COLLABORATIVE CROSS GRAPHICAL GENOME

Hang Su

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Bioinformatics and Computational Biology Curriculum.

Chapel Hill
2022

Approved by:

Leonard McMillan

Fernando Pardo Manuel de Villena

Martin T Ferris

Terrence Furey

William Valdar

ABSTRACT

Hang Su: Collaborative Cross Graphical Genome
(Under the direction of Leonard McMillan)

Reference genomes are the foundation of most bioinformatic pipelines. They are conventionally represented as a set of single-sequence assembled contigs, referred to as linear genomes. The rapid growth of sequencing technologies has driven the advent of pangenomes that integrate multiple genome assemblies in a single representation. Graphs are commonly used in pangenome models. However, there are challenges for graph-based pangenome representations and operations. This dissertation introduces methods for reference pangenome construction, genomic feature annotation, and tools for analyzing population-scale sequence data based on a graphical pangenome model. We first develop a genome registration tool for constructing a reference pangenome model by merging multiple linear genome assemblies and annotations into a graphical genome. Secondly, we develop a graph-based coordinate framework and discuss the strategies for referring to, annotating, and comparing genomic features in a graphical pangenome model. We demonstrate that the graph coordinate system simplifies assembly and annotation updates, identifying and segmenting updated sequences in a specific genomic region. Thirdly, we develop an alignment-free method to analyze population-scale sequence data based on a pangenome model. We demonstrate the application of our methods by constructing pangenome models for a mouse genetic reference population, Collaborative Cross. The pangenome framework proposed in this dissertation simplified the maintenance and management of massive genomic data and established a novel data structure for analyzing, visualizing, and comparing genomic features in an intra-specific population.

To Jesus Christ, my savior

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my savior Jesus Christ for leading and guiding me through this journey. The past 5 year at UNC is an adventure for me both physically, mentally and spiritually. I have no idea where I would end up without Him.

I would like to thank my advisor Leonard McMillan. He picked me up when I was lost like a homeless child. He welcomed me in his group and taught me everything, from the basic computer science and genetics knowledge, to how to develop idea and become an independent researcher. He holds group lunches, encourages collaborations, and push me coming out of my comfort zone to improve and grow. I could not have completed my dissertation without tremendous support from Leonard. I was fortunate to be his student during my PhD journey. His enthusiasm to science, creativity and innovation, and his generosity and kindness to people will keep guiding me in my future career. I would also like to thank my committees Fernando Pardo Manuel de Villena, Martin Ferris, Terrence Furey, and William Valdar for their constructive suggestions and help for my project and dissertation.

I would also like to thank my collaborators and labmates Ziwei Chen and Jay Rao. I learnt a lot from them in our collaboration. I would also like to thank Anwica Kashfeen, Sebastian Sigmon, Taksir Hasan Majumder, Yongze ”Sammy” Yin, Baqiao ”Charles” Liu and other lab members in CSbio group. I wish to thank all of the members of the Systems Genetics group, especially Fernando Pardo-Manuel de Villena and Martin Ferris for their suggestions and help for my understanding of CC and population genetics. I also want to thank all the support from Bioinformatics and Computational Biology program, especially Jonathon Cornett, and William Valdar for their assistance, patience and kindness for guiding me through the PhD study.

Finally, I want to thank my family, my mom Jing Cao and my dad Rongchi Su. Their unconditional love and support encourage me to get through all the difficulties in my life. Thank you to

Siyao Liu, Shu Zhang and all my friends at UNC and Duke for the laughter and every up and down moment sharing with you. A lot of joyful memories in the past 5 years are with you and getting to know you all is my fortune.

TABLE OF CONTENTS

LIST OF TABLES xii

LIST OF FIGURES xiii

LIST OF ABBREVIATIONS xvi

CHAPTER 1: INTRODUCTION 1

1.1 Reference Genome 2

1.2 Challenges of Sequence Analysis in Recombinant-Inbred Panel 6

1.3 Pangenome Models 8

1.4 Genome Graphs 10

1.5 Challenges in Pangenome Analysis 12

1.5.1 Graph Coordinate System 12

1.5.2 Graph Indexing 13

1.5.3 Sequence Analysis Based on a Graphical Pangenome 14

1.6 Contribution 15

1.6.1 Thesis statement 16

1.6.2 Innovation 16

1.6.3 Thesis Structure 18

CHAPTER 2: PREVIOUS WORK 19

2.1 Pangenome Construction and Representation 20

2.1.1 Inserting Well-documented Variants to a Reference Assembly 20

2.1.2 Assembly-based Pangenome construction tools 20

2.1.3 De Bruijn Graphs 21

2.1.4 Alignment-based Methods 21

2.2 Pangenome Analysis 22

2.2.1 BWT and FM Index 22

2.2.2 Graph Indexing and Graph Aligner 23

2.2.3 Genotyping in Pangenome Graph 25

2.2.4 Graphical Genome Browser and Visualization 25

2.3 Data Formats for Pangenome Representation 26

CHAPTER 3: GENOME REGISTRATION AND AN ANCHOR-BASED PANGENOME . 28

3.1 Background 28

3.2 Methods 31

3.2.1 Anchor Candidate Selection 31

3.2.2 Genome Registration and Anchor Selection 32

3.2.3 Edge Creation 34

3.2.4 Graph Annotation 37

3.2.5 Adding New Linear Genomes to the Graph 38

3.3 Results 40

3.3.1 Anchor-based Graphical Genome and Terminology 40

3.3.2 Graphical Genome Data Format 41

3.3.3 Computation Time for Adding Linear Assembly to the Graph 43

3.3.4 Detection of Structural Variants or Assembly Errors 45

3.4 Discussion 51

CHAPTER 4: COLLABORATIVE CROSS GRAPHICAL GENOME 53

4.1 Introduction 53

4.2 Methods 56

4.2.1 Identifying Anchor Candidates in the CC Population 56

4.2.2 CCGG Construction 56

4.2.3 Adding CC Paths ... 58

4.2.4 CCGG Annotation ... 59

4.3 Results ... 60

4.3.1 Collaborative Cross Graphical Genome ... 60

4.3.2 CC Linear Genomes Extracted from the CCGG ... 66

4.3.3 Feature Visualization in CCGG ... 73

4.4 Discussion ... 78

CHAPTER 5: VALIDATION OF THE COLLABORATIVE CROSS GRAPHICAL GENOME ... 82

5.1 Introduction ... 82

5.2 Methods ... 86

5.2.1 Anchor probes ... 87

5.2.2 Selecting Diversity probes ... 89

5.2.3 Fast bulk query algorithm ... 90

5.2.4 Probe Database for a Pangenome Model ... 91

5.3 Results and Discussion ... 91

5.3.1 A Probe Database for the Collaborative Cross ... 92

5.3.2 Visualization of the CCGG Probe Database ... 93

5.3.3 Validating the Sequence Content of the Graph ... 95

5.3.4 Adding Newly Released CC Sequenced Samples to the Graph ... 98

5.4 Conclusion ... 99

CHAPTER 6: GRAPH-BASED COORDINATE SYSTEM ... 101

6.1 Introduction ... 101

6.2 Method ... 103

6.2.1 Graph Coordinate System ... 104

6.2.2 Coordinate Mapping Method ... 106

6.3 Result ... 107

6.3.1 Mouse Reference Pangenome Model . . . 107

6.3.2 Graph Coordinate Framework . . . 108

6.3.3 Homologous Sequence Extraction and Comparison . . . 109

6.3.4 Updating New Mouse Reference Genome GRCm39 . . . 111

6.3.5 Graph Representation for Large Inversions . . . 114

6.4 Discussion . . . 117

6.5 Conclusion . . . 122

CHAPTER 7: GRAPH REFINEMENTS . . . 123

7.1 Introduction . . . 123

7.2 Method and Result . . . 125

7.2.1 Refining Sequence Content of CCGG . . . 125

7.2.2 Eliminate Redundant Paths . . . 126

7.2.3 Refining the CC Path on CCGG . . . 129

7.2.4 Integrating New Sequences to the Graph . . . 131

7.3 Future Direction . . . 133

7.3.1 Leveraging Long Read Sequences to Improve the Sequence Content of the CCGG . . . 133

7.3.2 Refine Recombination Boundary of CC Genomes . . . 136

7.3.3 Refine Graph Topology in Long Gaps . . . 138

CHAPTER 8: DISCUSSION AND CONCLUSION . . . 141

8.1 Summary . . . 141

8.2 Future Work . . . 142

8.2.1 Cancer Genomics . . . 142

8.2.2 Inter-specific Pangenome Models for Analyzing Sequence Data for PDX models . . . 143

8.2.3 Graphical Genome Browser . . . 144

8.2.4 Pangenome-based Association Study . . . 145

REFERENCES 146

LIST OF TABLES

Table 3.1 – Anchor Candidate Statistics 49

Table 4.1 – Statistics of CCGG Construction 58

Table 4.2 – Comparison of Overall Alignment Rate of CC MRCA Sequence data to GRCm38 and CC-specific Genomes 74

Table 6.1 – Sequence Similarity between Alternative Genome Assemblies and the GRCm38 Reference Genome 108

Table 6.2 – Statistics of Coordinate Mapping Accuracy 111

Table 6.3 – Inverted Regions on GRCm39 Relative to GRCm38 113

Table 6.4 – Reference-specific Inversion and its Coordinates. 113

LIST OF FIGURES

Figure 1.1 – Graphs ... 11

Figure 3.1 – Workflow for Genome Registration. ... 35

Figure 3.2 – Distribution of sequences shared by 8 founder genomes ... 37

Figure 3.3 – Demoting an Anchor ... 40

Figure 3.4 – An example of an anchor-based graphical genome ... 42

Figure 3.5 – Graphical Genome Representation ... 43

Figure 3.6 – Sequence Similarity and Diversity between DBA2 and CC Founders ... 46

Figure 3.7 – Distribution of Absent or Duplicated Anchor Candidates ... 47

Figure 3.8 – Distribution of Inverted or Translocated Anchor Candidates ... 48

Figure 3.9 – Mapping of inter-chromosome translocated anchor candidates between Mouse Reference Genome GRCm38 and the DBA2/J genome ... 50

Figure 4.1 – Genomes of Collaborative Cross (CC) mouse strains ... 57

Figure 4.2 – CCGG Properties ... 61

Figure 4.3 – Sequence Similarity and Diversity. ... 62

Figure 4.5 – Path Length Distribution ... 64

Figure 4.6 – CC Genomes Fix Variant Calls Resulted from Non-B6 Genetic Background ... 69

Figure 4.7 – CC Genome Capture the CC010 Insertion Relative to the GRCm38 reference Genome ... 70

Figure 4.8 – CC010 Deletion Relative to the Reference Genome ... 71

Figure 4.9 – Contrast of Read Alignment to GRCm38 and CC010 Genome ... 75

Figure 4.10 – Average number of Mismatches per Read Mapping to GRCm38 and CC010 Genome ... 76

Figure 4.11 – Distribution of Mismatch Numbers Aligning to GRCm38 and CC010 genome ... 77

Figure 4.12 – Graphical Genome Visualization of Gabra2 78

Figure 4.13 – Recombination in Genes 79

Figure 5.1 – Probe Selection 87

Figure 5.2 – The fast bulk query algorithm 88

Figure 5.3 – A probe database is constructed using the k-mer counts 92

Figure 5.4 – Visualization of the CCGG Probe Database 94

Figure 5.5 – Distribution of Unsupported K-mers on the CC-specific Path Versus the Reference Path in CC MRCA Data 96

Figure 5.6 – Potential Duplication Regions identified in the probe database 98

Figure 6.1 – Anchor Coordinate Framework 105

Figure 6.2 – Alignment-based Coordinate Mapping between Reference and Alternative Genomes 106

Figure 6.3 – Distribution of Haplotype Number in Gene and Exon intervals 110

Figure 6.4 – Visualization of the duplicated sequences on Chromosome 13 114

Figure 6.5 – Visualization of inverted anchor candidates 115

Figure 6.6 – Representation for large inversions on the graphical genome 116

Figure 7.1 – Distribution Ns and Unmapped Reads in CC010 Genome 127

Figure 7.2 – Distribution of mislabeled edges in CC010. 128

Figure 7.3 – Distribution of Fixed CC010 Variants 130

Figure 7.4 – Refined CC Haplotype Intervals 131

Figure 7.5 – Genome Refinement 132

Figure 7.6 – Integrating novel genomic discoveries to the CCGG 134

Figure 7.7 – Refining Sequence Content of CCGG 135

Figure 7.8 – Genomic Gaps represented by Ns 136

Figure 7.9 – Estimating the Actual Length of Ambiguous Regions 137

Figure 7.10 – Recombination Boundary of CC010 139

LIST OF ABBREVIATIONS

CC	Collaborative Cross
CCGG	Collaborative Cross Graphical Genome
NGS	Next Generation Sequencing
SNP	Single Nucleotide Polymorphism
VG	Variation Graph
BWT	Burrows–Wheeler Transform
FM index	Ferragina Manzini index
IUPAC	International Union of Pure and Applied Chemistry
JSON	JavaScript Object Notation
GAF	Graph Alignment Format
GMF	Graphical Mapping Format
GFA	Graphical Fragment Assembly Format
rGFA	Reference Graphical Fragment Assembly Format
GAM	Graph Alignment Map
PAF	Pairwise mApping Format
SAM	Sequence Alignment Map
BAM	Binary Alignment Map
BED	Browser Extensible Data
VCF	Variant Call Format
GFF	General Feature Format
GTF	General Transfer Format
PAV	presence/absence variations
HMM	Hidden Markov Model
MRCA	Most Recent Common Ancestor
ONT	Oxford Nanopore Technologies

CHAPTER 1: INTRODUCTION

The development of high throughput sequencing technology has revolutionized biomedical research and transformed the fields of genetics and genomics [1]. Sequence analysis and bioinformatics tools have become indispensable in studying disease, molecular biology, and genetics. Sequencing technologies, such as Next Generation Sequencing (NGS) or Third Generation Sequencing (long read sequencing, ONT or PacBio) are now a standard tool for scientific exploration.

One objective of sequence analysis is to establish a genome model, a reference genome, for a predefined population and often an entire species. Reference genomes, usually represented as a single-sequence genome assembly, are a fundamental resource for analysis [2, 3]. The quality and accuracy of the reference genome is critical for analyzing and interpreting the sequence data. However, using a single-sequence assembly as a reference for an entire population has limitations [4, 5]. The advances in sequencing technology have enabled an increased number of high-quality individualized assemblies but these are often difficult to contrast and compare [6]. This motivates the development of pangenome models, and compatible tools, to incorporate multiple genome assemblies to be used as a reference. Pangenome models not only incorporate additional sequence diversity information, which improves the accuracy of sequence analysis, but they also provide a computationally efficient framework for data analysis and interpretation. The main objectives for pangenome models include reducing bias, compressing redundant sequences shared between assemblies, and providing a framework for feature annotation, resource management, genome comparison, and visualization [4, 5]. The paradigm shift from linear reference genomes to pangenomes will transform the field of genetics and provide new insights to biomedical research [4].

In this thesis, I propose a reference pangenome model for a specific genetic reference population of recombinant-inbred mouse known as the Collaborative Cross (CC). This model represents the genomes of all available CC strains and their eight founder strains as a graph, and it is called the Collaborative Cross Graphical Genome (CCGG). It describes a process for constructing a series-parallel sequence graph from the eight CC founder linear genome assemblies. The founder assemblies as well as recombinant genomes for each CC strain can be extracted from the pangenome graph in a standard sequence file format, which can be used in common bioinformatics pipelines. A CC probe database of representative k-mers was constructed, covering every base pair of the graph and the k-mer frequency was counted in sequenced datasets of both individual and pooled CC samples. This effectively validated the sequence content of the pangenome model based on the CC probe database. The k-mer query tool serves as an alignment-free method for population-scale sequence analysis based on a pangenome model. I also discuss the operational strategies for referring to, annotating, and comparing genomic features in a pangenome graph. This includes developing a graph-based coordinate framework to identify homologous base pairs and calculate distances in a multi-genome context. We show that our graph-based coordinate system simplifies the maintenance and update of genome assemblies and annotations.

This Chapter discusses the important roles of reference genomes, the standard representation of reference genomes, and their limitations. It introduces the concept of a pangenome model and the related notations and terminology. It then provides a brief review of the current state-of-the-art in pangenome representations, discusses the challenges of applying pangenome models in the sequence analysis, and their practical limitations. Finally, I summarize the innovations and contributions of this thesis, and discuss the structure of this dissertation.

1.1 Reference Genome

The recent revolution in sequencing technologies has enabled the creation of multiple genome models each representing an entire species [2]. The first complete genome assemblies was released in 1995, for Haemophilus influenzae [7]. Later, the first eukaryotic genome, Saccharomyces cerevisiae genome, was sequenced and assembled [8]. Soon afterward, the genomes of other bacteria and archaea were sequenced and assembled. Later, in 2001, a draft of the entire human genome is released [9, 10]. In 2002, a genome assembly of the laboratory mouse, Mus musculus, was completed [11]. These sequence assemblies have been utilized as *reference genomes* and are widely used in the bioinformatics and genomic study. Reference genomes serve as a foundation for most bioinformatics pipelines [3]. They provide coordinate frames for referring to and annotating biological or sequence features[5], and serve as the substrates for computational analysis and functional discoveries. Reference genome assemblies also enable genetic mapping, pattern searching, and feature comparisons. Reference genomes provide information for examining unannotated genomic regions and are invaluable resources for wet-lab assays including genotyping probe and PCR primer designs.

Reference assemblies commonly take the form of a collection of assembled contigs [4], ideally with one contig per chromosome. We refer to these single-sequence reference assemblies as *linear genomes*. Linear genomes are well established frameworks for functional annotation and analyzing the resequenced samples from the same species [12]. In standard practice, sequenced reads are aligned to the reference and genetic variations are identified relative to the reference. Linear genomes provide a straightforward way of referring to genomic position by specifying the contig name and indices. In this framework, genomic features are represented by indices of one or more genomic intervals [13]. For example, most gene, exon and transcript annotations are reported as one or more genomic intervals in a linear reference coordinate system. Variants are reported relative to the reference genome using these same linear coordinates. This coordinate framework has been applied in multiple file formats, including SAM [14], BED [15], and VCF [16]. The use of linear reference genomes has shaped method development and data formats across the whole genomics and bioinformatics field.

However, linear reference assemblies come with limitations [2, 17, 4, 3]. Reference genomes serve as a template for read mapping, variant calling, and providing a coordinate framework for function annotation and experiment design. However, reference genomes are usually derived from a limited set of samples per species [12]. For example, the mouse reference assembly represents the genomic sequences collected from a single common mouse strain, C57BL/6J [11]. The human reference genome is derived from a specific set of individuals and largely drawn from a western European population [9, 10]. Thus, reference genome assemblies unintentionally represent sample-specific genome organizations. They are not a “common” genome, nor the most “comprehensive”, nor the most “conserved”, nor the most “healthy” [3] representation. As such, the reference alleles may not always represent the consensus of the species, they may even be the be the rare alleles or high-risk alleles over the larger population [18]. These biases and errors are hard to identify and evaluate, as most of the genomic pipelines adopt reference genome as the standard. Frequently only read sequences similar enough to the reference assemblies are taken into consideration in the downstream analysis [3]. This tendency to prefer sequences similar to the reference genome and ignore those that are inconsistent is referred to as *reference bias*. Reference bias leads to issues such as poor characterization of highly variable regions or regions with assembly errors in the reference genome [19]. Reference bias has huge impact in genomic regions with structural variants or the sequences that are absent from the reference genome. When differential mapping matters, such as in the study of allele-specific expression, the reference bias also has significant impact [12]. In the genomic regions with reference-specific features, the variant caller tends to call more variants that are, in fact, common in the population, but ignore the reference-specific variants, or even high risk alleles that are shared by the reference genomes [20, 21].

In addition, linear reference genomes fail to model the genetic variations within a species. Many efforts have focused on resequencing human genomes in population-specific cohorts as well as world-wide populations [22, 23, 24, 25]. These studies have found many novel sequences that are missing from the reference assemblies and the missing sequences are estimated up to 10% of the total genome size [26, 27]. Reads from these non-reference regions tend to be neglected or misinterpreted in the common bioinformatics pipelines. In order to create a comprehensive catalogue of human genetic variation, the HapMap [28] and 1000 Genomes Project [29] collected samples in an effort to cover a larger cross section of the human population, from individuals assumed to be free of genetic defects. However, incorporating this additional genomic variation information into common bioinformatics analysis still remains challenging. The information of sequence variations is conventionally stored in auxiliary databases to support the downstream analysis, such as VCF files. Yet the VCF files are unable to fully describe complex genomic regions or regions with large structural variants. In fact, it is estimated that the human genome contains about 2500 large SVs [25, 30]. Recent advances in long read technology revealed that the SVs are even more prevalent in human genomes than predicted [31]. Many large-scale genomic variations have been discovered in other species, including laboratory mouse [6], Arabidopisis thaliana [32], etc.

Reference assemblies are also continually being updated and improved. Many efforts to improve the reference assemblies have been done to fill in the gaps of the reference assemblies [2]. Recently more and more work has been shifting towards incorporating more sequence diversity to the reference assemblies [4]. However, any changes to the reference assemblies will require a significant effort to adapt to the new standard. For example, with the changes of coordinate framework, all the feature annotations such as variants, gene or exon intervals, have to be updated and transferred to the new coordinate system. All of the biological discoveries reported relative to the primary reference assembly have to be re-evaluated and updated as well. Moreover, with the growing number of available genome assemblies, there are challenges in the bioinformatics analysis to decide which assembly to use as the reference or how to integrate the multiple genome assemblies in downstream analysis. Bioinformaticians are pursuing solutions to manage these genomic resources consistently and integrate them altogether to analyze the sequence data.

These shortcomings of linear reference genomes has driven a paradigm shift towards pangenome representations[4, 5]. In the case where a population is derived from a known set of of ancestors, a linear genome derived from a single strain is not sufficient to be deemed as reference. A complete pangenome would be a more comprehensive model to be used as reference for such populations. These populations are common in model organisms, where they are commonly used for genetic mapping. One specific type of model population are recombinant-inbred panels that are derived from a common set of ancestors.

1.2 Challenges of Sequence Analysis in Recombinant-Inbred Panel

Recombinant inbred (RI) strains are important resources for genetic mapping [33]. RI strains are usually formed by crossing fully inbred strains, namely F1 cross, and then intercrossing these to get a mixed F2 generation [33]. After one or more intercross generations RI lines are then developed by continuous sibling mating until the genome mosaic is deemed fixed (i.e. until the strain is fully homozygous) thus establishing an new inbred line [33]. RI strains provide stable and replicable genomes for genetic study and the sequence variants that are largely common for the same line [34]. Genotyping only needs to be performed once for each strain of an RI panel, not for each individual as in F2 studies. RI strains have been widely used to study genetic factors underlying phenotypic differences and responses to environmental stimulus [35]. The Collaborative Cross (CC) is a panel of multi-parental recombinant-inbred mouse strains [34] all sharing a common set of ancestors, which are called the CC *founders*. CC mouse are derived from eight genetically diverse inbred strains, including A/J, C57BL/6J, 129S1/SvImJ, NOD/ShiLtJ, NZO/H1LtJ, CAST/EiJ, PWK/PhJ and WSB/EiJ [34]. These 8 founder strains represent the three major subspecies of house mouse . The CC panel serves as the genetic reference population in mouse species and has been widely used to study complex traits [34, 35].

The genome of each CC strain is a unique mosaic of the founder's genomes. Aside from the fact that all CC strains are derived from a common set of founders, the strains are otherwise independent (unrelated). Each CC strain can be modeled as a random assignment of chromosome blocks of different founder strains [33]. CC strains exhibit extensive genetic diversity, and provide a enriched resource for genetic mapping [37, 38, 39, 40, 41, 42]. However, the random mixing of genomes also introduces complications when analysing CC strains. As described previously, the first step of most of the sequence analysis pipelines is to align reads to a reference genome assembly. Typically, reference genomes represent the genome organizations of a single individual sample[3].

For example, the reference genome of *Mus Musculus* represents the genome of a single inbred strain C57BL/6J [11]. When mapping reads of RI samples to a standard reference genome, a reference bias is introduced that can have significant impacts for the regions inherited from non-reference strains, which might mislead the downstream analysis and data interpretation. For small SNPs or indels, these variants will cluster in the region where the genetic background of RI strain is significantly different from the reference. Instead, the true variants existing in the reference assembly are ignored. For larger sequence differences, reads from a genomic region that either not present in the reference genome or organized differently, tend to be neglected or misinterpreted in the analysis. Overall, a single-sequence reference assembly is insufficient for the sequence analysis of RI strains.

To address these issues, different approaches have been developed. In the previous work, pseudogenomes were constructed by inserting variants to a reference assemblies and a general framework for mapping annotation between these genome models was developed [43]. However, the inserted variants are mainly SNPs or small indels, while large structural variants, such as inversions, duplications, or translocations, are hard to implement in pseudogenomes. In addition, a multi-alignment pipeline has been developed to map reads to multiple reference assemblies and merge the result afterward [44].

Since the CC strains are descended from 8 inbred strains, others has tried aligning reads to all 8 linear genomes and then merging the results [45]. This approach is computationally expensive, and the merging step involves many heuristic decisions. Another alternative is to develop a consensus genome assembly for the RI panel that the treats ancestors of the RI strains more uniformly. But for sure, information of sequence diversity will be lost in the consensus genome as linear genomes are not capable to capture alternative alleles. Instead, pangenome models provide a promising solution for RI genome representation and genetic analysis. In the next section, we will talk about the concept of pangenome model and its potential applications in RI strains.

1.3 Pangenome Models

Pangenomes are defined as a set of genomic assemblies that are analyzed together as reference [4]. Broadly speaking, a pangenome model represents the genomic sequences of an intra-specific population, a species, a clade, or a metagenome [4]. Here we will focus on a *reference pangenome* as being both population-specific, and within species [46], in contrast to either a inter-specific *phylogenetic pangenome* that represents an evolutionary history of presently diverged clades [47, 48], or a *mutagenic pangenome* that represents somatic DNA variations within a single organ such as those proposed for cancer study [49]. Pangenomes have been first studied in microbiology [48, 50, 51, 52, 53], where genomic diversity and plasticity play roles on the adaptive fitness and pathogenicity of a strain. For example, Tettelin et al characterized *Streptococcus agalactiae* pangenome and found out that it is ’open’, i.e. the pan-genome size is infinite as more samples are added in [48]. Pangenomes also provide an enriched data structure, which allows for testing evolutionary hypotheses in bacteria [54]. Later on, many studies apply pangenome model to eukaryotic organisms [55, 56]. Li et al integrated the Asian, African de novo genome assemblies and the human reference genome in a pangenome model and identified novel sequences not present in the standard human reference [55]. Eupan, an eukaryotic pan-genome analysis toolkit, enables pangenome analysis for large eukaryotic genomes and benchmarked by constructing a pangenome with 453 rice genomes [56]. The widespread gene PAVs (Present Absent Variations) among rice genomes were detected [56].

The pangenome of a species consists of three differentiated components: core genome, the accessory genome, and strain-specific genome [54]. Core genome refers to the regions or genes that are present in every individual of the population. Accessory genome refers to the region or genes that are present in a subset of the population. The strain-specific genome refers to the genes that are present in a single isolate. Accessory genome and the strain-specific genome are also referred to as dispensable genomes [27]. The core and dispensable genomes comprise of pangenomes altogether. For prokaryotic genomes, the pangenome is defined at gene level, as their genomes are predominantly composed of genes, with little intergenic regions [27]. For eukaryotic genomes, pangenomes usually represent all sequences in their genome, including both genes and intergenic regions [27].

A pangenome represents multiple alternative DNA sequences, this requiring a more complex data structure than a single-sequence reference genome. Much of the efforts on pangenome representation have focused on minimizing redundancy [57, 58], estimating genome size [59], supporting the addition of new genomes [5], and establishing platform for sequence comparison [60, 13, 61]. One of the goal for pangenome representation is to reduce reference bias by relating sequences to a pangenome model with sequence diversities [5]. The desirable features of a pangenome include *completeness*, *stability*, *comprehensiveness*, and *efficiency* [4]. *Completeness* refers to incorporate genetic diversities or alternative sequences in the reference for analysis. *Stability* requires the pangenome provide a stable framework for assembly update or graph modification. *Comprehensibility* enables population-scale genomic study based on a pangenome model. *Efficiency* requires the pangenome provides an efficient data structure that facilitate downstream analysis. Significant efforts are required to achieve these goals, including an increasing amount of genome resources for the species or organisms of interest, a public platform for long-term resource sharing and maintenance, and the development of algorithms and methods to analyze the tremendous amount of data based on a pangenome model. The additional information provided by a pangenome model allows the bioinformatics and genomic pipelines to achieve better performance for read alignment, variant calling and genotyping [4]. Incorporating the sequence diversity in the population is essential for genome-wide association studies [30]. Using the graph-based pangenome model as a reference also introduces a novel data structure and format that benefits for the maintenance and management of the genomic resources.

Pangenome models provide an accessible and computationally efficient framework to deal with the issues in genetic analysis of RI strains. By integrating multiple ancestor linear genomes in a single pangenome graph and label the paths based on the haplotype information of each RI strain, the genome of RI strains can be represented as the recombinant paths of the genome graph. For genetic study focusing on a single RI strain, one extracts the specific linear genome of the RI strain from the pangenome and using public available tools to perform downstream analysis. For functional study, one can focus on a specific genomic region of interest, extracting the subgraph in the pangenome to compare distinct haplotypes. For the study for F1 or F2 strains, one can focus on a subset of the paths in the pangenome graph to perform analysis. One can also construct a consensus sequence from the pangenome graph to construct unbiased linear models. For population-scale sequence analysis, pangenome model incorporates extensive sequence variations in a single framework, which benefits for conduct analysis and comparison within a population than any linear model.

1.4 Genome Graphs

Graphs have been widely used in sequence analysis by providing a compact data structure to perform analysis, such as pattern searching [4, 30]. In general, we call the graphs that are used to represent a pangenome model as *Genome graphs* or *Sequence graphs* [5]. A graph usually compose of *nodes* and *edges*. Nodes often encode nucleotide sequences. Nodes are connected by edges. Edges are either directed or bidirected depending on the type of sequence graph. The degree of a node is the number of edges connecting with that node. For a directed graph, the indegree and outdegree of a node is the number of edges coming in and going out respectively. *Paths* or *Walks* refer to a series of graph entities that are passed by when traversing the graph. *Circuits* are walks that loop back to a same node. An *acyclic sequence graph* is a graph having no graph circuit. A *cyclic sequence graph* is a graph containing at least one circuit. A *bubble* is a directed acyclic subgraph separated by a pair of nodes [30]. They are often applied to represent parallel “ref|alt” alleles. Sequences are naturally implemented as a walk or a path in a sequence graph and can be reconstructed and extracted through graph traversal. The nodes shared by all paths in the graph can potentially serve as graph coordinates (Chapter 6).

A *directed graph* refers to a graph where the edges are directed, i.e. they asymmetrically connect one node to the other. De Bruijn graphs are typical directed graphs that have been widely used in genome assembly or read mapping [62, 63]. The nodes in a *de Bruijn* graph represents k-mers (substrings of length k). The directed edges in *de Bruijn* graphs represent the k-1 characters

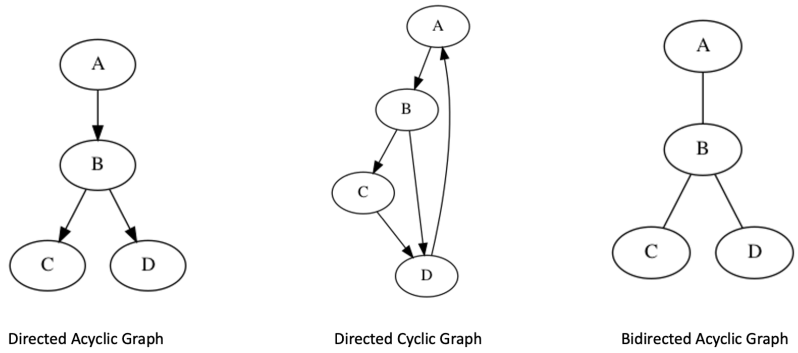


Figure 1.1: Different Types of Graphs. The directed acyclic graph, directed cyclic graph and bidirect acyclic graph are shown in this figure

overlapping between the prefix and suffix nodes. A *bidirected graph* refers to the edges connected two nodes with no specific orders. The strandness of nucleotide sequences can be implemented in the direction of edges, indicating whether the forward or the reverse complement of the sequences should be extracted [5]. This also enables sequence graph to represent complex genome organizations such as inversions or translocations, which are hard to reveal by colinear sequence comparison among multiple sequences.

A subset of graph with special graph structure, called *series-parallel graph*, refers to graphs with two distinguished vertices called terminals [64]. The series-parallel graphs can be constructed by a sequence of “series” and “parallel” compositions. Many combinatorial problems, which are NP-complete in general graphs, can be resolved in linear time if the input graph are restricted to the series-parallel graphs [64]. On the other hand, series-parallel graphs are a subset of planar graphs, i.e. the graphs that can be displayed on the plane that no edges cross each other. These desirable properties of series-parallel graphs make it a good choice for sequence graph representation, which benefit for pattern searching and genome visualization.

1.5 Challenges in Pangenome Analysis

Pangenome models provide comprehensive genomic information and a novel dataframe for sequence compression and management. However, pangenomes application requires the development of new data formats and compatible methods for querying, and operating on these pangenome graphs. The incorporation of pangenome models into the existing genomic practices faces challenges. Different methods adopt distinct types of graphs to achieve sequence compression and features representation, yet requires set of tools to deal with the specific graph structure [12]. In this section, we will review the major issues working with a graph-based pangenome model.

1.5.1 Graph Coordinate System

Linear genomes denotes genomic locations by specifying a contig name and indices, such as chr1:10Mb, referring to the base position 10 Mb from the beginning of chromosome 1. Genomic features, such as genes or exons, are represented using indices of one or more genomic intervals. This coordinate framework is the foundation of many bioinformatics pipelines and has shaped the data formats used across the whole genomic and bioinformatics filed [14, 15, 16]. The graph-based coordinates in a pangenome model are hard to incorporate into existing tools or software. To achieve the compatibility to available tools, the graph-coordinates should be easily interpreted and transformed to the conventional linear coordinates, which will smooth the transition from linear genomes to pangenome models. One of the major challenges for graph-based reference pangenomes is that the graph structure introduces complication referring to base positions or representing genomic intervals. Since there can be multiple sequence combinations within a single interval, referencing genomic features by a single pair of start and end coordinates, such as chr1:10-12Mb, can be ambiguous in a graph-based reference genome. Both sequential and vertical coordinates are required for a graph with alternative paths. A valid graph coordinate system should satisfy monotonicity, horizontal and vertical spatiality, backward-compatibility, and compatibility to linear coordinates [4, 5, 13].

- **Monotonicity**: *Monotonicity* of the graph-based coordinate system requires the coordinates of successive bases within a genome to be nondecreasing.
- **Spatiality**: Spatiality in graph-based coordinate system requires that the nearby bases in a graph should have similar coordinates. It includes *horizontal spatiality*, whether bases are close to each other along a single path, and *vertical spatiality*, where orthologous bases, *i.e*. identical position on parallel paths are close to each other.
- **Backward-compatibility**: *Backward-Compatibility* requires coordinates of the graphical genome should be stable after the genome update or topology modification. Genomic annotations should still be valid after the genome update or topology modification.
- **Linear coordinates compatibility**: Pervasive bioinformatic file formats and tools adopt linear coordinates for genomic features representation and analysis. To smooth the transition from linear coordinates to graph-based coordinate representation, we further proposed that the graph coordinate framework should be able to transform to linear coordinates conveniently.

1.5.2 Graph Indexing

Indexing the pangenome model is essential for the performance of the pangenome graph application such as read alignment, variant calling and genome browser. However, indexing pangenome graphs faces challenges. The common graph indexing methods include hash-based k-mer index, or FM index based on BWT data structure [5, 12]. The issue of graph indexing may introduce overhead, which can lead to significant computational memory and time costs [12]. For example, the genomic regions of many small variations will generate many sequence fragments. For the k-mer indexes, the complexity of graph traversal can exponentially increase with the character length k. Thus, an upper bound of k value should be set to limit the memory usage [65]. Another common structure is the BWT-based graph indexing. The Burrows-Wheeler Transform (BWT) data structure and the FM index (full-text index in minute space) have been widely used in sequence analysis [66]. The BWT includes a forward transformation, which permutes the input string text with a unique “end of text” symbol appended. One way is to index the assembly in the normal

BWT and FM index with increased alphabet size, representing the SNPs in the pangenome graph by using International Union of Pure and Applied Chemistry (IUPAC) code or "ref|alt" symbols [67, 68]. Increasing alphabetic size can achieve fast graph indexing but trade off querying speed when searching encounters variants and has to branch for each allele. Other tools index haplotypes in the VCF-based graphs such as PBWT(positional Burrows-Wheeler transform) [69], gPBWT(graph positional Burrows-Wheeler transform) [70], and GBWT(graph Burrows-Wheeler transform)[71].

1.5.3 Sequence Analysis Based on a Graphical Pangenome

Graph-based reference pangenome models preserve and document genetic variations, and provide a novel data structure to reveal genome landscape in the graph topology [4]. Applying a pangenome graph in the sequence analysis would reduce bias and improve the sequence analysis accuracy. Pangenome models also simplify the maintenance and management of genomic resources, benefit for visualizing genome structure and identify biological-interested genomic features in the population. Applying pangenome graph to the sequence analysis should no doubt bring new insights to the genomic field. However, sequence analysis based graphical pangenome models faces challenges. Different pangenome graphs are constructed tailored for different input dataset, and objective goals [72]. Many tools are designed for a specific type of genome graph, but are not generalizable to other graphs. With the emergence of new sequence technologies, such as the third generation sequencing techniques, these sequence analysis tools further diverge to handle different types of sequence data [12]. In general, the pangenomic field lacks standards and guidelines in both pangenome definition, construction, and applications. The lack of commonality between methods and tools make it hard to perform comprehensive surveys to evaluate and compare different methods.

Read alignment is usually the first step of typical bioinformatics and genetics pipelines. Recently many efforts has been made in the development of graph-based alignment algorithms, such as HISAT2 [73], GraphAligner [74]. However, it is general computational expensive and time consuming to align short-reads even to a single reference genome and this computational burden is amplified by aligning reads to pangenomes. In addition, there are differences in the types of graph that these tools can handle. A comprehensive comparison studies for the performance of these alignment-based tools is difficult to conduct due to these differences and the lack of gold standard. An alternative is to analyze sequence data based on the number of occurrences of a selected set of substrings of a fixed length of k in the raw data. Compared to alignment-based methods [75], the k-mer based methods are fast, flexible and more memory efficient.

In general, pangenome graph requires the development of novel tools, representations and data formats to facilitate the transition and application. It remains challenging how these genome graphs should fit into the existing linear-reference genome ecosystem and bring new insights to genomic field. The majority of the available bioinformatics pipelines are built on the linear reference framework. The community analyze data, design experiments and evaluate the results based on the linear reference framework and the standards. When and to what extent the pangenome model would be accepted by the community still remains questionable.

1.6 Contribution

In this thesis, I develop new methods for constructing a reference pangenome and applying a pangenome dataframe to analyze population-specific sequence data. I construct the first mouse reference pangenome model for a mouse genetic reference population, called the Collaborative Cross Graphical Genome (CCGG). **(1) I develop a genome registration method to construct reference pangenome model by integrating multiple genome assemblies and annotation files into a single framework.** Our method partitions multiple assemblies consistently by introducing anchor nodes, which are conserved, unique and consistently-ordered sequences with uniform size. **(2) We develop a graph-based coordinate system to reference, annotate and compare genomic features in a graph-based pangenome model.** The coordinate system normalizes linear coordinates to the unique occurrence position of the anchor sequences. It also benefits for assembly update, allowing backward-compatibility for feature annotation. **(3) We develop a k-mer query tool, an alignment-free method for analyzing population-scale sequence data based on a**

pangenome model. We select probes from the pangenome model to analyze the sequence similarity and diversity in the population. We count the frequency of these probes in the sequenced samples to achieve population-scale sequence analysis and genome comparison.

1.6.1 Thesis statement

"A reference pangenome model is a more compact, complete, accurate, and maintainable model for representing population-specific genome sequence data. By using an anchor-based series-parallel graph to represent a pangenome, it locally identify commonality and isolate differences between genomes of individual strains. This representation also allows for a graph-based coordinate system to refer to, annotate and compare genomic features. It supports the extraction of traditional linear genomes customized to each strain. Furthermore, an alignment-free method was developed to analyze sequence data based on this pangenome model."

1.6.2 Innovation

With the rapid growth in genomic sequencing capability and the reduction in sequencing costs, there is a emerging need to integrate multiple genomic assemblies into a single coherent representation. In my thesis, I constructed the first graph-based mouse reference pangenome model, Collaborative Cross Graphical Genome (CCGG), designed for the Collaborative Cross mouse genetic reference population. My thesis develops methods for constructing a specific graph-based reference pangenome model and discusses the principles for applying graphical pangenome models to analyze population-scale sequence data. The CCGG is derived from the 8 founder genome assemblies of the CC population. By overlaying the imputed haplotype intervals of each CC strain to the graph, the CCGG represents the genomes of 83 mouse strains in total. It incorporates multiple genomic annotations including genes, exons and repeated elements, which are overlaid on the graph. The CCGG provides a unified framework to simultaneously update and maintain annotations and assemblies.

The contribution of my thesis includes both conceptual and methodological aspects. Conceptually, anchor nodes are first proposed here, and are defined as topologically sorted k-mers that are common and unique to all genome assemblies incorporated in the graph. Anchor sequences provide a valid graph coordinate framework by normalizing the features coordinates relative to the unique occurrence of anchor sequences across all the represented genomes. In addition, anchor nodes partition multiple linear genomes into short fragments, which support a finer resolution for sequence comparison. The sequences between anchor pairs put the variants in a multiple sequence context and reveal the sequence diversity within each genomic region. The process of identifying potential anchors sheds light on the absent, duplicated or reversed sequences that provide insight into potential structural variants or assembly errors. The CCGG is stored in the standard FASTA file format, with annotations recorded as part of the header of each sequence fragment semantically encoded as a JSON string. The graphical genome simplifies the maintenance and management of genomic resources while supporting efficient assembly updates.

Methodologically, my thesis discusses a genome registration tool to construct an graph-based reference pangenome model from a set of genome assemblies. New assemblies can be incrementally added to the graph by a series of parallel compositions. Transcriptome or genomic regions of interests can be extracted and investigated by series compositions from the graph. The graph structure provides the benefits of compacting shared sequences, mapping counterpart sequences between multiple genomes, and visualizing sequence variations as a graph. The anchor-based coordinate system provides an efficient tool to manage assembly update and translate the gene or exon coordinates among assembly members of the pangenome model. An alignment-free method was developed to evaluate the population-scale sequence data based on a pangenome model. Overall, the pangenome graph framework proposed in this thesis serves as a platform for pangenome tool-chain development to better assess population-scale sequence diversity within a species.

1.6.3 Thesis Structure

This thesis is organized into 8 chapters that describe a graph-based reference pangenome framework and its usage.

- **Chapter 1. Introduction:** Provides the motivation for a pangenome reference model and an overview of this thesis dissertation.
- **Chapter 2. Previous Work:** Reviews the current status of the reference pangenome development and available tools.
- **Chapter 3. Genome Registration and An Anchor-based Pangenome Model:** Introduces a general methodology for constructing a graph-based reference pangenome model from an intra-specific population.
- **Chapter 4. Collaborative Cross Graphical Genome:** Describes the construction steps and the properties of a mouse reference pangenome model for the Collaborative Cross.
- **Chapter 5. Validation of the Graphical Genome:** Uses an alignment-free method to analyze and verify the CCGG reference pangenome.
- **Chapter 6. Graph-based Coordinate System:** Introduces a graph-based coordinate framework to reference, annotate and compare genome features in our graphical genome.
- **Chapter 7. Graph Refinements:** Discusses a process for incorporating novel strain-specific sequences to the graph and potential refinement strategies for improving the graphical genome.
- **Chapter 8. Discussion and Conclusion:** Summarizes the contributions and shortcomings of the proposed model and discusses future directions

CHAPTER 2: PREVIOUS WORK

Both tools and representations of pangenomes are under active development and have made tremendous progress in the recent years [76, 77, 78]. The research field of pangenomic analysis is young and many areas are still under exploration. A fundamental issue in computational pangenome field is the general lack of standards or pragmatic guidelines for constructing a pangenome graph and adapting them to common bioinformatics workflows.

Different types of pangenome graphs are designed for specific input and objectives, yet require specific methods to achieve the optimization goals [12]. The input for constructing a pangenome model includes one or more of the following 1) a reference assembly with variants information, such as a series of VCF files; 2) a set of sequences or assemblies; 3) a multiple sequence alignment of assemblies. Variation graphs are constructed by inserting well-documented sequence variants to a standard reference assembly [76, 77], yet in this frame large structural variants or sequence rearrangement are hard to express. Given a set of assemblies, methods have been developed to construct a compressed de Bruijn graph to represent these genomes [57, 79]. Sequence graphs can also be built upon a multiple sequence alignment results [61], which is computationally simpler but relies on the algorithms used to build the alignments [30]. In general, there are multiple ways for representing a set of genomes in a pangenome model. Given a fixed input set, the pangenome graph itself is not unique. There exists a diverse set of extant tools and algorithms for pangenome construction and analysis. It makes it hard to compare and benchmark different types of graphs and related approaches [30]. Users need to carefully select the methods depending on their input data type and the specific downstream needs. The following section discusses several available approaches for pangenome construction and analysis.

2.1 Pangenome Construction and Representation

2.1.1 Inserting Well-documented Variants to a Reference Assembly

Several graph-based pangenome construction pipelines have been proposed. A common way for graphical genome construction is to insert well-known variants to the linear reference assembly [80, 77, 76]. The resulting pangenome model is a directed graph, with bubbles representing variations. Consider Variation Graph (VG toolkit) as an example, VG toolkit takes VCF files and linear reference sequences as inputs to construct genome graphs [76]. The VG toolkit also incorporates external haplotype information into the sequence graph to facilitate the graph traversal. VG toolkit overlays paths to the graph and utilizes a designated path with offset to represent a genomic region. A graph node could occur in multiple paths and may loop back in a single path due to the cycles in the graph. The graph coordinate system provided by VG toolkit encodes paths with additional tags to place the stable linear reference coordinates to the graph. However, this tool is not backward-compatible to reference assembly update. Variants in VCF files have to be renewed to align to the new reference coordinates, and these variation graph should be reconstructed based on a new reference assembly. In addition, genomic complex region and large structure alterations are hard to implement in this framework. Reference-specific sequences, *i.e.* sequences private to a reference assembly, are especially difficult to represent.

2.1.2 Assembly-based Pangenome construction tools

Inserting well-documented variants from VCF files to a reference assembly is not sufficient to capture all the genomic features such as large structure variants or complex genomic regions. One potential solution is to construct *de novo* assemblies to preserve the haplotype information, and then analyze multiple assemblies altogether in a graph-based pangenome framework. Several approaches have been developed along this line. Panseq integrates a collection of genome sequences in a pangenome model, determines the core and accessory genome, extracts genomic regions unique to a genome or group of genomes and identifies SNPs within the core pangenome [81]. EUPAN was developed as a toolkit to analyze eukaryotic pangenomes (it has been benchmarked using rice genomes) by constructing pangenome model from de novo assemblies and using a "map-to-pan" strategy to detect within-species gene presence/absence variations (PAVs) at a relatively low sequencing depth [56]. HUPAN extends this approach to human genomes, taking 185 deep sequencing and 90 assembled Han Chinese genomes as input to find novel genomic sequences and novel protein-coding genes missing from the human reference genome (GRCh38)[82]. These assembly-based graphs vary in graph types and optimization goals. The graph representation depends on the construction algorithms. For example, the multi-sequence alignment results may vary with different algorithms and parameter settings. Thus, the same set of input assemblies could result in different graphs.

2.1.3 De Bruijn Graphs

Several algorithms have been proposed to construct compacted de Bruijn graphs directly from multiple assemblies, such as SplitMeM [57] and TwoPaco[79]. SplitMeM uses a suffix tree with suffix skip edges to construct compressed de Bruijn graphs [57]. TwoPaCo further provides a simple and scalable low memory representations and algorithms to support a pangenome graph construction for larger genomes [79]. Pantools introduces a generalized De Bruijn graph as a pan-genome representation, which is stored in a Neo4j graph database and is scalable to large eukaryotic genomes [83]. De Bruijn graphs with the addition of colors (a unique tag for each individual sequenced sample) to their nodes or unitigs (unipath nodes) enable the mapping of specific samples to the graph. Cortex used a colored de Bruijn graph to perform population-scale sequence analysis [84]. Other similar algorithms, such as Succinct colored de Bruijn graphs[85] and Bifrost [86] reduce the time and memory for constructing the colored de bruijn graphs.

2.1.4 Alignment-based Methods

Alignment-based pangenome models are another common approach used for pangenome construction[4]. Co-linear alignments are able to capture SNPs and small indels, which do not influence the sequence organizations [72]. However, larger structural changes involving large genomic alterations are hard to represent in alignment-based models, such as inversions, translocations or duplications. Non-colinear alignment methods segment the genome based on predicted homology and reveal local colinearities [72]. Progressive Cactus is a reference-free aligner of large vertebrate genomes with high alignment quality, which enables reference-free pangenome construction [87]. Seq-seq-pan builds a pangenome data structure by sequentially aligning a set of whole-genome sequences, allowing addition or removal of individual whole-genome sequences from a set of aligned sequences[88]. NovoGraph constructs a genome-wide multiple sequence alignment of de novo assembled contigs and outputs resulting graphs in VCF format [89]. Minigraph proposes a graph-based model and associated formats to represent a pangenome that incorporates multiple assemblies [90]. It adopts a "genome-to-graph" strategy to align chromosome-long sequences to a graph and constructs the pangenome model incrementally with high computational efficiency.

2.2 Pangenome Analysis

2.2.1 BWT and FM Index

The Burrows-Wheeler Transform (BWT) is a data structure that has been widely used in genome assembly and sequence analysis [66]. The BWT is commonly used to compress genome assemblies for pattern searching. When aligning reads to the reference assembly, algorithms first search seed k-mers from sequenced reads to place reads to a genomic location and perform small alignments to the reference assembly [91, 92]. The BWT data structure includes a forward transformation which permutes the input string text T with a unique "end of text" symbol (typically $\$$) appended. A suffix array (i.e. a lexicographically sorted array of all cyclical suffixes of the input string) is implicit in the BWT structure [93]. All the occurrences of the substring S in T are present within an interval of the suffix array. An auxiliary data structure of BWT, known as the FM-index, accelerates the process of accessing search sequence patterns and finding their occurrence interval in the BWT's implicit suffix array. BWT and FM index support exact searches for any kmers in O(k) time [94, 95].

The search is perform in reversed order of the kmer, leveraging a special property of BWT known as last-first (LF) mapping, where in the ith occurrence of a character, e.g. “A”, in the BWT string (the last column of the sorted cyclical suffixes) corresponds to the same character as the ith occurrence of “A” in the first column. Given a suffix of S, the search algorithm calculates the range of indices within the suffix array that starts with the suffix of S in query. Each symbol added to the suffix shrinks the range of the searching interval in the suffix array. By iteratively adding one character to the suffix of S, the searching range shrinks with steps. The length of the final index range indicates the occurrence time of the kmer appears in text. If the length of searching interval shrinks to 0, it indicates that string S does not appear at text T. Knowing the index range where S appears in the suffix array, one can retrieve the occurrence position of S in the original text.

