The Pennsylvania State University
The Graduate School
Molecular Cellular and Integrative Biosciences

IDENTIFICATION AND VISUALIZATION OF REGULATORY ELEMENTS AND 3D GENOME STRUCTURE

A Dissertation in
Integrative Biosciences

by

Yanli Wang

© 2017 Yanli Wang

Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

August 2017
The dissertation of Yanli Wang was reviewed and approved* by the following:

Feng Yue
Assistant Professor of Biochemistry and Molecular Biology
Dissertation Advisor
Co-Chair of Committee

James R. Broach
Professor of Biochemistry and Molecular Biology
Chair of the Department of Biochemistry and Molecular Biology
Co-Chair of Committee

Ross C. Hardison
T. Ming Chu Professor of Biochemistry and Molecular Biology

Jennifer W. Baccon
Associate Professor of Pathology and Neurosurgery

Peter J. Hudson
Willaman Professor of Biology

*Signatures are on file in the Graduate School
ABSTRACT

The three-dimensional (3D) organization of vertebrate genomes is intricately linked to genome function, specifically gene regulation. At the DNA level, distal regulatory elements such as enhancers need to be in physical contact with their target genes. At a larger scale, topological associating domains (TADs) have been suggested to be the basic unit of genome organization. To gain a better understanding of the relationship between chromatin structure and function, we designed and implemented a widely-used online platform for visualizing the spatial organization of the genome, which is also integrated with thousands of genomic datasets such as ChIP-Seq and RNA-Seq. This system is currently visited by thousands of people each month, as it facilitate researchers to hypothesize the function of non-coding elements and disease-related variants. We employed zebrafish as a model organism to explore its cis-regulatory landscape and investigate how the structure of the vertebrate genome impact its function.
# TABLE OF CONTENTS

List of Figures ............................................................................................................. vi
List of Tables ............................................................................................................... viii
List of Abbreviations .................................................................................................. ix
Acknowledgements ..................................................................................................... x

## Chapter 1 Structure and Function of the Vertebrate Genome ................................... 1

- Techniques to study the three-dimensional (3D) chromatin structure ................. 1
- The vertebrate genome architecture ........................................................................ 4
- The “glue” of vertebrate chromatin structure ....................................................... 9
- The implications of vertebrate chromatin structure on its function .................... 10
- Conclusion .............................................................................................................. 13

## Chapter 2 The 3D Genome Browser ....................................................................... 14

- Abstract .................................................................................................................. 14
- Introduction ............................................................................................................. 15
- Methods .................................................................................................................. 22
  - Explore published chromatin interactions datasets ............................................. 22
  - Browsing unpublished Hi-C datasets ................................................................. 24
  - Constructing the backend and user-interface ...................................................... 25
  - Supplementing Build-In Annotation .................................................................. 25
  - Embedding External Genome Browsers ............................................................. 26
- Results ..................................................................................................................... 27
  - Discovering promoter-enhancer interactions with the 3D Genome Browser ...... 27
  - Investigating functions of noncoding genetic variants with the 3D Genome Browser ........................................................................................................... 27
  - Uncovering structural variations with the 3D Genome Browser ....................... 31
  - Exploring chromatin structure conservation with the 3D Genome Browser ...... 34
  - Compressing Hi-C matrices with the BUTLR file format ................................. 37
- Conclusion and Discussion ......................................................................................... 40

## Chapter 3 The Cis-regulatory Landscape of Zebrafish ......................................... 43

- Abstract .................................................................................................................. 44
- Introduction ............................................................................................................. 45
- Methods .................................................................................................................. 47
  - Determining tissue-specific gene expression ....................................................... 47
  - Establishing putative promoters and enhancers .................................................. 47
  - Finding tissue-specific putative enhancers ......................................................... 48
LIST OF FIGURES

Figure 1-1. 3-C based technologies ................................................................. 3

Figure 2-1. Complexity of chromatin interactions data vs. other “omics” data ............ 17

Figure 2-2. Visualization of chromatin interactions data with the WashU Epigenome Browser ........................................................................ 18

Figure 2-3. Visualization of chromatin interactions data with Juicebox ................. 19

Figure 2-4. Overall design of the 3D Genome Browser ....................................... 21

Figure 2-5. Design of the BUTLR file format .................................................. 25

Figure 2-6. Hi-C heatmap mode denoting a promoter-enhancer interactions .......... 29

Figure 2-7. Gene expression of SCL25A37 across 109 tissues ........................... 30

Figure 2-8. Capture Hi-C of naïve B-cell displaying an enhancer-promoter interaction ................................................................................. 31

Figure 2-9. Using the virtual 4C mode of the 3D Genome Browser to identify interactions between cis-regulatory elements (enhancers) and their target genes ............................................................................ 33

Figure 2-10. Using the 3D Genome Browser to determine intrachromosomal structural variations ................................................................. 35

Figure 2-11. Using 3D Genome Browser to determine interchromosomal structural variations ............................................................................... 36

Figure 2-12. Using the 3D Genome Browser to explore chromatin structure conservation ................................................................................... 38

Figure 2-13. Comparison of storage memory required by different Hi-C matrix file formats ............................................................................. 39

Figure 3-1. Using histone modification to denote cis-regulatory elements ............. 50

Figure 3-2. UCSC browser snapshot of tissue-specific genes ............................... 51

Figure 3-3. Hierarchical clustering of tissue-specific genes ............................... 53

Figure 3-4. Gene ontology (GO) analysis of tissue-specific genes ....................... 54

Figure 3-5. Unannotated muscle-specific transcript ........................................... 55

Figure 3-6. Unannotated transcriptional start sites (TSSs) ................................... 55
Figure 3-7. Percentage of Distal H3K27ac compared to total H3K27ac ......................56
Figure 3-8. Hierarchical clustering of tissue-specific putative enhancers ..................57
Figure 3-9. Gene ontology (GO) analysis of tissue-specific putative enhancers ..........58
Figure 3-10. Conservation of putative enhancers across vertebrates .........................62
Figure 3-11. Conservation of cis-regulatory function in zebrafish and mouse ............63
Figure 3-12. Spatial organization of the zebrafish genome ..................................65
LIST OF TABLES

Table 2-1. List of Hi-C datasets hosted by the 3D Genome Browser.........................22

Table 2-2. List of ChIA-PET, Capture Hi-C, PLAC-Seq and HiChIP..........................23

Table 3-1. The numbers of tissue-specific genes for zebrafish brain, heart, kidney, liver and muscle..........................................................................................................................52

Table 3-2. The number of tissue-specific putative enhancers for zebrafish brain, heart, kidney, liver and muscle.................................................................................................................56

Table 3-3. Transcriptional factor (TF) motifs predicted to be over-represented in each zebrafish tissue ..............................................................................................................................59
# LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3C</td>
<td>Chromosome Conformation Capture</td>
</tr>
<tr>
<td>4C</td>
<td>Circularized Chromosome Conformation Capture</td>
</tr>
<tr>
<td>5C</td>
<td>Chromosome Conformation Capture Carbon Copy</td>
</tr>
<tr>
<td>ATAC-Seq</td>
<td>Assay for Transposase-Accessible Chromatin with High-Throughput Sequencing</td>
</tr>
<tr>
<td>BUTLR</td>
<td>Binary Upper Triangular Matrix (file format)</td>
</tr>
<tr>
<td>Cas9</td>
<td>CRISPR-associated protein 9</td>
</tr>
<tr>
<td>ChIA-PET</td>
<td>Chromatin Interaction Analysis by Pair-End Tag Sequencing</td>
</tr>
<tr>
<td>ChIP-Seq</td>
<td>Chromatin Immunoprecipitation followed by DNA Sequencing</td>
</tr>
<tr>
<td>CRISPR</td>
<td>Clustered, Regularly Interspaced, Short Palindromic Repeat</td>
</tr>
<tr>
<td>DHS</td>
<td>DNase I Hypersensitive Site</td>
</tr>
<tr>
<td>GO</td>
<td>Gene Ontology</td>
</tr>
<tr>
<td>GREAT</td>
<td>Genomic Regions Enrichment of Annotations Tool</td>
</tr>
<tr>
<td>KRAB</td>
<td>Krüppel Associated Box</td>
</tr>
<tr>
<td>PAM</td>
<td>Protospacer-Adjacent Motif</td>
</tr>
<tr>
<td>PLAC-Seq</td>
<td>Proximity Ligation Assisted ChIP-Seq</td>
</tr>
<tr>
<td>RNA-Seq</td>
<td>RNA Sequencing</td>
</tr>
<tr>
<td>sgRNA</td>
<td>Single Guide RNA</td>
</tr>
<tr>
<td>TF</td>
<td>Transcription Factor</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I would like to thank my loving parents, Xue Wang and Xiaoyuan Zhang, for helping me through the darkest times of the PhD journey.
Chapter 1

Structure and Function of the Vertebrate Genome

Inevitably, biological structure determines its function. Such mantra has dominated the entire field of biology but nowhere does it pertain more than in the field of genomics. With the advent high-throughput techniques, the field has undergone a revolution, giving unprecedented insight into the structure and function of the vertebrate genome.

One such insight encompasses that chromatin, a complex of DNA and proteins, is organized into layers of superimposed structures but this serves more than just the compactor of DNA; it is responsible for the diverse array of gene expression in determining cellular identities despite most autologous cells share the same DNA sequence. The control of chromatin structure is complex and dynamic with multiple layers of regulation, including DNA methylation patterns, chromatin remodeling, histone variants, post-translational histone modifications, and transcription factors (TFs) activity. While each of these regulatory mechanisms are important and are detected by interesting techniques, our studies focus on the spatial organization of the genome, that is, the chromatin that interacts in the three-dimensional space despite the distance defined by its linear genomic sequence.

Techniques to study the three-dimensional (3D) chromatin structure

Many methodologies have been developed to study chromatin structure. Historically, the technique used to study spatial organization of the genome is imaging through Fluorescence in situ Hybridization (FISH) (Gall and Pardue, 1969). This
cytogenetic technique involves employing fluorescent probes to highlight complementary genomic regions. Unfortunately, FISH suffers from low spatial resolution and is limited to a few loci of interest at one time. Nevertheless, it has provided important insight into the arrangement of higher-order chromatin within the nuclear space.

In more recent advances in technologies to determine chromatin interactions, chromosome conformation capture (3C) and its derivative techniques were invented. These techniques rely on the formaldehyde crosslinking of chromatin proteins to their associated DNA, which fixes the chromatin structure through covalent bonds. Next, after the digestion with restriction enzymes, proximity ligation, where DNA fragments are subjected to conditions that favor intramolecular (cross-linked fragments) over intermolecular (random fragments) ligation occurs. Finally, the ligated products are reverse cross-linked and quantified. In 3C, the ligation frequency is determined through real-time polymerase chain reaction (qPCR) with locus-specific primers (Dekker et al., 2002). Since these primers must be pre-determined, 3C could only interrogate interactions between a single pair of genomic loci in a one-vs-one approach.

In circular 3C (4C), a second round of restriction enzyme digestion with a distinct recognition site than that of the initial enzyme follows the reverse cross-link step, which leaves the fragments with sticky ends that permit them to undergo self-circularizing ligation (Simonis et al., 2006; Zhao et al., 2006). Then, these DNA are amplified through inverse PCR with nested primers common to all products in a one-vs-many or even one-vs-all approach. Finally, the chromatin contact profile from the viewpoint of a single locus of interest could be then uncovered through microarray (therefore 4C is also known as 3C-on-chip) or sequencing. Similar to 3C, 4C may be plagued by low throughput.

The 3C carbon copy (5C) combines 3C with multiplexed ligation-mediated amplification (LMA) (Dostie et al., 2006). Specifically, 5C employs primers that anneal
across the site of ligation and whose ends contain universal sequence that permits all products to be amplified. In a \textit{many-vs-many} strategy, 5C could detect chromatin interactions between many loci as allowed with primer availability. Unfortunately, this requirement for primer design could become laborious and therefore the method cannot interrogate the chromatin contacts genome-wide.

\textbf{Figure 1-1. 3-C based technologies.} A schematic of 3C, 4C, 5C and Hi-C methodologies.

The high-throughput 3C (Hi-C) is the first technique that permitted unbiased, genome-wide probing of chromatin interactions (\textit{all-vs-all}) (Lieberman-Aiden et al., 2009). It utilizes biotin-labelled nucleotides that anneal to 5’ sticky ends created by the restriction enzyme and then become involved in the proximity ligation process. The biotinylated nucleotides that remain non-ligated are then removed, allowing the biotin pulldown of only ligated DNA fragments, which then undergo high-throughput paired-end sequencing. Although Hi-C possesses genome-wide detection capabilities, it could only provide
superimposed average chromatin structure within a cell population. To measure the genomic architecture at a cell level resolution, single-cell Hi-C was developed, which added sorting cells and utilizing multiplexed PCR amplification to the Hi-C protocol (Nagano et al., 2013). The single-cell Hi-C method could be employed to detect variability of chromatin interactions among cells in the population.

To retain the genome-wide interrogation power of Hi-C while maintaining the cost-effectiveness in investigating the selected loci of interest, more techniques have been developed. Firstly, the chromatin interaction analysis by paired-end tag sequencing (ChIA-PET) utilizes chromatin immunoprecipitation (ChIP) before proximal ligation to enrich for regions that are bound by certain TFs or possess certain histone modifications/variants (Fullwood and Ruan, 2009). Secondly, the target enrichment Hi-C (Capture Hi-C) involves utilizing biotinylated oligonucleotide probes that hybridizes to specific sequences (e.g. promoters or disease risk loci) to select for regions containing those sequences before biotin pulldown and high-throughput sequencing (Martin et al., 2015; Mifsud et al., 2015).

The vertebrate genome architecture

Nucleosomes. At the most basic level, chromatin is organized into fundamental units known as nucleosomes, which consists of ~146-bp of DNA wrapped around a histone octamer composed of two of each of the histones H2A, H2B, H3 and H4 (Kornberg, 1974; Olins and Olins, 1974). Each nucleosome is connected to its neighbors via ~10-80-bp of linker DNA. This “beads on a string” chromatin then becomes tightly wrapped as 30 nm fibers stabilized by the linker histone H1. At the higher orders, the chromatin can be folded into complex structures through chromatin loops, topological
domains, nuclear compartments and chromosome territories that are subjected to more sophisticated regulation. Evidently, with such intricate arrangement, the levels of compactions vary to create distinct structures and each structure confers transcriptional regulations to the underlying DNA. Structures with more open configuration represents the region of the genome under active transcription, also known as the euchromatin. In contrast, the tightly bound chromatin structure that becomes inaccessible to TFs and RNA polymerase II (RNAPII), embodies the heterochromatin, which is classified into constitutive and facultative heterochromatin. Constitutive heterochromatin is mainly comprised of pericentric and telomeric repetitive elements that are permanently condensed and relegated to the nuclear periphery (Grunstein, 1998) while facultative heterochromatin consists of genomic regions that are reversibly condensed based on cellular or developmental context.

