Graph Embeddings with Applications in Genomic Experiments.

by

William Casey

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Mathematics

New York University

May, 2002

Bud Mishra

(Dissertation Advisor)

Dedicated

to the memory of Jack Alphonse Casey, George W. Droege, Sr., Ida Doepke Droege, and Noella Tetrault Casey

Acknowledgments

I acknowledge my advisors as those that introduced me to the state of the art: Bud Mishra who taught me constructive mathematics, Greg Budzban who introduced me to applied mathematics, and Calvin Ahlbrandt who introduced me to the science of mathematics.

I acknowledge my teachers for good instructions and insights I've seen over the years while pursuing the degree. This list includes: B. Mishra, P. Lax, H. McKean, C. Peskin, M. Wigler, J. Schwartz, C. Chicone, Y. Latushkin, C. Ahlbrandt, J. Beem, W. Patula, G. Budzban, M. Wright, D. Kammler, S. Mohammed.

My Friends have been and are true friends: Ryan, Curtis, Larel, Chris, Gabe, Ellen, Ryan Cutter, Josh, Dave, Per, Jeff, Scott, Link, Zack, Dan, Leon, Shelly, Ben, Todd, Lara, Tami, Eric, Drew, Greg, Boone, Chad, Ian, Jenna, Dan Kegal, Joel, Dave, Jessica, Lane, Sam, Craig, Christina, Dough and Jason Bredahl.

And most of all I acknowledge my family, Mike, Julie, Megan, Patrick, Brian, Geroge, Jerry, Alvin, Barbara and Father Bun and especially Greg and Joan, Dennis, Caitlin, and Kennedy Casey.

Doctoral Dissertation Abstract

The optimazation problem called MATRIX–TO–LINE consists of finding an embedding of a edge weighted complete graph, specified by a positive symetric matrix W, into a one dimensional vector space whose outcome produces pair–wise distance matrix $\tilde{W}$ such that $\left|\left|W - \tilde{W}\right|\right|$ is minimized over all possible embeddings. The problem arises in applications in genomics. and is NP–Complete for the general case. We develop the problem in a geometric setting and show that the problem although NP–Complete admits hueristics solutions. We construct one such hueristic in the form of a dynamical system. We discuss two important applications in genomics where hueristics to this general problem can be made useful, the first application is called Probe–Mapping, the second is called RFLP–Phasing.

In Probe–Mapping a map of genomic features is determined by the best embedding of pairwise distance data. In this application we can make percise statments about how frequently hueristic algorithms determine correct answers.

In RFLP–Phasing similar ideas are used to determine likely configurations of a set of polymorphic markers.

Contents

List of Appendix

List of Figures

Chapter 1

Motivation

Let $\mathcal{H}_0$ be the set of $N \times N$ non-negative, symmetric matrices with zero diagonal. Let Π be the set of non-negative, symmetric matrices arising from pairwise distances from a set of N points on the real line. The optimization problem called MATRIX–TO–LINE is to find an element $\tilde{W} \in \Pi$ so that:

$$\tilde{W} = \operatorname{argmin}\left(\left|\left|\tilde{W} - W\right|\right|\right), \text{ for any given } W \tag{1.1}$$

Pairwise distance matrices arising from points in $\mathbb{R}$ of a given order form a linear subspace in $\mathcal{H}_0$. The MATRIX–TO–LINE problem is shown to be NP–Complete. Thus the hardness of the problem could be attributed to the super–polynomially number of orderings of N points.

The Problem has become a relevant problem in applications to genomic experiments which focus on the inference of genome structure based on experiments that provide only pairwise distance or orientation information.

In probe–mapping, probes are important substring markers that can be string–matched via hybridization experiments to genomic material. Experiments are designed to observe an estimate of pairwise genomic distances from probe to probe.

In RFLP–Phasing a set of *restriction enzyme site* markers on both chromosomes are considered. Data consists of a mix of genomic copies taken from small regions of one of the two chromosomes. From small variations in the length between restriction sites due to restriction fragment length polymorphism (RFLP) on the two chromosomes and from the large number of copies involved we get a view of how these restriction enzyme sites should be mapped on each chromosome. This separation of *genotype* data into the two chromosomes or *haplotypes* is the *phasing* problem.

Probe–Mapping and RFLP-Phasing share common elements as mathematical problems. In both problems experiments are designed to discover pairwise similarity of the genome structure, from the pairwise data which is local information, we wish to infer what types of genome structure is possible globally. Both problems require considering a very large configuration space, which is exponential in the number of features. Such a search is confirmed to be computationaly difficult in the sense of NP–Completeness. In addition both problems admit to heuristics. One point the thesis shall make is that in applications experiments can be designed to produce data that make such heuristics reliable.

The thesis is organized into three main chapters and one appendix. In the first chapter a motivation is given, along with definitions sufficient enough to understand the geometry of the optimization problem. Formal definitions of the optimization problem as well as the result that the problem is NP–complete. Also is a discussion of heuristics and a dynamical system heuristic is demonstrated. Mathematical definitions of a basic nature, and an expanded proof of NP-completeness of GRAPH–

EMBEDDING are placed in the appendix for reference.

The second chapter contains the explanation of Probe–Mapping and develops heuristics and analysis showing that with probability close to one the heuristics obtain the correct result. We model the outcome of hybridization experiments with a probability distribution and are able to show that the DNA–Hybridization technology can be used to map and place probe positions on the genome. This is accomplished by noting that the problem is solved with easy heuristic algorithms with some inputs. In fact the same heuristic algorithm work for a set of input that our model of input assigns high probability to. This is made precise by formalizing an algorithm and the analysis using Chernoff bounds to provide exponentially decaying bounds on the probability that the optima attained by our heuristic algorithm is not a correct answer. The heuristics were inspired by ideas from the heat equation.

The third chapter explains the structure of RFLP–Phasing and develops heuristics, analysis and examples. Experiments are designed to detect co–association of restriction fragment length polymorphisms on haplotypes in contiguous regions, we model this as a lattice–spin system, where the spin is either up or down, and is considered an attribute at each restriction site. It is interesting to view the global configuration as an unknown and each contig operation branching the search space in half directing the search into regions which are appropriate for observed data. In the application of RFLPs we found an interesting connection between continuous groups and Maximum Likelihood estimates for pairwise–events. This is treated in a section on RFLPs.

1.1 Mathematical Definitions

In this section we build up the mathematical definitions to view the optimization problem MATRIX–TO–LINE as a geometric problem of finding the intersection of an ellipsoid ball and a star shaped set called Π. The set Π is the set union of subspaces determined by a permutation on N objects, that is each permutation $\sigma \in S_N$ determines a subspace of $\mathcal{H}_0$ denoted Π_σ having dimension $(N-1)$. Since there are $N!$ permutations in the permutation group S_N and the dimension of $\mathcal{H}_0$ is $\binom{N}{2}$ there is considerable overlapping. The structure of overlapping is explored and algorithms for computing the overlap and sums of such subspaces are given.

The optimization problem is shown to be NP–complete by reduction to GRAPH EMBEDDING which is reduced to 3–SAT. We discuss heuristics and present one heuristic as a dynamical system.

Definition 1.1.1 Linear space of linear transformations
Let $L(N, M)$ denote the set of linear transformations mapping vectors in $\mathbb{R}^N$ into $\mathbb{R}^M$ Let $L(N)$ denote the set of linear transformations of $\mathbb{R}^N$ into itself. The elements of $L(N, M)$ are made into a linear space over $\mathbb{R}$ by defining $+, \cdot$ as:

$$
\begin{aligned}
+ &: L(N, M) \times L(N, M) \to L(N, M) : (A, B) \mapsto A + B \\
&\quad A + B(x) = A(x) + B(x) \\
\cdot &: L(N, M) \times \mathbb{R} \to L(N, M) : (A, k) \mapsto kA \\
&\quad kA(x) = A(kx)
\end{aligned}
$$

Given an element A in $L(N, M)$ the *kernel* of A is defined to be the set $\{x \in \mathbb{R}^M : A(x) = 0\}$. This set is also a linear subspace.

All entries $a_{ij} : i,j \in [1 \dots N]$ of a matrix A in $L(N)$ are necessarily real numbers, and often will be denoted as $\mathbb{R}_{N\times N}$, indicating the $N \times N$ matrices over $\mathbb{R}$.

Definition 1.1.2 Isomorphism of linear spaces

An isomorphism is a one–to–one and onto mapping which maps sums onto sums and scalar multiples onto scalar multiples. X and Y are said to be isomorphic if there exists a mapping ϕ one–to–one and onto so that:

$$\phi(x_1 + k \cdot x_2) = \phi(x_1) + k \cdot \phi(x_2) \quad \forall x_1, x_2 \in X, k \in K \tag{1.2}$$

where the $+, \cdot$ operations on the left hand side are operations in X and the ones on the right hand side are operations in Y.

Claim 1.1.1 *The Basic Isomorphism: The space $L(N)$ is isomorphic to the space $\mathbb{R}^{N^2}$*

Proof 1 [Proof of claim 1.1.1] We construct an isomorphism called Φ, and show that it is one–to–one and onto. Finally, we show that it preserves sums and products. let $\nu(i,j) = (j-1)N + i$, Let $A \in L(N)$ and we define Φ by its components:

$$\Phi : L(N) \to \mathbb{R}^{N^2} : A \mapsto \Phi(A) : \Phi(A)_{\nu(i,j)} = A_{ij} \tag{1.3}$$

Φ is one–to–one: Suppose that $A \neq B \in L(N)$ but $\Phi(A) = \Phi(B)$ this would imply that $\Phi(A)_{\nu(i,j)} = \Phi(B)_{\nu(i,j)}\ \forall\ \nu(i,j)$ but the map ν is one–to–one and onto as a map of $[1 \dots N]^2$ to $[1 \dots N^2]$, Hence we have $A_{ij} = B_{ij}\ \ \forall i,j$, but this is a contradiction. Φ is onto: Suppose that $V \in \mathbb{R}^{N^2} \setminus \Phi(L(N))$ but consider the matrix $\tilde{A}$ which is defined as follows:

$$\tilde{A}_{i,j} = V_{(j-1)N+i} \tag{1.4}$$

clearly the image of $\tilde{A}$ must be V, but this is a contradiction.

Φ preserves sums and products: given $A, B \in L(N)$ and $k \in K$ we have:

$$\Phi(A + k \cdot B)_{\nu(i,j)} = (A + k \cdot B)_{ij} = A_{ij} + k \cdot B_{ij} = \Phi(A)_{\nu(i,j)} + k \cdot \Phi(B)_{\nu(i,j)} \tag{1.5}$$

□

The basic isomorphism concatenates matrix columns into one long column. We will make use of the isomorphism frequently, as well as linear transformations in the space $L(N^2)$, where conditions of a matrix are summarized by solutions spaces to problems of this form:

$$Lx = 0 \tag{1.6}$$

where $L \in L(N^2)$. The set of elements in $L(N)$ which satisfy a set of linear conditions in their entries will become isomorphic to the kernel of L.

1.2 Geometry and Important Subsets and Subspaces in $L(N)$

In this section we explore some of the structures important in the consideration of pairwise–distance data. Some of the structures explored will be sets, some will be convex sets or cones, others will be subspaces. When there is a subspace to consider we construct an operator in the space $L(N^2, N^2)$ operating on the isomorphic space $\mathbb{R}^{N^2}$, and whose kernel is isomorphic to the subspace.

The intended goal of these straightforward techniques is to observe the geometry involved for optimization problem requiring the embedding of points into a one dimensional subspace. This optimization problem is called MATRIX–TO–LINE.

1.2.1 Basic Notions

In a linear space the *Minkowski Sum* of sets S_1, S_2 is the set $\{s_1 + s_2 : s_1 \in S_1, s_2 \in S_2\}$, and is denoted $S_1 + S_2$.

A *Convex Set* C is a set with the property that if $x, y \in C$, then all the elements on the line segment joining x to y are also contained in C. That is:

$$x, y \in C \Rightarrow \lambda x + (1-\lambda)y \in C \quad \forall \lambda \in [0,1] \tag{1.7}$$

The line segment is denoted $[x : y] = \{\lambda x + (1-\lambda)y \in C : \lambda \in [0,1]\}$

The smallest convex set C containing an arbitrary set S is called the *convex hull* of S, denoted by $[S]$.

A *Cone Set* is a set C that is convex and has the property that:

$$C + C \subset C$$

$$x \in C, k \in \mathbb{R}^+ \Rightarrow kx \in C$$

The order relation of $\mathbb{R}$, is determined by a cone set. Similarly a cone set in a linear space can be used to order vectors.

Cones are closed under union, intersection, and Minkowski sum.

Further if P is a cone in $\mathbb{R}^n$, and $M \in L(n, m)$ then $M(P)$ is a cone in $\mathbb{R}^m$. If P is a cone in $\mathbb{R}^m$, then $\{x : M(x) \in P\}$ is a cone in $\mathbb{R}^n$.

In a linear space a set C is a *linear flat* if the following condition holds:

$$x, y \in C \Rightarrow \lambda x + (1-\lambda) y \in C \quad \forall \lambda \in \mathbb{R} \tag{1.8}$$

The smallest flat containing an arbitrary set S is called the *affine hull* of S, it is denoted by $\text{Aff}\,(S)$.

If a flat contains the point 0 then it is a subspace. Any flat in $\mathbb{R}^N$ can be written as the solution space of a linear system $Ax = b$ with $A \in L(N), b \in \mathbb{R}^N$. When the vector $b = 0$ then the flat is a subspace, otherwise it is a parallel translation of a the solution space of $Ax = 0$ by any z so that $Az = b$.

The smallest subspace containing an arbitrary set S is denoted by $\langle S \rangle$.

Example 1.2.1 *In $\mathbb{R}^n$ the set of vectors with strictly positive entries is a cone set. In $\mathbb{R}^n$ the set of all vectors x for which $||x||_p < \delta$ is a convex set when $p \geq 1$, where the p–norm $||x||_p$ is defined as $(\sum {x_i}^p)^{\frac{1}{p}}$*
In $L(N)$ the set of positive matrices (all entries are positive) is a cone.
In $L(N)$ the set of bi–stochastic matrices (rows sum to one, columns sum to one) is a convex set.

The transpose of a matrix $A \in L(N)$ is denoted A^t and is defined by the interchange of columns and rows:

$$A^t_{ij} = A_{ji} \tag{1.9}$$

1.2.2 Spaces of $\mathcal{H}_0$

Symmetric Matrices

A matrix is real–symmetric if $A \in \mathbb{R}^{N\times N}$ and:

$$A^t = A \tag{1.10}$$

Let $\mathcal{H}$ denote the subset of $L(N)$ which are real–symmetric. This family of matrices will be used to represent pair–wise–distance data in the applications.

This class of real–symmetric matrices are a subspace in $L(N)$, let $a, b \in \mathcal{H}$:

$$(a_{ij} = a_{ji}) \wedge (b_{ij} = b_{ji}) \Rightarrow a_{ij} + kb_{ij} = a_{ji} + kb_{ji} \tag{1.11}$$

This equation holding over the range of indices i, j shows that the space $\mathcal{H}$ is closed under the $+, \cdot$ operations of linear space $L(N)$.

Reformulating the condition of symmetry in the entries of a matrix $a \in \mathcal{H}$ as linear equations we get:

$$a_{ij} - a_{ji} = 0 \quad i, j \in [1, \ldots N]. \tag{1.12}$$

We can summarize the space of symmetric matrices as the kernel of a matrix in $L(N^2, N^2)$ acting on the space $\mathbb{R}^{N^2}$ isomorphic to $L(N)$.

We demonstrate that construction: Let us define an index function converting the indices of a 2–dimensional array to a 1–dimensional array.

$$(i, j) \mapsto \nu = (j-1) * N + i \tag{1.13}$$

$$\nu \mapsto (\nu_i, \nu_j) = (\nu \ \mathrm{mod} N, 1 + \lfloor \frac{\nu}{N} \rfloor) \tag{1.14}$$

$$\tag{1.15}$$

Then define $\Gamma_s \in L(N^2, N^2)$ by :

$$\left[\Gamma_s\right]_{\nu,k} = \begin{cases} 1 \text{ if } \nu_i = k, \nu_i \neq \nu_j \\ -1 \text{ if } \nu_j = k, \nu_i \neq \nu_j \\ 0 \text{ otherwise} \end{cases} \tag{1.16}$$

Define the solution space:

$$S_s = \{x \in \mathbb{R}^{N^2} : \Gamma_s x = 0\} \tag{1.17}$$

S_s is spanned by $\Phi(\mathcal{H})$.

Symmetric Matrices with Zero Diagonal

The set of matrices in $\mathcal{H}$ with zeros on the diagonal is denoted $\mathcal{H}_0$ and is a subspace of $\mathcal{H}$. We can summarize the space of symmetric matrices with zero diagonal as a solution space of a matrix in $L(N^2, N^2)$ acting on the space $\mathbb{R}^{N^2}$ isomorphic to $L(N)$. Using the map in equation (1.13). Define $\Gamma_0 \in L(N^2, N^2)$ by :

$$\left[\Gamma_0\right]_{\nu,k} = \begin{cases} 1 \text{ if } \nu_i = \nu_j = k \\ 0 \text{ otherwise} \end{cases} \tag{1.18}$$

Let S_0 be the solution space $\{x \in \mathbb{R}^{N^2} : \Gamma_0 x = 0\}$, and let:

$$S_{s0} = S_s \cap S_0 \tag{1.19}$$

S_{s0} is spanned by $\Phi(\mathcal{H}_0)$.

Positive Symmetric Matrices with zero diagonal

Let Υ_+ denote the subset of $\mathcal{H}_0$ which are real–symmetric and positive. This set Υ_+ is not a linear subspace of $L(N)$ but only a cone set and represents the pairwise distance data. All experiments designed in the applications produce data which in the worst case belongs to this class. We describe the cone set in $\mathbb{R}^{N^2}$.

We consider the condition that an element of a symmetric matrix is non–negative, letting $a \in \Upsilon_+$

$$a_{ij} \geq 0 \tag{1.20}$$

Define $\Gamma_1 \in L(N^2, N^2)$ by:

$$\Big[\Gamma_1\Big]_{\nu,k} = \begin{cases} 1, & \text{if } \nu = k; \\ 0, & \text{otherwise.} \end{cases} \tag{1.21}$$

Let $P_{N^2} = \{x \in \mathbb{R}^{N^2} : x_\nu \geq 0 \ \ \forall \nu\}$ be the non-negative cone in $\mathbb{R}^{N^2}$. Then the pre–image of a cone $P_1 = \{x \in \mathbb{R}^{N^2} : \Gamma_1(x) \in P_{N^2}\}$ is a cone in $\mathbb{R}^{N^2}$.

$$P_{1s0} = P_1 \cap S_{s0}$$

Since subspaces are cones then P_{1s0} is easily seen to be a cone in $\mathbb{R}^{N^2}$. The set (written as a Minkowski sum)

$$\Phi(\mathcal{H}_0) + P_{1s0} \tag{1.22}$$

is the cone set in $\mathbb{R}^{N^2}$ isomorphic to Υ_+.

Pairwise distance Matrices

Given arbitrary points $\{x_1, x_2, \ldots, x_N\}$ in a finite dimensional normed linear space $\langle X, ||\cdot||_p\rangle, p \geq 1$, we define the *Pairwise Distance Matrix* called W by assigning $w_{ij} = d(x_i, x_j) = ||x_i - x_j||_p$, $\forall 1 \leq i, j \leq N$. The elements of the matrix must inherit some properties from the norm:

$$w_{ij} = d(x_i, x_j) \geq 0 \text{ Non–Negative}$$

$$w_{ii} = d(x_i, x_i) = 0 \text{ Zero Diagonal}$$

$$w_{ij} = d(x_i, x_j) = d(x_j, x_i) = w_{ji} \text{ Symmetry}$$

$$w_{ik} = d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k) = w_{ij} + w_{jk} \text{ Triangle Inequality}$$

Let Υ be the set of pair–wise–distance matrices ranging over any N points in any finite dimensional normed linear space.

Claim 1.2.1 Υ *is a cone.*

Proof 2 (Proof of claim 1.2.1) We will show closure under the operations $+, \cdot$ for the triangle inequality: For $a, b \in \Upsilon$:

$$(a_{ij} + a_{jk} \geq a_{ik}) \wedge (b_{ij} + b_{jk} \geq b_{ik})$$

$$\Rightarrow a_{ij} + a_{jk} + b_{ij} + b_{jk} \geq a_{ik} + b_{ik}$$

$$\Rightarrow (a+b)_{ij} + (a+b)_{jk} \geq (a+b)_{ik}$$

$$k > 0, (a_{ij} + a_{jk} \geq a_{ik})$$

$$\Rightarrow ka_{ij} + ka_{jk} \geq ka_{ik}$$

□

We construct the isomorphic cone in $\mathbb{R}^{N^2}$.

Let S_{s0} be as in equation(1.19), the result of constructing solution spaces for Γ_s, Γ_0 in equations (1.16 , 1.18) earlier. S_{s0} contains points satisfing the equality constraints of zero diagonal and symmetry.

Let Γ_1 be defined as in (1.21), this covers the positive entry condition.

Now we focus on the triangle inequality.

Letting η be a index function converting triples of the form (i, j, k) into a single index over the range $[1 \dots N^3]$ given by the map:

$$(i, j, k) \to \eta = (k-1) * N^2 + (j-1) * N + i \tag{1.23}$$

$$\eta \to (\eta_i, \eta_j, \eta_k) = \Big(\eta \bmod N, 1 + \lfloor \frac{\eta}{N} \rfloor \bmod N, 1 + \lfloor \frac{\eta}{N^2} \rfloor\Big) \tag{1.24}$$

Define $\Gamma_2 \in L(N^3, N^2)$ as:

$$\Big[\Gamma_2\Big]_{\nu,v} = \begin{cases} 1 \text{ if } \eta_i = v, \eta_i \neq \eta_j, \eta_i \neq \eta_k, \eta_j \neq \eta_k \\ 1 \text{ if } \eta_j = v, \eta_i \neq \eta_j, \eta_i \neq \eta_k, \eta_j \neq \eta_k \\ -1 \text{ if } \eta_k = v, \eta_i \neq \eta_j, \eta_i \neq \eta_k, \eta_j \neq \eta_k \\ 0 \text{ otherwise} \end{cases} \tag{1.25}$$

Let P_{2f} be the non-negative cone in $\mathbb{R}^{N^3}$. Then the set $P_2 = \{x \in \mathbb{R}^{N^2} : \Gamma_2(x) \in P_{2f}\}$ is a cone in $\mathbb{R}^{N^2}$.

As $\Phi(\Upsilon_+)$ is the cone from equation (1.22) given by: $\Phi(\mathcal{H}_0) + P_{0,N^2}$, Note that upon intersection with cone P_2 we satisfy all equality and inequality conditions.

$$P_\triangle = \Phi(\mathcal{H}_0) + P_{0,N^2} \cap P_2 \tag{1.26}$$

And the Isomorphic space in $\mathbb{R}^{N^2}$ is the Minkowski sum given by:

$$\Phi(\Upsilon) + P_{\triangle} \tag{1.27}$$

Pairwise Distance Matrices of Points on a Line of a Given Order

Let us take N points $x_1 < x_2 < \ldots < x_N$ from the line $\mathbb{R}^1$.

Since all points are vectors in $\mathbb{R}^1$, pairwise angles are in the set $\{0, \pi\}$. All triples x_i, x_j, x_k with $i < j < k$ will attain equality in the triangle inequality:

$$i < j < k \Rightarrow (a_{ij} + a_{jk} = a_{ik}) \tag{1.28}$$

Let Π_e be the class of all matrices with the condition that $a \in \Pi_e$ implies $i < j < k \Rightarrow (a_{ij} + a_{jk} = a_{ik})$.

Claim 1.2.2 $\Pi_e \leq L(N)$

Proof 3 (Proof of claim 1.2.2) Let $a, b \in \Pi_e$:

$$\begin{aligned}
&i < j < k \Rightarrow a_{ij} + a_{jk} = a_{ik} \\
&i < j < k \Rightarrow b_{ij} + b_{jk} = b_{ik} \Rightarrow \gamma b_{ij} + \gamma b_{jk} = \gamma b_{ik} \\
&\quad \Rightarrow a_{ij} + a_{jk} + \gamma b_{ij} + \gamma b_{jk} = a_{ik} + \gamma b_{ik} \\
&\quad \Rightarrow (a + \gamma b)_{ij} + (a + \gamma b)_{jk} = (a + \gamma b)_{ik}
\end{aligned}$$

$\square$

Let Π_e^+ be the class of all matrices arising from pairwise distance data of any N points $\{x_1, x_2, \ldots, x_N\}$ on a line so that $x_{e_1} < x_{e_2} < \ldots < x_{e_N}$ (here $e_i = i$ the identity permutation in the permutation group S_N).

Claim 1.2.3 Π_e^+ *is a Cone in* $L(N)$

Proof 4 (Proof of claim 1.2.3) All element entries of Π_e^+ are non–negative. The diagonal entries of Π_e^+ are zero. Thus, cone closure property of Π_e^+ is obvious. $\square$

Further all entries in the matrix are determined by the upper-diagonal containing $N-1$ members.

We can represent the space of matrices in Π_e as a solution space of a matrix in $L(N^3, N^2)$ acting on the space $\mathbb{R}^{N^2}$ isomorphic to $L(N)$.

Let us translate these stiff triangle equality conditions into a matrix equation in the isomorphic space $\mathbb{R}^{N^2}$.

Let $\eta \leftrightarrow (i, j, k)$ be the index map defined in equations (1.23, 1.24).

Now define $\Gamma_e \in L(N^3, N^2)$

$$\Big[\Gamma_e\Big]_{\nu,v} = \begin{cases} 1 \text{ if } v = \nu_i, (\nu_i < \nu_j < \nu_k) \\ 1 \text{ if } v = \nu_j, (\nu_i < \nu_j < \nu_k) \\ -1 \text{ if } v = \nu_k, (\nu_i < \nu_j < \nu_k) \\ 0 \text{ otherwise} \end{cases} \tag{1.29}$$

Let S_e be the solution space:

$$S_e = \{x \in \mathbb{R}^{N^2} : \Gamma_e x = 0\} \tag{1.30}$$

Finally the space S_e is $\Phi(\Pi_e)$. With P_{N^2} the positive cone in $\mathbb{R}^{N^2}$ The cone set

$$P_{N^2} \cap S_e \tag{1.31}$$

is the isomorphic set of Π_e^+.

Pairwise distance Matrices of points on a line of a fixed ordering

Let us take N points $x_1, x_2, \ldots, x_N$ from the line $\mathbb{R}^1$, and let $\sigma \in S_N$ be the permutation of indices which sorts the set of vectors:

$$x_{\sigma_1} < x_{\sigma_2} < \ldots < x_{\sigma_N} \tag{1.32}$$

The triangle equality constraints are:

$$\sigma_i < \sigma_j < \sigma_k \Rightarrow (a_{\sigma_i\sigma_j} + a_{\sigma_j\sigma_k} = a_{\sigma_i\sigma_k}) \tag{1.33}$$

Let Π_σ be the class of all matrices with the condition that $a \in \Pi_\sigma$ if and only if $\sigma_i < \sigma_j < \sigma_k \Rightarrow (a_{\sigma_i\sigma_j} + a_{\sigma_j\sigma_k} = a_{\sigma_i\sigma_k})$

Claim 1.2.4 $\Pi_\sigma \leq L(N)$

Proof 5 (Proof of claim 1.2.4) Let $a, b \in \Pi_\sigma$:

$$\begin{aligned}
&\sigma_i < \sigma_j < \sigma_k \Rightarrow a_{\sigma_i\sigma_j} + a_{\sigma_j\sigma_k} = a_{\sigma_i\sigma_k} \\
&\sigma_i < \sigma_j < \sigma_k \Rightarrow b_{\sigma_i\sigma_j} + b_{\sigma_j\sigma_k} = b_{\sigma_i\sigma_k} \Rightarrow \gamma b_{\sigma_i\sigma_j} + \gamma b_{\sigma_j\sigma_k} = \gamma b_{\sigma_i\sigma_k} \\
&\quad \Rightarrow a_{\sigma_i\sigma_j} + a_{\sigma_j\sigma_k} + \gamma b_{\sigma_i\sigma_j} + \gamma b_{\sigma_j\sigma_k} = a_{\sigma_i\sigma_k} + \gamma b_{\sigma_i\sigma_k} \\
&\quad \Rightarrow (a + \gamma b)_{\sigma_i\sigma_j} + (a + \gamma b)_{\sigma_j\sigma_k} = (a + \gamma b)_{\sigma_i\sigma_k}
\end{aligned}$$

□

Let Π_σ^+ be the class of all matrices arising from pairwise distance data of any N points $\{x_1, x_2, \ldots, x_N\}$ on a line so that $x_{\sigma_1} < x_{\sigma_2} < \ldots < x_{\sigma_N}$ where $\sigma \in S_N$ is the permutation in the permutation group S_N, which sorts the set.

Claim 1.2.5 Π_σ^+ *is a Cone in* $L(N)$

Proof 6 (Proof of claim 1.2.5) All element entries of Π_σ^+ are non–negative. The diagonal entries of Π_σ^+ are zero, hence cone closure property of Π_σ^+ is obvious. $\square$

We can represent the space of matrices in Π_σ as a solution space of a matrix in $L(N^3, N^2)$ acting on the space $\mathbb{R}^{N^2}$ isomorphic to $L(N)$.

We translate these triangle equality stiffness conditions into a matrix equation in the isomorphic space $\mathbb{R}^{N^2}$.

Let $\eta \leftrightarrow (i, j, k)$ be the index map defined in equations (1.23, 1.24).

Now define $\Gamma_\sigma \in L(N^3, N^2)$

$$\left[\Gamma_\sigma\right]_{\nu,v} = \begin{cases} 1 \text{ if } v = \nu_i, (\sigma_{\nu_i} < \sigma_{\nu_j} < \sigma_{\nu_k}) \\ 1 \text{ if } v = \nu_j, (\sigma_{\nu_i} < \sigma_{\nu_j} < \sigma_{\nu_k}) \\ -1 \text{ if } v = \nu_k, (\sigma_{\nu_i} < \sigma_{\nu_j} < \sigma_{\nu_k}) \\ 0 \text{ otherwise} \end{cases} \tag{1.34}$$

Let S_σ be the solution space:

$$S_\sigma = \{x \in \mathbb{R}^{N^2} : \Gamma_\sigma x = 0\} \tag{1.35}$$

Finally the space S_σ is $\Phi(\Pi_\sigma)$. With P_{N^2} the positive cone in $\mathbb{R}^{N^2}$ The cone set

$$P_{N^2} \cap S_\sigma \tag{1.36}$$

is the isomorphic set of Π_σ^+.

