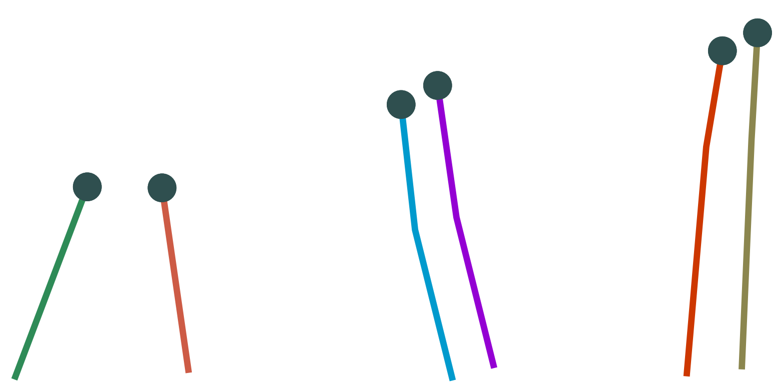
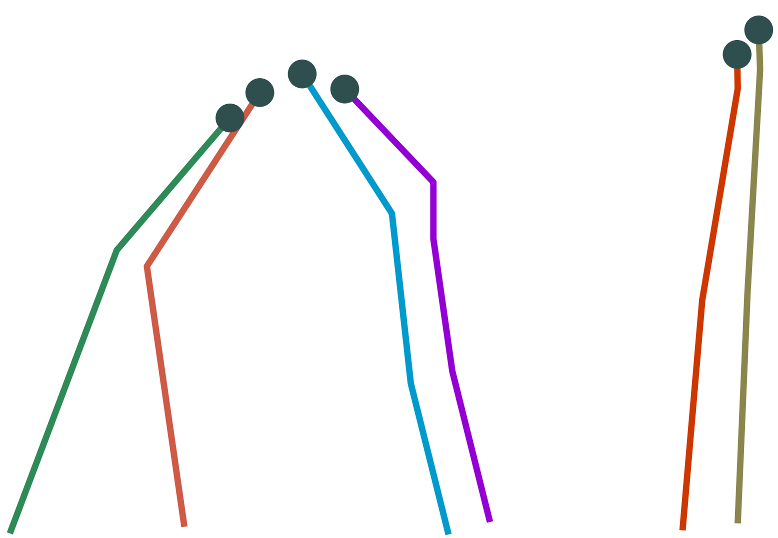
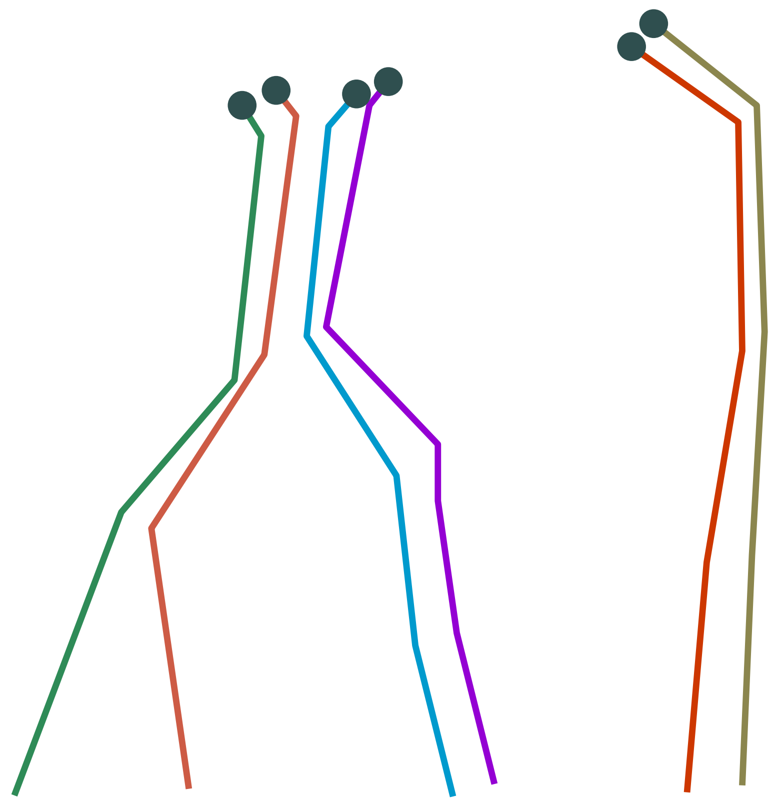
Trajectory Grouping Structure

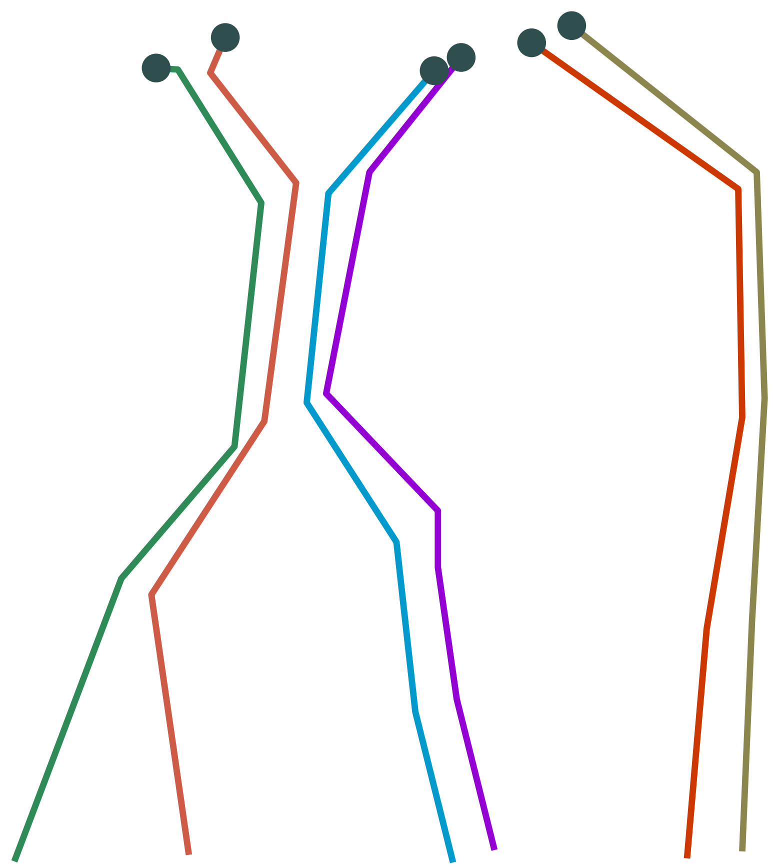
Kevin Buchin
Maike Buchin
Marc van Kreveld
Bettina Speckmann
Frank Staals

