

# 3D genome reconstruction and numerical algebraic geometry

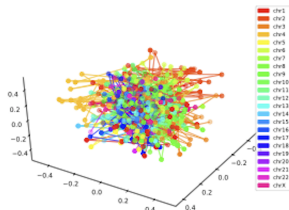
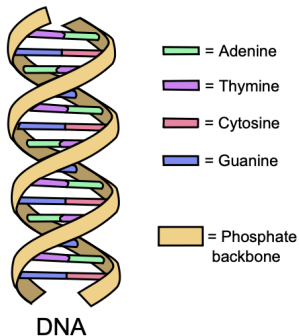
Kaie Kubjas



**Aalto University**

January 20, 2023

# The spatial organization of DNA



The spatial organization of the DNA in the cell nucleus plays an important role for gene regulation, DNA replication, and genomic integrity.

Figure 1: [https://commons.wikimedia.org/wiki/File:DNA\\_simple2.svg](https://commons.wikimedia.org/wiki/File:DNA_simple2.svg), Forluvoft, Public domain, via Wikimedia Commons

Figure 2: Belyaeva et al. Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry. *SIAM Journal on Mathematics of Data Science*, 2022. Copyright SIAM.

# Chromosome conformation capture techniques

- ▶ Chromosome conformation capture techniques measure the number of contacts between genomic loci over a population of cells [Lieberman-Aiden et al 2009].
- ▶ The results are recorded in a **HiC** or a **contact count matrix**.

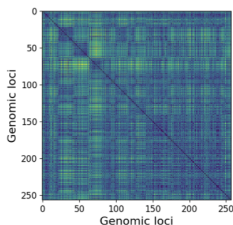
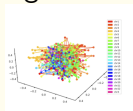


Figure: Belyaeva et al. Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry. *SIAM Journal on Mathematics of Data Science*, 2022. Copyright SIAM.

How to reconstruct from  
contact frequencies



the 3D organization of the  
genome



?

Figures: Belyaeva et al. Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry. SIAM Journal on Mathematics of Data Science, 2022. Copyright SIAM.



# Diploid organisms

Most eukaryotes including humans are **diploid organisms**, i.e. they carry two sets of chromosomes in a cell.

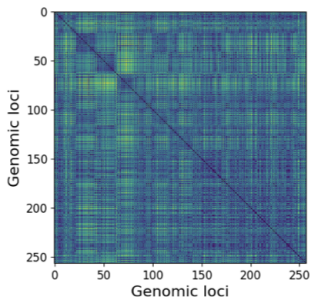
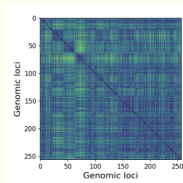


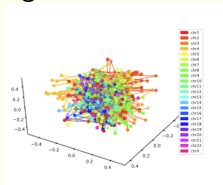
Figure: Belyaeva et al. Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry. *SIAM Journal on Mathematics of Data Science*, 2022. Copyright SIAM.

The contact frequency data is generally **unphased**, i.e. one cannot differentiate between different copies of a chromosome.

How to reconstruct from contact frequencies



the 3D organization of the genome



for diploid organisms?

Figures: Belyaeva et al. Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry. SIAM Journal on Mathematics of Data Science, 2022. Copyright SIAM.

# Partially phased data

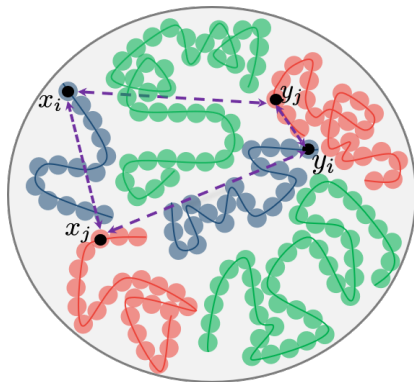
We assume that the data is **partially phased**, i.e., some of the contact counts can be associated with a homolog.

This is the case when SNPs can be used to assign a contact to a maternal or paternal homolog.



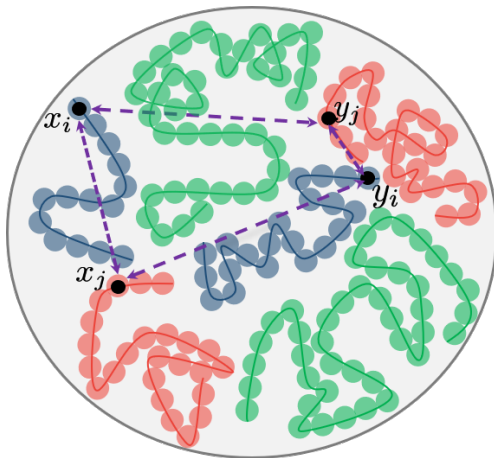
# Diploid organisms

The DNA is modeled as a string of beads consisting of two copies of each bead  $i$ , for  $1 \leq i \leq n$ . Denote the coordinates of the two copies of beads by  $x_i$  and  $y_i$ .



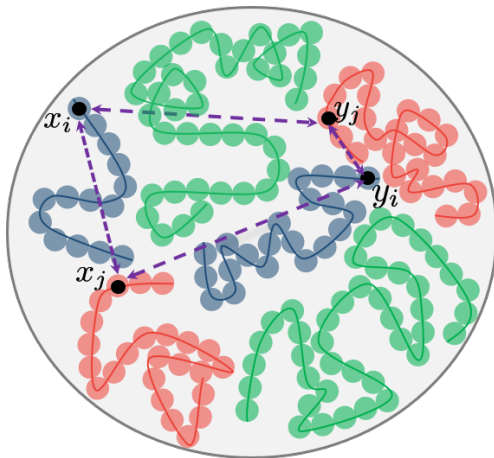
# Diploid organisms

If the contact count data is phased, then we know each of the four contact counts between two pairs of homologous loci  $x_i, y_i$  and  $x_j, y_j$ .



# Diploid organisms

If the contact count data is unphased, the measured contact counts between loci  $i$  and  $j$  correspond to the sum of four contact counts between two pairs of homologous loci  $x_i, y_i$  and  $x_j, y_j$ .



# Partially phased data

We will partition the bead pairs into **unambiguous** and **ambiguous** bead pairs.

# Partially phased data

We will partition the bead pairs into **unambiguous** and **ambiguous** bead pairs.

A locus is **unambiguous** if beads in the pair can be distinguished. The contact counts between two unambiguous loci are as in the phased case.

# Partially phased data

We will partition the bead pairs into **unambiguous** and **ambiguous** bead pairs.

A locus is **unambiguous** if beads in the pair can be distinguished. The contact counts between two unambiguous loci are as in the phased case.

A locus is **ambiguous** if beads in the pair cannot be distinguished. The contact counts between two ambiguous loci are as in the unphased case.

# Partially phased data

We will partition the bead pairs into **unambiguous** and **ambiguous** bead pairs.

