DNA Sequence Driven Machine Learning for Modelling Replication Timing

James Ashford

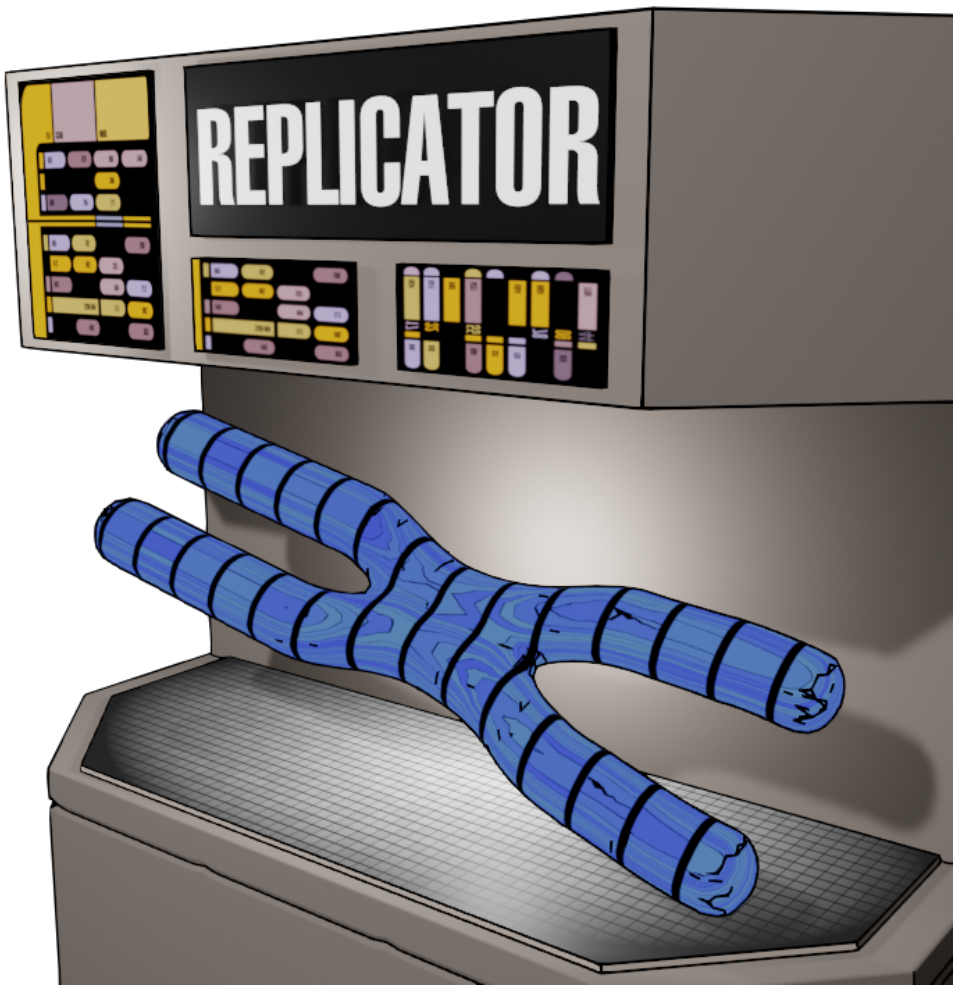
Brasenose College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2023



Live long, and prosper 🖖

Acknowledgements

The last 4 years have been an intense climb, and I owe a lot to the many people who've kept me going. First, for the dedication of their time and expertise in both medical sciences and surviving academia, I would like to thank my supervisors Aleksandr Sahakyan and Marella DeBruijn. Their unwavering certainty that I would make it to the top of this cliff-face has been a guiding beacon. I'd also like to thank the MRC for selecting me for the WIMM Prize Studentship that has made my studies in Oxford possible.

Secondly, I would like to express my gratitude to the colleagues and new friends I've met in my time at Oxford. I'm so glad I got to spend time with my Sahakyan group-mates, especially Patrick and Liezel who I can always count on being in the WIMM for a cup of coffee. Nicole and Tani, you were incredible housemates who brought so many of our cohort together for parties and catch ups, and I look back on our time in Jeune St. fondly. Outside the WIMM, I feel very fortunate to have lived and met people in Brasenose College, especially during my time on the HCR Committee. I'm especially thankful for meeting Ewan, and I hope we will always lose track of time as we walk around Christchurch Meadows sharing our geekiest ideas.

I know that the last few years would not have been manageable without help from old friends, in person and through a screen. I'd like to thank the always Eager Beans for the laughs and gossip, and in particular Edmond, Angus, Tom, and Catherine for our regular gaming sessions. I'd also like to thank Bill for his sage DPhil advice, pragmatism, and taste in coffee shops.

If love and support became words on the page, my loved ones have written this thesis for me 1000 times over. I'd like to thank my partner Ayesha for her support and understanding during the rollercoaster of emotions that were channeled into finishing the DPhil, and for helping me see it through to the end. My family have been a source of constant support and reassurance, and knowing that I can always turn to you all for advice and comfort makes me one of the luckiest people in the world. I love you all so much, and I hope one day I can help lift you all as high as you have me.

James Ashford
Brasenose College, Oxford
January 9, 2024

Abstract

All human somatic cells copy their entire genome during mitotic replication, in the S-phase of the cell cycle. Replication timing (RT) is the temporal order of genome replication in S-phase and has been shown to have consistent global "profiles" across a wide range of tissues and diseases. We demonstrate that while there are many factors that influence the specific RT characteristics of individual cell types, there is a strong link between the DNA sequence composition and the overall RT behaviour. This is achieved by accurately modelling the aggregate profiles from 131 RT experiments constituting 56 unique human cell types, using only engineered features of the DNA sequences as input. We then derive insight into how the composition of DNA sequences impacts RT values, by observing the impact of *in silico* sequence modifications on model predictions. We further extend our modelling towards cell-type specific predictions with a single model by incorporating a minimal source of extra information, ATAC-seq, which provides context for chromatin organisation. The obtained machine learning models, along with the underlying exploratory data analyses and feature engineering, are both useful for prediction of RT and shed light on the underlying DNA sequence basis of the replication phenomenon.

- The behaviour of replication timing (RT) across cell types is analysed by utilising a large cohort of biological samples.
- A model of the core, cell type invariant RT behaviour is developed from engineered DNA sequence features.
- Deep Neural Network models are created by embedding the feature engineering process into the model training.
- The DNA-based models are extended to produce cell-type specific predictions by using minimal extra information (ATAC-seq) to reflect cell-type specific RT modulations.

Contents

List of Figures

List of Abbreviations

1D, 2D . . . . One-, two- dimensional

SD Standard deviation

DNA . . . . . **D**eoxyribo**n**ucleic **a**cid

RNA . . . . . **R**ibo**n**ucleic **a**cid

ssDNA, dsDNA A **s**ingle, or **d**ouble **s**tranded **DNA**

RT . . . . . Replication timing

ASRT . . . . . Asynchronous replication timing

SINE . . . . . Short interspersed nuclear element

LINE . . . . . Long interspersed nuclear element

TF . . . . . Transcription factor

TSS . . . . . Transcription start site

ATAC . . . . . Assay for transpositionally active chromatin

ChIP . . . . . Chromatin immunoprecipitation

bp . . . . . Base pair

Kbp . . . . . Thousand base pair

Mbp . . . . . Million base pairs

TAD . . . . . Topologically associated domain

PWM . . . . . Position weight matrix

PFM . . . . . Position frequency matrix

CpG . . . . . C nucleotide followed by a G nucleotide (5'-3' direction)

TDP . . . . . Timing decision point

SNS . . . . . Short nascent strand

ORC . . . . . Origin recognition complex

ORI . . . . . Origin of replication initiation

OGRE . . . . . Origin G-rich repeated element

MCM . . . . . Minichromosome maintenance complex

IZ . . . . . Initiation zones

CMG . . . . . Cdc45-MCM-GINS complex

RD . . . . . Replication domain

CTR . . . . . Constant timing region

TTR . . . . . Timing transition region

FACS . . . . . Fluorescence activated cell sorting

vlncRNA . . . Very long non-coding RNA

LAD . . . . . Lamina associated domain

RAD . . . . . RIF1 associated domain

G4 . . . . . G-quadruplexes

TDP . . . . . Timing decision point

RIF1 . . . . . Rep1-interacting factor 1

rtQTL . . . . . Replication timing quantitative trait loci

iPSC . . . . . Induced pluripotent stem cell

RPE . . . . . Retinal pigmented epithelium

RC . . . . . Reverse-complement

PCA . . . . . Principal-component analysis

DNN . . . . . Deep neural network

CNN . . . . . Convolutional neural network

ISM . . . . . In-silico saturation mutagenesis

ML . . . . . Machine learning

DL . . . . . Deep learning

MLP . . . . . Multi-layer perceptron

OCR . . . . . Open chromatin region

SOTA . . . . . State of the art

XGBoost . . . eXtreme Gradient Boosting machine

LightGBM . . Light Gradient Boosting Machine

MSE . . . . . Mean squared error

RMSE . . . . . Root mean squared error

MAE . . . . . Mean absolute error

AUC . . . . . Areas under curve

ROC . . . . . Receiver operating characteristic

PR . . . . . Precision-recall

1

Genome Replication Literature

Understanding the central dogma of molecular biology, the storage and transmission of biological information, is a crucial stepping stone to understanding the workings of all biological systems[1]. Transmission of information is only possible where there are mechanisms for replication, and these underpin the ability of the information to propagate and adapt to best survive. Understanding replication within our genome, in a way that is primarily driven by the DNA sequence, encoded information, is likely to give us useful and generalisable insights.

While the goal of replication is unchanged at different scales of life, there exists a great deal of variety in the mechanisms and processes that are used to undertake it. In this work, we explore the sequence basis of DNA replication in modern human cells, a Metazoan organism with Eukaryotic cells and a canonically diploid genome[2,3]. Modern eukaryotes, as opposed to prokaryotes such as Bacteria and Archaea, are characterised by a strongly enclosed nucleus and complex endomembrane systems[4,5]. The formation of the modern eukaryote lineage from a symbiosis between Archea-Asgard host cells and a *proteobacteria* that is now our mitochondria, allowed for an explosion of cell diversity and laid the foundations for divergences in our replication dynamics.[6]

1.1 Composition of the Human Genome

Eukaryotic cell genomes are collections of DNA (deoxyribonucleic acid) molecules densely packed inside the cell's nucleus and mitochondria. Along the length of the DNA molecule are units composed of a deoxyribose sugar, a phosphate group, and a nucleotide "base". Each nucleotide in DNA is one of Adenine (A), Thymine (T), Cytosine (C), and Guanine (C). A and G are classified as *purines* with their two fused heterocyclin rings, and C and T are *pyrimidines* with only one ring[7]. When stacked, these units form a single strand of DNA which can attach to a complementary strand through hydrogen bonds, where A/T and G/C are "complements" to one another. The resulting structure is referred to as double stranded DNA (*dsDNA*), and is the primary information storage mechanism in the genome. This molecule preferentially coils into a structure called *B-DNA*, but there are other possible arrangements and spacings of the two strands including *A- and Z-DNA*[8–11]. More complex local deviations are possible, driven by the DNA composition and environment, such as *G-quadruplexes*, and *i-motifs*[9,12–14].

As with all structures on the single-digit nanometre scale, the shape and arrangement of the DNA sequence is driven fundamentally by DNA-DNA and DNA-protein interactions that can be modelled quantum-mechanically at high computational cost[15]. Long contiguous collections of these DNA sequences fold and arrange themselves, and within the nucleus are bound into larger structures[16]. The first layer of these structures are *nucleosomes*, formed by approximately 150bps DNA wrapping around a set of *histone* proteins[17]. The nucleosomes are then further folded into a structure called *chromatin* fibres of around 5-24nm in diameter, which can be dynamically unrolled to allow access for processes that require direct DNA binding.[18]. Chromatin undergoes a range of self-interacting folding processes, including the formation of *chromatin loops* which can join genomic ranges between single kilobases (kb) and multiple megabases (Mb) apart to fundamentally alter the behaviour of genomic processes such as gene-expression[19,20]. While many direct interactions between chromatin strands are considered transient, there are

regions of the genome that exhibit consistent proximity, forming chromatin regions called *topologically associated domains* (TAD) with relative density modulated to change gene expression[21,22]. Spatially adjacent collections of these TADs at around 10Mb range have been found to be broadly separable by their relative chromatin density into open and more generally expression-active *euchromatin* and closed *heterochromatin*, referred to as A/B compartments[18,23,24].

Finally, the chromatin bundles again into *chromatids*, which are joined by a *centromere* to form *chromosomes*[23]. The diploid nature of human cells arises when each cell keeps two copies of it's *autosomes* (somatic chromosomes), and some combination of the X and Y *allosomes* (sex chromosomes), typically XX or XY. These two copies can contain copies of the same DNA sequences, with minor variations, which are referred to as *alleles*, and this difference can result in modified expression or protein product (for protein coding genes), which is the cause of many genetic disorders[25]. Due to a variety of developmental and replicative factors it is possible for human cells to exhibit *aneuploidy* where the diploid characteristic of the genome does not hold which shifts cell dynamics[26,27]. However, for the sequence analysis that is carried out in this work, a discussion of the impact of sex chromosomes and aneuploidy on modelling would be out-of-scope.

The 3D structure of human genome in-situ can now be analysed at a single-cell level from a range of "3C sequencing technologies", and analysing how these structures evolve over both space and time has bloomed into the study of "4D nucleome"[20,28,29]. In addition, the local openness of chromatin structures can be profiled by techniques such as the "Assay for Transposase-Accessible Chromatin" (ATAC), which reveals local structure and the potential for DNA binding events[30–32]. These and other high resolution techniques are applied to more cells, and approaches that fuse these structural annotations of the original sequence behaviour are being developed, which reveal insights that generalise across the individual organism and species level[33–35].

1.2 Information in the Human Genome

The information in the genome is encoded in the sequence of DNA nucleotides alongside a series of modifications that are made to the bases, referred to as *epigenetic* coding. This information in the genome has been under rigorous selection for millions of years, through processes driven by the intrinsic properties of DNA and the pressures of Darwinian evolution. As such, it can be said that the genome of a modern human cell encodes a highly non-linear and self-modifying set of programs to modulate cell behaviour in pursuit of homoeostasis and the collective survival of the larger organism[36–39]. The nuclear and mitochondrial DNA sets are separate, with different hereditary pathways and mechanisms, and as such are often considered separately[40]. Due to these differences and relative lack of mitochondrial replication data, I will discuss the behaviours of DNA from the cell nuclei.

It is standard in modern genomics to refer to a species-specific reference genome as the "standard" for analysis, designed to capture the overall structure and some heterogeneity, and establish a baseline for comparison[41]. As the quantity of sequenced genomes increases, our understanding of global population heterogeneity expands, and the accuracy of performing genomic analysis with a single "patched" assembly as the baseline comes into question[42]. This has lead to ongoing work and the creation of a "pangenome" reference, that effectively combines information from a large number of individuals into a more dynamic and representative reference[43–45]. Creating and analysing these genomes by necessity requires processing larger ranges and quantity of information, and an ecosystem of powerful tools is growing to meet the demand[46,47]. These tools and genomes represent a paradigm shift in the way we consider diversity and in population genomics, and will provide powerful insights into human heterogeneity. However, for the scope of this work where we are primarily investigating bulk, aggregated behaviours, the traditional reference will be sufficient to provide modelling context. As of this work, the most recent reference genome is "GRCh38.p14" and is used throughout as the reference genome[41].

1. Genome Replication Literature

Previous research has revealed many of the patterns that have influenced the overall composition of the genome, and the encoding behaviours that we believe drive the genomes ability to create all the protein (and other) products that it needs to survive. The most basic of these compositional patterns is the relative abundance of each of the nucleotide bases, which has been found to have consistent relationships described in "Chargaffs Rules"[48–50]. The first states that the A/T ratio and C/G ratio are both 1 in dsDNA, and are a natural consequence of each ssDNA strand being matched to it's complement (A-T, C-G). The second states that the number of complementary bases are approximately equal in ssDNA, for which there is not yet a proven driving force but simulation-driven theory suggest is strongly driven by nucleotide mutation rates[51]. Beyond individual base-pair relationships, there is evidence to suggest that the relative abundance of G/C nucleotides compared to A/T regions contributes to large scale genome behaviours[52]. Computational methods have been created to measure the relative abundance of GC across entire genomes, and group areas with similar abundances into *"isochore"* regions[53]. All genomes are in a constant state of modification, with mutations occurring due to damage or insertion by exogenous factors, some of which are the results of transposable elements[54,55].

Perhaps the most staggering result of modern genomics is uncovering the relative compactness of the genomic code to describes individual organisms. While the amount of DNA stored in each cell is impressive, it is a pittance of information compared to the complexity of the resulting organism[56]. Modern genomics embraces the concept of coding and non-coding segments of the DNA, the former providing the blueprint for proteins, and the latter of which provide a key role is spacing or modulating the expression of the gene product[57]. Sections of the genome that contain the coded information for proteins/RNA products are referred to as *genes*. Previous research has determined that within the gene sequence there are a set of different subsections, which demarcate the edges (5' and 3' untranslated regions (*UTRs*), contain the coding sequence of the product (*exons*), or space out the coding sequences (*introns*)[58].

To convert between the stored information in DNA and a cellular product, be that RNA or a final protein, the gene sequence must be read (transcribed) by a protein complex called RNA Polymerase II (RNAPolII)[59]. These coding sequences embed information about the resulting cellular product in consecutive subsequences of the coding regions of length 3, called *codons*. The rate at which this process is carried out is governed by the environment of the gene sequences, that can be made more accessible (and thus copied more) by the relative shape of local DNA and the abundance of necessary factors. Shaping the expression is often carried out by the relative activation of promoter, enhancer and silencer regions in the genome caused by specialised proteins attaching to them[60]. The *promoters* are regions of the genome immediately adjacent to the gene which can be bound by transcription factor proteins (TFs) to begin transcription by RNAPol[61]. *Enhancers* are usually more distant to the gene, but still contained within the 3D space adjacent to the gene, can be moved by TF binding. Interactions between multiple enhancers that correspond to the same gene region are non-uniform and often non-linear, which make them a fascinating area of active study. *Silencers* are binding sites that can occur within promoters or at more distal positions, and TF activity on them results in down-regulation of the corresponding gene[62].

The expression-modifying protein interactions from TFs highlight the importance of understanding DNA sequence interactions, as most TFs bind to known DNA sequence motifs most of which are between 7 and 20 base pairs (bp) long. These motifs are not always fixed sequences, but due to inconsistency in the binding locations can be expressed as a set of frequencies or weights for each sequential position in the motif - resulting in position frequency and weight matrices (PFM and PWM)[63]. Isolating the binding motifs of different TFs is often carried out using experiments called ChIP-seq (Chromatin Immunoprecipitation sequencing) that can produce profiles of the binding of particular TFs to the genome. While identifying short sequences that bind to TFs highlights the importance of understanding short-DNA sequence behaviour, it has been found that palindromic DNA sequences (a sequence followed by its own reverse complement, thus identical on the forward

and reverse strand) can affect mutational stability depending on their length and have influence over gene expression and replication[64].

There are additional layers of regulation beyond the patterns of DNA around that modify gene expression (cis-sequence code), that are governed by the influence of additional molecules binding to the DNA, RNA, and related proteins. While a great deal of mechanisms have been identified for epigenetic modification, the most common alterations involve adding or removing some acetyl- or methyl- groups from the nucleobase of the DNA or the surface of histone proteins that make up the nucleosome[65]. These changes are often persistent through genome replication, as the modified DNA bases and histones can be kept by on of the copies after replication. For the purposes of this work, the most directly relevant classes of modification are those that directly methylate the DNA cytosines, as these modifications usually target CpG sites in the genome. As such, when there are regions of high density of CpGs, called "CpG Islands" in promoter regions, it is possible that they are being kept there by a selective pressure. Research has found that methylation of specific CpGs in promoters can modify gene expression by influencing TFs, and alter splicing variants by interaction with the transcription machinery[66]. Secondly, epigenetic modification to the histone proteins H3 and H4 have been found to influence local 3D chromatin structure, which has subsequent effects for gene expression[67].

Regulation of gene products in this way fundamentally determines "cell fate" - the type of cell that an un-differentiated cell becomes when specialising. Control of cell fate is determined by many intrinsic and external molecular factors, and interactions between different stages of information transmission play a vital role in this[68]. Determining which factors can reliably control the development of cells into useful tissues, and away from cancerous pathways is an area of fundamental cell research and holds great promise for an incredible number of disease cures. Due to the complex emergent interactions between gene products and the DNA that code them, it is difficult to reliably link the sequences of the gene to its behaviour and ultimate effect on the organism. That said, modelling the relationship between the genotype of a gene and its resulting phenotypical behaviour has provided a powerful

framework for analysing variant data and identifying oncogenes[69]. Additional large-scale efforts to model entire cells as complex maps between genotypes/phenotypes have predicted behaviour across an organism's full life cycle, such as this study of the human pathogen *Mycoplasma genitalium*[70].

1.3 Replicating the Human Genome

1.3.1 Intro: The Cell Cycle

Throughout their functional life, the vast majority of human cells will undergo replication by mitosis to proliferate or replace themselves in the cell population. Mitotic cell division involves the cell creating a full copy of its genome, and undergoing significant growth such that it can split to create two viable cells with all the genetic code and organelles for sustained functioning. For eukaryotes, the processes that govern self-replication are conventionally described in a cyclical set of state transitions referred to as the "Cell Cycle"[71]. The outlined processes highlight key activators of cell-state transitions, determinants of the processes in full-genome replication, and genome-wide changes that influence the rate of cell replication such as 3D structure[72]. Single-cell studies in the last decade have revealed that while this structure provides a useful overall framework, there are many ways for cells to deviate from these neat stages, which are particularly interesting to study in diseases such as cancer[73].

The cell cycle has two high-level phases: interphase where the cell grows and creates an entire copy of its genome, and mitosis where the cell splits into two daughter cells each of which contains a full copy of the genome. Mitosis is extensively studied elsewhere, with many intricate internal mechanisms separating the duplicated genome to different sides of the cell then creating two separate cells by cleaving the parent cell down the middle between the two copies of the genome. Each separate cell then reconstitutes its internal structure, forming cell-critical components such as the nuclear membrane, proliferating internal structures such as mitochondria, and unpacking/uncoiling its chromosomes into a more accessible

state to begin gene product manufacture. Interphase can be further broken into a sequence of growth (G) and synthesis (S) phases. When a cell is senescent and not preparing to replicate, it is described as being in "arrested phase" (G_0) where it undergoes native metabolic process and any specialised functions determined by its cell fate. When the cell enters its first "growth phase" (G_1), the cell begins to accumulate material for component synthesis, and passes through a chemical checkpoint system that acts as a precursor to genome synthesis, driven by proteins called cyclin-dependent kinases (Cdk)[71]. Once the cell has grown and accumulated sufficient material and energy for replication, the cell will enter S phase, after the production of cyclin reaches a critical level which triggers the Cdks. At this point, referred to as the "restriction" point, the cell is now committed to replication by mitosis and a copy of the full genome is produced, with the process only aborted if mitogen signalling is lost[74]. Once the internal assembly of the duplicate genome is complete the cell enters another growth phase (G_2) which promotes further resource accumulation in preparation for Mitosis (M). When mitosis is complete, the cells either enters G_0 to stop replicating, or G_1 to begin the replication process again. The time taken for human cells to proceed through the whole cell cycle varies significantly with age and cell type, ranging from 2 to 60 hours[75].

1.3.2 Replication Machinery and Dynamics

The preparations for replication begin in the G_1 phases when many sites along the genome are marked as potential origins of replication initiation (ORI) by binding with an origin recognition complex (ORC). This ORC is a large protein complex that binds to DNA at origin sites, along with the protein Cdc6 which attaches to the ORC through a Cy motif, forming the "pre-replication complex"[76]. Human ORCs have been found to show clear evolutionary differences to ORCs in other organisms, lacking the Orc4 α helix and the Orc2 loop that drives sequence specific binding in non-eukaryotes[77,78]. In prokaryotes and some budding yeasts, there is a fixed motif for the binding of ORCs to the DNA[77]. In these budding yeast cells the origin has a compositional motifs structure with an primary sequence,

a 17bp AT rich "autonomously replicated sequence" (ARS) referred to as "A", followed by one of a possible subset of sequences referred to as "B"[79]. In human cells there is not an analogous motif, but the distribution of these sites is driven by a relationship with DNA sequence factors and chromatin spatial organisation[80]. To ensure that sections of the genome are not replicated multiple times by new pre-replication complexes forming, the protein Geminin inhibits the complex assembly by binding Cdt1 during S phase[81].

Origin sites are non-uniformly distributed but can be profiled by a range of methods, reviewed in Hu et. al.[78], revealing that at least 10% of the human genome has the potential to act as an ORI. However, these predicted candidate ORIs do not overlap perfectly with known ORC binding locations, suggesting that there is still a gap in our understanding of the processes involved[82,83]. The usage of ORIs across different cells is not consistent, leading to researchers dividing origins into "core" origins that are always/frequently used and "stochastic" origins, which are used rarely or in very few cells types[84]. Computational analysis and modelling of experimentally marked ORIs revealed that human origins are at least 700bp in size, and strongly associate with having regions of high "G" density upstream of their location[85]. These regions are referred to as "origin G-rich repeated elements" (OGREs) and a likely to form a large number of G-quadruplexes and coincide with regions that have low nucleosome density, which is driven by nucleosome modification H3K64ac[86]. There is also earlier evidence to suggests that origins in *Saccharomyces cervisiae* show strong G-C skew[87]. While these markers of origins are promising, they are only able to identify origins with at most 80% accuracy. There is also evidence to suggest that active transcription in the genome can influence ORC's ability to bind with downstream proteins, with the transcription start sites (TSSs) of actively transcribed genes being enriched with ORCs.[88]

Once an ORC has been loaded onto the genome, it is subsequently "licensed" by the loading of two minichromosome maintenance (MCM) complexes (MCM2-7 helicase proteins) onto it with the assistance of Cdt1[78]. This combined ORC-MCM construct is referred to as the "pre-initiation complex"[78]. The licensing process is

mostly limited to G_1 phase with attempted licensing in S phase limited by Cdks, however, recent evidence of full replication in regions that have high transcriptional activity during replication (which detaches previously placed MCM complexes) suggests that areas can be re-licensed during S-phase to reduce replication stress[78,89]. HBO1, an H4-specific histone acetylase, is a co-activator of the DNA replication licensing factor Cdt1 and thus plays a central role in controlling which ORCs are licenced during the replication process[81]. MCM2-7 redistribution during the origin licensing process may represent an additional mechanism to establish a cell type specific DNA replication program, which will be explored in a subsequent chapter[83].

The resulting ORC-MCM complex is activated during the S-phase to form an CMG (Cdc45-MCM-GINS) complex[90], onto which a payload of DNA polymerases is loaded that will carry out the replication process, forming the "replisome"[91,92]. Each cell uses only around 20% of available origins, found by comparing ratio of origins found in bulk to single-molecule studies, and the activation of replication is governed by interactions between environmental proteins and the MCM-complexes[93]. Cdk proteins play a crucial dual role in this regulation, by inhibiting MCM loading in the G_1 phase by phosphorylating the ORC, then switching to activating the MCM complexes attached to ORCs by phosphorylating Sld2 and Sld3 in S phase[94]. Sld3 is known to bind to early-replicating origins in yeast during G_1 phase[95]. TICRR/TRESLIN are homologs of the Sld3/Sld7 proteins, and phosphorylated versions of them are available but inhibited in S phase to promote replication initiation, resulting in them affecting the number of origins that initiate[95]. MTBP also binds to replication initiation sites, and is proposed to form a complex with TRESLIN, but is not inhibited when the cell transitions into S phase, where depletion is known to inhibit origin firing[96]. Origins in close spatial proximity, that lie within 1Mb of each other and aligning with chromatin TADs, preferentially co-activate, as revealed by 3D-SIM observations of individual replication events[97]. These contiguous areas of co-activation are referred to as broad initiation zones (IZs), and have been profiled by investigating the directionality of replication machinery across the genome[98]. While a sequence-basis of activation order is not

fully characterised, there is evidence to suggest that replication origins that are highly initiated are flanked by fragile poly(A/T) sites that are nucleosome-free, or are close to histones with H4 acetylation[99,100].

At each replisome a structure called a "replication fork" forms, which move bidirectionally away from their origin[92]. Replication forks do not move at a continuous rate throughout S-phase, and are generally slower in early replication at gene rich regions where there is high replication-stress due to transcription interactions[101]. The rate of replication fork movement in single live cells at specific loci can be visualised with fluorescence microscopy, which has revealed a strong dependence between fork rate and the maturation of the Okazaki fragments[102]. The space of replicated DNA between a pair of active forks is referred to as a "replicon", and has also been historically referred to as a replication "eye" or "bubble"[103,104]. At the forward edges of each fork structure is a topoisomerase enzyme, "DNA gyrase", which moves ahead of the DNA uncoiling caused by the helicase to introduce temporary breaks in the DNA to release torsional tension[105]. In the replication fork, the process of "semi-conservative" replication is carried out, which duplicates the original DNA strand, and includes one of the original strands in each of the new dsDNAs[90]. Ensuring that one strand of the original DNA is included in each is thought to be powerful as it allows for direct error-checking. As DNA polymerase is only able to replicate in the 5' to 3' direction, the two resulting strands are replicated with different mechanisms. The "leading" strand has the relatively simple mechanism of a single DNA polymerase (Polϵ) receiving RNA primers from the DNA Polα-primase continuously and appending them directly onto the new strand it is constructing, producing a complete double helix in its wake[91]. The "lagging" strand has to undergo an additional step to account for DNA polymerase's one-directional copying. The DNA primase produces RNA template strands that are read by a DNA polymerase (Polα) and combined into 100-200bp long "Okazaki" fragments by Polδ[106]. These fragments then replace the RNA primers and are appended to the lagging strand[91]. Both Polβ and Polϵ are capable of DNA proofreading in their A subunit, whereas Polα cannot[107]. To ensure that duplication does not occur,

replication forks that collide will detach, allowing DNA repair mechanisms to join the strands. If a fork reaches the end of a chromosomes it can also undergo collapse near the end, which results in some of the remaining bases not being copied. To accommodate for this, every chromosome has a length of bases at the end called "telomeres" that act as an ablative armour against damage by DNA replication[108].

Throughout the *S* phase, replication *stress* is caused by processes that delay genome replication by inhibiting, stalling or mis-regulating the activation of origins and the progress of replication forks. The forks are maintained by many mechanisms, including the binding of the C-terminal domain of the RIF1 protein[109]. However, their initiation can be inhibited by Captothecin, or they can be stalled when moving though unprotected long (>20bp) poly(A/T) tracts or colliding with other active replication forks[72,99,110]. Fortunately, the large number of non-activated ("dormant") replication origins allow for redundancy in the replication process and can accommodate for replication stress and ensure genomic stability[111]. The length of S-phase is driven by arrangement and activation of ORCs, but can be modified through external chemical modifications[75]. For example, knocking down ENSA extends the S-phase by inhibiting transition from *S* phase into mitosis, and adding Trichostatin-A increase the initiation rate of replication, which speeds up S-phase[100,112].

1.4 Summary

- Genome is made of simple biomechanical parts, with very complex interaction pathways.
- Genome behaviour is heavily influenced by sequence composition, epigenetic modifications, and 3D structure.
- Complete, single genome copy during replication is required for cell viability.
- Rate of replication is governed by a wide-range of molecular factors.

- The pattern of origin activation is not fixed and not uniformly random, with clustered origin firing driving a characteristic replication order. Why might this be?

2

Replication Timing Literature

We have established that human genome replication is carried out from a redundantly large number of licensed origins locations across the genome[93]. The origin firing displays characteristic behaviour, with spatially local origins firing in concert and subsets of clustered origins preferentially firing earlier than others on average, Figure 2.1 A. This leads to regions of the genome being replicated earlier or later in S-phase. This temporal ordering of genome replication is referred to as replication timing (RT). RT has a knock-on impact on the behaviour of other genome features, such as epigenetic marking on histones, and the formation of 3D genome structure[113]. We are interested in establishing the key factors that influence these emergent RT patterns, and leveraging current experimental data to extract a meaningful sequence baseline for RT. In this chapter, I will discuss our current understanding of RT as a measurable emergent phenomena in modern genomes. I would like to highlight the work of Nicholas Rhind, and the ongoing "Replication Timing and Transcriptional Control" correspondence by Gilbert et. al. as excellent reviews of current state of the field[113,114].

2.1 Experimental Methodologies

To best analyse RT in human cells, it is essential to understand how we measure the RT values experimentally. In an ideal scenario, we would measure the individual replication forks and replicon formation across the full genome using absolute timing values from the initiation of the S phase, and examine how these values fluctuate between biological replicates and different cells. This would be the gold standard of data for studying replication, as coupled with recently developed imaging methods that provide nucleotide resolution 3D structures we would have the full spatial and temporal picture of replication. While such data is not yet available, there has been astonishing progress in the optical measurement of replication initiation sites down to single molecule resolution, which has allowed to accurate reconstruction of RT values when used as the driver of a stochastic mechanistic model of DNA replication[115,116]. In the absence of this gold-standard, we instead turn to measurements that we can make that provide a proxy for replication of given sequences in a genome. Over the last 3 decades, multiple methods have been derived to measure or extrapolate relative values of RT from cell samples. These revolve around the idea of measuring and comparing the amount of DNA replicated in cells in different stages of S-phase. By comparing the abundance of these sequences at different time steps, we can approximate the relative time in S-phase that these sequences from the genome were replicated.

Initially, a methodology comparing the results from PCR experiments was used to measure the expression of individual genes in early/late S-phase, which provided data on the relative expression of genes but did not allow for the analysis of genome-wide behaviour[117]. The first work that produced whole-genome RT values was done at 1Mb resolution across segments of the genome by Woodfine et. al.[118], which was then expanded upon by Hiratani et. al. in their 2008 work[119] improving the resolution to every 5.8Kb. In both of these whole genome works, bulk cell populations were cultured and then stimulated through the G_1 phase to begin genome synthesis. Early in the S phase, a subset of the population were frozen

and newly sequenced DNA is labelled by pulsing the cells with BrdU. Fluorescence activated cell sorting (FACS) is then used to separate those from early and late S-phase by comparing the amount of genome duplication that has occurred. Once the early and late populations were separated, the DNA within them is analysed through either hybridisation to a microarray plate or next-generation sequencing (NGS) resulting in the Repli-chip or Repli-seq processing pipeline respectively[120]. In most of the existing RT datasets, a reported RT value is calculated by comparing the reporter value (microarray fluorescences or sequencing depth) for all positions between the two populations. These resulting values are then transformed by taking a base 2 logarithm, to shift all values from a majority early position to be positive, and all values from a majority late sample to be negative. Subsequently the RT profiles are smoothed along the genome by fitting a loess multi-polynomial curve to compensate for experimental noise.

In more recent experiments, the number of cells analysed has allowed for additional fractions within the S-phase to be extracted beyond the original 2 (early/late), resulting in 16-stage Repli-seq profiles that are either reported as direct coverage values, or a weighted sum of all 16 stages where the weight is the stage number increasing from 1-16 for later S phase values[121]. This process allows for qualitative comparison with fewer normalisation steps, and can identify individual IZs and directional TTRs genome-wide[121]. Improved sequencing technologies have also allowed for the generation of telomere-to-telomere (T2T) genome sequencing profiles, which have revealed that centromeric regions generally replicate later in S-phase, and families of satellite DNA have a range of biases for late S phase replication[122]. To investigate the influence of single cell heterogeneity on RT behaviour, sc-Repli-seq was developed. Instead of comparing two or more fractions of multiple cells, a single population in mid-S phase is selected and amplified. The resulting sequenced genomes are then analysed to: identify early domains by finding regions that have a copy number of 2, and calculate a proxy of RT value by measuring the log_2 of read for a region over the genome-wide median of counted reads[123]. In complement to the above experimental methods of directly

measuring RT, other methods focussed on replication-based behaviour have been developed such as Repli-ATAC-seq, which profiles the chromatin composition of DNA after replication has occurred[124]. An alternative method of generating RT profiles for cell samples was discovered by isolating the influence of RT on local copy number of whole-genome sequencing data. Analysing the local fluctuations of sequencing depth after correction for GC content reveal a signal that capitulates the RT behaviours. The resulting method and software package, TIGER, can produce an RT profile for any deep sequencing dataset that has a corresponding reference genome; and produces > 0.8 correlation with experimental Repli-seq profiles in highly proliferating cells[125].

Depending on the dynamic-range of the experimental technique used to extract and calculate RT profiles, the explicit values assigned to a specific genome coordinate can range significantly. For fluorescence-based microarray experimental the log_2 RT values range between ± 2, in bulk Repli-seq experiments with sufficient coverage the values can range ± 6, and sc-Repli-seq values lie between ± 1[123]. It is important to note that the exact magnitude of RT in an individual experiment is not important, and the focus lies on the relative value between different regions of the genome within an experiment, as long as the prior data processing ensures that an RT value of 0 is the tipping points between prevalently early and late replication. To encourage consistency in data processing and allow for efficient pipelining the "Repliscan" software package was produced, that allows for processing different combinations of Repli-seq S phase fractions - ultimately advocating for a data processing pipeline that uses a G_1 phase references to effectively profile the start of cellular replication[126].

2.2 Human RT Behaviours

To describe the general characteristics of RT in the human genome, we must first lay the groundwork for patterns and terminology that are used to describe RT behaviours. When the RT values for contiguous spans of the genome are compared, it becomes immediately clear that there are patterns in adjacent positions in the

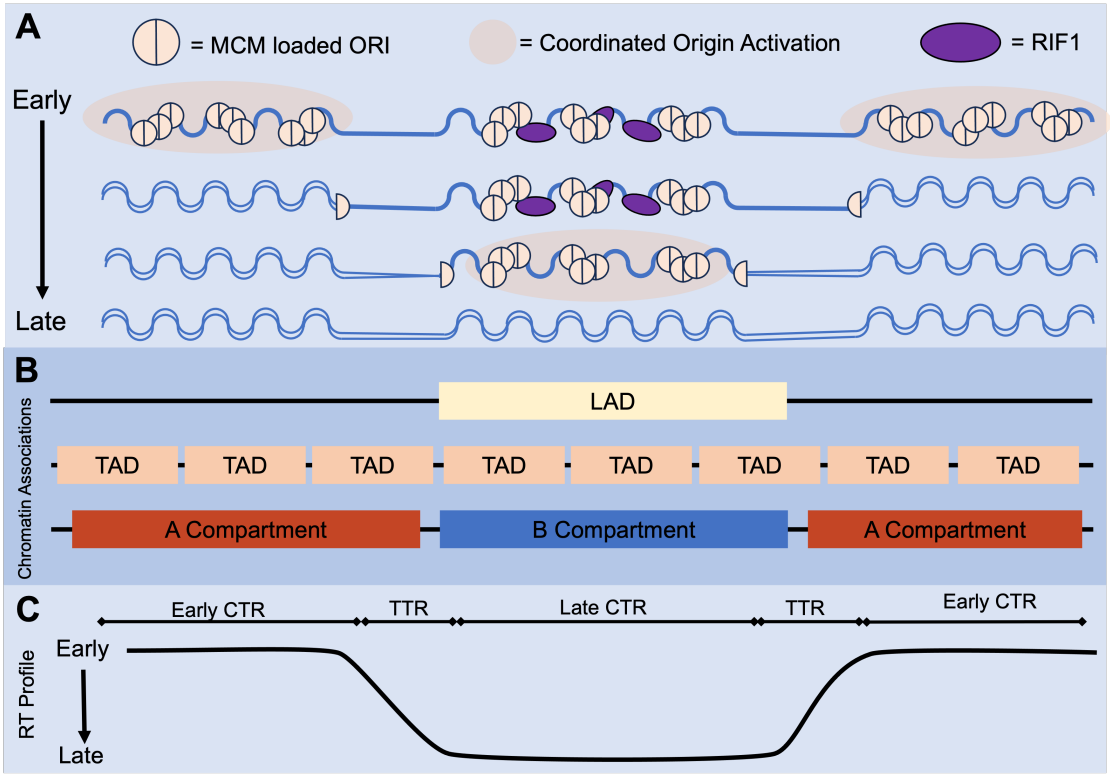


Figure 2.1: RT in the human genome. A) The pseudo-random but coordinated firing of MCM loaded ORI leads to cascades of local replication, with regions that are repressed by RIF1, and will replicate later once the protein is removed. B) RT behaviour is intimately linked to local chromatin structure, with lamina association enriched in B compartments that will replicate later. TADs usually contain regions of similar timing, however A/B compartments are the strongest candidate for structural units of RT. C) Comparing experimental replication data from different stages in S-phase results in an RT profile that reflects the dynamics of human RT, with CTRs corresponding to coordinated origins and TTRs arising when fewer redundant forks are activated to replicate a region.

genome. There are two key patterns that have been identified and validated to occur consistently across all known human cell types. The first are areas of consistent RT value, referred to as constant timing regions (CTRs). These contain multiple *replication domains (RDs)*, the result of coordinated firing of between ~ 1 to 4 adjacent replication origins[127,128]. The cumulative effect of these origins creates sections of the genome between 100kbs and 10Mbs that are copied at similar times, and are strongly associated with chromatin compartments A and B for early and late regions respectively[127]. The second key features lie between the CTRs, with regions where the RT value changes rapidly over a short section of the genome. These are often driven by the actions of very few or one single fork

traversing a segment sparsely occupied by replication origins, which are referred to at *timing transition regions (TTRs)*. Visualising these contiguous spans of RT values is referred to as an RT profile, and these illustrate the dynamics of RT for a given genomic region, Figure 2.1 C[129].

These RT profiles have been found to have strong intra-cell type reproducibility, and profound conservation among different cell types for the same organism with over 50% of the genome's RT unchanged[130,131]. RT has been found to be consistent in xenografts of cryogenically frozen human leukemic tissue into mouse hosts, and conserved during species evolution[130,132]. RT has been demonstrated to contribute to allelic choice among other developmental behaviours, and RT has shown allele-specific behaviour in mouse cells[133,134]. It has been shown that particular cell differentiations or pathologies can have localised modifications in RT behaviour at different positions in the genome in both bulk and single-cell data, and may persist through induced cellular reprogramming with possible implications for stem cell therapies[119,122,135]. RT behaviour is sufficiently differentiation-specific to allow the construction of gene-regulatory networks based on differential RT behaviour during differentiation from ESCs to different lineages[136]. ESCs have been recorded to have RT profiles that are earlier than later differentiated cells, but haploid ESCs display significant RT delays[137,138]. As cells enter senescence, there are signs of increased replication stress, however the RT behaviours remain globally unaffected, despite studies indicating that the increased genome fragility causes by replication stress can induce cell-type specific changes to RT that are heritable in daughter cells[139,140]. RT has been causally linked to the incidence of oncogenic chromosomal translocations in lymphomas, and leukemic lymphoblastoid cells show abnormal RT behaviour, which can be patient specific or shared in all leukemic samples[141,142]. Late replicating regions in prostate cancer display loss of DNA methylation, and are also prone to chromosomal rearrangements[143].

How is this behaviour regulated within the genome? Beyond the positioning of origins which has already been discussed, a crucial start is the timing decision point (TDP), a period of G_1 phase behaviour where the 3D structure of the genome is

organised in preparation for ordered genome synthesis. The aggregation of DNA into chromatin arrangements of different densities occurs in different parts of the nucleus, with closed heterochromatin forming around nuclear lamina and nucleoli, and open euchromatin populating the remainder of the nucleus. If a cell is forced into S phase before the proper establishment of the TDP, a random temporal program of replication is observed[144]. Pre-2000, models of eukaryotic DNA replication suggested that there were between 10-100 adjacent and simultaneous replicons each ~100kb in length[128]. However, with the rise of genome-scale RT databases, an alternative multi-scale hypothesis of genome RT controls has formed[113].

On the whole-genome scale, the protein RIF1 (Rap1-Interacting Factor 1) has been identified as a modulator of RT. RIF1 over expression modifies chromatin binding and inhibits replication in yeast, and depletion in mice results in the differential expression of over 600 genes[145,146]. The human homolog of this yeast gene has a PP1 binding motif, which when bound promotes helicase loading in G_1 and inhibits origin activation by de-phosphorylation in S phase[90,147]. RIF1 additionally assists with fork-reactivation[148]. RIF1 deficiency in mouse and humans dysregulates RT and inhibits post-replication chromatin structure[146,149]. Current models suggest that in wild type cells RIF1 binds preferentially to late domains associated with nucleolus and LADs, removing them from candidacy for activation until late S-phase and creating the RT profile, Figure 2.2 A[150]. Depletion of RIF1 leads to uniform activation of early and late origins, and loss of mid-S replicating foci, causing homogenous RT[113,151,152]. While the exact binding positions of RIF1 are not known, it has been shown that G-quadruplex (G4) like structures provide a strong binding location, implicating non-canonical DNA structures as a regulatory element of RT[151]. There are also suggestions that the proteins Fkh1/2 are intimately involved in the regulation of RT in humans, through regulating binding within chromatin structures[153].