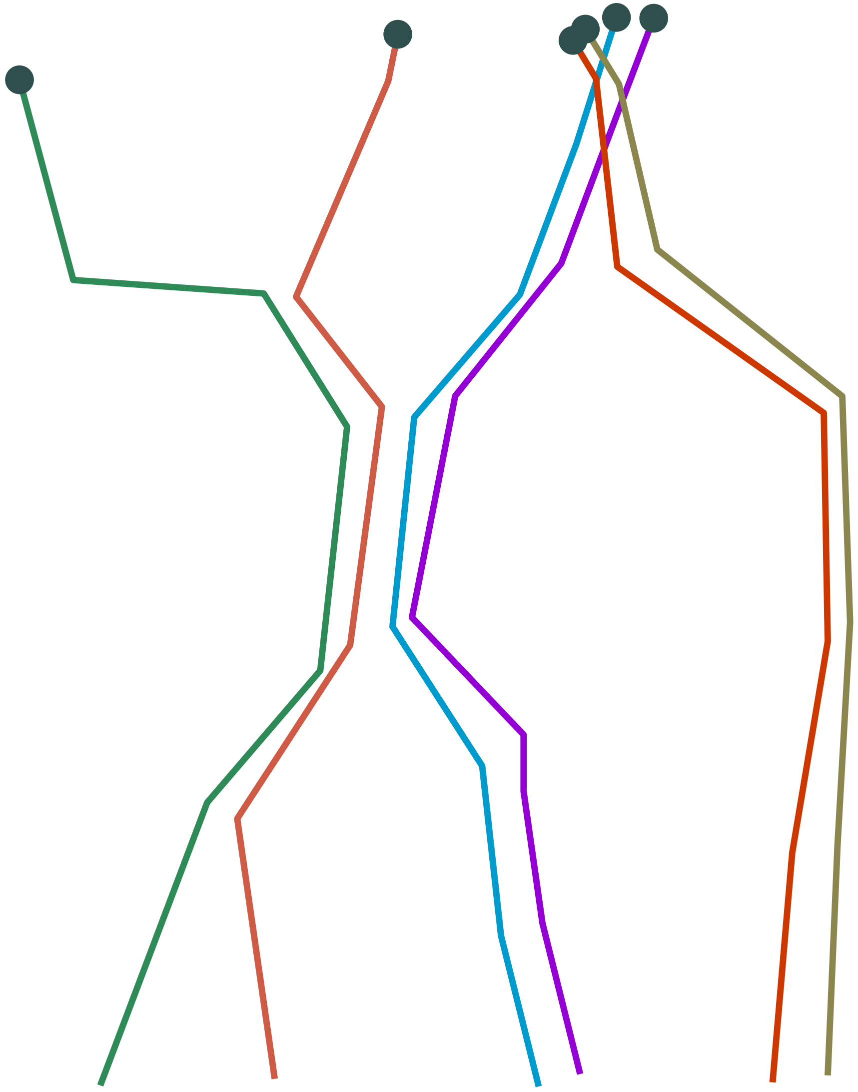
TU Eindhoven
Utrecht University

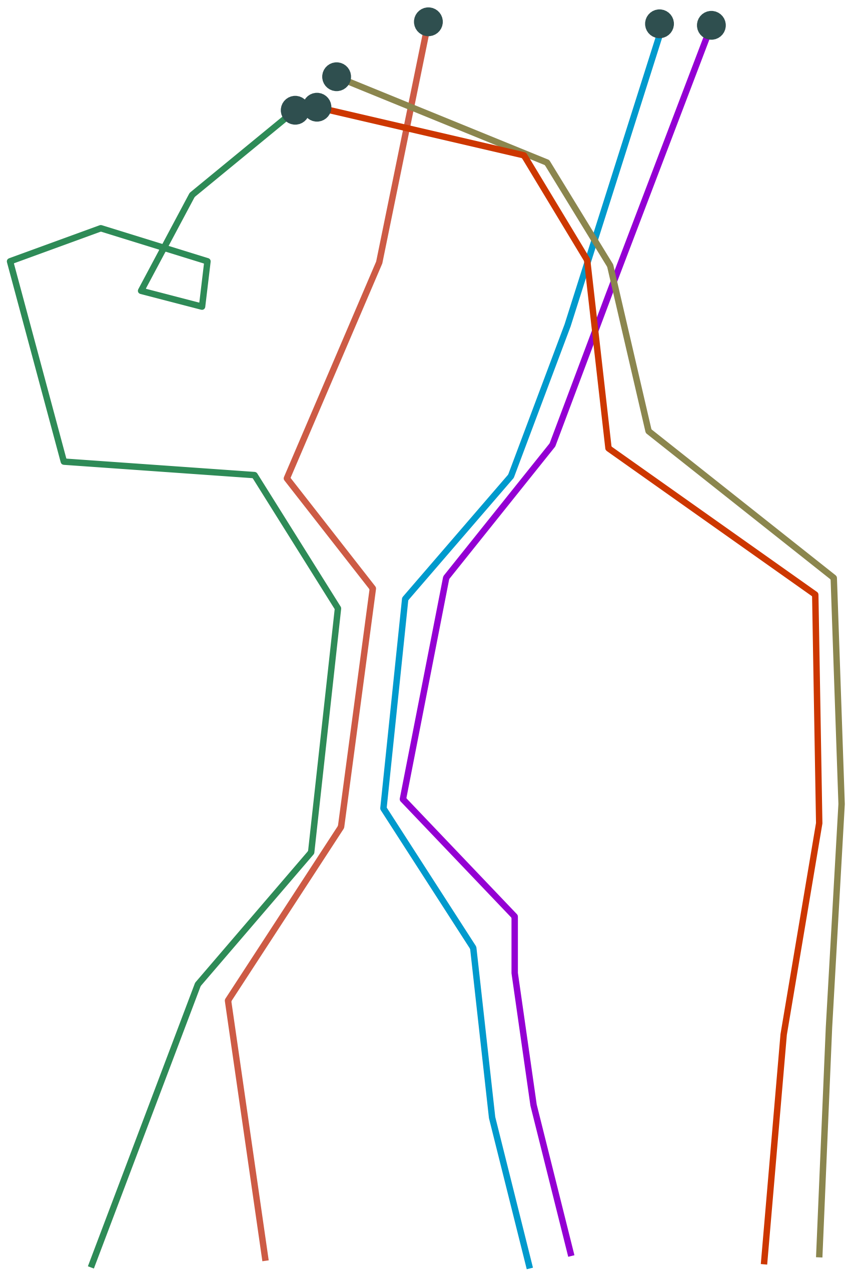


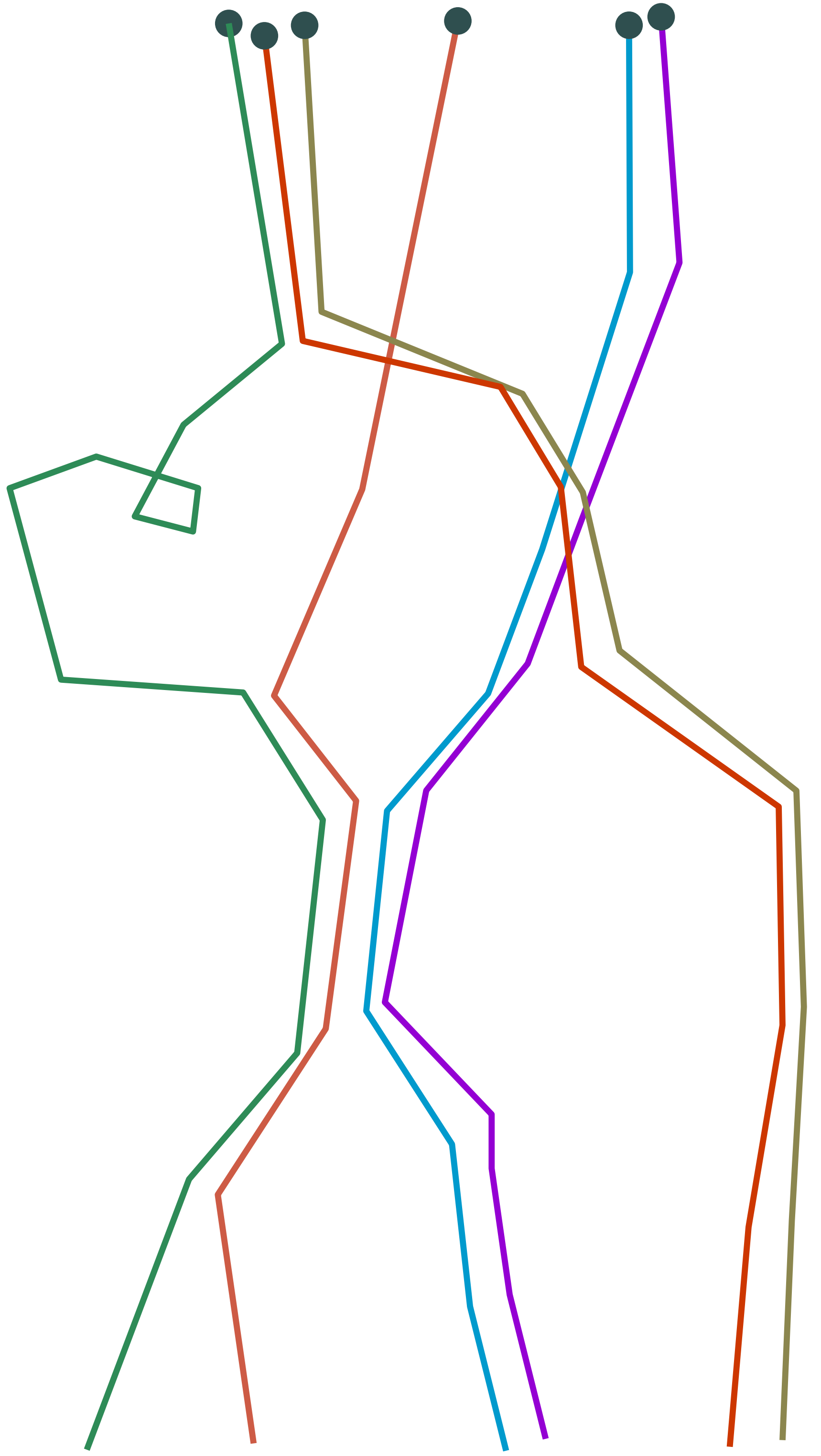


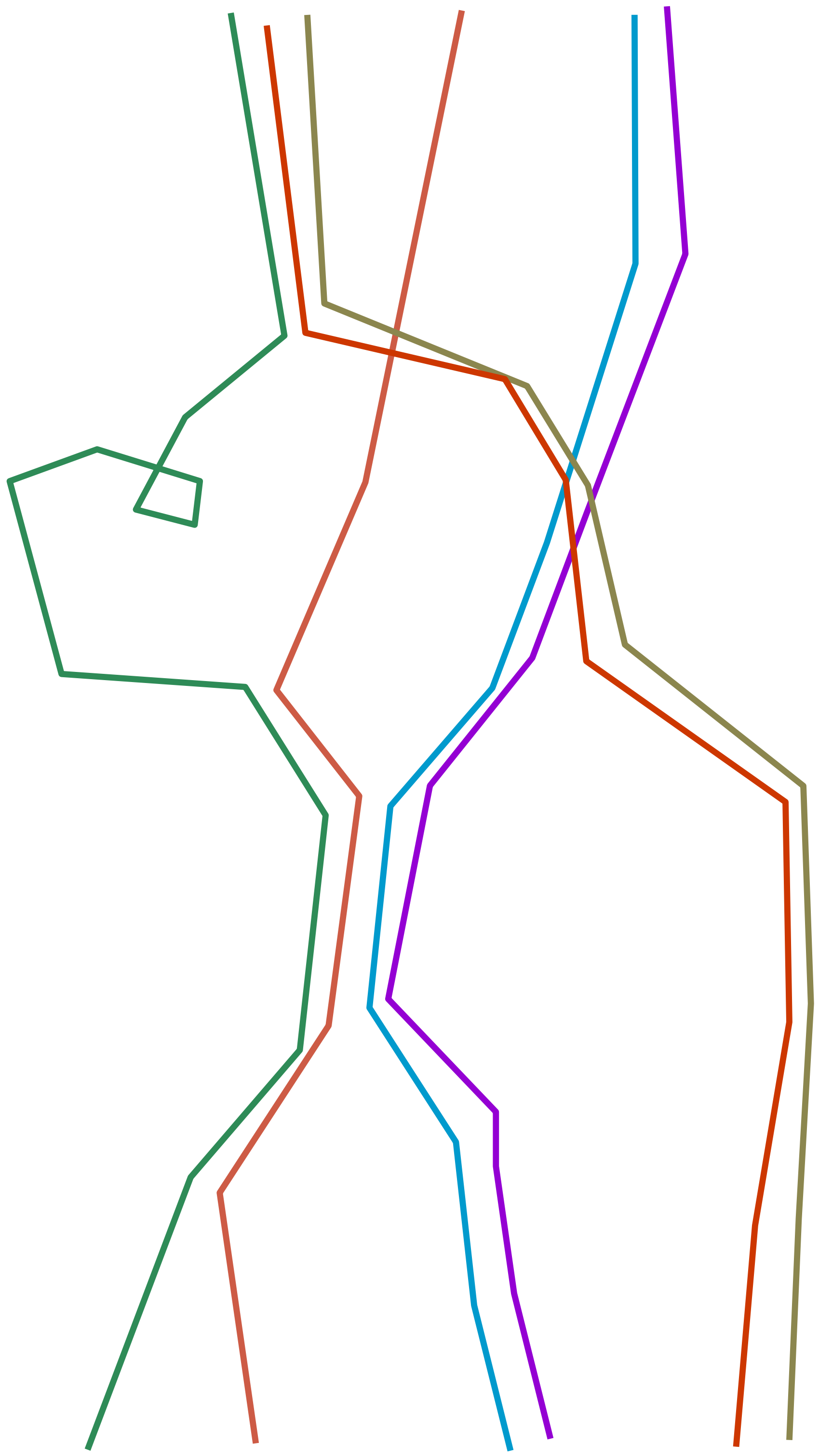








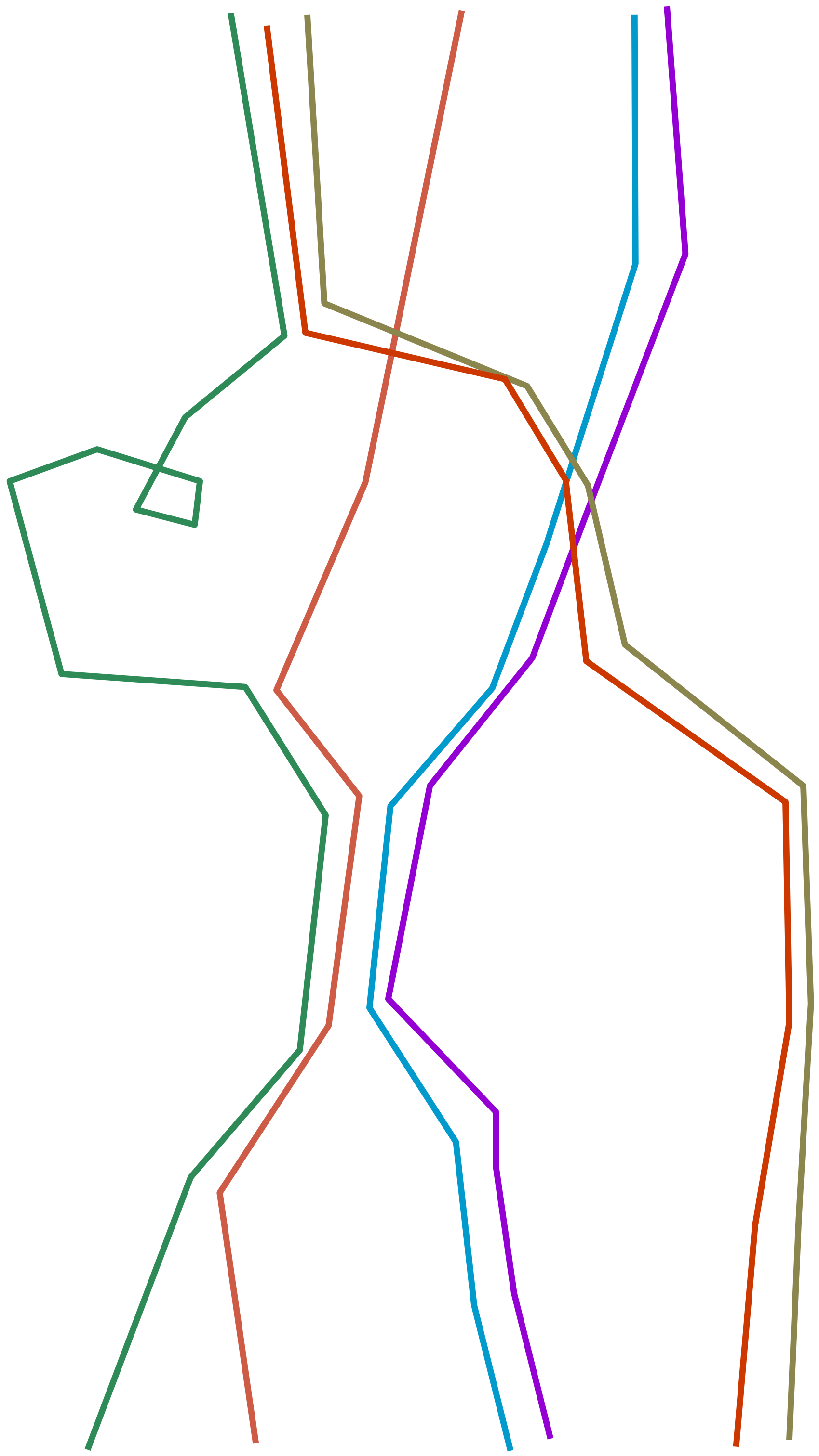




Groups

We want to find groups,

and how the groups change over time.

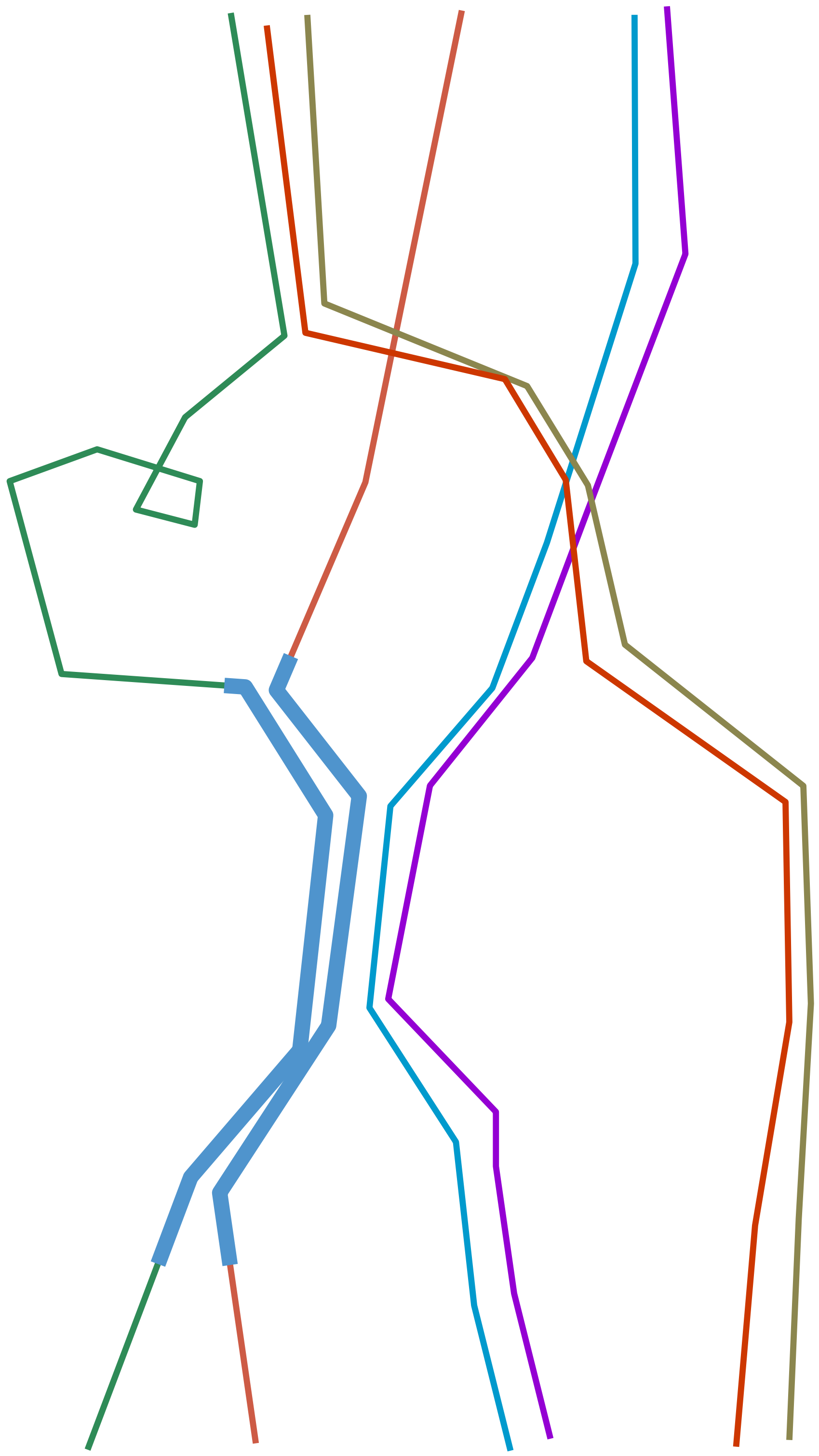


Groups

We want to find groups,

> Group: a large set of entities who travel together during a long period of time.

and how the groups change over time.