Definition 1.2.1 Degrees of Freedom

A set S in $L(N)$ is said to have J degrees of freedom if its affine hull has dimension J.

Lemma 1.2.6 *Degrees of Freedom: Degrees of freedom is preserved under isomorphism*

Proof 7 (Proof of lemma 1.2.6) Dimension of subspaces are preserved under isomorphism. $\square$

Lemma 1.2.7 *For any $\sigma \in S_N$ The space Π_σ has $N-1$ degrees of freedom.*

Proof 8 (Proof of lemma 1.2.7) We prove this for Π_e and reduce all other cases to Π_e by means of permuting rows and columns. Let $a \in \Pi_e$. $a_{i,i+j}$ due to the triangle equalities we can write the equation:

$$\begin{aligned} a_{i,i+j} &= a_{i,i+1} + a_{i+1,i+j} \\ &= a_{i,i+1} + a_{i+1,i+2} + a_{i+2,i+j} \\ &\vdots \\ &= a_{i,i+1} + a_{i+1,i+2} + \ldots + a_{i+j-1,i+j} \end{aligned}$$

for any $j > i$. This yields a system that can be solved over the free variables $a_{i,i+1}, i \in [1 \ldots (N-1)]$

General Case: Let $a \in \Pi_\sigma$, let $\mathcal{J}_\sigma$ be the permutation matrix obtained by permuting rows and columns by σ. Consider the matrix $b = \mathcal{J}_\sigma a \mathcal{J}_\sigma$ One can see that the matrix $b \in \Pi_e$ and hence has $N-1$ degrees of freedom. Note that $\mathcal{J}$ is an isomorphism and thus, it must have $N-1$ degrees of freedom as well. $\square$

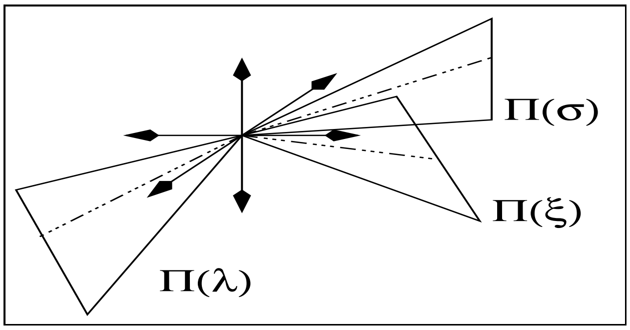

Figure 1.1: Permutation spaces

1.2.3 Permutation Spaces

So at this point a picture of the geometry of the these 'permutation spaces' starts to emerge: If each point $x_{\sigma_1} < x_{\sigma_2} < \ldots < x_{\sigma_N}$ is distinct then their associated pairwise–distance matrix sits in a linear subspace $\Pi_\sigma \leq \Upsilon$ of rank $N-1$. Thus for every $\sigma \in S_N$ there is a subspace fanning out of the origin in Υ which contains these data.

But consider the fact that $|S_N| = N!$ and since the rank of Υ is $\binom{N}{2}$, there surly must be some overlap.

Clearly only the origin belongs to the intersection of kernel spaces $\Gamma_e \cap \Gamma_\sigma$, unless $\sigma = e$, σ is the reversal of e, or there is some degeneracy in some of the dependent variables.

Intersections of Permutation Spaces : We give an algorithm for constructing a basis for the intersection of two such permutation spaces Π_{σ_1} and Π_{σ_2}. The clearest way to view this process is in the isomorphic space.

Algorithm 1 *We construct a matrix in $L(2N^3, N^2)$,*

$$\begin{bmatrix} \begin{bmatrix} \Gamma_{\sigma_1} \end{bmatrix} \\ \begin{bmatrix} \Gamma_{\sigma_2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \tag{1.37}$$

And construct a solution space for x using the LU factorization. The solution space is specified in terms of a basis. Let $\{b_1, b_2, \ldots, b_h\}$ be the basis for the solution subspace, convert back to a basis in $L(N)$ by Φ^{-1}. This provides a basis of elements in $L(N)$ describing the overlap of two permutation spaces.

This algorithm can be used inductively to compute the intersection of any finite number of permutation spaces.

Minkowski Sums of Permutation Spaces: We give an algorithm for constructing a basis for the Minkowski sum of two such permutation spaces Π_{σ_1} and Π_{σ_2}. The clearest way to view this process is in the Isomorphic space.

Algorithm 2 *We construct the matrix in $L(N^3, N^2)$, specified by equation(1.34):*

$$\begin{bmatrix} \Gamma_{\sigma_1} \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \tag{1.38}$$

And construct a solution space for x using the LU factorization. The solution space is specified in terms of a basis. Let $\{b_1^{(1)}, b_2^{(1)}, \ldots, b_{h_1}^{(1)}\}$ be the basis for the solution subspace. Next do the same thing for the matrix equation $\begin{bmatrix} \Gamma_2 \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = 0$ constructing a basis of its solution space: $\{b_1^{(2)}, b_2^{(2)}, \ldots, b_{h_2}^{(2)}\}$. Now we construct a matrix in $L(N^2, h_1 + h_2)$ with columns equal to the basis vectors. Setting up the *matrix equation:*

$$\begin{bmatrix} \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ b_1^{(1)} & b_2^{(1)} & \dots & b_{h_1}^{(1)} & b_1^{(2)} & b_2^{(2)} & \dots & b_{h_2}^{(2)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \tag{1.39}$$

finding a solution space for this problem in terms of basis elements $\{b_1^{(3)}, b_2^{(3)}, \dots b_{h_3}^{(3)}\}$. *Next we need only find a spanning set for the space*

$$\langle\{b_1^{(1)}, b_2^{(1)}, \dots, b_{h_1}^{(1)}, b_1^{(2)}, b_2^{(2)}, \dots, b_{h_2}^{(2)}\}\rangle \setminus \langle\{b_1^{(3)}, b_2^{(3)}, \dots b_{h_3}^{(3)}\}\rangle \tag{1.40}$$

This provides a basis in $\mathbb{R}^{N^2}$ *which spans the Minkowski sum of the two kernels, converting the basis back to* $L(N)$ *describes the Minkowski sum of two permutation spaces.*

The algorithm can be extended to compute the spans of a Minkowski Sum of any finite set of permutations Q, by iterating the previous algorithm and letting the set $\{b_1^{(1)}, b_2^{(1)}, \dots b_{h_1}^{(1)}\}$ be the running basis carried over as the solution of the previous problem.

$$\sum_{\sigma\in Q} \Pi_\sigma \leq \Upsilon \tag{1.41}$$

Pairwise distance Matrices of points on a line of unknown ordering

Since all points are vectors in $\mathbb{R}^1$ we know that pairwise angles are in the set $\{0, \pi\}$. All triples x_i, x_j, x_k will satisfy one of three triangle equalities.

Let us define a set of constraints among entries of a matrix:

$$C_{ijk} : A \to (a_{ij} + a_{jk} = a_{ik})$$

$$C_{jki} : A \to (a_{jk} + a_{ki} = a_{ji})$$

$$C_{kij} : A \to (a_{ki} + a_{ij} = a_{kj})$$

Now define a condition that one of the three are satisfied:

$$C_{(ijk)\triangle=} = (C_{ijk} \vee C_{jki} \vee C_{kij}) \tag{1.42}$$

We define a class of matrices representing pair–wise distances of N points that are found on a 1–d subspace in a normed linear space.

$$\Pi = \wedge_{i<j<k} C_{(ijk)\triangle=}(\Upsilon) \tag{1.43}$$

When a relation $\mathcal{R}_{ijk}$ holds then a point indexed by j can be seen to be in the convex hull of points indexed by i, k.

Lemma 1.2.8 *$w \in \Pi$ if and only if there is a $\sigma \in S_N$ so that $w \in \Pi_\sigma$*

Proof 9 (Proof of lemma 1.2.8) Let $\mathcal{R}_{ijk}$ for $i < j < k$ be a condition that satisfies the condition $C_{(ijk)\triangle=}$. Let $L[i:j] = \{k : (w_{kj} = w_{ki} + w_{ij})\}$, note condition $\mathcal{R}_{kij}$ and $\mathcal{R}_{jik}$ will imply that $k \in L[i:j]$.

sub claim 1

$$k \in L[i:j] \cap L[m:i] \Rightarrow k \in L[m:j] \text{ and } m \in L[i:j] \tag{1.44}$$

Π is defined to be a subset of Υ so each triple obeys the triangle inequality, hence:

$$w_{kj} = w_{ki} + w_{ij} \text{ as } k \in L[i:j] \tag{1.45}$$

$$= w_{km} + w_{mi} + w_{ij} \text{ as } k \in L[m:i] \tag{1.46}$$

$$\geq w_{km} + w_{mj} \text{ as } w_{mj} \leq w_{mi} + w_{ij} \tag{1.47}$$

$$\tag{1.48}$$

but also the triangle inequality gives that $w_{kj} \leq w_{km} + w_{mj}$ and we conclude:

$$w_{kj} = w_{km} + w_{mj} \text{ and } k \in L[m:j] \tag{1.49}$$

To prove that $m \in L[i:j]$:

$$w_{mi} = w_{ki} - w_{km} \text{ as } k \in L[m:i] \tag{1.50}$$

$$w_{ij} = w_{kj} - w_{ki} \text{ as } k \in L[i:j] \tag{1.51}$$

$$w_{kj} = w_{km} - w_{mj} \text{ as } k \in L[m:j] \tag{1.52}$$

$$\tag{1.53}$$

By eliminating variable w_{ki} followed by w_{kj} we find the relation:

$$w_{ij} = w_{mj} - w_{mi} \tag{1.54}$$

Thus showing that $m \in L[i:j]$, and concluding our sub-claim.

We conclude that relations $\tilde{\mathcal{R}}_{kmj}$ and $\tilde{\mathcal{R}}_{mij}$ are satisfied in addition to whatever relations are specified on triplets k, m, j and m, i, j, by w's inclusion in Π.

sub claim 2

If $y \in L[i:j]$ and $i \in L[x:j]$ then $y \in L[x:j]$.

$$
\begin{aligned}
w_{kj} &= w_{ki} + w_{ij} && (1.55)\\
&= w_{ki} + w_{ix} + w_{xj} \text{ as } x \in i \in L[x:j] && (1.56)\\
&\geq w_{kx} + w_{xj} \text{ as } w_{ki} + w_{ix} \geq w_{kx} && (1.57)\\
& && (1.58)
\end{aligned}
$$

This concludes Sub claim 2.

sub claim 3

If $b \in L[i:j]$ and $y \in L[i:j]$ and $y \notin L[b:j]$ and $w_{ij} \neq 0$ then $b \in L[y:j]$. If $y \notin L[b:j]$ then either $\mathcal{R}_{byj}$ or $\mathcal{R}_{bjy}$ must be satisfied for a $w \in \Pi$. $\mathcal{R}_{byj}$ implies $b \in L[y:j]$ which is in contradiction to our assumption, so we must investigate what happens if $b \in L[j:y]$

$$
\begin{aligned}
w_{bj} &= w_{bi} + w_{ij} && (1.59)\\
&\geq w_{by} + w_{yi} + w_{ij} && (1.60)\\
&\geq w_{ky} + w_{yj} \text{ as } y \in L[i:j] && (1.61)
\end{aligned}
$$

If $b \notin L[j:y]$ then we will have:

$$
\begin{aligned}
w_{by} &= w_{bj} + w_{jy} && (1.62)\\
& w_{by} + w_{yj} + w_{jy} && (1.63)\\
&\Rightarrow w_{jy} = 0 && (1.64)\\
&\Rightarrow w_{yi} + w_{ij} = 0 && (1.65)\\
& && (1.66)
\end{aligned}
$$

contradicting the assumption that $w_{ij} \neq 0$.

We outline an algorithm that will find a permutation $\sigma \in S_N$ satisfying all relations. We maintain a set of indices S for which every relation holds, at every step we may increase the size of S by another index and show that all relations hold after the insertion. We say a list of indices l has property L if:

$$\begin{cases} \text{size}(l) < 3 \\ l_1 \in L[l_2 : l_3], \text{size} l = 3 \\ i \leq j \leq k \Rightarrow l_i \in L[l_j : l_k], \text{size}(l) > 3 \end{cases} \tag{1.67}$$

Our algorithm shall grow list l so that it always has property L. The end result is a permutation σ so that $w \in \Pi_\sigma$. Identify all indices i, j so that $w_{ij} = 0$ in the given matrix, and for each equivalence class choose a representative index. Initialize the list by choosing any three distinct indices, and use $\mathcal{R}$ to place other indices in l so that l has property L. To insert an index x insert in front of the first l_m so that $x \in L[l_m : l_{\text{size}(l)}]$. If there is such a minimal element, sub–claim 2 shows that $x \in L[l_M : l_{\text{size}(l)}]$ for all $l_M \geq l_m$. Use sub–claim 3 to show that $b \in L[m : l_{\text{size}(l)}]$ implies that $b \in L[l_x : l_{\text{size}(l)}]$. This shows that all triples are still satisfied. If there is no such m for which $x \in L[l_m : l_{\text{size}(l)}]$, use the relation on $x, l_{\text{size}(l)-1}, l_{\text{size}(l)}$ to determine where to place x among the last three. In either case List l has property L.

□

A further question one may ask is given a set of such relation what pairwise distance must be zero. We address this quesiton in the appendix section *Connection between number of orderings and degeneracy of containing space* .

Claim 1.2.9 $\langle \Pi \rangle = \mathcal{H}_0$

Proof 10 (Proof of claim 1.2.9) Using permutations we give a basis for which any matrix in $\mathcal{H}_0$ can be expressed. Let a be a $N \times N$ matrix contained in the class $\mathcal{H}_0$. We define $M_{[a]b}$ to be the largest number $(a + jb) < N$ for $j \in \mathbb{Z}$. Let us enumerate $n - 1$ elements of S_N:

$$\sigma_1 = 1, 2, 3, 4, \ldots, (n-1), n \tag{1.68}$$

$$\sigma_2 = 1, 3, 5, \ldots, (1+2j), \ldots, M_{[1]2}, \tag{1.69}$$

$$2, 4, 6, \ldots, (2+2j), \ldots, M_{[2]2} \tag{1.70}$$

$$\sigma_3 = 1, 4, 7, \ldots, (1+3j), \ldots, M_{[1]3}, \tag{1.71}$$

$$2, 5, 8, \ldots, (2+3j), \ldots, M_{[2]3}, \tag{1.72}$$

$$3, 6, 9, \ldots, (3+3j), \ldots, M_{[2]3} \tag{1.73}$$

$$\vdots \tag{1.74}$$

$$\sigma_k = 1, 4, 7, \ldots, (1+kj), \ldots, M_{[1]k}, \tag{1.75}$$

$$\cdots, \tag{1.76}$$

$$a, a+k, a+2k, \ldots (a+kj), \ldots, M_{[a]k}, \tag{1.77}$$

$$\cdots, \tag{1.78}$$

$$k, 2k, 3k, \ldots, (k+kj), \ldots, M_{[k]k} \tag{1.79}$$

$$\sigma_{n-1} = 1, M_{[1]n-1}, M_{[2]n-1}, M_{[3]n-1}, \ldots M_{[n-2]n-1} \tag{1.80}$$

$$\tag{1.81}$$

Let Π_k be the permutation space associated with σ_k. For the Π_k we let V_k be a deficient basis set specified by:

$$V_k = \{w_{\sigma_k(i),\sigma_k(i+1)} : \sigma_k(i+1) - \sigma_k(i) = k\} \tag{1.82}$$

Notice that V_k spans the upper kth diagonal of $\mathcal{H}_0$. □

Definition 1.2.2 *We call G a generating set of permutations for Υ if and only if $\mathcal{H}_0 = \oplus_{\sigma \in G} \Pi_\sigma$. A necessary condition that G is a generating sets is that the Minkowski sum spans a space of dimension $\binom{N}{2}$:*

Claim 1.2.10 $\langle \Upsilon \rangle = \mathcal{H}_0$

Proof 11 (Proof of claim 1.2.10) If $a \in \Pi$ then $a \in \Upsilon$ thus by 1.2.3 we have $\mathcal{H}_0 \subset \langle \Pi \rangle \subset \langle \Upsilon \rangle$. We also have $\Upsilon \subset \mathcal{H}_0$ hence $\langle \Upsilon \rangle \subset \mathcal{H}_0$ and we conclude that:

$$\langle \Upsilon \rangle = \mathcal{H}_0 \tag{1.83}$$

□

We summarize the main results of this subsection by a table:

Vector Subset of $L(N)$	Type	Degrees of Freedom
Non–Negative	Cone	N^2
Symmetric, Zero Diagonal	Subspace $\mathcal{H}_0 \leq L(N)$	$\binom{N}{2}$
Pairwise Distance Data	Cone	$\binom{N}{2}$
PDD of points on a line	Subspace $\Pi_\sigma \leq \mathcal{H}_0$	$N-1$

1.3 NP–Completeness of MATRIX–TO–LINE Problem

In this section we formulate an optimization problem known as the MATRIX–TO–LINE problem. The main result of this section will be to show that the MATRIX–TO–LINE problem is NP–Complete. We proceed by showing that a solution to

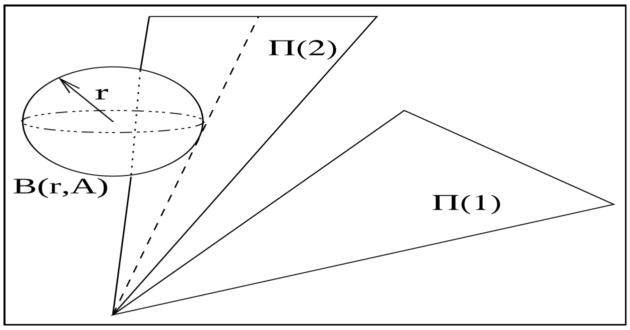

Figure 1.2: Optimization problem in the isomorphic space

the MATRIX–TO–LINE is verifiable in polynomial time, and then we show that MATRIX–TO–LINE reduces to GRAPH–EMBEDDING, which, in turn, is shown to be NP–Complete.

Problem 1 MATRIX–TO–LINE–OPTIMIZATION

Given a matrix $A \in \Upsilon$, and a semi–norm $||\cdot||$, find an element $\tilde{A} \in \Pi$ which minimizes the distance with respect to the norm.

$$\langle A, ||\cdot|| \rangle \rightarrow \tilde{A} : \tilde{A} = \ argmin_{x \in \Pi} ||A - x|| \tag{1.84}$$

Where $||A - x||$ denotes $||\Phi(A - x)||$.

Figure 1.2 indicates that the solution to the optimization problem in the case of a norm can be seen as the finding the first intersection of a ball centered at A intersected with Π.

1.3.1 MATRIX–TO–LINE is NP–Complete

We state the decision problem corresponding to the optimization problem.

Problem 2 MATRIX–TO–LINE–DECISION

Given input $\langle B \in \Upsilon_{\mathbb{F}}, b \in \mathbb{F}, ||\cdot||, A \in \Pi_{\mathbb{F}}\rangle$ *decide (Yes / No) if the given* A *is such that:*

$$||B - A|| < b \tag{1.85}$$

Problem 3 MATRIX–TO–LINE–BOUND

Given input $\langle B \in \Upsilon_{\mathbb{F}}, b \in \mathbb{F}, ||\cdot||\rangle$ *construct a matrix* $A \in \Pi_{\mathbb{F}}$*, or determine that one does not exist so that:*

$$||B - A|| < b \tag{1.86}$$

Clearly MATRIX–TO–LINE–BOUND is verifiable in polynomial time using straightforward matrix operations that can be computed in $\Theta(N^2)$ time.

Lemma 1.3.1 MATRIX–TO–LINE–BOUND *is equivalent to* GRAPH–EMBEDDING–BOUND

Proof 12 (Proof of lemma 1.3.1) Now we show that if we can solve instances of MATRIX–TO–LINE–BOUND we can solve instances of GRAPH–EMBEDDING–BOUND via a polynomial reduction.

Given the instance of GRAPH–EMBEDDING–BOUND, we construct B as:

$$B_{v,w} = \begin{cases} W(v,w) \text{ if } (v,w) \in E \\ 0 \text{ otherwise} \end{cases} \tag{1.87}$$

Let $\Lambda(\delta)$ be a matrix in $L(N^2)$ given by the equation:

$$\Lambda_{(v,w),(v',w')} = \begin{cases} 1 \text{ if } v = v', w = w', (v,w) \in E \\ \delta \text{ if } v = v', w = w', (v,w) \notin E \\ 0 \text{ otherwise} \end{cases} \tag{1.88}$$

And let $||x||_{\Lambda,p} = \left|\left|x\sqrt{\Lambda}\right|\right|_p$ be the norm in the MATRIX–TO–LINE instance. As δ goes to zero the limiting problem is a GRAPH–EMBEDDING rather than the complete graph embedding. In fact when $\delta = 0$ one can check that:

$$||x||_{\Lambda,p} = \left|\left|x\sqrt{\Lambda}\right|\right|_p \tag{1.89}$$

Is a semi–norm always equal to the p–norm of the GRAPH–EMBEDDING problem. Letting $\delta = 0$ we have:

$$\begin{aligned} \left|\left|B - \tilde{B}\right|\right|_{\Lambda,p}^p &= \sum_{e \in E} |(B - \tilde{B})_e|^p && (1.90) \\ &= \sum_{e \in E} |(W(e) - \tilde{B}_e)|^p && (1.91) \\ & && (1.92) \end{aligned}$$

Since $\tilde{B} \in \Pi_{\mathbb{F}}$ one can then find an embedding function f based on the values of $\tilde{B}_e$ for each $e \in E$. □

Lemma 1.3.2 GRAPH–EMBEDDING–BOUND *Problem is NP–Complete*

A proof very similar to the one by Jim Saxe [29] (by private communication), but expanding the cases and extending the result to P–Norms is given in the appendix.

Theorem 1.3.3 NP–COMPLETE

MATRIX–TO–LINE *is NP–Complete.*

Proof 13 (Proof of Theorem 1.3.3) By lemma (1.3.1) we note that GRAPH–EMBEDDING–BOUND reduces to MATRIX–TO–LINE, and by lemma (1.3.2) we conclude that MATRIX–TO–LINE is NP–Complete. □

This concludes the main result of the section.

1.4 Heuristics

To date it is not known if there is a polynomial time algorithm to solve MATRIX–TO–LINE problem in the general case: If the answer is yes then we can conclude that NP=P, if the answer is no then NP≠P. In either case, we would have solved the P vs. NP problem, one of the outstanding open problems in current mathematics.

Heuristics may be defined as algorithms which compute near optimal solutions, or locally optimal solutions. We are lead to ask, "can one design heuristics that are efficient?", if so are there conditions on the input which will ensure solutions are correct?"

We construct dynamical systems which define projections onto Π as the end results.

In the applications we focus on proving that local optima, however poor the guarantee may be, have computed perfect or correct results.

Definition 1.4.1 Heuristics

A Heuristic algorithm computes locally optimal solutions.

1.5 ODE Dynamical System Heuristic

We consider the construction of polynomial time algorithms which work in some cases for the MATRIX–TO–LINE problem. We then do an analysis of a 'basin of accuracy' for the heuristic by using the methods of Lyapunov. We begin by defining an evolution equation in the space of $\mathcal{H}_0$:

$$\gamma : [0, \infty) \to \mathcal{H}_0 \tag{1.93}$$

As an objective we seek to design the evolution of the matrix $\gamma(t)$ so that:

- The $\gamma(0)$ is arbitrary preassigned 'Initial Data' from the class Υ.
- The limit value is in Π.

$$\lim_{t \to \infty} \in \Pi \tag{1.94}$$

- The evolution is autonomous in time:

$$T_t : L(N) \to L(N) : \gamma(s+0) \to \gamma(s+t) \ \ \forall s \in [0, \infty] \tag{1.95}$$

For $\gamma(t)$ we define the underlying variables $\mathbf{X}(t)$ which determine γ and γ':

$$X_{ijk}(t) = (\gamma_{ij}, \gamma_{jk}, \gamma_{ik}, \theta_{jik}, \theta_{ikj}, \theta_{kji}, p_{ijk}) \quad i < j < k \in [1 \ldots N]$$

$$\begin{aligned}
\gamma_{ij} &= \text{ distance between points whose indices are } i, j \\
\gamma_{jk} &= \text{ distance between points whose indices are } j, k \\
\gamma_{ik} &= \text{ distance between points whose indices are } i, k \\
\theta_{jik} &= \text{ angle between line segments with endpoints indexed by } ik, ij \\
\theta_{ikj} &= \text{ angle between line segments with endpoints indexed by } ki, kj \\
\theta_{kji} &= \text{ angle between line segments with endpoints indexed by } jk, ji \\
p_{ijk} &= \text{ local gauge function}
\end{aligned}$$

The local gauge function is described by the following: Let,

$$\gamma_1 = \max\{\gamma_{ij}, \gamma_{jk}, \gamma_{ik}\}$$
$$\gamma_2 = \max\{\gamma_{ij}, \gamma_{jk}, \gamma_{ik}\} \setminus \{\gamma_1\}$$
$$\gamma_3 = \max\{\gamma_{ij}, \gamma_{jk}, \gamma_{ik}\} \setminus \{\gamma_1, \gamma_2\}$$
$$p_{ijk} = \{p : \gamma_1^p - (\gamma_2^p + \gamma_3^p) = 0\}$$

For the problem of embedding points in a line we will strongly encourage $p \to 1$, the only p–value to allow the relation $\gamma_1^p - (\gamma_2^p + \gamma_3^p) = 0$ while points are in a line.

Geometrically this vector is a set of variables defining model triangles, one for each of the $\binom{n}{3}$ distinct index triples. We note that the underlying variables X are an over–determined set of variables. We will use them anyway for easing the description of the ODE, but a smaller set of variables can be found. The dependencies amongst the variables in terms of free variables $\gamma_{ij} : i < j$:

$$\theta_{jik} = \arccos\left(\frac{\gamma_{ik}^2 + \gamma_{ij}^2 - \gamma_{jk}^2}{2\gamma_{ij}\gamma_{ik}}\right)$$
$$\theta_{ikj} = \arccos\left(\frac{\gamma_{ik}^2 + \gamma_{jk}^2 - \gamma_{ij}^2}{2\gamma_{jk}\gamma_{ik}}\right)$$
$$\theta_{kji} = \frac{\pi}{2} - (\theta_{jik} + \theta_{ikj})$$
$$p_{ijk} = \{p : \gamma_1^p - (\gamma_2^p + \gamma_3^p) = 0\}$$

Now we let η be a index array map:

$$\eta : i < j < k \to [1, \binom{N}{3}] \tag{1.96}$$

$$\mathbf{X}(t) = \langle X_\eta(t) \rangle \tag{1.97}$$

Since we want $\gamma(\infty) \in \Pi$, we design smooth bounded penalty functions favoring flow toward the set Π, and so that $x \in \Pi$ is a stable fixed point.

Penalty Functions

These penalty functions comprise two components:

- Angle displacement penalty

$$f(\phi, \theta; \nu) = \left(\frac{1}{3}\frac{4^2}{\pi^2}\right)^\nu \times (A + B + C)^\nu \tag{1.98}$$

$$A = \left(\left(\phi \bmod \frac{\pi}{2}\right) - \frac{\pi}{4}\right)^2 \tag{1.99}$$

$$B = \left(\left(\theta \bmod \frac{\pi}{2}\right) - \frac{\pi}{4}\right)^2 \tag{1.100}$$

$$C = \left(\left(\frac{\pi}{2} - \left((\phi + \theta) \mod \frac{\pi}{2}\right)\right) - \frac{\pi}{4}\right)^2 \tag{1.101}$$

$$\tag{1.102}$$

- Local gauge penalty function:

$$g(p; \tau) = \exp(-(1 - \max\{p, \frac{1}{p}\})/\tau) \tag{1.103}$$

These penalty functions are plotted in figure 1.3:

The angle displacement penalty function is plotted on a domain $0 : .05 : \pi$ for values of ν as .003, .1, 2, and 10.

