predict chromatin structure at higher resolution than the input Hi-C data we performed targeted Tiled-C. Tiled-C enriches all-vs-all chromatin interactions for focused windows of interest achieving sensitivities approaching the predictions of deepC but at an affordable sequencing depth. Tiled-C therefore allowed us to evaluate deepC's high sensitivity predictions at an all-vs-all map scale and to place some of the called boundaries back into a Hi-C context. We selected six regions for in detail characterisation where deepC resolved interactions at high sensitivity that were poorly detected in Hi-C (Figure 3.21). These highly sensitive Hi-C like interactions maps showed that predicted interactions which were

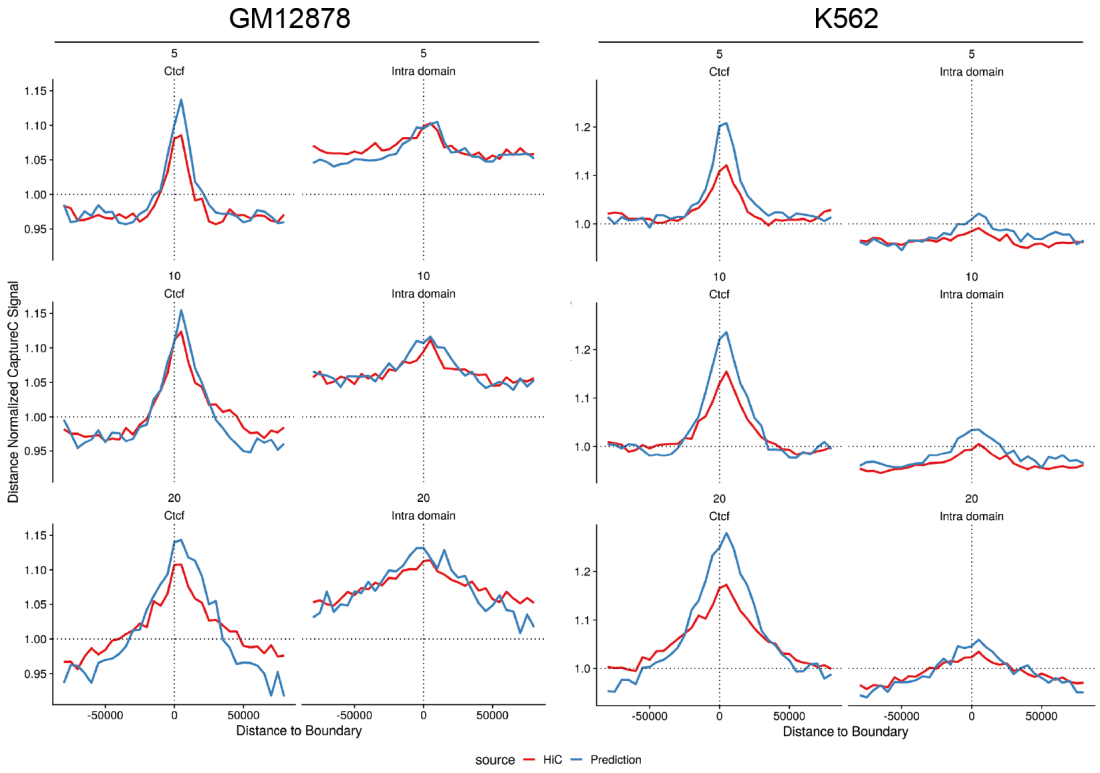

Figure 3.20: Validating distance normalised Capture-C interaction over boundaries called with TopDom Boundaries were called on Hi-C data and predictions. Profiles are stratified on cell type, viewpoint type and TopDom resolution parameter (window – w).

not evident in the standard Hi-C could be verified with Tiled-C 3C data. This was particularly striking in the less deeply sequenced K562 data. The predicted chromatin domains aligned better with Tiled-C resolved domains than the ones derived from the Hi-C data directly both when using the insulation score-based approach and TopDom (Figure 3.21).

3.2.3 Comparison of sequence-based prediction with chromatin feature based predictions

To compare deepC to currently available models, we opted for HiC-Reg[103]), a recently published tool that uses random-forest models to predict chromatin interactions from observed chromatin features such as accessible chromatin, TF binding and histone modifications. HiC-Reg predicts the interactions between pairs of genomic windows but fully integrates the window in between. This allowed

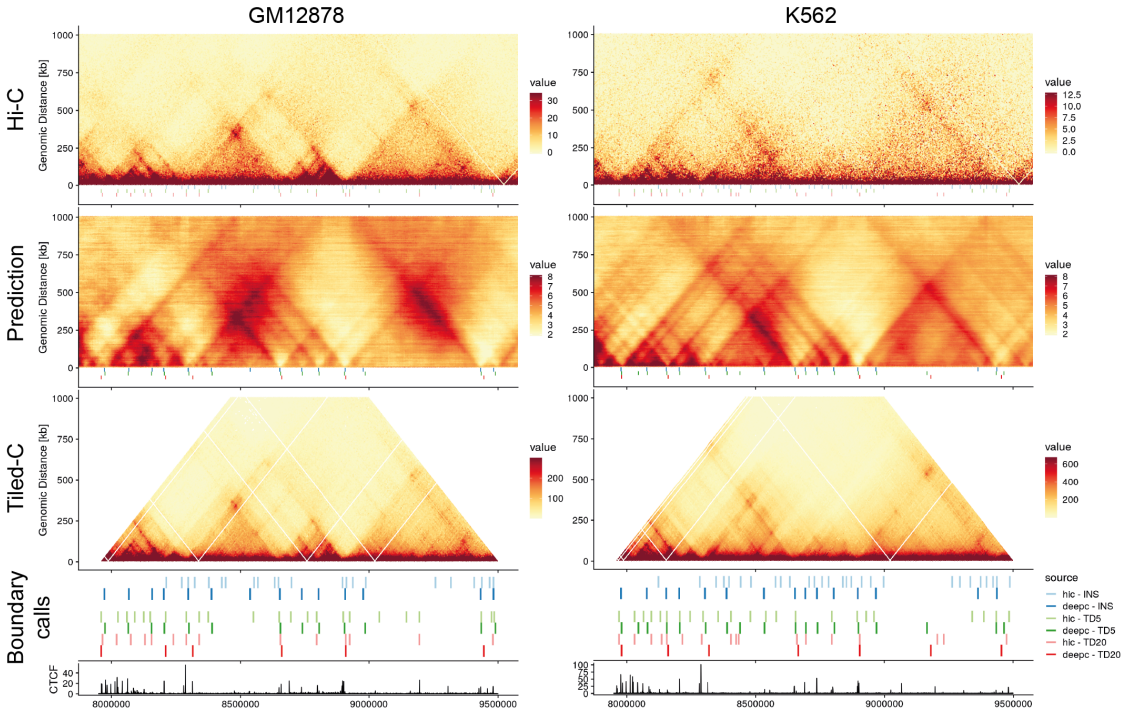

Figure 3.21: Comparing boundary calls on Hi-C data and deepC predictions. Boundary calls in Hi-C data, predictions and Tiled-C validation. Boundaries called using the high-resolution insulation score-based method (blue) and using TopDom at high resolution (green) or lower resolution (red) are aligned underneath. CTCF ChIP-seq is visualised for orientation in black.

HiC-Reg to be among the first approaches to predict and display comprehensive interaction maps in the style of Hi-C data, which facilitates direct comparison with deepC predictions. We sought to compare deepC's sequence-based predictions against HiC-Reg's chromatin feature based ones. HiC-Reg models are trained on a single chromosome and can then be used to predict interactions on hold out pairs on the same chromosome or on unseen chromosomes in "cross-chromosome" mode. To compare the generalisability of the two approaches we compared HiC-Regs cross-chromosome predictions against deepC predictions on hold out chromosome 17. The published HiC-Reg model was trained on chr14. We downloaded the cross-chromosome predictions associated with the HiC-Reg manuscript. To make the predicted maps readily comparable visually, we normalised the HiC-Reg predictions with the skeleton transformation. We downloaded, processed and visualised Hi-C data, HiC-Reg predictions, HiC-Reg skeleton, Hi-C skeleton and deepC predictions for GM12878 (Figure 3.22) and K562 cells (Figure 3.23). We observed that deepC

generalises better to the hold out chromosome and predicts the structure of large and of fine-grained domains. HiC-Reg appears to capture bigger domains but fails to accurately predict the finer structure both in GM12878 and K562 data (Figure 3.22 & Figure 3.23). To quantify this chromosome wide we calculated the distance stratified Pearson correlation between the Hi-C data and HiC-Reg or deepC predictions respectively and between deepC and HiC-Reg (Figure 3.24). We observed that HiC-Reg predictions showed very strong correlations with the Hi-C data across linear distance and in both cell types. Upon further investigation, we found that these correlations are inflated by regions where the Hi-C data has zero interactions. After removing all zero interaction regions, the HiC-Reg performance dropped drastically, while deepC's correlation with the Hi-C remained similar. We hypothesised that HiC-Reg ability to predict zero interactions may stem from using chromatin features as input as any mapability issues leading to gaps in the Hi-C data will most likely also be present in the ChIP-seq input data as well. We argue that predicting zero counts and mapability issues are less important than predicting the chromatin architecture characterised by high interaction frequencies correctly. In summary, deepC outperforms HiC-Reg in the hold out chromosome prediction task, while using DNA sequence as input, which is arguably a considerably harder task.

97

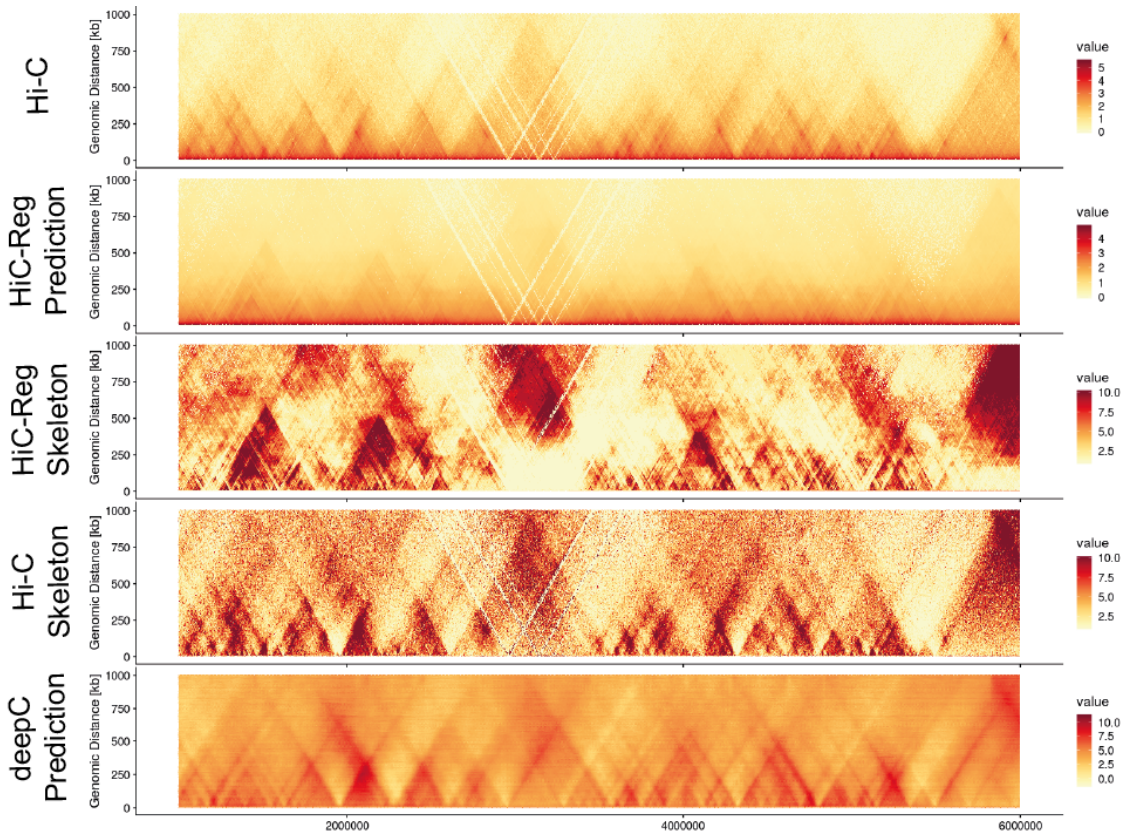

Figure 3.22: Comparing deepC and HiC-Reg predictions in hold out chromosome 17 in GM12878. Shown are Hi-C data, HiC-Reg predictions, HiC-Reg predictions skeleton transformed as well as the skeleton and the deepC predictions. Cross chromosome 5 kb predictions with a model trained on chr14 were downloaded from the HiC-Reg repository and visualised.

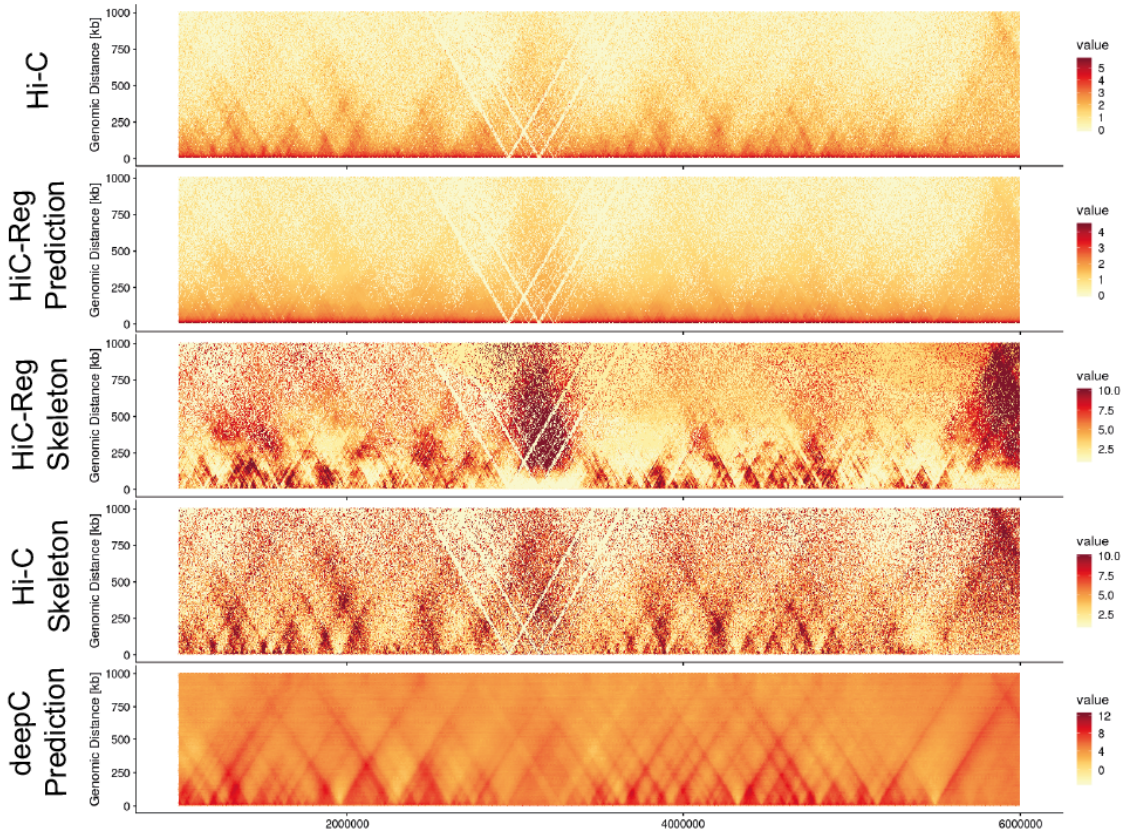

Figure 3.23: Comparing deepC and HiC-Reg predictions in hold out chromosome 17 in K562. Same as 3-22 but using K562 data and models. The less deeply sequenced data is much sparser.

99

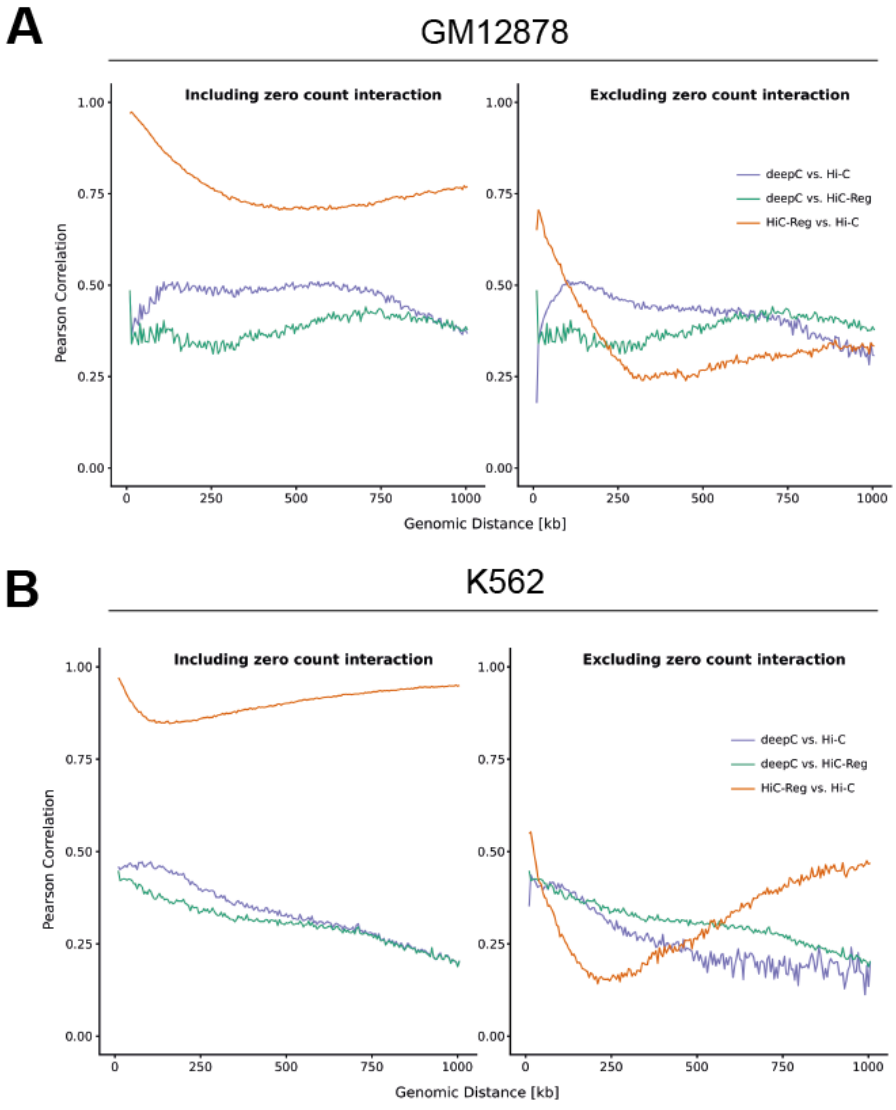

Figure 3.24: Distance stratified Pearson correlation between HiC-Reg and deepC predictions and the Hi-C data. HiC-Reg cross chromosome predictions were used with and without removing zero interaction count windows.

3.3 Discussion

To comprehensively predict the effect of non-coding sequence variants on gene regulation, we need to interpret the variant and its potential local effects in the three-dimensional genomic architecture that determines the interactions with other genomic elements. Machine learning approaches have shown promise in predicting the effects of variants locally with a sequence context of ~ 1 kb but they have not been extended to predict chromatin structure from more than two local windows at a time[100, 101]. Yet, the nature of chromatin architecture and the likely mechanisms driving it suggest that window-to-window based approaches will neglect critical information in the excluded regions between the windows. Rather a comprehensive integration of context over large genomic windows appears necessary for accurate predictions. Methods that predict genome folding by integrating a wider genomic context do so by coarse grained segregation into genomic features[47, 103] or polymer beads[97, 98]. This precludes the ability to map the sequence determinants of genome folding and to predict the impact of sequence variation at high resolution.

Here we developed deepC[107], a neural network approach that accurately predicts genome folding from DNA sequence at the megabase scale. By integrating a sequence context of ~ 1 Mb at base pair resolution, deepC bridges the base pair to TAD scale, allowing us to the predict the effect of sequence variants down to single base pair variants in an end-to-end framework. DeepC generalises well to hold out chromosomes and outperforms an existing approach based on experimentally derived chromatin features[103].

In a parallel study, Fudenberg and colleagues developed an alternative deep neural network approach to predicting chromatin interactions from large scale DNA sequence. Akita[110] also uses ~ 1 Mb sequence windows as input and predicts all-vs-all chromatin interactions. Akita has a similar convolutional module with max pooling at the start of the architecture but differs significantly in subsequent layers, the interaction encoding and the prediction task. Akita converts the 1D DNA signal

into a 2D window by averaging the hidden units pairwise in the corresponding 1 Mb x 1 Mb interaction map. Akita then performs subsequent 2D convolutions on top of this map. This allows Akita to predict a comprehensive interaction map one window at a time with dilated convolutions ensuring that information across the entire window is aggregated. This transformation allows Akita to work with a lower number of weights (deepC ~ 30 M. vs. Akita ~ 750 k). Akita is trained on the log_2 ratio of observed over expected reads, effectively normalising for the linear proximity signal. Compared to the percentile-based normalisation we employed, the log_2 ratio has the advantage of being more readily re-converted to interaction frequencies without losing information. We tried to train deepC models with a log_2 ratio encoding but these models failed to learn chromatin interactions. Rather, they learned residual variation across the linear distance gradient (Figure 3.6). In contrast to deepC, Akita can be trained *ab initio* without employing transfer learning. Training *ab inito* without transfer learning has the advantage of removing a pre-training phase from the framework, removing the need for a comprehensive compendium of chromatin data. However, this data is readily available publicly for human and key model organisms and is thus not a limiting factor in practice. Furthermore, training from randomly initialised weights may avoid biasing the training process by the selection of chromatin features present in the compendium. This may allow Akita to learn novel sequence patterns that are not informative for the chromatin feature task but may be important for the chromatin interaction task. In contrast, transfer learning may improve model interpretability. Since chromatin features such as CTCF binding sites and promoters are likely drivers of chromatin architecture, training neural networks to recognise the predictive sequence patterns of associated chromatin features may give deepC the ability to learn model internal representations of these elements. If sparser sequence patterns are important for chromatin features, transfer learning may also allow the network to learn their importance for chromatin interactions when the patterns are too rare to be trained *ab initio* given the limited number of examples from the genome.