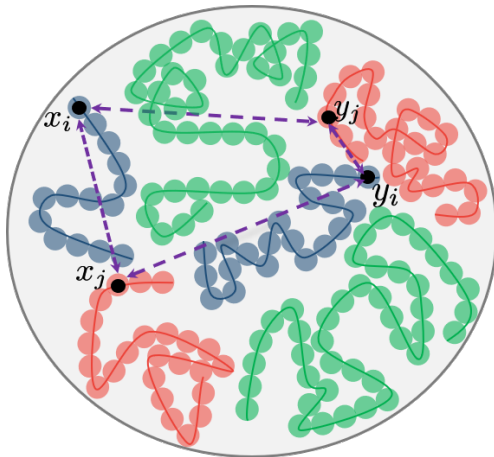
A locus is **unambiguous** if beads in the pair can be distinguished. The contact counts between two unambiguous loci are as in the phased case.

A locus is **ambiguous** if beads in the pair cannot be distinguished. The contact counts between two ambiguous loci are as in the unphased case.

What about contact counts between an ambiguous and an unambiguous locus?

# Partially phased data

If the locus  $i$  is unambiguous and  $j$  is ambiguous, then we know the sum of two contact counts between  $x_i$  and  $x_j, y_j$ , and similarly between  $y_i$  and  $x_j, y_j$ .





# Contact counts

The **unambiguous count matrix**  $C^U$  is a  $2n \times 2n$  matrix with the first  $n$  indices corresponding to  $x_1, \dots, x_n$  and the last  $n$  indices corresponding to  $y_1, \dots, y_n$ .

# Contact counts

The **unambiguous count matrix**  $C^U$  is a  $2n \times 2n$  matrix with the first  $n$  indices corresponding to  $x_1, \dots, x_n$  and the last  $n$  indices corresponding to  $y_1, \dots, y_n$ .

The **ambiguous count matrix**  $C^A$  is an  $n \times n$  matrix and we assume that each ambiguous count is the sum of four unambiguous counts:

$$c_{i,j}^A = c_{i,j}^U + c_{i,j+n}^U + c_{i+n,j}^U + c_{i+n,j+n}^U.$$

# Contact counts

The **unambiguous count matrix**  $C^U$  is a  $2n \times 2n$  matrix with the first  $n$  indices corresponding to  $x_1, \dots, x_n$  and the last  $n$  indices corresponding to  $y_1, \dots, y_n$ .

The **ambiguous count matrix**  $C^A$  is an  $n \times n$  matrix and we assume that each ambiguous count is the sum of four unambiguous counts:

$$c_{i,j}^A = c_{i,j}^U + c_{i,j+n}^U + c_{i+n,j}^U + c_{i+n,j+n}^U.$$

The **partially ambiguous count matrix**  $C^P$  is a  $2n \times n$  matrix and each partially ambiguous count is the sum of two unambiguous counts:

$$c_{i,j}^P = c_{i,j}^U + c_{i,j+n}^U.$$

# From contacts to distances

When convenient, we use the notation

$z_1 := x_1, \dots, z_n := x_n, z_{n+1} := y_1, \dots, z_{2n} := y_n$ . In this notation,

$$Z = [z_1, \dots, z_n, z_{n+1}, \dots, z_{2n}]^T \in \mathbb{R}^{2n \times 3}.$$

# From contacts to distances

When convenient, we use the notation

$z_1 := x_1, \dots, z_n := x_n, z_{n+1} := y_1, \dots, z_{2n} := y_n$ . In this notation,

$$Z = [z_1, \dots, z_n, z_{n+1}, \dots, z_{2n}]^T \in \mathbb{R}^{2n \times 3}.$$

Denoting the distance  $\|z_i - z_j\|$  between  $z_i$  and  $z_j$  by  $d_{i,j}$ , the [power law dependency](#) observed by Lieberman-Aiden et al.<sup>1</sup> can be written as

$$c_{i,j}^U = \gamma d_{i,j}^\alpha,$$

where  $\alpha < 0$  is a conversion factor and  $\gamma > 0$  is a scaling factor.

---

<sup>1</sup>Lieberman-Aiden, Erez, et al. Science, 2009. 

# From contacts to distances

When convenient, we use the notation

$z_1 := x_1, \dots, z_n := x_n, z_{n+1} := y_1, \dots, z_{2n} := y_n$ . In this notation,

$$Z = [z_1, \dots, z_n, z_{n+1}, \dots, z_{2n}]^T \in \mathbb{R}^{2n \times 3}.$$


Denoting the distance  $\|z_i - z_j\|$  between  $z_i$  and  $z_j$  by  $d_{i,j}$ , the **power law dependency** observed by Lieberman-Aiden et al.<sup>1</sup> can be written as

$$c_{i,j}^U = \gamma d_{i,j}^\alpha,$$

where  $\alpha < 0$  is a conversion factor and  $\gamma > 0$  is a scaling factor.

We set  $\gamma = 1$  and sometimes  $\alpha = -2$ . In general, the conversion factor  $\alpha$  depends on a dataset.

---

<sup>1</sup>Lieberman-Aiden, Erez, et al. Science, 2009. 

# From contacts to distances

We convert the empirical contact counts to Euclidean distances and then aim to reconstruct the positions of beads from the distances. This leads us to the following system of equations:

$$\begin{cases} c_{i,j}^A = \|x_i - x_j\|^\alpha + \|x_i - y_j\|^\alpha + \|y_i - x_j\|^\alpha + \|y_i - y_j\|^\alpha & \forall i, j \in A \\ c_{i,j}^P = \|x_i - x_j\|^\alpha + \|x_i - y_j\|^\alpha, \quad c_{i+n,j}^P = \|y_i - x_j\|^\alpha + \|y_i - y_j\|^\alpha & \forall i \in U, j \in A \\ c_{i,j}^U = \|x_i - x_j\|^\alpha, \quad c_{i,j+n}^U = \|x_i - y_j\|^\alpha, \\ c_{i+n,j}^U = \|y_i - x_j\|^\alpha, \quad c_{i+n,j+n}^U = \|y_i - y_j\|^\alpha & \forall i, j \in U \end{cases}$$

# Identifiability



# Unambiguous setting and Euclidean distance geometry

- ▶ If all pairs are unambiguous, i.e.,  $U = [n]$ , then constructing the original points translates to a classical problem in Euclidean distance geometry.

# Unambiguous setting and Euclidean distance geometry

- ▶ If all pairs are unambiguous, i.e.,  $U = [n]$ , then constructing the original points translates to a classical problem in Euclidean distance geometry.
- ▶ First we convert contacts to distances and then use tools from the first lecture to find the positions of beads from pairwise distances.

# Partially ambiguous setting

We denote the true but unknown coordinates by  $x^*$  and the symbol  $x$  stands for a variable that we want to solve for. We write  $\|\cdot\|$  for the standard inner product on  $\mathbb{R}^3$ .

**Theorem (Ciefuentes, Draisma, Henriksson, Korchmaros, K.)**

*Let  $\alpha$  be a negative rational number. Then for  $a^*, b^*, \dots, f^*, x^*, y^* \in \mathbb{R}^3$  sufficiently general, the system of six equations*

$$\|x - t^*\|^\alpha + \|y - t^*\|^\alpha = \|x^* - t^*\|^\alpha + \|y^* - t^*\|^\alpha \text{ for } t^* = a^*, b^*, \dots, f^*$$

*in the six unknowns  $x_1, x_2, x_3, y_1, y_2, y_3 \in \mathbb{R}$  has only finitely many solutions.*

# Partially ambiguous setting

Conjecture (Cieuentes, Draisma, Henriksson, Korchmaros, K.)

