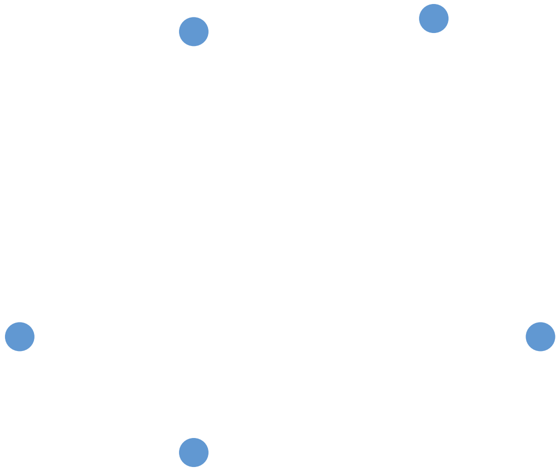
HyperSteiner: Computing Heuristic Hyperbolic Steiner Minimal Trees

A. García Castellanos*, A. A. Medbouhi*, G. L. Marchetti, E. J. Bekkers, D. Kragic

siam 2025
SIAM Symposium on Algorithm Engineering and Experiments

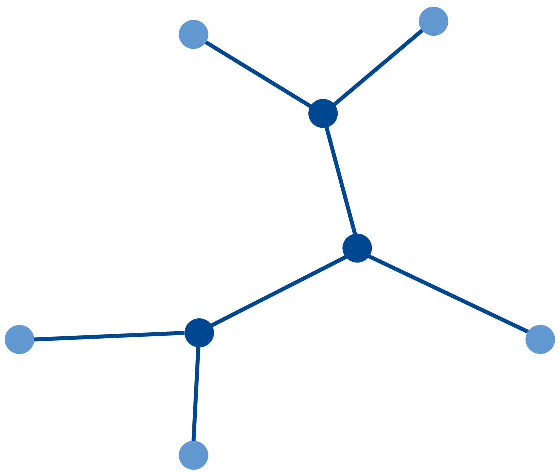
Steiner Minimal Trees

Problem: given a set of n points P in the plane,



Steiner Minimal Trees

Problem: given a set of n points P in the plane, find the set S of points and the tree with vertices P ∪ S minimizing the total edge length.



Applications

SMTs are crucial for (optimal) network design.

Hardness and Heuristics

The problem of obtaining the *exact* SMT is **NP-Hard**

→

Obtain approximate solutions using heuristics

Hardness and Heuristics

The problem of obtaining the *exact* SMT is **NP-Hard**

→

Obtain approximate solutions using heuristics

The **algorithm by Smith et al.** computes *suboptimal* Steiner Minimal Trees by a *divide-and-conquer* approach:

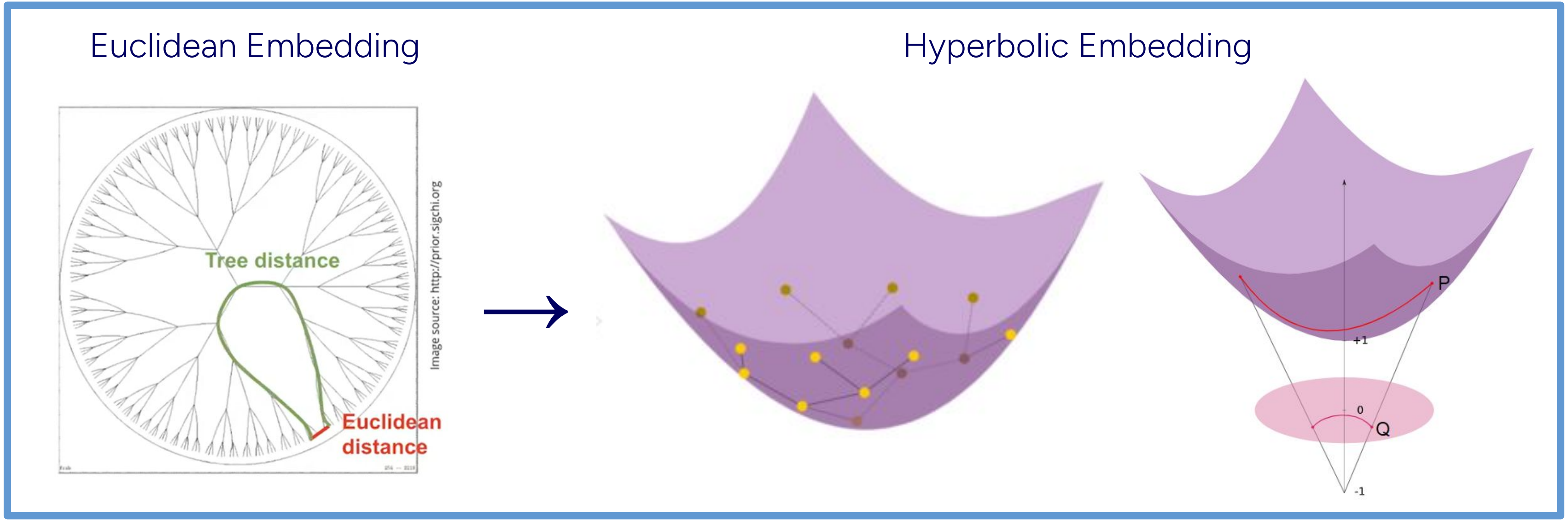
1. Reducing the problem to a local one via the Delaunay triangulation
2. Concatenating local Full Steiner Trees (FSTs)

Non-Euclidean SMTs

The SMT problem can be defined on any Riemannian manifold. We focus on the hyperbolic space,

Non-Euclidean SMTs

The SMT problem can be defined on any Riemannian manifold. We focus on the hyperbolic space, since its exponential growth is suitable for representing hierarchical data.