A particularly challenging aspect of chromatin architecture prediction is tissue-specificity. Large parts of the chromatin architecture are conserved across cell types showing only moderate levels of novel interaction domains or extensions of domains. For predicting chromatin interactions at large scale, we have a limited number of training examples available even when windowing the entire genome. Predicting nuanced difference within large conserved patterns is challenging. DeepC appears to capture tissues-specific patterns better than Akita, but only when training a dedicated model per cell type. Comparing models from independent training harbours the risk of comparing models in different local optima. Models trained jointly partly address this risk, although technically a similar convergence of multiple output neurons is not guaranteed. Joint cell type models predict an average chromatin interaction profile and do not pick up on tissues-specific nuances. This means that Akita, which is only trained on several cell types jointly appears to struggle with predicting tissue-specific chromatin interaction. Neither deepC nor Akita implicitly encode cell types in their model. Joint models, such as Akita treat different cell types as independent entities, making it harder for the model to share information across cell types. Therefore, the future development of deep learning models that are able to encode and/or learn a relationship between cell types may be better able to share information across this sphere and utilise this structure to learn finer tissue-specific details.

Overall, deepC and Akita represent distinct approaches for predicting genome folding, both with associated advantages and disadvantages. We are confident that future extensive studies of their respective merits and the consolidation into a unified approach will lead to more powerful and efficient models for studying chromatin architecture.

We validated deepC with high sensitivity chromosome confirmation capture approaches, demonstrating that deepC can predict fine-grained domain structure that can surpass the resolution of the input Hi-C data. Ideally, we would use a

genome-wide all-vs-all interaction maps at higher resolution and sensitivity than the input Hi-C data used. However, as the Hi-C data we have used to train deepC has not yet been surpassed in sequencing depth, we resorted to targeted chromatin interaction enrichments from viewpoints and regions of interest. The deepC predicted domains and boundaries were shown, on average,w to be true biophysical properties of the genome when probed with high-resolution Capture-C. However, the Capture-C based validation has limitations, in particular the number of viewpoints. We chose ~ 200 viewpoints covering a large sampling of CTCF bound sites and intra domain viewpoints. We have initially not included non CTCF bound enhancers and promoters, however a third set covering these elements in the future would add scope to validate that the interactions of other active elements can also be predicted efficiently. Overall, we designed the viewpoint to capture interactions within informative domain structures where deepC prediction where more detailed than the Hi-C input. We therefore likely biased the viewpoints to more challenging regions.

We expected that an even more comprehensive Capture-C based validation would substantiate the accuracy of deepC predictions further. Tiled-C allows us to experimentally go beyond Hi-C data sensitivity but is also limited to selected regions of interest. We observed promising validation of deepC accuracy at the tested regions. Ideally, however, we would validate deepC with all-vs-all interactions at a larger scale. One limitation of deepC we observed was in genomic regions associated with the nuclear lamina (Figure 3.11), where deepC predicted domain structure that was not detectable by Hi-C nor Tiled-C. No markers of lamina association were included in the chromatin feature training in step 1. Including such features in the initial training step may help learning sequence determinants of LADs and thus help deepC to better predict the associated chromatin structure.

After demonstrating deepCs accuracy and generalisability, we next turned to utilising deepC to dissect the sequence determinants of genome folding and to predict the impact of sequence variation.

3.4 Data & Code Availability

The deepC code and trained models are available via github: (https://github.com/rschwess/deepC). Hi-C data from Rao et al. is available under GSE63525. The used chromatin feature data from ENCODE, Roadmap and other publicly available sources are detailed in Appendix A. Probes used for Capture-C are detailed in Appendix B (https://doi.org/10.5281/zenodo.4751244). NG Capture-C and Tiled-C validation data are available under the GEO super series GSE137437.

3.5 Acknowledgments

The neural network architecture was designed and optimised with the input of Gerton Lunter and Yee Whye Tee. The validation strategy, using Capture-C and Tiled-C, was designed with the help of Jim Hughes, Marieke Oudelaar, Matthew Gosden and Damien Downes. The Capture-C experiments were performed by Matthew Gosden and Tiled Capture-C experiments were performed by Damien Downes.

4

Investigating the regulatory grammar of 3D chromatin interactions

Contents

4.1 Introduction 105
4.2 Results 108
4.2.1 Mapping sequence determinants at base pair resolution 108
4.2.2 Variant effect prediction 112
4.2.2.1 Predicting the effect of a validated boundary deletion 113
4.2.2.2 Predicting the impact of single base pair variants 115
4.2.3 Predicting the relative contribution of regulatory element classes to 3D genome structure 120
4.2.4 Dissecting the regulatory grammar of chromatin boundaries 123
4.3 Discussion 131
4.4 Data & Code Availability 136
4.5 Acknowledgments 136

4.1 Introduction

Genome architecture and its implications for genome function and gene regulation are extensively studied, however the key mechanisms and their implications for functional variation are still poorly understood. Mammalian genome architecture is comprised of an intricate framework of insulated domains, that shape the landscape

of interactions between distal regulatory elements. The dominant model explaining the formation of chromatin architecture is loop extrusion[35, 36, 115, 116]. Cohesin, loaded onto the DNA, is thought to extrude chromatin through its loop like structure in an ATP dependent manner. This extrusion is blocked to varying degrees by boundaries, predominantly CTCF bound sites. CTCF bound sites are enriched at domain boundaries in a convergent orientation. However, there are 10s of thousands of CTCF sites throughout the entire genome and their specificity is all but clear. Many CTCF sites do not show any insulating properties to chromatin interactions and many sites appear redundant. CTCF binding is largely conserved across cell types and development. However, the mechanistic implications of the CTCF demarcated chromatin domains vary depending on the cell-type-specific activity of the regulatory elements within[27].

Insights into the sequence determinants of boundary elements are limited. The core CTCF binding motif is well characterised, but the motif alone does not explain the varying boundary potentials. Studies in mouse have revealed an upstream and downstream motif that may modulate CTCF binding strength and hence may affect its boundary strength[117]. Co-motifs and co-occurring regulatory elements in the proximity of CTCF sites may be crucial to understand their insulating properties. A comprehensive integrated study is still missing. It remains unclear, which and to what extent genomic elements other than CTCF sites are acting as boundaries. Experiments with acute CTCF depletion have shown that the genome is largely reorganised into domains corresponding to active transcription[118–120]. It remains unclear to what extent promoters represent boundaries to loop extrusion and how they balance with CTCF driven insulation in an unperturbed cell[36]. Moreover, where and how Cohesin is loaded onto the DNA is still unclear, although speculations implicate active regulatory elements, namely promoters and enhancers in this process. Natural occurring genomic variants that alter chromatin structure offer crucial insights into the mechanisms of genome folding and to its importance for gene regulation. Importantly, assessing genomic variations associated with disease in

the context of chromatin architecture may help to explain the underlying molecular mechanism and prioritise affected genes and pathways. Genomic variation affecting boundary elements can lead to miss regulation of gene expression. These can be the drivers for drastic phenotypes such as cancer[94] or developmental malformations[95, 121, 122]. Although examples of chromatin architecture altering variants have been characterised in detail, the abundance of CTCF sites and the lack of understanding of their specificity renders it challenging to predict the effect of a given genomic variant. Moreover, most variants described in the literature encompass entire boundary elements or larger structural variations. The effect of small indels and single base pair variation on chromatin architecture has been less frequently studied. Isolated work points to the importance of genome folding alteration as an important mechanism for interpreting SNPs identified in genome-wide association studies[123].

The time and cost consideration of chromosome conformation assays make computational methods valuable to predict potential genome folding altering variants and prioritise candidates for functional follow up. Currently available methods have rarely been employed to predict the impact of genomic variants. Only polymer models have been used to model the effect of large-scale structural variations on chromatin architecture[98]. Polymer models and chromatin feature based prediction methods coarse grain the DNA into features or polymer beads. Predicting the effect of small indels or single base pair variants therefore requires the use of intermediate models to predict the effect of the variants on chromatin features first. Computational methods that use the DNA sequence and could thus potentially model the impact of single base pair variants have been limited to window-to-window based interactions, ignoring the wider sequence context between and around the windows. Integrating information about surrounding genomic elements such as boundaries appears crucial for accurate prediction of the impact of variants. DeepC offers a tool to predict the impact of genomic variants on genome folding, while considering DNA sequence at base pair resolution over megabase contexts. We hypothesised that this combination makes deepC well suited for variant effect

prediction. Our aim is to apply deepC to supplement existing and ongoing research into chromatin structure. We seek to employ deepC as tool for an integrative computational study of the determinants of chromatin architecture. By learning the sequence determinants important for predicting chromatin structure accurately, we seek to generate hypothesis about the relative contribution of different factors and of the different classes of regulatory elements. Furthermore, we aim to provide a computational framework for predicting the impact of sequence variation on genome folding, covering a range of variant sizes from structural variation to single base pair variants. We aim to utilise the variant impact predictions to further dissect the functional elements shaping genome structure. In addition, we seek to provide a framework for researchers to query discovered variants, to assess their potential effects on genome structure in silico without the need to perform expensive and time-consuming chromosome confirmation assays. In this chapter, we describe how we employ deepC for these tasks.

4.2 Results

4.2.1 Mapping sequence determinants at base pair resolution

Current understanding suggests that genome architecture is the emergent property of a combination of TF binding and histone modifications ultimately leading to the recruitment of key nuclear proteins and processes. The presence of all these chromatin features can be predicted from DNA sequence to varying degrees of accuracy[60, 61]. We therefore hypothesised that chromatin architecture may itself be predicted from DNA sequence and built deepC for this purpose. With models that predict genome folding at sufficient accuracy, we can now reverse engineer the predictive DNA features learned by the deep neural networks, to generate and test hypotheses *in silico*. To achieve this, we aim to dissect the importance of the sequence input for the chromatin folding prediction, both at single loci and genome-wide.

Dissecting the importance of features for deep neural networks predictions is an ongoing field of research. There are two principled ways for assigning the importance of DNA sequence at the base pair level[124]: gradient-based and perturbation-based. For the gradient-based approach, we can calculate the gradient of the prediction output with respect to the sequence input and multiply it by the one-hot-encoded sequence. We effectively ask how much the output would change if we change the 1 in the one-hot-encoding-based sequence representation marginally. Although this does not model a possible biological event, it yields the importance of each present base in the sequence. The importance of all bases in a sequence window is calculated with a single forward and backwards pass through the network, making it computationally efficient. In contrast, for the perturbation-based method we *in silico* mutate each base pair of interest in a given window to all three possible variants and compute the network output for every variant. This approach models the predicted effect of a single base pair change at a time and so more closely replicates a sequence variant. However, each variant requires a separate forward pass through the network, making this approach more computationally demanding, especially with the low batch size restriction of deepC (see Methods). To assess the difference between the gradient- and perturbation-based approach in the context of deepC, we selected a CTCF bound site and computed both the saliency score[68, 125] and the perturbation score for every possible variant over a 100 bp region centered on the CTCF core motif (Figure 4.1). We then computed the average change for the three possible variants at each position. Both saliency and perturbation recover the core CTCF motif at bp resolution and flag flanking sequences as important, although at different magnitudes. Although the saliency score appears slightly noisier, given the large computational benefits in terms of speed and throughput, we henceforth primarily employed the saliency score as a computationally efficient methods to get genome-wide base pair resolution important scores.

When calculating the saliency score using the GM12878 model genome-wide, we observe peaks of importance score. The trends we observe genome-wide are exemplified in a region around the *IKZF2* gene on chromosome 2 (Figure 4.2). The

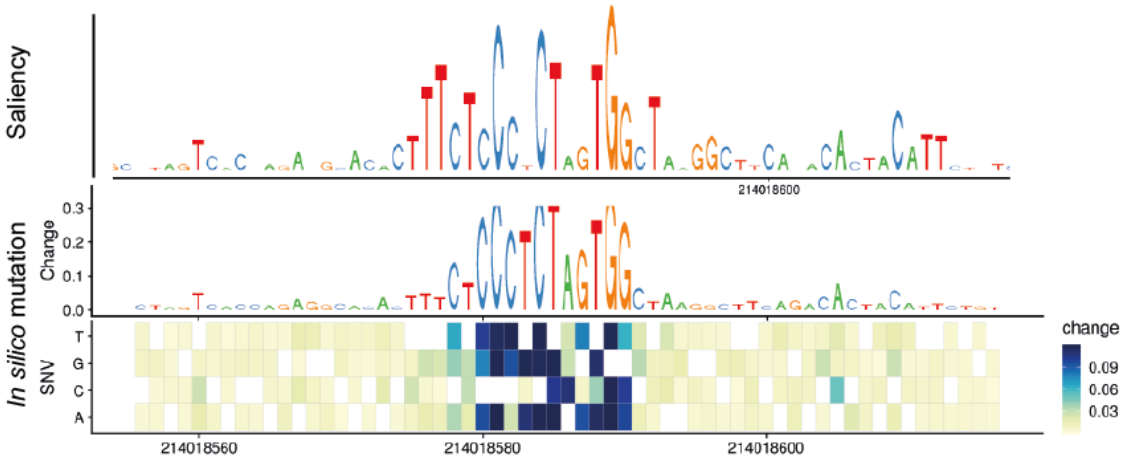

Figure 4.1: Comparison of deepC saliency score vs. *in silico* mutation of variant impact over CTCF site core motif using the GM12878 model. The saliency measures the importance of base pairs for chromatin architecture prediction. The *in silico* mutation score predicts the change on predicted genome folding upon single base pair mutation. The change is summarised as the average absolute change in all interactions within 1 Mb of the variant. For the profile, the three possible changes at each position are averaged.

primary signal are sharp peaks at CTCF bound sites, characterised by accessible chromatin and CTCF binding. These observations are in line with our expectation that CTCF sites play an important role for predicting chromatin folding. In addition, we observe broader peaks at a smaller magnitude primarily over promoter regions. Resolving the saliency at base pair resolution over the promoter region of *IKZF2* highlights the CTCF core motif in CTCF sites and general transcription factor binding motifs over the promoter. *IKZF2* is actively transcribed in GM12878 cells and the promoter is marked by H3K4me3 and H3K27ac, promoter and regulatory activity associated histone marks.

To quantify the saliency of regulatory elements genome-wide we calculated meta-profiles over accessible chromatin regions in GM12878 relative to the peak call centers of the elements. Regions were derived from peaks in DNase-seq profiles and classified by chromatin mark and CTCF ChIP-seq using chromatin segmentation with GenoSTAN[40]. CTCF sites show a high and sharp saliency meta-peak relative to other elements (Figure 4.3). CTCF sites bound more strongly by CTCF (CTCF

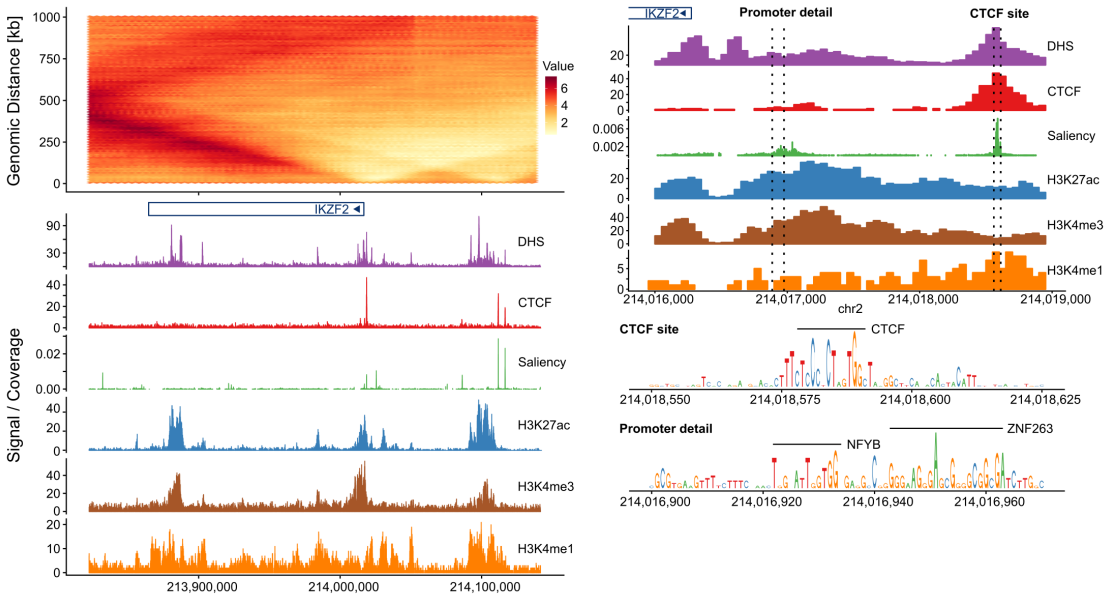

Figure 4.2: Saliency pattern over the *IKZF2* locus with promoter and CTCF site. Shown are predicted chromatin interactions in the GM12878 model. Accessible chromatin shown in purple and CTCF and histone modifications from ChIP-seq are aligned below. The saliency, the importance score for predicting chromatin interactions is shown in green. Zooming into the promoter region shows a broad peak over promoter and sharp peaks over the CTCF. At the base pair level, the saliency score highlights TF binding motifs indicated by black lines. Plot adapted from Schwessinger et al.[107].

strong) as indicated by higher CTCF ChIP-seq peaks show a stronger saliency peak than those over smaller CTCF ChIP-seq peaks (CTCF weak). The meta-peak is approximately 1 kb in size and is centered relative to the observed sites. Both promoters and enhancers show a sharp central meta-peak when they overlap with CTCF binding indicated by CTCF ChIP-seq. In contrast, promoters without CTCF binding show a broad peak in the meta profile. This is pronounced in the promoters associated with activity associated histone marks (Promoter active), while promoters lacking them show only a faint meta-peak. Enhancers without CTCF binding show no noticeable enrichment for saliency scores. Taken together, genome-wide saliency maps indicate CTCF sites and active promoters as predictive for chromatin folding. The saliency highlights TF motifs predictive for chromatin folding. We therefore hypothesied that the saliency scores could be used to assign functional importance for chromatin folding. This in turn may be used to fine map single base pair variants such as those identified by GWAS or eQTL studies with respect to consequences

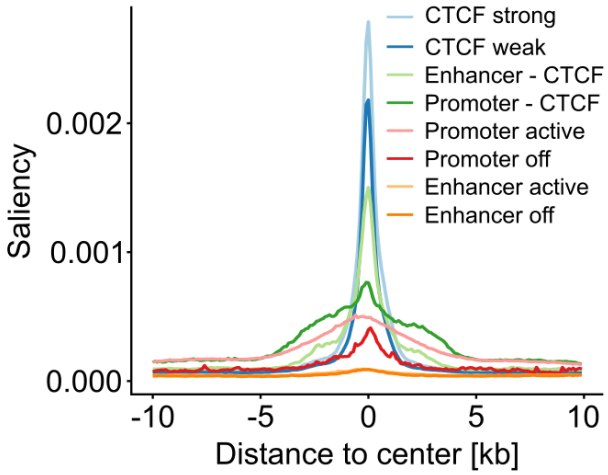

Figure 4.3: Genome wide meta profiles of the saliency score over accessible chromatin and CTCF ChIP-seq peaks relative to the peak centre. Peaks were stratified based on chromatin segmentation using CTCF and histone modification ChIP-seq. Plot adapted from Schwessinger et al. 2020[107].

on genome folding. To test this, we retrieved cell type specifc eQTLs identified in GM12878 by the GTEx consortium. We filtered the identified SNPs for overlap with accessible chromatin or CTCF sites and calculated the saliency score for each variant. We then randomly shuffled the SNP positions requiring them to be located within the same set of open chromatin and CTCF sites and calculated the saliency for the shuffled set. The eQTLs showed a significant enrichment of saliency score (Figure 4.4) (P-value $<$ 1e-85 using a two-sample Kolmogorov-Smirnov test). This suggested that the saliency score may indeed be used to fine map single base pair variants with respect to influence on genome-folding.

4.2.2 Variant effect prediction

Next, we assessed deepC's utility for predicting the impact of sequence variation on genome folding. We first sought to validate the variant effect predictions on well characterised examples from the literature before utilising deepC for novel variants and to perform *in silico* screens. Benchmarking chromatin folding predictions is challenging because comprehensive validation of a variants effect requires chromosome conformation capture assays performed in relevant wild type

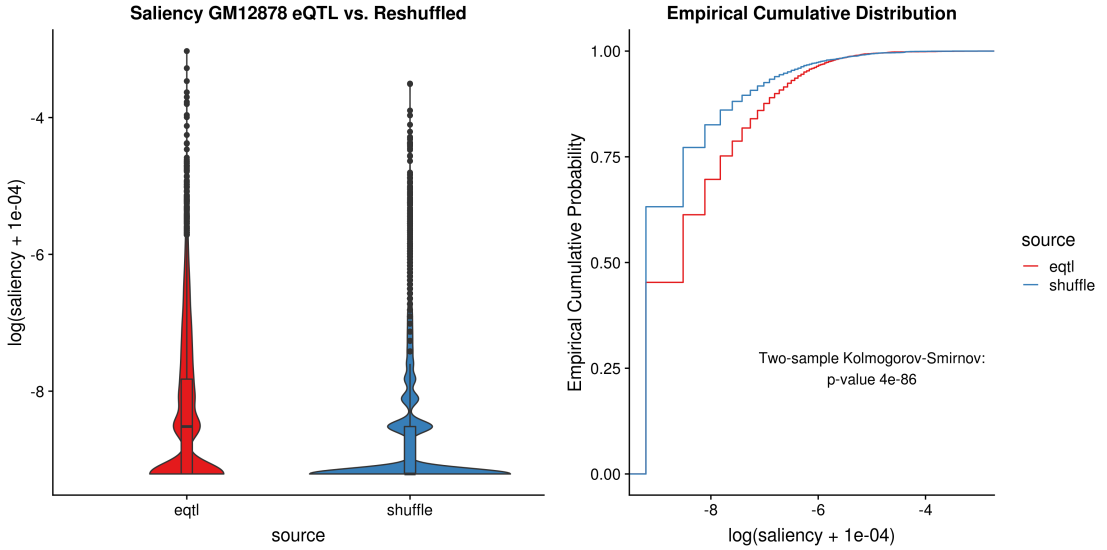

Figure 4.4: Saliency score of GM12878 specific eQTLs within accessible chromatin and CTCF sites vs. shuffled background SNPs. Background SNPs were shuffled by resampling SNPs from within the accessible chromatin and CTCF site set. Shown (left) are violin and boxplots representations of the saliency score. Shown (right) are the empirical cumulative distributions of both sets. A two-sample Kolmogrov-Smirnov test showed a significant enrichment of eQTLs for higher saliency scores. Plot adapted from Schwessinger et al.[107].

and variant genotype. Usually conformation captures are only performed in wild type. Variant effect predictions are derived by overlapping them with genomic features or statistically linking them with downstream effects, such as gene expression by sampling populations of genotypes. To address the effect of variants on genome organisation more directly, we selected studies where the chromosome conformation has been assessed in both wild type and variant.