Despite there being a strong local RT behaviour with TADs, a downregulation of cohesin, and thus significant TAD formations, does not disrupt the RT program suggesting that RT is linked more closely to A/B chromatin compartment

boundaries than TADs[24]. Similarly, a complete depletion of CTCF binding in the genome that affects TAD structure had no significant measurable effect on RT or A/B compartment separation beyond experimental variability[154]. The nuclear A compartment is generally more open than B, correlating closely with early and late RT respectively[155]. In addition, direct correlation experiments have shown that Hi-C adjacency frequency maps correlate strongly with RT profiles, achieving strong correlations across the full genome in Ryba et. al. 2010, Figure 2.1 B[130].

At the chromosome level, a collection of "Asynchronous replication and autosomal RNAs" (*ASAR*) genes encode vlncRNA that influence RT, and have been identified across the human genome. These vlncRNA are >200kb long, and disruption of their expression causes faulty transition from the end of the RT program into mitosis[113]. ASAR interacting genes have been identified, and 69 RNA binding proteins (RBPs) interact with them forming complexes that control replication and chromosome integrity[156]. ASARs display asynchronous RT (ASRT) between alleles, and the genes are subject to an epigenetic program that results in random Allelic Expression Imbalance (AEI)[157,158]. The ASAR vlncRNAs cover the chromosomes from which they are transcribed, and if inhibited the cell exhibits delayed replication and chromosome condensation, which causes potentially pathogenic genome rearrangements[159].

At the level of local domains within-chromosomes the composition of the genome, protein interactions driven by higher level RT regulators, and the epigenetic state of the nucleosomes become the most influential factors for determining replication timing.

Cis-sequence modifications that could modulate the RT behaviour in a reproducible way are considered the holy-grail of modern RT research, as it would bring our understanding and control of the phenomena on par with the relatively simple mechanisms in bacterial genomes. It has been well established in literature that RT values positively correlate with sequence GC density, the abundance of G-quadruplex structures, and this is likely causally linked to the positive correlation between RT values and replication origin density[151]. There are strong indications that RT behaviour is related to genome fragility and the presence of retro-transposon

elements in the genome, with single nucleotide variations (SNVs) occurring more frequently in late RT regions and GC-content-independent associations between RT values and SINE (but not LINE) elements present in germline cells[160,161].

A modification to the local sequences in the *Myc* locus causes an early-to-late RT switch that resulted in a loss of translocation[141]. Similarly, deletion of sequences that shift genome compartmentalisation is capable of producing locally perturbed RT[154]. Work carried out in over 400 iPSC cultures has brought to light *cis*-sequence modifications in the human genome that directly effect RT. RT quantitative trait loci (*rtQTLs*) are locations in the genome where natural genetic variability is accompanied by a significant shift in RT behaviour[162]. These sequence modifications can act in isolation, or in tandem with other rtQTLs producing effector patterns that mimic those of enhancers in gene expression. These rtQTLs were validated by experimentally modifying them in sections of the genome and experimentally verifying a change in the RT profile[162]. 1617 rtQTLs were identified across a panel of iPSC and ESC cell lines, and their locations in the genome were consistently enriched for the binding motifs of TFs including: SOX2, POU5F1 (OCT4), NANOG, EP300 (P300), SP1, and RBBP5. The work by Ding et. al. concludes that:

> "Replication timing is robustly encoded in DNA, yet multiple DNA sequences dictate DNA replication combinatorially via chromatin effectors. The replication timing program of the human genome emerges as being sequence-dependent, without being sequence-specific."

suggesting that a sequence code for RT is most likely to be found as a composition of multiple sequence features[162].

When investigating the relationship between RT, 3D genome structure from Hi-C experiments, and epigenetic histone markers, a significant pattern of enriched markers was found in regions that are early replicating, and were thus deemed early replication control elements (ERCEs)[154]. These ERCE regions show compositional effects on RT similar to those in rtQTLs, and can be analogised as "enhancers for early replication". The ERCEs are enriched in histone markers, and are the site of consistent co-binding of OCT4, SOX2, and NANOG (OSN). These

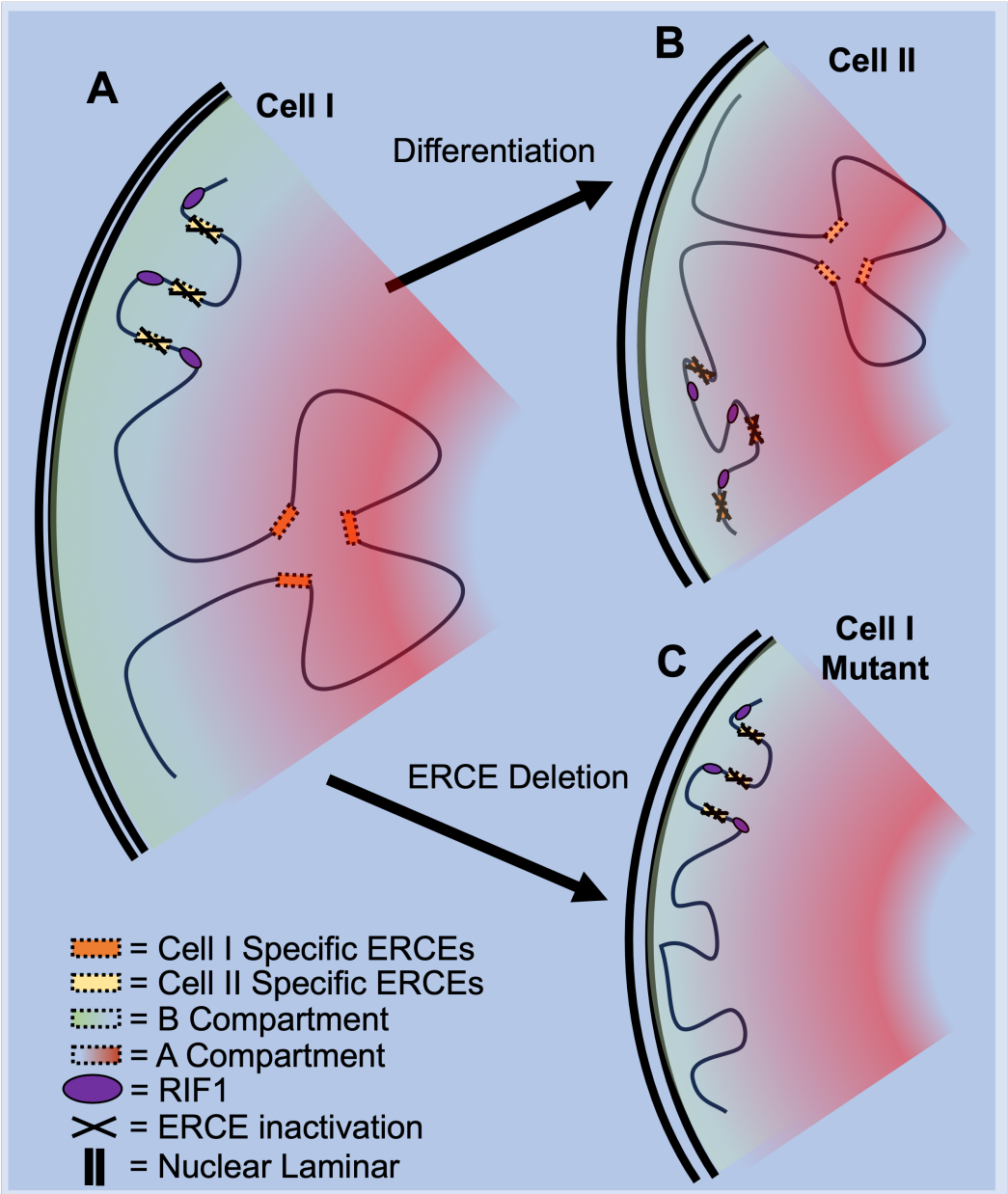


Figure 2.2: ERCEs modulate 3D structure and consequently dictate local RT. A) The selective activation of groups of ERCEs results in interactions that drive cell specific 3D chromatin structure, creating compartments that are replicated earlier or later. B) When the activation pattern is modulated through protein binding and epigenetic events, the 3D structure changes to prioritise other regions for replication by converting them into an A compartment. C) Experiments that selectively remove activate ERCE elements in a cell line repress the chromatin structure regulation resulting in no early replication DNA.

pluripotency factors (OCT4 and NANOG) had been previously shown to have a binding preference for early-replicating alleles in hESCs and iPSCs, along with EP300 and ATF3[162]. Inversions of these ERCE regions showed minimal perturbation to the RT behaviour in the region, suggesting that the presence of these *cis*-sequences is sufficient to control RT. Once validated around the DppA2/4 domain where they were originally identified, a further 1835 ERCEs were predicted across the mouse reference genome. 34% of these ERCEs overlap with mouse enhancers, and are active with many chromatin marks[154]. Crucially, ERCEs showed robust long-range interaction independently of CTCF, suggesting that ERCEs are responsible for 3D genome structure and A/B compartmentalisation, which have previously been highly correlated but never causally linked through sequence[154]. Activation of different ERCE sequences leading to changes in the genome structural arrangement is currently considered one of the strongest cis-sequence mediators of RT, and has been observed though experimental deletion of known ERCE regions Figure 2.2 C, and is hypothesised to occur through the process of cell differentiation, Figure 2.2 B[154]. Additionally, ERCEs may provide a "histone acetylation landing pad" for Brd2/4-Treslin to partially advance RT, as has been seen in previous work[163].

RT has a strongly reported epigenetic component, primarily based on patterns relating to histone and other chromatin modifications correlated with RT values. For example, histone acetylation stimulates earlier replication initiation, and the replication origin licensing factor Cdt1 is co-activated by the histone acetylase HBO1 which has knock-on implications for earlier RT[163,164]. TICRR-BET protein interaction regulates replication by recruiting replication factors, and are modulated by chromatin acetylation[163]. Local replication patterns have been associated with 4 key chromatin states, with the K9/K36me3 demethylase KDM4A controlling 11% of RT genome-wide through broad histone marks especially H3K36me2[165]. Euchromatic marks such as H3K4me1 and H3K18ac were positively correlated with RT, whereas heterochromatic marks H3K27me3 and H3K9me2 were negatively correlated, especially in senescent cells[90,166,167]. H4K20me3 has been implicated in enhancing origin formation in later replication heterochromatin, and restricting

replication licensing to prevent over-replication during S phase[88,168]. Heterochromatin regions near the centromere enriched with H3K27me3 have later RT[169]. Recent work has also indicated the presence of a "histone code" for RT at rtQTL locations, consisting of 3 methylation then hyper- (>2) acetylation mark[162]. There is also evidence to suggest that genome-wide DNA hypomethylation can introduce single-cell RT heterogeneity[170]

2.3 Existing Software

As we have discussed, RT possesses strong persistent behaviour across multiple orders of life, and as such many attempts have been made to effectively model RT *in silico*. The first of these were mechanistic in nature, attempting to recreate high-level RT behaviour in a simulated genome leaning on models of replication origin and fork mechanics. The work of Jun et. al. modelled the spread of replication forks as akin to a 1D nucleation problem, described as a "Kolmogorov, Johnson-Mehl, and Avrami" (KJMA) model[171]. Each origin of growth could represent a replication origin, and could be used to recover experimentally observed nucleation rates and domain sizes when simulating chromosomal replication in an ideal constant replication rate case[103]. Strong performance on RT prediction was then achieved by Gindin et. al. using mechanistic modelling of the activation and progress of replication machinery at a genome-wide scale[116]. Staggeringly, their "Replicon" model has only 1 free parameter in its baseline operation mode, the number of replication forks active simultaneously, but is capable of predicting RT profiles with correlations ~0.84. This differentiates it heavily from another attempt which fitted a model to existing RT data in other organisms by explicitly parametrising the firing probabilities of known origins[172]. For the human genome, this would require many thousands of parameters, and was considered an intractable and likely over-fit solution. Instead, the predictive power of Replicon is derived from it probabilistically initiating replication at different positions along the genome using a genome annotation as the probability density. The strongest performing Replicon model uses the results from a "DNase" footprinting assay as its initiation probability "landscape"

across the genome, which aligns with our understanding of RT being driven heavily by DNA accessibility[173]. Subsequently, Löb et. al. produced a model that similarly aimed to reproduce S phase behaviour, and forewent using a dense genome annotation by assuming that different chromatin states (open euchromatin, closed heterochromatin, or intermediate facultative chromatin) have different efficiencies of origin firing, and that origin firing is selectively inhibited or stimulated based on the relative distance to nearby active forks[104]. This model is parameterised by a set of 14 experimentally derived values, including the number and spontaneous firing probabilities of different chromatin states and estimates of fork replication speeds. Furthermore, by using the ball-on-string polymer model of DNA with some assumptions about chromatin local attraction, this model could qualitatively reproduce imaging results of cells undergoing different stages of S-phase. All of these models were able to faithfully reproduce RT and replication fork dynamics by integrating simplified states, stochastic mechanisms and/or an abundance of instructive datasets. However, despite their impressive reconstruction performance these models provided a limited window into the mechanisms that drive RT.

Using the ever growing body of genome annotations for different cell lines, Van Rechem et. al. collected and discovered a pattern of 3 distinct histone epigenetic marks (H3K9me3, H3K27me3, and H3K36me2) that strongly predict genome-wide RT with high correlation ($R^2 = 0.94$) in RPE cells. The models produced were generated with linear regression, for data in 50kb DNA sequence bins across the whole genome. They identify that genome-wide remodulation of histone marks by overexpressing KDM4A causes a change in RT across 11% of the genome, and their histone-mark predictor can predict the change in RT (δRT). Their model is highly suggestive of a RT influencing factor driven by the epigenomic markers - and it remains to be seen if these patterns of epigenome behaviour are found in other cell types, and are causative of RT. Finally, the most recent RT prediction model is "CONCERT" (CONtext-of-sequenCEs for Replication Timing), currently available as a pre-print on *bioRxiv*, which creates a model for each cell type to predict RT across the entire genome[174]. The model is constructed of two main modules: a "selector"

which highlights regions in the local context that are likely useful for prediction, and a "predictor" which uses features from the selected bins to predict the final RT values. This combination of modules allows the model to be used after training to identify loci in the genome which are highly predictive of local RT, and the work highlights strong overlaps between areas with CTCF motifs and a variety of chromatin marks. In addition, *in silico* deletion of different combinations of selected "high importance" regions showed combinatorial behaviour similar to those of ERCEs[154].

The experimental insights and modelling frameworks that have been presented reveal RT to be strongly driven by core features of the human genome, which are then built upon by cell specific behaviours resulting in high variability. While the field has progressed significantly since the Berezney et. al. paper in 2000[128], their closing remarks sum up the potential of the field:

> "Much remains to be done, but the fascinating results obtained in the studies reviewed here suggest that the story will continue to be interesting and exciting . . . "

3

Sequence-Based Modelling

Sequence based modelling seeks to predict a target cell behaviour with information derived primarily from the DNA base pair sequence. This ethos towards modelling stems from the hypothesis that the underlying behaviour common between all cells in the same organism is driven by the DNA sequence, as this is also shared between cells. In such cases where average behaviour predictions are being made, we believe that the effect of the non-sequence influences can be averaged out by considering a wide range of cell types. This is conditional on the distribution of the data across cell types permuting around a single average behaviour, and as such investigating whether there are cross-cell type clusters is also necessary for defining the value to predict. In this paradigm, the average behaviour of an organisms cells (or groups therein) can be derived from the sequence of base pairs alone, and the cell-specific behaviour is a permutation around one of these average behaviours. These permutations would be governed by external influences, and could be modelled by including extra data accordingly. Modelling and understanding the purely sequence driven behaviours of a cell has a broad spectrum of potential uses in genetics. For example, being able to computationally assess (without additional experimental effort) the basic effect of DNA modifications on a cell-level behaviour allows for *in silico* discovery of novel effector variant behaviours, and fast tracks

the initial search process for useful experiments to be carried out to validate them. In addition, the deviation of cell-type behaviour from the learned background behaviour could be calculated, and these residuals analysed to observe the cell type specific effects present in the data.

In this work, we take on the challenge of modelling RT from a sequence driven perspective to explore the advantages above. Working with DNA sequence information provides us with a great deal of flexibility for approaching the modelling process, and a lot of considerations have to be made to best explore the space of possible modelling choices. In this work, the majority of traditional modelling approaches are carried out in the R software package, and subsequent "deep learning" work is programmed in Python, using the PyTorch framework[175,176].

3.1 Encode: Informative DNA Encodings

We are most familiar with DNA being represented as a series of characters, canonically `"A", "C", "G", and "T"`. This representation of the sequence is intuitive for most observers, and allows us to see patterns when comparing short sequences. This is particularly helpful for visually identifying regions of heavily repeated sequences, which are common around promoters and telomeres, and for visualising trends/motifs through sequences logos[177,178]. However, this representation requires the image-processing and biological experience of a human user and is unintuitive for automated modelling.

To convert into a format that is more interpretable for algorithmic analysis, the sequence can be converted into a wide range of representations or *encodings*. There already exist a number of excellent surveys on possible DNA encodings[179]. The encoding chosen is influenced by the type of analysis we wish to carry out. Processing the encoding directly can benefit from including biologically-relevant encodings such as physiochemical information or structural relationships. Alternatively, the encoding used can be designed to make the sequence more amenable to techniques from traditional digital signal processing (DSP) such as spectral and fractal measures.

These encodings are not required to preserve the length of the sequence being analysed, but an immediate compression of the sequence can have drawbacks for some interpretability tasks. For many downstream tasks it is useful to apply constraints to final embedding used. For example, if every sequence is supposed to have a unique predicted value it is important that all sequences map to a unique part of the embedding space without unwanted collisions. For the discussions outlined here, I will refer to a DNA sequence S of length L such that:

$$\begin{aligned} S &= (D_1, D_2, D_3, ..., D_L) \\ D &\in \{A, C, G, T, N\} \end{aligned} \tag{3.1}$$

where N represents an unknown/unspecified base. D is a subset of the IUPAC DNA code set, which includes other forms of ambiguous base that can represent different combinations of bases[180].

3.1.1 Direct Mappings

The simplest form of DNA encoding is translating the canonical character representations into a series of numbers, with each sub-sequence being mapped to one or more numbers by a pre-determined relationship. If there is no priority to include further biological context, each D in S can be mapped to a single vector. All direct mappings produce a vector of size $(M \times L)$ where M is the size of the value/array each base is mapped to. Categorical mappings that convert each D to an arbitrary integer value are rarely used in machine learning (ML) applications, as the distances between pairs of bases in the resulting space are not grounded in any understanding of base-pair relationships.

Perhaps the most ubiquitous representation of a DNA sequence in modern DL applications is the Voss or *"one-hot"* encoding, where each base N is given a sparse representation, where only one index is active, as visualised in Equation (3.2)[181]. This transforms a sequence originally of shape $(1, N)$ into $(4, N)$, mapping each base onto it's own channel or dimension, producing a sparse output vector that spaces all bases equally distant to each another. This system is efficient to

interpret, and when considered as 4 individual functions of base occupancy at each position in S, it is amenable to analysis by the discrete fourier transform (DFT), which has revealed consistent periodicities across the human genome, such as a 3-periodicity in coding regions (driven by codons and nucleosome structure), and more discrete "hidden" periodicities of 10-11bps driven by nucleotide pairings and binding of DNA to histones[182,183].

$$\begin{aligned} A &= (1,0,0,0) \\ C &= (0,1,0,0) \\ G &= (0,0,1,0) \\ T &= (0,0,0,1) \end{aligned} \tag{3.2}$$

The Voss encoding also shares similarities to the PWM representation of DNA motifs. This property is particularly useful for interpreting the outputs of networks that dynamically learn features based on this encoding, such as convolutional neural networks (CNNs). By ordering the rows of the one-hot encoding such that complementary bases are mirrored around the centre of the channel dimension (for example, $G-A-T-C$) it is possible to convert an encoded sequence into it's reverse complement (other DNA strand) by simply reversing the direction of both axes of the matrix representation. This characteristic is useful in the model design process when the genome feature in question is strand-independent and thus should respond equally to the "forward" and "reverse complementary" versions of a sequence, which is true for RT.

Additionally, we can consider encodings that instead of being human curated are learned from the distribution of sequences in existing genomes. This concept often has been applied in natural language processing (NLP) to decompose long sentences and documents into "tokens"; which are then encoded into a vector space that may have semantically meaningful dimensions or distance measures. A simple tokenisation of the english language would break sentences by blank spaces, referred to as "word-level tokenisation". It is possible to learn an efficient token encoding

into an arbitrarily sized encoding space by attempting to compress and re-construct an existing corpus of data, as demonstrated in the word2vec work[184].

If you are comfortable generalising "language" to "a sequence with a fixed alphabet, which encodes information with consistent interactions" then these techniques can be broadly applied to DNA analysis as well[185]. Works such as "seq2vec", "dna2vec", "BioVec", "fastteRgram", "LSHVec", and "BioSequence2Vec" have used this strategy of embedding properties to enhance performance in prediction settings, such as effectively separating and classifying sequences from different organisms[185–188]. Increasingly, attention-based architectures that learn dependencies across an input sequence have been pre-trained on bodies of genomic data, such as reference genomes, to improve performance on downstream tasks. The embedding that is learnt within the model is difficult to extract, but can be used in the context of fine-tuning the model for particular tasks producing very strong predictors[185,189].

3.1.2 Count-based Encoding

The length of the encoded representation does not necessarily have to be equal or even a function of the original sequence length. Accumulating information from the full sequence into a compressed representation can ba a powerful way to extract useful and task-relevant features without the computational burden of trying to disentangle behaviours in the full sequence while a model is trained[190]. If there are known subsequences that have an experimental or theoretical relationship with the target variable, then counting the number of density of their occurrences provides a useful dimension for the encoding - for example G-quadruplexes have a strong association with replication origin location and as such could be used as a feature to predict the probability that an unknown sequence could act as a human origin site[82].

More generally, one would be interested in embedding the overall sequence composition of a DNA sample, which can be done by counting the occurrence of all permutations of subsequences of length K, referred to as "k-mers"[191]. K-mer counts present a strong baseline on unaligned sequence comparison tasks, and graphs that map their relative position in sequencing samples are essential for *de novo*

assembly. As a great deal of previous research has been done on the biochemical and structural properties of short k-mer sequences, there are software packages available that are able to generate features for DNA (and RNA) sequences that include counts and other information, such as PyFeat[192]. These features are powerful ways of encoding DNA sequences behaviour, however as the K value used grows, the number of possible features increases at a rate of 4^K, which makes including all high-K features prohibitively expensive for a model. It is possible to apply dimensionality reduction techniques such as PCA to the counts of high-K k-mers to capture the useful variance without providing all the original values, at the cost of exact information and easy interpretability[174]. It is also possible to explore the dynamics of transitions between k-mers in a DNA sequence, and these have been used to cluster similar sequences into phylogenetic trees[193]. DNA sequences have a strong periodicity of length 3 due in part to the presence of coding information in codons, and patterns in the 3-mer and codon information can be exploited to identify patterns that evolutionarily distinguish different organisms[27].

Finally, the information stored about the subsequences can be extended beyond simply counting. The "accumulated natural vector" is a compressed representation of a DNA sequence that guarantees a unique mapping of a sequence of any length to an 18 dimensional space, where the dimensions correspond to the counts, average distance from the end, divergence, and covariance of the 4 nuclear bases[194]. Comparing Euclidean distances between these encodings allows for efficiently carrying out alignment or similarity based tasks such as reconstruction of phylogenetic trees.

3.2 Embed: Information Propagation and Interpretation

In the prior discussion, we outlined some classes of ways that DNA information can be encoded into a format more easily readable by models. This can be conceptualised as a mapping between a difficult to interpret space (for machines) to a space that is

both easier for the machine to parse and has useful semantic relationships between its parts (or "dimensions") for downstream tasks. The natural question that follows this process is: how do we use this encoded DNA for these aforementioned tasks? The answer usually lies in projecting this new DNA encoded feature space in some way to create new *embedding* spaces, some of which could be single-dimensional and correspond directly to the target of the downstream task. The important job therefore is working out the best processes for carrying out these transitions; what are the most useful and semantically meaningful methods to carry out this re-embedding process? Can we choose a subset of these methods which improve our understanding of the phenomena being modelled?

Defining meaningful new embeddings and the transitions between them is a very broad question, and strongly contingent on the form of the input data. In general terms, we can consider the flow of information through the transitions we are creating and compare the relative amount of information that is present at any particular stage. For input encodings that already aggregate information into dimensions that correspond to genome features, such as count-based encodings, the relative positional structure of the input is not very useful, and we are interested in combining the information from the different dimensions. The fields of statistics and ML have defined an incredible spectrum of methods to accumulate information in feature dimensions and map them to a space that is useful for prediction, and a full description is beyond the scope of this work. In general, it is notable that relationships between biological features can be linear or highly non-linear, and it is rarely possible *a priori* to know which methods will most effectively "solve" the projection you need. There is no guarantee in genomics that the simplest solution is the best, however a simpler solution is often preferred if it produces a more interpretable result and is less likely to have accidentally "over-fit" to noise in the data that does not correspond to the biological information. If we are going from a very high-dimensional and sparse representation such as the "one-hot" DNA encoding, information is originally spread thinly and homogeneously throughout the input and we seek to condense it down into a much lower dimensional space

that corresponds to, in our case, a single RT value. This transition from sparse to dense will require a significant amount of information aggregation, and as such we need operations that can do efficiently by respecting the patterns and structure within DNA. In some genomics problems there are shared semantic anchoring points around which multiple sequences can be aligned, such as TSSs for gene expression prediction. This makes the relative position of each input base important and useful for downstream prediction tasks, and in those cases explicitly adding that information to the model through an engineered *positional embedding* can significantly improve performance, especially in transformer-like architectures[195].

The ability to take information from different parts of the sequence and use them as context for decision making is an essential component for complex reasoning over sequences in general, and is the driving force behind the development of recurrent neural networks (RNNs). These neural networks are capable of holding an internal "state" as they scan along a sequences, and "learn" the best domain-specific rules to accumulate and forget information in this state as new parts of the sequence are encountered. In the context of DNA, these models would move along the sequence or a compressed encoding of it (usually in the 5' to 3' direction) and learn during training to accumulate information about features to predict genome behaviours. RNNs are not limited to local interactions, and using more complicated memory and attention mechanisms can be generalised to "reason" about very long sequences - up to 1 million bases of DNA "context" simultaneously[189]. While these models are very powerful predictors in genomics and other fields, their internal mechanisms can be difficult to interpret, and the field of "mechanistic interpretability" attempts to disentangle the "brains" of these models to understand the algorithms they may have "learnt" to approximate during training[196].

Improving model interpretability is an essential stage of the modelling process, and is vital for trust in models that are being applied to tasks that have immediate human impact. In genomics, we are fortunate to have decades of theoretical and experimental knowledge to draw on when designing and interpreting our models, to ensure that their prediction mechanisms matches those observed. These include

a well-grounded model of protein-DNA interactions with many known "motifs" of DNA binding behaviour, and how the collective binding of proteins can influence the structure and expression of genes from the DNA in an additive way. For our models that work off pre-encoded features, if we have ensured that the features used are amenable to interpretation, we can look for patterns and interaction within our models that align with known regulatory behaviour. In models that work directly off the encoded DNA, such as one-hot encoded sequence, we can take advantage of the intrinsic behaviours of certain deep neural network (DNN) layers to both extract useful motifs/features, and identify interactions between them, reviewed in Novakovsky et. al.[197]. In brief, convolutional neural networks (CNNs) are a neural network layer parameterised by relatively small "kernels" that are "convolved" over the inputs to produce an output that relates how close each part of the input is to the kernels. In the context of DNA, each of the CNN kernels can be thought of as a DNA motif, and the response from the CNN layer as a measure of the similarity between each part of the DNA sequence and the motif. This is an exceptionally useful property, as it allows us to optimise these "kernels" for DNA motif finding tasks very efficiently. The convolution process means we do not have to learn all the weights to detect a motif at every location, we just need to learn the motif in a position invariant form, similar to a PWM. Actively modulating the optimisation these kernels can encourage sparsity (use few motifs where possible) and cohesion (do not split the motifs across multiple kernels) which makes it more likely that we can recover the optimised motifs after training simply by looking at the kernels[198]. However, the ease with which the kernels can be interpreted depends on both the training and architecture, with "shallow" networks often finding whole motifs and "deeper" models preferentially splitting the motifs across multiple kernels[199]. If we know that the feature we are interested in modelling is strand-invariant (RT) we can further capitalise on the CNN kernels structure to simultaneously detect if the reverse complement is present in a sequence with no more parameters. This can be used to produce models that are reverse-complement invariant, or efficiently parameterize a model that processes each strand independently downstream[200–202].

Identifying the learned motifs in a model can be approached in other ways than directly inspecting the CNN weights. These approaches assumes that a model has optimised its predictions sufficiently to closely approximate the underlying genomic behaviour, which may not always be the case as models with many parameters can over-fit easily. Model interpretability techniques from other modalities, such as images in computer vision, have been translated to genomics workflows, resulting in packages such as DeepLIFT that are able to highlight positions in an input sequence by the strength and sign of their contribution to the final prediction[200,203]. These "attributions" can then be provided to software such as TF-MoDisco (TF motif discovery) that isolates and clusters small motif regions that are frequently strongly attributed to the model's performance[204]. Other packages such as DeepResolve and NeuronMotif attempt to directly disentangle the motif syntaxes that the model has identified by analysing the models weights, utilising the symmetries and redundancies in some representations and side-stepping the requirement of additional inference passes on unseen data[205,206]. Other approaches to model interpretability often involve observing changes in the model's behaviour on a set of modified sequences, and has culminated in the study of "local" and "global" interactions between motifs and the performance of the model to determine learned behaviours[207,208]. Exhaustive approaches to investigate the effect of sequence changes are carried out with *in silico* saturation mutagenesis (ISM), which has recently been greatly accelerated by methods that reduce redundant recalculations[209,210].

3.3 Predict: What can be effectively modelled?

As we have discussed above, the field of genomics is very well aligned for analysis with modern ML approaches, and the flexibility of DL to process and extract features from a wide-range of input modalities is especially powerful given the diversity of assays available. A great deal has been written to summarise the application of DL to tasks in genetics, and we would like to signpost the review by Zhang et. al. on the application of DL to motif finding as a comprehensive primer on dominant architectures and problem settings[211]. In addition, the primer by Zhou

et. al. provides an overview of terminology, workflows, and existing subdomains of genomics that are being targetted with DL learning systems. In this section, I will discuss some exemplary tasks and models in the field, and some key tools in the software ecosystem available for researchers.

A cornerstone task in modern computational genomics is sequence alignment and similarity. While a great deal of traditional algorithms exist to discover patterns *de novo* from a pool of DNA sequences, the tasks can still benefit from the encoding and information processing capabilities of sequence-based learning. For example, the unsupervised compression of DNA sequences can expedite unaligned comparisons, and take advantage of atypical encodings such as the "chaos game" to speed up the extraction of meaningful features[194,212,213]. However, in contrast to alignment, many tasks in modern genomics revolve around detailed understanding of a particular sequence locus or behaviour, such as the (ongoing) search for a consensus sequence or set of motifs for human ORIs. After conceptual and pattern-mining work reveals sequence trends around origin sites, methods were developed to use extract DNA descriptive motifs (such as 3-length k-mer counts) to characterise and predict ORIs with some success. This tasks has been approached with a wide range of sequence-based ML architectures, with support vector machines (SVMs) and XGBoost classifiers going toe-to-toe with supervised DNA embeddings[214–216]. Similarly, XGBoost shines in other tasks that involve processing low-level DNA sequence features and patterns, producing leading performance in the prediction of G-quadruplex formation likelihood, or the probability of double-strand breakage, when additionally using epigenomic features[217,218].

One of the problems that has been most commonly approached with sequence based machine learning is attempting to learn the regulatory mechanisms of gene expression through *de novo* discovery of TF binding motif syntaxes. These works primarily employ the motif-mimicking capabilities of CNNs discussed above, and use the experimental ChIP-seq tracks as their supervision data[219]. The DeepSTARR architecture was designed to classify enhancers as either involved in "housekeeping" or "developmental" gene regulation, and uses this signal to encourage the model

to discover patterns of enhancer-promoter interaction, which can be subsequently disentangled from the model using interpretability score[220]. This approach is the latest in a lineage of "Deep-" models that classify TF binding at sites such as "DeepBind", "DeeperBind", and "DeepSNR"[221–223]. However, detection of TF binding is not the end of the story, and as single base-pair resolution binding data for different alleles became available, the ChromBPNet architecture was developed to predict these and identify hierarchical relationships in the binding patterns that represent enhancer activation[204,224,225]. The regulatory syntax of TF motifs has also been tackled with self-attention layers, one of the core components of Transformer architectures, in the SATORI model, which is able to produce interpretable interaction patterns between TF motifs for small DNA inputs[226]. A recent standout paper in the field detailed the "Puffin" architecture, which unravels the sequence basis of transcription initiation through a multi-step modelling process: first fitting a large model to predict motif activations, then using the outcomes of interpretability analysis to produce a smaller model that compositionally predicts initiation as a function of certain motifs and tri-nucleotide patterns, which they believe are caused by CpG islands[227].

A related problem in genomics is a sequence-driven understanding of genome 3D structure, ranging from local DNA openness and chromatin accessibility to long-distance interactions along chromosomes. Again, the field is dominated by CNN-based architectures that extract local features from 1-D ATAC-seq tracks or 2-D Hi-C image representations. "Basset" is a pioneering model in genome accessibility that predicts DNase-seq data for over 150 cell types to attempt to learn relevant sequence motifs and dependencies[228]. Similar attempts to model open chromatin as a proxy for structure and gene regulation have used ATAC-seq data, and "AtacWorks" ingeniously learns to simultaneously de-noise and predict peaks in ATAC, which encourages very strong predictive performance[229,230]. ATAC-seq data has also been used as proxy-information to create more efficient DNA sequence embeddings, which can be used downstream to predict the developmental affinities of cell-types by embedding TF motifs[231]. At a large scale, models have

been developed to predict chromatin loops, and consequently local chromatin structure and compartmentalisation again using a combination of CNN features and RNN/Attention information propagation[232–235]. Finally, a wide range of models have been produced to predict the full 3D structure of chromosomes from Hi-C data at as low as single kilobase resolution, with techniques also relying heavily on a CNN backbone over the DNA sequence that is then interpreted and expanded by later layers into a 2D representation[236]. Models such as "C. Origami" and Akita demonstrate *de novo* discovered interaction with CTCF motifs after training, and "siamese" networks have been trained to distinguish between biological and technical noise resulting in embeddings that provide meaningful distances between Hi-C patches[237–239].

Across all these domains of genomic biology, there is a commonality that the strongest performing models are often pre-trained and/or multi-objective. Models that are "pre-trained" first optimise for one or multiple auxiliary tasks which allow it to learn useful sequence features in its earlier layers[240]. "DeepC" is an example of a Hi-C prediction model that was originally trained to predict chromatin features, a related but very structurally different output modality[241]. Finally, the models that present the strongest potential for downstream tasks (or "Foundations") are those that have been pre-trained and predict for a wide range of modalities. The "Enformer" model uses transformer block and a wide (and very computationally intensive) context field along the genome to predict many tracks in both human and mouse genomes, and has already been used in a range of variant studies or as an embedding-module to use in other tasks in the "seq2cells" work[242,243]. Most recently, the "Sei" and "HyenaDNA" models have been trained and evaluated on large amounts of data and shown to contain very useful and sensitive DNA embedding within their hidden layers[189,244]. "Sei"'s DNA "latent space" shows high sensitivity to small DNA variants, and "HyenaDNA" provides an efficient alternative to Transformer layers allowing for much wider sequence contexts, up to 1 million bases. In addition, HyenaDNAs self-supervised pre-training on the human reference

genome allows it to beat state-of-the-art (SOTA) in existing GenomicBenchmark tasks with very little fine-tuning[189,245].

3.4 Tools for Sequence-Based Modelling

In our work, we trial multiple ML and DL approaches to work out which is best suited for the tasks of predicting RT. Over the last two decades, an ecosystem for machine learning modelling has flourished, catalysing many disciplines and lowering the bar for entry for many modelling approaches. Packages such as "scikit-learn" provide such an interface for many ML algorithms, and are used extensively in this work for model training, and evaluation[246]. In addition, implementations of powerful specific ML algorithms such as XGBoost, and more recently LightGBM, have becomes cornerstones of the academic and industrial ML workflow[247–249]. Creating powerful and fair datasets to train these systems on is of paramount importance, and software packages to make data loading more efficient, generate synthetic but representative datasets, and augment biological sequence inputs with additional contextual information are vital for boosting model performance[250–252]. Developing models to use these datasets is facilitated by the glut of high-level languages that are able to interface with faster low-level ML and DL components. As demonstrated by the range of implementations in the above, there are many options for approaching the creation and analysis of genomic models, such as "Selene"[250]. Fortunately, the open-source community provides software packages to assist in the developments and interpretation of model findings[253,254]. Finally, once these models have been created there is a growing ecosystem of depositories to host and test models for ease of reproducibility, such as "HuggingFace" and "Kipoi"[255].

3.4.1 Buyer Beware! Modelling Considerations and Concerns

Unfortunately, despite often possessing strong predictive power, ML is not resistant to human error, and ML in genomics presents many opportunities for the performance and interpretation of our models to be skewed. The excellent paper by Whalen

et. al. outlines 5 key pitfalls in modern ML that can influence the performance of models in genomics: distributional differences, dependent examples, confounding, leaky pre-processing, and unbalanced classes[256]. These traps are possible in most fields that employ ML, and some (especially confounding and dependent examples) are particularly hard to identify in genomic datasets. This article recommends clear best-practices for easily avoidable pitfalls (4 and 5), and suggests tools and processed that attempt to help mitigate the effects of the remaining pitfalls. Much remains to be done, and vigilance is paramount when writing our own or reviewing other literature to ensure we are not falling into these.

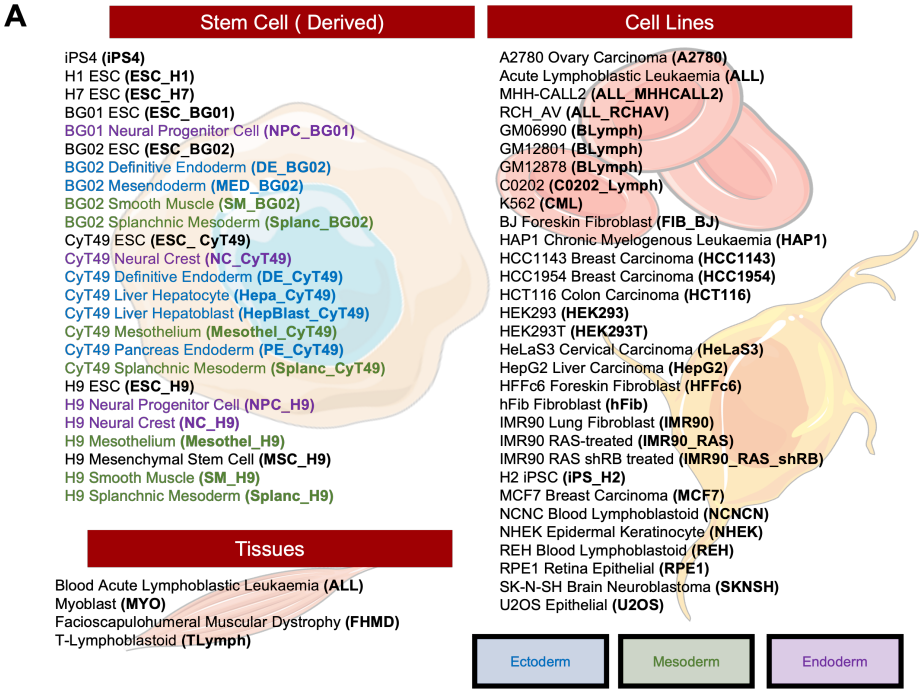
As data becomes available from more organisms and individuals, each at increasingly detailed resolution there is a trend toward the creation of "Foundation" models that incorporate these vasts amount of data as "pre-training" material to accelerate "fine-tuning" performance on a different downstream task. This has already provided a series of powerful models that are being used and interrogated readily throughout the field, providing a fruitful hub for the growing DL in genomics field. However, these models still require improvements before they can be confidently used for tasks beyond their original training data distribution. Studies showing the Enformer model struggled to generalised to predicting the effect on gene expression of SNV in the input sequences[257]. Furthermore, in another study there was strong disagreement between Enformer and 3 other state-of-the-art models[258]. There are growing concerns that a poorly-documented scattergun approach to Foundation models will lead to significant contamination of the training data to the point that it is impossible to reliably assess the performance of the model on unseen data.

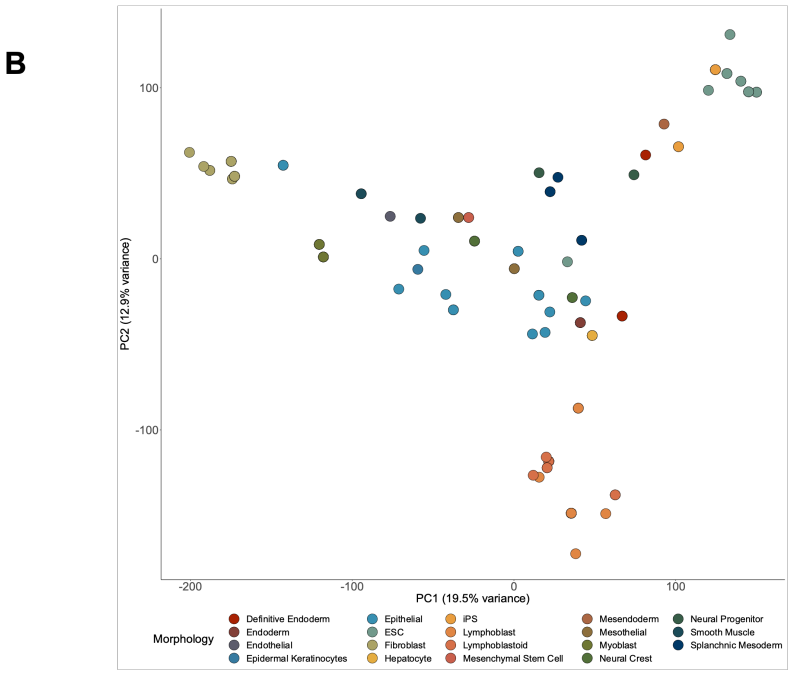
4

Replication Timing Dataset

4.1 Database Sourcing

The primary data source for this work was Replication Domain – a curated repository of RT data from many organisms and cell lines[259]. 197 samples of data for the human genome (version UCSC hg38) were downloaded from over 30 unique cell lines including iPSCs, embryonic stem cells (ESCs), and cells with leukemic or lymphoblastic origin. After consultation with original authors and depositors 22 datasets were not included as the original raw data is no longer publicly available. For all remaining datasets, the GEO and 4DNucleome accessions and associated publications are available for reproducibility as the Replication Domain website is no longer maintained. A further 66 datasets were excluded, such as biological replicates when the average profile was already included, samples from mouse or monkey lineages mislabeled as human samples, or datasets containing low genome coverage such as only data from a single chromosome. As some of the datasets were originally generated on different reference genomes, the sets were lifted over to hg38 on the Replication Domain website. In addition, we used 5 of the deposited processed RT datasets from cell lines in the Koren Laboratory, who have previously published high-throughput RT experiments and computational processing[125]. These were included in the analyses to ensure that our dataset was composed of RT


A
Stem Cell (Derived)
iPS4 (iPS4)
H1 ESC (ESC_H1)
H7 ESC (ESC_H7)
BG01 ESC (ESC_BG01)
BG01 Neural Progenitor Cell (NPC_BG01)
BG02 ESC (ESC_BG02)
BG02 Definitive Endoderm (DE_BG02)
BG02 Mesendoderm (MED_BG02)
BG02 Smooth Muscle (SM_BG02)
BG02 Splanchnic Mesoderm (Splanc_BG02)
CyT49 ESC (ESC_ CyT49)
CyT49 Neural Crest (NC_CyT49)
CyT49 Definitive Endoderm (DE_CyT49)
CyT49 Liver Hepatocyte (Hepa_CyT49)
CyT49 Liver Hepatoblast (HepBlast_CyT49)
CyT49 Mesothelium (Mesothel_CyT49)
CyT49 Pancreas Endoderm (PE_CyT49)
CyT49 Splanchnic Mesoderm (Splanc_CyT49)
H9 ESC (ESC_H9)
H9 Neural Progenitor Cell (NPC_H9)
H9 Neural Crest (NC_H9)
H9 Mesothelium (Mesothel_H9)
H9 Mesenchymal Stem Cell (MSC_H9)
H9 Smooth Muscle (SM_H9)
H9 Splanchnic Mesoderm (Splanc_H9)
Tissues
Blood Acute Lymphoblastic Leukaemia (ALL)
Myoblast (MYO)
Facioscapulohumeral Muscular Dystrophy (FHMD)
T-Lymphoblastoid (TLymph)
Cell Lines
A2780 Ovary Carcinoma (A2780)
Acute Lymphoblastic Leukaemia (ALL)
MHH-CALL2 (ALL_MHHCALL2)
RCH_AV (ALL_RCHAV)
GM06990 (BLymph)
GM12801 (BLymph)
GM12878 (BLymph)
C0202 (C0202_Lymph)
K562 (CML)
BJ Foreskin Fibroblast (FIB_BJ)
HAP1 Chronic Myelogenous Leukaemia (HAP1)
HCC1143 Breast Carcinoma (HCC1143)
HCC1954 Breast Carcinoma (HCC1954)
HCT116 Colon Carcinoma (HCT116)
HEK293 (HEK293)
HEK293T (HEK293T)
HeLaS3 Cervical Carcinoma (HeLaS3)
HepG2 Liver Carcinoma (HepG2)
HFFc6 Foreskin Fibroblast (HFFc6)
hFib Fibroblast (hFib)
IMR90 Lung Fibroblast (IMR90)
IMR90 RAS-treated (IMR90_RAS)
IMR90 RAS shRB treated (IMR90_RAS_shRB)
H2 iPSC (iPS_H2)
MCF7 Breast Carcinoma (MCF7)
NCNC Blood Lymphoblastoid (NCNCN)
NHEK Epidermal Keratinocyte (NHEK)
REH Blood Lymphoblastoid (REH)
RPE1 Retina Epithelial (RPE1)
SK-N-SH Brain Neuroblastoma (SKNSH)
U2OS Epithelial (U2OS)
Ectoderm
Mesoderm
Endoderm


B
100
0
-100
PC2 (12.9% variance)
-200
-100
0
100
PC1 (19.5% variance)
Morphology
Definitive Endoderm
Endoderm
Endothelial
Epidermal Keratinocytes
Epithelial
ESC
Fibroblast
Hepatocyte
iPS
Lymphoblast
Lymphoblastoid
Mesenchymal Stem Cell
Mesendoderm
Mesothelial
Myoblast
Neural Crest
Neural Progenitor
Smooth Muscle
Splanchnic Mesoderm

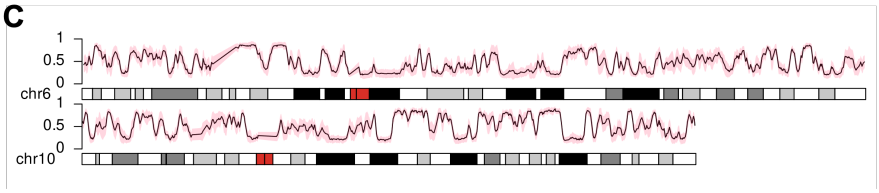

C
1
0.5
0
chr6
1
0.5
0
chr10

Figure 4.1: (A) Overview of the variety of cell types for which was have data, divided and coloured by broad categorisations and annotated with a shorthand **(cell type)** name for easy reference. (B) 2D PCA plot generated by appling PCA to the full genome RT profiles of each cell type and plotting the first two principal components (PCs). Each point is coloured by the labelled morphology extracted from Replication Domain[259]. While there is clear local structure that groups similar cell types together, the PCs plotted account for only 30% of the variance in the dataset. (C) Two sample RT profiles of the average RT behaviour across cell types from chromosome 6 and 10. These profiles show clear RT domain and transition region dynamics, and the shaded 1-standard deviation regions in red highlight that there are both areas of strong agreement and discordance between the cell types.

data from a range of laboratory sources and experimental methods, which ensures robustness of the final model created. After aggregation, 131 datasets remained for further analysis and are visualised in Appendix 1.