Let  $a^*, b^*, c^*, d^*, e^*, f^*, g^*, x^*, y^* \in \mathbb{R}^3$  be sufficiently general.  
The system of rational equations

$$\frac{1}{\|t^* - x^*\|^2} + \frac{1}{\|t^* - y^*\|^2} = \frac{1}{\|t^* - x\|^2} + \frac{1}{\|t^* - y\|^2} \text{ for } t^* = a^*, \dots, g^*$$

has precisely two solutions  $(x^*, y^*)$  and  $(y^*, x^*)$ .

# Partially ambiguous setting

Corollary (Ciefuentes, Draisma, Henriksson, Korchmaros, K.)

*Let  $\alpha$  be a negative rational number. Then for  $a^*, b^*, \dots, f^*, x^*, y^* \in \mathbb{R}^3$  and  $\epsilon_a, \epsilon_b, \dots, \epsilon_f \in \mathbb{R}$  sufficiently general, the system of six equations*

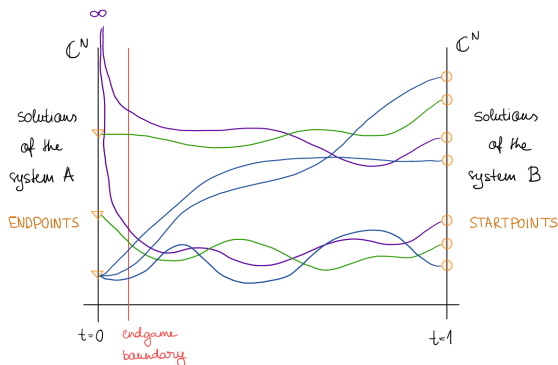
$$\|x - t^*\|^\alpha + \|y - t^*\|^\alpha = \|x^* - t^*\|^\alpha + \|y^* - t^*\|^\alpha + \epsilon_{t^*} \text{ for } t^* = a^*, b^*, \dots, f^*$$

*in the six unknowns  $x_1, x_2, x_3, y_1, y_2, y_3 \in \mathbb{R}$  has only finitely many solutions.*

# Numerical algebraic geometry

# Overview

- ▶ Main goal: To solve a system of equations  $A$ .
- ▶ Take a similar system of equations  $B$  for which solutions are known.
- ▶ Deform the solutions of  $B$  to the solutions of  $A$ .
- ▶ This approach is called **homotopy continuation**.



# Definitions

- ▶ A system of polynomial equations is called **square** if the number of equations is equal to the number of variables, i.e., the system has the form

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0.$$

- ▶ We will first consider square systems and later explain how the results can be extended to general systems.



# Definitions

- ▶ A system of polynomial equations is called **square** if the number of equations is equal to the number of variables, i.e., the system has the form

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0.$$

- ▶ We will first consider square systems and later explain how the results can be extended to general systems.
- ▶ A solution  $z^* \in \mathbb{C}^N$  is called **isolated** if it is the only solution in an open ball centered at  $z^*$ .

# Homotopy

## Definition

Given two continuous functions  $f, g : \mathbb{C}^N \rightarrow \mathbb{C}^N$ , a **homotopy** is a continuous function

$$H(z, t) : \mathbb{C}^N \times [0, 1] \rightarrow \mathbb{C}^N$$

satisfying  $H(z, 0) = f(z)$  and  $H(z, 1) = g(z)$ .

# Homotopy

## Definition

Given two continuous functions  $f, g : \mathbb{C}^N \rightarrow \mathbb{C}^N$ , a **homotopy** is a continuous function

$$H(z, t) : \mathbb{C}^N \times [0, 1] \rightarrow \mathbb{C}^N$$

satisfying  $H(z, 0) = f(z)$  and  $H(z, 1) = g(z)$ .

## Example

The easiest homotopy to consider is

$$H(z, t) = tg(z) + (1 - t)f(z).$$

# Intuition

Consider a square system

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0.$$

We want to find a **finite set  $\mathcal{S}$  of solutions** of this system **containing every isolated solution** of  $f(z) = 0$ .

# Intuition

Consider a square system

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0.$$

We want to find a **finite set  $\mathcal{S}$  of solutions** of this system **containing every isolated solution** of  $f(z) = 0$ .

1. Build and solve a **start system  $g(z)$** .
  - ▶  $g(z)$  is related to  $f(z)$ : it usually has the same degrees
  - ▶ It should be easy to solve  $g(z)$
  - ▶ The solutions of  $g(z)$  are called the **startpoints**

Consider a square system

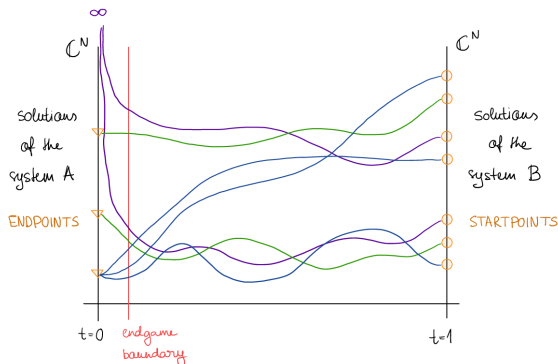
$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0.$$

We want to find a **finite set  $\mathcal{S}$  of solutions** of this system **containing every isolated solution** of  $f(z) = 0$ .

1. Build and solve a **start system**  $g(z)$ .
  - ▶  $g(z)$  is related to  $f(z)$ : it usually has the same degrees
  - ▶ It should be easy to solve  $g(z)$
  - ▶ The solutions of  $g(z)$  are called the **startpoints**
2. Construct a **homotopy** between  $f(z)$  and  $g(z)$ .
  - ▶ The homotopy  $H(z, t)$  gives a **parametrized family of equations** that specializes to  $f(z)$  and  $g(z)$  for different parameter values

## 3 Follow the solution paths from $t = 1$ to $t = 0$ .

- ▶ **Predictor-corrector methods** are used most of the way
- ▶ Close to  $t = 0$  more powerful **endgames** are used
- ▶ Some paths could approach infinity as  $t \rightarrow 0$ ; these paths are called **divergent**
- ▶ Other paths can merge at  $t = 0$



# Example

We want to solve  $f(z) = 0$  for the polynomial

$$f(z) = -2z^3 - 5z^2 + 4z + 1.$$

This particular example can be solved by the [cubic formula](#). We consider it to illustrate the steps of the homotopy continuation.



# Example

We want to solve  $f(z) = 0$  for the polynomial

$$f(z) = -2z^3 - 5z^2 + 4z + 1.$$

This particular example can be solved by the [cubic formula](#). We consider it to illustrate the steps of the homotopy continuation.

## 1. Start system

- ▶ Any cubic polynomial with three distinct roots that can be solved easily.
- ▶ We take  $g(z) = z^3 + 1$ .
- ▶ The roots of  $g(z)$  are  $z = -e^{2k\pi i/3}$ , where  $k = 0, 1, 2, 3$ .