A Multi-String BWT (msBWT) combines multiple texts into a single index and supports querying collections of strings simultaneously [96]. The msBWT has been applied to compress multiple sequences, such as contigs from multiple assemblies or short reads from a sequenced sample [97]. Many algorithms have been developed to construct an msBWT data structure [98, 99]. Similarly, the FM-index can be constructed for an msBWT, allowing access to its implicit suffix array and enabling the search of an arbitrary k-mer in O(k) time. Here, we use msBWTs and their FM-indices to index and compress raw short-read sequence data. These indices are used to perform fast bulk k-mer queries directly in the short-read sequence dataset.

2.2.2 Graph Indexing and Graph Aligner

Read mapping is fundamental for genomic analysis and has been widely applied in many sequence analysis pipelines [12]. However, mapping read to a genome graph genome with alternative sequences are not trivial. The performance of read mapping depends on the graph indexing of the sequence graph [12, 5]. Hash-based k-mer indices and FM-index based on the BWT structure are frequently used indexing nucleotide sequences [12]. For example, GenomeMapper [65] uses k-mer index for graph indexing. In highly polymorphic region, the number of alternative k-mers grows with its length k. Thus an upper bound of kmer length is set to limit the memory usage[12, 65]. The

BWT data structure and its FM-index are also widely used for graph indexing, such as in GCSA [100], BWBBLE [67], v-BWT [68]. BWBBLE and v-BWT index genome graph with increased alphabet size [67, 68]. Pan-genome Seed Index (PSI) [101] proposed a hybrid method combining an graph index of selected paths with an index of the queried reads. In addition, a series of efforts have focused on indexing haplotypes in Variation Graphs including PBWT [69], gPBWT [70], GBWT [71]. These tools provide efficient representations for compressing haplotypes embedded in sequence graphs and enable substring queries.

Many efforts have been focused on developing tools for aligning reads directly to a sequence graph. These graph aligners includes HISAT2 [73], VG [76], GraphAligner [74] *etc*. HISAT2 (hierarchical indexing for spliced alignment of transcripts 2) align sequence reads using a BWT-based data structure and FM index [73]. The HISAT adopts a hierarchical indexing strategy with a global index representing the whole reference genome and thousands of small, overlapping local indexes. The global index and the local indexes cover the whole genome and all alternative sequences together [73]. Rather than using a global index, this allows searching sequence pattern in a local region by using a small local index to speed up the query. In addition, HISAT2 indexes repeats separately. Instead of aligning reads to the real occurrence position of these repetitive sequences in the reference genome, HISAT2 align reads directly to an instance of a repeated elements. This decreases multiple alignments and reduces the computational burden required to resolve repetitive sequences. The VG toolkit provides a general framework both constructing and aligning reads to a sequence graph [76]. VG uses BWT-based index methods and longest-common-prefix array to achieve highly specific queries [12]. VG toolkit is capable to align both short read and long read to the graph, but the runtime for long read alignment tends to be slow [74]. GraphAligner is tuned to align long reads to a sequence graph [74]. Besides variation graphs, GraphAigner also aligns to de Bruijn graphs and provides a pipeline to perform error correction of long reads [74]. Other graph aligners, including GenomeMapper [65], V-MAP [102], have been developed to deal with different types of sequence graphs. However, the sequence graphs are of different types and these methods focus on different optimization objectives, which introduces confusion when deciding which tool should be used. A standard and guideline for pangenome construction and analysis should be provided for the community to build up a sound ecosystem for pangenome analysis.

2.2.3 Genotyping in Pangenome Graph

Genotyping methods based on a reference pangenome have been developed, which are mainly derived from graph alignments of sequenced reads to a pangenome graph [77, 80, 73, 103, 104]. The strategies include aligning reads to the graph, calling and filtering variants and iteratively implementing common variants to the graph [77]. Pangenome models provide an enriched and unbiased scaffold for variant calling and improves the performance in highly divergent genomic regions or regions with large structural variants [12]. Graphtyper aligns reads to a sequence graph and enables fast and highly scalable and accurate genotyping in population-scale datasets [80]. VG toolkit provides a SV genotyping tool by mapping reads to a whole-genome graph and inferring SVs based on path coverage ratio [104].

K-mer based genotyping tools are an alternative based on graphical genomes. Colored de Bruijn graphs have been proposed to overlay short-read sequenced samples onto a graph [19, 84]. These enable genotyping variants by using k-mer counts from the short-read sequence samples directly. BayesTyper uses the mapping of read k-mers to a reference graph to perform unbiased, probabilistic genotyping across the variation spectrum [105]. Dolle et al. proposed a reference-free compressed data structure, a population BWT, and used it to store and index the sequenced reads from the 1000 Genomes Project samples in a fashion similar to our query msBWT data structure [106].

2.2.4 Graphical Genome Browser and Visualization

Genome browsers are a essential tool for genomic research. They are used for searching, visualizing, and interpreting genomic data and alignment results. The commonly used genome browsers for standard linear reference genome include IGV (Integrative Genomics Viewer) [107], UCSC genome browser [108], *etc*. Recent progress has also been made to visualize multiple linear assemblies in a single web interface, such as MGV (Multiplge Genome Viewer) in mouse genome

[109]. Genome graphs are powerful tools for displaying the sequence similarity and differences between multiple genomes. However, the complex graph structures, such as a compressed de Bruijn graph, is hard to display in a planar space. There are challenges for pangenome graph visualization tool development. Extant visualization methods has been developed to display specific kinds of genome graphs. For example, Bandage was developed to display de novo assembly graphs, the de Bruijn graphs, and perform and visualize BLAST searches in the graph [110]. GfaViz was developed for the visualization of sequence graphs in GFA format [111]. Sequence Tube Map were designed for visualization of variation graphs constructed by VG [112]. MoMI-G (Modular Multi-scale Integrated Genome Graph Browser) were developed to visualize structural variants based on genome graphs and support visualizing alignments, coverage and annotations [113]. MoMI-G were designed to facilitate long read analysis and SV visualization [113].

2.3 Data Formats for Pangenome Representation

Pangenome representations require the development of novel data formats to represent the graph structure and to support pangenome analysis. A pangenome can be simply represented as a collection of sequences in standard FASTA files. Pangenome models can also be represented in forms of multi-sequence alignment or a k-mer based model, such as the de Bruijn graph. Another representation of pangenome models is a directed graph composed of nodes and edges. Alternative sequences in a pangenome model contains both SNPs, small indels or structural variants [5]. Graph-based pangenomes compress shared sequences in the input assemblies but still represent of the whole input sequences [12]. The input sequences can be extracted and reconstructed by traversing through a sequence graph. For variation graphs, the pangenome models can be represented as a set of FASTA files with VCF files representing the variations. To represent general sequence graphs, the Graphical Fragment Assembly (GFA) format has been developed and has been widely used for sequence graph representation, such as sequence assembly graphs [114]. GFA format generalizes the PAF (Pairwise Alignment Format) to describe sequence graphs. It is a tab-delimited text format for describing a set of sequences and their overlapping. However, the GFA format lacks a coordinate system for the entire genome [30]. Later, a new method, rGFA, extends GFA format to reference pangenome graphs [90]. It provides a stable coordinate system by placing tags to specify the position of a graph entity in the linear reference genomes incorporated in the genome graph [90]. The limitation of rGFA representation is that it only considers paths incorporating simple variants relative to the reference assembly, such as SNPs or small indels [30]. A Graphical Mapping Format (GAF) is derived from the Pairwise Alignment Format (PAF) to describe graph alignment results [90]. In addition, the VG tool kit has developed the Graph Alignment Map (GAM) format, which is a TAB delimited format to describe sequence-to-graph alignments [76]. GAM format is generalized from the SAM (Sequence Alignment Map) or BAM (Binary Alignment Map) [14] format to describe pangenome graphs. To annotate genomic features or regions in a sequence graph, a new file format, gGFF, extends the GFF format for pangenome graph annotation [115]. Furthermore, in order to compare the similarity and differences among pangenome graphs, PGVF (Pangenome Graph Variation Format) was developed to describe the graph-to-graph alignment. PGVF represents a collection of aligned graphs by merging sequence graphs. PGVF applies to general graphs and is a work in progress.

In this thesis, we propose a series-parallel graph, where graphs can be added or decomposed by a sequence of series and parralel operations. It simplifies the merging and comparison of genome graphs.

CHAPTER 3: GENOME REGISTRATION AND AN ANCHOR-BASED PANGENOME

3.1 Background

Reference genomes serve as the foundation of modern bioinformatics analysis [3]. They are commonly represented as one or more single-sequence assembled contigs [4]. Sequence reads collected from individuals are mapped to a standard linear reference genome for downstream analysis, such as abundance estimation or variant calling. Genomic features, such as genes or exons, are annotated as one or more genomic intervals on the reference assemblies. Variants are reported relative to these linear reference genomes. The use of linear reference genome has shaped method developments and data formats across the whole genomics and bioinformatics fields [5]. However, linear reference assemblies come with limitations [4, 2, 17, 3]. Reliance on a linear reference genome will introduce reference bias to the downstream analysis, *i.e.* the sequences divergent from the linear reference genome tend to be discarded or misinterpreted [12]. Thus, reference bias has strong impact on genomic regions with structural variants or sequences that are absent from the reference genome [12].

In addition, linear genomes fail to capture the genetic variations in populations. Sequence variants are conventionally stored separately in VCF files [16] to support the downstream analysis. Complex genomic regions or regions with large structural variants are hard to represent in VCF files. Moreover, the linear reference assemblies are being updated and improved regularly. These updates of the reference assemblies imply coordinate changes that require a large effort to map to the new standard. Genomic discoveries identified based on the old reference genome need to be transferred and re-evaluated in the new standard. Inexpensive and fast genome sequencing has enabled the construction multiple genome assemblies that capture more genetic diversity than any single reference assembly. However, it still remains challenging to represent and analyze multiple assemblies in standard bioinformatics or genetic analysis pipelines.

Most available variant discovery pipelines are based on genotyping or short-read sequencing technologies [116]. These methods rely on the standard linear reference genomes to identify and report SNPs, indels and structure variants [116]. However, when the genetic background of the sequenced sample is different from the reference strain, relating the sequence reads to the linear reference genome will generate many false positive calls [3]. In addition, structural variations fundamentally impact the genome organizations and are essential for many genomic and functional studies [117]. Yet identifying SVs is confronted with challenges introduced by the current sequence analysis scheme [116]. Prevalent SVs detection tools align pair-end short-reads to a reference genome to infer the presence of SVs [116, 118]. The mapping of paired-end reads are often used to assess the size and orientation of a SV and the read-coverage are used to detect deletions or duplications [116, 118]. However, due to the large variability of SVs in both size and types, it is technically difficult to detect and reveal the full range of SVs [116, 118]. In addition, the visualization of SVs is challenging in the linear genome scheme, as SVs usually introduce non-linear correspondence between sequence segments [30]. Graphs, on the contrary, provide a potential solution for representing non-linear mappings between paralogous sequences and visualizing SVs. The graphical data structure provides a more efficient tool than the linear genomes to describe duplication, translocations or inversions [12]. Cycles or directed edges in a genome graph can be applied to represent duplications, large deletions, inversions, or translocations by associating edges to correspondence between linearly far apart regions or inverted regions.

Reference pangenome models, which incorporate multiple genomes in a single representation, are a promising alternative [4, 5, 12]. Graphs are commonly used in pangenome representations, as they provide a natural way to merge shared sequences and represent alternative sequences [4]. Many efforts have been dedicated to constructing graph-based reference pangenome models, from microbial genomes [48] to large eukaryotic pangenomes such as human pangenomes [55]. The common pangenome construction methods include inserting well-documented variants to a linear reference genome, or merging multiple linear assemblies into a compact sequence graph. In this chapter, I introduce a novel approach to construct a graph-based pangenome model for a population with a known ancestry. Our pangenome framework is a directed, series-parallel sequence graph. Series-parallel graphs are a specific type of planar graph that can be constructed by a sequence of series and parallel composition. Many hard combinatorial problems in general graphs can be resolved in linear time if the input graph is series-parallel [64]. Assisted by the sequence data in the represented population, our method constructs a reference pangenome model from a set of ancestor linear assemblies. For populations derived from a known set of ancestors genomes can be represented as a unique recombinant path switching between the ancestor genomes in the graph.

Our graph-based pangenome framework establishes common 'anchor' sequences that are shared amongst all assemblies as well as every sequenced sample incorporated in the pangenome graph. Anchor sequences register and partition linear genome assemblies into short fragments and establish a topologically sorted ordering of these disjoint genomic regions. Sequences between anchor pairs are merged into parallel paths, which provide finer resolution for genome comparison. Annotations are overlaid onto the graph, and are preserved along with sequences and transferred between paths in the sequence graph. The graphical genome maintains and manages multiple sequence and annotations consistently, thus providing an enriched reference resource for unbiased sequence analysis. It also provides a computationally efficient and backward-compatible data structure that benefits for assembly update, genome comparison and visualization.

The additive feature of series-parallel graphs makes our graphical genome a flexible framework for different objectives. Large genomic contigs can be decomposed into a series of intervals between adjacent anchor pairs. One can focus on a specific functional regions, extracting the corresponding subgraphs to perform analysis for targeted genes or isoforms. In addition, linear genomes of new samples can be added to the graph by overlaying parallel sequences to the graph. This provides an efficient framework for genome assembly updates and annotation updates. Our approach identifies and segments the updated sequences, representing them as new edges to the graph. The update of the reference assemblies in the pangenome framework is of minimal impact on other genomic regions, thus most of the genomic annotations will not be changed during the assembly update. By tracking the mapping positions of anchor candidate sequences in a genome assembly, our method reveals the distinct sequence organizations between the reference and the alternative assemblies. Discordant mapping of candidate kmers provides insights for potential assembly errors or virtual structural variants (SVs). We benchmarked our methods by constructing a graph-based pangenome model for a mouse genetic reference population, Collaborative Cross. Guided by 69 sequenced samples of the CC population [119, 120], we merged the genome assemblies of the 8 CC founder strains into a single graphical genome. Each chromosome is represented as a series-parallel graph. In this Chapter, I describe the methodology used in the CCGG construction. The properties and statistics of the CCGG are described in Chapter 4.

3.2 Methods

In this section, we describe the workflow for anchor-based pangenome construction. The input of the CCGG was the genome assembly of the 8 CC founder strains, including GRCm38 and the other 7 assemblies released in [6]. Short read sequence data from 69 CC samples were used to select anchor candidates [119, 120]. The CCs are a panel of recombinant-inbred mouse descended from the 8 founder inbred mouse strains [34]. The genome of each CC strains can be regarded as a unique mosaic of the 8 founder genomes. The CC haplotype intervals are inferred from the CC sequenced samples using an HMM [119] and are overlaid to the founder path of the CCGG. The CC linear genomes are represented as a path or paths on the graph and have been further refined by incorporating the CC Most-Recent-Common-Ancestor sequence data (described in Chapter 7). The workflow of our method is shown in Figure 3.1.

3.2.1 Anchor Candidate Selection

The first step of constructing an anchor-based pangenome model is to identify a set of conserved sequences in the represented population to use as *anchor candidates*. Here, we define anchor candidates as the unique k-mers (in both forward and reverse strands) from a given genome deemed to be the reference. The choice of an anchor candidate set is not unique and involves certain heuristics. In the construction of CCGG, we considered every non-overlapping 45-mers from the GRCm38 reference assembly. The choice of k-mer length is influenced by the contig length and their expected divergence within the genomes included in the pangenome model [121]. The k-mer length is not unique. To avoid the situation where a k-mer and its reverse complement are identical, we prefer an odd value of k. The sequence content could vary if we choose to select candidate k-mer from GRCm39. For each of these non-overlapping 45-mers, we query its occurrence in the GRCm38 assembly in both forward and reverse complement orientations and record in a repeat vector. We selected the unique 45-mers in the GRCm38 reference assembly. Next, we test if these 45-mers are conserved in the CC population. In the previous work, we had queried the frequency of every non-overlapping 45-mer of the GRCm38 in the 69 raw sequence data of the CC population released in [119, 120]. We next filtered these k-mer candidates by testing whether these k-mers are present in every sample from the represented CC population. There are heuristics to select conserved k-mers considering different sequence depth of each sample. Here we decide to select k-mers that are supported by more than 3 reads in every CC sequenced sample. These unique and conserved k-mers are deemed as *anchor candidates*. These anchor candidate sequences represent the core genomes of the CC population.

3.2.2 Genome Registration and Anchor Selection

We next map these anchor candidate sequences to the other 7 founder assemblies released from [6] and merge the input assemblies into a series-parallel graph. We call this phase *genome registration*. We scan through every k-mer of each genome assembly and determine whether itself or its reverse complement are in the anchor-candidate set. We constructed a registration table to record the mapping position of each anchor candidate in each genome assembly. The registration table serves as a substrate for merging and comparing multiple linear genomes. We then exclude anchor candidates that 1) are absent in one or more founder genomes and 2) appear more than once in any genome (irregardless of orientation). We next examine the ordering of the remaining candidate sequences. The 45-mers that are consistently ordered in every linear assembly are selected as anchor nodes. Many anchors can be adjacent, we collapse such a series by removing all but the first and the last of a contiguous run. These anchors are merged into a single edge that are shared by every path in the graph.

Anchor candidate sequences were removed during the genome registration because they were absent, duplicated, or inconsistently ordered in any of the linear assembly. These removed anchor candidates can be further classified into 5 subsets: 1) The candidate sequences are absent in the registered linear genome, implying genomic gaps, sequence errors or mutations in the linear assembly. 2) The candidate sequences that appear more than one time in the linear assembly. These duplicated k-mers could appear either in forward or reverse complement, mapping to any contig. This suggests a potential duplication or redundant sequences in the linear genome. 3) The candidate sequences appears only in the reverse complement. The inverted sequences might occur on the same contig as the reference or on different contigs. It suggests either a potential inversion in the genome, or an assembly orientation issue appear in the registered genome. 4) The candidate sequences appear in different contigs other than the occurrence in the reference assembly, suggesting potential mis-assembled errors or inter-chromosome translocations in the registered genome. 5) The candidate sequences map to the same chromosome but are out of order relative to other anchors. This suggests potential assembly errors or intra-chromosome translocation events.

We found the removed anchor candidate sequences tended to be clustered in the founder genome assemblies. To validate the sequence similarity between the mapping regions, we extracted sequences between the first and the last 45-mer of a cluster and performed the pairwise alignments between the reference and the alternative sequences. If the sequence similarity (percentage of matching bases) is higher than 80%, we considered these intervals for further analysis. Recall that the initial anchor candidate set is composed of k-mers that are unique in the reference genome and are supported by more than 3 reads in every 69 sequenced CC sample. Thus, using this method to identify inversions or translocations or tandem duplications is biased towards finding large examples.

Also, large de novo deletions in a single sample impact anchor selection and may result in larger genomic intervals (large gaps between anchors) that are not indicative of shared structural variation.

3.2.3 Edge Creation

Once the anchor nodes are identified, the sequences between adjacent anchors are extracted and merged into edges. Between a pair of anchor nodes, edges with identical sequence content are merged and their assembly IDs are recorded in the *strain* attribute of the edge. Edges represent segregating haplotype sequences between a source and a destination anchor node. There is a list of edge-specific annotations, including "src", "dst", "strains", "variants", and "inversion". The "src" and "dst" attributes specify the proximal and distal nodes connected to an edge. The "strains" attribute annotates genomes that share the edge, including a founder genome, or a CC genome that are constructed via recombination of the founder genomes. Sequence similarity and diversity among genomes are reflected by the collection of strains sharing the same edges. The pair-wise alignment was performed between each non-reference edge and the reference path betweeen the same anchor nodes. The alignments are stored in the "variants" attribute of each edge in forms of a SAM-tools compatible cigar string ("=" for matches, "X" for mismatches, "I" for insertions and "D" for deletions). These cigar strings represent an alignment to the reference, which can be used to identify sequence variations. Cigar strings representing functional annotations, including genes and exons, are also included on both the anchor nodes and edges on the reference path.

The graph traversal in an anchor-based graphical genome is straight forward. Anchor nodes are shared by every path in the graph. There are 8 or fewer parallel paths between each pair of anchors in the current version of the CCGG. The graph topology is implicit in the "src" to "dst" attributes of edges. We traverse the graph from "src" to "dst" of an edge, except when an "inversion" flag is reached and the traversal direction is flipped (details in Chapter 6). We iterate through the node-edge and identify the paths according to a strain ID. The 8 founder assemblies as well as the CC genomes can be reconstructed by traversing the path between SOURCE to SINK. We also develop an algorithm to construct all the paths between a given pair of anchors. This algorithm keeps

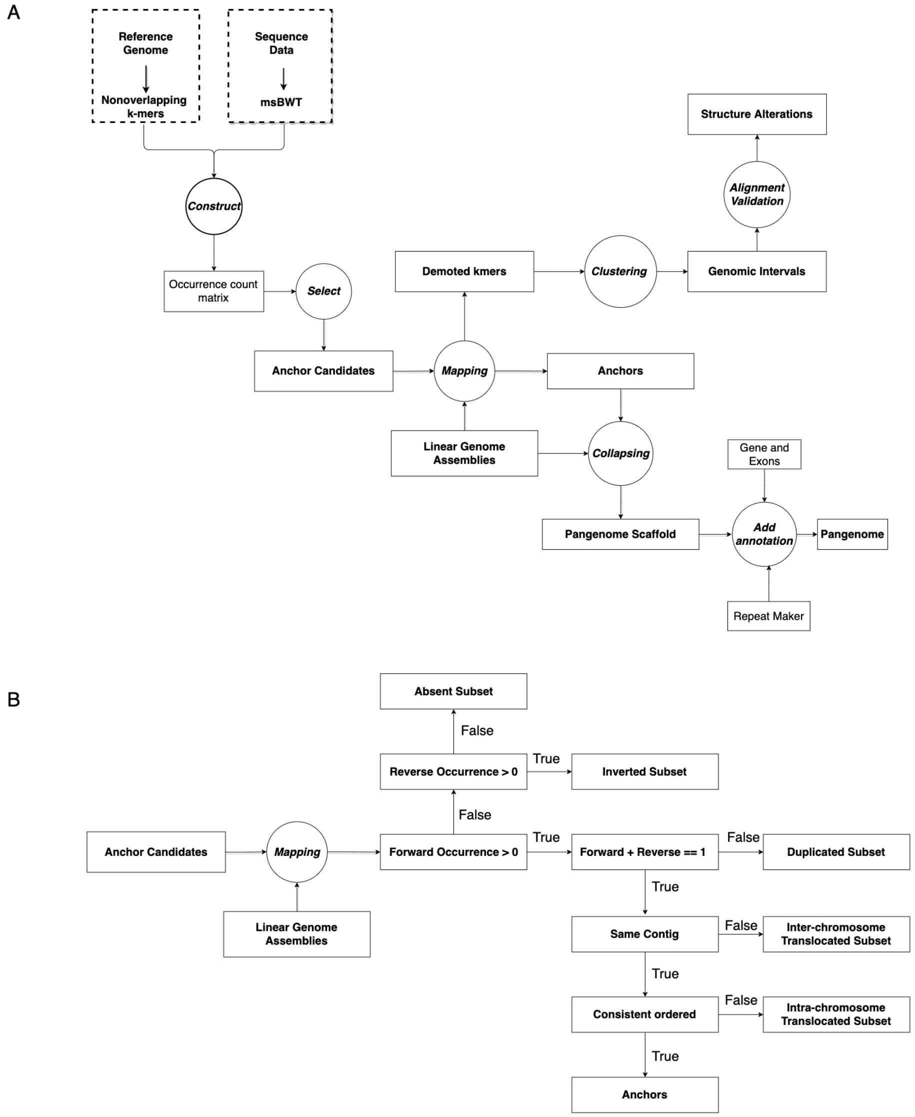


Figure 3.1: Workflow for Genome Registration. A) An occurrence-count matrix for non-overlapping k-mers of the linear reference assembly was constructed for all 69 sequenced CC samples. Candidate anchors were selected as they are unique in the reference assembly (both forward and reverse complement) and were supported by more than 3 reads in every sequenced sample. Each of these anchor candidates were tested in every other 7 CC founder genome assemblies. Those candidate sequences that were unique and topologically sorted in all linear assemblies were selected as anchor nodes. Sequences between anchors were extracted and merged as paths in the pangenome. Genomic features, such as the gene and exon annotations were then added to the reference path of the graphical genome. B) The demoted k-mer were filtered sequentially and classified into 5 subsets, including “Absent”, “Duplicated”, “Inverted”, “Intra-chromosome translocated” and “Inter-chromosome translocated” subset.

track of the shared founder strains along the path (memoization), and if the set of shared strains becomes empty, we terminate the search. For a large genomic region covering multiple anchor pairs, we construct all the parallel paths between each adjacent pair of anchors and sequentially concatenate the paths within each interval. Overall, the series-parallel graph structure reduces the complexity of graph traversal.

In many cases, all initial input sequences are merged into a single common edge, which we call a collapsed edge. Collapsed edges are shared by the 8 CC founder assemblies. Unlike anchors, collapsed edges, while common to all input sequences, may not be unique or may not be consistently ordered in every linear assembly in the graph. We classified collapsed edges into 3 categories. If any k-mers composed the collapsed edge are not supported by more than 3 reads in every CC sample, it's type attribute is set to "missing". Otherwise, if any k-mers are not unique in the genome, their type is set to "with repeats". And, if all the k-mers in the collapsed edge are conserved and unique, their type is set to"conserved". The kmers on the "conserved" collapsed edges can be promoted to anchor nodes if needed. The distribution of anchors and collapsed edges are shown in Figure 3.2, where the percentage of base pairs shared by 8 founders are shown in every 100kb bin.

As described in Chapter 4, anchors are densely distributed along autosomes in the mouse genome. However, there are genomic regions where the spacing between a pair of anchors is long. These long intervals impact the resolution of sequence comparison, coordinate mapping and sequence compression. To provide a better understanding of the sequence organizations over these *long gaps*, we allow floating nodes between adjacent pair of anchors where any path exceeds 1,000 bp. We partitioned the sequences in long gaps into non-overlapping k-mers. In the case where sequence lengths are not multiples of k, we allow for overlaps in the middle of the path. We identify shared k-mers among parallel edges and merged the shared sequences by inserting floating nodes at the beginning and the end of these kmers. The floating nodes themselves don't carry any sequence content or annotation. Floating nodes merely provide additional sequence compression. Floating nodes also create new parallel sequences between a subset of paths, which further benefit for sequence comparison and coordinate mapping.

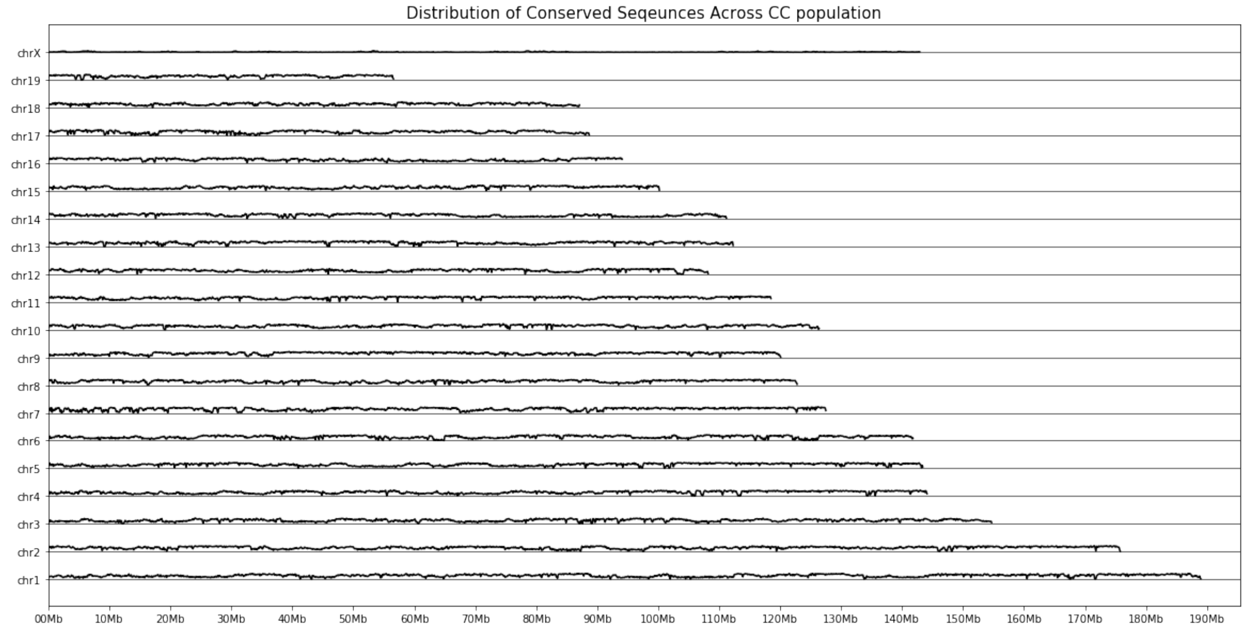


Figure 3.2: Distribution of sequences shared by 8 founder genomes. Both anchors and collapsed edges between a pair of adjacent anchors represent conserved sequences shared by the 8 founder assemblies. In this figure, we plot the distribution of the base pairs shared by 8 founder genome in every 100kb bin. The Y-axis represent the percentage of conserved sequences normalized by the total length of each 100kb bin.

3.2.4 Graph Annotation

We annotate anchor nodes and edges with genes, exons, and interspersed genomic repeat types [122]. We also annotated edges with a "variants" attribute to record the pair-wise alignment of the edge to its parallel reference sequence. All the graph annotations are stored in JSON format and as part of the sequence's FASTA header (Figure 3.5). The Gene and Exon intervals are provided for the GRCm38 reference genomes are obtained from Ensembl Biomart [123]. We overlay the gene interval to the anchor nodes and edges by using a SAMtools compatible cigar string, where "M" represents sequences overlapping with the feature and "S" represents sequences that are not overlapping [14]. Similarly, we annotate the exons on reference path of the CCGG. We also annotate the genomic repeats and their types in the graph using their Repeat-Masker annotations obtained from http://www.repeatmasker.org/. We record the class of the repeat, the name of the matching interspersed repeat, the sequence orientation and the cigar string describing the relative position between the repeat interval and the graph entity.

Variants on edges that are less than 1000 bp are annotated as follows. For each genomic region separated by a pair of anchors, we align each alternative path to its parallel reference path and find an alignment with minimum Levenshtein edit distance. A SAMtools compatible cigar string [14] is constructed for the alignment and is recorded as a ”variant” attribute on the edge. As a result, every edge along the reference path should contain a ’variant’ string of the form ’N=’ where N represents the length of the reference edge’s sequence. Other paths will contain mismatches (’X’), insertions (’I’), and deletions (’D’) sufficient to transform the sequence on the alternate path to the reference sequence. Alignments for longer paths were generally less informative, confounded by structural variants (SVs) that are enriched in those regions, such as large duplications, deletions, or inversions. We insert floating nodes within those long gaps to assist with sequence comparison and compression (details in Edge Creation section). Floating nodes create local parallel sequences, where we further perform pairwise alignment in those regions.

3.2.5 Adding New Linear Genomes to the Graph

The CCGG provides an efficient framework for adding additional reference assemblies and robustly updating old builds and annotations with backward-compatibility. The series-parallel graph structure of our anchor-based graphical genome allows new sequences to be incorporated by using only a limited set of operations [64]. The process of adding new assemblies to the graphical genome is described as follows. We first map the original set of anchor candidate sequences selected from the represented population to the given linear assembly. We record the mapping position of candidate sequences, selecting a new set of anchors that are conserved, unique and consistently ordered in the registered genome. We partition the large contigs into small segments based on the anchor nodes. Overlaying the fragments of the new assembly onto the graph, the identical sequences are merged to the existing graph entities with new path annotations added to their attributes. The unique sequences are inserted to the graph as new parallel paths between anchor nodes. Threading the linear genome to the graph not only compresses identical sequences, but also segments and identifies strain specific haplotypes by introducing new edges to the graph.

Here we propose a new data format for preserving those registered linear genomes relative to our graphical genome, as well as a fast pipeline to adding these linear genomes to the graphical genome. We provide meta files to describe these registered linear genomes by using an anchor information table recording all the segments sharing with the graph and a FASTA file of all the sequences with unique features in this assembly. The anchor information table records the mapping position of the anchor sequences in the new assembly and the outgoing edge from the anchor node. If the anchor sequence does not appear in the new assembly, the position column will be an empty string. If the sequence segments of the new assembly are merged to the existing edge in this graph, the edge name and the CC founder strains on this edge will record in the anchor information table. On the contrary, if the subsequence of the new assembly is different from any path between the pair of anchors, it will be recorded as a new edge in the FASTA file with *src* and *dst* annotations and sequence contents. When adding the linear genome to the graph, the assembly ID are appended to the strain attributes of the existing edges and the FASTA file of the new edges are added to the existing graph edge files.

During loading the registered genome to the graph, we tentatively allow edges to bypass anchors, as a result these anchors no longer satisfy the *conserved*, *unique* and *topologically sorted* properties in the graph. After the loading the new assembly to the graph, these anchors can be *demoted* by coneverting them to edges and creating two new floating nodes on both sides (Figure 3.3). The incoming edges of the demoted anchor are now assigned to the proximal floating node and the outgoing edges from the demoted anchor are assigned to the distal floating node. These new edges will inherit all of the annotations of the anchor except for the coordinates of each founder assemblies. A "strains" annotation will be added to the new edges, which is a union of strains from the edges on either side of the anchor prior to the added assembly.

Our pangenome framework provides a relatively stable framework for maintaining genomic features in their sequence context as well as for managing multiple versions of an assembly. Maintaining the version of the genome's assembly along with the sequence content simplifies the lift over of annotations. When a genome assembly is threaded through the graph, the annotations

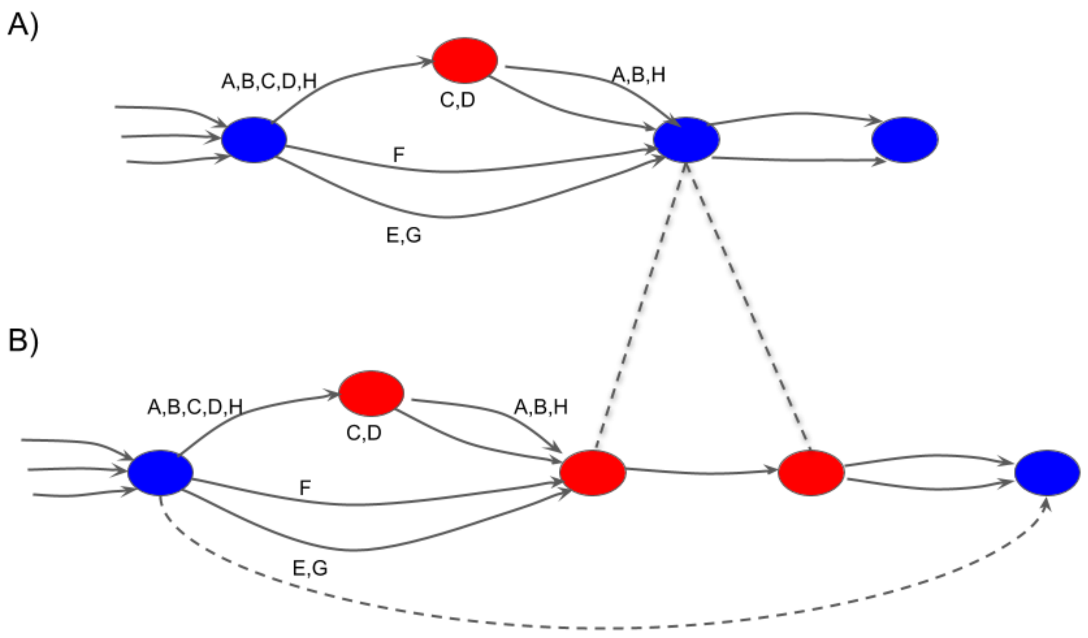


Figure 3.3: Demoting an Anchor: In the figure, blue nodes represent anchor nodes and red nodes represent floating nodes. In A), the anchor node in the middle is no longer conserved, or unique or consistently ordered in the graph and therefore needs to be demoted. A newly inserted edge by pass this demoted anchor is shown by the dashed line in B). To complete the demotion, the anchor node is replaced with an edge of 45 bp which is connected to two new floating nodes. The incoming edges of the old anchor are now given to the new floating node on the left and the outgoing edges from the old anchor are given to the floating node on the right. Floating nodes do not have sequence content but serve as connectors between edges.

are adjusted only around modified regions. This will not impact most annotations on the graphical genome. Thus, the anchor-based graphical genome significantly enhances backward-compatibility relative to traditional linear genomes. More details of graph coordinate system and lift over tool are discussed in Chapter 6.

3.3 Results

3.3.1 Anchor-based Graphical Genome and Terminology

Our anchor-based graphical genome is a series-parallel sequence graph composed of nodes and edges. There are four types of nodes: anchors, sources, sinks, and floating nodes. Only anchor nodes have associated genomic sequences. Anchor nodes satisfy three properties: 1) anchor sequences are of uniform length and appear on every path in the graph; 2) anchor sequences are unique in both forward and reverse complement and thus establish an one-to-one association between genomes; 3) anchors are topologically sorted in all the paths of the graphical genome.

Each anchor node has a unique name. Our anchor naming convention begins with the letter "A" followed by a contig identifier and a decimal point that is followed by 8 additional digits indicating their linear coordinates on a reference assembly. We assign anchor names such that a lexicographical sorting maintains the sequential ordering of anchors. Anchor nodes are annotated with their coordinate in each linear genome. The SOURCE and SINK nodes represent the start and end of genomic contigs. The floating nodes allow for sharing among a subset of paths. The inclusion of floating nodes enables sequence comparison and compression between a subset of paths between adjacent anchors. Edges are directed and represent alternative sequences between an adjacent pair of anchor nodes. Edges are annotated according to the one or more genomes upon which they lie. Edges provide the connectivity of graph by specifying source (*src*) and destination (*dst*) nodes. Paths are sequences between any pair of nodes, which are represented as a series of graph entities. Parallel paths are defined as the set of paths between a given pair of nodes. Parallel paths represent distinct haplotypes of a genomic region. The genomic regions separated by an adjacent anchor pairs are referred to as *gaps*. In general, anchor nodes establish an ordered series of shared sequences separating genomic gaps, whereas parallel paths reveal the orthogonal structures.

3.3.2 Graphical Genome Data Format

The anchor-based pangenome model is represented as a graph data structure, which is serialized as one or more standard FASTA files (Figure 3.5). Genomic attributes and annotations for both nodes and edges are stored in a dictionary encoded as a JSON string in each sequence's FASTA header. Typical annotations for edges include src, dst, strain, gene, exon, repeat class, and variant. Anchor nodes are typically annotated with their known positions in each of the linear genome assemblies. We provide a Python-based API for accessing, traversing, and editing the anchor-based

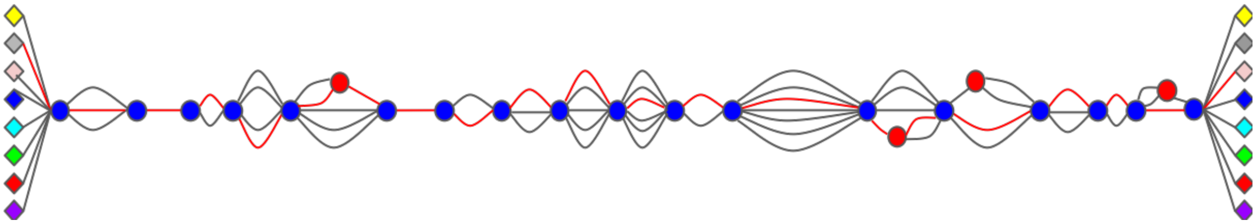
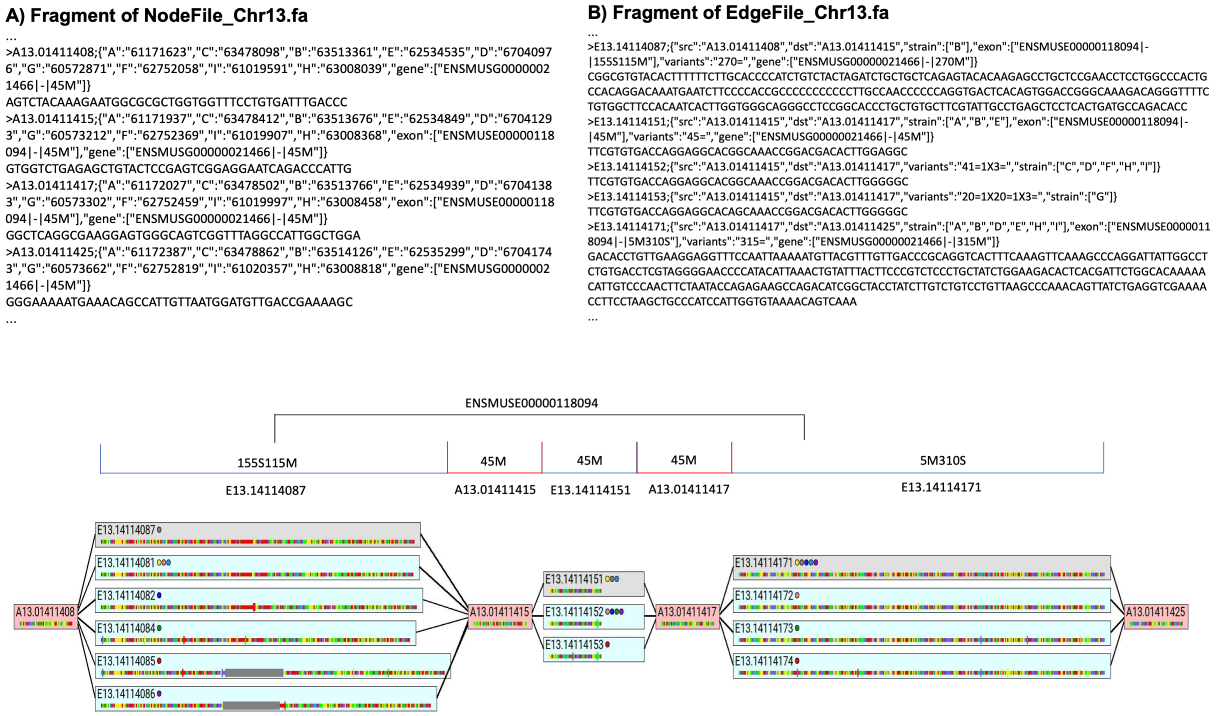


Figure 3.4: An example of an anchor-based graphical genome. In general, a graph is composed of nodes (or vertices) and edges. There are three types of nodes. Anchor nodes, shown in blue, contain unique and conserved sequences that are shared by every genome in the graph. SOURCE and SINK nodes, represented as diamonds, denote the start and the ends of specific genomic contigs in the graph as indicated by their color in this figure. SOURCEs and SINKs contain no sequence. Floating nodes, in red, support sequence mapping and compression between a subset of paths within the long gaps. Like SOURCE and SINK nodes, floating nodes contain no sequence. Edges represent the vast majority of sequence data, and always lie between a pair of adjacent nodes. Paths refer to the accumulated sequences between any pair of nodes. The red line in this figure represents a recombinant path beginning from the gray SOURCE and ending at the pink SINK.

graphical genome, as well as a command-line interface (CLI) for extracting various paths, subpaths, and other symbolic genomic representations from the graph. The API parses the CCGG FASTA files and builds a graph representation from them. Components of the graph are four dictionaries: *nodes*, *edges*, *incoming*, and *outgoing*. The *nodes* and *edges* dictionaries are indexed by names of anchor nodes and edges and return a dictionary of attributes. An attribute of each edge is the names of its source and destination nodes. These required key values are used to dynamically construct the incoming and outgoing dictionaries during parsing. Another required key is 'strain', which returns a list of paths/genomes that share the edge sequence. The incoming and outgoing dictionaries represent the graph's topology. Both incoming and outgoing are indexed by node name (either anchor, floating, source, or sink) and they return a list of adjacent edge names. The graph is traversed in the forward direction by iterating through the outgoing edges of a node and finding the edges lying along the desired path. When a matching edge is found, its destination node is used to traverse the graph by accessing its entry in the outgoing dictionary. The process iterates until either the SINK node or a specified destination node is reached. The graph can be traversed similarly in the reverse direction using the incoming dictionary.



A) Fragment of NodeFile_Chr13.fa

```
...
>A13.01411408;{"A":"61171623","C":"63478098","B":"63513361","E":"62534535","D":"6704097
6","G":"60572871","F":"62752058","I":"61019591","H":"63008039","gene":["ENSMUSG0000002
1466|-|45M"]}
AGTCTACAAAGAATGGCGCGCTGGTGGTTTCCTGTGATTTGACCC
>A13.01411415;{"A":"61171937","C":"63478412","B":"63513676","E":"62534849","D":"6704129
3","G":"60573212","F":"62752369","I":"61019907","H":"63008368","exon":["ENSMUSE00000118
094|-|45M"],"gene":["ENSMUSG00000021466|-|45M"]}
GTGGTCTGAGAGCTGTACTCCGAGTCGGAGGAATCAGACCCATTG
>A13.01411417;{"A":"61172027","C":"63478502","B":"63513766","E":"62534939","D":"6704138
3","G":"60573302","F":"62752459","I":"61019997","H":"63008458","exon":["ENSMUSE00000118
094|-|45M"],"gene":["ENSMUSG00000021466|-|45M"]}
GGCTCAGGCGAAGGAGTGGGCAGTCGGTTTAGGCCATTGGCTGGA
>A13.01411425;{"A":"61172387","C":"63478862","B":"63514126","E":"62535299","D":"6704174
3","G":"60573662","F":"62752819","I":"61020357","H":"63008818","gene":["ENSMUSG0000002
1466|-|45M"]}
GGGAAAAATGAAACAGCCATTGTTAATGGATGTTGACCGAAAAGC
...
```

B) Fragment of EdgeFile_Chr13.fa

```
...
>E13.14114087;{"src":"A13.01411408","dst":"A13.01411415","strain":["B"],"exon":["ENSMUSE00000118094|-
|155S115M"],"variants":"270=","gene":["ENSMUSG00000021466|-|270M"]}
CGGCGTGTACACTTTTTTCTTGCACCCCATCTGTCTACTAGATCTGCTGCTCAGAGTACACAAGAGCCTGCTCCGAACCTCCTGGCCCACTG
CCACAGGACAAATGAATCTTCCCCACCGCCCCCCCCCCCTTGCCAACCCCCCAGGTGACTCACAGTGGACCGGGCAAAGACAGGGTTTTC
TGTGGCTTCCACAATCACTTGGTGGGCAGGGCCTCCGGCACCCTGCTGTGCTTCGTATTGCCTGAGCTCCTCACTGATGCCAGACACC
>E13.14114151;{"src":"A13.01411415","dst":"A13.01411417","strain":["A","B","E"],"exon":["ENSMUSE00000118094|-
|45M"],"variants":"45=","gene":["ENSMUSG00000021466|-|45M"]}
TTCGTGTGACCAGGAGGCACGGCAAACCGGACGACACTTGGAGGC
>E13.14114152;{"src":"A13.01411415","dst":"A13.01411417","variants":"41=1X3=","strain":["C","D","F","H","I"]}
TTCGTGTGACCAGGAGGCACGGCAAACCGGACGACACTTGGGGGC
>E13.14114153;{"src":"A13.01411415","dst":"A13.01411417","variants":"20=1X20=1X3=","strain":["G"]}
TTCGTGTGACCAGGAGGCACAGCAAACCGGACGACACTTGGGGGC
>E13.14114171;{"src":"A13.01411417","dst":"A13.01411425","strain":["A","B","D","E","H","I"],"exon":["ENSMUSE0000011
8094|-|5M310S"],"variants":"315=","gene":["ENSMUSG00000021466|-|315M"]}
GACACCTGTTGAAGGAGGTTTCCAATTAAAAATGTTACGTTTGTTGACCCGCAGGTCACTTTCAAAGTTCAAAGCCCAGGATTATTGGCCT
CTGTGACCTCGTAGGGGAACCCCATACATTAAACTGTATTTACTTCCCGTCTCCCTGCTATCTGGAAGACACTCACGATTCTGGCACAAAAA
CATTGTCCCAACTTCTAATACCAGAGAAGCCAGACATCGGCTACCTATCTTGTCTGTCCTGTTAAGCCCAAACAGTTATCTGAGGTCGAAAA
CCTTCCTAAGCTGCCCATCCATTGGTGTAAAACAGTCAAA
...
```

Figure 3.5: Graphical Genome Representation. Both the nodes and edge files of the graphical genome are stored in the standard FASTA files. All the annotations are recorded in the header of each sequence fragment in JSON format. We provide tools for visualizing the graph topology within certain region. As shown in this picture, boxes enclose both anchor nodes and edges, with their sequences shown as a colored barcode below the node's or edge's name. Sequence variations are incorporated in edges and are denoted by long bars according to the "variants" attribute of each edge. Pink boxes represent anchors nodes. Light gray boxes indicate the standard reference path in the graph while light blue indicates alternate paths. The strain attributes of each edge are represented by dots with standard colors of founder strains next to the edge name. Sequences of each graph entity are represented by the color bars within each box, where sequence variations are denoted by long bars according to the pairwise alignment. The single-path intervals on the linear reference assembly, including the genes, exons and repeat maskers, are annotated to the graph entities on the reference path overlapping with these features. The relative position between the genomic feature and the graph entity is denoted by SAMtool cigar strings. As shown above, the first 115 bases of exon ENSMUSE00000118094 overlap with the edge E13.14114087, represented as "155S115M". The exon is encoded on the negative strand ("-"). The information is annotated in the "exon" attribute of this edge as [”ENSMUSE00000118094| − |155S115M”].

3.3.3 Computation Time for Adding Linear Assembly to the Graph

The series-parallel graphs provide an efficient framework for adding new intra-specific assemblies and updating the reference genomes when new reference assemblies are released. The anchor sequences divide the large genomic contigs into small segments and they can be merged to the existing graph entities or be inserted as new edges to the graph. We benchmarked our method by overlaying the new mouse reference genome GRCm39 onto the CCGG. We registered the GRCm39 assembly by using the same set of anchor candidates (recall these were originally selected based on the previous mouse reference, GRCm38). The sequence segments of GRCm39 are overlaid to the CCGG, where identical sequences are merged to the graph directly by appending the assembly ID to the "strain" attribute of the corresponding edges. Updated sequences on GRCm39 are inserted as new edges to the graph. Only 1,197 new edges were introduced to the CCGG pangenome, comprising only 2.4% of GRCm39's total length, whereas unaffected sequences were merged with the extant edges of the graphical genome. The threading process also flagged updated regions, thus highlighting the changes between builds. Also, 113 anchor sequences of the CCGG were eliminated during the GRCm39 update, as 14 of them were absent, and 99 were duplicated in the new build. The removal of anchors during the assembly update did not have a significant impact on the genomic annotations. This update was a fairly computationally trivial process. The registration phase of GRCm39 took 86 minutes for 20 chromosomes (263 Mb, autosomes + X chromosome), 5.18 minutes for chromosome 1 (195 Mb) and 4.34 minutes per chromosome on average. The merging phase and graph topology modification took 36 minutes, 1.8 minutes per chromosome on average (graph loading time excluded). The annotation phase, including gene and exon annotation and pairwise alignment between parallel paths, took 140.89 minutes, 7 min per chromosome on average. All the analysis was performed on a desktop computer (Intel Xeon CPU E5-2620, 2.00GHz, physical memory 32,860,064 kB) at the Department of Computer Science, UNC-Chapel Hill.