The processed RT datasets are the result of 4 major experimental and data processing pipelines. The majority come from Repli-chip experiments carried out before 2019, using the methodology proposed by Hiratani et. al.[119] The computational processing of the raw early and late hybridisation data begun with normalisation of individual datasets with median absolute deviation (MAD) scaling and are then smoothed with a loess-fit to average out experimental divergence. The remaining datasets use a variation on the 2-stage Repli-seq method, that utilises deep sequencing of early and late cell fractions to compute RT values at high dynamic ranges[121]. Repli-seq datasets from Replication Domain were processed with pipelines from ENCODE and the 4DN project[260,261], and the Koren Lab datasets used a modified Repli-seq method described in the accompanying publication[262].

4.2 Database Processing

The datasets that remained after filtering report RT values at different genome-wide sampling resolutions, ranging from 100bp to 5kbp. This caused disparities in the smoothness of the RT profiles and provided a barrier for reliably comparing the RT values between datasets. To combat this, we investigated methods of binning the RT data into fixed-size bins, into which each dataset would contribute multiple points

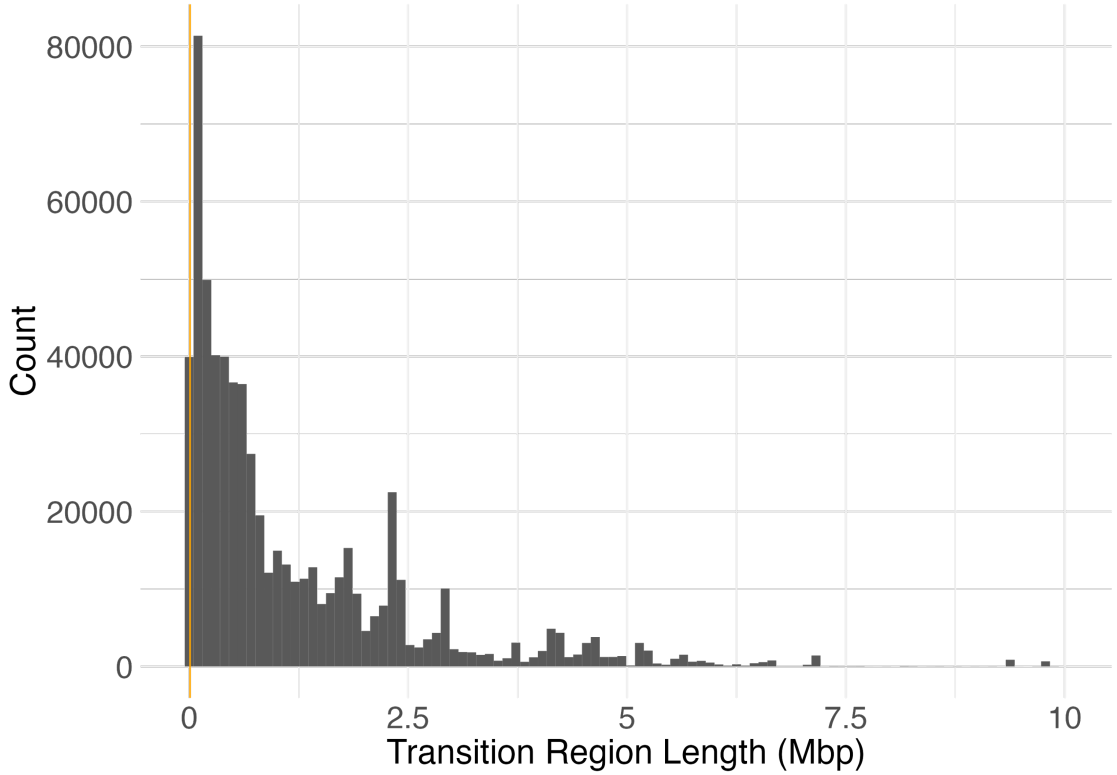


Figure 4.2: Distribution of RT transition timing region (TTR) widths accumulated along the full genome of each cell type profile. The vertical orange line indicates the chosen binning threshold of 10kb, and demonstrates that the vast majority of identified transition regions are larger than this resolution and will thus not be smoothed-over by the binning process.

that would be averaged. Previous works on large-scale RT data have used bin sizes of 50 and 100kb, but as binning comes with an inevitable loss of some information we wanted to be confident that the RT profile dynamics, such as TTRs, would not be smoothed over by the process[82,165]. We profiled the width of all of the RT "transition" regions in the dataset, Figure 4.2, and verified that they exceeded 10kb in length, with the vast majority doing so by an order of magnitude, in accordance with qualitative observation of the profiles in full-genome plots. While it would have been possible to choose a lower bin size, this would only have served to reduce the number of points that contribute to the average behaviour being observed, and as many datasets were already sampled at 5kbp we deemed 10kbp to be a reasonable compromise. After binning, all dataset that had been labelled as technical replicates of one another were averaged together to a single dataset.

All of these datasets are the result of taking the log_2 of the ratio of early over late values for the experimental RT assay. As each of the datasets originated from different experiments, cells, and laboratory conditions the resulting datasets were not all originally normalised to within the same range of RT values, which had to be corrected to ensure removal of batch effects. To perform this normalisation, we considered 3 different techniques to ensure that the data was free of detectable batch effects, stringently bring datasets from different experimental backgrounds to a consistent range, and do so without suppressing the individual RT dynamics of each dataset. The first method was quantile normalisation, a standard biostatistics tool often applied to normalisation of microarray data, which enforces strict normalisation through re-assigning values in each bin based on the order of the original values after sorting. The second method divides each value by the standard deviation of the dataset it comes from, in line with the `scale` operation that is often applied to data in machine learning pre-processing, which effectively normalised the data to within a range (-2, 2). Finally, we designed a normalisation technique which calculated a linear fit (through 0) between each dataset and a calculated average set (generated from all datasets whose values exclusively lie in the range ± 2) and used the coefficient of this fit to scale each dataset's values.

The effect of applying each of these normalisation methods can be observed in Figure 4.3, which reveals that they each effectively constrain the datasets to a consistent range, without producing significant artefacts in the individual RT profiles. Additionally, each normalisation technique results in a dataset with very similar overall dynamics and similarities, shown in Figure 4.4 where all the PCA reduced-spaces display similar shapes and structure. We chose to use the standard deviation based normalisation scheme due to it's relative simplicity and generalisability to new data, and because it only scales the RT data thus preserving the individual dynamics of the original datasets.

Using these visualisations, we identify one clear batch effect in the un-processed data where 4D Nucleome samples have a significantly wider range of RT values, Figure B.1, which is effectively removed by normalisation. By colouring the PCA

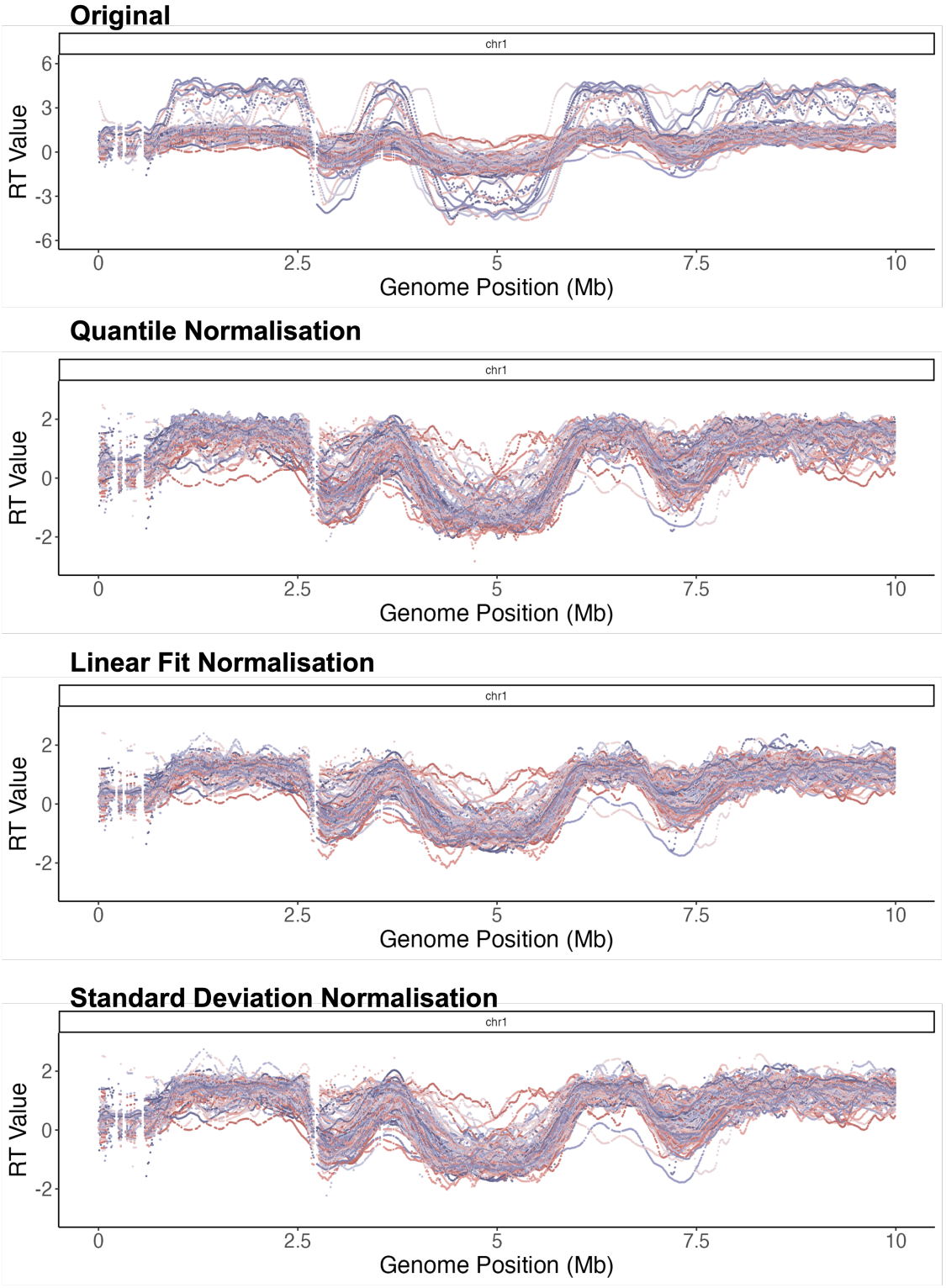

Original
chr1
RT Value
6
3
0
-3
-6
0
2.5
5
7.5
10
Genome Position (Mb)
Quantile Normalisation
chr1
RT Value
2
0
-2
0
2.5
5
7.5
10
Genome Position (Mb)
Linear Fit Normalisation
chr1
RT Value
2
0
-2
0
2.5
5
7.5
10
Genome Position (Mb)
Standard Deviation Normalisation
chr1
RT Value
2
0
-2
0
2.5
5
7.5
10
Genome Position (Mb)

Figure 4.3: Qualitative effect of normalising the original RT profiles with 3 different normalisation schemes. **Quantile normalisation** uses the standard rank-based normalisation technique to align each dataset. **Linear fit normalisation** scales each dataset by its linear fit coefficient to a generated 'reference' RT set from datasets that all lie between ±2. **Standard deviation normalisation** scales all values in each dataset by it's standard deviation. We observe that all normalisation techniques successfully unify the ranges of the datasets, and choose to use standard deviation normalisation as it is conceptually simple, doesn't require an arbitrary decision about a reference set, and ensures all points retain their original RT identity (Early/Late) as it only scales the points distance from 0.

space with different variables in the sample metadata, we identified a possible batch effect from many Lymphoblastoid samples coming from the same GEO series, GSE37987, Figure B.2. However after visualising the effect of deleting all data from GSE37987, we see that the structure of the PCA space is still similar and must be driven by the trends in the remaining dataset, so we consider the close grouping of GSE37987 datasets to be a primarily biological effect.

Finally, to ensure that specific cell types are not over-represented in the subsequent analysis, all dataset that share a cell type are averaged together resulting in 56 final profiles, visualised in Figure 4.5. Each of the plots in Figure 4.5 shows the distribution of data for a dataset after normalisation and binning, and separates by colour the data for each cell type that was contributed from a different original experimental method. We can clearly see that there is a great diversity in RT value distributions, with most but not all exhibiting peaks of Early and Late RT values around ±1 respectively. For all cell types that have contributions from multiple experimental methods, there is variation between the experimental types as would be expected; however, we note that all distributions contributing to a cell type have been brought to the same range and, where there are discernable peaks, they are broadly aligned. This processing resulted in a dataset of RT values for all 56 cell types in 144,000 unique 10kb bins.

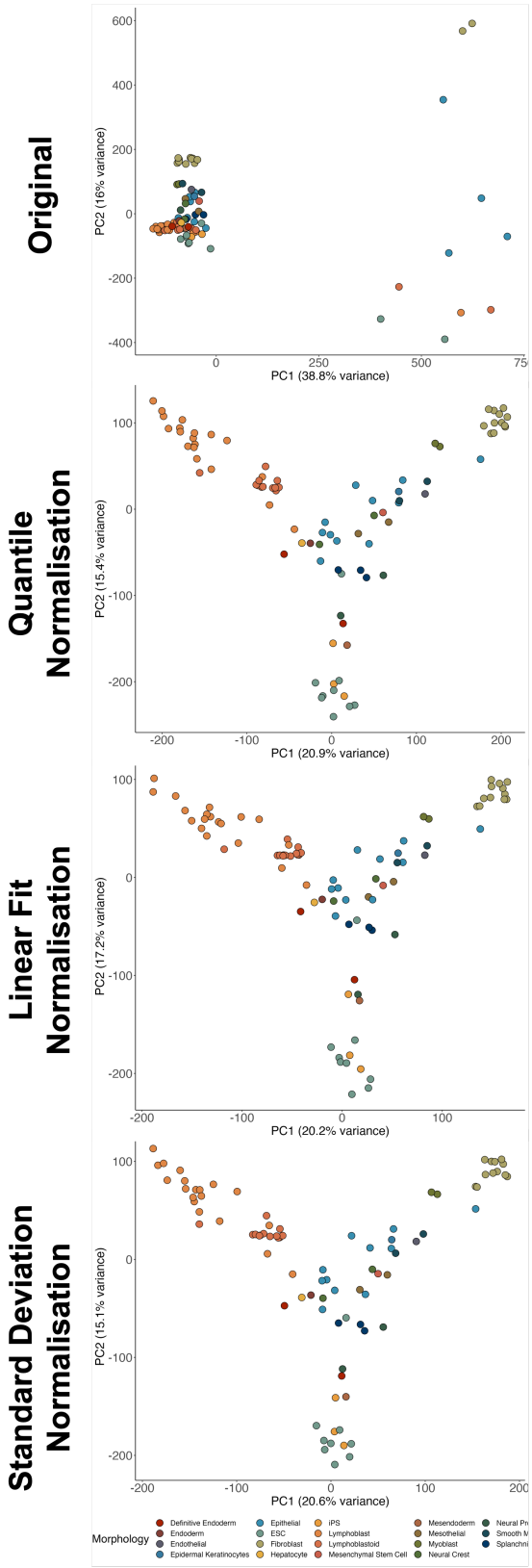

Original
PC2 (16% variance)
PC1 (38.8% variance)
Quantile Normalisation
PC2 (15.4% variance)
PC1 (20.9% variance)
Linear Fit Normalisation
PC2 (17.2% variance)
PC1 (20.2% variance)
Standard Deviation Normalisation
PC2 (15.1% variance)
PC1 (20.6% variance)
Morphology
Definitive Endoderm
Endoderm
Endothelial
Epidermal Keratinocytes
Epithelial
ESC
Fibroblast
Hepatocyte
iPS
Lymphoblast
Lymphoblastoid
Mesenchymal Stem Cell
Mesendoderm
Mesothelial
Myoblast
Neural Crest

Figure 4.4: Visualisation of the resulting low-dimensional space produced by carrying out PCA on the dataset before and after different normalisation strategies. Each point corresponds to an individual genome-wide RT profile, and is coloured by the morphology of the cell type the experiment was carried out on. A clear batch effect is visible in the un-normalised data resulting from the different range of values in all experiments from the 4D Nucleome (4DN) Repli-seq pipeline, visualised in Appendix B. This is efficiently removed by all normalisation strategies, which all produce very similar PCA spaces.

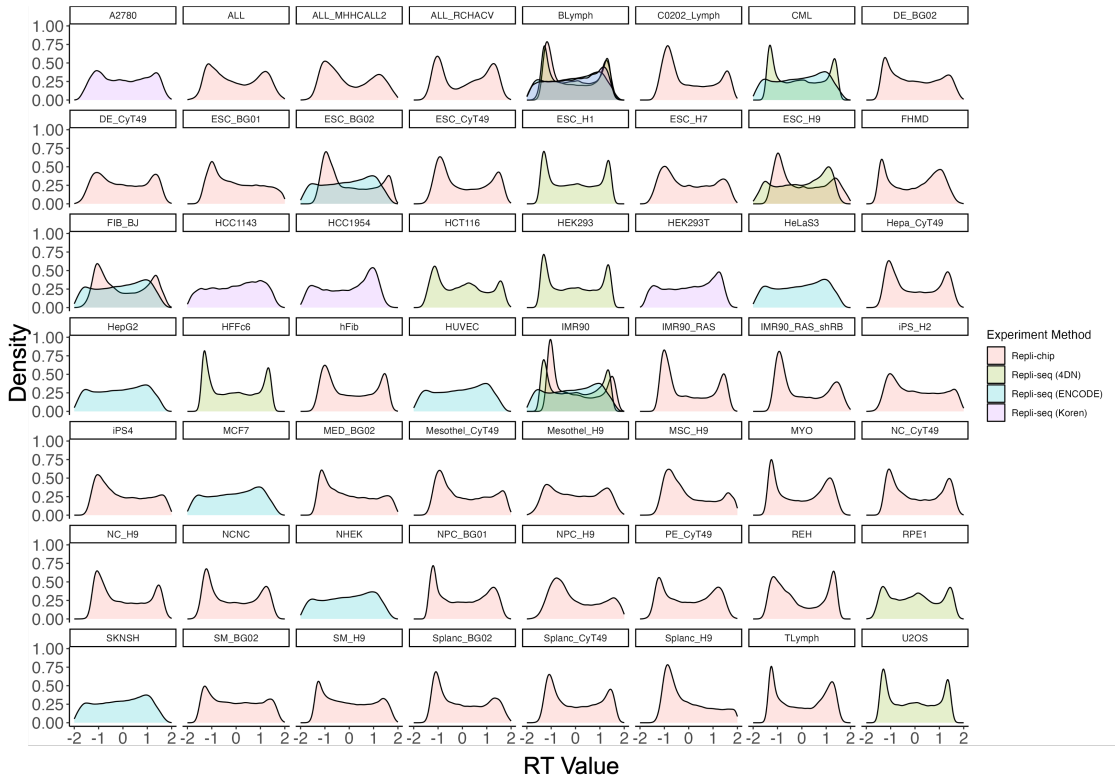
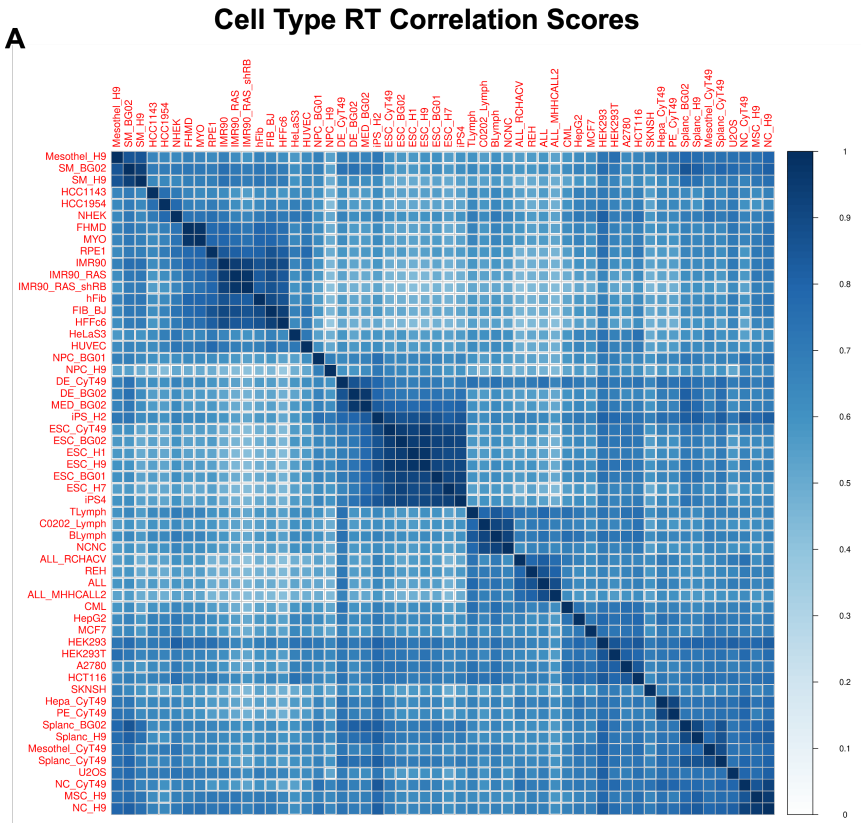
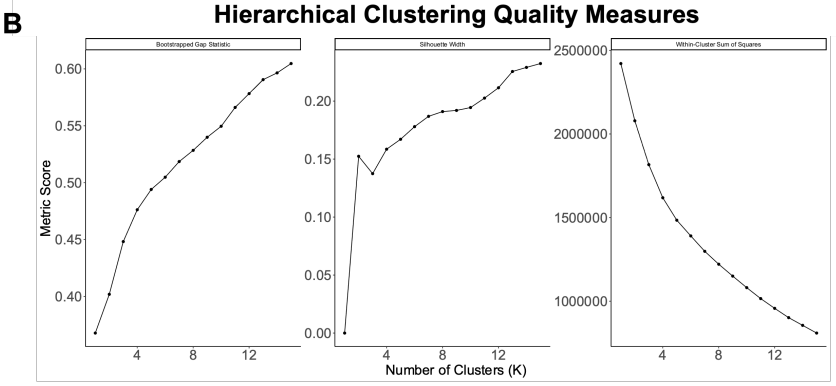


Figure 4.5: Density distribution of all RT values after binning and normalisation within each cell type, colour filled by experimental method. While the majority of the datasets display clear peaks at RT values symmetrical around 0, that have been aligned after the binning and normalisation methods there is clearly still a great diversity in the dataset values which provide interesting modelling challenges downstream.


A
Cell Type RT Correlation Scores
Mesothel_H9
SM_BG02
SM_H9
HCC1143
HCC1954
NHEK
FHMD
MYO
RPE1
IMR90
IMR90_RAS
IMR90_RAS_shRB
hFib
FIB_BJ
HFFc6
HeLaS3
HUVEC
NPC_BG01
NPC_H9
DE_CyT49
DE_BG02
MED_BG02
IPS_H2
ESC_CyT49
ESC_BG02
ESC_H1
ESC_H9
ESC_BG01
ESC_H7
iPS4
TLymph
C0202_Lymph
BLymph
NCNC
ALL_RCHACV
REH
ALL
ALL_MHHCALL2
CML
HepG2
MCF7
HEK293
HEK293T
A2780
HCT116
SKNSH
Hepa_CyT49
PE_CyT49
Splanc_BG02
Splanc_H9
Mesothel_CyT49
Splanc_CyT49
U2OS
NC_CyT49
MSC_H9
NC_H9
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


B
Hierarchical Clustering Quality Measures
Bootstrapped Gap Statistic
Silhouette Width
Within-Cluster Sum of Squares
Metric Score
Number of Clusters (K)

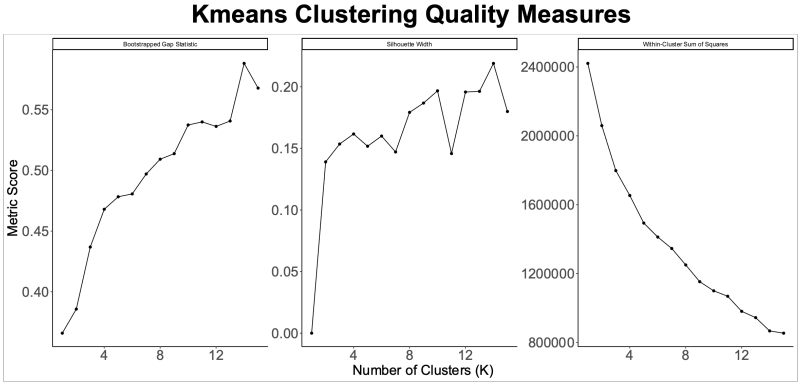

Kmeans Clustering Quality Measures
Bootstrapped Gap Statistic
Silhouette Width
Within-Cluster Sum of Squares
Metric Score
Number of Clusters (K)

Figure 4.6: RT cell type interrelations are not amenable to small K clustering. A) Correlation matrix of all cell type average RT profiles, revealing localised clustering among similar cell types but little in the way of overall trends. B) Clustering quality statistics for hierarchical and K-means clustering for K < 15. While all metrics show iterative improvement, other than some instability with K-means clustering, the improvement for small K values < 4 does not significantly outperform higher K clustering, indicative of little global aggregable behaviour.

4.2.1 Clustering Approaches

As one of the overall goals in this work is to understand the sequence driven component of RT, the first relevant question is to establish whether a single average behaviour best describes the data. We know from literature that, as well as overall RT plasticity, similar cell types that share precursor cell types or similar fates have more-similar replication timing profiles, which can be seen in the PCA plot of all binned cell types RT in Figure 4.1 B. These similarities are also visible in a correlation comparison between the RT profiles of all 56 final cell types, Figure 4.6 A, with subsets of the correlation matrix showing high correlations that visually correspond to known similar cell types, such as ESCs and Fibroblasts. However, despite strong cell type inter-relations, there is no immediately visible partitioning of the full "cell space" into a small number of sub-clusters that would suggest multiple overall average behaviours.

To investigate this more rigorously, we trialled K-means and hierarchical clustering on the cell type RT profiles for a range of possible clusters, and investigated measures of the clustering fit quality, Figure 4.6 B. A suggestive results that there are multiple average overall RT profiles would show a small number of clusters (< 4) possessing superior clustering capability than a larger numbers of clusters, indicating that a smaller number of clusters provided an accurate but more parsimonious clustering of the data. However, as is visualised for both Hierarchical and K-means clustering there is no significant increase, in "Gap Statistic" or "Silhouette Width", or sudden decreases, "Sum of Squares" which would be suggestive of this.

4.3 Consistent RT Behaviour

To our knowledge, no previous analysis of RT has collected and made-comparable data from this wide range of cell types, and as such this presents an opportunity to investigate the dynamic relationships between RT in different cells. The continuous RT data can be binarised into "Early" and "Late" regions by mapping the sign of the RT value. As RT is the log_2 ratio of "Early" divided by "Late" counts, a positive value is "Early" and negative is "Late". The "early consistency" (C_E) of each bin across the genome could then be calculated as the number of cell types that have an positive RT value in this bin. C_E counts are not evenly distributed across the database, with peaks of consistent "Early" or "Late" behaviour visible in Figure 4.7 A. Using these C_E values as a separating variable, we then investigated the distribution of genomic trends across these regions. CpG islands are a staple *cis*-sequence pattern that are known to be consistently maintained genome-wide, with a bias towards conservation near gene-coding regions and promoters. In Figure 4.7 B, it can be observed that there is a depletion of CpG islands between genes ("Inter") in regions that are consistently early (high C_E). Figure 4.7 C however reveals that this trend is not due to a reduction in genome GC content, as the number of high GC isochores ("H"-type) is higher in high C_E regions. Genes within the C_E regions have broadly similar characteristic composition, Figure 4.7 B "Genes", but demonstrate a gradual depletion of Introns, which may be related to the result in Figure 4.7 D that indicates that genes in high C_E regions are skewed to be much shorter.

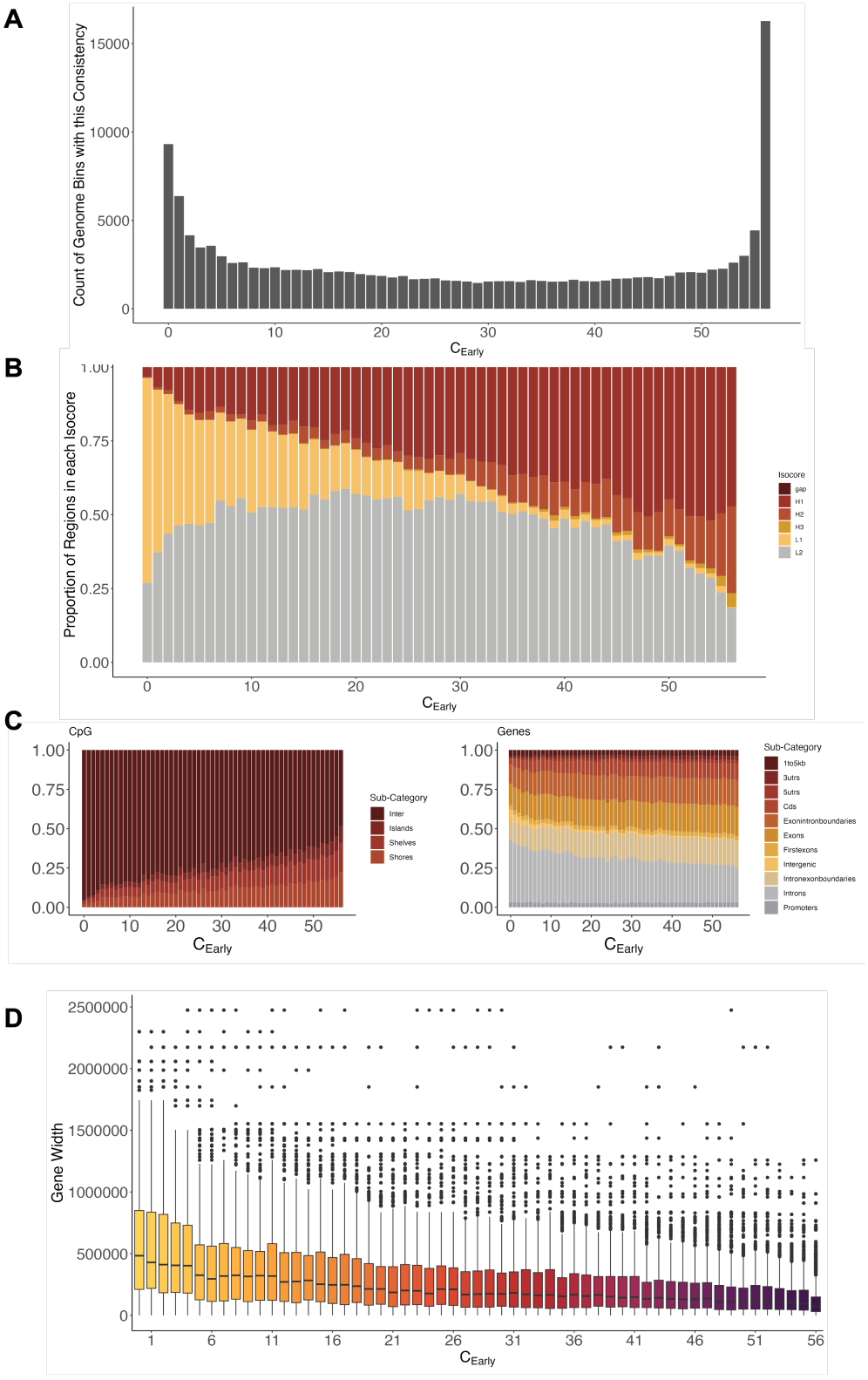

A
Count of Genome Bins with this Consistency
15000
10000
5000
0
0
10
20
30
40
50
CEarly
B
Proportion of Regions in each Isocore
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
50
CEarly
Isocore
gap
H1
H2
H3
L1
L2
C
CpG
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
50
CEarly
Sub-Category
Inter
Islands
Shelves
Shores
Genes
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
50
CEarly
Sub-Category
1to5kb
3utrs
5utrs
Cds
Exonintronboundaries
Exons
Firstexons
Intergenic
Intronexonboundaries
Introns
Promoters
D
Gene Width
2500000
2000000
1500000
1000000
500000
0
1
6
11
16
21
26
31
36
41
46
51
56
CEarly

Figure 4.7: RT consistency across available cell lines reveals sequence composition relationships. A) Binarised RT values (Early or Late) allows comparison of consistent behaviour across all 56 cell lines, revealing that many regions display consistently early ($C_E = 56$) or late ($C_E = 0$) behaviour. B) Isochore types are differentially related to RT, with much higher prevalence of GC-rich isochores (H) in regions that are consistently early. C) Sequence-annotation features show distinct sub-category patterns across regions with different RT consistency, with more CpG islands proximal to genes and genes containing lower intron content. D) Distribution of lengths of genes in each consistency type, reveals a bias towards short genes in consistently early regions.

5 Average-Behaviour Modelling

5.1 Modelling Dataset

We aim to create a model of RT that predicts the core sequence-driven behaviour that is independent of particular cell type deviations present in our dataset. As such, we need to select a subset of our data for training that displays the least variability across cell types and is thus representative of invariant behaviour. Bins that meet this criteria will contain data that is normally distributed across the cell types, with a standard deviation low enough to consistently report Early or Late behaviour across cell lines, hence is most reflective of core sequence effects. Visualising the relationship between RT mean and standard deviation in all bins, Figure 5.1 A, reveals that regions of extreme RT are in greater agreement across the datasets. Conversely, points with RT values around 0 displayed the highest degree of variation, which is likely driven by the relatively high rate of RT change in TTRs leading to broad sampling when aggregated across cell types. The threshold for the bin standard deviation was set to the mean of all bin standard deviations, 0.52, as this effectively removes all points with the highest deviation while still leaving points from across the full RT value ranges. As our points are distributed between ± 2, a standard deviation cutoff of 0.52 also ensures that the majority

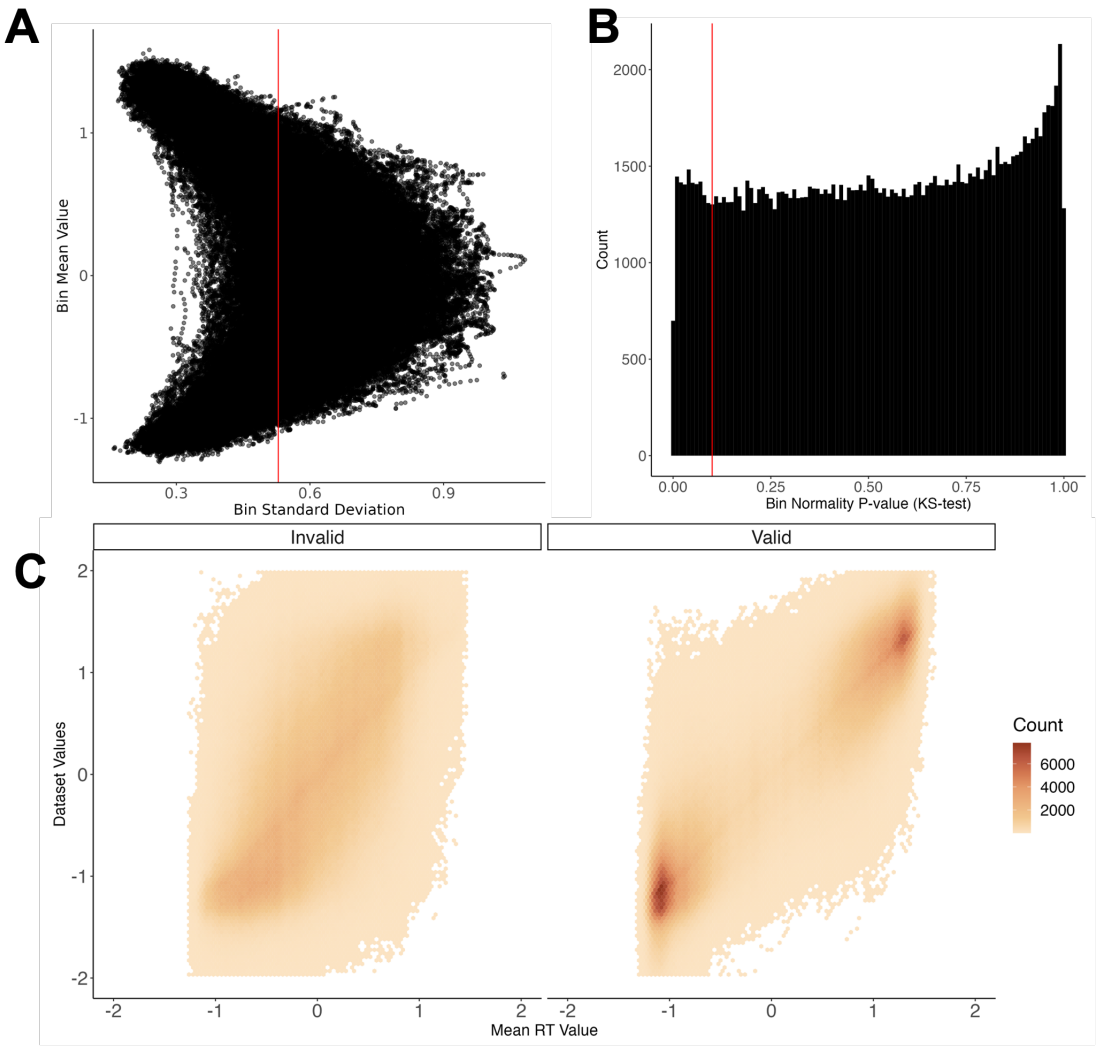


Figure 5.1: Visualisation and results of database filtering steps. The filtering thresholds imposed on bin standard deviation (A) and normality (B) provide the best opportunity for modelling the underlying sequence basis of RT. (C) visualises the change in the distribution of points on the "y" axis contributing the a final mean value in the dataset "x".

of bins that have their mean in an Early or Late region have fewer than 5% of the points on the opposite side of 0.

As an additional criteria to ensure that all points in the bin are representative of underlying behaviours, we evaluated the normality of each bin by carrying out a two-sided Kolmogorov-Smirnov test between the points in each bin and a normal distribution with mean/standard deviation matched to the data. The “null” hypothesis of this test is that values of each bin are drawn from the provided

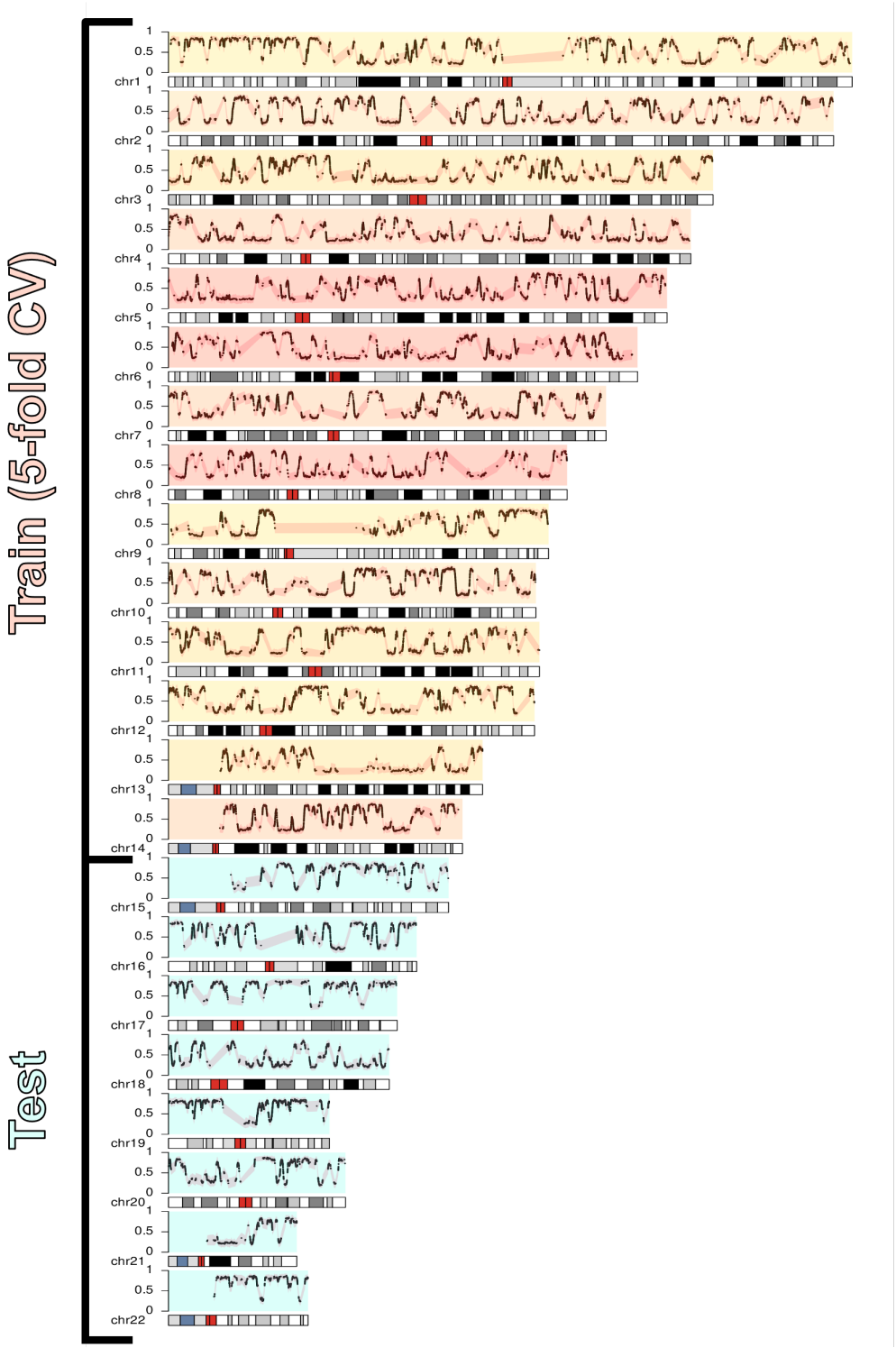


Figure 5.2: Genome-wide karyotype plot of the core sequence-driven RT values, highlighted by the test/train split, with each 5-fold validation fold within the training data highlighted in a different colour. All values are surrounded by a 1 deviation shaded border in red to demonstrate the local variability.

normal distribution, and a significant p-value would lead to the rejection of this null hypothesis. As such, a p-value > 0.05 is suggestive that the data is normally distributed. For increased stringency, we chose a higher normality p-value cutoff of 0.1 visualised in Figure 5.1 B, keeping all bins that had a p-value above this cutoff.