Src: http://summergeometry.org/sgi2021/embedding-hierarchical-data-in-hyperbolic-geometry/

Goals

- Adapt the heuristic method for computing SMTs by Smith et al. to the hyperbolic space
- Apply the method to hierarchy discovery

Smith-Lie-Liebman algorithm

The **algorithm by Smith et al.** computes *suboptimal* Steiner Minimal Trees by a *divide-and-conquer* approach:

1. Reducing the problem to a local one via the Delaunay triangulation
2. Concatenating local Full Steiner Trees (FSTs)

It has O(n log(n)) computational complexity 🚀

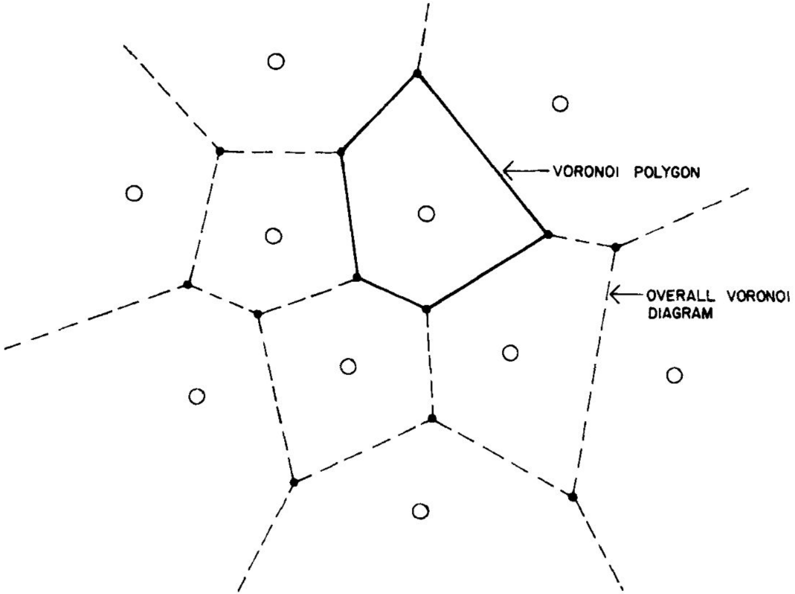
Algorithm 1 SLL Algorithm

Input: Terminals $P \subset \mathbb{R}^2$.

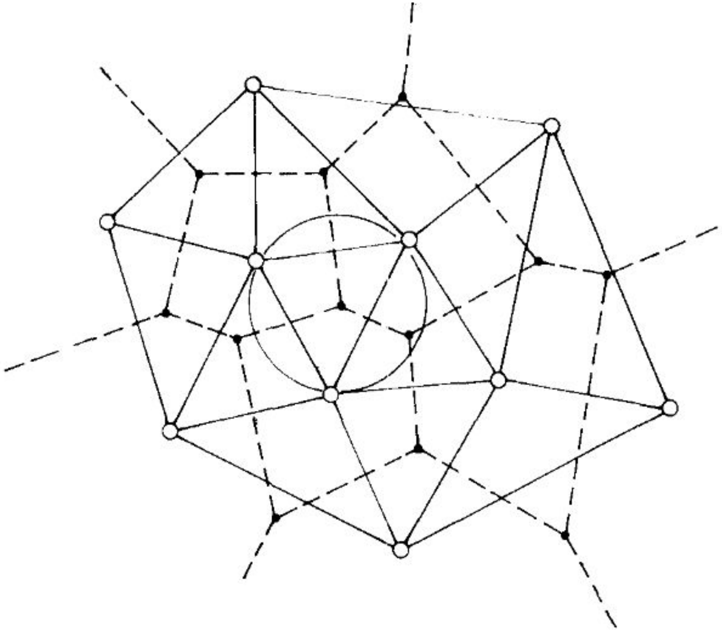
1. Construct the Delaunay triangulation, $\mathrm{DT}(P)$.
2. Construct $\mathrm{MST}(P)$ (Kruskal algorithm) and simultaneously build a priority queue as follows:
   - 2.1. Mark all the triangles $\sigma \in \mathrm{DT}(P)$ containing two edges of $\mathrm{MST}(P)$ and admitting an FST.
   - 2.2. Place the FSTs of marked triangles σ in a queue Q prioritized on $\rho(\sigma)$ (smaller first).
3. Add the FST of the 4-terminal subsets:
   - 3.1. For each marked triangle σ, find its adjacent triangles σ' such that σ and σ' contain three edges of the $\mathrm{MST}(P)$.
   - 3.2. Compute the FST $\sigma \cup \sigma'$ for each of the two possible topologies and add the minimal one to Q.
4. Convert Q to an ordered list and append to it the edges of $\mathrm{MST}(P)$, sorted in non-decreasing order.
5. Let T be an empty tree. An FST in Q is added to T if it does not create a cycle (greedy concatenation).

Output: T – a heuristic SMT of P.