Next the gauge penalty function is plotted for various values of τ:

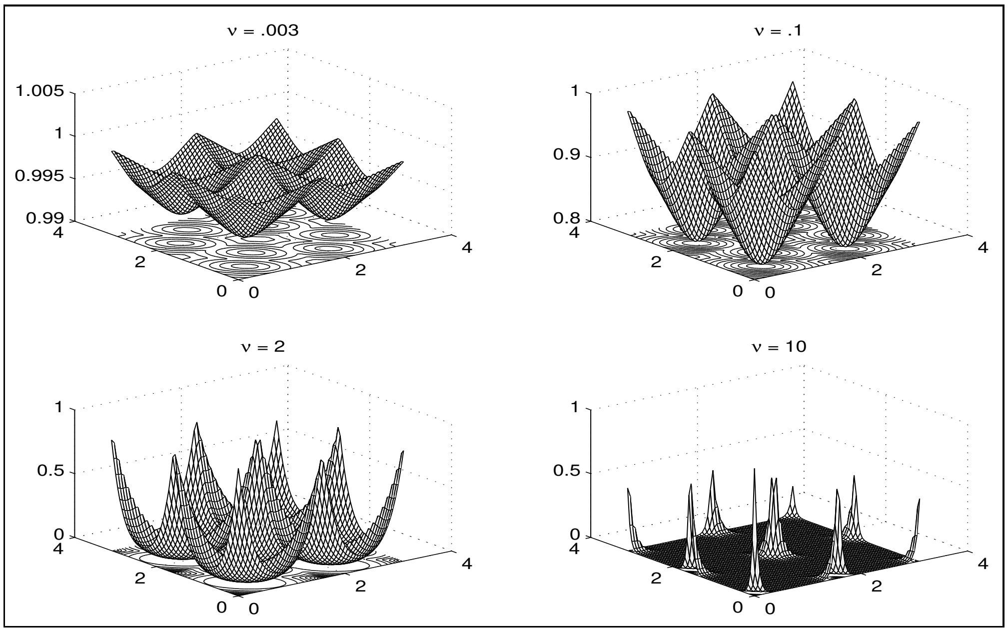

Figure 1.3: Angle penalty function family

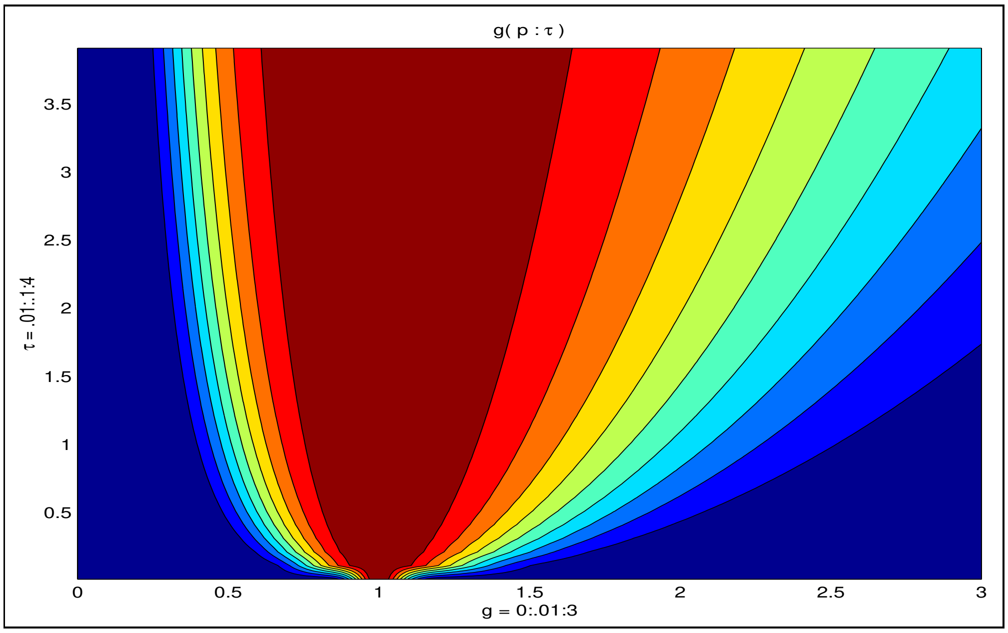

Figure 1.4: Gauge function family

Note that the function $f(\phi, \theta; \nu)$ is periodic, with periodic domain $[0, \frac{\pi}{2}]^2$, and favors values with one large angle and two tiny angles, as it happens for points on a line.

The function $g(p; \tau)$ on the other hand favors values of 1, which is the needed convexity of a norm to place points in a line so that they satisfy equality in a triangle inequality. The gauge penalty is not a Gaussian, as it places a slightly greater penalty on points with gauge less than 1, than on points with gauge greater than 1, this is to bias against index triplets that do not satisfy the triangle inequality. The p–function has a singularity when all edges are the the same length, this corresponds to the fact that $x^p = 2x^p$ if $p = \infty$, at this point the gauge penalty function is zero.

These functions together will impose a structure on the evolution of our data. Let:

$$\dot{\gamma}(t) = F(\gamma(t); \nu, \tau) \tag{1.104}$$

with:

$$F(\gamma_{ij}; \nu, \tau) = \gamma_{ij} - \frac{1}{Z_{ij}(\nu, \tau)} \sum_{k \in [1:N] \setminus \{i,j\}} f_{ikj} \times g_{ikj} \times (\alpha_{ikj}\gamma_{ik} + \beta_{ikj}\gamma_{kj}) \tag{1.105}$$

where:

$$f_{ikj} = f(\Theta_{ijk}, \Theta_{jki}; \nu) \tag{1.106}$$

$$g_{ikj} = g(p_{ijk}; \tau) \tag{1.107}$$

$$Z_{ij}(\nu, \tau) = \sum_{k \in [1:N] \setminus \{i,j\}} f(\Theta_{ijk}, \Theta_{jki}; \nu) g(p_{ijk}; \tau) \tag{1.108}$$

$$(\alpha_{ijk}, \beta_{ijk}) = \begin{cases} (\alpha = -1, \beta = 1) \text{ if } y_{ij} = y_{ik} = y_{jk} \\ (\alpha = -1, \beta = 1) \text{ if } y_{ij} < y_{ik} = y_{jk} \\ (\alpha = -1, \beta = 1) \text{ if } y_{ij} = y_{ik} < y_{jk} \\ (\alpha = -1, \beta = 1) \text{ if } y_{ij} < y_{ik} < y_{jk} \\ (\alpha = 1, \beta = -1) \text{ if } y_{ij} = y_{jk} < y_{ik} \\ (\alpha = 1, \beta = -1) \text{ if } y_{ij} < y_{jk} < y_{ik} \\ (\alpha = -1, \beta = 1) \text{ if } y_{ik} < y_{ij} = y_{jk} \\ (\alpha = -1, \beta = 1) \text{ if } y_{ik} < y_{ij} < y_{jk} \\ (\alpha = 1, \beta = 1) \text{ if } y_{ik} = y_{jk} < y_{ij} \\ (\alpha = 1, \beta = 1) \text{ if } y_{ik} < y_{jk} < y_{ij} \\ (\alpha = 1, \beta = -1) \text{ if } y_{jk} < y_{ij} = y_{ik} \\ (\alpha = 1, \beta = -1) \text{ if } y_{jk} < y_{ij} < y_{ik} \\ (\alpha = 1, \beta = 1) \text{ if } y_{jk} < y_{ik} < y_{ij} \end{cases} \tag{1.109}$$

The ordinary differential equation above is nonlinear as can be seen if we replace Θ variables with their dependencies in terms of γ variables.

After the needed conditions of F are verified we will treat the above ODE with a numerical analysis.

Claim 1.5.1 *$F(\gamma;\nu,\tau)$ is Lipschitz Continuous in a bounded set of $\mathcal{H}_0$ for fixed ν, τ*

Proof 14 (Proof of claim 1.5.1) For fixed parameters, τ,ν, f_{ijk} and g_{ijk} are clearly Lipschitz continious. Let us define:

$$T_{ijk} = f_{ijk}g_{ijk}t_{ijk} \tag{1.110}$$

$$t_{ijk} = (y_{ij} - \alpha_{ijk}y_{ik} + \beta_{ijk}y_{kj}) \tag{1.111}$$

The proof will consist of showing that either $t_{ijk}(y_{ij},y_{ik},y_{kj})$ or T_{ijk} is Lipschitz continious over a large number of cases given by the ordering of data y_{ij},y_{ik},y_{kj}. This is sufficient to show that F is Lipschitz continious, as F_{ij} is a normalized sum of Lipschitz continious functions.

We will show that t_{ijk} is Lipschitz continious in a large number of cases, then we will indicate how the other cases are identical, One case will use the smoothness properties of f and g. Underlying the technical details of the proof is that t_{ijk} is continious in y_{ij},y_{ik},y_{kj}. Variables α,β depend on the ordering of data y_{ij},y_{ik},y_{kj} and act as switches always producing a correction for value y_{ij}, in order that the points indexed by i, j, k fit on a line. The cases of the proof are enumerated below.

$$\begin{cases} (i) \;\; y_{ij} = y_{ik} = y_{kj} \;\; \alpha = -1, \beta = 1 \\ (ii) \;\; y_{ij} < y_{ik} = y_{kj} \;\; \alpha = -1, \beta = 1 \\ (iii) \;\; y_{ij} = y_{ik} < y_{kj} \;\; \alpha = -1, \beta = 1 \\ (iv) \;\; y_{ij} < y_{ik} < y_{kj} \;\; \alpha = -1, \beta = 1 \\ (v) \;\; y_{ij} = y_{kj} < y_{ik} \;\; \alpha = 1, \beta = -1 \\ (vi) \;\; y_{ij} < y_{kj} < y_{ik} \;\; \alpha = 1, \beta = -1 \\ (vii) \;\; y_{ik} < y_{ij} = y_{kj} \;\; \alpha = -1, \beta = 1 \\ (viii) \;\; y_{ik} < y_{ij} < y_{kj} \;\; \alpha = -1, \beta = 1 \\ (ix) \;\; y_{ik} = y_{kj} < y_{ij} \;\; \alpha = 1, \beta = 1 \\ (x) \;\; y_{ik} < y_{kj} < y_{ij} \;\; \alpha = 1, \beta = 1 \\ (xi) \;\; y_{kj} < y_{ij} = y_{ik} \;\; \alpha = 1, \beta = -1 \\ (xii) \;\; y_{kj} < y_{ij} < y_{ik} \;\; \alpha = 1, \beta = -1 \\ (xiii) \;\; y_{kj} < y_{ik} < y_{ij} \;\; \alpha = 1, \beta = 1 \end{cases} \tag{1.112}$$

We first consider cases $(iv), (vi), (viii), (x), (xii), (xiii)$:

$$t_{ijk}(y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}) - t_{ijk}(y_{ij}, y_{ik}, y_{kj}) \tag{1.113}$$

$$= y_{ij} + \epsilon_{ij} + (\alpha' (y_{ik} + \epsilon_{ik}) + \beta' (y_{kj} + \epsilon_{kj})) - y_{ij} + (\alpha y_{ik} + \beta y_{kj}) \tag{1.114}$$

$$= \epsilon_{ij} + \alpha' \epsilon_{ik} + \beta' \epsilon_{kj} + y_{ik} (\alpha' - \alpha) + y_{kj} (\beta' - \beta) \tag{1.115}$$

$$\tag{1.116}$$

here α', β' are the α, β variables for the possibly perturbed ordering $y_{ij} + \epsilon_{ij}, y_{ij} +$

$\epsilon_{ik}, y_{ij} + \epsilon_{kj}$. but for $||\epsilon|| < \min |y_{ij} - y_{ik}|, |y_{ik} - y_{kj}|, |y_{ij} - y_{kj}|$ we will have $\alpha' = \alpha$ and $\beta' = \beta$. Further we have that $|\alpha| \leq 1$ and $|\beta| \leq 1$ so we may conclude that:

$$t_{ijk}(y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}) - t_{ijk}(y_{ij}, y_{ik}, y_{kj}) = O(||\epsilon||) \tag{1.117}$$

And t_{ijk} has been shown to be Lipschitz continious in the cases (iv), (vi), $(viii)$, (x), (xii), $(xiii)$.

Now we consider cases $(iii), (v), (ix)$. We shall pick (iii) to proove and note that the other proofs use the same idea. In case (iii) we have $y_{ij} = y_{ik} < y_{kj}$. We can proove that the term t_{ijk} is Lipschitz continious.

$$t_{ijk}(y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}) - t_{ijk}(y_{ij}, y_{ik}, y_{kj}) \tag{1.118}$$

$$= y_{ij} + \epsilon_{ij} + (\alpha' (y_{ij} + \epsilon_{ik}) + \beta' (y_{kj} + \epsilon_{kj})) - y_{ij} + (\alpha y_{ij} + \beta y_{kj}) \tag{1.119}$$

$$\tag{1.120}$$

but in case (iii) we see that for ϵ small enough we will have: $y_{ij}+\epsilon_{ij}, y_{ik}+\epsilon_{ik}, y_{kj}+\epsilon_{kj}$ either fallin into one of the cases $(ii), (iii), (vii)$ all of which assign $\alpha' = -1, \beta' = 1$. we deduce that:

$$t_{ijk}(y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}) - t_{ijk}(y_{ij}, y_{ik}, y_{kj}) \tag{1.121}$$

$$= y_{ij} + \epsilon_{ij} + (-1 (y_{ij} + \epsilon_{ik}) + 1 (y_{kj} + \epsilon_{kj})) - y_{ij} + (-1 y_{ik} + 1 y_{kj}) \tag{1.122}$$

$$= O(||\epsilon||) \tag{1.123}$$

We conclude that these cases are safe, and that t_{ijk} is Lipschitz continious.

Now we consider cases $(ii), (vii), (xi)$, we shall pick (ii) to proove and note that the other proofs use the same idea. In case (ii) we have $y_{ij} < y_{ik} = y_{kj}$. We note that for $||\epsilon||$ small enough we will have $y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}$ in one of the cases

$(iv), (ii), (vi)$. If $y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}$ is in case (iv) or (ii) then the proof is similar to the one above but if $y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}$ is in case (vi) we will have, $(\alpha, \beta) = (-1, 1), (\alpha, \beta) = (1, -1)$:

$$t_{ijk}(y_{ij} + \epsilon_{ij}, y_{ik} + \epsilon_{ik}, y_{kj} + \epsilon_{kj}) - t_{ijk}(y_{ij}, y_{ik}, y_{kj}) \tag{1.124}$$

$$= y_{ij} + \epsilon_{ij} + (1\,(y_{ik} + \epsilon_{ik}) + -1\,(y_{ik} + \epsilon_{kj})) - y_{ij} + (-1y_{ik} + 1y_{ik}) \tag{1.125}$$

$$= O(||\epsilon||) \tag{1.126}$$

We conclude that these cases are safe in that t_{ijk} is Lipschitz continious.

We now consider our last case, case (i). We note that a pertubation of this case may become any of our 13 cases, so rather than go through these cases we shall prove that T_{ijk} is Lipschitz continious. The function g is zero when $y_{ij} = y_{ik} = y_{kj}$, In a neighborhood of the zero the function g decays exponential. This ends our proof. $\square$

Claim 1.5.2 *If* $\gamma(t_0) \in \Pi$ *then* $\gamma(t_0 + s) = \gamma(t_0)\ \forall s > 0$

Proof 15 (Proof of claim 1.5.2) If $\gamma(t_0) \in \Pi$ then $\gamma(t_0) \in \Pi_\sigma$ for some $\sigma \in S_N$ by lemma(1.2.8). We may conclude that:

$$\gamma_{ij} = \alpha_{ikj}\gamma_{ik} + \beta_{ikj}\gamma_{kj} \text{ for } i, j, k \tag{1.127}$$

Thus:

$$F(\gamma; \nu, \tau)_{ij} = \gamma_{ij} - \frac{1}{\sum_k f(\gamma)g(\gamma)} \sum_k f(\gamma)g(\gamma)\gamma_{ij} \tag{1.128}$$

$$= \gamma_{ij} - \gamma_{ij}\frac{1}{\sum_k f(\gamma)g(\gamma)} \sum_k f(\gamma)g(\gamma) \tag{1.129}$$

$$= 0 \tag{1.130}$$

□

Our next result indicates the stability of the set Π for our ODE. We shall consider a point $v \in \Pi_\sigma$ so that v is not in any other Π_λ which is not identical to Π_σ. For such a v there exists a $B_\delta(v)$ so that:

$$B_\delta(v) \cap \Pi_w = \varnothing \;\; \forall \;\; \Pi_w \neq \Pi_\sigma \tag{1.131}$$

Now let $Y = \mathcal{H}_0 \setminus \Pi_\sigma$, Y is the perpendicular subspace of Π_σ in $\mathcal{H}_0$. Let $U = B_\delta(v) \cap Y$ and $F_Y = F|_Y$ as a restriction map.

Claim 1.5.3 *For v, Y as above, there exists a δ so that $y \in U = G_\delta(v) \cap Y$, 0 is an Asymptotically Stable point, in the sense of Lyapunov. The point 0 corresponding to a point in Π_σ.*

Proof 16 (Proof of claim 1.5.3) We prove this claim using the method of Lyapunov, by developing a strict Lyapunov function for Y. We define the function

$$V : U \to \mathbb{R}^{\leq} : y \mapsto \frac{1}{2}||-\Gamma_\sigma \Phi\,(y)||_2^2 \tag{1.132}$$

This function V is the norm on the image vector $\Gamma_\sigma \Phi(y)$ where Γ_σ is defined in equation(1.34). We show that V is a strict Lyapunov function by showing three things.

- $V(0) = 0$
- $V(y) > 0 \;\; \forall y \in U \setminus \{0\}$
- $\mathrm{grad}V(y) \cdot F_Y < 0 \;\; \forall y \in U \setminus \{0\}$

$V(0) = 0$ is obvious and corresponds to a flow that is contained in $\Pi_\sigma \subset \Pi$. Also obvious is that $V(y) > 0$ if $y \neq 0$.

To show the third item in some $B_\delta(v)$ we can let: $d = \min_{k=2\dots n}\left\{v_{\sigma(k-1)\sigma(k)}\right\}$, and let $\delta' = \left(\frac{d}{4}\right)^p$. Now we have:

$$||y - v||_p < \delta' \Rightarrow ||y - v||_\infty < \frac{d}{4} \tag{1.133}$$

This is sufficient to conclude:

$$y_{\sigma(i)\sigma(k)} > y_{\sigma(i)\sigma(j)} \vee y_{\sigma(j)\sigma(k)} \Leftrightarrow v_{\sigma(i)\sigma(k)} > v_{\sigma(i)\sigma(j)} \vee v_{\sigma(j)\sigma(k)} \tag{1.134}$$

The gradient vector of V is computed as:

$$(\ \mathrm{grad} V(y))_{\sigma(i)\sigma(j)} = \sum_{\sigma(k)<\sigma(i)<\sigma(j)} - \left(-y_{\sigma(i)\sigma(k)} + y_{\sigma(k)\sigma(i)} + y_{\sigma(i)\sigma(j)}\right) \tag{1.135}$$

$$+ \sum_{\sigma(i)<\sigma(k)<\sigma(j)} - \left(y_{\sigma(i)\sigma(j)} - y_{\sigma(i)\sigma(k)} - y_{\sigma(k)\sigma(j)}\right) \tag{1.136}$$

$$+ \sum_{\sigma(i)<\sigma(k)<\sigma(j)} - \left(-y_{\sigma(i)\sigma(k)} + y_{\sigma(i)\sigma(j)} + y_{\sigma(j)\sigma(k)}\right) \tag{1.137}$$

$$\tag{1.138}$$

The field vector:

$$(F(y))_{\sigma(i)\sigma(j)} = \sum_{\sigma(k)<\sigma(i)<\sigma(j)} \frac{f_{\sigma(i)\sigma(j)\sigma(k)} g_{\sigma(i)\sigma(j)\sigma(k)}}{Z_{\sigma(i)\sigma(j)}} \left(-y_{\sigma(i)\sigma(k)} + y_{\sigma(k)\sigma(i)} + y_{\sigma(i)\sigma(j)}\right) \tag{1.139}$$

$$+ \sum_{\sigma(i)<\sigma(k)<\sigma(j)} \frac{f_{\sigma(i)\sigma(j)\sigma(k)} g_{\sigma(i)\sigma(j)\sigma(k)}}{Z_{\sigma(i)\sigma(j)}} \left(y_{\sigma(i)\sigma(j)} - y_{\sigma(i)\sigma(k)} - y_{\sigma(k)\sigma(j)}\right) \tag{1.140}$$

$$+ \sum_{\sigma(i)<\sigma(k)<\sigma(j)} \frac{f_{\sigma(i)\sigma(j)\sigma(k)} g_{\sigma(i)\sigma(j)\sigma(k)}}{Z_{\sigma(i)\sigma(j)}} \left(-y_{\sigma(i)\sigma(k)} + y_{\sigma(i)\sigma(j)} + y_{\sigma(j)\sigma(k)}\right) \tag{1.141}$$

Letting:

$$t_{ijk} = \begin{cases} \left(-y_{\sigma(i)\sigma(k)} + y_{\sigma(k)\sigma(i)} + y_{\sigma(i)\sigma(j)}\right) \text{ if } \sigma(k) < \sigma(i) < \sigma(j) \\ \left(y_{\sigma(i)\sigma(j)} - y_{\sigma(i)\sigma(k)} - y_{\sigma(k)\sigma(j)}\right) \text{ if } \sigma(i) < \sigma(k) < \sigma(j) \\ \left(-y_{\sigma(i)\sigma(k)} + y_{\sigma(i)\sigma(j)} + y_{\sigma(j)\sigma(k)}\right) \text{ if } \sigma(i) < \sigma(k) < \sigma(j) \end{cases} \tag{1.142}$$

and $h_{ijk} = \frac{f_{\sigma(i)\sigma(j)\sigma(k)} g_{\sigma(i)\sigma(j)\sigma(k)}}{Z_{\sigma(i)\sigma(j)}}$ we get:

$$(\ \mathrm{grad} V(y))_{\sigma(i)<\sigma(j)} \times (F(y))_{\sigma(i)<\sigma(j)} = \left(-\sum_k t_{ijk}\right)\left(\sum_k h_{ijk} t_{ijk}\right) \tag{1.143}$$

Now to finish up the proof we will show that in a small enough ball of v the terms h_{ijk} consentrate to values $\frac{1}{N}$.

We note that each $h_{\sigma(i)\sigma(j)\sigma(k)}$ is many times differentiable on a compact set and hence is Lipschitz continious. This implies that there is a K so that:

$$|h_{ijk}(y_1) - h_{ijk}(v)| \leq K|y_1 - v| \tag{1.144}$$

Noting that $h_{ijk}(v) = \frac{1}{N}$, and letting:

$$\epsilon_{ijk} = h_{ijk} - \frac{1}{N} \tag{1.145}$$

We proceed:

$$
\begin{aligned}
&(\ \mathrm{grad}V(y))_{\sigma(i)<\sigma(j)} \times (F(y))_{\sigma(i)<\sigma(j)} && (1.146)\\
&= -\left(\sum_k t_{ijk}\right)\left(\sum_k h_{ijk}t_{ijk}\right) && (1.147)\\
&= -\left(\sum_k t_{ijk}\right)\left(\sum_k \left(\frac{1}{N}+\epsilon_{ijk}\right)t_{ijk}\right) && (1.148)\\
&= \frac{-1}{N}\left(\sum_k t_{ijk}\right)^2 - \left(\langle t_{ijk},(1,1,\ldots 1)\rangle\right)\left(\langle \epsilon_{ij},t_{ij}\rangle\right) && (1.149)\\
&< \frac{-1}{N}||t_{ijk}||_2^2 + ||t_{ijk}||_2^2\sqrt{N}||\epsilon_{ij}||_2 && (1.150)\\
&= ||t_{ijk}||_2^2\left(-\frac{1}{N}+\sqrt{N}||\epsilon_{ij}||_2\right) && (1.151)
\end{aligned}
$$

But when $||y_1 - v|| < \frac{1}{N^{\frac{3}{2}}K}$ we will have:

$$||\epsilon_{ij}|| < K||y_1 - v|| \leq \frac{1}{N^{\frac{3}{2}}} \tag{1.152}$$

and the term $\left(-\frac{1}{N}+\sqrt{N}||\epsilon_{ij}||_2\right) < 0$. Letting $\delta = \min\left(\delta', \frac{1}{N^{\frac{3}{2}}K}\right)$ completes the proof. □

The above claim tells us that within a neighborhood of a point v (in only one subspace Π_σ) given by $B_\delta(v)$, the analytic result converges to a projection onto one of the permutation spaces.

Next, we will prove that a numerical solution converges to the analytic solution, the stage is set as we have shown that our field function F is C'.

Claim 1.5.4 *A numerical method such as Euler's method, backward Euler, or Ruge-Kutta converges*

Proof 17 (Proof of claim 1.5.4) We give a proof for the Euler's method, and note that the other methods are well known to have better convergence properties than that of Euler's or Forward Eulers method. let h be some small number.

$$t_n = t_0 + nh \tag{1.153}$$

And our approximate solution will given by the formula, also know as the forward Euler method:

$$\tilde{\gamma}_n = \tilde{\gamma}_{n-1} + hF(\tilde{\gamma}_{n-1}) \tag{1.154}$$

This as a approximation to the actual position of the flow:

$$\gamma_n = \gamma(t_n) \tag{1.155}$$

By Taylors series expansion we will have:

$$\gamma_n = \gamma\left(t_0 + nh\right) = \gamma\left(t_0 + (n-1)\,h\right) + h\dot{\gamma}\left(t_0 + (n-1)\,h\right) + O(h^2) \tag{1.156}$$

$$= \gamma_{n-1} + h\dot{\gamma}\left(t_0 + (n-1)\,h\right) + O(h^2) \tag{1.157}$$

Now let the error at time point n be given by:

$$\epsilon_n = ||\tilde{\gamma}_n - \gamma_n|| \tag{1.158}$$

and notice that $\epsilon_0 = 0$. We derive the bound:

$$
\begin{aligned}
|\epsilon_n| &= |||\tilde{\gamma}_n - \gamma_n||| && (1.159)\\
&= \left|\left|(\tilde{\gamma}_{n-1} + hF(\tilde{\gamma}_{n-1})) - \left(\gamma_{n-1} + h\dot{\gamma}\left(t_0 + (n-1)\,h\right) + O(h^2)\right)\right|\right| && (1.160)\\
&\leq ||\tilde{\gamma}_{n-1} - \gamma_{n-1}|| + ||hF(\tilde{\gamma}_{n-1}) - h\dot{\gamma}\left(t_0 + (n-1)\,h\right)|| + O(h^2) && (1.161)\\
&\leq \epsilon_{n-1} + h||F(\tilde{\gamma}_{n-1}) - F\left(\gamma\left(t_0 + (n-1)\right)\right)|| + O(h^2) && (1.162)\\
&\leq \epsilon_{n-1} + hK||\gamma_{n-1} - \gamma\left(t_0 + (n-1)\right)|| + O(h^2) && (1.163)\\
&\leq \epsilon_{n-1}\left(1 + hK\right) + O(h^2) && (1.164)\\
& && (1.165)
\end{aligned}
$$

We get equation (1.161) by the triangle inequality, we get equation (1.163) by the Lipschitz continuity of the function F. We conclude:

$$|\epsilon_n| \leq (1 + hK)^n \, O(h^2) \qquad (1.166)$$

For $n = \lfloor \frac{1}{2h} \rfloor + 1$ we can get an arbitrary precision accurate evaluation by letting $h \to 0$. This shows us that we can use the forward Euler step method to approximate $\gamma(t_n + \frac{1}{2})$ to arbitrary precision, from an accurate $\gamma(t_n)$ evaluation. We conclude that the numerical method Converges in the interval $[t_n, t_n + \frac{1}{2}]$ and we can use this iteratively to get Convergence for all $[0, \infty)$. $\square$

1.5.1 Numerical Solutions and Complexity

Numerical solutions are in general quick algorithms. We note the only problem we face is that we have blown a problem of $\binom{N}{2}$ up into about $\binom{N}{3}$ space vectors, however we are still within the grasp of tractable work, maybe $N = 100$.

We illustrate some examples and illustrate the effect of error.

1.5.2 Conclusions on Heuristics.

We note that the heuristic is able to tune itself to the region of data that most satisfies our assumptions that points are embedable into $\mathbb{R}$, namely that angles $(0, \pi, 0)$ can be found to match data.

What we find is somewhat interesting in terms of distributions of error matrices. Suppose that we have input A:

$$A = \pi + E \quad \pi \in \Pi \tag{1.167}$$

Given a distribution of on E we believe that the resulting subspaces are fairly robust under the variation in E. The algorithm is essentially choosing a basis for Π as it goes. It pursues a basis for its embedding by pruning back a large number of basis candidates as it moves. In some cases the solution is determined from a high signal to noise ratio in a critical band of matrix entries. Ultimately we believe that the determination of basis is somewhat related to the flow of heat in a conductive rod, and occurs very quickly. In the applications the algorithms presented are speedups of the central idea presented in the ODE.

Chapter 2

Application I, Probe Mapping

2.1 Mathematical Definitions

A probe is an isolated string of genomic material that may hybridize with genomic material (called the text string) if there is substantial similarity between the probe and any substring of the text string. The probe will stick to genomic material the text string contains a similar sub-string to the probe's complementary sequence.

Given a set of P probes listed as $\{p_1, p_2, \ldots, p_P\}$ and contained in some contiguous segment of the genome we define a *probe map* to be a pair of sequences, **ordering** $= \{p_{\pi(1)}, p_{\pi(2)}, \ldots, p_{\pi(P)}\}$ and **position** $= \{x_1,\ x_2,\ \ldots,\ x_P\}$. The ordering sequence is a description of an ordering for the P probes, while the position sequence is a description of the probe positions.

Assume that the underlying correct position of each probe remains unknown. We infer probe maps approximating the correct positions as best as possible from an experimental set of data which is stochastic.

We model various experimental errors arising from the hybridization experiment used to measure probe to probe distance. With the model we can understand the distribution of pairwise distance graphs as a random variable. Under certain parameters we can implement Bayes formula to build a Maximum Likelihood Estimator (MLE) for probe map reconstruction.

Formulating the Problem

Consider a genome represented by the interval $[0, G]$. Take P random short substrings, about 200bps (base pairs), which appear on the genome uniquely. Represent these strings as points $\{x_1, \ldots, x_P\}$ in the vector space $\mathbb{R}$, and indicating the genomic position of each string. Assume that the probe positions are i.i.d. with uniform random distribution over the interval $[0, G]$. Let S be a collection of intervals of the genome, each of length L (usually ranging from few 100kbs to Mbs). Suppose that the left-hand points of the intervals of S are i.i.d. uniform random variables over the interval $[0, G]$. Take a small, even in number sized subset of intervals $S' \subset S$, chosen randomly from S. Divide S' randomly into two equal-size disjoint subsets $S' = S'_R \cup S'_G$, where R indicates a red color class and G indicates a green color class. Now specify any point x in $[0, G]$ and consider the possible relationships between x, and the intervals in S':

- x is not covered by any interval in S'.
- x is covered by at least one interval of S'_R but no intervals of S'_G.
- x is covered by at least one interval of S'_G but no intervals of S'_R.
- x is covered by at least one interval of S'_R and at least one interval of S'_G.

If we perform a sequence of M such experiments then for each x we get a sequence of M outcomes represented as a color string of length M. We are interested in observing color string outcomes at each position $\{x_1, \ldots, x_P\}$.

For DNA the short sub-strings can be produced with the use of restriction enzymes, or synthesized as oligoes. The collection of covering intervals may be provided by a bacterial artificial chromosomes library (BACs) or yeast artificial chromosomes library (YACs). The division of a random sample taken from the clone library may be done with phosphorescent molecules added to the DNA and visible with a laser scanner. Hybridization microarrays allow us to observe such an outcome sequence for each of the 100,000 probes in a constant amount of time.

Consider an example with human. To make a set of Human Oligoe Probes we may use restriction enzymes to cut out P probe substrings of size 200bp to 1200bp from the genome and choose a low complexity representation (LCR) . We may arrange for a sequence of M random samples from the BAC library, suppose each sample has K BACs and coverage $c = \frac{KL}{G}$. Samples are then randomly partitioned into two color classes $\Sigma = \{R, G\}$, and then hybridized to a microarray, arrayed with P probes. If we pick one probe p_i, then the possible outcomes for one experiment are:

- p_i hybridizes to zero BACs. We say the outcome is 'B' (blank).
- p_i hybridizes to at least one red BAC and zero green BACs. We say the outcome is 'R' (red).
- p_i hybridizes to at least one green BAC and zero red BACs. We say the outcome is 'G' (green).