After the filtering process, 68,000 high-quality bins remained that we are confident have sequence-driven behaviour only and can be used for later machine-learning approaches. The effect of the filtering on the distribution of points contributing to final averages can be seen in Figure 5.1 C, where the distribution of values in the original dataset on the y-axis are plotted against the mean value they contributed into on the x-axis. Points labelled "Valid" were kept after the filtering process, and there is a clear improvement in the correlation between the two axes in the data that were kept after filtering. We believe that these two steps were most important to ensure we extracted the most indicative points of the sequence-driven RT component, and while it is possible that some valid data was discarded unnecessarily, carrying out the filtering in a bin-wise fashion ensured we did not accidentally sample from a skewed subset of the cell types. A TrackHub containing this sequence-driven "core" RT behaviours as a genome annotation can be viewed here.

For training the resulting models, we wanted to be confident that there is no contamination between sequences that are used in the validation and testing stages of modelling, as this would invalidate any measures of accuracy we use in the final evaluation. To ensure separation, we divided up the genome into Train and Testing sections by chromosome, as it would not be possible to accidentally overlap with a sequence from another chromosome. To carry out validation during the model selection and hyperparameter tuning phases, we also broke down the training chromosomes into 5 "folds", subsets of the original data of approximately equal size, which can be used in different combinations during training. Once again, to ensure there would be no accidental contamination of the train-validation loops we separated the folds on the chromosome level. A visualisation of the filtered RT data

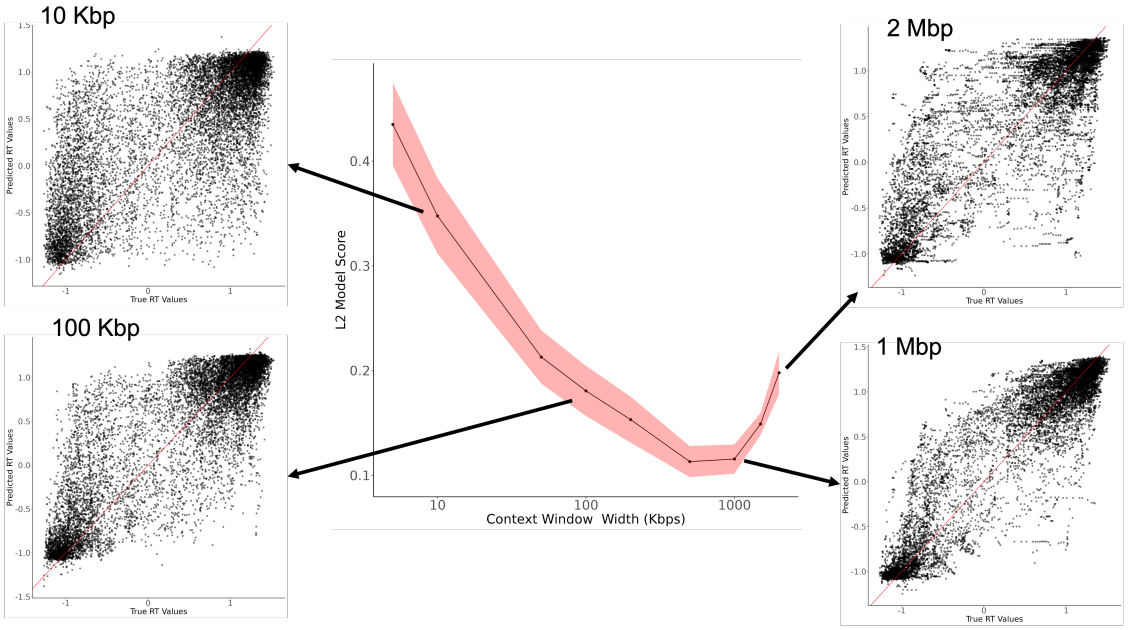


Figure 5.3: The effect of increased genome context on the performance of an LightGBM regressor trained on 3-mer counts

can be seen in Figure 5.2, with the final filtered data plotted against the chromosome it is from, each of which are coloured by their role in the model optimisation process.

For most of the modelling work in this thesis, we use the core sequence-driven value of RT as the target for prediction, resulting in a regression task that predicts a value between ±2 for each bin across the genome. However, with the context that RT has distinct replication "domains" that have consistent early/late replication timing, we can also reframe the prediction task as a classification of whether a region of the genome will be "Early"/"Late" using the a logical test of whether the core RT value is positive, producing a binary classification task. A separate inquiry was made into how classifiers can be used to predict this interpretation of RT, detailled in Appendix C.

5.2 Context Window

The selection of a 10kbp binning region for the RT data was motivated by previous research, such as the 50kb window used in Van Rechem et. al.[165], and an interest in not squashing extant RT dynamic behaviour. However, when choosing the size of

genome context to use for extracting features, there is a lot more flexibility, and a different class of constraints. Our separation by chromosome at both the test/train and validation level ensures that we can expand the context region arbitrarily and we will never accidentally cross a training region with a validation/testing one. Motivating factors for increasing the size of the context window were an understanding that local behaviour of RT is driven by chromatin structure, and is consistent within a single TAD. In addition, a larger sequence context provides more opportunities for infrequent motifs and larger patterns such as G-quadruplexes to occur, which increases the possible count-per-sample which gives the feature more dynamic influence as compared to simply counting if one is present or not at all. Conversely, a large sequence context comes with an alternate risk of over-smoothing, as the context sequence crosses over TAD boundaries or encounters the edge of an isochore region that might skew the feature counts in a way that is not predictive of RT. Finally, while it would be interesting to explore arbitrarily large context sequences, we are limited by the hardware available to us.

We conducted an investigation of the effect of modifying the DNA sequence context on the predictive power of an LightGBM classifier trained only on the counts of length 3 sequences (3-mers) in the context region. The model was initialised with the same parameters and seed, and were trained and evaluated across the 5-cross validation folds as standard, producing an average and standard deviation L2 (squared error) loss for each context size, shown in Figure 5.3. For this RT regression task, a decreased L2 score is an improvement in the model performance. As expected, a gradual increase of the context window size from 5kbp up to 500kbp produced an improved model, with the prediction scatter plots revealing a tightening of the predictions around the identity line. All points sitting on the red identity line would indicate perfect performance. As the context window exceeds 1Mb, we begin to see significant artifacts in the scatter plots of model performance and a corresponding worsening of the L2 score metric. In the CONCERT work, predicting RT from DNA sequence features and annotations, varying context widths were trialled to identify an optimal size for model performance and they found that

performance peaked when the context rose to 500kb and then plateaued up to 1Mb - which matches what have seen here[174]. While it is difficult to ascertain why this degradation takes place, it is clear that the distribution of the 3-mer features shifts significantly as the context window grows; beneficially at first and then somehow pathologically as the context window exceeds 1Mb. Possible explanations for this phenomena include the context windows exceeding the average size of a local cis-regulatory factor of RT such as GC content or number of G-quadruplexes.

As a tradeoff between feature extraction costs and model performance, we choose to use a context window of 100kb in creating our model. While this does not fully exploit the clear advantages of larger sequence contexts, we believe there is much that can be learned from interrogating the local distribution such as the influence of local GC content and the presence of existing regulatory sequences. Secondarily, as sequence length provides a clear advantage for prediction, we limit to range used here to allow for fair comparison with later more computationally demanding models which are unable to handle sequences that are significantly longer than 100kb.

5.3 Feature Extraction

As discussed previously, at the start of any modelling process, the data of interest must be translated into a format that can be easily imported to the modelling programs we intended to use. This process of converting the abstract DNA sequence into a suitable format is covered in the *"Sequence-based modelling"* background. To summarise the process, we must find a mapping function that takes in the DNA in a more abstract form and embeds it into a space whose dimensionality is governed by the number and type of features we extract. Crucially, for these "embedding" spaces to be useful they should try to satisfy the constraints that similar sequences are close to each other in the embedding, and sequences map uniquely into the space. Both of these constraints can be relaxed based on the final goal of the work, especially in cases where a specific subsequence pattern is being analysed and its presence/absence should have more impact on the space than

overall composition. There are arbitrarily large numbers of ways to do this, and significantly literature and tooling has been developed to capture the composition and structure of DNA sequence for modelling purposes.

RT is not a strand-specific DNA behaviour and interactions with DNA that are relevant to RT can start on either or both strands simultaneously. As such, any features that we extract for our modelling must be counted on the forward and combined with the count of the reverse-complement. Unlike in modelling of TF binding sites or enhancer prediction, we do not have a large number of *a priori* TF sequence motifs with known RT regulating properties from which to draw useful features from, as the binding of RIF1 and ASARs are not currently profiled down to a motif-driven binding event. Instead we approach the problem by first providing a general description of each DNA sequence in the form of large-scale metrics, and then drill down into more specific features that could be drivers of RT behaviour from the available database or the existing literature.

To provide a generalised baseline description of a sequence, we can look to broad-scale genome characteristics. We know from literature that the GC content of a genomic region is strongly related to RT behaviour, as gene dense regions are often replicated early and are generally GC rich. As such, we include the counts of the GC nucleotides and the GC-skew as features for each of these bins. Alone these GC measures provide a very sparse view of the complexity in each of the DNA sequence bins, so we also provide the counts of all length 3 sequences (3-mers) in each bin. Once the forward and reverse strand counts are combined, only the lexicographically-first k-mers are kept to avoid duplicated features, resulting in 32 features for the model.

As we have an RT value for every 10kb sequence, we reasoned it would be interesting to try and identify any motifs in a subset of these sequences that were unique to a particular RT range. We isolated the top 2000 10kbp bins with the highest RT values and submitted them to the XTREME web portal, part of the MEME suite[263]. We requested strongly enriched consensus motifs of up to 21bp in length, which resulted in 14 enriched motifs against a 3-mer shuffled background.

5. Average-Behaviour Modelling

As we knew from the context-window experiments that 3-mer content is highly predictive of RT performance that was considered a strong baseline to compare against. The resulting motifs are strongly enriched for contiguous sequences of "G" and "C" nucleotides, confirming our literature understanding of high GC content dependence, and we refer to these as the **RT Motifs**. As the previously detailed ERCEs[154] are the most prevalent *cis*-sequence regulatory elements of RT, we investigated possible motifs in their sequence in a similar way. All currently publicly available ERCE data has been profiled in mice, so we extracted the regions from the "mm10" reference genome and submitted them to the XTREME web portal as with the RT motifs. The resulting 19 consensus patterns, that we call **ERCE Motifs**, were significantly more diverse than the RT motifs. They were still dominated by high GC content, but some bucked the trend containing almost entirely "A" or oscillating patterns of two bases - pattern behaviours that are reminiscent of the discovery that activated replication origins often have poly(A/T) tails upstream of them[99]. Finally we extract the count of probably **G-quadruplex** locations in each of the bins using a pattern match from previous literature[217]. This is in line with existing literature that demonstrates a strong binding affinity between RIF1 and G-quadruplex regions, and a strong association with GC rich element (that might form G-Quadruplexes) and RT[151].

For the subsequent modelling, we concatenate all of the features described above: 3-mer counts, g-quadruplex count, GC content, GC skew, ERCE motif, and RT motif counts. There was no additional feature selection or processing (such as dimensionality reduction) during the modelling process. We believe that this fits best practice as we did not incorporate any features which could have a known cell type specific bias, did not manually or automatically exclude any class of features we investigated as that process could bias the training of the model, and only used training sequences in the *de novo* motif discovery process to avoid contamination. We visualise the distribution of each of the final feature types in Figure 5.4.

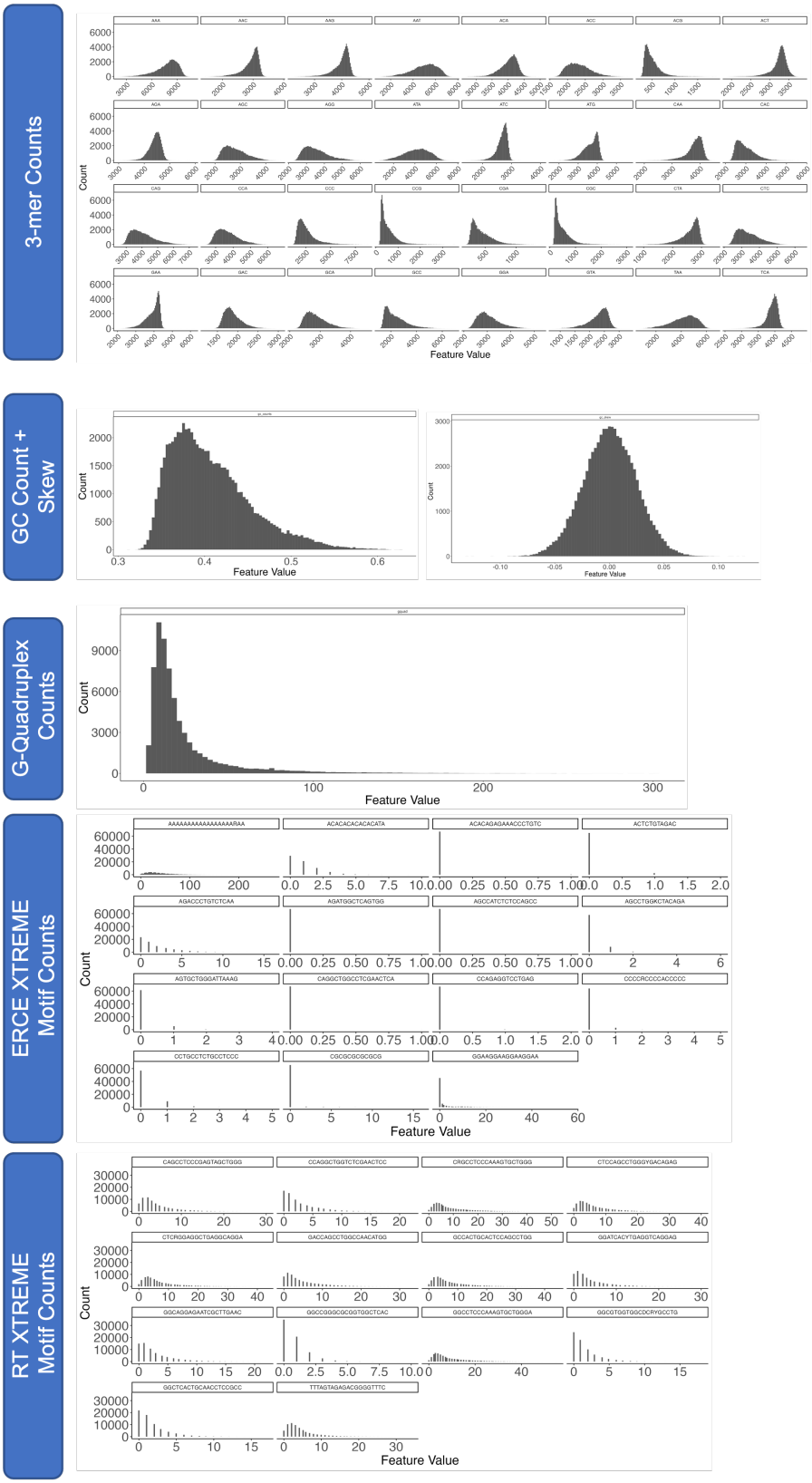


Figure 5.4: Distribution of sequence-based features extracted from the filtered bins

5.4 Comparing Model-Type Performance

As we have established our core feature set for predicting RT, we need to choose what model will perform best on the available data. The Python machine learning suite `scikit-learn` provides implementations of, or interoperability with, many ML algorithms that can be applied naïvely to a prediction task like this.

For this model comparison process I chose a combination of linear, Bayesian, dictionary, and tree-based models which covered a broad spectrum of available model types and hopefully form a representative sample of the available modelling space. Many of these models will only converge (or will perform significantly better) if the input features are centred and scaled. These models have a lot of adaptable hyper-parameters, which allow fine-grained control over the training and tradeoffs that are model during the "learning" process. To ensure as fair a comparison as possible, all tree-based models (DecisionTree, XGBoost, and LightGBM) were set to the same number of trees and learning rate. All other models were left at default parameters or tuned until they successfully converged when provided with the data.

The result of training each of the 8 model types to convergence are displayed in Figure 5.5, and highlights a few key inter-relations between the model types. As "Lasso" and "SGD" models are fundamentally linear models with different optimisation processes, the performance between them is very similar. The "BayesianRidge" model captures a similar range of values as the linear models, but is unable to constrain the predictions around the extreme Early and Late values in the way XGBoost and LightGBM are capable. MLP presents an interesting option as its internal layer of neurons can be made increasingly wide, improving its modelling performance at the cost of runtime and generalisation. Finally, the KNN performance was impressive given the relative simplicity of the model's dictionary-look-up method of learning. While it performed strongly on average across the full range, it was unable to correctly classify a subpopulation of samples with true Late behaviour.

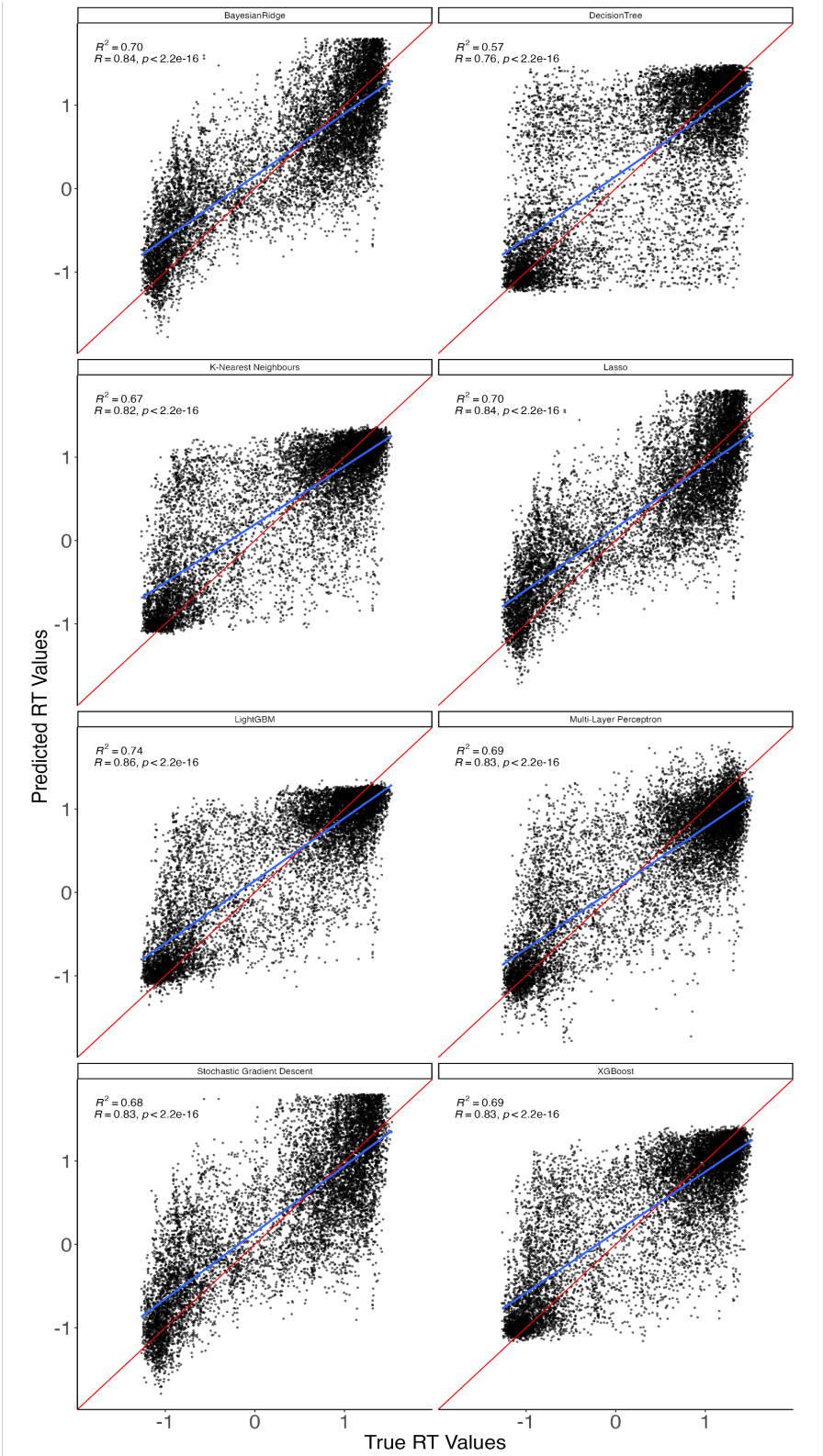


Figure 5.5: Comparison of different model class performance on RT prediction task. Red line in each axis along the diagonal corresponds to perfect prediction performance, and the blue lines represent a linear fit between the true and predicted RT values for each model. The R^2 for the linear fit, and the Pearson R is shown below.

As it achieved the strongest performance of 0.74 R^2, is directly interpretable through its tree-based structure/importance measures, and had the fastest runtime second only to the KNN, we decided to proceed with the LightGBM model.

5.5 Optimising Final Model

After selection, the LightGBM model architecture was subjected to a Bayesian optimisation hyperparameter search with 5-fold cross validation as detailed above. The search was carried out using the "Sweeps" functionality of the "Weights and Biases" logging toolkit, and 300 parameter combinations were trialled. The search was designed to minimise the mean validation-set root-mean-squared-error (RMSE) for each set of parameters, and would select subsequent parameters from a uniform distribution across each of the following ranges of hyperparameter values.

$$\begin{aligned} NumLeaves &: 25 - 100 \\ Learningrate &: 0.0025 - 0.1 \\ NumBoostRounds &: 500 - 5000 \\ BaggingFraction &: 0.1 - 1.0 \\ FeatureFraction &: 0.1 - 1.0 \end{aligned} \tag{5.1}$$

These values were chosen based on a combination of literature background and practical experience with the LightGBM classifier. The most powerful hyper-parameters to influence the nature of the model training are the `number of boosting rounds` and the `learning rate`, as their values influence each other in a reciprocal way. A low `learning rate` will result in each additional `boosting round` contributing a small amount to improving the performance of the model, implicitly slowing the pace of over-fitting but requiring more boosting rounds to reach high performance. Conversely, if one was to use a very high learning rate the model would quickly over-fit, resulting in a model that generalises poorly to unseen samples. The `number of leaves` in each tree also afford a tradeoff between detailed fitting to the training data and the risk of over-fitting, and can in theory

be set very high at the cost of significant increase in computational requirement to both train and carry out inference. Finally the `bagging` and `feature` fractions introduce additional randomness to the sampling of feature and other trees to use in the boosting process, allowing exploration over combinations of features avoids the model over-fitting to the behaviour of informative but limited subset of the features. After the sweep, an optimised parameter set was found and can be seen in (5.2).

$$\begin{aligned} NumLeaves &= 100 \\ Learningrate &= 0.003 \\ NumBoostRounds &= 4800 \\ BaggingFraction &= 0.8 \\ FeatureFraction &= 0.3 \end{aligned} \tag{5.2}$$

When trained on the full training data and tested on the held out test set, visualised in Figure 5.6 A, these parameters improved LightGBMs performance to an $R^2 = 0.76$, out-performing the baseline models we trained. Additionally, the model's absolute error rate revels that the model was consistently predicting values very close to to the true sequence-driven RT values, with 72% of the absolute errors in the "test" predictions lying within one SD of the original bin RT value, shown in Figure 5.6 B. Collectively, this pattern of prediction reveals that the model is able to perform consistently and strongly across the full range of sequence-driven RT values, and has captured a key link between the DNA sequence and the dynamics of human RT.

Finally, in order to provide a useful reference set of data for future work, we trained a pair of models on separate halves of the genome-wide RT dataset, using the optimised parameters above. To balance the amount of data given to each model, the first was trained on chromosomes 1-8 and the second on all data from 9-22. We then used these models to predict an RT value for the full human genome, binned at 10kb, ensuring that each model was used to predict on the chromosomes it didn't see during training. The TrackHub containing this genome-wide prediction of sequence-driven "core" RT can be viewed here.

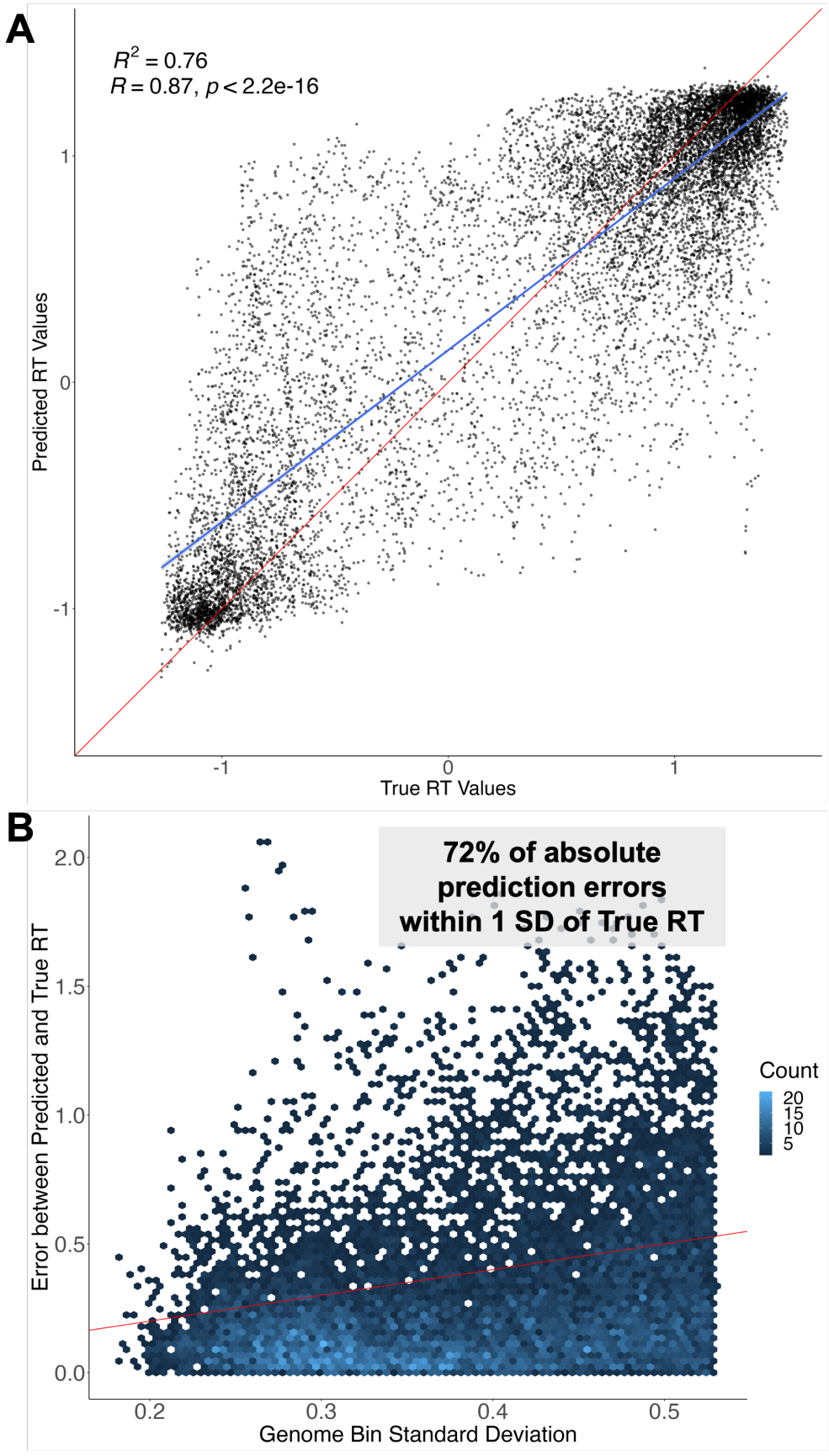

A
$R^2 = 0.76$
$R = 0.87$, $p < 2.2e\text{-}16$
Predicted RT Values
True RT Values
1
0
-1
-1
0
1
B
72% of absolute prediction errors within 1 SD of True RT
Error between Predicted and True RT
Genome Bin Standard Deviation
Count
20
15
10
5
2.0
1.5
1.0
0.5
0.0
0.2
0.3
0.4
0.5

Figure 5.6: (A) Performance of the tuned LightGBM model on the unseen "test" dataset. The red line shows the pattern of theoretically perfect performance, and the blue line shows a linear fit between true and predicted RT. The Pearson R and R^2 coefficients for the fit are shown in the top left. (B) Visualisation of the distribution of absolute error in the models predictions on the "test" set compare to the SD of the RT values in the genome bin being predicted on. In 72% of all test bins, the absolute error of the prediction lay within 1 SD, indicating strong predictive performance.

6

One-Hot Encoded Deep Modelling

For most ML applications, there is a significant pressure on the model designer to find a set of suitable features for the model to use to make predictions about unseen data. There are many ways to automate the removal or combination of features to improve model performance, but most of these are coarse heuristics that are unlikely to result in novel discoveries. After creating a model that is able to competently predict RT from sequence based features, we intended to produce a method that could capture a similar level of performance without requiring the domain knowledge or tools (such as XTREME). Ideally we want to utilise methods that make the modelling process do more of the work, while also ensuring that we are not missing key relationships between the sequence and the target behaviour. In short, we want to generalise the inference of genome scale behaviours by making the model identify and optimise the DNA sequence features that best match the target of interest. This can be achieved by operating directly on simple direct encodings that maintains the positional structure of the DNA sequence, so the final optimised features can be interpreted afterwards.

After the literature search, and examination of previous modelling efforts, the Voss/one-hot encoding was deemed to be the most applicable here, visualised in Figure 4.7 A. However, while we do not explore them in this work we do not

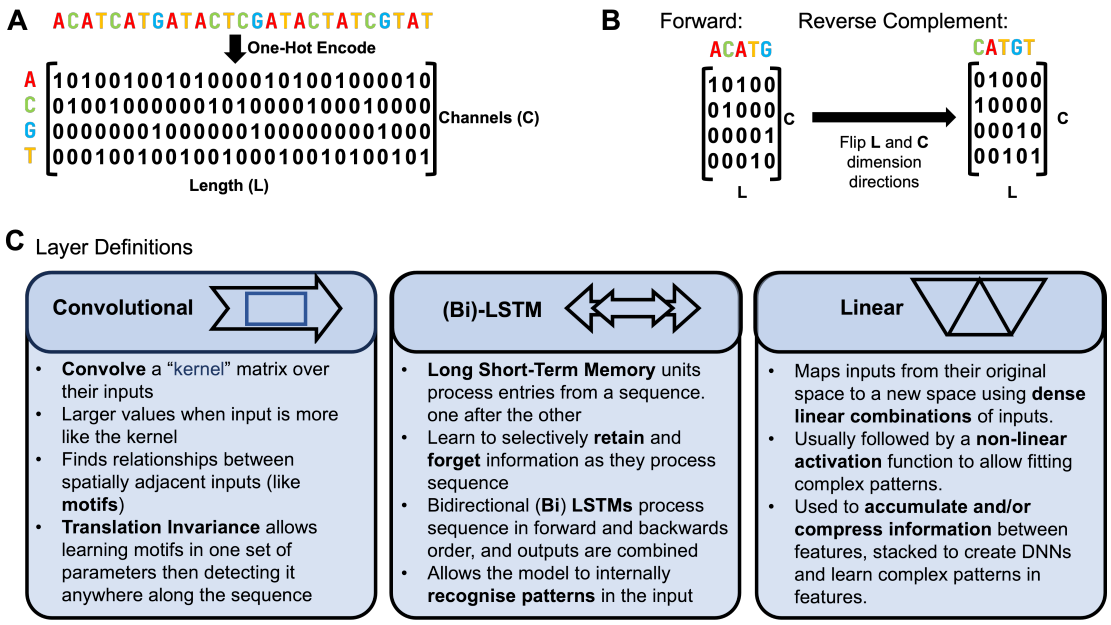


Figure 6.1: Visualisation of encoding (A) and reverse complementing (B) a DNA sequence in the One-Hot Encoding scheme. Additionally, the component parts of subsequent models are laid out in (C), detailling the primary benefits of convolutional (CNN) layers as motif learners, Bi-LSTM layers for identifying patterns of motifs, and linear layers as a fundamental layer for accumulating and compressing information.

discount the possibility that denser vector encodings such as "dna2vec" could be very powerful for modelling RT behaviours. In particular, using them in conjunction or as part of a large model pre-trained on the reference genome sequence presents a promising approach for accelerating the prediction of genome tasks, most recently seen in the impressive performance of the HyenaDNA model of the Genomics Benchmark tasks[245]. However, this excitement must be tempered by first looking for ways to quantitatively measure the influence of pre-training on all sequences/other genome tracks on the possibility of over-fitting and memorisation that does not reflect true generalisation. There are serious questions about how to verify the model's performance and generality on "unseen" DNA sequences if there is no guarantee that the model has not seen them before.

In this chapter, we can explore the capability of deep learning model to participate in "supervised novel discovery" by starting from a hand-crafted deep learning model that encodes our known understanding of what works well to predict RT,

then gradually freeing the model to find new helpful features that can improve the predictive performance.

6.1 Overall Modelling Decisions

We use the PyTorch Python library to create all of the models described in this section and, unless otherwise stated, all layers are setup with the default initialisation scheme[264]. The outer training loops and optimisation process was handled through the PyTorch lightning wrapper framework, allowed easy integration with the logging and hyper-parameter optimisation platform "Weights and Biases"[265,266]. All models presented here are trained on a single A6000 GPU with 40GB RAM. The optimisation process for deep learning models is significantly more unstable than the relatively refined methods for optimising linear regressors and tree-based methods. For most of the architectures testing in this and later chapters the random seed, which governs the initialisation of parameters and the shuffling of the batches of data has a significant effect on the training process of each of the model. This variability in training behaviour indicates that the model is likely capturing patterns in the data in a different way, however regardless of this the final predictive performance are stable to changes in the random seed, and performance is consistent as long as the model converges with the given hyper-parameters. This in many ways frustrates the process of architecture choices and final model selection, as it increases the computational load to confidently differentiate the performance change between updating the architecture and running a different random seed. However, it highlights a key and often under-discussed issue of reproducibility in deep "learning" methods. Where computationally feasible, all finalised architectures are trained with multiple random seeds, and the resulting mean/sd of training metrics are reported. For evaluation of the unseen test partition of the dataset, the model that achieved the best validation score is used.

Even before selecting the primary modelling components, a deep "learning" optimisation process is influenced by key hyper-parameters - the most powerful

of which are often the "batch size" and "learning rate". Batch size dictates the number of samples from the training set for which gradients are calculated at each optimisation step. While there are few proven rules on the effect of batch size on the behaviour of optimisation, the heuristics that have persisted in the deep learning community suggest that a large batch size can be considered to "stabilise" training, providing a conservative estimate of the best direction to "step" at each iteration, however smaller batch sizes allow for more fine-grained navigation of the loss space that may avoid the trap of the "safest" path leading into a local minima. There are also suggestions that keeping batch sizes equal to a power of 2 allows for more efficient processing, and while this is at best framework/machine specific we use it as there is no reason not to. For the experiments in this work, we found that using the largest possible batch size that was a power of 2 that could fit within GPU memory provided the most stable results between random initialisations.

The "learning rate" is another powerful hyper-parameter lever that is also hard to automatically tune, and is intimately linked with the model architecture and optimiser chosen. We use the "Adam" optimiser throughout this work as provided by PyTorch, which has been documented to provide strong across-the-board performance and does not require significant tuning beyond a suitable learning rate[267]. In the presence of significant noise between random seeds, we found minimal performance improvements for our models when tuning the learning rate to more than the best order of magnitude. For all experiments unless otherwise stated, the learning rate of the model was set to 3×10^{-4}.

Finally, it is often the case that NN models are over-parameterised for the task they are required to solve, resulting in fast over-fitting on the training data. Regularisation strategies are employed in most DNNs to explicitly punish models using more weights than necessary, or learning too heavily on a subset of the weights in a way that reduces generalisation. For all models in this section, an L2 regularisation penalty on every parameter is added during the model training to encourage weight "decay" if not being used for prediction. To combat pathological sub-modularisation a "dropout" layer is added after every Linear unit, which

randomly masks weights to zero during training to force the model to distribute its computation. From these descriptions, it is clear that while these two regularisation techniques share a common implementation goal, improve generalisation, they act on the weight space in opposition to one another, hence carefully balancing their influence is vital for training stability.

6.1.1 RT Dataset

The dataset used for all training in this chapter uses the same ranges, targets, and validation/test splits that are used in the machine learning modelling in the previous chapter. As the computational cost of loading and encoding many 100,000 bp DNA sequences is significant, we employ the fast memory-mapped "2bit" format to quickly load genome spans and reimplement the one-hot encoding scheme from the Enformer PyTorch implementation for minimal overhead[268]. Multiple efforts were made to find more efficient loading schemes, including pre-loading and encoding all sequences in RAM, but this counter-intuitively produced slower loading. It is possible that this occurred because this method could not benefit from fast cache-hits when the "2bit" loading function was called with the same arguments across multiple epochs. However, it was beyond the technical scope of this work to investigate this in greater detail, and we welcome further investigation into the fastest way to load and encode DNA sequences from file as it was demonstrably one of the main bottlenecks in the training process.

6.1.2 The Rationale for Convolutional Layers

In all of the models that interface directly with one-hot encoded sequences, we use a series of sequential **1D convolutional neural network layers** (1D CNN layers). The 1D convolutions are each described by a stack of kernel matrices, or "weights", and a vector of offsets, "biases" that are added to the output of the convolution. While a full description of convolution is beyond the scope of the thesis, it is useful to develop an intuitive analogue for the process of a convolution over a given input. Convolution in this context can be visualised as sliding one matrix (the kernel) along

another (the input) sideways and performing a multiplication between all elements that overlap at each position in the slide. This "slideways" movement is referred to as "convolving" the kernel over the input. The "stride" of a convolution describes the number of positions that are stepped sideways between each multiplication.

In PyTorch the kernel weight matrix for a 1D convolution is a 3D tensor with shape (N_{out}, N_{in}, $kernelwidth$), where N_{in} represents the first dimension of the input, which is 4 in the case of a one-hot encoded DNA sequence as there are 4 bases. N_{out} is the number of output channels we want to create in this convolution, and can be thought of as the number of individuals kernels we are convolving over the input. Finally, $kernelwidth$ is the size of each of the N_{out} kernels that are convolved over the input. As PyTorch tensors cannot be "ragged" and much have values for the full scope of all their dimensions, we can see that each convolution layer must scan N_{out} (N_{in} x $kernelwidth$) kernels across its input.

We can see that in this formulation there is no explicit mention of the length of the input sequence to the layer, and as the kernel slides over the full length it performs the same function regardless of its position. Thus, if the same pattern occurs in two positions in the sequence the kernel output will be the same when it slides over both. This disregard for the position in the sequence is called "translation invariance" and, when optimised during the "learning" process, allows the extraction of general patterns that work in a position-independent way. Crucially, in applications to DNA sequences, this allows the kernels to be optimised to represent all or part of a meaningful DNA motifs, which can be summarily extracted from the model after training through a range of analyses.

Without deliberate intervention, the output of the convolution process will result in a shorter tensor than the input, as you cannot calculate a value when the kernel overlaps with the "nothing" off the edge of a tensor. This mild compression of the input is an expected and sometimes useful side effect of convolution. This compression would be drastically increased if the convolutions "stride" is increased beyond the default of 1, scaling the output at a rate of $1/stride$. However there are situations where the output is required to be the same size as the input, and this

is achieved by "padding" the input tensor with additional values either by adding neutral values (often 0) or by mirroring/extending the tensor beyond its original bounds if that is a valid transformation for the application. In our case, it would be invalid to wrap the DNA around on itself or repeat the end bases to pad the input, so where padding is required, a neutral column of four 0's is appended to each end. When convolutions are used in this work the default behaviour will be to not pad the input, and any padding will be explicitly described.

6.1.3 Reverse-complement Equivariance

When a property or behaviour of the genome acts independently of which strand of a DNA region it interacts with, it is considered to be reverse-complement (RC) invariant. As RT is one of these properties, we need our model to produce identical outputs for an input sequence and its RC - displaying RC equivariance. As outlined in Chapter 3, creating a model that is RC equivariant has been tackled in multiple ways and there are architectural choices to encourage or directly guarentee that the model will produce identical responses[269]. The work of Zhou and Shrikumar et. al. 2020 outline a unified view of architectural choices that can guarentee RC equivariance in predictions called "Conjoined-RevComp Wrapper" (CJRCWrapped), and are focussed primarily on generating single base-pair resolution predictions of genome behaviour from DNA sequences for both the forward and reverse strand[202]. Their unified view provides proofs of how RC parameter sharing (RCPS) methods can efficiently guarentee RC-equivariant outputs for strand-specific predictors, and demonstrate how existing NN layers such as CNNs can be adapted to do this. It is notable that introducing RC-equivariant layers destabilised training in bp-resolution predictions in the paper, and the CJRC compliant methods were eventually surpassed in predictive power by models trained with data augmented with RC versions of the inputs on longer training - as of writing this disparity is still an active area of investigation.

For use in this work, we implement the RCPS model architecture in PyTorch, taking advantage of how the forward and RC representation of a one-hot encoded

sequence can be attained by simply `flip()`-ing a weight matrix along its "length" and "channel" axis if the order of the channel axis results in a complementary swap, Figure 6.1. In this work, we order the channels "ACGT" to allow this and align with other work[242]. We choose to use the RCPS model class instead of a joined "siamese" architecture also described in Zhou and Shrikumar as RCPS has no risk of unnecessarily learning both the forward and RC version of a motif, and we are interested in being efficient with our parameters.

6.1.4 Model Performance Evaluation

As with all machine "learning" optimisations, a target metric is required to measure the performance of the model and produce a loss value that can be back-propagated to provide gradients each of the trainable parameters. We chose the "mean squared error" (MSE) loss function (implemented in PyTorch) as the performance measure, and trained use the original split of genome regions described for the modelling in Chapter 5. As full 5-fold cross validation introduced too large a computational overhead, we separated out the first "fold" subset from the previous training as a validation set and used subset "fold" 2-5 of as the training set. At the end of each epoch, the performance of the model was measured on fold 1 as the validation set, which is reported to the user but not used in training. While this is sub-optimal validation compared to a full 5-fold cross-validation approach, it allowed for a significantly faster development cycle and allowed us to identify the models' sensitivity to batch size.

Out of the box, a PyTorch model can be trained indefinitely, and is very likely to over-fit if left unattended. We employed the commonly used "Early Stopping" criteria to halt likely over-fitting training runs, conditioned on the MSE loss of the validation fold. If the validation performance did not improve for 5 epochs, the training was terminated and a "checkpoint" of the model with the best validation performance is saved to disk for testing.