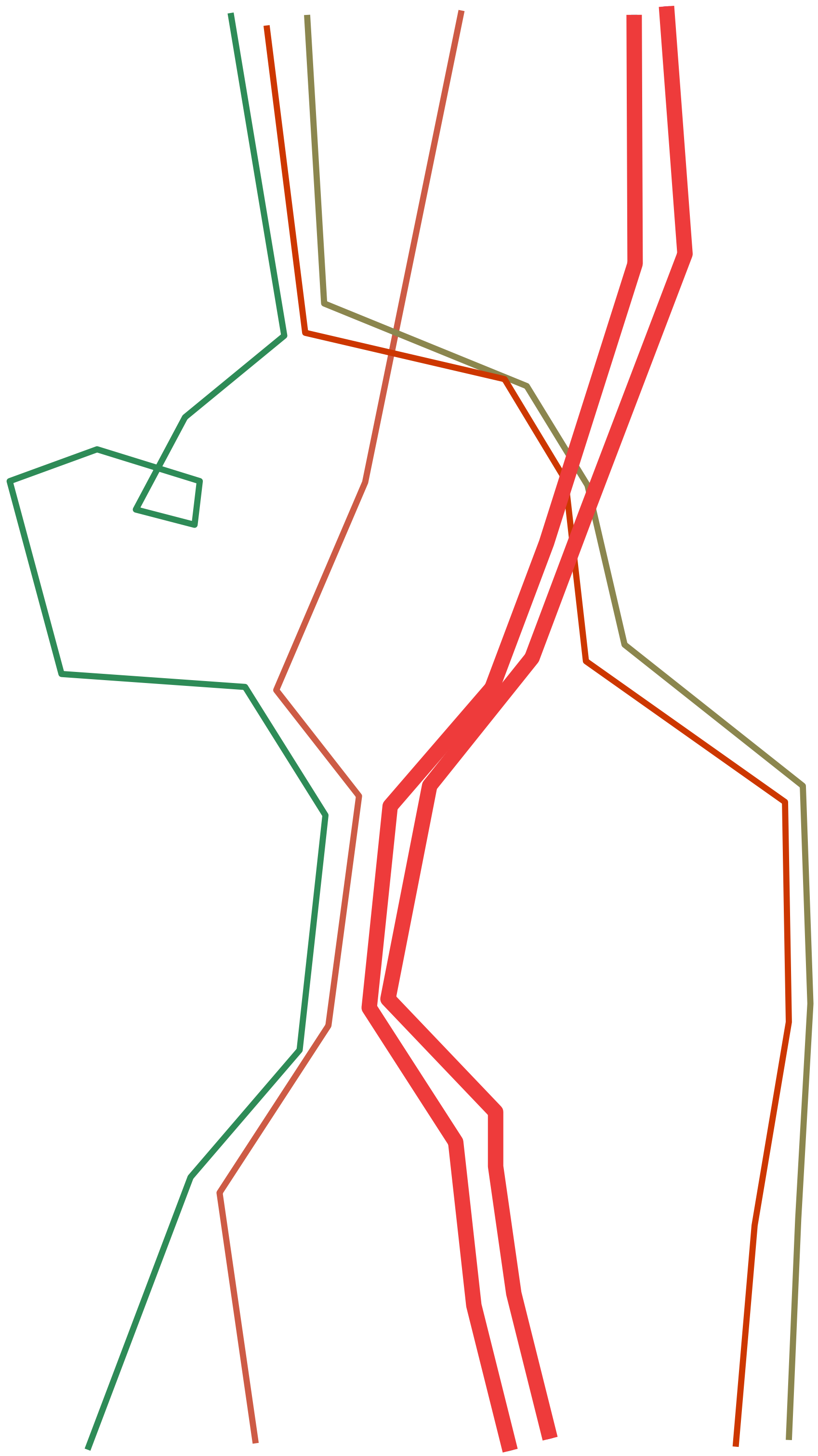


Groups

We want to find groups,

> Group: a large set of entities who travel together during a long period of time.

and how the groups change over time.

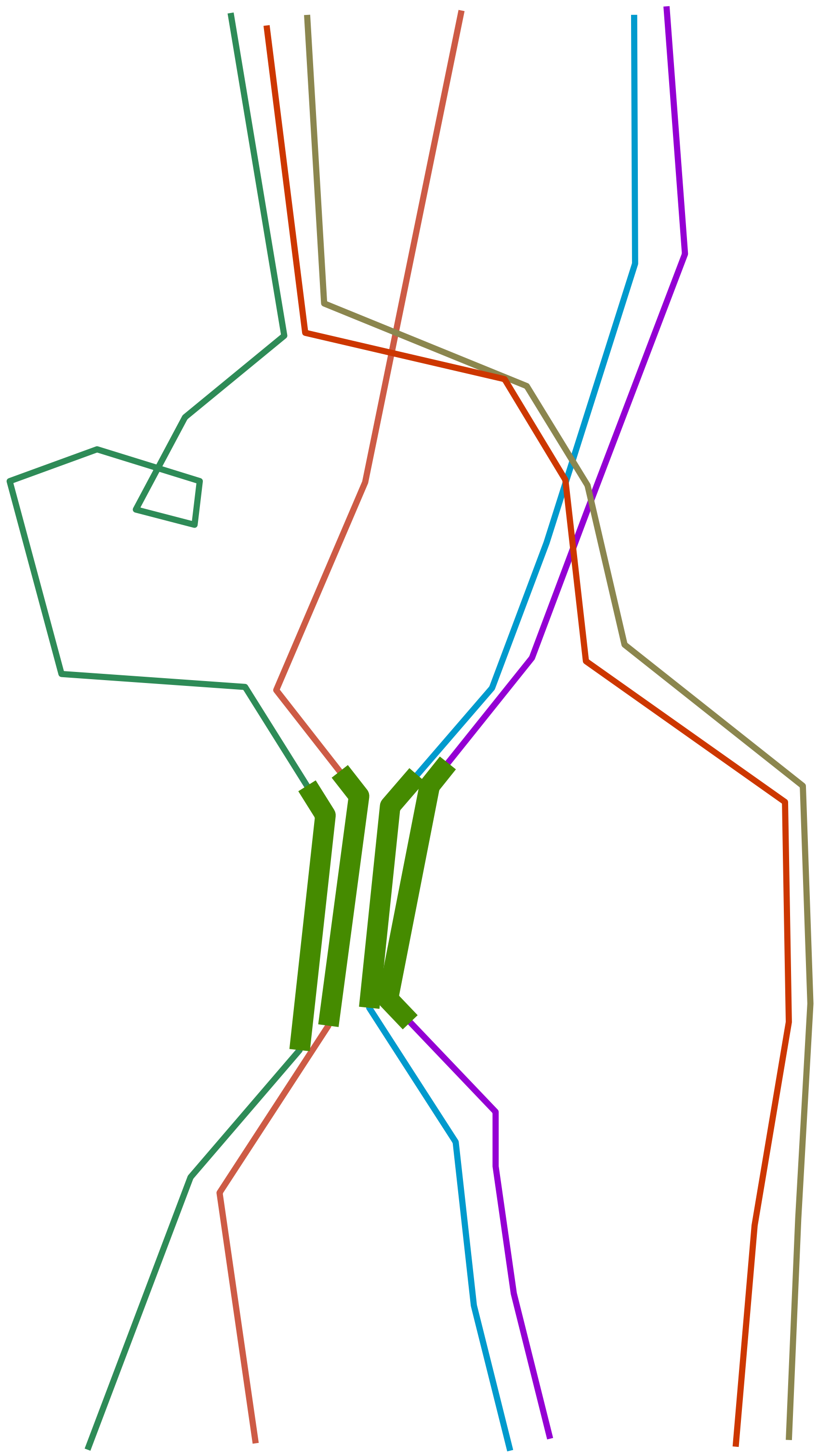


Groups

We want to find groups,

> Group: a large set of entities who travel together during a long period of time.

and how the groups change over time.

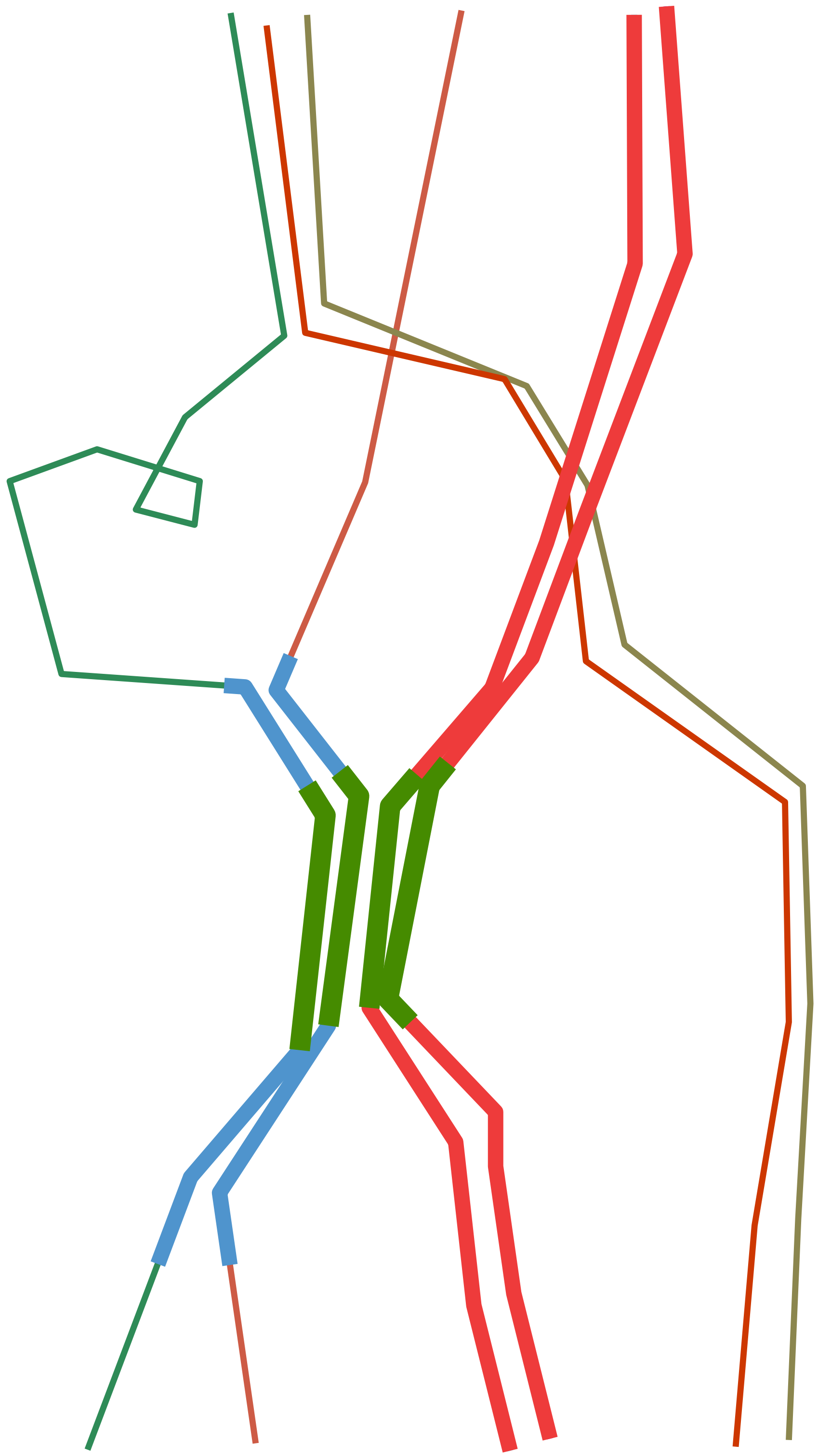


Groups

We want to find groups,

> Group: a large set of entities who travel together during a long period of time.

and how the groups change over time.



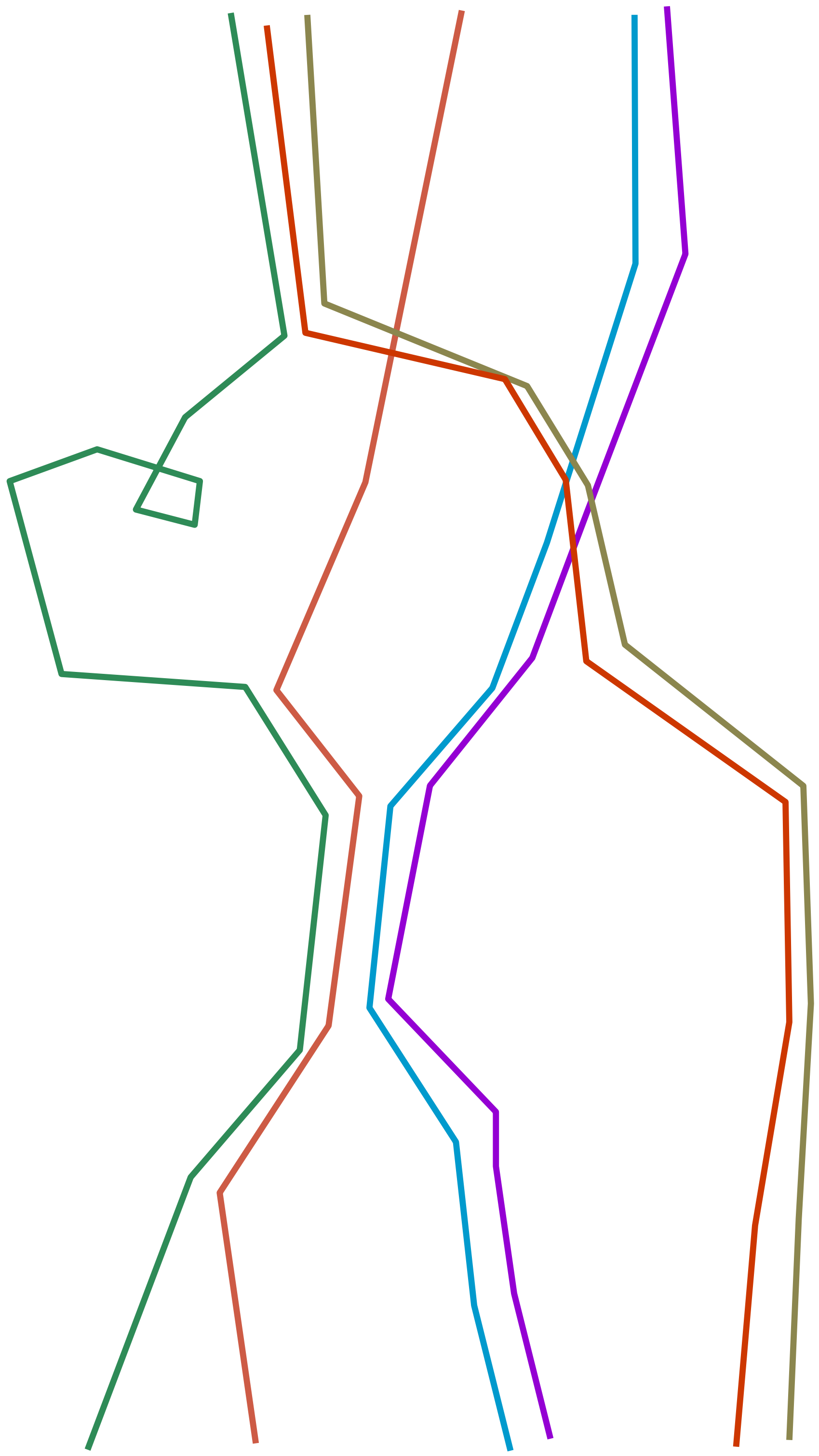
Groups

We want to find groups,

> Group: a large set of entities who travel together during a long period of time.

and how the groups change over time.

