Relaxations of the Seriation Problem and Applications to de novo Genome Assembly

Soutenance de thèse

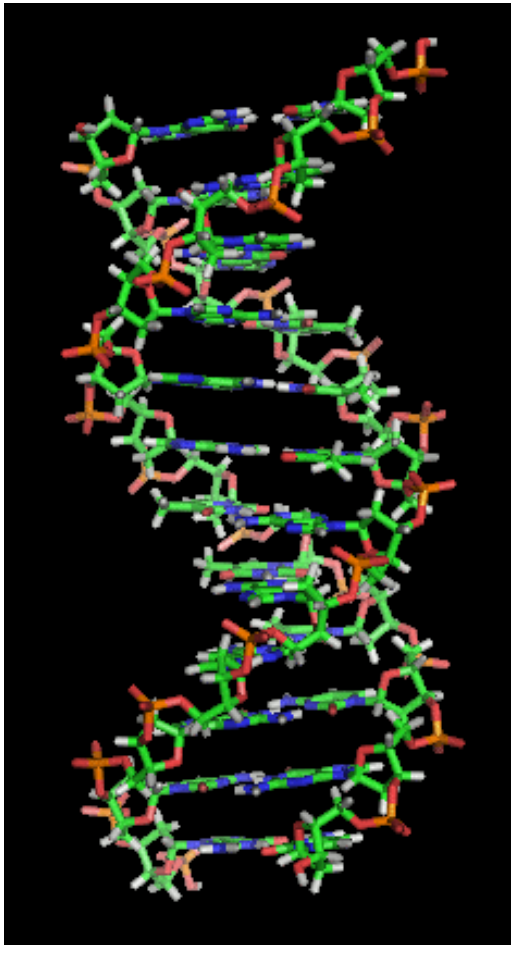
Antoine Recanati

sous la direction d'Alexandre d'Aspremont

29 Novembre 2018

Introduction

Genome sequencing



⇨

...ATGGCGTGCAATG...
...TACCGCACGTTAC...

DNA sequencing

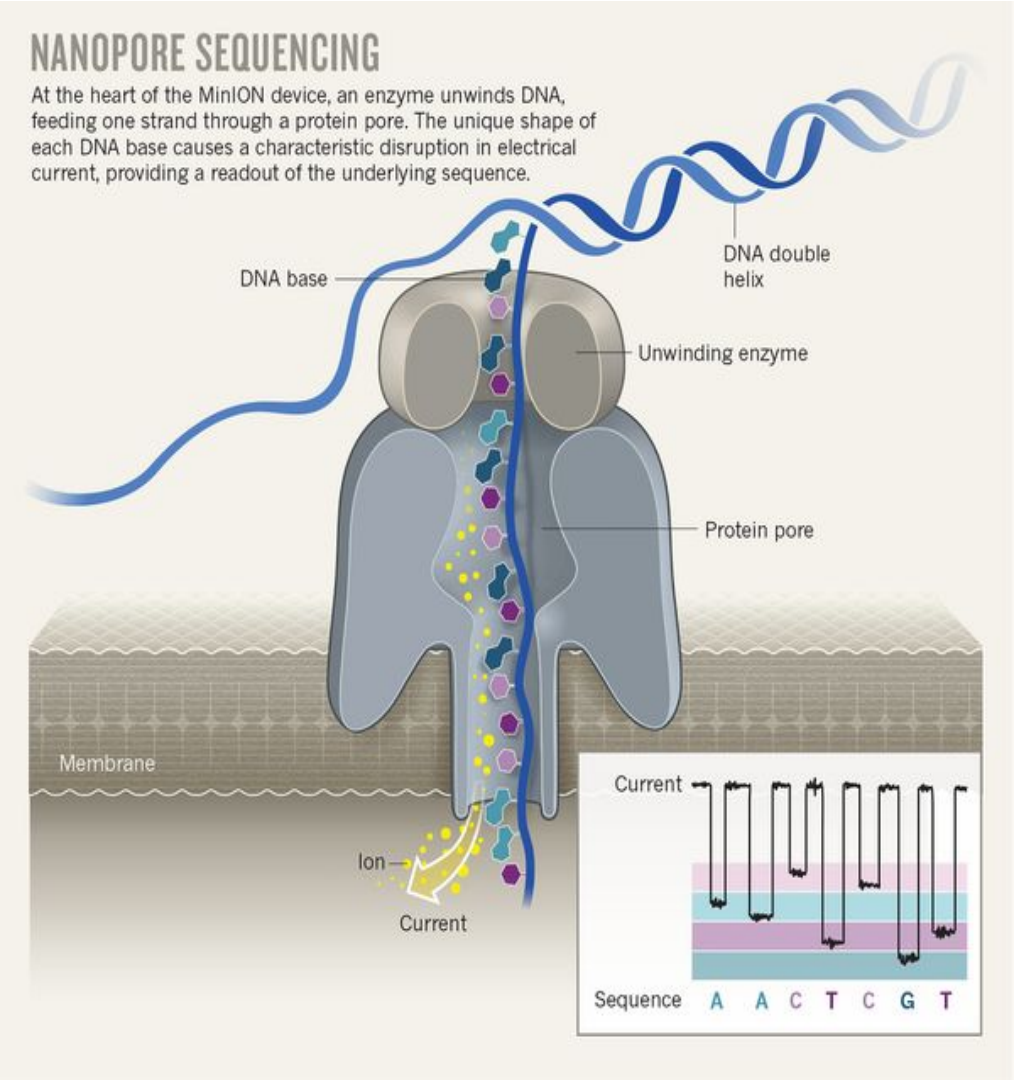


Image: Nik Spencer/Nature

Genome is cut into overlapping fragments (*reads*).

Ex:

ATGGCGTGCAATG

DNA sequencing

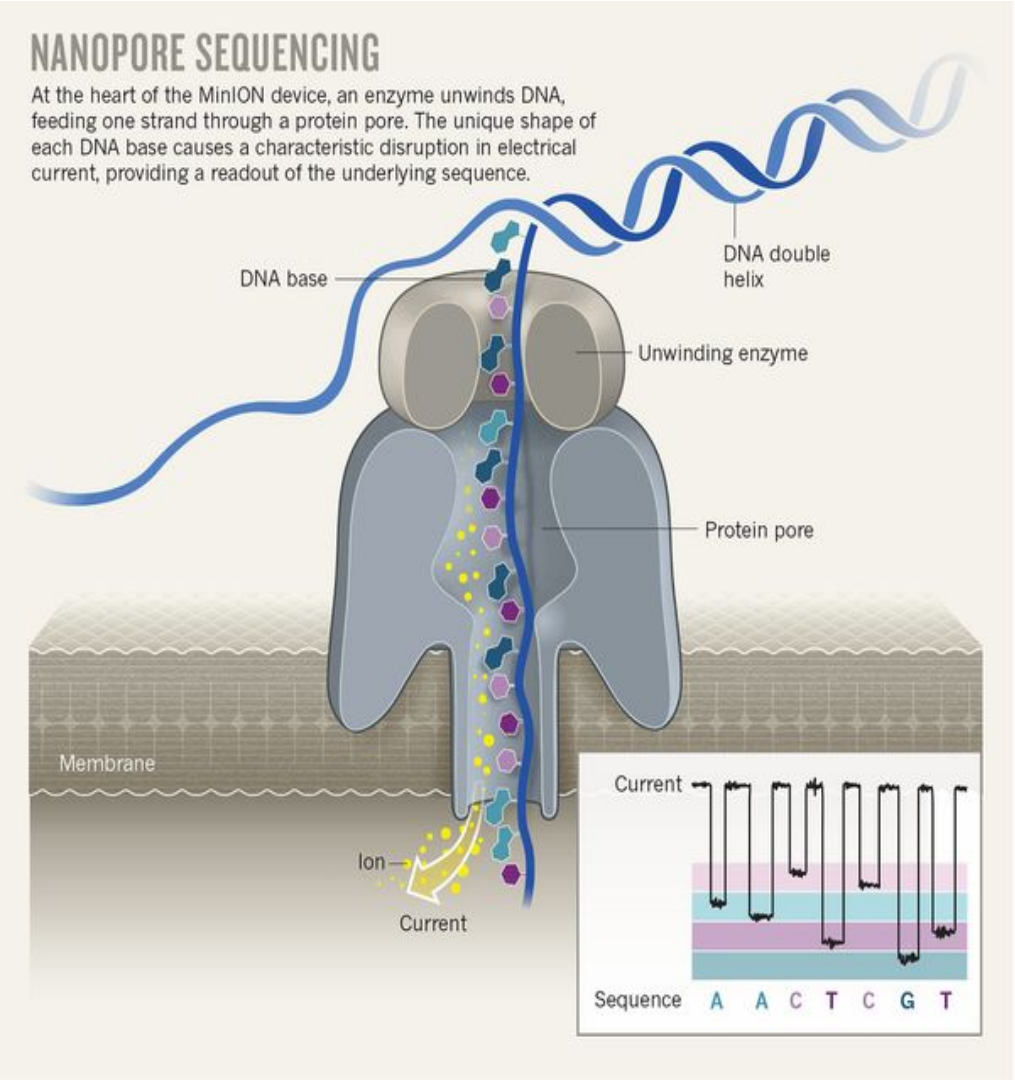


Image: Nik Spencer/Nature

Genome is cut into overlapping fragments (*reads*).

Ex:

ATGGCGTGCAATG

CGTGCAA

DNA sequencing

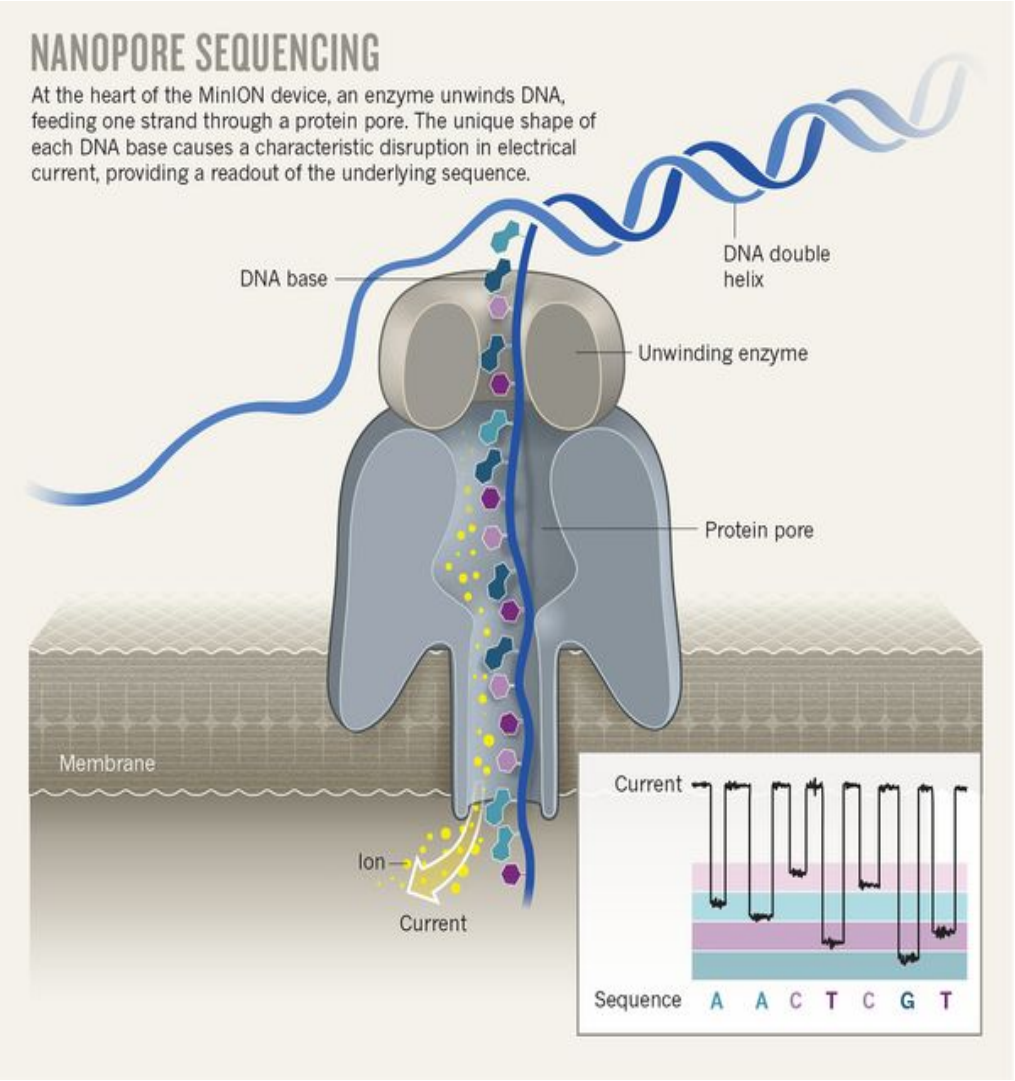


Image: Nik Spencer/Nature

Genome is cut into overlapping fragments (*reads*).

Ex:

ATGGCGTGCAATG

CGTGCAA

ATGGCGT

DNA sequencing

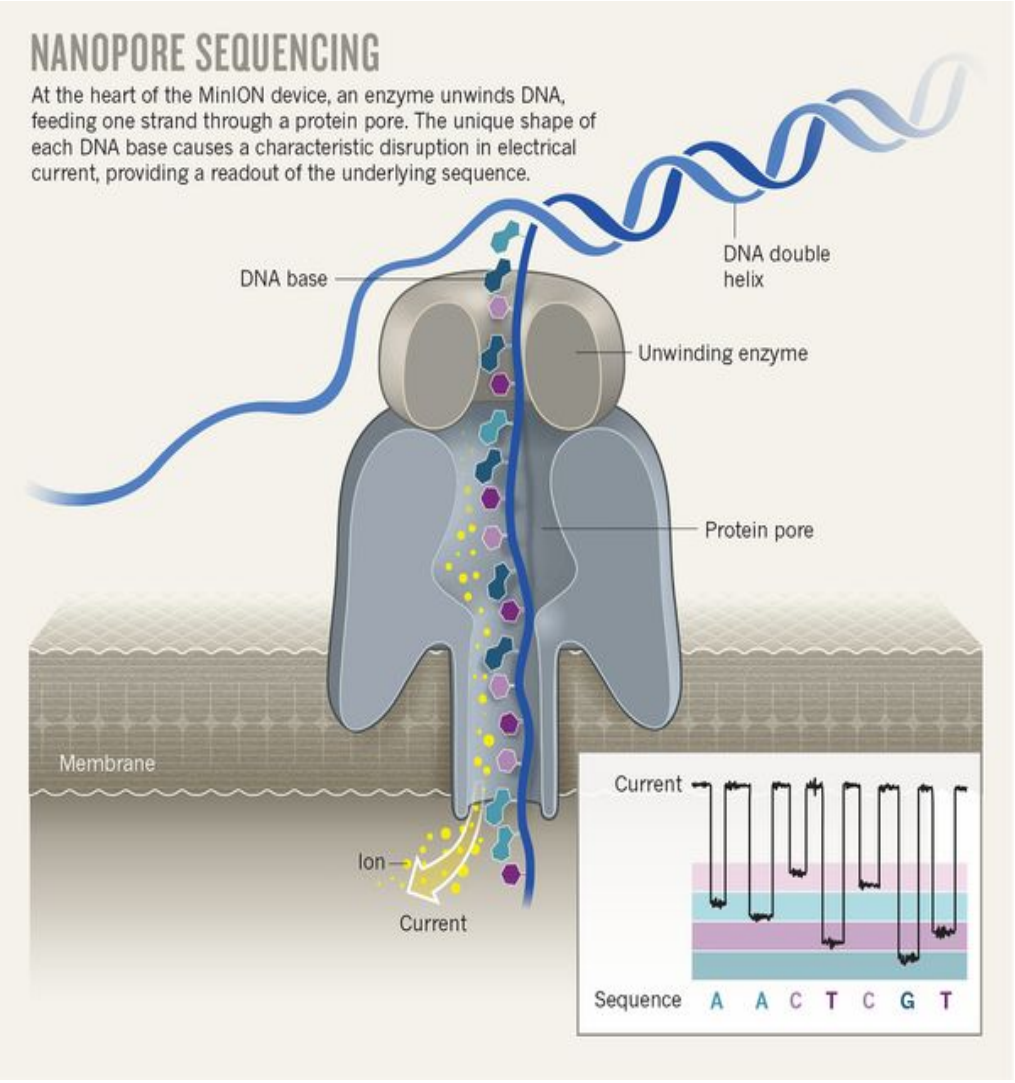


Image: Nik Spencer/Nature

Genome is cut into overlapping fragments (*reads*).