To test the model's performance on unseen data, the "test" fraction of the dataset is run through the model checkpoint and it's predictions visualised against the

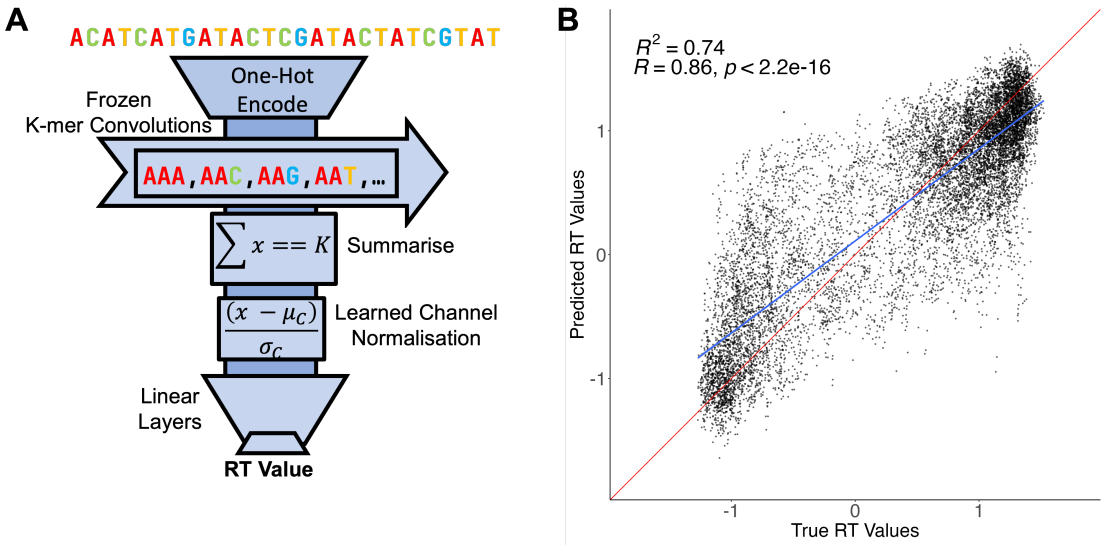


Figure 6.2: Schematic representation and performance of the "Handcrafted" model with frozen 3-mers are extractable features. The performance of the model on unseen "test" data is shown in (B), demonstrating the strong correlation of the predictions with the ground truth averages. The red line represents perfect prediction performance, and the blue line is a linear fit between true and predicted RT values. The Pearson R and R^2 coefficients for the fit are shown in the top left.

true dataset values as a qualitative measure of model performance. Quantitatively, we measure the MSE, RMSE, MAE, Pearson R and Spearman R coefficient to compare against the original machine learning models.

6.2 Hand-Crafted "Frozen" Feature Deep Learning Model

With these modelling considerations in mind, we first set out to show that deep "learning" methods could be used with our existing biological knowledge to create a predictor for RT from the one-hot encoded sequences. To this end, we create a model that uses a single convolutional layer initialised and frozen to use the all 3-mer sequences. "Frozen" weights are not changed during training, and are commonly used to make "backbone" models that are pre-trained on one task then applied to help with another. In this case, it could be said that the weights have been "pre-trained" by my understanding of how useful 3-mers are in predicting biological RT.

Subsequently, the channel activations of the frozen 3-mer convolutions are tested whether they are equal to 3 (the k-mer length) as this would indicate the presence of the 3-mer. These counts are then subsequently added within each output channel, introducing a global pooling operation along the full context sequence length. This multi-step process is analogous to pre-counting the 3-mers and passing them to the next layer un-normalised. We identified in the original machine "learning" model explorations that centering and scaling the input sequence pattern counts produced a significant performance boost for the non-gradient-boosting tree methods. We can emulate this process in the neural network and make it more adaptive to features that are not known *a priori* by introducing "centering" and "scaling" learnable parameters for each output channel of the first convolution. These will allow the model to shift the data distribution into a range that it is more capable of working with. Notably, we could have given the model these center/scaling parameters for the training set in advance, but this would not scale well to learnable parameters.

After this adaptive distribution-shift, the outputs are passed to a two layer Linear NN, referred to as a multi-layer perceptron (MLP) with rectified linear unit (ReLU) non-linear activations after the hidden layer. The size of this hidden layer can be tuned to improve model performance. The final layer outputs a single number without an activation function, which is the model's RT value estimation. A full visualisation of this model's processing can be seen in Figure 6.2 A. Models generated from this specification and trained with large batches of 256 samples are able to perform confidently on the unseen "test" data, exceeding the performance of the MLP trained with the same number of neurons in the previous average behaviour modelling comparison, increasing the correlation coefficient from 0.69 to 0.73, visualised in Figure 6.2 B. This increase in performance is an improvement in generality across the board, and demonstrates that learnable normalisation and data-modelling during training are a viable alternative to pre-generated summary statistic that may unintentionally hardcode bias.

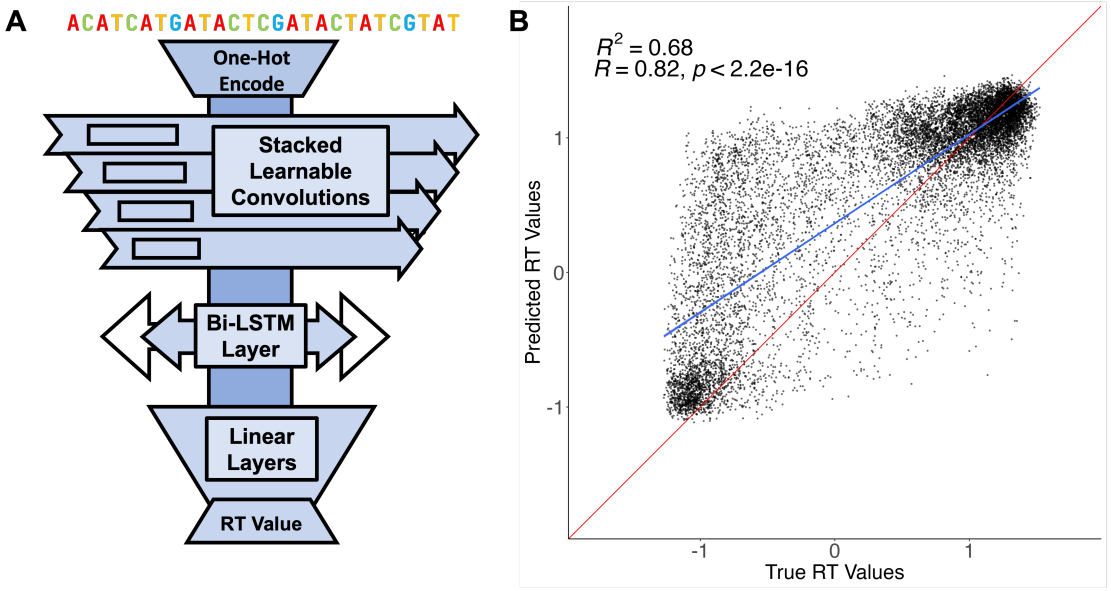


Figure 6.3: Schematic representation and performance of a fully automatic modelling setup given a random initialisation and engrained reverse complement equivariance by RCPS convolutions, (A). The performance of the model on unseen "test" data is shown in (B), with quantitatively weaker performances than models that were initialised with "knowledge" of 3-mer content but qualitatively stronger performance around high RT values. The Pearson R and R^2 coefficients for the fit are shown in the top left.

6.3 Fully Automatic Modelling

Building off this success, we can now address the shortcomings of our previous LightGBM models and our "handcrafted" model that used feature counting, a method of global pooling. While this is demonstrably very powerful for predicting the average RT profile, we know that cis-regulation of genome behaviour is often driven by the relative position of cis-regulatory elements. For example, the presence of enhancer regions around a gene that can be activated in certain cell types to modulate gene transcription. The capture of regulatory dynamics *ab initio* in a deep learning model trained on DNA sequence is currently a very active area of research. Approaches outlined in Chapter 3 such as ChromBPnet, Puffin, and Borzoi are beginning to identify patterns of TF motifs ands some long range interactions that align with enhancer association[204,227,270]. These models are trained to predict genome profiles such as RNA-seq, ChIP-seq and ATAC-seq profiles which are intimately related to the regulation of genes and the local 3D structure of the

genome, providing a strong guiding line for the model to infer regulatory syntax.

In our modelling task, we are starting with a significant paucity of relevant information by comparison. RT is causally linked to gene regulation through RIF1 inhibition, and to 3D structure through its close relationship with chromatin openness. However, it is the work of bioinformaticians comparing and combining different experimental datasets that has revealed these relationships, and few if any of these behaviours have been shown to have a simple motif-combinatorial behaviour. In addition, as we are predicting the behaviour of core-sequence RT profiles across different cell types, it is likely that the influence of any cell type specific drivers of the behaviour (such as specific promoter/enhancer regions) have been lost in the quest for core-sequence modelling. The closest potential target we have for cell-generic RT patterns are the recently discovered ERCE regions in mice, which are large (20kb on average) spans of DNA which contain a bounty of possibly regulatory binding sites. It is worth noting as well that, while we share much of our replication machinery with mice, these ERCEs have not been identified in human cells. Consequently, before more powerful modelling begins I must set expectations. The chance of us finding the regulatory syntax of RT from the core-sequence RT data is slim.

However, if there are drivers within the DNA sequence that extend beyond the overall composition of the context, then we will need to observe the patterns within sub-context chunk. The CONCERT architecture addressed this by first extracting and compressing counts from shorter regions (~5kb) and using DL approaches to combine the information from these bins into strong predictions of RT for a single cell type[174]. In addition, they trialled an end-to-end sequence approach called CONCERT-Hierarchical, which used CNNs to optimise features during training and then pass them down to a similar aggregation mechanism as in the base CONCERT. We aim to carry out the same process to predict core-sequence RT. A key takeaway from the CONCERT paper was that this hierarchical model underperformed the model with pre-generated feature distributions, which should temper our expectations for the performance of this model - learning both the features and solving the prediction outcome is a daunting optimisation problem.

We have described in detail the rationale of using a single layer of convolutions to extract known or learned sequence features from the DNA input and then pooling the results from the full input length, a "global" pooling strategy. However, by stacking layers of convolutions, activations, and smaller-scale pooling operations, it is possible to learn a model architecture which extracts and accumulates local features then passes those on to subsequent layers that in turn extract/accumulate more abstracted features. Iterating this process along many layers (increasing the models "depth") results in a hierarchy of features being learned, and simultaneously compresses the sequence information into a shorter sequence with a number of channels determined by the kernels used in the convolution. Combining this approach with the RCPS techniques produces a parameter efficient and expressive series of convolutions that can combine/accumulate information locally and along the full sequence.

While convolutions are powerful at collecting the local motifs and structure of the DNA, it requires many layers for this influence to extend along a wide range. One possible solution is to increase the "dilation" of the kernel which has had success in other DL models in genomics such as the Enformer work[242], but requires many CNN layers resulting in a deep and computationally expensive model. We chose to instead leverage a different type of NN layer altogether, a "long short-term memory" (LSTM) unit, which is a variant on the recurrent neural network (RNN) area. The LSTM operates by moving along the sequence, much like a convolution, but instead of completely independent calculations at each position it maintains an internal state or "memory" as it moves. By learning a selective information retention/loss program during training, it can accumulate information along the sequence an build up an awareness of long-range patterns in the underlying sequence. Crucially, unlike other forms of RNN, LSTM networks are much less prone to a phenomenon called the "vanishing gradient problem", where the influence of positions further up the sequence is lost as the RNN head "moves". In the context of DNA sequences, these learned patterns could be a co-localisation or spacing of DNA motifs, or a gradual shift in base pair composition over an isochore boundary. By creating a bidirectional LSTM (Bi-LSTM) we can move an LSTM "head" in both directions

along the convolutionally encoded sequence and concatenate the states from each direction to get the output of the layer. Finally, after the LSTM layers we can accumulate all the emitted hidden states for each position and pass them through an MLP (as in the handcrafted model) to produce a single RT value prediction. The full model architecture can be sene in Figure 6.3 A.

6.3.1 Comparing Performance

This model also performs well on the unseen "test" data after training, visualised in 6.3 B, and is able to easily capture the main dependencies between RT and DNA sequence despite starting from a completely random initialisation. Qualitatively, we can see in the scatter plots that the model has learned not to over-predict early regions much better than the "handcrafted" model, with the points staying closer to the identity line in early regions.

One could rightly ask; why is it that these DNNs that trained for longer are underperforming the LightGBMs? It is difficult to say in general, but it many cases, where the input for a problem can be transformed in a series of separate and independent features (referred to as "Tabular" problems as they are represented by a table of samples against features), there is previous research that suggests that in general gradient boosting methods will be better at modelling the underlying data structure[271]. Analysis carried out on a benchmark of non-synthetic datasets compared the performance of a range of boosting-tree methods, recurrent neural networks and transformers to compare their overall performance. The results indicated that the tree-based methods were much more robust to uninformative features, and DNNs tended to produce over-smoothed results that did not match target distributions as accurately[272]. In addition, in datasets where features are not normally distributed (i.e. heavily skewed or categorical) the partitioning-based logic of boosting trees was more robust than the continuous internal structure of DNNs[273]. An analysis of correlations between "meta-features", features that describe the features of available datasets, and the relative performance of different model

classes also revealed the gradient boosting methods tended to perform more strongly than DNNs on larger datasets that have a high ratio of samples to features[273].

The datasets generated in this work to model RT from sequence features meets many of the criteria that would suggest improved performance from gradient boosting methods over deep neural networks. In particular, the high ratio of samples/features and the irregular distribution of values in the input features. The 3-mer counts have a very wide and non-normal distribution, and are presented at the same time as the counts of very rare k-mers, some of which consistently occur < 10 times per sequence, resulting in a Poisson-like distribution. If used correctly and in a discrete fashion, this can provide a lot of information to the model, however this is not easily captured by linear DNN layers. Additionally, it is clear that the 3-mer composition describes a great deal of the variance exhibited by RT values and, as such, providing those upfront saves a huge amount of processing time.

7

In-Silico discovery with Sequence Models

7.1 Feature Attribution in LightGBM Model

By construction, as a gradient boosting tree derivative model, LightGBM models are composed of a series of comparisons between features, which result in their branching structure. This allows for unique interpretability options once a model has been trained, as the number and type of interactions that each input feature took part in reveals a dimension of its contribution towards the final prediction. As these models are constructed in part driven by a random seed, such as the subset of features available due to the `feature fraction` parameter, a model trained with a different set of starting conditions will follow the training slightly differently, resulting in different amounts of use for each feature. For models that fit our parameter specification after optimisation, with over 4000 rounds of boosting, there are great many possibilities for the behaviour of the tree to branch a different way.

While exploring even a modest fraction these possible paths is computationally infeasible, we can leverage the fact that our training data has been partitioned into folds, over which these parameter configurations performed the best. It follows that we can train models using our existing 5-fold cross validation split and be reasonably confident that the models final performance will be similar to the best performing model. As such, if we train those 5 models and investigate the patterns within their

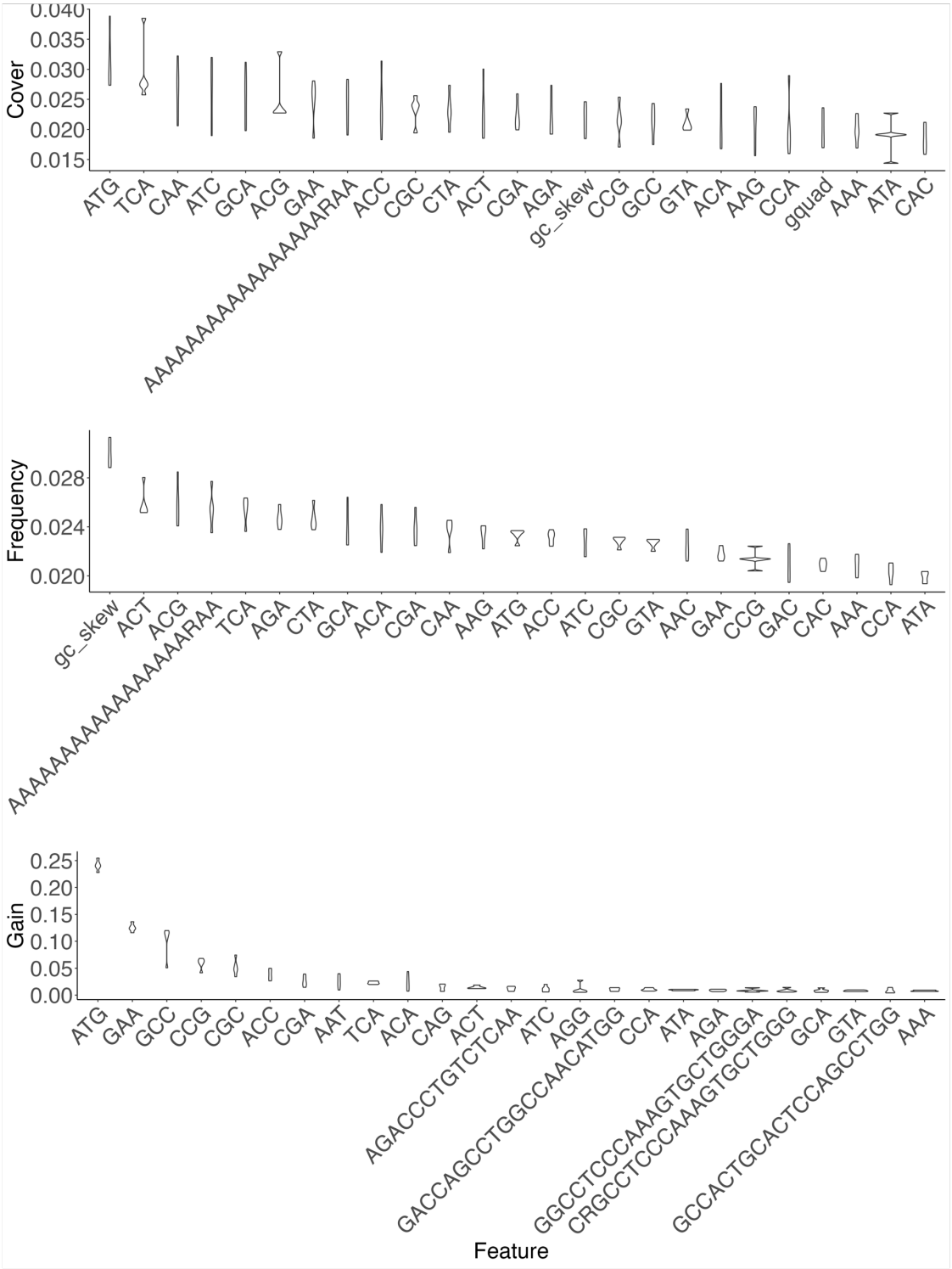


Figure 7.1: Top 25 most important features for the final LightGBM model from each importance category; Gain, Cover, and Frequency.

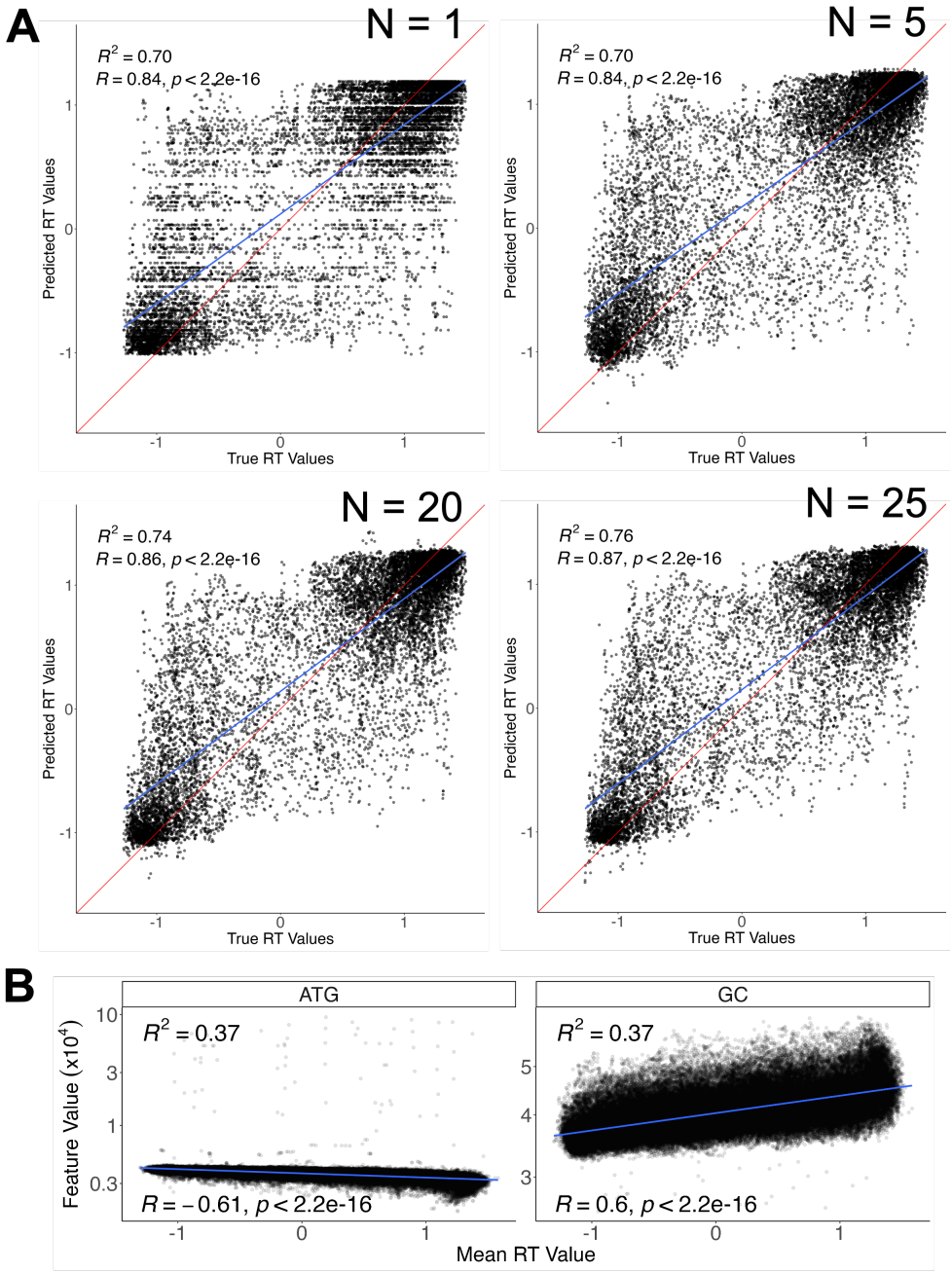


Figure 7.2: (A) Visualisation of model performance on "test" dataset restricting the feature space to the top N features by Gain. The red line shows the pattern of theoretically perfect performance, and the blue line shows a linear fit between true and predicted RT. The Pearson R and R^2 coefficients for the fit are shown in the top left.. (B) Scatter-plot indicating the relationship between the highest "Gain" feature "ATG" and RT value, alongside the previously best understood sequence predictor of RT, the "GC" content. The blue line is a linear fit between the RT values and feature values, with the R^2 coefficient for the fit shown in the top left.

assigned feature importance it should be possible to ascertain which of the features are most generally useful and which are only used occasionally with limited effect.

Visualised in Figure 7.1, we can see that the vast majority of the predictive power (Gain) comes from the a small number of 3-mer - primarily the "ATG"/"CAT" triad. These are supported by the next 4 key features, which are primarily the counts of all permutations of the exclusively G and C containing motifs - strongly reaffirming the role of GC in creating a strong predictor. Notably, while these do approve the qualitative appearance of the distribution in Figure 7.2 A 'N=5', removing significant banding artifacts where the "ATG" counts were the same for regions with widely varying RT, the correlation score does not improve reminding us of the fragility of individual metrics for assessing model performance. Furthermore, we note that it is not until the inclusion of some non-3-mer features, shown in the N=20 sub-figure, that a significant jump in the correlation is made from the N=1 "ATG" model. This is supporting evidence to suggest that there may be more complicated DNA motifs and patterns that influence the RT distribution. Conversely, we note that composite calculated such as "gc_count", "gquad" (G-quadruplex count), and "gc_skew" did not provide significant Gain, despite "gc_skew" being the more frequently used feature from the dataset. Additional features, such as many of the 20-mers from the RT XTREME runs, are not featured in the top 25 in either category and this is likely a product of their sparsity in the dataset and the genome as as whole. Based on this information, we select subsets of the top "Gain" features, and observe the performance of a model trained with only these in Figure 7.2 A. As already noted, the performance improvement from adding features beyond "ATG" is not realised until at least 10 features are added, which implies the majority of the variance in the dataset is covered by the "ATG" feature. This was a surprise, as based on previous work and the strong relationship between RT and GC-content we expected "gc_count" or an "SSS" motif (any combination of 3 G and C bases) to be the most predictive. However, due to its role as start codon and it's documented depletion in 5' UTRs, it is most likely that the strong predictive power of "ATG" arises from it's relationship with genes.

7.2 Profiling Poorly Predicted Regions

When assessed on the held out "test" regions, we see that the performance in Figure 5.6 is strong for the majority of points but is not without outliers. For example, there is a subset of points that have "Late" core sequence-based RT values which the models consistently predict as "Early", and vice-versa. In this section, we investigate what behaviours are different in these sequences that might explain why the model mis-predicts them more than others, and whether this can be used as a way of exploring what biological patterns the models has learned to use to predict RT.

To do this, we labelled all of the bins in the test dataset by their true and predicted behaviour using "Late" if the true/predicted value is less than 0, and "Early" if that values is greater than 0. We then annotated these regions with a set of genome content and sequence composition behaviours, the first of which was the distribution of CpG Islands and gene components in each of the four region types, shown in Figure 7.3. This demonstrated that the correctly predicted regions often have easily identifiable differences such as a large difference in CpG Island "Inter" category and more introns in "Late" regions. However, the mis-predicted regions don't display any trends with the same magnitude, with the main identifiable pattern between them being the preference for the model to predict late when the "Inter" CpG category is proportionally higher. We are aware from literature that there is a strong relationship between GC content/gene density and RT, but the patterns displayed here are insufficiently detailed to isolate the reason why these regions are being mis-predicted in the model.

Secondly, we investigated the distribution of common genomic repeat elements, visualised in Figure 7.4. Sub-figures B and C are organised to show the distribution of different features in the same layout as we see on the model performance scatter-plots. Thus that regions that are correctly predicted are shown in the facets on the rising diagonal, akin to the identity line in the scatter-plot, and mis-predicted regions are on the falling diagonal. These sub-figures highlight a shift in increased SINE abundance in regions that the model predicts are "Early", which that is

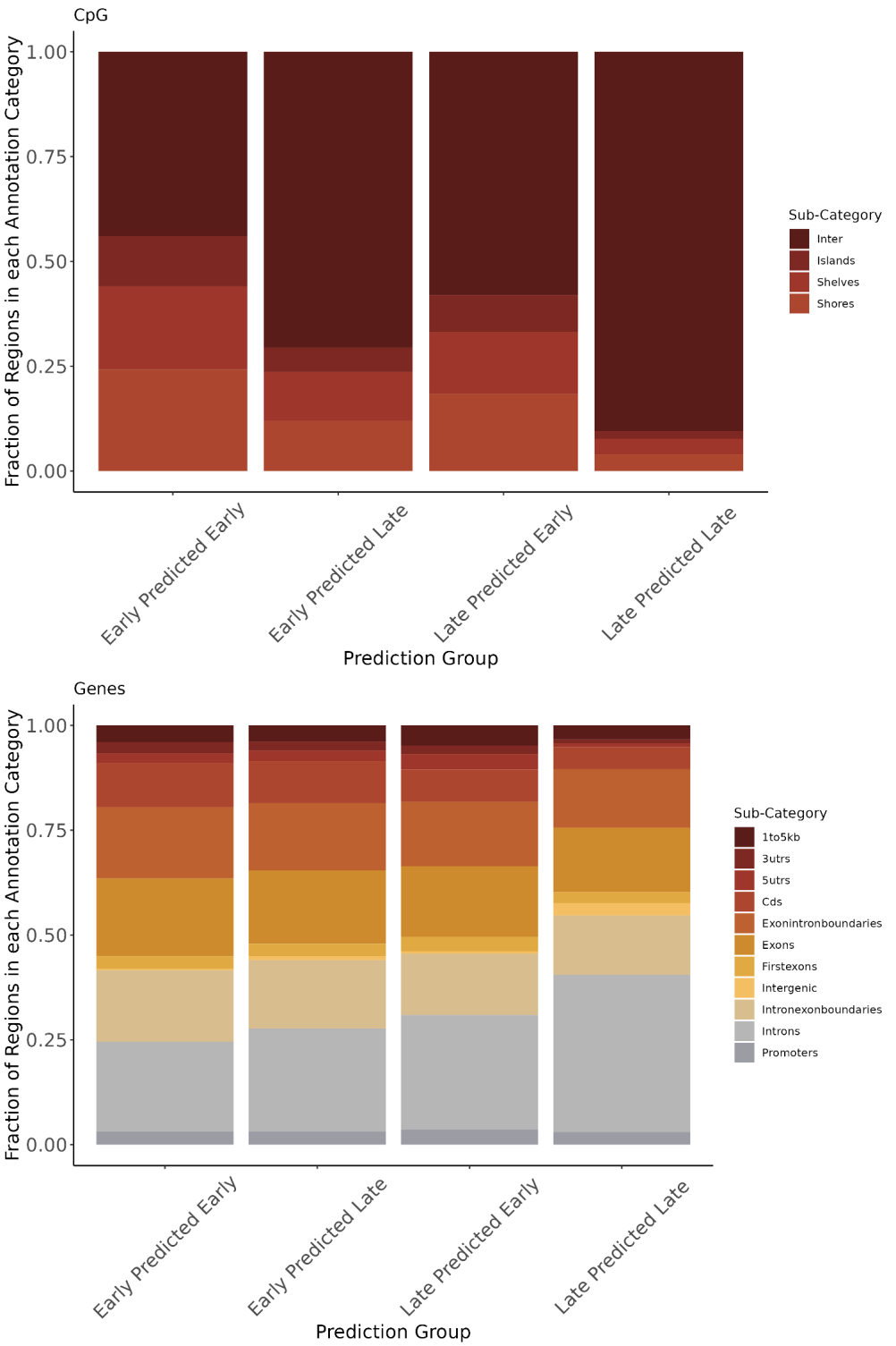

CpG
Fraction of Regions in each Annotation Category
1.00
0.75
0.50
0.25
0.00
Early Predicted Early
Early Predicted Late
Late Predicted Early
Late Predicted Late
Prediction Group
Sub-Category
Inter
Islands
Shelves
Shores
Genes
Fraction of Regions in each Annotation Category
1.00
0.75
0.50
0.25
0.00
Early Predicted Early
Early Predicted Late
Late Predicted Early
Late Predicted Late
Prediction Group
Sub-Category
1to5kb
3utrs
5utrs
Cds
Exonintronboundaries
Exons
Firstexons
Intergenic
Intronexonboundaries
Introns
Promoters

Figure 7.3: The distribution of CpG island and gene sequence-region subcategories for the 4 different genome bin classifications - where the classifications are defined by the combination of true and predicted RT behaviour. "Early" indicates an RT value > 0 and "Late" implies a value < 0. Clear patterns of different behaviour can be seen for the correctly predicted "Early" and "Late" regions, with the "Late" regions being dominated by "Inter" CpG island regions and containing proportionally more intronic sequences than the "Early" regions. Conversely, the two mis-predicted classifications demonstrate very similar behaviour in both families of features.

directly related to the global trend in SINE/LINE abundance in "Early"/"Late" regions, demonstrated by Figure 7.4 A. We observe that in all the regions where the RT value has been mis-predicted (falling diagonal of facets in Figure 7.4 B), the distribution of LINE/SINE elements matches those of the category they have been mistaken for, strongly suggesting that the model has identified a sequence pattern as part of its prediction process that correlates with the SINE density, and is making it's prediction based on the general relationship between SINE/LINE and RT demonstrated in Figure 7.4 A.

As the LINE distribution is stable between categories, we further investigated the distribution of SINE subfamilies, visualised in Figure 7.4 C, and identify *Alu*s as the primary varying subfamily that is driving this shift in SINE distribution. *Alu*s are retrotransposons rich in CpG sites, providing a possible link to information about genome methylation, which occupy about 11% of the human genome[55]. *Alu*s have a strong consensus template, resulting in sequences roughly 300bp in length with two preserved motifs of length 11 and 9 constituting an internal RNA polymerase III promoter separated by A-rich regions[55]. Neither of these tails contain the "ATG" motif which provides such strong predictive power in our LightGBM model, so we must assume that identification of these or related regions must be carried out in the model through a combination of the feature set provided. None of the feature set we identified with *de novo* motif discovery on "Early" RT and murine ERCE regions identified either of the preserved motif sequences exactly. Features from these sets with a >50% match to the preserved sequences did not appear in the top 20 features ranked by importance, suggesting that this was not utilised by the model

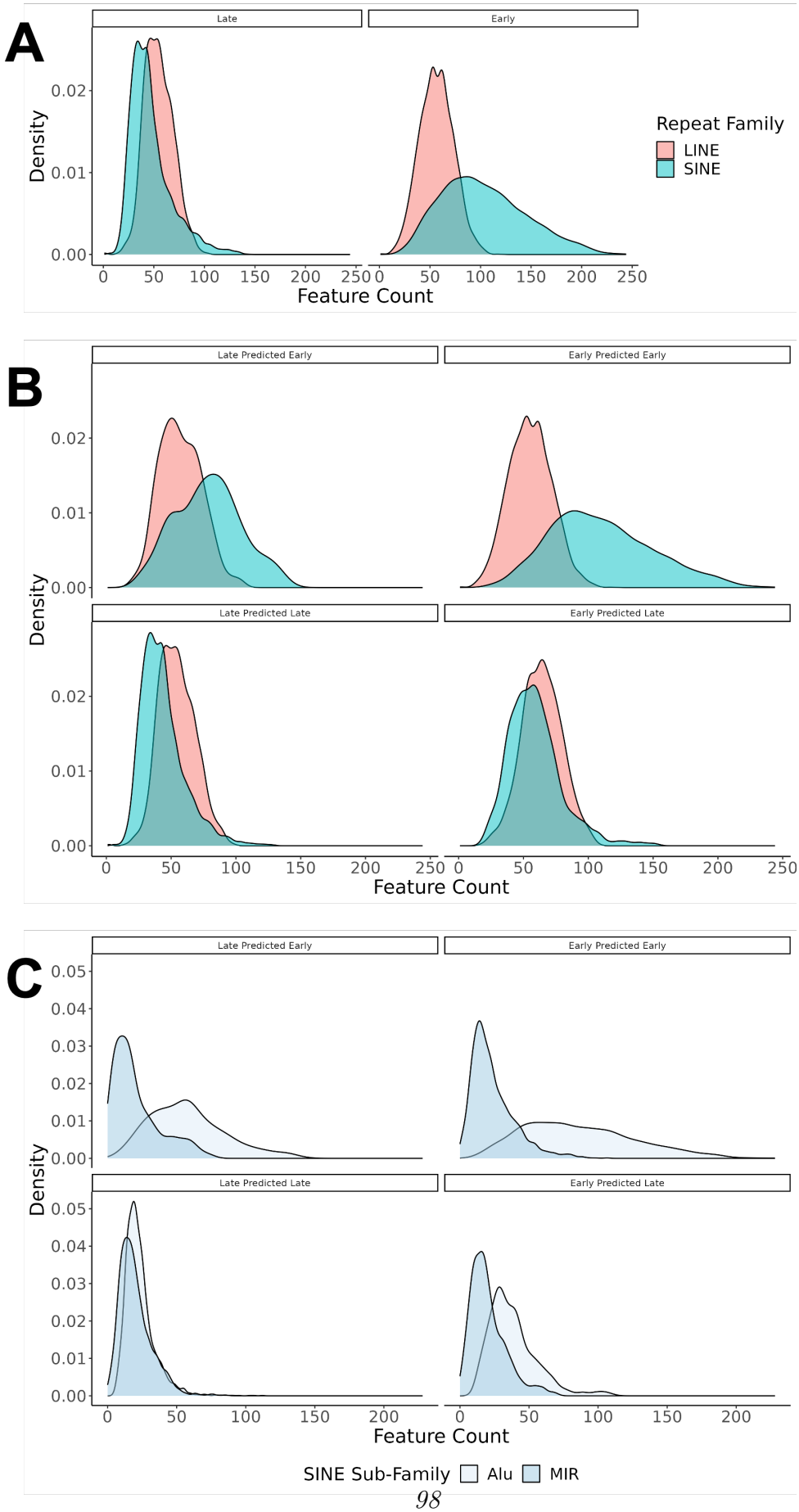

A
Late
Early
Density
0.02
0.01
0.00
0
50
100
150
200
250
Feature Count
Repeat Family
LINE
SINE
B
Late Predicted Early
Early Predicted Early
Late Predicted Late
Early Predicted Late
Density
Feature Count
C
Late Predicted Early
Early Predicted Early
Late Predicted Late
Early Predicted Late
Density
0.05
0.04
0.03
0.02
0.01
0.00
Feature Count
SINE Sub-Family
Alu
MIR

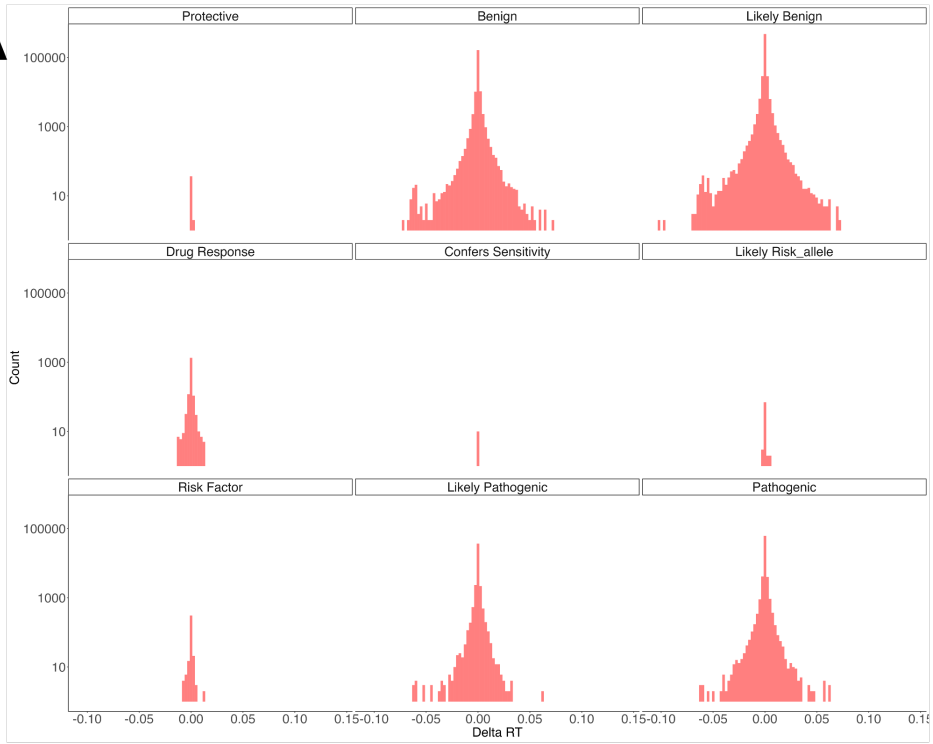
Figure 7.4: (A) Count density of LINE/SINE elements in regions that have positive ("Early") and negative ("Late") core sequence-driven RT values. (B) Count density of LINE/SINE elements in all genome bins in the test dataset, faceted by the combination of predicted and true "Early"/"Late" RT values. (C) Visualisation of a subset of SINE sub-families, with the same facets as (B). In sub-figures (B) and (C), looking along the rows of facets shows that sequences that are predicted "Early" have a shared and distinct distribution of SINEs from those that are predicted "Late", in contrast to the true distribution of SINEs for all "Early" and "Late" regions shown in the facet columns. Within the SINE family, we identify *Alu*s as the subfamily that displays the most variation between the predicted "Early" and "Late" RT regions.

to make it's predictions. Preliminary investigations were undertaken to retrain the LightGBM model on feature sets derived from a genome with these repeat elements deliberately masked out to observe the impact on model performance. We observed a marginal drop in the model's correlation coefficient (0.76 down to 0.74), and "ATG" was usurped by "GCC" as the most importance feature for model predictions, suggesting that the presence of repeats played a role in "ATG"s usefulness to the model. Future work will investigate whether deliberately detecting *Alu* regions in the genome can provide a strong predictor for RT, and how consensus sequences of the same complexity as *Alu*s might be automatically detected and included as features in subsequent models.

7.3 Model Applications

Now that we have an optimised and strongly generalising predictor of core-sequence RT across cell lines, it would be useful to demonstrate how this can be used as a lens onto the behaviour of the genome under modification, and derive novel insights into the behaviour of the human genome. As the model has optimised to predict the core-sequence RT extracted from consistent regions across many cell lines, it will not be possible to make any assertions about individual cell line behaviour. In addition, as each target value during training represented a point with a standard deviation of ~0.3, it will not be possible to expect accurate predictions of the exact core-sequence RT value for any region of the genome. As such, we must look to what

A



B

Source	GO Term	Term Name	Negative Delta P-Value	Positive Delta P-Value
GO BP	GO:0007417	central nervous system development	6.20E-07	
GO BP	GO:0048731	system development	1.00E-04	2.40E-02
GO BP	GO:0048856	anatomical structure development	1.40E-03	4.20E-04
GO BP	GO:0007399	nervous system development	4.50E-03	
GO BP	GO:0048513	animal organ development	4.50E-03	8.90E-06
GO BP	GO:0007275	multicellular organism development	8.80E-03	4.40E-02
GO BP	GO:0032502	developmental process	1.30E-02	5.40E-03
GO:CC	GO:0030286	dynein complex	1.40E-03	
GO:CC	GO:0099081	supramolecular polymer	1.70E-03	6.70E-05
GO:CC	GO:0034702	monoatomic ion channel complex	3.90E-03	
GO:CC	GO:0099512	supramolecular fibre	6.30E-03	3.70E-04
GO:CC	GO:0005856	cytoskeleton	6.70E-03	9.00E-03
GO:CC	GO:0097014	ciliary plasm	2.80E-02	2.70E-02
GO:MF	GO:0008569	minus-end-directed microtubule motor activity	1.20E-03	8.40E-03
GO:MF	GO:0051959	dynein light intermediate chain binding	4.90E-03	2.30E-02
GO:MF	GO:0005524	ATP binding	1.30E-02	
GO:MF	GO:0008331	high voltage-gated calcium channel activity	1.30E-02	
GO:MF	GO:0003774	cytoskeletal motor activity	1.80E-02	4.20E-04
GO:BP	GO:0009887	animal organ morphogenesis		2.90E-03
GO:BP	GO:0009653	anatomical structure morphogenesis		1.90E-02
GO:BP	GO:0072359	circulatory system development		2.70E-02
GO:BP	GO:0000226	microtubule cytoskeleton organization		2.90E-02
GO:BP	GO:0007010	cytoskeleton organization		3.20E-02
GO:CC	GO:0099080	supramolecular complex		2.40E-03
GO:CC	GO:0005858	axonemal dynein complex		1.20E-02
GO:CC	GO:0005737	cytoplasm		4.70E-02
GO:MF	GO:0003779	actin binding		1.80E-02
REAC	REAC:R-HSA-380259	Loss of NIP from mitotic centrosomes		4.60E-02
REAC	REAC:R-HSA-380284	Loss of proteins from interphase microtubule organization		4.60E-02

Figure 7.5: (A) Distribution of change in RT values (delta-RT) for each clinical classification of ClinVar SNV. (B) Panel of upregulated gene ontology (GO) terms related to the genes contained (or were adjacent to) the SNVs with the highest delta-RT values.

can be learnt from changes in the model's performance, bearing in mind that the model's feature has no explicit encoding of their original position in the sequence.

As such, to begin with we investigated the impact of modifications on the sequence on the predicted average RT value of the model, on the basis that if a modification produced an exceptionally large deviation from the original prediction we have at least identified a region of the genome that due to its composition is more variable/sensitive by the model's estimation. Given the position non-specific nature of the model's features, our baseline assumption would be that wider modifications to the sequence should produce larger changes to the resulting RT prediction.