**Chromatin Loop.** The chromatin loop is defined as the genomic sequences in *cis* (from the same chromosome) that have more spatial proximity compared to their intervening sequences. Chromatin loops are most famously embodied by promoter-enhancer loops. The promoter consists of noncoding *cis*-regulatory sequences typically proximal (relatively compared to other *cis*-regulatory elements) to and includes the transcription start site (TSS). It often contains a core promoter comprised of significant DNA motifs (i.e. TATA box, initiator (Inr) element, TFIIB recognition element (BRE), and the downstream promoter element (DPE)). The promoter along with the RNAPII and appropriate general TFs, could initiate and drive a basal level of transcription (Blackwood and Kadonaga, 1998). To refine the control of this basal transcriptional machinery, other *cis*-regulatory elements, such as enhancers (also: repressors, insulators and locus-control regions), must participate in the process. The enhancer also consist of noncoding sequences but those reside distal to its target gene of regulation. Unlike most promoter
sequences with characteristic motifs and CpG rich content, enhancer does not contain any defining sequences save for possible known TF binding motifs. Furthermore, the two elements have distinct histone modification landscape, with H3K4me3 enrichment at promoters (Zhou et al., 2011a) and H3K4me1 with H3K27ac or H3K27me enrichment for active and poised enhancers, respectively (Creyghton et al., 2010; Rada-Iglesias et al., 2011; Zentner et al., 2011). Finally, the enhancers could also be observed generating non-coding RNAs known as enhancer RNAs (eRNAs) (Kim et al., 2010).

For enhancer to activate the promoter and stimulate its level of transcription, the two elements must be juxtaposed by assuming the chromatin loop configuration (Deng et al., 2012). TFs bind to spaces of open chromatin on promoters and enhancers and actively recruit co-activators such as the multiprotein complex Mediator (Kim et al., 1994; Thompson et al., 1993), P300 (Eckner et al., 1994) or cyclic AMP-responsive element-binding (CREB)-binding protein (CBP) (Naar et al., 1998). The activity of the enhancer is cell-specific and the orchestration of these events allow enhancers to exact spatiotemporal control of gene expression, which is important for determining cellular identities and coordinating developmental sequences. Curiously, despite the three-dimensional proximity of the promoter and the enhancer, there is no pattern that describes the position of the enhancers in relation to the promoters in the linear genomic sequence, as enhancers exert their control in a location- and orientation-independent fashion. This contributes to the dearth of knowledge about the relationship between enhancers and their target genes. Moreover, enhancers may be promiscuous and redundant, as one enhancer could regulate multiple genes and multiple enhancers could regulate the same gene. Several enhancers could also span kilobases in near contiguity while exhibiting prominent TF and Mediator-binding density as super-enhancers (Whyte et al., 2013).
**Topological Associating Domains.** The genome-wide analysis of chromatin structure data has yielded genomic regions where intra-region chromatin interactions are dominant over inter-region ones, termed topologically associating domains (TADs) (Dixon et al., 2012; Nora et al., 2012; Sexton et al., 2012) that are interspersed with regions with comparatively less chromatin interactions. The sizes of these TADs range from 100-kb to 5-Mb with a median size of ~880kb. This compartmentalization of genomic regions into TADs has interesting implications regarding its function, as for example, it has been demonstrated that enhancer-promoter interactions mostly occur within a TAD (Shen et al., 2012), suggesting that the TAD functions to limit the search space of the promoter for its enhancer. Furthermore, the boundaries between TADs are enriched with housekeeping genes, histone modification of active transcription such as H3K4me3 and H3K36me3, certain repeat elements and architectural proteins that would be discussed later in this chapter. The TADs exhibit stability across different cell types and conservation among vertebrate species (Dixon et al., 2012). With advancement in high-throughput 3C-derived technologies, it has been shown that TADs are further subdivided into smaller domains called sub-TADs, whose sizes range from 40-kb to 3-Mb with median size of ~185kb (Phillips-Cremins et al., 2013; Rao et al., 2014). While TADs are for the large part tissue-invariant, sub-TADs are observed to be tissue-specific (Phillips-Cremins et al., 2013). Unfortunately, it may be challenging to tease out sub-TADs from TADs since the definition and classification of these domains rely on the resolution and methodology of the 3C-derived experiment.

**Nuclear Compartments.** Associations among TADs establish a higher level megabase-sized chromatin structure known as nuclear compartments, of which there are two types (Lieberman-Aiden et al., 2009). The compartment A consists of gene-rich euchromatic, active TADs while compartment B is comprised of gene-poor
heterochromatic, inactive TADs. As such, domains within A compartments favor interactions with other type A domains and vice versa. With high-resolution Hi-C, it is revealed that type A compartments could be further subdivided into two sub-compartments while type B ones contain four sub-compartments (Rao et al., 2014). While TADs are maintained across distinct cell types and tissues, nuclear compartments are dynamic and TADs could switch between different compartments depending on the cellular and developmental context (Dixon et al., 2015). Nuclear compartments appear to be a mechanism of positioning similar TADs in proximity of or segregated from other types of TADs.

**Lamina-Associated Domains.** At the nuclear periphery, certain heterochromatic regions of chromosomes becomes associated with the nuclear lamina – the inner lining of the nuclear envelope. These regions, also known as lamina-associated domains (LADs), exhibit repressive histone marks such as H3K27me3 and H3K9me2 (Guelen et al., 2008). Similar to heterochromatin, LADs are classified into constitutive LADs, which demonstrate conservation across cell types and tissues, and facultative LADs, which differ in a cell-type-specific manner (Meuleman et al., 2013). Similarly, chromatin near the nucleolus has also been characterized. These nucleolus-associated domains (NADs) are mainly composed of pericentric satellite repeats and enriched in A and T (van Koningsbruggen et al., 2010). The mechanisms responsible for relegating chromatin structures at the nuclear or nucleolar periphery could play a significant role in the organization of the interphase chromosomes.

**Chromosome Territories.** Within the interphase nucleus, each chromosome occupies a discrete space, called chromosome territory, in a nonrandom and cell type-dependent fashion which rarely intermingle, as noted by FISH studies (Lichter et al., 1988) and later confirmed by Hi-C (Lieberman-Aiden et al., 2009). This arrangement involves
having gene-rich chromosomes near the nuclear interior and gene-poor chromosomes in
the nuclear periphery (Croft et al., 1999). Furthermore, gene-rich chromosomes occupy a
larger region in the nucleus compared to gene-poor chromosomes, which may be
explained by the space taken up by the transcriptional machinery. Analysis of
chromosome territories using cryo-FISH has yielded places where interminglings of
chromosomes, or interchromosomal interactions, occur certain cells, which correlates
strongly with sites of recombination during chromosomal translocation following ionizing
radiation (Branco and Pombo, 2006). Evidently, the chromatin structure is defined by
physical proximity and positioning, which may be altered with environmental stimuli,
ultimately affecting gene expression.

The “glue” of vertebrate chromatin structure

As demonstrated by 3C-based technologies, contact probability of two genomic
segments decreases as their linear distance increases. Therefore, to maintain long-range
interactions, architectural “glue” is necessary to stabilize those chromatin structures. One
of such “glue” is the CCCTC-binding factor (CTCF) insulator protein. CTCF mediates
chromatin looping by stabilizing the structure at the base through binding to its DNA motif
and to other DNA-bound CTCF proteins. Here, these CTCF-CTCF interactions occur at
pairs of convergent sites, which describe the orientation of the motif as forward-reverse
(Rao et al., 2014). The CTCF proteins also demonstrate enrichment at compartment and
TADs boundaries, signaling their role in maintaining those structures. At these regions,
pairs of divergent CTCF sites are enriched (Gomez-Marin et al., 2015), suggesting a
diverse folding in the formation of those structures compared to the chromatin loop.
Another architectural protein that serves to stabilize chromatin structure is the protein cohesin, which consists of, in vertebrates, subunits SMC1 and SMC3 of the Structural Maintenance of Chromosomes (SMC) family, strand-break repair protein RAD21, and either stromal antigen SA1 or SA2 that together assemble into a ring-like complex (Haering et al., 2002). Cohesins are most well-known to be involved with sister chromatid association and separation during cell division (Losada et al., 1998; Sumara et al., 2000). As a chromatin architectural protein, it interacts with the Mediator to stabilize chromatin loop interactions (Kagey et al., 2010). Furthermore, it co-localizes with CTCF to maintain boundaries of chromatin loops, TADs and nuclear compartments (Rubio et al., 2008). Mutations within the cohesin subunit, SMC1A, SMC3 or RAD21 lead to Cornelia de Lange syndrome, an autosomal dominant disorder with characteristic facies of synophrys, long eyelashes, short nose with anteverted nares, long and smooth philtrum and thin upper vermilion with downturned corners of the mouth, developmental delay and autistic behavior, all associated with transcriptional dysregulation in many tissues (Ramos et al., 2015).

The implications of vertebrate chromatin structure on its function

Chromatin Loop. The impact of cis-regulatory elements on transcriptional regulation cannot be understated. In the disease β-thalassemia, most patients have single point mutations in or deletion of the β-globin gene (HBB). Nevertheless, there have been more cases of β-thalassemia where the deletion occurred upstream of HBB but otherwise left the gene itself devoid of changes (Van der Ploeg et al., 1980). The deleted region contains a super-enhancer, the locus control region (LCR), which functions to stimulate HBB transcriptional activation in a erythroid-cell specific manner (Grosveld et al., 1987).
These disruptions to noncoding regulatory regions of the genome leading to disease phenotypes underlie the observation that in genome-wide association studies (GWAS), over 90% of disease-associated genetic variants reside in the noncoding genomic areas (Maurano et al., 2012). While it remains challenging to parse disease-causal genetic variants from lead ones uncovered by GWAS, due to the effect of linkage disequilibrium as well as our coarse understanding of the enhancer code, the functional significance of enhancers and other cis-regulatory elements in maintaining health and driving disease pathogenesis is undeniable.

A variety mechanisms could disturb the wild-type promoter-enhancer loop interactions, which could manifest in distinct phenotypes. This is perhaps most effectively encapsulated by the sonic hedgehog gene (SHH), which is regulated by the ZRS (zone of polarizing activity regulatory sequence) enhancer located a megabase away within the intron of another gene LMBR1 (Lettice et al., 2003). The deletion of the ZRS enhancer leads to limb truncation (Sagai et al., 2005) while its duplication can cause Haas type polysyndactyly (Sun et al., 2008) and triphalangeal thumb-polysyndactyly syndrome (Klopocki et al., 2008). Furthermore, point mutations within the enhancer are associated with preaxial polydactyly (Lettice et al., 2003), triphalangeal thumb (Furniss et al., 2008) or Werner mesomelic syndrome (Wieczorek et al., 2010). Evidently, the disruption of the promoter-enhancer wiring involves more than just the inactivation of the cis-regulatory elements through deletion or point mutations, the increase in their activities through duplication or introduction of TF binding sites could also lead to diseases. Another way to perturb this wiring is through enhancer hijacking, which describes the phenomenon that chromosomal rearrangements place genes under the control of enhancers such that the inappropriate activation of those genes occurs (Northcott et al., 2014).
Higher-Order Chromatin Structure. Many cellular processes exploit the underlying 3D organization of the genome. During X-chromosome inactivation, the long noncoding RNA Xist coordinates the marking of genes for silencing on the inactive X based on the chromatin structure, where gene-rich neighborhoods near Xist are inactivated before the interspersed gene-poor domains (Engreitz et al., 2013; Simon et al., 2013). Furthermore, TADs are associated with the temporal order of replication (Pope et al., 2014). The TADs of compartment A are early replicating while TADs of compartment B and LADs are late replicating. Appropriate replication timing is crucial in limiting the perturbation of chromatin structure caused by cell division. Similar disruption could be instigated by development and differentiation, during which genes switch between A and B nuclear compartments and intra- and inter-domain interactions are established or eliminated frequently. (Dixon et al., 2015).

Architectural Proteins. The architectural “glue” proteins, CTCF and cohesin play significant roles in maintaining the integrity of the chromatin structure to ensure proper cellular function. This becomes apparent when studies show that CTCF and cohesin subunits are mutated in many types of cancers (Kandoth et al., 2013; Kon et al., 2013) and that CTCF- and cohesin-binding sites are also disrupted by cancer mutations, often near dysregulated genes in the disease (Hnisz et al., 2016; Katainen et al., 2015). Furthermore, as DNA methylation could abolish CTCF binding to the recognition site (Bell and Felsenfeld, 2000), the functions of CTCF could be inactivated in diseases characterized by hypermethylation. For example, in certain gliomas, hypermethylation at a CTCF binding site abrogates its insulation function, bringing the oncogene PDGFRA under the control of a constitutive enhancer located within a different TAD (Flavahan et al., 2016). Similarly, structural variations such as deletions could also disturb TAD boundaries causing ectopic interactions (Nora et al., 2012). The disruption of TAD
boundaries involving the \textit{EPHA4} TAD causes the limb enhancers to interact with gene promoters of neighboring TADs leading to limb malformation (Lupianez et al., 2015).

\textbf{Conclusion}

As this chapter illustrated, genomic structure and function are intrinsically linked. Therefore, elucidating the function of a genomic region would be accelerated when it is considered in its spatial context. In the next chapter, we will illuminate how the chromatin structure could be utilized in the understanding of its function.
Chapter 2

The 3D Genome Browser

Abstract

Recent advent of 3C-based technologies such as Hi-C and ChIA-PET provides us an opportunity to explore chromatin interactions and 3D genome organization in an unprecedented scale and resolution. However, it remains a challenge to visualize chromatin interaction data due to their size and complexity. Here, we introduce the 3D Genome Browser (http://3dgenome.org), which allows users to conveniently explore both their own and over 100 publicly available chromatin interaction datasets. We design a new binary data format for Hi-C data, which reduces the file size and increases the data query speed by a magnitude. Our browser provides the ability to explore inter-chromosomal interactions and presents a tool for identify potential structural variations in cancer genomes. Users can seamlessly integrate thousands of other “omics” data sets, such as ChIP-Seq and RNA-Seq for the same genomic region, to gain a complete view of both regulatory landscape and 3D genome structure. Finally, our browser provides multiple methods to link distal cis-regulatory elements with their potential target genes, including virtual 4C, ChIA-PET, Capture Hi-C and cross-cell-type correlation of proximal and distal DNase I hypersensitive sites (DHSs), and therefore represents an invaluable resource for the study of gene regulation in vertebrate genomes.
Introduction

As established in the last chapter, the 3D organization of vertebrate genomes is intricately linked to genome function, specifically gene regulation. At the DNA level, distal regulatory elements such as enhancers need to be in physical contact with their target genes. At a larger scale, TADs have been suggested to be the basic unit of mammalian genome organization (Dixon et al., 2012). Several recent high-throughput technologies based on 3C (Dekker et al., 2002) have emerged (such as Hi-C (Lieberman-Aiden et al., 2009), ChIA-PET (Li et al., 2010), Capture-C (Hughes et al., 2014), Capture Hi-C (Mifsud et al., 2015), PLAC-Seq (Fang et al., 2016) and HiChIP (Mumbach et al., 2016)) and have provided an unprecedented opportunity to study this spatial organization in a genome-wide fashion.