# Example

We want to solve  $f(z) = 0$  for the polynomial

$$f(z) = -2z^3 - 5z^2 + 4z + 1.$$

This particular example can be solved by the [cubic formula](#). We consider it to illustrate the steps of the homotopy continuation.

## 1. Start system

- ▶ Any cubic polynomial with three distinct roots that can be solved easily.
- ▶ We take  $g(z) = z^3 + 1$ .
- ▶ The roots of  $g(z)$  are  $z = -e^{2k\pi i/3}$ , where  $k = 0, 1, 2, 3$ .

## 2. Homotopy

- ▶ We choose linear homotopy  $h(z, s) = sg(z) + (1 - s)f(z)$ .
- ▶  $h(z, 1) = g(z)$  and  $h(z, 0) = f(z)$

# Example

## 3 Follow the solution paths

- ▶ The variable  $s$  is complex, so there are infinitely many paths from 1 to 0.
- ▶ Although the real line segment  $[0, 1]$  seems like a natural choice, it can be problematic.
- ▶ Instead consider the following family of **circular arcs**: Let  $\gamma \in \mathbb{C} \setminus \mathbb{R}$ . Then

$$q(t) = \frac{\gamma t}{\gamma t + (1-t)}, \quad t \in [0, 1]$$

connects  $s = 1$  to  $s = 0$ .

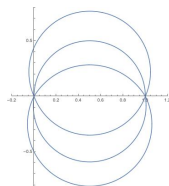
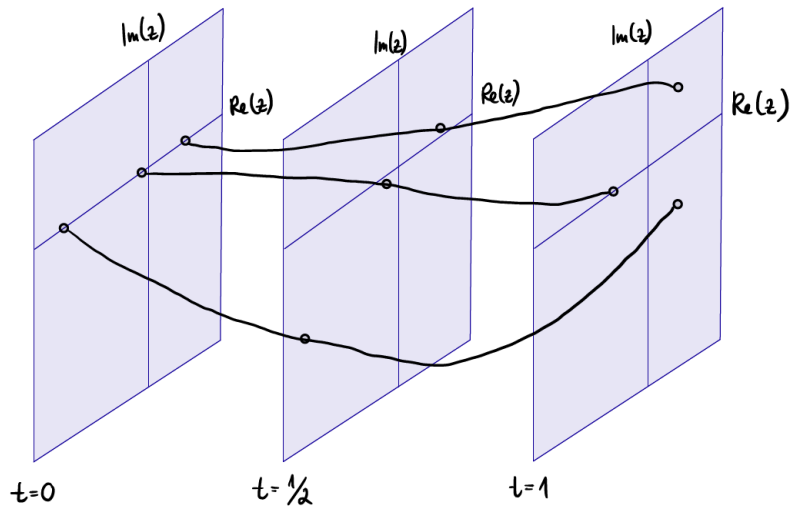


Figure: Plots are for six different values of  $\gamma$ .



# Example



# Choice of $\gamma$

- ▶ If  $\gamma$  is chosen uniformly randomly in  $\mathbb{C}$ , then with probability one the homotopy defines three smooth paths.
- ▶ To see this, we consider the behavior of  $h(z, s) = 0$  as  $s$  varies.
- ▶ For most  $s^* \in \mathbb{C}$ ,  $h(z, s^*) = 0$  is a cubic equation with three distinct roots.
- ▶ For a few  $s^*$  there are only two distinct solutions.
- ▶ The use of circular arcs to obtain a path between  $s = 1$  and  $s = 0$  and choosing  $\gamma$  randomly is known as the “gamma trick”.

# NumericalAlgebraicGeometry in Macaulay2

```
NAG-example.m2

needsPackage("NumericalAlgebraicGeometry")
R = CC[z];
F = {-2*z^3-5*z^2+4*z+1};
s = solveSystem F
realPoints s

U:-- NAG-example.m2 All L5 (Macaulay2)

+ M2 --no-readline --print-width 140
Macaulay2, version 1.14
--loading configuration for package "FourTiTwo" from file /Users/kubjaski/Library/Application Support/Macaulay2/init-FourTiTwo.m2
--loading configuration for package "Topcom" from file /Users/kubjaski/Library/Application Support/Macaulay2/init-Topcom.m2
with packages: ConwayPolynomials, Elimination, IntegralClosure, InverseSystems, LLLBases, PrimaryDecomposition, ReesAlgebra, TangentCone,
Truncations

i1 : needsPackage("NumericalAlgebraicGeometry")
--loading configuration for package "NumericalAlgebraicGeometry" from file /Users/kubjaski/Library/Application Support/Macaulay2/init-NumericalAlgebraicGeometry.m2
--loading configuration for package "PHCpack" from file /Users/kubjaski/Library/Application Support/Macaulay2/init-PHCpack.m2
--loading configuration for package "Bertini" from file /Users/kubjaski/Library/Application Support/Macaulay2/init-Bertini.m2

o1 = NumericalAlgebraicGeometry
o1 : Package

i2 : R = CC[z];
i3 : F = {-2*z^3-5*z^2+4*z+1};
i4 : s = solveSystem F
o4 = {{-3.09415}, {.796927}, {-202773}}
o4 : List

i5 : realPoints s
o5 = {{-3.09415}, {.796927}, {-202773}}
o5 : List

i6 : []
```

U:-- +M2\* All L34 (Macaulay2 Interaction:run)  
Beginning of buffer

## Definition

**Path tracking** is the numerical process of approximating the paths from startpoints to endpoints.

Path tracking gives approximations of the solutions of  $H(z, 0) = 0$  from the known solutions of  $H(z, 1) = 0$ .



# Definition of good homotopy

A **good homotopy** for a system  $f(z) = 0$  and a set of  $D$  distinct solutions  $S_1$  of  $g(z) = 0$  is a system of infinitely differentiable functions  $H(z, t) = (H_1(z, t), \dots, H_N(z, t)) = 0$  such that

1. for any  $t \in [0, 1]$ ,  $H(z, t)$  is a system of polynomials;
2. for any  $w \in S_1$ , there is a smooth map  $p_j(t) : (0, 1] \rightarrow \mathbb{C}^N$  satisfying  $p_j(1) = w$ ;
3. the associated paths do not cross;
4. for each  $t^* \in (0, 1]$  the points  $p_j(t^*)$  are smooth isolated solutions of  $H(z, t^*)$ .
5. The set

$$S_0 = \left\{ z \in \mathbb{C}^N \mid \|z\|_2 < \infty \text{ and } z = \lim_{t \rightarrow 0} p_j(t) \right\}$$

contains every isolated solution of  $f(z) = 0$ .

# Bezout's theorem

## Theorem (Bezout's theorem)

Assume that the system of polynomial equations

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0.$$

has finitely many solutions in  $\mathbb{C}^N$ . Let  $d_i = \deg f_i$ . Then the system  $f$  has *at most*  $d_1 \cdots d_N$  solutions.

