Inference of 3D Structure of Diploid Chromosomes

by

Lawrence Sun

S.B. Electrical Engineering & Computer Science, Massachusetts Institute of Technology (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author . . . .
Department of Electrical Engineering and Computer Science
May 22, 2018

Certified by . . . .
Caroline Uhler
Assistant Professor
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Inference of 3D Structure of Diploid Chromosomes

by

Lawrence Sun

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2018, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

The spatial organization of DNA in the cell nucleus plays an important role for gene regulation, DNA replication, and genomic integrity. Through the development of chromosome capture experiments (such as 3C, 4C, Hi-C) it is now possible to obtain the contact frequencies of the DNA at the whole-genome level. In this thesis, we study the problem of reconstructing the 3D organization of the genome from whole-genome contact frequencies. A standard approach is to transform the contact frequencies into noisy distance measurements and then apply semidefinite programming (SDP) formulations to obtain the 3D configurations. However, neglected in such reconstructions is the fact that most eukaryotes including humans are diploid and therefore contain two (from the available data) indistinguishable copies of each genomic locus. Due to this, the standard approach performs very poorly on diploid organisms. We prove that the 3D organization of the DNA is not identifiable from exclusively chromosome capture data for diploid organisms. In fact, there are infinitely many solutions even in the noise-free setting. We then discuss various additional biologically relevant constraints (including distances between neighboring genomic loci and to the nucleus center or higher-order interactions). Under these conditions we prove there are finitely many solutions and conjecture we in fact have identifiability. Finally, we provide SDP formulations for computing the 3D embedding of the DNA with these additional constraints and show that we can recover the true 3D embedding with high accuracy even under noise.

Thesis Supervisor: Caroline Uhler
Title: Assistant Professor

Acknowledgments

I would like to thank my advisor Caroline Uhler for the tremendous amount of help throughout this year on my thesis. She proposed a very interesting problem to work on and provided valuable input to aid in working on this problem.

I would also like to thank Kaie Kubjas for working jointly with me under Caroline on this problem. Working together we were able to tackle many seemingly intractable problems by appealing to algebraic geometry.

Finally, I would like to thank all the members in my research group for listening to my presentations and providing feedback.

Contents

List of Figures

Chapter 1

Introduction

The genome (for example, in humans) is formed out of chromosomes. Chromosomes within a cell nucleus form a 3D structure, where the organization of chromosomes affect many mechanisms such as gene regulation, DNA replication, epigenetic modification and maintenance of gene stability [2]. These factors make understanding the 3D structure of chromosomes a very important problem.

Thus the fact that genes lie in 3D space explains why genome sequencing is not sufficient to determine the 3D structure of chromosomes. The question then becomes: how can we determine the 3D structure? The first part of the answers lies in a technology Hi-C which computes a *contact frequency* matrix between genomes [4]. Specifically, the contact frequency matrix f has element f_{ij} indicating the number of times genes i and j interact with each other. We then expect two genes i, j to be close together if and only if their contact frequency is high. We can then estimate the distances d_{ij} between genes to satisfy [4]:

$$d_{ij} \propto \begin{cases} (1/f_{ij})^{\alpha} \text{ if } f_{ij} > 0 \\ \infty \text{ otherwise} \end{cases}$$

where α is a constant called the *conversion factor*, which is not necessarily known beforehand because it changes based on the resolution of the data [10]. This formula captures the intuition that the closer together genes are, the more often they inter-

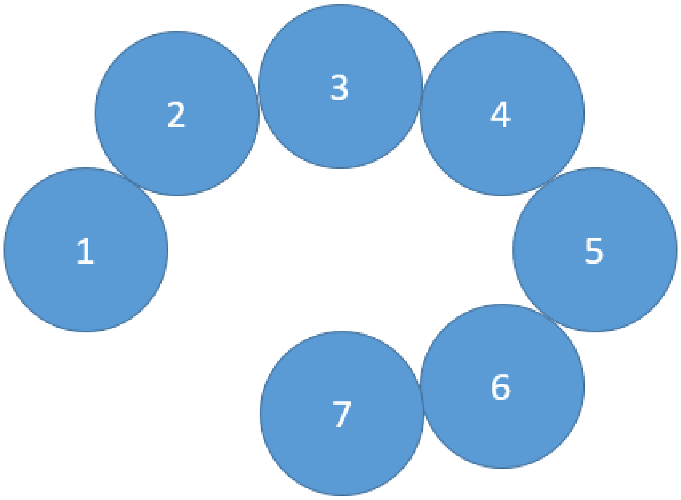

Figure 1-1: An example of a configuration of genes in 2D space.

act. Once we have these distance estimates, we need to uncover the true distances (because the estimated distances are noisy). This is most often resolved by phrasing the problem as an optimization problem, i.e. how close can we get to the estimated distances assuming we are in a 3 dimensional coordinate system.

One currently un-addressed problem is the case of diploid chromosomes, meaning there are two copies of each chromosome. All the work summarized above assumes there is only one copy of each chromosome, which is the monoploid case. This is not true in the case of the human genome: we possess two copies of each chromosome. Because of this, Hi-C data will give us contact frequencies which describe the combination of contact frequencies between the pairs of chromosomes. Intuitively this should still be solvable: we have $\binom{m}{2}$ constraints and $6m$ variables when there are m pairs of chromosomes present ($m = 23$ for the human genome). So despite having less information than in the monoploid case, we still can hope to recover the original coordinates of each pair of chromosomes. This problem going forward is the focus of our research.

1.1 Goals

Ultimately, we wish to determine the spacial configuration of diploid chromosomes from observations. The most basic data we can utilize is Hi-C data, which gives us contract frequencies between genes. However, we can add more data if it is necessary to uniquely determine the organization of genes.

We then have two main goals:

1. Identifiability: We would like to find a set of constraints generated from observed data which uniquely determines the organization of genes. Proving uniqueness would be the best case scenario.

2. Solvability: Simply knowing a unique solution exists is not sufficient because we must be able to solve the optimization problem. In many cases optimization problems are NP-complete so we need to devise an efficient algorithm in this case. The algorithm must also be robust because we cannot rely measured data to be exact, there will always be error.

1.2 Thesis Roadmap

The remainder of the thesis is organized as follows:

Chapter 2 goes over past work in this field which motivates many of the ideas later in this thesis and formally states the problem we will be discussing. In Chapter 3 we prove Hi-C contact frequencies are not sufficient for uniqueness and provide a construction of all possible solutions.

Starting in Chapter 4 we add constraints beyond those derived from Hi-C frequencies. We propose new constraints based on the beads on a string model. We then explore how this influences our problem and provide a proof of uniqueness in 2D while demonstrating in 3D we still do not have uniqueness.

In Chapter 5 we examine two possible new sets of constraints. Under these new constraints we prove there exist finitely many solutions and conjecture we in fact have uniqueness.

In Chapter 6 we present an algorithm to solve the constraint systems build up in this thesis. Using simulated data in small configurations we demonstrate that we can extract the true solution with high accuracy even under noise.

Chapter 2

Problem Introduction

First we discuss past work done in this field. The state of the art method is a system ChromSDE which solves for the 3D configuration of monoploid genes given Hi-C data. The algorithm is efficient and resistant to noise. However, the methods do not directly generalize to diploid chromosomes. Nevertheless, we utilize many similar ideas to the work done here so they are important to discuss.

2.1 Kernel Matrix

The first key insight to ChromSDE is to work with the *kernel matrix.* In general, given d-dimensional vectors $v_1, v_2, ..., v_m$, the kernel matrix for the vectors K is an $n \times n$ matrix where $K_{ij} = v_i^T \cdot v_j$. This is also referred to as the *Gram matrix.* The kernel matrix can alternatively be written as the product of a $m \times d$ matrix and its transpose, specifically:

$$K = \begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ \vdots & & \\ - & v_m & - \end{bmatrix} \cdot \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_m \\ | & | & & | \end{bmatrix}$$

where we simply insert the coordinates vectors as the rows/columns of our matrices. Because K is the product of these matrices, we know it has rank at most d. It is also a classical result that the kernel matrix must be positive semi-definite as a consequence of the factorization we give above. The reason why we work with the kernel matrix is because it helps give us *identifiability*: rotations, reflections and translations of the original coordinates yield infinitely many different configurations but all sharing the same distances between genomic regions, but if we center the kernel matrix then all these transformations do not change the kernel matrix. This in turn gives us hope of a unique solution for the Gram matrix.

2.2 Semi-Definite Programming

The next step in ChromSDE is to find some kernel matrix K which is close to our approximate distances. Specifically, suppose d_{ij} is the approximate distance between genes i and j given by the contact frequencies. We then specify a semi-definite program (SDP) where the kernel matrix K_{ij} adheres to the approximate distances d_{ij} as closely as possible; specifically we have:

$$d_{ij}^2 = \|v_i - v_j\|_2^2 = \|v_i\|_2^2 - 2v_i \cdot v_j + \|v_j\|_2^2 = K_{ii} - 2K_{ij} + K_{jj}$$

In the noiseless case, we'd like to find a rank 3 matrix K which satisfies $d_{ij}^2 = K_{ii} - 2K_{ij} + K_{jj}$ for all pairs i, j. However, this is a non-convex optimzation problem due to rank constraints being non-convex. Thus we perform a SDP relaxation and search for general positive semi-definite matrices K satisfying the constraints with minimal trace. Because trace is an L^1 norm on the necessarily non-negative eigenvalues of the matrix, this tends to force many of the eigenvalues of the solution to be 0, giving us a low rank solution. This is also now a convex optimization problem, so we can solve for the (unique) solution efficiently. It can be shown that for sufficiently large configurations and no noise, the solution to the SDP will in fact be equal to the true kernel matrix K. [9]

To adapt to noise, we essentially add slack to all the constraints. For each constraint of the form $f_i(K) = C_i$, we add a term of $(C_i - f_i(K))^2$ to our objective function (which is currently simply the trace of the matrix). This way K is strongly incentivized to follow the constraints, but if it is not desired, e.g. there is measurement error on C_i, the solution can stray from the constraint. This addition makes the method very robust to noise.

The key to this approach is that even in the presence of noise the SDP will return a matrix of rank mostly 3; to be precise, the majority of the mass of the matrix is concentrated on the first 3 eigenvalues. Let the top 3 eigenvalues of K be $\lambda_1, \lambda_2, \lambda_3$ with corresponding eigenvectors ν_1, ν_2, ν_3. Then letting

$$v_i = \begin{bmatrix} \sqrt{\lambda_1}\nu_{1i} & \sqrt{\lambda_2}\nu_{2i} & \sqrt{\lambda_3}\nu_{3i} \end{bmatrix}$$

yields a very high quality approximation of the original coordinates of the genes. When the matrix is correct, this will perfectly re-construct the original solution up to orthogonal transformations.

It should be noted that another powerful part of ChromSDE is its ability to adapt to different values of α. Specifically, the authors of the paper lay out techniques to search for the correct value of α based on evaluating metrics on the 3D coordinates generated by the previously described algorithm. They then search for the α value which scores the highest on their metrics.

Overall ChromSDE is a very powerful method for determining 3D structure of genes. However, as mentioned before, the approach does not perfectly generalize to diploid chromosomes. In particular, based on some preliminary numerical experiments we find that if we mimic their approach of performing an SDP relaxation, the resulting kernel matrix has rank approximately 5 rather than 3, implying the resulting method of taking eigenvectors as coordinates for our vectors will result in a very poor approximation.