SLL: Delaunay triangulation

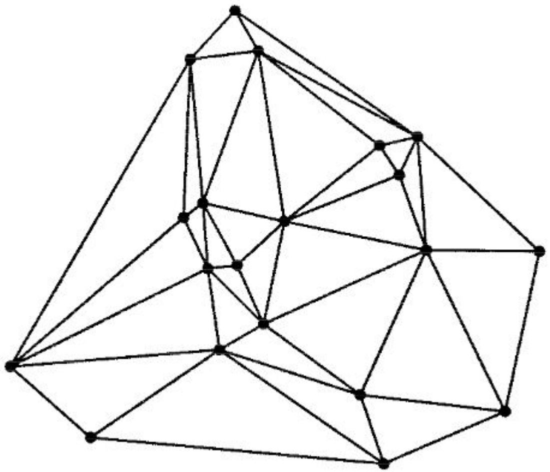


Voronoi diagram



Delaunay triangulation

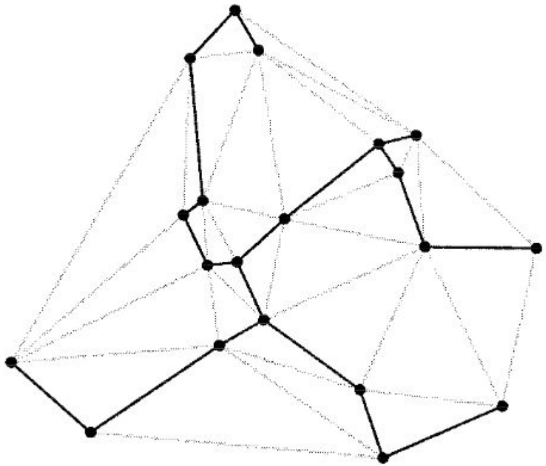
SLL: Minimum Spanning Tree



Delaunay triangulation

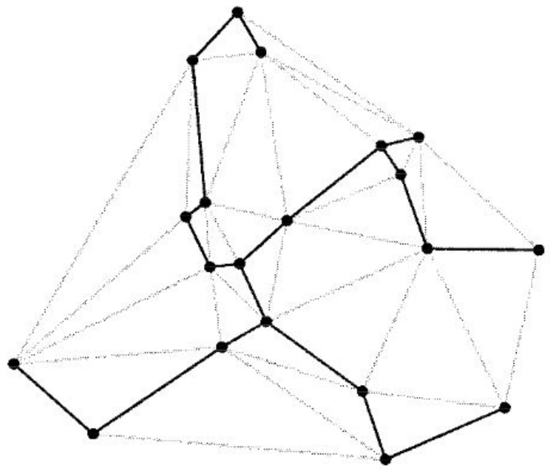
Kruskal

$\Rightarrow$



MST

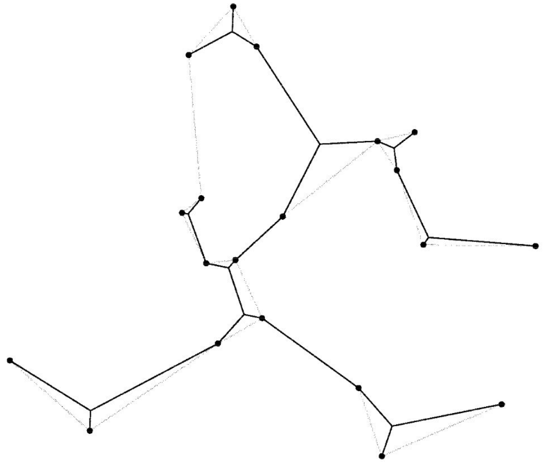
SLL: Heuristic Steiner Minimal Tree



MST

3 and 4 node
Full Steiner Trees

⇒



SMT

Method: Adapting Smith et al.

Goes from local to global solution via Delaunay triangles
⇒ Adaptation from Euclidean to hyperbolic needs:

Method: Adapting Smith et al.

Goes from local to global solution via Delaunay triangles
⇒ Adaptation from Euclidean to hyperbolic needs:

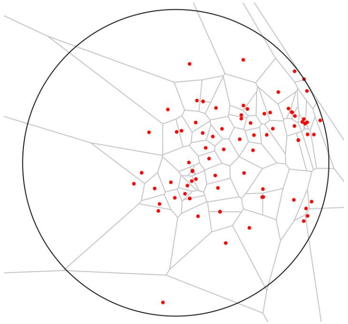
Hyperbolic Voronoi Diagram

Hyperbolic Delaunay Triangulation

3 and 4-point Full Steiner Trees in a Hyperbolic model

Method: Adapting Smith et al.

Goes from local to global solution via Delaunay triangles
⇒ Adaptation from Euclidean to hyperbolic needs:

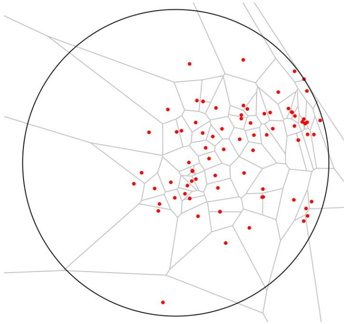


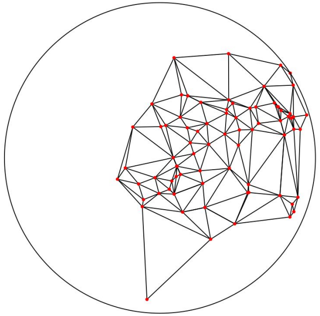
Hyperbolic Delaunay Triangulation

3 and 4-point Full Steiner Trees in a Hyperbolic model

Method: Adapting Smith et al.