4.2.2.1 Predicting the effect of a validated boundary deletion

To test deepC's ability to predict the effect of larger deletions we utilised a well-characterised example from the literature. Using CRISPR/Cas9 Hnisz et al. deleted a ~ 30 kb genomic fragment at the *LMO2* locus on chromosome 11 that overlaps four CTCF sites and acts a boundary in HEK-293T cells[94]. The deletion was associated with the merging of the two adjacent, insulated domains as demonstrated by 5C[126]. The deletion also led to a 2-fold increase in *LMO2* expression, likely caused by novel interactions of the *LMO2* promoter with upstream regulatory

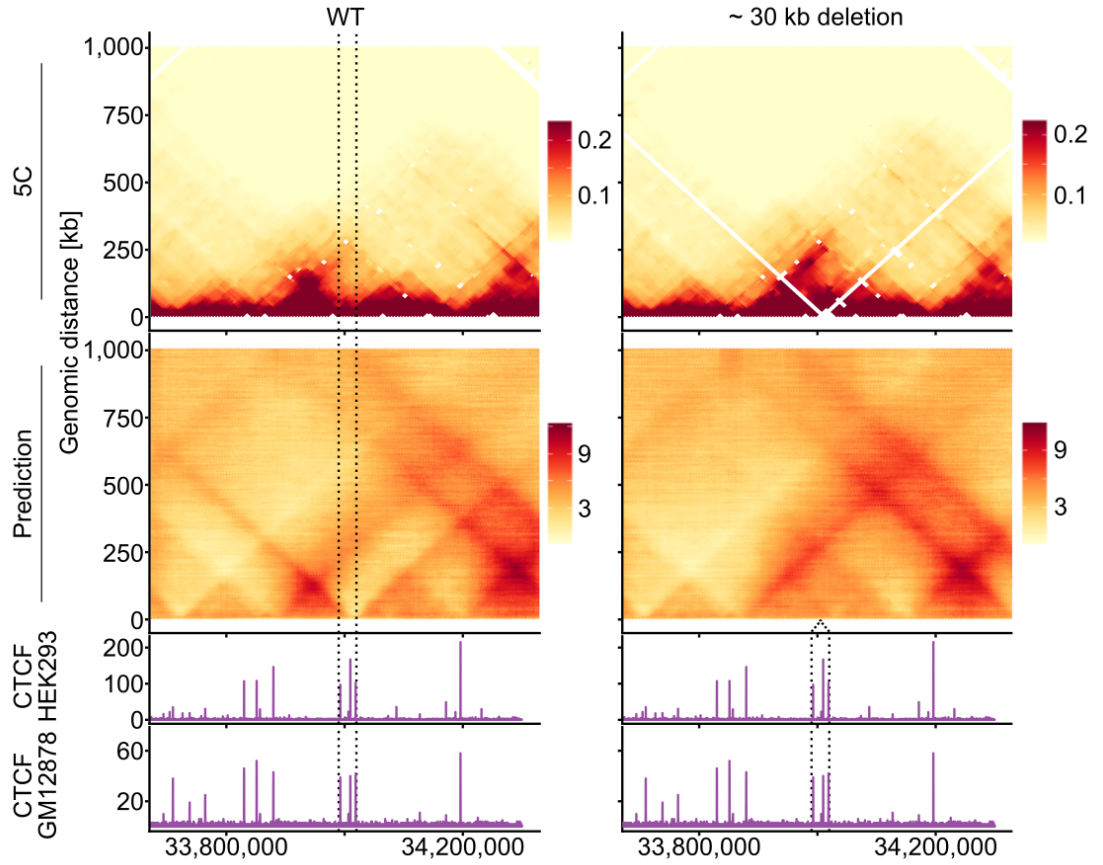

Figure 4.5: Effect of deleting a 30 kb fragment at the *LMO2* locus at chromosome 11 harbouring four CTCF sites in HEK-293T cells. Shown are chromatin interactions in wild-type and deletion measured by 5C by Hnisz et al.[94] and deepC predictions in GM12878. The deleted fragment is highlighted in dotted lines. CTCF ChIP-seq tracks for both cell lines are shown below in purple. Plot adapted from Schwessinger et al.[107].

elements. We used deepC to predict the effect of the deletion *in silico*. Because, no deep HEK-293T Hi-C data for training a deepC model was available we used the GM12878 model. The CTCF binding profiles in the *LMO2* locus are highly similar between the two cell lines as indicated by ChIP-seq (Figure 4.5). When deleting the same genomic region computationally in the GM12878 model, deepC recapitulates the effect observed by Hnisz et al. (Figure 4.5). The two insulated domains adjacent to the deletion merge into one.

The deletion encompassed four CTCF binding sites as indicated by peaks in the CTCF ChIP-seq profiles in HEK-293T and GM12878. The presence of multiple sites might indicate redundant boundaries. We thus used deepC to computationally

model deletions of the CTCF binding sites individually and in combination (Figure 4.6). Three binding sites are marked by strong CTCF ChIP-seq peaks and a fourth by a smaller peak. Deleting each CTCF site individually leaves a marked insulation between the domains, with CTCF site (2) having the strongest individual effect. When deleting all three strong binding sites in combination, deepC predicted a gradual fusion of the domains but with visible insulation left. Only when deleting all four CTCF sites in combination does the predicted effect equal the one predicted for the 30 kb fragment deletion. The predictions suggest a redundant boundary where no single CTCF site deletion is able to produce the same chromatin architectural changes.

4.2.2.2 Predicting the impact of single base pair variants

A key design feature of deepC is its ability to predict the impact of single base pair variants on large scale genome structure. The effect of single base pair variants on genome folding is less frequently studied than the effect of larger structural variation. To validate deepC's capabilities, we utilised one of the rare studies where single base pair variants have been implicated to alter genome folding and this effect has been validated by chromosome confirmation capture. Schmiedel et al. functionally dissected the GWAS identified 17q21 genetic risk locus associated with asthma[123]. From a haploblock containing 136 SNPs in tight linkage, Schmiedel et al. identified two SNPs overlapping CTCF binding motifs as candidate causal variants: rs12936231 and the rs4065275. The risk allele was associated with a three-fold increase in *ORMDL3* expression in CD4+ T-cells. Differential CTCF binding over the two SNPs was confirmed by ChIP-qPCR, showing significantly more CTCF binding over the rs12936231 non-risk and the rs4065275 risk allele respectively. To validate the effect on chromatin architecture Schmiedel et al. performed 4C[127] assays using the *ORMDL3* promoter as viewpoint in CD4+ T-cells homozygous for risk and non-risk allele. Average 4C profiles for the risk allele show increased interactions with the intronic regions of *IKZF3* (Figure 4.7). In contrast, the average non risk profile shows a lack of interactions in the intronic

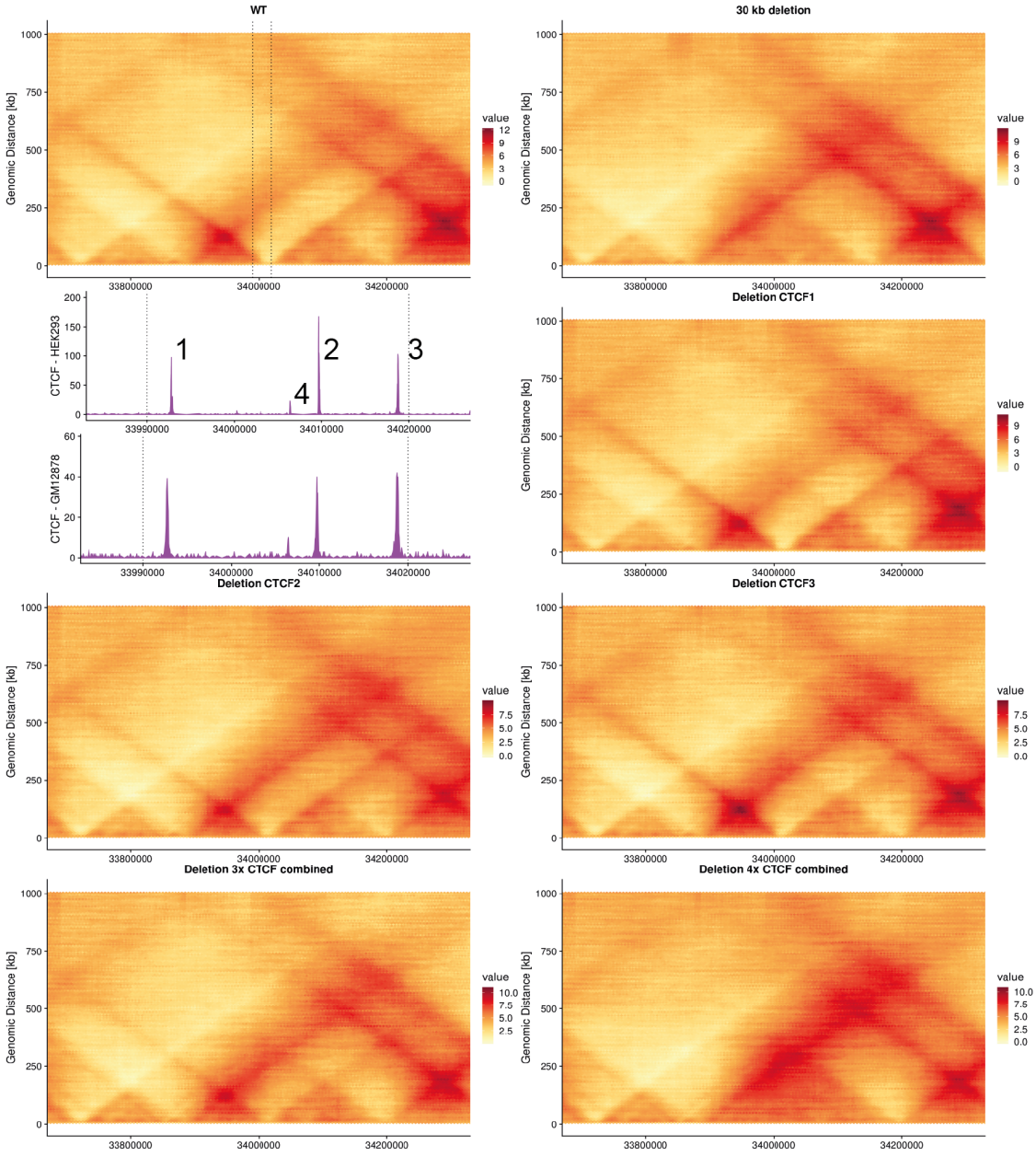

Figure 4.6: Single and combinatorial *in silico* deletions of CTCF sites. Shown are the predicted effects of CTCF site deletions within the genomic fragment deleted by Hnisz et al. CTCF ChIP-seq profiles in WT HEK-293T and GM12878 cells are shown in purple. CTCF sites are numbered. Predictions of chromatin interactions for WT, the 30 kb fragment deletion, individual deletions of site 1, 2 and 3 and of 1-3 and 1-4 in combination are shown. Plot adapted from Schwessinger et al.[107].

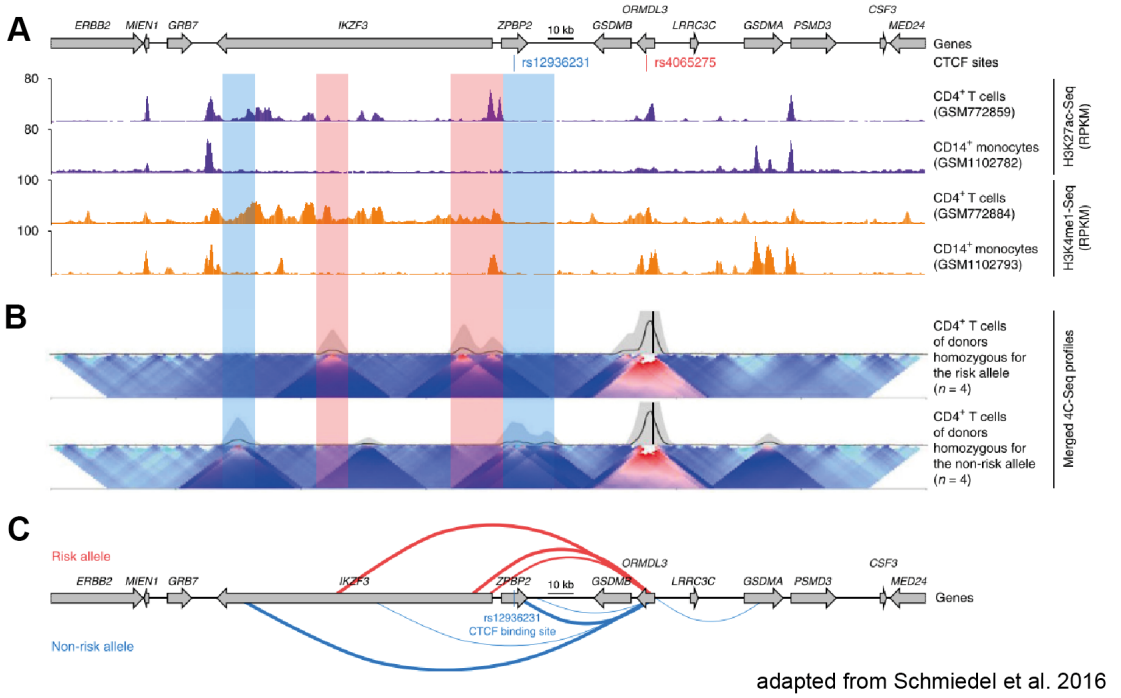

Figure 4.7: CD4+ T-cell chromatin marks and 4C experiments in allelic contexts of asthma-associated SNPs. The SNPs of interest are shown in blue and red. Shown are histone modifications on T-cell samples homozygous for risk and non risk allele. Shown below are average 4C-seq profiles from four donors each using the *ORMDL3* promoter as viewpoint. The risk allele shows more interactions with the intronic region of *IKZF3* loosing interactions over rs12936231. This figure was adapted from Schmiedel et al.[123].

region but increased interactions over the rs12936231 SNP and downstream of *IKZF3* both regions coinciding with CTCF binding peaks (Figure 4.7).

We used the GM12878 immortalised B-cell line deepC model to predict the effect of the two SNPs on chromatin architecture (Figure 4.8). Both SNPs as well as *ORMDL3* and *IKZF3* lie within a larger insulated domain. The non risk allele shows a distinct insulation over rs12936231 separating the larger domain into sub domains and insulating *ORMDL3* from the *IKZF3* region and intronic enhancers. The risk allele showed a marked loss of the insulation over rs12936231 effectively merging the predicted subdomains. The predicted pattern is consistent with the 4C profiles observed by Schmiedel et al. and with increased contacts of the *ORMDL3* promoter with the intronic *IKZF3* enhancer. In contrast, the region over rs4065275

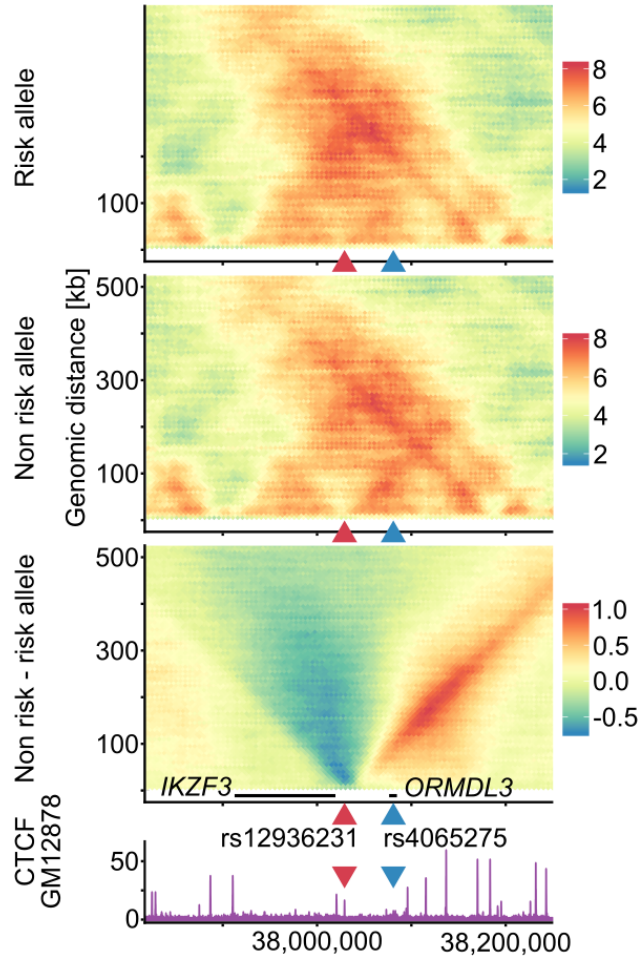

Figure 4.8: Non risk and risk allele SNP effect predictions on chromatin interactions in GM12878. Shown are the predicted chromatin interactions for the risk and non risk allele, the differential map and CTCF ChIP-seq track. SNP positions are highlighted by triangles. Only *IKZF3* and *ORMDL3* genes are plotted as black lines for clarity. Plot adapted from Schwessinger et al.[107].

shows no distinct predicted change in chromatin interactions.

To further characterise the potential role of each SNP we predicted their individual effects (Figure 4.9). Both SNPs show a differential interaction map characteristic for modulation of boundary potential but in different orientations and at a different magnitude. The rs12936231 SNP alone shows a stronger predicted effect than rs4065275 and we observe a marked increase of chromatin interactions spanning rs12936231 fitting the insulation loss observed in the combined prediction and in the 4C analysis by Schmiedel et al. The effect predicted based on rs12936231 alone

shows the same pattern of predicted changes but a slightly lower magnitude then the combined SNP prediction (compare Figure 4.8 & 4.9 B). In contrast, rs4065275 shows a loss of interactions directly over the SNP but a slight increase in interactions with the downstream region along a stripe. Although at a much smaller scale this might contribute to the amplification of *ORMDL3* promoter enhancer interactions due to its location close to the promoter. To put the strength of the single base pair changes into context, we compared their predicted effects to 1,000 randomly sampled SNPs from CTCF sites, promoters, enhancers and background sequences each. We computed the average absolute difference of interactions in the effected window, up to 1 Mb away from the SNP (Figure 4.9 D). Compared to randomly samples CTCF site SNPs, the predicted effect for the asthma associated SNPs was in the upper 11 % and 1 % for rs4065275 and rs12936231 respectively. Taken together, deepC prioritises rs12936231 as the most likely functional candidate SNP showing the strongest individual effect. However, rs4065275 still has an effect matching the observed molecular phenotype. The difference in magnitude between the two SNP and the single SNP predictions further suggests that both SNPs might act in concert if present on the same haplotype.

Schmiedel et al. narrowed down the effect of the asthma associated SNPs to immune cells, with the strongest effect on expression observed in CD4+ T-cells. We observed earlier that deepC can learn to predict tissue-specific chromatin interactions. We were therefore interested to evaluate if deepC can predict the tissue-specific nature of the asthma SNP effect. The predictions presented above were made using the GM12878 deepC model trained on Hi-C data from an immortalised B-cell line. When we predict the effect in the K562, immortalised erythroid progenitors and IMR90, immortalised fibroblast cell line, we observed no distinct predicted impact on chromatin architecture (Figure 4.10). While a small change in the IMR90 model can be observed it is ~ 100-fold lower than the GM12878 effect. These results suggest that deepC can predict tissue-specific effects of genomic variation even at the single base pair level.

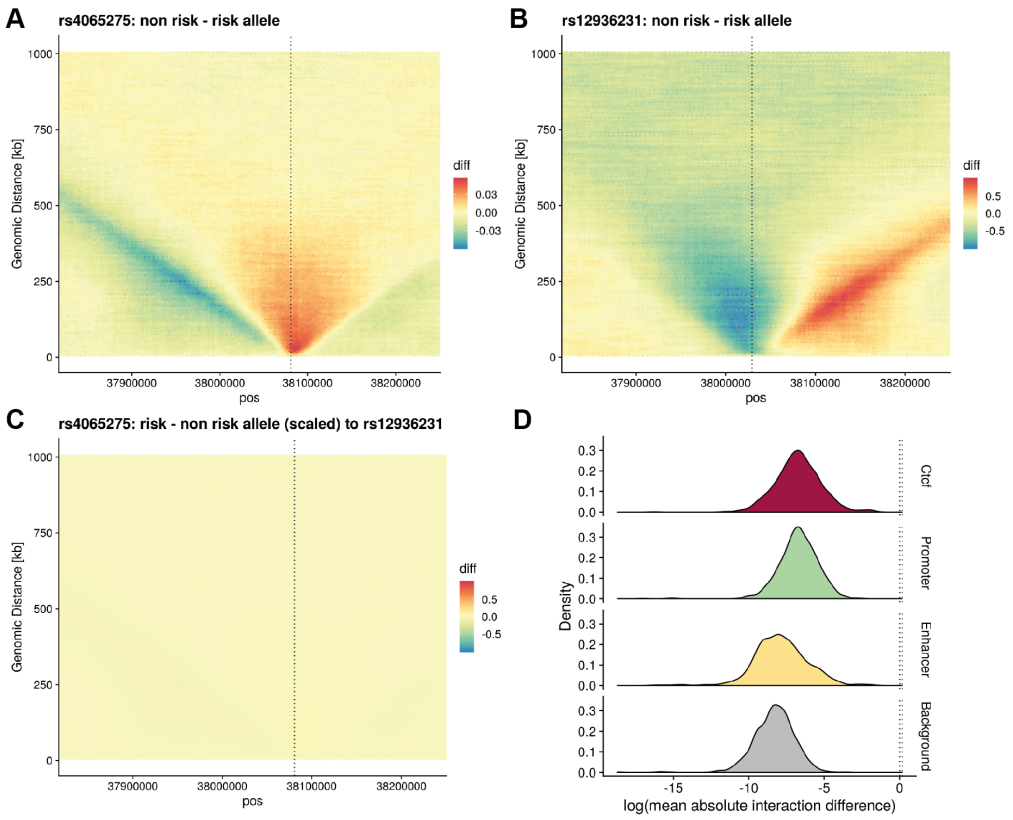

Figure 4.9: Predicting impact of single asthma-associated SNPs (A-C) on chromatin interactions. rs4068275 is plotted twice once to show the pattern of the predicted effect (A) and once at the same colour gradient magnitude (C) as rs12936231 to highlight the difference in strength. D) Placing the predicted changes into the context of 1,000 randomly sampled SNPs within CTCF sites, promoters, enhancers. Dotted lines mark the SNP effects. Plot adapted from Schwessinger et al.[107].