2.3 Diploid Case

We now give a formal presentation of our problem. Let $f \in \mathbb{R}_{\geq 0}^{m \times m}$ be a Hi-C interaction matrix, where f_{x_i,x_j} measures the interaction strength between genomic regions with coordinates x_i and x_j in $\mathbb{R}^2$ or $\mathbb{R}^3$. Let $d \in \mathbb{R}^{m \times m}$ be the pairwise distance matrix, where d_{x_i,x_j} is the distance between x_i and x_j. By [4], the relationship between the Hi-C interaction matrix and the pairwise distance matrix is

$$d_{x_i,x_j} = \begin{cases} (1/f_{x_i,x_j})^{\alpha} & \text{if } f_{x_i,x_j} > 0 \\ \infty & \text{otherwise} \end{cases} \tag{2.1}$$

where we ignore the earlier discussed constant of proportionality because we are interested in the spacial configuration, the absolute distances are not critical. A cell has two copies of each chromosome and gene. We denote the coordinates corresponding to the two genomic regions in a homologue pair by x_i and y_i. In practice, only a combination of true interaction strengths $f_{x_i,x_j}, f_{x_i,y_j}, f_{y_i,x_j}$ and f_{y_i,y_j} can be measured. We denote it by F_{ij}. We assume the following relationship between the measured interactions and the true interactions

$$\frac{1}{F_{ij}} = \frac{1}{f_{x_i,x_j}} + \frac{1}{f_{x_i,y_j}} + \frac{1}{f_{y_i,x_j}} + \frac{1}{f_{y_i,y_j}}.$$

Applying (2.1) gives

$$\frac{1}{F_{ij}} = (d_{x_i,x_j})^{1/\alpha} + (d_{x_i,y_j})^{1/\alpha} + (d_{y_i,x_j})^{1/\alpha} + (d_{y_i,y_j})^{1/\alpha}.$$

Defining $D_{ij} := 1/F_{ij}$ and setting $\alpha = \frac{1}{2}$ (which is often the case for many biological systems such as equilibrium globules [4]) gives

$$D_{ij} = (d_{x_i,x_j})^2 + (d_{x_i,y_j})^2 + (d_{y_i,x_j})^2 + (d_{y_i,y_j})^2,$$

or equivalently

$$D_{ij} = \|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2.$$

We note that any orthogonal transformation, translation or permutation of x_i and y_i keeps all D_{ij} the same. Our goal is then the same as before: identity what additional constraints (if any) are necessary for a unique solution to exist, and then devise an algorithm to extract the solution given the observed data.

Throughout this thesis we will assume we lie in a *generic* configuration. This means the spacial coordinates of all the genomic regions are algebraically independent. Non-generic configurations have measure 0 in the space of all configurations, so this is a safe assumption to work with due because even if they are not generic, any small perturbation will be with probability 1. It is also very convenient because it allows us to avoid algebraically degenerate configurations in our arguments.

Chapter 3

Hi-C Constraints

We first investigate the case where we only have access to Hi-C data. In the monoploid case this data was sufficient to uniquely determine the configuration of genes. However, here we will show that Hi-C constraints are not sufficient for us to have a unique solution up to translations and orthogonal transforms for dimensions greater than 1, demonstrating the solution is not identifiable.

In the rest of this thesis, we will denote the coordinates of true genomic regions by x_i^*, y_i^* and the true combinations of interaction strengths and distances by F_{ij}^* and D_{ij}^*, respectively. The symbols x_i, y_i will be variables in the systems of polynomial equations defined by $x_i^*, y_i^*, F_{ij}^*, D_{ij}^*$.

3.1 Non-identifiability

Here we show for dimensions greater than 1, the solution is not identifiable from F_{ij}^* up to orthogonal transformations, translations and permutations of x_i^* and y_i^* as we present an infinite class of solutions. Now take any other set of points x_i, y_i satisfying for each i:

$$x_i + y_i = x_i^* + y_i^*, \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \tag{3.1}$$

Note that these constraints are independent across different values of i. For each i, the set of (x_i, y_i) which satisfy the equations (3.1) are opposite points on a sphere with center $(x_i^* + y_i^*)/2$ and the second constraint defines the radius of the sphere. For dimensions > 1 this defines an infinite set of points.

Now observe that for a set of x_i, y_i satisfying the equations (3.1)

$$\begin{aligned} D_{ij}^* &= 2 \cdot (\|x_i^*\|^2 + \|y_i^*\|^2) + 2 \cdot (\|x_j^*\|^2 + \|y_j^*\|^2) - 2(x_i^* + y_i^*) \cdot (x_j^* + y_j^*) \\ &= 2 \cdot (\|x_i\|^2 + \|y_i\|^2) + 2 \cdot (\|x_j\|^2 + \|y_j\|^2) - 2(x_i + y_i) \cdot (x_j + y_j) \\ &= \|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2. \end{aligned}$$

Hence they constitute a valid solution to the exact problem. It is clear this produces infinitely many solutions because there are infinitely many possible values of $\|x_1 - x_2\|$ when there should only be 4 possible values.

It is interesting to note that the above construction only fails when $x_i^* = y_i^*$ for all i, i.e. the sphere has radius 0, in which case we have a monoploid genome.

3.2 Distances within homologous pairs are fixed

Despite not having identifiability, we can show the distances between homologue pairs is uniquely defined by the Hi-C constraints. Let x_i, y_i be any set of points satisfying for all $i \neq j$:

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D_{ij}^* \tag{3.2}$$

Note that this holds for x_i, y_i being equal to the real solution. We seek to show that $\|x_1 - y_1\|$ is equal to some constant, and then generalize the argument to all other indices. Perform a shift so that $x_1 = -y_1 = v$. This is valid because shifts preserve distances. Setting $i = 1$ we can write:

$$\|v - x_j\|^2 + \|v - y_j\|^2 + \| - v - x_j\|^2 + \| - v - y_j\|^2 = D_{1j}^*$$

Expanding this out into dot products and simplifying:

$$4\|v\|^2 + 2(\|x_j\|^2 + \|y_j\|^2) = D^*_{1j} \implies 2(\|x_j\|^2 + \|y_j\|^2) = D^*_{1j} - 4\|v\|^2$$

Now take the original distance equation for any $j \neq k$ and both not equal to 1. Substituting in the above:

$$D^*_{1j} + D^*_{1k} - 8\|v\|^2 - 2(x_j + y_j) \cdot (x_k + y_k) = D^*_{jk} \implies s_j \cdot s_k = T_{jk} - 8\|v\|^2$$

where $s_j = \sqrt{2}(x_j + y_j)$ and $T_{jk} = D^*_{1j} + D^*_{1k} - D^*_{jk}$, so a vector s_j is in the same dimension as our data points. Form a matrix $(d+1) \times (d+1)$ matrix T' satisfying $T'_{ij} = T_{(i+2)(j+d+2)}$. This matrix represents the dot products between $s_2, s_3, ..., s_{d+2}$ and $s_{d+3}, s_{d+4}, ..., s_{2d+3}$. Note that the entries of T' can be written as polynomials in terms of the coordinates of the chromosomes. It follows that $\det(T' - 8J\|v\|^2)$ can be viewed as a polynomial in terms of $\|v\|^2$, with coefficients that are polynomials in terms of the coordinates of the chromosomes. We claim that under generic configurations, $\det(T') \neq 0$. Because the coefficients are polynomials in terms of the original coordinates, $\det(T')$ can be written as a polynomial in the original coordinates. So $\det(T') \neq 0$ for generic configurations as long as the polynomial does not identically vanish.

In general dimensions it is difficult to argue if the polynomial vanishes, but for low dimensions it suffices to present one configuration where the polynomial is nonzero. We simply need to present a case where $\det(T') \neq 0$. For $d \leq 3$ we can check this using some python code:

```
import numpy as np
d = 3
x = []
y = []

for i in range(2*d + 3):
    x.append(np.random.randn(d))
    y.append(np.random.randn(d))

y[0] = -x[0]

def norm(v):
    return np.dot(v, v)
```

```python
def D(i,j):
    return norm(x[i]-x[j])+norm(x[i]-y[j])+norm(y[i]-x[j])+norm(y[i]-y[j])

matrix = np.zeros((d+1, d+1))

for i in range(d+1):
    for j in range(d+1):
        matrix[i][j] = D(0,i+1) + D(0,j+1+d+1) - D(i+1,j+1+d+1)

print(np.linalg.det(matrix))
```

The arrays $x[]$ and $y[]$ in our program represent x_i and y_i, but 0-indexed instead of 1-indexed. Thus we have for a generic configuration $\det(T') \neq 0$. Therefore T' has full rank so by the matrix determinant lemma:

$$\det(T' - 8J\|v\|^2) = (1 - 8\|v\|^2 u^T (T')^{-1} u) \det T'$$

where u is a column vector of all 1's. $u^T T'^{-1} u$ is a fixed scalar and $\det T' \neq 0$, so this is a linear equation in terms of $\|v\|^2$. It follows there is a unique solution for $\|v\|$ and thus the distance between the homologue pair x_1, y_1 can be uniquely determined in terms of the constraints as long as $m \geq 2d + 3$. Because this process arbitrarily selected the first index, we can argue the same for any other index i and therefore the distance between homologue pairs is uniquely determined by the F^*_{ij}.

3.3 Classification of all solutions from F^*_{ij}

We now show the construction in Section 3.1 classifies all solutions to $D^*_{ij} = \|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2$ up to translations and orthogonal transforms. By Section 3.2 we know $\|x_i - y_i\|$ for all i. Perform a translation such that $x_1 = -y_1 = v$ for some vector v. Note that we know $\|v\|$ but not the direction of v. Then for any $j \neq 1$, plug in $1, j$ into the constraint for i, j:

$$4\|v\|^2 + 2(\|x_j\|^2 + \|y_j\|^2) = D^*_{ij} \implies 2(\|x_j\|^2 + \|y_j\|^2) = D^*_{ij} - 4\|v\|^2$$

Because we know $\|v\|^2$, we have a constraint on what $\|x_j\|^2 + \|y_j\|^2$ is equal to. Similarly to Section 3.2, if we define $s_j = \sqrt{2}(x_j + y_j)$ and $T_{jk} = D_{ij} + D_{ik} - D_{jk} - 8\|v\|^2$ we find:

$$s_j \cdot s_k = T_{jk}$$

but note that because we have access to the diagonal constraints now, this relationship holds for all j, k and not just $j \neq k$. Thus we have T is a symmetric $(m-1) \cdot (m-1)$ matrix admitting a rank d factorization. Let S be the matrix formed with the vectors s_j. We then have $T = SS^T$. There is a result on rank factorizations of symmetric matrices that any other factorization $T = S'S'^T$ satisfies $S = S'Q$ for some orthogonal matrix Q. Thus for any other solution s'_j, we have $s_j = s'_j Q$, implying all solutions are simply orthogonal transformations of each other (rotations, reflections, etc.)

So in summary, we have shown that the quantities $x_j + y_j$ are unique up to orthogonal transforms once we've fixed $x_1 + y_1 = 0$ via translation. On top of this we also have $\|x_j\|^2 + \|y_j\|^2$ is unique. This shows that the solutions we derived in the previous section are exhaustive as desired. Going forward, we will make use of the fact that this solution set is exhaustive in our arguments.

Chapter 4

Beads on a String Model

As we saw in the previous chapter, the Hi-C constraints were not sufficient for identifiability in 3 dimensions. This is already an interesting finding, as often times it is assumed this data is sufficient to uncovering the spacial organization of genes. Due to this, we augment our constraints with inter-domain distances based on the beads on a string model [8].

Let $x_1^*, x_2^*, \ldots, x_n^*$ and $y_1^*, y_2^*, \ldots, y_n^*$ form chains of genomic regions where the distances between consecutive genomic regions are known, i.e. $\|x_i^* - x_{i+1}^*\| = a_i^*$ and $\|y_i^* - y_{i+1}^*\| = b_i^*$. Here, a chain represents the beads which make up a single chromosome. We refer to these constraints as *inter-domain distances*. Let x_i, y_i be any set of points satisfying for all $i \neq j$:

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D_{ij}^*, \tag{4.1}$$

$$\|x_i - x_{i+1}\| = a_i^* \text{ and } \|y_i - y_{i+1}\| = b_i^*. \tag{4.2}$$

From the results in Section 3.1, we have that for each i, (x_i, y_i) are diametrically opposite points on some sphere with known center and radius and we can explicitly compute the centers and radii from the results. Denote the i-th sphere by S_i and let it have center c_i and radius r_i. Then $\|c_i - x_i\| = r_i$ and $2c_i - x_i = y_i$.