As the volume of chromatin interaction data generated from individual labs and large collaborative projects such as the ENCODE (Consortium, 2012) and 4D Nucleome consortia increases, efficient visualization and navigation of these data becomes a major bottleneck for their biological interpretation. Due to the size and complexity of these interactome data (Fig. 2-1), it is difficult to store and explore them on a personal device. In tackling this challenge, several useful tools for visualizing chromatin interactions have been developed, and each of them has its unique features and limitations. The Hi-C Data Browser (Lieberman-Aiden et al., 2009) is the first web-based query tool that visualizes Hi-C data as heatmaps. Unfortunately, it does not support zoom functionalities and only hosts limited number of datasets. The WashU Epigenome Browser (Zhou et al., 2013; Zhou et al., 2011b) can display both Hi-C and ChIA-PET data, and it provides access to thousands of epigenomic datasets from the ENCODE and Roadmap Epigenome projects (Fig. 2-2). However, due to the file size of Hi-C matrices, which could reach hundreds of
gigabytes, its speed for uploading and exploring large Hi-C datasets is not optimal. Furthermore, it does not offer an option to display interchromosomal chromatin interaction data as heatmaps. Juicebox (Durand et al., 2016), a Java program that was introduced as a companion tool for exploring the high-resolution Hi-C data made available by the initial paper (Rao et al., 2014), has many practical features, such as zoom functionalities and built-in supplementation of other epigenomic data (Fig. 2-3). It is, however, not portable, as it requires installation. In addition, most aforementioned tools only display Hi-C as a heatmap, which is convenient for visualizing large domain structures such as TADs, but may not be the most intuitive way for visualizing enhancer-promoter interactions.
Figure 2-1. Complexity of chromatin interactions data vs. other “omics” data. Chromatin interactions data have a complexity of $O(n^2)$ while other “omics” data have a complexity of $O(n)$. Therefore, the time and space demand of chromatin interactions data dramatically outpace that of other “omics” data.
Figure 2-2. Visualization of chromatin interactions data with the WashU Epigenome Browser. The chromatin interactions of GM12878 and K562 surrounding the gene SLC25A37 as virtualized by WashU Epigenome Browser is shown. While the browser allows the comparison of more than one datasets, the loading speed of the chromatin interactions data is not optimal.
Figure 2-3. Visualization of chromatin interactions data with Juicebox. The Hi-C heatmap for all chromosomes is virtualized with Juicebox, supplemented with build-in gene annotation and "omics" datasets. While Juicebox is useful for visualizing larger chromatin structure, it is not optimized for investigating enhancer-promoter interactions.

Here, we present the 3D Genome Browser (http://www.3dgenome.org), which is a fast and portable web-based browser that allows users to smoothly explore both published and their own chromatin interaction data. Our 3D Genome Browser features six display modes: 1) intrachromosomal Hi-C contact matrices as heatmaps along with corresponding
TAD information, 2) stacked opposing intrachromosomal Hi-C heatmaps to facilitate cross-tissue or even cross-species comparisons of chromatin structures between two different Hi-C datasets, 3) interchromosomal Hi-C heatmaps, 4) Hi-C-derived virtual 4C plot along with ChIA-PET data, 5) ChIA-PET or other ChIP-based chromatin interaction data such as PLAC-Seq and HiChIP, and 6) Capture Hi-C or other capture-based chromatin interaction data. These distinct and versatile display modes encourage users to explore interactome data tailored to their own needs, from exploring organization of higher-order chromatin structures to investigating enhancer-promoter interactions. Our browser provides zoom and traverse functionalities in real time and supports querying inputs of genomic features such as gene name or SNP rsid. The 3D Genome Browser provides users with the ability to highlight loci of interest and to zoom into or center onto these regions as needed. In addition, the browser incorporates the University of California, Santa Cruz (UCSC) Genome Browser and the WashU Epigenome Browser, allowing users to simultaneously query and supplement chromatin interaction data with thousands of genetic, epigenetic and phenotypic datasets, including ChIP-Seq and RNA-Seq data from the ENCODE and Roadmap Epigenomics projects. The overall design of the 3D Genome Browser is summarized in Fig. 2-4.
Figure 2-4. Overall design of the 3D Genome Browser. The 3D Genome Browser links chromatin interactions data to all the “omics” data that are currently available.
Methods

Explore published chromatin interactions datasets

Our 3D Genome Browser hosts the most comprehensive and up-to-date high-quality chromatin interactions data, including more than 100 Hi-C datasets (Table 2-1) and 38 ChIA-PET, Capture Hi-C, PLAC-Seq and HiChIP datasets (Table 2-2). To assist users explore these domain structures estimated by Hi-C data, we systematically predicted topologically associating domains (TADs) for 38 Hi-C datasets when such information were not available in the original manuscript, using streamlined pipeline based on Dixon et al (Dixon et al., 2012). The TADs information are displayed alongside intrachromosomal heatmaps.

Table 2-1. List of Hi-C datasets hosted by the 3D Genome Browser.

<table>
<thead>
<tr>
<th>Species</th>
<th>Assembly</th>
<th>Tissue</th>
<th>Resolution</th>
<th>Interchromosomal Data</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>hg19</td>
<td>GM12878</td>
<td>1-kb, 5-kb, 10-kb, 25-kb, 50-kb, 100-kb, 250-kb, 500-kb</td>
<td>Yes</td>
<td>(Rao et al., 2014)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K562</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMR90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HMEC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HUVEC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NHEK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>KBM7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mouse</td>
<td>mm9, mm10</td>
<td>CH12</td>
<td>5-kb, 10-kb, 25-kb, 50-kb, 100-kb, 250-kb, 500-kb</td>
<td></td>
<td>(Dixon et al., 2012)</td>
</tr>
<tr>
<td>Human</td>
<td>hg19, hg18</td>
<td>GM12878</td>
<td>40-kb</td>
<td>No</td>
<td>(Shen et al., 2012)</td>
</tr>
<tr>
<td>Mouse</td>
<td>mm9, mm10</td>
<td>ESC</td>
<td>40-kb</td>
<td>No</td>
<td>(Jin et al., 2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cortex</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>hg19</td>
<td>IMR90</td>
<td>5-kb, 10-kb, 25-kb, 50-kb, 100-kb, 250-kb, 500-kb</td>
<td>No</td>
<td>(Leung et al., 2015)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H1-ESC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>H1-MES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>H1-MSC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>H1-NPC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>H1-TRO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>hg19</td>
<td>Aorta</td>
<td>40-kb</td>
<td>No</td>
<td>(Schmitt et al., 2016)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Left Ventricle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Liver</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thymus</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>hg19</td>
<td>Adrenal Gland</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bladder</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dorsolateral prefrontal cortex</td>
<td>40-kb</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Species</td>
<td>Method</td>
<td>Tissue</td>
<td>Enrichment Target</td>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>-------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>ChIA-PET</td>
<td>K562</td>
<td>CTCF</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H3K4me1</td>
<td>ENCODE3 (lab: Snyder)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H3K4me2</td>
<td>ENCODE3 (lab: Snyder)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H3K4me3</td>
<td>ENCODE3 (lab: Snyder)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H3K27ac</td>
<td>ENCODE3 (lab: Snyder)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RAD21</td>
<td>ENCODE3 (lab: Snyder)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POLR2A</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCT116</td>
<td>CTCF</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ESR1</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POLR2A</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MCF-7</td>
<td>GM12878</td>
<td>Rad21</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NB4</td>
<td>POLR2A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HeLa-S3</td>
<td>POLR2A</td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>HiChIP</td>
<td>GM12878</td>
<td>Cohesin</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td>Mouse</td>
<td>PLAC-Seq</td>
<td>ESC</td>
<td>H3K4me3</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H3K27ac</td>
<td>ENCODE2 (lab: Ruan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POL2</td>
<td>(Fang et al., 2016)</td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>Capture Hi-C</td>
<td>B Naïve</td>
<td>B Naïve</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B Total</td>
<td>B Total</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD4 Activated</td>
<td>CD4 Activated</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD4 Naïve</td>
<td>CD4 Naïve</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD4 Total</td>
<td>CD4 Total</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD4 Nonactivated</td>
<td>CD4 Nonactivated</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD8 Naïve</td>
<td>CD8 Naïve</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD8 Total</td>
<td>CD8 Total</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endothelial Precursors</td>
<td>Endothelial Precursors</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Erythroblasts</td>
<td>Erythroblasts</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fetal Thymus</td>
<td>Fetal Thymus</td>
<td>(Javierre et al., 2016)</td>
<td></td>
</tr>
</tbody>
</table>

Table 2-2. List of ChIA-PET, Capture Hi-C, PLAC-Seq and HiChIP.
Browsing unpublished Hi-C datasets

In addition to published datasets, the users could view their own Hi-C data by converting the contact matrices into a novel file format called Binary Upper Triangular Matrix (BUTLR file) developed by us. The BUTLR file encodes an entire genome-wide chromatin interactions data into a binary, indexed format. To compress the original contact matrices, BUTLR omits the storage of the most common value of the intrachromosomal and interchromosomal (if available) upper triangular matrix of the contact matrices, which as contact matrices are sparse matrices, is usually zero. The locations of each chromosome or chromosome-pair matrix, row indices of each matrix and column indices of non-omitted values along with those values are binarized and indexed within the BUTLR file structure (Fig. 2-5). Perl scripts that encode and decode BUTLR files are available at http://github.com/yuelab/BUTLRTools.

By hosting the BUTLR file on any HTTP-supported server and providing the URL to the 3D Genome Browser, a user can take full advantage of the features of our browser, without uploading their Hi-C, data since the browser would only query the selected region through binary indexing, rather than searching through the entire matrix. This capability is similar to the bigWig/bigBed mechanism invented by UCSC (Kent et al., 2010).
**Figure 2-5. Design of the BUTLR file format.** The BUTLR format encodes the chromosome or chromosome pair (interchromosomal interactions), row and column information as a series of pointers (represented by arrows). For purposes of compression, the most common values (MCVs) and their column locations are omitted from storage.

**Constructing the backend and user-interface**

The 3D Genome Browser is supported by the LAMP (Linux, Apache, MySQL, and PHP) stack architecture on the backend. At the user-interface level, the browser depends on HTML5, Cascading Styling Sheets (CSS) and JavaScript and its libraries JQuery and D3.js. All displays are rendered with HTML5 Canvas or Inline SVG. The user-interface has been extensively tested in Chrome and is compatible with Firefox and Safari.

**Supplementing Build-In Annotation**

In addition to TAD information, the 3D Genome Browser provides build-in gene annotations and DHS peaks when the user choose not to supplement the chromatin interactions data with external genome browsers. Such choice improves the loading times of the 3D Genome Browser and is useful for quick browsing of the large-scale chromatin
structure. The gene annotation is derived from the knownCanonical UCSC database while the DHS peaks are downloaded as ENCODE Broad Peaks files from UCSC.

**Embedding External Genome Browsers**

For the UCSC Genome Browser, we embed its sessions with the iframe and we align its content with our tracks by manipulating the scroll bars of the div HTML element containing the iframe. The WashU Epigenome Browser provides a JavaScript function for seamless integration into our browser. For both external browsers, it is possible for the user to embed a user-defined session consisted of user-selected tracks and options by providing the session URL to the 3D Genome Browser.
Results

Discovering promoter-enhancer interactions with the 3D Genome Browser

As enhancers exert their effects through contacts with their target promoters in the 3D space, such interactions are uncovered genome-wide by Hi-C and visualized with the 3D Genome Browser. Here, using the intrachromosomal Hi-C mode, we observed heatmap cells with high contact value involving the SLC25A37 gene in the 5-kb resolution contact map from the human erythroleukemic K562 cell line (Rao et al., 2014). To interpret its biological meaning, we integrated the WashU Epigenome Browser with gene annotation, histone modification H3K4me1, H3K4me3 and H3K27ac and chromHMM (Ernst and Kellis, 2012) for K562. We determined that the two interacting loci are the promoter of SLC25A37 and a putative enhancer as determined by histone modification patterns and chromHMM (Fig. 2-6). Utilizing the gene expression tool, we note that the expression of SLC25A37 is more than three times as compared to that of other tissues (Fig. 2-7), intimating that SLC25A37 expression may be tissue-specific for K562 and therefore driven by an enhancer. The putative enhancer has indeed been confirmed to exhibit activities that regulate SLC25A37 expression during late phase erythropoiesis (Huang et al., 2016).

Although a viable way to uncover promoter-enhancer pairs, interactions detected by Hi-C are not specific for these interactions. These interactions are most effectively encapsulated by capture-based chromatin ligation-based methods that seek long-range interaction which involves selected elements of interests captured with pre-determined sequences, such as promoters. The Capture Hi-C track is supplemented with DNase I hypersensitive site (DHS)-linkage data, which represents another method of linking distal
regulatory element with their target genes by computing Pearson correlation coefficients between the gene proximal and distal DHS pairs across more than 100 ENCODE cell types (Thurman et al., 2012). As an example, using the Capture Hi-C mode, we noticed several interactions involving the promoter of the \textit{PAX-5} gene and an H3K27ac-rich region downstream of the \textit{ZCCHC7} gene in the naïve B-cell Capture Hi-C dataset (Javierre et al., 2016) (Fig. 2-8). One region marked by histone modification was indeed determined to be an enhancer for \textit{PAX5} and its disruption leads to leukemogenesis (Puente et al., 2015). By integrating multiple lines of evidence, our browser provides a valuable resource for investigators to create hypotheses connecting distal regulatory element to their target gene.
Figure 2-6. Hi-C heatmap mode denoting a promoter-enhancer interactions. This mode demonstrates the putative interaction (indicated by arrow) between the promoter of SLC25A37 and an enhancer candidate.
Figure 2-7. Gene expression of SCL25A37 across 109 tissues. SLC25A37 expression in K562 is more than three times as much compared to other 108 tissues.
Figure 2-8. Capture Hi-C of naïve B-cell displaying an enhancer-promoter interaction. This mode displays a putative interactions between the promoter of PAX5 and a candidate enhancer downstream of ZCCHC7.

Investigating functions of noncoding genetic variants with the 3D Genome Browser

Although displaying Hi-C data as a heatmap is informative to visualize large genome structures such as TADs, it is not an intuitive way to highlight contacts between two loci of interest, such as enhancer-promoter interactions. Resolutions at locus-specific levels also hold significance in the discovery of the functions of noncoding genetic variants, such as single nucleotide polymorphisms (SNPs), which may disrupt transcription factor binding sites of cis-regulatory elements. To attain loci-specific interactions, we implemented Hi-C-derived virtual 4C plot in our browser. The 4C experiment is a chromatin ligation-based method that surveys for one-vs-many interactions in the genome, that is, to measure the interaction frequencies between a "bait"
locus and any other loci in its vicinity. Its data is plotted as a line histogram, where the center is the “bait” region and any peak signals in distal regions indicate frequency of chromatin interaction events. In our browser, we use the queried region as bait and extract Hi-C data centered on that region, hence, virtual 4C. To bolster the power of the virtual 4C plot, our browser also supplements ChIA-PET and DHS-linkage data. ChIA-PET is another implementation of chromatin ligation-based method, which detects long-range interactions between genomic regions that are enriched for a feature (either histone modification or transcription factor binding).

To illustrate the utility of this browser mode, we queried the SNP rs12740374 (Fig. 2-9). The SNP that has been associated with high plasma low-density lipoprotein cholesterol (LDL-C) (Sandhu et al., 2008), which could lead to coronary artery disease and myocardial infarction. Since LDLs are processed by the liver, we examine the histone modifications in the Hep2G cell line. We used virtual 4C and ChIA-PET data from K562, since high resolution Hi-C and numerous ChIA-PET data are only available for K562, but not for hepatic cell lines. The rs12740374 SNP resides within a candidate enhancer region as marked by H3K27ac shown in the UCSC genome browser. In this case, virtual 4C, ChIA-PET and DHS-Linkage all support that there is a putative interaction between the enhancer harboring this SNP and the promoter region of SORT1 and any changes to this arrangement possibly affects SORT1 expression, which may ultimately modify LDL-C levels. A Study (Musunuru et al., 2010) has demonstrated that the rs12740374 minor allele creates a C/EBPα binding site which enhances SORT1 expression leading to decreased LDL-C levels, potentially meaning that the minor allele confers a gain-of-function effect. Still, despite the unusual conclusions reached by the study – as most minor alleles are usually loss-of-function – the virtual 4C mode of our 3D Genome Browser could aid in the
hypothesis generation about not only the potential cis-regulatory elements and their potential target genes but also the effects of noncoding variants.