In addition, the BXD is a set of recombinant-inbred strains derived from crosses between C57BL/6J (B6) and DBA/2J (D2) inbred mouse strains [124]. The C57BL/6J strain is a CC founder, whose genome has already been included in the CCGG. We overlaid the DBA/2J linear genome assemblies from [6] to the CCGG. We keep the these registered linear genomes in forms of an anchor information file that records the assembly sequences that are merged to an exisitng graph edge, and a FASTA file recording all the new edges that the assembly introduced to the graph. When appending these registered linear genomes to the graph, we can loop through the anchor information file and append the assembly ID to the strain attributes of the existing edges sharing with the assembly. For unique sequences in the graph, we then append the new edge FASTA files to the extant graph edge FASTA files. Merging the DBA/2J sequences onto the CCGG introduced 2,216,394 private DBA/2J edges. The remainder already existed in the graph, and only required adding a path ID to the strain attribute. This process is straightforward, but anchors that violated the *unique*, *conserved* and *topologically sorted* properties are not eliminated during this loading phase. Further graph modification of graph topology are required and the details as described in the Method Section.

The graphical genome provides an overview of the sequence sharing information between DBA/2J and the other 8 CC founders. We compared the DBA/2J sequences with the CC founder genomes in terms of edges and base pairs respectively. As shown in Figure 3.6, 87.88% (16064356 in 18280750 DBA/2J edges) of DBA/2J edges are identical with the GRCm38 reference sequences, which comprised of 50.81% (1,310,280,525 in 2,578,793,109 bp) of the DBA/2J genome length. By further analyzing the alignment results between the reference and DBA/2J path, we found that 87.14% of the DBA/2J sequences are matching with the reference sequence. The top 15 reference and non-reference components in terms of edge number and base pair number of DBA/2J genome are provided in Figure 3.6. Among the non-reference sequences, the pattern "ACDEHI" is the second most common, after the DBA/2J, "I", private sequences. The pattern "ABCDEHI" pattern is the second most common edge label for the DBA/2J edges, and it is the third in terms of base pair numbers. This suggests that the DBA/2J genome is primarily from the domesticus mouse subspecies.

3.3.4 Detection of Structural Variants or Assembly Errors

Since anchor candidates are unique and sequentially ordered in the reference genome, the process of registering anchor candidates to a new linear assembly, provides insights related to the sequence organization of the newly registered assembly. The mapping position of demoted candidates helps identifying genomic regions that are discordant with the graph's sequence organization. The

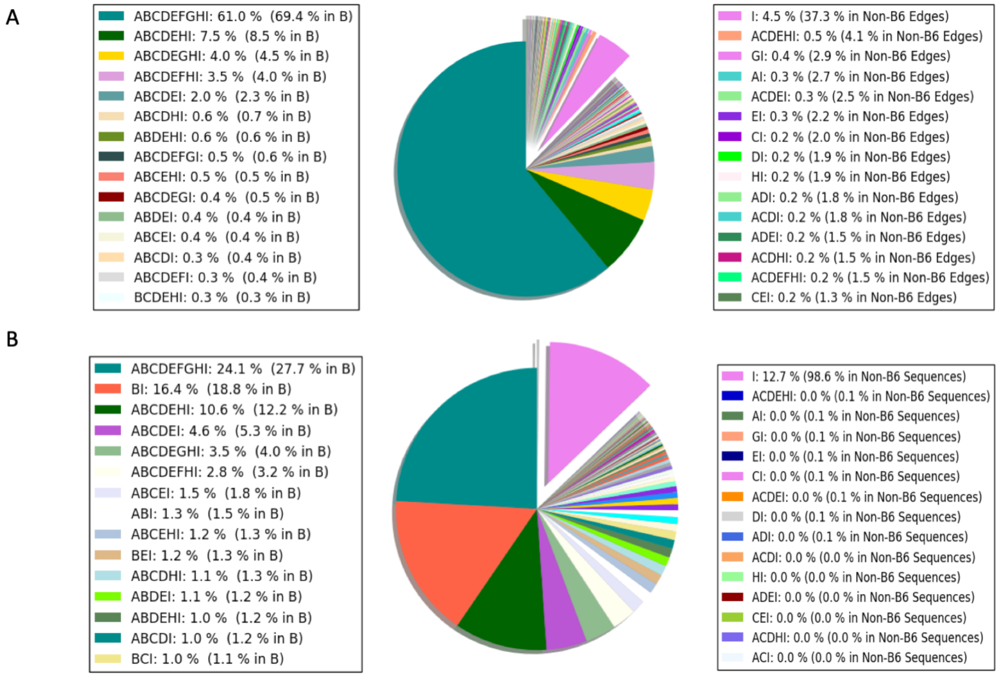
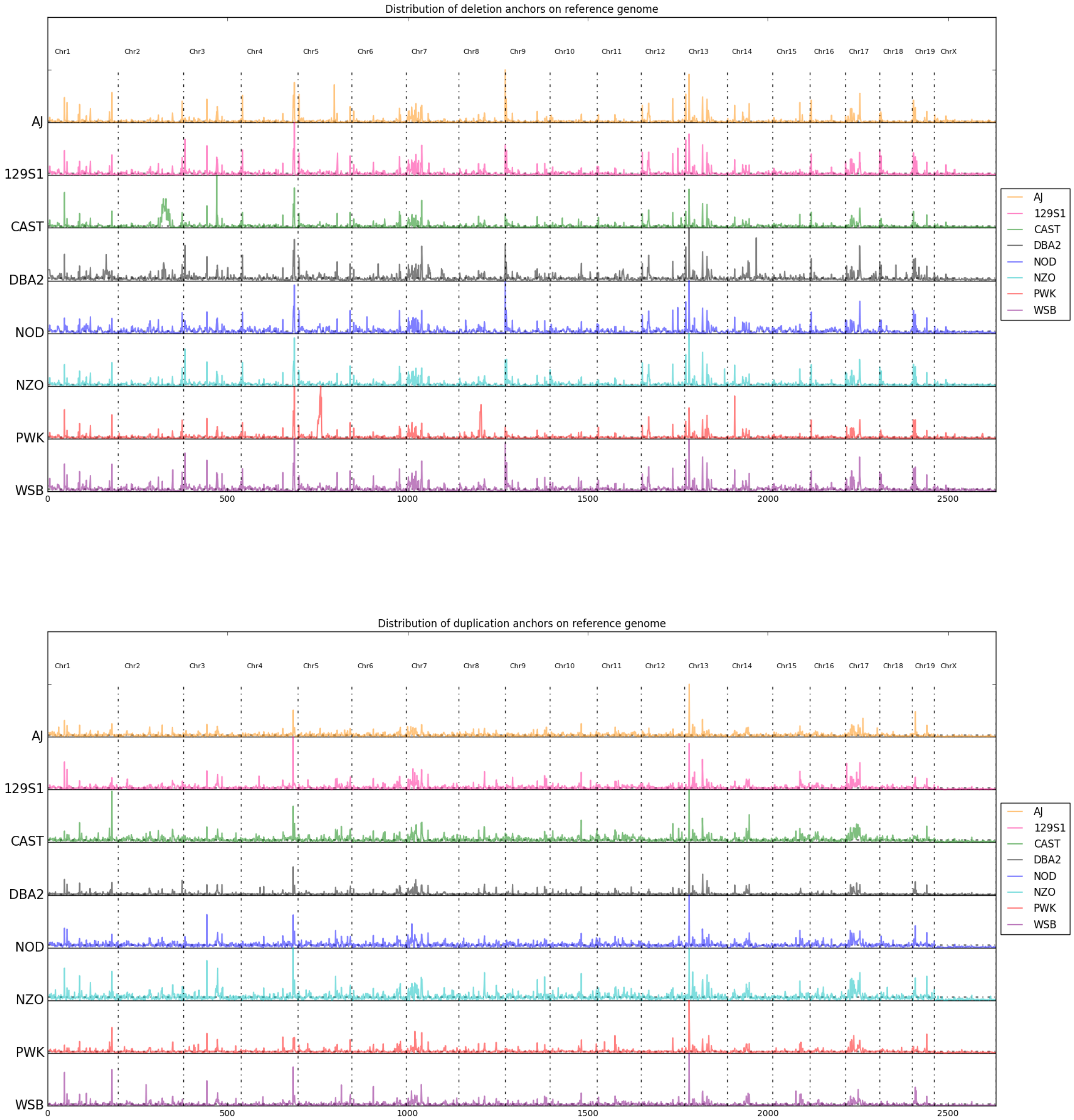


Figure 3.6: Sequence Similarity and Diversity between DBA/2J and CC Founders. A) A pie chart of shared edges between DBA2 and all 8 CC founders. The pie chart is plotted edges shared by CC founders and DBA/2J genome. It contains 18,280,750 DBA/2J edges in total, splitted into the reference sequences and non-reference sequences. The top 15 components in each category are shown in this figure. B) A pie chart of shared base pairs between DBA2 and all 8 CC founders. The pie chart shows the Strain Diversity Pattern (SDP) of 2,578,793,109 base pairs in the DBA/2J genome. The number of reference base pairs and the non-reference sequences are plotted in this figure. The non-reference sequences include mismatches (SNPs, “X”) and insertions (“I”) in the pairwise alignment between reference path and the DBA2 path. 87.14% of the DBA2 genomes are identical with the reference genome. The number of matching sequences in a non-reference path are recorded as the union of reference and alternative strain attributes.

mapping patterns of anchor candidates were next sequentially filtered and classified into subgroups (Table 3.1, more details are provided in the Methods section).

We found that demoted anchor candidate sequences usually cluster into genomic regions. These clusters are typically correlated structural variants in a linear genome, which vary in type and size, or potentially poorly assembled genomic regions. In Figure3.7, we plot the distribution of removed anchor candidates in every CC founder assembly except for C57BL/6J. In addition to the missing anchor sequences, there are 4 other categories of the demoted candidates including the *Duplicated*, *Inverted*, *Inter-chromosome translocated* and *Intra-chromosome translocated* sequences. The mapping of demoted sequences in the these subsets provide insights about the difference in


Distribution of deletion anchors on reference genome

Distribution of duplication anchors on reference genome

Figure 3.7: Distribution of Absent or Duplicated Anchor Candidates. This plot shows the distribution of demoted candidates that are absent or duplicated in at least one of the alternative genome. X-axis shows the reference genome position per 1Mb bin. Y-axis plots the normalized number of the deleted anchors in each 1 Mb bin, i.e. the raw number of deleted kmers divided by the max number of the demoted anchors within every 1Mb bin.

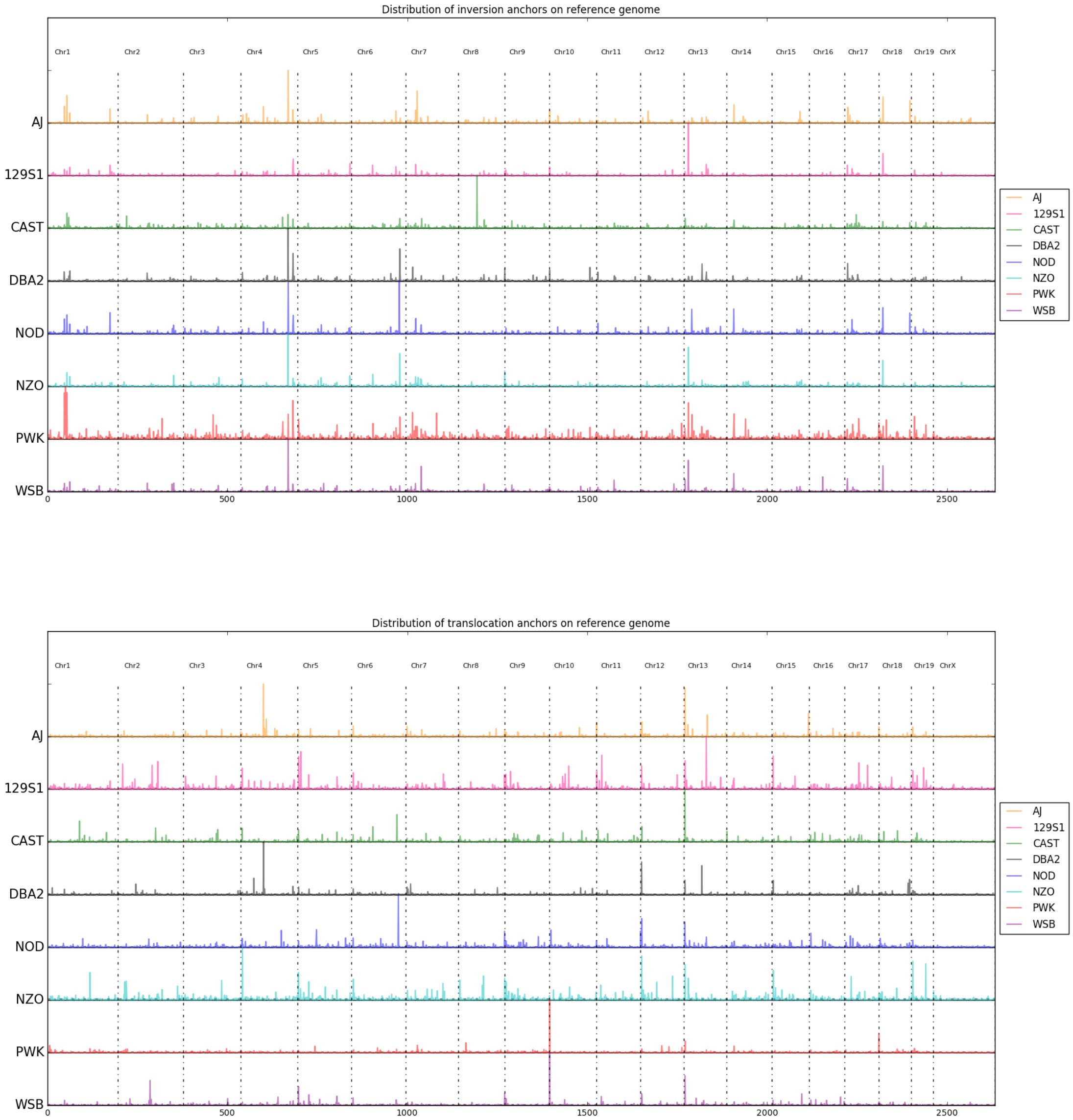


Figure 3.8: Distribution of Inverted or Translocated Anchor Candidates. The distribution of duplicated or translocated candidate sequences in at least one of the alternative genome. X-axis shows the reference genome per 1Mb bin. Y-axis plots the number of the deleted anchors in each 1 Mb bin, normalized between 0 and 1.

Table 3.1: Anchor Candidate Statistics. The total number of anchor candidates with discordant mapping in each linear genome is recorded in the “Total” column. “Absent”, “Duplicated”, “Inverted”, “Inter-chromosome translocated” or “Intra-chromosome translocated” breakdown the total according the mapping of the anchor candidate on each assembly.

Strain	Total	Absent	Duplicated	Inverted	Inter-chromo	Intra-chromo
AJ	287,013	175,027	70,416	12,596	4,537	24,437
129S1	377,734	252,158	88,497	11,924	5,357	19,798
CAST	493,466	309,672	81,946	56,619	11,220	34,009
NOD	479,384	282,309	108,812	15,284	6,087	66,892
NZO	354,391	227,680	92,399	11,890	5,526	16,896
PWK	422,817	238,712	54,179	76,654	12,108	41,164
WSB	419,048	329,197	55,756	11,206	4,980	17,909
DBA2	388,006	307,653	49,480	11,308	4,673	14,892
GRCm39	77,622	207	2,889	74,526	0	0

sequence organization between the reference and alternative assemblies. In Chapter 6, we overlaid the genome assembly of another inbred strain, DBA/2J from [6], which wass not a CC founder, onto the CCGG. We plotted the distribution of candidate sequence which map to a different chromosomes in the DBA/2J genome. A few clusters are identified in this map, which indicate translocations or potential assembly errors in either the DBA2/J or reference genome assemblies.

We selected all the clustered demoted anchor candidates on a continuous run in the reference assembly and tested if these sequences also consistently map to every non-B6 founder genome. Note that for the inverted and duplicated candidate sequences, their mapping order could be reversed relative to the reference. Among all these intervals, we further selected the intervals that are larger than 1kb and recorded their start and end coordinates in both the reference and the alternative genome. We extracted sequences in each interval and performed the pairwise alignment between the reference and the alternative sequence. If the percentage of matching are over 80%, we recorded these intervals for further SV analysis. For example, we found a cluster of duplicated sequences at the distal end of chromosome 4, spanning from 145.5Mb-147.8Mb on chromosome 4, GRCm38. These duplicated sequences are shared by 8 founder linear assemblies including A/J, 129S1/SvImJ, NOD/ShiLtJ, NZO/H1LtJ, CAST/EiJ, PWK/PhJ, WSB/EiJ and DBA2/J in the mouse pangenome model. This duplication was further validated using a double normalization tool developed in

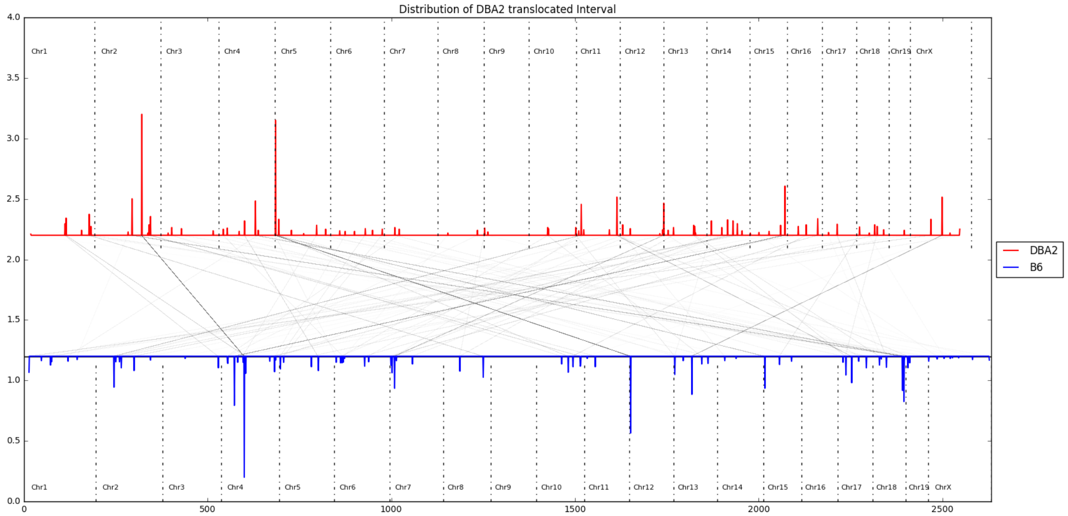


Figure 3.9: Mapping of inter-chromosome translocated anchor candidates between the mouse reference genome GRCm38 the DBA2/J genome. The anchor candidate sequences that appear on a different contig of the DBA2/J genome are indicated in the figure above. The density of these inter-chromosome translocated sequences and their mapping is shown in this figure. The gray lines show the sequence association between the GRCm38 reference genome and the DBA/2J genome.

our lab, where the raw count of each non-overlapping kmer on the GRCm38 mouse reference genome are normalized by the occurrence time of the kmer on the reference assembly and the mode of the kmer count in each sequenced sample. This double normalization tool maps the kmer copy numbers consistent with the reference assembly to near 1, thus indicating variations from the reference rather than an absolute copy number. Fold changes of the double-normalized kmer counts generally indicate a duplication event in the CC sequenced sample. We observe a fold change of kmer counts in this region in every founder samples, suggesting a duplicated events are prevalent in the CC founder strains. Another duplicated region appears on Chromosome 13: 12.5-13.5Mb on the GRCm38 reference assembly are observed and validated in founder strains. This analysis suggests that the genome registration methods offer benefits for identifying shared large structural variants among mouse strains.

3.4 Discussion

In this chapter I proposed a general methodology for linear genome registration based on anchor sequences, which are defined as conserved, unique, and topologically sorted sequences with uniform size. The graphical genome constructed via genome registration provides an unified framework to robustly maintain and manage multiple genomic resources, leveraging the existing standards for genome representations. The anchor nodes and edges of the graph are stored in a series of standard FASTA files, where all the annotations are stored in the headers of each sequence fragment in JSON format. The anchor-based pangenome is neither minimal nor unique. The main object of CCGG is to make best use of existing genomic resources and to provide useful functionality for the community conducting research using the CC genetics reference population. CC strain-specific genomes and annotations can be extracted from the CCGG in the standard file formats. The anchor-based pangenome framework shows both computational and biological advantages, especially suitable for the analysis of large eukaryotic genomes. It establishes a comprehensive picture for genome organization, facilitating searching, comparing, and visualizing the genomic structure in the CC population.

The natural products of the anchor-based graphical genome include 1) a graph-based coordinate framework based on anchor sequences; 2) the detection of assembly errors or sequence variations based on demoted anchor candidates. The selected anchor sequences in the graphical genome establish a ordering amongst disjoint genomic regions and provide a context-based coordinate framework that is more robust than offset-based models. When new reference assembly versions are introduced, our anchor-based pangenome framework supports the merging of identical sequences to the existing graph entities, while placing updated sequences in a specific genomic region separated by anchor pairs. Thus, only the anchor nodes need to be remapped when a new assembly is updated, and a specific edge's sequence should be compared to the new build, and potentially updated. In addition, the demoted anchor candidates identified during genome registration indicate discordant sequence organizations relative to the standard reference assembly, which implies either assembly errors or structural variants in the new genome being incorporated. It should be noted that the selection of anchor candidates involves heuristics. The first choice is k-mer size, which can be selected based on either biological knowledge (i.e. primer selection) or by characterizing the content of the given set of genomes [121]. In addition, as we selected anchor candidate sequences from every non-overlapping k-mer from an arbitrary choice of genome designated as the reference, such as the GRCm38 assembly in CCGG (Chapter 4). One could and would choose a different set of anchor candidates if starting from an another genome (ex. GRCm39). The graphical genomes resulting from different anchor choices would have a slightly different topology, but the overall pattern of sequence similarity and diversity reflected from the graphical genome should be similar.

The graphical genome can be extended to incorporate more linear genomes or sequence data from the same species. It establishes a platform for comparing large eukaryotic genomes. Furthermore, the graphical genome provides a scaffold for characterizing and refining individual genomes in the represented population. Leveraging the sequence data of individual samples, k-mer profiling based on graphical genome can be utilized to characterize genomic features at population-scale. In the following chapters, we explore the general features and genomic insights provided by CCGG.

CHAPTER 4: COLLABORATIVE CROSS GRAPHICAL GENOME

4.1 Introduction

The mouse reference genome is one of the most widely used, well annotated and accurately assembled mammalian genomes. It is the foundation for a wide range of bioinformatics and genetics tools. However, it represents the genome of a single inbred mouse strain, C57BL/6J (B6) [11]. There are many common genomic features reported in other mouse strains that are not included in the standard mouse reference genome [6]. Currently, most of the available variants discovery pipelines rely on alignments to the standard mouse reference genome to identify SNPs, indels and structure variants. However, relying on the mouse reference genome for variant discovery is blind to many *Mus musculus* genomic features where B6 is an outlier. As a result, alignments to it will generate both false positive and false negative variant reports due to the different genetic background. Recently, inexpensive and fast sequencing technologies has enabled the assembly of other common mouse strains at a high quality. However, integrating multiple assemblies in standard genomics analysis pipelines presents its own set of challenges.

Recombinant-inbred panels are derived from a known set of founder strains and the genome of each individual line is a unique mosaic of its founder genomes. As many recombinant-inbred panels of model organisms are developed, the standard reference assembly becomes insufficient as the choice of reference. The Collaborative Cross (CC) is a popular recombinant-inbred population. The Collaborative Cross is a widely used mouse genetic reference population, derived from 8 inbred mouse strains, including A/J, C57BL/6J, 129S1/SvImJ, NOD/ShiLtJ, NZO/H1LtJ, CAST/EiJ, PWK/PhJ and WSB/EiJ [34]. These founder strains are commonly abbreviated using the letters A through H. The CC recombinant-inbred panel provides stable and replicable genomes for genetic studies. Since CC strains are inbred, a given strain only requires genotyping once, and can produce many phenotype measurements for a set of individuals [125]. The CC strains capture about 90% of the common variants of laboratory mouse [36]. A large expansion of phenotypic variations has been observed for many traits measured using CC panel [37, 38, 39, 40, 41, 42]. The CC population thus has been widely used in complex traits and "systems genetics" studies. Characterization of CC genomes benefits for better analyzing and interpreting genetic experiments performed in the CC population. This has motivated the development of new genomic tools and resources to investigate the genomics and variations within the CC population [45, 44]. As most of the bioinformatics tools rely on the standard mouse reference genome as a substrate to perform analysis, sequence analysis in CC strains is complicated by biases introduced by having C57BL/6J, the basis of the mouse reference genome, as a founder [126, 127]. The actual genome of each CC strain is better modelled as a mosaic of its eight founder genomes. Some of these limitations can be improved by performing a joint analyses [45, 44, 128] using the new genome reference assemblies of the other CC founder strains [6]. However, these tools have limitations on both accuracy and computational efficiency. In the previous work, the short-read sequencing data for 69 CC mouse strains were released, each with 20x-30x coverage on avaerage and collected from a single male sample from each available CC line[119]. Later, sequencing data from sixadditional CC strains were released [120]. Meanwhile, standard "linear" reference genomes were released for sixteen common inbred inbred laboratory mouse strains, which included genomic assemblies from seven of the eight strains from which the CC was derived [6]. The remaining strain, C57BL6/J, was already the basis for the widely available *Mus Musculus* reference genome (GRCm38) [11]. There is a pressing need in the mouse genetics community to develop tools that leverage the available genomic resources to support both traditional and new genetics analysis of the CC population.

In this chapter, we present a pragmatic graph-based pangenome representation as a genomic resource for the widely-used recombinant-inbred mouse genetic reference population, called the Collaborative Cross (CC), and its eight founders. Our pangenome representation leverages existing standards for sequence representations with backward-compatible extensions that describe graph topology and genome-specific annotations along paths. It packs 83 mouse genomes (8 founders

+ 75 CC strains) into a single graph-based representation. The sequence content of CCGG is derived from the standard mouse reference genome (GRCm38) and the de novo assemblies of the 7 CC founder genomes [6] and the CC specific linear genomes are represented as mosaic paths of the 8 CC founder genomes. Each chromosome of CCGG is a series-parallel graph, composed of anchor nodes that are shared in every genome and edges between anchor nodes representing diverse haplotypes. The introduction of anchor nodes with sequence content divides large eukaryotic genomes consistently into short segments. Parallel edges between anchor pairs place variants within a common context and identify sequence homologies. The median spacing between adjacent anchors is about 180 bp on autosomes. Over 75.0% of genomic regions are separated by a pair of anchors having 3 or fewer edges. Larger genomic gaps between adjacent anchors also exist, which comprise about 22% of the standard reference genome's size. These gaps contain a wide range genetic diversity including copy-number variations and other types of structural variations, such as non-tandem duplications caused by transposable element activities. The linear genome of each CC strain can be extracted from the CCGG in the form of standard FASTA files that are compatible for common bioinformatics pipelines. In addition, sequence features (e.g. anchors, genes) with the strain-specific linear coordinates are provided as standard gff3 files to facilitate the genome navigation. Our series-parallel graph structure simplifies the maintenance and management of multiple assemblies, aids in lifting over annotations to alternative genomes, and allows for local sequence refinement. Although sequence compression is not the major goal for CCGG, the total sequence content in the CCGG is significantly smaller than the 8 founder linear genomes from which it was derived (10,653,041,524 bases vs 20,749,378,088 bases). The total size of the 83 linear genomes extracted from the graph could reach to approximately 200 billion base pairs. The Collaborative Cross Graphical Genome (CCGG) presents as the first mouse reference pangenome, which not only incorporates additional information to reduce reference bias for sequence analysis in CC panel, but also introduces a novel data structure and format that provides benefits for sequence mapping, comparison, and visualization.

4.2 Methods

4.2.1 Identifying Anchor Candidates in the CC Population

The CCGG builds upon the fact that intra-specific genomes share many common subsequences. We used a data-driven approach to select an initial set of anchor candidate sequences for the CC population. Multi-string Burrows-Wheeler Transforms (msBWTs) had been previously constructed for the raw sequence reads released from [119] and [120] using the package from [97]. These msBWTs were then queried for every non-overlapping 45-mers from the standard mouse reference (GRCm38) and the counts for the number of reads containing each 45-mer was saved in an occurrence-count matrix for all of the 69 sequenced CC samples reported in [119, 120].

To estimate the number of expected anchors present in the CC population, we simulated the process by sequentially adding the CC samples and measuring the number of anchor candidates lost as a function of the number of samples taken into account. We randomized the sample ordering and estimated the decay of anchor candidates by fitting the exponential decay function.

$$F = ae^{-\frac{n}{t}} + d$$

a is the amplitude of the exponential decay, t is the constant representing decay speed, n is the number of samples taken into account which are continuously extrapolated, and d is the estimated size of the anchor candidates for n approximating to infinity. The result suggests that the fraction of anchor candidates converges to a plateau value of $27.93\% \pm 0.24\%$. The conserved sequences in the CC population ensure the feasibility of constructing an anchor-based graphical genome for CC strains.

4.2.2 CCGG Construction

The CCGG were constructed by merging the standard mouse reference GRCm38 and the de novo assemblies of the 7 CC founder mouse genomes released in [6]. As described in Chapter

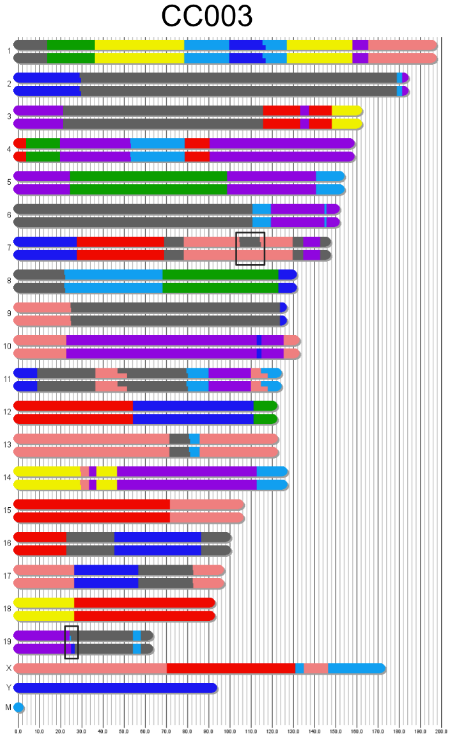


Figure 4.1: Genomes of Collaborative Cross (CC) mouse strains. The CC is a genetic reference population derived from a common set of 8 genetically diverse inbred founders. The genome of each CC strain is a mosaic of its 8 founder genomes. Each founder is represented by a standard color and a single letter label as follows: yellow and 'A' represents the A/J, gray and 'B' is used for C57BL/6J, pink and 'C' for 129s1/SvImJ, dark blue and 'D' for NOD/ShiLtJ, light blue and 'E' for NZO/HlLtJ, green and 'F' for CAST/EiJ, red and 'G' for PWK/PhJ, and purple and 'H' for WSB/EiJ. The ideogram above represents the composition of CC003 genome which is constructed based on the individual CC003 male sample sequenced in [119]. Most CC strains have residual heterozygous regions as indicated by the overlaid boxes. This suggests two possible genomic sequences, one for each haplotype. The ambiguity of recombination boundaries are represented by the small tails within a chromosome (ex. the pink to black transition on chromosome 10)

3, the first stage of the CCGG's construction focused on identifying a set of unique conserved sequences as a set of *anchors*. Anchor candidates were established as 45-mers that are 1) unique in both forward and reverse complement in the GRCm38 reference genome; 2) supported by more than 3 reads in every sequenced sample. As described in Chapter 3, we then determined if these anchor candidate sequences appeared uniquely in other founder assemblies, and that their relative order was consistent with the GRCm38 reference genome. Thus, every selected anchor sequences is conserved, unique, and consistently ordered in every genome represented in the graph. In all contiguous runs of more than three anchor candidate sequences were collapsed resulting in the final

Table 4.1: This table provides graph statistics for each step of the Graphical Genome Construction. The number of candidate 45-mers that are unique and conserved in each chromosome is recorded in the "Candidate" column. The number of candidates are not adjacent to each other is recorded in "Isolated" column. After filtering the unmapped and non-monotonically increased 45-mers, the number of anchor nodes is recorded in "Anchors" column. The edges between anchors are extracted and collapsed and the number of edges is recorded in "Edges" column.

Contig	Candidates	Isolated	Anchors	Edges
Chr1	1218407	793670	731878	2027568
Chr2	1255880	791769	729810	1973207
Chr3	925215	616790	563488	1602269
Chr4	957604	623644	576767	1599066
Chr5	962893	625721	570998	1616686
Chr6	903590	590565	547542	1518758
Chr7	901319	574732	527157	1441500
Chr8	813525	530775	490892	1389359
Chr9	865715	555932	522710	1428942
Chr10	799008	526287	490352	1346334
Chr11	895244	564678	536560	1469633
Chr12	695657	459804	426961	1211442
Chr13	749132	491003	450189	1247819
Chr14	733561	471157	434412	1180558
Chr15	657846	424380	395679	1087265
Chr16	592542	386902	361801	990604
Chr17	606135	388629	351157	988101
Chr18	556844	369483	346785	984510
Chr19	403417	259138	243307	659442
ChrX	109483	96773	76876	326086
ChrY	1927	1717	1717	1718

set of anchor nodes. We next extracted sequences between anchor pairs and merged the identical sequences into edges. The statistics of the anchor nodes and edges are described in Table 4.1.

4.2.3 Adding CC Paths

The founder mosaics of CC haplotypes were previously inferred from CC sequencing data using a forward-backward Hidden Markov Model [119]. The founder probability of a given genomic region was estimated using annotated founder variants of each CC strain [119]. The CC haplotype intervals were reconstructed based on the imputed probabilities of any given region having been descended from a specific founder strain[119]. Taking the genome of each CC strain as a mosaic of their 8 founder genomes, we annotated the CC haplotype intervals over the founder paths in CCGG. Recombination boundaries that occur in regions where the informative probes are sparse or the founder sequences are identical can be ambiguous. In theses cases, the haplotype intervals overlap extending to the variant where there are markers that can distinguish the founder paths. In these overlapping regions we set the recombination boundary by taking the mid-point of these overlapping regions. In heterozygous regions, we used ”a” and ”b” to denote the two different paths, e.g. ”001a” or ”001b”. We provide tools to traverse the graph and reconstruct a conventional linear genome for any specified CC strain from the CCGG. These reconstructed linear genomes are represented as a standard FASTA files, where each sequence represents a strain-specific linear contig. These linear genomes can be used by conventional bioinformatics pipelines. To facilitate feature mapping, we provide a standard .gff file to record the anchor and gene intervals in these CC linear genomes. These features can be viewed in the standard genome browsers such as IGV and help with finding homologous sequences in the strain-specific linear genomes. The haplotype intervals of each CC strain can be extracted from the CCGG by traversing the graph. In addition, We analyzed and identified the CC recombination events within genes and exons regions and reported a list of genes and exons affected by the CC recombination events (Figure 4.13). These CC recombinations in functional regions may result in novel isoforms and further affect downstream transcription or translation process.

4.2.4 CCGG Annotation

The annotations in the CCGG include genes, exons, repeat masks, and variants. The gene and exon intervals on the GRCm38 mouse reference genome were obtained from Ensembl Biomart [123]. The repeat-masker intervals on the GRCm38 mouse reference were obtained from repeat mask website. The variant annotations were obtained by aligning the parallel alternative sequences to the reference path using the same end-to-end scoring function in Bowtie2 [91]. Overall, 46063 genes, 726152 exons, 5700130 repeat-masker regions were annotated. The graph annotation process is described in Chapter 3 in detail.

4.3 Results

4.3.1 Collaborative Cross Graphical Genome

CCGG is of 10,653,041,524 bp in total. There are 9,212,137 anchor nodes (414,546,165 bp) and 30,252,594 edges (10,238,495,359 bp) on autosomes and chromosome X. The linear genomes of both founder and CC strains can be extracted from the graph as standard FASTA files, which can be applied in common bioinformatics or genetics pipelines. Anchors in the CCGG are densely distributed, though some long gaps exist. In Figure 4.2A, we plot the distribution of anchors over the entire genome. On autosomes, there are about 410,000 anchors within every 2Mb bin, giving an average of 200 bp between adjacent anchor pairs. Figure 4.2B illustrates the distribution of the gap spacing between adjacent anchors. The spacing between 94% of the anchor pairs are less than 1000 bp and 50% are less than 180 bp. Anchors are relatively sparse on chromosome X, as shown in the Figure 4.2A and B. We attribute this to two factors. First, repeats are more frequent on X chromosome [129]. Second, anchor sequences have approximately half the number of supporting reads than in autosomes since the sequenced CC samples are hemizygous (males). Selecting anchors supported by more than three reads in every CC sample result in less anchors in chromosome X and this can be refined by using a more flexible standard based on the kmer count distribution on a contig. As a consequence, the gap lengths on chromosome X are relatively longer than on autosomes (Figure 4.2B).

Anchor nodes and collapsed edges represent invariant sequences found in all 8 founder genomes and sum up to 660,797,730 bp (6.20% of the total bases in CCGG). We define collapsed edges as a single edge shared by all 8 founder genomes. Of 30,252,594 edges in total, 2,781,638 (9.2%) are collapsed edges spanning 246,251,565 bp. The parallel edges represent alternative sequences containing variants. The distribution of the number of paths between adjacent anchors is shown in

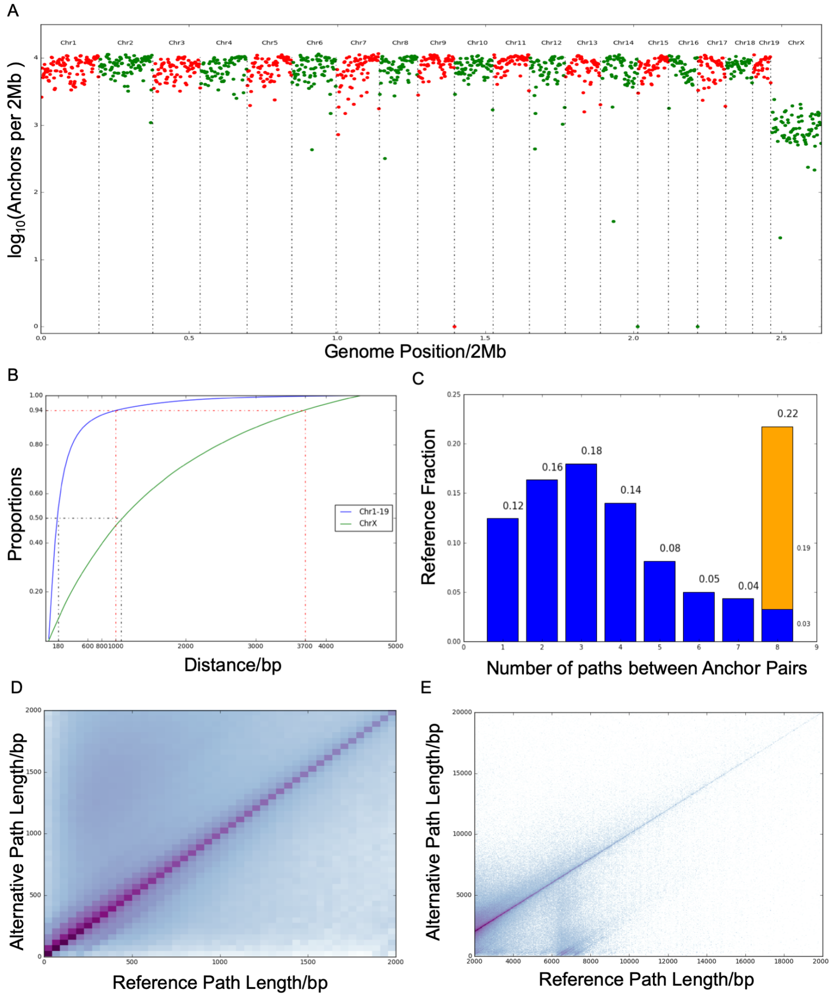


Figure 4.2: CCGG Properties. A) The number of anchors (log 10 scale) in every 2Mb genomic region. B) The cumulative distribution of spacing between anchor pairs. The blue curve represents the distribution in autosomes (chromosome 1-19). The green curve denotes the anchors on chromosome X. Half of the intervals between two anchors in autosomes are less than 180 bp. On chromosome X, the distance is larger (over 1000 bp). C) The distribution of parallel path number between adjacent anchor nodes. The mode is 3, suggesting many shared sequences existing in the CCGG. The length of each interval is normalized by the length of the GRCm38 mouse reference genome (autosomes + chromosome X). The orange bar represents the fraction of long gap sequences, where the reference path is over 1000bp. D) Path Length differences between anchor gaps compared relative to the reference sequence. The density is concentrated along the diagonal showing that reference and alternative paths are usually with similar length. E) For long gaps, the density distribution is more dispersed. It suggests that insertions and deletions are more common in long gaps. The cluster where reference paths are over 6000 bp suggests an excess of deletion events in alternative paths.

Figure 4.2C. The predominant number of edges between adjacent anchors is 3 (18% of the linear mouse reference genome), suggesting that a main fraction of the genome is composed of three

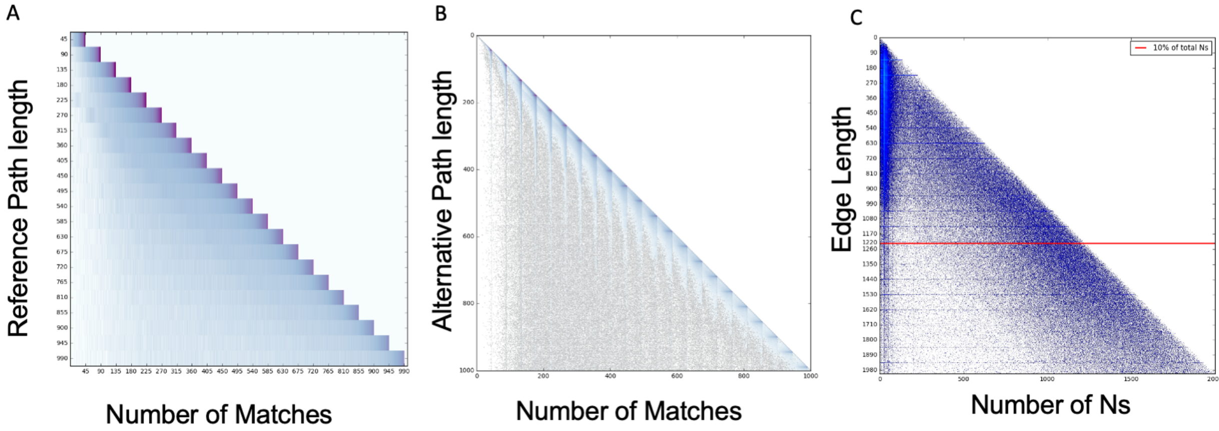
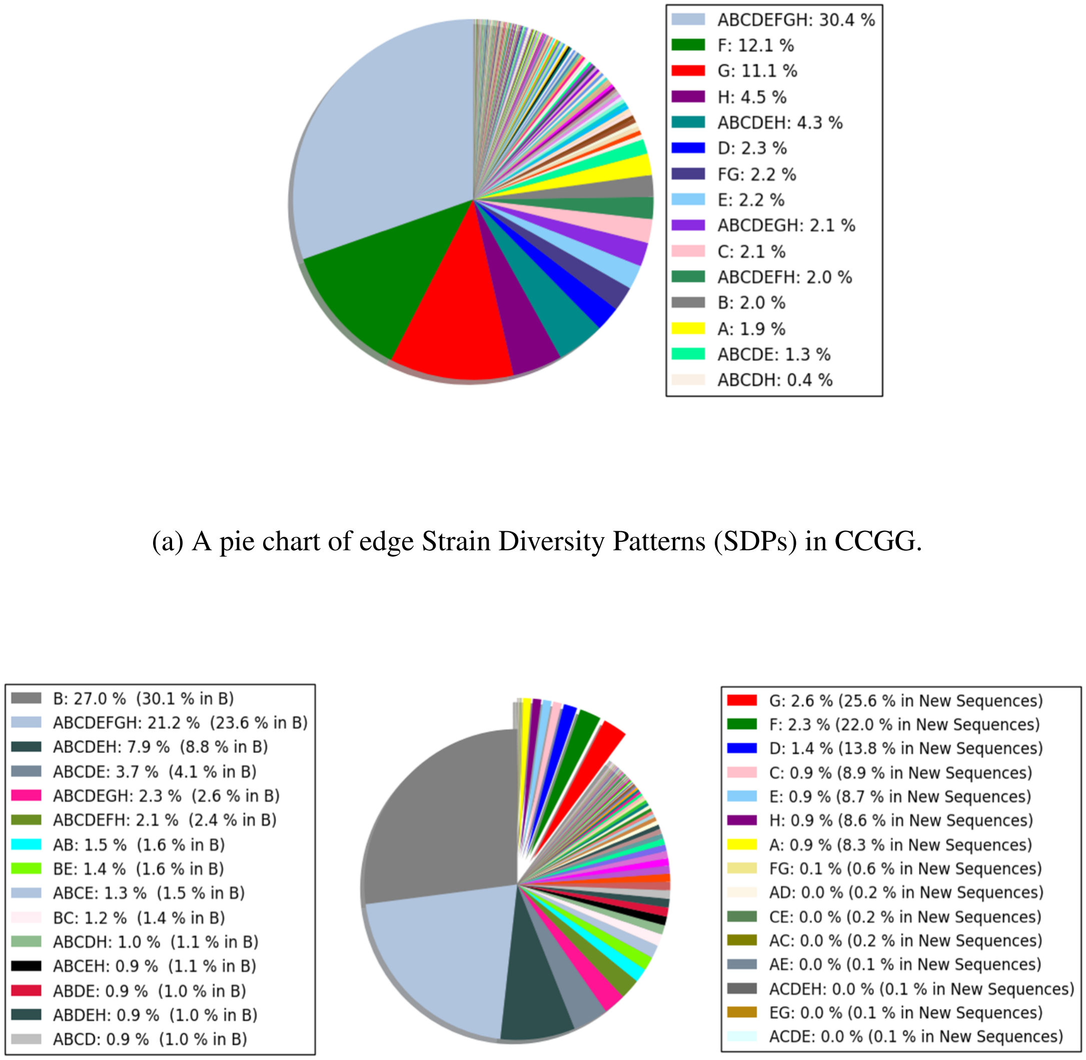


Figure 4.3: Sequence Similarity and Diversity. A) alignment matches between alternate and reference paths compared to the total length of the reference path. Region where reference paths are less than 1000 bp are included in this figure. The quantization visible is due to the fact that anchors divide the reference genome into a multiple of 45bp segments. B) alignment matches between reference and alternative sequences compared with the alternative path length. The density is centered along the diagonal, illustrating substantial sequence are identical between parallel paths. C) Ambiguous base statistics. Ambiguous bases are commonly represented by "Ns" in genome assemblies. The density of 'N' counts is compared by edge length (for edges less than 2000 bps). 10% of 'N's lie on edges less than 1220 bps as indicated by the red line.

haplotypes. A significant fraction of genomic regions is with 2 (16% of the reference genome) or 4 (14% of the reference genome) parallel paths. Fewer are with 5, 6, or 7 parallel paths. The fraction of genomic regions with 8 parallel paths is of a significant increase. This increase results from the correlation between sequence length and the number of variants. 19% of the reference genome with 8 parallel paths lies within long gaps, where the reference path is over 1kb. A small fraction of these 8-path genomic regions fall between relatively short anchor intervals (3% of the reference genome). Theses regions suggest highly variable regions exist in the mouse genome.

Pairwise sequence alignments between alternative and reference sequences are recorded in the "variants" annotation of each edge. The number of identical bases versus the length of the parallel reference path is shown in Figure 4.3A. The number of matches compared to alternative path lengths



(a) A pie chart of edge Strain Diversity Patterns (SDPs) in CCGG.

(b) Illustrates the percentage of new base pairs in the CCGG according to SDP.

Figure 4.4: A) Edge Strain Diversity Patterns (SDPs) in CCGG. Founder genomes sharing an edge are recorded in the 'strain' attribute of edges. There are 255 distinct possibilities. The fraction of edges that exhibit the same pattern are shown in this figure. The top 15 SDPs comprises 80.9% of total sequences and their fractions are given in the legend. B) Base pair percentage of SDPs in CCGG. This pie chart is splitted by reference base pairs and non-B6 base pairs. The non-B6 base pairs include mismatches (SNPs) and insertions on the alternative paths relative to the reference genome. The pie chart includes 2,844,298,072 bp in total, 297,974,379 bp are non-B6 bases.

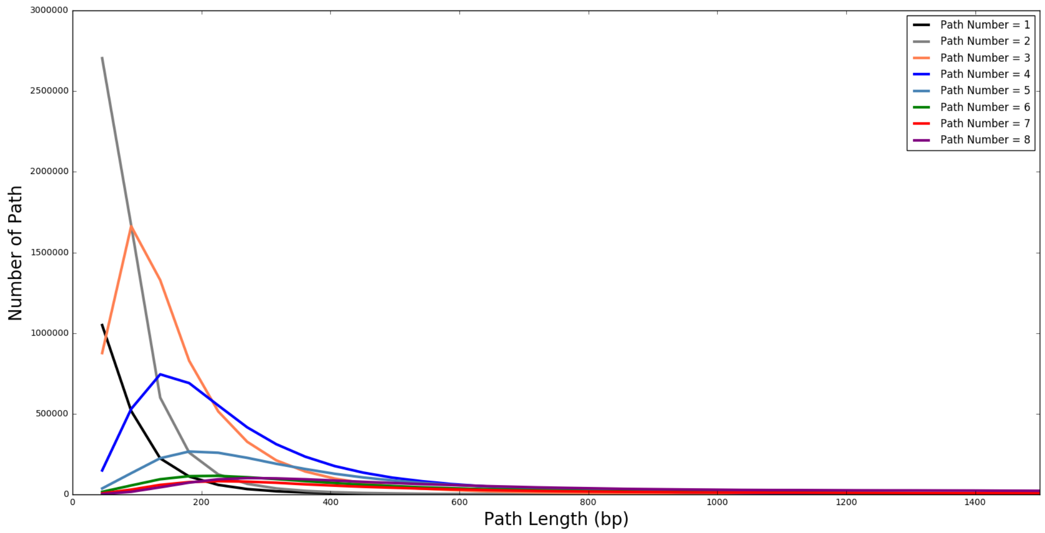


Figure 4.5: Path Length Distribution. The distribution of path lengths between anchor pairs is shown above. As the number of paths increases, the mode of the length distribution shifts to the right suggesting larger genomic regions exhibit more genetic diversity.

is shown in Figure 4.3B. The density is centered along the diagonal, indicating that a large fraction of sequences are shared between the parallel sequences. In addition, as shown in Figure 4.2D and E, most of the parallel edges are of similar length relative to the reference genome. This suggests most common variants are either SNPs or small indels. A large variance in path lengths could serve as an indicator for sizable insertions or deletions. We found that the alternative path lengths in long gaps varies more than in short gaps. Figure 4.2 D and E, illustrate that the density of reference path lengths versus alternative path lengths is more dispersed for long gaps as compared to short gaps. Figure 4.2E also illustrates a cluster of large deletions on the alternative paths, where the reference path is over 6 Kb longer than the alternative path.

The long gaps where the reference paths is more than 1,000 bp generally imply more sequence complexity in the mouse genomes. Both genome assembly issues and large structural variants could result in those long gaps. As discussed previously, unresolved bases (Ns) are over-represented in long edges. The edge length versus the number of Ns in these edge is shown in Figure 4.3C. There are 742,627,244 Ns (27.4% of the total 2,712,033,616 Ns in the CCGG) fall along short edges.