# Bezout's theorem

## Theorem (Bezout's theorem)

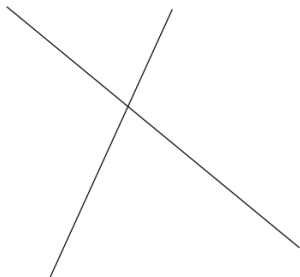
Assume that the system of polynomial equations

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0.$$

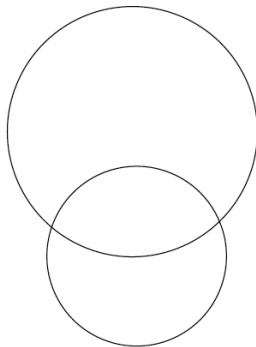
has finitely many solutions in  $\mathbb{C}^N$ . Let  $d_i = \deg f_i$ . Then the system  $f$  has *at most*  $d_1 \cdots d_N$  solutions.

- ▶ For general systems of polynomial equations the number of solutions equals this bound.
- ▶ The [Bernstein–Kushnirenko Theorem](#) gives better upper bounds for special systems, but it is more complicated.

# Bezout's theorem



$$d_1 = d_2 = 1$$
$$d_1 \cdot d_2 = 1$$
$$\# \text{ solutions} = 1$$



$$d_1 = d_2 = 2$$
$$d_1 \cdot d_2 = 4$$
$$\# \text{ solutions} = 2$$

# Total-degree homotopies

We construct a good homotopy

$$H(z, t) = (1 - t) \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} + \gamma t \begin{bmatrix} g_1(z_1, \dots, z_N) \\ \vdots \\ g_N(z_1, \dots, z_N) \end{bmatrix} = 0$$

as follows:

# Total-degree homotopies

We construct a good homotopy

$$H(z, t) = (1 - t) \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} + \gamma t \begin{bmatrix} g_1(z_1, \dots, z_N) \\ \vdots \\ g_N(z_1, \dots, z_N) \end{bmatrix} = 0$$

as follows:

- ▶ Let  $d_i = \deg f_i$ .
- ▶ Choose polynomials  $g_1, \dots, g_N$  such that they have **degrees**  $d_1, \dots, d_N$ , the system  $g(z) = 0$  is **easy to solve** and it **has exactly**  $D := d_1 d_2 \cdots d_N$  **solutions**.

# Total-degree homotopies

We construct a good homotopy

$$H(z, t) = (1 - t) \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} + \gamma t \begin{bmatrix} g_1(z_1, \dots, z_N) \\ \vdots \\ g_N(z_1, \dots, z_N) \end{bmatrix} = 0$$

as follows:

- ▶ Let  $d_i = \deg f_i$ .
- ▶ Choose polynomials  $g_1, \dots, g_N$  such that they have **degrees**  $d_1, \dots, d_N$ , the system  $g(z) = 0$  is **easy to solve** and it **has exactly  $D := d_1 d_2 \cdots d_N$  solutions**.
- ▶ For example, one can take  $g_i(z) = z_i^{d_i} - 1$ .
- ▶ In this case, the solution set of  $g(z) = 0$  is given by

$$\left\{ \left( e^{(j_1/d_1)2\pi i}, \dots, e^{(j_N/d_N)2\pi i} \right) : 0 \leq j_i \leq d_i \text{ for } i = 1, \dots, N \right\}.$$

# Total-degree homotopies

- ▶ Choose a random complex number  $\gamma \neq 0$ .
- ▶ In practice  $\gamma$  is chosen in a small band around the unit circle.
- ▶ If  $\gamma$  is chosen uniformly randomly, then with probability one we get a good homotopy.
- ▶ Total-degree homotopies are the simplest of all homotopies. Alternatively, one can use more special degree bounds.



# Path tracking

Assume that we have:

- ▶ a family of functions on  $\mathbb{C}^N$

$$H(z; q) = \begin{bmatrix} H_1(z_1, \dots, z_N; q_1 \dots, q_M) \\ \vdots \\ H_N(z_1, \dots, z_N; q_1 \dots, q_M) \end{bmatrix} = 0$$

such that  $H_i$  is a polynomial in  $z \in \mathbb{C}^N$  and analytic in  $q \in \mathbb{C}^M$ ;

# Path tracking

Assume that we have:

- ▶ a family of functions on  $\mathbb{C}^N$

$$H(z; q) = \begin{bmatrix} H_1(z_1, \dots, z_N; q_1, \dots, q_M) \\ \vdots \\ H_N(z_1, \dots, z_N; q_1, \dots, q_M) \end{bmatrix} = 0$$

such that  $H_i$  is a polynomial in  $z \in \mathbb{C}^N$  and analytic in  $q \in \mathbb{C}^M$ ;

- ▶ differentiable maps  $\phi : t \in [0, 1] \rightarrow q \in \mathbb{C}^M$  and  $\psi : t \in [0, 1] \rightarrow z \in \mathbb{C}^N$  satisfying
  1.  $H(\psi(t), \phi(t)) = 0$  for  $t \in (0, 1]$  and
  2. the Jacobian of  $H$  with respect to  $z_1, \dots, z_N$  has rank  $N$  for the points  $(\psi(t), \phi(t))$  with  $t \in (0, 1]$ .

# Path tracking

Assume that we have:

- ▶ a family of functions on  $\mathbb{C}^N$

$$H(z; q) = \begin{bmatrix} H_1(z_1, \dots, z_N; q_1, \dots, q_M) \\ \vdots \\ H_N(z_1, \dots, z_N; q_1, \dots, q_M) \end{bmatrix} = 0$$

such that  $H_i$  is a polynomial in  $z \in \mathbb{C}^N$  and analytic in  $q \in \mathbb{C}^M$ ;

- ▶ differentiable maps  $\phi : t \in [0, 1] \rightarrow q \in \mathbb{C}^M$  and  $\psi : t \in [0, 1] \rightarrow z \in \mathbb{C}^N$  satisfying
  1.  $H(\psi(t), \phi(t)) = 0$  for  $t \in (0, 1]$  and
  2. the Jacobian of  $H$  with respect to  $z_1, \dots, z_N$  has rank  $N$  for the points  $(\psi(t), \phi(t))$  with  $t \in (0, 1]$ .
- ▶ We construct  $H$  and  $\phi$  in such a way that  $\psi$  exists and  $\psi(1) = p_0$ . The objective is to compute  $p^* = \psi(0)$ .

# Path tracking

- ▶ Assume that  $M = 1$  and  $q_1 = t$ . Denote  $\psi(t)$  by  $z(t)$ .

# Path tracking

- ▶ Assume that  $M = 1$  and  $q_1 = t$ . Denote  $\psi(t)$  by  $z(t)$ .
- ▶ Differentiating  $H(z(t), t) = 0$  with respect to  $t$  gives

$$\frac{\partial H(z(t), t)}{\partial t} + \sum_{i=1}^N \frac{\partial H(z(t), t)}{\partial z_i} \frac{dz_i(t)}{dt} = 0 \text{ with } z(1) = p_0.$$

# Path tracking

- ▶ Assume that  $M = 1$  and  $q_1 = t$ . Denote  $\psi(t)$  by  $z(t)$ .
- ▶ Differentiating  $H(z(t), t) = 0$  with respect to  $t$  gives

$$\frac{\partial H(z(t), t)}{\partial t} + \sum_{i=1}^N \frac{\partial H(z(t), t)}{\partial z_i} \frac{dz_i(t)}{dt} = 0 \text{ with } z(1) = p_0.$$

- ▶ Let  $JH(z, t)$  denote the **Jacobian matrix** of  $H$  with respect to the variables  $z$

$$JH := \frac{\partial H}{\partial z} = \begin{bmatrix} \frac{\partial H_1}{\partial z_1} & \dots & \frac{\partial H_1}{\partial z_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial H_N}{\partial z_1} & \dots & \frac{\partial H_N}{\partial z_N} \end{bmatrix}$$

evaluated at  $(z, t)$  and let  $z(t) = [z_1(t), \dots, z_N(t)]^T$  denote the solution of the above differential equation.