4.1 Finitely many solutions in 2D

Let us consider the case of 2 dimensions. We have $y_1 = 2c_1 - x_1$ and $y_2 = 2c_2 - x_2$. Plugging this into $\|y_1 - y_2\| = b_i^*$ we get:

$$\begin{aligned} b_i^* &= \|(2c_1 - x_1) - (2c_2 - x_2)\|^2 \\ &= \|(2c_1 - 2c_2) - (x_1 - x_2)\|^2 \\ &= \|2c_1 - 2c_2\|^2 + \|x_1 - x_2\|^2 - 2(2c_1 - 2c_2) \cdot (x_1 - x_2) \end{aligned}$$

Note that $\|2c_1 - 2c_2\|^2, \|x_1 - x_2\|^2$ are fixed quantities. This implies there is exactly one solution for $(2c_1 - 2c_2) \cdot (x_1 - x_2)$ (and $c_1 \neq c_2$ by the generic property), and then because we know the length of $x_1 - x_2$ (its just a_i), this implies there are two possible angles for $x_2 - x_1$ and thus there are two possible solutions (this is where we use the 2D constraint).

Now that we know there's two possible solutions for $x_1 - x_2$, note that because x_1, x_2 are constrained to lie on circles, each solution for $x_1 - x_2$ leads to at most two possible solutions for (x_1, x_2). This is because two circles intersect at most twice. In turn this implies there are at most 4 solutions for x_2. We now investigate the four solutions and where they lie in the field $\mathbb{F}$, consisting of the rational numbers adjoined with the real coordinates of our points. Note that c_i, r_i^2 for all i actually belong in this field as $c_i = (x_i + y_i)/2$ and $4r_i^2 = \|x_i - y_i\|^2$.

Obviously x_2^* lies in the field. There is another solution which is x_2^* reflected over the line from c_1 to c_2. This other solution can then be written as simply $(x_2^* - c_2) - 2((x_2^* - c_2) \cdot p) \cdot p$ where p is simply $c_1 - c_2$ rotated 90 degrees (e.g. by swapping the x and y coordinates) which belongs in $\mathbb{F}$.

The other two solutions are slightly more complex. We reflect $x_2 - x_1$ over the line from c_1 to c_2 to get a new vector v. We then translate the circle centered at c_1 by v and intersect it with the circle centered at c_2. This leads to two solutions, whose coordinates can be written as the solutions to a quadratic with coefficients in $\mathbb{F}$. To see why it is a quadratic, note that if we subtract the equations governing the two circles, we get a linear equation, so the problem reduces to the case of intersecting a circle and a line. It follows that the four solutions consist of x_2^*, a reflection point which is some rational function in $\mathbb{F}$, and two points which lie in a quadratic extension of $\mathbb{F}$. We can extend this argument to deduce there are at most four solutions for (x_2, x_3) and so on, implying there are finitely many solutions overall.

4.2 Unique solution in 2D

Now, in the previous section we have done some work on pinning down the exact nature of the solutions. We can apply an identical argument to derive four solutions for x_2 but instead by solving various equations using the constraints describing x_2 and x_3. Again there are four solutions consisting of x_2^*, a reflection lying in $\mathbb{F}$, and two points lying in a quadratic extension.

We show that given certain rational functions not vanishing everywhere, for a generic configuration these four solutions intersect the earlier set of four solutions at only x_2^*. This would in turn imply x_2 has a unique solution of x_2^*. Suppose otherwise, so two solutions were equal at a point not x_2^*. If they are both reflection points, this implies two rational functions in $\mathbb{F}$ are equal, so their difference is equal to 0. If one is a reflection point and one is a root of a quadratic, by plugging in the reflection point into the quadratic we get a rational function in $\mathbb{F}$ evaluates to 0. Finally, if the equal solutions are roots of the quadratics, we can equate the two solutions by using the quadratic formula and by using squaring to get rid of square roots arrive at some rational function in $\mathbb{F}$ must be equal to zero.

We can show that these various rational functions relations cannot be identically zero by simply presenting an example where they do not vanish, e.g. there are 7 total

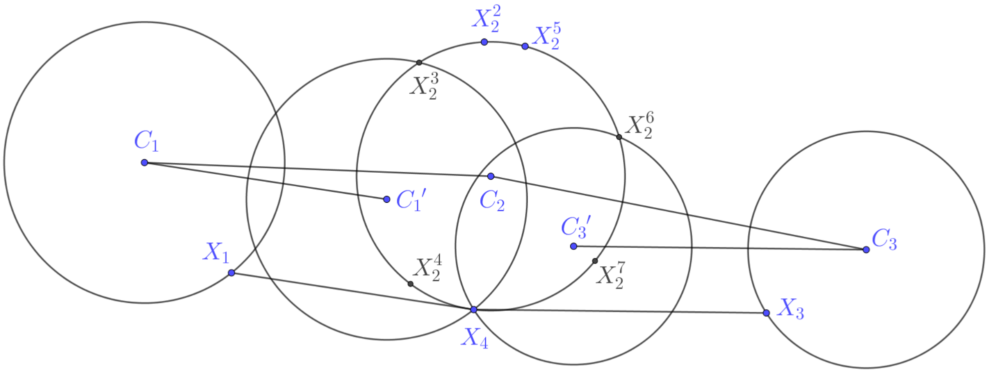

Figure 4-1: An example of 7 distinct solutions.

solutions for x_2 when we combine the two sets. With such an example our rational functions are not identically equal to 0, so under a generic configuration they will be nonzero and thus we will have a unique solution. Such an example is depicted in Figure 4-1.

$c_1, c_2, c_3, x_1, x_2, x_3$ are defined as the centers of the three circles and the true locations of the first three chromosomes. Point c_1' is c_1 translated by $x_2 - x_1$ and is the center of a circle with radius r_1. This circle's intersections with the circle centered at c_2 are x_2 and x_x^3. Point x_2^2 is the reflection of x_2 across the line from c_1 to c_2 and x_2^4 is the reflection of x_2^3 across the same line. The three points x_2^2, x_2^3, x_2^4 are the three alternative solutions derived using the constraints on x_1 and x_2.

Point c_2' is c_2 translated by $x_2 - x_3$ and is the center of a circle with radius r_3. This circle intersects the circle centered at c_2 at x_2 and x_2^6. Point x_2^5 is the reflection of x_2 across the line from c_3 to c_2 and x_2^7 is the reflection of x_2^6 across the same line. Again, x_2^5, x_2^6, x_2^7 are the three alternative solutions derived using the constraints on x_2 and x_3.

The figure shows 7 distinct solutions because none of the alternative solutions are the same point. This implies the certain rational functions specified above are nonvanishing. It follows that x_2 has a unique solution as desired, because the candidate solution sets derived from each side have an intersection of only x_2. This argument generalizes to show every value except for x_1 and x_n, the ends of the chain, are uniquely determined. Note that the above diagram shows that the four solutions for x_2 derived from either side are unique for a generic configuration (because by an identical argument to above, if two solutions were equal certain rational functions would need to be equal). This means once the value of x_2 has been fixed, there is a unique solution for x_1. The same argument then yields x_n is uniquely determined. We therefore have that all values in the chain are uniquely determined as long as there are at least 3 beads on the chain.

With uniqueness proved using the constraints within one chain, we can apply the same argument to every chain. Thus as long as each chromosome is composed of at least 3 beads and we have at least $2d + 3 = 7$ chromosomes total, we have a unique solution in 2D.

4.3 3 Dimensions

Despite having uniqueness in 2D, we do not have uniqueness in 3D. We can argue this by counting the number of constraints. We have $6n$ total variables, 3 for each genomic region. However, we only have $6n - 2$ constraints: we have $4n$ constraints based on x_i, y_i lying on various spheres and then $2n - 2$ constraints based on the inter-domain distances. Thus there are at least two remaining degrees of freedom, making it impossible for there to be a unique solution.

Chapter 5

Extra Constraints for 3D

In the previous chapter we saw the inter-domain distances were sufficient to prove uniqueness in 2D. However, they are not sufficient for 3D. In this chapter we explore two potential candidates for extra constraints. The first is based on Lamina-Associated Domains [7] while the second is based on higher order contact frequencies [5].

We omit most of the proofs in this section and present them in the appendix due to their length. Within this chapter we present the results and a high level overview of the proof flow.

5.1 Distances to the origin

In this section we assume that in addition to the information in the previous sections, we also know distances to the origin for the genomic regions x_1^* and x_n^*, i.e. $\|x_1^*\| = g_1^*$ and $\|x_n^*\| = g_n^*$. Biologically this would arise because Lamina-Associated Domains (LADs) are located near the nuclear membrane in a cell's nucleus. We can estimate the size of the nucleus to get an estimate of the distance of an LAD from the origin, which is the center of the nucleus.

The main result of this section is the following proposition.

Proposition 5.1.1. *Let x_i, y_i be any set of points satisfying for all $i \neq j$:*

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D^*_{ij}, \tag{5.1}$$

$$\|x_i - x_{i+1}\| = a^*_i, \|y_i - y_{i+1}\| = b^*_i, \tag{5.2}$$

$$\|x_1\| = g^*_1 \text{ and } \|x_n\| = g^*_n. \tag{5.3}$$

If $n \geq 2$, then generically the above system has finitely many solutions for $x_1, \ldots, x_n, y_1, \ldots, y_n$.

We recall from Section 3.1 that for each i, (x_i, y_i) are diametrically opposite points on the sphere S_i with center c_i and radius r_i.

The goal of the rest of this section is to prove Proposition 5.1.1. The main components of this proof are Lemma 5.1.5 and Lemma 5.1.6. Roughly speaking, Lemma 5.1.5 studies the above system of polynomial equations restricted to spheres $S_1, \ldots, S_{n-1}$ and Lemma 5.1.6 studies the same system restricted to spheres S_{n-1} and S_n. Corollary 5.1.4 to Lemma 5.1.3 is used in the proof of Lemma 5.1.5. Theorem on the Dimension of Fibers (Theorem 5.1.2) will be an ingredient in the proofs of Lemmas 5.1.3, 5.1.5 and 5.1.6.

Theorem 5.1.2 (Theorem on the Dimension of Fibers, Theorem 1.25 in [6])**.** *If $f : V_1 \to V_2$ is a regular map between irreducible varieties that is surjective, then*

(i) any irreducible component of any fiber has dimension at least $\dim(V_1) - \dim(V_2)$*, and*

(ii) in an open subset of V_2, the dimension of fibers is $\dim(V_1) - \dim(V_2)$*.*

Essentially this theorem allows us to show that the computation of the dimension of a random configuration bounds the dimensions in a generic configuration. For example, if compute a random configuration has a fiber with dimension 0, this implies $\dim(V_1) - \dim(V_2) \leq 0$. This bound will also hold for fibers of generic configurations which is vital for our arguments.