Figure 2-9. Using the virtual 4C mode of the 3D Genome Browser to identify interactions between cis-regulatory elements (enhancers) and their target genes. Here, we investigate the effect of SNP rs12740374 by plotting the virtual 4C from the 5-kb K562 Hi-C, DHS-linkage and K562 H3K4me3 and POL2A ChIA-PET and integrating the UCSC Genome Browser. Based on the HepG2 chromHMM, the SNP resides at a putative enhancer region (orange) while according to virtual 4C (not available for HepG2), there is a peak indicating an interaction between this enhancer and the promoter of SORT1 (transcribed right to left). This is supported by K562 H3K4me3 and POL2A ChIA-PET as well as DHS-linkage data.
Uncovering structural variations with the 3D Genome Browser

Hi-C data not only detect chromatin interactions, but also may denote structural variations. Certain structural variations, such as deletions, insertions, inversions and translocations, establish signature patterns within the Hi-C heatmaps and therefore could be discovered with high-resolution Hi-C. With the browser’s compare Hi-C mode, the users could contrast the similarities and differences of chromatin structure across different cells/tissues or even different species. Comparing the cell line K562 to the cell line KBM7, we noted deletions specific to K562, one of which encompasses the tumor suppressor genes \textit{CDKN2A} and \textit{CDKN2B} (Fig. 2-10), as previously confirmed (Sherborne et al., 2010). The deletion of tumor suppressor genes represents one of the structural variation that could contribute to the tumorigenesis of K562.
Figure 2-10. Using the 3D Genome Browser to determine intrachromosomal structural variations. Comparing human K562 to KBM7 yields a deletion encompassing CDKN2A and CDKN2B specific to K562 on chromosome 9.
Figure 2-21. Using 3D Genome Browser to determine interchromosomal structural variations. An interchromosomal translocation causes the BCR-ABL fusion in K562 and KMB7.

Perhaps more striking structural variations occur in the interchromosomal space. Browsing the interchromosomal heatmap mode, we confirmed the BCR-ABL translocations in chronic myelogenous leukemia lines K562 and KBM7 (Fig. 2-11). Based
on the reciprocity of interchromosomal Hi-C heatmap, we determined that the translocation is unbalanced in K562 but balanced in KBM7. We also noted that the breakpoint in BCR is similar in the two cell lines – both around p210 – while the breakpoint in ABL differs between the two cell lines.

**Exploring chromatin structure conservation with the 3D Genome Browser**

Studying the evolutionary conservation of TADs could yield a greater understanding of their functional significance. The compare Hi-C mode of the 3D Genome Browser facilitates this endeavor by stacking opposing Hi-C heatmaps from homologous regions of different species for visual contrast. In this mode, we observed the conservation of TADs and the genes near or at the TAD boundaries between human and mouse in their homologous region surrounding the BCL-6/Bcl-6 genes (Fig. 2-12), suggesting the chromatin structure plays a vital role in the regulation of the proto-oncogenes.

**Compressing Hi-C matrices with the BUTLR file format**

In addition to binary indexing, the BUTLR file format provides compression of Hi-C matrices to lower their storage memory through the omission of redundant values within the matrix. At all resolutions, the BUTLR file format compresses the Hi-C matrices more efficiently than coordinated list format (a list of row, column and values, omitting entries where the value is zero) while at high resolutions, BUTLR file format dramatically reduces the required storage memory compared to tab-delimited matrix files (Fig. 2-13). Most notably, while 1-kb resolution hg19 intrachromosomal Hi-C contact matrices in the tab-delimited format requires almost 1 terabyte, the BUTLR format of those same matrices
would only take 11 gigabytes. The binary file format also greatly improves the query speed: using pre-loaded Hi-C data sets, our 3D browser generally return the query results as a heatmap in the matter of seconds.

Figure 2-12. Using the 3D Genome Browser to explore chromatin structure conservation. Comparing human GM12878 to mouse CH12 at the region surrounding the BCL6/Bcl6 region demonstrates an evolutionary conservation of the chromatin structure between the two species.
Figure 2-13. Comparison of storage memory required by different Hi-C matrix file formats. The plot comparing the compression of Hi-C matrices by BUTLR, tab-delimited matrix and coordinated list file formats for intrachromosomal map only (top) or inter- and intrachromosomal maps (bottom).
Conclusion and Discussion

Our overall aim for this project was to visualize chromatin interactions data in a fast, portable and robust manner, making these data manageable to the members of the scientific community who may not have access to machines of computing prowess capable of processing gigabytes of data. To accomplish this, we have developed the web-based 3D Genome Browser (http://3dgenome.org), which features six display modes tailored to the distinct needs. The first mode, intrachromosomal Hi-C heatmap, allows the quick browsing of enhancer-promoter interactions at high resolutions and TAD organization at lower resolutions. The second mode, compare Hi-C, provides the comparison and contrast of two different Hi-C datasets, useful for investigating tissue-specific interactions, structural variations and cross-species structural conservation. The third mode, interchromosomal Hi-C heatmaps, permits the exploration of interchromosomal interactions as well as interchromosomal rearrangements. The fourth mode, virtual 4C, visualizes the Hi-C data in a locus-specific viewpoint, facilitating the discovery of promoter-enhancer interactions and the effects of noncoding genetic variants. The fifth mode, ChIA-PET, and the sixth mode, Capture Hi-C, also allow the users to explore interactions involving their locus of interest.

One of the most powerful aspect of the 3D Genome Browser is the embedding of external genome browsers, UCSC Genome Browser and WashU Epigenome Browser, which allows the chromatin interactions data to be aligned with a wealth of genomic functional data such as ChIP-Seq, RNA-Seq, phenotype, and conservation tracks. Superimposing structure and function data facilitates the discovery of transcriptional regulatory mechanisms.
The 3D Browser supports a variety of features that allow users to browse unpublished data. First, our browser encourages integration with customized UCSC/WashU sessions, in which the users could add or modify existing tracks or upload their own data to UCSC/WashU. To view a customized UCSC session, the user would only be required to enter the session URL. Next, the users could view their own Hi-C data by converting the contact matrices into the BUTLR file format. Not only does the BUTLR format dramatically reduces the storage of high-resolution Hi-C data through compression, its indexed binarization allows the users to browse unpublished Hi-C data in Hi-C heatmap mode, virtual 4C mode and interchromosomal heatmap mode through HTTP.

Besides adding new published, high-quality datasets to our browser, we aspire to improve the 3D Genome Browser in two ways in the near future. First, we plan to increase the speed of the browser even more by increasing its reliance on Asynchronous JavaScript and XML (AJAX) and user sessions instead of GET requests through PHP. Second, we aim to expand the compare Hi-C mode such that it would allow the comparison of more than two Hi-C datasets.
Acknowledgements

This chapter is a manuscript in preparation co-authored by Wang Y., Zhang, Bo., Zhang, L., An, L., Xu, J., Li, D., Choudhary, MNK., Li, Y., Hu, M., Hardison, R., Wang, T., and Yue, F. It has been made available as a preprint on bioRxiv with the doi https://doi.org/10.1101/112268.
Chapter 3
The Cis-regulatory Landscape of Zebrafish

State of Collaboration

In this chapter, Hongbo Yang collected and sequenced most of the zebrafish samples while Yanli Wang, the author of the thesis, performed most of the analysis. Yu Luan contributed to the Hi-C analysis. This work relied on the conserved non-genic elements (CNEs) in zebrafish as established by the Gill Bejerano laboratory and on transgenic mouse reporter assays conducted in the Len Pennacchio laboratory for conservation of sequence and function analysis.
Abstract

The zebrafish *Danio rerio* have served as important vertebrate models for human genomics and genetics studies as well as large-scale forward/reverse genetic and pharmacological screens, yet our understanding of its cis-regulatory landscape and genomic structure lags behind the mammalian model mouse and the invertebrate model *Drosophila melanogaster*. In this study, we attempt to close this knowledge gap by profiling histone modification marks of putative promoters and enhancers, H3K4me3 and H3K27ac, with ChIP-Seq and correlate them with transcription as detected by RNA-Seq in five different zebrafish tissues. Furthermore, we initialized the study of the spatial organization of the zebrafish with preliminary Hi-C data. We established promoter candidates with novel transcript candidates, tissue-specific enhancers and their impact on gene expression, highly conserved cis-regulatory elements that highly likely remained functional in other species and a coarse structural map of the zebrafish genome. With these results, we hope to build a foundation for future studies to completely annotate the cis-regulatory landscape and dissect the spatial organization of the zebrafish genome.
Introduction

In the past several decades, zebrafish (Danio rerio) have emerged as a prominent vertebrate model for genetic and genomic studies. A combination of high fecundity, rapid organogenesis, external fertilization, large number of offspring, relatively low infrastructure cost, and optical transparency at its embryo and larvae stages for in vivo visualization renders the organism efficacious for developmental biology investigations and large-scale genetic, phenotypic or pharmacological screens (Lieschke and Currie, 2007). Since zebrafish shares genetic similarity with humans as ~70% of human genes have zebrafish orthologues and vice versa (Howe et al., 2013), it has served as viable model organisms for many human diseases of various etiologies and pathologies such as Duchenne muscular dystrophy (Bassett et al., 2003), DiGeorge Syndrome (Piotrowski et al., 2003), polycystic kidney disease (Sun et al., 2004), melanoma (Patton et al., 2005), and myocardial infarction (Chablais et al., 2011). Despite the widespread usage of zebrafish models, however, there is still a considerable knowledge gap in the understanding of its genomic function and organization.

Over the past few years, the genome-wide functional noncoding elements have been characterized by large consortia for mammalian model organisms such as mice (Yue et al., 2014) and invertebrate models such as Drosophila melanogaster (Roy et al., 2010) and Caenorhabditis elegans (Gerstein et al., 2010). Meanwhile, our understanding of the cis-regulatory landscape of zebrafish remains coarse and incomplete. The knowledge derived from comparative genomics alone is insufficient: as a fish of the teleost infraclass of the ray-finned fish class, zebrafish has evolutionary ancestors who experienced an additional whole-genome duplication event (Meyer and Schartl, 1999), which created paralogous genes known as ohnologues. Furthermore, evolutionarily conserved
noncoding elements have mutated at higher rate in zebrafish compared to tetrapods, yet transgenic zebrafish assay of human noncoding elements lost to teleosts have demonstrated the conservation of certain trans-acting factors (Lee et al., 2011). To gain more insights into the genomic regulatory differences between fish and mammals, a more comprehensive annotation of the zebrafish functional genome is necessary.

Recent advances in high-throughput sequencing technologies have yielded ChIP-Seq (chromatin immunoprecipitation followed by DNA sequencing) (Barski et al., 2007; Johnson et al., 2007; Mikkelsen et al., 2007; Robertson et al., 2007), RNA-Seq (RNA sequencing) (Morin et al., 2008), and ATAC-Seq (assay for transposase-accessible chromatin with high throughput sequencing) (Buenrostro et al., 2013) that allow the genome-wide analysis of metrics of genomic function, including DNA transcription, post-translational histone modification, transcription factor binding and chromatin accessibility, as well as Hi-C (high-throughput C) (Lieberman-Aiden et al., 2009) and PLAC-Seq (proximity ligation-assisted ChIP-Seq) (Fang et al., 2016), which provide insight into the structure of the genome. We applied these techniques to profile the histone modifications and transcription levels of five zebrafish tissues and compared these results to human and mouse in order to understand the evolutionarily conserved as well as species-specific cis-regulatory landscape.
Methods

Determining tissue-specific gene expression

RNA-Seq fastq files of zebrafish brain, heart, kidney, liver and muscle were mapped to zebrafish genome assembly zv9 with Tophat (Trapnell et al., 2009) v.2.0.13 (default parameters). The transcriptome assembly of the bam files were achieved using Cufflinks (Trapnell et al., 2010) 2.2.1 (default parameters) with the RefSeq GTF file as guide. The RPKM (Reads Per Kilobase of transcript per Million mapped reads) values across the five tissues were standardized with quantile normalization (Bolstad et al., 2003). A tissue-specific gene was defined as the gene whose expression in one tissue was threefold enriched compared to its expression in the other four tissues. The gene ontology (GO) analysis was performed with Database for Annotation, Visualization, and Integrated Discovery (DAVID) (Huang da et al., 2009a, b).

Establishing putative promoters and enhancers

ChIP-Seq fastq files for H3K4me3, H3K27ac and input of zebrafish brain, heart, kidney, liver and muscle were mapped to zebrafish genome assembly zv9 utilizing bowtie (Langmead et al., 2009) 1.1.1 with parameters - -S -t -q -p 6 -n 2 -l 32 -e 3001. Peaks were called using Model-based Analysis of ChIP-Seq (MACS) (Zhang et al., 2008) 1.4.2 (default parameters) with the histone modifications (H3K4me3 or H3K27ac) and input for each tissue. After determining the midpoint of the peaks, the cis-regulatory element candidates were defined as the region ± 500 bps around the midpoints. Putative promoters were defined as regions with H3K4me3 peaks while putative enhancers were
defined as regions with H3K27ac distal (more than 3-kb away) to H3K4me3 peaks or transcription start sites (TSSs) of genes.

Finding tissue-specific putative enhancers

For each distal H3K27ac peak, the number of reads corrected by dividing the total number of mapped reads was determined. The normalized reads were then found by subtracting the corresponding corrected input reads from those corrected reads for each peak. A tissue-specific putative enhancer is defined as having twofold enrichment of number of resulting normalized reads compared to those of other four tissues. The regulatory domains surrounding the putative enhancers are annotated with GO terms with Genomic Regions Enrichment of Annotations Tool (GREAT) (McLean et al., 2010).

Discovering motifs

The motif discovery was accomplished using Hypergeometric Optimization of Motif Enrichment (HOMER) (Heinz et al., 2010) with default settings and one parameter change to -size 500).

Determining conservation

We downloaded the set of CNEs processed by the Bejerano Lab (Hiller et al., 2013) at http://bejerano.stanford.edu/zebrafish/. We overlapped this set of CNEs with our putative enhancer datasets with bedtools (Quinlan and Hall, 2010) v2.17.0. To determine whether these putative enhancers from zebrafish are functionally conserved in
human/mouse. We utilized the human/mouse coordinates of those CNEs whose sequences are conserved in human/mouse provided by the website and intersected them with, for humans, fetal brain male (E081), fetal heart (E083), fetal kidney (E086), liver (E066), and fetal skeletal muscle male (E107) H3K4me1 Broad Peaks datasets from the Road Epigenomics Project (Bernstein et al., 2010), and for mouse, fetal brain, combined fetal heart and adult heart, adult kidney, combined fetal liver and adult liver H3K4me1 Broad Peaks datasets from the Mouse ENCODE Project (Yue et al., 2014) as well as combined myoblasts and myotubes H3K4me1 peaks from the Asp et al (Asp et al., 2011).

**Processing Hi-C**

Hi-C fastq files were mapped to the zebrafish zv9 reference genome with BWA (Li and Durbin, 2009) 0.7.3a. Any duplicates from the resulting bam files were removed with markduplicates.jar from the Picard tools (http://broadinstitute.github.io/picard). All bam files were then merged with SAMtools (Li et al., 2009) and converted to contact matrices utilizing the pipeline from Dixon et al (Dixon et al., 2012). These matrices are then converted to BUTLR format and visualized with the 3D Genome Browser.
Results

Identifying putative promoters and enhancers in zebrafish

Various histone modifications are associated with certain cis-regulatory elements. The histone markers H3K4me1 and H3K4me3 correlate with active transcription (Barski et al., 2007) and H3K4me1 and H3K27ac with active enhancers (Creyghton et al., 2010). As H3K4me1 is associated in both promoters and active enhancers, in the interest of cost-effectiveness, it is omitted in favor of H3K4me3 and distal H3K27ac to mark promoters and active enhancers, respectively (Fig. 3-1).