- ▶ Using this notation, the above differential equation becomes

$$\frac{\partial H(z(t), t)}{\partial t} + JH(z(t), t) \cdot \frac{dz(t)}{dt} = 0.$$

- ▶ Using this notation, the above differential equation becomes

$$\frac{\partial H(z(t), t)}{\partial t} + JH(z(t), t) \cdot \frac{dz(t)}{dt} = 0.$$

- ▶ Since  $JH(z(t), t)$  is invertible on the path, this is equivalent to

$$\frac{dz(t)}{dt} = -[JH(z(t), t)]^{-1} \frac{\partial H(z(t), t)}{\partial t}.$$



# Path tracking

- ▶ Using this notation, the above differential equation becomes

$$\frac{\partial H(z(t), t)}{\partial t} + JH(z(t), t) \cdot \frac{dz(t)}{dt} = 0.$$

- ▶ Since  $JH(z(t), t)$  is invertible on the path, this is equivalent to

$$\frac{dz(t)}{dt} = -[JH(z(t), t)]^{-1} \frac{\partial H(z(t), t)}{\partial t}.$$

- ▶ This is an **initial value problem** that can be solved using numerical methods.

# First-order tracking

- ▶ We solve the initial value problem using [Euler's method](#) starting at  $t_0 = 1$  with  $p_0$  as the initial value and successively computing the approximations  $p_1, p_2, \dots$  at values  $t_0 > t_1 > t_2 > \dots > 0$ .

# First-order tracking

- ▶ We solve the initial value problem using **Euler's method** starting at  $t_0 = 1$  with  $p_0$  as the initial value and successively computing the approximations  $p_1, p_2, \dots$  at values  $t_0 > t_1 > t_2 > \dots > 0$ .
- ▶ The approximations are computed as

$$p_{i+1} = p_i - JH(p_i, t_i)^{-1} \frac{\partial H(p_i, t_i)}{\partial t} \Delta t_i,$$

where  $\Delta t_i = t_{i+1} - t_i$ .

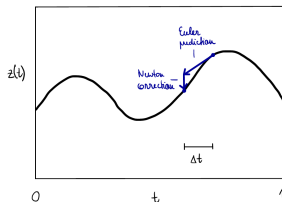
# First-order tracking

- ▶ We solve the initial value problem using **Euler's method** starting at  $t_0 = 1$  with  $p_0$  as the initial value and successively computing the approximations  $p_1, p_2, \dots$  at values  $t_0 > t_1 > t_2 > \dots > 0$ .
- ▶ The approximations are computed as

$$p_{i+1} = p_i - JH(p_i, t_i)^{-1} \frac{\partial H(p_i, t_i)}{\partial t} \Delta t_i,$$

where  $\Delta t_i = t_{i+1} - t_i$ .

- ▶ Geometrically this means predicting along the tangent line to the solution path at the current point of the path.



# Correction

- ▶ The prediction is often followed by the **correction** using the Newton's method.
- ▶ This means Newton's method is used for  $H(z, t_{i+1})$  starting with  $z_0 = p_{i+1}$ .
- ▶ **Newton's method** uses the iterative formula

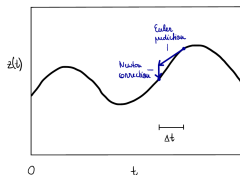
$$z_{i+1} = z_i - [JH(z_i, t_{i+1})]^{-1}H(z_i, t).$$

# Correction

- ▶ The prediction is often followed by the **correction** using the Newton's method.
- ▶ This means Newton's method is used for  $H(z, t_{i+1})$  starting with  $z_0 = p_{i+1}$ .
- ▶ **Newton's method** uses the iterative formula

$$z_{i+1} = z_i - [JH(z_i, t_{i+1})]^{-1}H(z_i, t).$$

- ▶ One or two iterations of Newton's method usually improves the prediction of  $z(t_{i+1})$ .
- ▶  $p_{i+1}$  is replaced with the corrected value before starting the next predictor-corrector cycle.



- ▶ In practice  $\Delta t_i$  is chosen **adaptively**.
- ▶ If the error after the correction is larger than the desired tracking accuracy, then  $\Delta t_i$  is halved.

- ▶ In practice  $\Delta t_i$  is chosen **adaptively**.
- ▶ If the error after the correction is larger than the desired tracking accuracy, then  $\Delta t_i$  is halved.
- ▶ Often **higher-order methods** (e.g. Runge-Kutta methods) are used in practice.
- ▶ They have the advantage that they often allow larger step sizes.



# From square systems to general systems

- ▶ Consider a general system

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_n(z_1, \dots, z_N) \end{bmatrix} = 0.$$

- ▶ If  $n < N$ , then the system is **underdetermined** and the solution set has positive-dimensional solution components.

# From square systems to general systems

- ▶ Consider a general system

$$f(z) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_n(z_1, \dots, z_N) \end{bmatrix} = 0.$$

- ▶ If  $n < N$ , then the system is **underdetermined** and the solution set has positive-dimensional solution components.
- ▶ If  $n > N$ , let  $A \in \mathbb{C}^{N \times n}$  be a **random matrix**. Instead of the system

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix},$$

we consider the system

$$A \cdot f.$$

# From square systems to general systems

- ▶ Every polynomial in the system  $A \cdot f$  has the form

$$a_{i1}f_1 + a_{i2}f_2 + \dots + a_{in}f_n,$$

where  $a_{ij}$  are random complex numbers.

# From square systems to general systems

- ▶ Every polynomial in the system  $A \cdot f$  has the form

$$a_{i1}f_1 + a_{i2}f_2 + \dots + a_{in}f_n,$$

where  $a_{ij}$  are random complex numbers.

- ▶ With probability one, all the isolated solutions of  $f$  are isolated solutions of  $A \cdot f$ .

# From square systems to general systems

- ▶ Every polynomial in the system  $A \cdot f$  has the form

$$a_{i1}f_1 + a_{i2}f_2 + \dots + a_{in}f_n,$$

where  $a_{ij}$  are random complex numbers.

- ▶ With probability one, all the isolated solutions of  $f$  are isolated solutions of  $A \cdot f$ .
- ▶ The system  $A \cdot f$  could have more solutions than  $f$ .
- ▶ The extra solutions can be detected because they do not satisfy  $f$ .

# From square systems to general systems

## Example

- ▶ Let  $p(z) = (z + 1)(z - 1)$  and  $q(z) = z(z - 1)$ .
- ▶ The system  $p(z) = q(z) = 0$  has one solution  $z = 1$ .