Lemma 5.1.3. *Let (x_1^*, y_1^*) and (x_2^*, y_2^*) be antipodal pairs on spheres S_1 and S_2, respectively. Generically, there are finitely many antipodal pairs (x_2, y_2) on the sphere S_2 satisfying*

$$\|x_1^* - x_2\| = \|x_1^* - x_2^*\| \text{ and } \|y_1^* - y_2\| = \|y_1^* - y_2^*\|. \tag{5.4}$$

Corollary 5.1.4. *Let $(x_1^*, y_1^*), (x_2^*, y_2^*), \ldots, (x_n^*, y_n^*)$ be antipodal pairs on spheres $S_1, S_2, \ldots, S_n$, respectively. Let $(x_2, y_2), \ldots, (x_n, y_n)$ be any antipodal pairs on spheres $S_2, \ldots, S_n$ satisfying for all $1 < i < n$:*

$$\|x_1^* - x_2\| = \|x_1^* - x_2^*\|, \|y_1^* - y_2\| = \|y_1^* - y_2^*\|,$$
$$\|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\|, \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\|.$$

Generically, the above system has finitely many solutions for $x_2, \ldots, x_n, y_2, \ldots, y_n$.

Proof. By Lemma 5.1.3, there are finitely many antipodal pairs (x_2, y_2) on S_2 such that $\|x_1^* - x_2\| = \|x_1^* - x_2^*\|$ and $\|y_1^* - y_2\| = \|y_1^* - y_2^*\|$. Similarly, for each of these points x_2 on S_2, there are finitely many points x_3 on S_3 satisfying $\|x_2 - x_3\| = \|x_2^* - x_3^*\|$ and $\|y_2 - y_3\| = \|y_2^* - y_3^*\|$ etc. □

These two results show that starting from a single solution on the first sphere S_1, the set of solutions grows to at most a finite amount when considering the set of solutions lying on any other sphere. This is important because it tells us if we have a solution set of dimension 1 lying on the first sphere, the solution set will have dimension 1 on any other sphere due to it multiplying by a finite number. This is formally stated in the following result:

Lemma 5.1.5. *Let $V \subseteq \mathbb{C}^{3\times 2n}$ be the set of points satisfying for all $i \neq j$:*

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D^*_{ij},$$
$$\|x_i - x_{i+1}\| = a^*_i, \|y_i - y_{i+1}\| = b^*_i \text{ and } \|x_1\| = g^*_1.$$

The projection $\pi_k : V \to \mathbb{C}^3$ which sends a solution in V to the coordinates x_k lying on sphere $S_k \subseteq \mathbb{C}^3$ can be written as

$$\pi_k(V) = \bigcup_{j=1}^{l_k} (W^k_j \backslash Z^k_j),$$

where $Z^k_j \subset W^k_j \subset \mathbb{C}^3$ are affine varieties. Furthermore, we can assume that W^k_j are irreducible and at most 1-dimensional, and Z^k_j are at most 0-dimensional.

It is important to note that this lemma is correct only over the complex numbers and not over the real numbers.

Lemma 5.1.6. *Let $S_{n-1}, S_n \subseteq \mathbb{C}^3$ be two spheres, let x^*_{n-1} be a point on S_{n-1} and let $(x^*_n)^{(1)}, (x^*_n)^{(2)}$ be two points on S_n. For $i = 1, 2$, let $T^{(i)}$ be the set of points x_{n-1} on S_{n-1} such that there exists x_n on S_n satisfying*

$$\|x_{n-1} - x_n\| = \|x^*_{n-1} - (x^*_n)^{(i)}\|, \|y_{n-1} - y_n\| = \|y^*_{n-1} - (y^*_n)^{(i)}\| \text{ and } \|x_n\| = \|(x^*_n)^{(i)}\|. \tag{5.5}$$

Generically, the intersection of the sets $T^{(1)}$ and $T^{(2)}$ is finite or empty.

Intuitively this lemma discusses the scenario where we fix a gene's location on S_{n-1} and then consider two different constraint sets generated by different locations of the gene on S_n, where we know the distance from the origin on sphere S_n. We then compute the sets $T^{(1)}, T^{(2)}$ which are the alternative solutions for the gene on S_{n-1} given the respective constraint systems. The lemma then states these two sets in general have finite intersection.

With these results proven, we can finally prove our main result of the section. The full proof can be found in the appendix. At a high level, the solution first sets up the 1 dimensional solution spaces on spheres S_1 and S_n derived in 5.1.5. Then using 5.1.4 we can determine the solution set on S_{n-1} is the intersection of one-dimensional sets. Finally by using 5.1.6, we can show this intersection has finite size, concluding the result.

First of all, note that we proved finiteness over the complex numbers. Of course, any real solution is also a complex solution, so in general we have finitely many solutions. Note furthermore that we assumed we knew the distances from the origin for x_1 and x_n. However, our proof adapts to more general settings. First of all, it does not matter if we instead knew y_1 rather than x_1 or y_n instead of x_n. This is because the solution set of y_i lying on a 1-dimensional set implies x_i does as well, and then our arguments still apply. Furthermore, it is not strictly necessary to know the distances from the origin of the endpoints of the chain. In fact, any two beads will suffice. Our argument will show there are finitely many solutions for the genes inbetween the two beads where we know the distance from the origin. This generalizes to finitely many solutions for all beads by 5.1.4.

We conjecture that knowing the distance from the origin for the two endpoint beads plus one bead anywhere else is sufficient for uniqueness. However, we so far have been unable to prove this.

5.2 Tensor constraints

A newly developed technology allows us to measure higher order interaction frequencies, which we term *tensor frequencies* due to being represented in a tensor rather than a matrix [5]. This section is very similar to the previous one where we prove finiteness of solutions and conjecture we also have uniqueness. The proofs techniques used are very similar as well.

Let $f \in \mathbb{R}^{m\times m\times\cdots\times m}$ be a Hi-C tensor, where $f_{x_{i_1},x_{i_2},\ldots,x_{i_k}}$ measures the interaction strength between genomic regions with coordinates $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. In practice, we can only measure a combination of interaction strengths for vectors in $\{x_{i_1}, y_{i_1}\} \times \{x_{i_2}, y_{i_2}\} \times \ldots \times \{x_{i_k}, y_{i_k}\}$, which we will denote by $F_{i_1 i_2 \ldots i_k}$. We will define $D_{i_1 i_2 \ldots i_k} := 1/F_{i_1 i_2 \ldots i_k}$ and assume

$$D_{i_1 i_2 \ldots i_k} = \min_{z_{i_j} \in \{x_{i_j}, y_{i_j}\}} (\sum_{j=1}^{k} (z_{i_j} - (z_{i_1} + \ldots + z_{i_k})/k)^2).$$

Proposition 5.2.1. *Let $x_i, y_i \in \mathbb{R}^3$ be any set of points satisfying for all i, j:*

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D^*_{ij}, \tag{5.6}$$

$$\|x_i - x_{i+1}\| = \|x^*_i - x^*_{i+1}\| \textit{ and } \|y_i - y_{i+1}\| = \|y^*_i - y^*_{i+1}, \| \tag{5.7}$$

$$\min_{z_i \in \{x_i, y_i\} \textit{ for } i=1,2,3} (\textstyle\sum_{j\in\{1,2,3\}} (z_j - (z_1 + z_2 + z_3)/3)^2) = D^*_{123}, \tag{5.8}$$

$$\min_{z_i \in \{x_i, y_i\} \textit{ for } i=n-2,n-1,n} (\textstyle\sum_{j\in\{n-2,n-1,n\}} (z_j - (z_1 + z_2 + z_3)/3)^2) = D^*_{n-2,n-1,n} \tag{5.9}$$

If $n \geq 4$, then generically the above system has finitely many solutions for $x_1, \ldots, x_n, y_1, \ldots, y_n$.

The aim of the rest of this section is to prove Proposition 5.2.1. Proofs in this section will be similar to the proofs in Section 5.1. Lemmas 5.2.3 and 5.2.4 will be the main components of the proof of Proposition 5.2.1. Roughly speaking, Lemma 5.2.3 studies the above system of polynomial equations restricted to spheres $S_1, \ldots, S_{n-1}$ and Lemma 5.2.4 studies the same system restricted to spheres S_{n-2}, S_{n-1}, S_n. Lemma 5.2.2 is used to prove Lemma 5.2.3.

Lemma 5.2.2. *Let $(x^*_1, y^*_1), (x^*_2, y^*_2), (x^*_3, y^*_3)$ be antipodal pairs on spheres S_1, S_2, S_3. Let V consist of all antipodal pairs $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ on spheres S_1, S_2, S_3 sat-*

isfying

$$\begin{gathered}\|x_1 - x_2\| = \|x_1^* - x_2^*\|, \|x_2 - x_3\| = \|x_2^* - x_3^*\|, \qquad (5.10)\\ \|y_1 - y_2\| = \|y_1^* - y_2^*\|, \|y_2 - y_3\| = \|y_2^* - y_3^*\|,\\ \min_{z_i \in \{x_i, y_i\} \textit{ for } i=1,2,3} (\textstyle\sum_{j \in \{1,2,3\}} (z_j - (z_1 + z_2 + z_3)/3)^2) = D_{123}^*.\end{gathered}$$

Generically, we have $\dim(V) = 1$*.*

Lemma 5.2.3. *Let* $V \subseteq \mathbb{C}^{3 \times 2n}$ *be the set of points satisfying for all* $i \neq j$*:*

$$\begin{gathered}\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D_{ij}^*,\\ \|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\|, \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\|,\\ \min_{z_i \in \{x_i, y_i\} \textit{ for } i=1,2,3} (\textstyle\sum_{j \in \{1,2,3\}} (z_j - (z_1 + z_2 + z_3)/3)^2) = D_{123}^*.\end{gathered}$$

Then the projection $\pi_k : V \to \mathbb{C}^3$ *which sends a solution in* V *to the coordinates* x_k *lying on sphere* $S_k \subseteq \mathbb{C}^3$ *can be written as*

$$\pi_k(V) = \bigcup_{j=1}^{l_k} (W_j^k \backslash Z_j^k),$$

where $Z_j^k \subset W_j^k \subset \mathbb{C}^3$ *are affine varieties. Furthermore, we can assume that* W_j^k *are irreducible and at most* 1*-dimensional and* Z_j^k *are at most* 0*-dimensional.*

These two results mimic 5.1.5 in the prior section. We show that one tensor constraint is sufficient to turn the solution set into a one-dimensional constructible set, just as one distance from the origin was able to accomplish.

Lemma 5.2.4. *Let* $S_{n-2}, S_{n-1}, S_n \subseteq \mathbb{C}^3$ *be three spheres, let* x_{n-2}^* *be a point on* S_{n-2}*, let* x_{n-1}^* *be a point on* S_{n-1} *and let* $(x_n^*)^{(1)}, (x_n^*)^{(2)}$ *be points on* S_n*. For* $i = 1, 2$*, let* $T^{(i)}$ *be the be the set of points* x_{n-2}, x_{n-1} *on* $S_{n-2} \times S_{n-1}$ *such that there exists* x_n *on*

S_n satisfying

$$\|x_{n-2} - x_{n-1}\| = \|x^*_{n-2} - x^*_{n-1}\|, \|y_{n-2} - y_{n-1}\| = \|y^*_{n-2} - y^*_{n-1}\|,$$

$$\|x_{n-1} - x_n\| = \|x^*_{n-1} - (x^*_n)^{(i)}\|, \|y_{n-1} - y_n\| = \|y^*_{n-1} - (y^*_n)^{(i)}\|,$$

$$\min_{z_i \in \{x_i, y_i\} \text{ for } i=n-2,n-1,n}(\textstyle\sum_{j\in\{n-2,n-1,n\}}(z_j - (z_1 + z_2 + z_3)/3)^2) = (D^*_{n-2,n-1,n})^{(i)},$$

where $(D^*_{n-2,n-1,n})^{(i)}$ is the distance for $x^*_{n-2}, x^*_{n-1}, (x^*_n)^{(i)}$. Generically, the intersection of sets $T^{(1)}$ and $T^{(2)}$ is finite or empty.

This lemma is the variant of 5.1.6 except we have access to a tensor constraint instead of a distance from the origin.

Then as before, with these results established we can prove the main result of 5.2.1 in the same manner as we proved the main result in the previous section. Therefore we have finitely many solutions in the case of tensor constraints. We conjecture that with at least 3 tensor constraints per chain, we will have uniqueness.