Figure 3-1. Using histone modification to denote cis-regulatory elements

The genome-wide distributions of H3K4me3 and H3K27ac were uncovered with ChIP-Seq in five zebrafish tissues. These datasets, along with transcription levels determined by RNA-Seq were assessed with expression patterns of known genes. The gene glutamate decarboxylase 1b (gad1b), which encodes the enzyme that converts the glutamate into γ-aminobutyric acid (GABA) indeed exhibited brain-specific expression (Fig. 3-2a), consistent with known literature (Martin et al., 1998). Furthermore, both of its putative promoter and enhancer in the neighborhood could also be recognized as brain-specific (Fig. 3-2a). Similarly, the cardiac atrial myosin heavy chain 6 (mhn6) demonstrated specificity of the heart tissue in its expression as well as proximity to H3K4me3 (putative promoter) and distal H3K27ac (putative enhancer) (Fig. 3-2b).
Figure 3-2. UCSC browser snapshot of tissue-specific genes: 

a. UCSC Genome Browser snapshot of gad1b showing its specificity for brain, with proximal H3K4me3 (orange arrow) and distal H3K27ac (violet arrow); 

b. UCSC Genome Browser snapshot of myh6 showing its specificity for brain, with proximal H3K4me3 (orange arrow) and distal H3K27ac (violet arrow)
Determining tissue-specific genes

We identified tissue-specific genes based on RNA-Seq RPKM values. The brain possesses the most number of tissue-specific genes while heart possesses the least (Table 3-1). Hierarchical clustering of these tissue-specific genes recapitulated clusters based on germ layer origin of the tissue (Fig. 3-3). The gene ontology (GO) analysis of these tissue-specific genes yielded terms describing the structure and function of each tissue (Fig. 3-4).

Table 3-1. The numbers of tissue-specific genes for zebrafish brain, heart, kidney, liver and muscle.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Number of known tissue-specific genes</th>
<th>Number of novel tissue-specific genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brain</td>
<td>1156</td>
<td>594</td>
</tr>
<tr>
<td>Heart</td>
<td>276</td>
<td>188</td>
</tr>
<tr>
<td>Kidney</td>
<td>560</td>
<td>285</td>
</tr>
<tr>
<td>Liver</td>
<td>601</td>
<td>298</td>
</tr>
<tr>
<td>Muscle</td>
<td>313</td>
<td>165</td>
</tr>
</tbody>
</table>
Figure 3-3. Hierarchical clustering of tissue-specific genes.
Tissue-specific novel genes were then identified with the aid of Cufflinks (Table 3-1). Fig. 3-5 illustrates an example of an unannotated muscle-specific gene along with its putative promoter and candidate enhancer. We examined the number of unannotated transcriptional start site (TSS) which are around 25% of all putative promoters as denoted by the H3K4me3 histone modification (Fig. 3-6).
Figure 3-5. Unannotated muscle-specific transcript. UCSC Genome Browser snapshot of an unannotated novel (grey) along with muscle-specific putative promoter (orange arrow) and putative enhancer (violet arrow) in the vicinity.

Figure 3-6. Unannotated transcriptional start sites (TSSs). Pie charts showing the proportion of H3K4me3 peaks that overlap known TSS.
Uncovering tissue-specific putative enhancers

We then sought to determine putative enhancers for all the zebrafish tissues. About a half to two-thirds of the total H3K27ac peaks marked putative enhancers (Figure 3-7). Similar to our identification of tissue-specific genes, the brain possesses the most number of tissue-specific putative enhancers in contrast to the heart, which possesses the least (Table 3-2). The hierarchical clustering of these putative enhancers resulted in clusters based on the similarities in tissue function. (Fig. 3-8).

Table 3-2. The number of tissue-specific putative enhancers for zebrafish brain, heart, kidney, liver and muscle.

<table>
<thead>
<tr>
<th>Putative enhancers</th>
<th>Tissue-Specific</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brain</td>
<td>15,843</td>
<td>34,019</td>
</tr>
<tr>
<td>Heart</td>
<td>5,707</td>
<td>17,603</td>
</tr>
<tr>
<td>Kidney</td>
<td>5,925</td>
<td>17,895</td>
</tr>
<tr>
<td>Liver</td>
<td>5,994</td>
<td>21,703</td>
</tr>
<tr>
<td>Muscle</td>
<td>6,246</td>
<td>20,010</td>
</tr>
</tbody>
</table>

Figure 3-7. Percentage of Distal H3K27ac compared to total H3K27ac. Pie charts showing the proportion of distal H3K27ac peaks not proximal to H3K4me3 and known TSS and H3K27ac peaks proximal to H3K4me3 only out of the total number of detected H2K27ac peaks.
Figure 3-8. Hierarchical clustering of tissue-specific putative enhancers.

We then employed GREAT to annotate regulatory domains represented by these tissue-specific enhancer candidates and found that the resulting GO terms provided comparatively more specific terms (Fig. 3-9). For instance, instead of GO terms such as cation transport, we obtained more precise terms such voltage-gated calcium channel. Unfortunately, the GREAT analysis in kidney produced some puzzling terms. This could be due to its low mapping quality (perhaps because of contamination): only 42.6% of its H3K27ac reads successfully mapped, in contrast to those of other tissues, which ranged from 80.9% to 100.0%. Therefore, all future analysis of putative enhancer of the kidney must be interpreted with caution.
Figure 3-9. Gene ontology (GO) analysis of tissue-specific putative enhancers. The GO terms of tissue-specific putative enhancers are more specific compared to those of tissue-specific gene.

We then explored the TF motifs that reside within the tissue-specific enhancer candidates. The prediction recapitulated the TFs involved in the development or the function for each tissue (Table 3-3). For example, the motif for Atoh1, also known as
Math1, was detected as over-represented in the brain-specific distal H3K27ac peaks. Its inclusion was appropriate as in the hindbrain, Atoh1 has been shown to be crucial for the differentiation of cerebellar granule cells (Gazit et al., 2004). Again, the kidney received seemingly unrelated results: motif for hematopoietic cells. Since the kidney in zebrafish possesses a marrow that fosters hematopoiesis, this could be contributed to the function of the kidney rather than contamination (although it is possibly both).

Table 3-3. Transcriptional factor (TF) motifs predicted to be over-represented in each zebrafish tissue.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Motif</th>
<th>P-value</th>
<th>TF (Tissue)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brain</td>
<td></td>
<td>1E-89</td>
<td>Olig2 (Neuron)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-75</td>
<td>Zic (Cerebellum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-66</td>
<td>Mef2d (Retina)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-52</td>
<td>Atoh1 (Cerebellum)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-39</td>
<td>Lhx3 (Neuron)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-27</td>
<td>Sox10 (Sciatic Nerve)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-22</td>
<td>Ascl1 ( Neural Tubes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-16</td>
<td>PAX6 (Forebrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-14</td>
<td>Tbr1 (Cortex)</td>
</tr>
<tr>
<td>Heart</td>
<td></td>
<td>1E-53</td>
<td>Gata4 (Heart)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1E-2</td>
<td>Tbx20 (Heart)</td>
</tr>
<tr>
<td>Kidney</td>
<td></td>
<td>1E-5</td>
<td>GATA3 (iTreg)</td>
</tr>
<tr>
<td>Gene</td>
<td>Liver</td>
<td>Muscle</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Egr1 (K562)</td>
<td>1E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bZIP:IRF (Th17)</td>
<td>1E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ets1 (CD4+)</td>
<td>1E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HNF4a (HepG2)</td>
<td>1E-63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Erra (HepG2)</td>
<td>1E-50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>THRb (Liver)</td>
<td>1E-28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foxa3 (Liver)</td>
<td>1E-19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foxa2 (Liver)</td>
<td>1E-18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FXR (Liver)</td>
<td>1E-16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hnf1 (Liver)</td>
<td>1E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HNF6 (Liver)</td>
<td>1E-5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Srebp2 (HepG2)</td>
<td>1E-5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMAL1 (Liver)</td>
<td>1E-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Srebp1a (HepG2)</td>
<td>1E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cux2 (Liver)</td>
<td>1E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyoG (C2C12)</td>
<td>1E-17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Myf5</td>
<td>1E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Six1 (Myoblast)</td>
<td>1E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyoD (Myotube)</td>
<td>1E-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MafK (C2C12)</td>
<td>1E-5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Discovering conservation of cis-regulatory elements

To determine the conservation of the putative enhancers, we compared them to a set of conserved non-genic elements (CNEs) established computationally in a previous study (Hiller et al., 2013). This set containing 54,533 CNEs were discovered based on alignments to 15 vertebrates including other teleosts such as medaka, tetraodon, fugu and stickleback as well as human and mouse. For each tissue, around 6-12% of these putative enhancers were conserved across at least two of the vertebrate species while around 1-2% were conserved in human or mouse (Fig. 3-10). Brain was the most conserved tissue, followed by closely by heart while kidney was the least conserved (unclear if due to contamination). We then determined whether these sequence-conserved putative enhancers also retained their functions. To accomplish this, we utilized the human/mouse conserved CNEs and tested whether they overlapped with H3K4me1 peaks in human/mouse brain, heart, kidney, liver or muscle tissues. The H3K4me1 histone mark was selected due to the inconsistent availability of H3K27ac datasets. We found that 673/1047 (64.28%), 356/509 (66.64%), 144/278 (51.80%), 249/409 (60.88%) and 294/494 (59.51%) of CNEs from the zebrafish brain, heart, kidney, liver and muscle, respectively, were possibly functional in their human tissue counterparts. Meanwhile, 550/873 (63.00%), 262/419 (62.35%), 71/237 (29.96%), 172/344 (50.00%) and 174/412 (42.23%) CNEs were possibly functional in each corresponding mouse tissue. As an example of a putative enhancer that was conserved in sequence and function, we examined a CNE downstream of the Meox1 gene that was enriched in H3K27ac for both zebrafish and mouse (Fig. 3-11).
Figure 3-10. Conservation of putative enhancers across vertebrates. Pie charts illustrating the percentage of putative enhancers that are conserved in each zebrafish tissue.
Figure 3-11. Conservation of cis-regulatory function in zebrafish and mouse: a. UCSC Browser snapshot of putative enhancer on chromosome 12 in zebrafish; b. UCSC Browser snapshot of the sequence conserved putative enhancer on the corresponding region on chromosome 11 in mouse.

The GO analysis of these CNEs yielded terms such as regulation of metabolic process and regulation of transcription, suggesting these elements serve as basic essentials for the vertebrate life and development.
Exploring the spatial organization of the zebrafish genome

In order to gain insight into the 3D structure of the zebrafish genome, we performed Hi-C of the zebrafish muscle tissue. With 10,325,133 mapped reads, we were able to construct a 100-kb resolution contact matrix. The Hi-C data was then visualized using the 3D Genome Browser. At the current resolution, we noted TAD-like structures (Fig. 3-12), intimating that the spatial organization of zebrafish genome is similar to that of the mammalian genomes. Unfortunately, the current resolution is inadequate in visualizing finer structures. Higher resolution Hi-C and PLAC-Seq experiments are currently underway to link putative enhancers with their target genes.
Figure 3-12. Spatial organization of the zebrafish genome. 3D Genome Browser snapshot of the zebrafish muscle Hi-C contact matrix for chromosome 7.
Conclusion and Discussion

In this project, we set out to explore the cis-regulatory landscape in zebrafish by determining the distribution of histone modifications with ChIP-seq and associating it with gene expression detected by RNA-seq. H3K4me3 and distal H3K27ac were employed as markers of putative promoters and enhancer candidates, respectively. We found that about 71% of known TSSs possess H3K4me3 and annotating the unmarked TSSs using GREAT, we retrieved GO terms such as sensory perception of smell and phototransduction, intimating that the unmarked TSSs are from genes that perform specialized tasks, possibly tissue-specific genes from tissues not included in our study. Meanwhile, we established that distal H3K27ac was an appropriate indicator of putative enhancer, as the regulatory domain analysis of nearby genes yielded GO terms that matched the biological structure and function of all of tissues except kidney. Furthermore, motif analysis for the putative enhancers indicated by distal H3K27ac confirmed the possible binding of TFs specific to each tissue function or development. To further verify these putative enhancers, p300 occupancy could be examined with ChIP-Seq as its binding is highly associated with active enhancers (Visel et al., 2009). In addition to pursuing the cis-regulatory landscape of each tissue, we aspire to expand our study to include profiling the changes in the landscape during distinct time points in the course of development.

The increase in the specificity of GO terms derived from tissue-specific gene expression versus those stemming from tissue-specific putative enhancers could highlight certain properties of the cis-regulatory elements. The same cis-regulatory element could coordinate the expression of multiple genes that would define a terminally differentiated
Such sets of co-regulated genes are known as gene batteries (Britten and Davidson, 1969). The concept of gene batteries explains that tissue-specific expression of single genes alone cannot determine or maintain cell identity, rather it also requires the co-expression of a set of genes. Such insight raises the possibility that while expression of certain genes (e.g. trans-acting factors) might not be cell-specific, its effect based on co-expression with other genes could be.

Teleost fish possess rapidly evolving CNEs that outpace those of other vertebrates even after considering the whole genome duplications event (Lee et al., 2011), perhaps owing to higher selective pressures. Therefore, only 1-2% of CNEs in zebrafish could be recapitulated in human and mouse. A significant proportion of these CNEs overlap with the H3K4me1 from the corresponding tissue in human or mouse, suggesting the conservation of function in addition to sequence retention. To experimentally validate the functionality of these CNEs, we would utilize reporter assays to confirm the enhancer activities of these CNEs in zebrafish and in human/mouse. To further dissect functional conservation, we would determine the target genes of these CNEs with methods that detect the enhancer-promoter loop structure such as 4C and PLAC-Seq and confirm that they still regulate the same genes/gene batteries as their human/mouse counterparts. Furthermore, as human enhancer remaining functional within the in vivo zebrafish reporter assay (Kawakami et al., 2004) has demonstrated, certain trans-acting factors are conserved. To gain more insight into these factors, it is imperative to study them with large-scale ChIP-Seq experiments and regulatory network computational analysis.