4.2.3 Predicting the relative contribution of regulatory element classes to 3D genome structure

The exact role genomic elements play in establishing and maintaining chromatin architecture is still poorly understood. CTCF sites are enriched at domain boundaries but their specificity is still unclear[14]. The role active promoters and enhancers play in shaping the 3D genome, perhaps in recruiting Cohesin to the DNA, remains to be clarified. Here, we thought to employ deepC to perform computational deletion screens to dissect which regulatory elements are predictive for chromatin architecture and lead to changes therein upon deletion. As an initial test for our experimental design, we first confirmed that computational deletion of a single CTCF site can have large predicted consequences on genome folding. We deleted

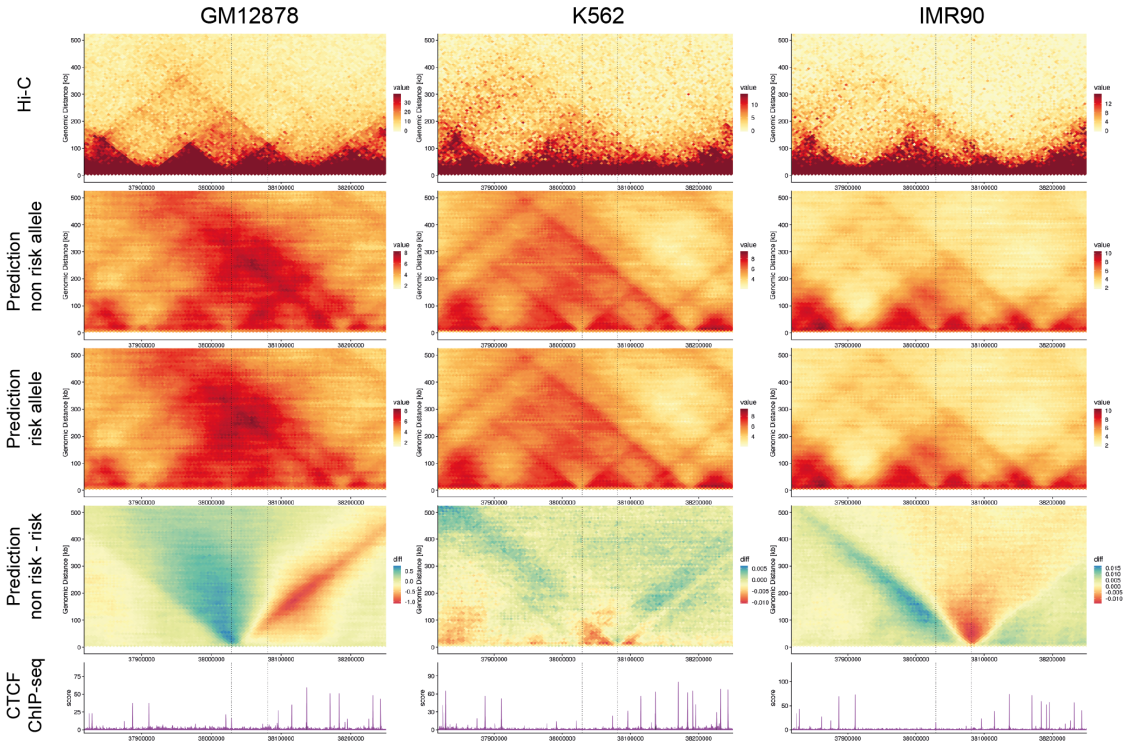

Figure 4.10: Predicting the impact of both asthma-associated SNPs in three different tissue models. Shown are deepC predictions in GM12878 (B-cell derived), K562 (erythroid) and IMR90 (fibroblasts). Differential maps are shown at different scales to highlight the predicted pattern of change. The effect in GM12878 is the strongest. Plot adapted from Schwessinger et al.[107].

500 bp spanning a single CTCF site on chromosome 17. When deleting the single site and predicting the change in chromatin architecture, we observed a merging of subdomains coupled with a reconfiguration of the insulated domains downstream (Figure 4.11).

Next, we performed a genome-wide *in silico* screen in GM12878 cells by deleting all accessible chromatin sites and CTCF binding sites as defined by peaks in the DNase-seq and CTCF ChIP-seq profiles. We predicted the effect of deleting every single element and computed the mean absolute interaction difference in the window of 1 Mb around the deletion. We then stratified the deleted elements by chromatin segmentation using CTCF and histone modification ChIP-seq for H3K4me1, H3K4me3 and H3K27ac. We then summarised the predicted effects in these chromatin segmentation stratified classes in boxplots (Figure 4.12). On

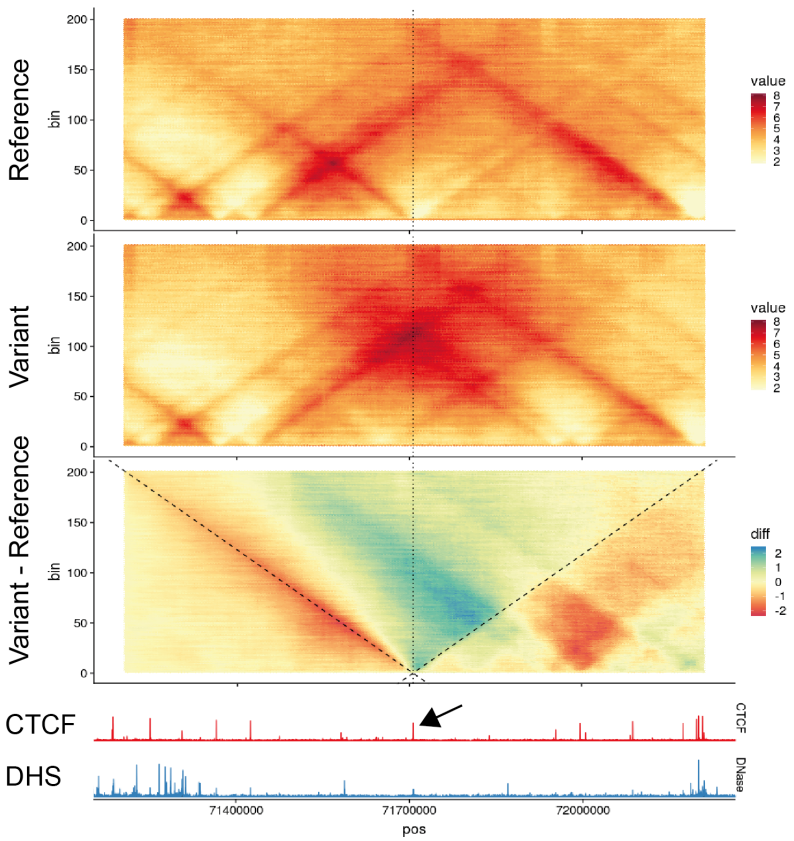

Figure 4.11: Exemplary *in silico* CTCF site deletion of a 500 bp fragment on chr17. Predictions were made in the 5 kb GM12878 model and. GM12878 CTCF ChIP-seq and DNase-seq tracks are shown below. Bins refer to the 5 kb genomic bins at increasing linear distance. Plot adapted from Schwessinger et al.[107].

average this matched our expectation from the literature that CTCF sites are important elements for predicting chromatin architecture. Those sites bound by more CTCF as measured by the height of the CTCF ChIP-seq peak (CTCF - strong) also have on average a stronger predicted change in genome folding. Both promoters and enhancers that overlap with CTCF binding sites have a strong predicted average change. However, the overlap makes it challenging to dissect the role each individual element plays. Interestingly, promoters show on average a stronger predicted impact than enhancers. Within both enhancers and promoters, it is the ones marked by regulatory activity associated histone modifications (Promoter – strong / Enhancer - strong) that have the larger predicted impact upon deletion. The full list of genomic positions and predicted impact of all element is available online (see Data availability).

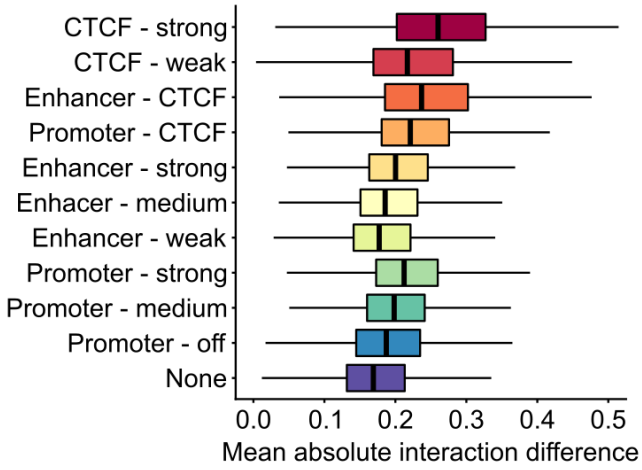

Figure 4.12: Boxplot summary of genome-wide deletion screen with chromatin segmentation stratification. Predictions are made using the 5 kb GM12878 model. Types of regulatory elements were defined by curating the chromatin segmentation results (see Methods 6.2.6.2). None refers to elements that have been identified as open chromatin or CTCF ChIP-seq peaks but that do not show any significant enrichment of CTCF, H3K4me1, H3K4me3 or H3K27ac ChIP-seq reads. Boxplots show the median as thick line, the 25th and 75th percentile as left and right hinge respectively. The whiskers stretch up 1.5 times the IQR (inter quartile range) away from the respective whisker. Outliers outside this range were omitted from plotting. Plot adapted from Schwessinger et al.[107].

4.2.4 Dissecting the regulatory grammar of chromatin boundaries

Tens of thousands of CTCF binding sites are present in the human genome. Their crucial role for establishing and maintaining chromatin architecture has been suggested by their enrichment at domain boundaries[14] and demonstrated by the consequences of deletions both naturally occurring[94, 128] and engineered[94, 129]. However, the specificity of CTCF sites is still poorly understood. Not all CTCF sites coincide with boundaries and redundant binding sites may have evolved to ensure robust gene regulation at crucial loci and TAD robustness[130]. The exact sequence determinants that make a strong boundary are unclear. The CTCF core motif is well established and motifs in convergent orientation are enriched at domain boundaries. It remains unclear which co-motifs may modulate CTCF binding and what additional factors play a role. For example, TF's enriched for co-binding at CTCF sites have been described[131] leading to speculations over their precise role

in chromatin folding. It is challenging to dissect the regulatory grammar of genomic elements by analysing co-occurring patterns alone. Ultimately, genome engineering experiments offer a tool for studying the role and interplay of individual elements precisely, by deleting elements one-by-one or by engineering elements into artificial loci. However, the throughput of precise genome engineering approaches is limited. Generating testable hypothesis and effective candidate prioritisation are therefore key. We sought to employ deepC for this task.

The predicted effects in our *in silico* deletion screen are in part affected by the surrounding genomic elements such as boundaries that determine how domains may reconfigure upon deletion of a single element. However, deepC offers us a unique opportunity to focus only on the sequence composition of individual elements independent of surrounding elements. Using deepC, we can study the grammar of chromatin boundaries by computationally engineering them into the same locus one-by-one and predicting the effect on the surrounding chromatin architecture. By evaluating genomic elements in the same locus, we can distinguish the effect of the element deletions from the contribution of surrounding elements. Furthermore, we can use multiple loci to affirm that the predicted effects are likely to generalise to other loci in the genome.

As a precursor, we tested deepC's ability to predict the effect of CTCF site insertions in different orientations and at different genomic loci. We selected five loci (see Methods). As an initial test, we computationally inserted three CTCF sites from the *HBA* locus spanning the respective CTCF ChIP-seq peak. The selected sites cover both orientations of the CTCF core motifs (see Methods) and different CTCF peak heights. To eliminate differences due to changes in size of the locus in the predictions we replaced 600 bp of the native locus with 600 bp of the test sequence. As a control to ensure that the predicted effects were not due to the deletion of the native sequence, potentially disrupting TF binding sites, we first inserted 600 bp of N's into the respective loci and predicted the effect on chromatin structure using the K562

model (Figure 4.13). We observed no strong predicted changes upon N insertion. We next inserted the selected CTCF sites into each locus and predicted the change of chromatin interactions relative to the N insertions. Depending on the inserted CTCF site we observed a predicted gain of insulation at different magnitudes and in different orientations matching the orientation of the respective core motifs. Interestingly, the predicted changes appeared very similar when comparing the insertions into a medium sized or larger TADs (loci 3 and 4) or into a complex region (loci 2).

To quantify the effect of a large number of CTCF site insertions into the five loci, we next inserted all CTCF sites on chromosome 16, a hold-out chromosome. A total of 2,932 CTCF sites were selected as peaks in the K562 CTCF ChIP-seq profile using a sequence of length 600 bp centered on each element. We predicted the change in chromatin interactions relative to equal length N insertions. We summarised the predicted changes as the mean absolute change in interactions over the sequence window of 1 Mb around the element and also calculated the gain of directionality in the insertion (see Methods). To compare the effects predicted across the five loci, we plotted the distribution for both metrics and calculated the pair-wise Pearson correlation coefficient (Figure 4.14). The mean absolute interaction difference was highly similar, although this measure is likely inflated by the large amount of low impact elements, while the distributions tails appear more distinct. The distributions over loci 3, 4 and 5 appear most similar and these are the loci where the elements are placed into single homogenous TADs. The directionality gain, while still showing high Pearson correlation coefficients, seems to depend more on the surrounding genomic structure. This suggests exercising caution when defining changes in chromatin architecture as they may vary depending on the metric used as proxy. For the purpose of dissecting the sequence determinants of boundaries elements, our analysis suggested that it is sufficient to screen the elements by insertion into a single locus. Locus 3, 4 and 5 appeared better suited for a genome-wide screen since the effects predicted in them were highly similar.

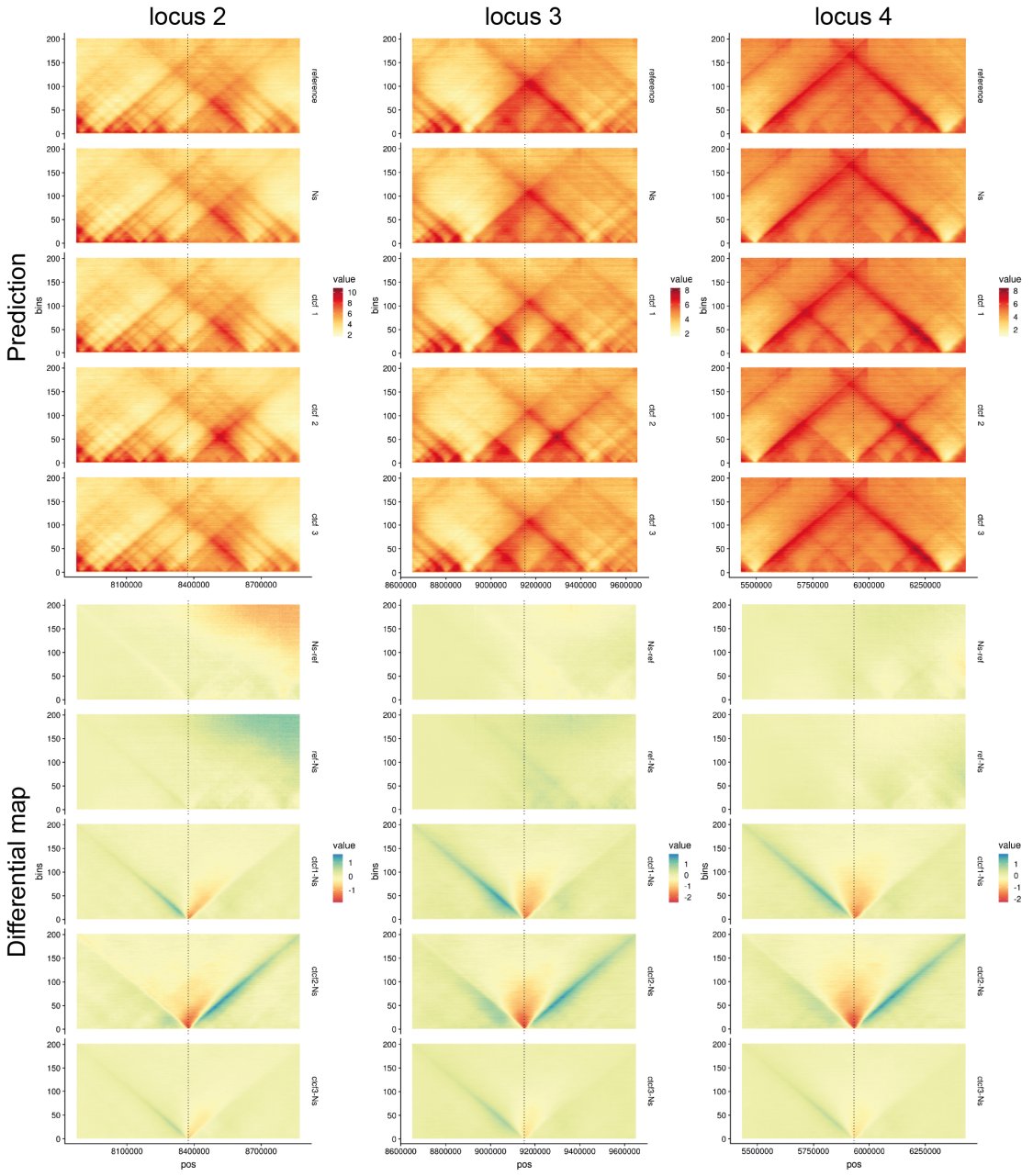

Figure 4.13: CTCF insertion prediction in different loci. Shown are CTCF insertions into three out of the five assessed loci, with predictions and differential maps using the K562 5 kb model. Shown are reference predictions, insertion of 600 bp of Ns and the insertion of 600 bp sequences centred on three CTCF sites from the alpha globin locus. Bins refer to the 5 kb genomic bins at increasing linear distance.

Moreover, the larger, homogenous TAD structure is better suited to visualise the variant effect and dissect it from the surrounding architecture. Because we had already designed Tiled-C capture probes over locus 3, we chose it for conducting our genome-wide boundary screen, allowing for potential experimental validation in the future.

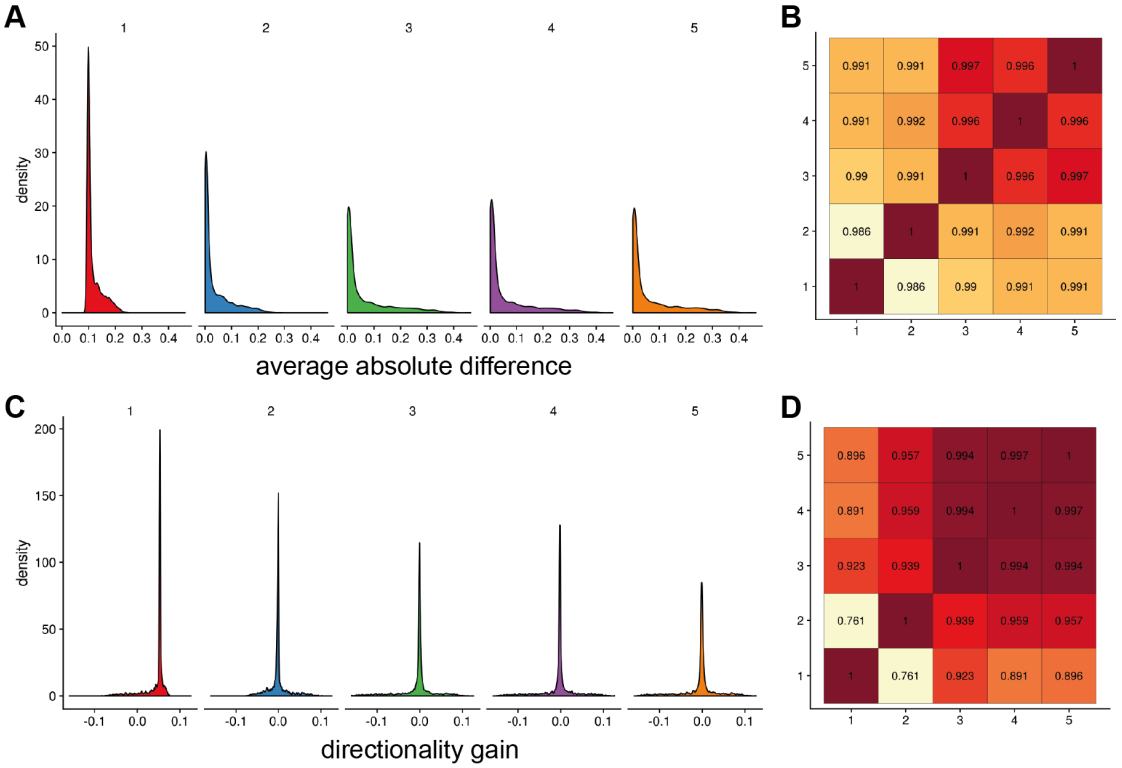

Figure 4.14: Characterising the boundary potential of CTCF sites on chromosome 16. A) Density of average absolute difference per interaction bin upon inserting all CTCF peaks into five genomic loci. B) Pearson correlation matrix between the predicted effects across loci. C) Density of directionality gain measure for predictions as in (A). D) Pearson correlation matrix of directionality gain predictions.

Next we screened all genome wide CTCF sites by computational insertion into locus 3. A total of 79,501 CTCF sites were selected as described for chr16. We predicted and summarised chromatin interaction changes in the K562 model as described above. We then dissected the predicted CTCF boundary strengths considering tissue-specify, strength of CTCF binding and motif directionality. First, we sought to relate the predicted boundary strength to the tissue-specificity of the CTCF

binding sites. We called CTCF peaks from three cell lines: K562, GM12878 and IMR90. We then intersected the peak calls to get unique and union sets for all combinations of cell types. Next, we stratified the predicted boundary strength in the K562 screen to the tissue intersection profile (Figure 4.15 A). We observed that tissue-invariant CTCF sites have on average a stronger predicted boundary effect, while the more dynamic and tissue-specific sites are on average weaker. However, all types of tissue profiles cover a wide range of predicted boundary strengths. Second, we correlated the predicted boundary strengths with the coverage of K562 CTCF ChIP-seq reads (Figure 4.15 B). After log transforming both the coverage and the mean absolute interaction difference, we observe a Pearson correlation of 0.52. The coverage of a ChIP-seq peak is determined by a multitude factors, including molecular but also technical details such as fixation time[132]. Our result suggests that the biological component of peak heights, for example TF binding frequency and residence time, are predictive for the insulating potential of a CTCF site. Third, we considered sequence matches to the CTCF core motif. We identified all matches, their similarity scores and their orientation. We then summarised both using a simple motif match score per CTCF site multiplying each similarity score with +1 for forward orientation (>) and -1 for reverse orientation (<) and calculating the weighted sum. We then plotted the predicted directionality gains upon insertion and compared them to the motif match score (Figure 4.15 C). The majority of CTCF have no clear bias in the directionality of differential interactions. Those that do correlate well with the motif match score, yielding an overall Pearson correlation of 0.52. Taken together tissue invariance, CTCF ChIP-seq peak height, motif similarity and orientation of core motif matches are predictive of the boundary potential of a CTCF site. They do, however, not entirely explain the predicted variability.