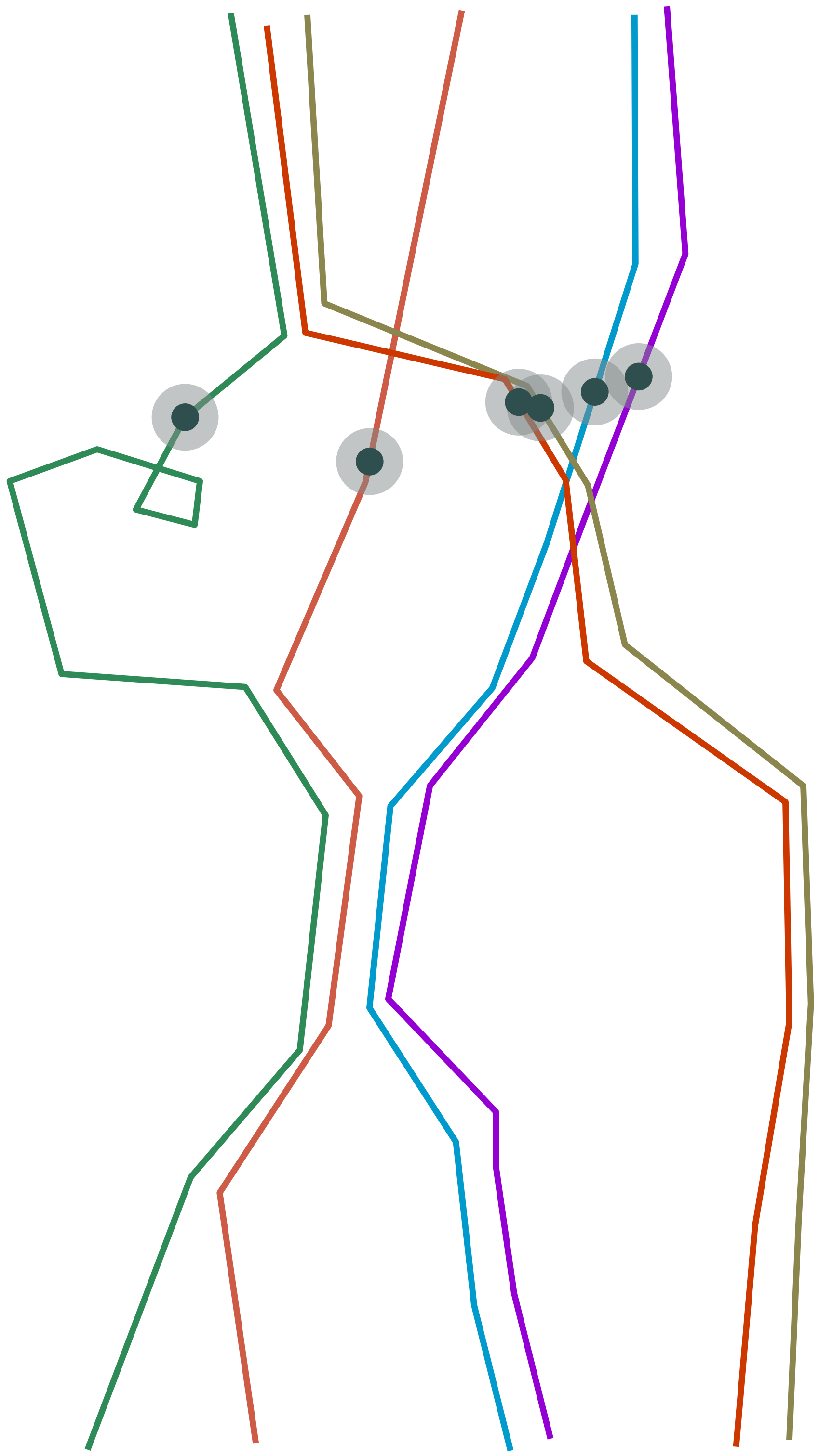
> Trajectory Grouping Structure



Groups

Group parameters:

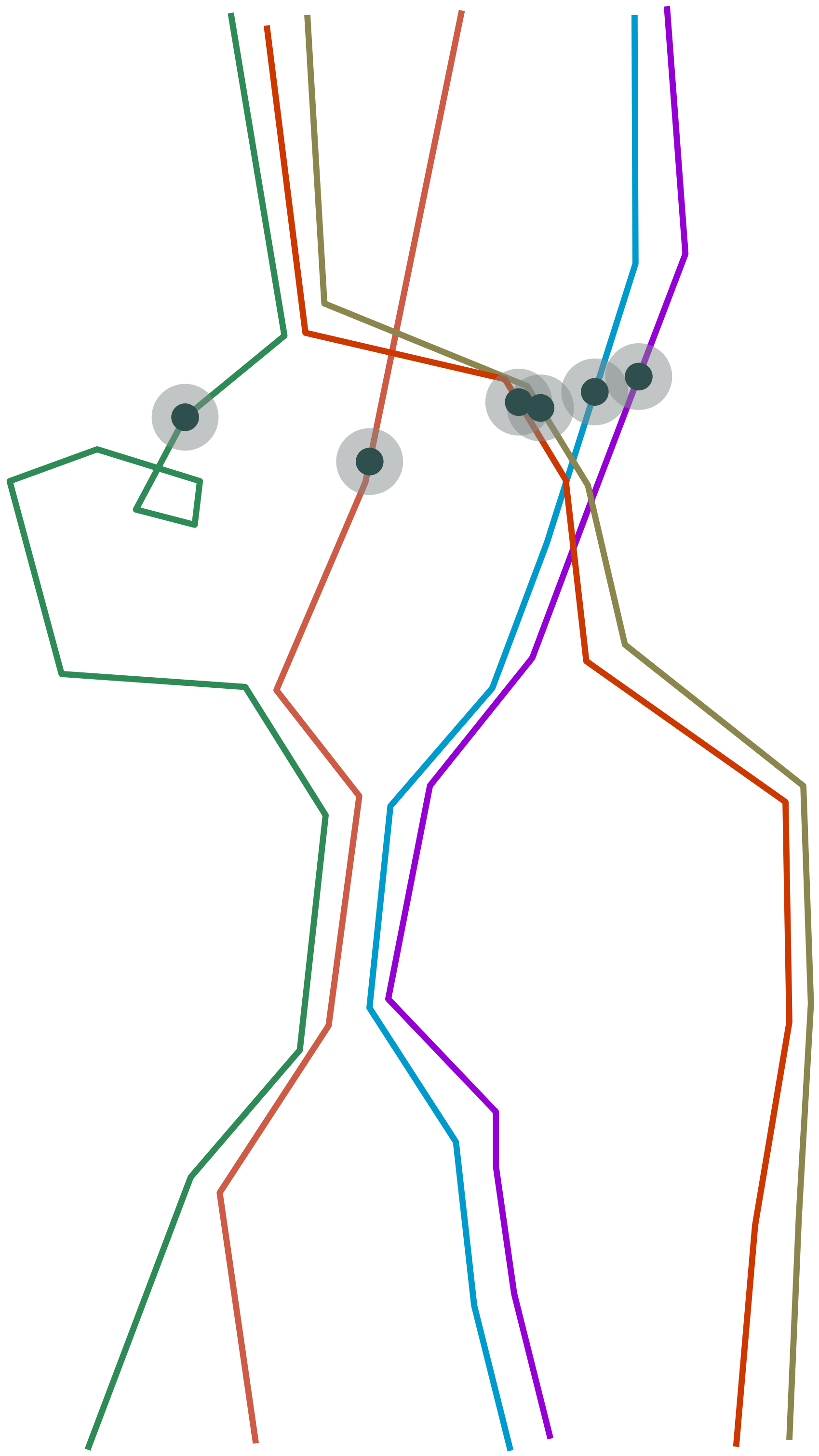
- min group size: m
- min duration: δ
- max distance between entities: ε



Groups

Group parameters:

- min group size: m
- min duration: δ
- max distance between entities: ε

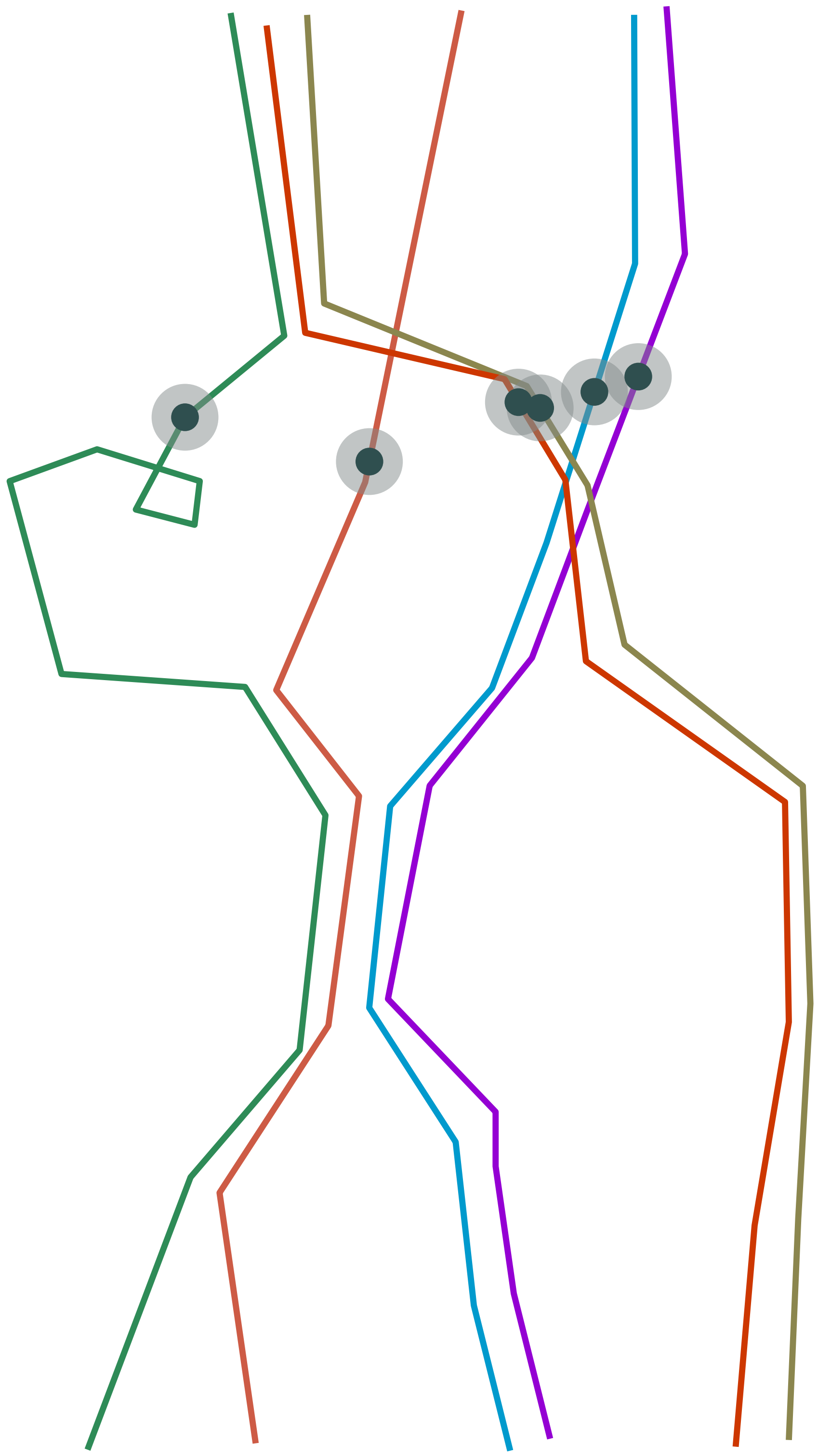


Groups

Group parameters:

- min group size: m
- min duration: δ
- max distance between entities: ε

The grouping structure should allow study of groups at different scales.



Groups

Group parameters:

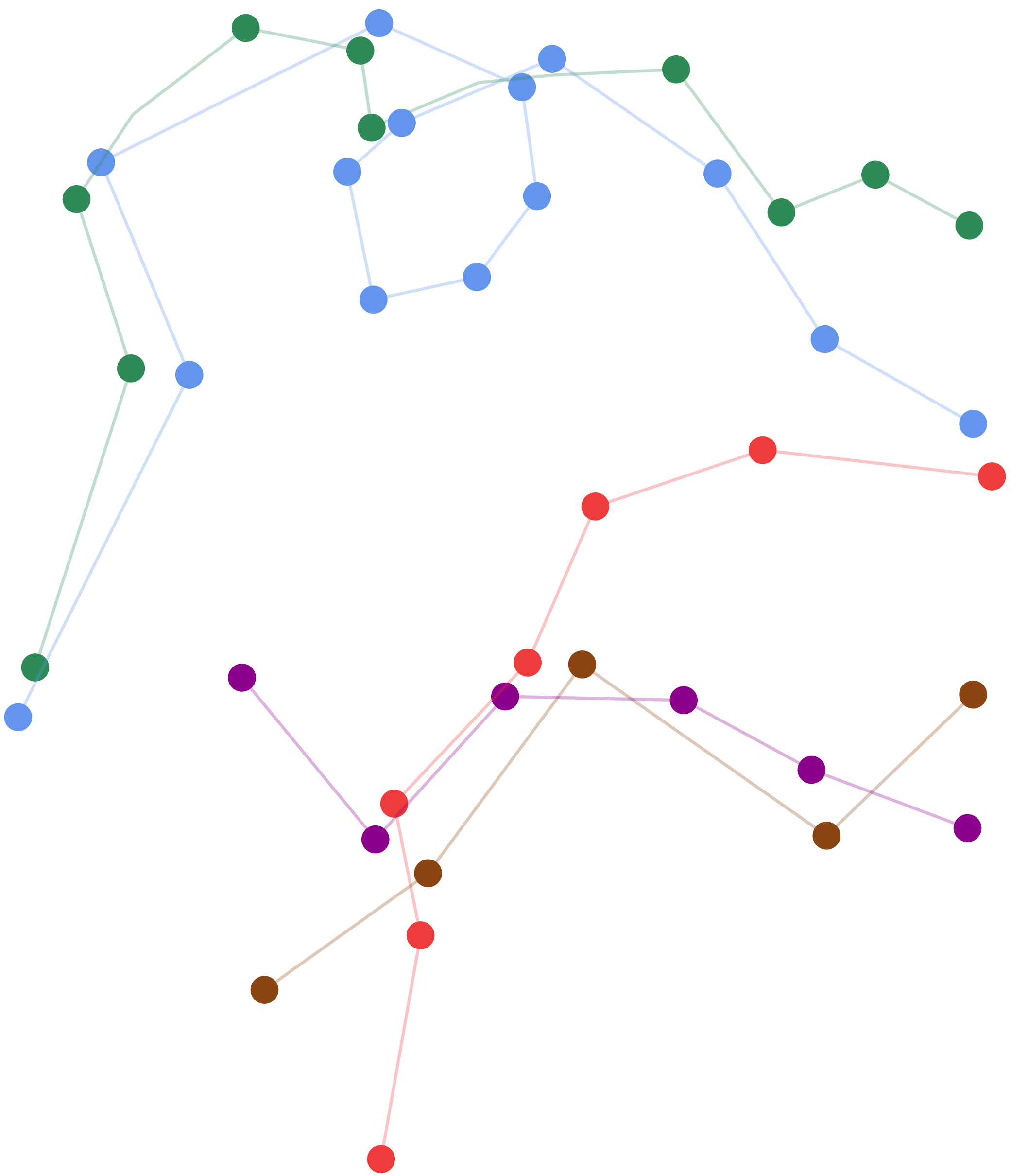
- min group size: m
- min duration: δ
- max distance between entities: ε

The grouping structure should allow study of groups at different scales.