The partitioning of the zebrafish genome into domains has been confirmed by 4C-Seq studies (Acemel et al., 2016; Gomez-Marin et al., 2015). Although our Hi-C data was of low resolution, we noticed TAD-like structures similar to those of the mammalian genomes. With higher resolution Hi-C, we hope to confirm the presence of TADs and
explore their properties in zebrafish. To link promoters and enhancers, we could perform high-resolution Hi-C to discover chromatin loops genome-wide. Unfortunately, such endeavor would be resource intensive and therefore unfeasible. The alternative would be to achieve enough resolution (e.g. 40kb) for the Hi-C dataset to visualize TADs. Since enhancers and their target genes most likely reside within the same TAD (Shen et al., 2012), we could limit our search space for target genes within the same TAD and correlate gene expression with intensity of the chromatin accessibility signal as well as correlate the intensities of chromatin accessibility signal with each other across multiple tissues similar to Thurman et al (Thurman et al., 2012). To accomplish this, we would need to first establish the accessibility landscape of the zebrafish genome by finishing our efforts in ATAC-Seq and as well as sequencing in other zebrafish tissues. Finally, we would confirm our results with other proximal ligation methods such as PLAC-Seq, which is also well-underway.
Chapter 4
Discussion – 3D Genome and Functional Non-Coding Elements of the Future

Finding the genomic structure of a single cell

In the previous chapters, we have explored proximal ligation 3C-based technologies and how they could provide insight into the possible target genes of cis-regulatory elements. Although Hi-C permits genome-wide interrogation of chromatin contacts, it possesses major weaknesses. In addition to the cost-prohibitive nature of high resolution Hi-C, conventional Hi-C contact maps could only encapsulate the superimpose structure of the cell population and therefore cannot account for cell-to-cell variability. Considering cell heterogeneity is crucial for organisms that exhibit large mix of different population of cells in a small space, such as zebrafish. In the first foray to tackle this challenge, small scale Hi-C (Noordermeer et al., 2011) and single cell Hi-C (Nagano et al., 2013) were invented, both demonstrating that individual cells adopt highly variable genome configurations. The technique has since then been propagated and developed into different improved iterations. In massively multiplex single-cell combinatorial indexed Hi-C (sciHi-C), we could identify the cell origins of Hi-C reads by “barcoding” each nucleus (Ramani et al., 2017). In another study, authors performed single Hi-C on haploid embryonic cell at the G1 phase, which further reduces the ambiguities of the chromosomal origin of the 3D structure (Stevens et al., 2017). The authors then solved the structure of the chromatin using computational modeling, reaffirming previous findings such as compartment A is located at the nuclear interior while compartment B and LADs are
relegated to the nucleolar and nuclear periphery. Furthermore, TADs exhibit different levels of compaction in different cells, intimating that they are not static, rather dynamic, structures.

Another strategy to analyze single cell chromatin structure lies in microscopy imaging. Although dismissed in the introduction as low-throughput, they are capable of capturing the 3D organization more organically and dynamically compared to the Hi-C contact heatmaps. Recent developments in custom oligonucleotide arrays such as Oligopaint (Beliveau et al., 2015; Beliveau et al., 2012) and super-resolution microscopy imaging (Rust et al., 2006) have permitted the direct visualization of the genomic structure at unprecedented resolutions. In one study, extensive DNA FISH experiments have confirmed the existence of TADs as well as the A and B sub-compartments that are spatially arranged at the opposite axis (Wang et al., 2016). With microscopy imaging, it would be feasible to visualize the changes in 3D structure of the single cell during a time course, which would be impossible by merely relying on Hi-C snapshots.

**Using Hi-C in genome assembly**

Despite its shortcomings, conventional Hi-C has provided immense insight into the spatial organization of the vertebrate genome. Furthermore, the observation that in Hi-C, chromatin contacts decay rapidly with distance has also rendered it effective in the genome assembling (Burton et al., 2013; Kaplan and Dekker, 2013). Recently, a group combined Pacific Biosciences (PacBio) long-read sequences, Bionano Genomics optical scaffolding and Hi-C proximal ligation mapping to perform *de novo* assembly of the genome from the domestic goat (*Capra hircus*) (Bickhart et al., 2017). This resulted in continuous sequences whose N50 is greater than 10 Mb, a feat that was achieved by
three mammalian species: human (International Human Genome Sequencing, 2004), mouse (Church et al., 2009) and gorilla (Gordon et al., 2016). The endeavor to assemble more genomes is currently underway (Dudchenko et al., 2017). Although only 3.78% of the zebrafish genome are currently not replaced within the 25 chromosomes, many genomic contigs were currently incorrectly assembled in the reverse orientation (Kelkar et al., 2014), which would thwart studies of its spatial organization. We hope to apply assembly with Hi-C scaffolding to the zebrafish genome to correct these errors before our structural analysis.

**Applying CRISPR-Cas9 screening for enhancers**

In our study of the zebrafish cis-regulatory landscape, we identified putative enhancers through ChIP-Seq of typical enhancer elements (H3K27ac in our studies, H3K4me1 and p300 are also markers). We must then test these putative enhancer through a reporter assay and perhaps identify their target genes through their contacts within the chromatin loop via proximal ligation methods such as PLAC-Seq. Recent advancement in genomics has enabled the development of screening tools for cis-regulatory elements. The II clustered, regularly interspaced, short palindromic repeat/CRISPR-associated protein 9 (CRISPR–Cas9) has emerged to become a prominent genomic/epigenomic editing tool (Cong et al., 2013; Jinek et al., 2012). The CRISPR–Cas9 system relies on the single guide RNA (sgRNA) to recognize a particular DNA sequence and direct Cas9, an endonuclease that introduce a double-strand break (DSB) near a short protospacer-adjacent motif (PAM) (Jinek et al., 2012). The DSB induces an endogenous DNA repair response that could be exploited to introduce mutations at the targeted site. In addition to its nuclease ability, Cas9 could function to
recruit proteins and RNA factors to the selected site; therefore nuclease-deactivated Cas9 (dCas9) could be fused with transcriptional activators and repressors to create a sequence-specific RNA-guided DNA-binding system (Gilbert et al., 2013; Qi et al., 2013). Such system could serve to physically hinder the transcriptional machinery to impede the transcription of the target gene, in which it becomes CRISPR interference (CRISPRi). Recent studies have taken advantage of the CRISPR–Cas9 platform to screen for enhancers. In one such study, tiling sgRNAs were utilized to target particular genomic regions, and by fusing dCas9 to the transcriptional repressor domain Krüppel associated box (KRAB), the functions of enhancers that become targeted were repressed and detected via a phenotypic screen (Fulco et al., 2016). In a similar fashion, another study operated with KRAB-dCas9 but choose to target all DHSs in a particular genomic region (Klann et al., 2017). A third study introduced cis-regulatory element scanning by tiling-deletion and sequencing (CREST-Seq), which employed overlapping sgRNA to guide the deletion of sequences and screen for those with deletions cis to the gene of interest (Diao et al., 2017). Although all effective methods of enhancer screening, the current tiling and screening approach could only focus on regions surrounding one gene of interest and would not scale well to a genome-wide screen. Still, the KRAB-dCas9 approach appears to be viable strategy in identifying functional enhancers in zebrafish in our endeavor to annotate the functional noncoding element in its genome.
References


zebrafish genome through an integrated transcriptomic and proteomic analysis. Mol Cell Proteomics 13, 3184-3198.


Appendix

BUTLR File Format Code

matrixToButlr.pl

#!/usr/local/bin/perl -w

#Yanli Wang
#3dgenome.browser@gmail.com
#Version 1.3
#Code to convert Hi-C matrix (contact matrices in multiple files) into the Binary Upper TrianguLer MatRix (BUTLR) format

use strict;
use warnings;

#Find the path of the required modules
use Cwd qw(abs_path);
use FindBin;
use lib abs_path("$FindBin::Bin/.");

#Required modules
use Getopt::Long qw(GetOptions);
use File::Basename;
##use List::Util qw(any);
use Scalar::Util qw(looks_like_number);
use Butlr;
no if ($\] >= 5.018), 'warnings' => 'experimental';
my $version = "1.3";
my $assembly;
my $genome_size_filename;
my $matrix_list_filename;
my $resolution;
my $output_filename;
my $header_end_n = 1;
sub get_use()
{
    return "BUTLR conversion tool version $version\n"
        "Converts contact matrices in tab-delimited text to
        BUTLR\n"
        "Usage: perl $0 <REQUIRED> <OPTIONAL>\n"
        "-a <assembly of data> [REQUIRED]\n"
        "-g <genome size file> [REQUIRED]\n"
        "tIf using new assembly, it is recommended to
        construct an assembly hub on UCSC and specify that URL, rather than
        default genome tracks.\n"
        "-g <genome size file> [REQUIRED]\n"
        "tA file with chromosome/scaffold names and their
        sizes, delimited by tab, one per line, like the *.chrom.sizes
        files.\n"
"-m <file containing list of chr *tab* matrix file location> [REQUIRED]\n"
"A file with chromosome/scaffold names and their corresponding matrix location/name, delimited by tab, one per line.\n"
"If interchromosomal interactions are included, then the file could, for each line, list chrom1, chrom2 and file, delimited by tab.\n"
"Please make sure that the chromosome/scaffold here matches those from the genome size file.\n"
"-r <resolution: bp as default, could enter Xk or Xm to specify resolutions in X kb or mb> [REQUIRED]\n"
"-h <row number when header ends/matrix begins, 1-based; default: 1 (no header)> [OPTIONAL]\n"
"-o <output butlr file> [OPTIONAL]\n"

GetOptions(
    'assembly|a=s' => \$assembly,
    'genome|g=s' => \$genome_size_filename,
    'matrix|m=s' => \$matrix_list_filename,
    'resolution|r=s' => \$resolution,
    'header_end|h=s' => \$header_end_n,
    'output|o=s' => \$output_filename,
) or die get_use();

unless (defined($assembly) &&
    defined($genome_size_filename) &&
"
defined($matrix_list_filename) && defined($resolution))
{
    die get_use();
}

my $res = get_resolution_in_bp($resolution);

# formulate the default output name if one is not provided by user
my ($basename, $dir, $ext) = fileparse($matrix_list_filename, qr/\.[^.]*/);
if (not defined $output_filename)
{
    $output_filename = "$dir$basename.$resolution" . "btr";
}

# read genome size file
if ( ! -e $genome_size_filename ) { die "$!
($genome_size_filename)\n" ];
my %genome_size;
open(FILE, $genome_size_filename) or die "Error opening
($genome_size_filename)\n";
while (my $line = <FILE>)
{
    chomp $line;
    my ($chr, $size) = split( ' ', $line);
    $genome_size{$chr} = $size;
}
close(FILE);

# sort by chromosome size, then by chromosome name (to put more "important" chromosomes first)
my @sorted_chr_list = ( sort { ($genome_size{$b} <=> $genome_size{$a}) || ($a cmp $b) } keys %genome_size );

my $interchrom_flag = 0;

# read matrix filenames
if ( ! -e $matrix_list_filename ) { die "$!
(matrix_list_filename)\n" }
my %matrix_name;
open(FILE, $matrix_list_filename) or die "Error opening
(matrix_list_filename)\n";
while ( my $line = <FILE> )
{
    chomp $line;
    my @recs = split( ' ', $line);
    if ( scalar(@recs) == 2 )
    {
        $matrix_name{ $recs[0] } = $recs[1];
    }
    elsif ( scalar(@recs) > 2 )
    {
        if ( $recs[0] eq $recs[1] )
        {
            $matrix_name{ $recs[0] } = $recs[2];
        }
$interchrom_flag = 1;

if ( is_chrom1_ahead($%genome_size, $recs[0],
$recs[1]) )
{
    $matrix_name{ $recs[1] . "\t" . $recs[0] } =
$recs[2];
}
else
{
    $matrix_name{ $recs[0] . "\t" . $recs[1] } =
$recs[2];
}
}
else
{
    die "The matrix file does not seem to be in the
correct format";
}
}

close(FILE);

open my $OUTFILE, "">", $output_filename or die "$!
($output_filename)\n";
binmode $OUTFILE;
#size of header (4-byte), accumulate size in this variable, then seek back to this location to overwrite it once all headers are written
my $head_size = 0;
print $OUTFILE pack("L", 0);
$head_size += 4;

#version of the code that produced the format (fixed 16 1-byte)
my @clist = convert_from_char_to_bin($version);
while ( scalar(@clist) < 16 )
{
    push @clist, 0;
}
print $OUTFILE pack("C*", @clist);
$head_size += 16;

#location of chromosome size/intrachromosomal
print $OUTFILE pack("L", 0);
$head_size += 4;

#location of interchromosomal information, 0 if does not exists
print $OUTFILE pack("L", 0);
$head_size += 4;

print STDERR "Assembly: " . $assembly . "\n";
#assembly
$assembly = trim($assembly);
print $OUTFILE pack("C*", convert_from_char_to_bin($assembly));
$head_size += length($assembly) + 1;

print STDERR "Resolution: " . $res . " bp\n";
#resolution (in bp)
print $OUTFILE pack("L", $res);
$head_size += 4;

my $mcv = 0.0;
#Most common value in matrix (zero for now)
print $OUTFILE pack("f<", $mcv);
$head_size += 4;

#Empty field holders x 4
print $OUTFILE pack("L", 0);
$head_size += 4;
print $OUTFILE pack("L", 0);
$head_size += 4;
print $OUTFILE pack("L", 0);
$head_size += 4;
print $OUTFILE pack("L", 0);
$head_size += 4;
print $OUTFILE pack("L", 0);
$head_size += 4;

#Denote the start of chromosome name, size, intrachromosomal information location
seek ($OUTFILE, 20, 0); #0 = SEEK_SET
print $OUTFILE pack("L", $head_size);
seek ($OUTFILE, 0, 2); #2 = SEEK_END, go to end of file
# stores location of the first row for each intrachromosomal, first with 0, then come back to rewrite to these locations

my %intrachrom_to_row_header_location;
my %intrachrom_to_row_body_location;

foreach my $chr (@sorted_chr_list)
{
    if (exists $matrix_name{$chr})
    {
        if ( ! -e $matrix_name{$chr}) {unlink $output_filename;

die "$! ($matrix_name{$chr})
"}
        print $OUTFILE pack("C*",
convert_from_char_to_bin($chr));
        $head_size += length($chr) + 1;
        print $OUTFILE pack("L", $genome_size{$chr});
        $head_size += 4;
        $intrachrom_to_row_header_location{ $chr } = $head_size;
        print $OUTFILE pack("Q", 0); # fill this later
        $head_size += 8;
    }
}

my %interchrom_to_row_header_location;
my %interchrom_to_row_body_location;
if ( $interchrom_flag )
{
    my $interchrom_loctn = $head_size;

    foreach my $c1 (0 .. $#sorted_chr_list)
foreach my $c2 (0 .. $c1)
{
    my $chrom1 = $sorted_chr_list[$c1];
    my $chrom2 = $sorted_chr_list[$c2];
    if ( $chrom1 eq $chrom2 ) { next; }

    my $key;
    if ( is_chrom1_ahead($genome_size, $chrom1, $chrom2) )
    {
        $key = $chrom2 . "$t" . $chrom1;
    }
    else
    {
        $key = $chrom1 . "$t" . $chrom2;
    }

    if ( exists $matrix_name{ $key } )
    {
        my $fname = $matrix_name{ $key };

        if ( ! -e $fname ) { unlink $output_filename; die "$!
($fname)\n"}

        print $OUTFILE pack("C*",
        convert_from_char_to_bin($key));