Ex:

ATGGCGTGCAATG

- CGTGCAA
- ATGGCGT
- TGCAATG

DNA sequencing

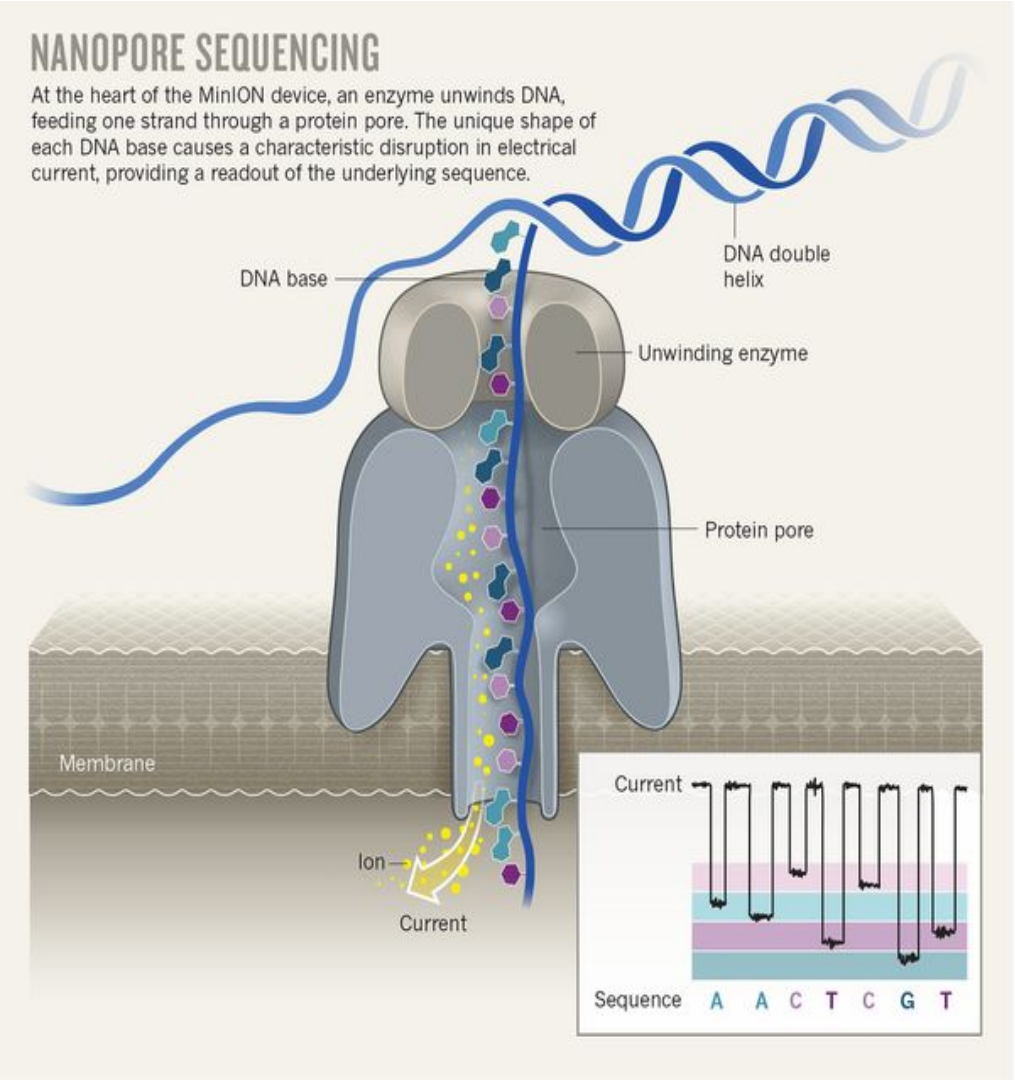


Image: Nik Spencer/Nature

Genome is cut into overlapping fragments (*reads*).

Ex:

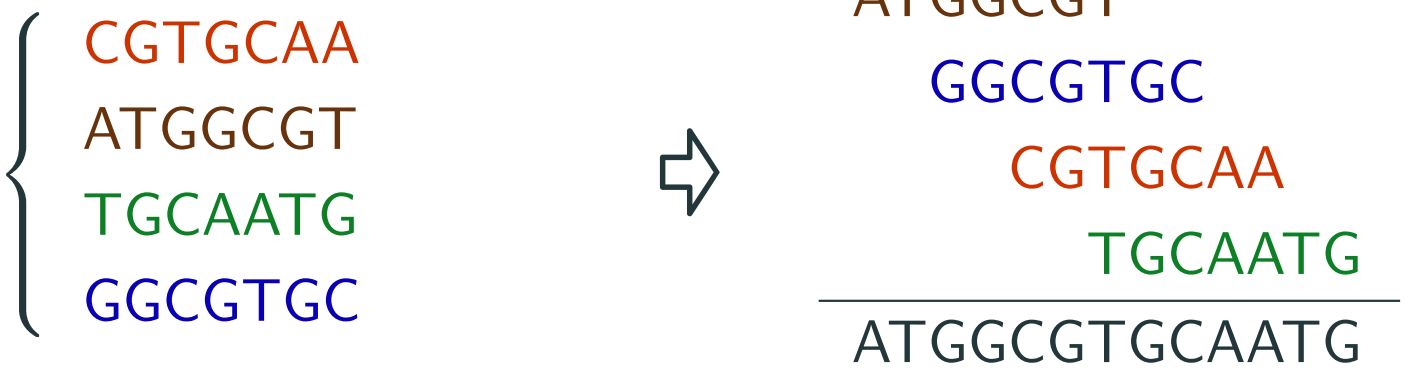
ATGGCGTGCAATG

- CGTGCAA
- ATGGCGT
- TGCAATG
- GGCGTGC

Assembly

Goal: assemble reads together to reconstruct the full sequence.

The position and ordering of the reads are unknown.



Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

AAGGCGTGCATTG (ref. (proxy))

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

AAGGCGTGCATTG (ref. (proxy))

CGTGCAA

ATGGCGT

TGCAATG

GGCGTGC

CGTGCAA

Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

AAGGCGTGCATTG (ref. (proxy))
CGTGCAA
ATGGCGT

Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

AAGGCGTGCATTG (ref. (proxy))
CGTGCAA
ATGGCGT
TGCAATG

Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

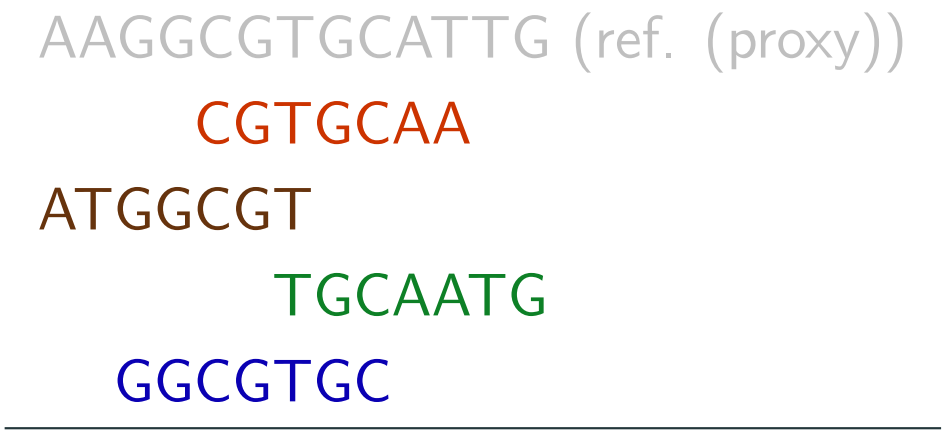
CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

AAGGCGTGCATTG (ref. (proxy))
CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC



Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

AAGGCGTGCATTG (ref. (proxy))
CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

ATGGCGTGCAATG (assembly)

Genome assembly: mapping

If reference genome available: map the fragments to it, then derive consensus sequence

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

AAGGCGTGCATTG (ref. (proxy))
CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

ATGGCGTGCAATG (assembly)

Genome assembly: *de novo*

No reference available. Greedy assembly: take one read, "add" the one with largest overlap, etc., until all reads are included.

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

Genome assembly: *de novo*

No reference available. Greedy assembly: take one read, "add" the one with largest overlap, etc., until all reads are included.

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

CGTGCAA

CGTGCAA

Genome assembly: *de novo*

No reference available. Greedy assembly: take one read, "add" the one with largest overlap, etc., until all reads are included.

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

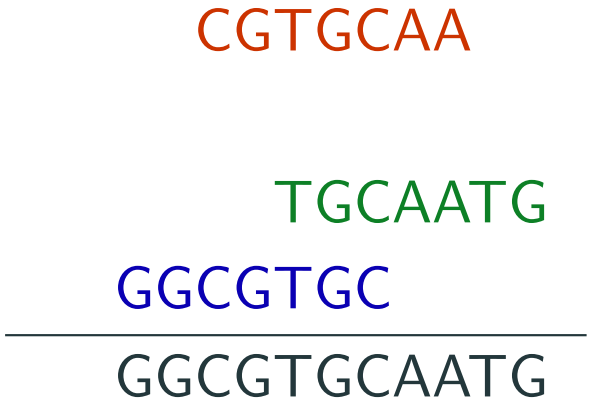
CGTGCAA
TGCAATG

CGTGCAATG

Genome assembly: *de novo*

No reference available. Greedy assembly: take one read, "add" the one with largest overlap, etc., until all reads are included.

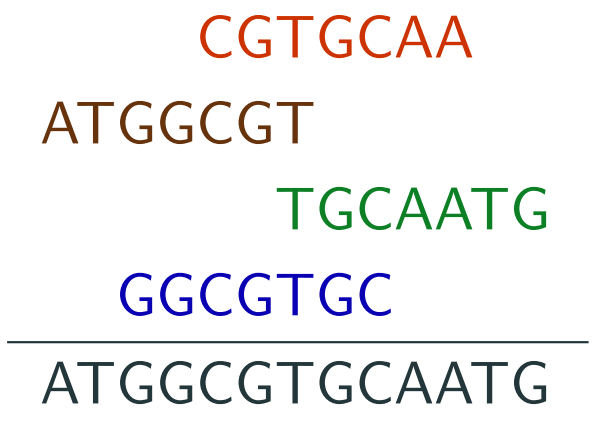
CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC



Genome assembly: *de novo*

No reference available. Greedy assembly: take one read, "add" the one with largest overlap, etc., until all reads are included.

CGTGCAA
ATGGCGT
TGCAATG
GGCGTGC

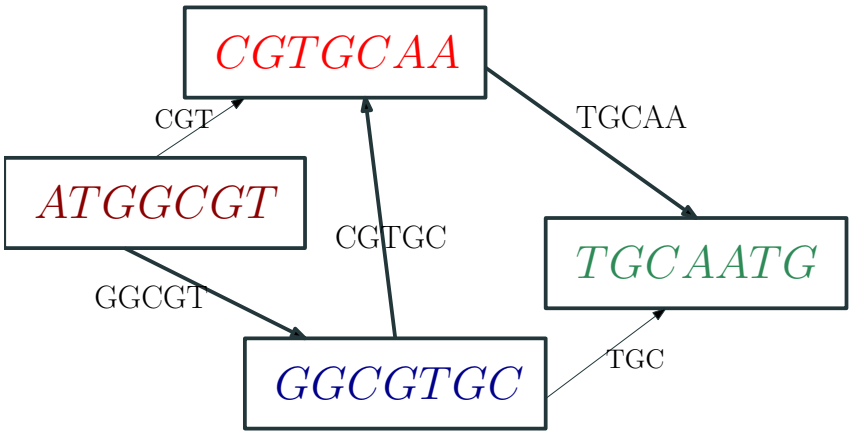


De novo **assembly paradigms**

- Greedy methods
- De Bruijn graphs
- Overlap-Layout-Consensus

Overlap-Layout-Consensus

- Compute overlaps between all read pairs
- Find tiling of reads consistent with overlaps
- Average reads values to create consensus sequence



ATGGCGT
GGCGTGC
CGTGCAA
TGCAATG

ATGGCGTGCAATG

Modern sequencing technologies

- 2nd gen. (SGS): **short** ($\sim$100bp), **accurate** ($<$ 2% err.)reads (Illumina/Solexa), with **pairing** information. **De Bruijn graphs** methods (on k-mers based graph) preferred.
- 3rd. gen.: **long** ($\sim$10000bp), **noisy** ($\sim$10%) reads (Pacific Biosciences [PacBio], Oxford Nanopore Technology [ONT]). Come-back of **OLC** methods.
- Can be combined to have both accuracy and length (**hybrid methods**)

De novo assembly methods with ONT reads

State of the art: Canu (ex. Celera Assembler). Heavy **pre-processing**, many **heuristics**

- correction: (uses [hash-based] overlaps for consensus)
- trimming: recalculate overlaps to filter low-coverage/high-error regions
- re-computation of overlaps with specific target errors (uses a priori model of errors)
- assemble *unitigs* (unambiguous sequences) first, then incremental scaffolding

De novo assembly methods with ONT reads

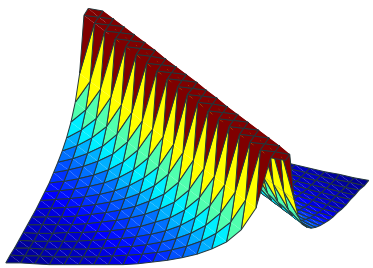
- ONT-only assemblers (non-hybrid): active field of research 2015-now
- Canu: **complex** pipeline, **high quality** consensus.
- Miniasm: ideas of Canu assembly, no pre-processing, smart heuristics. **Ultra-fast**, **low-quality**.
- Naive OLC approach with clean mathematical formulation ?