Goes from local to global solution via Delaunay triangles
⇒ Adaptation from Euclidean to hyperbolic needs:

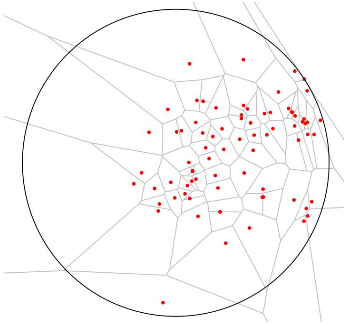


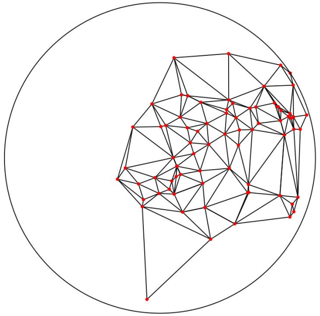


3 and 4-point Full Steiner Trees in a Hyperbolic model

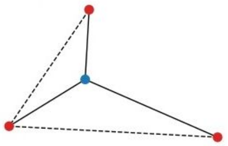
Method: Adapting Smith et al.

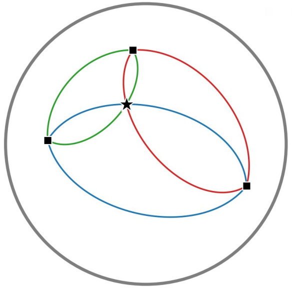
Goes from local to global solution via Delaunay triangles
⇒ Adaptation from Euclidean to hyperbolic needs:





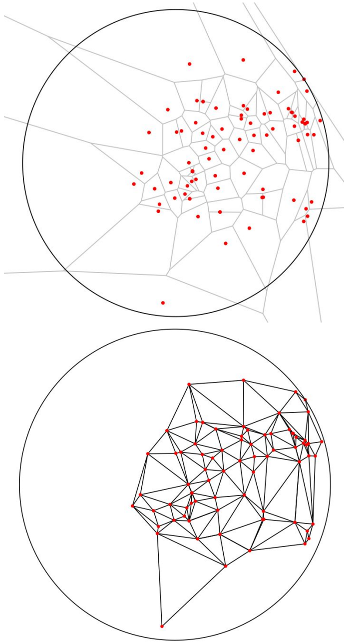
<u>3-points</u>
System of algebraic equations



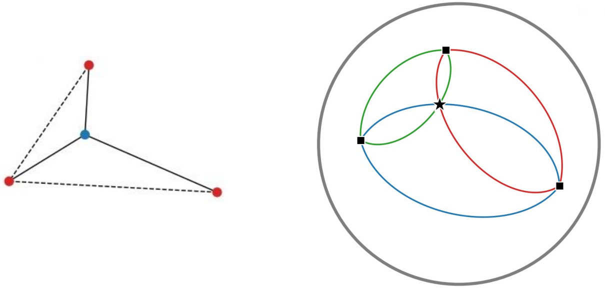


Method: Adapting Smith et al.

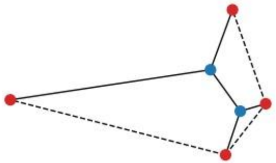
Goes from local to global solution via Delaunay triangles
⇒ Adaptation from Euclidean to hyperbolic needs:



3-points
System of algebraic equations



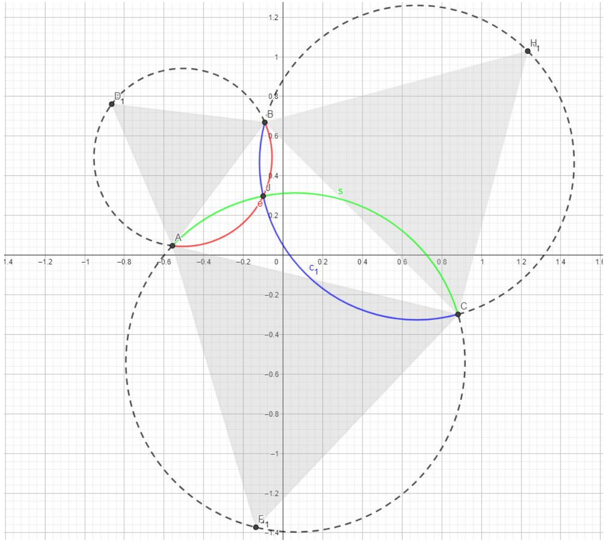
4-points
Approximated recursive algorithm



Method: Solving the 3-point case with isoptic curves

In the 3-point case the Steiner point (if exists) is the locus where all segments of the triangle $\triangle ABC$ are seen under 120° *(i.e., intersection of isoptic curves)*

Euclidean Case (aka Fermat point)