In general we are more optimistic about using tensor frequencies than LADs. There are not strong guarantees on which genes are LADs and how many there are in each chromosome. However, we have access to contact frequency tensors of the whole genome. Because of this, for our numerical studies we use make use of tensor constraints rather than distances from the origin.

Chapter 6

Algorithms

We derived the theoretical framework to establish when we should have unique identifiability of the solution (note we have not proved it, but we strongly believe there is a unique solution). However, a unique solution does not necessarily we can find it efficiently, as in many cases finding the solution may be NP-hard or we may have noise.

6.1 Noiseless Case

We first deal with the noiseless case on simulated data. We make use of a very similar approach to the one used in ChromSDE [10]. We form a $2m \times 2m$ Gram matrix G which tracks the dot products between our $2m$ chromosomes and seek to determine G. The convention we use throughout this paper is for $1 \leq i \leq m$ column/row i correspond to x_i and column/row $m + i$ correspond to y_i. So for example, $G_{1,m+3}$ would be the dot product between x_1 and y_3. G is easier to determine than the true solutions, because it is rotation invariant and by imposing a constraint on the sum of the entries in G we can also fix the translational axis.

We re-phrase our various constraints in terms of the Gram matrix. In general this is possible because distances can be re-written in terms of the Gram matrix:

1. $\|x_i - x_j\|^2 = G_{ii} + G_{jj} - 2G_{ij}$
2. $\|x_i - y_j\|^2 = G_{ii} + G_{(j+m)(j+m)} - 2G_{i(j+m)}$
3. $\|y_i - y_j\|^2 = G_{(i+m)(i+m)} + G_{(j+m)(j+m)} - 2G_{(i+m)(j+m)}$

and all of the constraints we have make use of the three quantities above. It follows that once we translate the constraints to using the entries of the Gram matrix, the constraints will become linear expressions. Because the expressions become quite long, here we omit the derivations, because translating them into using the Gram matrix instead of the distances between chromosomes is straightforward but tedious.

Once our constraints have been translated, we add an extra constraint of $\sum_{i,j} G_{ij} = 0$ and also the values of distances between homologue pairs. The work done in Section 3.2 allows us to compute the distance between homologue pairs using the frequency data. Our objective function is then to minimize $\operatorname{tr} G$. We would rather minimize the rank of G, as then the minimum would be guaranteed to align with the true solution by our theoretical work from earlier. However, then our optimization problem would not be convex and thus very difficult to optimize. tr is a good proxy for minimizing rank so we use it instead [3]. Our resulting convex optimization problem then resembles:

$$\begin{array}{lll}
\text{minimize} & \operatorname{tr} G & \\
\text{subject to} & G_{i,i} - 2G_{i,j} + G_{i+m,i+m} = d^2_{x_i,x_{i+m}} & \forall 1 \leq i \leq m \\
& G_{i,i} + G_{j,j} + G_{i+m,i+m} + G_{j+m,j+m} - & \\
& G_{i,j} - G_{i+m,j} - G_{i,j+m} - G_{i+m,j+m} = D_{ij}/2 & \forall 1 \leq i < j \leq m \\
& \text{Constraints on inter-domain distances and tensors go here} & \\
& \sum_{i,j} G_{i,j} = 0 & \\
& G \text{ positive semidefinite} &
\end{array}$$

where again $d_{x_i,x_{i+m}}$ denotes the distance between x_i and x_{i+m} which we derive in

3.2, and D_{ij} is the sum of the squares of the four distances between chromosomes i and j.

To solve the convex optimization problem, we made use of the solvers SDPT3 and SeDuMi implemented in cvx within MATLAB. After that, we mimic the approach in [10] to re-construct the coordinates of the genes. To assess the accuracy of a solution we then compare the pair-wise distances between genes.

For the dimension one case, we deduced the Hi-C constraints by themselves were sufficient when $m \geq 3$ to re-construct the original solution and in this case the convex optmization problem leads to the correct solution. For the dimension two case we proved knowing the inter-domain distances lead to uniqueness. Indeed, in dimension two when we set the number of chromosomes to 1 and the number of domains per chromosome to at least 15 the output solution matches the true values. For dimension 3, when there are 2 chromosomes with over 40 domains each we can find the original solution, but when 3 chromosomes are present we require only 12. As the number of chromosomes increases, the number of required domains further decreases.

6.2 Adding Noise

In true biological data we cannot expect measurements to always return the true values, especially in this case of contact frequencies where variance comes from that fact that we are measuring a random variable. Due to this, strict constraints such as the ones we use before become highly ineffective.

To combat this, we make use of two common techniques. The first is translating a constraint of the form $f(x) = y$ to adding $(f(x) - y)^2$ to the objective function. This way, the optimizer will be incentivized to set $f(x) = y$, but if this is impossible (i.e. there is measurement error in y), it is fine allowing the equality to be slightly off at some penalty.

The second is through the use of slack variables. Our tensor constraints are of the form $f(x) \leq y$. If there is error in y, it may be the true solution satisfies $f(x) > y$. Because of this, we add slack variables ϵ^+ and ϵ^-, one for each inequality constraint, and change the constraint to $f(x) + \epsilon^+ - \epsilon^- = y$. We then specify $\epsilon^+, \epsilon^- \geq 0$ and add ϵ^- to the objective function. ϵ^- in this case captures how much $f(x)$ exceeds y, so minimizing it encourages the solution to follow our constraint but it is not mandatory.

We can apply these two techniques to make our optimization problem more noise resilient. But there are two issues remaining. The first one is our computation of the distance between homologue pairs. Previously in this thesis we perform a complex computation to compute the homologue pairs, one step of which involves inverting a matrix. Even if the error in our measurements is very small, noise can propagate very strongly through this so we need a more numerically robust method of computation.

Luckily this ends up being straightforward. Recall that we needed to compute a $\|v\|^2$ such that

$$\det(T' - 8J\|v\|^2) = 0$$

where T' is an invertible matrix. Our work showed that this is a linear equation in $\|v\|^2$, so we can simply empirically compute the slope and intercept of this linear equation rather than inverting a matrix to figure it out. For example, one can simply input $\|v\|^2 = 0, 1$ and compute the determinants to determine the slope and intercept. This allows us to compute the unique solution to $\|v\|^2$ using only determinants.

Furthermore, to smooth out the end result even further, recall that T' was formed by selecting a set of $2d + 2$ indices. One can perform the same process but selecting another set of indices and then averaging the resulting solutions to $\|v\|^2$ for both sets. Performing this many times will drastically decrease the variance on our estimate of the distance.

The other issues is more subtle. While now our optimization problem is quite resilient to noise, it is very difficult to solve. Most general convex optimization programs can handle problems with relatively few variables but an enormous number of constraints. Our problem has an extremely large number of variables and constraints, so solvers such as SDPT3 run into memory errors extremely quickly (even around 15 genes total overwhelms them). Because of this, we implement a system of sharing slack variables. For example, we define a global parameter ϵ and specify that $f(x) \leq y + \epsilon$ for all of our inequality constraints. We then add $\epsilon \geq 0$ as another constraint and put ϵ into the objective function. If the errors in our measurements are uniformly random, this is mostly equivalent to the formulation of one slack variable per inequality constraint. For more accuracy more slack variables can be used, but for our small configurations we found 1 was sufficient.

After these transformations, our optimization problem will resemble:

$$\begin{array}{lll} \text{minimize} & \operatorname{tr} G + \sum_{i=1}^{s_1} (f_i(G) - b_i)^2 + \epsilon & \\ \text{subject to} & g_j(G) \leq d_j + \epsilon & \forall 1 \leq j \leq s_2 \\ & \epsilon \geq 0 & \\ & \sum_{i,j} G_{i,j} = 0 & \\ & G \text{ positive semidefinite} & \end{array}$$

where we have s_1 constraints of the form $f_i(G) = b_i$ and s_2 constraints of the form $g_j(G) \leq d_j$.

In order to still get good results, we must attach constants to balance the priorities of minimizing either the trace or aligning with the constraints. In our numerical experiments we found attaching a constant ρ in front of $\operatorname{tr} G$ is sufficient, however one must carefully choose the value of this constant.

In general, the accuracy of the solution is extremely dependent on ρ. Through numerical experiments we find for any given configuration there is a small range of ρ values which will yield a correct solution. Fortunately, there is a simple way to determine whether a given ρ value succeeded or not without knowing the true solution beforehand. When a solution is output, we compute the eigenvalues of the output Gram matrix. The quality of the solution generally scales directly with the $(d+1)^{th}$ eigenvalue of the matrix, where we are working in dimension d. If the eigenvalue is large, the solution will be very poor, but the closer it is to 0, the higher quality the solution is. Therefore something simple such as a binary search should be sufficient to determine the optimal value of ρ quickly. For 2D $\rho \approx 10^{-2}$ and for 3D $\rho \approx 10^{-5}$ lead to fairly accurate solutions most of the time for the minimal configurations specified in the previous section. For larger configurations we expect ρ needs to be re-tuned using a binary search.

We have also experimented with injecting artificial noise. When we add noise drawn from a small uniform distribution to every observation, we find that we can still re-construct the configuration of genes with high accuracy. This is to be expected due to the nature of how we constructed our optimization problem, but it is re-assuring to check the numerical results agree with the theoretical ones.

For smaller configurations such as ones we have discussed, general solvers are sufficient. However, very large systems will often cause memory issues or too slow convergence for these algorithms. Because of this a specialized solver such as the one detailed in [3] can be used to retain high performance.

Chapter 7

Conclusion and Future Work

7.1 Future Work

Moving forward the goals are quite clear. On the theoretical side, we would like to show we have a unique solution when in possession of either 3 distances from the origin or 3 tensor constraints per chromosome. Based on computational results, we are strongly confident this is true. One idea we had involved computing mixed volume and applying the Bernstein-Kushnirenko theorem. However, we did not have much success in completing this approach.

In terms of the algorithmic side, there are two extensions. The first is more straightforward, where we implement our constraint system into the PPA Smoothing algorithm to demonstrate our approach can generalize to larger configurations. On top of this, we would like to obtain real data sets (in particular, the tensor frequencies) to test how our approach handles real scenarios.

7.2 Conclusion

In this thesis, we have presented an algorithm to infer the 3D structure of diploid chromosomes. We make use of Hi-C contact frequencies, inter-domain distances between beads within chromosomes and tensor contact frequencies to achieve this. We have proven this data is sufficient to narrow the solution down to finitely many possibilities and conjecture we in fact have uniqueness. This is supported by our numerical experiments, where even in the presence of noise we can accurately solve for the true configuration of genes.

Appendix A

Proofs of Chapter 5

Here we present the previously omitted proofs of results in Chapter 5. Most of these proofs were developed in large part by Kaie Kubjas.

A.1 Proofs for 5.1

Proof of 5.1.3. We will first show by a `Macaulay2` computation that the statement of the lemma holds for randomly chosen (x_1^*, y_1^*) and (x_2^*, y_2^*). Then we will use Theorem 5.1.2 to prove it for generic values.