Edges less than 1,224 bps (shown as a red line) contain 10% of Ns. The remaining 90% of Ns lie on long gaps. Long gaps between anchors often imply the presence of structural variants, including large deletions (in either a founder or CC sample), large insertions, translocations, inversions or duplications. Determined by sequencing samples of Diversity Outbred (DO), Morgan et al identified 1,749 copy-number variants (CNVs) in the CC founder strains in [130]. We found that 71.58% (1,252 in 1,749) of these CNVs fall within long gaps. The average length (reference path length between adjacent anchor pairs) of these region with CNVs is 752,856 bp. 88.16% of the regions are longer than 1,000 bp. Considering the fact that more than 90% of the total anchor pairs are less than 1000 bp (Figure 4.2B), this suggests the CNVs are overrepresented in the long gaps.

The Strain Diversity Patterns (SDPs) patterns of CCGG edges also provide interesting insights. We analyzed SDPs of sequences based on founder strain annotations in the CCGG. Since anchors sequences are shared by 8 founder genomes, we treat equivalently to an edge shared by all eight founders. The proportions of the 255 possible SDP patterns in the CCGG are shown in Figure 4.4A. The Top 15 SDPs are illustrated in the legend. The most common component in the CCGG are sequences shared by all 8 founders and comprise 30.3% of the total number of graph entities. SDP private to a wild-derived mouse strains, CAST/EiJ and PWK/PhJ, contribute 12.1% and 11.1% of CCGG entities respectively. Sequences shared by all classical laboratory mouse strains contribute 4.3% of CCGG entities. These results agree with the phylogeny of the 8 founders strains [125, 131, 36]. We estimate the number of non-B6 bases that differ from the mouse reference (GRCm38) in the CCGG. We estimate the new base pairs by mismatches and insertions that are not present in the reference assembly excluding Ns. Since alignments in long gap regions are typically ambiguous, a conservative approximation of the number of additional bases are estimated as the difference in path lengths between alternative and the reference paths in the long gaps, excluding any Ns. As shown in Figure 4.4B, we plot the SDP patterns at base pair level. The pie chart is splitted by reference-genome bases and the new bases. There are 2,844,214,470 bp included in the pie chart in total, 297,890,777 (10.47%) of which are new bases. Private founder sequences comprise of more than 95.9% of these new bases. The wild-derived strains (PWK/PhJ and CAST/EiJ) alone contribute

47.6%. Of the SDP patterns that include C57BL/6J, the most common pattern are sequences on C57BL/6J private edges (B, 27.0% of CCGG bases, 30.1% of GRCm38 bases) which may capture B6-specific genomic features. This followed by conserved patterns shared by all 8 founder genomes (ABCDEFGH, 21.2% of CCGG bases, 23.6% of GRCm38 bases). The next common SDP patterns is the sequences shared between Mus musculus domesticus haplotypes (ABCDEH, 7.9% of CCGG bases, 8.8% of GRCm38 bases), followed by sequences common to all laboratory mouse strains (ABCDE, 3.7% of CCGG bases, 4.1% of GRCm38 bases). Overall, these CCGG statistics provide an overview of the sequence similarity and genetic diversity of the mouse strains.

4.3.2 CC Linear Genomes Extracted from the CCGG

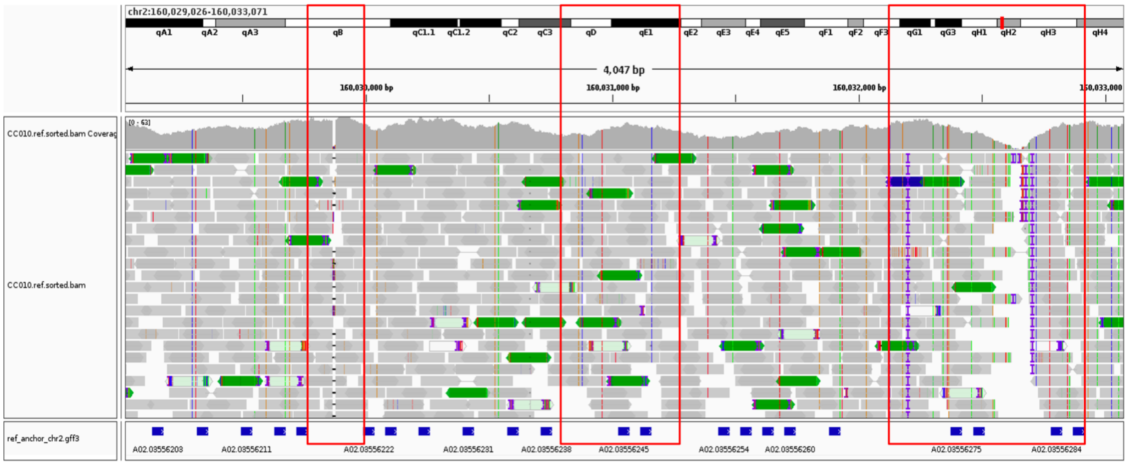
The standard mouse reference genome represents the genomic assemblies of a single inbred mouse strain, C57BL/6J [11]. Most of the genetics and bioinformatics pipelines adopt the mouse reference genome as the standard. Sequence reads are related to the reference assemblies and downstream analysis such as variants calling is performed [12]. CC strains derived from 8 genetically diverse inbred strains and C57BL/6J is one of the CC founder strains [34]. Many non-B6 features have been reported in the CC population [6]. When analyzing CC sequence data, relying on the mouse reference genome will introduce reference bias, especially for the genomic regions descended from non-reference strains. This will lead to an extensive amount of variant calls as the reference allele may not be the present allele in the CC strain. On the other hand, reference bias also have significant impact where large structural variants or non-B6 sequences appear in the CC strain but are absent in the reference assemblies. When aligning reads to the standard reference genome, reads derived from these non-reference regions tend to be discarded or misinterpreted in the downstream analysis.

The CCGG provides CC-specific genomes and annotation files for the research community using CC panels. The linear genomes of both founder and CC strains can be extracted from the graphical genome in forms of standard FASTA files. The CC linear genomes, regarded as a unique mosaic of the 8 founder assemblies, integrate multiple assemblies and ancestor variants in their sequence content. Since anchor sequences are unique and consistently ordered in every linear genome represented on the CCGG, they provide a coordinate framework on the graph by normalizing the individual linear coordinates to the unique occurrence position of the anchor sequence. We provide a standard *.gff3* file to record the anchor location for each reconstructed linear genome. These files can be loaded in a genome browser such as IGV [107] in order to search and visualize the homologous regions in different linear genomes. The CC linear genomes integrate the sequence variations inherited from the 8 founder genomes. The CC linear genomes extracted from the CCGG effectively reduce reference bias introduced by using the *Mus musculus* genome of C57BL/6J as the reference.

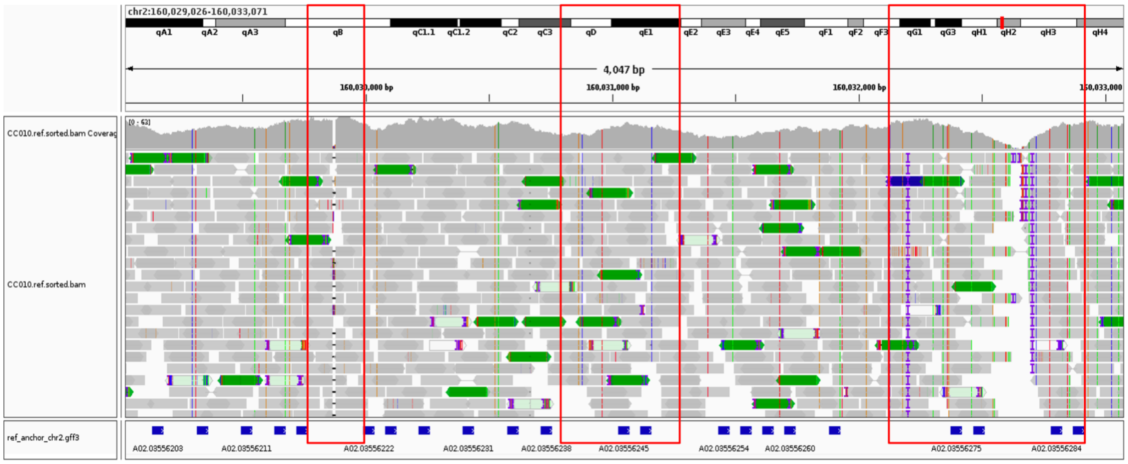
Most Recent Common Ancestor (MRCA) data of the CC strains refers to the sequence data collected from a pool of obligate breeding animals in the core CC colony [132]. These pooled sequence data captures the fixed genomic features of a CC strain and are more robust in characterizing the CC-strain-specific variants than any sequenced individual. We have the pair-end MRCA sequence data for 20 CC strains with deep coverage. We applied a standard aligner, Bowtie2 [91], to align MRCA data to both the standard mouse reference genome GRCm38 and the corresponding CC linear genome extracted from CCGG. Taking CC010 strain as an example, we extracted the linear genome of CC010 from the CCGG. As described previously, we align the CC010 MRCA short-read sequence data to the mouse reference genome (GRCm38) and the CC010 linear genome respectively by using a standard short-read aligner Bowtie2 [91]. We navigate the alignment of CC MRCA reads to GRCm38 and CC linear genomes in a genome browser, IGV[107]. In the region shown in Figure 4.6, CC010 is descended from CAST/EiJ around Chr2:160Mb (GRCm38). CAST/EiJ is a wild-derived inbred mouse strain, whose genome incorporates many non-B6 features [131]. Due to the different genetic background in this region, using the standard mouse reference genome as the template will lead to many variant calls result from the genetic difference between the CAST/EiJ genome and the reference assembly. As shown in Figure 4.6 (a), there are multiple mismatch positions reported in the reference alignment (highlighted by colored bars). A deletion was identified before anchor *A02.03556222* with a low coverage gap, and a insertion before anchor *region before A02.03556275*, reported by the consistent insertions in the aligned reads. When using the CC010 linear genome constructed from the CCGG, where the assembly in this region is constructed from CAST/EiJ assembly, we found the alignment result is much cleaner than to the GRCm38 reference genome. The insertions, deletions as well as the SNPs reported in the reference alignment are fixed. As a result, the read coverage for the indels reported in the reference genome are restored as well. These results demonstrate that using CC010 genome would improve the read coverage and reduce false positive variant calls.

Reference bias will have significant impact where non-reference sequences appear in the sample but are absent in the reference assemblies. These sequences tend to be misaligned or discarded and thus will introduce bias in the downstream analysis. In the previous work, a Transposable Element insertion (TEi) was predicted by using a TEi identification pipeline called Elite [133]. This TEi located on Chr2:150,669,746 bp, positive strand, and is predicted in NOD/ShiLtJ, CAST/EiJ, PWK/PhJ as well as a set of CC strains descended from these founder strains in this region. CC010 is descended from founder strain CAST/EiJ, thus inherit this TEi in its genome. Yet, the TEi sequence is absent in the standard mouse reference assembly (GRCm38). The TEi located in the gap between *A02.03348213* and *A02.03348220*, where the reference path is 270 bp, and the CC010 path (CAST/EiJ path) is 504 bp. Figure 4.8 shows the alignment of the CC010 sequenced sample to the mouse reference genome (GRCm38). When aligning CC010 short reads to the GRCm38, there is a low coverage gap present in this region, suggesting the discordancy of the reference assembly in the CC010 sample (Figure 4.8). In addition, multiple variant spots are reported in this region.On the contrary, when we aligned the CC010 MRCA data to the CC010 linear genome reconstructed from CCGG, we found the read coverage is restored (Figure4.8). Due to the repetitive sequences in the TEi body, this genomic region attracts multiple reads that are either duplicated or their mates mapped to other chromosomes. The result suggests that the linear CC genomes captures these non-B6 strucral variants and refines the sequence alignment in the large insertions.

In addition, another TEi is predicted on chr2:3,567,797(positive strand) by using the Elite [133]. This TEi appears in the A/J, C57BL/6J, NOD/ShiLtJ, WSB/EiJ and the CC strains descended

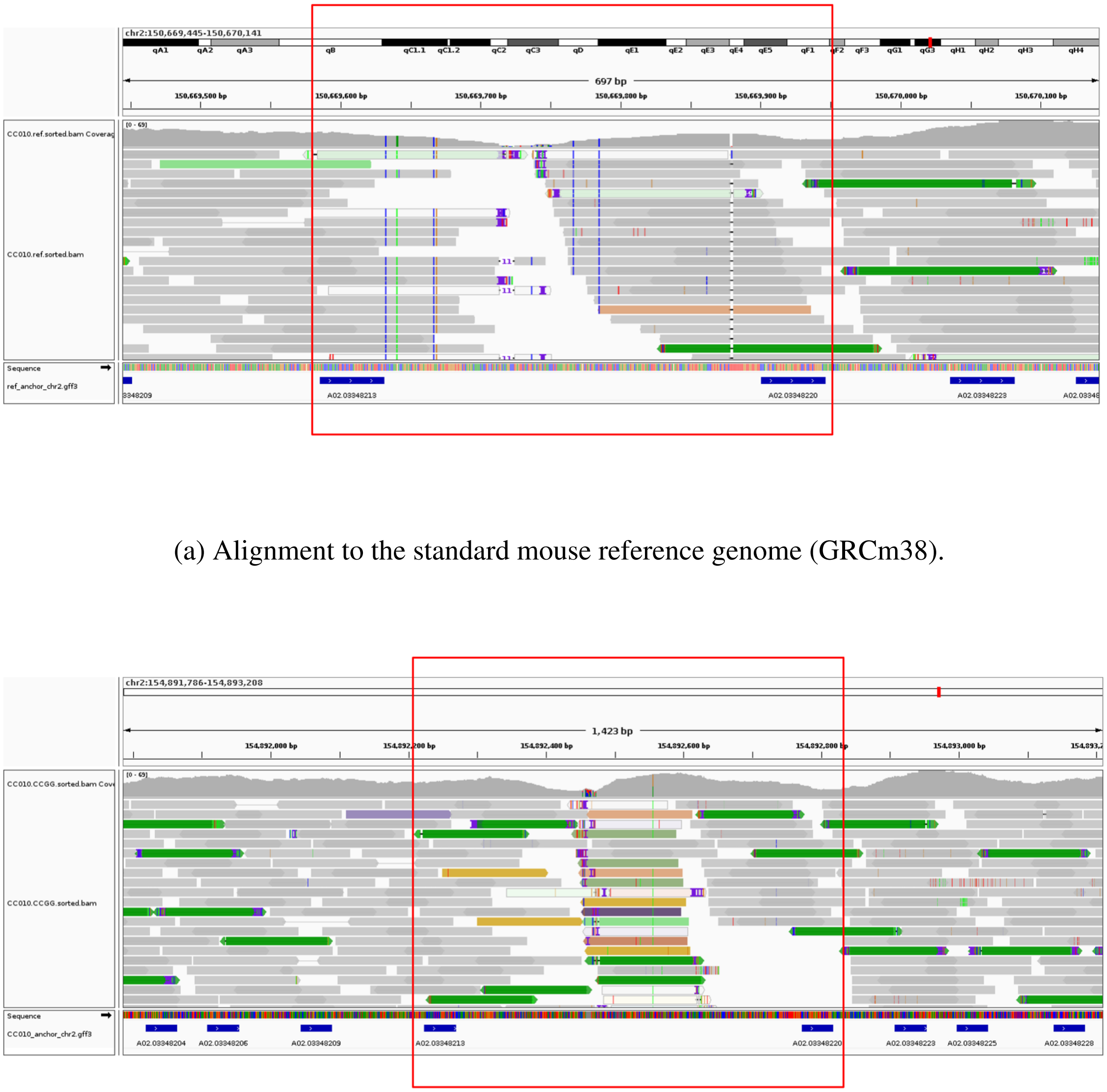


(a) Alignment to the standard mouse reference genome (GRCm38).



(b) Alignment to CC010 linear genome reconstructed from CCGG.

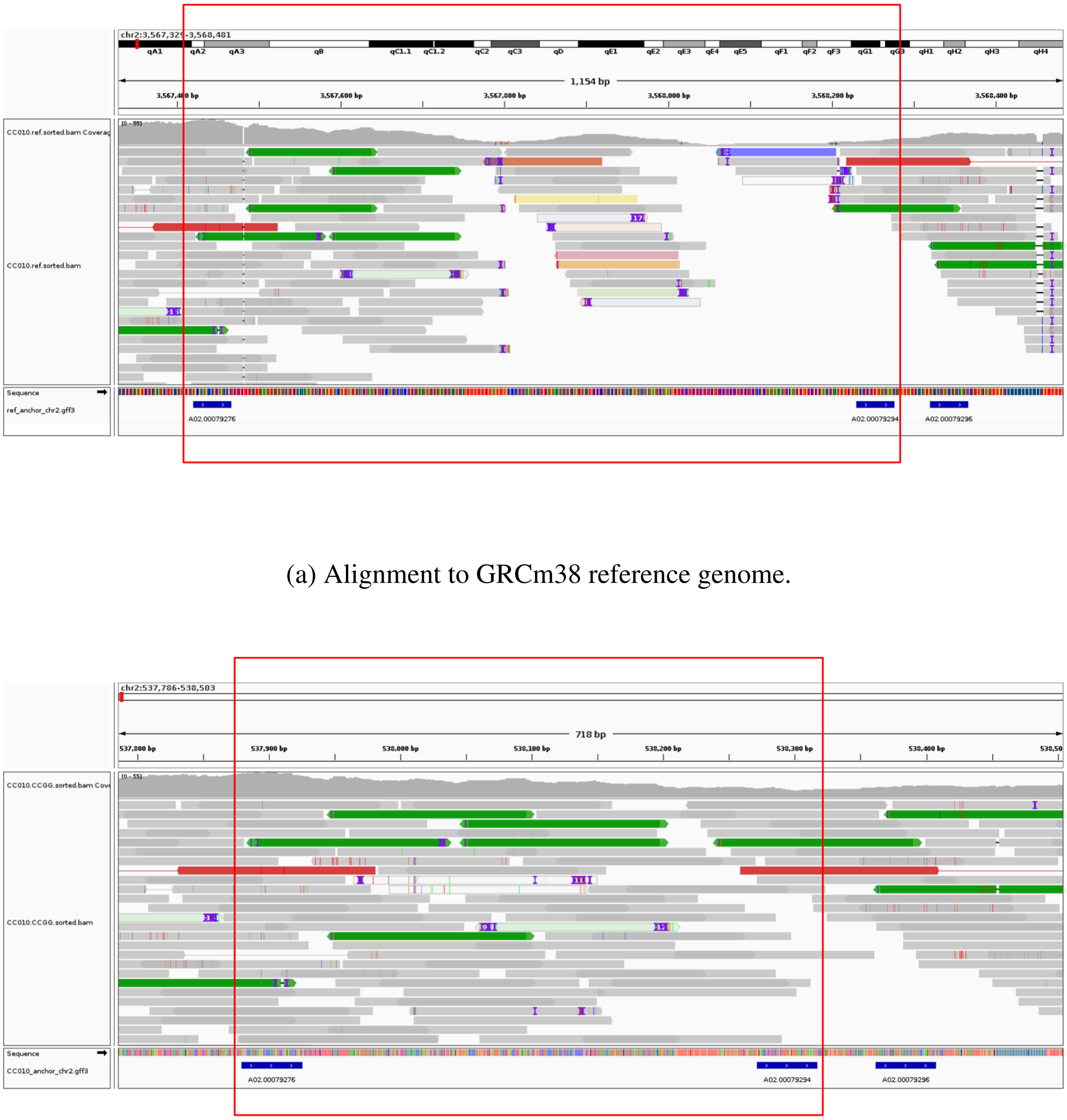
Figure 4.6: CC Genome Fix Variants Calls Resulted from Non-B6 Genetic Background. CC010 is derived from CAST/EiJ in chr2:160Mb. When aligning CC010 MRCA short read sequence data to the GRCm38 genome, multiple variant spots are reported including a deletion (a low coverage gap in the region before A02.03556222), insertions (region before A02.03556275, highlighted by I in the reads), and multiple SNPs (color bars) in this region. Instead, aligning reads to the CC010 linear genomes, these variant calls are fixed, which suggests that these variants are common in CAST/EiJ genome but are not in GRCm38 reference genome. The read coverage in this region is also restored when aligning to the CC genome. Anchor sequences are shown in the bottom track.



(a) Alignment to the standard mouse reference genome (GRCm38).

(b) Alignment to CC010 linear genome reconstructed from CCGG.

Figure 4.7: CC010 Insertion. The plot shows the CC010 alignment in a region with a transposable element insertion relative to the reference genome. A TE insertion are reported in CC010 (Chromosome 2:150,669,746 on the positive strand), which locate between *A02.03348213* and *A02.03348220*. The CC010 path between the anchor pairs is longer than the reference path (504 bp versus 270 bp). Anchor sequences are shown in the track at the bottom. When aligning to the reference genome, few reads cross the low coverage boundary due to the insertion relative to the reference assembly. The CC genome, which incorporate TEi sequences, rescue a bunch of reads aligning in this reads and due to the repetitive nature of these TEi sequences, these reads may map to multiple places in the genome (indicated by the colors). The coverage between the anchor pairs are also improved.



(a) Alignment to GRCm38 reference genome.

(b) Alignment to CC010 linear genome reconstructed from CCGG.

Figure 4.8: The plot shows the CC010 alignment in a region with a reference-specific transposable element insertion. A TE insertion are reported in B6 samples (Chromosome 2:3,567,797 on the positive strand, GRCm38), which is not present in the CC010 path. The reference path between anchor *A02.00079276* and *A02.00079294* is longer than the CC010 path (765 bp versus 347 bp). Anchor sequences are shown in the bottom track. Due to the deletion relative to the reference genome, we observe a low coverage region between these two anchor, yet a few reads are attracted in this region as the TEi sequences are repetitive. This low coverage region is fixed using the CC010 linear genome and the coverage are restored between the anchor pairs.

from these founders. The reference assembly GRCm38 incorporates this TEi sequence. CC010 is descended from the NZO/HlLtJ in this region thus do not acquire the TEi sequence. This TEi locates between anchor A02.00079276 and A02.00079294 in the CCGG, where the reference path is 765 bp, the CC010 (NZO/HlLtJ) path is 347 bp. When aligning CC010 MRCA data to the reference assembly, we found low read coverage in this region. Only a few reads mapped here, as the sequences in the TEi body are usually highly repetitive and appear multiple times across the whole genome. When aligning CC010 MRCA data to the CC010 genome, we found the low coverage gap reported in the reference alignment is fixed and the read coverage are restored. This suggested that our CC010 genome captures the CC010 deletion relative to the reference assemblies and help better interpreting the sequence data.

We compare the global alignment of CC010 MRCA sequence data to the reference genome versus to the CC010 linear genome. For each 100kb non-overlapping sliding window, we identified the bounding anchor of each bin and the corresponding genomic region in in GRCm38 and CC010 genome respectively. We counted the raw read number of CC010 MRCA sample falling within each anchor intervals. The average coverage when aligning to CC010 linear genome is 31.86, which is higher than aligning to GRCm38, 30.50. We then calculated the contrast between the reads coverage when aligning to GRCm38 reference genome and to the CC010 linear genome by

$$C = \frac{R - T}{R + T}$$

where R refers to the raw read count aligned to the GRCm38 reference genome, and T refers to the raw read count aligned to the CC010 genome in each counterpart region. The statistics C ranges from -1 to 1, where 1 refers to no read aligned to CC genome but there are at least one read aligned to GRCm38, -1 refers to at least one read aligned to the CC genome but no read align to the same region in GRCm38. The negative values of C indicate the CC010 genome win over GRCm38 while the positive value indicate GRCm38 win over CC010 genome. We plot the C value across the whole genome in Figure 4.9, where the positive C values are shown in blue, the negative C values are shown in red. It points out genomic regions that CC010 genome attracting more reads than the

GRCm38 genome. In addition, we calculated the number of mismatches per read when aligning CC010 MRCA data to GRCm38 and CC010 genome respectively. Figure 4.10 shows the violin plot of the average mismatches using the GRCm38 and CC specific genome respectively. When aligning CC010 MRCA reads to GRCm38, the violin plot shows bimodal distribution, suggesting a number of reads acquire many mismatches when aligning to the GRCm38 genome. These mismatches are fixed by using CC010 genome extracted from CCGG as shown in Figure 4.10. The median of mismatches using GRCm38 is 2.02, which is higher than the median number of mismatches using CC010 genome of CCGG, 1.90. The standard derivation (std) of mismatches using GRCm38 is 0.71, while using CC010 genome the std is 0.53, suggesting a larger variation of mismatch number when aligning reads to GRCm38. The distribution of mismatch number aligning to GRCm38 and CC010 genome is shown in Figure 4.11.

However, the overall alignment rate aligning to CC genomes did not improve due to the draft quality of the genome assemblies. As shown in Table 4.2, we found that the alignment rate to the CCGG is generally lower than using the GRCm38, which is largely due to the draft-quality of the other 7 founder assemblies except for C57BL/6J. The fraction of ambiguous bases (Ns) in GRCm38 is 2.86% (chromosome 1-19, X, Y, MT). The fraction of Ns in the other seven founder genomes are 10.43% in A/J, 14.85% in 129S1/SvImJ, 22.71% in NOD/ShiLtJ, 13.35% in NZO/HlLtJ, 13.70% in CAST/EiJ, 8.92% in PWK/PhJ, and 15.64% in WSB/EiJ respectively. The percentage of the ambiguous regions in these assemblies are significant higher than the GRCm38. Since the CC genomes extracted from the CCGG can be regarded as a mosaic of the 8 founder assemblies, the CC genomes also comprise of larger fraction of Ns (Table 4.2). These ambiguous regions impact the overall alignment rate of CC genomes. In Chapter 7, we will talk about the strategies we used to improve the sequence content and other potential methods to refine the graph.

4.3.3 Feature Visualization in CCGG

The CCGG anotate genome features, such as genes or exons by overlaying intervals to the reference path of the graph. Gene, exon, and interspersed repeat element are annotated on the

Table 4.2: Comparison of Alignment Rate to GRCm38 and CC-specific Genomes. We align the Most Recent Common Ancestor sequence data to the GRCm38 and CC-specific genome respectively. The overall alignment rate to GRCm38 and to the CC genome is shown in this table. The fraction of ambiguous bases (Ns) on each CC linear genome is also shown in the table.

CC Strain	Alignment Rate to GRCm38	Alignment Rate to CC Genome	Fraction of Ns on CC Genome
CC009	79.04%	76.89%	13.22%
CC010	92.47%	89.50%	13.36%
CC019	92.16%	87.57%	11.97%
CC025	92.16%	89.88%	14.01%
CC029	87.11%	83.76%	12.75%
CC032	93.97%	92.29%	10.01%
CC035	89.96%	87.62%	14.01%
CC043	93.38%	90.96%	14.12%
CC046	91.31%	85.08%	13.75%
CC049	92.35%	84.67%	13.01%
CC059	93.16%	91.08%	11.14%
CC060	90.12%	85.44%	15.67%
CC061	91.25%	89.67%	11.14%
CC072	90.31%	88.68%	12.53%

reference path (B) using CIGAR strings (see Method section). On average, genes span 110 anchor nodes, and exons span 1.58 anchor nodes. The CCGG provides a platform for multi-genome comparisons within these annotated functional regions and strain-specific variants identification. The graph structure places variants in the haplotype context, facilitating visualization and comparisons of these genomic features. The number of paths indicate the distinct haplotypes of a given region. If there is only a single path in the annotated region, the sequence is conserved. For the regions with multiple paths, one can further perform multi-sequence-alignment and identify sequence variants within the regions. Most of the SNPs and tandem indels relative to reference genome can be identified by scanning the variant cigar string on each path.

The reference-specific variants, which are difficult to be detected and represented in the standard VCF files, can be identified and intuitively visualized on the graph. Figure 4.12 illustrates how the graphical genome help visualize and represent a reference-specific variant in the Gabra2 gene. A recent study reported a C67B/L6J specific, functional non-coding variant in the Gabra2 gene body [134]. It is a single base-pair deletion that were fixed in C67BL/6J between 1976 and 1991, which is ahead of the creation of CC panel. Thus the CC population should inherit the C67BL/6J-specific

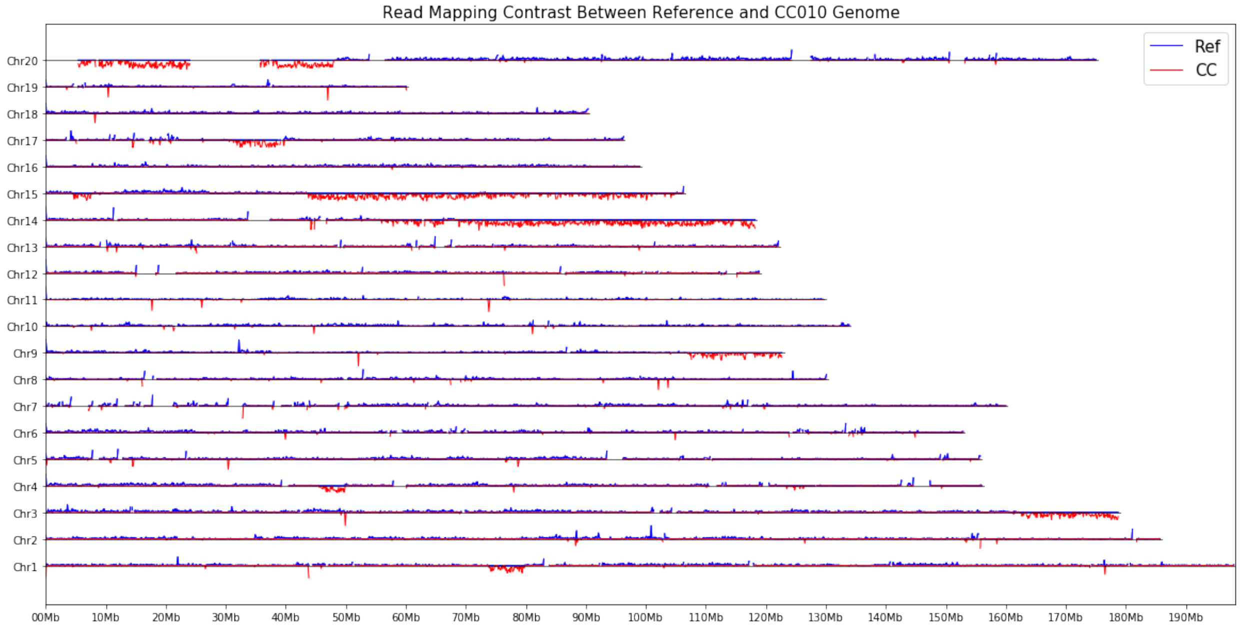


Figure 4.9: Contrast of Read Alignment to GRCm38 and CC010 Genome. Aligning CC010 MRCA short-read sequence data to GRCm38 and CC010 genome extracted from CCGG, we counted the raw reads aligned to each non-overlapping 100kb bin in the CC010 and the GRCm38. The CC010 and GRCm38 coordinates are aligned by using anchor pairs at the boundary of each bin. We calculated a statstic C to evaluate the alignment to GRCm38 and CC010 genome, where the positive value indicates GRCm38 aligned better and the negative value indicate CC010 linear genome is better. The positive values are shown in blue and the negative values are shown in red. Null values indicate no reads are aligned to these genomic regions.

deletion. We examined the CCGG subgraph in Gabra2 (Figure 4.12) and found 387 anchors in this region and 27 anchors fall within exons. Given the position reported by [134], we found that the variant locates between anchors A05.01578094 and A05.01578114. There are 5 distinct paths between these two anchors. The reference path (B path) is private, suggesting C57BL/6J specific sequence features in this region. In addition, all the parallel edges share an insertion of "A" where 411 base pairs distal from anchor A05.01578094, indicating a deletion of "A" in the reference. In addition, differences in lengths between parallel and reference path length indicate potential insertions or deletions. There is another deletion located between anchors A05.01579089 and A05.01579115, where the path length of PWK/PhJ and CAST/EiJ haplotypes are significantly shorter than other founder sequences. This is consistent with the Sanger Institute reports that

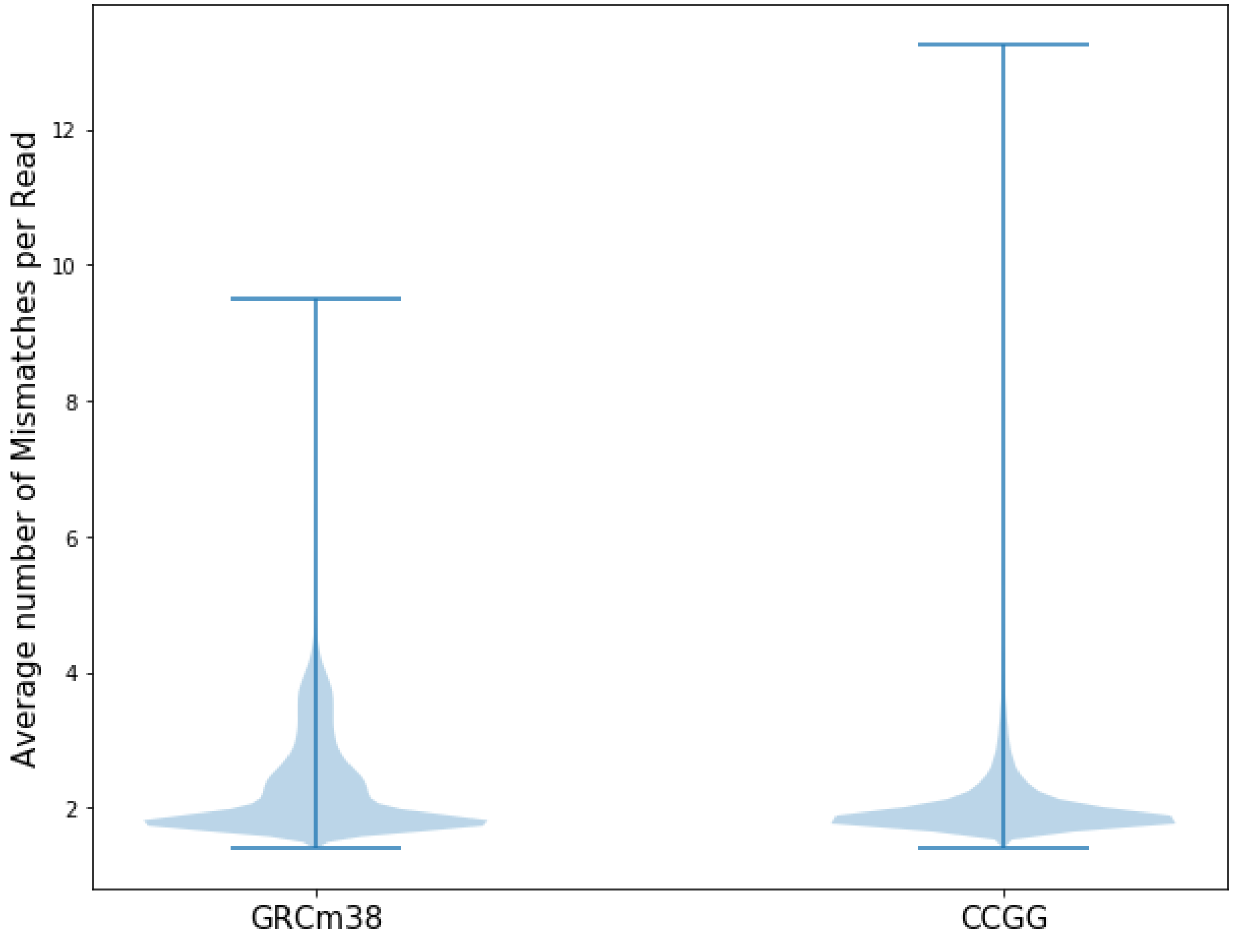


Figure 4.10: Average number of Mismatches per Read Mapping to GRCm38 and CC010 Genome. The average number of mismatches per reads are shown in this picture when aligning CC010 MRCA data to the reference GRCm38 and the CC010 genome extracted from CCGG. The range of mismatch values are represented by the bars.

a structure variant (Chr5:71,059,388-71,059,864 bp) in the wild-derived strains, PWK/PhJ and CAST/EiJ.

CCGG also provides a integrative framework to understand the recombination events occurred in the CC population. On average, recombination occurs every 358,190 bp (roughly 3 per Mb), and is enriched at the end of each chromosome. The CC recombination events that occur in the functional regions, such as genes or exons, may result in novel genome structure and further affect downstream transcription or translation process. We analyzed 46,036 protein coding regions in all 75 CC strains. We found 3,293 distinct recombination events occurred in 2108 annotated gene regions in 75 CC

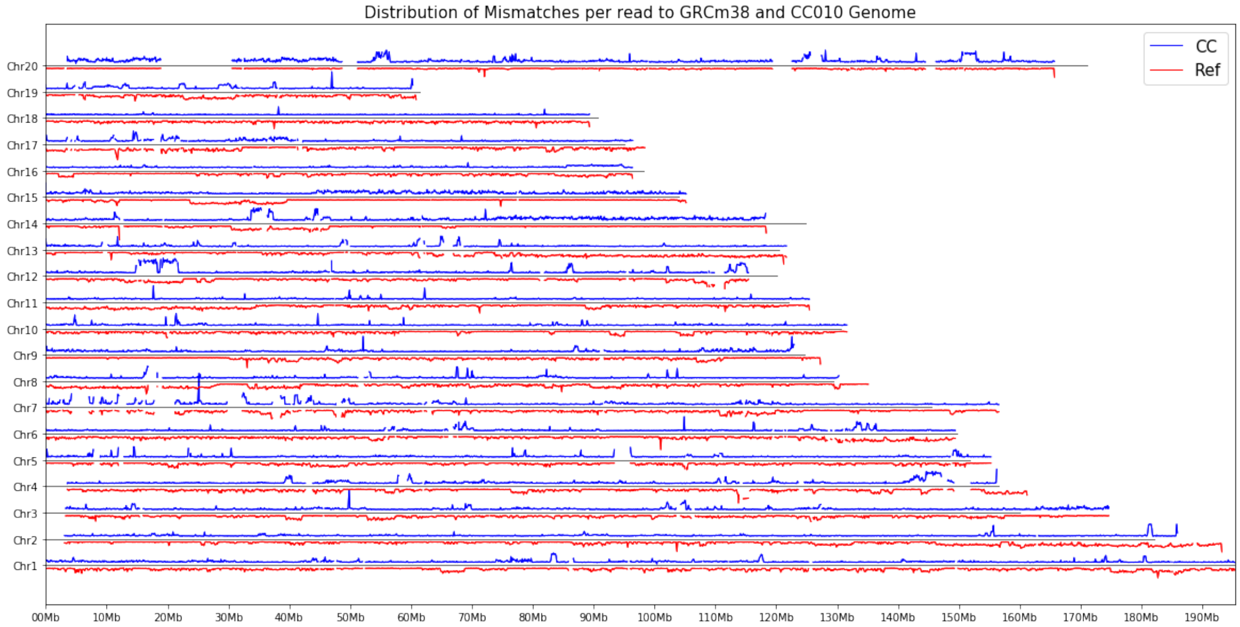


Figure 4.11: Distribution of Mismatch Numbers Aligning to GRCm38 and CC010 genome. Aligning CC010 MRCA data to the reference GRCm38 and the CC010 genome extracted from CCGG, we calculated the average number of mismatches per reads in every 100Kb bin. The distribution of mismatch numbers are shown in this picture (CC010 genome are represented by positive values and reference genome are represented by negative values centered by the grey line). The coodinates of GRCm38 and CC010 linear genome are aligned by using the bounding anchors of each bin.

strains; 96 distinct recombination events occurred in 90 exon regions in 50 CC strains. Details can be seen in http://devel.csbio.unc.edu/GraphicalGenome. As shown in Figure 4.13, three CC strains, CC052, CC061 and CC081, have recombination within a single gene, *Amt*. Exons exist both before and after the recombination boundaries, which might lead to novel isoforms in those CC strains. Fgfr2, Gm20388, Osbpl10, Pde4d, and Prkg1 contain more than 10 recombination events in the CC population. Among them, Gm20388 (gene length 4,434,881 bp) contains the largest number of recombination events involving 17 distinct CC strains. In addition, the recombination within gene bodies can be complex. For example, as shown in Figure 4.13B, CC078 transits from haplotype C to A and back to C within Il20ra. We provide a list of genes with potential new isoforms in the CC population and are interesting targets for further biological exploration.

Overall, the CCGG combine genomic resources from multiple linear genome assemblies to achieve a comprehensive reference model for CC population. It enables the exploration of genomic variants and recombination events across different CC samples.

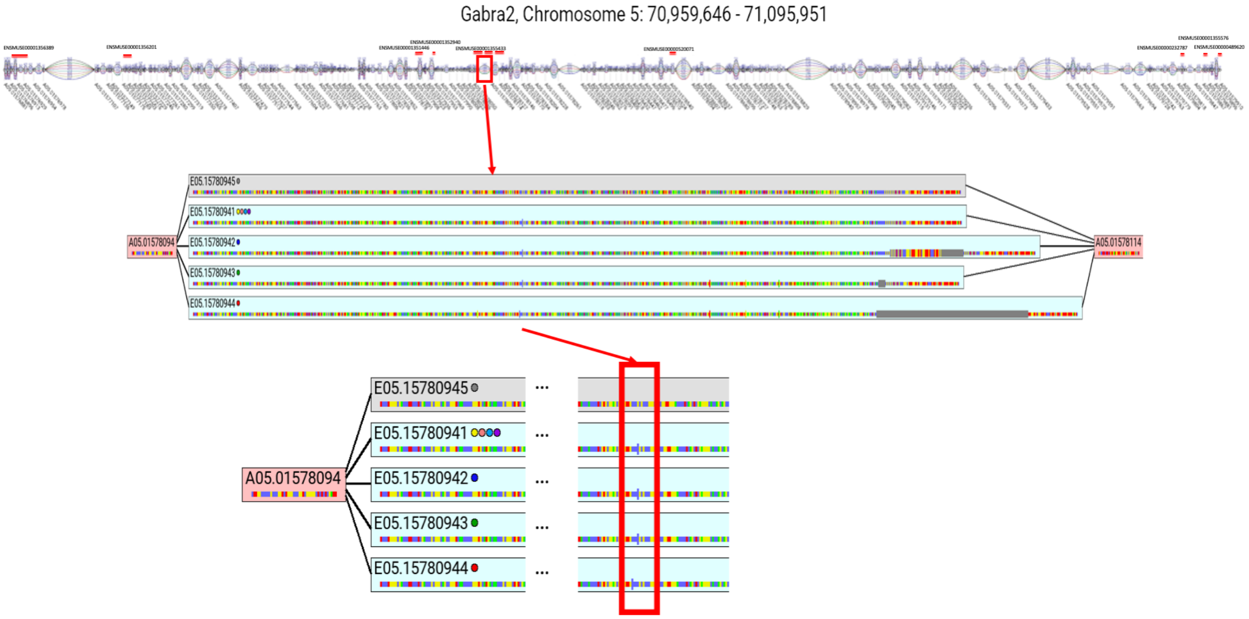


Figure 4.12: CCGG help identify, compare and visualize strain-specific variants. A C57BL/6J specific functional variant has been identified in Gabra2. This functional deletion exists between anchors A05.01578094 and A05.01578114. All the paralleled path except for B path have the insertion of "A" in the same region, indicating a deletion of "A" in B path.

4.4 Discussion

In this Chapter, we introduce a pangenome resource for searching, annotating, and comparing genomic assemblies for a commonly used mouse genetic reference population, Collaborative Cross. It represents genomes of 8 founder and 75 CC strains with 2,091 contigs in a single graph representation. CCGG captures important genetics aspects such as conserved genomes, and highly variable genomic regions. A special class of conserved nodes, which we call anchors, are introduced in CCGG, which organizes genomes into disjoint segments and provides a new framework for labeling genomic regions while maintaining monotonicity and backward-compatibility. It also allows annotations in multiple genomic contexts. CCGG also supports incremental updates as the new reference assembly is released without globally lifting over all the annotations. The CC strain-specific genomes can be extracted as classical linear genome from CCGG. These genomes can be used in the standard bioinformatics pipelines, such as bowtie2. We show that using the CC-specific linear genome can reduce the reference bias and improve coverage of the CC MRCA

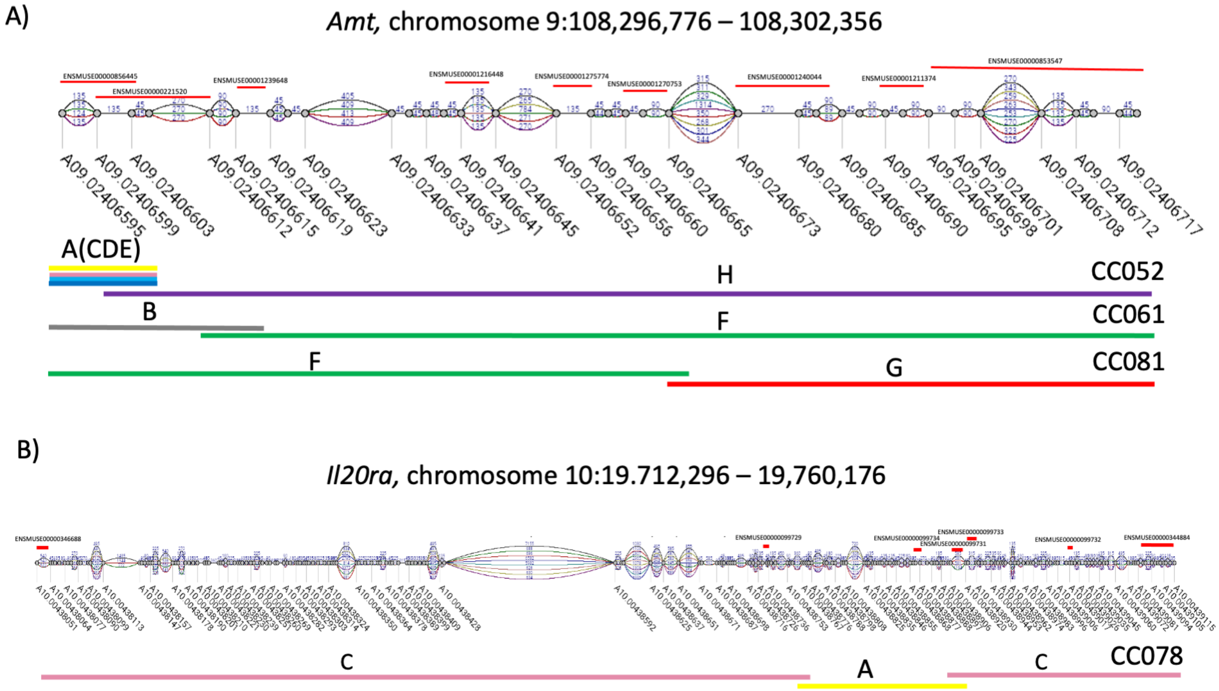


Figure 4.13: CC Recombinations in Genes. A) Three CC strains, CC052, CC061 and CC081, have recombination within a single gene, *Amt*. B) CC078 has two recombination events (C-A-C) within a single gene *Il20ra* that impacts exon sequence.

sample. The graph structure also facilitate searching, and visualizing genomic features and supports the tool chain development for pangenome analysis. The properties of graphical genome are informative for genomic discovery, such as the identification of structural variants and genome-wide association studies. Combining the inferred CC haplotype intervals with CCGG helps the investigation of recombination events and the functional analysis of genes. Lastly, the CCGG is a versatile representation of genomic sequences with a trade-off between space savings and provide useful functionality by preserving the variants within the haplotype context. The CCGG can be extended to incorporate new inbred strains and other recombinant inbred lines. For example, the utility of the CCGG can be expanded to include nearly 100 additional BXD strains by including a single strain, DBA/2J [124, 135, 136]. The CCGG is also useful for characterizing outbred mouse population, such as the Diversity Outcross (DO) population [137], an outbred population derived from the same set of founder strains with fine-mapping potential. Sequences of DO strains can be aligned to the CCGG to identify the optimal path of each individual and perform downstream analysis. This will reduce the reference bias introduced by using B6 genome as the standard for downstream analysis.

The future direction includes genomic comparison studies of CC strains and to further refine the recombination boundaries. The chain of anchor-edge sequences provides a high-level abstraction of the genome that allows synteny to compare between homologous paths much more efficiently than direct sequence alignments at chromosome scale. Sequences between anchor pairs can be compared directly to assess their similarities and differences at a more accurate level. Combining with comparative phylogenetic methods, informative illustrations for the CC genealogy can be obtained from the CCGG. We also propose that genome-wide association methods can be developed using the CCGG, which provide a more comprehensive profile of genomic diversities in the CC population. Currently, most of the genome wide association studies focus on SNPs. The CC graphical genome provides an effective way to identify potential structural variants and incorporate them for further association studies.

Another direction is to improve the sequence content of the CCGG. The sequence content of the current version of CCGG are derived from the 8 founder linear assemblies. However, CC lines will introduce private mutations fixed during the breeding process to the graph. Examples include the private SNPs and large deletions reported in [119]. Moreover, many pooly-assembled genomic regions, represented by runs of N's, exist in the 7 new assemblies from [6]. Long read sequencing technology can be applied to resolve these genomic complex region. Furthermore, taking anchor sequences as seeds enables local assembly tasks and will greatly simplify the genomic assembly process. Not only the founder sequence, but also the CC strains descended from the same founder can be used for the local assembly tasks. There are more than eight CC strains on average representing each of the eight founders in a given genomic region. This is equivalent to leveraging sequence data of biological replicates with more than 160x genomic coverage to perform the assembly task. This should greatly reduce the impact of private variants or technical noise introduced by any individual sample.

Meanwhile, much efforts should be dedicate to constructing or refining the graphical genome of chromosome X, Y and mitochondria genome. The chromosome X, Y and mitochondria genome are composed of many repetitive elements [129]. This implies that there will be less anchor nodes in these chromosomes. For chromosome X (171,031,299 bp), the number of anchors is 89.5% less than the comparably sized chromosome 2 (182,113,224 bp) (Table 4.1). The cumulative distribution of anchor distances on X is significantly different from the distribution in autosomes (Figure 4.2B). These results suggest that anchors are relatively sparse on chromosome X than autosomes. In addition, the founder genome assemblies released from [6] do not include sequence for Y. Thus, we can potentially construct a draft chromosome Y graph using unique 45-mers present in all male CC samples as anchors and perform local assembly between each pair of anchors.

It stands as a draft for future assembling better models for Y in CC strain genomes. We also constructed a mitochondria graph by integrating the *de novo* assemblies of both founders and CC strains based on the Sanger sequence data. We concatenate the starting and ending edges and construct a circular graphical genome for the mitochondria genome representation. The potential application includes identifying CC private or segregating mutations in the mitochondria genomes. The preliminary results suggest that there are about 8 CC-specific mutations in the Mitochondria genome.

CCGG presents a comprehensive view of sequence diversity in Collaborative Cross and provides an alternative genome model for maintain the genome resources of the CC panel. It assists the development of future bioinformatic tools and biological assays for the CC analysis. We show that relying on a single linear reference genome derived from an inbred strain is not sufficient for the recombinant-inbred population. Multiple reference assemblies need to be applied to the common bioinformatics analysis of the CC strains. The CCGG contributes for providing the CC-specific genomes and annotation in the standard file formats to facilitate the research using CC mouse. It served as a substrate for further reference pangenome construction and downstream analysis tool-chain development.