Our first investigation looked at the impact of ClinVar single nucleotide variations (SNVs) across the full human genome. ClinVar collects data from many human genotype-phenotype pairs, and annotates their effect with supporting evidence[274]. The change in RT (delta RT) for each SNV is shown in Figure 7.5 grouped by their clinical assessment. The patterns in these delta RT distributions seem to suggest that large effects are possible in both benign- and pathogenic-associated SNVs, but the sampling rates for different conditions are very variable suggesting that other categories may be capable of producing large delta-RTs if there were more of them in the database. Taking the SNV positions that had the most extreme negative and positive delta-RT values and associating them with their overlapping or adjacent genes reveals a diverse web of ontological associations, in 7.5 B. While some genes are found in both categories, the regions that have an increase in the RT values after the SNV, associated with a cluster of gene pathways related to structural development in cell, visible at the bottom of the table. While a direct causal relationship between a shift in RT and genome behaviour has not been proven, there is evidence to suggest that a shift towards early RT encourages more permissive transcription as early regions are associated with more-open chromatin. Thus these positive RT shifts could

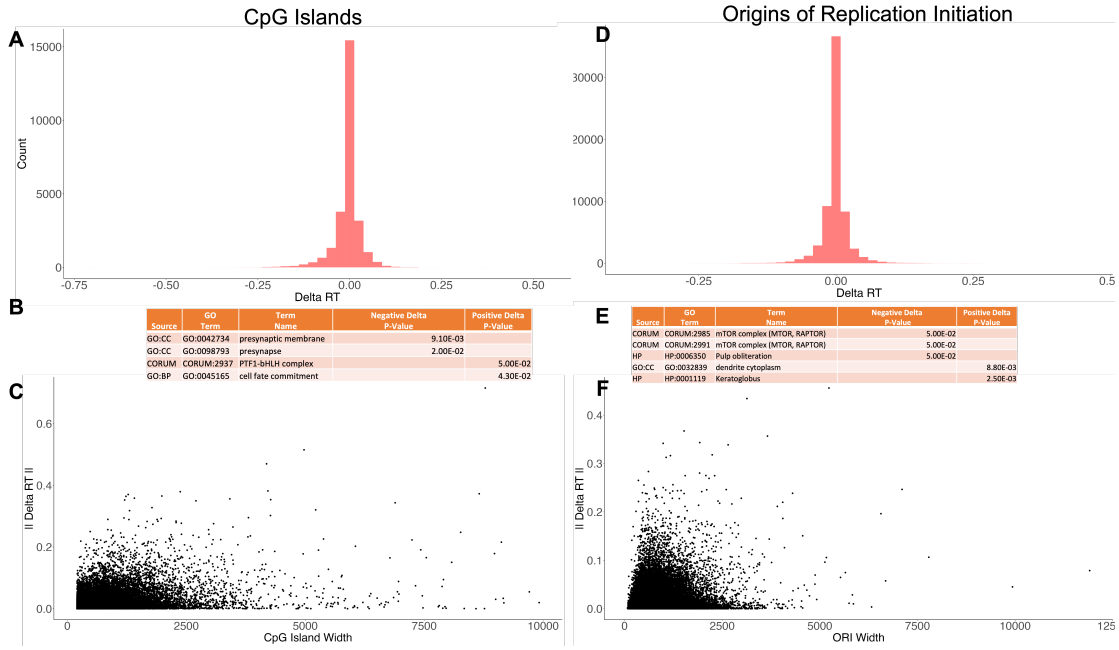


Figure 7.6: The impact of CpG Island and Origin of Replication Initiation (ORI) sites deletion on the predictions from our RT model. (A) and (D) show the direct delta-RT outputs when all annotated CpG/ORI sites in the human genomes are deleted. (B) and (E) show the most strongly related GO networks to the deleted regions with the largest changes both positive and negative, after deletion. (C) and (F) demonstrates the possibility of a relationship between the length of the sequence and magnitude of the absolute delta RT predicted by the model. However, given the low number of long CpG islands and ORIs, it is not possible to draw strong conclusions about a general trend without adjusting the sampling of different width features.

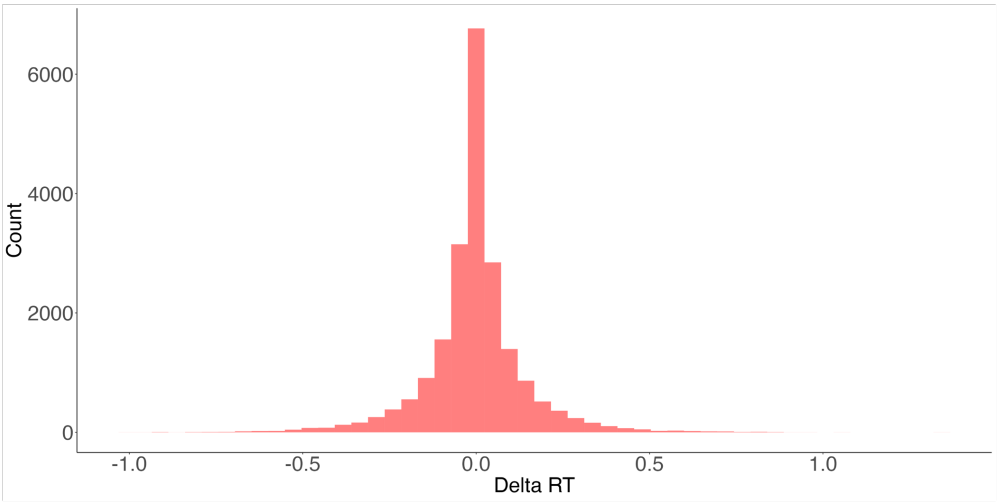
indicate a gene expression change, reinforcing and encouraging cell proliferation. However, it is also worth noting that due to their location in relatively gene-sparse sections of the genome, the "adjacency" condition we used for associating genes to SNVs might be over-representing developmental genes, and de-biasing this method against this will be important for future downstream applications of this model.

To investigate the effect of other genomic sequence features, and thus discern the impact of that sequences presence, we isolated regions that have high CpG density (CpG islands) and regions that have been linked to origins of replication (ORIs) across the full hg38 genome. Both of these region types have a well-documented link to RT, and should therefore produce large shifts in the RT value when they are removed. Intersecting regions between these two sets of genome regions covered 20% of the CpG islands and 8% of the ORIs, indicating that while there is similarity

between these sets, the majority of the sequences are unique and can be explored separately. In Figure 7.6 A and D, the range of delta-RT values for CpG islands and ORIs are shown respectively, revealing that both groups removal are capable of producing delta-RT effects routinely greater than 0.2. CpGs produce occasional larger shifts, in absolute delta-RT units, but this is not due to either the CpG or ORI region deletion being wider on average. Figure 7.6 C and F demonstrate that the ORIs are on average shorter than the CpG islands that are being removed. In both classes of deletion, it appears that a wider region is not a direct indicator of the absolute delta-RT response after deletion, with the effect saturating after the deletion reaches 2500bps long. It is likely that this pattern is the result of undersampling the true relationship between these sequences length and their effect on RT. Intuitively, it follows that the removal of regions with high GC content such as CpG islands would produce a slightly stronger effect on RT value than with a more complicated sequence identity. However, the overall similarity in distribution between these two classes of deletion suggest that the combinations of simple k-mers used in the model are sufficient to discern the impact of regions that are semantically closely-related to RT such as ORIs. Finally, investigating the CpG Islands and ORIs that produced the strongest change in RT revealed very few strong GO associations, but those present are shown in Figure 7.6 B and E.

Finally, due to the recorded significance of gene activation and clustering in the literature surrounding RT regulation we set out to investigate whether there are genes whose sequence intrinsically confers a particular RT value on itself in opposition to the RT value of the surrounding area. We explored this by undertaking a similar deletion experiment as above, except we deleted the entire gene span and measured the RT values of the region centred on the gene before and after it's deletion. Investigating the intersections between the isolated gene sequences, which include the introns, and previously examined genome regions reveal that the majority of ORIs (61%) and CpG islands (72%) are included in the gene deletion experiment. Conversely, the combined intersection between the gene regions and both ORIs and CpG islands is less than 3% of the gene span.

A



B

Source	GO Term	Term Name	Negative Delta P-Value	Positive Delta P-Value
CORUM	CORUM:7263	ABCG5-ABCG8 complex	8.30E-03	
CORIUM	CORUM:6186	ArPIKfyve-Sac3-Sph1 complex	2.50E-02	
GO:BP	GO:0009410	response to xenobiotic stimulus	5.90E-03	
GO:BP	GO:0045796	negative regulation of intestinal cholesterol absorption	4.60E-02	
GO:BP	GO:1904562	phosphatidylinositol 5-phosphate metabolic process	4.60E+02	
GO:BP	GO:0060752	intestinal phytosterol absorption	4.60E+02	
GO:BP	GO:0010949	negative regulation of intestinal phytosterol absorption	4.60E+02	
GO:CC	GO:0044853	plasma membrane raft	2.70E-03	
GO:CC	GO:0005901	caveola	7.90E-03	
GO:CC	GO:0045121	membrane raft	9.30E-03	
GO:CC	GO:0098857	membrane microdomain	9.50E-03	
GO:MF	GO:0030554	adenyl nucleotide binding	4.50E-03	
REAC	REAC:R-HSA-5679096	Defective ABCG5	8.40E-03	
REAC	REAC:R-HSA-5679090	Defective ABCG8 causes GBD4	8.40E-03	
CORUM	CORUM:6699	KCNQ1 homotetramer		5.00E-02
GO:BP	GO:0003096	renal sodium ion transport		1.50E-02

C

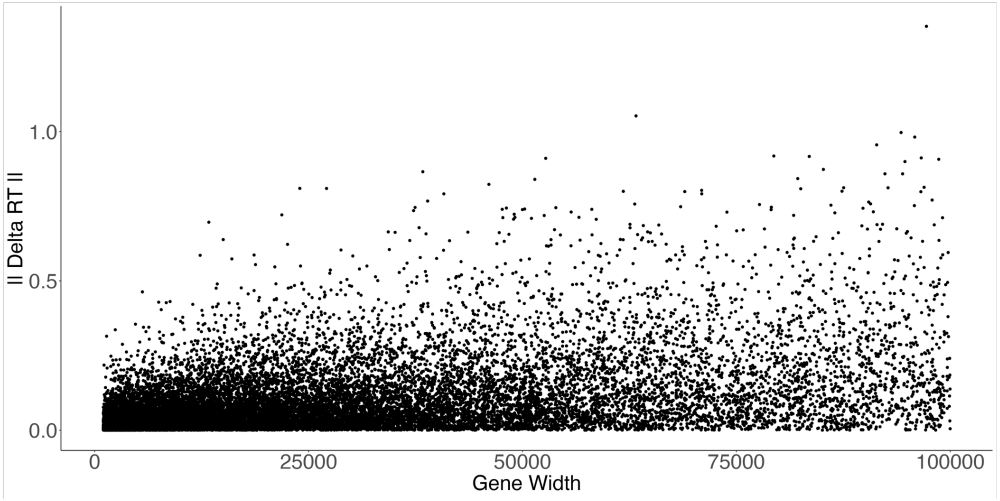


Figure 7.7: (A) The impact of gene full coding-sequence deletion on the predictions from our RT model, with the gene ontologies in (B) containing the genes which had the largest deltas. (C) The length of the deleted gene does positively correlate with the magnitude of the change in RT value, the largest changes are not seen exclusively at the longest ranges as also observed in the ORI and CpG deletions.

The resulting change in RT values are plotted in Figure 7.7 A, revealing deltas exceeding a magnitude of 1 which suggests that the most influential gene regions are capable of shifting the RT behaviour of their local region between early and late. Looking at the more conservative changes around a magnitude of 0.5, we wanted to investigate whether the genes whose deletion caused those changes were significantly enriched for different genes features such as the CpGs and ORIs studied above. To do this, we isolated the sets of genes which produced delta RTs greater than 0.5 and less than -0.5, and designated the rest of the genes with low absolute delta RT as "low delta" genes. We could then calculate the percentage of the isolated gene sets that overlapped with the genome feature of interest, and compare that to the bootstrapped distribution of overlap percentages between the genome feature and "low delta" gene spans. The genes that has delta RTs greater than 0.5 intersected with CpG islands for roughly 0.6% of their span, and with ORIs for 1% of their total span. Both of these overlaps were significantly lower than the bootstrapped population average overlaps, 2% and 3% respectively, with the p-value of observing the CpG intersection being 3.9×10^{-6}, and the p-value of the ORI intersection 1×10^{-4}. The genes with delta RTs less than -0.5 were also compared against the "low delta" background, but did not contain a statistically significant trend. It is difficult to extract a biological significance from this bulk assessment of genes, but the lower intersection between the genes that causes a high delta RT and gene features follows from our understanding of the general links between the gene features and RT. As deleting the gene led to a large increase in the local RT, the gene would likely contain fewer features that are positively correlated with RT, such as high GC content or many ORIs.

These large delta regions in Figure 7.7 C indicate that the length of the deletion does correlate positively with the magnitude of the change in RT prediction, however quantifying the linearity of that relationship is difficult from the sparsity of the data at large lengths. As the length of the deletion increases, the variance of possible deltas seems to increase compared to the deletions below 20,000bps which are all heavily localised in magnitudes < 0.3.

8

Cell-Type Specific Modelling

8.1 Key Principles

Now that we have explored the potential modelling variations for the averaged cluster behaviour, we have the opportunity to exploit the breadth of the available RT data. Our previous predictions have been on the core sequence behaviour that was shared between cell types, however it is clear from the correlation plots between datasets that, besides the core sequence pattern, each cell type also features its own variation and specific dynamics.

Producing a model that is capable of cell-specific prediction requires conditioning the output of the model on an indicator provided by the user. In this work, we first consider 3 different ways of approaching this conditioning:

1. train a **separate model for each cell state**, and use the model that is semantically closest to our target cell type when making new predictions;

2. encode the **cell state properties as a categorical feature in a single model**, and use this feature to modulate the model's output when making new predictions;

3. **condition the model on auxiliary biological data and genome tracks** which are characteristic of a cell's behaviour and easier to acquire than Repli-Seq or other experimental RT data.

Of these 3 options, #1 is clearly the weakest - training the `N` models required to achieve this will require orders of magnitude more model selection, feature engineering, and hyper-parameter optimisations than training the single model in #2. In addition, there will be significant waste of training time as the vast majority of learned patterns will be similar between these models; this is especially wasteful for model architectures that are capable of natively sharing trained components, such as neural networks. Finally, grouping by cell type results in each model having access to only $1/N^{th}$ of the overall RT data, which misses opportunities to exploit correlations in behaviour during learning, and leads to models that are unable to generalise across cell type.

Option #2 overcomes a few of these shortcomings: treating the cell-state identity as another feature allows the option for the model to train on significantly more data and be more data-efficient in its decision-making between cell types. However, there are no ways to reliably encode the cell-state descriptions available in a continuous embedding due to the natural-language style in which they are presented. Dividing the cell-state description into separate features for Cell Type, Morphology, Tissue, and Diseases Present, would allow each of these to influence separately. However, these must be Categorically encoded (either as a factor or one-hot encoded) as there is no way to smoothly semantically interpolate between the Tissue states "Blood" and "Eye", for example.

Finally, option #3 provides a candidate that addresses the majority of concerns leveraged against #1 and #2 due to the many options that become available when provided with more biological data. In keeping with the original spirit of this project, where models should be primarily sequence-driven in the interests of broad generalisation, any extra data leveraged here should be readily or cheaply available for a large number of existing cell types. The data must however provide a

A

CELL TYPE	GEO SOURCE	SRA SOURCE	MODELLING ROLE
NPC_H9	GSE115044	SRR72304{29, 36}	Test
HUVEC	GSE121781	SRR81100{45, 47, 49}	Test
HeLaS3	GSE121840	SRR81712{96..99}	Train
iPS_H2	GSE122319	SRR81764{46..47}	Train
FIB_BJ	GSE126376	SRR85581{22..23}	Train
ESC_H9	GSE139099	SRR103117{91..92}	Train
U2OS	GSE139190	SRR103199{09..10}	Train
Hepa_CyT49	GSE148368	SRR115185{41..42}	Train
A2780	GSE149145	SRR115895{88..90}	Train
ALL_RCHACV	GSE149885	SRR117300{45..46}	Valid
DE_CyT49	GSE167596	SRR137906{86..88}	Train
IMR90	GSE169767	SRR141028{75..76}	Train
MCF7	GSE169929	SRR141033{47..48}	Train
HepG2	GSE170012	SRR141035{81..82}	Train
CML	GSE170214	SRR141041{25..26}	Test
HCT116	GSE170407	SRR141052{65..66}	Train
BLymph	GSE170918	SRR141077{37..40}	Train
ALL	GSE172903	SRR143059{33..35}	Train
SKNSH	GSE172943	SRR143059{87..90}	Test
HEK293	GSE197009	SRR18072315	Valid
ESC_H7	GSE85066	SRR36897{59..60}	Train
ESC_H1	GSE85879	SRR40458{96..97}	Train
RPE1	GSE99287	SRR5601872	Train
PE_CyT49	GSM3175562	SRR7261717	Train

B

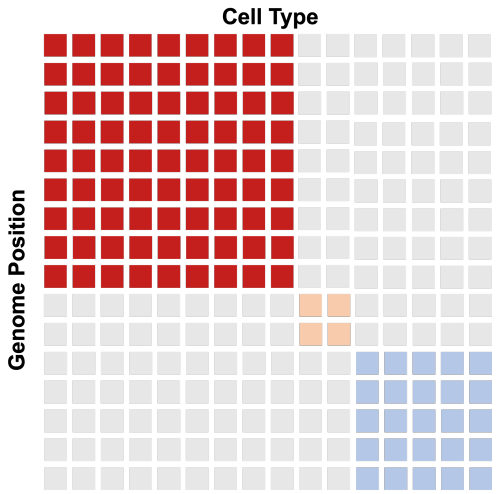


Figure 8.1: (A) Table of ATAC-seq sequence reads for each of the cell types we could find a direct match for. (B) To ensure that the model is learning as generalised a representation of RT from the DNA and additional ATAC-seq data, we enforced a strict separation along both genome position and cell line 'axes' for each the training, validation, and testing splits. Visualisation inspired by Fig 2(a) Partition 4 from Yan et. al.[275]

sufficiently unique signature to differentiate between the cell types that are present in the dataset and demonstrate generality. ATAC-seq data is chosen as the primary candidate for this role, as it meets all the criteria for ready availability and very little barrier to generation if new cell types are needed.

8.2 Database Preparation

To prepare for cell type specific modelling, we need to focus on bins that have easily comparable data. This requires there to be data present from a wide range of the available cell types, and for the data in that bin for each cell line being tightly distributed to ensure consistency. Finally, we want to avoid any spurious outliers that might interfere with the generalisation of the model. We averaged the available dataset by their cell type, which we describe as the combination of the Morphology and any Diseases that were recorded, as these are likely to have the highest impact on the behaviour of the cell. Within each cell type group of each bin, we then calculated summary statistics: mean value, standard deviation of values, and the p-value of the Shapiro-Wilks normality test. All bin cell type

groups which report an absolute RT mean value of > 2, a standard deviation > 0.5, and are not normally distributed have their values set to `NA`. With all potentially spurious data removed, we then selected only the DNA bins with average values for all cell types, resulting in 144,059 bins for each cell type to train with.

To provide the additional data for option #3, GEO and ENCODE were searched for ATAC-seq data corresponding to the cell types present in each cell state. Due to most of the original data being extracted from "ReplicationDomain", primary sources with detailed morphology and differentiation information were not available for all cells. We were able to match 24 of the original cell types with comparable data in the external databases. On investigation of available ATAC-seq data in public repositories, it was clear that the different processing routes from original sequence data to final coverage values were sufficiently different that the data could not be compared between samples. As such, it was necessary to process the data from the original sequence data using a consistent processing pipeline. Following the recommendations in Yan et. al. 2020[276] and avoiding the normalisation pitfalls highlighted by Reske et. al 2020[32], we produced a pipeline that ingested the raw sequencing data and produced comparable ATAC-seq tracks for each of the cell types in Figure 8.1.

This pipeline is broken down into these key stages:

1. Quality control of the original reads with `FastQC`[277]
2. Aligning to reference genome with `Bowtie2`[278]
3. Filter out non-paired or allosome reads with `Samtools`[279]
4. Calculate the library complexity of all samples with `ATACseqQC` R package[280]
5. Subsample to a consistent read-count and complexity with `Samtools`[279]
6. Generate genome tracks with the `deeptools` function `bamCoverage`[281]
    - Filtering regions with the ENCODE blacklist[282]
    - Bin size: 128bp, smoothing-length: 640bp, BPM normalisation

Deliberate efforts were made to ensure that validation and testing data was fully unseen, such that evaluation on these datasets would reflect real-world generalisation. This was achieved by separation out not only chromosomes for different validation folds and testing, but also keeping entire cell-line ATAC-seq datasets held-out from the model training, such that no data from any of those cell types were seen during training, Figure 8.1 B. Due to similarities between cell types visible in the ATAC-seq data it is likely that certain held out cell types will be more similar to those present in the training data, but in the process of exploring the data no data trends that entirely encapsulated cell type differences were found, so this splitting process was deemed to be a good test for generality.

8.3 DNA and ATAC Model Architecture

To create the model that is able to make cell-type specific predictions and efficiently incorporate the ATAC-seq data, we trialled two methods of combining our existing DNA sequence based RT predictors and methods of encoding the ATAC-seq data. Our "main" model, shown in Figure 8.2 A, reuses the compressive convolutional framework from the "average" model for core-sequence RT, and creates a new convolutional "trunk" for the ATAC-seq data. The outputs of each of these convolutional "trunks" were then merged and fed into a set of "linear" layers to produce the final prediction. The second model referred to as the "minimal" model, shown in Figure 8.2 B, uses the best frozen 3-mer model from section 6 as the DNA "encoder", reducing the input sequence from a $4 \times 100,000$ input to a 1×100 embedding from the middle layer of the output "linear" tail of the 3-mer model. The ATAC-seq data is processed by counting the number of positions in the ATAC input that lie between two optimisable threshold values, approximating the amount of each ATAC-seq track, which is in a "peak" of openness rather than exhibiting background closed behaviour. This compressed each $1 \times 100,000$ ATAC-seq input into a single value, with only 2 tunable parameters, hence is considered to be the more "minimal" approach. As determining whether the values in a tensor are above or below a set threshold is not a differentiable operation (the gradient at all

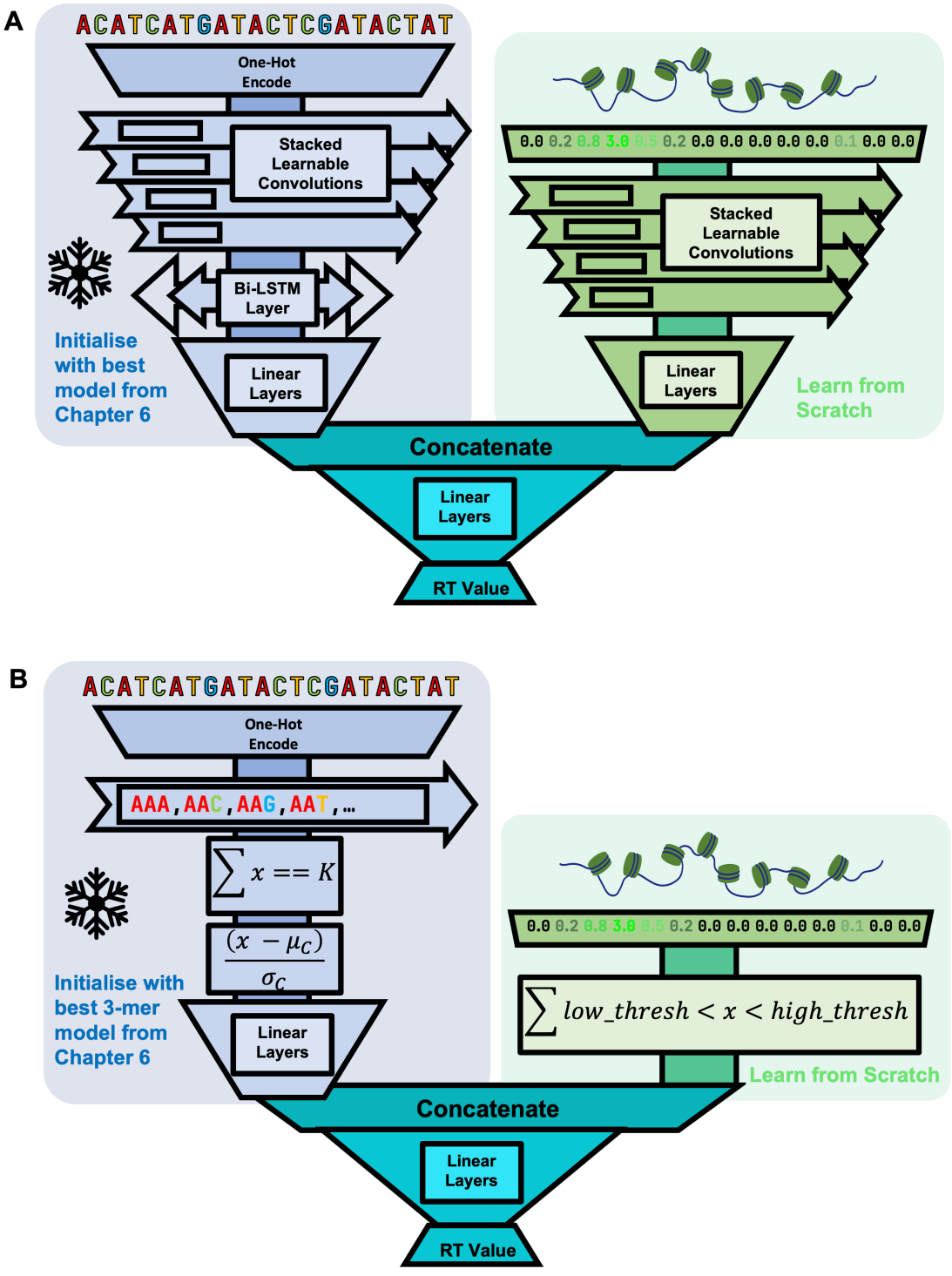

A
ACATCATGATACTCGATACTAT
One-Hot Encode
Stacked Learnable Convolutions
Bi-LSTM Layer
Linear Layers
Initialise with best model from Chapter 6
0.0 0.2 0.8 3.0 0.5 0.2 0.0 0.0 0.0 0.0 0.0 0.1 0.0 0.0
Stacked Learnable Convolutions
Linear Layers
Learn from Scratch
Concatenate
Linear Layers
RT Value
B
ACATCATGATACTCGATACTAT
One-Hot Encode
AAA, AAC, AAG, AAT, ...
$\sum x == K$
$\frac{(x - \mu_C)}{\sigma_C}$
Linear Layers
Initialise with best 3-mer model from Chapter 6
0.0 0.2 0.8 3.0 0.5 0.2 0.0 0.0 0.0 0.0 0.0 0.1 0.0 0.0
$\sum low\_thresh < x < high\_thresh$
Learn from Scratch
Concatenate
Linear Layers
RT Value

Figure 8.2: Schematic visualisation of the ATAC-seq enhanced RT prediction architectures, which combine learned features from the sequence and ATAC-seq tracks into a single predictor that can generalise to unseen cell types. The snowflake symbol indicates regions of each model, which are initialised from the best performing core-sequence RT behaviour, then "frozen" and unchanged during subsequent model optimisation. Architecture (A) uses a series of stacked convolutions to embed the DNA sequence and ATAC-seq data independently, after which they are concatenated and fed through a final linear layer to produce the cell specific RT prediction. Similarly, Architecture (B) also extracts features from both input modalities separately, but does so using methods with significantly fewer parameters due to user design. The DNA sequence embedding is generated using the best 3-mer initialised model from Chapter 6, and the ATAC "embedding" is calculated by counting the number of positions that lie between two optimisable thresholds, emulating peak counting and outlier removal.

positions will be 0), we substitute the hard threshold for the output of a sigmoid curve applied to the input after transforming them by subtracting the threshold from the tensor or vice versa to approximate the "<=" and ">=" operations respectively. The transformed vectors are multiplied by 10 to "sharpen" the sigmoid value, producing numbers much closer to 0 and 1, while still allowing for gradient propagation through the threshold parameters. The outputs of the 3-mer model and the ATAC-seq thresholded-count were then concatenated and sent through 2 "linear" layers to produce the final RT prediction.

By carrying over the parameters from the previously trained core-sequence RT predictors, this training process hoped to take advantage of the training already carried to speed the convergence of these models to cell type specific predictions. The "fine-tuning" approach is commonly carried out when applying a pre-trained model to a different problem which is closely related to the original one. The generalisation from core-sequence to cell type specific RT is not trivial, but as the DNA sequence input is common to all cell types we believe that useful transforms were learned in the original linear output layers of the core-sequence RT models. As in the core-sequence RT model, we use the MSE function as our loss value to minimise and include dropout and weight decay factors between 0 and 0.02 to regularise and encourage generalisable performance.

These model architectures both inherited a sub-model from Chapter 6, and to use the learned parameters those sub-models had their parameter shapes unchanged for training. To process the ATAC-seq data, the "main" model used a series of stacked convolutions, conceptually identical to the DNA encoder in the model from Chapter 6.3, with 9 CNN layers with $[8, 8, 8, 8, 4, 4, 4, 4, 4, 1]$ kernels, each of width 5, and each followed by batch normalisation (BatchNorm), a maximum pooling (MaxPool) of width 2, and ReLU activation function. The ATAC-seq processing in the "minimal" model was significantly simpler, as stated above, with only two tunable parameters that defined the upper and lower threshold. Both models then concatenate their DNA and ATAC-seq encodings and send them through a central hidden layer of size 100, before mapping that to a single RT prediction value.

8.4 Model Performance

During training, a number of key behaviours were observed that characterise some of the challenges inherent to training deep learning models on DNA sequence data. The sensitivity of the training dynamics to the random seed that determines the original weight values and the data shuffling order have already been discussed, and were accentuated here due to the increased diversity of target values. Tracing the influence on the model of the repeated exposure to the same DNA sequence with multiple accompanying ATAC-seq tracks was difficult, especially as it had to be disentangled from the other changes during the modelling process, such as architecture changes and weight random initialisation. The type of analysis required to do this, on a task such as this with such large inputs and a diverse range of internal components, is not in the scope of modern machine learning - although we believe that detailed understanding of the influence of data composition on DNN training dynamics will likely play a crucial role in making these models more efficient to train and interpret.

Regardless, in order to help improve the stability of the model training on this range of data despite memory constraints, we employed a process called "gradient

accumulation" to update the model's parameters only after multiple batches had been fed through the model, effectively increasing the batch size without extra memory requirements. For each architecture we trialled gradient accumulation values of 4, 8, and 16 batches but found no significant impact on the final model performance. Finally, we investigated the impact of not freezing the core-sequence RT predictors that are used as the DNA encoders in each model, on the premise that updating these additional parameters might allow the final model to capture more nuanced patterns in the relationship between DNA sequence and RT. However, the additional degrees of freedom afforded by these extra parameters exclusively led the model to over-fit even when trained with a range of learning rates, thus the models were kept frozen for all the subsequent reported results.

8.4.1 Unseen Test Combinations

After finalising the model training parameters through a combination of sweeps for the main hyper-parameters (learning rate and batch size) and heuristically updating the internal architectures, we arrived at final models for both architecture paradigms above. The "main" model archetype was first tested on the fully unseen combinations of DNA sequence and cell ATAC-seq data, with predictions visualised in FIgure 8.3. These predictions show positive correlations with experimental values, and crucially a diversity of prediction responses across the different cell types. This strongly suggests that the model has learned to leverage both the useful information in the DNA sequence and the relative exposure described by the ATAC-seq data to produce predictions that reflect the cell specific RT state. While the performance metrics of the model are weaker than those of the model trained to predict the core-sequence RT behaviour, there is significantly more variability for this model to try to capture than the core sequence. Surprisingly, the "minimal" model's performance, when measured on this same unseen test dataset, shows stronger correlations and similar diversity of behaviour, Figure 8.4. This suggests that a large number of parameters is not necessary to differentially predict the behaviours of different cell-type RT, and a simple function of relative genomic exposure is

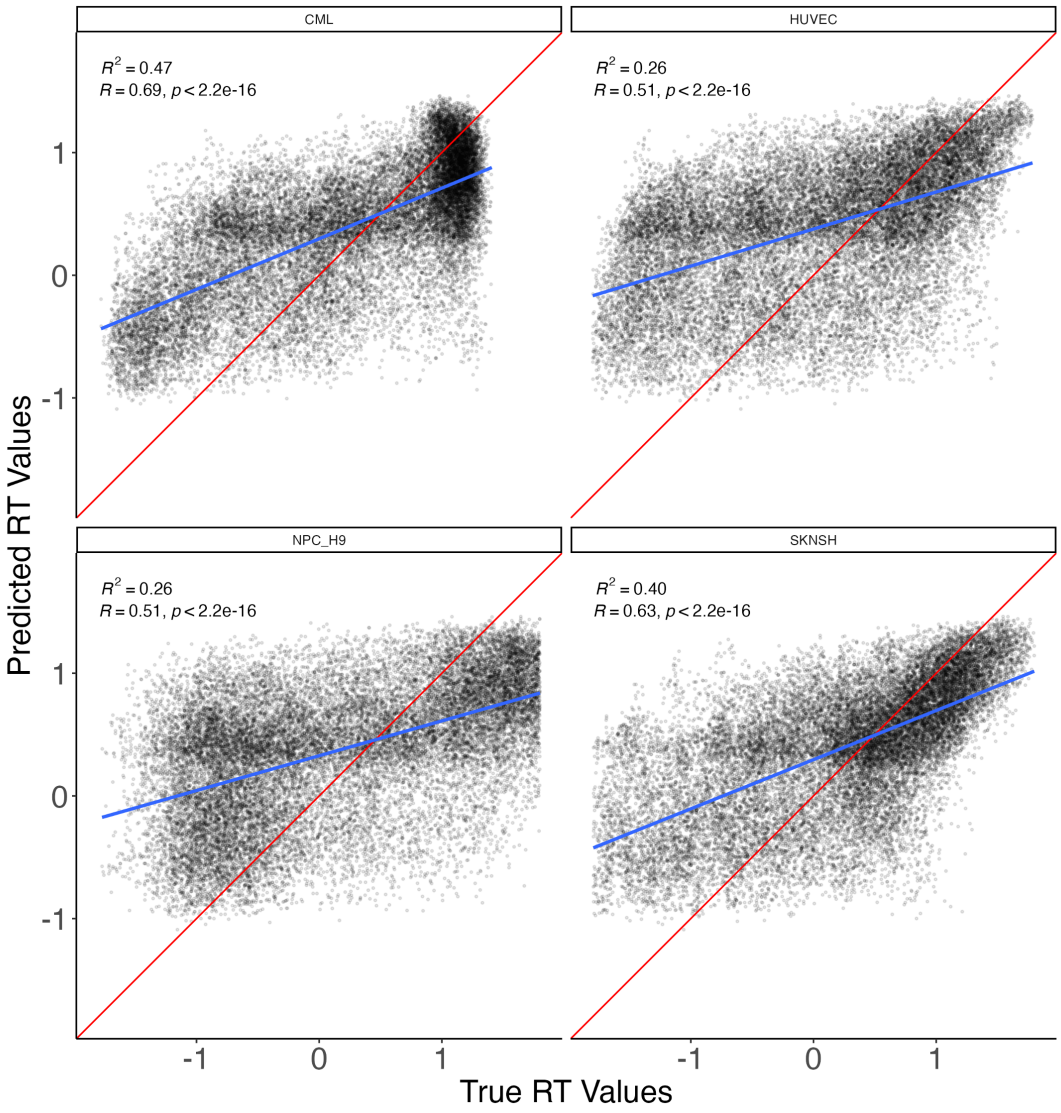


Figure 8.3: Performance of the "Main" modelling architecture on data from unseen cell types and genome bins. All subplots are the prediction outcomes for the single model with the "test" set of genome regions conditioned on the ATAC-seq data for each unseen cell type. The red annotation line shows the "identity" line, corresponding to theoretically perfect predictions from the model, and the blue line shows a linear fit between the "True RT Values" and the "Predicted RT values". The Pearson R and R^2 coefficients for the fit are shown in the top left.

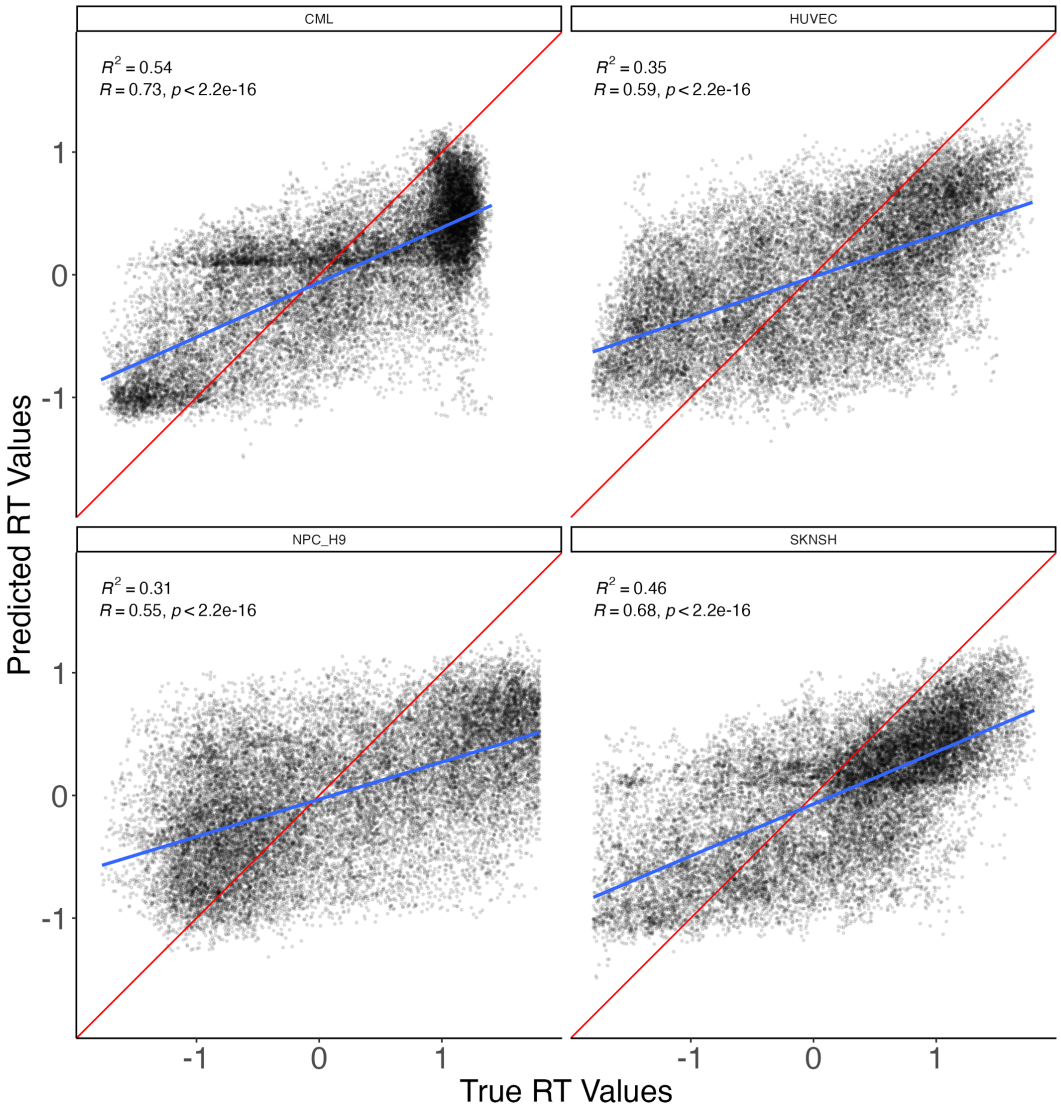


Figure 8.4: Performance of the "Minimal" modelling architecture on data from unseen cell types and genome bins. All subplots are the prediction outcomes for the single model with the "test" set of genome regions conditioned on the ATAC-seq data for each unseen cell type. The red annotation line shows the "identity" line, corresponding to theoretically perfect predictions from the model, and the blue line shows a linear fit between the "True RT Values" and the "Predicted RT values". The Pearson R and R^2 coefficients for the fit are shown in the top left.

sufficient to create this distinction. Qualitatively, it can be seen that the "minimal" model has learned to predict "later" values on average, which makes it unable to correctly predict early regions as well as the "main" model but leads to the overall performance being significantly better. This shift, along with the banding of some predictions around 0, is not easy to disentangle by examining the resulting model but is possibly an artefact on each model failing to generalise an identified pattern in the genome to a wide enough set of predictions. Additionally, we can see when comparing the overall structure of predictions for individual cell types that both models display similar distributions despite differences in architecture, suggesting that both models have converged on a similar representation of the DNA sequence and interpretation of the influence of ATAC-seq data.

8.4.2 Relative Generalisation Performance

As the minimal model showed the best performance on the unseen cell types and genome regions, we used it to investigate which of those two novel generalisations are more difficult for the model to capture. This was carried out by selecting subsets of the possible combinations of DNA sequence and ATAC-seq data that had not been used for training, which either introduced unseen sequence or unseen cell types. We first trialled the "minimal" model on predicting the RT values for DNA sequences in the "test" section using ATAC-seq data from the cell types it was trained on, visualised in Figure 8.6. We can see that, on average, the R^2 and Pearson R values are on average higher than for the unseen cell types, suggesting that the performance of the model has generalised well onto unseen DNA sequences. Furthermore, the qualitative differences across cell types that was visible in Figures 8.3 and 8.4 also occur here, suggesting that the model has not defaulted to a best fitting "single-mode" and is able to use the ATAC-seq data to effectively alter the RT predictions where needed. Conversely, when the same "minimal" model was tasked to predict the RT values for ATAC-seq data from unseen cell types on the DNA sequences it was trained on, the performance of the model was marginally reduced when compared to its performance on the test DNA sequences/ATAC-seq

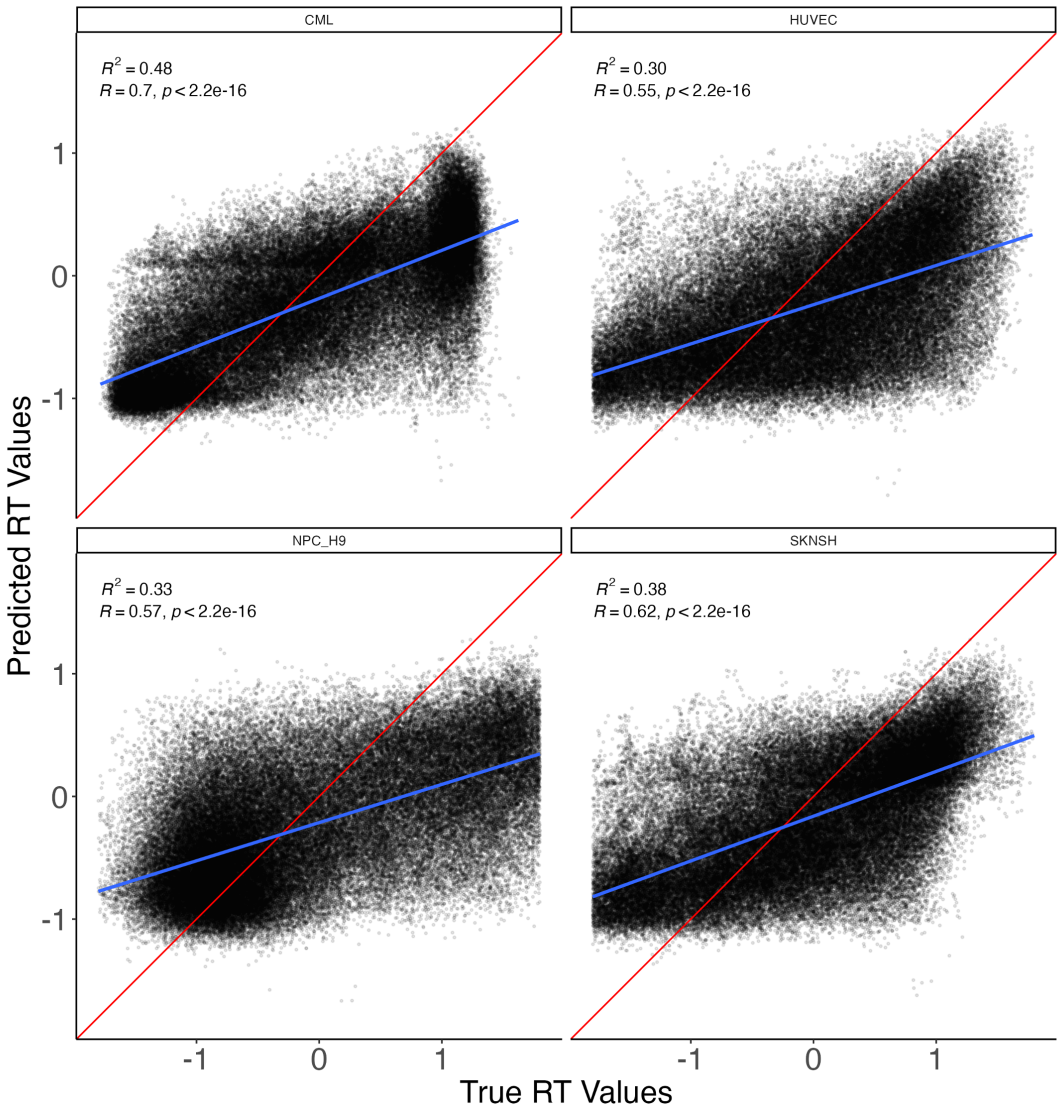


Figure 8.5: Performance of the "minimal" model on the training DNA sequences with ATAC-seq data from cell types unseen during training. The red line shows the pattern of theoretically perfect performance, and the blue line shows a linear fit between true and predicted RT. The Pearson R and R^2 coefficients for the fit are shown in the top left.

data, visualised in Figure 8.5. This surprising result indicates that the model is perhaps under-fitting to the available data, and this is perhaps a reflection of the huge information bottleneck introduced by the ATAC-seq filtering. Crucially, while the performance metrics are lower, there is little loss of cell-type specific behaviour and the general trends of, for example, a tendency to predict early SK-N-SH cell regions as later is preserved between the two sets of genome regions.