The statement can be shown for randomly chosen values by the following `Macaulay2` code.

```
R = QQ[x1s1,x1s2,x1s3,x2s1,x2s2,x2s3,c11,c12,c13,c21,c22,c23,x21,x22,x23]
--first polynomial: x2 and x2s lie on the same sphere
--second polynomial: d(x1s,x2)=d(x1s,x2s)
--third polynomial: d(ys1,y2)=d(ys1,ys2) (expressed through x's and c's)
I = ideal(
     (x21-c21)^2+(x22-c22)^2+(x23-c23)^2-((x2s1-c21)^2+(x2s2-c22)^2+(x2s3-c23)^2),
     (x1s1-x21)^2+(x1s2-x22)^2+(x1s3-x23)^2-((x1s1-x2s1)^2+(x1s2-x2s2)^2+(x1s3-x2s3)
     ^2),
     (-x1s1+2*c11+x21-2*c21)^2+(-x1s2+2*c12+x22-2*c22)^2+(-x1s3+2*c13+x23-2*c23)^2-
     ((-x1s1+2*c11+x2s1-2*c21)^2+(-x1s2+2*c12+x2s2-2*c22)^2+(-x1s3+2*c13+x2s3-2*c23)
     ^2));
--substitute parameter variables with random values
C11 = random(1,10);
C12 = random(1,10);
C13 = random(1,10);
C21 = random(1,10);
C22 = random(1,10);
C23 = random(1,10);
```

```
X1S1 = random(1,10);
X1S2 = random(1,10);
X1S3 = random(1,10);
X2S1 = random(1,10);
X2S2 = random(1,10);
X2S3 = random(1,10);
J = sub(I,{x1s1=>X1S1,x1s2=>X1S2,x1s3=>X1S3,
     x2s1=>X2S1,x2s2=>X2S2,x2s3=>X2S3,
     c11=>C11,c12=>C12,c13=>C13,
     c21=>C21,c22=>C22,c23=>C23})
J2=sub(J,QQ[x21,x22,x23])
--correctness check: x2s should be in the ideal
sub(J2,{x21=>X2S1,x22=>X2S2,x23=>X2S3})
dim J2
```

Let $V_1 \subseteq \mathbb{C}^{15}$ be the variety defined by the ideal I in the code above. Let $V_2 = \mathbb{C}^{12}$, the space of values for x_1^*, x_2^*, and centers of S_1 and S_2. Let f be the coordinate projection from V_1 to V_2. By the `Macaulay2` computation, there exists a fiber of dimension zero. By the first part of Theorem 5.1.2, we have $\dim(V_1) - \dim(V_2) \leq 0$. In fact $\dim(V_1) - \dim(V_2) = 0$, because the ideal of V_1 is generated by three polynomials. By the second part of Theorem 5.1.2, there exists some open set where the dimension of fibers is $\dim V_1 - \dim V_2 = 0$. The statement of the lemma follows, because open sets are dense. □

Proof of 5.1.5. The first statement follows from [1, Theorem 7 in §4.7]. We can assume that W_j^k are irreducible, because otherwise we can replace $(W_j^k \backslash Z_j^k)$ with the union over the irreducible components of W_j^k. By [1, Proposition 10 in §9.4], we have $\dim(Z_j^k) < \dim(W_j^k)$.

To show that W_j^k are at most 1-dimensional, we will show that V is at most 1-dimensional by applying Theorem 5.1.2. We take $V_1 \subseteq \mathbb{C}^{3\times 2n}$ to be an irreducible component of V, $V_2 \subseteq \mathbb{C}^3$ to be the set of points on the sphere S_1 satisfying $\|x_1\| = g_1^*$ and the map f to be the projection to the x_1 coordinates. The map f is surjective by definition and regular due to being a projection. The set of points on sphere S_1 satisfying $\|x_1\| = g_1^*$ is a circle and hence $\dim(V_2) = 1$. By Corollary 5.1.4, a generic fiber of f over a point in V_2 is 0-dimensional. By Theorem 5.1.2, the variety V is 1-dimensional. This in turn implies that the W_j^k are at most 1-dimensional as desired. □

Proof of 5.1.6. We will first show the statement of the lemma for randomly chosen points $x^*_{n-1}, (x^*_n)^{(1)}, (x^*_n)^{(2)}$ and centers c_{n-1}, c_n of the spheres S_{n-1}, S_n by a `Macaulay2` computation. We will then apply Theorem 5.1.2 to deduce the statement for generic values.

```
R = QQ[x1s1,x1s2,x1s3,x21s1,x21s2,x21s3,x22s1,x22s2,x22s3,c11,c12,c13,c21,c22,c23,
    x11,x12,x13,x211,x212,x213,x221,x222,x223]
--first four polynomials: x1 and x1s lie on the same sphere; x21 and x21s lie on the same sphere;
--x22 and x22s lie on the same sphere; x21s and x22s lie on the same sphere
--fifth and sixth polynomial: d(x1,x2i)=d(x1s,x2is) for i=1,2
--seventh and eighth polynomial: d(y1,y2i)=d(y1s,y2is) for i=1,2
--ninth and tenth polynomial: d(x2i)=d(x2is) for i=1,2
I = ideal((x11-c11)^2+(x12-c12)^2+(x13-c13)^2-((x1s1-c11)^2+(x1s2-c12)^2+(x1s3-c13)^2),
    (x211-c21)^2+(x212-c22)^2+(x213-c23)^2-((x21s1-c21)^2+(x21s2-c22)^2+(x21s3-c23)^2),
    (x221-c21)^2+(x222-c22)^2+(x223-c23)^2-((x22s1-c21)^2+(x22s2-c22)^2+(x22s3-c23)^2),
    (x21s1-c21)^2+(x21s2-c22)^2+(x21s3-c23)^2-((x22s1-c21)^2+(x22s2-c22)^2+(x22s3-c23)^2),
    (x11-x211)^2+(x12-x212)^2+(x13-x213)^2-((x1s1-x21s1)^2+(x1s2-x21s2)^2+(x1s3-x21s3)^2),
    (x11-x221)^2+(x12-x222)^2+(x13-x223)^2-((x1s1-x22s1)^2+(x1s2-x22s2)^2+(x1s3-x22s3)^2),
    (-x11+2*c11+x211-2*c21)^2+(-x12+2*c12+x212-2*c22)^2+(-x13+2*c13+x213-2*c23)^2-
    ((-x1s1+2*c11+x21s1-2*c21)^2+(-x1s2+2*c12+x21s2-2*c22)^2+(-x1s3+2*c13+x21s3-2*c23)^2),
    (-x11+2*c11+x221-2*c21)^2+(-x12+2*c12+x222-2*c22)^2+(-x13+2*c13+x223-2*c23)^2-
    ((-x1s1+2*c11+x22s1-2*c21)^2+(-x1s2+2*c12+x22s2-2*c22)^2+(-x1s3+2*c13+x22s3-2*c23)^2),
    x211^2+x212^2+x213^2-(x21s1^2+x21s2^2+x21s3^2), x221^2+x222^2+x223^2-(x22s1^2+x22s2^2+x22s3^2));
--substitute parameter variables with random values
C11 = random(1,10);
C12 = random(1,10);
C13 = random(1,10);
C21 = random(1,10);
C22 = random(1,10);
C23 = random(1,10);
R2 = random(1,10);
X1S1 = random(1,10);
X1S2 = random(1,10);
X1S3 = random(1,10);
X21S1 = C21+R2;
X21S2 = C22;
X21S3 = C23;
X22S1 = C21;
X22S2 = C22+R2;
X22S3 = C23;
J = sub(I,{x1s1=>X1S1,x1s2=>X1S2,x1s3=>X1S3,
  x21s1=>X21S1,x21s2=>X21S2,x21s3=>X21S3,
  x22s1=>X22S1,x22s2=>X22S2,x22s3=>X22S3,
  c11=>C11,c12=>C12,c13=>C13,
  c21=>C21,c22=>C22,c23=>C23})
J2=sub(J,QQ[x11,x12,x13,x211,x212,x213,x221,x222,x223])
--correctness check: (x1s,x21s,x22s) should be in the ideal
sub(J2,{x11=>X1S1,x12=>X1S2,x13=>X1S3,
  x211=>X21S1,x212=>X21S2,x213=>X21S3,
   x221=>X22S1 , x222=>X22S2 , x223=>X22S3})
dim J2
```

Let V_1 be the variety defined by the equations (5.5), where $x^*_{n-1}, (x^*_n)^{(1)}, (x^*_n)^{(2)}$ and centers c_{n-1}, c_n of the spheres S_{n-1}, S_n are considered as variables and antipodal points are expressed using centers of spheres. Let V_2 be the variety of feasible values for $x^*_{n-1}, (x^*_n)^{(1)}, (x^*_n)^{(2)}$ and centers c_{n-1}, c_n of the spheres S_{n-1}, S_n, i.e. it is defined by the equation that $(x^*_n)^{(1)}, (x^*_n)^{(2)}$ lie on the same sphere.

The coordinate projection from V_1 to V_2 is surjective, because an element of V_2 always gives an element of V_1. The code above shows that the fiber of a random point of V_2 is 0-dimensional. By the Theorem 5.1.2, the statement holds for generic parameters. □

Proof of Proposition 5.1.1. We want to show that for generic $x^*_1, \dots, x^*_n, y^*_1, \dots, y^*_n$, the solution set of (5.1) is finite. By contradiction, assume that this is not the case.

Fix generic $x^*_1, \dots, x^*_{n-1}, y^*_1, \dots, y^*_{n-1}$. Let V be the solution set of the following equations for all $1 \leq i \neq j \leq n-1$:

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D^*_{ij},$$
$$\|x_i - x_{i+1}\| = a^*_i, \|y_i - y_{i+1}\| = b^*_i \text{ and } \|x_1\| = g^*_1.$$

We will consider $\pi_{n-1}(V)$, which by Lemma 5.1.5 is at most 1-dimensional constructible set. Let $k = l_{n-1}$ be the number of irreducible components of $\pi_n(V)$.

We also fix generic $(x^*_n)^{(1)}, \dots, (x^*_n)^{(k+1)}, (y^*_n)^{(1)}, \dots, (y^*_n)^{(k+1)}$ on S_n. Let $T^{(i)}$ be the set of points x_{n-1} on S_{n-1} such that there exists x_n on S_n satisfying

$$\|x_{n-1} - x_n\| = \|x^*_{n-1} - (x^*_n)^{(i)}\|, \|y_{n-1} - y_n\| = \|y^*_{n-1} - (y^*_n)^{(i)}\| \text{ and } \|x_n\| = \|(x^*_n)^{(i)}\|.$$

By Lemma 5.1.5, the sets $T^{(i)}$ are constructible. Moreover, by our assumption that the solution set of (5.1) is infinite, the intersections of $\pi_{n-1}(V)$ and $T^{(i)}$ are infinite, because they are equal to the projections of the solution set of (5.1) with $x_n^* = (x_n^*)^{(i)}$ to S_{n-1}.