Computational Topolgy

Related Work

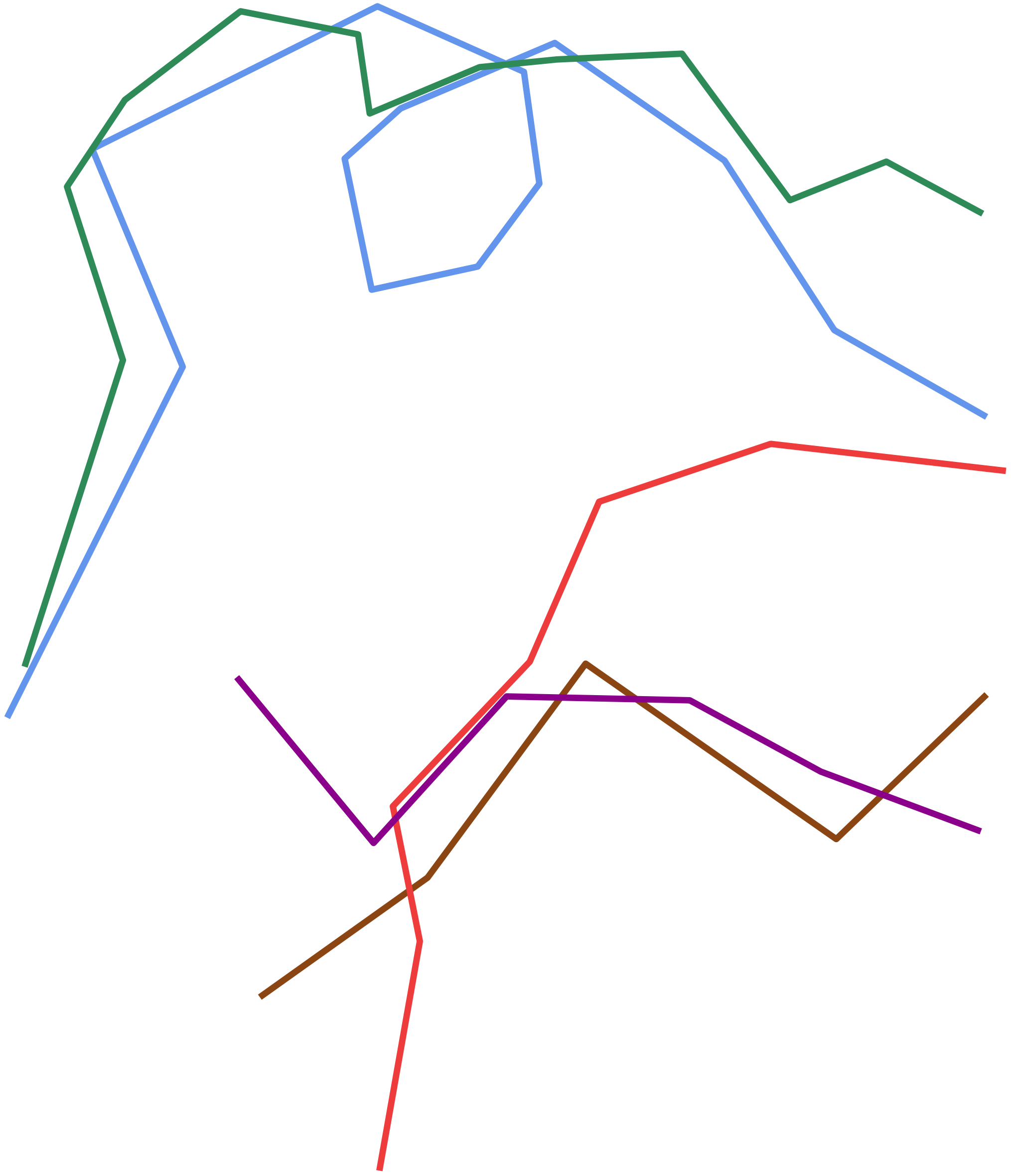
convoys, moving clusters, mobile groups, swarms, flocks, herds, ...



Related Work

convoys, moving clusters, mobile groups, swarms, flocks, herds, ...

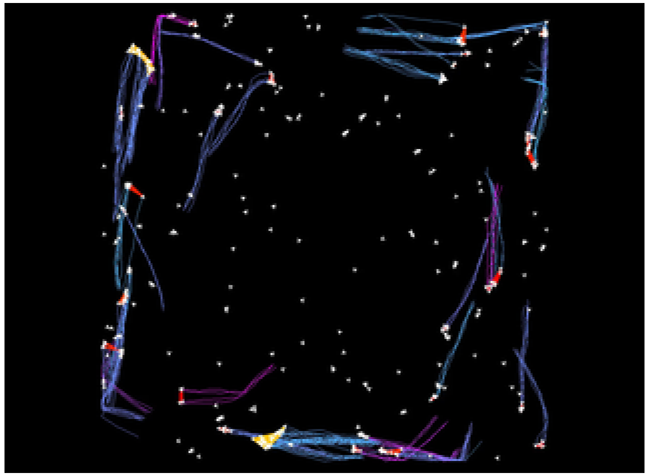
Discrete trajectory data



Related Work

convoys, moving clusters, mobile groups, swarms, flocks, herds, ...

Continuous trajectory data

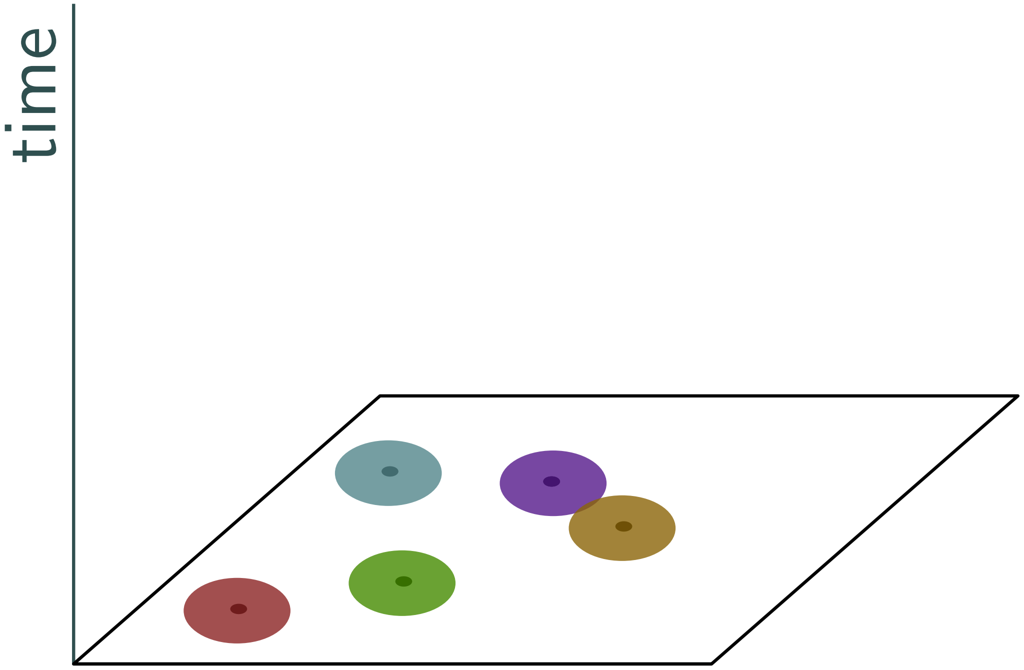


Related Work

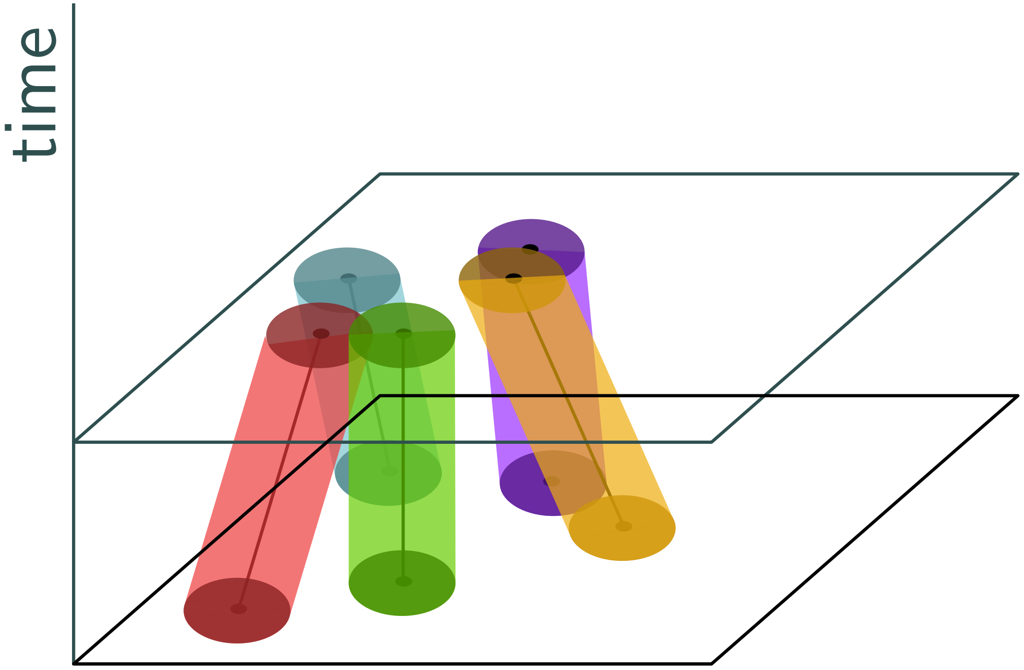
convoys, moving clusters, mobile groups, swarms, flocks, herds, ...

Continuous trajectory data

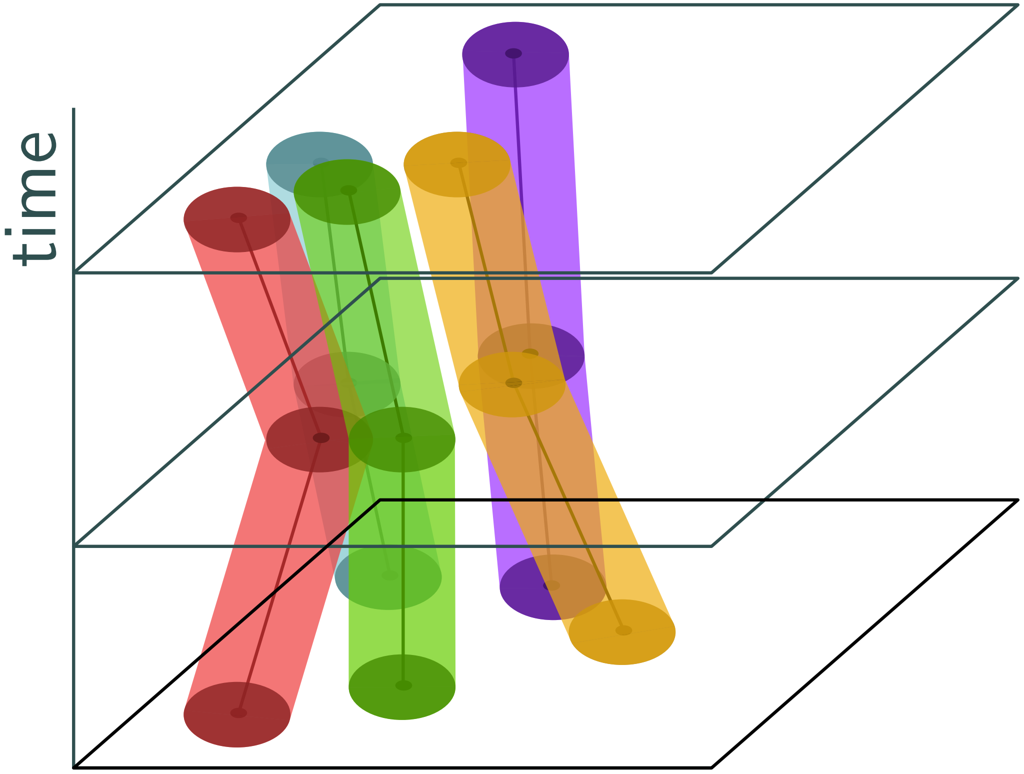
We provide an experimental evaluation.



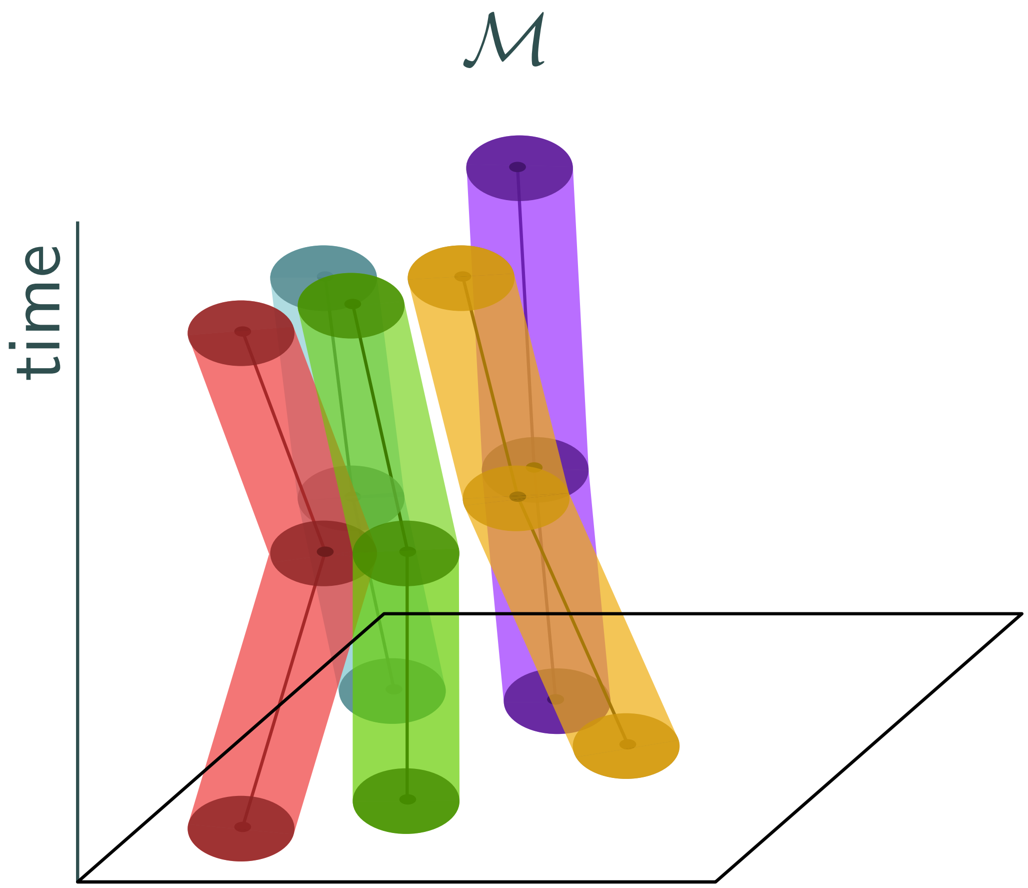
Consider tracing ε-discs over the trajectories.