Finally, we wanted to assess how strong of an effect single promoter and enhancer insertions could exert relative to CTCF sites. We selected well characterised regulatory elements from the human *HBA* locus and inserted them into loci 4 as

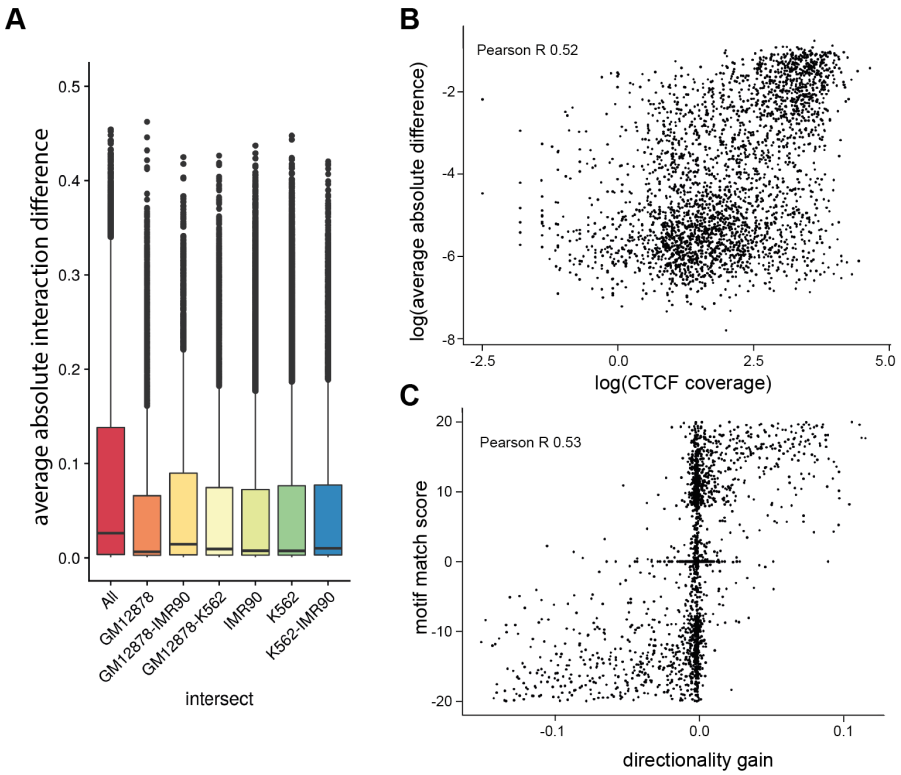

Figure 4.15: Boundary strength vs. CTCF cell-type-specificity, ChIP-seq peak strength and motifs. A) Intersection of CTCF sites with 3 cell types and stratifying this against predicted change in chromatin interactions. B) Correlation of predicted interaction change and CTCF ChIP-seq coverage. C) Correlation of a naïve integrated motif match score, weighted sum of motif similarity times orientation (+1/-1), with directionality gain predictions. Only chr16 results are plotted for clarity but the same trends were observed genome wide.

described above. We chose the human *HBA1* and *HBA2* promoters using 1 kb of sequence (see Methods). Because K562 shows a weak CTCF ChIP-seq peak close to the *HBA1* promoter, we also selected a 250 bp sequence window focused on the TSS and avoiding the CTCF peak. In addition, we selected 500 bp windows centered on the human alpha globin enhancers R1 and R2 (see Methods), the mouse orthologs of which have been characterised as the strongest enhancers of alpha globin expression[133]. In all three promoter insertions we observed a highly local increase of chromatin interactions in the insertion and a weak insulation towards up and downstream (Figure 4.16). This effect was strongest for the *NPRL3* promoter. Although on a much smaller magnitude, these results suggest that promoters may themselves act as weak boundaries in chromatin architecture. Interestingly, when

only querying the TSS focussed 250 bp of the *HBA1* promoter the predicted effect was a stronger gain of interactions with the downstream region, not recapitulating the prediction of the 1 kb promoter fragments. Neither one of the enhancer insertions had a noticeable effect on chromatin interaction predictions, fitting our importance score and deletion screen results.

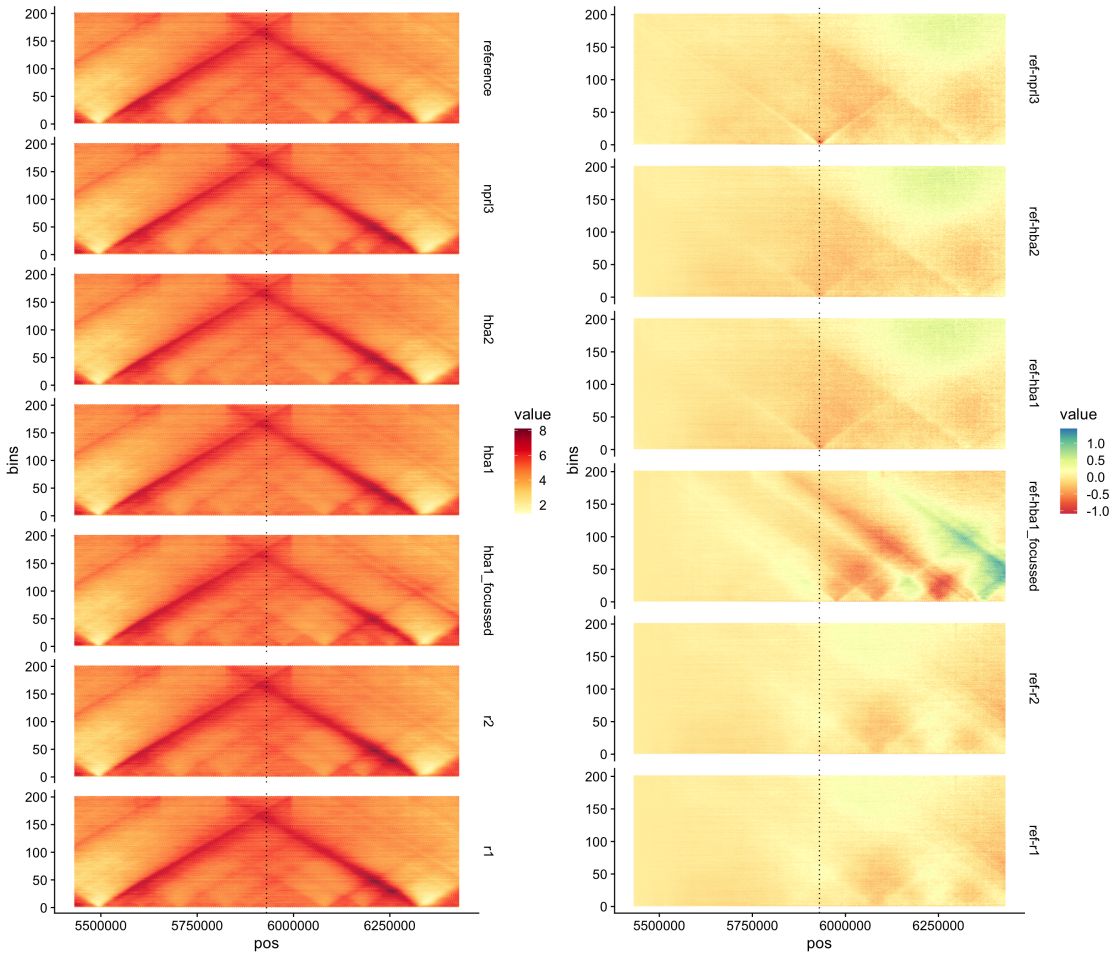

Figure 4.16: Predicting the effect of inserting promoters and enhancers into a locus on chr17. Predicted chromatin interactions using the K562 5 kb model are shown on the left. Differential maps $reference - variant$ are shown on the right. Bins refer to the 5 kb genomic bins at increasing linear distance. The *NPRL3* and the *HBA1* and *HBA2* promoters were selected as 1 kb fragments as well as a 250 bp fragment focussed on the *HBA1* TSS. The two strongest alpha globin enhancers, as characterised in mouse orthologs, R1 and R2 were chosen.

4.3 Discussion

Here, we used deepC trained on high resolution, tissue-specific Hi-C data to dissect the sequence determinants of genome folding computationally. We employed a gradient-based score, the saliency[125], to map the importance of regions and of individual base pairs. We compared the saliency score to a perturbation-based importance score showing similar results, while the saliency score is computationally much more efficient. The saliency score can appear noisy at the base pair resolution and smoothing filters are necessary to assess larger patterns of importance such as element wise importance and meta profiles. More robust methods for importance score calculation have been proposed but come at the cost at significant computational cost such as integrated gradients[134] or cannot be flexibly used for all network architectures[135]. Some require the definition of reference sequences without any features of interest or these features scrambled[135], which is hard to evaluate when the sequence features predictive for an outcome are not known. Genome-wide, we observe sharp peaks of saliency over CTCF sites. Promoters and enhancers overlapping with CTCF sites show a saliency pattern dominated by the sharp CTCF peak. Interestingly, promoters without CTCF overlaps show broader saliency peaks, specifically the ones with histone marks associated with active promoters. In contrast, enhancers show no distinct saliency peaks. Within CTCF sites and promoters, the saliency highlights single TF motifs.

Next, we have utilised deepC, to predict the effect of sequence variants on chromatin architecture. Most computational algorithms developed for predicting genome folding have in fact not been applied to predicting the effect of sequence variants. Only polymer models have been explicitly suggested to model the effect of larger structural variations[98]. We designed deepC with variant effect prediction in mind. The combination of sequence input and large genomic context allows us to predict the effect of genomic variants from SV down to single base pair variation.

We first sought to validate deepC's ability to accurately predict the impact of sequence variants. This is a challenging task since we lack gold standards. Only a few examples of variants with an effect of chromatin architecture have been fully characterised, demonstrating the effect by chromosome confirmation assays in WT and variant. More frequently, the genome folding is only experimentally measured in WT or reference Hi-C maps of related cell types are used. The effect of genomic variants is then frequently suggested by showing overlap with domain boundaries or CTCF sites and motifs. Especially on single base pair level, we lack a coherent understanding of how much single base pair changes are capable of re-configuring large-scale genome folding and how likely and common such events are. We believe deepC offers a tool to address these short comings, by allowing us to quickly predict the effects of variants computationally. We utilised two well characterised examples from the literature to demonstrate that deepC can accurately predict the effect of larger deletions and single base pair changes on genome folding.

Having evaluated deepC's variant effect predictions, we sought to utilise it to dissect the importance of sequence features by comprehensive deletion screens. A long-standing question in the field of genome folding has been which regulatory elements contribute to the formation and maintenance of genomic architecture. We used deepC to perform an *in silico* deletion screen of all accessible chromatin and CTCF bound genomic elements in turn. We found that CTCF sites, especially the strongly bound ones have on average the highest predicted impact on genome folding, followed by weaker CTCF sites and promoter and enhancers with CTCF site overlap. Noteworthy, promoters had on average a larger predicted impact than enhancers. Both within promoters and enhancers, it was the ones marked by histone modifications associated with regulatory activity that had the highest predicted impact on average. These results are in agreement with the genome wide saliency patterns described above. In combination, these results confirm the expected crucial role of CTCF sites for chromatin architecture. Interestingly, we observed that active promoters are also predictive of chromatin structure suggesting an active role in

the establishment and maintenance of genome folding. It is important to note that all sequence patterns and important genomic elements identified are predictive of chromatin architecture. Although this makes causal relationship more likely these correlative results have to carefully assessed and tested experimentally to establish causality.

Next, we focused specifically on CTCF sites aiming to dissect the specificity of CTCF sites potentials to act as boundary to genome folding. CTCF sites have been identified as key factors in shaping chromatin architecture by acting as insulating boundary to Cohesin mediated loop extrusion[35, 136–138]. CTCF sites are found enriched at TAD boundaries and loop domains[14, 24] primarily in a convergent orientation. However, tens of thousands of CTCF bound sites are found in the human genome. While many CTCF sites are conserved across tissues and frequently coincide with conserved chromatin structure, a large proportion of sites is also tissue-specific. Apart from the core CTCF motif, the sequence determinants of CTCF binding are less well characterised. The structure of the CTCF molecule allows flexibility in the recognition of variatiants in the core motif and spacing between additional motifs (reviewed in Wu et al.[139]). A motif upstream of the core binding motif found in 15 % of CTCF binding sites has been implicated in orienting CTCF binding when the core motif is palindromic[138]. The impact of co-binding factors on CTCF specify and insulating properties also requires further characterisation. A plethora of proteins are found co-binding with CTCF many of which have their own enriched sequence binding motifs such as YY1, MYC, MAX and ZNF143[131].

Computational tools such as deepC offer a valuable strategy to dissect the sequence determinants of CTCF binding and boundary potential. We demonstrated how we can query boundary potential for all CTCF binding sites *in silico* to begin to dissect the predictive sequences learned by the model. We stratified CTCF boundary potential by tissue-specificity suggesting that tissue-invariant CTCF sites are on

average stronger boundaries. DeepC further showed that CTCF binding strength, as measured by CTCF ChIP-seq peak coverage, is predictive of boundary potential. This suggests that the boundary potential depends on CTCF binding frequency and/or residence time. Furthermore, we demonstrated that the predicted boundary potential correlated with a naïve integrative motif score that summarises CTCF core motifs strengths and orientations found within each site. This demonstrated that deepC learns a complex grammar of binding sites and not just motif presence. Taken together, deepC predictions offer a rich resource and tool for further investigation of sequence determinants of boundary potential not covered in this work.

Finally, we predicted the boundary potential of active promoters and enhancers *in silico*. It has been suggested that promoters themselves may act as boundaries to loop extrusion[24, 140]. This has not been suggested for enhancers. The promoters of *NPRL3*, *HBA1* and *HBA2* showed insulating potential *in silico* although on much smaller magnitude than CTCF sites. Interestingly, in the case of *HBA2*, the insulating potential depended on a larger genomic fragment of 1 kb that included a weak CTCF peak. The predicted changes upon inserting a smaller, TSS focused fragment, excluding the weak CTCF peak exhibited a reduction of the local insulation and a gain of more wide spread interactions within the farming TAD (Figure 4.16). This may suggest that the sequence determinants predictive for promoter associated insulation are found further away from the TSS coinciding with weak CTCF peaks. In contrast, the different effect of a focused promoter insertion may suggest that core promoters alone harbour a signal for more wide spread chromatin interaction changes, that might be balanced by the more distal promoter signal. However, a dedicated screen for these properties is needed to derive a firm hypothesis. The enhancers we tested computationally showed no insulating potential. This may suggest that they do not act as boundaries themselves or reflects that the contribution of promoters to chromatin architecture is more readily learned by the neural network.

135

Ultimately, genomic engineering experiments will be needed to verify predicted sequence determinants and effect. However, deepC offers a valuable resource for generating hypothesis and effect.

4.4 Data & Code Availability

Deletion results were made available as part of the deepC publication[107]. CTCF screen results are available under: (https://doi.org/10.5281/zenodo.4751244).

4.5 Acknowledgments

All analysis presented in this chapter was done by me. For genomic element annotation I relied on publicly available chromatin feature data from the ENCODE consortium. The example sequence variants for validating deepC were adapted from Hnisz et al.[94] and Schmiedel et al.[123]. GM12878 eQTLs were downloaded from the GTEx Consortium (accessed 01/03/2019).

5
Conclusion & Perspectives

Contents

5.1 Chromatin features, the non-coding genome and GWAS 138
5.2 3D chromatin architecture predictions 141
5.3 Towards integrated models of transcriptional regulation 142
5.4 Interpretability 143
5.5 Conclusion 146

The overarching aim of a large branch of functional genomics research is to interpret sequence variation in the human non-coding genome and understand the molecular mechanisms and contributions to molecular phenotypes and disease. Specifically, we want to understand gene expression and the effect of regulatory variation to map effects to pathways and find drug-able targets. To identify functional regulatory sequence variation in the non-coding genome, researchers need to identify the type of regulatory element the variant overlaps and which molecular mechanism it affects. Furthermore, variants in cis-regulatory elements need to be linked to their effector genes. Cis-regulatory elements and chromatin interactions can be highly cell-type specific. Identifying the correct cell type is therefore crucial to map the functional effects of regulatory sequence variants. Computational and specifically machine learning approaches have been established as valuable tools to facilitate variant prioritisation and interpretation. In the work presented in this

thesis, we have developed, evaluated and applied machine learning approaches to aid researchers in variant effect interpretation.

5.1 Chromatin features, the non-coding genome and GWAS

Directly modelling the effect of a sequence variant on gene expression is challenging because a number of molecular mechanisms may be involved over a large genomic distance, to mediate the effect. Studies that aim to identify direct links between genomic variation and gene expression, such as eQTL studies, usually need to probe a very large number of individuals. Still, the amount of variation identified is not sufficient to directly learn the rules governing the effects on expression. Rather researchers seek to identify the molecular mechanisms and regulatory interactions through and for which eQTLs act as validation of their predicted causal effect. In the absence of comprehensive models of transcriptional regulation and the effect of sequence variants, researchers use proxies for regulatory function and the effect of sequence variants on these proxies as a means of interpreting non-coding variation. Chromatin features such as TF binding, histone modifications and accessible chromatin are used frequently to demarcate and annotate regulatory elements[39, 40]. The overlap with regulatory elements provides information for interpreting non-coding sequence variants. For example, SNPs identified from GWAS are enriched in sites of accessible chromatin[72, 73]. However, localisation does not necessarily imply a mechanistic effect due to the large amount of polymorphism in the human population and the large amount of linkage between sequence variants.

Variants that effect chromatin features are better candidates for causal effects on transcription. Variants may alter TF binding and consequently alter histone modifications and chromatin accessibility. To predict such effects, a variety of computational approaches has been proposed. Most focus on learning patterns in the DNA sequence that are predictive for chromatin features such as open chromatin[46] or TF binding[76] or proxies of TF binding such a DNase footprints[77, 78]. Models

can then be used to predict how disruptive a sequence variant may be by estimating how much less likely a variant renders a DNA sequence pattern to be associated with a predicted chromatin feature.

The regulatory DNA code harbours complex patterns of TF binding sites, for which the sequence preferences of less than 700 out of the 2,000 – 3,000 known human TFs have been characterised[4, 5]. This renders the regulatory grammar challenging to decipher even at the level of individual elements with a DNA sequence context of around 1 kb. Consequently, machine learning approaches suited for learning complex relationships have been employed with ever increasing complexity. Deep learning approaches, specifically CNNs, are the next iteration of algorithms with a higher capacity to learn more complex patterns compared to traditional ML algorithms and an architecture that is intrinsically suited to modelling TF binding to DNA sequences[110].

Caution is warranted when interpreting the predictions made by CNNs as with all pattern discovery algorithms. The identified patterns and relationships are purely correlative. The underlying assumption that justifies their use it that a causal link from DNA sequence to chromatin features and ultimately transcription is the most plausible. For example, DNA sequence determines which TFs bind to the DNA, allowing them to interact with the PIC at promoters to promote transcription. However, this assumption may not always hold true. For example, accessible chromatin and TF binding may influence DNA methylation[141], which in turn may affect mutational frequencies[142] and therefore DNA sequence during evolution at regions with atypical methylation patterns.

Nevertheless, CNNs are the most accurate predictors of chromatin features from DNA sequence to date. Here, we demonstrated that CNNs can learn to predict a variety of chromatin features with high accuracy in a single framework. We showed that CNNs learn complex DNA sequence patterns and combinations thereof

associated with function and could therefore be used to effectively interpret and prioritise non-coding sequence variation from GWAS. A main benefit is that CNNs learn to predict the chromatin context from DNA sequence, alleviating the need to have a comprehensive map of regulatory elements in a region of interest or chromatin data from specific genotypes; allowing us to predict potential gain of function effects leading to cryptic regulatory sites. Importantly, CNNs trained to predict chromatin features in a number of cell types also offer an approach to identifying the correct cell type in which a regulatory variant may exert its effect. By predicting the effect on a chromatin feature proxy, such as accessible chromatin, across cell types, the prediction can be used to identify the cell types where a region of interest is accessible, bound by TF and thus likely to be functional. In this work, we focused on dissecting SNPs associated with red blood cell phenotypes and interpreted them using deepHaem predictions in erythroid progenitors. DeepHaem, especially the extended data compendium version, covers a range of cell types including cell lines and primary tissues in a single model. We believe such models will be extremely helpful for pin-pointing the cell- and tissue-types of interest for interpreting non-coding variation.

To fully characterise the effect of non-coding sequence variants effecting cis-regulatory elements, they need to be evaluated in a framework that considers their 3D interactions in the genome. Currently, researchers rely on publicly available or specifically generated chromatin conformation capture data of varying resolutions. The major challenges are the need to probe chromatin interactions in sufficient numbers of cells in purified populations of the right cell type and achieving sufficient resolution to map chromatin interactions precisely. Furthermore, using interactions data from wild-type cells to interpret the effect of variants assumes that the variant in question does not alter chromatin structure itself. Computational models predicting chromatin architecture in the context of wildtype and variant haplotype would be extremely valuable for the establishment of testable hypotheses before experimental validation, to guide experimental design and so limit the expenditure of the costs

and time required for these demanding assays. Caution is warranted as with all predictive methods since false negatives, SNPs wrongly predicted to have no effect on chromatin features could lead to missed causal SNPs. Overall, we believe the shear number of variants makes computational methods for prioritisation essential to make follow-up analysis and experiments tractable, outweighing the risk of false negatives.

5.2 3D chromatin architecture predictions

Currently existing methods for predicting chromatin architecture are limited by their narrow context, their low resolution or both. High resolution methods that use DNA sequence as input focus on pair-wise interactions ignore the wider genomic context, while polymer models spanning large genomic regions lack the resolution to model individual cis-regulatory elements and the effect of sequence variation within them. The size, hierarchical nature and intricacy of chromatin architecture suggests that computational models should integrate sequence context and regulatory elements over a large scale but also at high resolution.

Here, we have developed deepC, a model with these properties. DeepC can predict chromatin interactions from DNA sequence and predict the effect of sequence variants from larger structural changes down to single base pair variants. Fudenberg et al. have proposed an alternative model to predict chromatin interactions from DNA sequence[110]. Consolidating work is necessary to evaluate the specific advantages and limitations of either approach. Importantly, both works offer a solution to integrating chromatin interactions into a framework for predicting the effect of non-coding sequence variants using DNA sequence. Currently, both models only predict chromatin interactions. Yet, their predictions could in turn be used to refine chromatin feature and gene expression predictions or they could be integrated into a more comprehensive, multi-modal model. Models that predict genome architecture and the effect of sequence variants thereupon are a crucial step towards integrated models of transcriptional regulation. However, they are only a contribution to overarching solutions.

5.3 Towards integrated models of transcriptional regulation

Our overarching aim is to interpret the effect of non-coding sequence variants on gene expression. We strive to develop predictive models that integrate the full range of regulatory mechanisms possible. A first yet critical step is the development of models that integrate chromatin features at distal regulatory elements with the gene promoters they regulate using chromatin interactions as the scaffold for regulatory interactions. Computational models that predict gene expression from promoters have been proposed either using chromatin features or DNA sequence as input[68, 143–147]. These approaches can work well for housekeeping genes but tend to fall short of accurately predicting tissue-specific dynamics of gene expression, likely due to neglecting cis-regulatory elements that are known to facilitate tissue-specific gene expression.