However, this is a contradiction to Lemma 5.1.6 that states that the Zariski closures of $T^{(1)}, \dots, T^{(k+1)}$ intersect pairwise at finite number of points: The set $\pi_{n-1}(V)$ can be written as $\cup_{j=1}^{l_{n-1}}(W_j^{n-1} \backslash Z_j^{n-1})$ and $T^{(i)}$ can be written as $\cup_{j=1}^{l'}(W'_j \backslash Z'_j)$. By the previous paragraph, there exist j and j' such that the intersection of $W_j^{n-1} \backslash Z_j^{n-1}$ and $W'_{j'} \backslash Z'_{j'}$ is infinite. Hence $W_j^{n-1} = W'_{j'}$, because otherwise $W_j^{n-1} \cap W'_{j'}$ would be 0-dimensional by [1, Proposition 10 in §9.4] and hence $(W_j^{n-1} \backslash Z_j^{n-1}) \cap (W'_{j'} \backslash Z'_{j'})$ would be finite. Since there are more different $T^{(i)}$ than W_j^{n-1}, then there exists 1-dimensional W_j^{n-1}, a component $W' \backslash Z'$ of $T^{(i')}$ and a component $W'' \backslash Z''$ of $T^{(i'')}$ such that $W' = W_j^{n-1}$ and $W'' = W_j^{n-1}$. The intersection of $W' \backslash Z'$ and $W'' \backslash Z''$ is $W_j^{n-1} \backslash (Z' \cup Z'')$ and hence it is 1-dimensional. This is a contradiction to Lemma 5.1.6, so it follows the solution set of (5.1) is finite. □

A.2 Proofs for 5.2

Proof of 5.2.2. To prove that $\dim(V) = 1$, we first show it with a `Macaulay2` computation for randomly chosen parameters and then apply Theorem 5.1.2 to prove $\dim(V) = 1$ for generic parameters. The computation is for eight different cases, each case corresponding to different triple (z_1, z_2, z_3) achieving the minimum in the equation containing D^*_{123}.

```
R = QQ[x1s1,x1s2,x1s3,x2s1,x2s2,x2s3,x3s1,x3s2,x3s3,c11,c12,c13,c21,c22,c23,c31,c32,
    c33,
    x11,x12,x13,x21,x22,x23,x31,x32,x33]
--first three polynomials: xi and xis lie on the same sphere for i=1,2,3
--next two polynomials: d(x1,x2)=d(x1s,x2s) and d(x2,x3)=d(x2s,x3s)
--last two polynomials: d(y1,y2)=d(y1s,y2s) and d(y2,y3)=d(y2s,y3s)
I0 = ideal((x11-c11)^2+(x12-c12)^2+(x13-c13)^2-((x1s1-c11)^2+(x1s2-c12)^2+(x1s3-c13)
    ^2),
    (x21-c21)^2+(x22-c22)^2+(x23-c23)^2-((x2s1-c21)^2+(x2s2-c22)^2+(x2s3-c23)^2),
    (x31-c31)^2+(x32-c32)^2+(x33-c33)^2-((x3s1-c31)^2+(x3s2-c32)^2+(x3s3-c33)^2),
    (x11-x21)^2+(x12-x22)^2+(x13-x23)^2-((x1s1-x2s1)^2+(x1s2-x2s2)^2+(x1s3-x2s3)^2),
    (x21-x31)^2+(x22-x32)^2+(x23-x33)^2-((x2s1-x3s1)^2+(x2s2-x3s2)^2+(x2s3-x3s3)^2),
    (-x11+2*c11+x21-2*c21)^2+(-x12+2*c12+x22-2*c22)^2+(-x13+2*c13+x23-2*c23)^2-
    ((-x1s1+2*c11+x2s1-2*c21)^2+(-x1s2+2*c12+x2s2-2*c22)^2+(-x1s3+2*c13+x2s3-2*c23)
    ^2),
    (-x21+2*c21+x31-2*c31)^2+(-x22+2*c22+x32-2*c32)^2+(-x23+2*c23+x33-2*c33)^2-
    ((-x2s1+2*c21+x3s1-2*c31)^2+(-x2s2+2*c22+x3s2-2*c32)^2+(-x2s3+2*c23+x3s3-2*c33)
    ^2));

--the next part constructs an ideal for the tensor constraint in each of the eight
    different cases
threeDimDistance = (p11,p12,p13,p21,p22,p23,p31,p32,p33) -> (
    (p11-1/3*(p11+p21+p31))^2+(p12-1/3*(p12+p22+p32))^2+(p13-1/3*(p13+p23+p33))^2+
    (p21-1/3*(p11+p21+p31))^2+(p22-1/3*(p12+p22+p32))^2+(p23-1/3*(p13+p23+p33))^2+
    (p31-1/3*(p11+p21+p31))^2+(p32-1/3*(p12+p22+p32))^2+(p33-1/3*(p13+p23+p33))^2
    );
antipodalPairs1 = {
    {x11,-x11+2*c11},
    {x12,-x12+2*c12},
    {x13,-x13+2*c13},
    {x21,-x21+2*c21},
    {x22,-x22+2*c22},
    {x23,-x23+2*c23},
    {x31,-x31+2*c31},
    {x32,-x32+2*c32},
    {x33,-x33+2*c33}
    };
antipodalPairs2 = {
    {x1s1,-x1s1+2*c11},
    {x1s2,-x1s2+2*c12},
    {x1s3,-x1s3+2*c13},
    {x2s1,-x2s1+2*c21},
    {x2s2,-x2s2+2*c22},
    {x2s3,-x2s3+2*c23},
    {x3s1,-x3s1+2*c31},
    {x3s2,-x3s2+2*c32},
    {x3s3,-x3s3+2*c33}
    };
idealsList = flatten flatten for i to 1 list for j to 1 list for k to 1 list ideal (
    threeDimDistance(antipodalPairs1#0#i,antipodalPairs1#1#i,antipodalPairs1#2#i,
    antipodalPairs1#3#j,antipodalPairs1#4#j,antipodalPairs1#5#j,antipodalPairs1#6#k,
    antipodalPairs1#7#k,antipodalPairs1#8#k) -
    threeDimDistance(antipodalPairs2#0#i,antipodalPairs2#1#i,antipodalPairs2#2#i,
    antipodalPairs2#3#j,antipodalPairs2#4#j,antipodalPairs2#5#j,antipodalPairs2#6#k,
    antipodalPairs2#7#k,antipodalPairs2#8#k));

--loop that finds dimensions in eight different cases
for i to #idealsList-1 do (
    I = I0+idealsList#i;
    print(i);
    --substitute parameters with random values
    C11 = random(1,10);
    C12 = random(1,10);
    C13 = random(1,10);
    C21 = random(1,10);
    C22 = random(1,10);
    C23 = random(1,10);
    C31 = random(1,10);
    C32 = random(1,10);
    C33 = random(1,10);
    X1S1 = random(1,10);
    X1S2 = random(1,10);
    X1S3 = random(1,10);
    X2S1 = random(1,10);
    X2S2 = random(1,10);
    X2S3 = random(1,10);
    X3S1 = random(1,10);
    X3S2 = random(1,10);
    X3S3 = random(1,10);
    J = sub(I,{x1s1=>X1S1,x1s2=>X1S2,x1s3=>X1S3,
      x2s1=>X2S1, x2s2=>X2S2, x2s3=>X2S3,
      x3s1=>X3S1, x3s2=>X3S2, x3s3=>X3S3,
      c11=>C11, c12=>C12, c13=>C13,
      c21=>C21, c22=>C22, c23=>C23,
      c31=>C31, c32=>C32, c33=>C33});
    J2=sub(J,QQ[x11,x12,x13,x21,x22,x23,x31,x32,x33]) ;
    --correctness check: (x1s,x2s,x3s) should lie in the ideal
    print(sub(J2,{x11=>X1S1,x12=>X1S2,x13=>X1S3,
      x21=>X2S1,x22=>X2S2,x23=>X2S3,
      x31=>X3S1,x32=>X3S2,x33=>X3S3}));
    print(dim J2);
  )
```

We get $\dim(V) = 1$ for generic parameters by Theorem 5.1.2. More precisely, let V_1 be the variety defined by the equations (5.10) where true points x_1^*, x_2^*, x_3^* and the centers c_1, c_2, c_3 of the spheres S_1, S_2, S_3 are considered as variables and let V_2 be $\mathbb{C}^{3\times 6}$ corresponding to the space of values for the true points x_1^*, x_2^*, x_3^* and the centers c_1, c_2, c_3 of the spheres S_1, S_2, S_3. Let the map from V_1 to V_2 to be the coordinate projection onto V_2. Since a fiber of this map has dimension 1, a generic fiber has dimension ≤ 1. By dimension count, the dimension of a generic fiber has to be equal to 1. □

Proof of 5.2.3. This proof is similar to the proof of 5.1.5. To show that W_j^k are at most 1-dimensional, we will show that V is 1-dimensional by applying Theorem 5.1.2. We take $V_1 \subseteq \mathbb{C}^{3\times 2n}$ to be an irreducible component of V, we take $V_2 \subseteq \mathbb{C}^3 \times \mathbb{C}^3 \times \mathbb{C}^3$ to be the variety in the statement of Lemma 5.2.2, i.e. the projection of V on $S_1 \times S_2 \times S_3$, and the map to be the projection to the x_1, x_2, x_3 coordinates. By Lemma 5.2.2, we have $\dim(V_2) = 1$. By Corollary 5.1.4, a generic fiber of f has dimension 0 and thus $\dim(V_1) = 1$ follows from Theorem 5.1.2. □