CHAPTER 5: VALIDATION OF THE COLLABORATIVE CROSS GRAPHICAL GENOME

5.1 Introduction

Recent advances in sequencing technology have enabled the assembly of multiple intra-specific genomes [4]. These studies suggests that using a standard linear reference genome imposes limitations to the genetic and bioinformatics analysis [27]. For example, large amounts of sequences in the human sub-populations, such as African populations, are not present in the human reference assembly [26]. Sherman et al showed that the novel sequences of the African-specific genome comprise of about 10% of the total genome size, which do not appear in the reference assembly [26]. In order to obtain a comprehensive spectrum of the common human genetic variation, the 1000 Genome project are conducted collecting openly consented healthy individuals from worldwide populations [29, 25]. The data resources of the 1000 Genome project has been applied in many biomedical research [25]. However, these population-scale genomic analysis has not yet been generalized to other species. The genomes of other species, such as mouse genomes or plants are more variable than the human genomes [27]. A recent study on maize genomes suggest a large fraction of genomes between inbred lines are not alignable using the standard linear reference genome [138, 30]. The reference bias could impacts the downstream study in the analysis of these species.

Aligning short-read data to a reference assembly is usually the first step in typical bioinformatics and genetics pipelines [72]. Based on the read alignments, downstream analysis identifies genetic differences from the standard reference genome and assigns these genetic differences to haplotypes. However, aligning short-reads to a single reference assembly is blind to many non-reference structural variants and can lead to errors, or even failures in highly divergent regions [2, 4]. Meanwhile, as complex recombinant-inbred panels of model organisms are developed, it becomes insufficient to use a single-sequence assembly as the choice of reference [119, 43]. An example is the Collaborative Cross (CC), whose genomes are a mosaic of eight genetically diverse founder genomes [34]. However, applying multiple genome assemblies to the sequence analysis practices faces challenges. It has been suggested that performing sequence analysis based on pangenome models is a promising way using multiple genome assemblies [4, 5]. Pangenome models are commonly represented as graphs with directed or bidirected edges [4]. These sequence graphs provide a efficient framework to represent sequence variations and compress redundant sequences amongst multiple intra-specific genomes [4, 5]. Algorithms related to pangenome construction and analysis have been rapidly developed [57, 79, 77, 76, 80]. A common pangenome graph representation is to insert well-documented sequence variants (ex. Single Nucleotide Polymorphisms, SNPs, small insertions, or deletions, indels) into a linear reference genome [77, 76, 80]. Genotyping methods based on a reference pangenome have been developed, which are mainly derived from graph alignments of short reads or long reads to a pangenome graph [77, 80, 73, 103, 104].

The Burrows-Wheeler Transform (BWT) is a data structure that has been widely used in genomic sequence analysis [66]. The BWT is commonly used to represent genome assemblies as the alignment template, where seed k-mers from sequenced reads are used to place read to a reference assembly [91]. A Multi-String BWT (msBWT) supports searching substrings in a collection of sequences [96]. The msBWT has been applied to compress multiple sequences, such as contigs from multiple assemblies or short reads from a sequenced sample [97]. Many algorithms have been developed to construct an msBWT data structure [98, 99]. BWT-based query tools employ an auxiliary data structure known as the FM-index, that provides the access to intervals of an implicit suffix array of the BWT. This allows for exact searchs for any substring of length k within a BWT in O(k) time [94, 95]. The searching algorithm leverages a property of BWT, last-first (LF) mapping, i.e. the ith occurrence of a character in the BWT string corresponds to the ith occurrence of the same character in the first column of the sorted cyclical suffixes. It is perform in reversed order of the substring, from the last to the first symbol. Each symbol identifies a shrinking range of the suffix array. The extent of the final index range indicates how many times that the substring appears in the template. A zero-length interval indicates that the substring does not appear in the template. One can retrieve the occurrence position of the substring in the template based on the index range in the suffix array.

Aligning sequences to multiple genome assemblies or sequence graph is both challenging and time consuming. When variant branches appear in close proximity, the graph needs to be augmented to account for path combinations that co-occur in observed haplotypes in order to avoid exponential searches. Moreover, the linear genome assemblies adopt different linear coordinate systems, where homologous bases correspond to completely different indices. Aligning short reads to multiple genome assemblies introduces complications when merging inconsistent or even conflicting alignment results. An alternative is to analyze sequence data based on the number of occurrences of a selected set of k-mers (substrings of a fixed length of k) as the probes in the raw short-reads. Many k-mer counting methods have been developed and applied to the problems of genome assembly, abundance estimation, and error correction [75, 84, 139, 105, 106]. Leveraging k-mer count information in raw short-read data, approaches have been developed for genotyping variants in a specific genomic region or at whole genome level [105, 106]. Several k-mer based genotyping tools have been developed as well. Colored de Bruijn graphs have been proposed to overlay short-read sequenced samples onto a graph [19, 84]. These enable genotyping variants by using k-mer counts from the short-read sequence samples directly. BayesTyper uses the mapping of read k-mers to a reference graph to perform unbiased, probabilistic genotyping across the variation spectrum [105]. Dolle et al. proposed a reference-free compressed data structure, a population BWT, and used it to store and index the sequenced reads from the 1000 Genomes Project samples [106]. This BWT enables non-reference queries across the entire population and the discovery of SNPs and indels using this compressed data structure.

The k-mer based methods bypass the alignment step and are fast, flexible and more memory efficient when compared to alignment-based methods [75]. However, analyzing sequence data by k-mer queries loses the k-mer connectivity implied in contigs, which is problematic in repetitive regions or complex genomic regions [140]. Combining the haplotype information with these kmer counting tool is a potential solution to address this problem. Pangenome models commonly utilize graphs to merge and partition multiple genome assemblies to avoid sequence redundancy [4]. Thus they both preserves the long range k-mer connectivity and compresses the redundant sequences among multiple assemblies. Haplotypes can be overlaid onto a pangenome graph to idenitfy paths [84]. The graph-based pangenome models reduce the amount of k-mers to be processed and identify the informative probes that contain sequence variants in a subset of strains. One can count the k-mer frequency in the short-read sequence data to characterize the genomic features in each individual sample. However, few methods have been developed to apply k-mer counting methods to access the origin of and differences between genomic regions in samples from an admixed population.

In this Chapter, we validate our graphical genome by constructing a CC probe database. The CC probe database integrates population-scale sequenced CC samples with the pangenome model. We propose a sequence analysis pipeline that applies the fast k-mer counting tool with a pangenome model, while maintaining the genomic origin and graph topology of each k-mer. We first select a set of k-mers from the pangenome model, that covers every base pair in its constituent genomes. The selected probes capture both shared and variable sequences in the population. We then collected these probes and their reverse complements, sorted them by their suffixes, and removed repeated sequences. We next counted how many times each probe occurs in the sequenced sample. We developed a fast bulk-query method to speed up the overall execution time of querying a large k-mer set. It exploits shared suffixes of k-mers to perform efficient FM-index accesses in a Multi-String BWT (msBWT) [98, 96, 120, 119] of sequenced reads. As a preprocess, the k-mers are sorted by suffix. We keep track of index ranges between successive queries by using a simple stack data structure. This ordered search significantly reduces the execution time for bulk k-mer queries as compared to executing searches in random order. This algorithm is initially proposed by a former lab member Maya Najarian [119]. We then unravel the sorted order to map both forward and reverse-complemented k-mers to a count matrix. All haplotype paths in the pangenome are represented as a series of row indices in a count matrix, thus eliminating redundancy. The CC probe database serves as a genomic resource for analyzing population-scale sequence data and characterizing genomic features across multiple samples. This representation can then be used to assess any new sample's relationship to the pangenome. When new sequenced samples are released, the kmer count of the CC probe database can help identify optimal path to explain the sequenced sample in the graph.

Overall, we present a k-mer tool to analyze population-scale short-read sequence data based on a pangenome model. We validated the sequence content of CCGG by examining the CC probe counts in 113 sequenced samples. The CC probes serve as a standard set of k-mers that can be queried when new CC samples are released and their counts can be incrementally appended as new columns to the count matrix. Our method scales well when adding new genome assemblies or haplotypes to the pangenome model. When threading other intra-specific assemblies to the pangenome, only a small fraction of new k-mers will be added to the k-mer set, as most are shared with other genome assemblies.

5.2 Methods

Our method assumes that the CCGG captures all of the expected haplotypes and sequence variations of the CC population. This assumption is feasible for the CC population as it is derived from a common set of ancestors with minimal genetic drift. The anchor nodes are connected by one or more labelled edges that include all of the genomic variants in the population. This includes simple point variants like SNPs, indels, as well as larger structural variants like transposable element insertions, segmental duplications, inversions, and other rearrangements. Edges are typically shared by a subset of assemblies in the model. The connectivity of the graph is specified by the source and destination attributes of each edge. Edges incorporate sequence variations in the graphical genome.

We constructed our probe database by selecting a representative set of k-mers from the CCGG. These k-mers covers every base pair in the pangenome model. Furthermore, we classify those probes into three categories: the anchor probes that are conserved in the founder and CC genomes, the informative probes that discriminate between haplotypes and the bridging probes that are shared

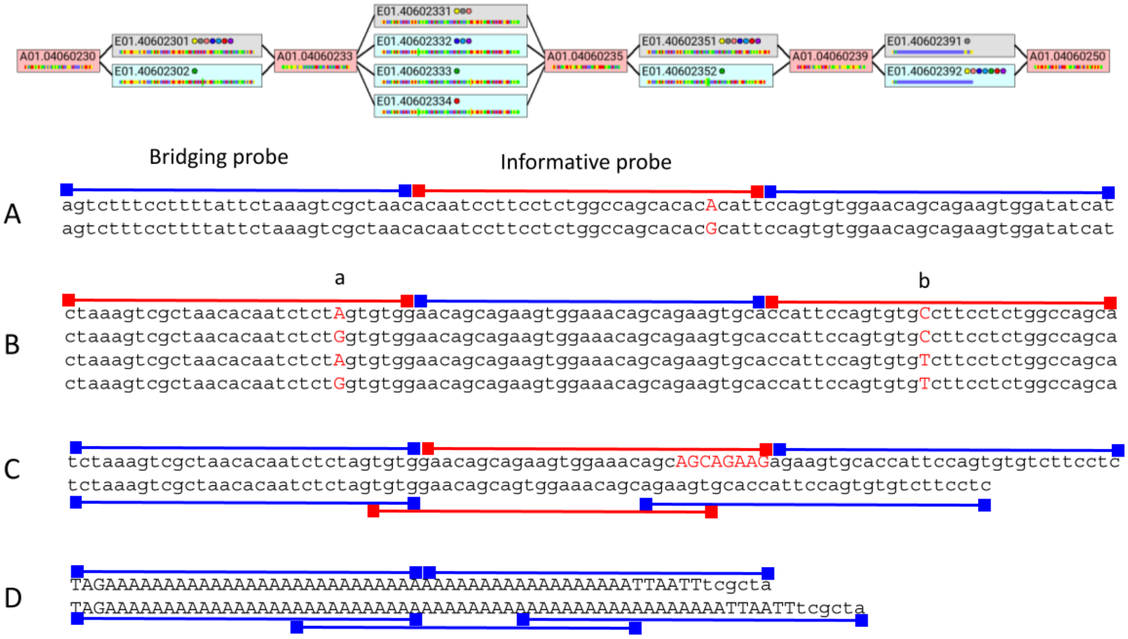
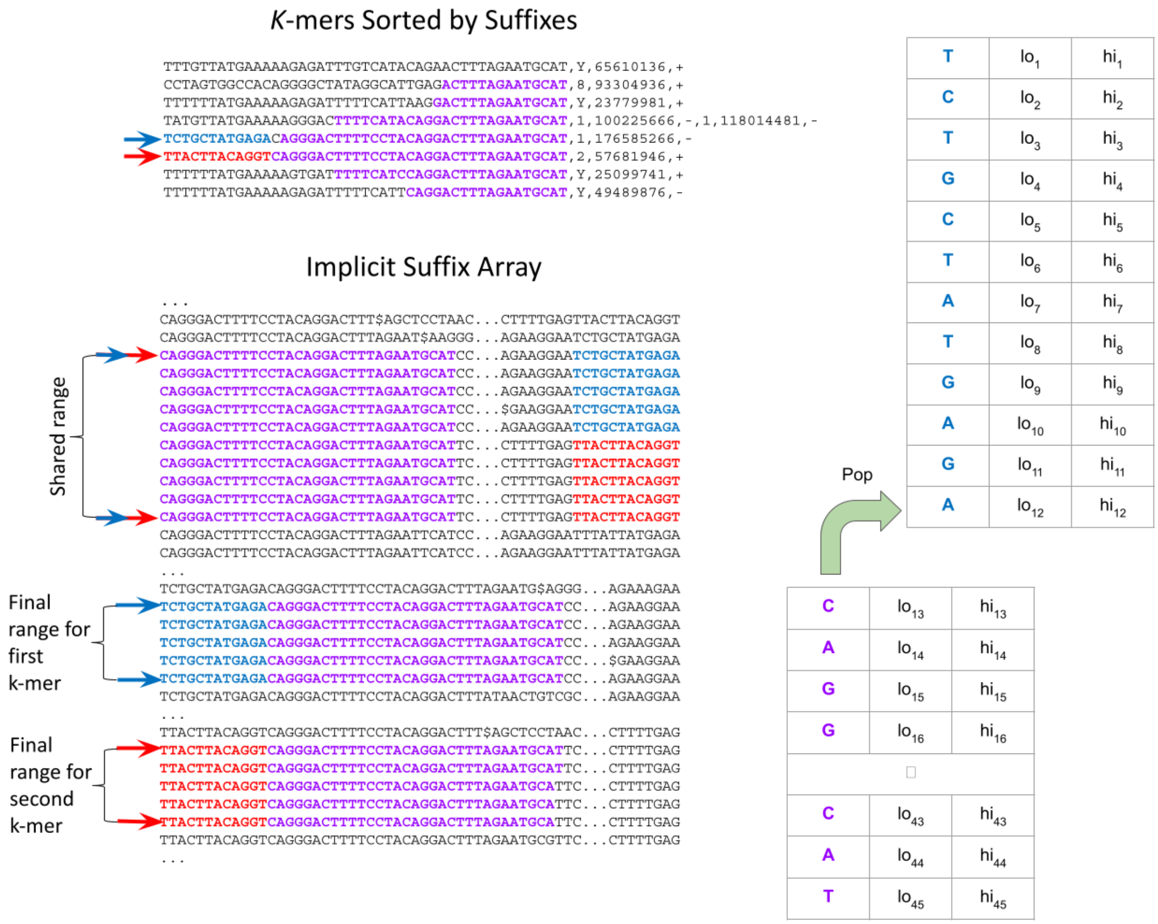

A01.04060230
E01.40602301
E01.40602302
A01.04060233
E01.40602331
E01.40602332
E01.40602333
E01.40602334
A01.04060235
E01.40602351
E01.40602352
A01.04060239
E01.40602391
E01.40602392
A01.04060250
Bridging probe
Informative probe
A
agtctttccttttattctaaagtcgctaacacaatccttcctctggccagcacacAcattccagtgtggaacagcagaagtggatatcat
agtctttccttttattctaaagtcgctaacacaatccttcctctggccagcacacGcattccagtgtggaacagcagaagtggatatcat
a
b
B
ctaaagtcgctaacacaatctctAgtgtggaacagcagaagtggaaacagcagaagtgcaccattccagtgtgCcttcctctggccagca
ctaaagtcgctaacacaatctctGgtgtggaacagcagaagtggaaacagcagaagtgcaccattccagtgtgCcttcctctggccagca
ctaaagtcgctaacacaatctctAgtgtggaacagcagaagtggaaacagcagaagtgcaccattccagtgtgTcttcctctggccagca
ctaaagtcgctaacacaatctctGgtgtggaacagcagaagtggaaacagcagaagtgcaccattccagtgtgTcttcctctggccagca
C
tctaaagtcgctaacacaatctctagtgtggaacagcagaagtggaaacagcAGCAGAAGagaagtgcaccattccagtgtgtcttcctc
tctaaagtcgctaacacaatctctagtgtggaacagcagtggaaacagcagaagtgcaccattccagtgtgtcttcctc
D
TAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATTAATTtcgcta
TAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATTAATTtcgcta

Figure 5.1: Probe Selection. Informative (red) and bridge (blue) probes are selected from parallel paths. A) In many cases, parallel edges are differentiated by simple variants which can be represented using an alternate k-mer. B) When haplotypes are distinguished by multiple variants separated by more than k bases, multiple k-mers were selected together as informative probes. C) When haplotypes are distinguished by indels, the k-mers that cover the indel are selected as informative probes. Overlaps are allowed between adjacent probes as the overall edge length is not guaranteed to be a multiple of k. D) When haplotypes are distinguished by micro-satellites, there may not exist any informative probes on the interval. Our approach begins by determining informative probes (red), and then selecting bridge probes (blue) that cover the remaining base pairs of the edge.

by a subset of haplotypes. The details of the probe selection, probe querying, and organizing the results in a database are discussed in the following sections.

5.2.1 Anchor probes

As described in the previous chapter, anchors are the unique, non-overlapping and topologically sorted k-mers that are shared by every linear genome assembly represented in the pangenome model. We select anchor probes from the linear genome assemblies. We first divide the reference assembly into a set of non-overlapping k-mers. We then select a subset k-mers that are unique in the reference assembly. We refer to this subset as anchor candidates. We next test whether these


K-mers Sorted by Suffixes
Implicit Suffix Array
Shared range
Final range for first k-mer
Final range for second k-mer
Pop

```
TTTGTTATGAAAAAGAGATTTGTCATACAGAACTTTAGAATGCAT,Y,65610136,+
CCTAGTGGCCACAGGGGCTATAGGCATTGAGACTTTAGAATGCAT,8,93304936,+
TTTTTTATGAAAAAGAGATTTTTCATTAAGGACTTTAGAATGCAT,Y,23779981,+
TATGTTATGAAAAAGGGACTTTTCATACAGGACTTTAGAATGCAT,1,100225666,-,1,118014481,-
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTAGAATGCAT,1,176585266,-
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTTAGAATGCAT,2,57681946,+
TTTTTTATGAAAAAGTGATTTTTCATCCAGGACTTTAGAATGCAT,Y,25099741,+
TTTTTTATGAAAAAGAGATTTTTCATTCAGGACTTTAGAATGCAT,Y,49489876,-
```

```
...
CAGGGACTTTTCCTACAGGACTTT$AGCTCCTAAC...CTTTTGAGTTACTTACAGGT
CAGGGACTTTTCCTACAGGACTTTAGAAT$AAGGG...AGAAGGAATCTGCTATGAGA
CAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAATCTGCTATGAGA
CAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAATCTGCTATGAGA
CAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAATCTGCTATGAGA
CAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...$GAAGGAATCTGCTATGAGA
CAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAATCTGCTATGAGA
CAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAGTTACTTACAGGT
CAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAGTTACTTACAGGT
CAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAGTTACTTACAGGT
CAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAGTTACTTACAGGT
CAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAGTTACTTACAGGT
CAGGGACTTTTCCTACAGGACTTTAGAATTCATCC...AGAAGGAATTTATTATGAGA
CAGGGACTTTTCCTACAGGACTTTAGAATTCATCC...AGAAGGAATTTATTATGAGA
...
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTAGAATG$AGGG...AGAAAGAA
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAA
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAA
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAA
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...$GAAGGAA
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTAGAATGCATCC...AGAAGGAA
TCTGCTATGAGACAGGGACTTTTCCTACAGGACTTTATAACTGTCGC...AGAAGGAA
...
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTT$AGCTCCTAAC...CTTTTGAG
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAG
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAG
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAG
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAG
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTTAGAATGCATTC...CTTTTGAG
TTACTTACAGGTCAGGGACTTTTCCTACAGGACTTTAGAATGCGTTC...CTTTTGAG
...
```

T	lo_1	hi_1
C	lo_2	hi_2
T	lo_3	hi_3
G	lo_4	hi_4
C	lo_5	hi_5
T	lo_6	hi_6
A	lo_7	hi_7
T	lo_8	hi_8
G	lo_9	hi_9
A	lo_{10}	hi_{10}
G	lo_{11}	hi_{11}
A	lo_{12}	hi_{12}

C	lo_{13}	hi_{13}
A	lo_{14}	hi_{14}
G	lo_{15}	hi_{15}
G	lo_{16}	hi_{16}
□		
C	lo_{43}	hi_{43}
A	lo_{44}	hi_{44}
T	lo_{45}	hi_{45}

Figure 5.2: The fast bulk query algorithm speeds up overall execution time by taking advantage of shared suffixes between successive queries. All k-mers are initially sorted by their suffix. Successive queries reuse the common suffix of their predecessor (shown in purple) to initialize their search in the msBWT. This sharing is managed via a stack that tracks the msBWT range of each suffix from a k-mer, The successive k-mer begins by popping off the intervals of its unshared prefix. On average this results in a significant improvement relative to independent FM-index traversals for each k-mer. In the example shown, the red k-mer leverages the shared intervals of the previous blue k-mer shown in the stack depiction on the right.

k-mers appear in, are unique, and are consistently ordered in all other assemblies. When anchor candidates are adjacent, we keep only the first and last k-mers of a continuous run, and one at random when only two are adjacent (the others are incorporated into edges). As a result, we select a set of non-overlapping anchor probes that are conserved, unique, consistently ordered in all linear genome assemblies. Sequences between anchor probes are next extracted and identical sequences are merged into a single edge. The source and destination anchors are then added to represent the graph's connectivity.

5.2.2 Selecting Diversity probes

In our anchor-based graphical genome, edges include all variants. The parallel edges that share the same source and destination anchor nodes represent the haplotype diversity in the genomic region. In this phase, we divide edges into k-mers, that cover all of their bases. The objective is to maximize the number of shared k-mers between the parallel edges in a gap, while identifying a set of probes that are exclusive to an edge. Edge probes, called *diversity probes*, include the selected k-mers that cover every non-anchor base pair in the pangenome and capture all genomic differences between parallel edges. These diversity probes can be classified into two types: the informative probes that discriminate parallel edges, and bridging k-mers that are shared by parallel edges. The informative probes usually incorporate SNPs, small indels, and are representative of a specific path (Figure 5.1).

We first select informative probes from parallel edges. The selection of informative probes involves heuristics. There are different scenarios for variant representations within an edge. In the simplest case, parallel edges can be discriminated by a single SNP or a small indel (Figure 5.1A). In such cases, the sequence variants of an edge can be represented using a single informative k-mer. There are multiple choices for selecting an informative probe to cover the variant. We preferably choose the informative k-mer with a multiple-of-k offset from the first base of the edge. Edges can possibly contain multiple sequence variants separated by more than k base pairs. In this case, multiple k-mers are used together to distinguish one edge from another. For example, as shown in Figure 5.1B, these parallel edges contain 2 variants, where the the first and the third path share the same allele at a, while the first and the second path share the same allele at b. By associating multiple informative k-mers together, we distinguish different haplotypes. A third case occurs on edges with indels. Indels may introduce one or more new k-mers for large insertions or introduce overlapping k-mers in the case of deletions. A fourth case occurs when the diversity of an edge involves tandem duplication or micro-satellites for which no informative k-mer can be selected (Figure 5.1D).

In the last three cases, we utilize a minimum set cover approach to find the minimal set of k-mers to distinguish parallel paths. We first scan through all possible k-mers on parallel edges and encode them with a set of assembly IDs that share the k-mer. We then select the minimal set of informative k-mers that discriminate all pairs of parallel paths in the gap. Finally, the bridging k-mers between informative probes are selected to cover every base pair of the gap's parallel edges. Overlapping is allowed between k-mers when the edge length is not a multiple of k. For edges with ambiguous bases (Ns), we select a probe set with a minimal number of k-mers containing Ns.

5.2.3 Fast bulk query algorithm

Once a set of probes that covers the pangenome is selected, we perform bulk queries against an msBWT that takes advantage of the shared suffixes between successive queries. Our search approach leverages the fact that many k-mers will be processed at once and attempts to maximally reuse previous FM-index accesses. It begins by sorting the k-mers via suffix, and using a stack to track the common shared suffix between successive queries, which corresponds to the shared searching ranges in the FM-index (Figure 5.2).

We first reorder the k-mer list by lexicographically sorting the k-mers by their suffix. The process of sorting by suffix can be done using any sort routine, e.g. a radix sort [141]. Queries with similar suffixes will access the same intervals of the FM-index, which allows successive queries to share parts of their computation. We keep track of the shared suffix and the range of their search intervals between consecutive queries in a stack. Before each subsequent k-mer is processed, the top of the stack that does not correspond to the two entries' common suffixes is popped off the stack and the standard FM-index traversal restarts from that point. This effectively avoids redundant searches of shared suffixes between successive queries, thereby reducing the O(k) running time required when the query was executed independently. The algorithm is implemented via a stack coupled with a depth first traversal of a shared suffix tree. The bulk query algorithm is developed based on the msbwt Python toolkit (https: //github.com/holtjma/msbwt) developed by Holt and McMillan [96].

The msBWTs used in our case were previously constructed for all lanes and pair ends of the short-read sets of DNA sequenced samples [119, 120]. Once the counts for each k-mer in a data set are determined, the ordering of k-mers is unravelled, forward and reverse complement counts are combined and the results are loaded into a database representing the pangenome.

5.2.4 Probe Database for a Pangenome Model

We construct a mapping vector to map each probe in the pangenome to a row of a k-mer count matrix (Figure 5.3). Every edge in the pangenome is mapped to a range of indices in the mapping vector, whose length is identical to the number of probes on the selected edge. The mapping vector indicates a row index of the k-mer count matrix, thus each k-mer is represented once. A single row on the k-mer count matrix can map to multiple genomic positions on the pangenome. Anchors are unique and shared by every path and are, thus, fixed rows in the count matrix. Parallel paths between an adjacent pair of anchors were represented by alternative blocks in the mapping vector. When displaying the k-mer counts on a path, a block of indices were extracted from the mapping vector and the corresponding rows of the k-mer count occurrence matrix were extracted and stacked together. The k-mer frequency shows how many reads include the specific k-mer in each sequenced sample (both forward and reverse complement). Anchors are specified by their linear coordinates on the reference path. Coordinate offsets from the proximal anchor positions are also provided for each k-mer of an edge. The probe database integrates multiple genome assemblies and sequenced samples in the represented population. It provides a straightforward way to visualize and analyze population-scale sequence data based on a pangenome model.

5.3 Results and Discussion

In this chapter, we introduced an k-mer query tool that is built for a widely used genetic reference population of laboratory mice, called the Collaborative Cross (CC) [142]. Details of the k-mer

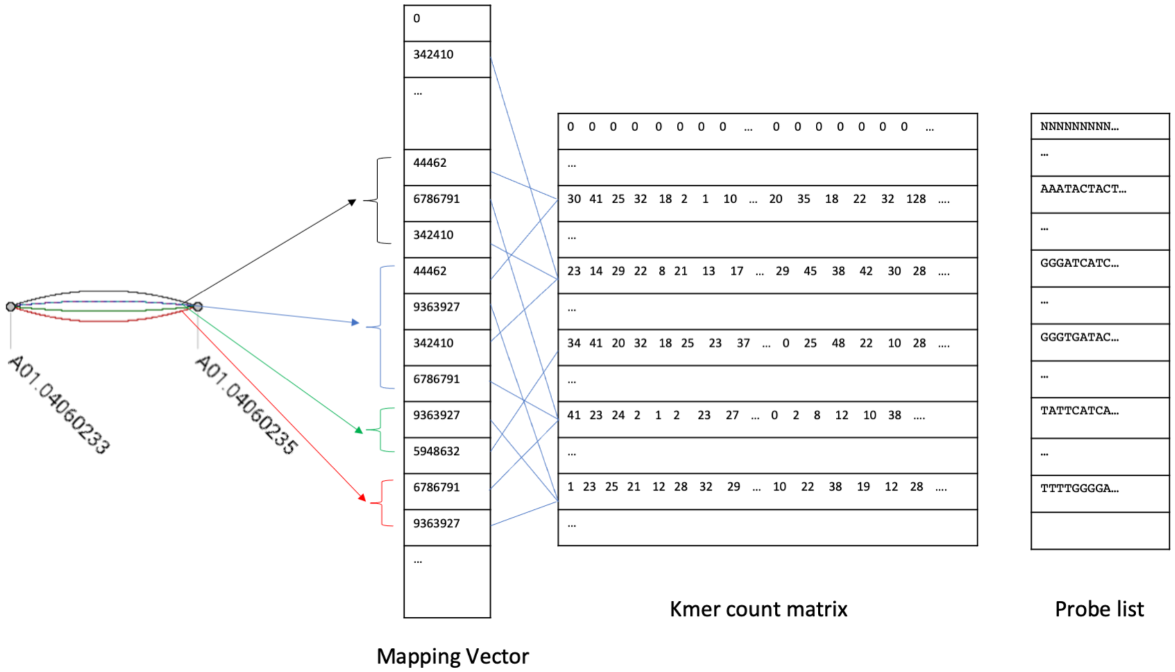


Figure 5.3: A probe database is constructed using the k-mer counts. The k-mers of edges are specified indirectly using a mapping vector. An edge specified the interval of a block in the mapping vector which, in turn, references rows in a k-mer count matrix. The mapping vector establishes a many-to-one mapping of k-mers to a matrix row. The actual k-mer sequences are kept in a list that is also addressed using the mapping vector indices.

selection, k-mer queries and database construction for the CC are described in the Method section. We present a web-based tool that can be used to explore the genomes of this population.

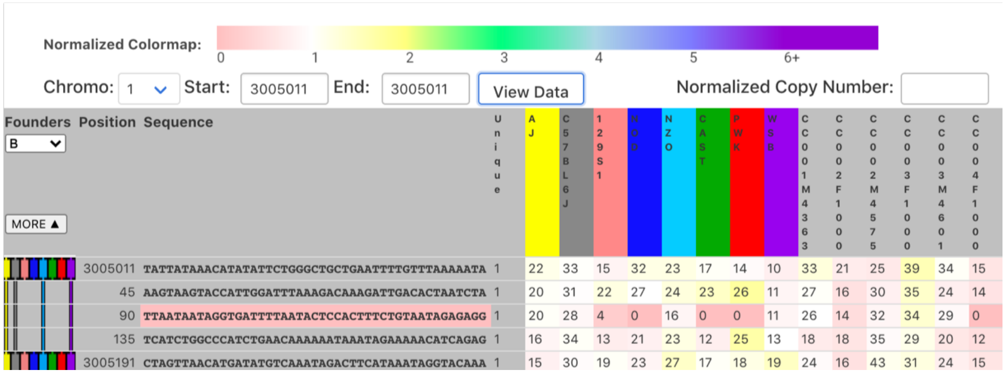
5.3.1 A Probe Database for the Collaborative Cross

As described previously, the sequence content of the Collaborative Cross Graphical Genome (CCGG) is derived from genome assemblies of the 8 founder strains of the CC population. This includes the mouse reference genome, GRCm38, and the *de novo* assemblies of the other 7 founder genomes from [6]. We combined these 8 genome assemblies into a pangenome model and selected probes to cover every base pair of the CCGG. As described in the methods selection, we collected 289,683,815 45-mer probes from the CCGG (autosomes and chromosome X). Identical probes were merged, which further compressed the 8 linear assemblies into a representative set of k-mers. We obtained a set of 148,183,468 distinct k-mers ($k = 45$), which is about 51% of the original probes size. Among these distinct k-mers, 103,593,844 appear in GRCm38 reference genome, comprising of 70% of the total distinct k-mers. 58,528,361 of them are non-overlapping 45-mers that cover the mouse reference genome. Sequence variations in the CCGG are captured by the informative probes in the probe database. There are 24,727,746 informative probes in the CCGG, which comprise of 16.69% of the distinct k-mer set.

These distinct k-mers were sorted by their suffixes. We then applied our fast bulk query method to query these probes in the short-read sequence data collected from 113 samples including the 8 CC founders and 75 CC strains with replicates. The k-mer counts were stored as an occurrence matrix with 113 columns. When new sequenced samples are released, our pipeline incorporates them by appending their k-mer counts as new columns in the occurrence matrix. A mapping vector maps the informative and bridging probes on each edge to the k-mer count occurrence matrix, which establishes the connectivity among k-mers. The parallel paths are encoded with a unique ID and assigned to a block of mapping indices pointing to the row indices of the k-mer count matrix. When examining a specific path, the k-mers on that path and the corresponding rows in the count matrix are extracted and stacked together to show the probe information on that path. The probe database provides an efficient tool for revealing the pangenome structure and the genomic origins of sequenced samples in a specific genomic region. Mapping the k-mer counts in each sample to the pangenome model, downstream analysis such as variant calling can be performed based on the k-mer count information.

5.3.2 Visualization of the CCGG Probe Database

We provide a web tool for displaying the CC probe database that includes the probes and counts in each sequenced sample. The visualization of the CC probe database is shown in Figure 5.4. Anchor sequences appear on every path in the CCGG, and are displayed in the first columna as blocks of 8 colors representing the founder strains (A/J, yellow; C57BL/6J, gray; 129S1/SvImJ, pink; NOD/ShiLtJ, dark blue; NZO/H1LtJ, light blue; CAST/EiJ, green; PWK/PhJ, red; and WSB/EiJ, purple [132]).



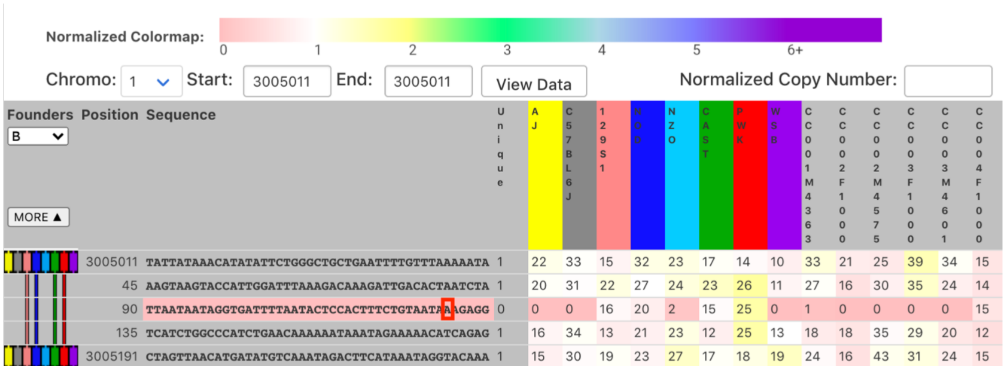
Normalized Colormap: 0 1 2 3 4 5 6+

Chromo: 1 Start: 3005011 End: 3005011 View Data Normalized Copy Number:

B MORE ▲

Founders	Position	Sequence	Unique	AJ	C57BL6J	129S1	NOD	NZO	CAST	PWK	WSB	CC001M4363	CC002F1000	CC002M4575	CC003F1000	CC003M4601	CC004F1000
	3005011	TATTATAAACATATATTCTGGGCTGCTGAATTTTGTTTAAAAATA	1	22	33	15	32	23	17	14	10	33	21	25	39	34	15
	45	AAGTAAGTACCATTGGATTTAAAGACAAAGATTGACACTAATCTA	1	20	31	22	27	24	23	26	11	27	16	30	35	24	14
	90	TTAATAATAGGTGATTTTAATACTCCACTTTCTGTAATAGAGAGG	1	20	28	4	0	16	0	0	11	26	14	32	34	29	0
	135	TCATCTGGCCCATCTGAACAAAAAATAAATAGAAAAACATCAGAG	1	16	34	13	21	23	12	25	13	18	18	35	29	20	12
	3005191	CTAGTTAACATGATATGTCAAATAGACTTCATAAATAGGTACAAA	1	15	30	19	23	27	17	18	19	24	16	43	31	24	15

(a) Path that shared by ABEH.



Normalized Colormap: 0 1 2 3 4 5 6+

Chromo: 1 Start: 3005011 End: 3005011 View Data Normalized Copy Number:

B MORE ▲

Founders	Position	Sequence	Unique	AJ	C57BL6J	129S1	NOD	NZO	CAST	PWK	WSB	CC001M4363	CC002F1000	CC002M4575	CC003F1000	CC003M4601	CC004F1000
	3005011	TATTATAAACATATATTCTGGGCTGCTGAATTTTGTTTAAAAATA	1	22	33	15	32	23	17	14	10	33	21	25	39	34	15
	45	AAGTAAGTACCATTGGATTTAAAGACAAAGATTGACACTAATCTA	1	20	31	22	27	24	23	26	11	27	16	30	35	24	14
	90	TTAATAATAGGTGATTTTAATACTCCACTTTCTGTAATAAAGAGG	0	0	0	16	20	2	15	25	0	1	0	0	0	0	15
	135	TCATCTGGCCCATCTGAACAAAAAATAAATAGAAAAACATCAGAG	1	16	34	13	21	23	12	25	13	18	18	35	29	20	12
	3005191	CTAGTTAACATGATATGTCAAATAGACTTCATAAATAGGTACAAA	1	15	30	19	23	27	17	18	19	24	16	43	31	24	15

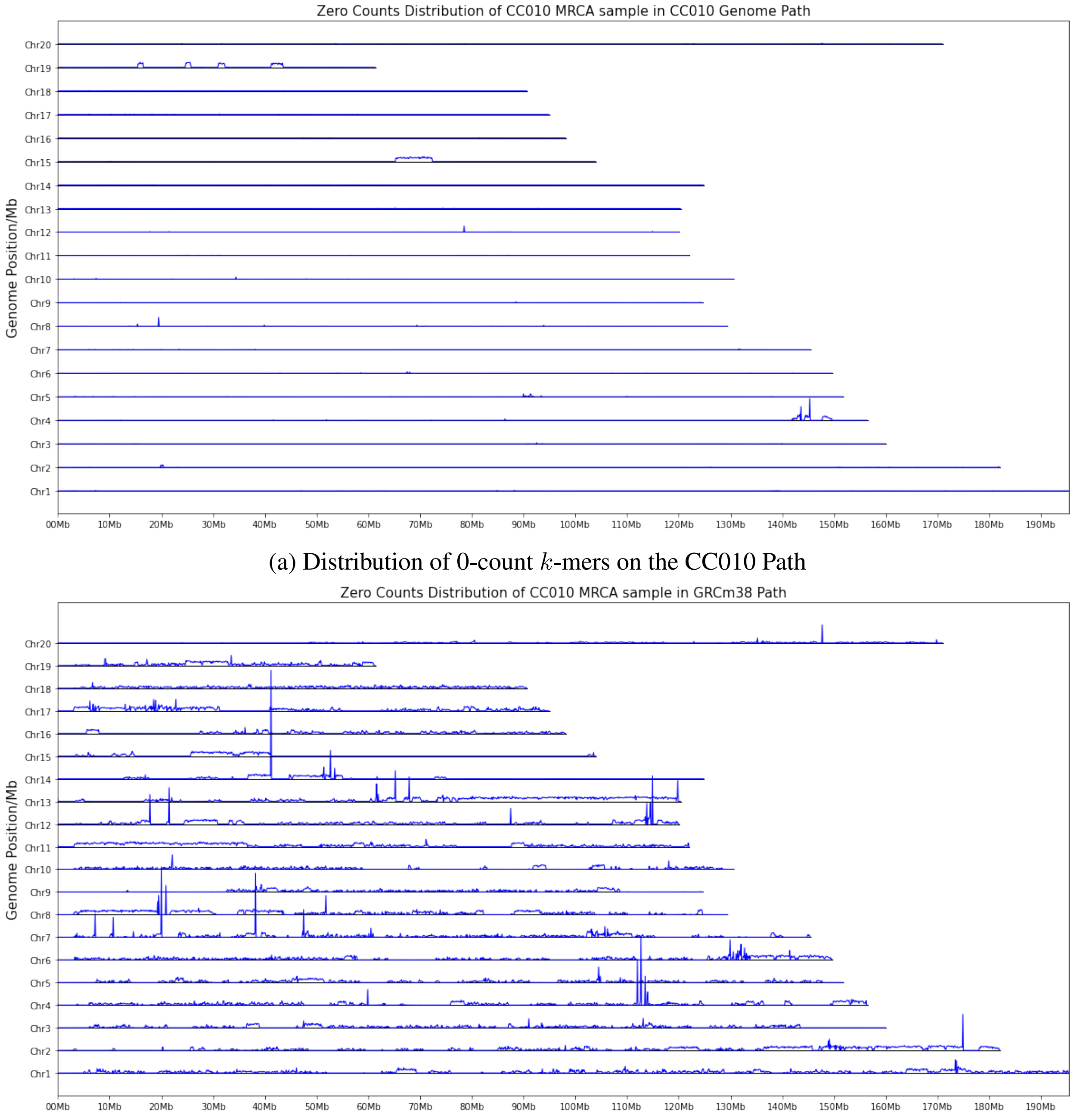
(b) Path that shared by CDFG.

Figure 5.4: Visualization of the CCGG Probe Database. A web-based tool for visualizing the CCGG database has been developed. Two parallel paths between a common pair of anchors are shown. These two paths are distinguished by a SNP (highlighted by red). Given the genomic position on the GRCm38 reference genome, the paths information and probe counts can be retrieved. The "Founders" column indicates which founder paths these probes lie on. The colored series of lines between anchors indicate the sharing of haplotypes between founders. The "Position" column provides the genomic position of each probe. Anchors (depicted as colored rectangles in the leftmost column) are identified by their position in the mouse reference genome, GRCm38. Edge probes are referenced by their relative offset from their proximal anchor. The user can cycle through available edges by clicking on the gaps between anchors in the "Founders" column. This brings in an alternate k-mer set and counts. The probe sequences are recorded in the "Sequence" column. The "Unique" column records how many times the k-mer probe occurs in the GRCm38 mouse reference genome. The following 113 columns record the read counts of each k-mer in the 113 short-read sequenced samples. The cell background of a count represents the normalized k-mer counts. The proximal and distal genomic gaps can be accessed by clicking the "More" button in this interface.

The coordinates of anchors in the mouse reference genome, GRCm38, are shown in the "Position" column. Paths between anchor pairs are shared by a subset of founder strains and are depicted by a set of lines connecting anchors. The standard colors are again used to represent founder strains sharing a common edge. The genomic positions of edge probes are given as offsets relative to their source (proximal) anchor nodes and displayed in the “Position” column. The sequence contents of anchor and edge probes are shown in the “Sequence” column, where the informative probes are highlighted in red. The “Unique” column indicates how many times the probe sequence appears in the GRCm38 reference genome (both forward and reverse complement). The following columns display the k-mer counts of each probe in the 113 sequenced samples. The k-mer counts are normalized by the mode of k-mer frequency in each sample. The cell’s background color indicates the normalized count. Hovering over a cell, the value of the normalized count is displayed in the “Normalized Copy Number” text box on the top right. When clicking on the interval between a pair of anchors displayed in the “Founders” column, the CCGG viewer will cycle through all the parallel paths and display the corresponding count matrix for the probes on that path. Alternatively, one can click the drop down menu beneath the “Founders” to select a specific founder’s path. The “MORE” buttons, load additional data up to the next anchor in both the proximal and distal directions. The web tool can be accessed at this website (http://devel.unc.edu/GraphicalGenome).

5.3.3 Validating the Sequence Content of the Graph

The CCGG probe database serves as a visualization tool to examine the genomic differences in a population. The sequence subgraph within each genomic region can be extracted from the database and the k-mer counts can be analyzed across the whole population. We applied the k-mer count matrix to validate the sequence content of the CCGG. Instead of using a single sequenced sample, our probe database integrates multiple samples descended from the same founder strain to validate the sequence graph, which reduces the technical noise or mutations introduced by a single sequenced sample. The unsupported k-mers potential result from assembly issues, mislabeling of founder paths, the segregating features in the CC strain, or the fixed CC private mutations introduced during the breeding process in a single CC strain. The raw counts and normalized counts in each sample can be utilized to refine recombination boundaries of a CC strain or to identify mutations, copy-number variations, and heterozygous regions.



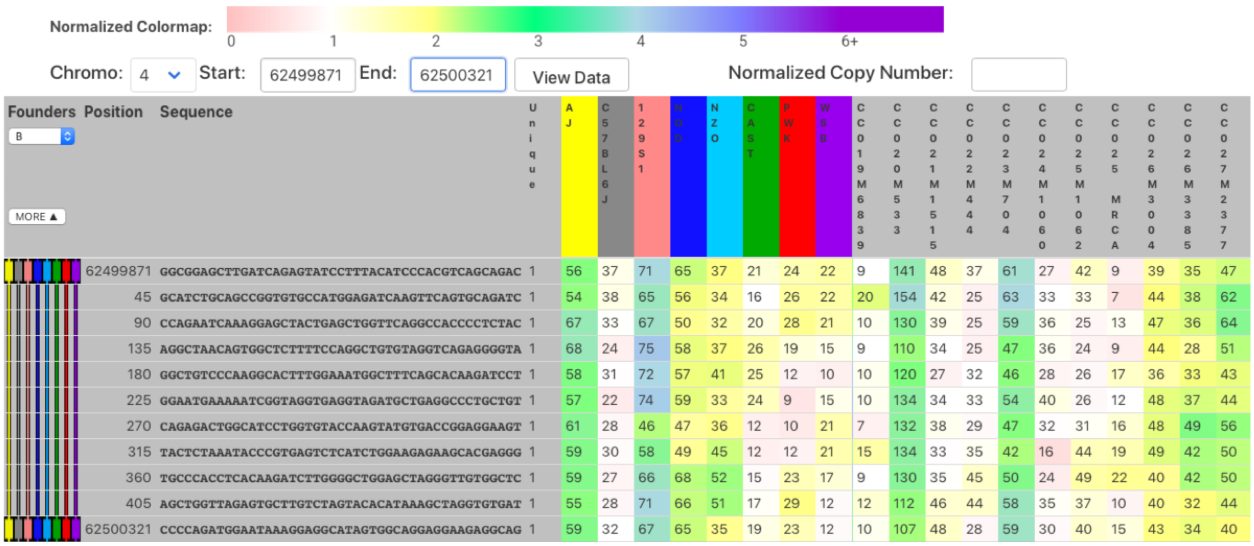
(a) Distribution of 0-count k-mers on the CC010 Path

(b) Distribution of 0-count k-mers on the B Path (GRCm38)

Figure 5.5: Distribution of Unsupported K-mers on the CC-specific Path Versus the Reference Path Using CC MRCA Data. We query the CC probes in the CC010 MRCA sequenced data. Traversing the CC010 path and the reference path, we plot the distribution of unsupported k-mers with 0 counts. The k-mers with Ns are excluded. The k-mer position are normalized by its proximal anchor position on the GRCm38 coordinates. The density of k-mers are normalized by the number of k-mers for every 100Kb non-overlapping sliding window. The x-axis represents the genomic position on the GRCm38. The y-axis plot the autosomes and chromosome X and the 0-count kmer density within each bin.

In the previous work, the haplotype intervals of CC strains are inferred from a Hidden Markov Model (HMM) based on the sequenced CC male samples [119]. We assigned the CC path in the pangenome model based on the haplotype intervals released in [119]. We then traversed the graph to identify the unsupported k-mers with 0 counts in the sequenced MRCA data. We found that the CC-specific path significantly reduce the amount of k-mers that are not supported by the MRCA sequence data. Taking CC010 as an example, we count the number of unsupported probes in the CC010 MRCA data on both the CC010 path and the reference path (GRCm38), excluding all the k-mers with Ns. There are 293,665 0-counts on the CC010 path, yet 5,647,803 0-counts on the B6 path. We plot the distribution of these 0-count k-mers on each path. As shown in Figure 5.5, the amount of unsupported k-mers in the CC010 MRCA data are significantly reduced along the CC010 path. The result suggests that the CC010 genome extracted from the CCGG capture more features of the CC MRCA sequenced samples than the standard reference genome.

The unsupported sequences on the CC genome could result from the mislabeling of founder assemblies or the mutations in the CC strains that are not incorporated in the founder strains. A few clusters of 0-count k-mers are reported in the CC010 path (Figure 5.5 a). These clusters of unsupported k-mers indicate the genomic regions that should be refined in the CC genome. The mislabeling of founder assemblies could result from the ambiguity near the recombination boundary or the errors in the HMM model prediction introduced by low density of informative SNPs. The k-mer count matrix provides an effective way to identify the optimal path to describe the CC sequence data. The CC-specific paths can be refined by relabeling the sequenced sample to an optimal path with minimal number of unsupported sequences (discussed in Chapter 7). In addition, there are CC specific features that are not incorporated in the founder assemblies. These CC-specific genomic features can be introduced to the graph as a set of new edges. The technical noise introduced by individual sample can be reduced by jointly considering replicates or CC samples descended from the same ancestors. CC private variants fixed in the strain can be identified by searching for k-mers that are absent in the replicates of a single strain, but are supported by every other CC strains decended from the same founder. *De novo* founder variants that are segregating in the CC strains descended from the same founder can be identified by the variants that are shared by only a subset of corresponding CC strains. Assembly errors can also be identified where none of the CC samples as well as the founder sample supports the corresponding k-mer in the assembly. In addition, Copy Number Variants (CNVs) can be identified according to the fold change of the k-mer frequency on a continuous run. For example, the k-mer on Chr4:62.50Mb appear only one time in the mouse reference genome, GRCm38, yet the normalized copy number of these k-mer probes is more than 2 in A/J, 129S1/SvImJ, NOD/ShiLtJ, NZO/H1LtJ and the CC samples descended from these founder strains (Figure 5.6). It suggests a CNV occurred in these founder strains in this region.



Chromo: 4 Start: 62499871 End: 62500321

Normalized Copy Number:

Founders: B

Founders	Position	Sequence	Unique	AJ	C57BL6J	129S1	NOD	NZO	CAST	PWK	WSB	CC019M6839	CC020M533	CC021M1515	CC022M444	CC023M704	CC024M1060	CC025M1062	CC025 MRCA	CC026M3004	CC026M3385	CC027M2377
	62499871	GGCGGAGCTTGATCAGAGTATCCTTTACATCCCACGTCAGCAGAC	1	56	37	71	65	37	21	24	22	9	141	48	37	61	27	42	9	39	35	47
	45	GCATCTGCAGCCGGTGTGCCATGGAGATCAAGTTCAGTGCAGATC	1	54	38	65	56	34	16	26	22	20	154	42	25	63	33	33	7	44	38	62
	90	CCAGAATCAAAGGAGCTACTGAGCTGGTTCAGGCCACCCCTCTAC	1	67	33	67	50	32	20	28	21	10	130	39	25	59	36	25	13	47	36	64
	135	AGGCTAACAGTGGCTCTTTTCCAGGCTGTGTAGGTCAGAGGGGTA	1	68	24	75	58	37	26	19	15	9	110	34	25	47	36	24	9	44	28	51
	180	GGCTGTCCCAAGGCACTTTGGAAATGGCTTTCAGCACAAGATCCT	1	58	31	72	57	41	25	12	10	10	120	27	32	46	28	26	17	36	33	43
	225	GGAATGAAAAATCGGTAGGTGAGGTAGATGCTGAGGCCCTGCTGT	1	57	22	74	59	33	24	9	15	10	134	34	33	54	40	26	12	48	37	44
	270	CAGAGACTGGCATCCTGGTGTACCAAGTATGTGACCGGAGGAAGT	1	61	28	46	47	36	12	10	21	7	132	38	29	47	32	31	16	48	49	56
	315	TACTCTAAATACCCGTGAGTCTCATCTGGAAGAGAAGCACGAGGG	1	59	30	58	49	45	12	12	21	15	134	33	35	42	16	44	19	49	42	50
	360	TGCCCACCTCACAAGATCTTGGGGCTGGAGCTAGGGTTGTGGCTC	1	59	27	66	68	52	15	23	17	9	130	35	45	50	24	49	22	40	42	50
	405	AGCTGGTTAGAGTGCTTGTCTAGTACACATAAAGCTAGGTGTGAT	1	55	28	71	66	51	17	29	12	12	112	46	44	58	35	37	10	40	32	44
	62500321	CCCCAGATGGAATAAAGGAGGCATAGTGGCAGGAGGAAGAGGCAG	1	59	32	67	65	35	19	23	12	10	107	48	28	59	30	40	15	43	34	40

Figure 5.6: Potential Duplication Regions identified in the probe database. The probe counts on the reference path around Chr4:62.50Mb are shown above. This path is shared by all founder assemblies. As shown in the "Unique" column, most of the probes appear only once in the GRCm38 mouse reference genome. This is supported by the C57BL/6J sample, whose normalized count number is about 1. However, the probe counts suggest higher coverage over this region in A/J,129S1/SvImJ,NOD/ShiLtJ,NZO/H1LtJ and CC samples descended from these founder strains. This suggest that a potential shared duplication inherited from a founder.