- p_i hybridizes to at least one green BAC and at least one red BAC. We say the outcome is 'Y' (yellow).

We call these events i_B, i_R, i_G, and i_Y respectively. We use M random samples to complete the full experiment. The parameter domain for the full experiment is $\langle P, L, K, M\rangle$, where P is the number of probes, L is the average length of the genomic material used (for BACs, $L = 160$kb), K is the sampling size, and M is the number of samples. The output is a color sequence for each probe. The sequence corresponding to probe p_j is $\mathbf{s}_j = \langle s_{j,k}\rangle_{k=1}^{M}$ with $s_{j,k} \in \{B, R, G, Y\}$.

How the distances are measured

With the resulting color sequences $\mathbf{s}_j$ we can compute the pairwise Hamming distance. Let

$$H_{i,j} = \#\text{ places where } \mathbf{s}_i \text{ and } \mathbf{s}_j \text{ differ ,} \tag{2.1}$$

$$C_{i,j} = \#\text{ places where } \mathbf{s}_i \text{ and } \mathbf{s}_j \text{ are the same but } \mathbf{s}_i \neq B, \tag{2.2}$$

$$T_{i,j} = \#\text{ places where } \mathbf{s}_i \text{ and } \mathbf{s}_j \text{ are } B. \tag{2.3}$$

$$\tag{2.4}$$

The Hamming distance defines a distance metric on the set of probes.

Lemma 2.1.1 *Consider an experiment with parameters $\langle P, L, K, M\rangle$, and $c = \frac{KL}{G}$. Let i and j be arbitrary indices from the clone set and x_{ij} is the actual distance (in number of bases) separating probe p_i from probe p_j on the genome. Let $\hat{x}_{ij} =$*

$\min\{x_{ij}, L\}$. *Then:*

$$H_{i,j} \sim Bin\ (M, \frac{2ce^{(\frac{-c}{2})}\hat{x}_{ij}}{L} + O((\hat{x}_{ij})^2)) \tag{2.5}$$
$$C_{i,j} \sim Bin\ (M, 1 - e^{-c} + \frac{c}{2}(e^{-c} - 2e^{-\frac{c}{2}})\hat{x}_{ij} + O((\hat{x}_{ij})^2)) \tag{2.6}$$
$$T_{i,j} \sim Bin\ (M, (e^{-c(1+\frac{\hat{x}_{ij}}{L})})) \tag{2.7}$$
$$\tag{2.8}$$

Where $V \sim Bin\ (M, r)$ *indicates that* V *is a binomial random variable with parameters* (M, r).

Proof 18 (Proof of lemma 2.1.1) Fix i, j and let i_Y be the event that probe i is observed to have a Y outcome; similar events are defined for the symbols $i_B, i_R, i_G, i_Y, j_B, j_R, j_G, j_Y$. Let C be the event that i and j are observed to have the same non–blank color, let T be the event that i and j have the same blank outcome, and let H be the event that probe i or j are not the same color.

Since the M samples are done independently the proof reduces to showing that when $M = 1$ the probabilities are Bernoulli with respective parameters. Let us define events $T = (i_B \wedge j_B)$, $C = ((i_R \wedge j_R) \vee (i_G \wedge j_G) \vee (i_Y \wedge j_Y))$, and $H = (\neg T \wedge \neg C)$. Given a set of K BACs on a genome $[0, G]$ the probability that none start in an interval of length l is $(1 - \alpha)^l \approx e^{-\alpha l}$ where $\alpha = \frac{K}{G}$.

Shown below in figure 2.1 is a diagram that is helpful in computing the probabilities for events C, H, T when $x < L$. The heavy dark bar labeled a represents a set of BACs which covers probe p_i but not p_j; the bar labeled b represents a set of BACs that covers probe p_i and p_j; finally, the bar labeled c represents a set of BACs that covers p_j but not p_i. Hence we derive:

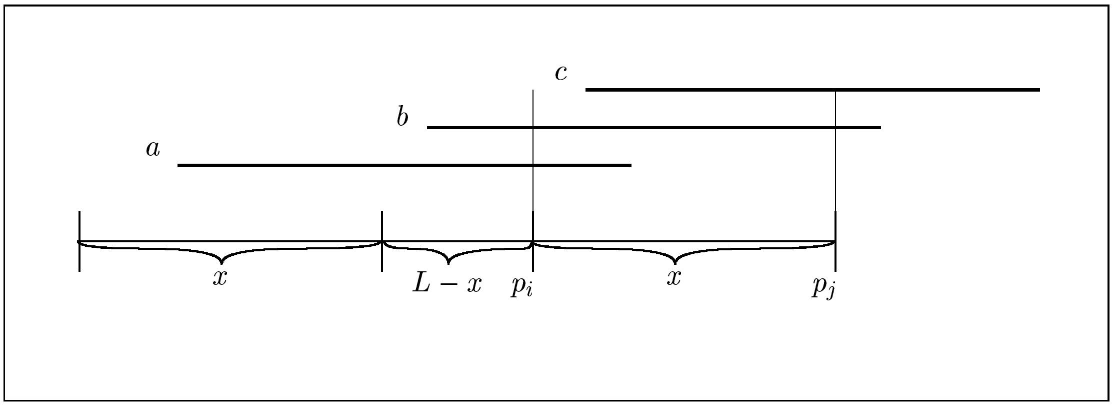

Figure 2.1: Probabilities dependent on distance x

$$
\begin{aligned}
\Pr(T|x \leq L) &= \exp(-(\alpha_R + \alpha_G)(L+x)) && (2.9)\\
\Pr(i_R \wedge j_R|x < L) &= e^{-\alpha_G(L+x)}\{(1 - e^{-\alpha_R(L-x)}) && (2.10)\\
&\quad +(1 - e^{-\alpha_R x})(e^{-\alpha_R(L-x)})(1 - e^{-\alpha_R x})\} && (2.11)\\
&= e^{-\alpha_G(L+x)}\{1 - 2e^{-\alpha_R L} + e^{-\alpha_R(L+x)}\} && (2.12)\\
\Pr(i_G \wedge j_G|x \leq L) &= e^{-\alpha_R(L+x)}\{1 - 2e^{-\alpha_G L} + e^{-\alpha_G(L+x)}\} && (2.13)\\
\Pr(i_Y \wedge j_Y|x \leq L) &= \left(1 - 2e^{-\alpha_R L} + e^{-\alpha_R(L+x)}\right) \times && (2.14)\\
&\quad \left(1 - 2e^{-\alpha_G L} + e^{-\alpha_G(L+x)}\right) && (2.15)\\
\Pr(C|x \leq L) &= P(i_R \wedge j_R|x \leq L) + P(i_G \wedge j_G|x \leq L) && (2.16)\\
&\quad +P(i_Y \wedge j_Y|x \leq L) && (2.17)\\
\Pr(H|x \leq L) &= 1 - [P(T|x \leq L) + P(C|x \leq L)] && (2.18)\\
& && (2.19)
\end{aligned}
$$

When $x \geq L$ the probabilities are:

$$\Pr(T|x \geq L) = \exp(-(\alpha_R + \alpha_G)(2L)) \tag{2.20}$$

$$\Pr(i_R \wedge j_R|x \geq L) = e^{-\alpha_G(2L)}\{(1 - e^{-\alpha_R L})^2\} \tag{2.21}$$

$$\Pr(i_G \wedge j_G|x \geq L) = e^{-\alpha_R(2L)}\{(1 - e^{-\alpha_G L})^2\} \tag{2.22}$$

$$\Pr(i_Y \wedge j_Y|x \geq L) = (1 - e^{-\alpha_R L})^2(1 - e^{-\alpha_G L})^2 \tag{2.23}$$

$$\Pr(C|x \geq L) = P(i_R \wedge j_R|x \geq L) + P(i_G \wedge j_G|x \geq L) \tag{2.24}$$

$$+P(i_Y \wedge j_Y|x \geq L) \tag{2.25}$$

$$\Pr(H|x \geq L) = 1 - [P(T|x \geq L) + P(C|x \geq L)] \tag{2.26}$$

$$\tag{2.27}$$

Because $\alpha_R = \alpha_G$, $\alpha_R L = \alpha_G L = \frac{c}{2} = \frac{KL}{2G}$. Let $q = q(x) = P(H)$ and $p = p(x) = P(C)$. In general $q(x)$ and $p(x)$ are complicated functions of x, below we derive a first order approximation of $x(q)$ to be used as a biased estimator.

$$\Pr(H) = (1 - (1 - 2e^{\frac{-c}{2}} + 2e^{\frac{-c}{2}(1+\frac{x}{L})})^2) = \frac{2c\exp(\frac{-c}{2})x}{L} + O(x^2) \tag{2.28}$$

$$\Pr(T) = \left(e^{-c(1+\frac{x}{L})}\right) \tag{2.29}$$

$$\Pr(C) = 1 - e^{-c} + \frac{c}{2}(e^{-c} - 2e^{-\frac{c}{2}})x + O(x^2) \tag{2.30}$$

With independent sampling:

$$\Pr(H_{i,j}) \sim \text{Bin}\ (M, \frac{2c\exp(\frac{-c}{2})x}{L} + O(x^2)) \tag{2.31}$$

$$\Pr(C_{i,j}) \sim \text{Bin}\ (M, 1 - e^{-c} + \frac{c}{2}(e^{-c} - 2e^{-\frac{c}{2}})x + O(x^2)) \tag{2.32}$$

$$\Pr(T_{i,j}) \sim \text{Bin}\ (M, (e^{-c(1+\frac{x}{L})})) \qquad \square \tag{2.33}$$

$$\tag{2.34}$$

□

These computations for small x lead to an accurate estimator:

Corollary 2.1.2 *The estimator of x_{ij} is given by $\tilde{x}_{ij} = H_{i,j}\frac{e^{\frac{c}{2}}L}{2cM}$ is good in the sense that there are values of c so that:*

$$f(\tilde{x}_{ij} = d|x_{ij}) \to \begin{cases} \frac{1}{\sqrt{2\pi}\sigma\sqrt{x_{ij}}} e^{-\frac{(d-x_{ij})^2}{2\sigma^2 x_{ij}}} & \text{if } x_{ij} < L; \\ \frac{1}{\sqrt{2\pi}\sigma\sqrt{L}} e^{-\frac{(d-x_{ij})^2}{2\sigma^2 L}} & \text{if } x_{ij} \geq L; \end{cases} \quad \text{as } M \to \infty. \tag{2.35}$$

with $\sigma^2 = \left(\frac{e^{\frac{c}{2}}}{2c}\right)$.

Proof 19 (Proof of corollary 2.1.2) It is based on a standard approximation. □

Lemma 2.1.3 *The distribution for distance d is a function of x and is approximated by*

$$f(d|x) = \mathbb{I}_{0 \leq x < L} \frac{e^{-(d-x)^2/2\sigma^2 x}}{\sqrt{2\pi x}\sigma} + \mathbb{I}_{L \leq x \leq G} \frac{e^{-(d-L)^2/2\sigma^2 L}}{\sqrt{2\pi L}\sigma}.$$

Proof 20 (Proof of lemma 2.1.3) Simple restatement of corollary 2.2 □

Since we have assumed that any given probe is distributed uniformly randomly over the genome, the density function for the probe's position is:

$$f(x) = \frac{1}{G} \tag{2.36}$$

Our next lemma is an application of Bayes' formula to compute $f(x|d)$ from $f(x)$ and $f(d|x)$ computed above.

Lemma 2.1.4 *If $f(d|x) = \mathbb{I}_{0 \leq x < L} \frac{e^{-(d-x)^2/2\sigma^2 x}}{\sqrt{2\pi x}\sigma} + \mathbb{I}_{L \leq x \leq G} \frac{e^{-(d-L)^2/2\sigma^2 L}}{\sqrt{2\pi L}\sigma}$. Then*

$$f(x|d) \approx \mathbb{I}_{d<L} \frac{e^{-(x-d)^2/2\sigma^2 d}}{\sqrt{2\pi d}\sigma} + \mathbb{I}_{d \geq L}\, \mathbb{I}_{L \leq x \leq G}\, \frac{1}{G-L}. \tag{2.37}$$

Proof 21 (Proof of lemma 2.1.4)

$$\begin{aligned} f(x|d) &= \frac{f(d|x)f(x)}{\int_0^G f(d|x)f(x)\ dx} && (2.38)\\ &= \frac{\frac{1}{G}\left(\mathbb{I}_{0\le x<L}\frac{e^{-(d-x)^2/2\sigma^2 x}}{\sqrt{2\pi x}\sigma} + \mathbb{I}_{L\le x\le G}\frac{e^{-(d-L)^2/2\sigma^2 L}}{\sqrt{2\pi L}\sigma}\right)}{\frac{1}{G}\int_0^G\left(\mathbb{I}_{0\le x<L}\frac{e^{-(d-x)^2/2\sigma^2 x}}{\sqrt{2\pi x}\sigma} + \mathbb{I}_{L\le x\le G}\frac{e^{-(d-L)^2/2\sigma^2 L}}{\sqrt{2\pi L}\sigma}\right)\ dx} && (2.39)\end{aligned}$$

For small values of σ^2 the denominator in the above expression can be approximated as follows:

$$\begin{aligned} f(d) &= \frac{1}{G}\int_0^L \frac{e^{-(d-x)^2/2\sigma^2 x}}{\sqrt{2\pi x}\sigma}\ dx + \frac{G-L}{G}\frac{e^{-(d-L)^2/2\sigma^2 L}}{\sqrt{2\pi L}\sigma} && (2.40)\\ &\approx \frac{1}{G}\mathbb{I}_{d<L} + \left(1-\frac{L}{G}\right)\delta_{d=L}. && (2.41)\end{aligned}$$

Thus, we make further simplifying assumptions and choose the following likelihood function:

$$f(x|d) \approx \mathbb{I}_{d<L}\frac{e^{-(x-d)^2/2\sigma^2 d}}{\sqrt{2\pi d}\sigma} + \mathbb{I}_{d\ge L}\ \mathbb{I}_{L\le x\le G}\ \frac{1}{G-L}, \qquad \Box \qquad (2.42)$$

$\Box$

With conditional $f(x|d)$ we can now define the Maximum Likelihood Estimation problem:

Given an arbitrary pair-wise distance edge weighted complete graph $\mathcal{G}$ of P vertices, representing probes, and each edge (i,j) labeled with $d_{i,j}$, a sampled value of a random variable with the distribution $f(d||x_i - x_j|)$, we would like to choose an embedding of $\mathcal{G}$ (or more precisely, an embedding of the vertices of $\mathcal{G}$) into the real line:

$$\{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_P\} \subset [0, G], \qquad (2.43)$$

that maximizes a likelihood function $F(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_P | d_{ij} : i, j \in [1, P])$. By ignoring the weak dependencies, we approximate F as:

$$\prod_{1 \leq i,j \leq P} f(|\tilde{x}_i - \tilde{x}_j| | d_{ij}). \tag{2.44}$$

Hence, we can minimize a related cost function

$$\sum_{1 \leq i,j \leq P} - \ln f(|\tilde{x}_i - \tilde{x}_j| | d_{ij}). \tag{2.45}$$

Lemma 2.1.5 *The Optimization problem of finding $\tilde{x}_j$ to minimize $f(\tilde{x}_j | \{\tilde{x}_i : i < j\}, \{d_{i,j} : i < j\})$ is approximated by solving the following optimization problem:*

$$\textit{minimize} \sum_{1 \leq i < j \leq P} W_{ij}(|\tilde{x}_i - \tilde{x}_j| - d_{ij})^2, \tag{2.46}$$

where W_{ij}'s are positive real valued weight functions:

$$W_{ij} = \begin{cases} \dfrac{1}{2\sigma^2 d_{ij}} & \textit{if } d_{ij} < L; \\ \epsilon & \textit{otherwise,} \end{cases} \tag{2.47}$$

and $\epsilon = O\left(\frac{1}{(G-L)^2}\right)$.

Proof 22 (Proof of lemma 2.1.5)

$$- \ln f(x|d) \approx \begin{cases} \dfrac{(x-d)^2}{2\sigma^2 d} + \ln(\sqrt{2\pi d}\sigma) & \text{if } d < L; \\ \ln(G-L) - \ln \mathbb{I}_{L \leq x \leq G} & \text{otherwise.} \end{cases} \tag{2.48}$$

Hence

$$\sum_{1 \leq i,j \leq P} - \ln f(|\tilde{x}_i - \tilde{x}_j| | d_{ij}) = \sum_{1 \leq i < j \leq P} W_{ij}(|\tilde{x}_i - \tilde{x}_j| - d_{ij})^2. \tag{2.49}$$

Note that $\epsilon = \frac{1}{2\sigma_M^2 d_{ij}} \leq \frac{1}{2\sigma_M^2 L} \leq \frac{1}{2(G-L)^2 L}$ as σ_M being the maximum variance is bounded by $(G - L)$. $\square$

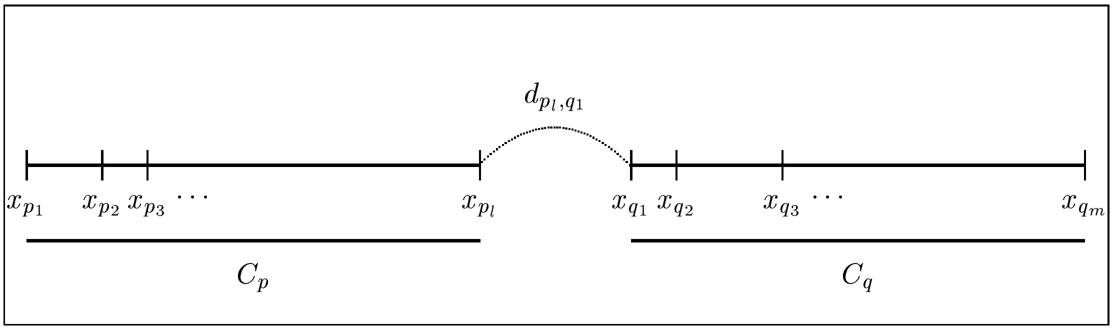

Figure 2.2: Contig operation

Simple Algorithm

In the following description of an algorithm, Contigs are sets of probes, that are believed to be in a contiguous region of the genome. Contigs are local regions where the map may be understood. The algorithms shall grow the contigs monotonically obtaining the largest possible set of probes that can be mapped, mapped means understanding probe positions in terms of order and placement.

The simplest algorithm to place probes proceeds as follows: Initially, every probe occurs in just one singleton contig, and the relative position of a probe $\tilde{x}_i$ in contig C_i is at the position 0. At any moment, two contigs $C_p = [\tilde{x}_{p_1}, \tilde{x}_{p_2}, \ldots, \tilde{x}_{p_l}]$ and $C_q = [\tilde{x}_{q_1}, \tilde{x}_{q_2}, \ldots, \tilde{x}_{q_m}]$ may be considered for a "join" operation: the result is either a failure to join the contigs C_p and C_q or a new contig C_r containing the probes from the constituent contigs. Without loss of generality on set is not smaller than the other, assume that $|C_p| \geq |C_q|$, and that the probe corresponding to the right end of the first contig (x_{p_l}) is closet to the left end of the other contig (x_{q_1}). That is the estimated distance d_{p_l,q_1} is smaller than all other estimated distances: d_{p_1,q_1}, d_{p_1,q_m} and d_{p_l,q_m}.

Let $0 < \theta \leq 1$ be a parameter to be explored further later, and $L' = L\theta \leq L$. If $d_{p_l,q_1} \geq L'$ then the join operation fails. Otherwise, the join operation succeeds with the probes of C_p placed to the left of the probes of C_q, with all the relative positions of the probes of each contig left undisturbed. We will estimate the distance between the probes in C_p and the probe x_{q_1} by minimizing the function:

$$\text{minimize} \sum_{i\in\{p_1,\ldots,p_l\}:d_{i,q_1}<L'} \frac{(\tilde{x}_{q_1} - \tilde{x}_i - d_{i,q_1})^2}{2\sigma^2 d_{i,q_1}}, \tag{2.50}$$

where $\tilde{x}_i$'s ($i \in \{p_1, \ldots, p_l\}$) are fixed by the locations assigned in the contig C_p. Thus taking a derivative of the expression above with respect to $\tilde{x}_{q_1}$ and equating it to zero, we see that the optimal location for x_{q_1} in C_r is

$$d^* = \max\left[\tilde{x}_{p_l}, \frac{\sum_{i\in\{p_1,\ldots,p_l\}:d_{i,q_1}<L'} \left(\tilde{x}_i + d_{i,q_1}\right)/\sigma^2 d_{i,q_1}}{\sum_{i\in\{p_1,\ldots,p_l\}:d_{i,q_1}<L'} 1/\sigma^2 d_{i,q_1}}\right]. \tag{2.51}$$

Once the location of x_{q_1} is determined in C_r at d^*, the locations of all other probes of C_q in the new contig C_r are computed by shifting them by the value d^*. Thus

$$C_r = [\tilde{x}_{r_1}, \ldots, \tilde{x}_{r_l}, \tilde{x}_{r_{l+1}}, \ldots, \tilde{x}_{r_{l+m}}], \tag{2.52}$$

where $r_i = p_i$ and $\tilde{x}_{r_i} = \tilde{x}_{p_i}$, for $1 \leq i \leq l$; $r_{l+i} = q_i$ and $\tilde{x}_{r_{l+i}} = d^* + \tilde{x}_{q_i}$, for $1 \leq i \leq m$. Note that when the join succeeds, the distance between the pair of consecutive probes $\tilde{x}_{r_l}$ and $\tilde{x}_{r_{l+1}}$ is

$$0 \leq \tilde{x}_{r_{l+1}} - \tilde{x}_{r_l} \leq L', \tag{2.53}$$

and the distances between all other consecutive pairs are exactly the same as what they were in the original constituent contigs. Thus, in any contig, the distance between every pair of consecutive probes takes a value between 0 and L'. Note

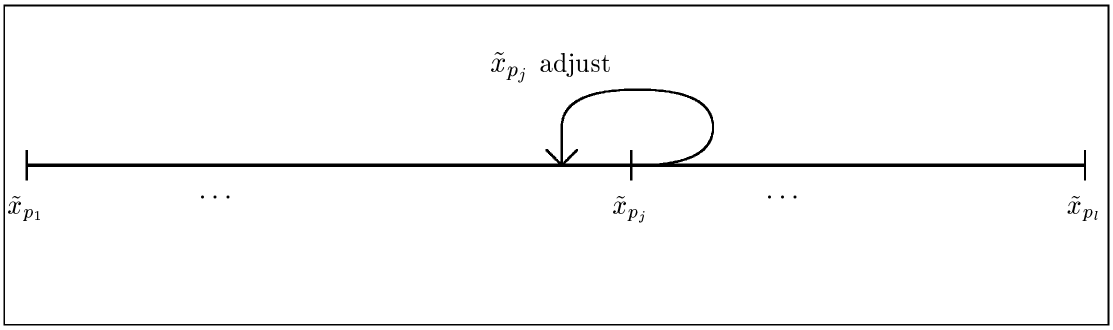

Figure 2.3: Adjust operation

that one may further simplify the distance computation by simply considering the k nearest neighbors of $\tilde{x}_{q_1}$ from the contig C_p: namely, $\tilde{x}_{p_{l-k+1}}$, ..., $\tilde{x}_{p_l}$.

$$d_k^* = \max\left[\tilde{x}_{p_l}, \frac{\sum_{i\in\{p_{l-k+1},\ldots,p_l\}:d_{i,q_1}<L'}\left(\tilde{x}_i + d_{i,q_1}\right)/\sigma^2 d_{i,q_1}}{\sum_{i\in\{p_{l-k+1},\ldots,p_l\}:d_{i,q_1}<L'} 1/\sigma^2 d_{i,q_1}}\right]. \qquad (2.54)$$

In the greediest version of the algorithm $k = 1$ and

$$d_1^* = \tilde{x}_{p_l} + d_{p_l,q_1}, \qquad (2.55)$$

as one ignores all other distance measurements.

At any point we can also improve the distances in a contig, by running an "adjust" operation on a contig C_p with respect to a probe $\tilde{x}_{p_j}$, where

$$C_p = [\tilde{x}_{p_1}, \ldots, \tilde{x}_{p_{j-1}}, \tilde{x}_{p_j}, \tilde{x}_{p_{j+1}}, \ldots, \tilde{x}_{p_l}.] \qquad (2.56)$$

We achieve this by minimizing the following cost function:

$$\text{minimize} \sum_{i\in\{p_1,\ldots,p_l\}\setminus\{p_j\}:d_{i,p_j}<L'} \frac{(|\tilde{x}_{p_j} - \tilde{x}_i| - d_{i,p_j})^2}{2\sigma^2 d_{i,p_j}}, \qquad (2.57)$$

where $\tilde{x}_i$'s ($i \in \{p_1, \ldots, p_l\} \setminus \{p_j\}$) are fixed by the locations assigned in the contig C_p.

Let:

$$I_1 = \{i_1 \in \{p_1, \ldots, p_{j-1}\} : d_{i_1,p_j} < L'\} \tag{2.58}$$

$$I_2 = \{i_2 \in \{p_{j+1}, \ldots, p_l\} : d_{i_2,p_j} < L'\} \tag{2.59}$$

$$x^* = \frac{\sum_{i_1 \in I_1} \left(\tilde{x}_{i_1} + d_{i_1,p_j}\right)/\sigma^2 d_{i_1,p_j} + \sum_{i_2 \in I_2} \left(\tilde{x}_{i_2} - d_{i_2,p_j}\right)/\sigma^2 d_{i_2,p_j}}{\sum_{i_1 \in I_1} 1/\sigma^2 d_{i_1,q_1} + \sum_{i_2 \in I_2} 1/\sigma^2 d_{i_2,p_j}}. \tag{2.60}$$

At this point, if $x^* \neq \tilde{x}_{p_j}$, then the new position of the probe $\tilde{x}_{p_j}$ in the contig C_p is x^*. As before, one can use various approximate version of the update rule, where only k probes from the left and k probes from the right are considered and in the greediest version only the two nearest neighbors are considered. Note that the "adjust" operation always improves the quadratic cost function of the contig locally and since it is positive valued and bounded away from zero, the iterative improvement operations terminate.

2.2 Implementation of the $k-$neighbor Algorithm

`INPUT`

The input domain is a probe set V, and a symmetric positive real-valued distance weight matrix $D \in \mathbb{R}_+^{P \times P}$, where $P = |V|$.

`PRE-PROCESS`

Construct a graph $\mathcal{G}' = \langle V, E' \rangle$, where $E' = \{e_k = (x_i, x_y) | d_{i,j} < L'\}$. The edge set of the graph $\mathcal{G}'$ is sorted into an increasing order as follows: e_1, e_2, …, e_Q, with $Q = |E'|$ such that for any two edges $e_{k_1} = [x_{i_1}, x_{j_1}]$ and $e_{k_2} = [x_{i_2}, x_{j_2}]$, if

$k_1 < k_2$ then $d_{i_1,j_1} \leq d_{i_2,j_2}$. $\mathcal{G}'$ can be constructed in $O(|V|^2)$ time, and its edges can be sorted in $O(|E'|\log(|V|))$ time. In a simpler version of the algorithm it will suffice to sort the edges into an "approximate" increasing order by a parameter $H_{i,j}$ (related to $d_{i,j}$) that takes values between 0 and M. Such a simplification would result in an algorithm with $O(|E'|\log M)$ runtime.

`MAIN ALGORITHM`

Data-structure: Contigs are maintained in a modified union–find structure designed to encode a collection of disjoint unordered sets of probes which may be merged at any time. Union-find supports two operations, *union* and *find* , union merges two sets into one larger set, find identifies the set an element is in. At any instant, a is represented by the following:

- Doubly linked list of probes giving left and right neighbor with estimated consecutive neighbor distances.
- Boundary probes: each contig has a reference to left and right most probes.

In the kth step of the algorithm consider edge $e_k = [x_i, x_j]$: if find(x_i) and find(x_j) are in distinct contigs C_p and C_q, then join C_p and C_q, and update a single distance to neighbor entry in one of the contigs.

At the termination of this phase of the algorithm, one may repeatedly choose a random probe in a randomly chosen contig and apply an "adjust" operation.

`OUTPUT`

A collection of probe contigs with probe positions relative to the anchoring probe for that contig.