Proof of 5.2.4. As in some earlier proofs, we will first show the statement of the lemma for specific parameters using `Macaulay2` and then prove it for generic parameters by applying Theorem 5.1.2. Since the minimum for $(D^*_{n-2,n-1,n})^{(i)}$ can be achieved by eight different configurations both for equations including $(x_n^*)^{(1)}$ and for equations including $(x_n^*)^{(2)}$, we have to do the computation for 64 different cases.

```
R = QQ[x1s1,x1s2,x1s3,x2s1,x2s2,x2s3,x31s1,x31s2,x31s3,x32s1,x32s2,x32s3,c11,c12,c13,
    c21,c22,c23,c31,c32,c33,
    x11,x12,x13,x21,x22,x23,x311,x312,x313,x321,x322,x323]
--first four polynomials: x1 and x1s lie on the same sphere; x2 and x2s lie on the
    same sphere;
--x3i and x3is lie on the same sphere for i=1,2
--next three polynomials: d(x1,x2)=d(x1s,x2s) and d(x2,x3i)=d(x2s,x3is) for i=1,2
--last three polynomials: d(y1,y2)=d(y1s,y2s) and d(y2,y3i)=d(y2s,y3is) for i=1,2
I0 = ideal((x11-c11)^2+(x12-c12)^2+(x13-c13)^2-((x1s1-c11)^2+(x1s2-c12)^2+(x1s3-c13)
    ^2),
    (x21-c21)^2+(x22-c22)^2+(x23-c23)^2-((x2s1-c21)^2+(x2s2-c22)^2+(x2s3-c23)^2),
    (x311-c31)^2+(x312-c32)^2+(x313-c33)^2-((x31s1-c31)^2+(x31s2-c32)^2+(x31s3-c33)
    ^2),
    (x321-c31)^2+(x322-c32)^2+(x323-c33)^2-((x32s1-c31)^2+(x32s2-c32)^2+(x32s3-c33)
    ^2),
    (x11-x21)^2+(x12-x22)^2+(x13-x23)^2-((x1s1-x2s1)^2+(x1s2-x2s2)^2+(x1s3-x2s3)^2),
    (x21-x311)^2+(x22-x312)^2+(x23-x313)^2-((x2s1-x31s1)^2+(x2s2-x31s2)^2+(x2s3-x31s3
    )^2),
    (x21-x321)^2+(x22-x322)^2+(x23-x323)^2-((x2s1-x32s1)^2+(x2s2-x32s2)^2+(x2s3-x32s3
    )^2),
    (-x11+2*c11+x21-2*c21)^2+(-x12+2*c12+x22-2*c22)^2+(-x13+2*c13+x23-2*c23)^2-
    ((-x1s1+2*c11+x2s1-2*c21)^2+(-x1s2+2*c12+x2s2-2*c22)^2+(-x1s3+2*c13+x2s3-2*c23)
    ^2),
    (-x21+2*c21+x311-2*c31)^2+(-x22+2*c22+x312-2*c32)^2+(-x23+2*c23+x313-2*c33)^2-
    ((-x2s1+2*c21+x31s1-2*c31)^2+(-x2s2+2*c22+x31s2-2*c32)^2+(-x2s3+2*c23+x31s3-2*c33
    )^2),
    (-x21+2*c21+x321-2*c31)^2+(-x22+2*c22+x322-2*c32)^2+(-x23+2*c23+x323-2*c33)^2-
    ((-x2s1+2*c21+x32s1-2*c31)^2+(-x2s2+2*c22+x32s2-2*c32)^2+(-x2s3+2*c23+x32s3-2*c33
    )^2));

--the next part constructs the tensor constraint for eight different cases both for
    x31 and x32
threeDimDistance = (p1x,p1y,p1z,p2x,p2y,p2z,p3x,p3y,p3z) -> (
    (p1x-1/3*(p1x+p2x+p3x))^2+(p1y-1/3*(p1y+p2y+p3y))^2+(p1z-1/3*(p1z+p2z+p3z))^2+
    (p2x-1/3*(p1x+p2x+p3x))^2+(p2y-1/3*(p1y+p2y+p3y))^2+(p2z-1/3*(p1z+p2z+p3z))^2+
    (p3x-1/3*(p1x+p2x+p3x))^2+(p3y-1/3*(p1y+p2y+p3y))^2+(p3z-1/3*(p1z+p2z+p3z))^2
    );
antipodalPairs1 = {
    {x11,-x11+2*c11},
    {x12,-x12+2*c12},
    {x13,-x13+2*c13},
    {x21,-x21+2*c21},
    {x22,-x22+2*c22},
    {x23,-x23+2*c23},
    {x311,-x311+2*c31},
    {x312,-x312+2*c32},
    {x313,-x313+2*c33},
    {x321,-x321+2*c31},
    {x322,-x322+2*c32},
    {x323,-x323+2*c33}
    };
antipodalPairs2 = {
    {x1s1,-x1s1+2*c11},
    {x1s2,-x1s2+2*c12},
    {x1s3,-x1s3+2*c13},
    {x2s1,-x2s1+2*c21},
    {x2s2,-x2s2+2*c22},
    {x2s3,-x2s3+2*c23},
    {x31s1,-x31s1+2*c31},
    {x31s2,-x31s2+2*c32},
    {x31s3,-x31s3+2*c33},
    {x32s1,-x32s1+2*c31},
    {x32s2,-x32s2+2*c32},
    {x32s3,-x32s3+2*c33}
    };
idealsList1 = flatten flatten for i to 1 list for j to 1 list for k to 1 list ideal (
    threeDimDistance(antipodalPairs1#0#i,antipodalPairs1#1#i,antipodalPairs1#2#i,
    antipodalPairs1#3#j,antipodalPairs1#4#j,antipodalPairs1#5#j,antipodalPairs1#6#k,
    antipodalPairs1#7#k,antipodalPairs1#8#k)-
    threeDimDistance(antipodalPairs2#0#i,antipodalPairs2#1#i,antipodalPairs2#2#i,
    antipodalPairs2#3#j,antipodalPairs2#4#j,antipodalPairs2#5#j,antipodalPairs2#6#k,
    antipodalPairs2#7#k,antipodalPairs2#8#k));
idealsList2 = flatten flatten for i to 1 list for j to 1 list for k to 1 list ideal (
    threeDimDistance(antipodalPairs1#0#i,antipodalPairs1#1#i,antipodalPairs1#2#i,
    antipodalPairs1#3#j,antipodalPairs1#4#j,antipodalPairs1#5#j,antipodalPairs1#9#k,
    antipodalPairs1#10#k,antipodalPairs1#11#k)-
    threeDimDistance(antipodalPairs2#0#i,antipodalPairs2#1#i,antipodalPairs2#2#i,
    antipodalPairs2#3#j,antipodalPairs2#4#j,antipodalPairs2#5#j,antipodalPairs2#9#k,
    antipodalPairs2#10#k,antipodalPairs2#11#k));

--a loop that finds dimensions in 64 different cases
for i to #idealsList1-1 do for j to #idealsList2-1 do (
    print(i*8+j);
    I = I0+idealsList1#i+idealsList2#j;
    --substitute parameter variables with random values
    C11 = random(1,10);
    C12 = random(1,10);
    C13 = random(1,10);
    C21 = random(1,10);
    C22 = random(1,10);
    C23 = random(1,10);
    C31 = random(1,10);
    C32 = random(1,10);
    C33 = random(1,10);
    R3 = random(1,10);
    X1S1 = random(1,10);
    X1S2 = random(1,10);
    X1S3 = random(1,10);
    X2S1 = random(1,10);
    X2S2 = random(1,10);
    X2S3 = random(1,10);
    X31S1 = C31+R3;
    X31S2 = C32;
    X31S3 = C33;
    X32S1 = C31;
    X32S2 = C32+R3;
    X32S3 = C33;
    J = sub(I,{x1s1=>X1S1,x1s2=>X1S2,x1s3=>X1S3,
      x2s1=>X2S1,x2s2=>X2S2,x2s3=>X2S3,
      x31s1=>X31S1,x31s2=>X31S2,x31s3=>X31S3,
      x32s1=>X32S1,x32s2=>X32S2,x32s3=>X32S3,
      c11=>C11,c12=>C12,c13=>C13,
      c21=>C21,c22=>C22,c23=>C23,
      c31=>C31,c32=>C32,c33=>C33});
    J2 = sub(J,QQ[x11,x12,x13,x21,x22,x23,x311,x312,x313,x321,x322,x323]);
    --correctness check: the true point should be in the ideal
    print(sub(J2,{x11=>X1S1,x12=>X1S2,x13=>X1S3,
      x21=>X2S1,x22=>X2S2,x23=>X2S3,
      x311=>X31S1,x312=>X31S2,x313=>X31S3,
      x321=>X32S1,x322=>X32S2,x323=>X32S3}));
    print(dim J2);
)
```

Let V_1 be the variety in the code with $x^*_{n-2}, x^*_{n-1}, (x^*_n)^{(1)}, (x^*_n)^{(2)}$ and centers c_{n-2}, c_{n-1}, c_n considered as variables. Let $V_2 \subseteq \mathbb{C}^{7\times 3}$ be the variety where $x^*_{n-2}, x^*_{n-1}, (x^*_n)^{(1)}, (x^*_n)^{(2)}$ and c_{n-2}, c_{n-1}, c_n take values, i.e. it is defined by the constraint that $(x^*_n)^{(1)}, (x^*_n)^{(2)}$ lie on the same sphere centered at c_n. Let f be the projection from V_1 to V_2. The

`Macaulay2` code shows that the fiber of a random point of V_2 is 0-dimensional. By Theorem 5.1.2, the statement of the lemma holds for a generic element of V_2. □

Proof of Proposition 5.2.1. We want to show that for generic $x_1^*, \dots, x_n^*, y_1^*, \dots, y_n^*$, the solution set of (5.6) is finite. By contradiction, assume that this is not the case.

Fix generic $x_1^*, \dots, x_{n-1}^*, y_1^*, \dots, y_{n-1}^*$. Let V be the solution set of the following equations for all $1 \leq i \neq j \leq n-1$:

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D_{ij}^*,$$
$$\|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\|, \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\|,$$
$$\min_{z_i \in \{x_i, y_i\} \text{ for } i=1,2,3} (\textstyle\sum_{j \in \{1,2,3\}} (z_j - (z_1 + z_2 + z_3)/3)^2) = D_{123}^*.$$

We will consider $\pi_{n-1}(V)$, which by Lemma 5.2.3 is at most 1-dimensional constructible set. Let $k = l_{n-1}$ be the number of irreducible components of $\pi_n(V)$.

We also fix generic $(x_n^*)^{(1)}, \dots, (x_n^*)^{(k+1)}, (y_n^*)^{(1)}, \dots, (y_n^*)^{(k+1)}$ on S_n. Let $T^{(i)}$ be the set of points x_{n-1} on S_{n-1} such that there exists x_n on S_n satisfying

$$\|x_{n-2} - x_{n-1}\| = \|x_{n-2}^* - x_{n-1}^*\|, \|y_{n-2} - y_{n-1}\| = \|y_{n-2}^* - y_{n-1}^*\|,$$
$$\|x_{n-1} - x_n\| = \|x_{n-1}^* - (x_n^*)^{(i)}\|, \|y_{n-1} - y_n\| = \|y_{n-1}^* - (y_n^*)^{(i)}\|,$$
$$\min_{z_i \in \{x_i, y_i\} \text{ for } i=n-2,n-1,n} (\textstyle\sum_{j \in \{n-2,n-1,n\}} (z_j - (z_1 + z_2 + z_3)/3)^2) = D_{n-2,n-1,n}^*,$$

By Lemma 5.2.3, the sets $T^{(i)}$ are constructible. Moreover, by our assumption that the solution set of (5.1) is infinite, the intersections of $\pi_{n-1}(V)$ and $T^{(i)}$ are infinite, because they are equal to the projections of the solution set of (5.1) with $x_n^* = (x_n^*)^{(i)}$ to S_{n-1}.

However, this is a contradiction to Lemma 5.2.4 that states that the Zariski closures of $T^{(1)}, \dots, T^{(k+1)}$ intersect pairwise at finite number of points: The set $\pi_{n-1}(V)$ can be written as $\cup_{j=1}^{l_{n-1}} (W_j^{n-1} \backslash Z_j^{n-1})$ and $T^{(i)}$ can be written as $\cup_{j=1}^{l'} (W_j' \backslash Z_j')$. By the previous paragraph, there exist j and j' such that the intersection of $W_j^{n-1} \backslash Z_j^{n-1}$ and $W'_{j'} \backslash Z'_{j'}$ is infinite. Hence $W_j^{n-1} = W'_{j'}$, because otherwise $W_j^{n-1} \cap W'_{j'}$ would be 0-dimensional by [1, Proposition 10 in §9.4] and hence $(W_j^{n-1} \backslash Z_j^{n-1}) \cap (W'_{j'} \backslash Z'_{j'})$ would be finite. Since there are more different $T^{(i)}$ than W_j^{n-1}, then there exists 1-dimensional W_j^{n-1}, a component $W' \backslash Z'$ of $T^{(i')}$ and a component $W'' \backslash Z''$ of $T^{(i'')}$ such that $W' = W_j^{n-1}$ and $W'' = W_j^{n-1}$. The intersection of $W' \backslash Z'$ and $W'' \backslash Z''$ is $W_j^{n-1} \backslash (Z' \cup Z'')$ and hence it is 1-dimensional. This is a contradiction to Lemma 5.2.4, so it follows the solution set of (5.1) is finite. □

Bibliography

[1] David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer, Cham, fourth edition, 2015. An introduction to computational algebraic geometry and commutative algebra.

[2] Job Dekker. Gene regulation in the third dimension. *Science Signalling*, 319(5871), 2008.

[3] Kaifeng Jiang, Defeng Sun, and Kim-Chuan Toh. A partial proximal point algorithm for nuclear norm regularized matrix least squares problems. *Mathematical Programming Computation*, 6(3), 2014.

[4] Erez Lieberman-Aiden, Nynke L. van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, Bradley Bernstein, M. A. Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A. Mirny, Eric S. Lander, and Job Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome . *Science*, 326(5950), 2009.

[5] Sofia A Quinodoz, Noah Ollikainen, Barbara Tabak, Ali Palla, Jan Marten Schmidt, Elizabeth Detmar, Mason Lai, Alexander Shishkin, Prashant Bhat, Vickie Trinh, Erik Aznauryan, Pamela Russell, Christine Cheng, Marko Jovanovic, Amy Chow, Patrick McDonel, Manuel Garber, and Mitchell Guttman. Higher-order inter-chromosomal hubs shape 3-dimensional genome organization in the nucleus. 2017.

[6] Igor R. Shafarevich. *Basic algebraic geometry. 1*. Springer, Heidelberg, third edition, 2013. Varieties in projective space.

[7] Bas van Steensel and Andrew S. Belmont. Lamina-Associated Domains: Links with Chromosome Architecture, Heterochromatin, and Gene Repression. *Cell*, 169(5), 2017.

[8] F. Thoma, TH. Koller, and A. Klug. Involvement of histone H1 in the organization of the nucleosome and of the salt-dependent superstructures of chromatin. *Journal of Cell Biology*, 83(2), 1979.

[9] Yinyu Ye and Anthony Man-Cho So. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2), 2007.

[10] ZhiZhuo Zhang, Guoliang Li, Kim-Chuan Toh, and Wing-Kin Sung. 3D Chromosome Modeling with Semi-Definite Programming and Hi-C Data. *Journal of Computational Biology*, 20(11), 2013.