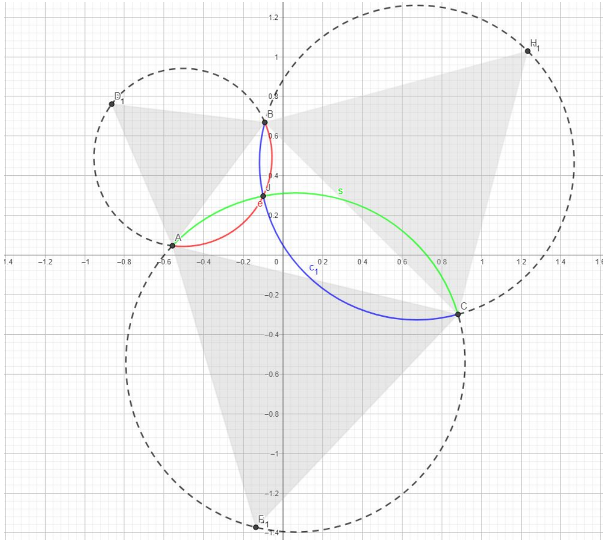
→

Hyperbolic Klein model

Method: Solving the 3-point case with isoptic curves

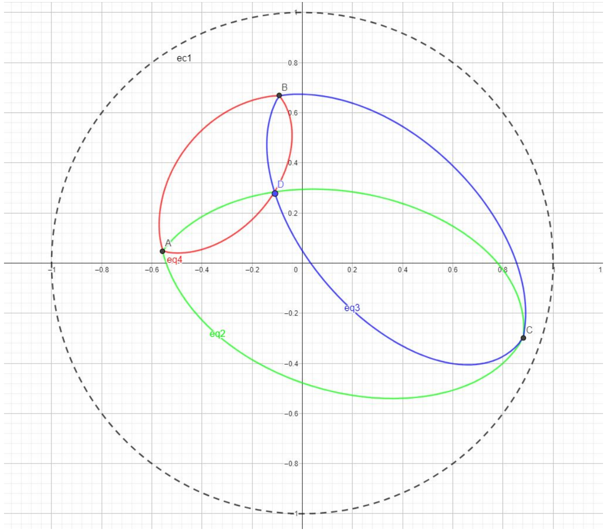
In the 3-point case the Steiner point (if exists) is the locus where all segments of the triangle △ABC are seen under 120° *(i.e., intersection of isoptic curves)*

Euclidean Case (aka Fermat point)



→

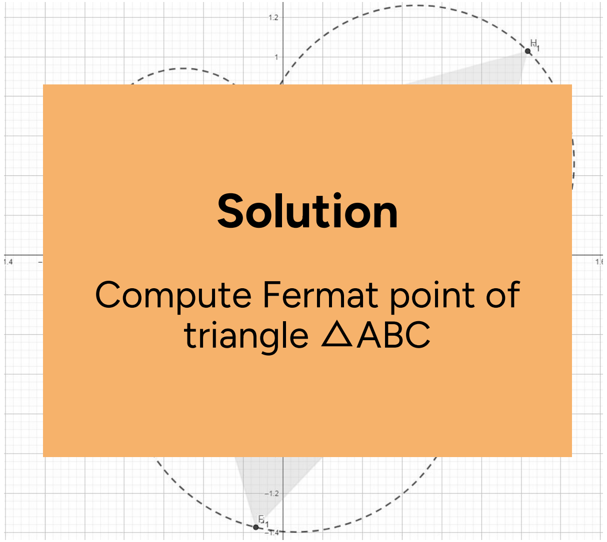
Hyperbolic Klein model



Method: Solving the 3-point case with isoptic curves

In the 3-point case the Steiner point (if exists) is the locus where all segments of the triangle $\triangle ABC$ are seen under 120° *(i.e., intersection of isoptic curves)*

Euclidean Case (aka Fermat point)