Time Complexity

First we estimate the time complexity of the main algorithm implementing the $k-$neighbor version: For each $e \in E'$ there are two find operations. The number of union operations cannot exceed the number of probes $P = |V|$, as every successful join operation leading to a union operation involves a boundary vertex of a contig. Any vertex during its life time can appear at most twice as a boundary vertex of a contig, taking part in a successful join operation. The time cost of a single find operation is at most $\gamma(P)$, where γ is the inverse of Ackermann's function. Hence the time cost of all union-find operations is at most $O(|E'|\gamma(P))$. The join operation on the other hand requires running the $k-$neighbor optimization routine which is done at a cost $O(k)$. Thus the main algorithm has a worst case time complexity of:

$$O\Big(|E'|\gamma(|V|) + k|V|\Big) \tag{2.61}$$

The Full Algorithm including preprocessing is:

$$O\Big(|E'|\log(|V|) + |V|^2\Big) \tag{2.62}$$

In a slightly more robust version the contigs may be represented by a dynamic balanced binary search tree which admit find and implant operations. Each operation has worst case time complexity of $O(\log(|V|))$. Thus after summing over all $|E'|$ operations the worst case runtime for the main algorithm is:

$$O\Big(|E'|\log(|V|) + k|V|\Big) \tag{2.63}$$

and for the full algorithm is:

$$O\Big(|E'|\log(|V|) + |V|^2\Big) \tag{2.64}$$

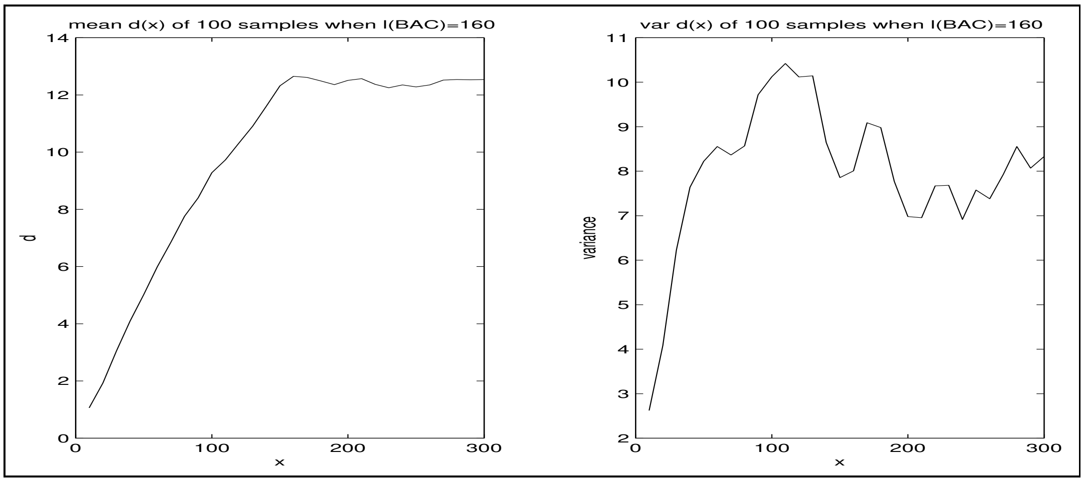

Figure 2.4: Simulation statistics

2.3 Examples and Analysis

Simulation: observed distance

The sample mean and variation of the distance function are computed with a simple simulation done in-silico. Bacterial artificial chromosomes (BACs) are 160Kb in length, we generate $1,200$ BACs and place them randomly on a genome of size $G = 32,000$Kb, This gives a $6\times$ BAC set. In this experiment a random point is chosen on the genome and we use a hypothetical clone library covering this point as well as points $10, 20, 30, \ldots 300$ Kb to the right on the Genome to compute the Hamming distance, we repeat the experiment for 100 such random points. Color sequences are computed by using 20 samples of 130 randomly chosen BACs of which half are likely to be red and the other half green.

Simulation: full experiment

Below we describe an in-silico experiment for a problem with 150 probes. On a Genome of size 5,000 Kb we randomly place 150 probes, their positions are graphed as a monotone function in the probe index. Next we construct a population of 500 randomly placed BACs. From the population we repeat a sampling experiment using a sample size of 32 BACS 16 are colored red, and 16 are colored green. Each sample is hybridized in-silico to the probe set. Here we assume a perfect hybridization so there are no cross hybridizations or failures in hybridizations associated with the experiment. We repeat the sample experiment 130 times. This produces the observed distance matrix, whose distribution we modeled earlier. This is the input for the algorithm presented in this paper. In the distance vs observed data plot we see that using a large $M = 130$ (suggested by the Chernoff Bounds) has its benefits in cutting down the rate of the false positives. The observed distance matrix is input into the ($10-$neighbor, $\theta = \frac{11}{16}$) algorithm without the use of the adjust operation, the result is 7 contigs. The order within contigs had five mistakes. We look at the the 4th contig and plot the relative error in probe placement.

2.3.1 Analyzing the Algorithm Results

In this section we develop an idea for a proof of correctness for a randomized algorithm.

Chernoff bounds for Proof of Correctness, False Positives, False Negatives

We treat the problem of false positives, and false negatives with Chernoff's tail bounds. We find upper bounds on the probability of getting a false positive or false

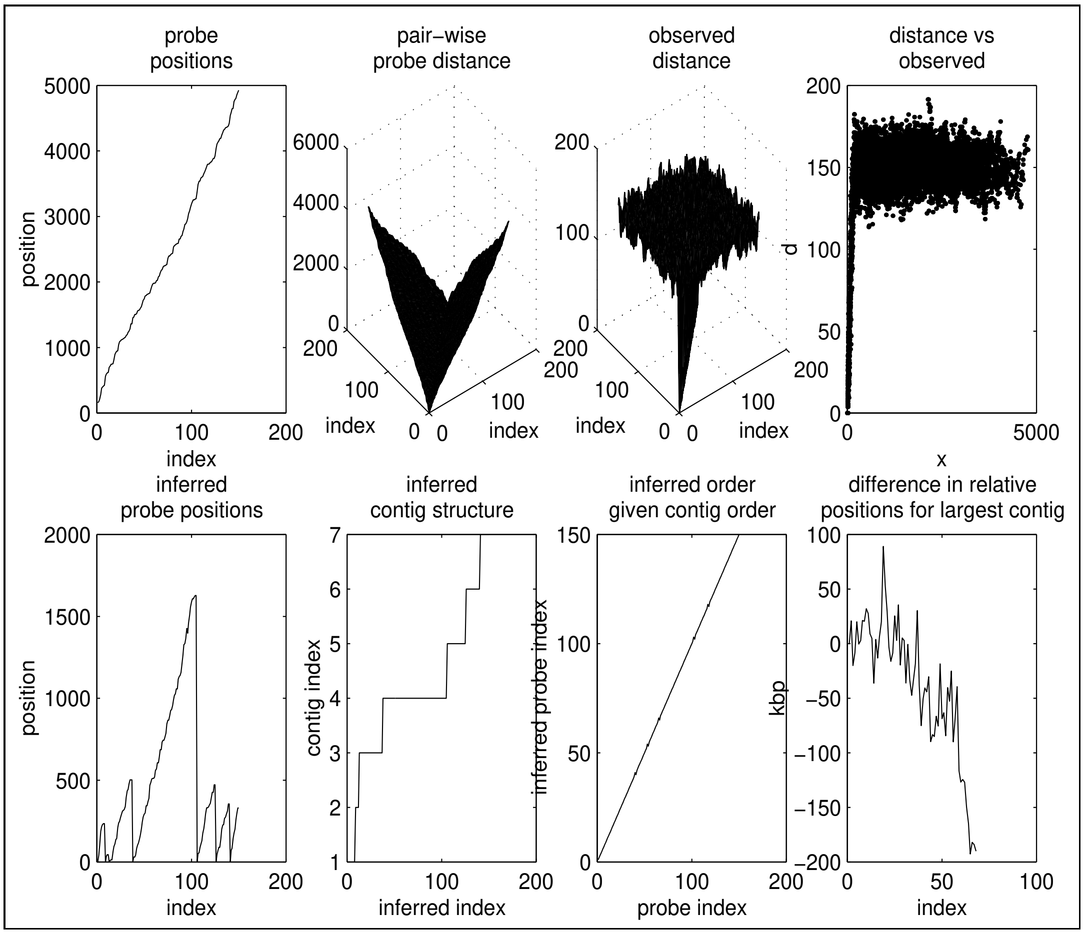

Figure 2.5: Simulation results

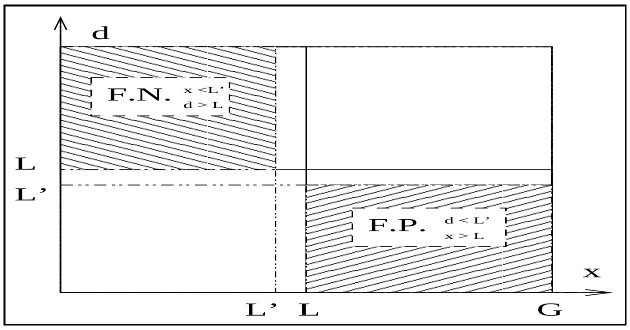

Figure 2.6: Events to bounds

negative in terms of the parameters $\theta, M, c = \frac{KL}{G}$, $0 \leq \theta \leq 1$, $L' = L\theta \leq L$.

A *false positive* is a pair of probes that appear to be close by the Hamming Distance but are actually far apart on the genome. We denote the event as:

$$\text{F.P.} = (d < L') \wedge (x > L)$$

A *false negative* is a pair of probes that appear to be far by the Hamming Distance but are actually close on the genome. We denote the event as:

$$\text{F.N.} = (x < L') \wedge (d > L)$$

In the following picture the volume of data which are false positives and false negatives are indicated by the squares noted F.P. and F.N. respectively.

We develop a Chernoff Bound to bound the probability that the volume of false positive data is greater than a specified size.

The Chernoff Bounds for a Binomial Distribution with parameters (M, q) are given by:

$$P(H > (1+v)Mq) < \left(\frac{e^v}{(1+v)^{(1+v)}}\right)^{Mq} \qquad \text{with } v > 0 \tag{2.65}$$

$$P(H < \theta Mq) < e^{\frac{-Mq(1-\theta)^2}{2}} \qquad \text{with } 0 \leq \theta < 1 \tag{2.66}$$

$$\tag{2.67}$$

Let $H(M)$ be the Hamming distance when M phases are complete. Let $q(L) = P(H|x \geq L) \approx \frac{2\alpha L}{e^{\frac{c}{2}}} = \frac{2c}{e^{\frac{c}{2}}}$ We start by noting equivalent events:

$$\begin{aligned}
(d < \theta L | x > L) &= (\sigma^2 H(M) < \theta L | x > L) && (2.68)\\
&= (H(M) < \theta \frac{L}{\sigma^2} | x > L) && (2.69)\\
&\subset (H(M) < \theta \frac{2cM}{e^{\frac{c}{2}}}) && (2.70)\\
&= (H(M) < \theta M q(L)) && (2.71)\\
& && (2.72)
\end{aligned}$$

Using the Chernoff bound we have:

$$P(d < \theta L | x > L) \leq P(H(M) < \theta M q_L) < e^{\frac{-Mc(1-\theta)^2}{e^{\frac{c}{2}}}}$$

For the False Negatives we begin by noting that:

$$\begin{aligned}
(d > L | x \leq \theta L) &= (\sigma^2 H(x) > L | x < L') && (2.73)\\
&= (\sigma^2 H(x) > (1+v)L' | x < L') \text{ where } v = (\frac{1}{\theta} - 1) && (2.74)\\
&= (H(x) > \frac{(1+v)L'}{\sigma^2} | x < L') && (2.75)\\
&\subset (H(x) > (1+v)Mq(x)) && (2.76)\\
& && (2.77)
\end{aligned}$$

The last event inclusion is due to the following:

$$(x \leq L') \Rightarrow \big(\frac{2cMx}{e^{\frac{c}{2}}L} \leq \frac{2cML'}{e^{\frac{c}{2}}L}\big) \Rightarrow \big(Mq(x) \leq \frac{1}{\sigma^2}L'\big)$$

Applying the Chernoff bound we get:

$$\begin{aligned} P(\text{ F.N. }) &\leq P(H > (1+v)Mq(x)) && (2.78) \\ &< \left(\frac{e^v}{(1+v)^{(1+v)}}\right)^{Mq(x)} && (2.79) \\ &< (e^{(\frac{1}{\theta}-1)}\theta^{\frac{1}{\theta}})^{Mq_L} = (e^{(\frac{1}{\theta}-1)}\theta^{\frac{1}{\theta}})^{M\frac{2c}{e^{\frac{c}{2}}}} && (2.80) \\ & && (2.81) \end{aligned}$$

Chernoff bounds are:

$$\begin{aligned} P(\text{ F.P. }) &< e^{\frac{-Mc(1-\theta)^2}{e^{\frac{c}{2}}}} && (2.82) \\ P(\text{ F.N. }) &< (e^{(\frac{1}{\theta}-1)}\theta^{\frac{1}{\theta}})^{M\frac{2c}{e^{\frac{c}{2}}}} && (2.83) \\ & && (2.84) \end{aligned}$$

The Chernoff bounds for typical parameters are shown below. And one can use the bounds to prescribe good parameters for probe mapping experiments. Further they give us confirmation that our heuristic algorithms provide the correct solutions.

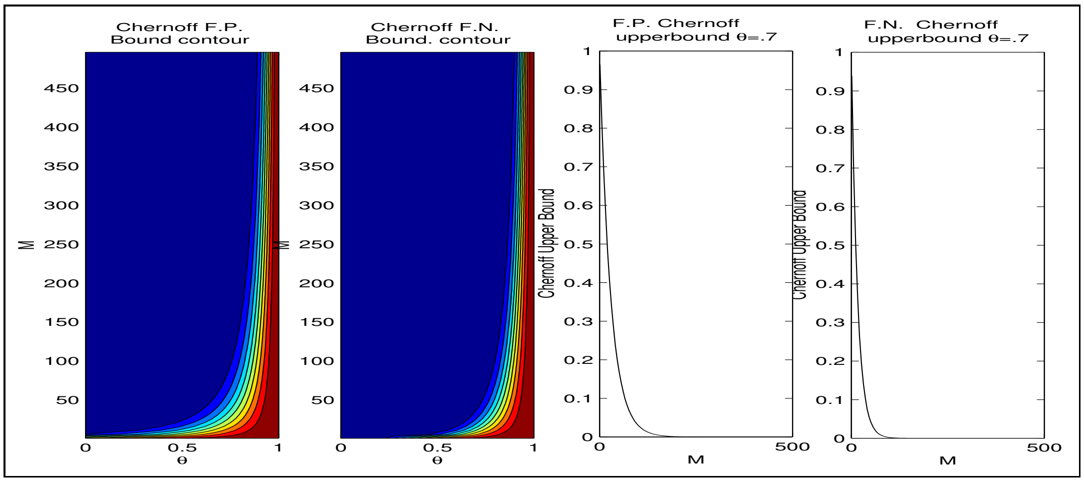

Figure 2.7: Chernoff bounds

Chapter 3

Application II, RFLP Phasing

3.0.2 Restriction Fragment Length Polymorphism or RFLPs

Restriction enzyme site are sequence where specific restriction enzymes may bind to genomic DNA, they are important markers for several applications in genome mapping or analysis. Each chromosome copy *Haplotype* in the diploid organism contains a roughly similar map of restriction enzyme sites. Knowing a restriction enzyme site map for each haplotype provides a possible tool to investigate genetic mutation and disease. Genomic material from the cell cannot be extracted for each Haplotype by Haplotype with current technologies, so we focus on the problem of sorting out haplotypes from the genotype data.

Building on the technology of optical mapping which provides a local view of restriction fragment sites found on molecules derived from one of the two haplotypes, we develop tools to understand the haplotypes underlying the mixture of data. The basic unit of measurement is the length between two consecutive restriction fragment sites, this is know as the restriction fragment length, and under some conditions there is a significant difference in the corresponding length between two consecutive restriction fragment sites on the two haplotypes. This difference is called a *Restriction fragment length Polymorphism* or RFLP, and in combination with optical mapping tools lengths of RFLPs may be understood pairwise if the RFLPs are found on any molecule derived from one of the Haplotypes, in this way co–association of lengths may be understood for each of the haplotypes.

In this application we focus on the development of feasible algorithms for finding the co–associations of lengths for each haplotype, we present a contig algorithm which discovers the co–association of RFLPs in increasingly larger contiguous regions of genotype data for which restriction site maps are known. Finally the results of these feasible algorithms are analyzed by providing probabilities of successful computation given that the data conforms to our error model. Chernoff bounds which provide limiting bound for the the probabilities of False Positives and False Negatives are computed over the parameter space of experiments, these give a probability bound that our computation has failed to find the correct answer.

3.0.3 Formulating the Problem

We consider a set of M fragments of average length L which cover the genome of length G with coverage $c = \frac{ML}{G}$. On this genome we have a set of N restriction sites. Each molecule is a contiguous region from one of two haplotypes, and contained on the molecule are some restriction sites. Each molecule provides a local view of the ordered restriction sites taken from one haplotype. For each of the restriction sites found on a molecule we have data for position in the interval $[1, G]$. We assume that non–digestion rates are negligible, and that distance data may be scaled to a consensus map so that positional data may be understood.

We model the position of a particular restriction site as a random variable whose p.d.f. function is a convex sum of two normal random p.d.f.'s with means representing positions of the restriction sites on each of the two haplotypes. Further we assume a common standard deviation as a constant for all data derived from restriction sites[2]. The genotype restriction site position is modeled as a random variable with p.d.f.:

$$\theta \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-\left(x-\mu_1\right)^2}{2\sigma^2} + (1-\theta) \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-\left(x-\mu_2\right)^2}{2\sigma^2} \tag{3.1}$$

Index the restriction sites from 1 to N and index the molecules from 1 to M. The input is assumed to be a set of restriction sites with positions for each molecule. Since we may assume that the map of these restriction sites is complete, each restriction site found on any molecule may be associated with a 'consensus' restriction site for the genome map. The input may be formalized as a large banded matrix D. D is an $M \times N$ matrix where non-zero entries in column j represent restriction site positions found in the data which correspond the restriction fragment site indexed by j. Non-zero entries found in row i correspond to restriction sites positions that were found on the molecule indexed by i.

D is a large banded matrix whose band width is equal to the coverage c in expectation, hence the expected sparsity, ratio of non–zero entries to total entries, of matrix D is $\frac{cF}{MF} = \frac{c}{M}$ or $\frac{L}{G}$.

Our first problem is to determine $RFLP$ events accurately, sometimes with small number of data points. This is called the Mixture Problem and we give an EM algorithm to solve the mixture problem. Once $RFLP$s have been estimated we would like to understand how they fit together pairwise, we work locally by developing a *contig* algorithm based on the relative phasing for pairwise RFLPs found in phased contigs stretching distances greater than L bases apart. The contig–algorithm is done with a combination of Union-Find [30] data structures for the maintenance of disjoint sets and Tree data structures for maintaining the history of contigs and computed $RFLP$ phasing.

We also provide analysis for results arising from model data.

3.0.4 EM for Mixture Problems

In model problems where data is derived from a mixture of two Gaussian, mixture in the sense that the cumulative distribution function for the data is a convex combination of two cumulative distribution functions taken from two separate Gaussians. The mixture problem is to infer the parameters of our mixture distribution. On approach to inferring parameters for such a distribution is the expectation maximization algorithm or EM algorithm for short. The expectation maximization algorithm finds parameter values Θ for the model distribution which best describe the data D, This is stated as an optimization problem, maximizing the objective function:

$$P(D|\Theta). \tag{3.2}$$

Consider the jth column of D and take the non-zero entries as a column vector denoted by a and consider it as a derivate of random vector A, an $n \times 1$ vector with $\langle A_i \rangle_{i=1:n}$ i.i.d. random variables representing the position of restriction sites taken from a distribution with p.d.f. function:

$$f(A_i = x) = q_1 \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x-\mu_1)^2}{2\sigma^2} + q_2 \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x-\mu_2)^2}{2\sigma^2} \tag{3.3}$$

We do not know the parameters yet but without any loss of generality we may assume that $\mu_1 \leq \mu_2$. Also $q_i, i = \{1, 2\}$ may be interpreted as a probability that point x is derived from the Gaussian with mean μ_i, further we have $\sum_i q_i = 1$.

For each random column A of D there is an estimation problem: We must determine the values of $\Theta := \langle \mu_1, \mu_2, \sigma, q_1 \rangle$. Once this is complete, we may attempt to detect RFLPs as events involving the distance between μ_1 and μ_2, we may also compute probabilities of pairwise events.

The approach that is typically performed in such problem instances is Maximum Likelihood, whereby one evaluates the probability that any particular parameter vector Θ may produce the observed data.

$$L(\Theta) = P(A = a : \Theta) = \Pi_i f_\Theta(A_i = a_i) \tag{3.4}$$

A necessary condition for L to attain a maxima at vector Θ^* is that the gradient vanishes:

$$\frac{\partial L}{\partial \Theta_\zeta}(\Theta^*) = 0 \tag{3.5}$$

for all ζ indexing parameters. If the Hessian or second variation is non-positive then it is a sufficient condition for local maxima and this suggests a value Θ^* for the argument of L which provides the best possible model parameters for the data.

Note that a log likelihood function $L' = \log L$ follows the same principle but has the feature that products of independent random variables are transformed to sums.

We choose to treat one of the parameters q_{1j} as the probability of a hidden random variable for each derivate a_i let Y_{1i} be a Bernoulli random variable whose p-value is equal to q_{1i} and represents the probability that the data point a_i is derived from the Gaussian with the left most mean.

We note that we have a distribution:

$$\begin{aligned} P(A_i = x, Y_{1i} = \nu|\Theta) = \mathbb{I}_{\nu=1} f_1(a_i) + \mathbb{I}_{\nu=2} f_2(a_i) \\ f_1(a_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-\left(a_i - \mu_1\right)^2}{2\sigma^2} \\ f_2(a_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-\left(a_i - \mu_2\right)^2}{2\sigma^2} \end{aligned}$$

whose marginals are:

$$\begin{aligned} P(A_i = x|\Theta) &= E_\nu[\mathbb{I}_{\nu=1} f_1(a_i) + \mathbb{I}_{\nu=2} f_2(a_i)] \\ &= q_{i1} f_1(a_i) + q_{i2} f_2(a_i) \end{aligned}$$

We now formulate the EM-Algorithm by use of Jensen's Inequality

$$\begin{aligned}
L'(\Theta) &= \log(L(\Theta)) = \log \Pi_{i=1:N} P(A_i = x|\Theta) \\
&= \log \Pi_{i=1:N} \sum_{\nu=1:2} P(A_i = x, Y_i = \nu|\Theta) \\
&= \log \Pi_{i=1:N} \sum_{\nu=1:2} Q(Y_i = \nu) \frac{P(A_i = x, Y_i = \nu|\Theta)}{Q(Y_i = \nu)} \\
&= \sum_{i=1:N} \log \sum_{\nu=1:2} Q(Y_i = \nu) \frac{P(A_i = x, Y_i = \nu|\Theta)}{Q(Y_i = \nu)} \\
&\geq \sum_{i=1:N} \sum_{\nu=1:2} Q(Y_i = \nu) \log \left(\frac{P(A_i = x, Y_i = \nu|\Theta)}{Q(Y_i = \nu)} \right) \\
&= \sum_{i=1:N} \sum_{\nu=1:2} Q(Y_i = \nu) \log P(A_i = x, Y_i = \nu|\Theta) + H(Q_i)
\end{aligned}$$

Here Q is an arbitrary measure and H is the Entropy Function on probability vectors:

$$H(Q) = \sum_i Q_i \log(\frac{1}{Q_i})$$

We define the function:

$$F(Q, \Theta) := \sum_{i=1:N} \sum_{\nu=1:2} Q(Y_i = \nu) \log P(A_i = x, Y_i = \nu|\Theta) + H(Q_i)$$

From now on, the EM–algorithm may be defined as a process of increasing the F function value [27]. We note that a gradient ascent may be performed on the Likelihood surface by successively maximizing Q followed by Θ.

E-Step $Q_{k+1} \leftarrow \{Q^* : \max_Q F(Q, \Theta_k) = F(Q^*, \Theta_k)\}$.

Lemma 3.0.1 (E-Step) *Let Q be a vector $\langle q_{i\nu} \rangle_{i=1:N, \nu=1:2}$ where N is the number of non–zero entries in the column of data. The Arg–Max can be solved for explicitly* *with :*

$$Q_{k+1} = \langle q_{i1}^{k+1} \rangle_{i=1:N}$$
$$q_{i1}^{(k+1)} \leftarrow \left(\frac{1}{\exp\left(\frac{(a_i-\mu_1^{(k)})^2}{2\sigma^{(k)2}} - \frac{(a_i-\mu_2^{(k)})^2}{2\sigma^{(k)2}} \right) + 1} \right)$$

Proof 23 (Proof of lemma 3.0.1) Consider the calculus problem of optimizing:

$$f(\phi) = \phi\,(A_1 - \log(B\phi) + (1-\phi)(A_2 - \log(B(1-\phi)))$$
$$f'(\phi) = 0 \Rightarrow \phi^* = \left(\frac{1}{e^{A_1 - A_2} + 1} \right)$$

And notice that $\phi^* \in (0,1)$. Applying this fact to the optimization problem of finding numbers $\hat{Q} = \langle q_{i\nu} \rangle_{i=1:N,\nu=1:2}$ so that the following function is optimized:

$$\sum_{i=1:n} \sum_{\nu=1:2} ((q_{i\nu} A_{i\nu} - \log(B q_{i\nu}))) \tag{3.6}$$
$$= \sum_{i=1:n} ((q_{i1} A_{i1} - \log(B q_{i1})) + ((1-q_{i1}) A_{i2} - \log(B(1-q_{i1})))) \tag{3.7}$$
$$\tag{3.8}$$

Where $A_{i1} = \frac{(a_i-\mu_1)^2}{2\sigma^2}$ and $A_2 = \frac{(a_i-\mu_2)^2}{2\sigma^2}$ and $B = \sqrt{2\pi\sigma^2}$. We see that the answer is given by maximizing each summand and hence given by:

$$q_{i1} = \left(\frac{1}{e^{\left(\frac{(a_i-\mu_1)^2}{2\sigma^2} - \frac{(a_i-\mu_2)^2}{2\sigma^2}\right)} + 1} \right) \tag{3.9}$$

$$= \left(\frac{e^{\left(\frac{-(a_i-\mu_1)^2}{2\sigma^2}\right)}}{e^{\left(\frac{-(a_i-\mu_1)^2}{2\sigma^2}\right)} + e^{\left(\frac{-(a_i-\mu_2)^2}{2\sigma^2}\right)}} \right) \tag{3.10}$$

$$\tag{3.11}$$

□

M-Step $\Theta_{k+1} \leftarrow \{\Theta^* : \max_\Theta F(Q_{k+1}, \Theta) = F(Q_{k+1}, \Theta^*)\}$.

Lemma 3.0.2 (M-Step) *The Arg-Max can be solved for explicitly with :*

$$\Theta_{k+1} = \langle \mu_1^{(k+1)}, \mu_2^{(k+1)}, \sigma^{(k+1)} \rangle$$
$$\mu_1^{(k+1)} \leftarrow \frac{\sum_{i=1:N} q_{i1}^{(k+1)} a_i}{\sum_{i=1:N} q_{i1}^{(k+1)}}$$
$$\mu_2^{(k+1)} \leftarrow \frac{\sum_{i=1:N} q_{i2}^{(k+1)} a_i}{\sum_{i=1:N} q_{i2}^{(k+1)}}$$
$$\sigma^{(k+1)} \leftarrow \sqrt{\frac{1}{2} \sum_{\nu=1:2} \frac{1}{N} \sum_{i=1:N} q_{\nu i}^{(k+1)} (a_i - \mu_\nu^{(k)})^2}$$

Proof 24 (Proof of lemma 3.0.2) Consider the calculus problem of optimizing the function:

$$
\begin{aligned}
f(\mu_1, \mu_2, \sigma) &= \sum_{i=1:N} \sum_{\nu=1:2} q_{\nu i} \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(a_i - \mu_\nu)^2}{2\sigma^2} \right) + H \\
&= \sum_{i=1:N} \sum_{\nu=1:2} q_{\nu i} \left(\frac{-(a_i - \mu_\nu)^2}{2\sigma^2} - \log \sqrt{2\pi\sigma^2} \right) + H
\end{aligned}
$$

Where H is constant in (μ_1, μ_2, σ). Consider the partial derivative of f with respect to μ_ν:

$$
\begin{aligned}
\frac{\partial f}{\partial \mu_\nu} &= \sum_{i=1:N} \frac{\partial}{\partial \mu_\nu} \Big(\frac{q_{\nu i}}{2\sigma^2} (-a_i^2 + 2a_i\mu_\nu - \mu_\nu^2 - 2\sigma^2 \log \sqrt{2\pi\sigma^2}) \Big) \\
&= \frac{1}{\sigma^2} \sum_{i=1:N} q_{\nu i} a_i - \mu_\nu q_{\nu i}
\end{aligned}
$$

Thus we get

$$
\frac{\partial f}{\partial \mu_\nu} = 0 \Leftarrow \mu_\nu = \frac{\sum_{i=1:N} q_{i\nu} a_i}{\sum_{i=1:N} q_{i\nu}}
$$

Now consider the partial of f with respect to σ:

$$
\begin{aligned}
\frac{\partial f}{\partial \sigma} &= \sum_{i=1:N} \sum_{\nu=1:2} q_{\nu i} \left(\frac{(a_i - \mu_\nu)^2}{2\sigma^2} - 1 \right) \\
&= \frac{1}{2\sigma^2} \sum_{i=1:N} \sum_{\nu=1:2} q_{\nu i} \left((a_i - \mu_\nu)^2 - 2\sigma^2 \right)
\end{aligned}
$$

Thus we get

$$
\frac{\partial f}{\partial \sigma} = 0 \Leftarrow \sigma = \sqrt{\frac{1}{2} \sum_{\nu=1:2} \frac{1}{N} \sum_{i=1:N} q_{i\nu} (a_i - \mu_\nu)^2}
$$

Hence the necessary condition for $(\mu_1^*, \mu_2^*, \sigma^*)$ to be the maximizing argument is that:

$$\mu_1^* = \frac{\sum_{i=1:N} q_{i1} a_i}{\sum_{i=1:N} q_{i1}}$$
$$\mu_2^* = \frac{\sum_{i=1:N} q_{i2} a_i}{\sum_{i=1:N} q_{i2}}$$
$$\sigma^* = \sqrt{\frac{1}{2} \sum_{\nu=1:2} \frac{1}{N} \sum_{i=1:N} q_{\nu i}(a_i - \mu_\nu)^2}$$

□

Lemma 3.0.3 *EM Algorithm finds a non-deceasing Likelihood function limit in the parameter space.*

Proof 25 (Proof of lemma 3.0.3) Consider the E-Step, and distribution $Q_{(k+1)} = P(A_i = x | Y_i = \nu, \Theta_{(k)})$ We note that Jensen's Inequality attains equality:

$$\log \int_x Q_{(k+1)} \frac{P(A_i = x, Y_i = \nu | \Theta_{(k)})}{Q_{(k+1)}} dx \tag{3.12}$$
$$= \log \int_x P(A_i = x, Y_i = \nu | \Theta_{(k)}) dx \tag{3.13}$$
$$= \log P(Y | \Theta_{(k)}) \tag{3.14}$$
$$= \int_x \log P(Y_i = \nu | A_i = x, \Theta_{(k)}) P(A_i = x | Y_i = \nu, \Theta_{(k)}) dx \tag{3.15}$$
$$= \int_x \log \left(\frac{P(A_i = x, Y_i = \nu | \Theta_{(k)})}{P(A_i = x | Y_i = \nu, \Theta_{(k)})} \right) P(A_i = x | Y_i = \nu, \Theta_{(k)}) dx \tag{3.16}$$
$$= \int_x Q_{(k+1)} \log \left(\frac{P(A_i = x, Y_i = \nu | \Theta_{(k)})}{Q_{(k+1)}} \right) dx \tag{3.17}$$

We can argue that if we can complete the E–Step then we have evaluated $Q_{(k+1)} \leftarrow P(A_i = x | Y_i = \nu, \Theta_{(k)})$ At this value of $Q_{(k+1)} = P(A_i = x | Y_i = \nu, \Theta_{(k)})$and we have an equality:

$$F(Q_{(k+1)}, \Theta_{(k)}) = L(\Theta_{(k)})$$

Showing that at each of the M-Step we are increasing the Likelihood function by finding a $\Theta_{(k+1)}$. $\square$

With the Lemmas we assume we have procedures called ESTEP and MSTEP. The EM-Algorithm is now:

Algorithm 3

```
EM( A )
    QPREV ← .5*ONES( MAX(SIZE(A)), 2)
    M ← MEAN(A )
    S ← STD(A )
    TPREV ← ( M( 1- S), M( 1 + S), S )
    QNEW ← INF
    TNEW ← INF
    WHILE( MAX( NORM( QPREV - QNEW ) , NORM( TPREV - TNEW ) ) > ε )
        QNEW ← ESTEP( QPREV, TPREV )
        TNEW ← MSTEP( QNEW, TPREV )
    ENDWHILE
    return ( QNEW, TNEW )
```

Detection of RFLPs

We define a detected RFLP as an outcome to our EM algorithm, it is an event such that $|\mu_2 - \mu_1| > \delta$ for some positive δ possibly related to σ, see the section on EM and Detection of RFLPs for further results.