Introduction: The Seriation Problem

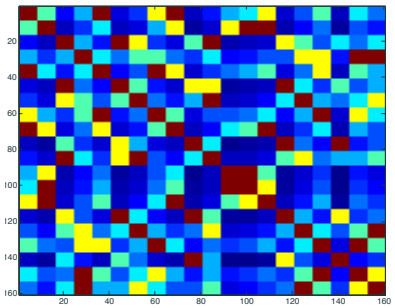
- Pairwise **similarity information** A_{ij} on n variables.
- Suppose the data has a **serial structure**, i.e. there is an order π such that

$$A_{\pi(i)\pi(j)} \text{ decreases with } |i-j| \quad \textbf{(R-matrix)}$$

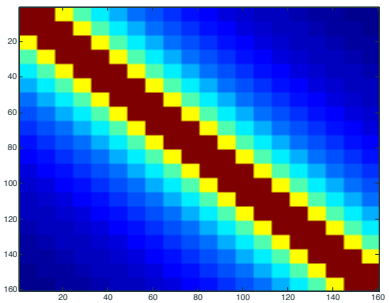
Recover π?



Similarity matrix

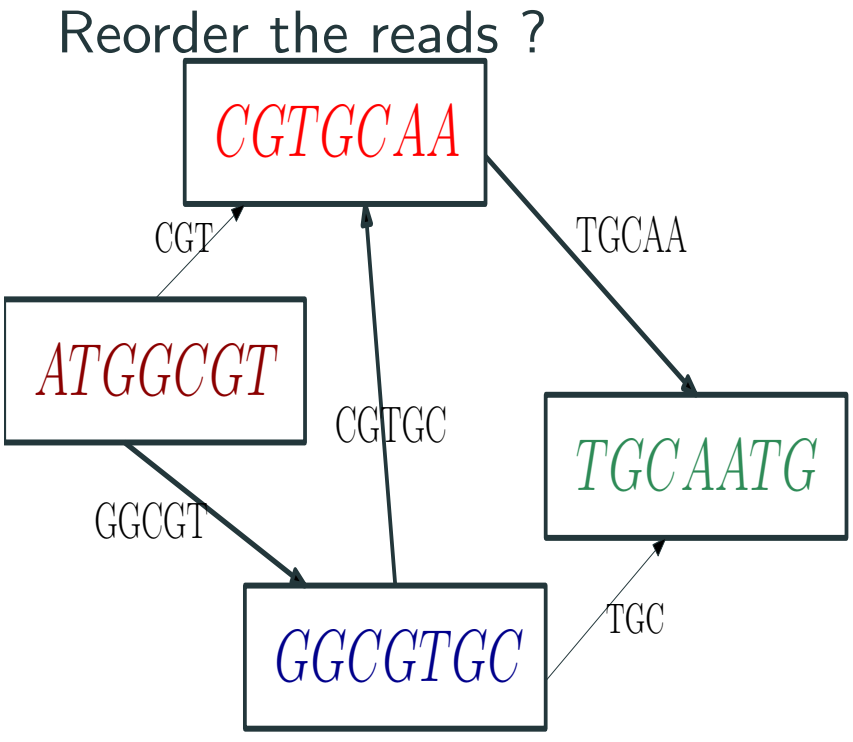


Input



Reconstructed

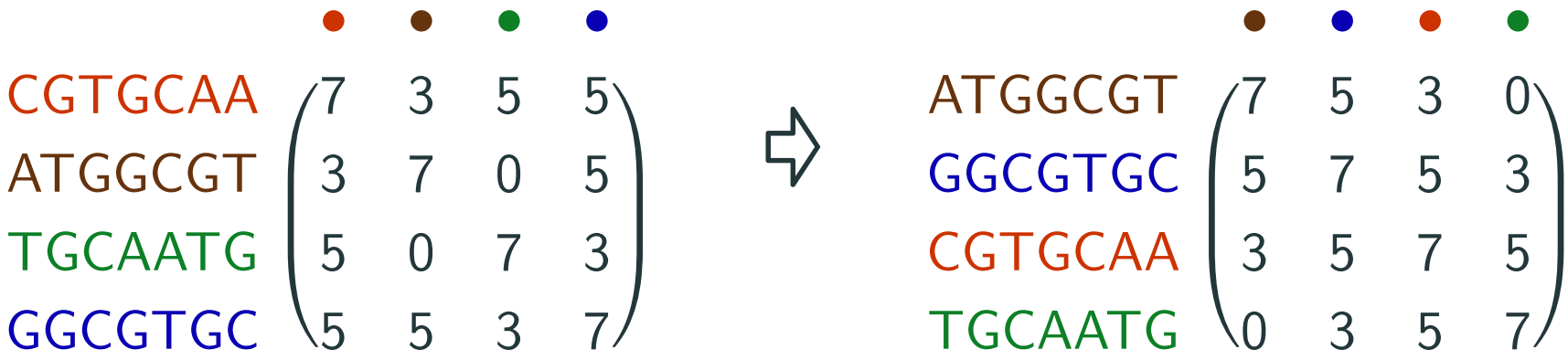
Seriation in OLC assembly



	●	●	●	●
CGTGCAA	7	3	5	5
ATGGCGT	3	7	0	5
TGCAATG	5	0	7	3
GGCGTGC	5	5	3	7

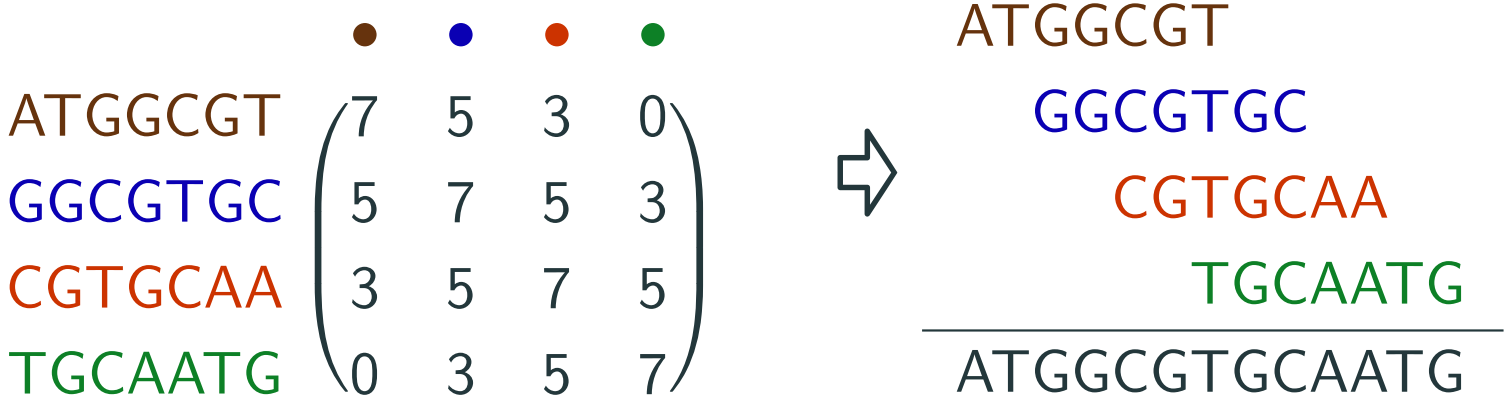
Seriation in OLC assembly (ctd.)

Solve Seriation to reorder reads:



Seriation in OLC assembly (ctd.)

The ordering yields the layout:



The 2-SUM combinatorial problem

- The **2-SUM problem** is written

$$\min_{\pi \in \mathcal{P}} \sum_{i,j=1}^{n} A_{ij}(\pi(i) - \pi(j))^2 \qquad \text{(2-SUM)}$$

- optimal π^*: high $A_{ij} \Leftrightarrow$ low $|\pi(i) - \pi(j)|$, *i.e.*, i and j are nearby.

The 2-SUM combinatorial problem

$$f_{2\text{SUM}} = \frac{1}{2} \sum_{i,j=1}^{n} A_{ij}(\pi_i - \pi_j)^2$$

$$\begin{array}{c} \\ \text{CGTGCAA} \\ \text{ATGGCGT} \\ \text{TGCAATG} \\ \text{GGCGTGC} \end{array} \begin{array}{c} \begin{array}{cccc} \bullet & \bullet & \bullet & \bullet \end{array} \\ \begin{pmatrix} 7 & 3 & 5 & 5 \\ 3 & 7 & 0 & 5 \\ 5 & 0 & 7 & 3 \\ 5 & 5 & 3 & 7 \end{pmatrix} \end{array} \Rightarrow$$

$$\begin{aligned} f_{2\text{SUM}} &= (1/2) * 4 * 7 * 1^2 \\ &+ 2 * 3 * 2^2 \\ &+ 2 * 5 * 3^2 \\ &+ 1 * 5 * 4^2 \\ &= 416 \end{aligned}$$

The 2-SUM combinatorial problem

$$f_{2\text{SUM}} = \frac{1}{2} \sum_{i,j=1}^{n} A_{ij}(\pi_i - \pi_j)^2$$

$$\begin{array}{l} \\ \text{ATGGCGT} \\ \text{GGCGTGC} \\ \text{CGTGCAA} \\ \text{TGCAATG} \end{array} \begin{array}{c} \begin{array}{cccc} \bullet & \bullet & \bullet & \bullet \end{array} \\ \begin{pmatrix} 7 & 5 & 3 & 0 \\ 5 & 7 & 5 & 3 \\ 3 & 5 & 7 & 5 \\ 0 & 3 & 5 & 7 \end{pmatrix} \end{array} \Rightarrow$$

$$\begin{aligned} f_{2\text{SUM}} &= (1/2) * 4 * 7 * 1^2 \\ &+ 3 * 5 * 2^2 \\ &+ 2 * 3 * 3^2 \\ &+ 1 * 0 * 4^2 \\ &= 142 \end{aligned}$$

Seriation vs 2SUM

- The optimal permutation for 2SUM solves Seriation [Fogel].

Seriation vs 2SUM

- The optimal permutation for 2SUM solves Seriation [Fogel].
- Solve 2-SUM ?

2-SUM is a quadratic

The 2-SUM objective is quadratic in π

$$\begin{aligned} f_{\mathsf{2SUM}}(\pi) &= \sum_{i,j=1}^{n} A_{ij}(\pi_i - \pi_j)^2 \\ &= \sum_{i,j=1}^{n} L_{ij}\pi_i\pi_j \\ &= \pi^T L\pi. \end{aligned}$$

with $L = \mathbf{diag}(A\mathbf{1}) - A$
(*i.e.*, $L_{ii} = \sum_{j\neq i} A_{ij}$, $L_{ij} = -A_{ij}$, $i \neq j$).

2-SUM is a quadratic

$$f_{2\mathrm{SUM}}(x) = \sum_{i,j=1}^{n} A_{ij}(x_i - x_j)^2 = x^T L x. \tag{1}$$

- L is symmetric and positive semi-definite.
- L has n non-negative, real-valued eigenvalues, $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.
- $\mathbf{1} = (1, \ldots, 1)^T$ is eigenvector associated to eigenvalue 0.
- Used in Spectral Clustering.

Relaxations of the set of permutations

How to optimize over the set of permutations ?

Set of permutation vectors
$\pi = (\pi_1, \ldots, \pi_n)$:

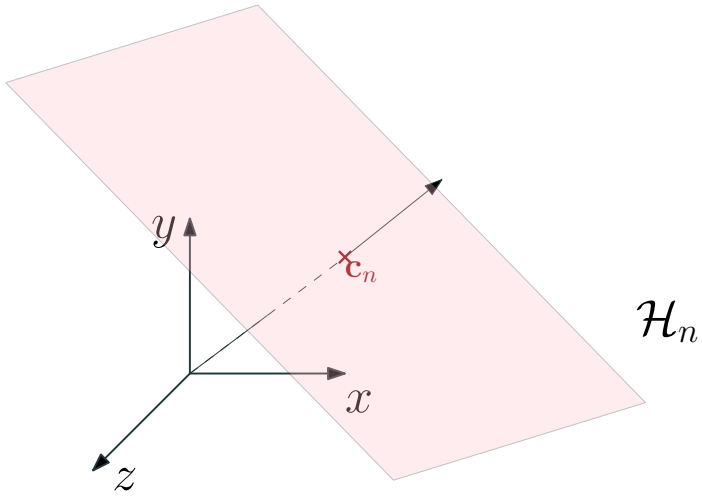
$$\left\{ \right.$$

Relaxations of the set of permutations

How to optimize over the set of permutations ?

Set of permutation vectors $\pi = (\pi_1, \ldots, \pi_n)$:

$$\begin{cases} \sum_i \pi_i = n(n+1)/2 \\ \end{cases}$$

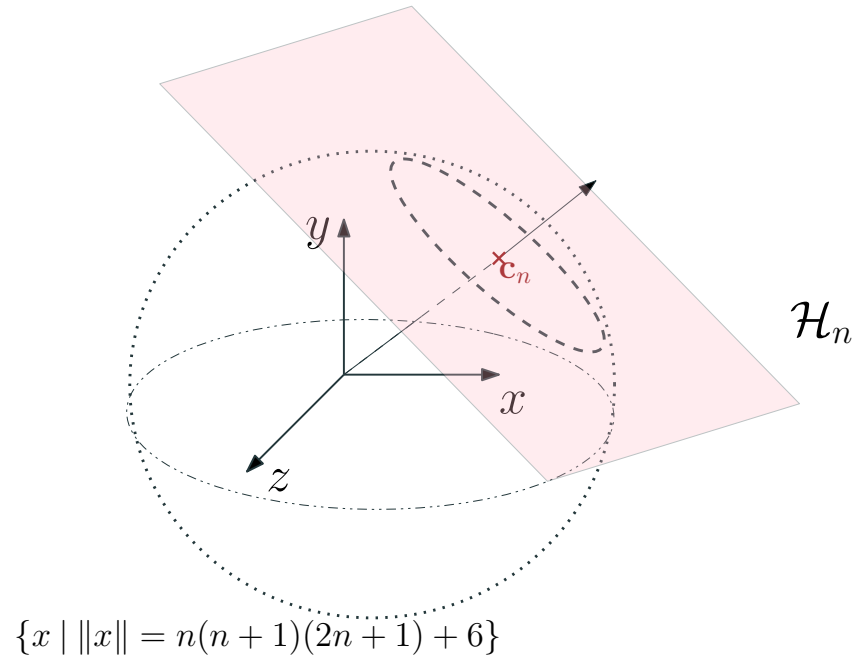


Relaxations of the set of permutations

How to optimize over the set of permutations ?