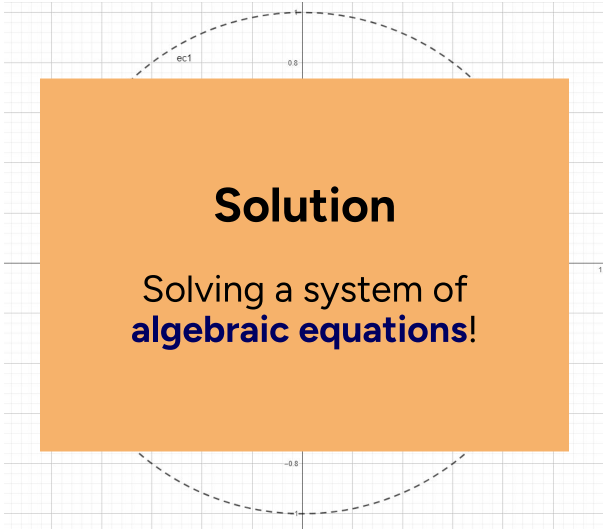


Solution

Compute Fermat point of triangle $\triangle ABC$

→

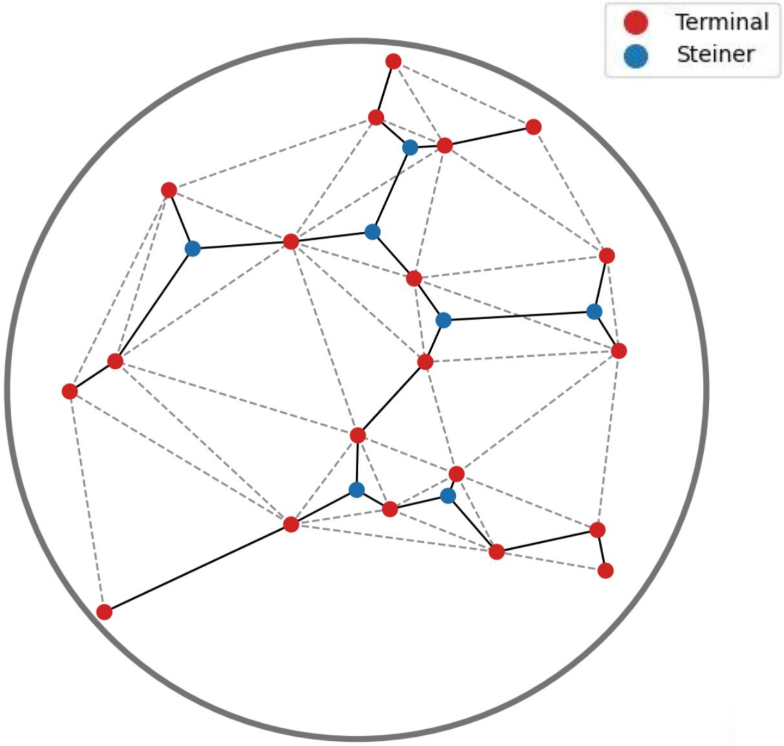
Hyperbolic Klein model



Solution

Solving a system of **algebraic equations**!

Example



Steiner ratio

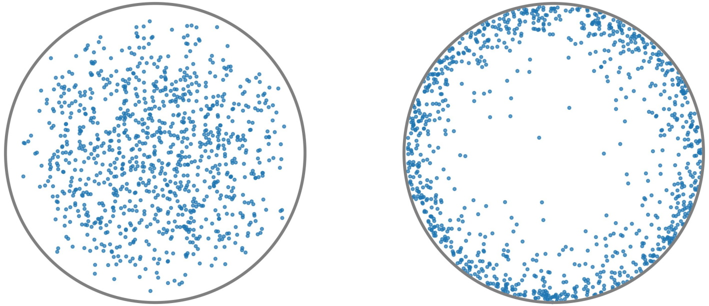
$$\frac{\text{Steiner tree}}{\text{Minimum Spanning Tree}} = 94\%$$

Experiments

Method validation on synthetic data sampled from different distributions

Experiments

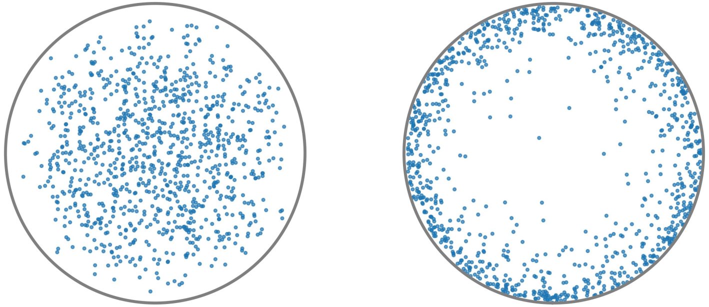
Method validation on synthetic data sampled from different distributions



Experiments

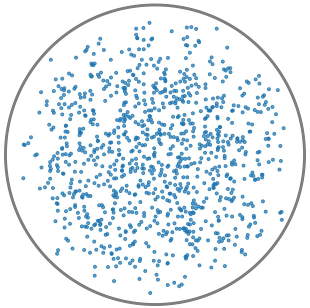
Method validation on synthetic data sampled from different distributions

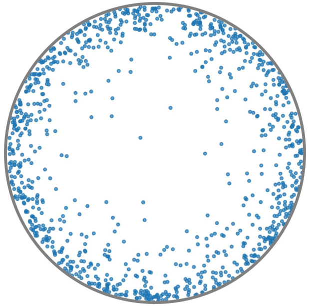
Application to hierarchy discovery on real biological data



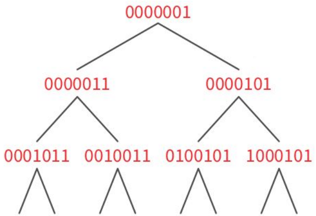
Experiments

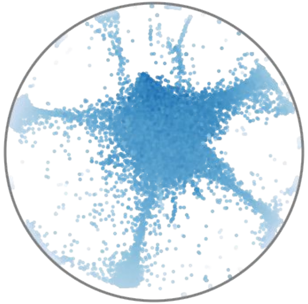
Method validation on synthetic data sampled from different distributions