5.3.4 Adding Newly Released CC Sequenced Samples to the Graph

With more sequenced CC strain available, our graphical genome maintains the capacity to incorporate more CC data and their recombinant genomes to the graph. When a sequenced CC sample is released, we can construct msBWT for the raw read data and query the CC probes in the msBWT of the sequenced sample. The new sample can be appended to the kmer count occurrence matrix as a new column. One can further leverage the founder informative probes and the SNP intensity to infer the haplotype intervals by using the HMM model [119]. Once the haplotype intervals are obtained, the new strain ID can be overlaid to the founder path. Alternatively, we can infer the optimal path between each anchor pair, which describe the sequenced sample with minimal unsupported probes. Once the optimal paths between anchor pairs are defined, the haplotypes can be reconstructed from forward and backward graph traversal.

Private CC paths can be added to the graph if evidence suggest the CC-specific features are fixed in a strain. For regions with such variations, we can take the proximal and distal probes as seed to perform local assembly around those absent kmers. The alternative sequences can then be inserted as private edges to the graph. The other advantage of this method is that it can analyze population-scale samples cross many individuals. For the CC panel, there are multiple strains descended from the same founder strains in a specific genomic region. Using these prior knowledge, one can reduce the sequence noise and robustly interpret the genetic data conducted in a CC strain.

5.4 Conclusion

Applying the k-mer counting tool based on a pangenome model, where k-mer connectivity is tracked along paths, we have demonstrated an alignment-free method for population-scale sequence analysis. We maintain a k-mer based pangenome model as a list of representative k-mers sorted by their suffixes. These k-mers are selected to cover every base pair in the pangenome model and to capture sequence variations in the population. We have developed a method to speedup bulk queries against an msBWT using a preprocess that sorts a list of k-mers by their suffix and an algorithm to reuse shared searching ranges from previous queries. We applied our methods to build a CC probe database of 113 sequenced samples collected from the CC population and its founder strains. We provide a web-based tool (http://devel.unc.edu/GraphicalGenome) to visualize genomic differences in the CC population. The probe database serves as a resource and a used-friendly interface for the research community using the CC panel to understand and interpret the CC genomic data using multiple genome assemblies.

CHAPTER 6: GRAPH-BASED COORDINATE SYSTEM

6.1 Introduction

The number of species, whose genome has been sequenced, has been steadily increasing since the 1990s [143]. These initially sequenced and assembled genomes have been taken as *reference genomes* for the common genetic and genomic analysis. Reference genomes play an essential role by providing a coordinate framework for referring to and annotating genomic features. The most prevalent representation of reference genomes is a set of single-sequence assembled contigs, which are referred to as linear genomes in the following discussion. Linear genomes provide a straightforward way for denoting genomic locations by specifying a contig name and indices, such as chr1:10Mb, representing the base position 10 Mb from the beginning of chromosome 1. In the linear reference genomes, genes or exons are represented using indices of one or more genomic intervals per feature. This coordinate framework is the foundation of many bioinformatics pipelines and has been applied in multiple file formats used for functional annotations or to represent alignment results, including GFF3 (Generic Feature Format Version 3), SAM [14], BED [15], VCF [16]. Despite their critical role in our understanding of the genomic organizations and features representation, linear reference genomes come with limitations. Linear reference genomes fail to incorporate sequence variation (either single nucleotide polymorphisms, SNPs, or more complex genomic rearrangements) within a species [4, 144]. Reliance on a single-sequence reference assembly leads to issues such as the poor characterization of highly divergent genomic regions [19] and regions with non-reference sequences [3]. Meanwhile, since the genomic features are reported with respect to the reference coordinates, adapting all the annotation and biological discoveries to an updated reference assembly with coordinate changes is time consuming and a heavy burden to the community.

With the advent of improved sequencing technologies, increasing number of intra-specific genome assemblies have been constructed with a high quality. The rapid growth of genomic sequence data has driven a paradigm shift from the linear reference genome towards the pangenome models that integrate multiple assemblies into a single representation [4, 5]. Graphs are commonly used in pangenome models to merge and partition sequences [4]. However, graph-based pangenome models introduce complications when referring to base positions or annotating features [4, 5, 13]. Since there can be multiple sequence combinations within a single interval, referencing genomic features by a single pair of start and end coordinates, such as chr1:10-12Mb, can be ambiguous in a graph-based reference genome. A graph-based coordinate system should satisfy a set of properties, including: *monotonicity*, i.e. the coordinates of successive bases along a path should be incremental, *vertical/horizontal spatiality*, i.e. nearby bases on the graph (both vertically and horizontally) should have similar coordinates, and *backward-compatibility*, i.e. the graph coordinates should be robust to assembly update and graph topology modification [4, 5, 13]. Several graph-based coordinate frameworks have been proposed, yet they struggle with fulfilling the above-described requirements, and some are computationally intensive and difficult to interpret [13, 61, 76]. For example, *Rand et al* proposed an offset-based coordinate system and introduced methods for representing intervals in graph-based genome models [13]. However, this graph-based coordinate framework lacks a consistent naming convention to illustrate the horizontal and vertical ordering between genomic contigs and is also sensitive to graph topology modification, i.e. partitioning or reconstructing sequences from the graph would result in different coordinates for a single base. *Gärtner, Fabian, et al* proposed a heuristic algorithm to construct a common coordinate system for a supergenome based on Multiple Sequence Alignment (MSA) [61]. It is computationally expensive to apply such an MSA-based coordinate system for large eukaryotic genomes. Another common graphical genome implementation is to insert well-known variants in the VCF files to the reference assembly and iteratively adding new sequences through graph alignment [80, 77, 76]. All the coordinates preserved in this framework are reported with respect to a stable linear reference coordinate framework. However, large structural variants and other complex genome rearrangement are hard to represent in this framework. Reference-specific variants are especially difficult to detect. Computationally, the complexity of graph traversal grows exponentially with the new variants added in without built-in haplotype information. In general, a graph coordinate framework requires a consistent sequence partitioning and naming convention to denote the sequential and vertical ordering of the contigs presented in a graph, yet there is no well-developed standard available. All together, these issues have hindered the application of pangenome models to conventional bioinformatic tools, leaving researchers unable to move forwards to utilize pangenome models in the extensive genomic studies.

In this Chapter, we propose a novel graph-based coordinate framework involving common 'anchor' sequences that are shared amongst assemblies incorporated in the pangenome. The anchor-based coordinate framework satisfies the major requirements of a graph-based pangenome coordinate framework yet has not been satisfied by available proposals. When a new genome build is released, our approach partition the large genome assemblies by anchor sequences, identifying and localizing the updated sequences to specific genomic intervals seperated by anchor pairs. Inserting the updated sequences as new edges to the graph, the anchor-based coordinate framework robustly preserves genomic annotations that the sequence update is of minimal or no impact on other genomic regions. Our approach also enables local corrections of genome assemblies. Tracking the mapping positions of anchor candidate sequences in a new assembly, our approach identifies distinct genome organizations relative to the standard reference assembly, which implies potential assembly errors or structural variants (SVs) in the strains. Furthermore, our approach provides an easily accessible linear coordinate translation approach for when end users wish to compare genomic features between members of the pangenome.

6.2 Method

Anchor nodes are defined as the unique and topologically sorted k-mers common to every linear genome assembly in the graph. Anchors partition multiple linear genomes consistently and are separated by one or more *edges* (sequences shared by a subset of linear assemblies). Anchor nodes are annotated with their position on each assembly. Relative to an anchor, any base on the graph can be specified as an offset with a path identifier. Genomic features are referenced relative to the specified anchor sequence with cumulative offsets. We further develop a coordinate mapping method to translate coordinates between multiple linear genomes. Coordinate mapping leverages the sequence sharing information naturally implied in the graph and pairwise alignments between parallel paths. The correspondence of homologous sequences (vertical spatiality) is represented by mapping of parallel paths according to these alignments. Referring to graph bases relative to special "SOURCE" nodes provides graph-coordinates that are compatible with traditional linear genome coordinates. The coordinate mapping between linear genomes in our pangenome representation establishes an efficient framework for multi-genome comparison. The anchor-based graphical genome also provides an efficient framework for assembly update with backward-compatibility.

6.2.1 Graph Coordinate System

Our anchor-based coordinate system is composed of three components: 1) an anchor name; 2) an offset from the start of the anchor sequence; and 3) a path identifier. This formulation enables unambiguous specification of any genomic base. The coordinate of a base is not, however, unique as it can be specified as an offset and path from any proximal anchor. The preferred or canonical coordinate of a base is generally specified from the closest proximal anchor, but exceptions to this rule are allowed for specifying bases of intervals that extend over multiple gaps. As shown in Figure 3.5, a genomic interval, Chr13: 63,513,561 – 63,513,815 bp on a reference linear genome, corresponds to completely different indices in all of the constitute genomes. The same interval can be represented by using the anchor-coordinate framework by referring to relative to the proximal anchor sequences, "A13.01411408:200:B-A13.01411417:49:B". Anchor coordinates of an interval can always be translated to refer to a common (the most proximal) anchor, thus allowing for a more compact interval specification and convenient distance calculation. Using a common anchor basis simplifies sequence comparison on a graph, since offsets can be used to determine interval sizes and calculate distances. In the extreme, the offsets from a contig's SOURCE node are identical to

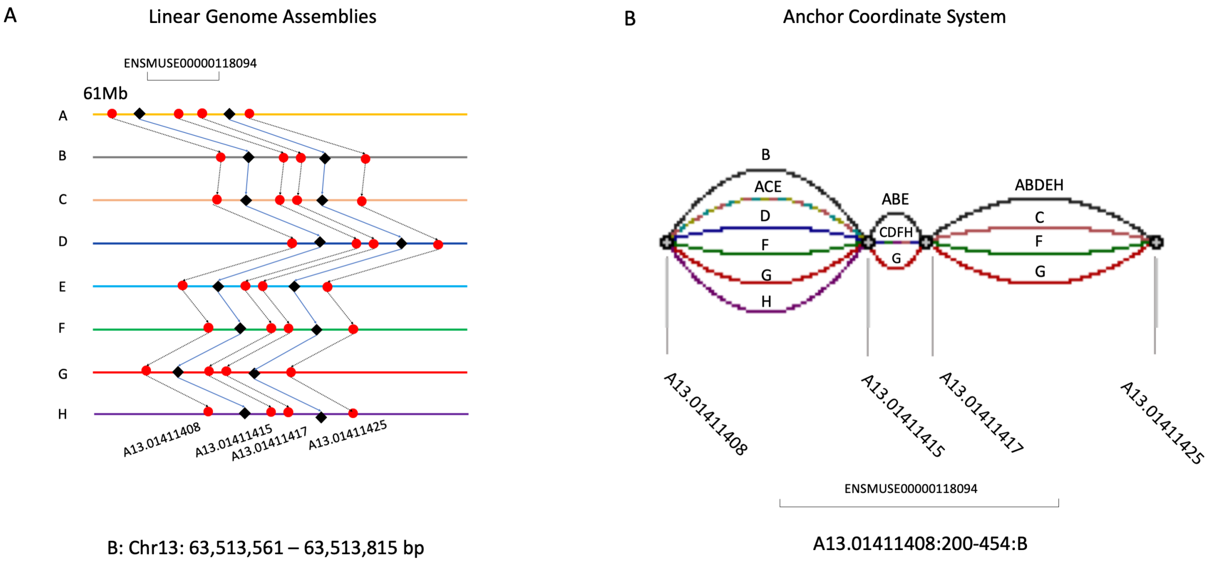


Figure 6.1: Anchor Coordinate Framework. A) Linear genomes adopt different coordinate system where homologous bases correspond to completely different indices. It makes it hard to map base positions or calculate feature distances between different linear genomes. **A)** illustrates a schematic plot of the homologous base mapping between multiple linear genomes. The red dots represent anchor nodes. The black diamond represent the start and the end of an exon ENSMUSE00000118094. The coordinate mapping among the linear genomes are illustrated by dotted lines. B) In our anchor-based coordinate system, a base position on the graphical genome can be referred to the unique occurrence position of its proximal anchor sequence with an offset and the path identifier. Mapping the graph coordinates to alternative paths, the homologous sequences can be identified and compared easily. This graph coordinates can be generalized to refer to any given anchor sequence with cumulative offsets, which allows for comparing base positions far apart from each other. When traversing back to the "SOURCE" node, the cumulative offset of the graph coordinates are identical to linear coordinates of the represented genome.

their conventional linear coordinates. Since anchor nodes are annotated with their linear genomic position in each genome included in the graph, the mapping of anchor-based offsets to their original linear coordinates is simplified. Anchor-based graph coordinates establish a convenient platform for multi-genome comparison, and they tend to be more stable when genomes are updated. It should be noted that the graphical genome compacts common sub-sequences from multiple linear genomes into shared edges. Thus, an edge can be on multiple paths, and a list of paths sharing the edge is given by the "strain" attribute of the edge. Where multiple genomes share either an anchor node or edge, a common base would likely map to distinct linear coordinates in the original genome, but that same base would have an identical canonical anchor-based graph coordinate in every genome that includes it.

6.2.2 Coordinate Mapping Method

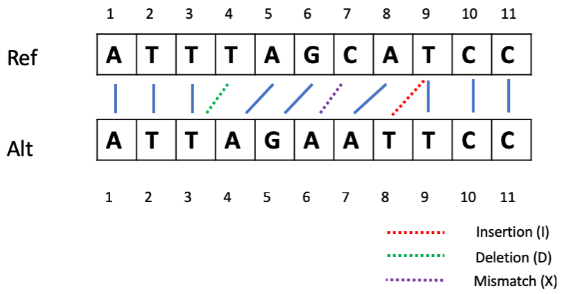


Figure 6.2: Alignment-based Position Mapping between Reference and Alternative Genomes. The pairwise-alignment results between the reference sequences and the parallel alternative were recorded in the “variants” annotation of each edge. The cigar string provides instructions to map one sequence to another by a series operations, including substitutions (X), insertions (I) or deletions (D). Scanning the expanded cigar string, the accumulated number of shifted bases pointed one position to its counterpart on the alternative path.

An advantage of our anchor-based graph coordinates is that it supports both mapping and establishing the correspondence of base positions between the linear genomes represented in the graph. It leverages the anchor node annotations and the pair-wise alignments stored in “variants” attributes of edges to provide unambiguous mappings of homologous base positions between genomes. Where genomes share an edge, it is straightforward to map base coordinates by adding a common offset to a genome coordinate annotated in an adjacent anchor node. In the case where genomes lie on a parallel path, we apply the pairwise alignment recorded in the “variants” attribute of edges to establish correspondences between base positions of the two genomes. Identical bases between two sequences are established by the matching characters (=) in the “variant” cigar string, while substitutions (X), insertions (I) or deletions (D) indicate genomic variants. By scanning the expanded cigar and summing up the number of shifted bases affected by insertions (I) or deletions

(D), one position on a path can be mapped to an alternative path. This can lead to cases where multiple bases have the same coordinate due to an insertion, or no coordinate in the case of a deletion, but these ambiguities are easily detected in the cigar string and can be reported if necessary. To provide mappings between non-reference genomes, we first map the base position to the reference path, and then map the corresponding reference coordinate to the destination path.

6.3 Result

6.3.1 Mouse Reference Pangenome Model

We demonstrate the utility of an anchor-based graphical pangenome using the genomes from a common model organism, the laboratory mouse. We constructed the mouse pangenome by integrating the genome assemblies of the 9 inbred mouse strains, A/J, C57BL/6J, 129S1/SvImJ, NOD/ShiLtJ, NZO/H1LtJ, CAST/EiJ, PWK/PhJ, WSB/EiJ and DBA/2J. These assemblies incorporate the mouse reference genome (GRCm38 derived from C57BL/6J), as well as the founder strains' genomes of 2 widely used recombinant-inbred mouse population: the Collaborative Cross (CC) [34] and the BxD [124] populations. The sequence content of the mouse pangenome includes the mouse reference (GRCm38) and the de novo assemblies of the other 8 inbred strains, which were released in [6]. Across the mouse genome (roughly 2.9 billion base pairs), we identified 15,603,130 anchor candidates (702,140,850 bp in our anchor selection, or roughly 25% of the GRCm38 mouse reference genome). These anchor candidates are supported by multiple reads (>3) in the CC and DBA/2J sequenced samples. As such, recombination of these 9 inbred strains' genomes can potentially represent of a larger cohort for approximately 160 mouse strains. We further tested whether the linear assemblies incorporated in the pangenome capture those conserved anchor sequences. We eliminated 3.2 million anchor candidate sequences as they are either absent, duplicated, or inconsistently ordered in at least of one of the alternative genome assemblies (Table 3.1). The anchors were also annotated with their linear coordinates on the 9 genome assemblies. We further annotated multiple genomic features to the reference path of the graph (GRCm38) to establish a comprehensive pangenome model. Across the 9 inbred strains we used here, the mouse pangenome is comprised of 9,098,352 anchor sequences and 31,152,388 edges, with 11,715,146,916 total base pairs (48.9% of the combined sizes of the 9 constituent genomes). The sequence similarities and differences between the reference genome and each of the alternative assemblies are shown in Table 6.1. The mouse pangenome helps to maintain and manage multiple genomic resources in a unified framework, which also benefit for visualizing and comparing the sequences among multiple genomic assemblies.

Table 6.1: Sequence Similarity between Alternative Genome Assemblies and the GRCm38 Reference Genome. Pairwise alignment was performed between the reference sequence and alternative sequences within each gap. The number of matching ("=" of cigar string) in the alignment results were calculated, characterizing the sequence similarity between each founder assembly and the reference genome.

Genomes	Identical Bases	Total Length	Percentage
AJ	2,248,849,993	2,593,672,261	86.71%
129S1	2,237,280,694	2,695,557,728	83.00%
NOD	2,222,081,545	2,943,734,294	75.49%
NZO	2,243,505,110	2,664,379,183	84.20%
CAST	2,015,260,461	2,635,057,786	76.48%
PWK	2,017,650,901	2,533,792,055	79.63%
WSB	2,208,402,950	2,671,831,734	82.66%
DBA/2J	2,247,253,162	2,578,793,109	87.14%

6.3.2 Graph Coordinate Framework

Many genomic features are best described in terms of intervals. Interval specifications from linear genomes are typically difficult to generalize to graphs. As described in Chapter 3, genomic features are overlaid on anchors and edges of the reference path of the graph. These features include genes, exons and repeat masker intervals. We annotate the relative position between the genomic features and the graph entity (anchor or edge) sequence using run-length encoded soft-clipping primitives. "M" represents base pair overlapping with the interval and , "S" represents base pair outside of the annotated interval. As an example in Figure 3.5 the exon ENSMUSE00000118094 (chromosome 13, 63,513,561 - 63,513,815 bp) of gene *Ptch1* overlaps 3 edges and 2 anchors of the graphical genome. The “gene”, “exon” and “repeatclass” annotations of graph entities were recorded in separate lists of name-strand-cigar triples. By traversing the matches, “M”, in the cigar strings along a path in the graph, the sequence content of every annotated biological feature can be easily extracted. Maintaining the genomic annotations along with the sequence content simplifies the updates of annotations. Since anchors are annotated with their linear coordinates in every genome assembly, both the graph-based and linear coordinates of a annotated interval can be extracted from the graph through simple calculations. When a genome assembly is updated, the annotations are adjusted only around modified regions. Meanwhile, most annotations on the graphical genome are unaffected. Thus the anchor-based coordinate system provided significantly improved backward-compatibility over traditional linear genomes.

6.3.3 Homologous Sequence Extraction and Comparison

Genomes in an intraspecific population are usually highly similar. However, individual genomes vary from one to another due to the accumulation of segregating mutations, ranging from single nucleotide polymorphisms, or SNPs, to larger and more complex genomic rearrangements. Importantly, while in any single region, only a few variants will segregate between samples, and many genomics tools are built to account for these variants, across a genome, the accumulated weight of these changes can confound common mapping of feature coordinates between samples. As described in Chapter 3, we annotated the well-documented GRCm38 features, including genes, exons and repeat regions (masked via RepeatMasker in standard bioinformatic pipelines) on the reference path (“B” path) of the graphical genome. We also provide a coordinate mapping tool (details described in the Methods section) for transforming starting and ending offsets from the reference path to an alternative path. This allows homologous sequences of a functional interval to be extracted from the graph. To demonstrate the accuracy of the anchor-based coordinate mapping method, we mapped the GRCm38 gene intervals to every alternative founder strain genomes including the A/J, 129S1/SvImJ, NOD/ShiLtJ, NZO/H1LtJ, CAST/EiJ, PWK/PhJ, WSB/EiJ and DBA/2J. We took the strain-specific gene annotation files released in [6] as the gold standard in order to test the deviations from the mapping intervals to the strain-specific annotations. In total, 533,258 pairs of indices were compared, where 493,031 of them (92.46%) completely agree with the annotations. If both the start and the end indices of a gene agree with the annotation, we consider it as complete matching. As shown in Table 6.2, the mapping of over 90% genes in domestic mouse strains completely agree with the strain-specific annotations. For wild-derived strains CAST and PWK, over 86% of the gene intervals completely match with the annotations.

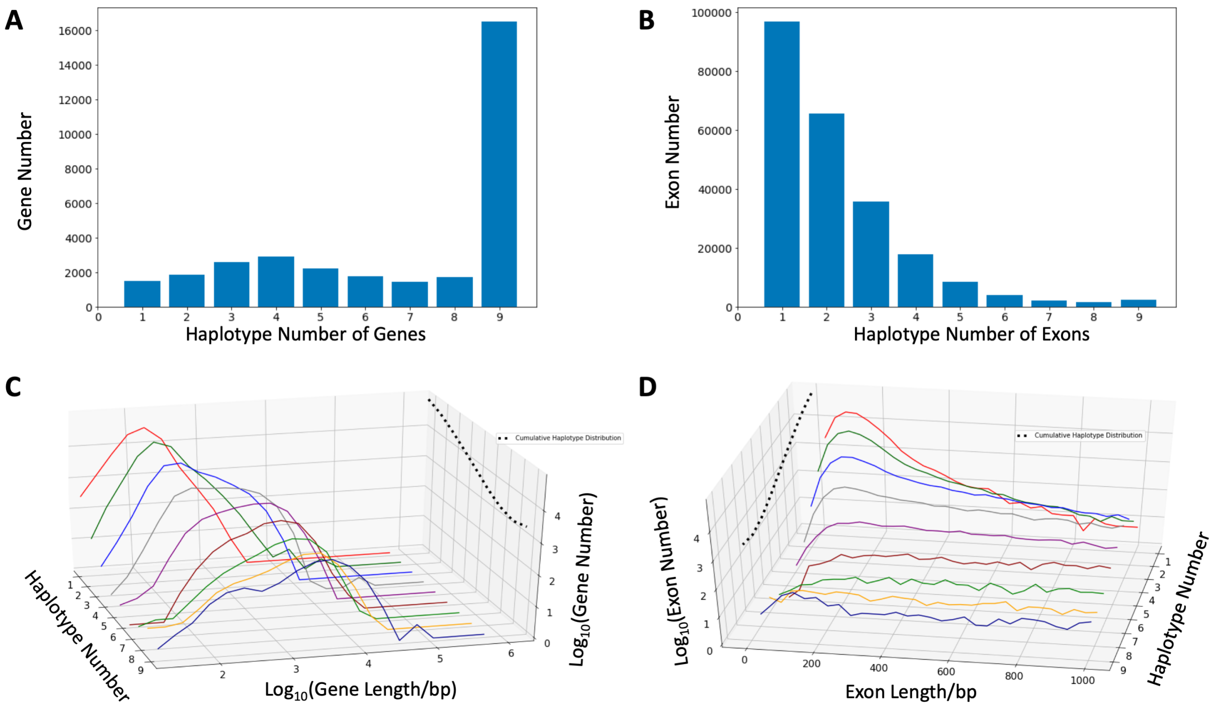


Figure 6.3: Distribution of Haplotype Number in Gene and Exon intervals. Gene and exon intervals in the GRCm38 reference genome were annotated on the B path of the graph. We mapped the start and end indices of each interval to every non-B6 linear assemblies. We extracted and compared the homologous sequences of each feature. The distinct sequence number for gene and exon intervals were plotted in A and B respectively. The length of gene and exon intervals versus the number of distinct haplotype sequences were plotted in C and D.

We extracted the homologous sequences of each genomic feature in 9 linear assemblies. The identical sequences were merged. Homologous sequences of 37,003 genes and 332,264 exons intervals are extracted from graphical genome. The haplotype number distribution of gene and exons are shown in Figure 6.3. Most of the gene intervals are relatively long (>1K) and contain both the intron and exon regions. These long intervals tend to cover more variants than short intervals. 53.43% of the gene intervals are of 9 distinct homologous sequences. Instead, 44.17% of the exon regions are of a single haplotype sequence, suggesting the sequence conservation in the coding regions. Overall, the graphical genome is capable of representing both the single-path and multi-path intervals and aids in the understanding the sequence similarity and diversity of functional features among multiple strains.

Table 6.2: Statistics of Coordinate Mapping Accuracy. The “Exact Match” refers to the number of genes whose mapping indices completely agree with the annotation files. The “Partial Match” refers to the number of genes with one index (either the start or the end position) agree with the annotated interval. The “Not Match” refers to the number of genes neither the start nor the end index agrees with the annotation.

Genomes	TotalNum	Exact Match	Percentage	Partial Match	Percentage	NotMatch	Percentage
AJ	33,216	30,309	91.25%	895	2.69%	2,012	6.06%
129S1	33,144	30,072	90.73%	998	3.01%	2,074	6.26%
NOD	32,713	29,499	90.18%	1,068	3.26%	2,146	6.56%
NZO	33,321	30,468	91.44%	928	2.79%	1,925	5.78%
CAST	31,327	27,188	86.79%	1,433	4.57%	2,706	8.64%
PWK	31,183	26,851	86.11%	1,586	5.09%	2,746	8.81%
WSB	32,051	28,859	90.04%	1,034	3.23%	2,158	6.73%

6.3.4 Updating New Mouse Reference Genome GRCm39

Our anchor-based registration approach provides an efficient framework for managing genome assembly and annotation updates. A new mouse reference genome, GRCm39, was released in July 2020, after our initial GRCm38 assembly of this pangenome. We registered and overlaid GRCm39 onto the mouse pangenome using anchor nodes to partition and register the large genomic contigs. Our pangenome framework provide a relatively stable framework to maintain genomic features with their sequence context and manage multiple versions of annotations. Maintaining the genomic annotations along with the sequence content simplifies the updates of annotations. When a genome assembly is updated, the annotations are adjusted only around modified regions. Meanwhile, most annotations on the graphical genome are unaffected. Thus, the anchor-based coordinate system significantly improves backward-compatibility over traditional linear genomes. We translated the GRCm38 gene and exon intervals to the GRCm39 assembly based on the anchor coordinate framework. We found that 98.94% of our mapping interval of genes and 99.58% of exons completely agree with the GRCm39 annotations (both the start and the end indices agree with the annotations). Overall, our anchor-based graphical genome provides a stable framework for assembly and annotation update and an efficient approach to translating coordinates among multiple assembly versions with a high accuracy.

In addition, the discordant mapping of the candidate sequences during the registration phase identifies distinct sequence organizations relative to the standard reference assembly, and benefit for stabilizing the downstream coordinates, which shall introduce globally impact in the linear coordinate system. As shown in Figure 6.4, there are 1,122 anchor candidate sequences duplicated on chromosome 13 of GRCm39. In our anchor-based graphical genome, this duplicated sequence in GRCm39 is inserted as a new edge between anchor “A13.00279625” and “A13.00287364”, while in the next gap the GRCm39 sequence is merged to the GRCm38 path with no impact from this insertion in GRCm39. In a linear coordinate system, this sequence duplication has a global effect on downstream coordinates, whereas the canonical anchor-based coordinates remain stable. Another genomic region originally located in Chr17:58,196,566 - 58,209,031 bp on GRCm38 appears as a non-tandem duplication in the new build. The candidate anchor sequences in GRCm39 now appear on Chromosome 17: 58,503,561-58,516,026 bp and, in reverse complement order, on Chromosome X: 29,094,739-29,107,204 bp, with over 99% sequence similarity. The mapping of anchor candidates provides insights for large structure alterations in the registered assembly, which are not easily identified and represented by colinear sequence comparison. Genomic intervals with enriched inverted candidates were found on GRCm39 (Table 6.3). We plot the inverted sequence mapping within an interval on chromosome 11 and chromosome 9 (Figure 6.5). There are 75 candidate k-mers mapping consistently in a reversed order between the same pair of anchors on GRCm39. The similarity between the GRCm38 and the reverse complement GRCm39 of the sequences is over 99%, suggesting a inversion occur in GRCm39 relative in GRCm38. However, the overall sequence similarity of the cluster of 899 anchor candidates on chromosome 9 is about 60%. This is due to the complex sequence organization in this region. Figure 6.6 shows kmer density and the mapping between GRCm38 and GRCm39, where 8 distinct inverted segments locate at this region. The k-mer mapping provides an intuitive way to reveal the distinct sequence organization between genome assemblies.

In addition, reference-specific sequence structure can be identified via anchor candidates that discordantly map to every non-B6 genome. We examined the clusters of demoted anchor candidates and found 3 intervals with contiguous inverted anchor candidates shared by every non-B6 strains. The length of these intervals are over 1 kb. The longest one was found on Chr1: 61,501,006 - 61,511,086 bp in the GRCm38, spanning 10,080 kb. These sequences appear on the primary strand in both GRCm38 and GRCm39 assemblies, while they appear in reverse complement in the other 7 founder assemblies (Table 6.4). We found that all the corresponding sequences located in the same long gap between the anchor pair “A01.01366479” and “A01.01367224”. The pairwise alignment confirms that the reverse complement of the GRCm38 sequence in this interval appears in every other founder genome with more than 80% sequence similarity. This suggests either a reference-specific sequence inversion or an assembly orientation issue in this region.

Table 6.3: Inverted regions on GRCm39 relative to GRCm38. The genomic regions were shown in this table, where a cluster of anchor candidates appears in the reverse compliment in GRCm39. The “Sequence Similarity” column represents the percentage of matching bases between the GRCm38 and the reverse complemented GRCm39 sequences.

Chromosome	Proximal Anchor	Distal Anchor	GRCm38 Coordinates	GRCm39 Coordinates	Sequence Similarity
Chromosome4	A04.02900369	A04.02901765	Chr4:130516606-130579426	Chr4:130354583-130291803	99.36%
Chromosome8	A08.00450186	A08.00454262	Chr8:20258371-20441791	Chr8:20182864-19999444	99.60%
Chromosome9	A09.02763077	A09.02766152	Chr9:124338466-124476841	Chr9:124100958-124235998	63.83%
Chromosome11	A11.00172384	A11.00172653	Chr11:7757281-7769386	Chr11:7719446-7707341	99.60%

Table 6.4: Reference-specific Inversion and its Coordinates.

Genomes	Startpos (linear)	Endpos (linear)	Startpos (graph)	Endpos (graph)	Orientation
GRCm38	chr1:61501006	chr1:61511086	A01.01366479:9450:B	A01.01366479:19530:B	Forward
GRCm39	chr1:61540166	chr1:61550201	A01.01366479:9450:Q	A01.01366479:19485:Q	Forward
AJ	chr1:59412384	chr1:59424775	A01.01366479:3434:A	A01.01366479:15825:A	Reverse Complement
129S1	chr1:61584812	chr1:61596566	A01.01366479:3968:C	A01.01366479:15722:C	Reverse Complement
NOD	chr1:68919834	chr1:68930623	A01.01366479:6708:D	A01.01366479:17497:D	Reverse Complement
NZO	chr1:61022017	chr1:61032019	A01.01366479:3388:E	A01.01366479:13390:E	Reverse Complement
CAST	chr1:60555705	chr1:60568221	A01.01366479:3237:F	A01.01366479:15753:F	Reverse Complement
PWK	chr1:58194909	chr1:58205248	A01.01366479:3250:G	A01.01366479:13589:G	Reverse Complement
WSB	chr1:60737025	chr1:60750390	A01.01366479:11325:H	A01.01366479:24690:H	Reverse Complement

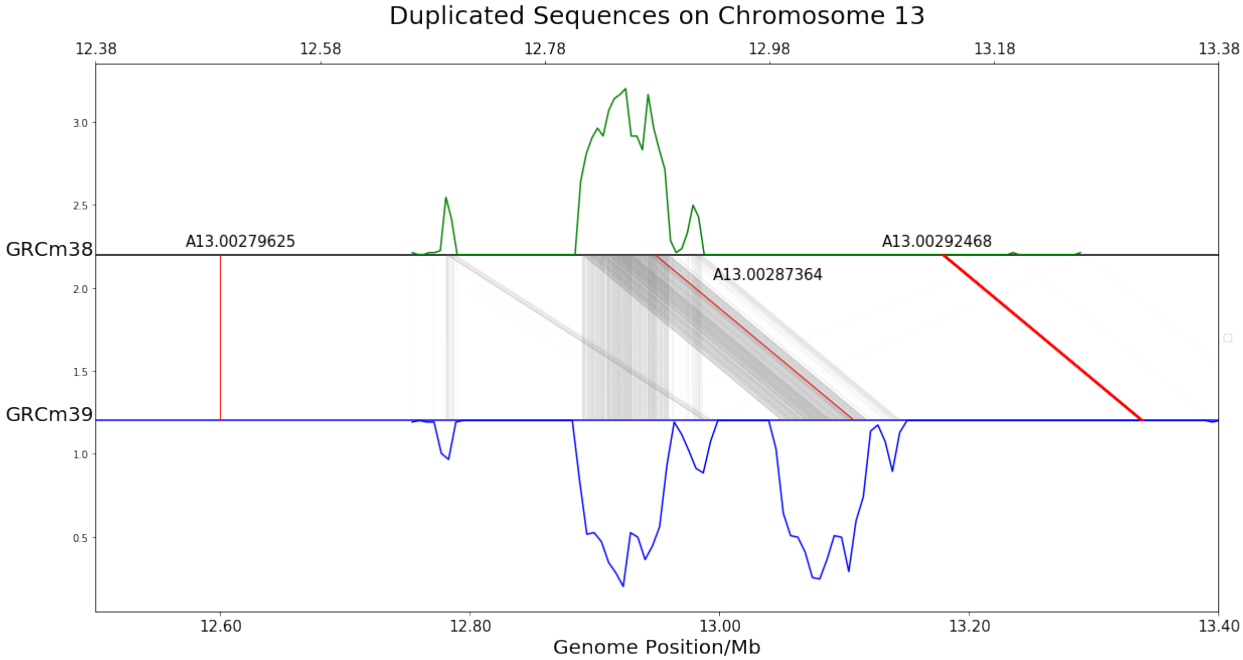


Figure 6.4: Visualization of the duplicated sequences on Chromosome 13. A cluster of 1122 candidate sequences spanning from Chr13:12.7Mb - 13.0Mb on GRCm38 are duplicated on Chr13:12.9Mb-13.2Mb, GRCm39. The duplicated sequences in the new build are inserted as a new edge (508,004 bp) between anchor “A13.00279625” and “A13.00287364” in the graphical genome, where the GRCm38 path is of 348,300 bp on the same gap. In the following gap between anchors “A13.00287364” and “A13.00292468”, the GRCm39 and GRCm38 sequence are identical and are merged into a single path. The density of these duplicated candidate anchor sequences on GRCm38 and GRCm39 is plotted in green and blue respectively. The mapping between GRCm38 and GRCm39 is denoted by grey lines. The anchor node sequences and their mapping are plotted as red lines.

6.3.5 Graph Representation for Large Inversions

When threading GRCm39 to the CCGG, we found a large inverted region spanning over 20Mb at the beginning of chromosome 14. We found that 42,176 anchor sequences appear contiguously in reverse complement in this region. A long gap is found at the boundary of this inversion, where 50,000 Ns are inserted in both GRCm38 (Chr14:19,419,705-19,469,705 bp) and GRCm39 (Chr14:19,469,773-19,474,772 bp). This suggests a major orientation revision at the beginning of chromosome 14 in GRCm39 as compared to GRCm38. Simply demoting these inverted anchors will leave a large gap at the beginning of chromosome 14, where parallel sequences are in different orientation and is not easily compared based by co-linear sequence alignment. Instead, we introduced

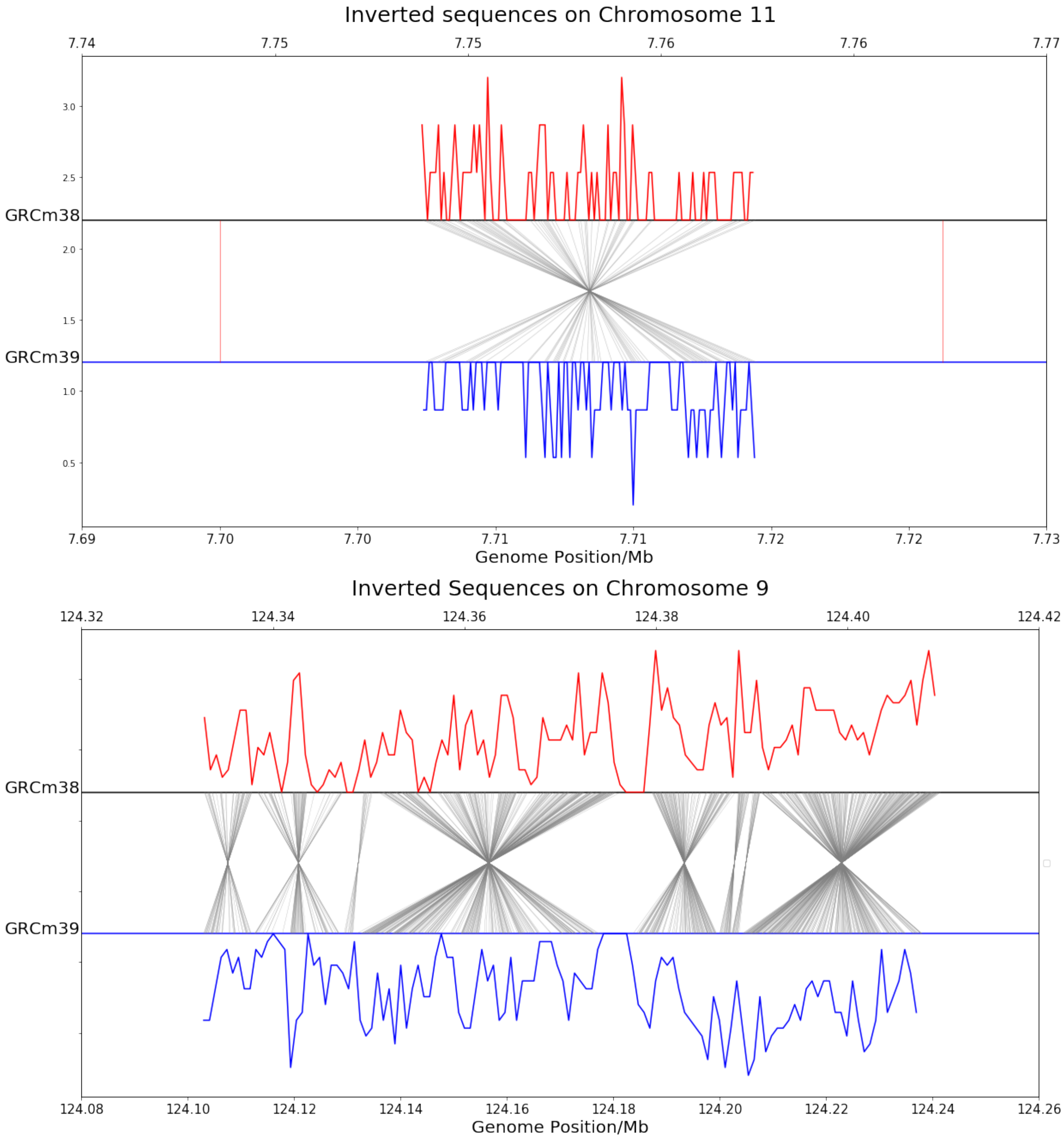


Figure 6.5: Visualization of inverted anchor candidates. A) A cluster of 75 anchor candidats (Chr11:7.75Mb on GRCm38) appear in reverse compliment on the new build GRCm39. The overall sequence similarity between GRCm38 and the reverse complement of GRCm39 sequences is about 99.60%, suggesting these are identical sequences. B) A cluster of 899 anchor candidats (Chr9:124Mb on GRCm38) appear in reverse compliment on the new build GRCm39. As shown in Table 6.3, the overall sequence similarity between GRCm38 and GRCm39 is about 60%, due to the complex sequence organization in this region. The density of inverted candidate sequences is plotted, normalized by the maximum number of sequences for each bin in this region. The mapping between GRCm38 to GRCm39 is denoted by grey lines, showing the relative ordering between these inverted sequences.

edges with “inversion” attribute at the boundary of inverted region and a set of *reversed edges* in between with a “-” annotation after the GRCm39 genome ID in the edge annotation. Incorporating this change had minimal impact on the graphical genome. The details of the topology design for representing large inversion in the graph is described as follows.

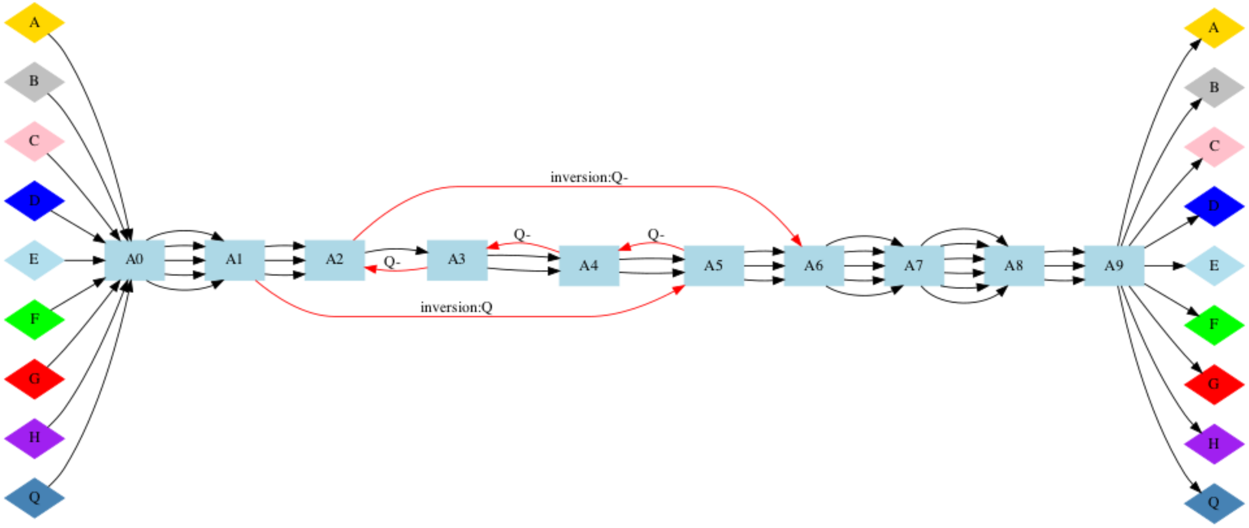


Figure 6.6: Representation for large inversions on the graphical genome. In this figure, anchors are represented by blue boxes; edges are represented by arrows with their traversal directions. The diamonds represent the ”SOURCE” and ”SINK” nodes of each linear genome, abbreviated by 9 distinct letters. We introduce a set of reversed edges (red arrows) to represent large inversions in the graphical genome. The reverse complement of the primary sequences in linear genome Q are stored in these edges with the annotation “Q-”. We annotated the bounding edges with a list of strain attributes in the “inversion” field, as potentially more than one strains could be inverted in the same region. The direction of graph traversal is flipped when pass through the “inversion” signal. The reverse complement of the following anchors and reversed edges are extracted to constitute the path.

The graph topology is implied in the edge attributes by specifying the source and destination nodes of a directed edge. In general cases, the anchor-based graph is traversed from the source to the destination, except when an edge is flagged with an “inversion” attribute. The “Inversion” attribute records a list of paths whose subsequent sequences are inverted relative to other paths and is used to annotate the boundaries of the inverted region. When an “inversion” signal is received during graph traversal, subsequent edges are traversed from “dst” to “src” node, until the next “inversion” signal is received. At that point, graph traversal reverts to its normal direction. While traversing the graph in the “inversion” state, all anchors are reverse complemented, and edges used on the path can indicate whether their sequence should be reverse-complemented by appending a "-" suffix to the path ID, thus allowing sequences to be shared on both orientations (Figure 6.6). We refer to these edges as *reversed edges*. Only the bounding edges of the inverted region include the "inversion" attribute. Small inversions that do not span over multiple gaps are represented as edges. Sequences in the inverted regions are stored in a consistent orientation with the other parallel paths, thus can be compared and mapped coherently. When reconstructing the linear genome, the reverse complement sequences of the anchor and reversed edges in the inverted region are extracted to constitute the path. The preferred sequence orientation in an inverted region is arbitrary, and it is straightforward to edit the graph such that forward and inverted paths are exchangeable. Overall, the graphical genome provides a method to represent large inversions and to compare assemblies in a consistent orientation.

6.4 Discussion

Our graph-based pangenome integrates multiple genomic sequences and annotation resources in a single graph representation. However, referring a base-position unambiguously in a graphical genome faces challenges, since both sequential and vertical coordinates are required to describe a base pair position on a graph with alternative path. A desirable graph-based coordinate system should satisfy *monotonicity*, *spatiality*, *backward-compatibility* [13, 5, 4]. We claim that the anchor-coordinate framework improves all these properties and retains linear-coordinate compatibility, which benefits for the transition from the conventional linear coordinates to graph-coordinates of the pangenome representation.

- **Monotonicity**:

  Monotonicity of the graph-based coordinate system requires the coordinates of successive bases to be incremental. As discussed previously, anchors are shared by every genome presented in the graphical genome. They are unique in the linear assemblies in both forward and reverse complement and are topologically sorted in every linear assembly. A lexicographical sorting of anchor names maintains the sequential ordering of anchors in the linear reference assembly. Thus, anchors partition multiple genomes consistently and establish an sequential ordering between disjoint genomic regions. Referring to base position on a graph relative to a specific anchor fill the graph base pairs into the incremental ordering across the whole genome. Thus, the anchor coordinate framework, which refer graph base position by a combination of anchor name, cumulative offsets relative to the anchor and the path attributes (”Anchor name : Offsets: PathID”), satisfies the *Monotonicity* property of graph-based coordinates. The anchor-based coordinate framework also simplifies the calculation of the horizontal distances between base pairs on the graph.

- **Spatiality**:
  Spatiality in graph-based coordinate system refers to that nearby bases should have similar coordinates. It includes *horizontal spatiality* for bases close to each other along a single path, and *vertical spatiality* for orthologous bases, *i.e.* alternative alleles on parallel paths. As described previously, anchor partition the linear genomes into disjoint genomic regions. Parallel paths between a pair of anchor represent segregating haplotypes. The offsets relative to an anchor defines the horizontal position of a base, while the base pair mapping between parallel paths reveals the vertical spatiality. Successive bases along a path will be represented relative to the proximal anchor with the similar offsets, which full-fill the requirement of the horizontal spatiality. The orthologous bases on linear genomes would either be collapsed to a single base position on the graph or can be mapped to the similar reference coordinates by the alignment-based position mapping methods (details in *Method* section). Thus, anchor coordinate system reveals both the horizontal and vertical spatiality between nearby bases, thus full-fills the *spatiality* property of graph-based coordinate system.

- **Backward-compatibility**:
  Backward-Compatibility requires coordinates of the graphical genome should be unambiguously refer to the same bases after the assembly update or graph modification. Genomic annotations

should still be valid after the assembly update. When incorporating a new genome assembly into the graph, only the anchor sequences are mapped to the new genome. The anchors that satisfy the *conserved*, *unique*, and *monotonic* properties in the new assemblies are retained as anchor nodes. The intervening sequences between anchor pairs in the new genome can be merged to the primary paths or updated as a set of new edges. Thus, most of the anchor-based coordinates from a previous version of the graph should be still valid, except for the region with revised sequences. The annotation in the updated regions should be checked and revised accordingly. Thus, anchor coordinate framework improves the backward-compatibility of the graph-based coordinate system.

- **Linear Coordinates Transformation**:

  Pervasive bioinformatic file formats and pipelines adopt linear coordinates for genomic features representation and analysis. To smooth the transition from linear coordinates to graph-based coordinate representaion, we further proposed that the graph coordinate framework should be able to transform to linear coordinates conveniently. We annotated anchors with their linear coordinates of the CC founder genome and DBA2/J genome in the node files. Any base position on the graphical genome in *anchor:offset:path* format can be transformed into linear coordinates by adding the offset to the annotated linear coordinates of the specified anchor. For recombinant genomes regarded as a mosaic of the founder genomes, the linear coordinates of anchor position can be calculated through graph traversal. Thus, anchor coordinate system is compatible to the conventional linear coordinates.

Unambiguously referring to a base position in a graph-based pangenome faces challenges. Several graph-based coordinate formalization theories have been proposed. A common way for graphical genome construction is to insert well-documented variants in the VCF files to the reference assembly [80, 77, 76]. The genetic variant coordinates are reported with respect to the reference coordinates preserved in this framework. For example, the graph coordinate system provided by VG toolkit encodes paths with additional tags to place the linear reference coordinates to the graph. However, with reference assembly update, variants in VCF files have to be updated to align to the new reference coordinates, and these variation graph should be reconstructed based on a new reference assembly. In addition, genomic complex region and large structure alterations are hard to implement in VCF files. Reference-specific features are especially difficult to represent. The correspondence between orthologous bases is hard to interpret in these compacted sequence graph as well. Computationally, without additional haplotype information, the complexity of graph traversal is exponentially increasing with the new variants added in. To address this issue, VG toolkit implement external haplotype information to the sequence graph to facilitate the graph traversal.

An alternative is to place sequence variations in large contigs to preserve the haplotype information, and then merge multiple assemblies altogether in a graph-based pangenome model. However, there are few standards for partitioning and naming contigs in a graphical genome to address the horizontal and vertical spatiality among the genomic segments. The backward compatibility is also hard to achieve in the conventional offset-based coordinate system proposed by [13], i.e. coordinates are sensitive to graph topology modification. The MSA-based coordinate systems have been proposed [61], yet it is computationally expensive to apply such a MSA-based pangenome and graph coordinate system to large eukaryotic genomes, such as mouse genomes. Several algorithms have been proposed to construct compacted de Bruijn graphs directly from multiple assemblies, such as SplitMeM [57] and TwoPaco [79]. Despite one can encode all the occurrence positions of a vertex on all linear genomes of the graph, the compacted data structure generally introduces complications for graph coordinate and haplotype interpretation.

The anchor-based graphical genome presents a straightforward way to identify, segment, and compare homologous sequences among multiple linear genomes. It preserves sequence variations in the haplotype context, revealing the orthologous structure within each genomic region. The anchor coordinate framework address the major issues working with a graphical genome. The base positions on the graphical genome can be referred to relative to any given anchor sequences as well as the “SOURCE” node of the graph, thus tolerates graph modifications and enables fast linear assembly and annotation update. Instead of a single ”stable” linear coordinate system preserved in the VCF-based pangenome framework, the anchor-based coordinate system enables coordinate representation and comparison in any given linear coordinate system presented in the graph. The vertical spatiality can be naturally represented and compared by using graph coordinate mapping tool. The reference-specific features can be identified by the common features shared by every non-reference genomes.