As described above, computational models such as neural networks can be successfully used to predicting chromatin features from DNA sequence at promoters and cis-regulatory elements and to predict the effect of sequence variants on those chromatin features. Two models have been proposed for connecting chromatin feature predictions at cis-regulatory elements to gene expression at promoters in a single framework. Kelley et al. proposed a dilated CNN with a total spatial reach of 32 kb to connect proximal enhancers to promoters[68]. Zhou et al. proposed regularised linear models to integrate chromatin features over spatial range[145]. Although these models represent significant steps forward, they ignore a critical contributor to gene regulation, the chromatin architecture.

Truly integrated models of transcriptional regulation should incorporate genome architecture and evaluate the combined effect of sequence and structural variation on each individual contributor to gene expression and their combined effect. For example, a variant that creates a new enhancer for a gene might not be relevant if its interactions are directed away from the gene in question by a boundary lying

in between. Conversely, disrupting a gene enhancer might not be significant if the gene promoter is interacting with a completely novel set of regulatory elements due to structural rearrangements. Multi-modal models will be key to capturing such interdependence between different molecular mechanism in the genome. Tissue-specificity is another aspect that is poorly explored in current computational models. Currently, the tissue-specificity of chromatin features is frequently modelled independently as separate classes missing out on the chance to learn shared patterns between cell types. The effect of a sequence variant in closely related cell types might be similar, while cell types with more different gene expression profiles and transcription factor repertoires are more likely to be affected differently, for example because the overlapping enhancer is only active in one of the cell types. Innovative solutions to encoding cell identities and integrating these into predictive models of gene regulation may offer a significant boost to their overall performance and interpretability. Wnuk et al. proposed a model to predict accessible chromatin in tumour cells while using the gene expression from RNA-seq as proxy for cell identity[148]. Here, the cell identity was only used as an auxiliary input to the final predictive layer. In future models, the encoding could be integrated more directly into the convolutional architecture of neural networks, modulating what patterns should be prioritised for a prediction. Furthermore, a more sophisticated encoding may allow for extrapolation between cell types offering the possibility to predict chromatin features in cell types not tractable for experiments.

5.4 Interpretability

The high capacity of DNNs that renders them well suited to learn complex patterns also substantially complicates the interpretation of the final models. More traditional ML approaches such as Random Forrest classifiers allow a straightforward extraction of the most important predictive features such as k-mers. In contrast, the multi-layered transformations by convolutional filters render it hard to identify which patterns and relationships a CNN has learned. Three practises are common for interpreting CNNs in genomics:

1. Visualise the first layer of convolutional filters. After a minor transformation[149], these filters take the form of position weight matrices commonly used in the field of genomics and are thus readily relatable to TF motifs identified by orthologous approaches. CNNs trained to predict chromatin features from DNA sequence learn convolutional filters that resemble the TF binding motifs already known[59–61]. Remarkably CNNs trained on a large data compendium learn these motifs for a range of TFs simultaneously. However, visualising convolutional filters past the first layer is nearly intractable, as those involve a combination of all hidden unit outputs of the previous layer, leading to ever more convoluted outputs.

2. Calculating the contribution of every sequence input to the output prediction. By calculating the gradient of a NNs output with respect to the input sequence we can derive an importance score at base pair resolution, termed the saliency[125]. In other words, the saliency measures how much the prediction output would change if a sequence base would be changed infinitesimally away from its representation as a 1 in the hot-encoded sequence. The saliency has the benefit of yielding a score for every input sequence base in one pass through the network and is thus computationally efficient. The saliency score effectively highlights important sequence determinants but has limitations. Depending on the model, the saliency can appear noisy and on-going research is continually proposing ways of making the saliency score more robust[135]. Furthermore, depending on the prediction output, the sign of the saliency score may not readily be interpretable, if it is calculated with respect to a multifaceted output. Lastly, the saliency score does not represent a true biological event. A base pair cannot be changed from a 1 to a 0.9.

3. Saturated *in silico* mutagenesis predicts the effect of all possible base pair changes at every base pair of interest, yielding three predicted changes per position. The predicted changes highlight important sequence determinants. However, one network evaluation per variant is required. Thus, *in silico*

mutagenesis is substantially slower than the saliency score calculation but represents actually possible biological events. In our work, the results from both methods are comparable.

In the work presented here, we have employed multiple techniques to start to interpret both the chromatin features as well as the chromatin interaction networks. However, highlighting the predictive sequence determinants is only the first step in fully understanding trained models. The saliency or *in silico* mutation score alone does not reveal the relationship between patterns learned nor does it uncover the learned weighting for the final predictions. Multiple layers of convolutions make pattern relationships hard to track. One potential solution from the image analysis field, has been to calculate the saliency up to an intermediate layer of interest and project this score to the input features that affect the intermediate output the most [150]. This approach highlights a connection between input features beyond the first convolutional layer, but it does also not reveal exact relationships.

Two principal pathways to more interpretable models are currently emerging in the field of ML and genomics. The first approach is to interpret entire trained models *post hoc*. For example, Alaa et al. developed an approach to train neural networks constructed from G-functions to resemble the output of the original network as closely as possible[151]. G-functions can flexibly take on the form of many other functions by adjustment of their real valued parameters. These parameters can be trained via gradient descent, allowing the G-functions to adapt their shape. This allows networks to be represented as algebraic functions and for relative feature contribution to be interpreted. Such *post hoc* methods, generally have the major advantage of being employable to all existing models, regardless of what considerations to interpretability went into the original model's design. The second approach is to encode restraints into the model architecture that represent biological prior information. Training such models ensures that the constraints can be interpreted as fixed entities and machine learning is used to learn the connections and relative contributions of these fixtures. For example, de Boer et

al. trained a model to predict gene expression in yeast by explicitly encoding TF binding preferences and then learn their relative contribution to gene expression from data[152]. Depending on the level of constraints introduced, such models offer a tractable path to fully interpretable models. However, encoding prior information always risks missing opportunities to learn novel patterns and relationships.

It remains to be explored which type of approaches are most suited for machine learning approaches in genomics or if hybrid solutions can be designed. Future work on machine learning models for functional genomics will demand more consideration of model interpretability both to enable wider biological insights to be derived from the models and to facilitate their translation to applications for biomedicine. Specifically, for models of transcriptional regulation, interpretability considerations will be crucial for effectively connecting the different predictive modalities into a comprehensive useful algorithmic framework.

5.5 Conclusion

In this thesis work, we have explored, developed and utilised deep learning approaches for functional genomics. We re-implemented CNNs to predict chromatin features from DNA sequence and used them to interpret non-coding sequence variants identified by GWAS. We effectively integrated the developed models into a pipeline for prioritisation and functional validation of candidate causal variants. We demonstrated how CNNs can be used as a key computational tool for characterising the regulatory DNA sequence and variation therein.

In addition, we developed a novel CNN architecture to predict chromatin interactions at high resolution and large-scale using DNA sequence at the megabase scale. We significantly extended the utilised DNA sequence context from previously about 1 kb to 1 Mb, while fully integrating the DNA at base pair resolution. We validated the predicted high-resolution architectures using high sensitivity chromatin confirmation assays, establishing that our developed model predicts

intricate chromatin architecture of biological significance. We then used our developed model to probe the sequence determinants of chromatin folding genome-wide and at base pair resolution. Finally, we evaluated the utility of our model to predict the effect of both larger structural and single base pair variations. The developed models bridge the gap from DNA sequence to chromatin architecture, connecting these essential tiers of transcriptional regulation.

Taken together, the thesis work has expanded the utility of deep neural networks for the field of genomics spanning DNA sequence, the epigenome and chromatin architecture.

6 Methods

Contents

6.1 Chromatin feature networks 149
6.1.1 Data pre-processing 149
6.1.1.1 Chromatin feature data 149
6.1.1.2 Pre-processing and data encoding 150
6.1.2 Design & architecture 151
6.1.2.1 Chromatin feature prediction 151
6.1.2.2 Re-implementing existing architectures 151
6.1.2.3 Dilated CNNs 151
6.1.2.4 Optimised CNN architecture - deepHaem . . . 152
6.1.3 Training & evaluation 152
6.1.4 Application 153
6.1.4.1 Running window predictions 153
6.1.4.2 SNP effect predictions deepHaem 153
6.1.4.3 Union open chromatin set 153
6.1.4.4 Comparison to Sasquatch 153
6.1.4.5 Read coverage vs. probability scores 154
6.2 Chromatin interaction networks 154
6.2.1 Data pre-processing 154
6.2.1.1 Hi-C data 154
6.2.1.2 Hi-C data encoding and distance normalisation 154
6.2.2 Design & architecture 155
6.2.2.1 Predicting chromatin interactions 155
6.2.2.2 Transfer learning and architecture – deepC . . 155
6.2.3 Training & evaluation 157
6.2.3.1 Training deepC networks 157
6.2.3.2 Distance-stratified correlation 158
6.2.3.3 Comparison against HiC-Reg 158
6.2.3.4 Re-aligning and downsampling Hi-C data . . . 159

6.2.4 Validation 159
6.2.4.1 Capture-C design 159
6.2.4.2 Cell culture and formaldehyde fixation 159
6.2.4.3 3C library preparation 160
6.2.4.4 NG Capture-C 160
6.2.4.5 Tiled-C 160
6.2.4.6 Data analysis 161
6.2.4.7 Distance-normalised NG Capture-C tracks . . 161
6.2.5 Application 162
6.2.5.1 Predicting changes in chromatin interactions . 162
6.2.5.2 Boundary calling 162
6.2.5.3 Distance-normalised NG Capture-C signal over boundaries 163
6.2.5.4 Virtual4C 163
6.2.5.5 Saliency score 163
6.2.5.6 eQTL data analysis 163
6.2.5.7 Deletion screen 164
6.2.5.8 CTCF boundary potential screen 164
6.2.6 Auxiliary 165
6.2.6.1 Additional chromatin feature data 165
6.2.6.2 Chromatin segmentation 165
6.2.6.3 5C 166
6.2.6.4 SNP sampling 166
6.3 Software & Data 166
6.3.1 Neural networks 166
6.3.2 Additional software 167
6.3.2.1 Tools 167
6.3.2.2 Additional R packages 167
6.3.2.3 Additional Python libraries 167
6.3.3 Data availability 168
6.3.4 Code availability 168

6.1 Chromatin feature networks

6.1.1 Data pre-processing

6.1.1.1 Chromatin feature data

A total of three chromatin feature datasets, two human and one mouse, were collected throughout the course of the thesis work. Multiple experiment types were covered, including DNase-seq, ATAC-seq and ChIP-seq targeting histone modifications and TFs. These collections of datasets are organised as three compendia: 1) First, 918 human experiments used by deepSEA[60] were downloaded. The data were

supplemented with additional erythroid linage data. Five sets of ATAC-seq data from Corces et al.[82], two DNase-seq experiments[78], ten ATAC-seq plus one CTCF ChIP-seq experiment from Downes et al.[74], and in house erythroid differentiation were used. All data used are listed in Appendix A. 2) A second, larger human chromatin feature compendium was assembled using a total of 4,384 publicly available data from the ENCODE[70] and Roadmap[71] data portals. There are also several smaller sources included as part of this compendium (see Appendix A). 3) A total of 1021 publicly available chromatin feature data from the mouse ENCODE portal[153] were collected. Full details of all chromatin feature datasets are available in Appendix A.

For the re-implementation and evaluation of the CNNs and the testing of dilated CNNs, the training data from deepSEA[60] and Basset[61] were downloaded from the respective repositories (http://deepsea.princeton.edu/help/ and https://github.com/davek44/Basset).

6.1.1.2 Pre-processing and data encoding

For the first set of human chromatin features, genomic coordinates of the features were downloaded from the ENCODE and Roadmap data portal (narrowPeaks format, aligned to human reference genome hg19). All additional data were aligned to hg19 using the NGseqBasic pipeline[154] and peaks were called using MACS2[155] as described above. Mouse datasets were retrieved from mouse ENCODE. For histone modification datasets, peak calls were downloaded from the ENCODE data portal. For DNase-seq, ATAC-seq and transcription factor ChIP-seq data, the aligned bam files were downloaded and peak called with MACS2 as described above. Replicates were collapsed into unions. All mouse data used are listed in Supplementary Table 1. For each set, the peak signals were aggregated following the procedure described in Zhou et al.[60]. The genome was split into 200 bp bins. Peak calls were then intersected with these bins and labelled positive for the given feature if at least 50 % of the bin overlapped with the peak. Bins that

did not intersect with at least one peak call were discarded. Only autosomes were used for all analyses. For every sequence with at least one overlapping chromatin feature peak, the 1 kb sequence centered on the labelled genomic bin was extracted. The DNA sequence was stored in one-hot encoding format and the chromatin feature classes in a binary target vector.

6.1.2 Design & architecture

6.1.2.1 Chromatin feature prediction

For the CNN training, the chromatin feature data were provided as 1 kb DNA sequences in one-hot encoding $[observations * length(1000) * bases(4)]$. Each sequence was associated with a binary label vector yielding the observation matrix $[observations * chromatin\ features\ classes]$ where 1 indicates that the sequence's central 200 bp overlapped with the respective chromatin feature and 0 otherwise. The predictive task of each CNN version described below was to, for every sequence input, compute a probability score between 0 and 1 for each chromatin feature. Network performance, the loss, was measured by the sum of the binary cross entropy per chromatin feature class, performing a multi label - multi class classification.

6.1.2.2 Re-implementing existing architectures

The DeepSEA and Basset architectures were reimplemented in Tensorflow[81] v1.8 following the architectures described in the respective publication (see Table 1).

6.1.2.3 Dilated CNNs

The dilated chromatin feature CNNs were constructed using an initial convolutional layer with 300 hidden units (filter size 20) with ReLU activation and max pooling (width & stride = 5), followed by a stack of dilated, gated CNNs[84] with residual connections[63] as described in WaveNet[83] – but with important modifications. No causal masking was applied. Instead, a dilated convolutional filter size of 3 was used. Gated convolutional layers require training double the amount of filter weights but can model more complex functions through their multiplicative units. The residual units allow information to flow more easily through the deep networks

by providing alternative information flow paths to the subsequent convolutional operations[63]. Iterations with varying numbers of hidden units in the convolutional layer (100-300), dilated hidden units (20-300), and the number of dilated layers (4-14) were trained, and the best performing training curves visualised.

6.1.2.4 Optimised CNN architecture - deepHaem

Several iterations of modified architectures were trialled. After hyperparameter optimisation via grid search the best working architecture was selected for all future work, henceforth referred to as deepHaem. The deepHaem architecture consist of five convolutional layers with ReLU activation and max pooling. Convolutional filter widths of 20, 10, 8, 4, 8; with convolutional strides of 1 along the sequence dimension and hidden units of 300, 600, 900, 900, 900 and max pooling windows of 3, 4, 5, 10, 4 with strides equal to the max pooling windows; finally, a single fully connected layer with output neurons with sigmoid activation, one per chromatin feature class. For regularisation, dropout with a rate of 0.2 was applied after every max pooling operation and an L2 norm with a strength (alpha) of 0.001 was added to the binary cross entropy output. Networks were trained to optimise the combined training error with a batch size of 100, a learning rate of 0.0001 and the ADAM optimiser[156] using default parameters. Weighs were initialised using the Glorot normal initialisation[51]. For the large human data compendium, a version of deepHaem with 1000 hidden units per convolutional layers was also trained and evaluated.

6.1.3 Training & evaluation

For evaluating the reimplementation models, the training data provided with the publications of deepSEA and Basset were used. For all subsequent work the data were split into training, test and validation sets on a chromosome basis, using chr11 and 12 for testing and chr15, 16 and 17 for validation. Hyperparameters (learning rate, L2 norm alpha, hidden units, conv. filter widths, dropout rate, batch size) were optimised via grid search. Training was continued until a plateau of training and test error was observed, or early stopping was employed if the test error exceeded

the train loss. The test error was evaluated after every full epoch through all training data. The data from the validation chromosomes were used for evaluation. Predictions for the validation set were computed and the ROC and PRCs with AUCs were calculated per chromatin feature. Macro (mean of individual AUCs) and micro (computing the average of the aggregate of the individual predictions concatinated together) averages were computed for each dataset. Only genomic bins intersecting at least one of the accessible chromatin features were considered for calculating the evaluation metrics when comparing the performance on open chromatin classifiers across trained models using variable datasets.

6.1.4 Application

6.1.4.1 Running window predictions

Chromatin feature probabilities for six classes were predicted for the region on chr16:90,000-243,250 (hg19) in 250 bp increments and visualised below their respective read coverage tracks.

6.1.4.2 SNP effect predictions deepHaem

The effect of a SNP was predicted by computing the network output for the reference sequence and for the variant sequence centered on the SNP of interest. The difference $(reference - variant)$ was computed as damage score per chromatin feature class.

6.1.4.3 Union open chromatin set

As a reference map for SNP intersections, a union set of accessible chromatin and CTCF sites in erythroid progenitors was created using the chromatin features data listed as WIMM data sets in Appendix A. The union set was created using deeptools merge (-d 10).

6.1.4.4 Comparison to Sasquatch

Sasquatch damage scores were calculated as described in Schwessinger et al.[78] using the WIMM erythroid DNase-footprinting dataset, 7-mers and exhaustive scoring. DeepHaem and Sasquatch damage scores were compared by performing

an *in silico* mutagenesis over the R2 enhancer region, predicting the effect of every possible base pair substitution to every reference base in the window of chr16:163,375-163,875 (hg19).

6.1.4.5 Read coverage vs. probability scores

For nine chromatin features, the predicted probabilities across the genome were compared against the read coverage of the input signal. Coverage tracks were created from alignment files using deepTools (bamCoverage)[157] while removing duplicates. The average coverage over each union open chromatin peak was calculated and visualised as $log_{10}(coverage)$ against the deepHaem probability scores.

6.2 Chromatin interaction networks

6.2.1 Data pre-processing

6.2.1.1 Hi-C data

Hi-C data of seven publicly available human cell lines (Rao et al.[14] – GEO: GSE63525) were used. The available 5 kb and 10 kb resolution intra-chromosomal contacts maps were downloaded and normalised using the provided KRnorm factors. In addition, the primary GM12878 replicate was realigned from raw fastqs using HiCPro[158]. The valid Hi-C interactions were downsampled to achieve 1 billion, 100 million and 10 million valid interaction maps. Mouse ES cell Hi-C data from four replicates were retrieved as raw fastqs from Bonev et al.[22] (GEO: GSE96107). These data were then aligned and processed using HiCPro with default parameters, creating 5 kb interaction maps.

6.2.1.2 Hi-C data encoding and distance normalisation

The genome was binned into windows matching the desired Hi-C resolution, 5 kb bins for the majority of analyses. Every DNA window of size 1 Mb + bin size (e.g. 1,005,000 for 5 kb bins) was associated with the chromatin interactions in a vertical, zig-zag pole of bin-to-bin interaction windows over the centre of the sequence window (Figure 3.1). Every bin-to-bin interaction window encodes the Hi-C interaction

frequencies observed between two bin sized windows of increasing distance (up to 1 Mb away). Sliding this DNA window over whole chromosomes recovers the Hi-C map up to 1 Mb distance. Genomic windows whose interaction pole had a median interaction frequency of 0 were excluded from training. To remove the linear proximity signal, the Hi-C data was percentile normalised across individual chromosomes in a distance stratified manner (Figure 3.4). For every bin-to-bin interaction distance, all interaction frequencies over the respective chromosome were grouped into percentiles in a pyramid scheme, decreasing in size from low to high: 2 x 20 %, 4 x 10 %, 4 x 5 %. The identifier of the respective pyramid percentile (1 – 10) was stored and the chromatin interaction network was then trained to predict the percentile identifier as regressor. The transformed Hi-C matrix is referred to as the Hi-C skeleton. In the exploratory phase, Hi-C data were also encoded using equal sized 10 % percentiles and classification and regression approaches were trialed. Regression of the pyramid percentile scheme yielded the best performance and was subsequently used throughout the thesis work.

6.2.2 Design & architecture

6.2.2.1 Predicting chromatin interactions

We experimented with defining the chromatin interaction predicting problem as classification, predicting the percentile identity of the chromatin interactions split into 10 % percentiles, minimising the binary cross-entropy during training. The regression approach yielded much finer interaction predictions at higher accuracy. After pyramid percentile normalisation the percentile identifiers (1-10) were used to train a neural network regressor, minimising the sum of the squared error between the predicted interaction strength and the percentile identifier.

6.2.2.2 Transfer learning and architecture – deepC

We experimented with *ab initio* training but settled on a transfer learning approach as it yielded substantially better predictions (Figure 3.5). A two-step training

process was used:

In the first step, a convolutional neural network was trained to predict chromatin features from 1 kb of DNA sequence as described for deepHaem above. The chromatin feature compendium of 936 human datasets described above was used for training. The network architecture resembles the previously described deepHaem architecture, with minor modifications to the filter and max-pooling: Five convolutional layers (hidden units: 300, 600, 600, 900, 900; filter widths: 8, 8, 8, 4, 4, 4) with ReLU activation, max pooling (widths: 4, 5, 5, 5, 2) and dropout during training (rate: 0.2), followed by a fully connected layer with sigmoid activation yielding chromatin feature class probabilities. The network was trained as the deepHaem network described above.

In the second step, the chromatin interaction network (deepC, Figure 6.1) was trained to predict the percentile normalised Hi-C interaction frequencies from DNA sequence. Input is a DNA sequence window of 1 Mb + 1x Hi-C bin size. The first module consists of five convolutional layers, with ReLU, max pooling and dropout with the dimensions and hyperparameters matching the chromatin feature network trained in the first step. The weights were initialised by transferring the trained weights from the chromatin feature model. Only the convolutional weights and biases were transferred, the fully connected layer from step one was ignored. All chromatin features across cell types and assays were used for pre-training and the same weights were used for seeding the chromatin interaction network training independent of the Hi-C data cell type used for training. Next a 1x1 convolutional layer maps the 900 hidden units from the previous layer to 100 hidden units for the dilated module. The second module is a series of ten dilated, gated 1D convolutional layers with residuals[63, 83, 84] (Figure 6.1) using 100 hidden units and exponentially increasing dilation rates to reach the full DNA sequence context in the last layer (dilation rates: 1, 2, 4, 8, 16, 32, 64, 128, 256, 1). Finally, a fully connected layer is used to output predicted interaction strengths in units matching the percentile

normalisation. The model was trained using the ADAM optimiser (epsilon 0.1 and the tensorflow 1.8 default settings otherwise) with a learning rate of 0.0001 to minimise the sum of squares error between the outputs and the true percentile identities. A GPU memory of 12 GB limited the batch size to 1.