Set of permutation vectors $\pi = (\pi_1, \ldots, \pi_n)$:

$$\begin{cases} \sum_i \pi_i = n(n+1)/2 \\ \|\pi\|_2^2 = n(n+1)(2n+1)/6 \end{cases}$$

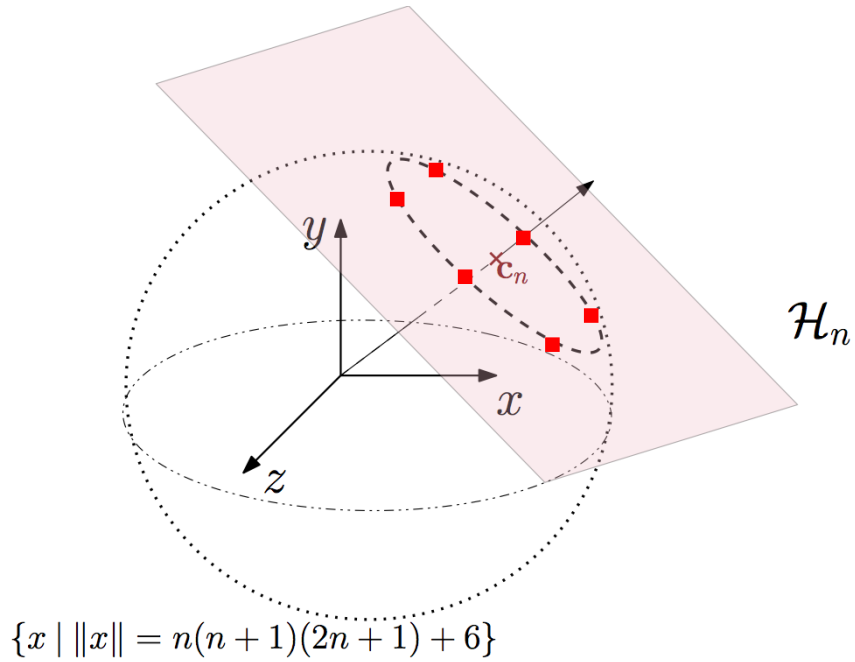


Relaxations of the set of permutations

How to optimize over the set of permutations ?

Set of permutation vectors $\pi = (\pi_1, \ldots, \pi_n)$:

$$\begin{cases} \sum_i \pi_i = n(n+1)/2 \\ \|\pi\|_2^2 = n(n+1)(2n+1)/6 \\ \pi_i \in \{1, ..., n\} \end{cases}$$

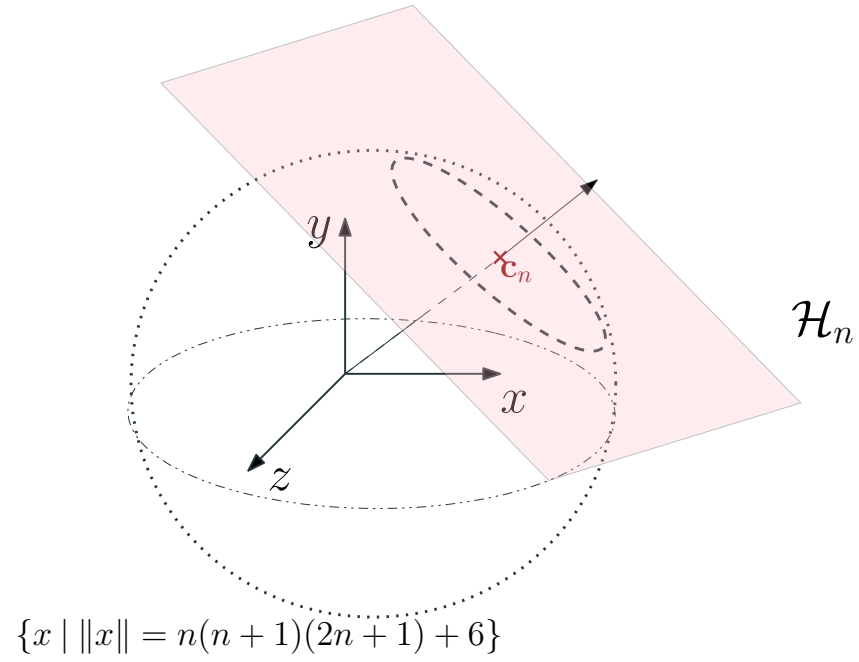


Spectral relaxation

Drop integer constraints

Relaxed set
$\pi = (\pi_1, \ldots, \pi_n)$:

$$\begin{cases} \sum_i \pi_i = n(n+1)/2 \\ \|\pi\|_2^2 = n(n+1)(2n+1)/6 \\ \pi_i \in \mathbb{R} \end{cases}$$



Spectral relaxation

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{such that} & \sum_i x_i = n(n+1)/2, \\ & \|x\|_2^2 = n(n+1)(2n+1)/6. \end{array}$$

- $L\mathbf{1} = 0$: $x \leftarrow x - \frac{(n+1)}{2}\mathbf{1}$

Spectral relaxation

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{such that} & \sum_i x_i = 1, \\ & \|x\|_2^2 = n(n+1)(2n+1)/6. \end{array}$$

- $L\mathbf{1} = 0$: $x \leftarrow x - \frac{(n+1)}{2}\mathbf{1}$

Spectral relaxation

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{such that} & \sum_i x_i = 1, \\ & \|x\|_2^2 = n(n+1)(2n+1)/6. \end{array}$$

- $L\mathbf{1} = 0$: $x \leftarrow x - \frac{(n+1)}{2}\mathbf{1}$
- homogeneous function optimize over sphere: $x \leftarrow x/\left(n(n+1)(2n+1)/6\right).$

Spectral relaxation

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{such that} & \sum_i x_i = 1, \\ & \|x\|_2^2 = 1. \end{array}$$

- $L\mathbf{1} = 0$: $x \leftarrow x - \frac{(n+1)}{2}\mathbf{1}$
- homogeneous function optimize over sphere: $x \leftarrow x/\left(n(n+1)(2n+1)/6\right).$

Spectral relaxation

$$\begin{aligned} \text{minimize} \quad & x^T L x \\ \text{such that} \quad & x^T \mathbf{1} = 0, \\ & \|x\|_2^2 = 1. \end{aligned}$$

- eigenvalue problem on L ($\mathbf{1}$ is first eigenvector).
- From NP hard to $O(n^2)$ (extremal eigenvalue)

Spectral ordering

Define the Laplacian of A as $L = \mathbf{diag}(A\mathbf{1}) - A$. The Fiedler vector f of A is the second smallest eigenvector of L:

$$f = \underset{\mathbf{1}^T x = 0,\ \|x\|_2 = 1}{\operatorname{argmin}} \quad x^T L_A x.$$

f reorders a R-matrix in the noiseless case.

Theorem ([Atkins)

Spectral Seriation] Suppose $A \in \mathbf{S}_n$ is a pre-R matrix, with a simple Fiedler value whose Fiedler vector f has no repeated values. Suppose that $\Pi \in \mathcal{P}$ is such that the permuted Fielder vector Πv is monotonic, then $\Pi A \Pi^T$ is an R-matrix.

Spectral Ordering Algorithm

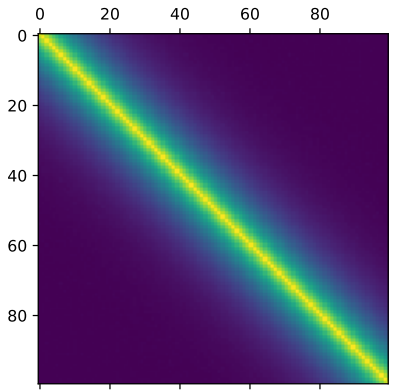
Input: Connected similarity matrix $A \in \mathbb{R}^{n \times n}$

1: Compute Laplacian $L = \mathbf{diag}(A\mathbf{1}) - A$

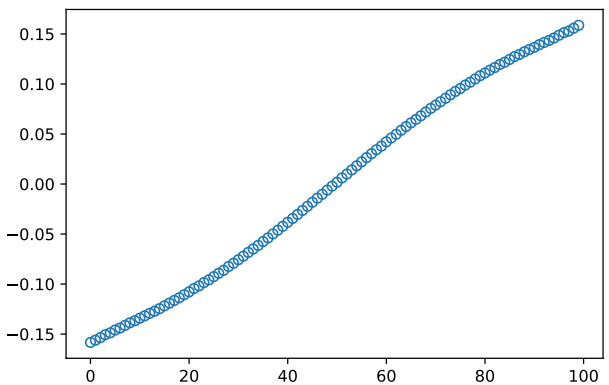
2: Compute second smallest eigenvector of L, $\mathbf{f}$

3: Sort the values of $\mathbf{f}$

Output: Permutation $\pi : \mathbf{f}_{\pi(1)} \leq \mathbf{f}_{\pi(2)} \leq ... \leq \mathbf{f}_{\pi(n)}$



Similarity matrix



Fiedler vector

Application of the Spectral Method to Genome Assembly

Related publication

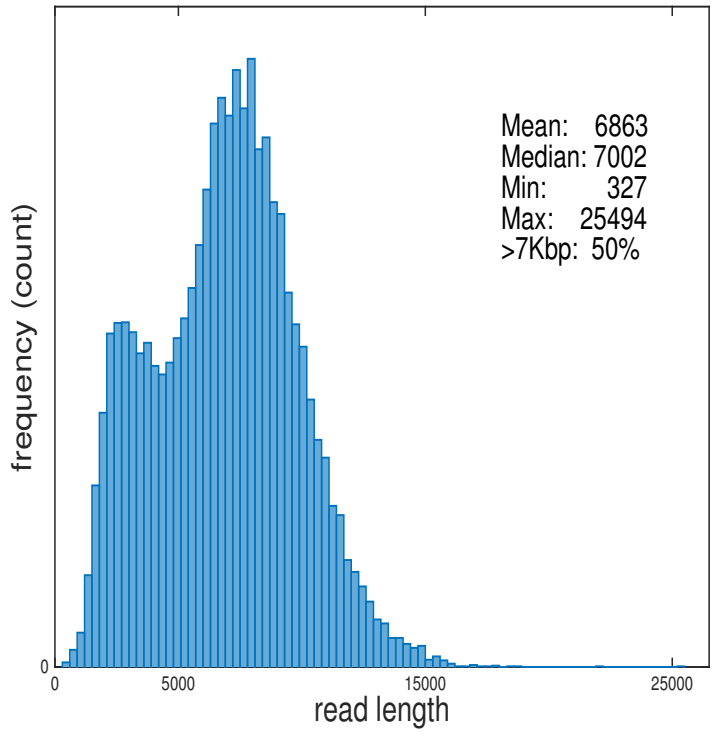
This section is based on the following publication:

Antoine Recanati, Thomas Brüls, and Alexandre d'Aspremont. **A spectral algorithm for fast de novo layout of uncorrected long nanopore reads**. *Bioinformatics*, 2016.

`https://github.com/antrec/spectrassembler`

Data: ONT and PacBio long noisy reads

- bacteria: *E. coli* and *A. baylyi*. Circular, prokaryotic, small (∼4Mbp) genomes, n_{reads} ∼20000, cov∼30X.
- yeast: *S. cerevisiae*. Eukaryotic genome, 16 chromosomes, ∼12Mbp, n_{reads} ∼100000, cov∼80X.



(*E. coli* read length hist.)

Overlap Layout Consensus (OLC) pipeline

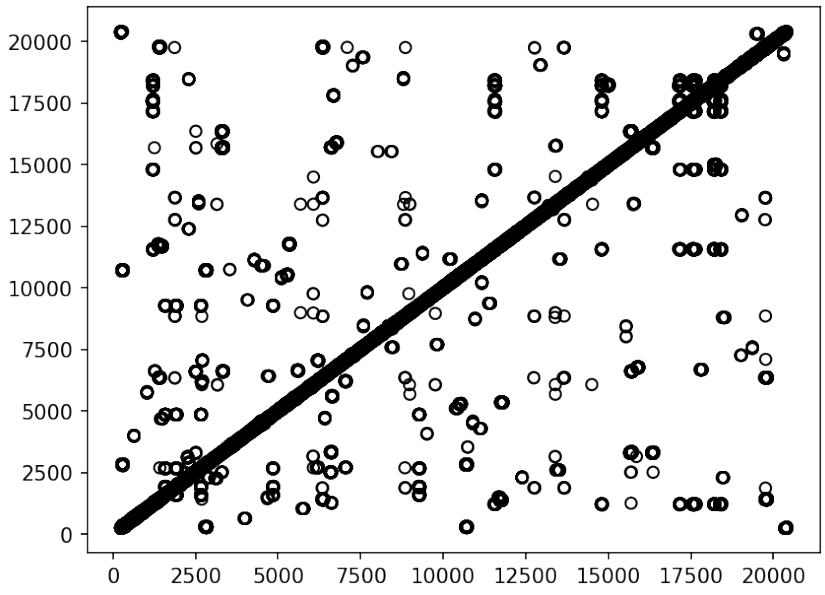
Input: Sequenced reads

1: Compute overlaps for all read pairs
2: Define similarity matrix from overlaps
3: Solve Seriation to reorder reads
4: Refine layout with overlap information
5: Compute consensus sequence by multiple sequence alignment

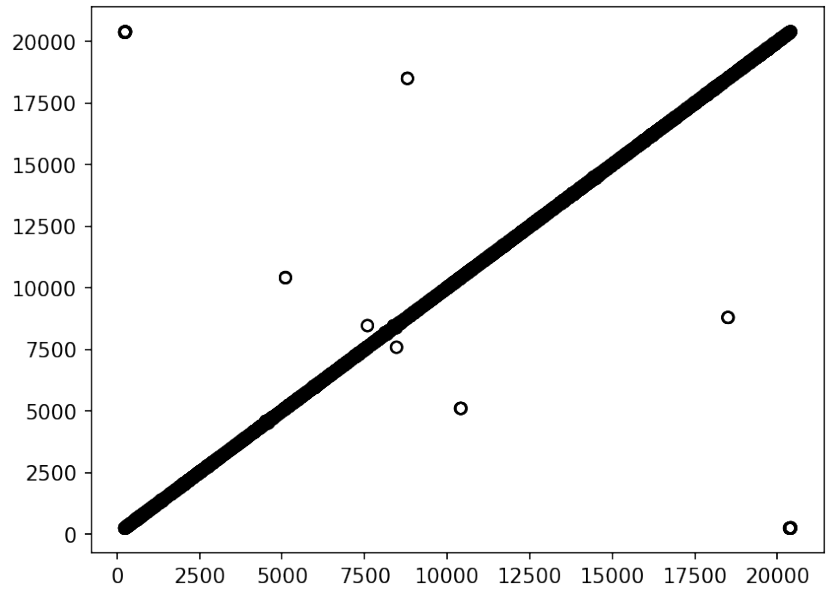
Output: Assembled sequence

Overlap-based similarity matrices

Repeats induce **false overlaps** between far-apart reads. In general shorter than "real" overlaps.