We noted that in general unseen cell types with RT ranges ±2 have worse

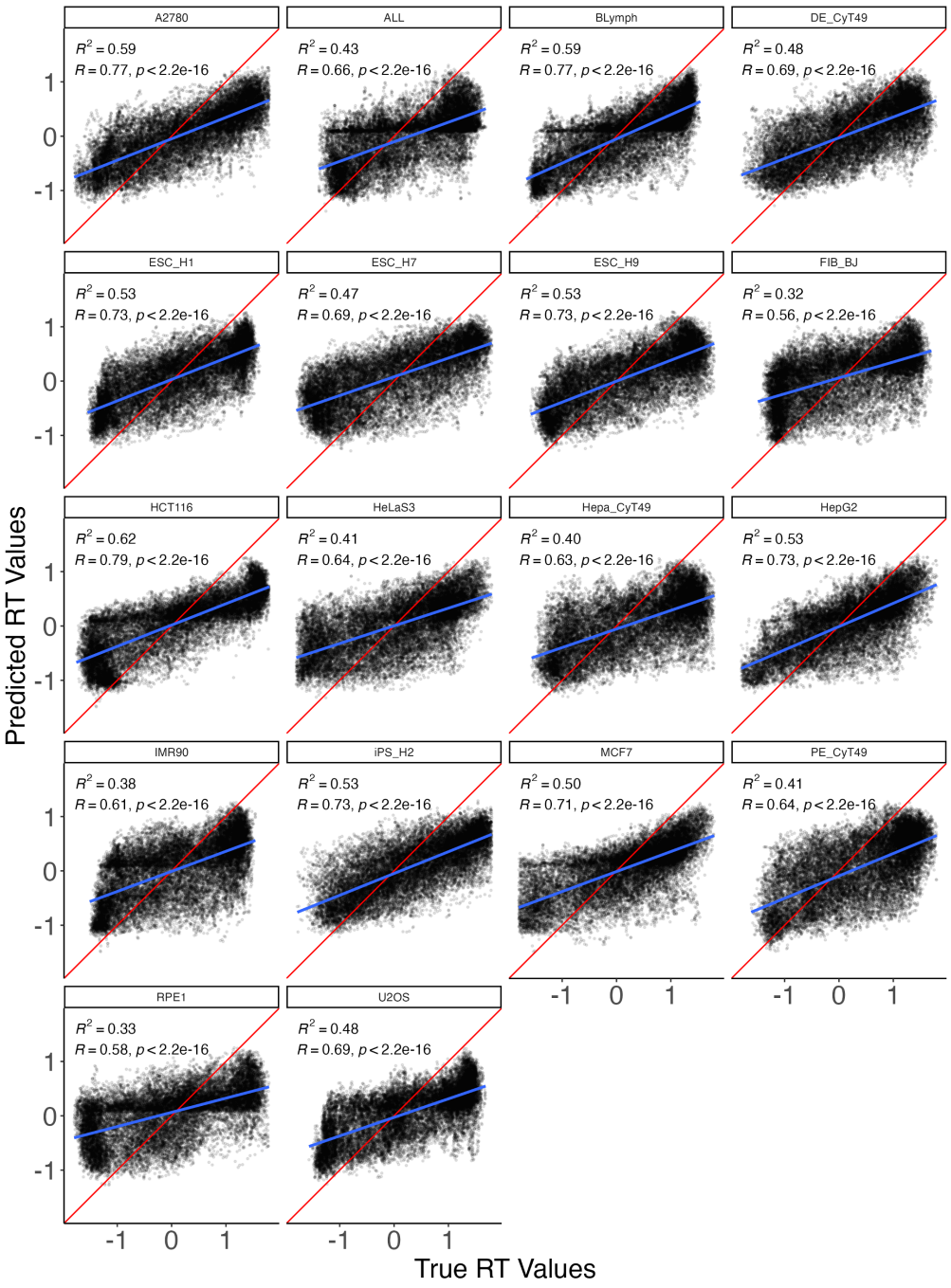


Figure 8.6: Performance of the "minimal" model on the "test" DNA sequences with ATAC-seq data from cell types seen during training. The red line shows the pattern of theoretically perfect performance, and the blue line shows a linear fit between true and predicted RT. The Pearson R and R^2 coefficients for the fit are shown in the top left.

predictions, as the model seems to have picked up a bias from many sets that have a dominant range of values between ±1 and has struggled to generalised. In future work, this could be alleviated by transforming the input data through a sigmoid function for training, converting all RT values to the range 0 and 1 where earlier is higher. This would ensure that extreme RT values (absolute value > 1) would have transformed values that are closer together, possibly providing the model with a stronger signal to differentiate between early and late values. A similar normalisation was used in the CONCERT model, where all values were brought into the range 0-1, and may be one of the factors that helped improve the model performance[174].

8.4.3 Comparison to Existing Models

It is likely that with additional sequence context, requiring more computing power, we could achieve even stronger performance across the genome. It has already been shown by the CONCERT model that increasing the amount of context can boost similar architectures to greater performance, with systematic improvements to each of their individual cell type models performance when using architectures with larger (505kb) context on pre-processed sequences[174]. The models trained in the CONCERT paper provide the closest possible comparison in the literature for our models here, as they predict (transformed) RT values for specific cells types. In particular, the "CNN-BiLSTM-local" model that they generated as a reference is very conceptually similar in architecture to the DNA-sequence processing "trunks" of our models, but their "Concert-Hierarchical" architecture is most similar as it is their only one that works directly from un-processed DNA sequence. Encouragingly, the performance of our "minimal" model matches the range of R^2 performance metrics of their "CNN-BiLSTM-local" models across cell types, which we believe is very promising evidence in favour of our model as it is a single model able to predict for all these cell types, whereas each "CNN-BiLSTM-local" is trained and tested on one cell type and the sequence is the only thing that is "unseen". Our model's performance in the best test case also approaches that of the "Concert-Hierarchical" model, with our best unseen Pearson's R of 0.730 on CML approaching

the only reported "Concert-Hierarchical" performance of 0.785 on H1-hESC, with the additional benefit that our model uses only $1/5^{th}$ of the sequence context and can predict for any other cell type with ATAC-seq data[174].

9 Conclusions

9.1 RT has Strong DNA Sequence Dependencies

Previous experiments have shown that RT displays broadly consistent but permuted behaviour across human cell types. Further, established literature states that RT is mechanistically rooted in patterns in the DNA sequence, and has been found to be moderated by specific sequences with identifiable composition. We have demonstrated through analysis of a large body of existing RT experimental data that patterns of consistent RT values across cell types correspond to broad changes in DNA sequence composition and the presence of various genomic features (such as CpG Islands). Additionally, we identified that the distribution and structure of genes in the genome is correlated with RT consistency, with regions that are consistently early being heavily enriched in short length genes, Figure 4.7 D. We also show that the RT profile of cell types is sufficient to group similar cell types together, but does not on its own provide a strong low-dimensional space in which to predict the behaviour of arbitrary unseen cell types, Figure 4.1 B.

Investigating the distribution of RT values at the same genomic regions across different cell types reveals that some regions display consistent RT behaviour, which we believe is driven by DNA sequence as this is nominally shared between all cells of the same organism. Separating out the bins which display RT values

with a normal distribution and a low spread (SD), as in Figure 5.1 C, allows for the creation of a large dataset of genome regions and associated RT that can be used to create a sequence driven model. The distribution of commonly used DNA sequence features, such as k-mers, is not-normal across all length scales, and patterns in these distributions drift as the size of the genomic context window is changed. We discovered that a simple linear predictor can use the 3-mers from context regions ranging from 5kb to 1Mb showing increased performance as more context is incorporated to produce the counts, 5.3. However, between the 500kb and 1Mb context range the performance improvement plateaus, in line with findings from previous modelling work, and the prediction quality can even get worse as the context increases up to 2Mb possible due to the regions overlapping with regions of change in local sequence compositions such as isochores[174].

9.2 Core Sequence-Driven RT Behaviour can be Predicted from Sequence

We derived a set of 67 sequence features ranging in complexity inspired by literature of known behaviour of RT, and previous sequence based modelling efforts, and counted them (and their reverse complements) along bins of length 100kb to explore different modelling approaches to RT. We found that the strongest performing models on this dataset were gradient boosting tree models, Figure 5.5, and thus chose to optimise the parameters of a LightGBM model with a Bayesian optimised hyper-parameter search. The resulting model outperformed its predecessors and produced predictions that consistently had error less than or equivalent to the known variance of the underlying core sequence-driven RT behaviour, Figure 5.6 A and B, and provided attribution scores for each of its features to aid interpretability. These attribution scores revealed a surprisingly powerful triad sequence component "ATG", that outperformed all other features including the GC count and other GC rich K-mers as the most useful feature for prediction Figure 7.1, and as a solo feature was able to predict the core-sequence RT behaviour with and R^2 value of 0.7

between the true and predicted RT. One reason that it could be so predictive of core-sequence RT could be it's relationship with genes as the primary eukaryotic "start" codon, and it would be interesting to explore if similarly exceptional predictors exist for different K values, or if triads are uniquely powerful.

9.3 Deep Learning Provides Efficient and Expressive DNA Sequence-Based Learning

While the LightGBM model provided a powerful predictor of core-sequence RT using globally aggregated features, we sought to investigate whether DNNs could be trained to perform the same prediction directly from the DNA sequence without the need for engineering the 67 features with prior literature and biological knowledge. This resulted in the creation and training of series of CNNs that read a one-hot encoded representation of the DNA sequence and extract features to use downstream. We found that "handcrafting" the CNN kernels to extract 3-mers from the sequence provided a powerful boost to the models performance early on, Figure 6.2, and we could mimic the process of centering and scaling the data within the DNN without risk of train/test leakage.

To create a more flexible system we combined the motif extraction capabilities of the CNN with the pattern-combining logic of a recurrent Bi-LSTM layer to create an adaptive model that could be trained from random initialisation to predict the core-sequence RT. This model underperformed the more heavily pre-engineered models in quantitative performance on the test set, Figure 6.3 B, but solved a considerably harder task in the process - learning what features to extract and how to combine them into an effective prediction of core sequence-driven RT. Furthermore, this fully automatic model leveraged a custom designed CNN layer to guarantee reverse complement equivariance for any input without needing to augment the dataset.

9.4 Incorporating Different Data Modalities in Deep Models allows Cell Type Specific Performance

Encouraged by the success of the core-sequence driven RT predictions, we established a new problem stage of predicting the RT values for individual cell types by leveraging readily available data to distinguish the cell samples from one another. We settled on ATAC-seq data as it was readily available for 24 of the cell types for which we had RT data, and because local chromatin structure is known to be causally linked to RT. We processed the sequencing data for each of these dataset from scratch and normalised them to create a set of comparable data to guide a single model that can predict for any cell type. We again explored two different architecture paths for this model, Figure 8.2. The first was considered the "main", an extension of the fully automatic model from Chapter 6, that would use that model as a DNA encoder and append to its output an encoding of the ATAC-seq data for each region, leveraging a similar CNN architecture. The second model was considered "minimal", and used the "handcrafted" RT model along with a very simple ATAC-seq "encoder" that counted the number of positions in the ATAC-seq data between two learnable thresholds.

Surprisingly, the simpler "minimal" model outperformed the larger model on unseen DNA regions with ATAC-seq from previously unseen cells, Figure 8.3 and 8.4. To explore the generalisation properties of this model further, we trialled the "minimal" model on predicting the values of ATAC-seq data from known cell types on the test DNA sequences, Figure 8.6, and ATAC-seq from unseen cells on the training DNA regions, Figure 8.5. The results of this comparison suggest that the model was more reliably able to generalise its internal representation of already seen cell types to unseen sequences than vice versa, due to the better overall performance in Figure 8.6. Counterintuitively, the performance of the model predicting for unseen cells ATAC-seq data on the training DNA regions was worse consistently than its performance with the same cells on unseen testing DNA regions. This would suggest that the model has perhaps over-fit to the training sequence data.

Introducing additional training mechanisms such as adaptive learning rate and batch control could help close this gap.

9.5 Future Work

This work opens up a great deal of potential future avenues for further investigation, and these can be broadly categorised as incremental or structural changes. Incremental improvements could be made by any strategy that furthers the development of models similar to those we have already made, such as the generation of more experimental data and improving data loading speeds to increase the rate of model training and experimentation. As we explored in Chapter 6, increasing the sequence context beyond 100kb has great potential for improving model performance, and the increased memory cost of this could be balanced by lowering the batch size and applying gradient accumulation to recover training stability. If we are interested in improving the internal representations that the model is developing during training, there is a potential avenue for applying more dramatic regularisation to the weights of the model to encourage sparsity, or change the target function from a regression to a classification (of "early" or "late", for example) to provide a more direct signal to the training process. Alternatively, there may be benefits to shifting the target value range for the regression from ±2 to 0-1, as this appears to have worked well for the CONCERT model[174]. This could be achieved with a reversible sigmoid mapping or similar.

Larger scale structural changes could also be applied to improve the model performance at the cost of significantly more computation and leaving behind much of the domain knowledge developed during this work. There are many DNA-sequence attention based models being pre-trained on generic DNA prediction tasks, such as HyenaDNA, that might make a powerful backbone for an RT predictor. Alternatively, one could explore other efficient parametrisations of input data processing, like the ATAC-seq thresholds used in our minimal models, that might allow efficient and interpretable downstream applications.

Appendices

A

RT Datasets

Dataset	Cell Line	Tissue	Morphology	Disease	Cell Type	Experiment Method
KorenA2780	A2780	Ovary	Epithelial	Carcinoma	A2780	Repli-seq (Koren)
Ext38079077	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int71645241	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int44247008	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int39419315	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int17615485	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int56827427	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int81363378	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int42700050	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int94072292	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int23614946	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int25421514	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int45825531	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int37567550	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int67853626	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int22656701	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int39446261	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Int95787414	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip

Dataset	Cell Line	Tissue	Morphology	Disease	Cell Type	Experiment Method
Int29665185	Patient-Blood	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL	Repli-chip
Ext93694043	MHH-CALL2	Blood	Lymphoblast	Acute Lymphoblastic Leukemia	ALL_MHHCALL2	Repli-chip
Ext43197738	RCH-ACV	Bone Marrow	Lymphoblast	Acute Lymphoblastic Leukemia	ALL_RCHACV	Repli-chip
Int80298719	GM06999	Blood	Lymphoblastoid	NA	BLymph	Repli-chip
Ext54054609	GM06990	Blood	Lymphoblastoid	NA	BLymph	Repli-chip
Int80766291	GM06990	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (ENCODE)
Int48673064	GM12801	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (ENCODE)
Int30870010	GM12812	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (ENCODE)
Int71952664	GM12813	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (ENCODE)
Int12122254	GM12878	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (ENCODE)
Int90901931	GM12878	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (4DN)
Int72761980	GM12878	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (4DN)
Int61574576	GM12878	Blood	Lymphoblastoid	NA	BLymph	Repli-seq (4DN)
KorenGM12878	GM12878	Peripheral blood	Lymphoblast	NA	BLymph	Repli-seq (Koren)
Ext63868885	C0202	NA	Lymphoblastoid	NA	C0202_Lymph	Repli-chip
Ext13588990	C0202	NA	Lymphoblastoid	NA	C0202_Lymph	Repli-chip
Int85041976	K562	Bone Marrow	Lymphoblast	Chronic Myelogenous Leukemia	CML	Repli-seq (ENCODE)
Int48435379	K562	Bone Marrow	Lymphoblast	Chronic Myelogenous Leukemia	CML	Repli-seq (4DN)
Int37482971	K562	Bone Marrow	Lymphoblast	Chronic Myelogenous Leukemia	CML	Repli-seq (4DN)
Int60800204	BG02	NA	Definitive Endoderm	NA	DE_BG02	Repli-chip
Int63667653	CyT49	NA	Definitive Endoderm	NA	DE_CyT49	Repli-chip
Int72643985	CyT49	NA	Definitive Endoderm	NA	DE_CyT49	Repli-chip
Int61646100	BG01	Blastocyst	ESC	NA	ESC_BG01	Repli-chip
Int10734660	BG02	NA	ESC	NA	ESC_BG02	Repli-seq (ENCODE)
Int87960943	BG02	NA	ESC	NA	ESC_BG02	Repli-chip
Int73235540	CyT49	Blastocyst	ESC	NA	ESC_CyT49	Repli-chip
Int83608519	CyT49	Blastocyst	ESC	NA	ESC_CyT49	Repli-chip
Int88694775	H1	NA	ESC	NA	ESC_H1	Repli-seq (4DN)
Int18370565	H1	NA	ESC	NA	ESC_H1	Repli-seq (4DN)
Ext35479608	H7	NA	ESC	NA	ESC_H7	Repli-chip
Ext29405702	H9	NA	ESC	NA	ESC_H9	Repli-chip
Int38166296	H9	NA	ESC	NA	ESC_H9	Repli-seq (4DN)
Int58433514	H9	NA	ESC	NA	ESC_H9	Repli-seq (4DN)
Ext78061553	Patient-Myoblast	NA	Myoblast	Facioscapulohumeral Muscular Dystophy	FHMD	Repli-chip
Int58331187	Patient-Myoblast	NA	Myoblast	Facioscapulohumeral Muscular Dystophy	FHMD	Repli-chip
Int79271655	Patient-Myoblast	NA	Myoblast	Facioscapulohumeral Muscular Dystophy	FHMD	Repli-chip
Int54006031	Patient-Myoblast	NA	Myoblast	Facioscapulohumeral Muscular Dystophy	FHMD	Repli-chip
Int27232215	Patient-Myoblast	NA	Myoblast	Facioscapulohumeral Muscular Dystophy	FHMD	Repli-chip
Ext35826210	BJ	Foreskin	Fibroblast	NA	FIB_BJ	Repli-chip
Ext56950408	BJ	Foreskin	Fibroblast	NA	FIB_BJ	Repli-chip
Int34501930	BJ	Foreskin	Fibroblast	NA	FIB_BJ	Repli-seq (ENCODE)
Ext47996945	BJ	Foreskin	Fibroblast	NA	FIB_BJ	Repli-chip
Ext55562945	BJ	Foreskin	Fibroblast	NA	FIB_BJ	Repli-chip
KorenHCC1143	HCC1143	Breast	Epithelial	Carcinoma	HCC1143	Repli-seq (Koren)
KorenHCC1954	HCC1954	Breast	Epithelial	Carcinoma	HCC1954	Repli-seq (Koren)
Int97243322	HCT-116	Colon	Epithelial	Carcinoma	HCT116	Repli-seq (4DN)
Int90617792	HCT-116	Colon	Epithelial	Carcinoma	HCT116	Repli-seq (4DN)
Int76222264	HEK293	Embryonic Kidney	Epithelial	NA	HEK293	Repli-seq (4DN)
Int57383924	HEK293	Embryonic Kidney	Epithelial	NA	HEK293	Repli-seq (4DN)
KorenHEK293T	HEK293T	Kidney	ESC	NA	HEK293T	Repli-seq (Koren)
Int93773609	HeLaS3	Uterus/Cervix	Epithelial	Carcinoma	HeLaS3	Repli-seq (ENCODE)
Int17909837	CyT49	Liver	Hepatocyte	NA	Hepa_CyT49	Repli-chip

A. RT Datasets

Dataset	Cell Line	Tissue	Morphology	Disease	Cell Type	Experiment Method
Int81158282	CyT49	Liver	Hepatocyte	NA	Hepa_CyT49	Repli-chip
Int68719891	CyT49	Liver	Hepatocyte	NA	Hepa_CyT49	Repli-chip
Int49277082	HepG2	Liver	Epithelial	Carcinoma	HepG2	Repli-seq (ENCODE)
Int62605959	HFFc6	Foreskin	Fibroblast	NA	HFFc6	Repli-seq (4DN)
Int95577547	HFFc6	Foreskin	Fibroblast	NA	HFFc6	Repli-seq (4DN)
Ext21612944	hFib	NA	Fibroblast	NA	hFib	Repli-chip
Ext45249226	hFib	NA	Fibroblast	NA	hFib	Repli-chip
Int46463494	HUVEC	Umbilical Cord Vein	Endothelial	NA	HUVEC	Repli-seq (ENCODE)
Int83403571	IMR90	Lung	Fibroblast	NA	IMR90	Repli-chip
Int94339003	IMR90	Lung	Fibroblast	NA	IMR90	Repli-seq (ENCODE)
Int49605910	IMR90	Lung	Fibroblast	NA	IMR90	Repli-seq (4DN)
Int78679848	IMR90	Lung	Fibroblast	NA	IMR90	Repli-seq (4DN)
Ext57181431	IMR90	Lung	Fibroblast	NA	IMR90_RAS	Repli-chip
Ext79014737	IMR90	Lung	Fibroblast	NA	IMR90_RAS_shRB	Repli-chip
Ext23228347	H2	NA	iPS	NA	iPS_H2	Repli-chip
Ext30484475	H2	NA	iPS	NA	iPS_H2	Repli-chip
Ext47609172	iPS4	NA	iPS	NA	iPS4	Repli-chip
Ext76527981	iPS4	NA	iPS	NA	iPS4	Repli-chip
Ext52445028	iPS5	NA	iPS	NA	iPS4	Repli-chip
Ext77645606	iPS5	NA	iPS	NA	iPS4	Repli-chip
Int67688290	MCF-7	Breast	Epithelial	Carcinoma	MCF7	Repli-seq (ENCODE)
Int43634063	BG02	NA	Mesendoderm	NA	MED_BG02	Repli-chip
Int71609683	CyT49	NA	Mesothelial	NA	Mesothel_CyT49	Repli-chip
Int33069021	CyT49	NA	Mesothelial	NA	Mesothel_CyT49	Repli-chip
Int51173422	H9	NA	Mesothelial	NA	Mesothel_H9	Repli-chip
Int98741332	H9	NA	Mesothelial	NA	Mesothel_H9	Repli-chip
Int36642440	H9	NA	Mesenchymal Stem Cell	NA	MSC_H9	Repli-chip
Int25284992	H9	NA	Mesenchymal Stem Cell	NA	MSC_H9	Repli-chip
Int39149059	Patient-Myoblast	NA	Myoblast	NA	MYO	Repli-chip
Ext45267227	Patient-Myoblast	NA	Myoblast	NA	MYO	Repli-chip
Int68113562	Patient-Myoblast	NA	Myoblast	NA	MYO	Repli-chip
Int68568106	Patient-Myoblast	NA	Myoblast	NA	MYO	Repli-chip
Int40538991	CyT49	NA	Neural Crest	NA	NC_CyT49	Repli-chip
Int69547455	CyT49	NA	Neural Crest	NA	NC_CyT49	Repli-chip
Int91488607	H9	NA	Neural Crest	NA	NC_H9	Repli-chip
Int35728573	H9	NA	Neural Crest	NA	NC_H9	Repli-chip
Int71405114	H9	NA	Neural Crest	NA	NC_H9	Repli-chip
Int62546708	NC-NC	Blood	Lymphoblastoid	NA	NCNC	Repli-chip
Int92817591	NHEK	Ectoderm	Epidermal Keratinocytes	NA	NHEK	Repli-seq (ENCODE)
Int93941761	BG01	NA	Neural Progenitor	NA	NPC_BG01	Repli-chip
Int89790558	H9	NA	Neural Progenitor	NA	NPC_H9	Repli-chip
Int56654336	CyT49	Pancreas	Endoderm	NA	PE_CyT49	Repli-chip
Int77430856	CyT49	Pancreas	Endoderm	NA	PE_CyT49	Repli-chip
Int77282288	CyT49	Pancreas	Endoderm	NA	PE_CyT49	Repli-chip
Int99268934	CyT49	Pancreas	Endoderm	NA	PE_CyT49	Repli-chip
Ext83220597	REH	Blood	Lymphoblastoid	NA	REH	Repli-chip
Int80406555	REH	Blood	Lymphoblastoid	NA	REH	Repli-chip
Int28397865	RPE-1	Eye	Epithelial	NA	RPE1	Repli-seq (4DN)
Int42543431	RPE-1	Eye	Epithelial	NA	RPE1	Repli-seq (4DN)
Int67184500	SK-N-SH	Brain	Epithelial	Neuroblastoma	SKNSH	Repli-seq (ENCODE)
Int67900463	BG02	NA	Smooth Muscle	NA	SM_BG02	Repli-chip
Int16405711	H9	NA	Smooth Muscle	NA	SM_H9	Repli-chip
Int42371887	H9	NA	Smooth Muscle	NA	SM_H9	Repli-chip
Int50319428	BG02	NA	Splanchnic Mesoderm	NA	Splanc_BG02	Repli-chip
Int32039340	CyT49	NA	Splanchnic Mesoderm	NA	Splanc_CyT49	Repli-chip
Int30401469	CyT49	NA	Splanchnic Mesoderm	NA	Splanc_CyT49	Repli-chip
Int64319442	H9	NA	Splanchnic Mesoderm	NA	Splanc_H9	Repli-chip
Int92004230	H9	NA	Splanchnic Mesoderm	NA	Splanc_H9	Repli-chip
Int67044443	T-Lymphoblastoid	Blood	Lymphoblastoid	NA	TLymph	Repli-chip
Int29241958	U2OS	Bone Marrow	Epithelial	NA	U2OS	Repli-seq (4DN)
Int66343918	U2OS	Bone Marrow	Epithelial	NA	U2OS	Repli-seq (4DN)

B

RT Normalisation

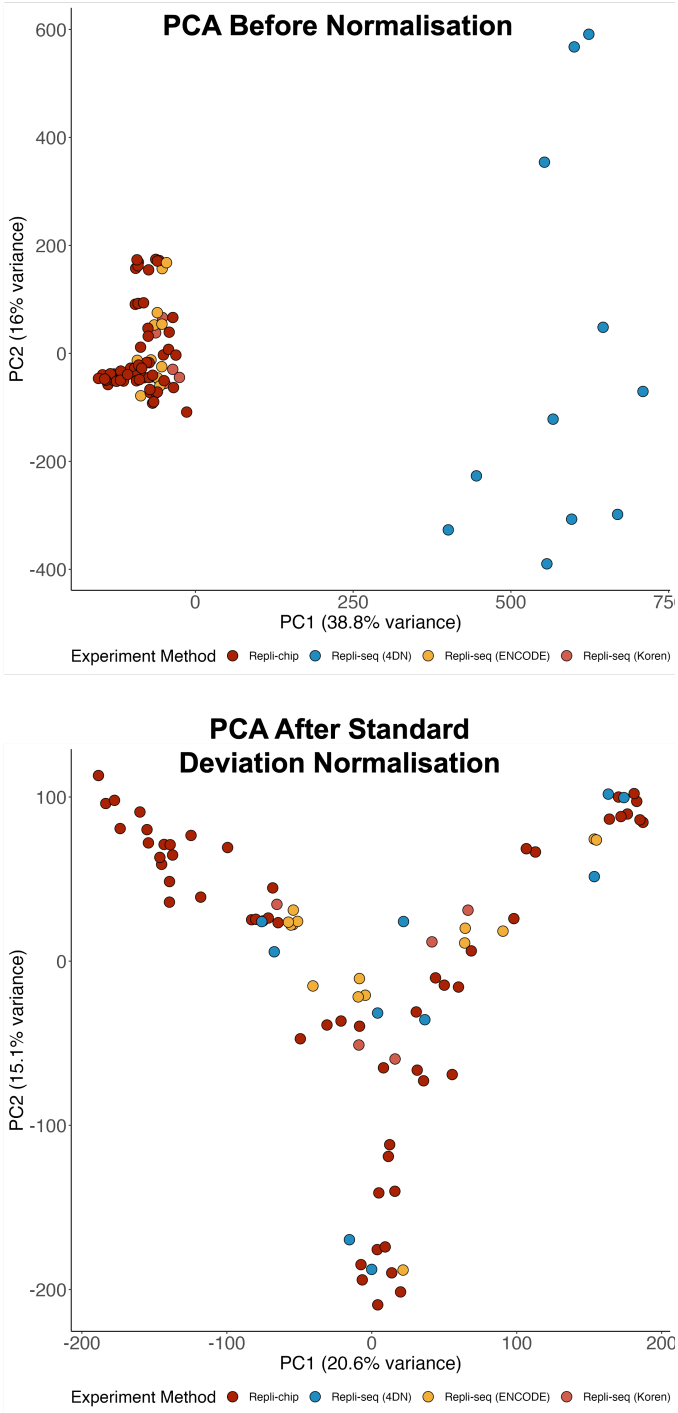


Figure B.1: Identifying the source of a batch effect in the dataset driven by the 4D Nucleome (4DN) processing pipeline, which created datasets with different RT-value ranges. Each point in this visualisation is coloured by the processing pipeline used. The 4DN batch effect is removed in our pipeline after standard deviation normalisation.

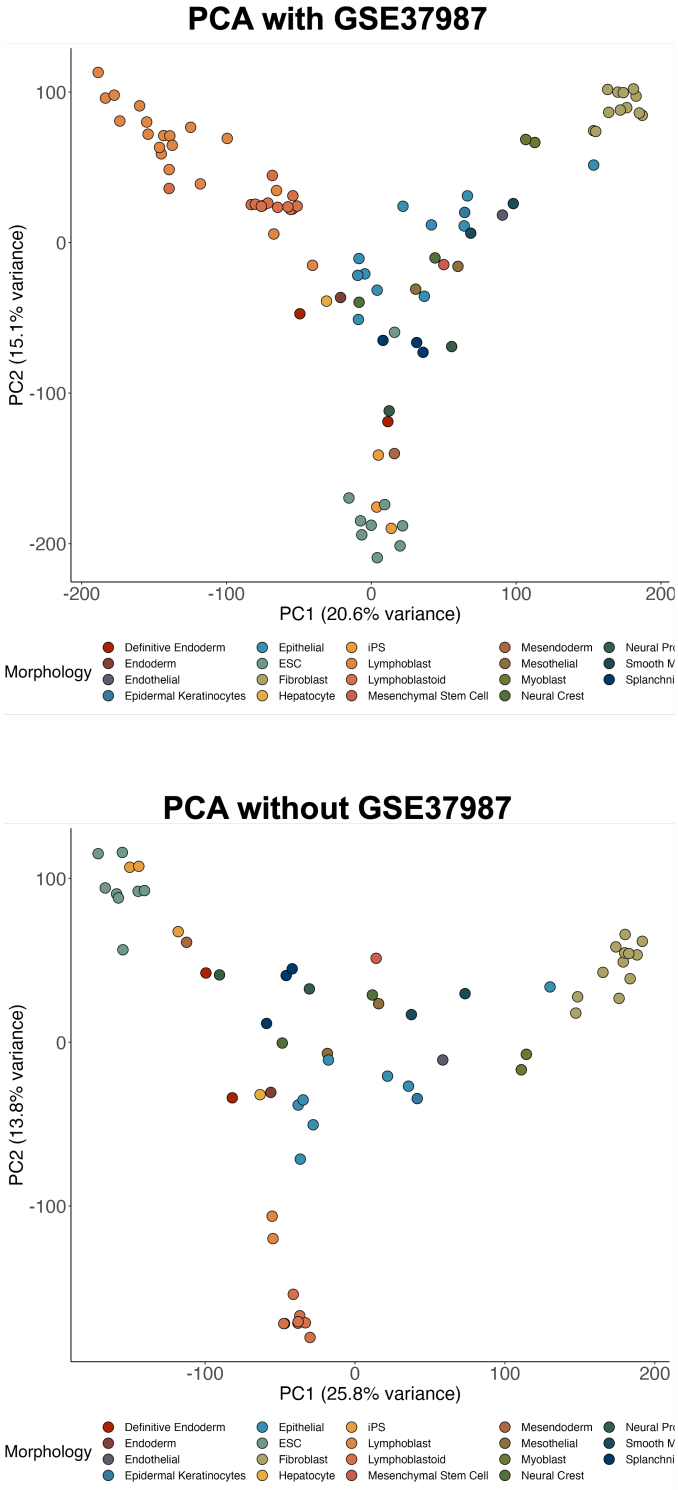


Figure B.2: PCA visualisation revealed that one experimental source, `GSE37987`, produced the majority of Lymphoblastoid experiments in our dataset all of which are grouped at the extreme of one "spoke" of the PCA space. This self-similar behaviour suggests a possible batch effect, so we re-ran the PCA embedding without any data from `GSE37987`. We observed that the 3-spoke PCA space shape still occurs, and other Lymph-based samples (B/T-Lymph) still occupy the same relative position. This suggests that the behaviour seen is driven by the biological similarity between these cell types and is not considered a detectable batch effect.

C
RT Classifiers

C.1 Reframing as a Classification Task

As previously discussed, the RT values that we use as our primary target during this work are usually continuous values between ±2, regardless of whether we are trying to predict the core sequence-driven RT or the cell type specific values. However, it is also possible to approach the prediction of RT behaviour as a classification task. This is made simpler due to the construction of the original dataset, which is centred on zero and normalised by the standard deviation, allowing us to make general claims about the relationship between groups of RT values.

As positive RT values are considered "Early" and negative RT values as "Late", we converted the continuous RT values into a binary classification by checking if the bin value is greater than 0. Applying this to the core sequence-driven RT dataset resulted in a dataset of identical size to the regression task (~66,000 bins) where each set of features corresponds to a categorical outcome; either "Late" or "Early". This reframing makes sense in the context of regions that are clearly within replication domains, however it is poorly defined for those that are in TTRs. Consequently, we can alternatively frame the prediction task as a multi-class classification problem where each bin is either "Early", "Late", or "Middle". As we know all our RT values are scaled by their standard deviation, statistically the vast majority of

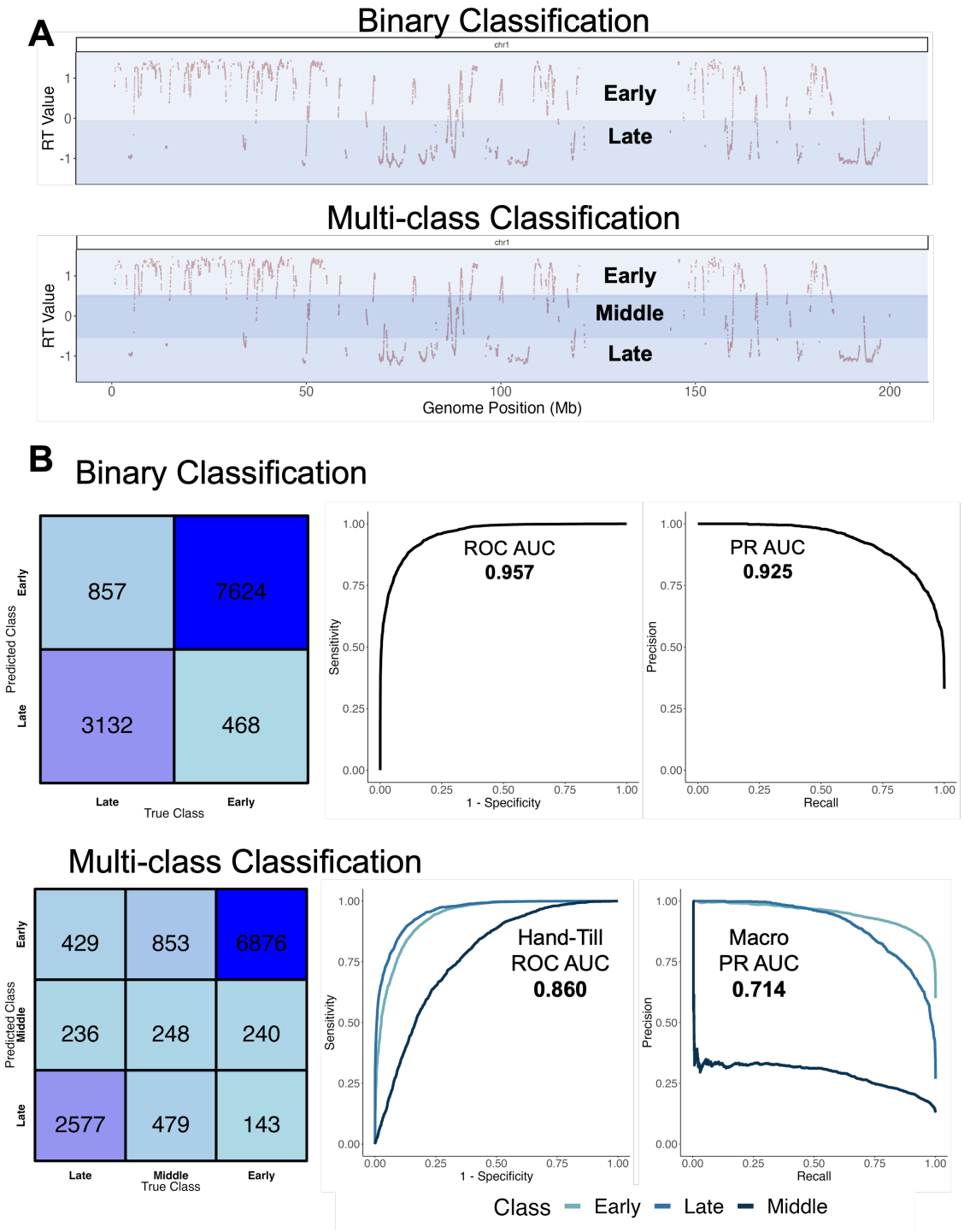


Figure C.1: (A) Category extraction from continuous RT data, for the "Binary" and "Multi-class" classification tasks. (B) Performance of LightGBM models using the original sequence features on the "Binary" and "Multi-class" targets. For each model the confusion matrix, receiver operating characteristic (ROC) curve, and precision-recall (PR) curve are shown with their corresponding metrics.

their data-points lie within ±2, and our additional knowledge of the distribution of RT allows us to choose a central cut-off at ±0.5 to act as the threshold for genome bins being assigned to the "Middle" category. It should be noted that this is not expected to overlap perfectly with all points in TTRs, as automatically detecting points in transition regions is not perfectly solved, but is expected to capture the majority of them. This choice of threshold for "Middle" RT values is constrained to avoid classifying any genome regions in RDs as "Middle", and could be tuned in a dataset specific way if we were making cell type specific predictions. For this classification problem, we choose a generic threshold from inspection of the training data distribution and don't tune it further to avoid influencing the models validation performance artificially. For this subsequent work, we will tackle both prediction of these prediction tasks and refer to them as the "Binary" and "Multi-class" RT classifier respectively. A schematic for the categorisation process for each task is shown in Figure C.1 A.

For each classification task, we use the same model architecture (LightGBM) and feature set described in Chapter 5, as these have already shown strong performance on the related task of regressing RT and do not introduce significant computational overhead. Similarly, we keep the context window from which the features are counted at 100kb around the centre of each 10kb bin. It is importance to address the relative amount of each category in the dataset, to avoid inaccurate measures of model performance based on the balance of the dataset - such as inflated accuracy on highly imbalanced data caused by only predicting the majority class. Investigating the training data, we observe that in the "Binary" task the two classes are almost balanced by default with 27896 "Late" and 27330 "Early" samples, however in the "Multi-class" tasks there are only 8931 "Middle" samples compared to 22780 "Early" and 23515 "Late". To remedy this, we used class re-balancing during training to ensure the model sees an approximately equal number of examples from each class to improve generalisation. The choice of rebalancing strategy can depends on the dataset, as down- or up- sampling can have deleterious effects if they reduce the number of data-points below a usable level or over-represent a pathological

pattern in the minority class data. The re-sampling can be carried out by removing samples, repeating sample in the training data (which for many model architectures is systematically equivalent to applying a per-sample weight) or using a heuristic to generate synthetic representative data sample, such as the SMOTE algorithm[283]. In this work, we applied downsampling to the majority classes using the `themis` R library from the `tidymodels` framework.

C.2 Predicting RT Classification with LightGBM

The performance of the "Binary" classifier on the held-out test dataset is visualised on the top panel of Figure C.1 B. These indicate that the model fitted to the task overall very strongly, consistently reporting the correct categorisation with a balanced accuracy on the unseen test regions of 0.864. The values of the receiver operating characteristic (ROC) area under the curve (AUC) and precision-recall (PR) AUC are very high, indicating that the model is able to reliably generalise to unseen sequences even in this unbalanced test set. The confusion matrix suggests that the model is marginally more consistent at correctly predicting "Early" regions of the genome, and this aligns with the outcomes from the regression models where "Late" being mis-predicted as "Early" was the most common prediction error. Overall, these results would suggest that the "Binary" RT classification we have extracted can be very consistently predicted with the sequence features we provided. During model parameter optimisation, we identified that the models were able to train on very small sub-samples of the training set, with models recovering >95% of the balanced accuracy score achieved with the full dataset using only 500 randomly selected samples (~1%) of training data. This is indicative of the dataset containing a lot of redundant information. Future work is necessary to explore whether this redundancy indicates that core sequence-driven RT behaviours are simply predictable from only few very informative features, and whether there are more useful features whose subtler influence may have been overlooked by these model architectures and could improve prediction quality further.

C. RT Classifiers

The "Multi-class" modelling problem was considerably more difficult, as it specifically isolates the regions that we believe to be most difficult to predict into their own category, which is also the minority category. Similar to the "Binary" classifier, the performance for accurately predicting "Early"/"Late" regions is still high, with strong ROC and PR curves, resulting in an overall balanced accuracy of 0.753. However, the true "Middle" regions are consistently mis-classified into the "Early" and "Late" categories, with the regions that are predicted as "Middle" drawn nearly uniformly from the true classes. This would suggest that from the perspective of this LightGBM model trained on this dataset of features, the regions that have a core RT value near 0 are significantly harder to distinguish from adjacent "Early"/"Late" regions. This, coupled with similar difficulties shown by the regression model, could suggest that the concept of core sequence driven RT is not easy to accurately define for regions of the genome that reside consistently in TTRs. Consequently, for future work in this area we would suggest that the absolute RT value could be interpreted as a confidence that the region is "Early"/"Late", and that Bayesian architectures or `conformal predictors` that explicitly model data-point uncertainty while carrying out classifications could lead to improved and more interpretable performance.

References

1. Tyson, J. J. & Novak, B. A Dynamical Paradigm for Molecular Cell Biology. *Trends in Cell Biology* **30**, 504–515 (2020).

2. Levy, S. *et al.* The Diploid Genome Sequence of an Individual Human. *PLoS Biology* **5**, e254 (2007).

3. Soifer, llya *et al.* Fully Phased Sequence of a Diploid Human Genome Determined *de Novo* from the DNA of a Single Individual. *G3 Genes\textbarGenomes\textbarGenetics* **10**, 2911–2925 (2020).

4. Vellai, T. & Vida, G. The origin of eukaryotes: The difference between prokaryotic and eukaryotic cells. *Proceedings of the Royal Society of London. Series B: Biological Sciences* **266**, 1571–1577 (1999).

5. McInerney, J. O. & O'Connell, M. J. Mind the gaps in cellular evolution. *Nature* **541**, 297–299 (2017).

6. Zaremba-Niedzwiedzka, K. *et al.* Asgard archaea illuminate the origin of eukaryotic cellular complexity. *Nature* **541**, 353–358 (2017).

7. Nam, I., Nam, H. G. & Zare, R. N. Abiotic synthesis of purine and pyrimidine ribonucleosides in aqueous microdroplets. *Proceedings of the National Academy of Sciences* **115**, 36–40 (2018).

8. Pauling, L. & Corey, R. B. A Proposed Structure For The Nucleic Acids. *Proceedings of the National Academy of Sciences* **39**, 84–97 (1953).

9. Franklin, R. E. & Gosling, R. G. The structure of sodium thymonucleate fibres. I. The influence of water content. *Acta Crystallographica* **6**, 673–677 (1953).

10. Watson, J. D. & Crick, F. H. C. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature* **171**, 737–738 (1953).

11. Oyelade, J. *et al.* Overview of the human genome. in *Genome Plasticity in Health and Disease* 9–26 (Elsevier, 2020). doi:10.1016/B978-0-12-817819-5.00002-4.

12. Ravichandran, S., Subramani, V. K. & Kim, K. K. Z-DNA in the genome: From structure to disease. *Biophysical Reviews* **11**, 383–387 (2019).

13. Marsico, G. *et al.* Whole genome experimental maps of DNA G-quadruplexes in multiple species. *Nucleic Acids Research* **47**, 3862–3874 (2019).

14. Cheng, M. *et al.* Thermal and pH Stabilities of i-DNA: Confronting in vitro Experiments with Models and In-Cell NMR Data. *Angewandte Chemie International Edition* **60**, 10286–10294 (2021).

15. Privalov, P. L. & Crane-Robinson, C. Forces maintaining the DNA double helix and its complexes with transcription factors. *Progress in Biophysics and Molecular Biology* **135**, 30–48 (2018).