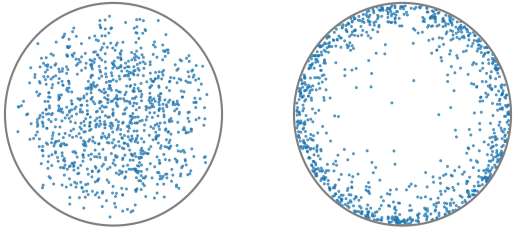


Application to hierarchy discovery on real biological data

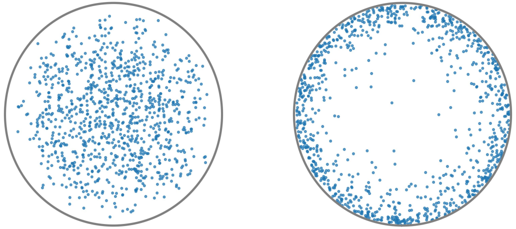




Results for validation on synthetic data

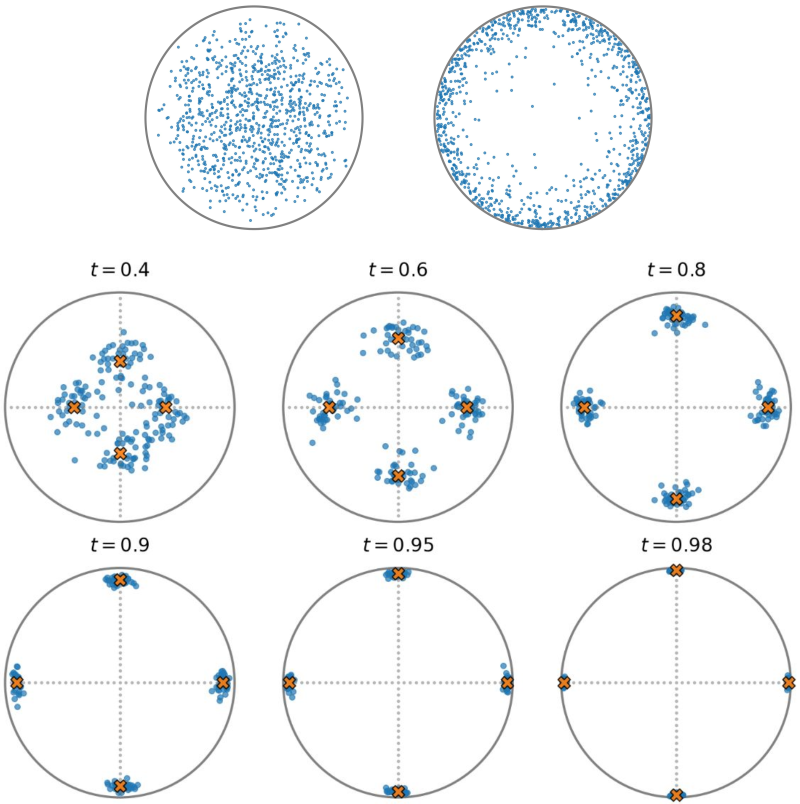


Results for validation on synthetic data



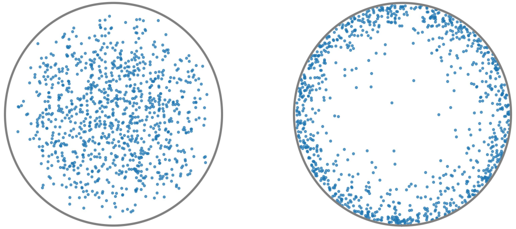
1) HyperSteiner leads to 2-3% improvement over the MST in standard scenarios, similarly to the Euclidean case.

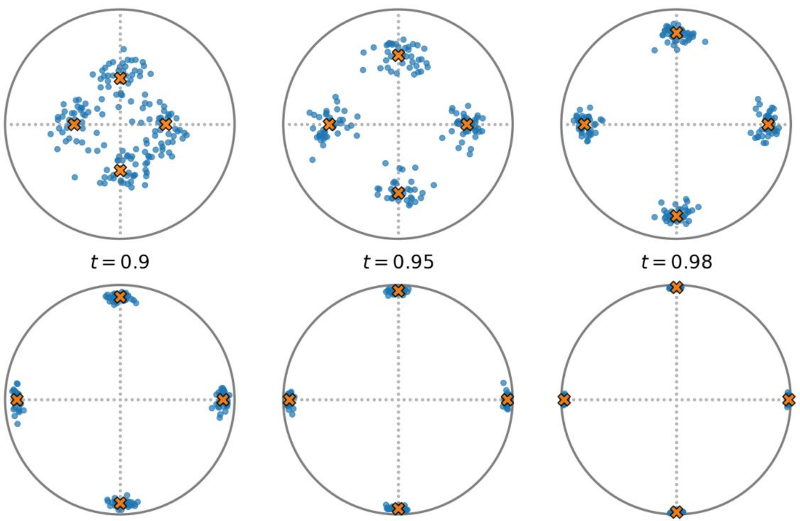
Results for validation on synthetic data



1) HyperSteiner leads to 2-3% improvement over the MST in standard scenarios, similarly to the Euclidean case.