        $head_size += length($key) + 1;
    }
$interchrom_to_row_header_location( $key ) =
$head_size;
print $OUTFILE pack("Q", 0); #fill this later
$head_size += 8;
}
}
}

seek($OUTFILE, 24, 0);
print $OUTFILE pack("L", $interchrom_loctn);
}

#Write the header size
seek ($OUTFILE, 0, 0);
print $OUTFILE pack("L", $head_size);
seek ($OUTFILE, 0, 2);

my $file_location = $head_size;

foreach my $chr (@sorted_chr_list)
{
    if (exists $matrix_name{$chr})
    {
        my $total_bin_num = int($genome_size{$chr} / $res) + 1;
        my $total_matrix  = ( $total_bin_num ** 2 + $total_bin_num ) / 2;
        print STDERR "Processing $chr : $total_bin_num length and $total_matrix bins with $matrix_name{$chr}\n";
    }
my @row_locations;

if (open my $MXFILE, "<", $matrix_name{$chr})
{
    my $n = 0;
    #Store sparse matrix in memory
    my $i = 0; #row number

    while (my $line = <$MXFILE>)
    {
        $n += 1;
        if ($n < $header_end_n)
        {
            next;
        }
        if ($i >= $total_bin_num)
        {
            warn "Warning: larger matrix ($i >= $total_bin_num) than expected. Please make sure that the assembly genome file, matrix file and resolution are correct. If any header exists, please specify with the -h options. All excess rows toward the end were disregarded.\n";
            last;
        }
    }
    chomp $line;
    $line = trim($line);
    my @recs = split(' ', $line);
    my $rec_size = scalar(@recs);
# Make sure that the number of columns in the matrix file >= the calculated # of bin based on chromosome size and resolution

```perl
if ( $rec_size >= $total_bin_num )
{
    if ( $rec_size > $total_bin_num )
    {
        warn "Warning: larger matrix ($rec_size > $total_bin_num) than expected. Please make sure that the assembly genome file, matrix file and resolution are correct. All excess columns at the beginning were disregarded.\n";
    }
    @recs = splice(@recs, $rec_size - $total_bin_num, $total_bin_num);
    # record row location
    push @row_locations, $file_location;
    # record col location
    if ( $mcv ~~ @recs )
    #if ( any { $_ != $mcv } @recs )
    { for (my $j = $i; $j < $total_bin_num; $j++)

        { if ( $recs[$j] != $mcv )

            {
                my $v = $recs[$j];
                if ( looks_like_number($v) )
```
{ 
    if (lc $v eq "nan")
    {
        $v = 0.0;
    }
    if (lc $v eq "inf" || lc $v eq "-inf")
    {
        if (lc $v eq "inf")
        {
            $v = 1.0e38;
        }
        if (lc $v eq "-inf")
        {
            $v = -1.0e38;
        }
    }
    else
    {
        warn "Warning: non-number ($v) detected at row $i, column $j (0-based). It will be interpreted as 0.0.\n";
        $v = 0.0;
    }
}
print $OUTFILE pack("L", $j);
$file_location += 4;
print $OUTFILE pack("f<", $v);
$file_location += 4;
} else { unlink $output_filename; die "Error: Incongruent matrix sizes - not enough columns at $i (0-based) ($rec_size compared to the calculated bin number of $total_bin_num)\n";
}

$i += 1;
}

close $MXFILE;
push @row_locations, $file_location;
}
else { unlink $output_filename; die "$!
(matrix_name{$chr})\n";
      my $row_number = scalar(@row_locations);
      if ($row_number < $total_bin_num)
         { unlink $output_filename; die "Error: Incongruent matrix sizes - not enough rows ($row_number compared to the calculated bin number of $total_bin_num)\n";
         }
      $intrachrom_to_row_body_location{ $chr } = $file_location;
      for (my $row = 0; $row < scalar( @row_locations ); $row++)
         {
         print $OUTFILE pack("Q", $row_locations[$row]);
         $file_location += 8;
         }
      }
foreach my $chr (@sorted_chr_list)
{
    if (exists $matrix_name{$chr} && -e $matrix_name{$chr})
    {
        seek ($OUTFILE, $intrachrom_to_row_header_location{ $chr }, 0);
        print $OUTFILE pack("Q", $intrachrom_to_row_body_location{ $chr });
    }
}

seek ($OUTFILE, 0, 2);

if ( $interchrom_flag )
{

    #!!!!!!!!! Changed as of v1.2.2 to only process interchromosomal files with # row < # column
    foreach my $c1 ( 0 .. $#sorted_chr_list )
    {
        foreach my $c2 ( 0 .. $c1 )
        {
            my $chrom1 = $sorted_chr_list[$c1];
            my $chrom2 = $sorted_chr_list[$c2];
# Skip intrachromosomal datasets

if ( $chrom1 eq $chrom2 ) { next; }

my $key;

if ( is_chrom1Ahead(\%genome_size, $chrom1, $chrom2) )
{
    $key = $chrom2 . "\t" . $chrom1;
}
else
{
    $key = $chrom1 . "\t" . $chrom2;
}

if ( exists $matrix_name{ $key } )
{
    my $fname = $matrix_name{ $key };

    print $key . " : " . $fname . "\n";

    if ($fname)
    {
        my $chrom_a_bin = int($genome_size{ $sorted_chr_list[$c1] } / $res) + 1;
        my $chrom_b_bin = int($genome_size{ $sorted_chr_list[$c2] } / $res) + 1;
        my $total_bin = $chrom_a_bin * $chrom_b_bin;
    }
Print STDERR "Processing interchromosomal

$sorted_chr_list[$c1] ($chrom_a_bin rows) x $sorted_chr_list[$c2] ($chrom_b_bin columns) [total: $total_bin bins] with $fname\n"

# Get the number of rows and columns
my $num_of_row = 0;
my $num_of_col = 0;
open my $MXFILE, "<", $fname or die "$!";
while (my $line = <$MXFILE>)
{
  if ( $. == $header_end_n )
  {
    my @recs = split(' ', $line);
    $num_of_col = scalar(@recs);
  }
}
$num_of_row = $.;
if ( $num_of_row > $num_of_col )
{
  unlink $output_filename;
  die "Error: number of rows ($num_of_row) is

greater than number of columns ($num_of_col). Please transpose the

matrix and try again.\n"
}

###########################################################################
##########################
my @row_locations;
if (open my $MXFILE, "<", $fname)
{
    my $n = 0;
    my $i = 0; # row number
    while (my $line = <$MXFILE>)
    {
        $n += 1;
        if ($n < $header_end_n)
        {
            next;
        }
        if ($i >= $chrom_a_bin)
        {
            warn "Warning: larger matrix ($i >= $chrom_a_bin) than expected. Please make sure that the assembly genome file, matrix file and resolution are correct. If any header exists, please specify with the -h options. All excess rows toward the end were disregarded.\n";
            last;
        }
        chomp $line;
        $line = trim($line);
        my @recs = split(' ', $line);
        my $rec_size = scalar(@recs);
        if ( $rec_size >= $chrom_b_bin )
        {
            # Code continues here.
        }
    }
if ( $rec_size > $chrom_b_bin )
{
    warn "Warning: larger matrix
($rec_size > $chrom_b_bin) than expected. Please make sure that the
assembly genome file, matrix file and resolution are correct. All
excess columns at the begining were disregarded.\n";
}
@recs = splice(@recs, $rec_size - $chrom_b_bin, $chrom_b_bin);
#record row location
push @row_locations, $file_location;
#record col location
if ( $mcv ~~ @recs )
#if ( any { $_ != $mcv } @recs )
{
    for (my $j = 0; $j < $chrom_b_bin; $j++)
    {
        if ( $recs[$j] != $mcv )
        {
            my $v = $recs[$j];
            if ( looks_like_number($v) )
            {
                if (lc $v eq "nan")
                {
                    $v = 0.0;
                }
if ( lc $v eq "inf" || lc $v eq "-inf" )
{
    if ( lc $v eq "inf" )
    {
        { $v = 1.0e38; }
    }
    else
    {
        { $v = -1.0e38; }
    }
}
else
{
    warn "Warning: non-number ($v) detected at row $i, column $j (0-based). It will be interpreted as 0.0.
"
$v = 0.0;
}
print $OUTFILE pack("L", $j);
$file_location += 4;
print $OUTFILE pack("f<", $v);
$file_location += 4;
#matches if ( ($rec_size >= $chrom_b_bin) 

else { unlink $output_filename; die "Error: Incongruent matrix sizes - not enough columns at row $i (0-based) ($rec_size compared to the calculated bin number of $chrom_b_bin)\n"; } 

$i += 1;
}
}

push @row_locations, $file_location;
} #if (open my $MXFILE, "<", $fname)
else {unlink $output_filename; die "$!
($fname)\n"
}

my $row_number = scalar(@row_locations);
if ( $row_number < $chrom_a_bin )
{
    unlink $output_filename; die "Error: Incongruent matrix sizes - not enough rows ($row_number compared to the calculated bin number of $chrom_a_bin)\n";
}
$interchrom_to_row_body_location{ $key } =
$file_location;

for (my $row = 0; $row < scalar( @row_locations ); $row++)
{
    print $OUTFILE pack("Q",
$row_locations[$row]);
    $file_location += 8;
}

#foreach my $c2 ($c1 .. $#sorted_chr_list)
} #foreach my $c1 (0 .. $#sorted_chr_list)
} #if ( $interchrom_flag )

foreach my $c1 ( 0 .. $#sorted_chr_list )
{
    foreach my $c2 ( 0 .. $c1 )
    {
        my $chrom1 = $sorted_chr_list[$c1];
        my $chrom2 = $sorted_chr_list[$c2];
        if ( $chrom1 eq $chrom2 ) { next; }

        my $key;
        if ( is_chrom1_ahead(\%genome_size, $chrom1, $chrom2) )
        {
            $key = $chrom2 . "\t" . $chrom1;
else
{
  $key = $chrom1 . "\t" . $chrom2;
}

if ( exists $matrix_name{ $key } && -e $matrix_name{ $key } )

{
  my $fname = $matrix_name{ $key };
  seek ($OUTFILE, $interchrom_to_row_header_location{ $key }, 0);
  print $OUTFILE pack("Q", $interchrom_to_row_body_location{ $key });
}

}

}

close $OUTFILE;

print STDERR "Written $resolution" . " matrix as binary to
$output_filename\n";

butlrToMatrix.pl

#!/usr/local/bin/perl -w
Code to extract Hi-C matrix (contact matrices in multiple files)
from the Binary Upper Triangulated Matrix (BUTLR) format

use strict;
use warnings;

#Find the path of the required modules
use Cwd qw(abs_path);
use FindBin;
use lib abs_path("$FindBin::Bin/.");

#Required modules
use Getopt::Long qw(GetOptions);
use File::Basename;
use Butlr;

use lib qw(/hpc/home/yzw125/modulo/lib64/perl5/);
use List::Util qw(any first);

my $version = "1.3";
my $STDOUT  = "-";
my $butlr_filename;
my $output_prefix;
my $location_str;
my $bin_str;
my $usr_req_tp_flag;

my $chrom1_name  = "";
my $chrom2_name  = "";

my $UNINIT_VALUE     = -0.123456789;
my $chrom1_start     = $UNINIT_VALUE;
my $chrom1_endin     = $UNINIT_VALUE;
my $chrom1_start_bin = $UNINIT_VALUE;
my $chrom1_endin_bin = $UNINIT_VALUE;

my $chrom2_start     = $UNINIT_VALUE;
my $chrom2_endin     = $UNINIT_VALUE;
my $chrom2_start_bin = $UNINIT_VALUE;
my $chrom2_endin_bin = $UNINIT_VALUE;

my $DATA_BYTE_SIZE = 4;
my $LOCN_BYTE_SIZE = 8;

sub get_use
{
    return  "BUTLR conversion tool version $version\n".
           "Converts BUTLR to tab-delimited text format\n\n" .
           "Usage: perl $0 <REQUIRED> <OPTIONAL>\n\n".
           "-i <input butlr filename> [REQUIRED]\n\n".
           "-1 <location, 0-based, format: chromosome1[:start1-
end1],chromosome2[:start2-end2]> [OPTIONAL]\n".
}
"Extract the matrix based on the location.
Examples: chr1:0-100000 (both row and col of matrix) or chr1
chr1:0-100000,chr1:249150621-249250621 (row and col of same chromosome/intrachromosomal) or chr1 (entire chromosome)

chr1:0-100000,chr2:0-100000 or chr1,chr2
(whole interchromosomal) interaction

-b <bin (= location / resolution), 0-based, use bin instead of absolute position> [OPTIONAL]

-o <output prefix> [OPTIONAL] (Default: stdout)

If neither location or bin is provided, only the header information is printed.

GetOptions(
  'input|i=s' => $butlr_filename,
  'location|l=s' => $location_str,
  'bin|b=s' => $bin_str,
  'transpose|t' => $usr_req_tp_flag,
  'output|o=s' => $output_prefix,
) or die get_use();

unless (defined($butlr_filename))
{
  die get_use();
}
if (defined($location_str) && defined($bin_str))
{
    print STDERR "Both location and bin are defined. Please select one or the other.\n";
    die get_use();
}

#Input:
#  1) string: "chrom[:start-end]"
#Output:
#  array: (chrom, start, end)
sub extract_location
{
    my @loc_temp = split(":\", $_[0]);
    my $c = $loc_temp[0];

    if ( scalar @loc_temp == 1 ) #chr only
    {
        return (@_[0], $UNINIT_VALUE, $UNINIT_VALUE);
    }
    elsif ( scalar @loc_temp == 2 )
    {
        my @rec_temp = split("-", $loc_temp[1]);
        if ( scalar @rec_temp == 1 )
        {
            return ($c, $rec_temp[0], $UNINIT_VALUE);
        }
        elsif ( scalar @rec_temp == 2 )
        {

{ return ($c, $rec_temp[0], $rec_temp[1]); }
else #Inappropriate parsing of ':
{
    print STDERR "Error parsing the chromosome location information.\n\n";
    die get_use();
}
else
{
    print STDERR "Error parsing the chromosome location information.\n\n";
    die get_use();
}

#my ($basename, $dir, $ext) = fileparse($butlr_filename, qr/\.[^.]*/);
if (not defined $output_prefix)
{
    # $output_prefix = "$dir$basename";
}

#Input:
#  1) Pointer to BUTLR file
#  2) Number of bytes to read
Subroutine:

Read binary file into buffer

Output:

Binary String

sub read_bytes
{
    my $butlr_fp = $_[0];
    my $byte_num = $_[1];
    my $buffer;
    my $n;
    if ( $n = read( $butlr_fp, $buffer, $byte_num ) )
    {
        if ( $n != $byte_num )
        {
            die "[Error] File reading has yielded unexpected results (incorrect file size?).\n";
        }
    }
    else
    {
        die "[Error] File reading has failed.\n";
    }
    return $buffer;
}

Input:

1) Pointer to BUTLR file

Subroutine:

Reads binary files as characters and convert to string
sub read_chars
{
    my $butlr_fp =$_[0];
    my $c;
    my $string = '';
    while ( my $n = read $butlr_fp, $c, 1 )
    {
        my @clist;
        push @clist, unpack('C', $c);
        if ( unpack('C', $c) == 0 )
        {
            last;
        }
        $string .= convert_from_bin_to_char(@clist);
    }
    return $string;
}

# Input:
# 1) Pointer to BUTLR file
# 2) Location
# 3) Row
# 4) Column Start
# 5) Column End
# 6) Flag: true: interchromosomal; false: intrachromosomal
# Subroutine:
# Read butlr file to extract matrix
sub get_values {
  my $butlr_fp = $_[0];
  my $location = $_[1];
  my $row_offset= $_[2]; #i
  my $col_start = $_[3]; #j
  my $col_endin = $_[4];
  my $inter_flag= $_[5];
  
  if (!$inter_flag) {
    if ($row_offset > $col_start && $col_start == $col_endin) {
      my $temp = $col_start;
      $col_start  = $row_offset;
      $col_endin  = $row_offset;
      $row_offset = $col_start;
    }
  }
  
  seek( $butlr_fp, $location + $LOCN_BYTE_SIZE * $row_offset, 0 );
  my $row_locn_start = unpack('Q', read_bytes( $butlr_fp, $LOCN_BYTE_SIZE ));
}
seek( $butlr_fp, $location + $LOCN_BYTE_SIZE * ($row_offset+1), 0 );

my $row_locn_endin = unpack('Q', read_bytes( $butlr_fp, $LOCN_BYTE_SIZE ));

my %index_to_value;

for (my $i = $row_locn_start; $i < $row_locn_endin; $i+=$DATA_BYTE_SIZE*2)
{
    seek( $butlr_fp, $i, 0 );
    my $l = unpack('L', read_bytes($butlr_fp, $DATA_BYTE_SIZE));

    if ($l >= $col_start && $l <= $col_endin)
    {
        my $v = unpack('f<', read_bytes($butlr_fp, $DATA_BYTE_SIZE));