From the representation point of view, the reference Graphical Fragment Assembly (rGFA) was developed as a file format to represent sequence graphs [90]. With a brand new file format, it creates barriers for applying the graphical genomes in the conventional sequence analysis pipelines. Efforts should be pressed to develop much compatible tools and pipelines for these specific graph representation. Instead, our graphical genome preserves all the sequence and annotations in the standard FASTA file format, leveraging the extant standards for genomic sequence representations. The anchor-based graphical genome allows to extract linear assemblies as well as genomic features in conventional file formats, such as FASTA, VCF, or GFF3 files. The reconstructed linear genomes and anchor nodes can be loaded in the common genome browsers such as IGV for visualization. The compatibility of our anchor-based graphical genome simplifies the maintenance and management of genomic resources and eases the transition from the linear genomes to the graph-based pangenome.

The limitation of the current anchor-based coordinate framework is that, when mapping base positions between non-reference genomes, we adopt the reference genome as an intermediate for the coordinate exchange. This could introduce bias especially where reference-specific variants existed. One possible solution is to perform multiple sequence alignment for all the parallel sequences within each gap. When new assemblies are incorporated in the graph, the segments seperated by anchor pairs can incrementally added to the MSA alignment. We can also construct the consensus sequencec or maximal-length sequence within each gap as a path in the graph. The coordinate mapping can be performed based on the multiple sequence alignment or the maximal sequences instead of relative to the reference assembly. This would effectively reduce the reference bias introduced in the coordinate mapping method.

On the other hand, for some genomic complex regions, there is no valid alignment in the gap, affected by the potential large structural alterations or genome rearrangement. The long gaps in the graphical genome usually incorporate large structural alterations thus may not be compared by colinear alignment. The anchor candidates that are removed during genome registration could potentially be applied to understand the sequence organization and assist position mapping within these complex regions. For example, the genomic regions with large inversions are usually characterized by a continuous run of inverted anchor candidate sequences. As described in previously, we inserted a set of reversed edges with the reverse complement of the primary sequences stored in the graph (Figure 6.6). For each gap on the graph, the parallel sequence are preserved in a consistent orientation and thus the coordinates can be compared by co-linear alignment and mapped.

6.5 Conclusion

Integrating multiple linear genome assemblies and sequence data from the represented population, we selected *conserved*, *unique* and *topologically ordered* k-mers as anchors, which organized multiple genome assemblies into a series of disjoint homologous regions. The anchor sequences establish a valid graph-based coordinate framework that fulfills the desirable properties for a graph-based pangenome representation, including monotonicity, horizontal and vertical spatiality, backward-compatability. It also enables the mapping of coordinates between linear genomes. The graph structure enables extracting homologous sequence within functional regions, facilitating downstream analysis such as genome comparison and visualization. Overall, the anchor-based coordinate framework establishes a practical platform for graph-based feature reference and comparison and assists the transition from the linear genomes to the graph-based pangenomes.

CHAPTER 7: GRAPH REFINEMENTS

7.1 Introduction

The objectives for graph-based pangenome models include reducing reference bias, improving sequence compression, and establishing a coordinate system to reference and annotate genomic features [5]. We construct a graph-based pangenome model, CCGG, for the genome representation of 83 mouse strains, including the 8 CC founder strains and 75 CC strains. We then thread the DBA2/J assemblies to the CCGG, the founder genome of another recombinant-inbred (RI) mouse panel, BxD [124, 135]. The resulting mouse pangenome model can potentially represent more than a hundred mouse strains in a single graph representation. The recombinant genome of each RI strain can be extracted as unique paths from the graph. For the first version of the CCGG, the sequence content of the graph is from the 8 founder assemblies, including the GRCm38 and the other 7 founder assemblies released from [6]. Our CCGG is a series-parallel graph, which can be composed and decomposed by a sequence of series and parallel operations. Thus, our graph retains the capacity to add more linear assemblies or represent specific functional regions by using the subgraphs extracted from the whole genome graph. We show that applying the CC linear genome extracted from the CCGG can reduce reference bias in the sequence analysis of CC MRCA sequence data, the pooled sequence data of CC strain (Chapter 4). We further construct a CC probe database to validate the sequence content of the graph and show that the CC path can reduce the number of unsupported kmers than using the reference path (GRCm38 path, abbreviated by B path, Chapter 5).

There is room of improvement of the current version of CCGG. First of all, the sequence content of CCGG is derived from the 8 founder assemblies including the GRCm38 and the other 7 founder assemblies released from [6]. However, the quality of the other 7 founder assemblies are not as good as the GRCm38. There are many genomic gaps in these founder assemblies. There is a larger fraction of ambiguous bases in the other 7 founder genomes (10.43% in A/J, 14.85% in 129S1/SvImJ, 22.71% in NOD/ShiLtJ, 13.35% in NZO/HlLtJ, 13.70% in CAST/EiJ, 8.92% in PWK/PhJ, 15.64% in WSB/EiJ). Yet, ambiguous bases (Ns) are only comprised of 2.86% of the total length of GRCm38 (chromosome 1-19, X, Y, MT). When aligning the MRCA short-read sequence data to the CC linear genomes extracted from CCGG, the overall alignment rate of CC MRCA data is lower than to the GRCm38 (Table 4.2). Further efforts need to be done on closing the sequence gap in the CC genomes, including collecting more sequence data of CC strains and developing algorithms to refine these complex genomic regions.

In addition, the haplotype intervals of each CC strain can be refined by aligning the corresponding CC MRCA sequence data to the CC genome. The current CC haplotype intervals on the graph are inferred from a HMM model based on the SNP density of short-read sequence data of each individual male sample of a CC strain [119]. Single male samples are actually missing haplotypes present in the colonies. Many factors impact the quality of the CC genomes extracted from the CCGG, including the sequence error introduced by the individual male sample, the prediction errors in HMM model due to the low density of informative SNPs in a genome region, and the ambiguity around the recombination boundaries. In addition, the founder assemblies do not incorporate the CC-specific genomic features introduced during the breeding process. Thus, these CC-specific genomic features are not incorporated in the current version of the CCGG. We applied our CCGG probe database to refine the haplotype interval and identify the genomic regions where an alternative path on the graph that can better interpret the MRCA data than the assigned edges. We also introduced private CC edges with CC-specific genomic mutations that none of the founder assemblies incorporates.

The graph topology of the CCGG can be further refined as well. There are long gaps existing in the CCGG, comprising over 19% of the total length of the mouse reference genome, GRCm38. These long gaps either result from assemblies issues, *e.g.* Ns in the founder assemblies, or large structural variants. For example, a CC026 private deletion is identified on Chromosome17:57Mb, GRCm38 [119]. Due to the special property of anchor sequences that they are conserved in all CC sequenced sample, this private deletion of a single CC strain results in a long gap over 100kb in the CCGG. We can improve the resolution of this region by excluding CC026 sample when selecting anchor candidate sequences and inserting additional anchors in this region. By allowing a private CC026 edge bypassing multiple anchor nodes within this long gap, the graph topology can be refined, enabling a finer resolution of sequence comparison in this region. In addition, we apply the strategies representing large inversions in the graphical genome by allowing reversed edges in the inverted region (as described in Chapter 6). These editions will allow better sequence compression in the graph and benefit for sequence comparison and coordinate mapping.

7.2 Method and Result

7.2.1 Refining Sequence Content of CCGG

As described in Chapter 4, we extract CC linear genomes from CCGG and aligned the CC MRCA short read sequence data to its corresponding CC genome by using a standard short read aligner Bowtie2 [91]. We compare the result aligning the CC MRCA data to the CC linear genome and to the GRCm38 respectively (Table 4.2). The results shows that the overall alignment rate to CC linear genome is lower than to the GRCm38. This results from a larger fraction of ambiguous base pairs in the CCGG (over 10% Ns, Table 4.2) than in the GRCm38 genome (2%). There are many genomic gaps in the other 7 founder assemblies except for GRCm38, with an arbitrary number of Ns inserted in these unresolved regions. Ns comprise 10.43% of A/J genome, 14.85% of 129S1/SvImJ genome, 22.71% of NOD/ShiLtJ genome, 13.35% of NZO/HlLtJ genome, 13.70% of CAST/EiJ genome, 8.92% of PWK/PhJ genome, and 15.64% of WSB/EiJ genome respectively (chr1-19, chrX). As the CC genome is a mosaic of the 8 founder assemblies, each CC linear genome inherits different combinations of these unresolved gaps. Taking CC010 as an example, the overall alignment rate of CC010 MRCA data is 89.50% to the CC010 genome and 92.47% to the GRCm38 reference genome. The fraction of Ns in the CC010 genome is 13.36%, which is significantly higher than the proportion of Ns in GRCm38 (2.86%). We found 34,565,650 unmapped reads when aligning CC010 MRCA short read sequence data to the reference genome GRCm38, and

47,495,134 unmapped reads aligned to the CC010 genome. 30,376,828 of these reads are not aligned to neither GRCm38 nor CC010 genome. 4,188,822 reads only map to CC010 genome. 17,118,306 reads only map to the GRCm38. We plot the distribution of these 17,118,306 reads in Figure 7.1. We also plot the distribution of the number of Ns in the CC010 genome and map them to the GRCm38 coordinates. As shown in Figure 7.1, the regions with many Ns usually colocalized with the unmapped reads to the CC010 genome, such as chr14:75Mb or chr15:70Mb. It suggests that the genomic gaps with Ns in the CC010 genome is responsible for many unmapped reads in the CC010 genome compared to the GRCm38 reference genome. Nevertheless, we found a few genomic regions with unmapped reads, where the fraction of Ns are low, such as Chr5:95Mb. This suggests a potential error of the CC010 genome in the CCGG, either resulting from mislabeled edges or CC-specific mutations in this regions. It is essential to close the assembly gap of Ns in the CC genomes and improve the sequence content of the CC genomes.

7.2.2 Eliminate Redundant Paths

Anchor pairs divide genome assemblies into short fragments and the parallel paths between each anchor pair represent segregating haplotypes. However, for the genomic regions where the tandem repeats or duplication exist, the founder assemblies may only differ from the number of Ns or the number of repetitive elements inserted in each assembly. Thus, one of our attempt to refine the sequence content of the graph is to integrate information of parallel paths within each anchor pairs, using multiple CC sequenced samples to validate and eliminate the redundant paths in the graph. In other words, we refine the sequence content of the CCGG by merging the redundant sequences in the input assemblies. These redundant paths may not carry any novel genomic features. Leveraging the short-read sequence data of the available CC samples [119, 120], we first identify the identical pairs of parallel paths between adjacent anchors. As described in Chapter 5, we developed a k-mer query tool for population-scale sequence analysis based on a pangenome model [145]. The CCGG probe database was constructed, where a set of representative probes were selected to cover every base pair of the CCGG. The occurrence count of each selected probe was calculated in each

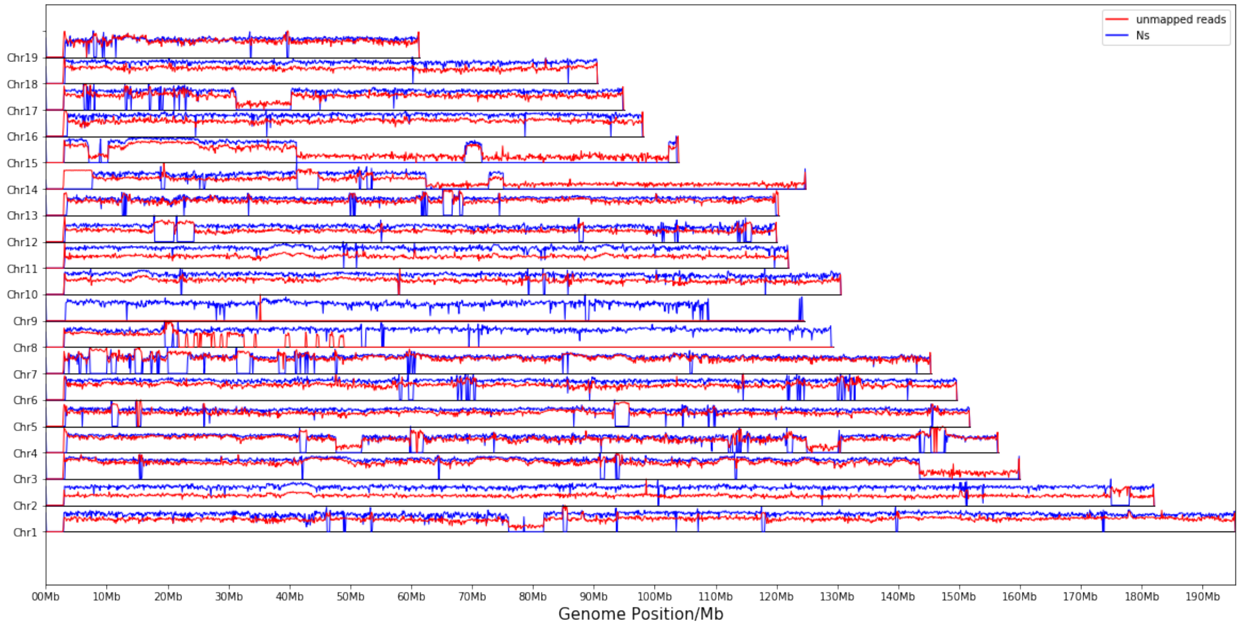


Figure 7.1: Distribution Ns and Unmapped Reads in CC010 Genome. We mapped the CC010 MRCA short reads to the GRCm38 reference genome and the CC010 genome reconstructed from CCGG respectively. We identify the read set that is unmapped to the CC010 genome but are aligned to the GRCm38 reference genome. We plot the distribution of these reads on the GRCm38 genome coordinates (red lines). We further calculated the number of Ns on the CC010 genome and plot the distribution of these Ns (blue lines) on the GRCm38 coordinates. The normalized values of both unmapped reads and Ns are calculated by the log10 value of the raw counts in 100kb non-overlapping window, divided by the maximal value across each chromosome.

sequenced CC sample (details in Chapter 5). We applied the CCGG probe database to identify the identical paths between a pair of anchors. Here, we define identical paths as every probe on one path are supported by multiple reads in all the sequenced samples labeled on the other path, and vice versa. We used the probe database to testify if any pair of parallel paths can be considered as identical. Among a set of fully connected identical paths, *i.e.* any pair of paths in the set are considered as identical in the set, we chose the path with the longest sequences (excluding Ns) as the optimal path. Other identical paths were removed and their annotations were merged to the optimal one. This process removes a significant number of Ns in the CCGG.

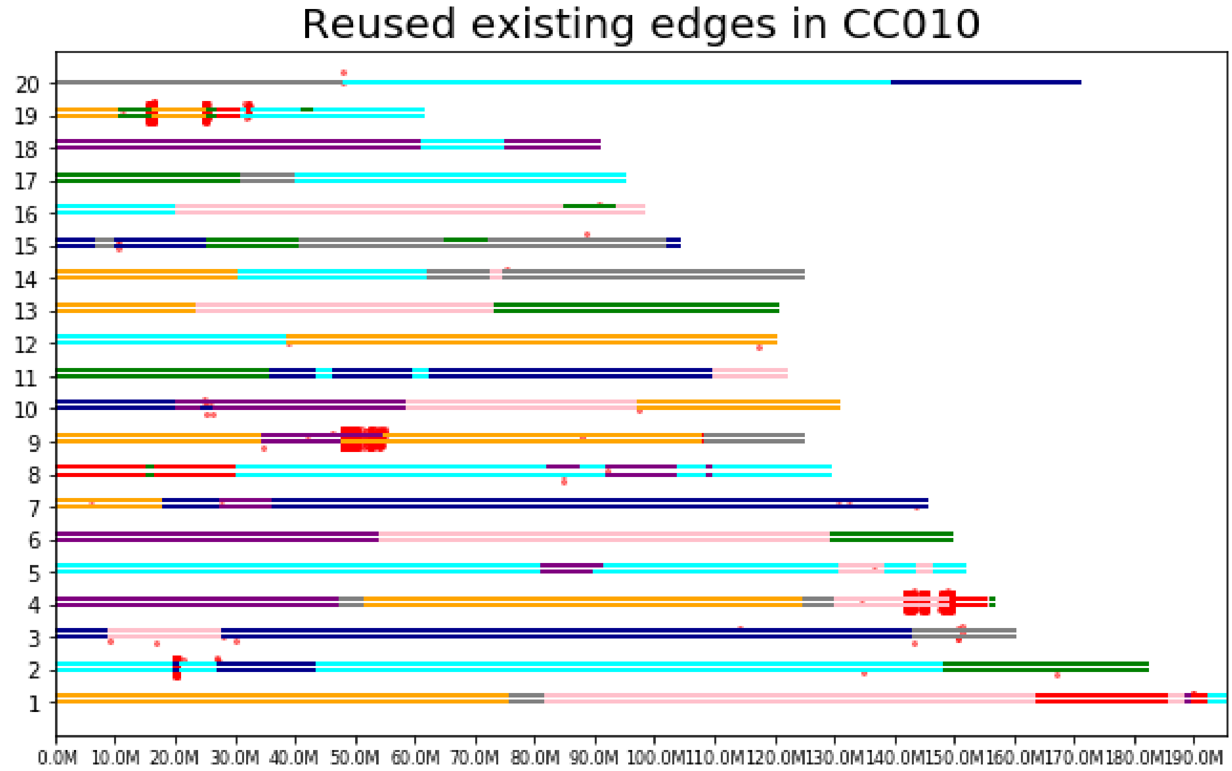


Figure 7.2: Distribution of mislabeled edges in CC010. The contigs of CC010 genome are represented as unique paths in the CCGG. Its descended founder haplotypes are represented by the standard colors in this figure. The overlaid CC haplotype intervals are initially inferred from sequenced male sample [119]. The heterozygous regions are displayed by parallel lines with two founder colors. The mislabeled edges are represented by red dots. They are result from the ambiguity of recombination boundaries (Chromosom 19), heterozyguous regions being fixed (Chromosome 9), or errors in HMM prediction based on single male samples (Chromosome 4).

7.2.3 Refining the CC Path on CCGG

As described in Chapter 5, we developed k-mer query tool and constructed a CCGG probe database to assess the sequence diversity in the CC population [145]. The *Most Recent Common Ancestor* (MRCA) data of CC strains, refered to as the sequence data collected from a pool of individual samples of each CC strain, captures the fixed genomic features of a CC strain and are more robust characterizing the CC-specific features than any individual sequenced sample. We applied the k-mer counts of the MRCA data to identify the features fixed in a CC strain. The CC haplotype intervals labeled to the CCGG are inferred from a hidden markov model (HMM) based on the sequenced data of individual CC male samples [119]. Traversing a prior CC path inferred from single male sample on the graph, we selected the k-mers that are absent from MRCA data but are supported by multiple reads in the founder sample and all the CC samples descended from the same founder. These absent k-mers result either from mislabeling of the founder path or the private mutations that are fixed in a CC strain. We first test if any other parallel path in the same region could characterize the CC MRCA data better than the primary path. If every k-mer on the alternative path are supported by multiple reads in the MRCA data, we will transfer the CC annotation to the alternative path. We plotted the distribution of these relabeled edges in CC0010 genome (Figure 7.2). Several clusters of the relabeled edges were identified in CC010 genome, suggesting potential mislabeling in these genomic regions. These may result from the ambiguity around recombination boundary, heterozygosity, unfixed features introduced by individual sample, or errors in HMM prediction.

Another possibility is to apply the standard bioinformatics pipelines to analyze CC sequence data based on the CC linear genome extracted from the CCGG. We use a standard sequence aligner bowtie2 [91] to align the CC010 MRCA data to the CC010 linear genome extracted from CCGG. We then call variants directly from the alignment to the CC linear genome. We use a standard variant caller freebayes [146] to call variants based on the alignment results of CC010 MRCA data to the CC010 genome. Variants are reported relative to the linear CC010 genome (Figure 7.3). Since the MRCA data are the sequence samples from a pool of individuals of a CC strain, the

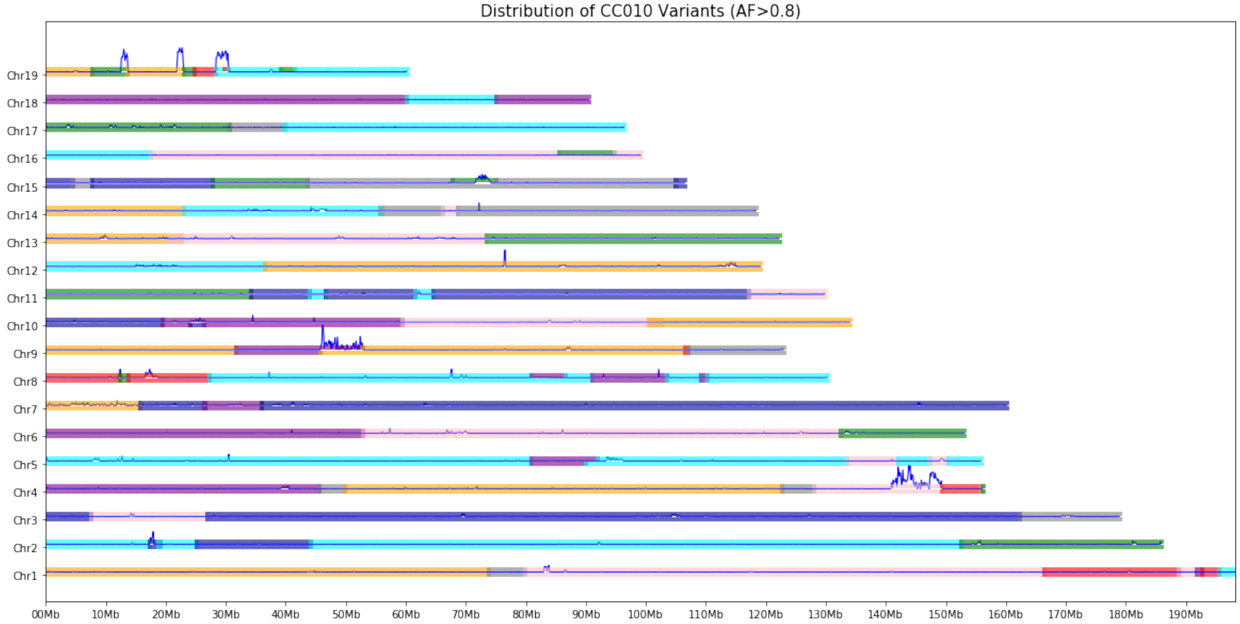


Figure 7.3: Distribution of Fixed Variants in CC010 MRCA sample (Allele Frequency over 0.8). We align CC010 MRCA sequenced data to CC010 genome and plot the distribution of fixed variants (AF $>$ 0.8). The variants density are represented by blue lines, calculated in 100kb bins. The haplotype intervals of CC010 genome are displayed by standard colors of founder genomes in this figure.

homozygous variants represent the fixed features in a CC strain. We filtered the variants based on its quality score and the allele frequency (AF $>$ 0.8). We found a few genomic regions that are enriched with fixed variants (Allele Frequency $>$ 0.8). The variant distribution points out the genomic regions to be refined, which largely agree with the mislabeled regions inferred based on the CC probe database (Figure 7.2, Figure 7.3). The potential reasons for those variant clusters include ambiguity in recombination regions (chromosome 19), heterogeneous regions (chromosome 15), or errors in HMM prediction (chromosome 4). We refine the CC010 linear genome by substituting founder alleles with these fixed variants. If the new sequence is identical with any alternative founder sequences between the same anchor pairs, we will re-label the CC010 annotation to the corresponding founder path. We identified the mis-labeled genomic region by finding a cluster of variants due to the mislabeling of the founder path. For these regions, we identify the correct founder path with the highest votes of each anchor interval and consistently switch the CC010 label to the corrected founder path (Figure 7.4). To validate our result, we took the refined contig and

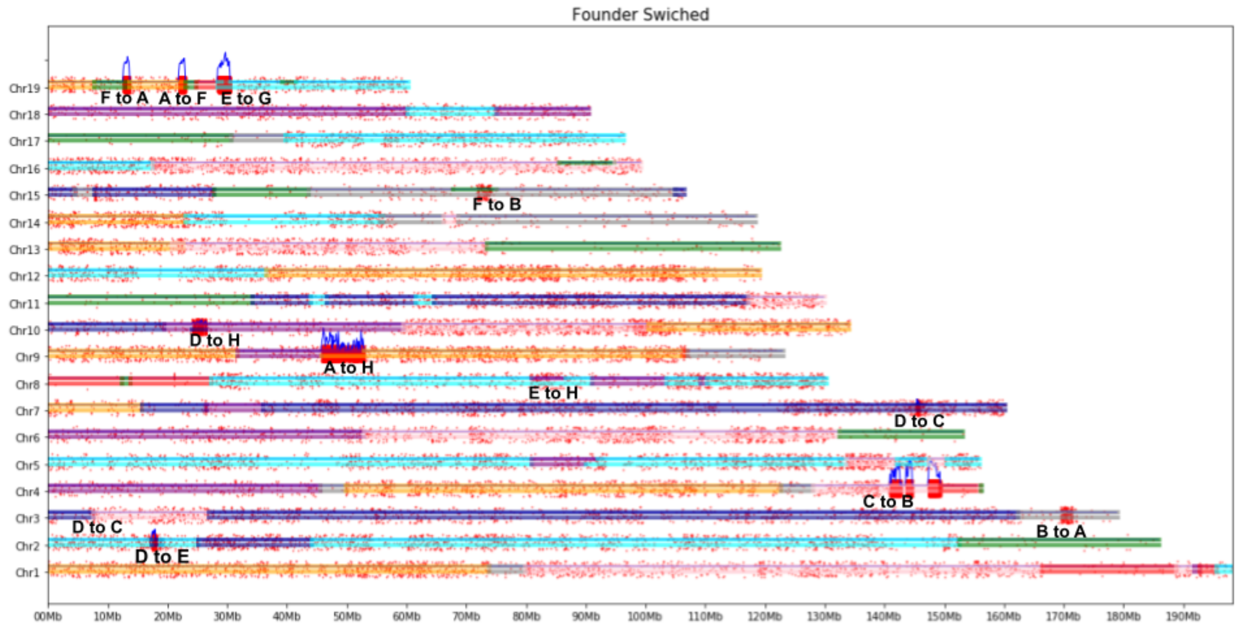


Figure 7.4: Refined CC Haplotype Intervals. We substitute the fixed variants to the CC genome and compare with the parallel founder paths between each anchor pairs. Each red dot represent the revised sequences identical to an alternative founder edge. We found genomic regions where the founder haplotype consistently switch to the other strain, suggesting the mislabelling of the haplotype interval of the CC strain.

realign the CC010 MRCA short-reads to the contig (Figure 7.5). We found the graph refinement process effectively improves the alignment in this region and reduces the number of fixed variants reported. We also identify the heterozygous regions that have not been fixed in each CC strains according to the variance of allele frequency.

7.2.4 Integrating New Sequences to the Graph

Combining multiple sequence data of the CC population, we identify genomic features at population scale, including the private mutation in a single strain, the segregating features descended from the same founder or the shared features across the whole population. This can be either achieved by adopting the CC probe database or performing joint variant calling based on the CCGG. The CC private mutations are implicated by the sequences absent in a single strain but appear in the founder and every other CC strains descended from the same founder (Figure 7.6A). Taking the

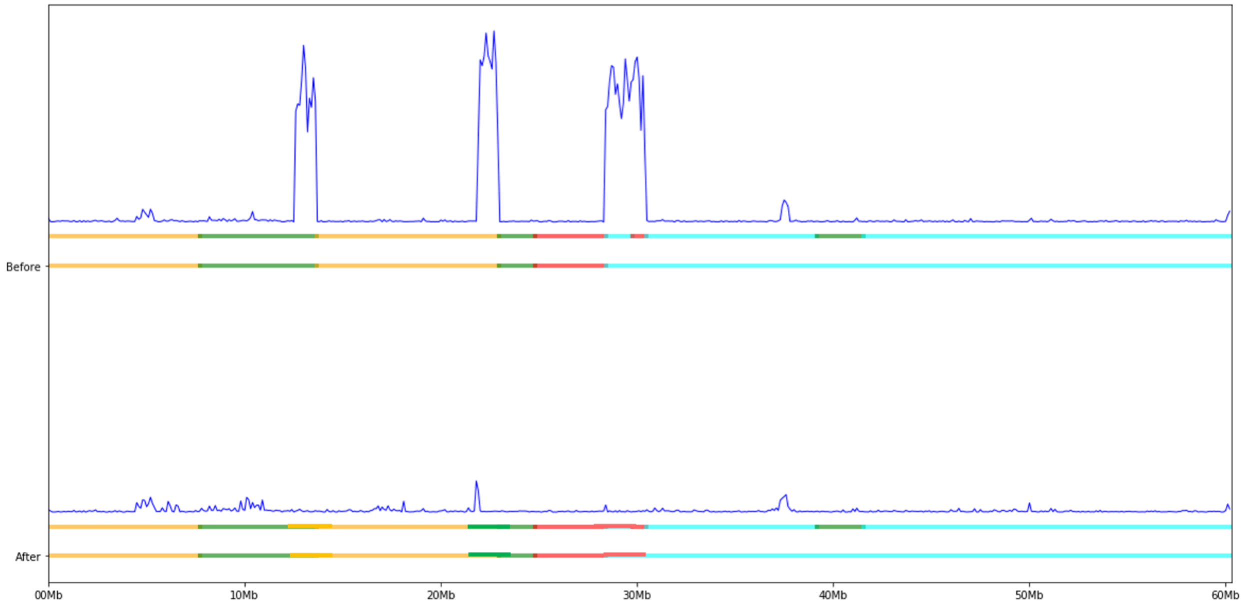


Figure 7.5: Genome Refinement. We refine the CC010 genome by replacing the prior CC010 path with the predicted founder path by the fixed variants (AF $>$ 0.8). The contrast of fixed variants before and after the genome refinement are shown in this plot on chromosome 19. The density of variants are shown in blue, normalized by the total base pairs in each 100kb bin. The prior and refined haplotype intervals are shown in the horizontal line colored by standard founder colors.

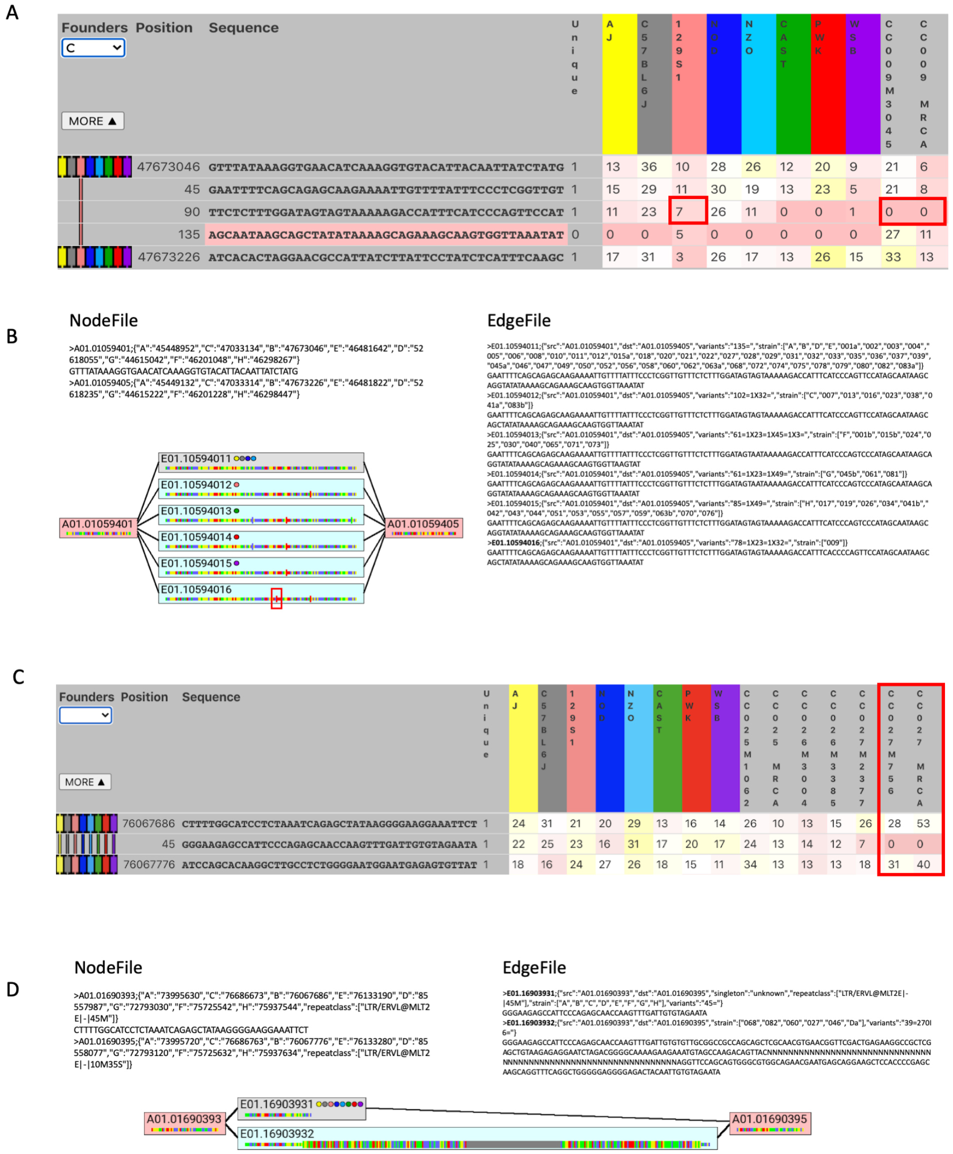
adjacent sequences as seeds, we perform local sequence assembly to construct the alternative k-mer in place of these absent k-mer with variants. K-mers that contain ambiguous bases (Ns) or polyNs were excluded. As described in the Method section, we took the prefix and suffix of the absent k-mer as probes to extract reads from the MRCA data. Consensus sequences were constructed from both ends and aligned. If the assembled sequences from both ends agree with each other, the alternative sequences were extracted between prefix and suffix and were implemented to the CCGG as the CC-specific edges. If the consensus sequences do not pass the alignment test, we will drop those replacements. We inserted 618 CC private edges across 10 CC strains to the CCGG based on MRCA data. These private edges incorporate private mutations fixed in CC strains. Alternatively, as described in the previous section, we can perform variant call and substitute the founder allele with the fixed variants. We can validate the k-mers with these new variants by counting the k-mer occurrence in all the founder and CC sequence data to determine if its a CC private mutation, or segregated mutation or a mutation shared by the founder strain and CC strains descended from the the founder.

In addition, genomic discoveries identified in previous works can be integrated to the CCGG. *De novo* transposable element insertions (TEis) were identified in the founder and CC strains [133]. Six *de novo* TEis in founder strains were identified that segregate in CC strains descended from the founder strains. We first applied the CC probe database to test if the genome assemblies released from [6] captures these TEi sequences. As shown in Figure 7.6, a TEi sequence (Chr1:76,067,686-76,067,776) was segregating in the NOD/ShiLtJ and the CC strains descended from NOD/ShiLtJ in this region. We found that, in the CCGG, the 8 founder assemblies share a single edge in this region with no repetitive elements, suggesting that the released assemblies in [6] did not capture the *de novo* TEi. The CC probe database suggests that CC068, CC082, CC060, CC027, CC046 possess the TEi sequence as they do not support the k-mer present in the founder assemblies in this region. On the other hand, CC020, CC025, CC037, CC050 have multiple reads supporting the shared k-mer in the founder assembly, indicating they do not acquire this TEi. This conclusion is consistent with the genetic mapping results of the TEi sequence [133]. We took the proximal and distal anchor sequences of the TEis from the CC probe database. We queried the sequenced CC samples that share the TEi and extracted reads supporting the probes. We constructed the consensus sequences from both ends and added “Ns” in the middle to represent the repetitive sequences of the TEis. We implemented the new sequences to the CCGG by annotating the CC strains sharing this TEi sequence to the *strain* attribute of the edge. Overall, CCGG provides as a platform to integrate novel genomic discoveries of the CC population in a single pangenome framework.

7.3 Future Direction

7.3.1 Leveraging Long Read Sequences to Improve the Sequence Content of the CCGG

The sequence content of CCGG comes from the 8 founder genome assemblies. The short-read sequence data of the CC founder strains have been collected and utilized to validate the founder



Founders	Position	Sequence	Unique	AJ	C57BL6J	129S1	NOD	NZO	CAST	PWK	WSB	CC009M3045	CC009 MRCA
	47673046	GTTTATAAAGGTGAACATCAAAGGTGTACATTACAATTATCTATG	1	13	36	10	28	26	12	20	9	21	6
	45	GAATTTTCAGCAGAGCAAGAAAATTGTTTTATTTCCCTCGGTTGT	1	15	29	11	30	19	13	23	5	21	8
	90	TTCTCTTTGGATAGTAGTAAAAAGACCATTTCATCCCAGTTCCAT	1	11	23	7	26	11	0	0	1	0	0
	135	AGCAATAAGCAGCTATATAAAAGCAGAAAGCAAGTGGTTAAATAT	0	0	0	5	0	0	0	0	0	27	11
	47673226	ATCACACTAGGAACGCCATTATCTTATTCCTATCTCATTTCAAGC	1	17	31	3	26	17	13	26	15	33	13

NodeFile

```
>A01.01059401;{"A":"45448952","C":"47033134","B":"47673046","E":"46481642","D":"52
618055","G":"44615042","F":"46201048","H":"46298267"}
GTTTATAAAGGTGAACATCAAAGGTGTACATTACAATTATCTATG
>A01.01059405;{"A":"45449132","C":"47033314","B":"47673226","E":"46481822","D":"52
618235","G":"44615222","F":"46201228","H":"46298447"}
```

EdgeFile

```
>E01.10594011;{"src":"A01.01059401","dst":"A01.01059405","variants":"135=","strain":["A","B","D","E","001a","002","003","004","
005","006","008","010","011","012","015a","018","020","021","022","027","028","029","031","032","033","035","036","037","039",
"045a","046","047","049","050","052","056","058","060","062","063a","068","072","074","075","078","079","080","082","083a"]}
GAATTTTCAGCAGAGCAAGAAAATTGTTTTATTTCCCTCGGTTGTTTCTCTTTGGATAGTAGTAAAAAGACCATTTCATCCCAGTTCCATAGCAATAAGC
AGGTATATAAAAGCAGAAAGCAAGTGGTTAAATAT
>E01.10594012;{"src":"A01.01059401","dst":"A01.01059405","variants":"102=1X32=","strain":["C","007","013","016","023","038","0
41a","083b"]}
GAATTTTCAGCAGAGCAAGAAAATTGTTTTATTTCCCTCGGTTGTTTCTCTTTGGATAGTAGTAAAAAGACCATTTCATCCCAGTTCCATAGCAATAAGC
AGCTATATAAAAGCAGAAAGCAAGTGGTTAAATAT
>E01.10594013;{"src":"A01.01059401","dst":"A01.01059405","variants":"61=1X23=1X45=1X3=","strain":["F","001b","015b","024","0
25","030","040","065","071","073"]}
GAATTTTCAGCAGAGCAAGAAAATTGTTTTATTTCCCTCGGTTGTTTCTCTTTGGATAGTAATAAAAAGACCATTTCATCCCAGTCCCATAGCAATAAGCA
GGTATATAAAAGCAGAAAGCAAGTGGTTAAGTAT
>E01.10594014;{"src":"A01.01059401","dst":"A01.01059405","variants":"61=1X23=1X49=","strain":["G","045b","061","081"]}
GAATTTTCAGCAGAGCAAGAAAATTGTTTTATTTCCCTCGGTTGTTTCTCTTTGGATAGTAATAAAAAGACCATTTCATCCCAGTCCCATAGCAATAAGCA
GGTATATAAAAGCAGAAAGCAAGTGGTTAAATAT
>E01.10594015;{"src":"A01.01059401","dst":"A01.01059405","variants":"85=1X49=","strain":["H","017","019","026","034","041b","
042","043","044","051","053","055","057","059","063b","070","076"]}
GAATTTTCAGCAGAGCAAGAAAATTGTTTTATTTCCCTCGGTTGTTTCTCTTTGGATAGTAGTAAAAAGACCATTTCATCCCAGTCCCATAGCAATAAGC
AGGTATATAAAAGCAGAAAGCAAGTGGTTAAATAT
>E01.10594016;{"src":"A01.01059401","dst":"A01.01059405","variants":"78=1X23=1X32=","strain":["009"]}
GAATTTTCAGCAGAGCAAGAAAATTGTTTTATTTCCCTCGGTTGTTTCTCTTTGGATAGTAGTAAAAAGACCATTTCACCCCAGTTCCATAGCAATAAGC
AGCTATATAAAAGCAGAAAGCAAGTGGTTAAATAT
```

Founders	Position	Sequence	Unique	AJ	C57BL6J	129S1	NOD	NZO	CAST	PWK	WSB	CC025M1062	CC025 MRCA	CC026M3004	CC026M3385	CC027M2377	CC027M756	CC027 MRCA
	76067686	CTTTTGGCATCCTCTAAATCAGAGCTATAAGGGGAAGGAAATTCT	1	24	31	21	20	29	13	16	14	26	10	13	15	26	28	53
	45	GGGAAGAGCCATTCCCAGAGCAACCAAGTTTGATTGTGTAGAATA	1	22	25	23	16	31	17	20	17	24	13	14	12	7	0	0
	76067776	ATCCAGCACAAGGCTTGCCTCTGGGGAATGGAATGAGAGTGTTAT	1	18	16	24	27	26	18	15	11	34	13	13	13	18	31	40

NodeFile

```
>A01.01690393;{"A":"73995630","C":"76686673","B":"76067686","E":"76133190","D":"85
557987","G":"72793030","F":"75725542","H":"75937544","repeatclass":["LTR/ERVL@MLT2
E|-|45M"]}
CTTTTGGCATCCTCTAAATCAGAGCTATAAGGGGAAGGAAATTCT
>A01.01690395;{"A":"73995720","C":"76686763","B":"76067776","E":"76133280","D":"85
558077","G":"72793120","F":"75725632","H":"75937634","repeatclass":["LTR/ERVL@MLT2
E|-|10M35S"]}
```

EdgeFile

```
>E01.16903931;{"src":"A01.01690393","dst":"A01.01690395","singleton":"unknown","repeatclass":["LTR/ERVL@MLT2E|-
|45M"],"strain":["A","B","C","D","E","F","G","H"],"variants":"45="}
GGGAAGAGCCATTCCCAGAGCAACCAAGTTTGATTGTGTAGAATA
>E01.16903932;{"src":"A01.01690393","dst":"A01.01690395","strain":["068","082","060","027","046","Da"],"variants":"39=270I
6="}
GGGAAGAGCCATTCCCAGAGCAACCAAGTTTGATTGTGTGTTGCGGCCGCCAGCAGCTCGCAACGTGAACGGTTCGACTGAGAAGGCCGCTCG
AGCTGTAAGAGAGGAATCTAGACGGGGCAAAAGAAGAAATGTAGCCAAGACAGTTACNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNAGGTTCCAGCAGTGGGCGTGGCAGAACGAATGAGCAGGAAGCTCCACCCCGAGC
AAGCAGGTTTCAGGCTGGGGGAGGGGAGACTACAATTGTGTAGAATA
```

Figure 7.6: Integrating novel genomic discoveries to the CCGG. A) A private mutation fixed in CC009 were identified from the CCGG probe database. B) A CC009 private edge containing this private SNP were introduced to the CCGG. C) CCGG probe database verified a de novo TEi insertion discovered in CC population. D) The de novo TEi insertion were introduced to the CCGG as a new edge shared by multiple CC strains.

assemblies and further refine the CC genome reconstructed from the CCGG. As described previously, there are a large fraction of genomic gaps in the other seven founder assemblies except for C57BL/6. The number of Ns in these gaps does not always represent the length of these ambiguous regions, and is largely arbitrary. The length of short-reads sequences collected at UNC (150 bp) is longer than the source of the founder genomes released in [6], the Sanger sequence data (100 bp). We can assemble through a certain number of genomic gaps with small tandem repeats or repetitive sequences. For example, as shown in Figure 7.7, we align CC010 MRCA short-read sequence data to the CC010 linear genome extracted from CCGG. A series of Ns were inserted in Chr16:82.5Mb in CC010 linear genome to represent repeatitive sequences. CC010 is derived from 129S1/SvImJ in this region. We found that using our CC010 MRCA sequence data can assemble through the regions with Ns. Sequences can be assembled to replace these Ns with valid base pairs in this gap. Once assembled, the kmers that carry the novel sequences in this region can be further validated by querying their occurrence counts in 113 CC sequenced samples to testify if these are features segregating in multiple strains or common across the whole CC population. These new kmers can be further added to the CC probe database.

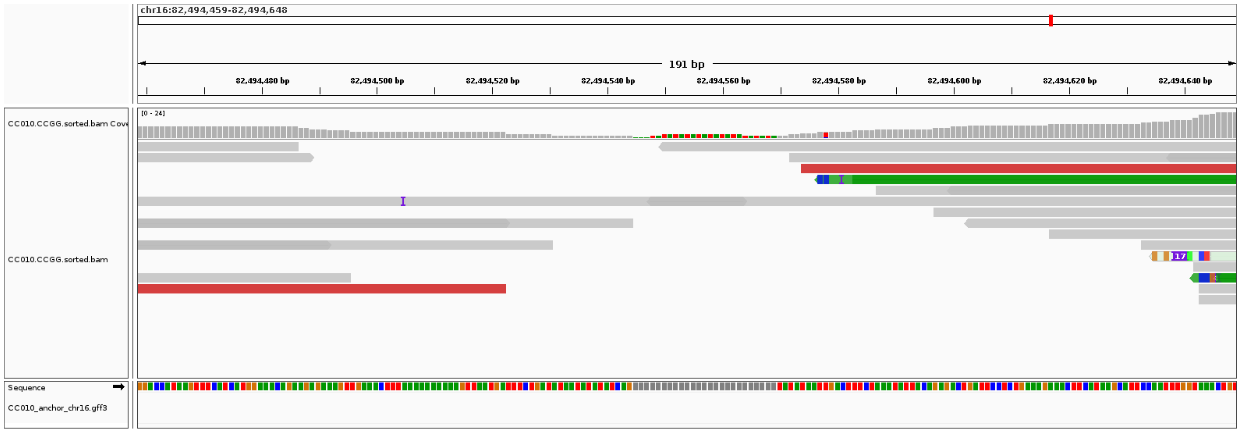


Figure 7.7: Refining Sequence Content of CCGG. The other 7 CC founder assemblies released in [6] comprise of many genomic gaps represented by a series of Ns. We can refine the sequence content of CCGG by using the CC MRCA short-read sequence data, taking the advantage that the sequence length of the CC MRCA short-read sequencing data (150 bp) is longer than the raw read data (100bp) used to assemble the genome assemblies.

A series of Ns in the founder assemblies creates unaligned regions in the CC linear genomes extracted from the CCGG (Figure 7.8). These genomic gaps impact the global alignment rate to the CC genomes (Table 4.2). Efforts should be pressed in resolving these genomic gaps and refining the sequence contents. One possibility is to adopt long-read sequence data collected from CC population to estimate the repeat expansions and the actual length of these ambiguous regions. For example, as shown in Figure 7.9, when aligning CC019 ONT long read sequence data (collected by Feinberg Lab at John Hopkins University) to the CC019 genome, a consensus deletion is reported in the reads covering the region with Ns, which suggests that an extra amount of Ns were inserted in this region. We can estimate the actual length of this unassembled gap and delete the extra Ns inserted in these regions. This could rescue more reads aligned to this region, which help better characterize the repetitive sequences and refine the sequence content in this region.

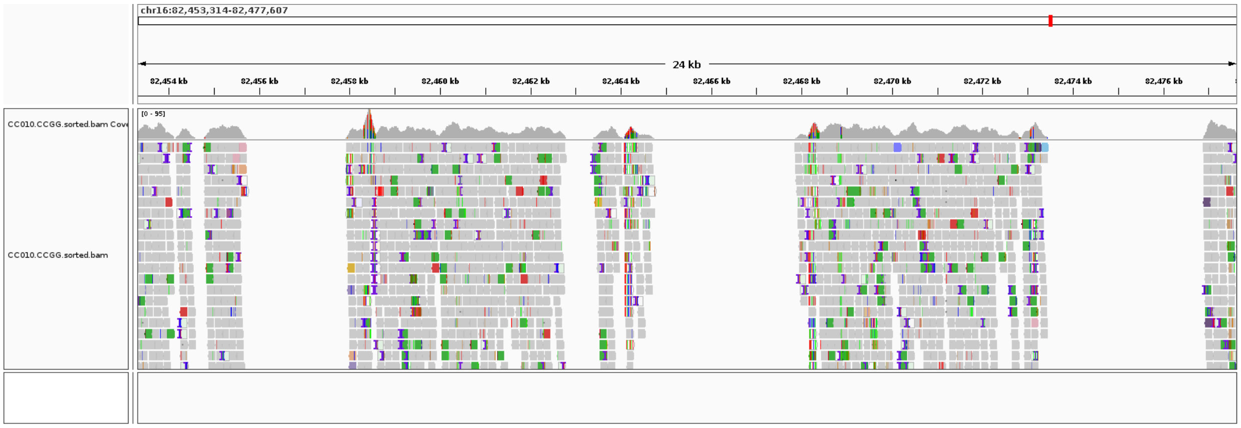


Figure 7.8: Genomic Gaps represented by Ns. We found many genomic gaps in the founder assemblies except for C57BL/6, with an arbitrary number of Ns inserted in the unresolved region. Since the CC linear genomes extracted from CCGG are represented as a recombination of the founder assemblies, this leads to many unaligned gaps in the CC genome. The random number of Ns in the CC genome impact the global alignment rate to the CC genome.

7.3.2 Refine Recombination Boundary of CC Genomes

In the previous work, CC haplotype intervals were reconstructed from a HMM model based on the sequenced CC male sample [119]. The HMM model predict descended founder probabilities based on the informative probes in every 5-kb non-overlapping windows across the whole genome [119]. When estimating the recombination boundaries, the haplotype intervals could be ambiguous due to the sparsity of the informative probes or the identical founder sequences. The inferred haplotype intervals will be extended to the region where the founder paths can be distinguished. We initially set the recombination boundary by the mid-point of these ambiguous regions. We then refine the recombination boundary of CC genomes based on CCGG. One approach is to use the

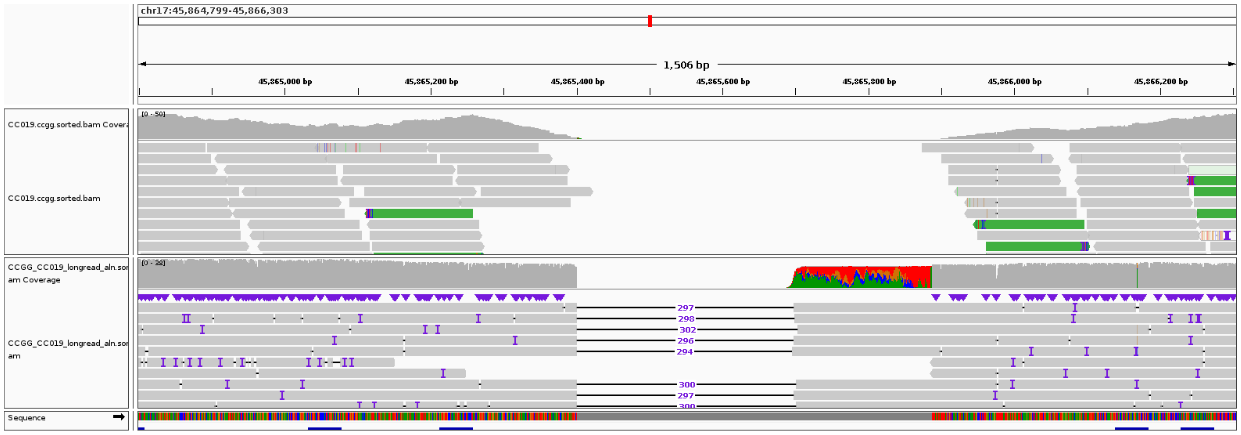


Figure 7.9: Estimating the Actual Length of Ambiguous Regions. We align CC019 MRCA short read sequence data and ONT long read sequence data to the CC019 linear genome extracted from the CCGG. We found that a series of Ns result in an unaligned gap in the short read sequence alignment. The ONT long reads cover the whole region and a consensus deletion were identified in this region, suggesting extra number of Ns were inserted.