# From square systems to general systems

## Example

- ▶ Let  $p(z) = (z + 1)(z - 1)$  and  $q(z) = z(z - 1)$ .
- ▶ The system  $p(z) = q(z) = 0$  has one solution  $z = 1$ .
- ▶ Consider

$$2p(z) - 3q(z) = 2(z + 1)(z - 1) - 3z(z - 1) = (2 - z)(z - 1).$$

- ▶ This system has **two solutions**  $z = 1$  and  $z = 2$ .
- ▶ For  $z = 2$ , we have  $p(2) = 3$  and  $q(2) = 2$ , so it is **not a solution of the original system**.

# From square systems to general systems

## Example

- ▶ Let  $p(z) = (z + 1)(z - 1)$  and  $q(z) = z(z - 1)$ .
- ▶ The system  $p(z) = q(z) = 0$  has one solution  $z = 1$ .
- ▶ Consider

$$2p(z) - 3q(z) = 2(z + 1)(z - 1) - 3z(z - 1) = (2 - z)(z - 1).$$

- ▶ This system has **two solutions**  $z = 1$  and  $z = 2$ .
- ▶ For  $z = 2$ , we have  $p(2) = 3$  and  $q(2) = 2$ , so it is **not a solution of the original system**.
- ▶ Since for most choices of constants we get a **degree two polynomial**, there are necessarily two solutions.
- ▶ This second solution changes when different coefficients are used.



# Numerical algebraic geometry packages

- ▶ Bertini
- ▶ Julia Homotopy Continuation
- ▶ NumericalAlgebraicGeometry package in Macaulay2
- ▶ PHCpack



## An introduction to the numerical solution of polynomial systems

The basics of the theory and techniques behind HomotopyContinuation.jl

- 01 A first example
- 02 Homotopy continuation methods
- 03 Tracking solution paths
- 04 Constructing start systems and homotopies
- 05 Case Study: Optimization
- 06 Solving the critical equations
- 07 Computing critical points repeatedly
- 08 Alternative start systems
- 09 More information

# Reconstruction algorithm

# Reconstruction algorithm when $\alpha = -2$

The method consists of the following main steps:

1. Estimation of the unambiguous beads  $\{x_i, y_i\}_{i \in U}$  through semidefinite programming (Euclidean distance problem).

## Reconstruction algorithm when $\alpha = -2$

The method consists of the following main steps:

1. Estimation of the unambiguous beads  $\{x_i, y_i\}_{i \in U}$  through semidefinite programming (Euclidean distance problem).
2. A preliminary estimation of the ambiguous beads using numerical algebraic geometry.

## Reconstruction algorithm when $\alpha = -2$

The method consists of the following main steps:

1. Estimation of the unambiguous beads  $\{x_i, y_i\}_{i \in U}$  through semidefinite programming (Euclidean distance problem).
2. A preliminary estimation of the ambiguous beads using numerical algebraic geometry.
3. A refinement of this estimation using local optimization.

# Reconstruction algorithm when $\alpha = -2$

The method consists of the following main steps:

1. Estimation of the unambiguous beads  $\{x_i, y_i\}_{i \in U}$  through semidefinite programming (Euclidean distance problem).
2. A preliminary estimation of the ambiguous beads using numerical algebraic geometry.
3. A refinement of this estimation using local optimization.
4. A final clustering step, where we make a choice between the estimations  $(x_i, y_i)$  and  $(y_i, x_i)$  for each  $i \in A$  based on the assumption that homolog chromosomes are separated in space.

## Preliminary estimation using numerical algebraic geometry

- ▶ Let  $x, y$  be the unknown coordinates in  $\mathbb{R}^3$  of a pair of ambiguous beads.



# Preliminary estimation using numerical algebraic geometry

- ▶ Let  $x, y$  be the unknown coordinates in  $\mathbb{R}^3$  of a pair of ambiguous beads.
- ▶ We pick six unambiguous beads with already estimated coordinates  $a, b, c, d, e, f \in \mathbb{R}^3$ .

# Preliminary estimation using numerical algebraic geometry

- ▶ Let  $x, y$  be the unknown coordinates in  $\mathbb{R}^3$  of a pair of ambiguous beads.
- ▶ We pick six unambiguous beads with already estimated coordinates  $a, b, c, d, e, f \in \mathbb{R}^3$ .
- ▶ For each  $t \in \{a, \dots, f\}$ , let  $c_t \in \mathbb{R}$  be the corresponding partially ambiguous counts between  $t$  and the ambiguous bead pair  $(x, y)$ .

# Preliminary estimation using numerical algebraic geometry

- ▶ Let  $x, y$  be the unknown coordinates in  $\mathbb{R}^3$  of a pair of ambiguous beads.
- ▶ We pick six unambiguous beads with already estimated coordinates  $a, b, c, d, e, f \in \mathbb{R}^3$ .
- ▶ For each  $t \in \{a, \dots, f\}$ , let  $c_t \in \mathbb{R}$  be the corresponding partially ambiguous counts between  $t$  and the ambiguous bead pair  $(x, y)$ .
- ▶ Clearing the denominators, we obtain a system of polynomial equations

$$\|x - t\|^2 + \|y - t\|^2 = c_t \|x - t\|^2 \|y - t\|^2 \text{ for } t = a, b, c, d, e, f.$$

# Preliminary estimation using numerical algebraic geometry

- ▶ Let  $x, y$  be the unknown coordinates in  $\mathbb{R}^3$  of a pair of ambiguous beads.
- ▶ We pick six unambiguous beads with already estimated coordinates  $a, b, c, d, e, f \in \mathbb{R}^3$ .
- ▶ For each  $t \in \{a, \dots, f\}$ , let  $c_t \in \mathbb{R}$  be the corresponding partially ambiguous counts between  $t$  and the ambiguous bead pair  $(x, y)$ .
- ▶ Clearing the denominators, we obtain a system of polynomial equations

$$\|x - t\|^2 + \|y - t\|^2 = c_t \|x - t\|^2 \|y - t\|^2 \text{ for } t = a, b, c, d, e, f.$$

- ▶ This system has finitely many complex solutions both in the noiseless and noisy setting, which can be found using homotopy continuation.

# Preliminary estimation using numerical algebraic geometry

- ▶ We make a number  $N \geq 2$ , choices of sets of six unambiguous beads, and solve the corresponding  $N$  square systems.

# Preliminary estimation using numerical algebraic geometry

- ▶ We make a number  $N \geq 2$ , choices of sets of six unambiguous beads, and solve the corresponding  $N$  square systems.
- ▶ For each system, we pick out the approximately real solutions, and obtain  $N$  sets  $\mathcal{S}_1, \dots, \mathcal{S}_N \subseteq \mathbb{R}^6$  consisting of the real parts of the approximately real solutions.

# Preliminary estimation using numerical algebraic geometry

- ▶ We make a number  $N \geq 2$ , choices of sets of six unambiguous beads, and solve the corresponding  $N$  square systems.
- ▶ For each system, we pick out the approximately real solutions, and obtain  $N$  sets  $\mathcal{S}_1, \dots, \mathcal{S}_N \subseteq \mathbb{R}^6$  consisting of the real parts of the approximately real solutions.
- ▶ Up to the symmetry  $(x, y) \mapsto (y, x)$ , we expect these sets to have a unique “approximately common” element.