Results for validation on synthetic data

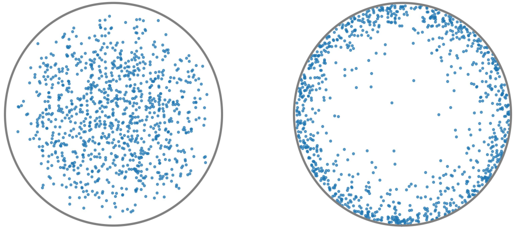


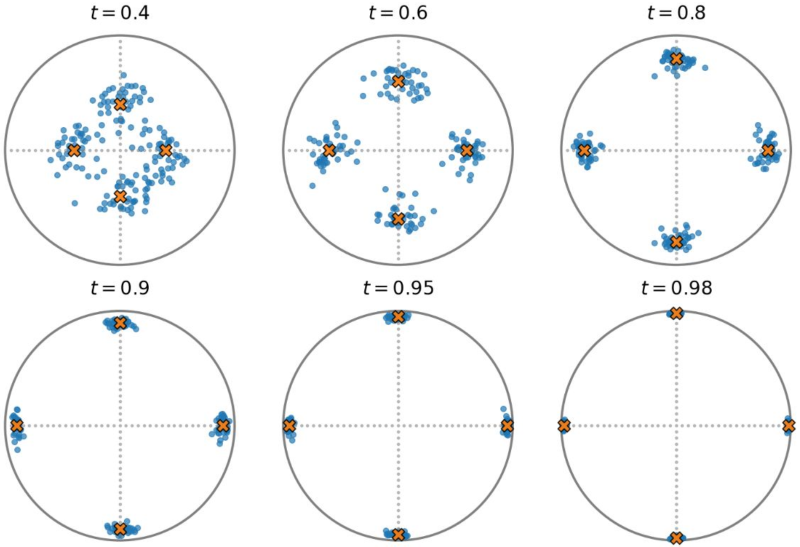


1) HyperSteiner leads to 2-3% improvement over the MST in standard scenarios, similarly to the Euclidean case.

	$d = 3$		$d = 4$	
t \ $\lvert P\rvert/d$	1	40	1	30
0.40	13.48 ± 0.33	2.52 ± 0.39	9.64 ± 0.27	2.57 ± 0.36
0.60	14.59 ± 0.19	2.33 ± 0.51	11.55 ± 0.17	2.27 ± 0.37
0.80	16.31 ± 0.11	2.44 ± 1.27	14.95 ± 0.11	1.89 ± 0.35
0.90	17.76 ± 0.07	3.16 ± 2.42	18.02 ± 0.08	1.81 ± 0.95
0.95	18.91 ± 0.05	4.47 ± 3.56	20.54 ± 0.06	2.18 ± 1.93
0.98	20.04 ± 0.04	6.63 ± 4.56	23.04 ± 0.05	4.02 ± 4.10

Results for validation on synthetic data





1) HyperSteiner leads to 2-3% improvement over the MST in standard scenarios, similarly to the Euclidean case.

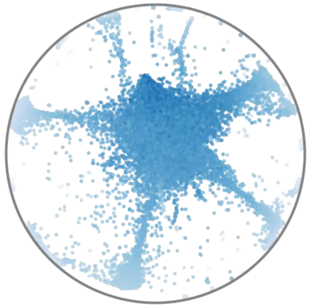
2) HyperSteiner achieves higher improvement over the MST than the maximum theoretical improvement on the Euclidean case (Gilbert-Pollak conjecture ~13.4%).

	$d=3$		$d=4$	
t \ $\|P\|/d$	1	40	1	30
0.40	13.48 ± 0.33	2.52 ± 0.39	9.64 ± 0.27	2.57 ± 0.36
0.60	14.59 ± 0.19	2.33 ± 0.51	11.55 ± 0.17	2.27 ± 0.37
0.80	16.31 ± 0.11	2.44 ± 1.27	14.95 ± 0.11	1.89 ± 0.35
0.90	17.76 ± 0.07	3.16 ± 2.42	18.02 ± 0.08	1.81 ± 0.95
0.95	18.91 ± 0.05	4.47 ± 3.56	20.54 ± 0.06	2.18 ± 1.93
0.98	20.04 ± 0.04	6.63 ± 4.56	23.04 ± 0.05	4.02 ± 4.10

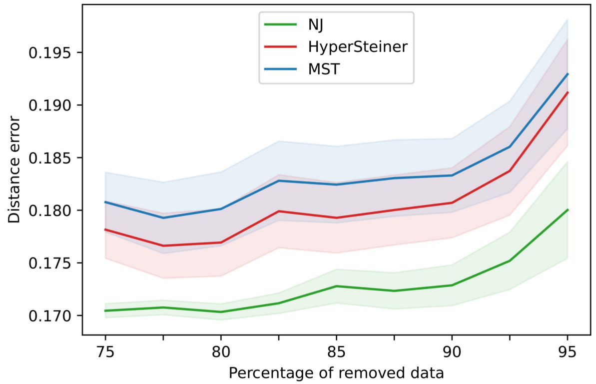
Results for cell age prediction on real data

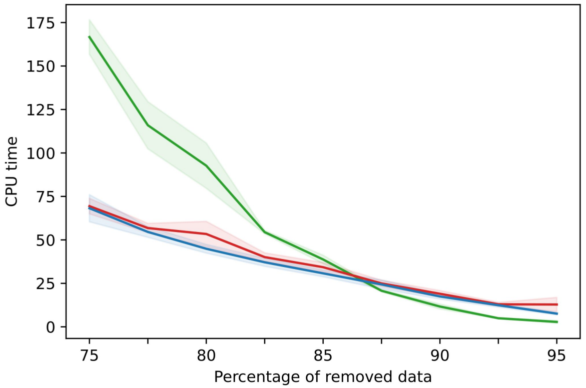
Whole *Planaria* dataset

Method	NJ	HyperSteiner	MST
Distance error	0.17	0.18	0.19
CPU	12604	995	914



Missing data





HyperSteiner:

1) Scales much better than Neighbor Joining to large datasets.

2) Performs better than Minimum Spanning Tree.

Conclusion

We developed a method to compute a heuristic Steiner tree connecting points in a hyperbolic space, with applications to hierarchy discovery in biology.

Questions?