        $index_to_value{$l} = $v;
    }
    if ($l > $col_endin) { last; }
}

my @list;

for (my $i = $col_start; $i <= $col_endin; $i++)
{
    if (exists $index_to_value($i)) { push @list, $index_to_value($i); }
    else { push @list, 0.0; }
}
return @list;

if ( ! -e $butlr_filename ) { die "$! ($butlr_filename)\n" ];
open(my $butlr_fp, $butlr_filename) or die "Error opening ($butlr_filename)\n";

my $header_size = unpack('L', read_bytes($butlr_fp, $DATA_BYTE_SIZE));

my @clist = unpack('C*', read_bytes($butlr_fp, 16));
print STDERR " BUTLR Converter Version: " . convert_from_bin_to_char(@clist) . "\n";

print STDERR " Size of header: " . $header_size . "\n";

my $intra_locn = unpack('L', read_bytes($butlr_fp, $DATA_BYTE_SIZE));
print STDERR " Location of chr information: " . $intra_locn . "\n";

my $inter_locn = unpack('L', read_bytes($butlr_fp, $DATA_BYTE_SIZE));
print STDERR " Location of interchromosomal: " . $inter_locn . "\n";

my $assembly = read_chars($butlr_fp);
print STDERR " Assembly: " . $assembly . "\n";

my $res = unpack('L', read_bytes($butlr_fp, $DATA_BYTE_SIZE));
print STDERR " Resolution: " . $res . "-bp\n";

seek( $butlr_fp, 20, 1 );

my @sorted_chr_list;
my %chr_to_size;
my %chr_to_locn;
print STDERR "chromosome/scaffold\tsize\tlocation\n";

my $curr_byte = $intra_locn;
my $intra_end = $inter_locn;
if (! $intra_end) { $intra_end = $header_size; }
while ($curr_byte < $intra_end)
{
    my $chr = read_chars( $butlr_fp );
    push @sorted_chr_list, $chr;
    my $size = unpack('L', read_bytes($butlr_fp, $DATA_BYTE_SIZE));
    $chr_to_size{ $chr } = $size;
    my $location = unpack('Q', read_bytes($butlr_fp, $LOCN_BYTE_SIZE));
    $chr_to_locn{ $chr } = $location;
    $curr_byte += length($chr) + 1 + $DATA_BYTE_SIZE + $LOCN_BYTE_SIZE;
    print STDERR "\t$chr\t$size\t$location\n"
my $chrom1_rowcol_num;
my $chrom2_rowcol_num;

if (defined($location_str))
{
    my @chr_temp = split("", $location_str);
    ($chrom1_name, $chrom1_start, $chrom1_endin) = extract_location( $chr_temp[0] );

    if    ( scalar @chr_temp == 1 )
    {
        ($chrom2_name, $chrom2_start, $chrom2_endin) = ($chrom1_name, $chrom1_start, $chrom1_endin);
    }
    elsif ( scalar @chr_temp == 2 )
    {
        ($chrom2_name, $chrom2_start, $chrom2_endin) = extract_location( $chr_temp[1] );
    }
    else
    {
        print STDERR "Error parsing the chromosome location information.\n\n";
        die get_use();
    }
}
if ( !exists $chr_to_size{ $chrom1_name } )
{
    die "Chromosome/Scaffold $chrom1_name not in the file.\n";
}
if ( !exists $chr_to_size{ $chrom2_name } )
{
    die "Chromosome/Scaffold $chrom2_name not in the file.\n";
}
$chrom1_rowcol_num = int( $chr_to_size( $chrom1_name ) / $res ) + 1;
$chrom2_rowcol_num = int( $chr_to_size( $chrom2_name ) / $res ) + 1;
if ( $chrom1_start == $UNINIT_VALUE ) { $chrom1_start = 0; }
if ( $chrom2_start == $UNINIT_VALUE ) { $chrom2_start = 0; }
if ( $chrom1_endin == $UNINIT_VALUE ) { $chrom1_endin = $chr_to_size{ $chrom1_name }; }
if ( $chrom2_endin == $UNINIT_VALUE ) { $chrom2_endin = $chr_to_size{ $chrom2_name }; }
$chrom1_start_bin = int($chrom1_start / $res);
$chrom1_endin_bin = int($chrom1_endin / $res);
$chrom2_start_bin = int($chrom2_start / $res);
$chrom2_endin_bin = int($chrom2_endin / $res);
if (defined($bin_str))
{
    my @chr_temp = split("","", $bin_str);
    ($chrom1_name, $chrom1_start_bin, $chrom1_endin_bin) =
    extract_location($chr_temp[0]);

    if ( scalar @chr_temp == 1 )
    {
        ($chrom2_name, $chrom2_start_bin, $chrom2_endin_bin) =
        ($chrom1_name, $chrom1_start_bin, $chrom1_endin_bin);
    }
    elsif ( scalar @chr_temp == 2 )
    {
        ($chrom2_name, $chrom2_start_bin, $chrom2_endin_bin) =
        extract_location($chr_temp[1]);
    }
    else
    {
        print STDERR "Error parsing the chromosome bin
information.

";
        die get_use();
    }

    if ( !exists $chr_to_size{ $chrom1_name } )
    {
        die "Chromosome/Scaffold $chrom1_name not in the
file.
";
    }
}
if ( !exists $chr_to_size{ $chrom2_name } )
{
    die "Chromosome/Scaffold $chrom2_name not in the file.\n";
}

$chrom1_rowcol_num = int( $chr_to_size{ $chrom1_name } / $res ) + 1;
$chrom2_rowcol_num = int( $chr_to_size{ $chrom2_name } / $res ) + 1;

if ( $chrom1_start_bin == $UNINIT_VALUE ) { $chrom1_start_bin = 0; }
if ( $chrom2_start_bin == $UNINIT_VALUE ) { $chrom2_start_bin = 0; }
if ( $chrom1_endin_bin == $UNINIT_VALUE ) { $chrom1_endin_bin = $chrom1_rowcol_num - 1; }
if ( $chrom2_endin_bin == $UNINIT_VALUE ) { $chrom2_endin_bin = $chrom2_rowcol_num - 1; }

my $inter_chrom_jump = 0;
my %inter_chrom_table;
if ( $inter_locn )
{
    my $chroml_index = first { $sorted_chr_list[$_] eq $chroml_name } 0 .. $#sorted_chr_list;
}
my $chrom2_index = first { $sorted_chr_list[$_ ] eq $chrom2_name } 0 .. $#sorted_chr_list;

my $chrom1_exit = 0;
my $chrom2_exit = 0;
if (!defined($chrom1_index))
{
    $chrom1_exit = 1;
}
if (!defined($chrom2_index))
{
    $chrom2_exit = 1;
}

print STDERR "chromosome/scaffold1\tchromosome/scaffold2\tlocation\n";
while ( $curr_byte < $header_size )
{
    my $key  = read_chars( $butlr_fp );
    my $location = unpack('Q', read_bytes($butlr_fp,
$LOCN_BYTE_SIZE));
    $inter_chrom_table{$key} = $location;
    $curr_byte += length($key) + 1 + $LOCN_BYTE_SIZE;
    print STDERR "\t$key\t$location\n";
}

if ($chrom1_exit && $chrom1_name)
{

die "$chrom1_name is not encoded by the file";

if ($chrom2_exit && $chrom2_name)
{
    die "$chrom2_name is not encoded by the file";
}

}
else
{
    if ( $chrom1_name && $chrom1_name ne $chrom2_name )
    {
        die "The file does not encode interchromosomal matrices.

    }
}

unless (defined($location_str) || defined($bin_str))
{
    exit;
}

if ($chrom1_name)
{
    print STDERR " Location1 (binned): $chrom1_name : $chrom1_start_bin, $chrom1_endin_bin\n";
}

if ($chrom2_name)
print STDERR " Location2 (binned): $chrom2_name:
$chrom2_start_bin, $chrom2_endin_bin\n";
}

# Sanity checks
if ( $chrom1_name && $chrom2_name && ($chrom1_start_bin < 0 ||
$chrom1_endin_bin < 0 || $chrom1_endin_bin < $chrom1_start_bin ))
{
    die "The chromosome location/bin information is invalid.\n\n";
}

my @chrom1_list;
my @inter_chrom_list;
my @chrom2_list;
my $all_flag = 0;
my %done_list;

if ( !defined($location_str) && !defined($bin_str) )
{
    $all_flag = 1;
    @chrom1_list = @sorted_chr_list;
    @chrom2_list = @sorted_chr_list;
}
else
{
    push @chrom1_list, $chrom1_name;
push @chrom2_list, $chrom2_name;
}
my $out;

#Interchromosomal Locations
if ( $chrom1_name ne $chrom2_name || $all_flag )
{
    foreach my $chrom1 ( @chrom1_list )
    {
        my $reverse_flag = 0;
        foreach my $chrom2 ( @chrom2_list )
        {
            if ( $chrom1 eq $chrom2 ) {next;}
            if ( $all_flag )
            {
                $chrom1_start_bin = 0;
                $chrom2_start_bin = 0;
                $chrom1_endin_bin = int( $chr_to_size{ $chrom1 } / $res );
                $chrom2_endin_bin = int( $chr_to_size{ $chrom2 } / $res );
            }
            my $inter_chrom_jump = 0;
            my $key1 = $chrom1 . "\t" . $chrom2;
            my $key2 = $chrom2 . "\t" . $chrom1;
        }
if (defined($done_list{ $key1 }) || defined($done_list{ $key2 })) { next; }

if ( $inter_chrom_table{ $key1 } ) {
    $inter_chrom_jump = $inter_chrom_table{ $key1 };
    $done_list{ $key1 } = 1;
    $done_list{ $key2 } = 1;
}
else {
    if ( $inter_chrom_table{ $key2 } ) {
        $inter_chrom_jump = $inter_chrom_table{ $key2 };
        $reverse_flag = 1;
        $done_list{ $key1 } = 1;
        $done_list{ $key2 } = 1;
    }
}

if ( !$inter_chrom_jump ) { next; }

my $output_filename;

if ( defined $output_prefix ) {
    $output_filename = "$output_prefix.$chrom1.$chrom2.matrix";
}
if ( defined $output_prefix )
{
    open $out, '>', $output_filename;
}

if ($reverse_flag)
{
    my %location_to_value;
    for (my $i = $chrom2_start_bin; $i <= $chrom2_endin_bin; $i++)
    {
        my @row_list = get_values( $butlr_fp, $inter_chrom_jump, $i, $chrom1_start_bin, $chrom1_endin_bin, 1 );
        for (my $j = 0; $j < scalar @row_list; $j++)
        {
            if ( $row_list[$j] != 0 )
            {
                my $h = $j + $chrom1_start_bin;
                $location_to_value{ $i . "\t" . $h } = $row_list[$j];
            }
        }
    }
}

for (my $i = $chrom1_start_bin; $i <= $chrom1_endin_bin; $i++)
{
    my @row_list = get_values( $butlr_fp, $inter_chrom_jump, $i, $chrom1_start_bin, $chrom1_endin_bin, 1 );
    for (my $j = 0; $j < scalar @row_list; $j++)
    {
        if ( $row_list[$j] != 0 )
        {
            my $h = $j + $chrom1_start_bin;
            $location_to_value{ $i . "\t" . $h } = $row_list[$j];
        }
    }
}
my @row_list = ();
for(my $j = $chrom2_start_bin; $j <= $chrom2_endin_bin; $j++)
{
    if ( defined $location_to_value{ $j . 
        "\t" . $i })
    {
        push @row_list, $location_to_value{ $j . "\t" . $i };
    }
    else
    {
        push @row_list, 0;
    }
}

if ( defined $output_prefix )
{
    print $out join( "\t", @row_list ) . "\n";
}
else
{
    print join( "\t", @row_list ) . "\n";
}
else

for (my $i = $chrom1_start_bin; $i <= $chrom1_endin_bin; $i++)
{
    my @row_list;
    push @row_list, get_values( $butlr_fp, $inter_chrom_jump, $i, $chrom2_start_bin, $chrom2_endin_bin, 1 );
    if ( defined $output_prefix )
    {
        print $out join( "\t", @row_list ). "\n";
    }
    else
    {
        print join( "\t", @row_list ). "\n";
    }
}
if ( defined $output_prefix )
{
    close $out;
}
}

# Intrachromosomal Locations
if ( $chrom1_name eq $chrom2_name || $all_flag )
foreach my $chrom1 ( @chrom1_list )
{
    if ( $all_flag )
    {
        $chrom1_start_bin = 0;
        $chrom2_start_bin = 0;
        $chrom1_endin_bin = int( $chr_to_size{ $chrom1 } / $res );
        $chrom2_endin_bin = int( $chr_to_size{ $chrom1 } / $res );
    }

    if ( defined $output_prefix )
    {
        my $output_filename = "$output_prefix.$chrom1.matrix";
        open $out, '>', $output_filename;
    }

    my %location_to_value;
    for (my $i = $chrom1_start_bin; $i <= $chrom1_endin_bin; $i++)
    {
        for (my $j = $chrom2_start_bin; $j <= $chrom2_endin_bin; $j++)
        {
            if ($j + 1 == $i)
my @row_list = get_values( $butlr_fp, $chr_to_locn{ $chrom1 }, $j, $i, $chrom1_endin_bin, 0 );
for (my $h = 0; $h < scalar @row_list; $h++)
{
  my $k = $i + $h;
  $location_to_value{ $j . "\t" . $k } = $row_list[$h];
}
if ($j >= $i)
{
  last;
}
for (my $i = $chrom1_start_bin; $i <= $chrom1_endin_bin; $i++)
{
  my @row_list;
  for (my $j = $chrom2_start_bin; $j < $i; $j++)
  {
    if ( defined( $location_to_value{ $j . "\t" . $i } ) )
      push @row_list, $location_to_value{ $j . "\t" . $i };
  }
}
elsif ( defined( $location_to_value{ $i . "\t" . $j } ) )
{
  push @row_list, $location_to_value{ $i . "\t" . $j };
}
else
{
  push @row_list, 0;
}

push @row_list, get_values( $butlr_fp, $chr_to_locn{ $chrom1 }, $i, $i, $chrom2_endin_bin, 0 );

if ( defined $output_prefix )
{
  print $out join( "\t", @row_list ) . "\n";
}
else
{
  print join( "\t", @row_list ) . "\n";
}

if ( defined $output_prefix )
{
    close $out;
}
}
}
}

close $butlr_fp;
# VITA

**Yanli Wang**

## EDUCATION

<table>
<thead>
<tr>
<th>Institution</th>
<th>Program</th>
<th>Dates</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pennsylvania State University College of Medicine</td>
<td>M.D./Ph.D. Candidate</td>
<td>Aug 2010 - present</td>
<td></td>
</tr>
<tr>
<td>University of Pittsburgh</td>
<td>B.S. Computer Science and Biological Sciences</td>
<td>Aug 2006 – Dec 2009</td>
<td>Graduated <em>summa cum laude</em></td>
</tr>
</tbody>
</table>

## PUBLICATIONS