- ▶ A disadvantage of the numerical algebraic geometry based estimation discussed in the previous subsection is that it only takes into account “local” information about the interactions for one ambiguous locus at a time, which might make it more sensitive to noise.



# Local optimization

- ▶ A disadvantage of the numerical algebraic geometry based estimation discussed in the previous subsection is that it only takes into account “local” information about the interactions for one ambiguous locus at a time, which might make it more sensitive to noise.
- ▶ We refine this preliminary estimation of  $\{x_i, y_i\}_{i \in A}$  further in a local optimization step that takes into account the “global” information of all available data.

- ▶ The idea is to estimate  $\{x_i, y_i\}_{i \in A}$  by solving the optimization problem

$$\min_{\{x_i, y_i\}_{i \in A}} \sum_{i \in U, j \in A} \left( \left( c_{i,j}^P - \frac{1}{\|x_i - x_j\|^2} - \frac{1}{\|x_i - y_j\|^2} \right)^2 + \left( c_{i+n,j}^P - \frac{1}{\|y_i - x_j\|^2} - \frac{1}{\|y_i - y_j\|^2} \right)^2 \right)$$

while keeping the estimates of  $\{x_i, y_i\}_{i \in U}$  fixed.

- ▶ The idea is to estimate  $\{x_i, y_i\}_{i \in A}$  by solving the optimization problem

$$\min_{\{x_i, y_i\}_{i \in A}} \sum_{i \in U, j \in A} \left( \left( c_{i,j}^P - \frac{1}{\|x_i - x_j\|^2} - \frac{1}{\|x_i - y_j\|^2} \right)^2 + \left( c_{i+n,j}^P - \frac{1}{\|y_i - x_j\|^2} - \frac{1}{\|y_i - y_j\|^2} \right)^2 \right)$$

while keeping the estimates of  $\{x_i, y_i\}_{i \in U}$  fixed.

- ▶ We use Matlab Optimization Toolbox for this step.

- ▶ The idea is to estimate  $\{x_i, y_i\}_{i \in A}$  by solving the optimization problem

$$\min_{\{x_i, y_i\}_{i \in A}} \sum_{i \in U, j \in A} \left( \left( c_{i,j}^P - \frac{1}{\|x_i - x_j\|^2} - \frac{1}{\|x_i - y_j\|^2} \right)^2 + \left( c_{i+n,j}^P - \frac{1}{\|y_i - x_j\|^2} - \frac{1}{\|y_i - y_j\|^2} \right)^2 \right)$$

while keeping the estimates of  $\{x_i, y_i\}_{i \in U}$  fixed.

- ▶ We use Matlab Optimization Toolbox for this step.
- ▶ The already estimated coordinates of  $\{x_i, y_i\}_{i \in A}$  from the numerical algebraic geometry step are used for the initialization.

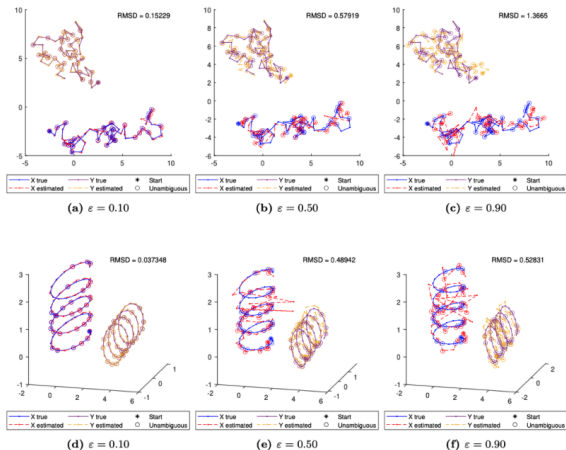
# Breaking symmetry

- ▶ Our objective function remains invariant if we exchange  $x_i$  and  $y_i$  for any  $i \in A$ .

# Breaking symmetry

- ▶ Our objective function remains invariant if we exchange  $x_i$  and  $y_i$  for any  $i \in A$ .
- ▶ We can break symmetry by relying on the empirical observation that homologous chromosomes typically are spatially separated in different so-called compartments of the nucleus.

# Reconstruction examples



**Figure 3.** Examples of reconstructions for varying noise levels, for a chromosome pair with 60 loci, out of which 50% are ambiguous. Subfigures (a)–(c) show chromosomes simulated with Brownian motion (projected onto the  $xy$ -plane), whereas figure (d)–(e) show helix-shaped chromosomes.

# Real data reconstruction

Reconstructions based on the real data set, which is obtained from Hi-C experiments on the X chromosomes in the Patski (BL6xSpretus) cell line. The data has been recorded at a resolution of 500 kb, which corresponds to 343 bead pairs in the model.

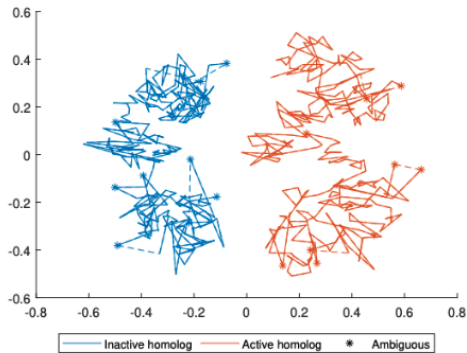


Figure: Cifuentes et al. 3D genome reconstruction from partially phased Hi-C data. Preprint.



# Real data reconstruction

It was discovered in [Deng et al, 2015] that the inactive homolog in the Patski X chromosome pair has a bipartite structure, consisting of two superdomains with frequent intra-chromosome contacts within the superdomains and a boundary region between the two superdomains. The active homolog does not exhibit the same behaviour.

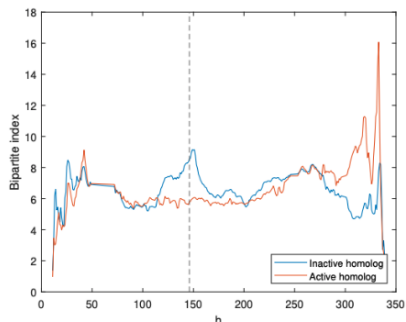


Figure: Cifuentes et al. 3D genome reconstruction from partially phased Hi-C data. Preprint.

# Summary

## Summary:

- ▶ We discussed how to find the 3D structure of genome from contact count matrices with focus on partially phased data.
- ▶ Theoretical results to have finitely many reconstructions.
- ▶ Algorithm to find a reconstruction using semidefinite programming and numerical algebraic geometry.

- ▶ Bates, D. J., Sommese, A. J., Hauenstein, J. D., & Wampler, C. W. (2013). Numerically solving polynomial systems with Bertini. Society for Industrial and Applied Mathematics.
- ▶ Belyaeva, A., Kubjas, K., Sun, L. J., & Uhler, C. (2022). Identifying 3D genome organization in diploid organisms via euclidean distance geometry. *SIAM Journal on Mathematics of Data Science*, 4(1), 204-228.
- ▶ Cifuentes, D., Draisma, J., Henriksson, O., Korchmaros, A., & Kubjas, K. (2023+). 3D genome reconstruction from partially phased Hi-C data.