Consider tracing ε-discs over the trajectories.

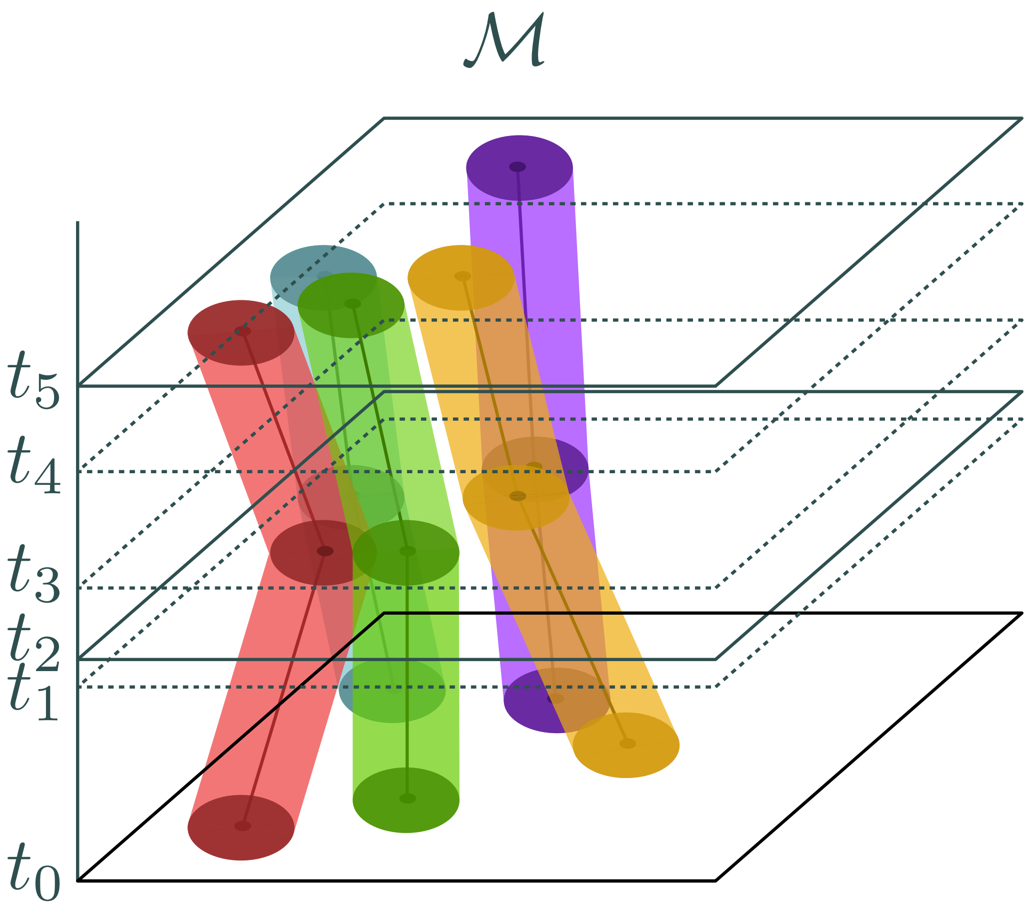


Consider tracing ε-discs over the trajectories.



Consider tracing ε-discs over the trajectories.

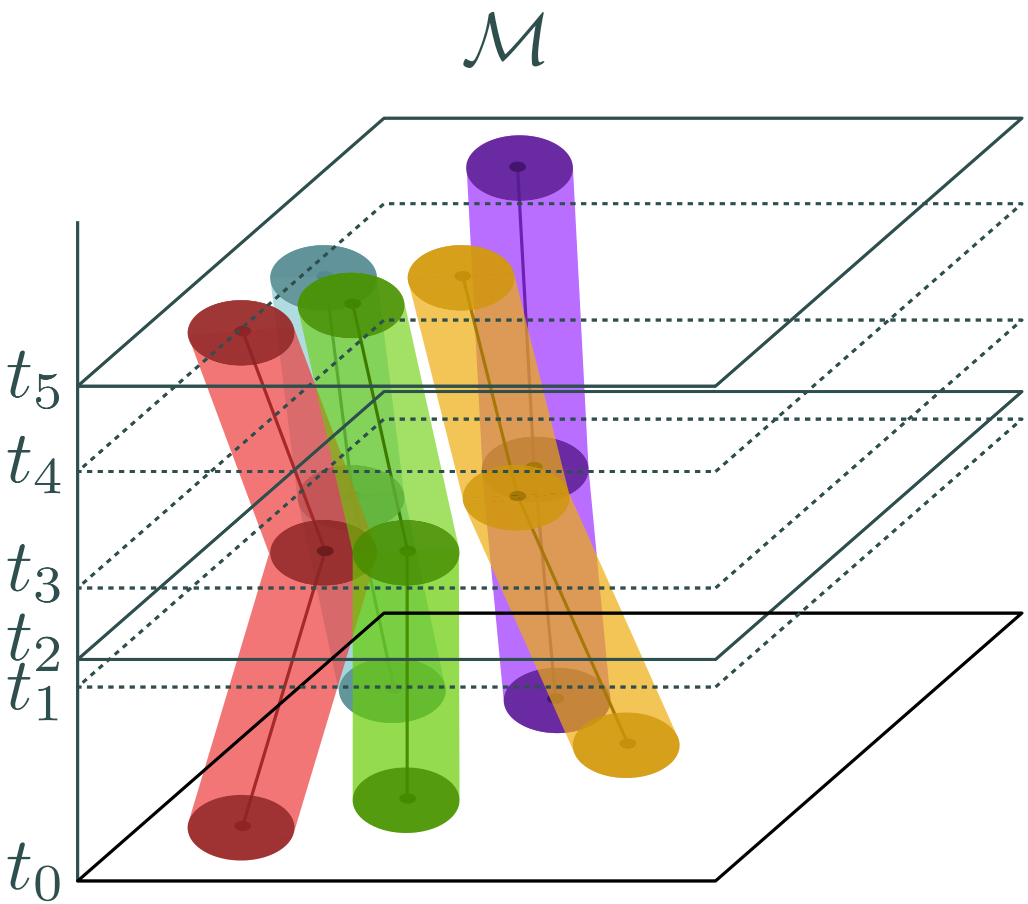
Each entity produces a "tube". Let $\mathcal{M}$ be the union of those n "tubes".



Consider tracing ε-discs over the trajectories.

Each entity produces a "tube". Let $\mathcal{M}$ be the union of those n "tubes".

Groups change when connectivity in $\mathcal{M}$ changes.



Consider tracing ε-discs over the trajectories.

Each entity produces a "tube". Let $\mathcal{M}$ be the union of those n "tubes".

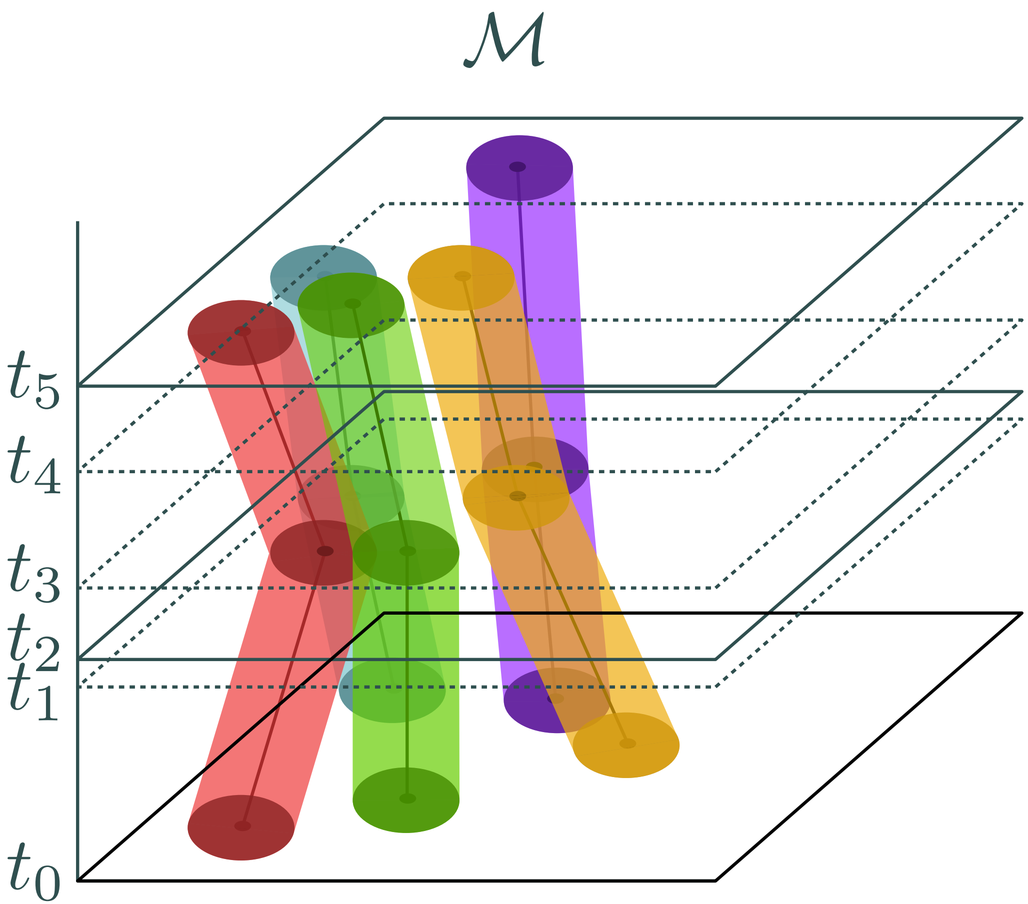
Groups change when connectivity in $\mathcal{M}$ changes.

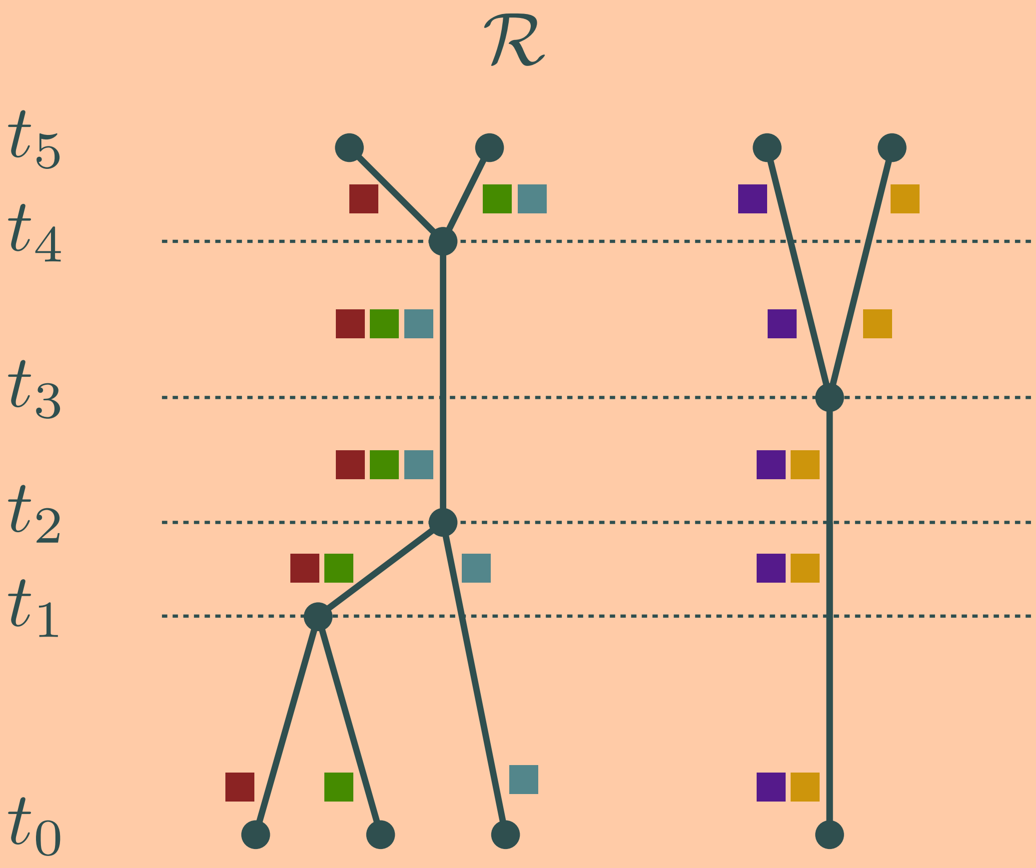
The Reeb graph $\mathcal{R}$ of $\mathcal{M}$ captures connectivity changes.

$\Longrightarrow$

Reeb graph captures all group changes.