CCGG probe database to estimate the optimal path between each pair of adjacent anchors for a CC sample. The haplotype intervals can then be reconstructed by traversing the graph both forwardly and reversely to identify the shared or consensus founder paths across a large genomic region. An alternative approach is to extract CC linear genome and align MRCA short read sequence data of the corresponding strain to identify the fixed variants in a CC strain. For example, the haplotype of CC010 transitions from G (PWK/PhJ) to F (CAST/EiJ) is predicted between chr8:15,424,296 - 15,481,998 bp (GRCm38) based on a HMM from CC010 male sequence sample (Figure 7.10). This boundary locates between anchor *A08.00342765* and *A08.00344047* in CCGG. This corresponds to chr8:12.395 - 12.50Mb on CC010 genome. A 54.7kb long gap lies between these two anchors. Before the anchor *A08.00342765*, the CC010 genome is derived from G path and few variants are reported. The CC010 edge in this large gap is derived from founder F. When aligning CC010 sequences to the F path in this region, a large amount of variants are reported at the beginning of this edge. A clear recombination boundary around chr8:12.448Mb is identified based on the alignment result (Figure 7.10). This is due to the CC010 genome is still of haplotype G at the beginning of this edge, and differences between founder strain PWK/PhJ and CAST/EiJ lead to a large amount of variants inconsistent with the CAST/EiJ edge. We tested this hypothesis by realigning the reads in this region to the alternative G edge (Figure 7.10). We observe an opposite that sequences before the predicted boundary align well but many variants relative to the G assembly are reported after the boundary. This further validate the recombination boundary exists within this edge. The recombination event lead to a CC010 recombinant edge between the PWK/PhJ and the CAST/EiJ sequences and the recombination boundary is predicted around chr8:12.45Mb on CC010 genome. Similar strategies can be applied to identify other recombination boundaries and to refine the sequence contents of each CC genome.

7.3.3 Refine Graph Topology in Long Gaps

The median spacing between anchor pairs in the CCGG is about 180bp in autosomes. Yet there are genomic regions where the anchors are sparse and the distance between adjacent anchor pairs are over 2kb. We refer to these genomic intervals as *long gaps*. As described in Chapter 4, Figure 4.2, about 19% of the reference genome size fall in long gaps, where there is no anchors in these regions. Sequences that are not well assembled (Ns) were found enriched in these long gaps. Other potential reasons for those long gaps are large structural variants, such as inversions, duplications or deletions. We design strategies to refine the graph topology in these long gaps and to improve the resolution for sequence comparison and coordinate mapping. As described in Chapter 6, we represent large inversions by using reversed edges, where the reverse complement of the initial assemblies sequences are stored in those edges. For example, there are about 20Mb sequences are inverted in GRCm39 relative to GRCm38 at the beginning of chromosome 14. We preserved the anchor sequences in this region, and introduced a set of reversed edges of GRCm39 to represent the large inversion in this region. The parallel paths in this region can be preserved in the graph in a consistent orientation, which facilitates the sequence searching, comparison and visualization of these regions. In addition, *de novo* deletions in a single CC strain could also result in long gaps. For example, *de novo* large deletion in CC026 is discovered previously in chr17:57Mb on GRCm38 [119]. Since anchors sequences are common in every sequenced CC sample, this large deletion in the CC026 strain result in a long gap over 100kb on chromosome 17. A potential strategies to refine

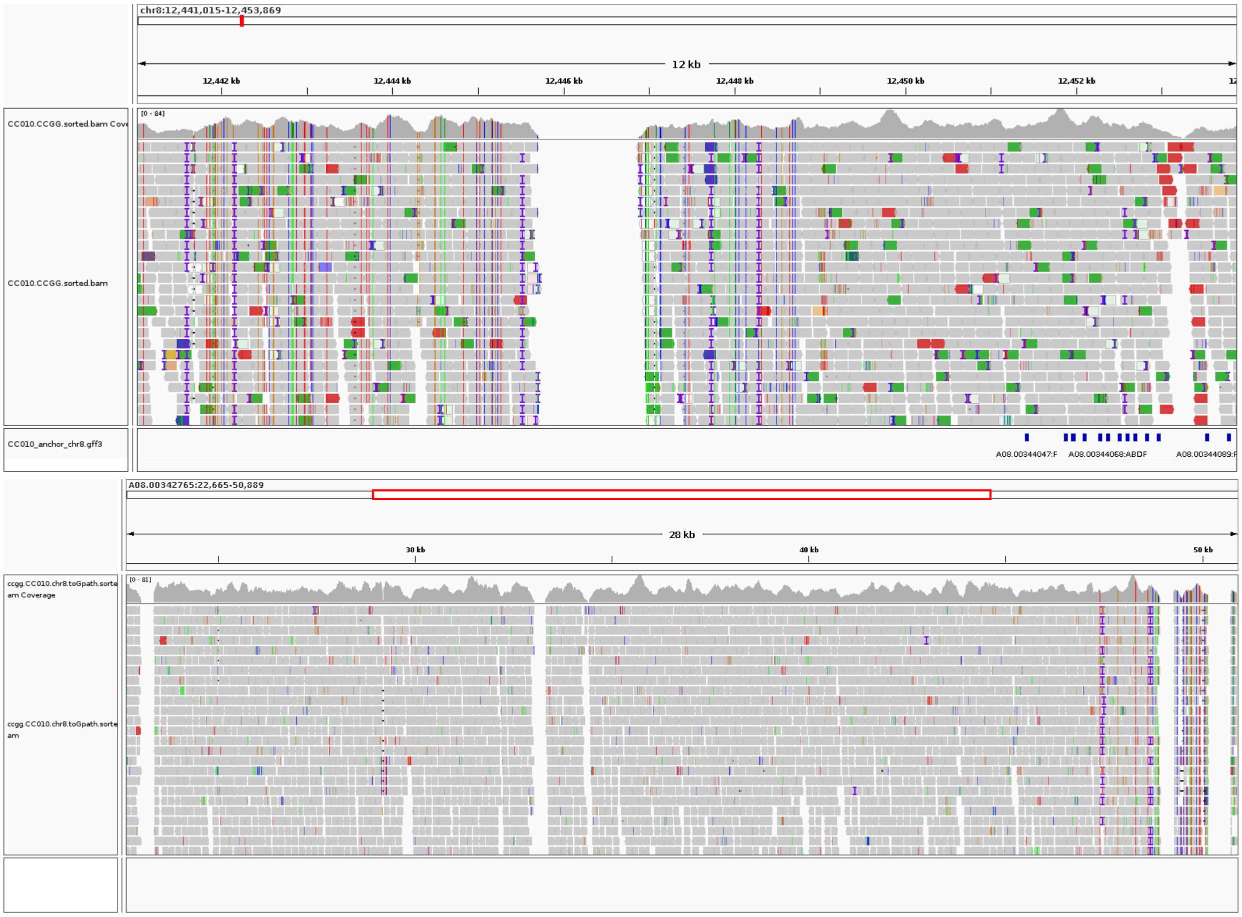


Figure 7.10: Recombination Boundary of CC010. The haplotype of CC010 transitions from G (PWK/PhJ) to F ((CAST/EiJ)) on chr8:15.4Mb, GRCm38 (chr8:12.4Mb on CC010 linear genome). A long gap between *A08.00342765* and *A08.00344047* locates on the recombination boundary. The sequence of CC010 genome is derived from G path before *A08.00342765* and F path afterwards. The alignment shows that a lot of variants are reported until chr8:12.450kb, which identifies the recombination boundary around chr8:12,448kb-12,449kb (CC010 genome). We extracted reads aligned in this region and re-aligned them to the G path between the pair of anchor. We found an opposite observation that CC010 sequences aligns well at the proximal part of this gap, while variants are reported after the recombination boundary. This identifies the recombination within this long gap and the founder genome switch from G to F around chr8:12.448-12.449Mb. To further refine this region, a recombinant edge (G-F) could be constructed and inserted as a CC010 private edge within this gap.

the graph topology around these CC-private *de novo* structural variants is allowing the violation of anchor properties in a small proportion of samples when selecting anchors. This CC private deletion can be represented by inserting a CC026 private edge bypassing multiple anchors rescured in this large gap. For duplications, we can introduce loops or circuits in the graph to represent duplicated sequences or add pointers or annotations to identify the paralogous copies across the graph.

CHAPTER 8: DISCUSSION AND CONCLUSION

8.1 Summary

In this dissertation, we present a novel framework for constructing a graph-based reference pangenome for Collaborative Cross and discuss the operational strategies for using the pangenome model to analyze sequence data. This pangenome framework is built upon the assumption that a core genome exists in an intra-specific population. We selected the anchor sequences that are conserved, unique and topologically sorted in every assembly member of the pangenome. We merge sequences between anchor pairs into edges, which represents segregating haplotypes in a genomic region. Each chromosome is represented as a series-parallel graphical genome. Both the sequences and annotations are stored as a list of standard FASTA files of anchors and edges of the graphical genome. We discuss the pangenome construction pipeline in Chapter 2. We apply our method to construct a graphical genome for a recombinant-inbred mouse population, Collaborative Cross (Chapter 3). By merging the 8 CC founder genome into a graph and overlying the CC haplotype intervals to the founder path, the CCGG represent genomes of 83 mouse strain, including the 8 CC founder strains and 75 CC strains. Each CC linear genome can be represented as a unique path on the graph and can be extracted in forms of standard FASTA file format. These CC-specific genomes can be adopted in conventional bioinformatics pipelines for sequence analysis. We then developed an alignment-free method to analyze population-scale sequence data and construct and CC probe database based on a pangenome model (Chapter 5). We further developed a graph-based coordinate system and coordinate mapping tool for feature annotation and comparison in a multi-genome context (Chapter 6). We assign base pair positions on the graph by using anchor name with offsets and path identifiers. We show that the anchor-based coordinate system provides an efficient platform to preserve and update assembly and annotations.

The development of a comprehensive pangenome model will no doubt open up new insights for genetic research. An unbiased graphical reference pangenome would help for reducing reference bias and improving genomic resources management. While many efforts focus on developing new tools and data formats for pangenome representation, the compatibility with the available bioinformatics tools should also be taken into consideration. Our anchor-based pangenome model provides a unified framework for both preserving multiple sequences in a graph and extracting conventional strain-specific genomes for common sequence analysis. It serves as a bridge between the conventional bioinformatics tools and pangenome models and can ease the transition of the graphical reference pangenome application. For example, public databases or platforms could preserve the graphical genome in their infrastructure, and users can extract linear genomes or variants from the graph as needed.

In the following sections, we will discuss the future work that we can do under this pangenome framework.

8.2 Future Work

8.2.1 Cancer Genomics

Cancer cells can accumulate numerous mutations during tumorigenesis, from SNPs to large structural alterations [147]. Recent studies have found extensive inter-patient and intra-patient heterogeneity of somatic mutations [148]. Furthermore, fast and economical sequencing technology has enabled individual genome assemblies. The genome of an individual, James D. Watson, has been constructed and released [149]. Combining tissue-specific sequence data, these personalized genome assemblies can help identify somatic variants and serve as a scaffold for cancer genome studies.

The identification and interpretation of somatic mutations in cancer genomes are essential for translational research. However, calling somatic mutations faces challenges. First of all, tumors are heterogeneous. A possible way is to first use the matched normal tissue samples to define the genetic background, then call somatic mutations from the pair-wise tumor samples [150]. Pangenome graphs will provide an enriched data structure for characterizing and identifying the novel somatic mutations in the tumor samples. Secondly, structural variants are prevalent in cancer genomes [150]. Short-read sequencing techniques have been widely used for genotyping, but most of these tools focus on small mutations, such as SNPs or short indels. Detecting structural variants and mutations in repetitive regions is challenging for short reads due to the limitation of read length [116]. Phasing is also challenging to conduct based on short-read sequencing [150]. One way to resolve this is to leverage third-generation sequencing technology to characterize the structural variants, copy-number variants, and complex genome rearrangement in cancer genomes. The long-read sequencing also helps phasing genomic mutations in cancer tissues. For transcriptome analysis, long read sequencing is able to identify transcripts with aberrant structures [150]. On the other hand, haplotype information is naturally implied in the long reads. Graph-based pangenome representation shows advantages in the representation and visualization of genetic variations and haplotypes. In addition, the cancer somatic mutations are progressively accumulated with time, leaving dynamically-changed, complicated genome landscapes [148]. It is a promising direction to trace and investigate tumor evolution based on a pangenome model built from heterogeneous tumor samples. Leveraging modern sequencing technology and the graphical pangenome models, methods can be developed to better characterize somatic mutations and decode their functional roles in carcinogenesis [150].

8.2.2 Inter-specific Pangenome Models for Analyzing Sequence Data for PDX models

Patient-Derived Xenograft (PDX) Models are mouse models of cancer where the tumor tissue from a patient is implanted in an immunedeficient mouse model [151]. PDX models simulate the process of human tumorigenesis, allowing for investigating and perturbing the cancer genesis and development process *in vivo*. PDX models have been widely applied in biomedical study and for patient's treatment design [151]. However, the tissue collected from PDX models derived from both human and mouse. It is critical and challenging to discriminate sequence reads derived mouse and human to accurately identify somatic mutations and copy number variations. Previous results show that errors when discriminating human reads from mouse model lead to incorrect variant calls and fail to identify the true variants associated with the phenotype [151]. The inter-specific pangenome models is a potential solution for dealing with these issues. We can construct an inter-species pangenome model, integrating human and mouse genome assemblies in a graph and perform analysis based on such a pangenome model. However, due to the different karyotype, i.e. different number of chromosomes between human and mouse, the pangenome structure would be more complex than the intra-specific pangenome models. To simplify this issue, one possibility is to focus on the genomic regions of interests, leveraging the available annotation resources to extract and merge the homologous sequences between the human and mouse genome. A subgraph can be constructed from a pangenome model for each region of interests, e.g. genes. These genomic regions are defined by anchor sequences and are more stable identifying functional features in different mouse strains. We can then apply both the alignment-based methods or the kmer-based methods described in Chapter 5 to analyze the PDX sequence data from both host and tumor tissues altogether and perform variant call given the pangenome model.

8.2.3 Graphical Genome Browser

Traditional genome browsers utilize the linear reference genome and its coordinates to organize and visualize the annotations or sequence alignments. Visualizing multiple genome assemblies at the same time introduces complications for referring to and comparing homologies. The variation graph or compact De Bruijn graph involve complex graph topology and are non-planar to display in a 2D viewer. Previously, the multiple genome viewer (MGV) [109] has been developed to visualize, explore, and compare the sixteen mouse genomes released in [6]. In MGV, equivalent genomic features are highlighted by using vertical connectors among multiple genome tracks. Similar idea can be adopted to develop a graph-based pangenome browser with multiple panels. The anchor-based pangenome model is a series-parallel graph, which is planar, and allows for displaying genome features and comparing homologous sequences in a 2D genome browser. One can extracted all the parallel sequences in a specific subgraph, display the sequence context in multiple track, and draw the connections among homologous sequences. The other way is to show the graph topology with overlaid labels or colors to represent annotations or features on the graph. The read alignments can also be related to different paths on the graph and jointly analyzed. It is of great interest of the community and is feasible to develop such a graphical genome browser for future genomic study.

8.2.4 Pangenome-based Association Study

Genome-wide association studies (GWAS) have been widely applied in genetic study, biomedical and clinical research. However, conventional GWAS study focus on the SNPs identified relative to a single reference genome. As we mentioned previously, the choice of reference assemblies are largely arbitrary, lean to specific sample and the reference allele may not be the average of the represented population. This lead to bias to the downstream GWAS study. Instead, the pangenome model provide a comprehensive depiction of the genomic variations in the intra-specific population and also captures the complex variations such as structural variations, inversions or translocations. Incorporating the unbiased variation to the association study could potential improve the ability to identify the genetic variations associated with phenotypic differences. Similar ideas has been proposed to identify regions in the compressed De Bruijn graphs and apply machine learning methods to identify features (pangenomic regions) associated with phenotypical traits [152]. The CCGG probe database and kmer count matrix could serves as a resource for performing such association analysis. The structural variants, such as repetitive regions or indels are identified in the graph and these could also serves as features for association study.

REFERENCES

Jason A Reuter, Damek V Spacek, and Michael P Snyder. High-throughput sequencing technologies. *Molecular cell*, 58(4):586–597, 2015.

Deanna M Church, Valerie A Schneider, Tina Graves, Katherine Auger, Fiona Cunningham, Nathan Bouk, Hsiu-Chuan Chen, Richa Agarwala, William M McLaren, Graham RS Ritchie, et al. Modernizing reference genome assemblies. *PLoS biology*, 9(7), 2011.

Sara Ballouz, Alexander Dobin, and Jesse A Gillis. Is it time to change the reference genome? *Genome biology*, 20(1):1–9, 2019.

Computational pan-genomics: status, promises and challenges. *Briefings in bioinformatics*, 19(1):118–135, 2018.

Benedict Paten, Adam M Novak, Jordan M Eizenga, and Erik Garrison. Genome graphs and the evolution of genome inference. *Genome research*, 27(5):665–676, 2017.

Jingtao Lilue, Anthony G Doran, Ian T Fiddes, Monica Abrudan, Joel Armstrong, Ruth Bennett, William Chow, Joanna Collins, Stephan Collins, Anne Czechanski, et al. Sixteen diverse laboratory mouse reference genomes define strain-specific haplotypes and novel functional loci. *Nature genetics*, page 1, 2018.

Robert D Fleischmann, Mark D Adams, Owen White, Rebecca A Clayton, Ewen F Kirkness, Anthony R Kerlavage, Carol J Bult, Jean-Francois Tomb, Brian A Dougherty, Joseph M Merrick, et al. Whole-genome random sequencing and assembly of haemophilus influenzae rd. *science*, 269(5223):496–512, 1995.

André Goffeau, Bart G Barrell, Howard Bussey, Ronald W Davis, Bernard Dujon, Heinz Feldmann, Francis Galibert, Jörg D Hoheisel, Claude Jacq, Michael Johnston, et al. Life with 6000 genes. *Science*, 274(5287):546–567, 1996.

International Human Genome Sequencing Consortium et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860, 2001.

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

Mouse Genome Sequencing Consortium et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420(6915):520, 2002.

Jordan M Eizenga, Adam M Novak, Jonas A Sibbesen, Simon Heumos, Ali Ghaffaari, Glenn Hickey, Xian Chang, Josiah D Seaman, Robin Rounthwaite, Jana Ebler, et al. Pangenome graphs. *Annual review of genomics and human genetics*, 21:139–162, 2020.

Knut D Rand, Ivar Grytten, Alexander J Nederbragt, Geir O Storvik, Ingrid K Glad, and Geir K Sandve. Coordinates and intervals in graph-based reference genomes. *BMC bioinformatics*, 18(1):263, 2017.

Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

Aaron R Quinlan and Ira M Hall. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

Arash Bayat, Bruno Gaëta, Aleksandar Ignjatovic, and Sri Parameswaran. Improved vcf normalization for accurate vcf comparison. *Bioinformatics*, 33(7):964–970, 2017.

Deanna M Church, Valerie A Schneider, Karyn Meltz Steinberg, Michael C Schatz, Aaron R Quinlan, Chen-Shan Chin, Paul A Kitts, Bronwen Aken, Gabor T Marth, Michael M Hoffman, et al. Extending reference assembly models. *Genome biology*, 16(1):13, 2015.

Rong Chen and Atul J Butte. The reference human genome demonstrates high risk of type 1 diabetes and other disorders. In *Biocomputing 2011*, pages 231–242. World Scientific, 2011.

Alexander Dilthey, Charles Cox, Zamin Iqbal, Matthew R Nelson, and Gil McVean. Improved genome inference in the mhc using a population reference graph. *Nature genetics*, 47(6):682, 2015.

Alberto Magi, Romina D'Aurizio, Flavia Palombo, Ingrid Cifola, Lorenzo Tattini, Roberto Semeraro, Tommaso Pippucci, Betti Giusti, Giovanni Romeo, Rosanna Abbate, et al. Characterization and identification of hidden rare variants in the human genome. *BMC genomics*, 16(1):1–16, 2015.

Yury A Barbitoff, Igor V Bezdvornykh, Dmitrii E Polev, Elena A Serebryakova, Andrey S Glotov, Oleg S Glotov, and Alexander V Predeus. Catching hidden variation: systematic correction of reference minor allele annotation in clinical variant calling. *Genetics in Medicine*, 20(3):360–364, 2018.

Birte Kehr, Anna Helgadottir, Pall Melsted, Hakon Jonsson, Hannes Helgason, Adalbjörg Jonasdottir, Aslaug Jonasdottir, Asgeir Sigurdsson, Arnaldur Gylfason, Gisli H Halldorsson, et al. Diversity in non-repetitive human sequences not found in the reference genome. *Nature Genetics*, 49(4):588–593, 2017.

Jesper Eisfeldt, Gustaf Mårtensson, Adam Ameur, Daniel Nilsson, and Anna Lindstrand. Discovery of novel sequences in 1,000 swedish genomes. *Molecular biology and evolution*, 37(1):18–30, 2020.

Swapan Mallick, Heng Li, Mark Lipson, Iain Mathieson, Melissa Gymrek, Fernando Racimo, Mengyao Zhao, Niru Chennagiri, Susanne Nordenfelt, Arti Tandon, et al. The simons genome diversity project: 300 genomes from 142 diverse populations. *Nature*, 538(7624):201–206, 2016.

1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68, 2015.

Rachel M Sherman, Juliet Forman, Valentin Antonescu, Daniela Puiu, Michelle Daya, Nicholas Rafaels, Meher Preethi Boorgula, Sameer Chavan, Candelaria Vergara, Victor E Ortega, et al. Assembly of a pan-genome from deep sequencing of 910 humans of african descent. *Nature genetics*, 51(1):30–35, 2019.

Rachel M Sherman and Steven L Salzberg. Pan-genomics in the human genome era. *Nature Reviews Genetics*, 21(4):243–254, 2020.

Richard A Gibbs, John W Belmont, Paul Hardenbol, Thomas D Willis, FL Yu, HM Yang, Lan-Yang Ch'ang, Wei Huang, Bin Liu, Yan Shen, et al. The international hapmap project. 2003.

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

Jasmijn A Baaijens, Paola Bonizzoni, Christina Boucher, Gianluca Della Vedova, Yuri Pirola, Raffaella Rizzi, and Jouni Sirén. Computational graph pangenomics: a tutorial on data structures and their applications. *Natural Computing*, pages 1–28, 2022.

Parsoa Khorsand, Luca Denti, Human Genome Structural Variant Consortium, Paola Bonizzoni, Rayan Chikhi, and Fereydoun Hormozdiari. Comparative genome analysis using sample-specific string detection in accurate long reads. *Bioinformatics Advances*, 1(1):vbab005, 2021.

Carlos Alonso-Blanco, Jorge Andrade, Claude Becker, Felix Bemm, Joy Bergelson, Karsten M Borgwardt, Jun Cao, Eunyoung Chae, Todd M Dezwaan, Wei Ding, et al. 1,135 genomes reveal the global pattern of polymorphism in arabidopsis thaliana. *Cell*, 166(2):481–491, 2016.

Sydney Brenner and Jeffrey H Miller. *Brenner's encyclopedia of genetics*. Elsevier Science, 2014.

Gary A Churchill, David C Airey, Hooman Allayee, Joe M Angel, Alan D Attie, Jackson Beatty, William D Beavis, John K Belknap, Beth Bennett, Wade Berrettini, et al. The collaborative cross, a community resource for the genetic analysis of complex traits. *Nature genetics*, 36(11):1133–1137, 2004.

David W Threadgill and Gary A Churchill. Ten years of the collaborative cross. *Genetics*, 190(2):291–294, 2012.

Hyuna Yang, Jeremy R Wang, John P Didion, Ryan J Buus, Timothy A Bell, Catherine E Welsh, François Bonhomme, Alex Hon-Tsen Yu, Michael W Nachman, Jaroslav Pialek, et al. Subspecific origin and haplotype diversity in the laboratory mouse. *Nature genetics*, 43(7):648, 2011.

David L Aylor, William Valdar, Wendy Foulds-Mathes, Ryan J Buus, Ricardo A Verdugo, Ralph S Baric, Martin T Ferris, Jeff A Frelinger, Mark Heise, Matt B Frieman, et al. Genetic analysis of complex traits in the emerging collaborative cross. *Genome research*, 21(8):1213–1222, 2011.

Vivek M Philip, Greta Sokoloff, Cheryl L Ackert-Bicknell, Martin Striz, Lisa Branstetter, Melissa A Beckmann, Jason S Spence, Barbara L Jackson, Leslie D Galloway, Paul Barker, et al. Genetic analysis in the collaborative cross breeding population. *Genome research*, 21(8):1223–1238, 2011.

Daniel Bottomly, Martin T Ferris, Lauri D Aicher, Elizabeth Rosenzweig, Alan Whitmore, David L Aylor, Bart L Haagmans, Lisa E Gralinski, Birgit G Bradel-Tretheway, Janine T Bryan, et al. Expression quantitative trait loci for extreme host response to influenza a in pre-collaborative cross mice. *G3: Genes, Genomes, Genetics*, 2(2):213–221, 2012.

Sarah R Leist, Carolin Pilzner, Judith MA van den Brand, Leonie Dengler, Robert Geffers, Thijs Kuiken, Rudi Balling, Heike Kollmus, and Klaus Schughart. Influenza h3n2 infection of the collaborative cross founder strains reveals highly divergent host responses and identifies a unique phenotype in cast/eij mice. *BMC genomics*, 17(1):143, 2016.

Remco T Molenhuis, Hilgo Bruining, Myrna JV Brandt, Petra E Van Soldt, Hanifa J Abu-Toamih Atamni, J Peter H Burbach, Fuad A Iraqi, Richard F Mott, and Martien JH Kas. Modeling the quantitative nature of neurodevelopmental disorders using collaborative cross mice. *Molecular autism*, 9(1):63, 2018.

Kelly Orgel, Johanna M Smeekens, Ping Ye, Lauren Fotsch, Rishu Guo, Darla R Miller, Fernando Pardo-Manuel de Villena, A Wesley Burks, Martin T Ferris, and Michael D Kulis. Genetic diversity between mouse strains allows identification of the cc027/geniunc strain as an orally reactive model of peanut allergy. *Journal of Allergy and Clinical Immunology*, 143(3):1027–1037, 2019.

Shunping Huang, Chia-Yu Kao, Leonard McMillan, and Wei Wang. Transforming genomes using mod files with applications. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, pages 595–604, 2013.

James Holt, Shunping Huang, Leonard McMillan, and Wei Wang. Read annotation pipeline for high-throughput sequencing data. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, pages 605–612, 2013.

Shunping Huang, James Holt, Chia-Yu Kao, Leonard McMillan, and Wei Wang. A novel multi-alignment pipeline for high-throughput sequencing data. *Database*, 2014, 2014.

Ruiqiang Li, Yingrui Li, Hancheng Zheng, Ruibang Luo, Hongmei Zhu, Qibin Li, Wubin Qian, Yuanyuan Ren, Geng Tian, Jinxiang Li, et al. Building the sequence map of the human pan-genome. *Nature biotechnology*, 28(1):57, 2010.

Hervé Tettelin, David Riley, Ciro Cattuto, and Duccio Medini. Comparative genomics: the bacterial pan-genome. *Current opinion in microbiology*, 11(5):472–477, 2008.

Hervé Tettelin, Vega Masignani, Michael J Cieslewicz, Claudio Donati, Duccio Medini, Naomi L Ward, Samuel V Angiuoli, Jonathan Crabtree, Amanda L Jones, A Scott Durkin, et al. Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: implications for the microbial “pan-genome”. *Proceedings of the National Academy of Sciences*, 102(39):13950–13955, 2005.

Siyuan Zheng, Andrew D Cherniack, Ninad Dewal, Richard A Moffitt, Ludmila Danilova, Bradley A Murray, Antonio M Lerario, Tobias Else, Theo A Knijnenburg, Giovanni Ciriello, et al. Comprehensive pan-genomic characterization of adrenocortical carcinoma. *Cancer cell*, 29(5):723–736, 2016.

Justin S Hogg, Fen Z Hu, Benjamin Janto, Robert Boissy, Jay Hayes, Randy Keefe, J Christopher Post, and Garth D Ehrlich. Characterization and modeling of the haemophilus influenzae core and supragenomes based on the complete genomic sequences of rd and 12 clinical nontypeable strains. *Genome biology*, 8(6):1–18, 2007.

Tristan Lefébure and Michael J Stanhope. Evolution of the core and pan-genome of streptococcus: positive selection, recombination, and genome composition. *Genome biology*, 8(5):1–17, 2007.

Annika Jacobsen, Rene S Hendriksen, Frank M Aaresturp, David W Ussery, and Carsten Friis. The salmonella enterica pan-genome. *Microbial ecology*, 62(3):487–504, 2011.

Zhemin Zhou, Inge Lundstrøm, Alicia Tran-Dien, Sebastián Duchêne, Nabil-Fareed Alikhan, Martin J Sergeant, Gemma Langridge, Anna K Fotakis, Satheesh Nair, Hans K Stenøien, et al. Pan-genome analysis of ancient and modern salmonella enterica demonstrates genomic stability of the invasive para c lineage for millennia. *Current Biology*, 28(15):2420–2428, 2018.

Hervé Tettelin and Duccio Medini. The pangenome: Diversity, dynamics and evolution of genomes, 2020.

Ruiqiang Li, Yingrui Li, Hancheng Zheng, Ruibang Luo, Hongmei Zhu, Qibin Li, Wubin Qian, Yuanyuan Ren, Geng Tian, Jinxiang Li, et al. Building the sequence map of the human pan-genome. *Nature biotechnology*, 28(1):57–63, 2010.

Zhiqiang Hu, Chen Sun, Kuang-chen Lu, Xixia Chu, Yue Zhao, Jinyuan Lu, Jianxin Shi, and Chaochun Wei. Eupan enables pan-genome studies of a large number of eukaryotic genomes. *Bioinformatics*, 33(15):2408–2409, 2017.

Shoshana Marcus, Hayan Lee, and Michael C Schatz. Splitmem: a graphical algorithm for pan-genome analysis with suffix skips. *Bioinformatics*, 30(24):3476–3483, 2014.

Uwe Baier, Timo Beller, and Enno Ohlebusch. Graphical pan-genome analysis with compressed suffix trees and the burrows–wheeler transform. *Bioinformatics*, 32(4):497–504, 2016.

Pascal Lapierre and J Peter Gogarten. Estimating the size of the bacterial pan-genome. *Trends in genetics*, 25(3):107–110, 2009.

Alexander Herbig, Günter Jäger, Florian Battke, and Kay Nieselt. Genomering: alignment visualization based on supergenome coordinates. *Bioinformatics*, 28(12):i7–i15, 2012.

Fabian Gärtner, Christian Höner zu Siederdissen, Lydia Müller, and Peter F Stadler. Coordinate systems for supergenomes. *Algorithms for Molecular Biology*, 13(1):1–19, 2018.

Phillip EC Compeau, Pavel A Pevzner, and Glenn Tesler. How to apply de bruijn graphs to genome assembly. *Nature biotechnology*, 29(11):987–991, 2011.

Antoine Limasset, Bastien Cazaux, Eric Rivals, and Pierre Peterlongo. Read mapping on de bruijn graphs. *BMC bioinformatics*, 17(1):1–12, 2016.

Kazuhiko Takamizawa, Takao Nishizeki, and Nobuji Saito. Linear-time computability of combinatorial problems on series-parallel graphs. *Journal of the ACM (JACM)*, 29(3):623–641, 1982.

Korbinian Schneeberger, Jörg Hagmann, Stephan Ossowski, Norman Warthmann, Sandra Gesing, Oliver Kohlbacher, and Detlef Weigel. Simultaneous alignment of short reads against multiple genomes. *Genome biology*, 10(9):1–12, 2009.

Michael Burrows and David Wheeler. A block-sorting lossless data compression algorithm. In *Digital SRC Research Report*. Citeseer, 1994.

Lin Huang, Victoria Popic, and Serafim Batzoglou. Short read alignment with populations of genomes. *Bioinformatics*, 29(13):i361–i370, 2013.

Sorina Maciuca, Carlos del Ojo Elias, Gil McVean, and Zamin Iqbal. A natural encoding of genetic variation in a burrows-wheeler transform to enable mapping and genome inference. In *International Workshop on Algorithms in Bioinformatics*, pages 222–233. Springer, 2016.

Richard Durbin. Efficient haplotype matching and storage using the positional burrows–wheeler transform (pbwt). *Bioinformatics*, 30(9):1266–1272, 2014.

Adam M Novak, Erik Garrison, and Benedict Paten. A graph extension of the positional burrows–wheeler transform and its applications. *Algorithms for Molecular Biology*, 12(1):1–12, 2017.

Jouni Sirén, Erik Garrison, Adam M Novak, Benedict Paten, and Richard Durbin. Haplotype-aware graph indexes. *Bioinformatics*, 36(2):400–407, 2020.

Birte Kehr, Kathrin Trappe, Manuel Holtgrewe, and Knut Reinert. Genome alignment with graph data structures: a comparison. *BMC bioinformatics*, 15(1):1–20, 2014.

Daehwan Kim, Joseph M Paggi, Chanhee Park, Christopher Bennett, and Steven L Salzberg. Graph-based genome alignment and genotyping with hisat2 and hisat-genotype. *Nature biotechnology*, 37(8):907–915, 2019.

Mikko Rautiainen and Tobias Marschall. Graphaligner: rapid and versatile sequence-to-graph alignment. *Genome biology*, 21(1):1–28, 2020.

G Marcais and C Kingsford. Jellyfish: A fast k-mer counter. *Tutorialis e Manuais*, 1:1–8, 2012.

Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Shilpa Garg, Charles Markello, Michael F Lin, et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology*, 36(9):875–879, 2018.

Goran Rakocevic, Vladimir Semenyuk, Wan-Ping Lee, James Spencer, John Browning, Ivan J Johnson, Vladan Arsenijevic, Jelena Nadj, Kaushik Ghose, Maria C Suciu, et al. Fast and accurate genomic analyses using genome graphs. *Nature genetics*, 51(2):354–362, 2019.

Daniel Valenzuela, Tuukka Norri, Niko Välimäki, Esa Pitkänen, and Veli Mäkinen. Towards pan-genome read alignment to improve variation calling. *BMC genomics*, 19(2):123–130, 2018.

Ilia Minkin, Son Pham, and Paul Medvedev. Twopaco: An efficient algorithm to build the compacted de bruijn graph from many complete genomes. *Bioinformatics*, 33(24):4024–4032, 2017.

Hannes P Eggertsson, Hakon Jonsson, Snaedis Kristmundsdottir, Eirikur Hjartarson, Birte Kehr, Gisli Masson, Florian Zink, Kristjan E Hjorleifsson, Aslaug Jonasdottir, Adalbjorg Jonasdottir, et al. Graphtyper enables population-scale genotyping using pangenome graphs. *Nature genetics*, 49(11):1654, 2017.

Chad Laing, Cody Buchanan, Eduardo N Taboada, Yongxiang Zhang, Andrew Kropinski, Andre Villegas, James E Thomas, and Victor PJ Gannon. Pan-genome sequence analysis using panseq: an online tool for the rapid analysis of core and accessory genomic regions. *BMC bioinformatics*, 11(1):1–14, 2010.

Zhongqu Duan, Yuyang Qiao, Jinyuan Lu, Huimin Lu, Wenmin Zhang, Fazhe Yan, Chen Sun, Zhiqiang Hu, Zhen Zhang, Guichao Li, et al. Hupan: a pan-genome analysis pipeline for human genomes. *Genome biology*, 20(1):1–11, 2019.

Siavash Sheikhizadeh, M Eric Schranz, Mehmet Akdel, Dick de Ridder, and Sandra Smit. Pantools: representation, storage and exploration of pan-genomic data. *Bioinformatics*, 32(17):i487–i493, 2016.

Zamin Iqbal, Mario Caccamo, Isaac Turner, Paul Flicek, and Gil McVean. De novo assembly and genotyping of variants using colored de bruijn graphs. *Nature genetics*, 44(2):226–232, 2012.

Martin D Muggli, Alexander Bowe, Noelle R Noyes, Paul S Morley, Keith E Belk, Robert Raymond, Travis Gagie, Simon J Puglisi, and Christina Boucher. Succinct colored de bruijn graphs. *Bioinformatics*, 33(20):3181–3187, 2017.

Guillaume Holley and Páll Melsted. Bifrost: highly parallel construction and indexing of colored and compacted de bruijn graphs. *Genome biology*, 21(1):1–20, 2020.

Joel Armstrong, Glenn Hickey, Mark Diekhans, Ian T Fiddes, Adam M Novak, Alden Deran, Qi Fang, Duo Xie, Shaohong Feng, Josefin Stiller, et al. Progressive cactus is a multiple-genome aligner for the thousand-genome era. *Nature*, 587(7833):246–251, 2020.

Christine Jandrasits, Piotr W Dabrowski, Stephan Fuchs, and Bernhard Y Renard. seq-seq-pan: Building a computational pan-genome data structure on whole genome alignment. *BMC genomics*, 19(1):1–12, 2018.

Evan Biederstedt, Jeffrey C Oliver, Nancy F Hansen, Aarti Jajoo, Nathan Dunn, Andrew Olson, Ben Busby, and Alexander T Dilthey. Novograph: human genome graph construction from multiple long-read de novo assemblies. *F1000Research*, 7, 2018.

Heng Li, Xiaowen Feng, and Chong Chu. The design and construction of reference pangenome graphs with minigraph. *Genome biology*, 21(1):1–19, 2020.

Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357, 2012.

Heng Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

U. Manber and G. Myers. Suffix arrays: A new method for on-line string searches. 22(5):935–948, July 2006.

P. Ferragina and G. Manzini. An experimental study of an opportunistic index. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, SODA '01, pages 269–278, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 390–398, Redondo Beach, CA, USA, 2000.

J. Holt and L. McMillan. Constructing burrows-wheeler transforms of large string collections via merging. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '14, pages 464–471, New York, NY, USA, 2014. ACM.

James Holt and Leonard McMillan. Merging of multi-string bwts with applications. *Bioinformatics*, 30(24):3524–3531, 2014.

Markus J Bauer, Anthony J Cox, and Giovanna Rosone. Lightweight algorithms for constructing and inverting the bwt of string collections. *Theoretical Computer Science*, 483:134–148, 2013.

Anthony J Cox, Markus J Bauer, Tobias Jakobi, and Giovanna Rosone. Large-scale compression of genomic sequence databases with the burrows–wheeler transform. *Bioinformatics*, 28(11):1415–1419, 2012.

Jouni Sirén, Niko Välimäki, and Veli Mäkinen. Indexing graphs for path queries with applications in genome research. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(2):375–388, 2014.

Ali Ghaffaari and Tobias Marschall. Fully-sensitive seed finding in sequence graphs using a hybrid index. *Bioinformatics*, 35(14):i81–i89, 2019.

Kavya Vaddadi, Rajgopal Srinivasan, and Naveen Sivadasan. Read mapping on genome variation graphs. In *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

Sai Chen, Peter Krusche, Egor Dolzhenko, Rachel M Sherman, Roman Petrovski, Felix Schlesinger, Melanie Kirsche, David R Bentley, Michael C Schatz, Fritz J Sedlazeck, et al. Paragraph: a graph-based structural variant genotyper for short-read sequence data. *Genome biology*, 20(1):1–13, 2019.

Glenn Hickey, David Heller, Jean Monlong, Jonas A Sibbesen, Jouni Sirén, Jordan Eizenga, Eric T Dawson, Erik Garrison, Adam M Novak, and Benedict Paten. Genotyping structural variants in pangenome graphs using the vg toolkit. *Genome biology*, 21(1):1–17, 2020.

Jonas Andreas Sibbesen, Lasse Maretty, and Anders Krogh. Accurate genotyping across variant classes and lengths using variant graphs. *Nature genetics*, 50(7):1054–1059, 2018.

Dirk D Dolle, Zhicheng Liu, Matthew Cotten, Jared T Simpson, Zamin Iqbal, Richard Durbin, Shane A McCarthy, and Thomas M Keane. Using reference-free compressed data structures to analyze sequencing reads from thousands of human genomes. *Genome research*, 27(2):300–309, 2017.

Helga Thorvaldsdóttir, James T Robinson, and Jill P Mesirov. Integrative genomics viewer (igv): high-performance genomics data visualization and exploration. *Briefings in bioinformatics*, 14(2):178–192, 2013.

Donna Karolchik, Robert Baertsch, Mark Diekhans, Terrence S Furey, Angie Hinrichs, YT Lu, Krishna M Roskin, Matt Schwartz, Charles W Sugnet, Daryl J Thomas, et al. The ucsc genome browser database. *Nucleic acids research*, 31(1):51–54, 2003.

Joel E Richardson, Richard M Baldarelli, and Carol J Bult. Multiple genome viewer (mgv): a new tool for visualization and comparison of multiple annotated genomes. *Mammalian Genome*, pages 1–11, 2021.

Ryan R Wick, Mark B Schultz, Justin Zobel, and Kathryn E Holt. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, 31(20):3350–3352, 2015.

Giorgio Gonnella, Niklas Niehus, and Stefan Kurtz. Gfaviz: flexible and interactive visualization of gfa sequence graphs. *Bioinformatics*, 35(16):2853–2855, 2019.

Wolfgang Beyer, Adam M Novak, Glenn Hickey, Jeffrey Chan, Vanessa Tan, Benedict Paten, and Daniel R Zerbino. Sequence tube maps: making graph genomes intuitive to commuters. *Bioinformatics*, 35(24):5318, 2019.

Toshiyuki T Yokoyama, Yoshitaka Sakamoto, Masahide Seki, Yutaka Suzuki, and Masahiro Kasahara. Momi-g: modular multi-scale integrated genome graph browser. *BMC bioinformatics*, 20(1):1–14, 2019.

Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110, 2016.

Bastien Llamas, Giuseppe Narzisi, Valerie Schneider, Peter A Audano, Evan Biederstedt, Lon Blauvelt, Peter Bradbury, Xian Chang, Chen-Shan Chin, Arkarachai Fungtammasan, et al. A strategy for building and using a human reference pangenome. *F1000Research*, 8(1751):1751, 2019.

Steve S Ho, Alexander E Urban, and Ryan E Mills. Structural variation in the sequencing era. *Nature Reviews Genetics*, pages 1–19, 2019.

Geòrgia Escaramís, Elisa Docampo, and Raquel Rabionet. A decade of structural variants: description, history and methods to detect structural variation. *Briefings in functional genomics*, 14(5):305–314, 2015.

Can Alkan, Bradley P Coe, and Evan E Eichler. Genome structural variation discovery and genotyping. *Nature Reviews Genetics*, 12(5):363–376, 2011.

Anuj Srivastava, Andrew P Morgan, Maya L Najarian, Vishal Kumar Sarsani, J Sebastian Sigmon, John R Shorter, Anwica Kashfeen, Rachel C McMullan, Lucy H Williams, Paola Giusti-Rodríguez, et al. Genomes of the mouse collaborative cross. *Genetics*, 206(2):537–556, 2017.

John R Shorter, Maya L Najarian, Timothy A Bell, Matthew Blanchard, Martin T Ferris, Pablo Hock, Anwica Kashfeen, Kathryn E Kirchoff, Colton L Linnertz, J Sebastian Sigmon, et al. Whole genome sequencing and progress toward full inbreeding of the mouse collaborative cross population. *G3: Genes, Genomes, Genetics*, 9(5):1303–1311, 2019.

Gregory E Sims, Se-Ran Jun, Guohong A Wu, and Sung-Hou Kim. Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions. *Proceedings of the National Academy of Sciences*, 106(8):2677–2682, 2009.

Arian FA Smit, Robert Hubley, and P Green. Repeatmasker, 1996.

Rhoda J Kinsella, Andreas Kähäri, Syed Haider, Jorge Zamora, Glenn Proctor, Giulietta Spudich, Jeff Almeida-King, Daniel Staines, Paul Derwent, Arnaud Kerhornou, et al. Ensembl biomarts: a hub for data retrieval across taxonomic space. *Database*, 2011, 2011.

Benjamin A Taylor, Christopher Wnek, Brett S Kotlus, Nicholette Roemer, Tammy MacTaggart, and Sandra J Phillips. Genotyping new bxd recombinant inbred mouse strains and comparison of bxd and consensus maps. *Mammalian genome*, 10(4):335–348, 1999.

Collaborative Cross Consortium. The genome architecture of the collaborative cross mouse genetic reference population. *Genetics*, 190(2):389–401, 2012.

Robert H Waterston, Kerstin Lindblad-Toh, Ewan Birney, Jane Rogers, Josep F Abril, Pankaj Agarwal, Richa Agarwala, Rachel Ainscough, Marina Alexandersson, Peter An, et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420(6915):520–562, 2002.

Deanna M Church, Leo Goodstadt, LaDeana W Hillier, Michael C Zody, Steve Goldstein, Xinwe She, Carol J Bult, Richa Agarwala, Joshua L Cherry, Michael DiCuccio, et al. Lineage-specific biology revealed by a finished genome assembly of the mouse. *PLoS biology*, 7(5), 2009.

Steven C Munger, Narayanan Raghupathy, Kwangbom Choi, Allen K Simons, Daniel M Gatti, Douglas A Hinerfeld, Karen L Svenson, Mark P Keller, Alan D Attie, Matthew A Hibbs, et al. Rna-seq alignment to individualized genomes improves transcript abundance estimates in multiparent populations. *Genetics*, 198(1):59–73, 2014.

Jacob L Mueller, Shantha K Mahadevaiah, Peter J Park, Peter E Warburton, David C Page, and James MA Turner. The mouse x chromosome is enriched for multicopy testis genes showing postmeiotic expression. *Nature genetics*, 40(6):794–799, 2008.

Andrew P Morgan, Daniel M Gatti, Maya L Najarian, Thomas M Keane, Raymond J Galante, Allan I Pack, Richard Mott, Gary A Churchill, and Fernando Pardo-Manuel de Villena. Structural variation shapes the landscape of recombination in mouse. *Genetics*, 206(2):603–619, 2017.

Andrew P Morgan and Catherine E Welsh. Informatics resources for the collaborative cross and related mouse populations. *Mammalian Genome*, 26(9):521–539, 2015.

Catherine E Welsh, Darla R Miller, Kenneth F Manly, Jeremy Wang, Leonard McMillan, Grant Morahan, Richard Mott, Fuad A Iraqi, David W Threadgill, and Fernando Pardo-Manuel de Villena. Status and access to the collaborative cross population. *Mammalian Genome*, 23(9):706–712, 2012.

Anwica Kashfeen, Harper B Fauni, Timothy A Bell, Fernando Pardo-Manuel de Villena, and Leonard McMillan. Elite: Efficiently locating insertions of transposable elements. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 183–189, 2019.

Megan K Mulligan, Timothy Abreo, Sarah M Neuner, Cory Parks, Christine E Watkins, M Trevor Houseal, Thomas M Shapaker, Michael Hook, Haiyan Tan, Xusheng Wang, et al. Identification of a functional non-coding variant in the gaba. 2019.

Jeremy L Peirce, Lu Lu, Jing Gu, Lee M Silver, and Robert W Williams. A new set of bxd recombinant inbred lines from advanced intercross populations in mice. *BMC genetics*, 5(1):7, 2004.

VM Philip, S Duvvuru, B Gomero, TA Ansah, CD Blaha, MN Cook, KM Hamre, WR Lariviere, DB Matthews, G Mittleman, et al. High-throughput behavioral phenotyping in the expanded panel of bxd recombinant inbred strains. *Genes, Brain and Behavior*, 9(2):129–159, 2010.

Karen L Svenson, Daniel M Gatti, William Valdar, Catherine E Welsh, Riyan Cheng, Elissa J Chesler, Abraham A Palmer, Leonard McMillan, and Gary A Churchill. High-resolution genetic mapping using the mouse diversity outbred population. *Genetics*, 190(2):437–447, 2012.

Silong Sun, Yingsi Zhou, Jian Chen, Junpeng Shi, Haiming Zhao, Hainan Zhao, Weibin Song, Mei Zhang, Yang Cui, Xiaomei Dong, et al. Extensive intraspecific gene order and gene structural variations between mo17 and other maize genomes. *Nature genetics*, 50(9):1289–1295, 2018.

Stefan Kurtz, Apurva Narechania, Joshua C Stein, and Doreen Ware. A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes. *BMC genomics*, 9(1):1–18, 2008.

Jana Ebler, Wayne E Clarke, Tobias Rausch, Peter A Audano, Torsten Houwaart, Jan Korbel, Evan E Eichler, Michael C Zody, Alexander T Dilthey, and Tobias Marschall. Pangenome-based genome inference. *BioRxiv*, 2020.

Ian J. Davis. A fast radix sort. *The computer journal*, 35(6):636–642, 1992.

Hang Su, Ziwei Chen, Jaytheert Rao, Maya Najarian, John R Shorter, Fernando Pardo-Manuel de Villena, and Leonard McMillan. The collaborative cross graphical genome. *bioRxiv*, page 858142, 2019.

Harris A Lewin, Gene E Robinson, W John Kress, William J Baker, Jonathan Coddington, Keith A Crandall, Richard Durbin, Scott V Edwards, Félix Forest, M Thomas P Gilbert, et al. Earth biogenome project: Sequencing life for the future of life. *Proceedings of the National Academy of Sciences*, 115(17):4325–4333, 2018.

Adam Ameur. Goodbye reference, hello genome graphs. *Nature biotechnology*, 37(8):866–868, 2019.

Hang Su, Ziwei Chen, Maya L Najarian, Martin T Ferris, Fernando Pardo-Manuel de Villena, and Leonard McMillan. A k-mer query tool for assessing population diversity in pangenomes. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–9, 2021.

Erik Garrison and Gabor Marth. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*, 2012.

Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology*, 19(1A):A68, 2015.

Lucy R Yates and Peter J Campbell. Evolution of the cancer genome. *Nature Reviews Genetics*, 13(11):795–806, 2012.

David A Wheeler, Maithreyan Srinivasan, Michael Egholm, Yufeng Shen, Lei Chen, Amy McGuire, Wen He, Yi-Ju Chen, Vinod Makhijani, G Thomas Roth, et al. The complete genome of an individual by massively parallel dna sequencing. *nature*, 452(7189):872–876, 2008.

Yoshitaka Sakamoto, Sarun Sereewattanawoot, and Ayako Suzuki. A new era of long-read sequencing for cancer genomics. *Journal of human genetics*, 65(1):3–10, 2020.

Xing Yi Woo, Anuj Srivastava, Joel H Graber, Vinod Yadav, Vishal Kumar Sarsani, Al Simons, Glen Beane, Stephen Grubb, Guruprasad Ananda, Rangjiao Liu, et al. Genomic data analysis workflows for tumors from patient-derived xenografts (pdxs): challenges and guidelines. *BMC medical genomics*, 12(1):1–19, 2019.

Buwani Manuweera, Joann Mudge, Indika Kahanda, Brendan Mumey, Thiruvarangan Ramaraj, and Alan Cleary. Pangenome-wide association studies with frequented regions. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 627–632, 2019.