16. Cazenille, L., Baccouche, A. & Aubert-Kato, N. Automated exploration of DNA-based structure self-assembly networks. *Royal Society Open Science* **8**, 210848 (2021).

17. Kujirai, T. & Kurumizaka, H. Transcription through the nucleosome. *Current Opinion in Structural Biology* **61**, 42–49 (2020).

18. Misteli, T. The Self-Organizing Genome: Principles of Genome Architecture and Function. *Cell* **183**, 28–45 (2020).

19. Dekker, J. & Misteli, T. Long-Range Chromatin Interactions. *Cold Spring Harbor Perspectives in Biology* **7**, a019356 (2015).

20. Aljahani, A. *et al.* Analysis of sub-kilobase chromatin topology reveals nano-scale regulatory interactions with variable dependence on cohesin and CTCF. *Nature Communications* **13**, 2139 (2022).

21. Rowley, M. J. & Corces, V. G. Organizational principles of 3D genome architecture. *Nature Reviews Genetics* **19**, 789–800 (2018).

22. Klemm, S. L., Shipony, Z. & Greenleaf, W. J. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics* **20**, 207–220 (2019).

23. Ou, H. D. *et al.* ChromEMT: Visualizing 3D chromatin structure and compaction in interphase and mitotic cells. *Science* **357**, eaag0025 (2017).

24. Oldach, P. & Nieduszynski, C. A. Cohesin-Mediated Genome Architecture Does Not Define DNA Replication Timing Domains. *Genes* **10**, 196 (2019).

25. Masny, P. S. *et al.* Analysis of allele-specific RNA transcription in FSHD by RNA-DNA FISH in single myonuclei. *European Journal of Human Genetics* **18**, 448–456 (2010).

26. Ben-David, U. & Amon, A. Context is everything: Aneuploidy in cancer. *Nature Reviews Genetics* **21**, 44–62 (2020).

27. Li, D. J. Distributional features of triplet codons in genomes underlie the diversification of life. *Biosystems* **217**, 104681 (2022).

28. Stevens, T. J. *et al.* 3D structures of individual mammalian genomes studied by single-cell Hi-C. *Nature* **544**, 59–64 (2017).

29. Di Stefano, M., Paulsen, J., Jost, D. & Marti-Renom, M. A. 4D nucleome modeling. *Current Opinion in Genetics & Development* **67**, 25–32 (2021).

30. Corces, M. R. *et al.* An improved ATAC-seq protocol reduces background and enables interrogation of frozen tissues. *Nature methods* **14**, 959–962 (2017).

31. Lu, R. J.-H. *et al.* ATACgraph: Profiling Genome-Wide Chromatin Accessibility From ATAC-seq. *Frontiers in Genetics* **11**, 618478 (2021).

32. Reske, J. J., Wilson, M. R. & Chandler, R. L. ATAC-seq normalization method can significantly affect differential accessibility analysis and interpretation. *Epigenetics & Chromatin* **13**, 22 (2020).

33. Makai, D., Cseh, A., Sepsi, A. & Makai, S. A Multigraph-Based Representation of Hi-C Data. *Genes* **13**, 2189 (2022).

34. Zhang, K. *et al.* A single-cell atlas of chromatin accessibility in the human genome. *Cell* **184**, 5985–6001.e19 (2021).

35. Dekker, J. *et al.* Spatial and temporal organization of the genome: Current state and future aims of the 4D nucleome project. *Molecular Cell* S1097276523004653 (2023) doi:10.1016/j.molcel.2023.06.018.

36. Schoenfelder, S. & Fraser, P. Long-range enhancer–promoter contacts in gene expression control. *Nature Reviews Genetics* **20**, 437–455 (2019).

37. Robson, M. I., Ringel, A. R. & Mundlos, S. Regulatory Landscaping: How Enhancer-Promoter Communication Is Sculpted in 3D. *Molecular Cell* **74**, 1110–1122 (2019).

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

39. Talbert, P. B., Meers, M. P. & Henikoff, S. Old cogs, new tricks: The evolution of gene expression in a chromatin context. *Nature Reviews Genetics* **20**, 283–297 (2019).

40. Nicholls, T. J. & Gustafsson, C. M. Separating and Segregating the Human Mitochondrial Genome. *Trends in Biochemical Sciences* **43**, 869–881 (2018).

41. Schneider, V. A. *et al.* Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Research* **27**, 849–864 (2017).

42. Halldorsson, B. V. *et al.* The sequences of 150,119 genomes in the UK Biobank. *Nature* **607**, 732–740 (2022).

43. Sherman, R. M. & Salzberg, S. L. Pan-genomics in the human genome era. *Nature Reviews Genetics* **21**, 243–254 (2020).

44. Wong, K. H. Y. *et al.* Towards a reference genome that captures global genetic diversity. *Nature Communications* **11**, 5482 (2020).

45. Liao, W.-W. *et al.* A draft human pangenome reference. *Nature* **617**, 312–324 (2023).

46. Ebler, J. *et al.* Pangenome-based genome inference allows efficient and accurate genotyping across a wide spectrum of variant classes. *Nature Genetics* **54**, 518–525 (2022).

47. Jarvis, E. D. *et al.* Semi-automated assembly of high-quality diploid human reference genomes. *Nature* **611**, 519–531 (2022).

48. Chargaff, E. Chemical specificity of nucleic acids and mechanism of their enzymatic degradation. *Experientia* **6**, 201–209 (1950).

49. Fariselli, P., Taccioli, C., Pagani, L. & Maritan, A. DNA sequence symmetries from randomness: The origin of the Chargaff's second parity rule. *Briefings in Bioinformatics* **22**, 2172–2181 (2021).

50. Mitchell, D. & Bridge, R. A test of Chargaff's second rule. *Biochemical and Biophysical Research Communications* **340**, 90–94 (2006).

51. Pflughaupt, P. & Sahakyan, A. B. Generalised interrelations among mutation rates drive the genomic compliance of Chargaff's second parity rule. *Nucleic Acids Research* gkad477 (2023) doi:10.1093/nar/gkad477.

52. Courel, M. *et al.* GC content shapes mRNA storage and decay in human cells. *eLife* **8**, e49708 (2019).

53. Cozzi, P., Milanesi, L. & Bernardi, G. Segmenting the Human Genome into Isochores. *Evolutionary Bioinformatics* **11**, EBO.S27693 (2015).

54. International Human Genome Sequencing Consortium *et al.* Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001).

55. Deininger, P. Alu elements: Know the SINEs. *Genome Biology* **12**, 236 (2011).

56. Nurk, S. *et al.* The complete sequence of a human genome. *Science* **376**, 44–53 (2022).

57. Barrett, L. W., Fletcher, S. & Wilton, S. D. Regulation of eukaryotic gene expression by the untranslated gene regions and other non-coding elements. *Cellular and Molecular Life Sciences* **69**, 3613–3634 (2012).

58. Lim, C. S., T. Wardell, S. J., Kleffmann, T. & Brown, C. M. The exon–intron gene structure upstream of the initiation codon predicts translation efficiency. *Nucleic Acids Research* **46**, 4575–4591 (2018).

59. Wada, Y. *et al.* A wave of nascent transcription on activated human genes. *Proceedings of the National Academy of Sciences* **106**, 18357–18361 (2009).

60. Uyehara, C. M. & Apostolou, E. 3D enhancer-promoter interactions and multi-connected hubs: Organizational principles and functional roles. *Cell Reports* **42**, 112068 (2023).

61. Jolma, A. *et al.* DNA-Binding Specificities of Human Transcription Factors. *Cell* **152**, 327–339 (2013).

62. Doni Jayavelu, N., Jajodia, A., Mishra, A. & Hawkins, R. D. *An atlas of silencer elements for the human and mouse genomes.* (2018) doi:10.1101/252304.

63. Wasserman, W. W. & Sandelin, A. Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics* **5**, 276–287 (2004).

64. Ganapathiraju, M. K., Subramanian, S., Chaparala, S. & Karunakaran, K. B. A reference catalog of DNA palindromes in the human genome and their variations in 1000 Genomes. *Human Genome Variation* **7**, 40 (2020).

65. Rothbart, S. B. & Strahl, B. D. Interpreting the language of histone and DNA modifications. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* **1839**, 627–643 (2014).

66. Zhang, P., Torres, K., Liu, X., Liu, C.-G. & Pollock, R. E. An Overview of Chromatin-Regulating Proteins in Cells. *Current Protein & Peptide Science* **17**, 401–410 (2016).

67. Hörnblad, A. & Remeseiro, S. Epigenetics, Enhancer Function and 3D Chromatin Organization in Reprogramming to Pluripotency. *Cells* **11**, 1404 (2022).

68. Oksuz, O. *et al.* Transcription factors interact with RNA to regulate genes. *Molecular Cell* S1097276523004343 (2023) doi:10.1016/j.molcel.2023.06.012.

69. Hsu, T.-K. *et al.* A general calculus of fitness landscapes finds genes under selection in cancers. 43.

70. Karr, J. R. *et al.* A Whole-Cell Computational Model Predicts Phenotype from Genotype. *Cell* **150**, 389–401 (2012).

71. Tyson, J. J. & Novak, B. Regulation of the Eukaryotic Cell Cycle: Molecular Antagonism, Hysteresis, and Irreversible Transitions. *Journal of Theoretical Biology* **210**, 249–263 (2001).

72. Hayakawa, T., Suzuki, R., Kagotani, K., Okumura, K. & Takebayashi, S. Camptothecin-Induced Replication Stress Affects DNA Replication Profiling by E/L Repli-Seq. *Cytogenetic and Genome Research* **161**, 437–444 (2021).

73. Stallaert, W. *et al.* The structure of the human cell cycle. *Cell Systems* **13**, 230–240.e3 (2022).

74. Cornwell, J. A. *et al.* Loss of CDK4/6 activity in S/G2 phase leads to cell cycle reversal. *Nature* (2023) doi:10.1038/s41586-023-06274-3.

75. Greenberg, A. & Simon, I. S Phase Duration Is Determined by Local Rate and Global Organization of Replication. *Biology* **11**, 718 (2022).

76. Hossain, M., Bhalla, K. & Stillman, B. *Cyclin binding Cy motifs have multiple activities in the initiation of DNA replication.* (2019) doi:10.1101/681668.

77. Hu, Y. *et al.* Evolution of DNA replication origin specification and gene silencing mechanisms. *Nature Communications* **11**, 5175 (2020).

78. Hu, Y. & Stillman, B. Origins of DNA replication in eukaryotes. *Molecular Cell* **83**, 352–372 (2023).

79. Li, S. *et al.* Nucleosome-directed replication origin licensing independent of a consensus DNA sequence. *Nature Communications* **13**, 4947 (2022).

80. Fragkos, M., Ganier, O., Coulombe, P. & Méchali, M. DNA replication origin activation in space and time. *Nature Reviews Molecular Cell Biology* **16**, 360–374 (2015).

81. Miotto, B. & Struhl, K. HBO1 Histone Acetylase Activity Is Essential for DNA Replication Licensing and Inhibited by Geminin. *Molecular Cell* **37**, 57–66 (2010).

82. Besnard, E. *et al.* Unraveling cell type–specific and reprogrammable human replication origin signatures associated with G-quadruplex consensus motifs. *Nature Structural & Molecular Biology* **19**, 837–844 (2012).

83. Prioleau, M.-N. & MacAlpine, D. M. DNA replication origins—where do we begin? *Genes & Development* **30**, 1683–1697 (2016).

84. Mesner, L. D. *et al.* Bubble-seq analysis of the human genome reveals distinct chromatin-mediated mechanisms for regulating early- and late-firing origins. *Genome Research* **23**, 1774–1788 (2013).

85. Hyrien, O. Peaks cloaked in the mist: The landscape of mammalian replication origins. *Journal of Cell Biology* **208**, 147–160 (2015).

86. Cayrou, C. *et al.* The chromatin environment shapes DNA replication origin organization and defines origin classes. *Genome Research* **25**, 1873–1885 (2015).

87. Marsolier-Kergoat, M.-C. Asymmetry Indices for Analysis and Prediction of Replication Origins in Eukaryotic Genomes. *PLoS ONE* **7**, e45050 (2012).

88. Kirstein, N. *et al. Active transcription regulates ORC/MCM distribution whereas replication timing correlates with ORC density in human cells.* (2019) doi:10.1101/778423.

89. Gilbert, D. M. Replication licensing during S phase: Breaking the law to prevent breaking DNA. *Nature Structural & Molecular Biology* **30**, 406–408 (2023).

90. Masai, H. Replicon hypothesis revisited. *Biochemical and Biophysical Research Communications* **633**, 77–80 (2022).

91. Yeeles, J. T. P., Janska, A., Early, A. & Diffley, J. F. X. How the Eukaryotic Replisome Achieves Rapid and Efficient DNA Replication. *Molecular Cell* **65**, 105–116 (2017).

92. Costa, A. & Diffley, J. F. X. The Initiation of Eukaryotic DNA Replication. *Annual Review of Biochemistry* **91**, 107–131 (2022).

93. Gilbert, D. M. Replication origins run (ultra) deep. *Nature Structural & Molecular Biology* **19**, 740–742 (2012).

94. Yeeles, J. T. P., Deegan, T. D., Janska, A., Early, A. & Diffley, J. F. X. Regulated eukaryotic DNA replication origin firing with purified proteins. *Nature* **519**, 431–435 (2015).

95. Wittig, K. A., Sansam, C. G., Noble, T. D., Goins, D. & Sansam, C. L. The CRL4DTL E3 ligase induces degradation of the DNA replication initiation factor TICRR/TRESLIN specifically during S phase. *Nucleic Acids Research* **49**, 10507–10523 (2021).

96. Boos, D., Yekezare, M. & Diffley, J. F. X. Identification of a Heteromeric Complex That Promotes DNA Replication Origin Firing in Human Cells. *Science* **340**, 981–984 (2013).

97. Chagin, V. O. *et al.* 4D Visualization of replication foci in mammalian cells corresponding to individual replicons. *Nature Communications* **7**, 11231 (2016).

98. Wu, X. *et al.* Genome-wide measurement of DNA replication fork directionality and quantification of DNA replication initiation and termination with Okazaki fragment sequencing. *Nature Protocols* **18**, 1260–1295 (2023).

99. Tubbs, A. *et al.* Dual Roles of Poly(dA:dT) Tracts in Replication Initiation and Fork Collapse. *Cell* **174**, 1127–1142.e19 (2018).

100. Kemp, M. G. The histone deacetylase inhibitor trichostatin A alters the pattern of DNA replication origin activity in human cells. *Nucleic Acids Research* **33**, 325–336 (2005).

101. van den Berg, J., van Batenburg, V. & van Oudenaarden, A. *Acceleration of genome replication uncovered by single-cell nascent DNA sequencing.* (2022) doi:10.1101/2022.12.13.520365.

102. Dovrat, D. *et al.* A Live-Cell Imaging Approach for Measuring DNA Replication Rates. *Cell Reports* **24**, 252–258 (2018).

103. Jun, S. & Bechhoefer, J. Nucleation and growth in one dimension. II. Application to DNA replication kinetics. *Physical Review E* **71**, 011909 (2005).

104. Löb, D. *et al.* 3D replicon distributions arise from stochastic initiation and domino-like DNA replication progression. *Nature Communications* **7**, 11207 (2016).

105. Lee, J. H. & Berger, J. M. Cell Cycle-Dependent Control and Roles of DNA Topoisomerase II. *Genes* **10**, 859 (2019).

106. Koyanagi, E. *et al.* Global landscape of replicative DNA polymerase usage in the human genome. *Nature Communications* **13**, 7221 (2022).

107. Raia, P., Delarue, M. & Sauguet, L. An updated structural classification of replicative DNA polymerases. *Biochemical Society Transactions* **47**, 239–249 (2019).

108. Wu, W., Hickson, I. D. & Liu, Y. The prevention and resolution of DNA replication–transcription conflicts in eukaryotic cells. *Genome Instability & Disease* **1**, 114–128 (2020).

109. Mukherjee, C. *et al.* RIF1 promotes replication fork protection and efficient restart to maintain genome stability. *Nature Communications* **10**, 3287 (2019).

110. Reusswig, K.-U. *et al.* *Unscheduled DNA replication in G1 causes genome instability through head-to-tail replication fork collisions.* (2021) doi:10.1101/2021.09.06.459115.

111. Kawabata, T. *et al.* Stalled Fork Rescue via Dormant Replication Origins in Unchallenged S Phase Promotes Proper Chromosome Segregation and Tumor Suppression. *Molecular Cell* **41**, 543–553 (2011).

112. Charrasse, S. *et al.* Ensa controls S-phase length by modulating Treslin levels. *Nature Communications* **8**, 206 (2017).

113. Vouzas, A. E. & Gilbert, D. M. Replication timing and transcriptional control: Beyond cause and effect — part IV. *Current Opinion in Genetics & Development* **79**, 102031 (2023).

114. Rhind, N. DNA replication timing: Biochemical mechanisms and biological significance. *BioEssays* **44**, 2200097 (2022).

115. Wang, Y., Alangari, M., Hihath, J., Das, A. K. & Anantram, M. P. A machine learning approach for accurate and real-time DNA sequence identification. *BMC Genomics* **22**, 525 (2021).

116. Gindin, Y., Valenzuela, M. S., Aladjem, M. I., Meltzer, P. S. & Bilke, S. A chromatin structure-based model accurately predicts DNA replication timing in human cells. *Molecular Systems Biology* **10**, 722 (2014).

117. Hiratani, I., Leskovar, A. & Gilbert, D. M. Differentiation-induced replication-timing changes are restricted to AT-rich/long interspersed nuclear element (LINE)-rich isochores. *Proceedings of the National Academy of Sciences* **101**, 16861–16866 (2004).

118. Woodfine, K. *et al.* Replication timing of the human genome. *Human Molecular Genetics* **13**, 191–202 (2004).

119. Hiratani, I. *et al.* Global Reorganization of Replication Domains During Embryonic Stem Cell Differentiation. *PLoS Biology* **6**, e245 (2008).

120. Dileep, V., Didier, R. & Gilbert, D. M. Genome-wide analysis of replication timing in mammalian cells: Troubleshooting problems encountered when comparing different cell types. *Methods* **57**, 165–169 (2012).

121. Zhao, P. A., Sasaki, T. & Gilbert, D. M. High-resolution Repli-Seq defines the temporal choreography of initiation, elongation and termination of replication in mammalian cells. *Genome Biology* **21**, 76 (2020).

122. Massey, D. J. & Koren, A. *Telomere-to-telomere human DNA replication timing profiles.* (2022) doi:10.1101/2022.03.28.486072.

123. Sakamoto, M. *et al.* scRepli-Seq: A Powerful Tool to Study Replication Timing and Genome Instability. *Cytogenetic and Genome Research* **162**, 161–170 (2022).

124. Stewart-Morgan, K. R. & Groth, A. Profiling Chromatin Accessibility on Replicated DNA with repli-ATAC-Seq. in *Chromatin Accessibility* (eds. Marinov, G. K. & Greenleaf, W. J.) vol. 2611 71–84 (Springer US, 2023).

125. Koren, A., Massey, D. J. & Bracci, A. N. TIGER: Inferring DNA replication timing from whole-genome sequence data. *Bioinformatics* btab166 (2021) doi:10.1093/bioinformatics/btab166.

126. Zynda, G. J. *et al.* Repliscan: A tool for classifying replication timing regions. *BMC Bioinformatics* **18**, 362 (2017).

127. Pope, B. D. *et al.* Topologically associating domains are stable units of replication-timing regulation. *Nature* **515**, 402–405 (2014).

128. Berezney, R., Dubey, D. D. & Huberman, J. A. Heterogeneity of eukaryotic replicons, replicon clusters, and replication foci. *Chromosoma* **108**, 471–484 (2000).

129. Marchal, C., Sima, J. & Gilbert, D. M. Control of DNA replication timing in the 3D genome. *Nature Reviews Molecular Cell Biology* **20**, 721–737 (2019).

130. Ryba, T. *et al.* Evolutionarily conserved replication timing profiles predict long-range chromatin interactions and distinguish closely related cell types. *Genome Research* **20**, 761–770 (2010).

131. Hiratani, I. & Gilbert, D. M. Replication timing as an epigenetic mark. *Epigenetics* **4**, 93–97 (2009).

132. Sasaki, T. *et al.* Stability of patient-specific features of altered DNA replication timing in xenografts of primary human acute lymphoblastic leukemia. *Experimental Hematology* **51**, 71–82.e3 (2017).

133. Bergman, Y., Simon, I. & Cedar, H. Asynchronous Replication Timing: A Mechanism for Monoallelic Choice During Development. *Frontiers in Cell and Developmental Biology* **9**, 737681 (2021).

134. Rivera-Mulia, J. C. *et al.* Allele-specific control of replication timing and genome organization during development. *Genome Research* **28**, 800–811 (2018).

135. Edwards, M. M., Wang, N., Massey, D. J., Egli, D. & Koren, A. *Incomplete Reprogramming of DNA Replication Timing in Induced Pluripotent Stem Cells*. (2023) doi:10.1101/2023.06.12.544654.

136. Rivera-Mulia, J. C. *et al. Replication Timing Networks: A novel class of gene regulatory networks*. (2017) doi:10.1101/186866.

137. Santos, M. M., Johnson, M. C., Fiedler, L. & Zegerman, P. Global early replication disrupts gene expression and chromatin conformation in a single cell cycle. *Genome Biology* **23**, 217 (2022).

138. Edwards, M. M. *et al.* Delayed DNA replication in haploid human embryonic stem cells. *Genome Research* **31**, 2155–2169 (2021).

139. Rivera-Mulia, J. C. *et al.* Cellular senescence induces replication stress with almost no affect on DNA replication timing. *Cell Cycle (Georgetown, Tex.)* **17**, 1667–1681 (2018).

140. Courtot, L. *et al.* Low Replicative Stress Triggers Cell-Type Specific Inheritable Advanced Replication Timing. *International Journal of Molecular Sciences* **22**, 4959 (2021).

141. Peycheva, M. *et al.* DNA replication timing directly regulates the frequency of oncogenic chromosomal translocations. *Science* **377**, eabj5502 (2022).

142. Ryba, T. *et al.* Abnormal developmental control of replication-timing domains in pediatric acute lymphoblastic leukemia. *Genome Research* **22**, 1833–1844 (2012).

143. Du, Q. *et al.* Replication timing and epigenome remodelling are associated with the nature of chromosomal rearrangements in cancer. *Nature Communications* **10**, 416 (2019).

144. Lu, J., Li, F., Murphy, C. S., Davidson, M. W. & Gilbert, D. M. G2 phase chromatin lacks determinants of replication timing. *Journal of Cell Biology* **189**, 967–980 (2010).

145. Kanoh, Y., Ueno, M., Hayano, M., Kudo, S. & Masai, H. Aberrant association of chromatin with nuclear periphery induced by Rif1 leads to mitotic defect. *Life Science Alliance* **6**, e202201603 (2023).

146. Cornacchia, D. *et al.* Mouse Rif1 is a key regulator of the replication-timing programme in mammalian cells: Mouse Rif1 controls replication timing. *The EMBO Journal* **31**, 3678–3690 (2012).

147. Richards, L., Das, S. & Nordman, J. T. Rif1-Dependent Control of Replication Timing. *Genes* **13**, 550 (2022).

148. Blasiak, J., Szczepańska, J., Sobczuk, A., Fila, M. & Pawlowska, E. RIF1 Links Replication Timing with Fork Reactivation and DNA Double-Strand Break Repair. *International Journal of Molecular Sciences* **22**, 11440 (2021).

149. Klein, K. N. *et al.* Replication timing maintains the global epigenetic state in human cells. 9 (2021).

150. Yamazaki, S. *et al.* Rif1 regulates the replication timing domains on the human genome: Rif1 regulates the replication timing domains. *The EMBO Journal* **31**, 3667–3677 (2012).

151. Alavi, S. *et al.* G-quadruplex binding protein Rif1, a key regulator of replication timing. *The Journal of Biochemistry* **169**, 1–14 (2021).

152. Foti, R. *et al.* Nuclear Architecture Organized by Rif1 Underpins the Replication-Timing Program. *Molecular Cell* **61**, 260–273 (2016).

153. Aparicio, O. M. Location, location, location: It's all in the timing for replication origins. *Genes & Development* **27**, 117–128 (2013).

154. Sima, J. *et al.* Identifying cis Elements for Spatiotemporal Control of Mammalian DNA Replication. *Cell* **176**, 816–830.e18 (2019).

155. Yaffe, E. *et al.* Comparative Analysis of DNA Replication Timing Reveals Conserved Large-Scale Chromosomal Architecture. *PLoS Genetics* **6**, e1001011 (2010).

156. Thayer, M. J., Heskett, M. B., Smith, L. G., Spellman, P. T. & Yates., P. A. *ASAR lncRNAs control DNA replication timing through interactions with multiple hnRNP/RNA binding proteins.* (2022) doi:10.1101/2022.06.04.494840.

157. Heskett, M. B., Smith, L. G., Spellman, P. & Thayer, M. J. Reciprocal monoallelic expression of ASAR lncRNA genes controls replication timing of human chromosome 6. *RNA* **26**, 724–738 (2020).

158. Heskett, M. B. *et al.* Epigenetic control of chromosome-associated lncRNA genes essential for replication and stability. *Nature Communications* **13**, 6301 (2022).

159. Stoffregen, E. P., Donley, N., Stauffer, D., Smith, L. & Thayer, M. J. An autosomal locus that controls chromosome-wide replication timing and mono-allelic expression. *Human Molecular Genetics* **20**, 2366–2378 (2011).

160. Liu, L., De, S. & Michor, F. DNA replication timing and higher-order nuclear organization determine single-nucleotide substitution patterns in cancer genomes. *Nature Communications* **4**, 1502 (2013).

161. Yehuda, Y. *et al.* Germline DNA replication timing shapes mammalian genome composition. *Nucleic Acids Research* **46**, 8299–8310 (2018).

162. Ding, Q. *et al. The Genetic Architecture of DNA Replication Timing in Human Pluripotent Stem Cells.* (2020) doi:10.1101/2020.05.08.085324.

163. Sansam, C. G. *et al.* A mechanism for epigenetic control of DNA replication. *Genes & Development* **32**, 224–229 (2018).

164. Miotto, B. & Struhl, K. HBO1 histone acetylase is a coactivator of the replication licensing factor Cdt1. *Genes & Development* **22**, 2633–2638 (2008).

165. Van Rechem, C. *et al.* Collective regulation of chromatin modifications predicts replication timing during cell cycle. *Cell Reports* **37**, 109799 (2021).

166. Eaton, M. L. *et al.* Chromatin signatures of the Drosophila replication program. *Genome Research* **21**, 164–174 (2011).

167. Chandra, T. *et al.* Independence of repressive histone marks and chromatin compaction during senescent heterochromatic layer formation. *Molecular Cell* **47**, 203–214 (2012).

168. Shoaib, M. *et al.* Histone H4K20 methylation mediated chromatin compaction threshold ensures genome integrity by limiting DNA replication licensing. *Nature Communications* **9**, 3704 (2018).

169. Takebayashi, S. *et al.* The Temporal Order of DNA Replication Shaped by Mammalian DNA Methyltransferases. *Cells* **10**, 266 (2021).

170. Du, Q. *et al.* DNA methylation is required to maintain both DNA replication timing precision and 3D genome organization integrity. *Cell Reports* **36**, 109722 (2021).

171. Jun, S., Zhang, H. & Bechhoefer, J. Nucleation and growth in one dimension, part I: The generalized Kolmogorov-Johnson-Mehl-Avrami model. *Physical Review E* **71**, 011908 (2005).

172. Hyrien, O. & Goldar, A. Mathematical modelling of eukaryotic DNA replication. *Chromosome Research* **18**, 147–161 (2010).

173. Gindin, Y., Meltzer, P. S. & Bilke, S. Replicon: A software to accurately predict DNA replication timing in metazoan cells. *Frontiers in Genetics* **5**, (2014).

174. Yang, Y., Wang, Y., Zhang, Y. & Ma, J. Concert: Genome-wide prediction of sequence elements that modulate DNA replication timing. 23.

175. R Core Team. *R: A language and environment for statistical computing*. (R Foundation for Statistical Computing, 2022).

176. Paszke, A. *et al.* Automatic differentiation in PyTorch. (2017).

177. Durand, P., Mahé, F., Valin, A.-S. & Nicolas, J. Browsing repeats in genomes: Pygram and an application to non-coding region analysis. *BMC Bioinformatics* **7**, 477 (2006).

178. Maguire, E., Rocca-Serra, P., Sansone, S.-A. & Chen, M. Redesigning the Sequence Logo with Glyph-based Approaches to Aid Interpretation. *EuroVis - Short Papers* 5 pages (2014) doi:10.2312/EUROVISSHORT.20141159.

179. Bonidia, R. P., Domingues, D. S. & Sanches, D. S. MathFeature: Feature extraction package for DNA, RNA and protein sequences based on mathematical descriptors. *Briefings in Bioinformatics* 10 (2022).

180. IUPAC. Abbreviations and symbols for nucleic acids, polynucleotides, and their constituents. *Biochemistry* **9**, 4022–4027 (1970).

181. Voss, R. F. Evolution of long-range fractal correlations and $1/f$ noise in DNA base sequences. *Physical Review Letters* **68**, 3805–3808 (1992).

182. Kotlar, D. & Lavner, Y. Gene Prediction by Spectral Rotation Measure: A New Method for Identifying Protein-Coding Regions. *Genome Research* **13**, 1930–1937 (2003).

183. Pich, O. *et al.* Somatic and Germline Mutation Periodicity Follow the Orientation of the DNA Minor Groove around Nucleosomes. *Cell* **175**, 1074–1087.e18 (2018).

184. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed Representations of Words and Phrases and their Compositionality. 9.

185. Shi, L. & Chen, B. LSHvec: A vector representation of DNA sequences using locality sensitive hashing and FastText word embeddings. in *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics* 1–10 (ACM, 2021). doi:10.1145/3459930.3469521.

186. Ng, P. Dna2vec: Consistent vector representations of variable-length k-mers. *arXiv:1701.06279 [cs, q-bio, stat]* (2017).

187. Asgari, E. & Mofrad, M. R. K. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLOS ONE* **10**, e0141287 (2015).

188. Ali, S., Sardar, U., Patterson, M. & Khan, I. U. BioSequence2Vec: Efficient Embedding Generation For Biological Sequences. (2023).

189. Nguyen, E. *et al.* HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution. (2023).

190. Deng, M., Yu, C., Liang, Q., He, R. L. & Yau, S. S.-T. A Novel Method of Characterizing Genetic Sequences: Genome Space with Biological Distance and Applications. *PLoS ONE* **6**, e17293 (2011).

191. Lee, D. LS-GKM: A new gkm-SVM for large-scale datasets. *Bioinformatics* **32**, 2196–2198 (2016).

192. Muhammod, R. *et al.* PyFeat: A Python-based effective feature generation tool for DNA, RNA and protein sequences. *Bioinformatics* **35**, 3831–3833 (2019).

193. Mao, G. Association Matrix Method and Its Applications in Mining DNA Sequences. in *Advances in Artificial Intelligence, Software and Systems Engineering* (ed. Ahram, T.) vol. 965 154–159 (Springer International Publishing, 2020).

194. Dong, R., He, L., He, R. L. & Yau, S. S.-T. A Novel Approach to Clustering Genome Sequences Using Inter-nucleotide Covariance. *Frontiers in Genetics* **10**, 234 (2019).

195. Zheng, J., Ramasinghe, S. & Lucey, S. Rethinking Positional Encoding. *arXiv:2107.02561 [cs]* (2021).

196. Liu, Z. *et al.* Towards Understanding Grokking: An Effective Theory of Representation Learning. (2022) doi:10.48550/ARXIV.2205.10343.

197. Novakovsky, G., Dexter, N., Libbrecht, M. W., Wasserman, W. W. & Mostafavi, S. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics* (2022) doi:10.1038/s41576-022-00532-2.

198. Ploenzke, M. & Irizarry, R. *Interpretable Convolution Methods for Learning Genomic Sequence Motifs.* (2018) doi:10.1101/411934.

199. Koo, P. K. & Eddy, S. R. Representation learning of genomic sequence motifs with convolutional neural networks. *PLOS Computational Biology* **15**, e1007560 (2019).

200. Shrikumar, A., Greenside, P., Shcherbina, A. & Kundaje, A. Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. (2017).

201. Brown, R. C. & Lunter, G. An equivariant Bayesian convolutional network predicts recombination hotspots and accurately resolves binding motifs. *Bioinformatics* **35**, 2177–2184 (2019).

202. Zhou, H., Shrikumar, A. & Kundaje, A. *Towards a Better Understanding of Reverse-Complement Equivariance for Deep Learning Models in Regulatory Genomics.* (2020) doi:10.1101/2020.11.04.368803.

203. Shrikumar, A., Greenside, P. & Kundaje, A. Learning Important Features Through Propagating Activation Differences. (2019).

204. Avsec, Ž. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics* **53**, 354–366 (2021).

205. Liu, G., Zeng, H. & Gifford, D. K. Visualizing complex feature interactions and feature sharing in genomic deep neural networks. *BMC Bioinformatics* **20**, 401 (2019).

206. Wei, Z. *et al.* NeuronMotif: Deciphering cis-regulatory codes by layer-wise demixing of deep neural networks. *Proceedings of the National Academy of Sciences* **120**, e2216698120 (2023).

207. Koo, P. K. & Ploenzke, M. *Interpreting Deep Neural Networks Beyond Attribution Methods: Quantifying Global Importance of Genomic Features.* (2020) doi:10.1101/2020.02.19.956896.

208. Toneyan, S. & Koo, P. K. Interpreting Cis-Regulatory Interactions from Large-Scale Deep Neural Networks for Genomics.

209. Schreiber, J., Nair, S., Balsubramani, A. & Kundaje, A. Accelerating in-silico saturation mutagenesis using compressed sensing. (2021).

210. Nair, S., Shrikumar, A., Schreiber, J. & Kundaje, A. fastISM: Performant *in Silico* saturation mutagenesis for convolutional neural networks. *Bioinformatics* **38**, 2397–2403 (2022).

211. Zhang, S. *et al.* Assessing deep learning methods in *Cis* -regulatory motif finding based on genomic sequencing data. *Briefings in Bioinformatics* **23**, bbab374 (2022).

212. Millán Arias, P., Alipour, F., Hill, K. A. & Kari, L. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. *PLOS ONE* **17**, e0261531 (2022).

213. Uddin, M., Islam, M. K., Hassan, Md. R., Jahan, F. & Baek, J. H. A fast and efficient algorithm for DNA sequence similarity identification. *Complex & Intelligent Systems* **9**, 1265–1280 (2023).

214. Akerman, I. *et al.* A predictable conserved DNA base composition signature defines human core DNA replication origins. *Nature Communications* **11**, 4826 (2020).

215. Do, D. T. & Le, N. Q. K. Using extreme gradient boosting to identify origin of replication in Saccharomyces cerevisiae via hybrid features. *Genomics* **112**, 2445–2451 (2020).

216. Wu, F., Yang, R., Zhang, C. & Zhang, L. A deep learning framework combined with word embedding to identify DNA replication origins. *Scientific Reports* **11**, 844 (2021).

217. Sahakyan, A. B. *et al.* Machine learning model for sequence-driven DNA G-quadruplex formation. *Scientific Reports* **7**, 14535 (2017).

218. Mourad, R., Ginalski, K., Legube, G. & Cuvier, O. Predicting double-strand DNA breaks using epigenome marks or DNA at kilobase resolution. *Genome Biology* **19**, 34 (2018).

219. Ranawana, R. & Palade, V. A neural network based multi-classifier system for gene identification in DNA sequences. *Neural Computing and Applications* **14**, 122–131 (2005).

220. De Almeida, B. P., Reiter, F., Pagani, M. & Stark, A. DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. *Nature Genetics* **54**, 613–624 (2022).

221. 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).

222. Hassanzadeh, H. R. & Wang, M. D. DeeperBind: Enhancing Prediction of Sequence Specificities of DNA Binding Proteins. 17 (2020).

223. Salekin, S., Zhang, J. M. & Huang, Y. Base-pair resolution detection of transcription factor binding site by deep deconvolutional network. *Bioinformatics* **34**, 3446–3453 (2018).

224. Brennan, K. J. *et al. Chromatin accessibility is a two-tier process regulated by transcription factor pioneering and enhancer activation.* (2022) doi:10.1101/2022.12.20.520743.

225. Miraldi, E. R., Chen, X. & Weirauch, M. T. Deciphering cis-regulatory grammar with deep learning. *Nature Genetics* **53**, 266–268 (2021).

226. Ullah, F. & Ben-Hur, A. A self-attention model for inferring cooperativity between regulatory features. *Nucleic Acids Research* **49**, e77–e77 (2021).

227. Dudnyk, K., Shi, C. & Zhou, J. *Sequence basis of transcription initiation in human genome.* (2023) doi:10.1101/2023.06.27.546584.

228. 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).

229. Balci, A. T., Ebeid, M. M., Benos, P. V., Kostka, D. & Chikina, M. *An intrinsically interpretable neural network architecture for sequence to function learning.* (2023) doi:10.1101/2023.01.25.525572.

230. Lal, A. *et al. AtacWorks: A deep convolutional neural network toolkit for epigenomics.* (2019) doi:10.1101/829481.

231. Tayyebi, Z., Pine, A. R. & Leslie, C. S. *Scalable sequence-informed embedding of single-cell ATAC-seq data with CellSpace.* (2022) doi:10.1101/2022.05.02.490310.

232. Lv, H. *et al.* A sequence-based deep learning approach to predict CTCF-mediated chromatin loop. *Briefings in Bioinformatics* bbab031 (2021) doi:10.1093/bib/bbab031.

233. Kuang, S. & Wang, L. Identification and analysis of consensus RNA motifs binding to the genome regulator CTCF. *NAR Genomics and Bioinformatics* **2**, lqaa031 (2020).

234. Salameh, T. J. *et al.* A supervised learning framework for chromatin loop detection in genome-wide contact maps. *Nature Communications* **11**, 3428 (2020).

235. Prost, J. A., Cameron, C. J. & Blanchette, M. *SACSANN: Identifying sequence-based determinants of chromosomal compartments.* (2020) doi:10.1101/2020.10.06.328039.

236. Zhou, J. Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. *Nature Genetics* **54**, 725–734 (2022).

237. Tan, J. *et al.* Cell-type-specific prediction of 3D chromatin organization enables high-throughput in silico genetic screening. *Nature Biotechnology* (2023) doi:10.1038/s41587-022-01612-8.

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

239. Al-jibury, E. *et al.* A deep learning method for replicate-based analysis of chromosome conformation contacts using Siamese neural networks. *Nature Communications* **14**, 5007 (2023).

240. Zheng, A. *et al.* Deep neural networks identify sequence context features predictive of transcription factor binding. *Nature Machine Intelligence* **3**, 172–180 (2021).

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

242. Avsec, Ž. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods* **18**, 1196–1203 (2021).

243. Schwessinger, R., Deasy, J., Woodruff, R. T., Young, S. & Branson, K. M. *Single-cell gene expression prediction from DNA sequence at large contexts.* (2023) doi:10.1101/2023.07.26.550634.

244. Chen, K. M., Wong, A. K., Troyanskaya, O. G. & Zhou, J. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature Genetics* **54**, 940–949 (2022).

245. Gresova, K., Martinek, V., Cechak, D., Simecek, P. & Alexiou, P. Genomic benchmarks: A collection of datasets for genomic sequence classification. *bioRxiv : the preprint server for biology* (2022).

246. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

247. Chen, Y., Li, Y., Narayan, R., Subramanian, A. & Xie, X. Gene expression inference with deep learning. *Bioinformatics* **32**, 1832–1839 (2016).

248. Ke, G. *et al.* LightGBM: A Highly Efficient Gradient Boosting Decision Tree. 9.

249. Anghel, A., Papandreou, N., Parnell, T., De Palma, A. & Pozidis, H. Benchmarking and Optimization of Gradient Boosting Decision Tree Algorithms. *arXiv:1809.04559 [cs, stat]* (2019).

250. Chen, K. M., Cofer, E. M., Zhou, J. & Troyanskaya, O. G. Selene: A PyTorch-based deep learning library for sequence data. *Nature Methods* **16**, 315–318 (2019).

251. Prakash, E., Shrikumar, A. & Kundaje, A. *Towards More Realistic Simulated Datasets for Benchmarking Deep Learning Models in Regulatory Genomics*. (2021) doi:10.1101/2021.12.26.474224.

252. Lee, N. K., Tang, Z., Toneyan, S. & Koo, P. K. EvoAug: Improving generalization and interpretability of genomic deep neural networks with evolution-inspired data augmentations. *Genome Biology* **24**, 105 (2023).

253. Budach, S. & Marsico, A. Pysster: Classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. *Bioinformatics* **34**, 3035–3037 (2018).

254. Koo, P. K., Ploenzke, M., Anand, P., Paul, S. & Majdandzic, A. ResidualBind: Uncovering Sequence-Structure Preferences of RNA-Binding Proteins with Deep Neural Networks. *Methods in Molecular Biology (Clifton, N.J.)* **2586**, 197–215 (2023).

255. Avsec, Ž. *et al.* Kipoi: Accelerating the community exchange and reuse of predictive models for genomics. 31.

256. Whalen, S., Schreiber, J., Noble, W. S. & Pollard, K. S. Navigating the pitfalls of applying machine learning in genomics. *Nature Reviews Genetics* **23**, 169–181 (2022).

257. Sasse, A. *et al.* *How far are we from personalized gene expression prediction using sequence-to-expression deep neural networks?* (2023) doi:10.1101/2023.03.16.532969.

258. Huang, C. *et al.* *Personal transcriptome variation is poorly explained by current genomic deep learning models*. (2023) doi:10.1101/2023.06.30.547100.

259. Weddington, N. *et al.* ReplicationDomain: A visualization tool and comparative database for genome-wide replication timing data. *BMC Bioinformatics* **9**, 530 (2008).

260. Gilbert Repli-seq Pipeline – ENCODE.

261. Marchal, C. *et al.* Genome-wide analysis of replication timing by next-generation sequencing with E/L Repli-seq. *Nature Protocols* **13**, 819–839 (2018).

262. Massey, D. J., Kim, D., Brooks, K. E., Smolka, M. B. & Koren, A. Next-Generation Sequencing Enables Spatiotemporal Resolution of Human Centromere Replication Timing. *Genes* **10**, 269 (2019).

263. Bailey, T. L. STREME: Accurate and versatile sequence motif discovery. *Bioinformatics* **37**, 2834–2840 (2021).

264. He, K., Zhang, X., Ren, S. & Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. (2015).

265. Falcon, W. *et al.* PyTorchLightning/pytorch-lightning: 0.7.6 release. (2020) doi:10.5281/ZENODO.3828935.

266. Biewald, L. Experiment tracking with weights and biases. (2020).

267. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. (2017).

268. Wang, P. Lucidrains/enformer-pytorch. (2023).

269. Shrikumar, A., Greenside, P. & Kundaje, A. *Reverse-complement parameter sharing improves deep learning models for genomics.* (2017) doi:10.1101/103663.

270. Linder, J., Srivastava, D., Yuan, H., Agarwal, V. & Kelley, D. R. *Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation.* (2023) doi:10.1101/2023.08.30.555582.

271. Shwartz-Ziv, R. & Armon, A. Tabular Data: Deep Learning is Not All You Need. (2021).

272. Grinsztajn, L., Oyallon, E. & Varoquaux, G. Why do tree-based models still outperform deep learning on tabular data? (2022).

273. McElfresh, D. *et al.* When Do Neural Nets Outperform Boosted Trees on Tabular Data? (2023).

274. Landrum, M. J. *et al.* ClinVar: Improvements to accessing data. *Nucleic Acids Research* **48**, D835–D844 (2020).

275. Yan, J. *et al.* LightGBM: Accelerated genomically designed crop breeding through ensemble learning. *Genome Biology* **22**, 271 (2021).

276. Yan, F., Powell, D. R., Curtis, D. J. & Wong, N. C. From reads to insight: A hitchhiker's guide to ATAC-seq data analysis. *Genome Biology* **21**, 22 (2020).

277. Andrews, S. FastQC. (2023).

278. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **9**, 357–359 (2012).

279. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *GigaScience* **10**, giab008 (2021).

280. Ou, J. *et al.* ATACseqQC: A Bioconductor package for post-alignment quality assessment of ATAC-seq data. *BMC Genomics* **19**, 169 (2018).

281. Ramírez, F. *et al.* deepTools2: A next generation web server for deep-sequencing data analysis. *Nucleic Acids Research* **44**, W160–W165 (2016).

282. Amemiya, H. M., Kundaje, A. & Boyle, A. P. The ENCODE Blacklist: Identification of Problematic Regions of the Genome. *Scientific Reports* **9**, 9354 (2019).

283. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **16**, 321–357 (2002).