The Reeb Graph



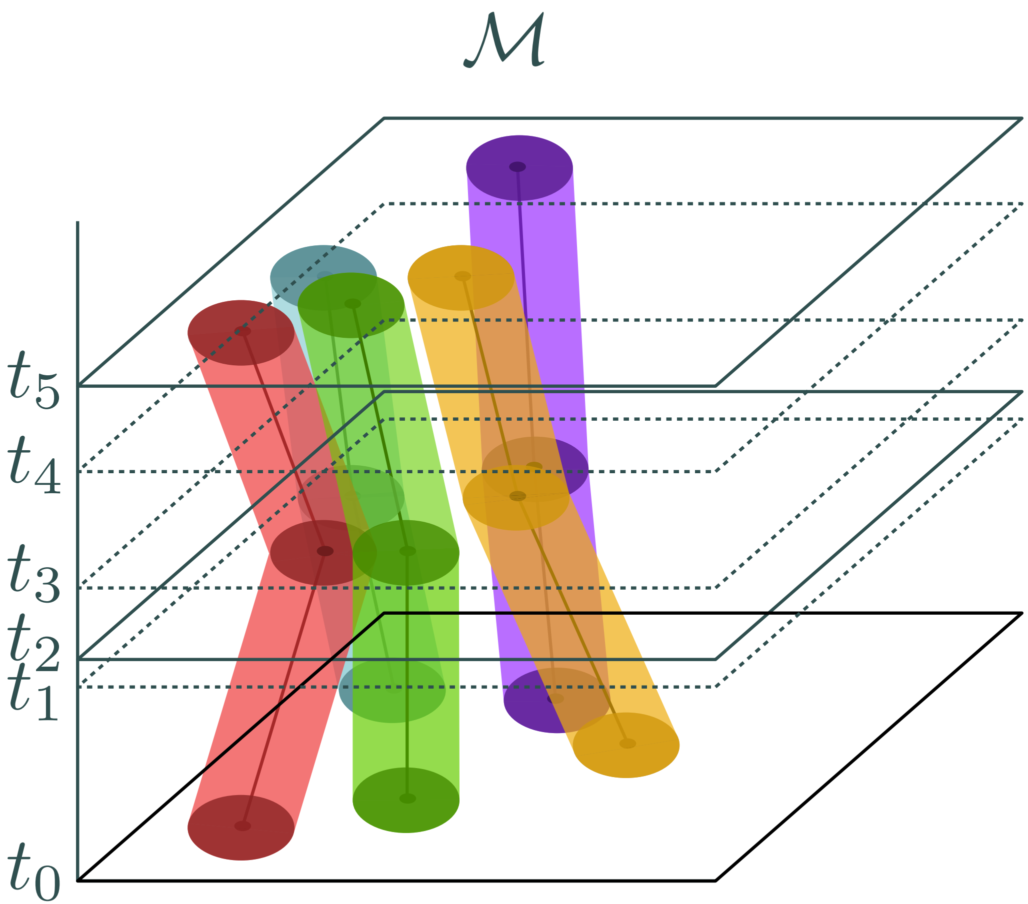


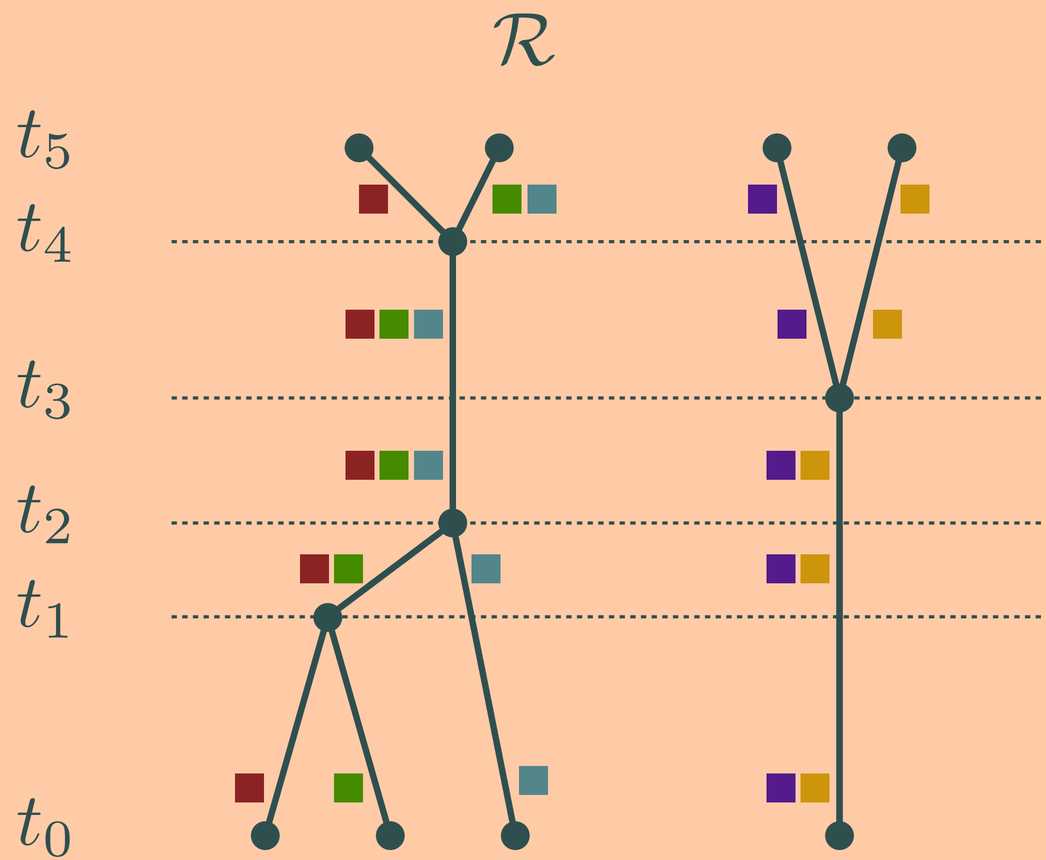
The Reeb Graph

Annotate $\mathcal{R}$ with all groups

$\Longrightarrow$

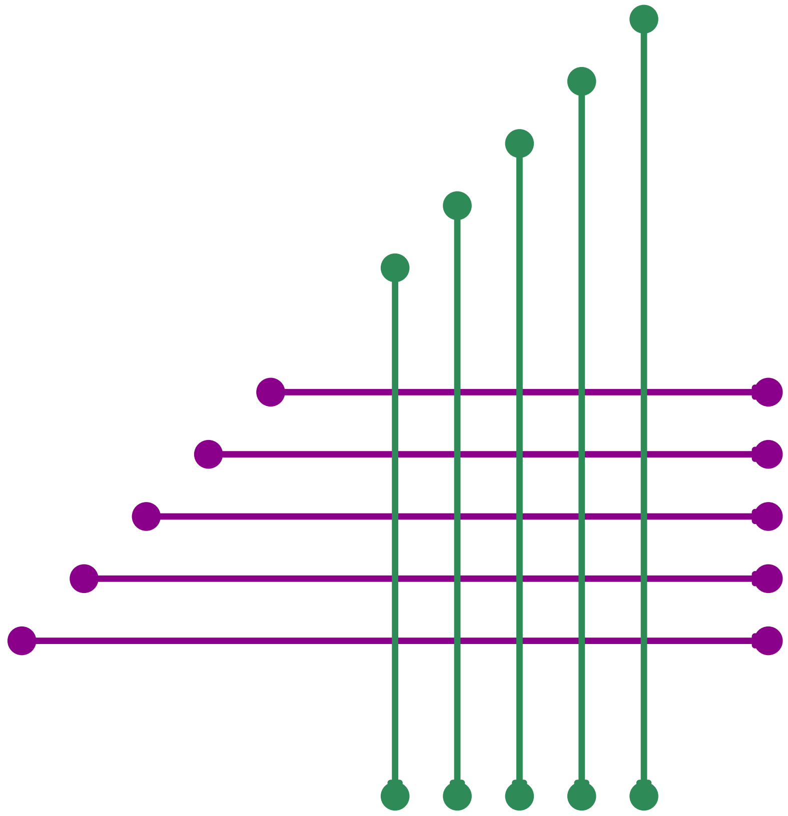
Trajectory Grouping Structure





The Reeb Graph

Complexity $\mathcal{R}$?

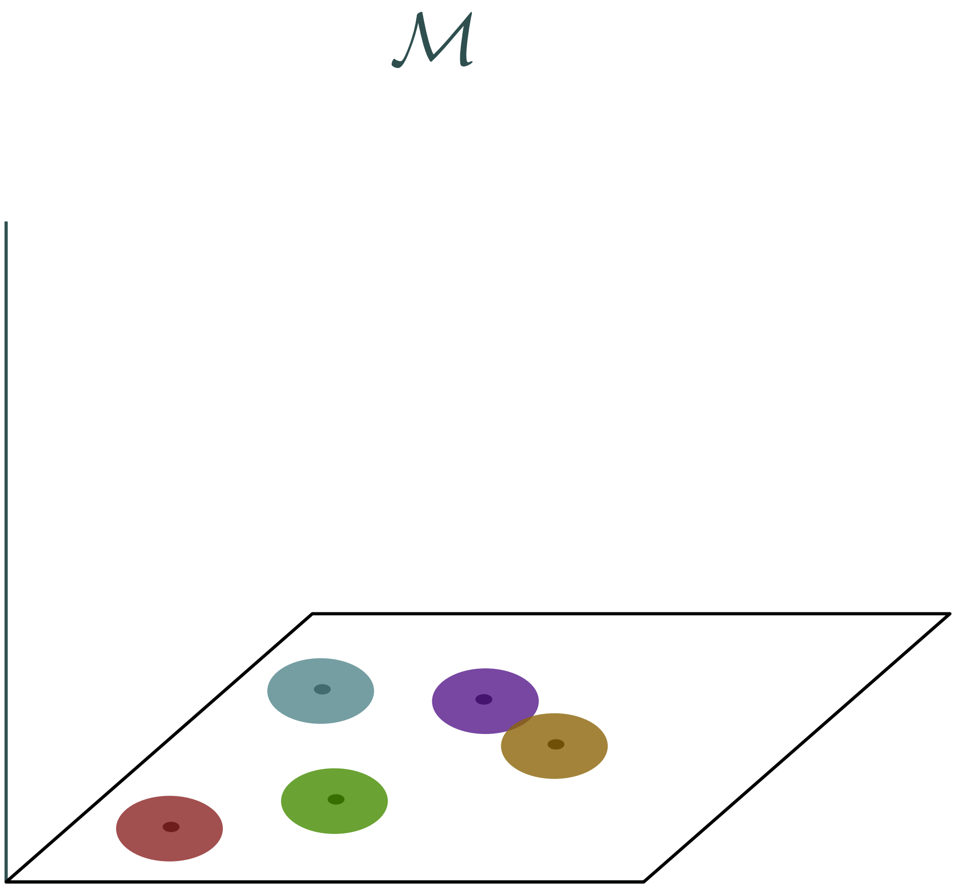


The Reeb Graph

Complexity $\mathcal{R}$?

$\Theta(\tau n^2)$

$\tau =$ trajectory length



The Reeb Graph

How to compute $\mathcal{R}$?

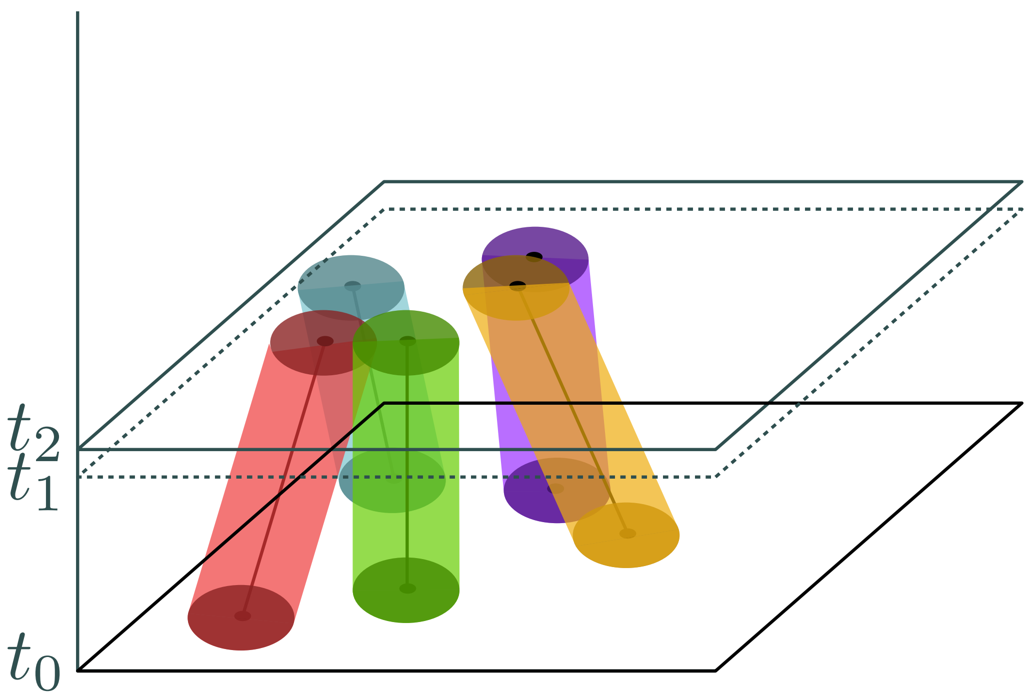
Sweep $\mathcal{M}$ while maintaining connected components.

The Reeb Graph

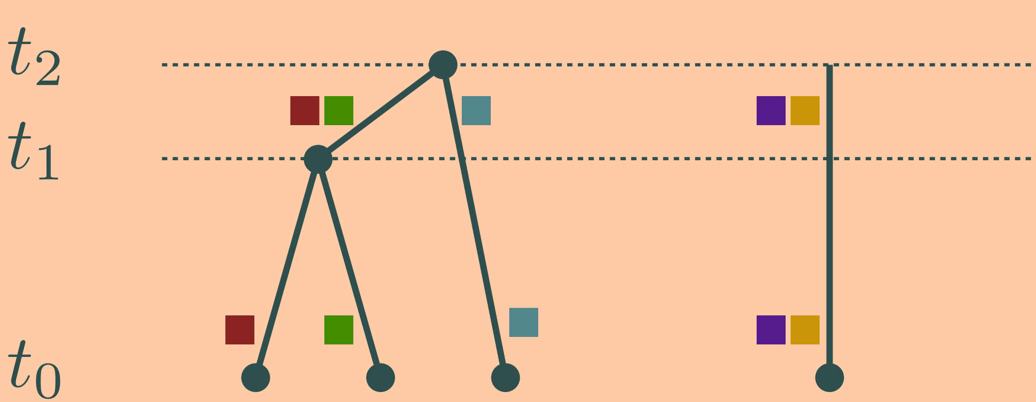
How to compute $\mathcal{R}$?

Sweep $\mathcal{M}$ while maintaining connected components.