3.0.5 Data Maps to Group Elements, MLE homomorphism

The results of EM on each column A_j of D is value $(Q(j), \Theta(j))$.

Consider a data point in the jth column $d_{i'j}$, as such it is derived from the distribution given by $Q(j), \Theta(j)$ i.e.:

$$f(x) = q(x)\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp\frac{-(x-\mu_{1j})^2}{2\sigma_j^2} + (1-q(x))\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp\frac{-(x-\mu_{2j})^2}{2\sigma_j^2}$$

Where $q(x)$ is some random function giving p–values of the point at x being derived from the distribution with μ_{1j}. But our EM algorithm has done a fit for this function at the points of interest, so corresponding to $d_{i'j}$ is a value q_i which the EM algorithm has estimated, as the probability that $d_{i'j}$ derives from the left distribution.

Let $p_i = 1 - q_i$ and we identify the data point d_{ij} to the 2×2 matrix:

$$\begin{bmatrix} q_i & p_i \\ p_i & q_i \end{bmatrix}$$

We similarly define a map for each element in the jth column of data, and denote the dependence on column with an additional subscript j.

$$\Phi_j : d_{ij} \to \begin{bmatrix} q_{ji} & p_{ji} \\ p_{ji} & q_{ji} \end{bmatrix}$$

The map is an injection into the set $\mathcal{G}'$ the 2×2 symmetric matrices whose entries are such that each column adds to one and each row adds to one. $\mathcal{G}'$ is a set with a natural group structure.

Let us define a continuous Abelian group by its set members and its operation:

$$\mathcal{G} = \left\{ \begin{bmatrix} a & b \\ b & a \end{bmatrix} : a \neq b, a + b = 1 \right\}$$

$$* : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G} : \begin{bmatrix} a & b \\ b & a \end{bmatrix} \begin{bmatrix} A & B \\ B & A \end{bmatrix} \rightarrow \begin{bmatrix} (aA + bB) & (aB + bA) \\ (aB + bA) & (aA + bB) \end{bmatrix}$$

To show that $\mathcal{G}$ is closed under the operation (i.e. matrix multiplication) we note that $aA + bB + aB + bA = a(A + B) + b(A + B) = (a + b)(A + B) = 1$ is implied by $(a+b = 1) \cap (A+B = 1)$, and also we have $(a - b \neq 0) \cap (A - B \neq 0) \Rightarrow (a - b)(A - B) \neq 0 \Rightarrow aA + bB \neq aB + bA$. To show that there is an identity element notice that $I_{2\times 2} \in \mathcal{G}$ by satisfying the conditions $1 + 0 = 1$ and $1 \neq 0$, further $\mathcal{G}$ contains inverses for every element as:

$$\begin{bmatrix} a & b \\ b & a \end{bmatrix} \begin{bmatrix} \frac{a}{a^2-b^2} & \frac{-b}{a^2-b^2} \\ \frac{-b}{a^2-b^2} & \frac{a}{a^2-b^2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Observe that $\frac{a}{a^2-b^2} + \frac{-b}{a^2-b^2} = \frac{a-b}{(a-b)(a+b)} = \frac{1}{a+b} = 1$ follows when $a + b = 1$.

We note that $a \neq b$ implies that the matrix has full rank and excludes the case $\begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix}$ which will however be an interesting part of our algorithm as it shall be the designated value of non-RFLPs. Note that the case $a = b$ acts like 0 in the ring $\langle \mathbb{R}, +, * \rangle$. It has no inverse, acts as an idempotent under the operation of multiplication: $\begin{bmatrix} a & a \\ a & a \end{bmatrix} \begin{bmatrix} A & B \\ B & A \end{bmatrix} = \begin{bmatrix} (aA + aB) & (aB + aA) \\ (aB + aA) & (aA + aB) \end{bmatrix}$ but clearly $a(B + A) = a(A + B)$.

Additional Operation and Closure

The set $\mathcal{G}'$ is closed under any affine combination of elements, that is to say : $g_1, g_2, \ldots, g_n \in \mathcal{G}'$ then:

$$\sum_{k=1:n} w_k g_k \in \mathcal{G}'$$

Here $+$ is the usual matrix addition and coefficients w_k are such that $\sum_{k=1:n} w_k = 1$

Computing Pairwise Events

Pairwise events are events which consider the placement of two restriction sites on the same molecule.

Let us focus on two Restriction Sites, site i and site j that are believed to be RFLPs, these sites have non–constant $q_i(x)$,$q_j(x)$ functions.

Consider molecules that span both RFLPs, these molecules contain data points x and y which were used to estimate both the functions $q_i(x)$ and $q_j(x)$, In these estimates the haplotype is revealed, and statistically with a large number of observations we will observe these limiting frequencies.

Let us look at the possible sets of haplotypes:

- case1: haplotype 1 contains the left most restriction site of i and the left most restriction site of j while haplotype 2 contains the right most restriction site of i and the right most restriction site of j. See figure 3.1, denote this event as $(i_{11}, j_{11}) \cap (i_{22}, j_{22})$

- case2: haplotype 1 contains the left most restriction site of i and the right most restriction site of j while haplotype 2 contains the right most restriction site of i and the left most restriction site of j. See figure 3.2, denote this event as $(i_{11}, j_{12}) \cap (i_{22}, j_{21})$

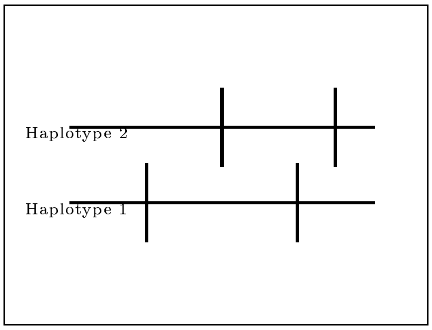

Figure 3.1: Haplotypes, case 1

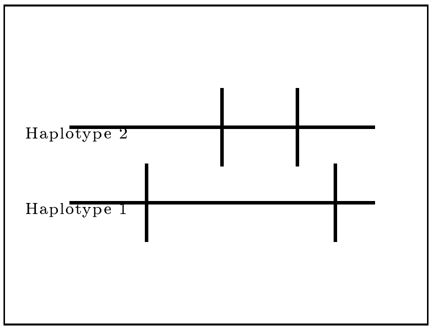

Figure 3.2: Haplotypes, case 2

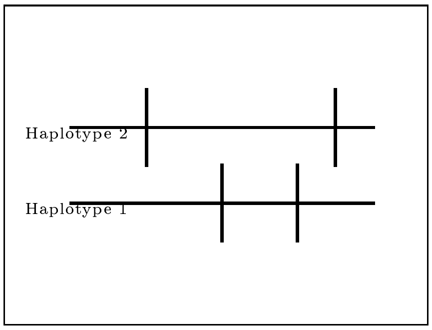

Figure 3.3: Haplotypes, case 3

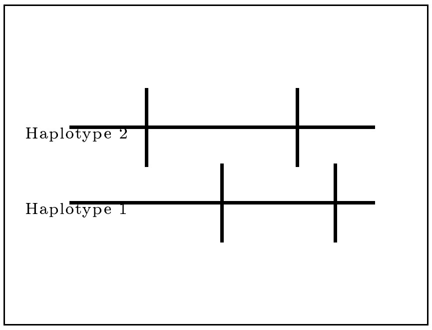

Figure 3.4: Haplotypes, case 4

- case3: haplotype 1 contains the right most restriction site of i and the left most restriction site of j while haplotype 2 contains the left most restriction site of i and the right most restriction site of j . See figure 3.3, denote this event as $(i_{12}, j_{11}) \cap (i_{21}, j_{22})$

- case4: haplotype 1 contains the right most restriction site of i and the right most restriction site of j while haplotype 2 contains the left most restriction site of i and the left most restriction site of j. See figure 3.4, denote this event as $(i_{12}, j_{12}) \cap (i_{21}, j_{21})$

Since we don't care to determine which haplotype the pairs are on, but rather just to determine which pairs are found together on a haplotype. The events of interest are:

$$E_1 = ((i_{11}, j_{11}) \cap (i_{22}, j_{22})) \cup ((i_{12}, j_{12}) \cap (i_{21}, j_{21}))$$
$$E_2 = ((i_{11}, j_{12}) \cap (i_{22}, j_{21})) \cup ((i_{12}, j_{11}) \cap (i_{21}, j_{22}))$$

Now we computing the probability that molecule ζ with point x_ζ and point y_ζ support the event E_1:

$$\begin{aligned}
P(E_1|\zeta) &= P\left((i_{11}, j_{11}, \zeta \in H_1) \cup (i_{22}, j_{22}, \zeta \in H_2)\right) \cup ((i_{12}, j_{12}, \zeta \in H_1) \cap (i_{21}, j_{21}, \zeta \in H_2)) \\
&= P\left((i_{11}, j_{11}) \cup (i_{21}, j_{21})|\zeta \in H_1\right) P(\zeta \in H_1) \\
&\quad + P\left((i_{22}, j_{22}) \cup (i_{21}, j_{21})|\zeta \in H_2\right) P(\zeta \in H_2) \\
&= (q_i(x_\zeta)q_j(x_\zeta) + p_i(y_\zeta)p_j(y_\zeta))(P(\zeta \in H_1) + P(\zeta \in H_2)) \\
&= q_i(x_\zeta)q_j(x_\zeta) + p_i(y_\zeta)p_j(y_\zeta)
\end{aligned}$$

While similarly

$$P(E_2) = q_i(x_\zeta)p_j(x_\zeta) + p_i(y_\zeta)q_j(y_\zeta)$$

But notice the connection with the group structure, $P(E_1)$ is the entry on the diagonal while $P(E_2)$ is the entry on the off diagonal of the product:

$$\begin{bmatrix} P(E_1|\zeta) & P(E_2|\zeta) \\ P(E_2|\zeta) & P(E_1|\zeta) \end{bmatrix} = \begin{bmatrix} q_i(x_\zeta) & p_i(x_\zeta) \\ p_i(x_\zeta) & q_i(x_\zeta) \end{bmatrix} *_{\mathcal{G}} \begin{bmatrix} q_j(y_\zeta) & p_j(y_\zeta) \\ p_j(y_\zeta) & q_j(y_\zeta) \end{bmatrix}$$

Sites on different molecules are independent, and to evaluate the probability of events E_1 given a set of molecules that contain data for both sites i and j we see:

$$\begin{bmatrix} P(E_1 | \cup_{v=1:m} \zeta_v) & P(E_2 | \cup_{v=1:m} \zeta_v) \\ P(E_2 | \cup_{v=1:m} \zeta_v) & P(E_1 | \cup_{v=1:m} \zeta_v) \end{bmatrix} = \sum_{v=1:m} w_v \begin{bmatrix} q_i(x_{\zeta_v}) & p_i(x_{\zeta_v}) \\ p_i(x_{\zeta_v}) & q_i(x_{\zeta_v}) \end{bmatrix} *_{\mathcal{G}} \begin{bmatrix} q_j(y_{\zeta_v}) & p_j(y_{\zeta_v}) \\ p_j(y_{\zeta_v}) & q_j(y_{\zeta_v}) \end{bmatrix}$$

$$\text{with } \sum_{v=1:m} w_v = 1$$

When all molecules are equally informative one should let:

$$w_v = \frac{1}{m}$$

Given two restriction sites α and β, we define the **support** of the pair as: $\text{Supp } (\alpha, \beta) = \{\zeta : d_{\zeta\alpha} \neq 0 \wedge d_{\zeta\beta} \neq 0\}$ or equivalently as the number of molecules indexed by ζ that span both sites. The **Phase** between two sites: RFLP α and RFLP β may be defined as:

$$\phi(\alpha, \beta) = \frac{1}{|\text{ Supp } (\alpha, \beta)|} \sum_{\zeta \in \text{ Supp } (\alpha,\beta)} \Phi_\alpha(x_\zeta) *_{\mathcal{G}} \Phi_\beta(y_\zeta)$$

We can also define the distance between two fragments as:

$$d_{\alpha,\beta} = \frac{1}{|\text{ Supp } (\alpha, \beta)|}$$

Computing all pairwise spins can be done with a few sparse matrix multiplications:

Algorithm 4

PWS (P)

DIST ← (P != 0)'*(P != 0)

Q ← ONES(SIZE(P)) - P

θ ← ((P'*P) + (Q'*Q)) ./DIST

return (θ)

For use in large data sets we use a threshold to guard against a worst case, This idea is explained in the section on Chernoff bounds. We define a **dead state** as a spin $\begin{bmatrix} p & q \\ q & p \end{bmatrix}$ where p is within a $.5\epsilon$ ball of 0.

Algorithm 5

PWSDeadState (P)

$\quad$ pws $\leftarrow$ PWS(P);

$\quad$ pws(((pws $> .5 - \epsilon$) && (pws $< .5 + \epsilon$))) $\leftarrow .5$;

$\quad$ **return** (pws)

3.0.6 Algorithms

We define the phasing problem as: Given a sequence of mixture models whose parameters and distributions have been estimated, use pairwise data to assemble monotonically growing disjoint sets (contigs) where a phasing structure can be inferred, in the sense of maximum likelihood.

Weighted $k-$Neighbor Phase–Contig Algorithm

We can define the phased contigs recursively as follows:

The base cases $C_i = \{F_i\}$ shall be phased $\forall i$ by the function:

$$\Phi(C_i) = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \text{if } F_i \text{ is a detected RFLP, this contig type is Non Trivial} \\ \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix} & \text{otherwise this contig type is Trivial} \end{cases}$$

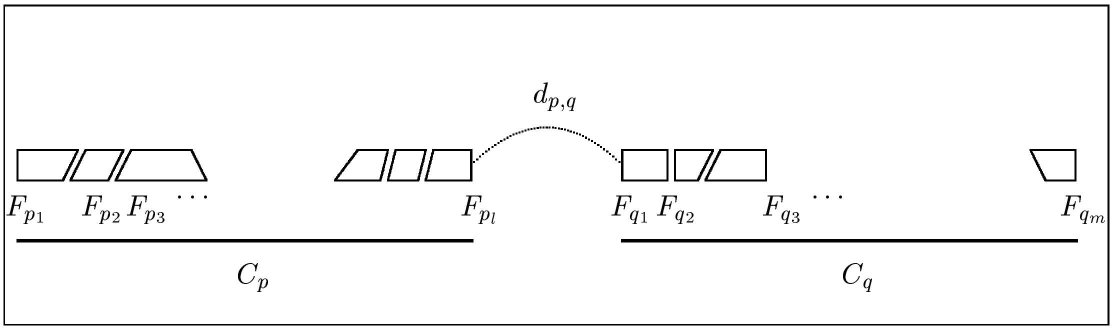

Figure 3.5: Contig distance

If $C_p = \{F_{p_1}, F_{p_2}, \ldots, F_{p_l}\}$ and $C_q = \{F_{q_1}, F_{q_2}, \ldots, F_{q_m}\}$ are phased contigs with well defined **distance** then the union $C_p \cup C_q$ may be phased by our **phase–join** operation.

distance

We define the distance between two phased contigs as the minimum distance between two fragments within contigs.

phase–join

The **phase–join** operation may be performed on C_p and C_q if and only if there is a molecule ζ which contains a data point x_ζ from a restriction site F_{p_l} found in contig C_p, and a data point y_ζ from a restriction site F_{q_1} found in contig C_q, as otherwise the distance will not be defined.

Our phasing may never stretch further a contiguous regions of a map.

For every pair $F_\alpha \in C_p$ and $F_\beta \in C_q$ there are pairwise spins variables to consider in the phase–join. These pairwise spins tell us how to orient the phased–contig C_q relative to the phased–contig C_p we will consider a weighted combination of this information, Weights will depend on distance between fragments, confidence in RFLP assignment etc.

To attempt a join of C_p to C_q we compute a quantity called *mean group action* which is a 'least squares' rotation to be applied similarly to all variables in the right contig to

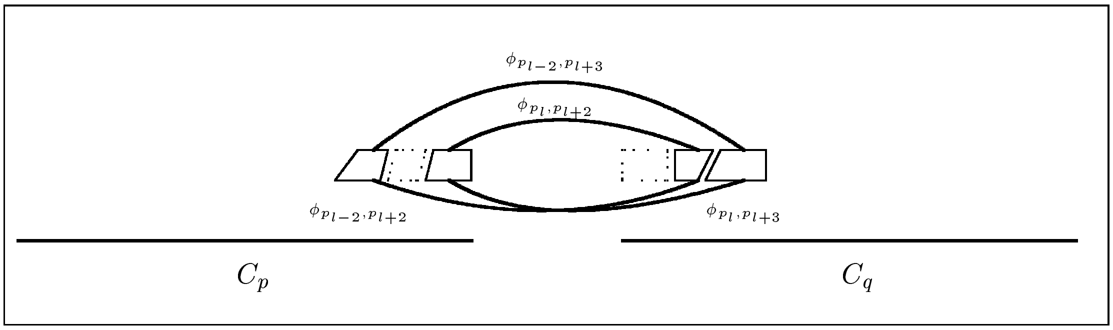

Figure 3.6: Phase weights

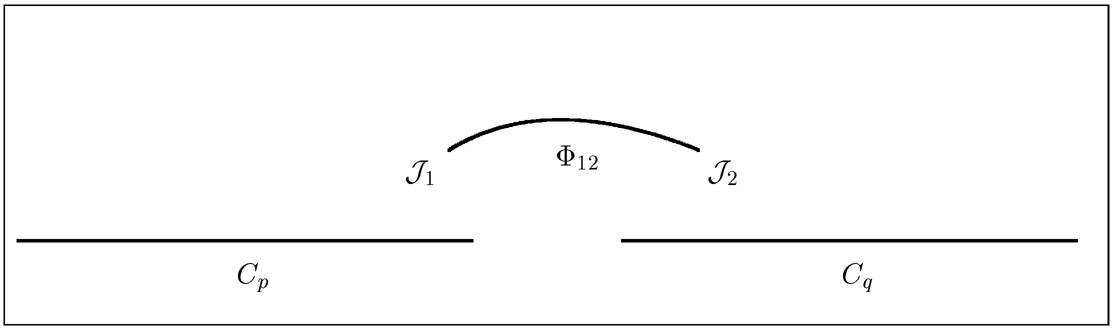

Figure 3.7: Mean group action

make a nice fit for all pairwise spins in the union of $C_p \cup C_q$.

To compute the *group action* for a pair of RFLPs one in each of the phased–contigs with spin assignment $\mathcal{J}_1$ and $\mathcal{J}_2$, and pairwise spin Φ_{12}, we derive the chain of computations, let k_{12} be the *group action* for pair $1, 2$.

$$k_{12}\mathcal{J}_2 = \Phi_{12}\mathcal{J}_1 \tag{3.18}$$

$$k_{12} = \mathcal{J}_2^{-1}\Phi_{12}\mathcal{J}_1 \tag{3.19}$$

$$\tag{3.20}$$

Solving for k_{12} we find the best rotation for these pairs in that after we update the spin $\mathcal{J}_2 \leftarrow k_{12}\mathcal{J}_2$ the variables would be in a state which is satisfies the pairwise spin data. Note that when we work in a group $\mathcal{G}'$ everything is solvable uniquely.

Thus in our algorithm the pair $1, 2$ cast a vote of $k_{12} = \mathcal{J}_2^{-1}\mathcal{J}_1\Phi_{12}$ as the *mean group element* needed to rotate contig C_q into the correct phase.

For each such i, j whose spins are of interest, one can solve uniquely the *pair group action* element k_{ij}

The **mean group action** computed for C_p and C_q is denoted by Φ_{MGA} and is defined by the 'weighted combinations' of *pair group actions* .

$$\Phi_{MGA} = \sum_{F_\alpha \in C_p, F_\beta \in C_q \beta - \alpha < k+1} w_{\alpha\beta} k_{\alpha\beta}$$
$$\sum w_{\alpha\beta} = 1$$

We use weight schemes placing uniform distribution over the detected RFLPs, and in the future will consider the local coverage of the fragments as well, one could also generalize to placing weights on molecule data.

$$\Phi_{MGA} = \sum_{F_\alpha \in C_p, F_\beta \in C_q \beta - \alpha < k+1} \sum_{\zeta \in \text{ Supp } (\alpha,\beta)} w_{\zeta\alpha\beta} k_{\alpha\beta}$$
$$\sum w_{\zeta\alpha\beta} = 1$$

Now if the resulting *mean group action* $\Phi_{MGA} = \begin{bmatrix} p & q \\ q & p \end{bmatrix}$ is not a dead state (i.e. $|p - .5| > \epsilon$), this will be a successful **join** with a parent node $C_p \wedge C_q$ assigned the value $\Phi(C_p \wedge C_q) \leftarrow \Phi_{MGA}$ this node will indicate the new contig as a union of the constituents and will be a non trivial contig. To complete the operation the element in the right hand side will be reassigned spin states.

$$\forall F_\beta \in C_q \Phi_\beta \leftarrow \Phi_{MGA}\Phi_\beta$$

This completes the **JOIN** operation, when the *mean group action* is not in the dead state.

When the *mean group action* $\Phi_{MGA} = \begin{bmatrix} p & q \\ q & p \end{bmatrix}$ is in the dead state (i.e. $|p - .5| < \epsilon$) we will be able to complete the **JOIN** if either of the contigs C_p or C_q is Trivial. If both C_p and C_q are Trivial the resultant **JOIN** $C_p \wedge C_q$ contig is Trivial and assigned spin $\Phi(C_p \wedge C_q) = \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix}$. If only one of the contigs C_p or C_q is Non Trivial the **JOIN** $C_p \wedge C_q$ is Non Trivial, and assigned the spin variable $\Phi(C_p \wedge C_q) = \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix}$. In the last two cases no recomputing is done as this is a *Dead Join* or *Join from the dead state.*

If the *mean group action* is in the dead state and if both contigs C_p and C_q are Non Trivial then the **JOIN** cannot be completed, the operation is aborted, and no **Join** Contig is created. The number of disjoint phased islands goes up by one.

At any given time the history of join–operation is stored as a set of disjoint binary–trees and each node C in any of the contig trees corresponds either to the base case (leaves) or to a historical join–operation (interior), at each node we store the Contig Type and spin variable $\Phi(C)$ which is the *mean group action* in the case of C being a join–operation.

Phase–Adjust

We may also attempt to improve on the phasing locally with an 'adjust' operation which considers a possibly larger neighborhood than the phase–join operation did. This is because the later in time the fuller the possible k–neighborhood is developed.

Phase–Adjust constitutes computing the *mean group action* for all pairwise Fragments in the $k-$neighborhood of F_α let C denote the contig for which F_α is in at the time of the **Adjust**.

$$\Phi_{MGAnew} = \sum_{F_\beta \in C : \beta - \alpha < k+1} w_{\alpha\beta} k_{\alpha\beta}$$

$$k_{\alpha\beta} = \mathcal{J}_\alpha^{-1} \Phi_{\alpha\beta} \mathcal{J}_\beta$$

$$\sum w_{\alpha\beta} = 1$$

Then the Adjust affects only the spin variable at F_α and we have:

$$\mathcal{J}_\alpha \leftarrow \Phi_{MGAnew} \mathcal{J}_\alpha$$

to complete the Adjust Operation.

Implementation of the weighted k–neighbor algorithm

`INPUT`

The input domain is a restriction fragment map F, these are encoded as a large banded matrix D. D is an $M \times N$ matrix where non-zero entries in column j represent restriction site points found in the data which correspond to the left end of fragment j. Non-zero entries found in row i correspond to restriction sites that were found on molecule number i. The band width is equal to the coverage c in expectation. Let us assume that the maximum band-width is c' (slightly greater than c in reasonable examples).

`PRE-PROCESS`

Compute the fragment–to–fragment distance matrix. This can be done by banded matrix multiplication and hence takes time $O(KM)$ where K is related to $\frac{c'M}{N}$ Compute the fragment–to–fragment pairwise spins. This can be done by a small number of banded matrix multiplication and banded matrix additions and hence takes time $O(KM)$ where K is related to $\frac{c'M}{N}$ Construct a sparse graph $\mathcal{K}' = \langle F, E' \rangle$, where $E' = \{e_k = (F_i, F_y) | d_{i,j} < \infty\}$. The edge set of the graph $\mathcal{K}'$ is sorted into an increasing order as follows: e_1, e_2, $\ldots$, e_Q, with $Q = |E'|$ such that for any two edges $e_{k_1} = [F_{i_1}, F_{j_1}]$ and $e_{k_2} = [F_{i_2}, F_{j_2}]$, if $k_1 < k_2$ then $d_{i_1,j_1} \leq d_{i_2,j_2}$. $\mathcal{K}'$ can be constructed in $c'O(|F|)$ time, where constant K is related to the maximum depth of the fragment library, and the edges can be sorted in $O(|F|)$ time as our distance takes on a finite set of values.

`MAIN ALGORITHM`

Data-structure: Contig–Trees are simple binary trees with a spin variable at each node. Contig–Trees are maintained in a Union-Find structure [30] designed to encode a collection of disjoint Contig–Trees which may be merged at any time. To support the set of disjoint Contig–Trees, two operations *union* and *find* are given: union merges two sets into one larger set, find identifies the set an element is in.

At any instant, a phasing is a set of Contig–Trees represented by the following:

- A rooted tree, giving a historical description of how subtrees have been phased.

Initially, every Fragment is its own contig.

In the kth step of the algorithm, consider edge $e_k = [F_i, F_j]$: if find(F_i) and find(F_j) are in distinct contigs C_p and C_q, then phase–join C_p and C_q.

At the termination of this phase of the algorithm, one may repeatedly choose a random fragment in a randomly chosen contig and apply an "adjust" operation.

POSTPROCESS

Starting from the left most fragment assigned a spin $I_{2\times 2}$ in any phased contig, we may compute phasing of every fragment to the right in the phased–contig by multiplying 'through the tree.' If done in a directed fashion all fragments can be assigned a phase in $O(M)$ time, After spins have been resolved we can use the values as a maximum likelihood estimator (MLE) and round to the nearest possible spin state $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. The post–process costs $O(M)$ time .

OUTPUT

A collection of phased RFLPS and non-Phased Restriction sites within a contiguous region of a restriction fragment map.

Time Complexity

First we estimate the time complexity of the main algorithm implementing the $k-$neighbor: For each $e \in E'$ there are two find operations. The number of union operations cannot exceed the number of fragments $N = |F|$, as every join operation resulting in a union, involves a boundary vertex of a contig. Any vertex during its life time can appear at most twice as a boundary vertex of a contig, taking part in a successful join operation. The time cost of a single find operation is at most $\gamma(N)$ where γ is the inverse Ackerman function. Hence the time cost of all union-find operations is at most $O(|E'|\gamma(N))$. The join operation on the other hand requires running the $k-$neighbor phase–join routine which is done at a cost $O(k)$ with looking up tables computed in the pre–process. Unfortunately there is a need to re–compute at least $\min\{k, |C_q|\}$ spins from the right contig C_q with every successful join operation. Thus the main algorithm has a worst case time complexity dominated by the sort of edges:

$$O\Big(|E'|(\gamma(|E|)+k)\Big) \approx 2(c')^2 k O\Big(F\Big)$$

Where c' is a bound on molecule coverage. The full algorithm including preprocessing is:

$$2(c')^2 k O\Big(M\Big) + \big(\frac{2(c')^2 M}{N}\big) O\Big(M\Big) = \big(2(c')^2(k + \frac{M}{N})\big) O\Big(M\Big)$$

where $M = F$ as the number of columns in the banded input matrix.

3.0.7 Analysis

We consider a few problems that aid in our quantitative understanding of the algorithm's results.

Expectation Maximization and the Detection of RFLPs

The results of the EM algorithm are parameter vectors and $q(x)$ function, leaving us with the problem of choosing which estimates indicate RFLPs.

We develop a cut–off function which is designed by considering the Chernoff bounds in the next section, for now we will state the events, and say that we may be liberal in RFLP assignment for sites with large local coverage.

The distinction between an RFLP site and non RFLP site is that the $q(x)$ function which non constant or constant .5. The non RFLP case follows from the condition that $\mu_2 - \mu_1 = 0$. We look upon this the separation of means as the independent variable determining RFLPs. We shall define the event RFLP as:

$$\begin{aligned} \text{detected RFLP } &= (|\hat{\mu}_2 - \hat{\mu}_1| > \delta(c)) \\ \text{RFLP } &= (|\mu_2 - \mu_1| > 0) \end{aligned}$$

Let F denote a restriction length:

$$\begin{aligned} \text{F.P.} &= \Big((F \text{ is not an RFLP }) \wedge \Big(\hat{F} \text{ is a detected RFLP } \Big) \Big) \\ \text{F.N.} &= \Big((F \text{ is an RFLP }) \wedge \Big(\hat{F} \text{ is not a detected RFLP } \Big) \Big) \end{aligned}$$

False Negatives are of no consequence to the correctness of the phasing. They only effectively shorten contiguous phased regions. See the section on computational reductions.

False positives may have consequences to the correctness of the phasing.