(thr=0%)

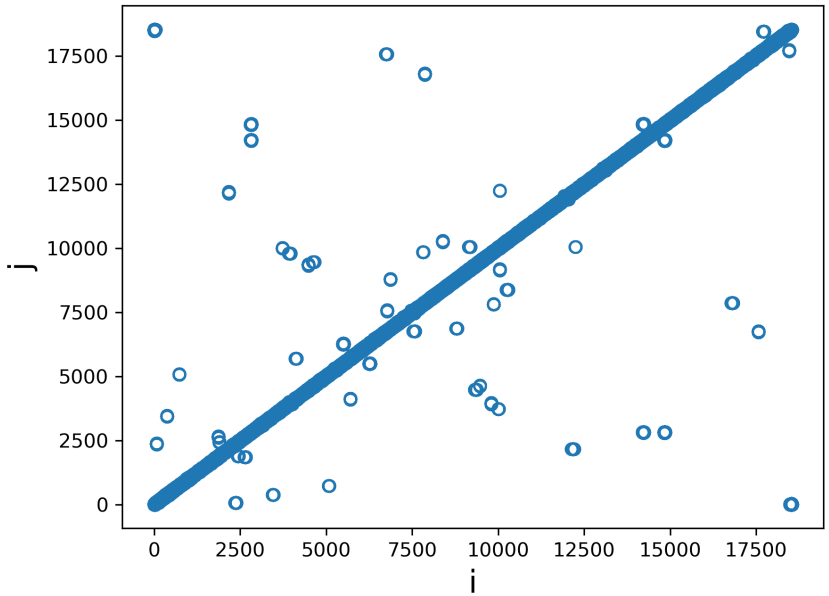


(thr=90%)

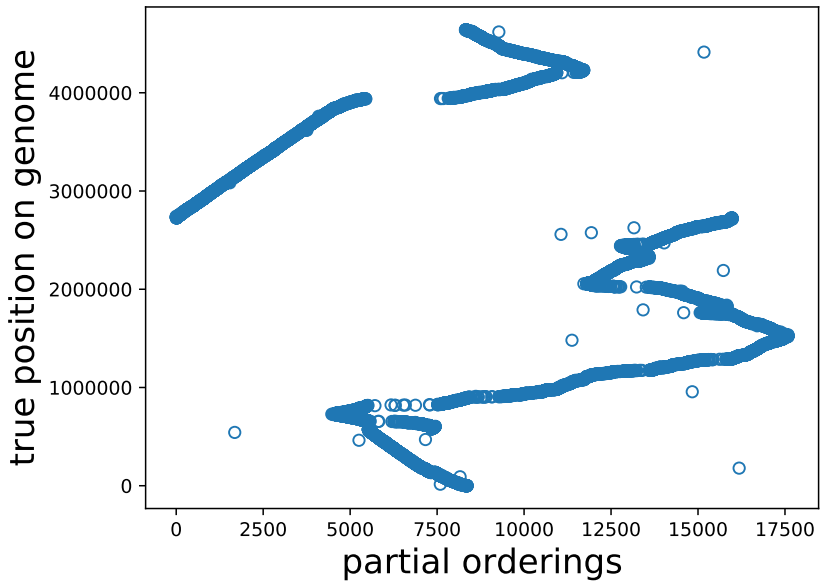
Basic spectral ordering method

Repeats make spectral method fail

$$f_{2\mathrm{SUM}} = \frac{1}{2}\sum_{i,j=1}^{n} A_{ij}(\pi_i - \pi_j)^2$$



(thr=50%)



(true vs spectral order)

Band heuristic

Keep only **large overlaps** to remove **repeat-induced** overlaps.

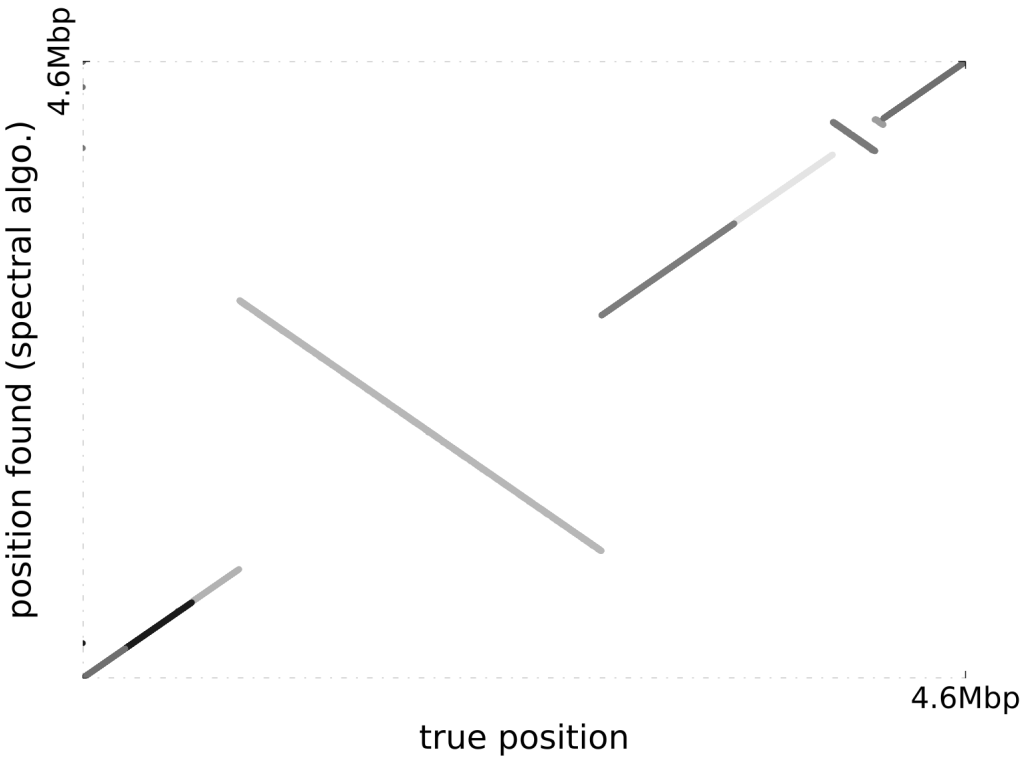
Input: Overlap-similarity matrix S

1: **for all** Connected component A of S **do**
2: Reorder A with spectral algorithm
3: **if** bandwidth of $A_{reordered} \geq 2\times$ Coverage **then**
4: increase threshold on A and try again
5: **end if**
6: Compute layout from the ordering found and overlaps
7: Derive consensus sequence (contig)
8: **end for**

Output: Contig consensus sequences

Results: ONT bacterial genomes (layout)

Yields **correct** but **fragmented** assemblies.

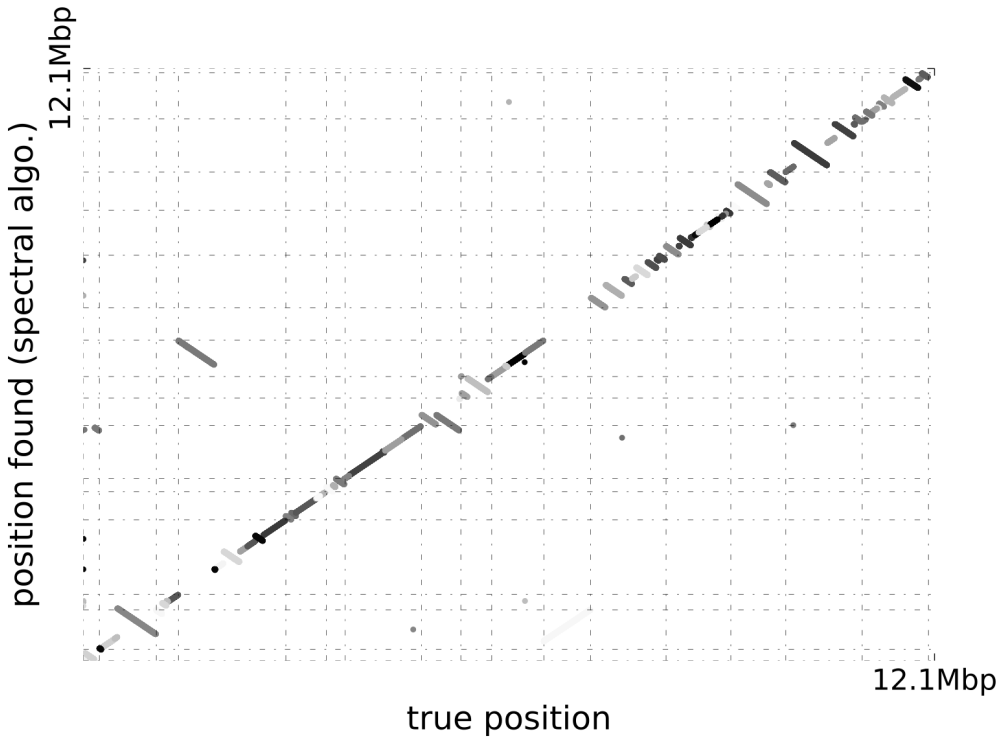


(*E. coli* ONT layout)

Contigs may overlap and be **merged** (one contig)

Results: ONT yeast genome (layout)

Even more **fragmented** assemblies. Cannot be merged into chromosome sized contigs.



(*S. cerevisiae* ONT layout)

Results (assembly quality)

Avg. identity with ref. (%) [# contigs]

	Ours	Canu	Miniasm
A. baylyi	**98.17**	97.59	87.31
E. coli	98.80	**99.40**	89.28
S. cerevisiae[1]	98.00 [71]	**98.33** [36]	89.00 **[29]**
S. cerevisiae[2]	98.81 [48]	**99.02 [26]**	93.55 [30]

[1](R7.3 chemistry, coverage 68X)

[2](R9 chemistry, coverage 86X)

Robust Seriation

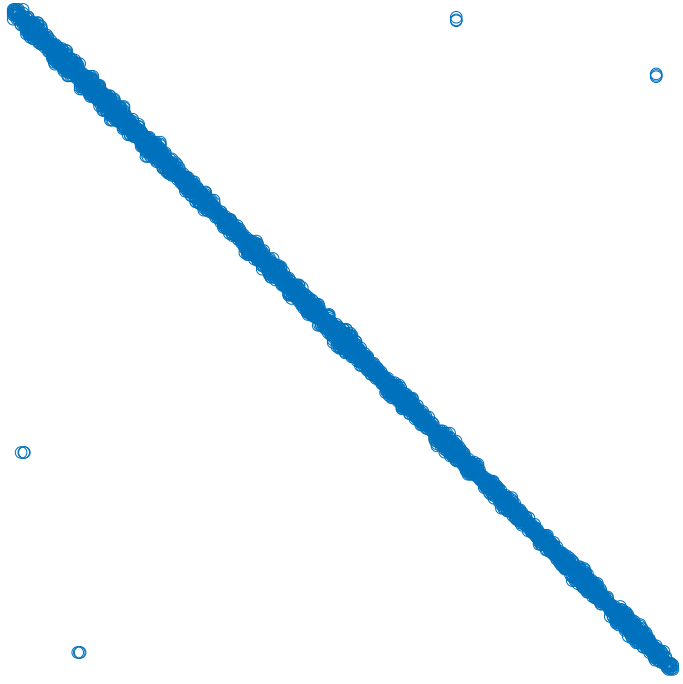
Related publication

This section is based on the following preprint:

Antoine Recanati, Nicolas Servant, Jean-Philippe Vert, and Alexandre d'Aspremont. **Robust seriation and applications to cancer genomics**. *arXiv preprint arXiv:1806.00664*, 2018.

Overlap-based similarity matrices

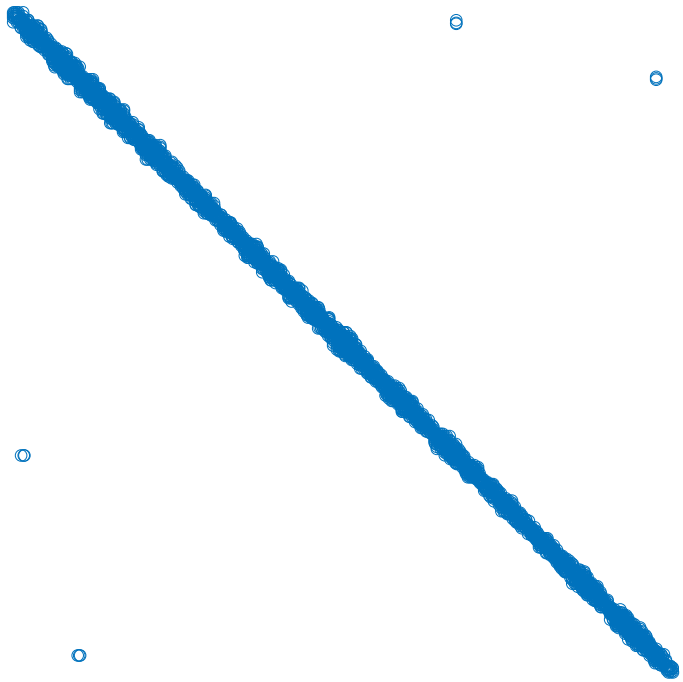
Similarity matrices arising in *de novo* assembly are the sum of a **banded matrix** (overlaps between neighboring reads) and a **sparse** matrix (repeat-induced overlaps).



Overlap-based similarity matrices

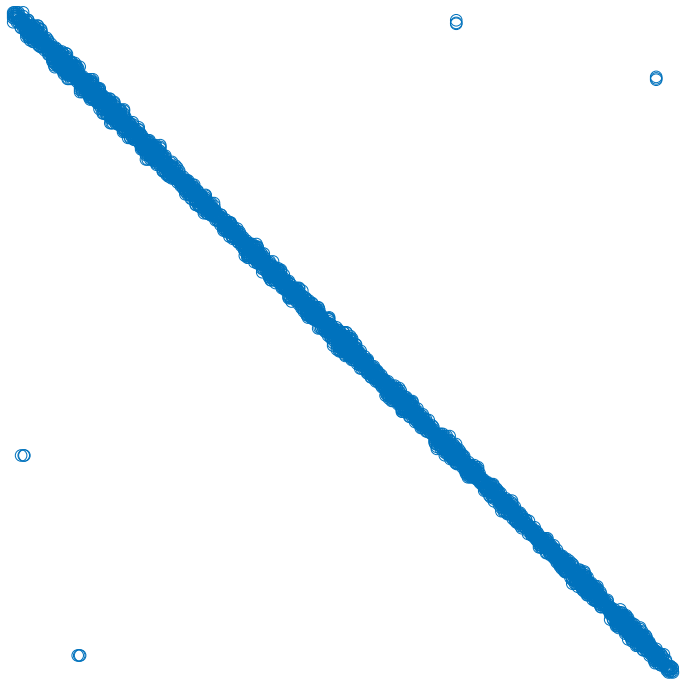
$$f_{2\text{SUM}} = \frac{1}{2} \sum_{i,j=1}^{n} A_{ij}(\pi_i - \pi_j)^2$$

The **2-SUM** function strongly penalizes **out-of-band** terms, although there are few of them.