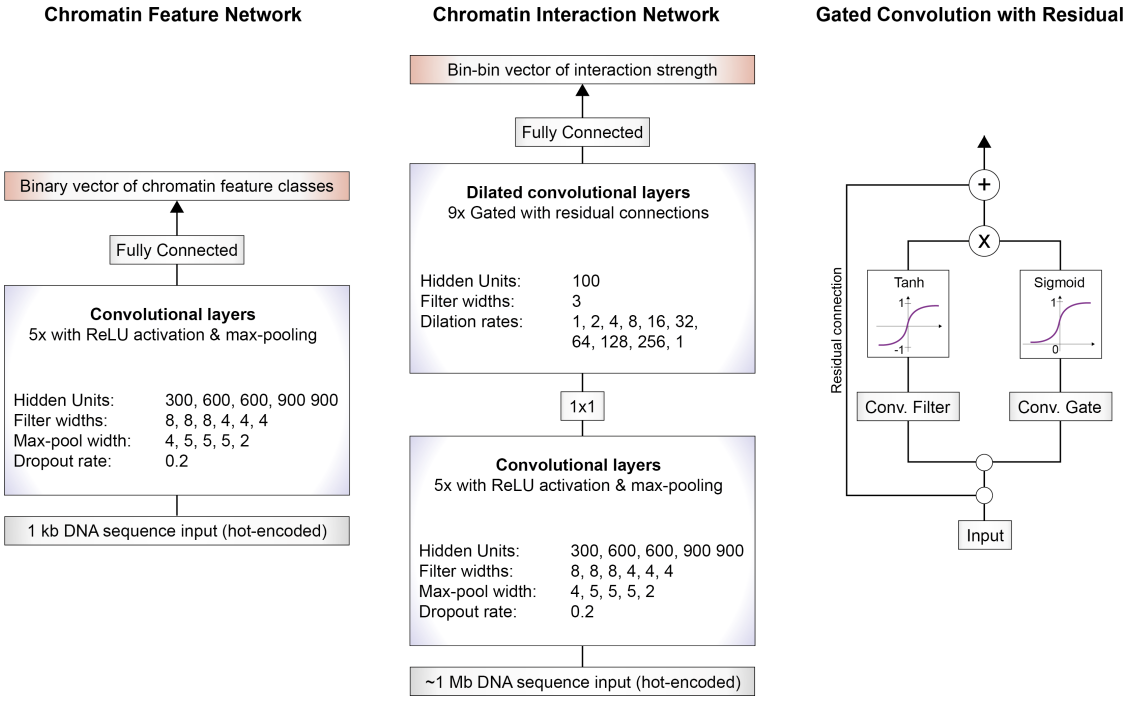

Figure 6.1: Convolutional network architectures employed and detail of the gated convolutions with residuals. The residual units have been adapted from Oord et al.[83]

6.2.3 Training & evaluation

6.2.3.1 Training deepC networks

The hidden units of the dilated layer, the dropout rate, the learning rate and the ADAM epsilon were optimised using grid search. The same L2 regularisation penalty as described for deepHaem was used. For both the chromatin feature and the chromatin interaction training procedures the data were split into training, validation and test chromosomes. For the chromatin feature network, chr11 and 12 were used for validation and chr15, 16 and 17 for testing. For the chromatin interaction network, only chr16 and 17 were used as test chromosomes to increase the number of training examples. All models were trained on NVIDIA Titan V cards

with 12 GB of video memory. Training on smaller cards is possible, but slower. Fully training the chromatin feature network required 14 epochs with about 8 hours per epoch (reshuffling the training data after every epoch). The chromatin interaction networks were trained on one chromosome at a time to make the DNA sequence pre-loading more efficient. Within a chromosome, training example order was chosen at random. Transfer-learning supplemented chromatin interaction networks converged relatively quickly, usually after training on 3 - 6 chromosomes. Models were trained for one full epoch as no significant improvement could be observed after. For cross-validation, multiple training iterations were performed holding out different chromosomes from training. While training networks is only feasible with GPU support, predictions with trained models can be run on CPU only. For example, predicting the impact of a single variant requires ~ 30 sec - 5 min with a GPU, depending if the tensorflow session is started for a single variant or if multiple variant effects are predicted, and ~ 2 h with only CPU support.

6.2.3.2 Distance-stratified correlation

To evaluate the accuracy of the deepC predictions, the Pearson correlation coefficient between the Hi-C skeleton and the deepC predicted regression scores was calculated in a distance stratified manner for each interaction distance in 5 kb bin size steps. Note that the skeleton percentiles are discrete (percentile identifiers 1 – 10), while the regression score is continuous. The correlation was calculated on both raw 5 kb resolution predictions and after applying a mean filter (5x5 bins) to smooth the skeleton.

6.2.3.3 Comparison against HiC-Reg

The available HiC-Reg[103] CrossChrom predictions, trained on chr14 and predicted on chr17, were downloaded from the supplementary material and distance-normalised as described above.

6.2.3.4 Re-aligning and downsampling Hi-C data

The primary GM12878 replicate from Rao et al. was realigned from raw fastqs using HiC-Pro to produce 5 kb interaction maps at default settings. The valid Hi-C interactions were downsampled to 1 billion, 100 million and 10 million valid interactions, respectively.

6.2.4 Validation

6.2.4.1 Capture-C design

DeepC predictions were validated using NG Capture-C[15] in GM12878 and K562 cells. Capture probes were designed for 220 viewpoints, specifically where the Hi-C data and deepC predictions differed in detail or where the deepC predicted structures were only very faintly noticeable in the Hi-C data. A set of 81 CTCF sites and 139 intra domain viewpoints within distinct interaction domains were created, avoiding intersections with potential functional elements as determined by the respective GM12878 and K562 open chromatin peaks. Capture probes were designed using CapSequm, removing repetitive probe regions as described in the online documentation http://apps.molbiol.ox.ac.uk/CaptureC/cgi-bin/CapSequm.cgi. The final probe design is detailed in Appendix B.

6.2.4.2 Cell culture and formaldehyde fixation

The human GM12878 lymphocyte cell line was obtained from the NIGMS Human Genetic Cell Repository (Coriell Institute for Medical Research). Cells were cultured in RPMI 1640 supplemented with 15 % Fetal Bovine Serum (FBS), 2mM L-Glutamine and 100 U/ml Pen-Strep at 37 °C in a 5 % CO_2 incubator. K562 cells were supplied by the WIMM transgenics facility. Cells were maintained in RPMI 1640 media supplemented with 10 % FBS at 37 °C in a 5 % CO_2 incubator. Both cell types were processed following the same protocol. Cells were resuspended at 1x106 cells per ml and fixed at room temperature with 2 % v/v formaldehyde for 10 minutes. Fixation was quenched with 120 mM glycine. Cells were washed with ice cold phosphate buffered saline (PBS). Cells were resuspended in cold lysis

buffer (10 mM Tris, 10 mM NaCl, 0.2 % Igepal CA-630 and complete proteinase inhibitor (Roche)) and snap frozen to -80 °C.

6.2.4.3 3C library preparation

3C libraries were prepared as described in Davies et al.[15] with the following modifications: Centrifugations were performed at 500 rcf, thermomixer incubations were set to 500 rpm, and ligation chromatin was pelleted by centrifugation (15 min, 4 °C, 500 rcf). The supernatant was discarded. To increase sequencing depth and minimise PCR duplicates, experiments were performed in technical triplicates and four unique adapters were used per replicate.

6.2.4.4 NG Capture-C

Double capture was performed as described in Davies et al.[15] with biotinylated oligonucleotides (IDT xGen Lockdown Probes) in two pools (CTCF and intra domain) with 3 pg of each oligonucleotide per 3C library. The generated NG Capture-C libraries were sequenced using Illumina sequencing platforms (V2 chemistry; 150-bp paired-end reads) and data collected using the NextSeq System Suite (v2). To resolve even subtle changes in chromatin interaction domains at high resolution, the libraries were deeply sequenced (GM12878 CTCF ~ 128 M; GM12878 intra domain ~ 118 M reads; K562 CTCF ~ 302 M; K562 intra domain ~ 289 M reads). All technical replicates were merged for the analysis.

6.2.4.5 Tiled-C

Tiled oligonucleotides were devised using the design approach and tool described in Oudelaar and Beagrie et al.[112] (https://oligo.readthedocs.io/en/latest/). A panel of double-stranded capture oligonucleotides from Twist Bioscience (Custom probes for NGS target enrichment) was used. The Tiled-C procedure was performed as described in Oudelaar and Beagrie et al. using the 3C libraries from GM12878 and K562 cells. All technical replicates were merged for the analysis. Three loci were selected for validation by Tiled-C: 1) chr17:7,950,000-9,500,000 over *NDEL1* with

3,790 probes 2) chr16:80,850,000-81,940,000 over *CMIP* with 2,582 probes and 3) chr2:65,750,000-66,850,000 over *MEIS1* with 2,950 probes.

6.2.4.6 Data analysis

NG Capture-C data were mapped to hg19, quality controlled and visualised using the CCseqBasicS[159] pipeline (https://github.com/Hughes-Genome-Group/CCseqBasicS) following the procedure described in Davies et al.[15]. Tiled-C data were mapped to hg19 and quality controlled using the tiled mode of CCseqBasicS outlined in Oudelaar and Beagrie et al.[112]. Each region was ICE normalised using the ICE implementation of HiC-Pro[158] with default parameters.

6.2.4.7 Distance-normalised NG Capture-C tracks

To compare NG Capture-C to the Hi-C skeleton and deepC prediction, the NG Capture-C viewpoint tracks were distance normalised. For each viewpoint and for each restriction enzyme fragment, the number of interactions was extracted and the distance to the viewpoint recorded. The interactions were then normalised for the total number of cis interactions for each respective viewpoint. Pooled over all viewpoints, the distance decay in interaction frequency relation follows approximately a log – log linear trend when split into three distance bins (close, intermediate and far). The distance thresholds for these bins were empirically optimised for every NG Capture-C set produced (GM12878 - CTCF & intra domain: 125,000, 1,250,000; K562 - CTCF: 170,000, 3,250,000; K562 – intra domain: 40,000, 3,500,000 [bp]), excluding all interactions closer than 2.5 kb to the respective viewpoint. The distance decay was then approximated by three linear regression fits, one for each distance bin per cell type and Capture set. The distance-normalised interactions were calculated per viewpoint by dividing the observed cis normalised interactions with the expected interactions from the linear fit at the respective distance.

6.2.5 Application

6.2.5.1 Predicting changes in chromatin interactions

Changes in chromatin interactions were predicted as the difference between the reference and variant deepC predictions, covering all reference sequence interactions within 1 Mb (plus 1x Hi-C bin size) of the sequence variant. The variant interactions were retrieved by modifying the DNA sequence to match the respective variant. The difference was then quantified by calculating the absolute difference between reference and variant prediction at each interaction bin and summarised as the mean absolute difference over all covered interaction windows. In addition, the directionality gain centered on a variant was calculated as the directionality score $reference \breve{} variant$, where the directionality score is the difference between the interactions upstream and downstream from a viewpoint of interest.

6.2.5.2 Boundary calling

Chromatin boundaries were called using the Hi-C and Hi-C skeleton transformed data and the deepC predictions. Two approaches were used: an insulation score-based algorithm and TopDom[114]. The insulation score-based boundary calling was adapted from Crane et al.[30] for higher resolution calls. The mean insulation score profile was calculated based on a 25 kb window. The first derivative (delta) of the insulation score profile was approximated using a 1D Sobel operator. Zero crossings in the first derivative were identified and those corresponding to local maxima in the insulation score profile were discarded. In addition, the second order derivative was approximated by applying a 1D Sobel operator to delta, yielding delta2. The putative boundaries were filtered based on delta2 reflecting the sharpness of a boundary. Boundaries with a delta2 score smaller than 0.1 were discarded and the identified boundaries were stratified based on their delta2 score. TopDom was retrieved from the gitHub implementation[160] (https://github.com/HenrikBengtsson/TopDom). Boundaries were called using the window parameters 5, 10 and 20. Junctions between all types of TopDom identified domains were used as boundaries.

6.2.5.3 Distance-normalised NG Capture-C signal over boundaries

Over the identified chromatin boundaries, the average, distance-normalised Capture-C interaction profiles were calculated relative to the boundary centres. The profiles were stratified over the Capture-C datasets and boundary sharpness. Boundaries that were more than 1 Mb away from a viewpoint were excluded.

6.2.5.4 Virtual4C

Virtual4C profiles from the Hi-C skeleton and deepC predictions were created by extracting all interactions with the viewpoint bin of interest, yielding distance-normalised virtual4C profiles. The Pearson correlation between the distance-normalised NG Capture-C tracks and the virtual4C profiles from the Hi-C skeleton and deepC predictions were calculated respectively for all interactions up to 1 Mb from the viewpoint. The skeleton virtual 4C profiles were smoothed using a running mean (25 kb smoothing window).

6.2.5.5 Saliency score

The saliency score approach was adapted from image analysis[125]. The gradient of the model output, here the interaction frequency predictions, was calculated with respect to the DNA sequence input. The saliency score is the dot product of the gradient with the one-hot encoded DNA sequence, effectively masking the impact of non-present bases. For a given DNA sequence window the saliency score relates to the interactions over the central vertical interaction pole. Saliency tracks were computed by moving the sequence window in 5 kb increments. At every base pair the saliency score was averaged over all sequence that include the respective base pair. To simplify visualisation and interpretation, the absolute value of the saliency score was used.

6.2.5.6 eQTL data analysis

EBV transformed lymphocyte specific eQTLs were retrieved from GTEx (v7 accessed from the GTEx portal 01/03/2019). The eQTL SNPs were filtered for intersection

with a union set of GM12878 CTCF ChIP-seq and DNase-seq peaks. Indels were removed yielding a total of 6,607 SNPs. A matching set background control SNPs was sampled by shuffling the eQTL SNP positions on each respective chromosome and forcing them to fall within the union peak set of DNase-seq – CTCF ChIP-seq peaks created using bedtools[161] (bedtools shuffle -chrom -incl), and randomly sampling one of the three possible base substitutions at each position relative to the reference base. The absolute saliency scores over the SNP bases were calculated from the 5 kb GM12878 model, and the empirical cumulative distributions for eQTL and background set were derived and tested for significance using a two sample Kolmogorov-Smirnov test (R, ks.test, reshuffled SNP saliency (n = 6,607) vs. eQTL set (n = 6,607) saliency, alternative hypothesis: "less").

6.2.5.7 Deletion screen

Separate DNase-seq and CTCF ChIP-seq peak sets for GM12878 were produced. Within each set, peaks overlapping or within 1.5 kb of one another were merged (bedtools merge -d 1500). All peaks were extended to a minimum of 300 bp. DNase peaks overlapping with CTCF peaks were removed from the DNase set. This resulted in a total of 45,635 CTCF and 47,320 DNase candidate elements. The 5 kb GM12878 model was then used to predict the effect of deleting every site individually. Chromatin segmentation classes were assigned based on overlap with the GenoSTAN chromatin segmentation. As background control, 3,850 sites were selected by shuffling all CTCF and open chromatin sites on chr16 forcing no overlap (bedtools shuffle -chrom -noOverlapping).

6.2.5.8 CTCF boundary potential screen

The CTCF boundary potential screen was performed using the 5 kb K562 model. For the initial screen, all 2,932 CTCF ChIP-seq peaks were used from the K562 data on chr16 with a 600 bp sequence window centred on each peak. The impact on chromatin architecture was predicted for every peak relative to a 600 bp insertion of N's. The initial screen was conducted in 5 genomic loci: 1) chr16:197,600-197,600; 2) chr17:8,373,400-8,374,000; 3) chr17:9150000-9150600; 4) chr17:5,930,000-5,930,600;

5) chr11:110,750,000-110,750,600. The mean absolute interaction difference and directionality gain was calculated for every insertion as described above, and the Pearson correlation between predictions for the five loci were computed. The genome wide screen was performed in loci (3) using a total of 79,501 sites. K562 CTCF sites were intersected with GM12878 and IMR90 CTCF ChIP-seq peaks to stratify them based on tissue-specificity. The K562 CTCF ChIP-seq read coverage at coordinates of peaks was calculated using deepTools[157].

6.2.6 Auxiliary

6.2.6.1 Additional chromatin feature data

Chromatin feature data from GM12878, K562 and IMR90 were downloaded as read-depth normalised bigwig tracks for visualisation. For genome segmentation, GM12878 and K562 chromatin data were downloaded from the ENCODE data portal as duplicate filtered alignment files (bam). Accession details are listed in Table 6.1. Available replicates were merged.

Table 6.1: Additional chromatin feature data used for visualisation and chromatin segmentation

Experiment ID	Cell type	Target - assay
ENCSR000EMT	GM12878	DNase-seq
ENCSR000AKB	GM12878	CTCF ChIP-seq
ENCSR057BWO	GM12878	H3K4me3 ChIP-seq
ENCSR000AKF	GM12878	H3K4me1 ChIP-seq
ENCSR000AKC	GM12878	H3K27ac ChIP-seq
ENCSR921NMD	K562	DNase-seq
ENCSR000AKO	K562	CTCF ChIP-seq
ENCSR668LDD	K562	H3K4me3 ChIP-seq
ENCSR000AKS	K562	H3K4me1 ChIP-seq
ENCSR000AKP	K562	H3K27ac ChIP-seq
ENCSR000DTW	HEK293	CTCF ChIP-seq
ENCSR000EFI	IMR90	CTCF ChIP-seq

6.2.6.2 Chromatin segmentation

For all GM12878 and K562 chromatin data listed in Table 2 peaks were called using MACS2 (Y. Zhang et al. 2008) with default settings and -q 0.01. Peaks from DNase-seq and CTCF ChIP-seq were merged into a union set using bedtools (Quinlan

and Hall 2010) (bedtools merge -d 10) and extended to 600 bp centered on the respective union peak. The read coverage over each union peak was calculated using deeptools multiBamSummary, considering an additional 200 bp window at each site of the union peak to better capture histone modification reads. Based on the count matrix, segmentation into chromatin classes was performed using GenoSTAN (Zacher et al. 2017). Rather than using a running window approach, GenoSTAN was used on a per-element basis. The HMM model was trained using the Poisson log-normal distributions, fitting twelve numericly labeled classes and manually curating them into eleven classes, merging highly similar chromatin signatures. The classes were manually classified into promoter, enhancer and CTCF sites with varying activity levels based on H3K27ac coverage. Union peaks that did not sow significant enrichment of any ChIP-seq signal used were classified as None.

6.2.6.3 5C

Processed 5C data from Hnisz et al.[94] were downloaded (GEO: GSE77142), binned into 5 kb windows and visualised.

6.2.6.4 SNP sampling

1000 SNPs were sampled randomly using a uniform distribution from strong CTCF sites, strong promoters and strong enhancers, as classified by the chromatin segmentation. An additional 1000 SNPs were sampled as a background set from the flanking regions (400 bp) of the regulatory elements while avoiding overlap with other elements. SNPs positions were sampled using bedtools shuffle, and variant bases were randomly selected from the three bases not present in the hg19 reference at the respective position.

6.3 Software & Data

6.3.1 Neural networks

All neural networks were implemented in python (v3.5) and tensorflow[81] (developed under 1.8.0). Compatibility versions for tensorflow 1.14, 2.0 and 2.1+ were

created subsequently.

6.3.2 Additional software

6.3.2.1 Tools

- samtools[162] (v1.3)
- FastQC (v0.11.4) (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)
- Bowtie[163] (v1.1.2)

6.3.2.2 Additional R packages

- cowplot (v0.6.2) (https://github.com/wilkelab/cowplot)
- GenomicRanges[164] (v1.30.3)
- ggplot2[165] (v3.1.0)
- RcolorBrewer (v1.1.1-2)
  (https://cran.r-project.org/web/packages/RColorBrewer/index.html)
- rtracklayer[166] (v1.30.4)
- tidyverse (v1.3.0) (https://www.tidyverse.org)
- zoo[167] (v1.8.1)

6.3.2.3 Additional Python libraries

- numpy[168] (1.16.4)
- h5py (v2.9.0) (http://www.h5py.org)
- pysam (0.15.2) (https://github.com/pysam-developers/pysam)

6.3.3 Data availability

Details of the publicly available chromatin feature data from ENCODE, Roadmap and other sources are listed in Appendix A. Capture-C Validation probe design is available in Appendix B. (https://doi.org/10.5281/zenodo.4751244). Tiled-C and NG Capture-C validation data are available under the GEO super series GSE137437.

6.3.4 Code availability

All code used for data pre-processing, model construction and training are publicly available via github: DeepHaem (https://github.com/rschwess/deepHaem) & DeepC (https://github.com/rschwess/deepC). The deepC repository also hosts trained models and tutorials.

References

1. Krebs, J. E., Goldstein, E. S. & Kilpatrick, S. T. *Lewin's Genes XI* 11th (2011).
2. Heinz, S., Romanoski, C. E., Benner, C. & Glass, C. K. The selection and function of cell type-specific enhancers. *Nature Reviews Molecular Cell Biology* **16,** 144–154 (2015).
3. Long, H. K., Prescott, S. L. & Wysocka, J. Ever-Changing Landscapes: Transcriptional Enhancers in Development and Evolution. *Cell* **167,** 1170–1187 (2016).
4. Vaquerizas, J. M., Kummerfeld, S. K., Teichmann, S. A. & Luscombe, N. M. A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics* **10,** 252–263 (2009).
5. Kulakovskiy, I. V. *et al.* HOCOMOCO: Towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Research* **46,** 252–259 (2018).
6. Kim, T.-K. & Shiekhattar, R. Architectural and Functional Commonalities between Enhancers and Promoters. *Cell* **162,** 948–959 (2015).
7. Kleinjan, D. A. & Lettice, L. A. Long-range gene control and genetic disease. *Advances in Genetics* **61,** 339–88 (2008).
8. Song, L. & Crawford, G. E. DNase-seq: A High-Resolution Technique for Mapping Active Gene Regulatory Elements across the Genome from Mammalian Cells. *Cold Spring Harbor Protocols,* 5384 (2010).
9. Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y. & Greenleaf, W. J. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nature Methods* **10,** 1213–8 (2013).
10. Barski, A. *et al.* High-resolution profiling of histone methylations in the human genome. *Cell* **129,** 823–837 (2007).
11. Johnson, D. S., Mortazavi, A., Myers, R. M. & Wold, B. Genome-wide mapping of in vivo protein-DNA interactions. *Science* **316,** 1497–1502 (2007).
12. Bannister, A. J. & Kouzarides, T. Regulation of chromatin by histone modifications. *Cell Research* **21,** 381–395 (2011).
13. Lieberman-Aiden, E. *et al.* Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science* **326,** 289–293 (2009).
14. Rao, S. S. *et al.* A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell* **159,** 1665–1680 (2014).
15. Davies, J. O. J. *et al.* Multiplexed analysis of chromosome conformation at vastly improved sensitivity. *Nature Methods* **13,** 74–80 (2016).

16. Beagrie, R. A. *et al.* Complex multi-enhancer contacts captured by genome architecture mapping. *Nature* **543,** 519–524 (2017).

17. Flyamer, I. M. *et al.* Single-nucleus Hi-C reveals unique chromatin reorganization at oocyte-to-zygote transition. *Nature* **25,** 82 (2017).

18. Sunwoo, H. *et al.* Polycomb Repressive Complex 1 Generates Discrete Compacted Domains that Change during Article Polycomb Repressive Complex 1 Generates Discrete Compacted Domains. *Molecular Cell* **65,** 432–445.e6 (2017).