3.0.8 Simulations and Examples

We demonstrate our algorithm on a small simulated data set, along the way we point out some features. The view below contains the simulated haplotypes in the bottom–most band of the layout. Above that is the Haplotype Molecule Map for a Diploid Organism, these molecule maps are available to the algorithm the segmentation of the two haplotypes is hidden from the algorithm. The third band indicates estimate values where we can see what sites the EM algorithm chooses as RFLPs. Mistakes occur with the lack of a deep library. The forth band in the Layout indicates the history of contig–operations and from this tree one can 1) view the developing maximum of k–neighbor neighborhoods to compute mean group action, and 2) See the distinct Phased contigs. At last the fifth band in the layout gives the algorithmic output to this problem, complete with phasing in subsets that span the distance indicated by the bars. Notice that where bars cross are areas of interest for extension of the phasing and one may choose to return to these areas with additional target sequences to understand what phasing is occurring through–out these regions.

Parameters of the simulations are:

$$
\begin{aligned}
\text{M} &= \text{number of molecules}\\
\text{F} &= \text{number of fragments RFLP and non RFLP}\\
\text{G} &= \text{size of the genome}\\
\text{EMS} &= \text{expected molecule size}\\
\text{VMS} &= \text{variance in molecule size}\\
\text{VFS} &= \text{variance in fragment length size}\\
\text{P–BIMODE} &= \text{P–value that any given Fragment is an RFLP}\\
\text{ERFLPSEP} &= \text{Expected separation of means for RFLP}\\
\text{VRFLPSEP} &= \text{Variance in the separation of means for RFLP}
\end{aligned}
$$

Any parameter with both an expectation and variance is generated with a normal distribution. From these parameters one can compute some additional symbols that we use in the paper $L =$ EMS and $c = \frac{LM}{G}$.

For the first simulation a relatively small set is chosen so that one can view the action of the algorithm, here the neighborhood size is set to $k = 5$ and there is no ϵ guard of the dead state, still things work pretty well, and one can see that any mistakes are due to the low coverage library.

$$
\begin{aligned}
\text{M} &= 80\\
\text{F} &= 20\\
\text{G} &= 12000\\
\text{EMS} &= 2000\\
\text{VMS} &= 50\\
\text{VFS} &= 1\\
\text{P–BIMODE} &= .5\\
\text{ERFLPSEP} &= 10\\
\text{VRFLPSEP} &= .01
\end{aligned}
$$

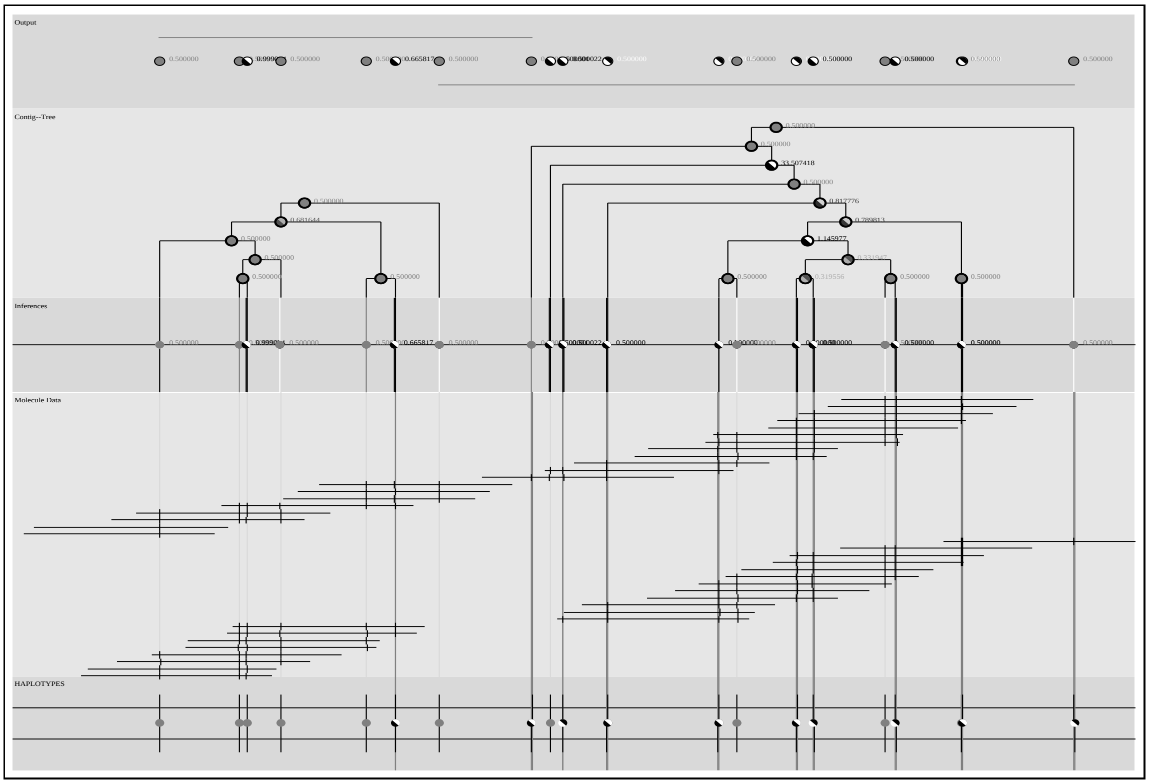

Figure 3.8: Data set

Here we give an illustration that similar results may be achieved on large data sets.

$$
\begin{aligned}
\mathrm{M} &= 150 \\
\mathrm{F} &= 100 \\
\mathrm{G} &= 50000 \\
\mathrm{EMS} &= 2000 \\
\mathrm{VMS} &= 500 \\
\mathrm{VFS} &= 20 \\
\text{P–BIMODE} &= .3 \\
\mathrm{ERFLPSEP} &= 50 \\
\mathrm{VRFLPSEP} &= 6
\end{aligned}
$$

Analyzing the Results, Chernoff bounds for the effect of False Positives

F.P. events may affect the performance of our algorithm. We are particularly interested in bounding the probability of such a restriction length X causing **phase error**, specially when the left neighbors of X are phased incorrectly to the right neighbors of X.

To consider this problem we study the probability that a single F.P. may rotate the right contig in the wrong direction relative to the left contig. In figure 3.10 below indicates the adversary case of interest.

Suppose that molecules $[1, k] \cup [2k+1, 3k]$ are from Haplotype 1 while molecules $[k+1, 2k] \cup [3k+1, 4k]$ are from Haplotype 2. In the image, the dotted lines surround the molecules from haplotype 1. Now let us suppose that there are three sites Site 1, Site 2, and Site 3, site 1 and site 3 are oriented in the same way, while site 2 is not an RFLP but due to an unlikely set of data estimated as an RFLP.

The worst possible F.P. outcome of EM at Site 2 is an estimate of the $q(x)$ function as:

$$q = \begin{cases} 0 & x < \mu \\ 1 & x \geq \mu \end{cases}$$

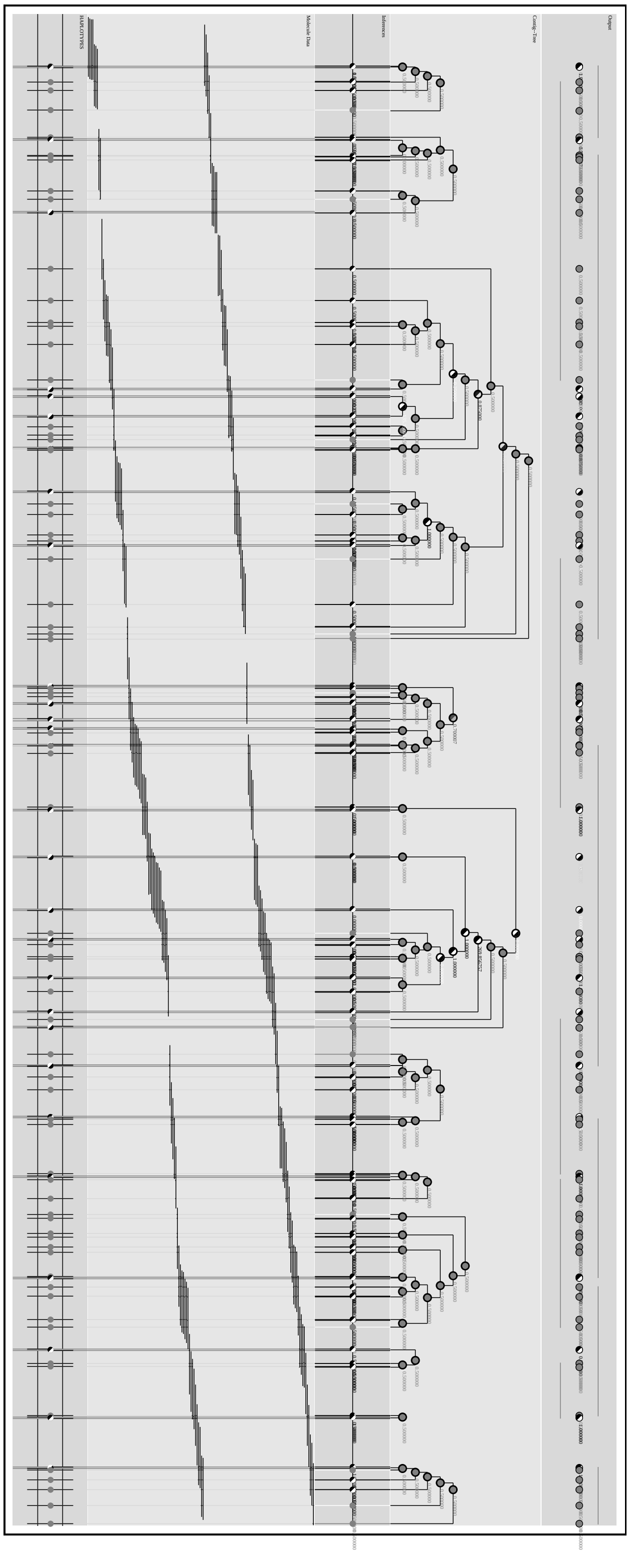

Figure 3.9: Large data set

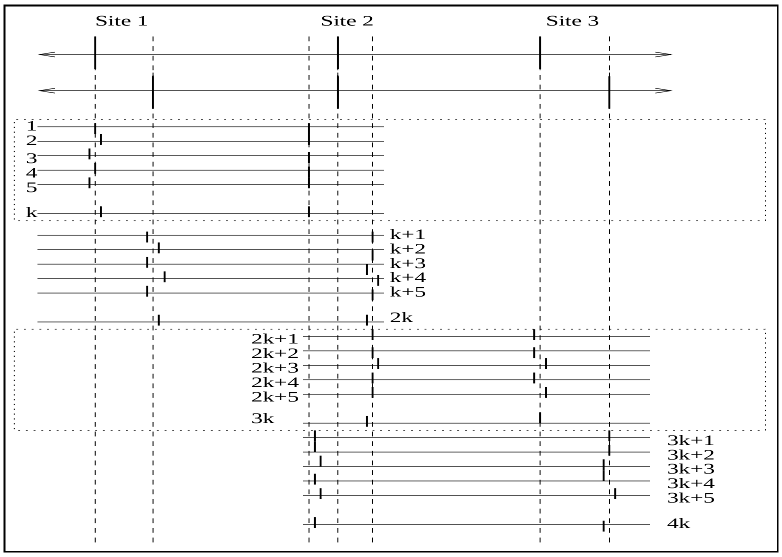

Figure 3.10: Dangerous phasing event

where μ is the actual mean of the site. Given that this worst case is found in our data, let us determine its action in the algorithm as a function of k with $4k$ the local coverage. Let x_j be a Bernoulli random variable with parameter $p = .5$, x_j shall take on the values of { left of mean , right of mean } and indicate whether the point at Site 2 of molecule j falls to the left of μ or to the right, both of which happen with model frequency .5. The sequence $\langle x_j \rangle_{j=1:4k}$ then has an outcome space of cardinality 2^{4k}. When all possible pairwise spins between Site 1, Site 2, and Site 3 are computed, denoted the spins as ϕ_{12}, ϕ_{23}, we are interested in the events that the spins are diabolically opposed meaning $\phi_{12}\phi_{23} = \begin{bmatrix} .5-\delta & .5+\delta \\ .5+\delta & .5-\delta \end{bmatrix}$, for a $\delta > 0$.

Recall that the dead state is the event that $p \in [.5-\epsilon, .5+\epsilon]$. A necessary condition that ϕ_{12} and ϕ_{23} are a set of diabolically opposed spins whose pairwise spin will be incorrectly phased is that:

$$p_{12}p_{23} + (1 - p_{12})(1 - p_{23}) < .5 - \epsilon$$

Let $F(p_{12}, p_{23}) := p_{12}p_{23} + (1 - p_{12})(1 - p_{23})$. The contour plot is depicted below for $p_1, p_2 \in [0, 1]^2$. We develop the event inclusions:

$$\{p_1, p_2 : F(p_1, p_2) < .5 - \epsilon\} \subset ((p_1 < .5 - \epsilon) \wedge (p_2 > .5 + \epsilon)) \vee ((p_1 > .5 + \epsilon) \wedge (p_2 < .5 - \epsilon))$$

Of course the value of p_{12} depends on the *selection* of variables $\langle x_j \rangle_{j=1:2k}$ independent of the *selection* of variables $\langle x_j \rangle_{j=2k+1:4k}$ determining p_{23}.

Redefine the X_j variable as having outcome: restriction site 2 of molecule j falls on ζ side of μ_2 and restriction site 1 of molecule j falls on the ζ side of μ_1 when $j \in [1, 2k]$. Similarly for $j \in [2k + 1, 4k]$ let the outcome space be restriction site 2 of molecule j falls on ζ side of μ_2 and restriction site 3 of molecule j falls on the ζ side of μ_3, here $\zeta \in \{$ Left , Right $\}$. X_j are still Bernoulli trials for all j for Gaussians and bi-mode distributions our means are also medians.

Letting $p_{12} = \frac{1}{2k} \sum_{j=1:2k} X_j$ and $p_{23} = \frac{1}{2k} \sum_{j=2k+1:4k} X_j$ we note that $2k * p_{12}$ is $\mathrm{Bin}(2k, .5)$ and that $2k * p_{23} = \mathrm{Bin}(2k, .5)$. We next apply the Chernoff bounds for binomial random variables.

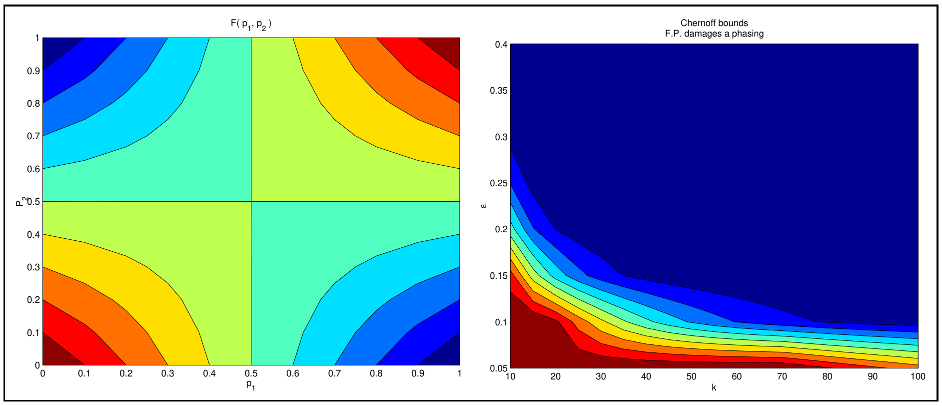

Figure 3.11: Chernoff bounds

Letting $p_1 = p_{12}$ and $p_2 = p_{23}$ Thus:

$$
\begin{aligned}
P(\{p_1, p_2 : F(p_1,p_2) < .5-\epsilon\}) \le P\left((p_1 < .5-\epsilon) \wedge (p_2 > .5+\epsilon)\right) \\
+ P\left((p_1 > .5+\epsilon) \wedge (p_2 < .5-\epsilon)\right) \text{ independence} \\
= 2 * P\left((p_1 < .5-\epsilon) \wedge (p_2 > .5+\epsilon)\right) \text{ symmetry} \\
= 2 * P\left((2kp_1 < 2k(.5-\epsilon)) \wedge (2kp_2 > 2k(.5+\epsilon))\right) \\
= 2 * P\left((2kp_1 < 2k(.5)\Theta) \wedge (2kp_2 > 2k(.5)(1+\nu))\right) \\
\le e^{-.5k(1-\Theta)^2} \left(\frac{e^{(\Theta-1)}}{\Theta^{\Theta}}\right)^k \text{ Chernoff bounds} \\
\le \left(\frac{e^{2(\epsilon-\epsilon^2)}}{(1+2\epsilon)^{(1+2\epsilon)}}\right)^k
\end{aligned}
$$

In the chain above the substitutions $\Theta = 1-2\epsilon$, and $\nu = 1-\theta$ were made. Below some values of Chernoff bound are contoured over the space $k = 5 : 5 : 100$ and $\epsilon = .05 : .05 : .4$.

In an actual problem we will estimate $k = (\frac{1}{4})c_{\text{loc}}$.

These bounds are not even so bad, when weighed against the small probability that such an occurrence of data may happen. The only real problems may occur when k is small.

Combinatorial Reductions, Expectations

If there are F RFLPs there are possibly 2^F possible RFLP configurations, after our algorithm has completed this number will be reduced substantially but by how much?

Essential for our algorithm is that molecules are capable of bridging two RFLPs so that we can observe the correlation, thus the length of molecules L as well as the number of RFLPs F is important.

We will let T be a distance for which we can be almost certain that 2 RFLPs are found within this distance. Then we can declare a random variable:

$V(a, a+T) =$ number of molecules overlaping the region of genome indexed by $[a, a+T]$

Since this event depends on the left end–point of a molecule falling in the region $[a-(L+T), a)$ we can rewrite V as a sum of indicator variables,

$$V(a, a+T) = \sum_{i=1:M} \sum_{j=a-(L+T):(a-1)} S_{ij}$$

$$S_{ij} = \begin{cases} 1 & \text{if Molecule } i \text{ starts at Genome position } j \\ 0 & \text{otherwise} \end{cases}$$

Since S_{ij} is Bernoulli($p = \frac{1}{G}$) where G is the size of the Genome we have $V(a, a+T)$ is Binomial$\big((L-T)M, \frac{1}{G}\big)$ and we use a standard approximation such as Brun's Sieve to approximate the random variable as a Poisson($c\sigma$) with $c = \frac{LM}{G}$ as coverage and $\sigma = (1 - \frac{T}{L})$.

Supposing that we set T large enough so that with probability greater than $1 - \frac{1}{10M}$ we have two RFLPs within a length of T.

We use the above distribution to evaluate:

$$P(V(a, a+T) = 0) \approx e^{-c\sigma}$$

and hence the Probability that any particular Molecule is the left 'anchor' of a phased contig is $e^{-c\sigma}$ and in expectation we'll have

$$M * e^{-c\sigma}$$

Phased contiguous regions.

Thus our algorithm is expected to reduce the possible phasing by a factor of:

$$2^{F - M * e^{-c\sigma}}$$

We derive $\sigma(r_{\text{RFLP}})$ Since we can look at the expected length before the next RFLP as a geometric RV with parameter r_{RFLP} in order to insure two RFLPs within a distance of T with a probability greater than $1 - \frac{1}{10M}$ we compute:

$$
\begin{aligned}
\frac{1}{10M} &> \sum_{n=T}^{\infty} r_{\mathrm{RFLP}}(1-r_{\mathrm{RFLP}})^{n-1} \\
&= r_{\mathrm{RFLP}}(1-r_{\mathrm{RFLP}})^{T-1}\sum_{n=0}^{\infty} r_{\mathrm{RFLP}}(1-r_{\mathrm{RFLP}})^{n} \\
&= r_{\mathrm{RFLP}}(1-r_{\mathrm{RFLP}})^{T-1}\frac{1}{1-(1-r_{\mathrm{RFLP}})} \\
&= (1-r_{\mathrm{RFLP}})^{T-1}
\end{aligned}
$$

Hence

$$
\begin{aligned}
(T-1)-\log(\frac{1}{1-r_{\mathrm{RFLP}}}) &< -\log 10M \\
T &> \frac{\log 10M}{\log(\frac{1}{1-r_{\mathrm{RFLP}}})}+1
\end{aligned}
$$

Letting $T(r)=\frac{\log 10M}{\log(\frac{1}{1-r_{\mathrm{RFLP}}})}+1$ we should expect at least a phasing reduction of

$$
2^{F-M*e^{-c(1-\frac{T(r)}{L})}}
$$

So with this formula one can compute the effect of F.N. on the phasing reduction factor which our algorithm ensures.

Appendix A

mathematical definitions

Sets, Conditions, Set Algebras, and Probability Spaces

Sets are collections of objects. We denote set membership as $x \in X$ while $x \notin X$ denotes non–membership.

Y is said to be a subset of X if every element of Y is also an element of X, we denote the correspondence with $Y \subset X$. Sets are said to be equal if and only if $X \subset Y$ and $Y \subset X$.

Given sets X, Y the intersection is defined to be the collection of elements found in both sets, while the union is the collection of elements found in either of the sets. We denote the set intersection and union by $X \cap Y$ and $X \cup Y$.

We define a *condition* as a 2-tuple denoted $\langle C, X\rangle$ whose members are a domain set X, and a boolean function C, well defined for any element of the set X:

$$C : X \to \{0, 1\} : x \to C(x)$$

When reference to a condition is made by specifying only the boolean function, we assume the existence of an *ambient domain set.* The ambient set could be defined as the largest such set for which the function makes sense, or as the minimum set covering the scope of arguments needed.

For elements of the real line $\mathbb{R}$ we assume one component of the trichotomy property as a condition:

$$\langle C_>, \mathbb{R}\rangle$$

$$C_>(x) = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ otherwise} \end{cases}$$

Binary-Relations are conditions on 2–tuples. With the field properties of $\mathbb{R}$, in addition to the trichotomy property we construct a binary order relation:

$$\langle C_>, \mathbb{R} \times \mathbb{R}\rangle$$

$$C_>(x, y) = C_>(x - y) = \begin{cases} 1 \text{ if } x > y \\ 0 \text{ otherwise} \end{cases}$$

To select members of a set Y which satisfy a condition $\langle C, X\rangle$ we use the set inverse map:

$$C^{-1}(1) \cap Y = \{x \in Y : C(x) = 1\}$$

We use the notation $C(Y) = C^{-1}(1) \cap Y$ to indicate the elements of Y which satisfy the condition $\langle C, X\rangle$. If C is well defined on X it will be well defined on $Y \cap X$, usually $Y \subset X$.

With condition $\langle C, X\rangle$, we can define the complement condition, $\langle \neg C, X\rangle$:

$$\neg C : X \to \{0, 1\} : x \mapsto \begin{cases} 0 \text{ if } C(x) = 1 \\ 1 \text{ if } C(x) = 0 \end{cases} \tag{A.1}$$

With conditions $\langle C_1, X\rangle$, $\langle C_2, X\rangle$ we can define an intersection condition:

$$C_1 \wedge C_2 : X \to \{0,1\} : x \mapsto C_1(x) \times C_2(x) \tag{A.2}$$

for $Y \subset X$:

$$C_1 \wedge C_2(Y) = C_1(Y) \cap C_2(Y) \tag{A.3}$$

Example A.0.1 *We can define the condition of equality to zero for elements $x \in \mathbb{R}$ as an intersection condition:*

$$\langle C_=, \mathbb{R}\rangle$$
$$C_=(x) = \neg C_>(x,0) \wedge \neg C_>(0,x)$$

denote the condition $C_=(x)$ by $(x = 0)$.

De Morgan's law for sets states that:

$$\neg (W \cap Y) = (\neg W) \cup (\neg Y) \tag{A.4}$$

We use it to define the *inclusive disjunction* or the *OR* of conditions $\langle C_1, X\rangle, \langle C_2, X\rangle$ as:

$$C_1 \vee C_2 : X \to \{0,1\} : x \to \neg ((\neg C_1) \wedge (\neg C_2)) (x) \tag{A.5}$$

The operations of $\vee$ and $\wedge$ acting on sets have a *Lattice Structure.*

A *Lattice Structure* is a binary relation $\mathcal{R}$ on set X so that:

$$
\begin{aligned}
&x \in X \Rightarrow \mathcal{R}(x,x) = 1 \\
&x, y \in X : \mathcal{R}(x,y) = \mathcal{R}(y,x) = 1 \Rightarrow x = y \\
&x, y, z \in X : \mathcal{R}(x,y) = \mathcal{R}(y,z) = 1 \Rightarrow \mathcal{R}(x,z) = 1 \\
&\forall x, y \in X \exists z \in X : \mathcal{R}(z,x) = \mathcal{R}(z,y) = 1, \text{ and} \\
&\quad \forall w \in X : \mathcal{R}(w,x) = \mathcal{R}(w,y) = 1 \Rightarrow \mathcal{R}(w,z) = 1 \\
&\forall x, y \in X \exists z \in X : \mathcal{R}(x,z) = \mathcal{R}(y,z) = 1, \text{ and} \\
&\quad \forall w \in X : \mathcal{R}(x,w) = \mathcal{R}(y,w) = 1 \Rightarrow \mathcal{R}(z,w) = 1
\end{aligned}
$$

The *Power Set* of a set X is the collection of all subsets of X this is denoted:

$$2^X = \{A : A \subset X\} \tag{A.6}$$

Set Algebras, $\sigma-$Algebras, and Probability Space

Definition A.0.1 Set Algebra

An algebra of sets over X is a collection of subsets $\mathcal{A}$, $A \in \mathcal{A}$ means that $A \subset X$. A set algebra is any subclass of 2^X, which satisfy these closure conditions:

$$
\begin{aligned}
&\varnothing \in \mathcal{A} \\
&A \in \mathcal{A} \Rightarrow \neg A \in \mathcal{A} \\
&A, B \in \mathcal{A} \Rightarrow A \cap B \in \mathcal{A}
\end{aligned}
$$

A set algebra is a lattice structure with relation $\mathcal{R}(A,B) \Leftrightarrow A \subset B$ and with minimum element $\varnothing$ and maximal element X.

Definition A.0.2 σ**–Algebra**

A σ–algebra is an algebra of sets $\mathcal{A}$ with a countable closure condition:

$$A_i \in \mathcal{A} \text{ for } i \in \{1, 2, \ldots\} \Rightarrow \cap_{i \in \{1,2,\ldots\}} A_i \in \mathcal{A} \tag{A.7}$$

Definition A.0.3 Measure Space

A measure space is a 2–tuple $\langle X, \mathcal{A} \rangle$ with a set X and a σ–algebra $\mathcal{A}$ over X.

Definition A.0.4 Probability Space

Probability spaces are measure spaces $\langle X, \mathcal{A} \rangle$ with a positive, normalized, countably additive set function:

$$P : \mathcal{A} \to [0, 1] : A \mapsto P(A) \tag{A.8}$$

The set function is positive, normalized, and countably additive if the following hold:

$$\begin{aligned}
&P(\varnothing) = 0 \\
&P(\neg A) = 1 - P(A) \\
&A \cap B = \varnothing \Rightarrow P(A \cup B) = P(A) + P(B) \\
&A_i \in \mathcal{A} \quad \forall i \in [1, \ldots \infty) \text{ and } A_i \cap A_j = \varnothing \text{ when } i \neq j \\
&\quad \Rightarrow \sum_{i=1}^{\infty} P(A_i) = P(\cup_{i=1}^{\infty} A_i)
\end{aligned}$$

This last condition is called countable additivity of the set function.

Definition A.0.5 Borel σ**–algebra**

The Borel σ–algebra of sets on $\mathbb{R}$ is a class of sets contained in $2^{\mathbb{R}}$ having these properties:

- *It contains all of the open sets of $\mathbb{R}$.*

- *It is the smallest such σ–algebra to contain the open sets.*

We denote the Borel σ–algebra by: $\langle \mathbb{R}, \mathcal{B} \rangle$

Definition A.0.6 Measurable Map

Let $\langle X, \mathcal{A} \rangle$,$\langle Y, \mathcal{B} \rangle$ be two measure spaces, a measurable map on measure spaces is a mapping:

$$\Phi : \langle X, \mathcal{A} \rangle \to \langle Y, \mathcal{B} \rangle \tag{A.9}$$

So that the set inverse map (often called the pull–back) takes on values in $\mathcal{A}$:

$$\Phi^{-1}(B) \in \mathcal{A} \textit{ for every } B \in \mathcal{B} \tag{A.10}$$

Definition A.0.7 Random Variable

A random variable is a measurable map from a probability space onto the measure space $\langle \mathbb{R}, \mathcal{B} \rangle$

As such the properties of the probability space are translated into an *induced measure* on $\langle \mathbb{R}, \mathcal{B} \rangle$ continuous with respect to the Lebesgue measure. For many useful random variable there are well understood correspondences:

Useful Items in correspondence with a Random Variable
CDF function $F : \mathbb{R} \to [0,1] : \zeta \to P((-\infty, \zeta])$
PDF function $f : \mathbb{R} \to [0,1] : \zeta \to \frac{d}{d\zeta} P((-\infty, \zeta])$
Characteristic function $\chi : \mathbb{C} \to \mathbb{C} : \zeta \to \int_{\mathbb{R}} \exp{(ix\zeta)} f(x) dx$
Moment Sequences $\langle \frac{d^k}{d\zeta^k} \chi(0) \rangle_{k=0,1\ldots}$

A *probability distribution* of a random variable is the CDF function induced on $\langle \mathbb{R}, \mathcal{B} \rangle$ by the probability space $\langle X, \mathcal{A}, P \rangle$.