Stylized similarity matrices

(Definition) $\mathcal{M}_n(\delta, s)$: binary matrices that are the sum of a band matrix of bandwidth δ and a sparse out-of-band matrix with s non-zero elements.



Robust Seriation

Goal: Solve **Seriation** on an approximation of the matrix yielding cleaner serial structure.

$$\begin{aligned} &\text{find} && \Pi \in \mathcal{P} \\ &\text{such that} && \Pi A \Pi^T \in \mathcal{R}. \end{aligned} \qquad \text{(Seriation)}$$

$$\begin{aligned} &\text{minimize} && \|S - \Pi A \Pi^T\| \\ &\text{such that} && \Pi \in \mathcal{P}, \quad S \in \mathcal{R}^*. \end{aligned} \qquad \text{(Robust Seriation)}$$

Robust 2-SUM

Goal: Solve **2-SUM** on an approximation of the matrix yielding cleaner serial structure.

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} S_{ij}|\pi_i - \pi_j|^2 + \lambda\|A - S\|_1 \\ \text{such that} & \pi \in \mathcal{P}, \quad S \in \mathbf{S}_+. \end{array} \qquad \text{(R2S}(\lambda)\text{)}$$

Can be re-written as:

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} A_{ij} \min(\lambda, |\pi_i - \pi_j|^2) \\ \text{such that} & \pi \in \mathcal{P}. \end{array} \qquad \text{(R2SUM}(\lambda)\text{)}$$

Robust Seriation vs. Robust 2-SUM

We proved that both problems are equivalent for stylized matrices:

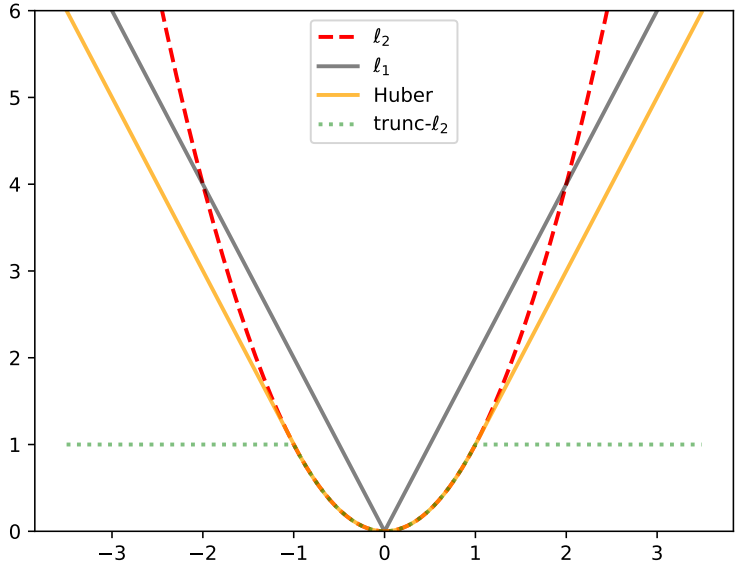
Proposition

For $s \leq s_{\lim} \triangleq (n - \delta - 1)$ *and* $A \in \mathbf{S}_n$*, if* A *can be permuted to belong to* $\mathcal{M}_n(\delta, s)$*, i.e., if there is* $\Pi \in \mathcal{P}_n : \Pi A \Pi^T \in \mathcal{M}_n(\delta, s)$*, then* Π *solves both Robust Seriation and R2SUM(*λ*) with parameter* $\lambda = \delta^2$*, and the* ℓ_1 *norm in Robust Seriation.*

Convex relaxation of Robust 2-SUM

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} A_{ij} \min(\lambda, |\pi_i - \pi_j|^2) \\ \text{such that} & \pi \in \mathcal{P}. \end{array}$$

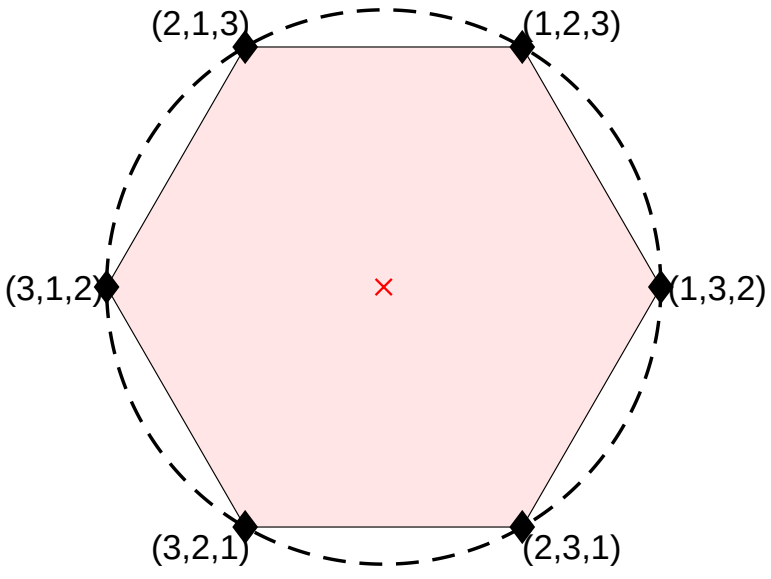
- Relax the objective: Huber-loss instead of quadratic



Convex relaxation of Robust 2-SUM

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} A_{ij} h_\lambda(|\pi_i - \pi_j|) \\ \text{such that} & \pi \in \mathcal{P}. \end{array}$$

- Relax the objective: Huber-loss instead of quadratic
- Relax set of permutation vectors

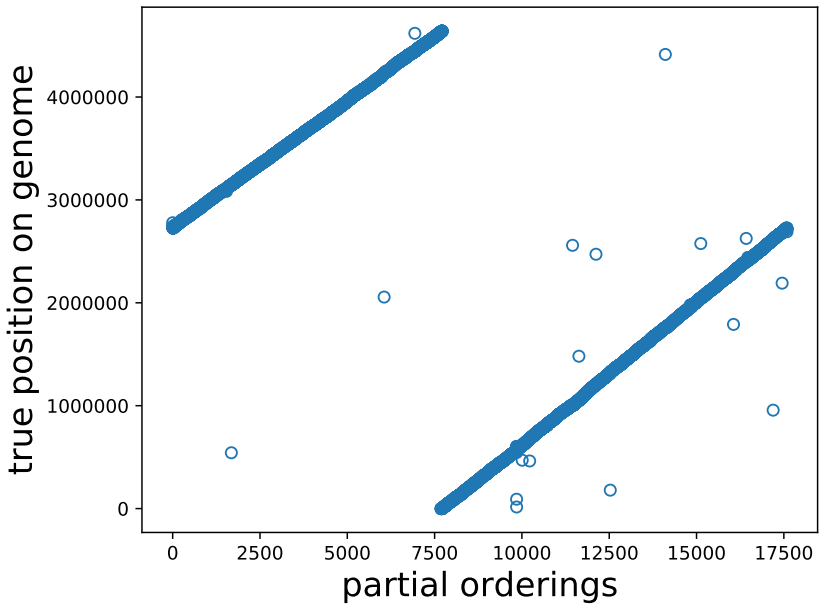


Results on synthetic $\mathcal{M}_n(\delta, s)$ matrices

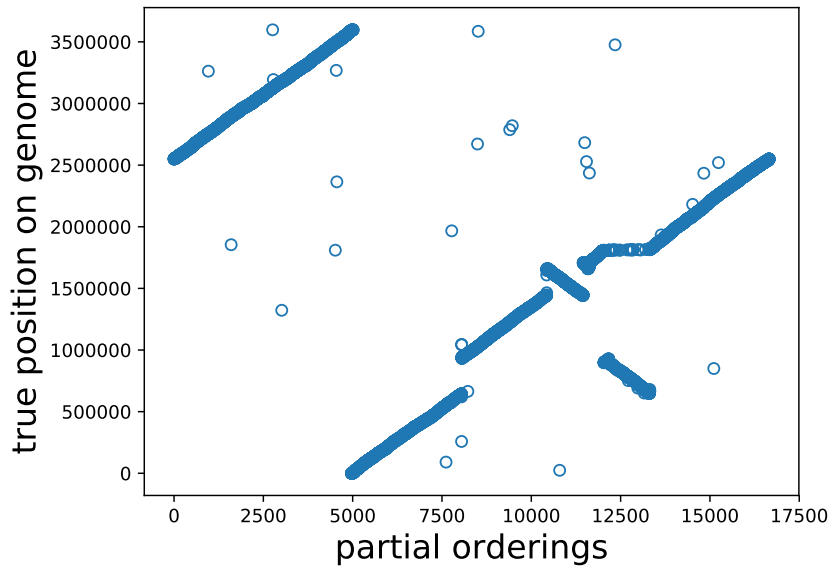
Kendall-Tau scores

s/s_{lim}:	2.5	5	7.5	10
spectral	0.91	0.86	0.84	0.80
HGnCR	**0.99**	0.89	0.85	0.83
η-Spectral	0.98	**0.97**	**0.96**	**0.94**
H-UBI	0.98	**0.97**	**0.96**	**0.94**

Results on ONT bacterial genomes



(E. coli)



(A. baylyi)

Multi-dimensional spectral ordering

Related publication

This section is based on the following preprint:

Antoine Recanati, Thomas Kerdreux, and Alexandre d'Aspremont. **Reconstructing latent orderings by spectral clustering**. *arXiv preprint arXiv:1807.07122*, 2018.

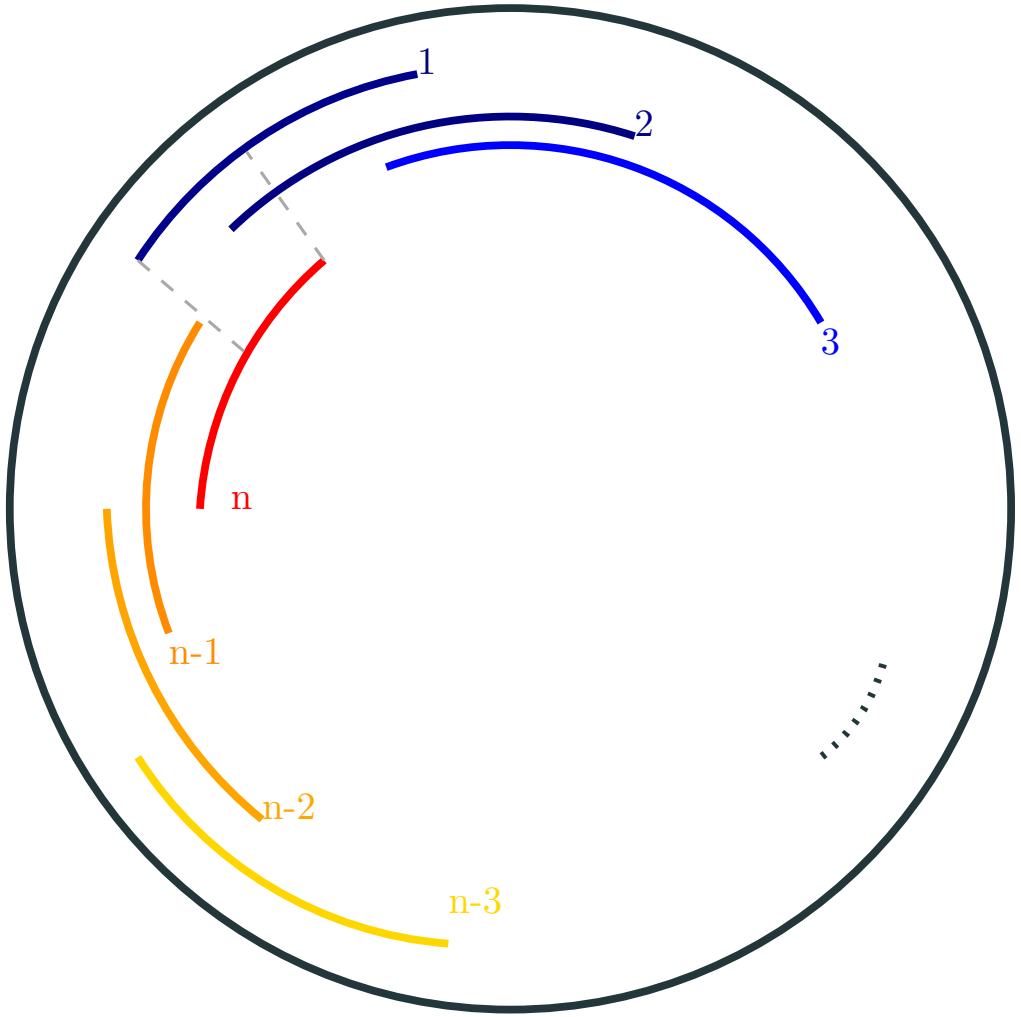
`https://github.com/antrec/mdso`

Spectral method for Seriation

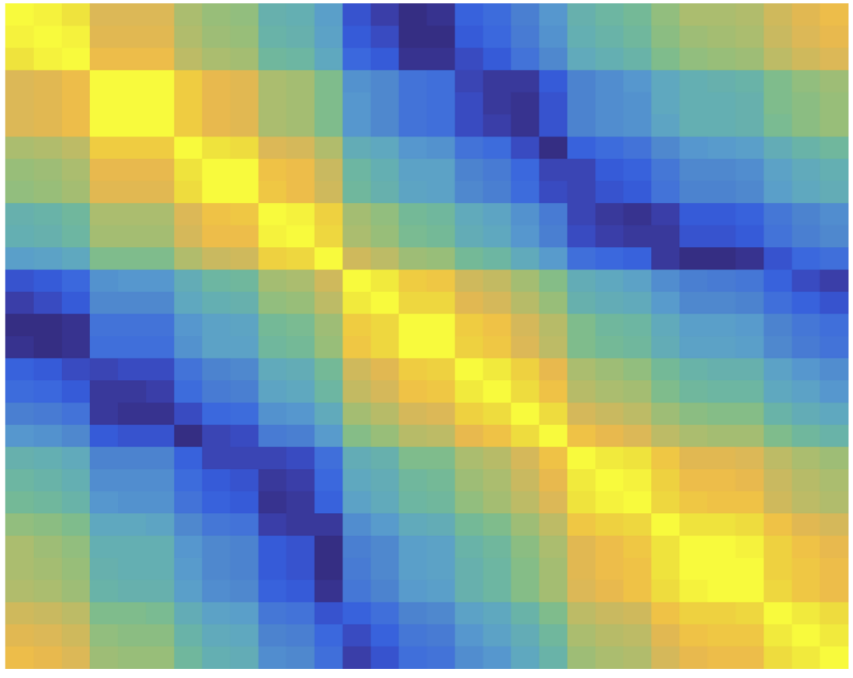
- Scalable (hence, widely used by practitioners)
- Theoretical guarantee to solve Seriation in noiseless setting
- Sentitive to noise in practice
- Limited to linear orderings