19. Fraser, J. *et al.* Hierarchical folding and reorganization of chromosomes are linked to transcriptional changes in cellular differentiation. *Molecular Systems Biology* **11,** 1–14 (2015).

20. Dixon, J. R., Gorkin, D. U. & Ren, B. Chromatin Domains: The Unit of Chromosome Organization. *Molecular Cell* **62,** 668–680 (2016).

21. Bonev, B. & Cavalli, G. Organization and function of the 3D genome. *Nature Review Genetics* **17,** 661–678 (2016).

22. Bonev, B. *et al.* Multiscale 3D Genome Rewiring during Mouse Neural Development. *Cell* **171,** 557–572.e24 (2017).

23. Kalhor, R., Tjong, H., Jayathilaka, N., Alber, F. & Chen, L. Genome architectures revealed by tethered chromosome conformation capture and population-based modeling. *Nature Biotechnology* **30,** 90–98 (2011).

24. Dixon, J. R. *et al.* Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485,** 376–380 (2012).

25. Dowen, J. M. *et al.* Control of cell identity genes occurs in insulated neighborhoods in mammalian chromosomes. *Cell* **159,** 374–387 (2014).

26. Ji, X. *et al.* 3D Chromosome Regulatory Landscape of Human Pluripotent Cells. *Cell Stem Cell* **18,** 262–275 (2016).

27. Hnisz, D., Day, D. S. & Young, R. A. Insulated Neighborhoods: Structural and Functional Units of Mammalian Gene Control. *Cell* **167,** 1188–1200 (2016).

28. Phillips-Cremins, J. E. *et al.* Architectural protein subclasses shape 3D organization of genomes during lineage commitment. *Cell* **153,** 1281–1295 (2013).

29. Oudelaar, A. M. *et al.* A revised model for promoter competition based on multi-way chromatin interactions at the α-globin locus. *Nature Communications* **10,** 1–8 (2019).

30. Crane, E. *et al.* Condensin-driven remodelling of X chromosome topology during dosage compensation. *Nature* **523,** 240–244 (2015).

31. Zufferey, M., Tavernari, D., Oricchio, E. & Ciriello, G. Comparison of computational methods for the identification of topologically associating domains. *Genome Biology* **19,** 217 (2018).

32. Downes, D. J. *et al.* High-resolution targeted 3C interrogation of cis-regulatory element organization at genome-wide scale. *Nature Communications* **12,** 531 (2021).

33. Nichols, M. H. & Corces, V. G. A CTCF Code for 3D Genome Architecture. *Cell* **162,** 703–705 (2015).

34. Canela, A. *et al.* Genome Organization Drives Chromosome Fragility. *Cell* **170,** 507–521.e18 (2017).

35. Sanborn, A. L. *et al.* Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. *Proceedings of the National Academy of Sciences of the United States of America* **112,** E6456–E6465 (2015).

36. Fudenberg, G. *et al.* Formation of Chromosomal Domains by Loop Extrusion. *Cell Reports* **15,** 2038–2049 (2016).

37. Davidson, I. F. *et al.* DNA loop extrusion by human cohesin. *Science* **366,** 1338–1345 (2019).

38. Bailey, T. L. *et al.* MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Research* **37,** W202–W208 (2009).

39. Ernst, J. & Kellis, M. ChromHMM: automating chromatin-state discovery and characterization. *Nature Methods* **9,** 215–216 (2012).

40. Zacher, B. *et al.* Accurate Promoter and Enhancer Identification in 127 ENCODE and Roadmap Epigenomics Cell Types and Tissues by GenoSTAN. *PLOS ONE* **12,** e0169249 (2017).

41. Karlić, R., Chung, H.-R., Lasserre, J., Vlahovicek, K. & Vingron, M. Histone modification levels are predictive for gene expression. *Proceedings of the National Academy of Sciences of the United States of America* **107,** 2926–2931 (2010).

42. Rentzsch, P., Witten, D., Cooper, G. M., Shendure, J. & Kircher, M. CADD: Predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Research* **47,** D886–D894 (2019).

43. Cortes, C. & Vapnik, V. Support-vector networks. *Machine Learning* **20,** 273–297 (1995).

44. Libbrecht, M. W. & Noble, W. S. Machine learning applications in genetics and genomics. *Nature Reviews Genetics* **16,** 321–332 (2015).

45. Fletez-Brant, C., Lee, D., McCallion, A. S. & Beer, M. a. kmer-SVM: a web server for identifying predictive regulatory sequence features in genomic data sets. *Nucleic Acids Research* **41,** 544–556 (2013).

46. Lee, D. *et al.* A method to predict the impact of regulatory variants from DNA sequence. *Nature Genetics* **47,** 955–961 (2015).

47. Belokopytova, P. S., Nuriddinov, M. A., Mozheiko, E. A., Fishman, D. & Fishman, V. Quantitative prediction of enhancer–promoter interactions. *Genome Research* **30,** 72–84 (2020).

48. Farley, B. & Clark, W. Simulation of self-organizing systems by digital computer. *Transactions of the IRE Professional Group on Information Theory* **4,** 76–84 (1954).

49. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* **65,** 386–408 (1958).

50. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323,** 533–536 (1986).

51. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research* **9,** 249–256 (2010).

52. Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S. Self-Normalizing Neural Networks. *Advances in Neural Information Processing Systems* **Dec,** 972–981. arXiv: 1706.02515 (2017).

53. Clevert, D. A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings.* arXiv: 1511.07289 (2016).

54. Xiong, H. Y. *et al.* The human splicing code reveals new insights into the genetic determinants of disease. *Science* **347** (2015).

55. LeCun, Y. *et al.* Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* **1,** 541–551 (1989).

56. Jarrett, K., Kavukcuoglu, K., Ranzato, M. & LeCun, Y. What is the best multi-stage architecture for object recognition? *Proceedings of the IEEE International Conference on Computer Vision,* 2146–2153 (2009).

57. Zeiler, M. D. & Fergus, R. Visualizing and Understanding Convolutional Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **8689,** 818–833. arXiv: 1311.2901 (2013).

58. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* **60,** 84–90 (2017).

59. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* **33,** 831–838 (2015).

60. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods* **12,** 931–934 (2015).

61. Kelley, D. R., Snoek, J. & Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research* **26,** 990–999 (2016).

62. Szegedy, C. *et al.* Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 1–9. arXiv: 1409.4842 (2015).

63. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *Multimedia Tools and Applications* **77,** 10437–10453. arXiv: 1512.03385 (2015).

64. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **9,** 1735–1780 (1997).

65. Quang, D. & Xie, X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Research* **1,** 032821 (2016).

66. Singh, S., Yang, Y., Poczos, B. & Ma, J. Predicting Enhancer-Promoter Interaction from Genomic Sequence with Deep Neural Networks. *bioRxiv* (2016).

67. Yu, F. & Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. arXiv: 1511.07122 (2015).

68. Kelley, D. R. *et al.* Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Research* **28,** 739–750 (2018).

69. Schaid, D. J., Chen, W. & Larson, N. B. From genome-wide associations to candidate causal variants by statistical fine-mapping. *Nature Reviews Genetics* **19,** 491–504 (2018).

70. Bernstein, B. E. *et al.* An integrated encyclopedia of DNA elements in the human genome. *Nature* **489,** 57–74 (2012).

71. Kundaje, A. *et al.* Integrative analysis of 111 reference human epigenomes. *Nature* **518,** 317–330 (2015).

72. Hindorff, L. A. *et al.* Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proceedings of the National Academy of Sciences of the United States of America* **106,** 9362–9367 (2009).

73. Maurano, M. T. *et al.* Systematic localization of common disease-associated variation in regulatory DNA. *Science* **337,** 1190–1195 (2012).

74. Downes, D. J. *et al.* An integrated platform to systematically identify causal variants and genes for polygenic human traits. *bioRxiv,* 813618 (2019).

75. Stenson, P. D. *et al.* Human Gene Mutation Database (HGMD ®): 2003 update. *Human Mutation* **21,** 577–581 (2003).

76. Coetzee, S. G., Coetzee, G. A. & Hazelett, D. J. MotifbreakR: An R/Bioconductor package for predicting variant effects at transcription factor binding sites. *Bioinformatics* **31,** 3847–3849 (2015).

77. Moyerbrailean, G. A. *et al.* Which Genetics Variants in DNase-Seq Footprints Are More Likely to Alter Binding? *PLOS Genetics* **12,** e1005875 (2016).

78. Schwessinger, R. *et al.* Sasquatch: predicting the impact of regulatory SNPs on transcription factor binding from cell- and tissue-specific DNase footprints. *Genome Research* **27,** 1730–1742 (2017).

79. Guo, Y. *et al.* CRISPR-mediated deletion of prostate cancer risk-associated CTCF loop anchors identifies repressive chromatin loops. *Genome Biology* **19,** 160 (2018).

80. Kircher, M. *et al.* Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. *Nature Communications* **10,** 3583 (2019).

81. Abadi, M. *et al.* TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16),* 265–284. arXiv: `1605.08695` (2016).

82. Corces, M. R. *et al.* Lineage-specific and single-cell chromatin accessibility charts human hematopoiesis and leukemia evolution. *Nature Genetics* **48,** 1193–1203 (2016).

83. Van den Oord, A. *et al.* WaveNet: A Generative Model for Raw Audio. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing,* 3437–3440. arXiv: `1609.03499` (2016).

84. Van den Oord, A. *et al.* Conditional Image Generation with PixelCNN Decoders. arXiv: `1606.05328` (2016).

85. Astle, W. J. *et al.* The Allelic Landscape of Human Blood Cell Trait Variation and Links to Common Complex Disease. *Cell* **167,** 1415–1429.e19 (2016).

86. Wang, K., Li, M. & Hakonarson, H. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* **38,** e164–e164 (2010).

87. Jaganathan, K. *et al.* Predicting Splicing from Primary Sequence with Deep Learning. *Cell* **176,** 535–548.e24 (2019).

88. Ran, F. A. *et al.* Genome engineering using the CRISPR-Cas9 system. *Nature Protocols* **8,** 2281–2308 (2013).

89. De Gobbi, M. *et al.* A regulatory SNP causes a human genetic disease by creating a new transcriptional promoter. *Science* **312,** 1215–1217 (2006).

90. Krasimirov Bozhilov, Y. *Analysis of a gain-of-function mutation that produces a new transcriptional unit in the α-globin locus* Doctoral Thesis (University of Oxford, 2017).

91. Wesolowska-Andersen, A. *et al.* Deep learning models predict regulatory variants in pancreatic islets and refine type 2 diabetes association signals. *eLife* **9** (2020).

92. Ludwig, L. S. *et al.* Transcriptional States and Chromatin Accessibility Underlying Human Erythropoiesis. *Cell Reports* **27,** 3228–3240.e7 (2019).

93. Krijger, P. H. L. & de Laat, W. Regulation of disease-associated gene expression in the 3D genome. *Nature Reviews Molecular Cell Biology* **17,** 771–782 (2016).

94. Hnisz, D. *et al.* Activation of proto-oncogenes by disruption of chromosome neighborhoods. *Science* **351,** 1454–1458 (2016).

95. Franke, M. *et al.* Formation of new chromatin domains determines pathogenicity of genomic duplications. *Nature* **538,** 265–269 (2016).

96. Dekker, J. Capturing Chromosome Conformation. *Science* **295,** 1306–1311 (2002).

97. Buckle, A., Brackley, C. A., Boyle, S., Marenduzzo, D. & Gilbert, N. Polymer Simulations of Heteromorphic Chromatin Predict the 3D Folding of Complex Genomic Loci. *Molecular Cell* **72,** 786–797.e11 (2018).

98. Bianco, S. *et al.* Polymer physics predicts the effects of structural variants on chromatin architecture. *Nature Genetics* **50,** 662–667 (2018).

99. Todd, S. *et al.* CSynth: an interactive modelling and visualization tool for 3D chromatin structure. *Bioinformatics* (2020).

100. Schreiber, J., Libbrecht, M., Bilmes, J. & Noble, W. S. Nucleotide sequence and DNaseI sensitivity are predictive of 3D chromatin architecture. *bioRxiv,* 103614 (2017).

101. Li, W., Wong, W. H. & Jiang, R. DeepTACT: predicting 3D chromatin contacts via bootstrapping deep learning. *Nucleic Acids Research* **47,** e60–e60 (2019).

102. Whalen, S., Truty, R. M. & Pollard, K. S. Enhancer–promoter interactions are encoded by complex genomic signatures on looping chromatin. *Nature Genetics* **48,** 488–496 (2016).

103. Zhang, S., Chasman, D., Knaack, S. & Roy, S. In silico prediction of high-resolution Hi-C interaction matrices. *Nature Communications* **10,** 5449 (2019).

104. Zhang, Y. *et al.* Enhancing Hi-C data resolution with deep convolutional neural network HiCPlus. *Nature Communications* **9,** 750 (2018).

105. Liu, Q., Lv, H. & Jiang, R. hicGAN infers super resolution Hi-C data with generative adversarial networks. *Bioinformatics* **35,** i99–i107 (2019).

106. Hong, H. *et al.* DeepHic: A generative adversarial network for enhancing Hi-C data resolution. *PLoS Computational Biology* **16,** e1007287 (2020).

107. Schwessinger, R. *et al.* DeepC: predicting 3D genome folding using megabase-scale transfer learning. *Nature Methods* **17,** 1118–1124 (2020).

108. Imakaev, M. *et al.* Iterative correction of Hi-C data reveals hallmarks of chromosome organization. - Supplement. *Nature methods* **9,** 999–1003. arXiv: `NIHMS150003` (2012).

109. Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems* **4,** 3320–3328. arXiv: `1411.1792` (2014).

110. Fudenberg, G., Kelley, D. R. & Pollard, K. S. Predicting 3D genome folding from DNA sequence with Akita. *Nature Methods* **17,** 1111–1117 (2020).

111. Yang, T. *et al.* HiCRep: assessing the reproducibility of Hi-C data using a stratum-adjusted correlation coefficient. *Genome Research* **27,** 1939–1949 (2017).

112. Oudelaar, A. M. *et al.* Dynamics of the 4D genome during in vivo lineage specification and differentiation. *Nature Communications* **11,** 2722 (2020).

113. Briand, N. & Collas, P. Lamina-associated domains: peripheral matters and internal affairs. *Genome Biology* **21,** 85 (2020).

114. Shin, H. *et al.* TopDom: An efficient and deterministic method for identifying topological domains in genomes. *Nucleic Acids Research* **44,** e70 (2015).

115. Nasmyth, K. Disseminating the Genome: Joining, Resolving, and Separating Sister Chromatids During Mitosis and Meiosis. *Annual Review of Genetics* **35,** 673–745 (2001).

116. Alipour, E. & Marko, J. F. Self-organization of domain structures by DNA-loop-extruding enzymes. *Nucleic Acids Research* **40,** 11202–11212 (2012).

117. Nakahashi, H. *et al.* A Genome-wide Map of CTCF Multivalency Redefines the CTCF Code. *Cell Reports* **3,** 1678–1689 (2013).

118. Merkenschlager, M. & Nora, E. P. CTCF and Cohesin in Genome Folding and Transcriptional Gene Regulation. *Annual Review of Genomics and Human Genetics* **17,** 17–43 (2016).

119. Schwarzer, W. *et al.* Two independent modes of chromosome organization are revealed by cohesin removal. *bioRxiv,* 094185 (2016).

120. Hyle, J. *et al.* Acute depletion of CTCF directly affects MYC regulation through loss of enhancer–promoter looping. *Nucleic Acids Research* **47,** 6699–6713 (2019).

121. Lupiáñez, D. G. *et al.* Disruptions of Topological Chromatin Domains Cause Pathogenic Rewiring of Gene-Enhancer Interactions. *Cell* **161,** 1012–1025 (2015).

122. Laugsch, M. *et al.* Disruption of the TFAP2A Regulatory Domain Causes Banchio-Oculo-Facial Syndrome (BOFS) and Illuminates Pathomechanisms for Other Human Neurocristopathies. *SSRN Electronic Journal* (2018).

123. Schmiedel, B. J. *et al.* 17q21 asthma-risk variants switch CTCF binding and regulate IL-2 production by T cells. *Nature Communications* **7,** 13426 (2016).

124. Eraslan, G., Avsec, Ž., Gagneur, J. & Theis, F. J. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics* **20,** 389–403 (2019).

125. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings.* arXiv: 1312.6034 (2014).

126. Dostie, J. *et al.* Chromosome Conformation Capture Carbon Copy (5C): A massively parallel solution for mapping interactions between genomic elements. *Genome Research* **16,** 1299–1309 (2006).

127. Simonis, M. *et al.* Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture-on-chip (4C). *Nature Genetics* **38,** 1348–1354 (2006).

128. Kaiser, V. B. & Semple, C. A. When TADs go bad: chromatin structure and nuclear organisation in human disease. *F1000Research* **6,** 314 (2017).

129. Hanssen, L. L. *et al.* Tissue-specific CTCF-cohesin-mediated chromatin architecture delimits enhancer interactions and function in vivo. *Nature Cell Biology* **19,** 952–961 (2017).

130. Kentepozidou, E. *et al.* Clustered CTCF binding is an evolutionary mechanism to maintain topologically associating domains. *Genome Biology* **21,** 5 (2020).

131. Hu, G. *et al.* Systematic screening of CTCF binding partners identifies that BHLHE40 regulates CTCF genome-wide distribution and long-range chromatin interactions. *Nucleic Acids Research* **48,** 9606–9620 (2020).

132. Baranello, L., Kouzine, F., Sanford, S. & Levens, D. ChIP bias as a function of cross-linking time. *Chromosome Research* **24,** 175–181 (2016).

133. Hay, D. *et al.* Genetic dissection of the α-globin super-enhancer in vivo. *Nature Genetics* **48,** 895–903 (2016).

134. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. *34th International Conference on Machine Learning, ICML 2017* **7,** 5109–5118. arXiv: 1703.01365 (2017).

135. Shrikumar, A., Greenside, P. & Kundaje, A. Learning Important Features Through Propagating Activation Differences. arXiv: 1704.02685 (2017).

136. De Wit, E. *et al.* CTCF Binding Polarity Determines Chromatin Looping. *Molecular cell* **60,** 676–684 (2015).

137. Vietri Rudan, M. *et al.* Comparative Hi-C reveals that CTCF underlies evolution of chromosomal domain architecture. *Cell reports* **10,** 1297–1309 (2015).

138. Guo, Y. *et al.* CRISPR Inversion of CTCF Sites Alters Genome Topology and Enhancer/Promoter Function. *Cell* **162,** 900–910 (2015).

139. Wu, Q., Liu, P. & Wang, L. Many facades of CTCF unified by its coding for three-dimensional genome architecture. *Journal of Genetics and Genomics* **47,** 407–424 (2020).

140. Ulianov, S. V. *et al.* Active chromatin and transcription play a key role in chromosome partitioning into topologically associating domains. *Genome Research* **26,** 70–84 (2016).

141. Héberlé, É. & Bardet, A. F. Sensitivity of transcription factors to DNA methylation. *Essays in Biochemistry* **63** (ed Blewitt, M.) 727–741 (2019).

142. Xia, J., Han, L. & Zhao, Z. Investigating the relationship of DNA methylation with mutation rate and allele frequency in the human genome. *BMC genomics* **13,** S7 (2012).

143. Durek, P. *et al.* Epigenomic Profiling of Human CD4+ T Cells Supports a Linear Differentiation Model and Highlights Molecular Regulators of Memory Development. *Immunity* **45,** 1148–1161 (2016).

144. Schmidt, F. *et al.* Combining transcription factor binding affinities with open-chromatin data for accurate gene expression prediction. *Nucleic Acids Research* **45,** 54–66 (2017).

145. Zhou, J. *et al.* Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature Genetics* **50,** 1171–1179 (2018).

146. Agarwal, V. & Shendure, J. Predicting mRNA Abundance Directly from Genomic Sequence Using Deep Convolutional Neural Networks. *Cell Reports* **31,** 107663 (2020).

147. Schmidt, F., Kern, F. & Schulz, M. H. Integrative prediction of gene expression with chromatin accessibility and conformation data. *Epigenetics & Chromatin* **13,** 4 (2020).

148. Wnuk, K. *et al.* Deep Learning Implicitly Handles Tissue Specific Phenomena to Predict Tumor DNA Accessibility and Immune Activity. *iScience* **20,** 119–136 (2019).

149. Ding, Y., Li, J. Y., Wang, M., Tu, X. & Gao, G. An exact transformation for CNN kernel enables accurate sequence motif identification and leads to a potentially full probabilistic interpretation of CNN. *bioRxiv* (2017).

150. Selvaraju, R. R. *et al.* Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision* **128,** 336–359. arXiv: `1610.02391` (2020).

151. Alaa, A. M. & van der Schaar, M. Demystifying black-box models with symbolic metamodels. *Advances in Neural Information Processing Systems* **32,** 11304–11314 (2019).

152. De Boer, C. G. *et al.* Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nature Biotechnology* **38,** 56–65 (2020).

153. Stamatoyannopoulos, J. A. *et al.* An encyclopedia of mouse DNA elements (Mouse ENCODE). *Genome Biology* **13,** 418 (2012).

154. Telenius, J., Consortium, T. W. & Hughes, J. R. NGseqBasic - a single-command UNIX tool for ATAC-seq, DNaseI-seq, Cut-and-Run, and ChIP-seq data mapping, high-resolution visualisation, and quality control. *bioRxiv,* 393413 (2018).

155. Zhang, Y. *et al.* Model-based Analysis of ChIP-Seq (MACS). *Genome Biology* **9,** R137 (2008).

156. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. arXiv: 1412.6980 (2014).

157. Ramírez, F., Dündar, F., Diehl, S., Grüning, B. A. & Manke, T. deepTools: a flexible platform for exploring deep-sequencing data. *Nucleic Acids Research* **42,** W187–W191 (2014).

158. Servant, N. *et al.* HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. *Genome Biology* **16,** 259 (2015).

159. Telenius, J. M. *et al.* CaptureCompendium: a comprehensive toolkit for 3C analysis. *bioRxiv,* 952572 (2020).

160. Bengtsson, H. & Shin, H. *TopDom: An Efficient and Deterministic Method for Identifying Topological Domains in Genomes.* 2018.

161. Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26,** 841–842 (2010).

162. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).

163. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology* **10,** R25 (2009).

164. Lawrence, M. *et al.* Software for Computing and Annotating Genomic Ranges. *PLoS Computational Biology* **9,** e1003118 (2013).

165. Wickham, H. *ggplot2: Elegant Graphics for Data Analysis.* (Springer New York, 2009).

166. Lawrence, M., Gentleman, R. & Carey, V. rtracklayer: an R package for interfacing with genome browsers. *Bioinformatics* **25,** 1841–1842 (2009).

167. Zeileis, A. & Grothendieck, G. zoo: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software* **14,** 1–27 (2005).

168. Van Der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering* **13,** 22–30 (2011).