Of particular use to us will be the *Gaussian Distribution* . The Probability Space for the Gaussian Distribution with parameters μ, σ is given by:

$$\langle \mathbb{R}, \mathcal{B}, \lambda_{\mu,\sigma} \rangle \tag{A.11}$$

Where the set $\mathbb{R}$ is the real line, $\mathcal{B}$ is the Borel σ–algebra of sets, and $\lambda_{\mu,\sigma}$ is the set function:

$$\lambda_{\mu,\sigma} : \mathcal{A} \to [0,1] : A \to \int_A \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2} \right) dx \tag{A.12}$$

This probability set function will be referred to as the Gaussian Distribution with mean μ and variance σ. We will denote the CDF and the items in correspondence with the set function:

$$\mathcal{N}(\mu, \sigma) \tag{A.13}$$

Basic Sets and Spaces

Definition A.0.8 Linear Space over K

A linear space is a set X, a field K, and operations:

$$+ : X \times X \to X \tag{A.14}$$

$$\cdot : X \times K \to X \tag{A.15}$$

$$\tag{A.16}$$

satisfying these conditions for all $x, y, z \in X$ and $h, k \in K$:

$$x + (y + z) = (x + y) + z$$

$$x + y = y + x$$

$$\exists 0 \in X : x + 0 = x$$

$$\forall x \in X \ \exists - x \in X : x + -x = 0$$

$$h(\cdot k \cdot x) = (hk) \cdot x$$

$$h \cdot (x + y) = h \cdot x + h \cdot y$$

$$(h + k) \cdot x = h \cdot x + k \cdot x$$

Definition A.0.9 Linear Subspace

If X is a linear space then Y is a linear subspace of X, if it is closed under the operation of $+, \cdot$ in X, this follows if:

$$y_1, y_2 \in Y \Rightarrow y_1 + y_2 \in Y$$

$$y \in Y, k \in K \Rightarrow ky \in Y$$

We denote Y as a subspace of X by: $Y \leq X$.

Definition A.0.10 Norm, Normed linear space

A **norm** *is a function arising from a even gauge function, or a symmetric convex set* *containing zero, a function is a norm if it satisfies these properties for* $x, y \in X, k \in K$*:*

$$p(x) > 0 \text{ for all } x \neq 0$$

$$p(x) = 0 \Rightarrow x = 0$$

$$p(x + y) \leq p(x) + p(y)$$

$$p(kx) = |k| p(x)$$

The **p–norm** $p \geq 1$ *on the linear space* $\mathbb{R}^n$ *is the function:*

$$||\cdot||_p : \mathbb{R} \to \mathbb{R}^{\geq} : x \to \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}} \tag{A.17}$$

Definition A.0.11 Linear Transformation

If X *and* Y *are linear spaces, A linear transformation from* X *to* Y *is any map* L *which preserves linear structure:*

$$L : X \to Y : x \to L(x) \tag{A.18}$$

To preserve linear structure it is necessary that for $x_1, x_2 \in X, k \in K$*:*

$$L(x_1 + k \cdot x_2) = L(x_1) + kL(x_2) \tag{A.19}$$

Where the $+, \cdot$ *on the left–hand–side of the equation are the operations defined on* X *and the* $+, \cdot$ *on the right–hand–side of the equations are the operations defined on* Y*.*

A.1 Connection between number of orderings and degeneracy of containing space

Order and Degeneracy Now if $x \in L[i:j]$ and $y \in L[i:j]$ then we may order $x \leq_{ij} y$ if and only if $w_{xj} \geq w_{yj}$.

sub-claim

We claim that if $x \leq_{ij} y$ then $x \in L[yj]$. proof

On the index triplet x, y, i one of the relations must hold:

$$\text{case 1 } \mathcal{R}_{xyi} \text{ or } \mathcal{R}_{iyx} \tag{A.20}$$

$$\text{case 2 } \mathcal{R}_{yix} \text{ or } \mathcal{R}_{xiy} \tag{A.21}$$

$$\text{case 3 } \mathcal{R}_{ixy} \text{ or } \mathcal{R}_{yxi} \tag{A.22}$$

$$\tag{A.23}$$

We give proofs case by case.

Case 1 is done by definition, we have We may conclude that $x \in L[y : j]$.

For case 2 we have:

$$x \in L[i : j] \tag{A.24}$$

$$y \in L[i : j] \tag{A.25}$$

$$x \in L[i : y] \tag{A.26}$$

$$\tag{A.27}$$

producing equations:

$$w_{xi} + w_{ij} = w_{xj} \tag{A.28}$$

$$w_{yi} + w_{ij} = w_{yj} \tag{A.29}$$

$$w_{xi} + w_{iy} = w_{xy} \tag{A.30}$$

$$w_{xj} = w_{yj} + K \quad K \geq 0 \tag{A.31}$$

$$\tag{A.32}$$

By eliminating variable w_{ij} and replacing $w_{xj} - w_{yj}$ by K we get:

$$w_{xi} - K = w_{yi} \tag{A.33}$$

$$w_{xi} + w_{iy} = w_{xy} \tag{A.34}$$

$$\tag{A.35}$$

Now eliminating variable w_{yi} we get:

$$\begin{bmatrix} 2 & -1 & -1 \end{bmatrix} \begin{bmatrix} w_{xi} \\ K \\ w_{xy} \end{bmatrix} = 0 \tag{A.36}$$

We conclude that there is a choice of two free variables to determine the relations. We arrive at the conclusion that $w_{yi} = 0$ and $w_{xy} = w_{xi}$ and that $x \in L[y : j]$.

In case 3 we arrive at these equations:

$$w_{yi} + w_{ix} - w_{yx} = 0 \tag{A.37}$$

$$w_{yj} + w_{ji} - w_{yi} = 0 \tag{A.38}$$

$$w_{xi} + w_{ij} - w_{xj} = 0 \tag{A.39}$$

$$w_{xi} - w_{ji} \geq 0 \tag{A.40}$$

$$\tag{A.41}$$

Whose solution requires that $w_{xy} = 0$ and $w_{xi} = w_{yi}$. We may conclude that $x \in L[y : j]$. This shows that the ordering $\leq_{ij}$ is a linear ordering for the set $L[ij]$. This being the case let $m(ij)$ be the set of minimal elements in the chain $\langle L[ij], \leq_{ij} \rangle$. We will show that there is a minimal element m below every chain. Let $m(ij)$ and $m(km)$ be two minimal elements from chains $L[ij]$ and $L[km]$. For i, j, m one of the following cases must occur: case 1: $m \in L[ij]$ or case 2: $i \in L[mj]$ or case 3: $i \in L[jm]$ as one of the following six relations are given $\{\mathcal{R}_{mij} \vee \mathcal{R}_{jim}, \mathcal{R}_{imj} \vee \mathcal{R}_{jmi}, \mathcal{R}_{ijm} \vee \mathcal{R}_{mji}\}$. In case 1 we have $k \in L[ij]$ so $m(ij) \leq m(km)$. In case 2 we have $i, k \in L[mj]$. In case 3 we have $m(km) \leq m(ij)$ and we have a minimum element. Now let $\mathcal{L}[ij] = \{\zeta : \zeta \in L[a, j] : a \in L[ij]\}$

Π is not a subspace, but a rather elaborate set contained in $L(N)$. The set is not a subspace, not a cone, not convex, it is however a star–shaped set meaning that it is closed under the positive scalar multiplication.

Appendix B

Further Remarks on the equivalence of problems

B.1 Embedding Bounds

Our next result indicates that for any given $B \in \Upsilon$ and a given norm $||\cdot||_\Lambda$ there exists an interval containing solutions for all $b \in \mathbb{R}$.

Lemma B.1.1 *For $B \in \Upsilon$ and $||\cdot||_\Lambda$, there is a bounded interval $[0, D]$ for all embedding, independent of b. That is, there exists a $D \in \mathbb{R}^+$ so that for every $b \in \mathbb{R}^+$, if the instance of* MATRIX–TO–LINE *given by $\langle B, b, ||\cdot||_\Lambda \rangle$ has a solution then there exists a solution $\tilde{B} \in \Pi$ so that:*

$$\left|\left|B - \tilde{B}\right|\right|_\Lambda < b \tag{B.1}$$

and $\tilde{B}_{ij} < D$ for all i, j.

Proof 26 (Proof of lemma B.1.1) Let ρ be the ratio of the minimal and maximal eigenvalue of Λ, and let

$$D = (1+\epsilon)\,(N-1)\rho^{-1}||B||_{\Lambda} \tag{B.2}$$

for some negligible $\epsilon > 0$.

If $b > ||B||_{\Lambda}$ we may note that $b > ||B-0||_{\Lambda}$ so the solution of placing all points at the origin is possible. If a solution $\tilde{B}$ exists for $b < ||B||_{\Lambda}$ then we note that:

$$||B||_{\Lambda} > b > \left|\left|B - \tilde{B}\right|\right|_{\Lambda} > \rho|\left(B - \tilde{B}\right)_{ij}| \quad \forall i,j \tag{B.3}$$

Since $\tilde{B} \in \Pi$ we may assume that there is a permutation $\sigma \in S_N$ so that $\tilde{B} \in \Pi_{\sigma}$ by lemma( ), but we have:

$$\tilde{B}_{\sigma(i),\sigma(i+1)} < \rho^{-1}||B||_{\Lambda} \text{ by equation(B.3)} \tag{B.4}$$

Hence we have:

$$D > \tilde{B}_{\sigma(1),\sigma(N)} \tag{B.5}$$

concluding the proof. □

B.2 Annular bounds

Problem 4 MATRIX–TO–LINE–ANNULAR–BOUND

Given input $\langle B \in \Upsilon, b > a \in \mathbb{R}, ||\cdot||\rangle$ construct or find a matrix $A \in \Pi$ or determine that one doesn't exist so that:

$$a < ||B - A|| < b \tag{B.6}$$

Lemma B.2.1 MATRIX–TO–LINE–BOUND *and* MATRIX–TO–LINE–ANNULAR–BOUND *are Polynomial Equivalent*

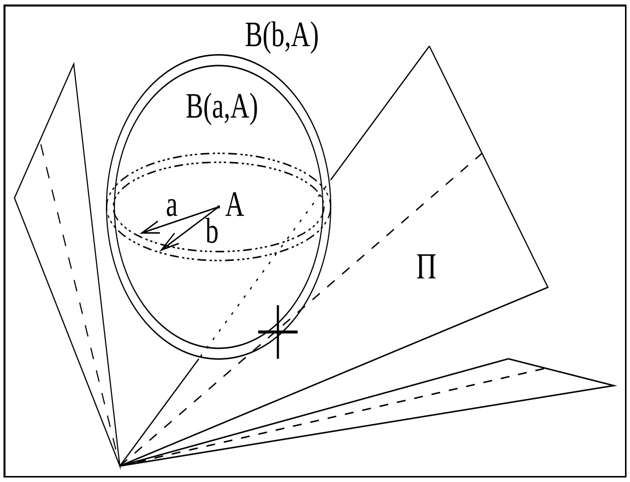

Figure B.1: Annular bound

Proof 27 (Proof of lemma B.2.1) one–way MATRIX–TO–LINE–ANNULAR–BOUND can be used to solve MATRIX–TO–LINE–BOUND:

Let $a = 0$ for the $a < b$ pair in the annular bound.

other–way MATRIX–TO–LINE–BOUND can be used to solve MATRIX–TO–LINE–ANNULAR–BOUND:

If a solution $\gamma \in \Pi_\sigma \leq \Upsilon$ for some $\sigma \in P_N$ can be found with the MATRIX–TO–LINE–BOUND some scalar multiple of γ will contact the boundary of the sphere. So given annular bounds $a < b$ solve MATRIX–TO–LINE–BOUND with bound b to find solution γ, now scale γ until it is found in the annular region. □

B.3 GRAPH-EMBEDDING–BOUND solves MATRIX–TO–LINE–BOUND

Proof 28 Given an instance of MATRIX–TO–LINE–BOUND given by $\langle B, ||\cdot||_{\Lambda,p}, b\rangle$ where Λ has spectral values in $\mathbb{Z}^+$, we shall construct an instance of GRAPH–EMBEDDING–BOUND to solve the instance of MATRIX–TO–LINE–BOUND. Let B be a $N \times N$ matrix, With out loss of generality we may assume that Λ is a diagonal matrix. For each value $\Lambda_{(i,j),(m,n)}$ we let $k(\Lambda;p)_{(i,j),(m,n)} = \min_{j\in\mathbb{Z}^+} j^p < \Lambda_{(i,j),(m,n)}$ We construct a vertex set V :

$$V_i : i \in [1 \ldots N] \tag{B.7}$$

$$V_{ijl} : i < j \in [1 \ldots N] : l \in [1 \ldots k(\Lambda;p)] \tag{B.8}$$

$$V_{jil} : i < j \in [1 \ldots N] : l \in [1 \ldots k(\Lambda;p)] \tag{B.9}$$

We construct an edge set E with weights W :

$$((V_i, V_{ijl}), 0) : i < j \in [1 \ldots N] : l \in [1 \ldots k(\Lambda;p)] \tag{B.10}$$

$$((V_j, V_{jil}), 0) : i < j \in [1 \ldots N] : l \in [1 \ldots k(\Lambda;p)] \tag{B.11}$$

$$((V_{ijl}, V_{jil}), B_{ij}) : i < j \in [1 \ldots N] : l \in [1 \ldots k(\Lambda;p)] \tag{B.12}$$

$$((V_i, V_{jil}), B_{ij}) : i < j \in [1 \ldots N] : l \in [1 \ldots k(\Lambda;p)] \tag{B.13}$$

$$((V_j, V_{ijl}), B_{ij}) : i < j \in [1 \ldots N] : l \in [1 \ldots k(\Lambda;p)] \tag{B.14}$$

$$\tag{B.15}$$

And we note that if we multiply Λ by a large integer M, we may approximate the p–norm slightly better. This sketches the main idea involved in showing the equivalence. For M sufficiently large and $\epsilon = \kappa b$ scaled accordingly we are actually emulating the MATRIX–TO–LINE instance with positive definite bilinear form Λ, p–norm. $\square$

B.4 Proof of GRAPH–EMBEDDING NP–Completeness with P–Norm

Definition B.4.1 ϵ, p**–Embedding**

A graph–embedding function $f : V \to \mathbb{R}$ *is an* ϵ, p*–Embedding if:*

$$\sum_{(u,v)\in E} ||f(u) - f(v)| - W((u,v))|^p < \epsilon^p \tag{B.16}$$

GRAPH–EMBEDDING–BOUND instances $\langle V, E, W, p, b\rangle$ There is a solutions if and only if there is a b, p–Embedding.

Lemma B.4.1 GRAPH–EMBEDDING–BOUND *Problem is NP–Complete*

Proof 29 (Proof of lemma B.4.1) We reduce 3–SAT to GRAPH–EMBEDDING–BOUND . An instance of 3–SAT is a set of clauses:

$$\wedge_{j=1:m} C_j \tag{B.17}$$

each clause is a disjunction of literals of the form given by:

$$C_j = (L_{j1} \vee L_{j2} \vee L_{j3}) \tag{B.18}$$

Each L_{jy} is equal to X_i or $\neg X_i$, for some $i \in [1 \dots n]$ one of the n boolean variables. Similar to Saxe [29], we construct a graph with vertex set:

$$\begin{aligned} V = \{&\{x_i, \neg x_i : i \in [1 \dots n]\}, \\ &A, B, \\ &\{\{C_{jv} : v \in [1 \dots 10]\}, z_j : j \in [1 \dots m]\}\} \end{aligned}$$

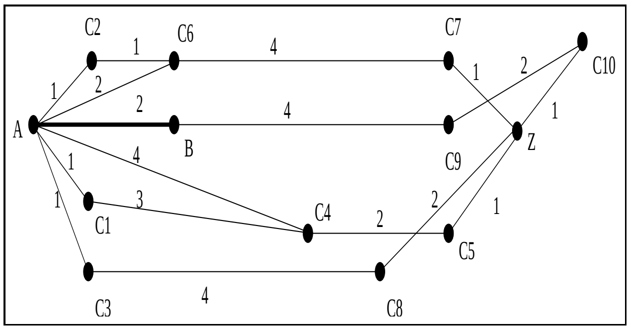

Figure B.2: Clause graph

And Edges with Weight:

$$E \times W = \{\{((x_i, \neg x_i), 2) : i \in [i \dots n]\} \tag{B.19}$$

$$((A, B), 2) \tag{B.20}$$

$$\{((A, x_i), 1), ((A, \neg x_i), 1) \;\; \forall i \in [1 \dots n]\} \tag{B.21}$$

$$\{((A, C_{j1}), 1), ((C_{j1}, C_{j4}), 3), ((C_{j4}, C_{j5}), 1), ((C_{j5}, z_j), 1)\} \tag{B.22}$$

$$\{((A, C_{j4}), 4)\} \tag{B.23}$$

$$\{((A, C_{j2}), 1), ((C_{j2}, C_{j6}), 1), ((C_{j6}, C_{j7}), 4), ((C_{j7}, z_j), 1)\} \tag{B.24}$$

$$\{((A, C_{j6}), 2)\} \tag{B.25}$$

$$\{((A, C_{j3}), 1), ((C_{j3}, C_{j8}), 4), ((C_{j8}, z_j), 2)\} \tag{B.26}$$

$$\{((B, C_{j9}), 4), ((C_{j9}, C_{j10}), 2), ((C_{j10}, z_j), 1)\} \tag{B.27}$$

$$\{((C_{jv}, L_{jv}), 0) \; j \in [1 \dots m] v \in [1, 2, 3]\}\} \tag{B.28}$$

$$\tag{B.29}$$

Below we draw a figure of the Clause Representation for reference: Further we name some paths and edges through the graph:

$$\alpha_1 = (A, C_4), (C_4, C_5), (C_5, z) \tag{B.30}$$

$$\alpha_2 = (A, C_6), (C_6, C_7), (C_7, z) \tag{B.31}$$

$$\alpha_3 = (A, C_3), (C_3, C_8), (C_8, z) \tag{B.32}$$

$$\alpha_4 = (A, B), (B, C_9), (C_9, C_10), (C_10, z) \tag{B.33}$$

Please notice that if point A is at zero and point B is at 2 then all points can be embedded exactly on the line for these configurations listed by z's position:

$$z = 7 : c_1 = 1, c_4 = 4, c_5 = 6 \tag{B.34}$$

$$: c_2 = 1, c_6 = 2, c_7 = 6 \tag{B.35}$$

$$: c_3 = 1, c_8 = 5 \tag{B.36}$$

$$: c_9 = 6, c_{10} = 8 \tag{B.37}$$

$$z = 5 : c_1 = 1, c_4 = 4, c_5 = 6 \tag{B.38}$$

$$: c_2 = 1, c_6 = 2, c_7 = 6 \tag{B.39}$$

$$: c_3 = -1, c_8 = 3 \tag{B.40}$$

$$: c_9 = 6, c_{10} = 4 \tag{B.41}$$

$$z = 3 : c_1 = 1, c_4 = 4, c_5 = 2 \tag{B.42}$$

$$: c_2 = -1, c_6 = -2, c_7 = 2 \tag{B.43}$$

$$: c_3 = 1, c_8 = 5 \tag{B.44}$$

$$: c_9 = 6, c_{10} = 4 \tag{B.45}$$

$$z = 1 : c_1 = 1, c_4 = 4, c_5 = 2 \tag{B.46}$$

$$: c_2 = -1, c_6 = -2, c_7 = 2 \tag{B.47}$$

$$: c_3 = -1, c_8 = 3 \tag{B.48}$$

$$: c_9 = -2, c_{10} = 0 \tag{B.49}$$

$$\tag{B.50}$$

$$z = -1 : c_1 = -1, c_4 = -4, c_5 = -2 \tag{B.51}$$

$$: c_2 = -1, c_6 = -2, c_7 = 2 \tag{B.52}$$

$$: c_3 = 1, c_8 = -3 \tag{B.53}$$

$$: c_9 = -2, c_{10} = 0 \tag{B.54}$$

$$z = -3 : c_1 = -1, c_4 = -4, c_5 = -2 \tag{B.55}$$

$$: c_2 = 1, c_6 = 2, c_7 = -2 \tag{B.56}$$

$$: c_3 = -1, c_8 = -5 \tag{B.57}$$

$$: c_9 = -2, c_{10} = -4 \tag{B.58}$$

$$z = -5 : c_1 = -1, c_4 = -4, c_5 = -6 \tag{B.59}$$

$$: c_2 = -1, c_6 = -2, c_7 = -6 \tag{B.60}$$

$$: c_3 = 1, c_8 = -3 \tag{B.61}$$

$$: c_9 = -2, c_{10} = -4 \tag{B.62}$$

$$\tag{B.63}$$

We note that if $A = 0$, $z = -7$ is the only other possible position for z incurring no cost on paths $\alpha_1, \alpha_2, \alpha_3$. However a positive cost must be paid if we are to place B in any location other than -2 in particular the embedding of such a clause requiring that $B = 2$ will have to be done with ϵ, p–embedding and cannot be done without incurring error greater than $\frac{1}{8}$. Here-forward we use the symbols C_{jl} as both the vertex and the position of vertex embedding on $\mathbb{R}$. After an embedding we note that the signs of $\langle C_1 - A, C_2 - A, C_3 - A \rangle$ encode the assignments of variables,

$$L_{jl} = \begin{cases} L_{jl} = 1 \text{ if } (C_{jl} - A) > 0 \\ L_{jl} = 0 \text{ otherwise} \end{cases} \tag{B.64}$$

We illustrate this encoding of values of literals with the position of z_j :

$$z_j = 7 : L_{j1} = 1, L_{j2} = 1, L_{j3} = 1 \tag{B.65}$$

$$z_j = 5 : L_{j1} = 1, L_{j2} = 1, L_{j3} = 0 \tag{B.66}$$

$$z_j = 3 : L_{j1} = 1, L_{j2} = 0, L_{j3} = 1 \tag{B.67}$$

$$z_j = 1 : L_{j1} = 1, L_{j2} = 0, L_{j3} = 0 \tag{B.68}$$

$$z_j = -1 : L_{j1} = 0, L_{j2} = 0, L_{j3} = 1 \tag{B.69}$$

$$z_j = -3 : L_{j1} = 0, L_{j2} = 1, L_{j3} = 0 \tag{B.70}$$

$$z_j = -5 : L_{j1} = 0, L_{j2} = 0, L_{j3} = 1 \tag{B.71}$$

Now we show a sub–claim.

sub–claim: If f is a ϵ, p–Embedding with $\epsilon < \frac{1}{8}$ of clause C_j. If $A = 0$ and $B = 2$ then the sign of $(C_{j1} - A), (C_{j2} - A), (C_{j3} - A)$ encode a satisfying assignment of literals L_{j1}, L_{j2}, L_{j3} by assignment:

$$L_{jl} = \begin{cases} 1 \text{ if } C_{jl} - A < 0 \\ 0 \text{ OW} \end{cases} \tag{B.72}$$

Let:

$$F(A, B, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, z; p) =$$

$$(|(|B - A|) - 2|)^p + (|(|C_1 - A|) - 1|)^p + (|(|C_2 - A|) - 1|)^p + (|(|C_3 - A|) - 1|)^p$$

$$+ (|(|C_4 - A|) - 4|)^p + (|(|C_6 - A|) - 2|)^p + (|(|C_4 - C_1|) - 3|)^p + (|(|C_6 - C_2|) - 1|)^p$$

$$+ (|(|C_7 - C_6|) - 4|)^p + (|(|z - C_7|) - 1|)^p + (|(|C_5 - C_4|) - 2|)^p + (|(|z - C_5|) - 1|)^p$$

$$+ (|(|C_8 - C_3|) - 4|)^p + (|(|z - C_8|) - 1|)^p + (|(|C_9 - B|) - 4|)^p + (|(|C_{10} - C_9|) - 2|)^p$$

$$+ (|(|z - C_{10}|) - 1|)^p$$

If $A = 0$ and $B = 2$ and $C_1 - A < 0$, $C_2 - A < 0$, $C_3 - A < 0$ then the function $F - \frac{1}{8}$ can be shown to be positive with a computational proof. Further we can show that if the vector $(A, B, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, z)$ is a ϵ, p–embedding then the only solution to $F(\cdot; p) = 0$ is the solution with the same signs for $(C_1 - A)$, $(C_2 - A)$, $(C_3 - A)$, when $[A : B]$ is taken to be the positive direction. Note that $\frac{1}{8}$ is chosen in connection with the longest cycle length 8 given by cycles $[A, C_2, C_6, C_7, z, C_5, C_4, C_1, A]$, $[A, B, C_9, C_{10}, z, C_5, C_4, C_1, A]$, and $[A, B, C_9, C_{10}, z, C_7, C_6, C_2, A]$. Now if two clauses C_{j_1} and C_{j_2} including the edges between clauses if any are ϵ, p–embedable then the result is that each must individually be ϵ, p–embedable, and thus if any literals are in common the clauses agree upon the placement for each, and thus the encoded assignment satisfies both clauses. So at this point one can reason that there is an ϵ, p–embedding with $\epsilon < \frac{1}{8}$ for the graph specified if and only if there is a satisfying assignment for the 3–SAT instance. $\square$

Chapter 4

Bibliography

[1] A. Aho, J. Hopcroft,and J. Ullman. **The Design and Analysis of Computer Algorithms**,
In *Addison–Wesley Series in Computer Science and Information Processing*, Addison–Wesley Publishing Company, Reading Massachusetts, 1974.

[2] T. AnantharamanD. Schwartz,and B. Mishra. "Genomics via Optical Mapping II: Ordered Restriction Maps," **Journal of Computational Biology, 4(2)**:91–118,1997

[3] F. Alizadeh,R.M. Karp,D.K. Weisser, and G. Zweig. "Physical Mapping of Chromosomes Using Unique Probes," **Journal of Computational Biology, 2(2)**:159–185, 1995.

[4] S. Asakawa,and N. Shimizu. "High-Fidelity Digital Hybridization Screening," **Genomics, 49(2)**:209–217, 1998.

[5] E. Barillot,J. Dausset, and D. Cohen. "Theoretical Analysis of a Physical Mapping Strategy Using Random Single–Copy Landmarks," **Journal of Computational Biology, 2(2)**:159–185, 1995.

[6] A. Ben–Dor,B. Chor. "On constructing radiation hybrid maps," **Proceedings of the First International Conference on Computational Molecular Biology,** 17–26, 1997

[7] W. Casey,B. Mishra,and M. Wigler. "Placing Probes on the Genome with Pairwise Distance Data," **Algorithms in Bioinformatics: first international workshop: proceedings WABO 2001,** 52–68 Springer,New York, 2001

[8] C. Cantor,and C. Smith. **Genomics: The Science and Technology Behind the Human Genome Project**,
John Wiley and Sons, New York, 1999.

[9] T. Christof,M. Junger,J. Kececioglu,P. Mutzel,and G. Reinelt. "A Branch–and–Cut Approach to Physical Mapping of Chromosomes by Unique End–Probes," **Journal of Computational Biology, 4(4)**:433–447, 1997.

[10] H. Chernoff. "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," **Annals of Mathematical Statistics, 23**:483–509,1952.

[11] A. Clark. "Inference of haplotypes from PCR-amplified samples of diploid populations," **Mol. Biol. Evol, 7**:111–122,1990.

[12] A. Clark,K. Weiss,and D. Nickerson. "Haplotype structure and population genetic inferences from nucleotide=sequence variation in human lipoprotein lipase," **Am. J. Human Genetics, 63**:595–612,1998.

[13] A.J. Cuticchia,J. Arnold,and W.E. Timberlake "ODS: Ordering DNA Sequences–A physical Mapping Algorithm Based on Simulated Annealing," **Comput Appl Biosci. 9(2)**: 215–219, 1993

[14] R. Drmanac,S. Drmanac,Z. Strezoska,T. Paunesku,I. Labat,M. Zeremski,J. Snoddy,W.K. Funkhouser, B. Koop,L Hood,et al. "DNA sequence determination by hybridization: a strategy for efficient large–scale sequencing," **Science, 163(5147)**:596, Feb 4, 1994.

[15] P.W. Goldberg,M.C. Golumbic,H. Kaplan,R. Shamir. "Four Strikes Against Physical Mapping of DNA," **Journal of Computational Biology, 2(1)**:139–152, 1995.

[16] D. Greenberg,and S. Istrail. “Physical mapping by STS hybridization: Algorithmic strategies and the challenge of software evaluation,” **Journal of Computational Biology, 2(2)**:219–273, 1995.

[17] A. Grigoriev,R. Mott,H. Lehrach. “An Algorithm to Detect Chimeric Clones and Random Noise in Genomic Mapping,” **Genomics, 22(2)**:482–486, 1995.

[18] D. Gusfield. “Inference of Haplotypes from Samples of Diploid Populations: Complexity and Algorithms,” **Journal of Computational Biology, 8-3**:305–323,2001.

[19] J. Håstad,L. Ivansson,J. Lagergren “Fitting Points on the Real Line and its Application to RH Mapping,” **Lecture Notes in Computer Science, 1461**:465–467, 1998.

[20] R.J. Lipshutz “Likelihood DNA Sequencing by Hybridization,” **Biomol Struct Dyn. 11(3)**:637–653, 1993.

[21] N. Lisitsyn,and M. Wigler “Cloning the differences between two complex genomes,” **Science, 258**:946–951,1993.

[22] R. Lucito, J. West, A. Reiner, J. Alexander, D. Esposito, B. Mishra, S. Powers, L. Norton,and M. Wigler “Detecting Gene Copy Number Fluctuations in Tumor Cells by Microarray Analysis of Genomic Representations,” **Genome Research, 10(11)**: 1726–1736, 2000.

[23] R. Lucito, M. Nakimura, J. West, Y. Han, K. Chin, K. Jenson, R. McCombie, M. Wigler “Genetic analysis using genomic representations,” **Proc. Natl. Acad. Sci. USA, 95**:4487–4492, 1998.

[24] M. Jain, and E.W. Myers. “Algorithms for Computing and Integrating Physical Maps Using Unique Probes,” **Journal of Computational Biology, 4(4)**:449–466, 1997.

[25] D.O. Nelson,T.P. Speed “Predicting progress in directed mapping projects,” **Genomics 24(1)**:41–52, 1994.

[26] M.J. Palazzolo,S.A. Sawyer,C.H. Martin,D.A. Smoller,and D.L. Hartl "Optimized Strategies for Sequence–Tagged–Site Selection in Genome Mapping," **Proc. Natl. Acad. Sci. USA, 88(18)**:8034–8038, 1991.

[27] S. Roweis,Z. Ghahramani "A Unifying Review of Linear Gaussian Models," **Neural Computation, 11(2)**:305–345, 1999

[28] D. Slonim,L. Kruglyak,L. Stein,E. Lander "Building human genome maps with radiation hybrids," **Journal of Computational Biology, 4(4)**:487–504, 1997.

[29] J.B. Saxe "Embeddability of weighted graphs in k-space is strongly NP-hard." **Proceedings of the 17th Allerton Conference in Communications, Control and Computing,** 480–489, 1979.

[30] R. E. Tarjan. **Data Structures and Network Algorithms**, CBMS 44
SIAM, Philadelphia, 1983.

[31] D.C. Torney. "Mapping Using Unique Sequences," **J Mol Biol, 217(2)**:259–264, 1991.