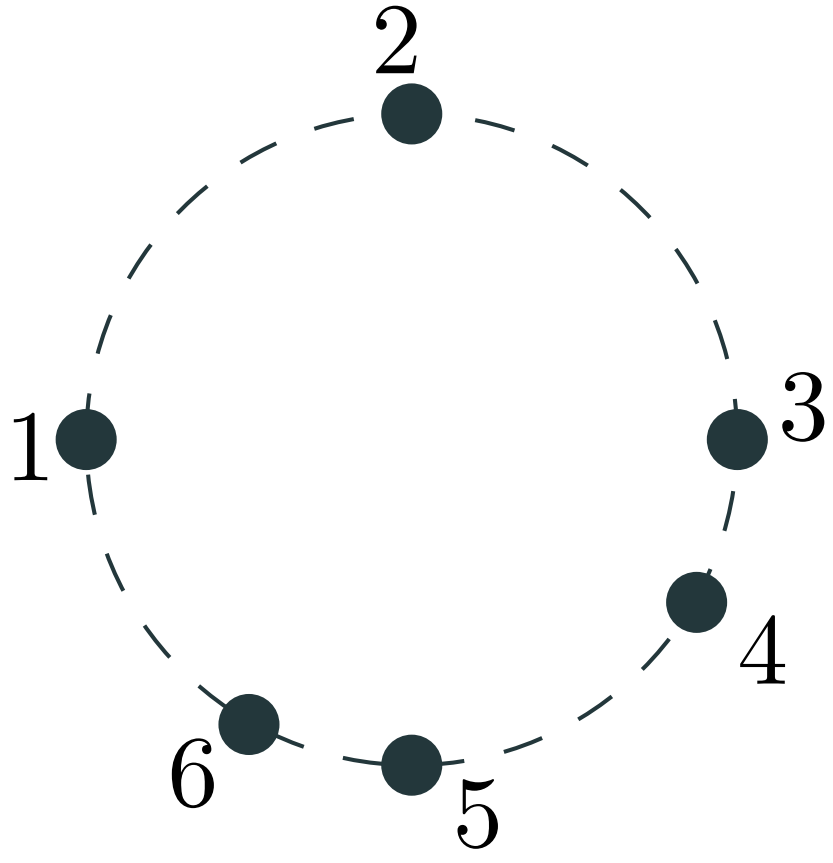
Bacterial genomes are circular

Bacterial genomes are circular



Circular-Robinson matrices





- Analog of Linear Seriation results ?

Laplacian embedding

Generalize Spectral relaxation of **2-SUM** with multiple **dimensions** (d)

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} A_{ij} \left(x_i - x_j\right)^2 = x^T L x \\ \text{such that} & x \in \mathbb{R}^n \\ & x^T \mathbf{1} = 1, \\ & \|x\|_2^2 = 1. \end{array}$$

Laplacian embedding

Generalize Spectral relaxation of **2-SUM** with multiple **dimensions** (d)

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} A_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \mathbf{Tr}(X^T L X) \\ \text{such that} & X \in \mathbb{R}^{n\times d} \\ & X^T\mathbf{1}_n = \mathbf{1}_d, \\ & X^T X = \mathbf{I}_d. \end{array} \tag{2}$$

Laplacian embedding

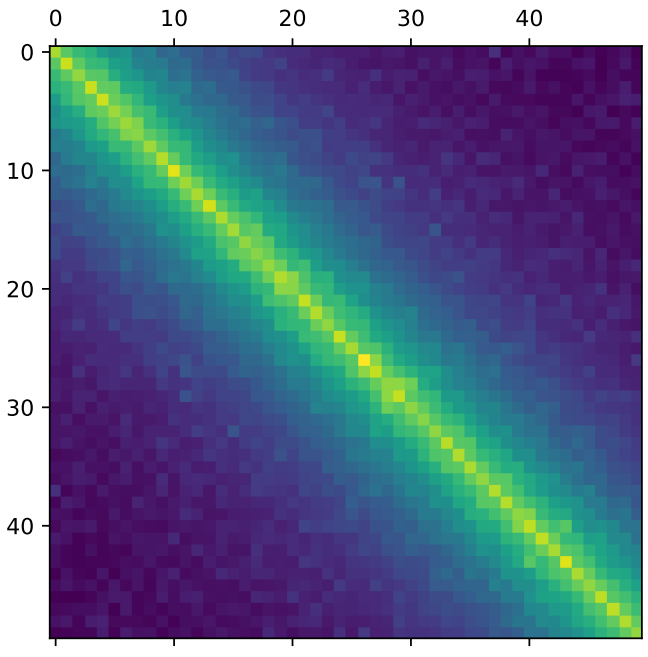
Let $\Phi = \left(\mathbf{1}, f_{(1)}, \ldots, f_{(n-1)}\right)$ eigenvectors of L.
For $d < n$, $\Phi^{(d)} \triangleq \left(f_{(1)}, \ldots, f_{(d)}\right)$ is a d-dim. embedding:

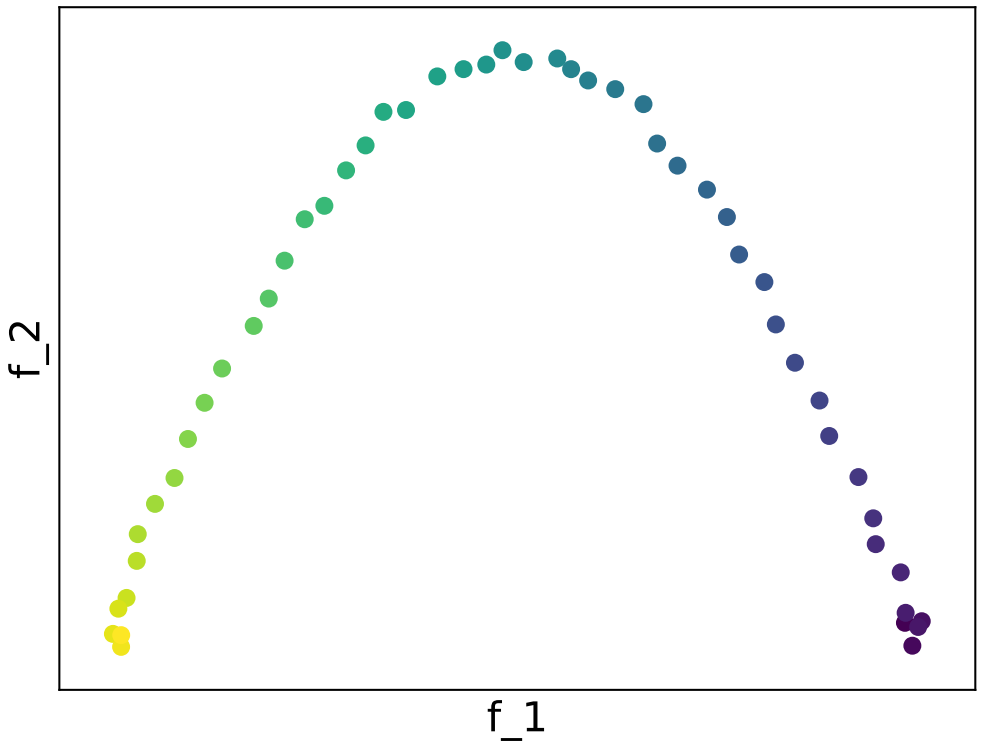
$$\mathbf{x}_i = \left(f_{(1)}(i), f_{(2)}(i), \ldots, f_{(d)}(i)\right)^T \in \mathbb{R}^d \qquad \text{(d-LE)}$$

The **Laplacian embedding** solves the **multi-dimensional 2-SUM problem** (2).

Laplacian embedding for R-matrices

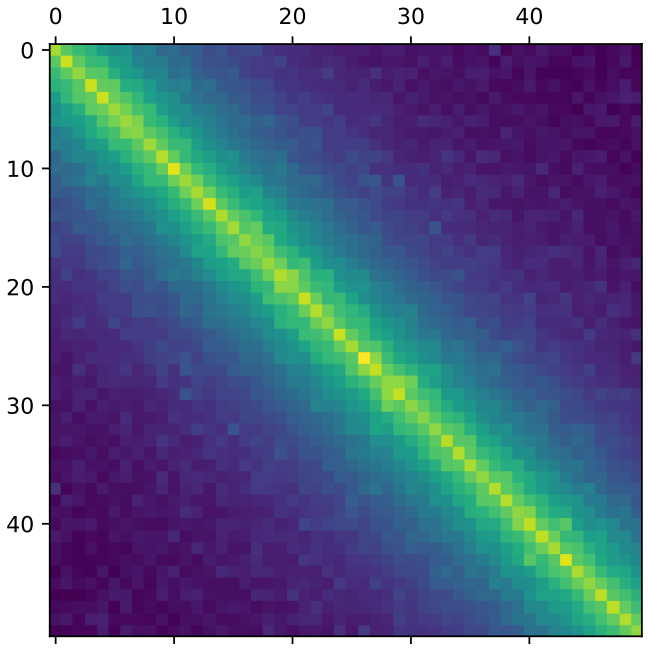
Observation: Laplacian embedding: 1d manifold

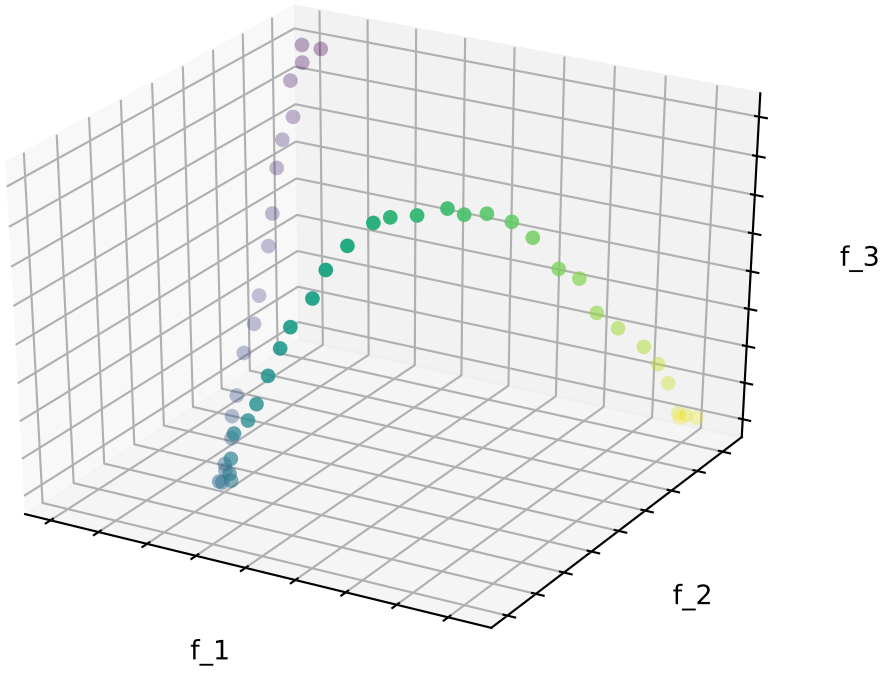




Laplacian embedding for R-matrices

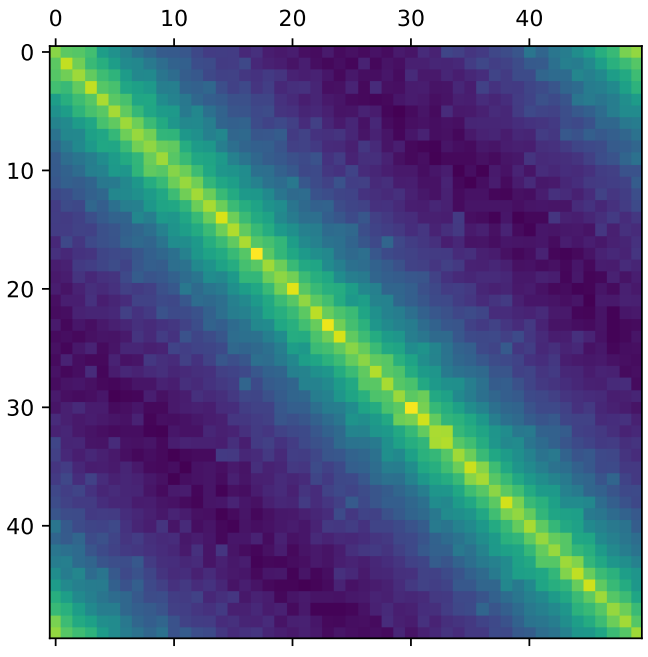
Observation: Laplacian embedding: 1d manifold

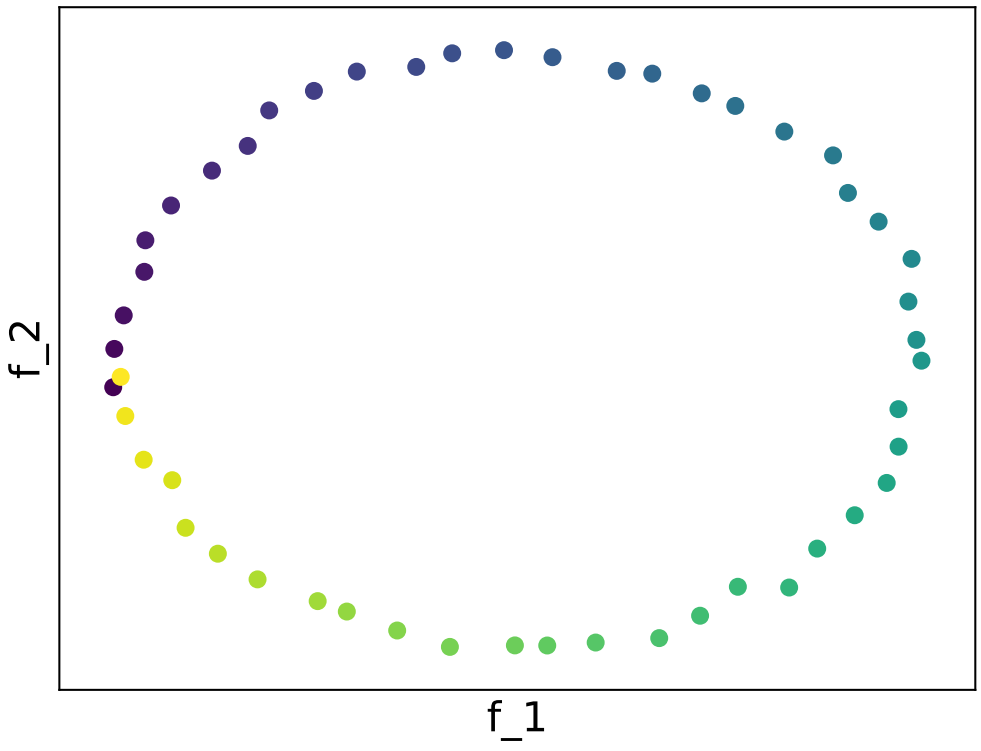




Laplacian embedding for circular R-matrices

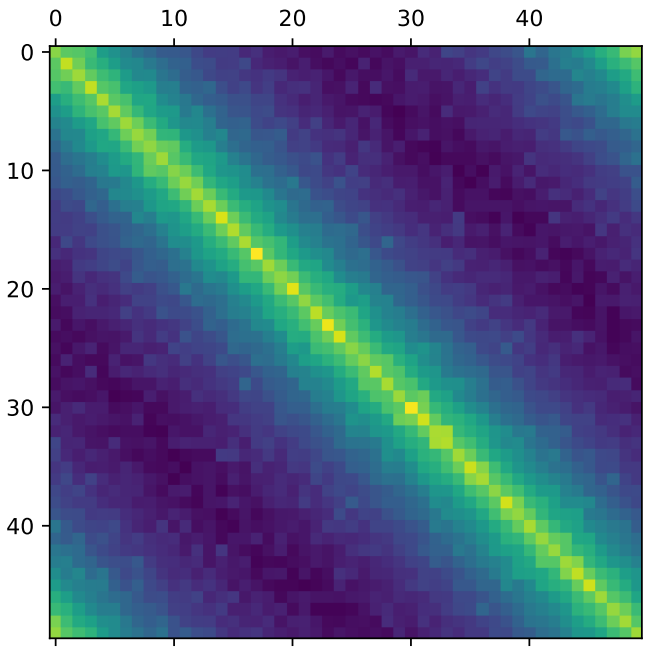
Observation: Laplacian embedding: **closed** 1d manifold

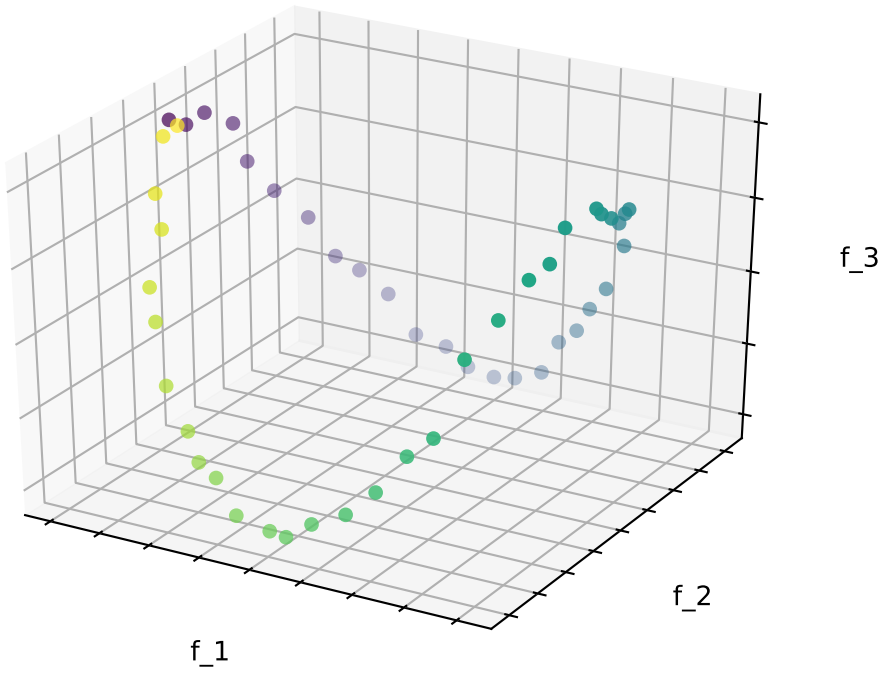




Laplacian embedding for circular R-matrices

Observation: Laplacian embedding: **closed** 1d manifold





In theory ?

Only **asymptotical** results with stronger assumptions on the data than in linear Seriation results.

- Toeplitz (circular) R matrices converge towards an operator whose eigenfunctions are harmonic (frequency increases with eigenvalues).
- No result for n finite

Spectral circular ordering method [1]

Input: Connected similarity matrix A

1: Compute Laplacian L

2: Compute the two first non-trivial eigenvectors of L, (f_1, f_2)

3: Sort the values of $\theta(i) \triangleq \tan^{-1}(f_2(i)/f_1(i)) + \mathbb{1}[f_1(i) < 0]\pi$

Output: Permutation $\sigma : \theta(\sigma(1)) \leq \ldots \leq \theta(\sigma(n))$

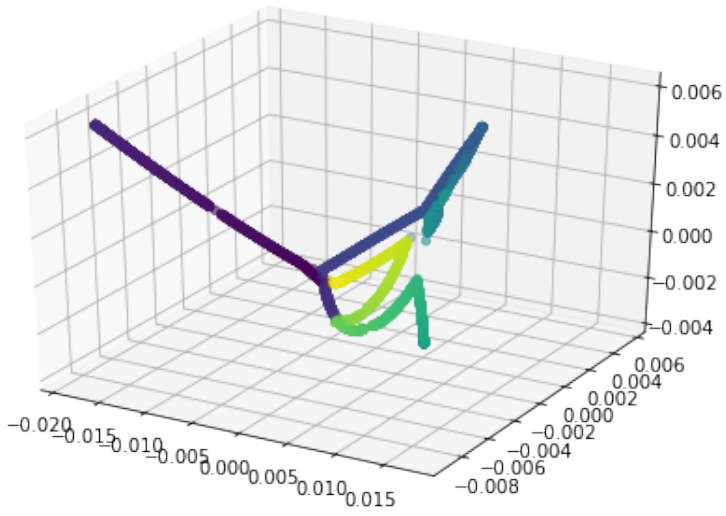
Theoretical guarantee for circular seriation

We proved the following non-asymptotical result, analog to linear seriation.

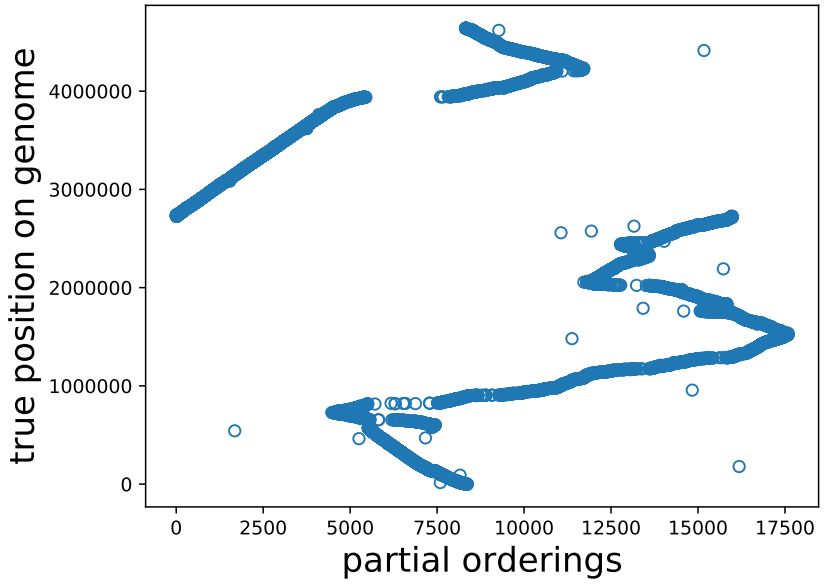
Proposition (Circular Spectral Seriation)

For Toeplitz, circulant R-matrices, the previous 2d Spectral ordering algorithm solves Circular Seriation.

3-LE of overlap similarity matrix (*E. coli*)



(3d)



(1d: spectral ordering)

Multi-dimensional spectral ordering

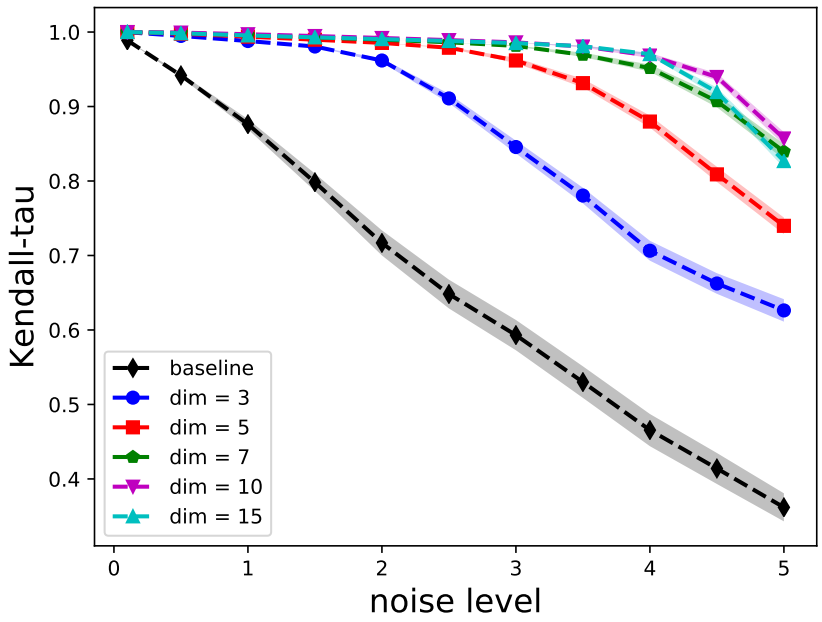
Claim: the latent ordering of points is easier to recover with multi-dimensional embeddings in noisy settings.

Input: Similarity matrix

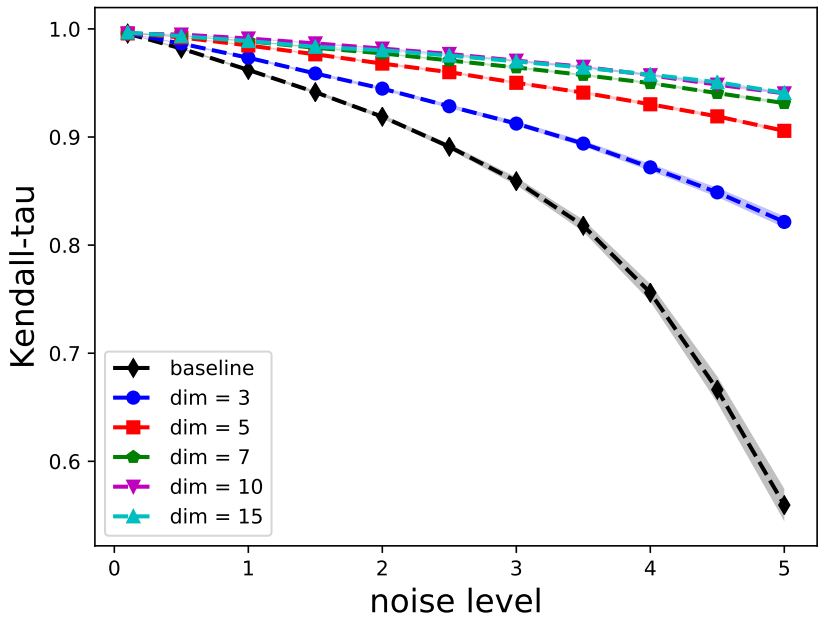
1: Compute d-LE
2: For all points, locally fit NNs by a line
3: Use projections on line to define new pairwise distance
4: Solve Seriation on the new matrix

Output: Ordering

Results on synthetic, noisy data



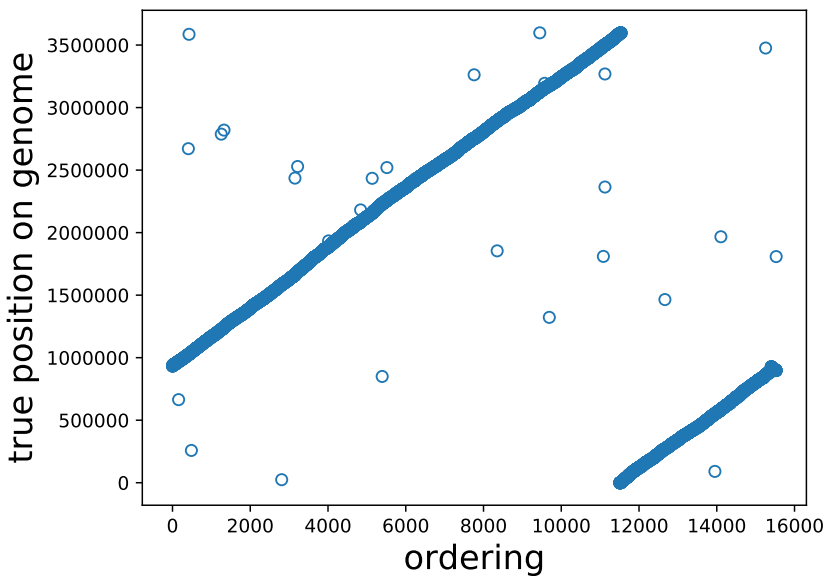
(linear)



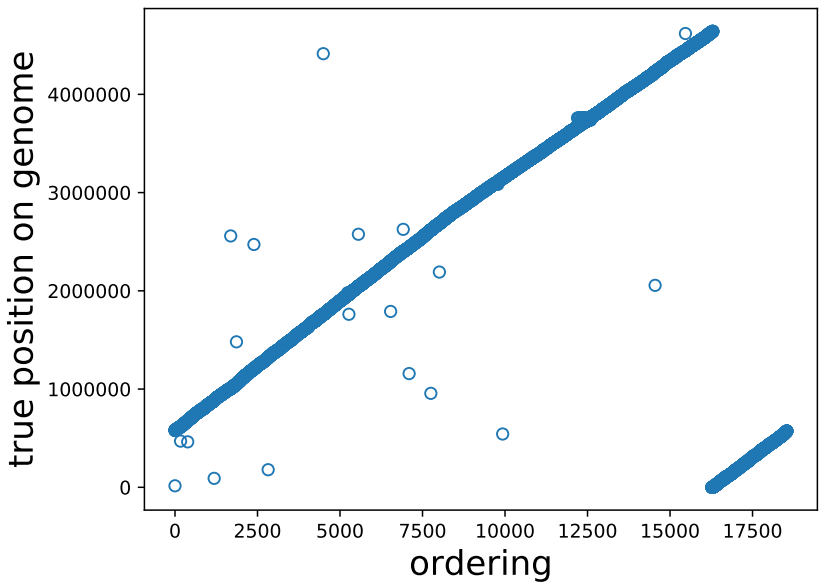
(circular)

Results on ONT bacterial genomes

Successfully reorders the reads



(A. baylyi)



(E. coli)

Conclusion

Summary

- Seriation: clean **mathematical** framework for layout in OLC
- Competitive **in practice**, although **challenged by repeats**
- Robust Seriation: essentially going from ℓ_2 to Huber loss.
- **Robust Seriation** increases robustness **in practice** but repeats remain challenging
- **Multi-dimensional Spectral Ordering**: simple extension of spectral baseline method, significant gains, notably in *de novo* context.

Extensions

- Seriation with Duplications: motivated by assembly of genomes with structural variants from Hi-C data. Related to Robust Seriation framework.
- Take multiple chromosomes into account ?