$\mathcal{M}$



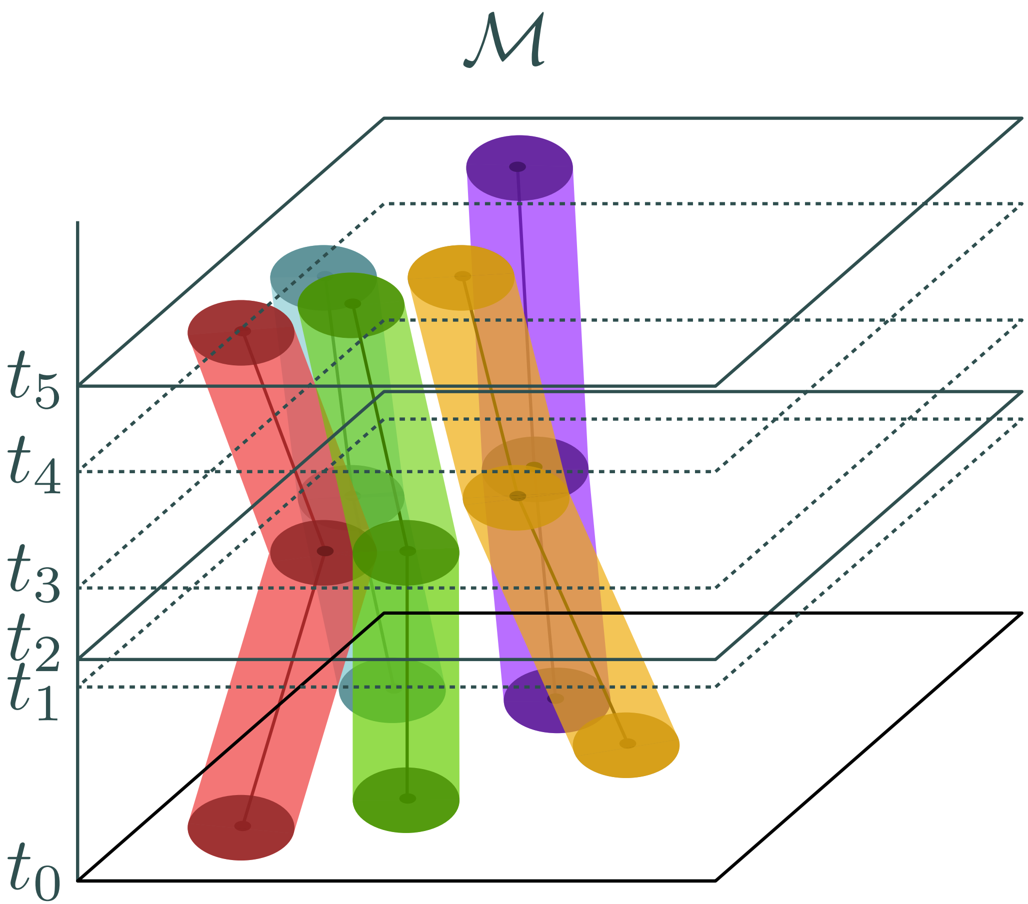
$\mathcal{R}$

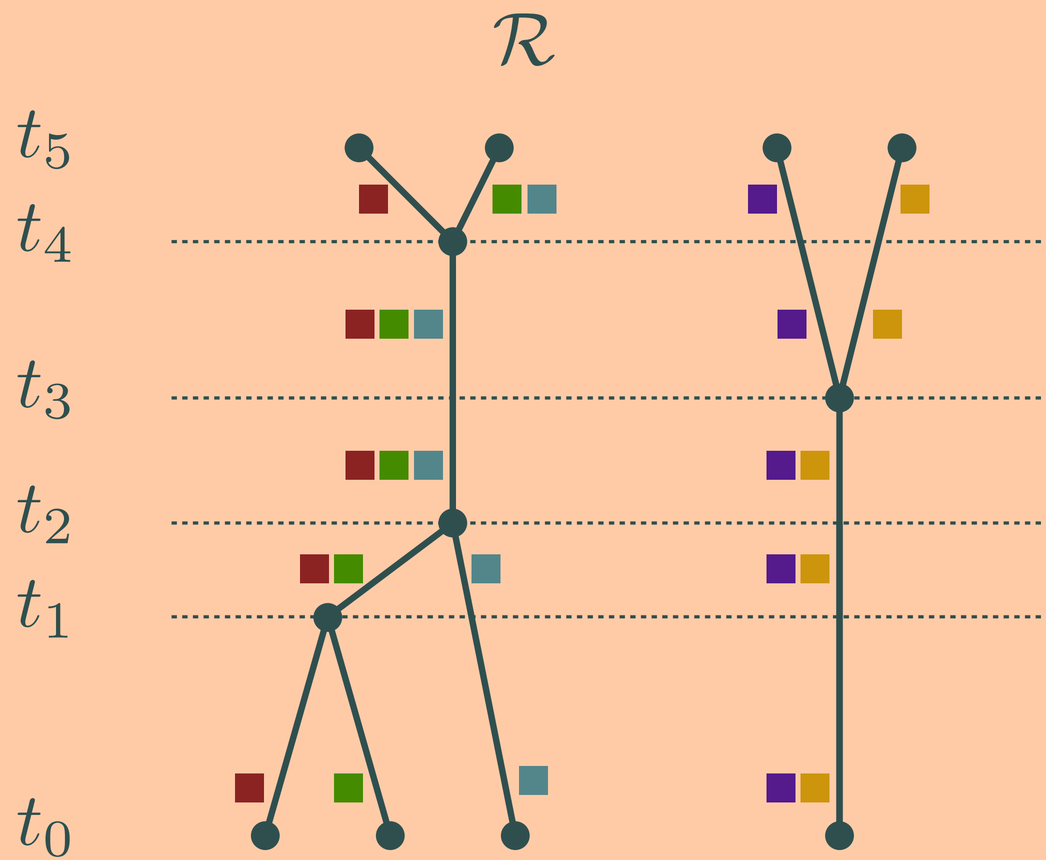


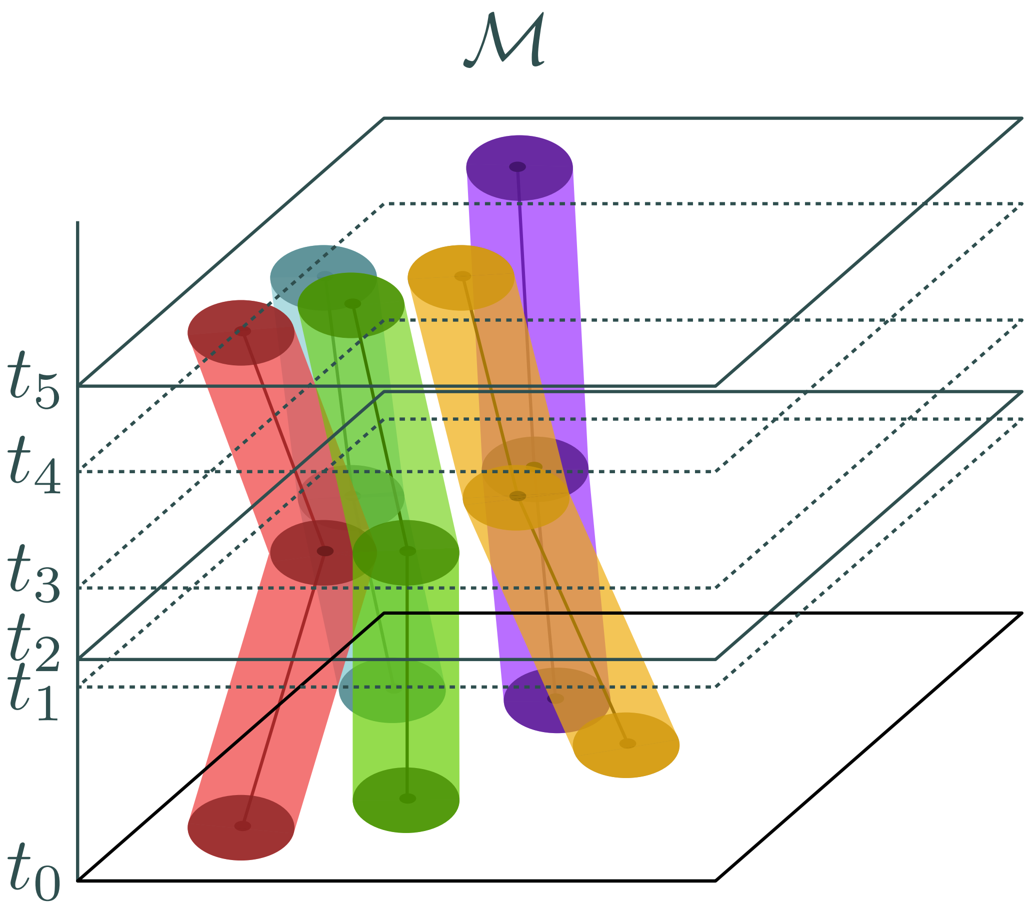
The Reeb Graph

How to compute $\mathcal{R}$?

Sweep $\mathcal{M}$ while maintaining connected components.







The Reeb Graph

How to compute $\mathcal{R}$?

Sweep $\mathcal{M}$ while maintaining connected components.

Running time?

Compute & Sort all events:

$O(\tau n^2 \log n)$ time

Initialize graph:

$O(n^2)$ time

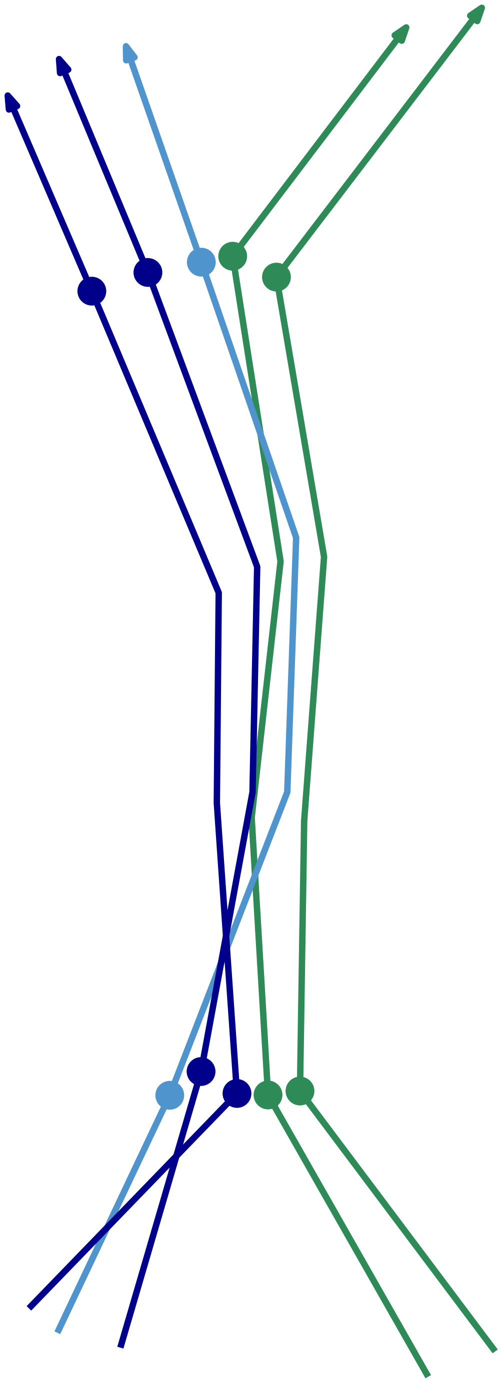
Handle an event:

$O(\log n)$ time

Total: $O(\tau n^2 \log n)$ time

Computing all Groups

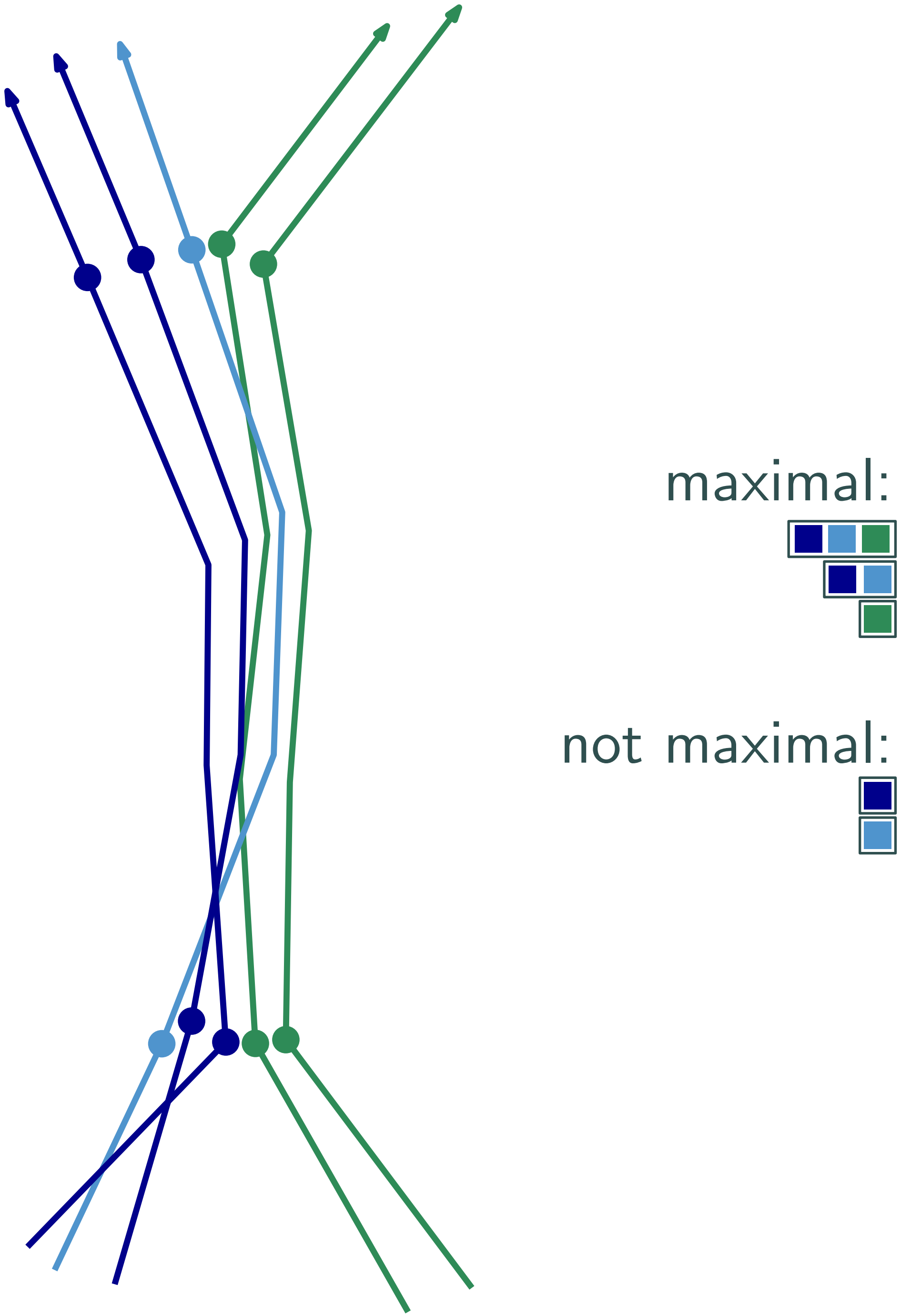
How many groups?



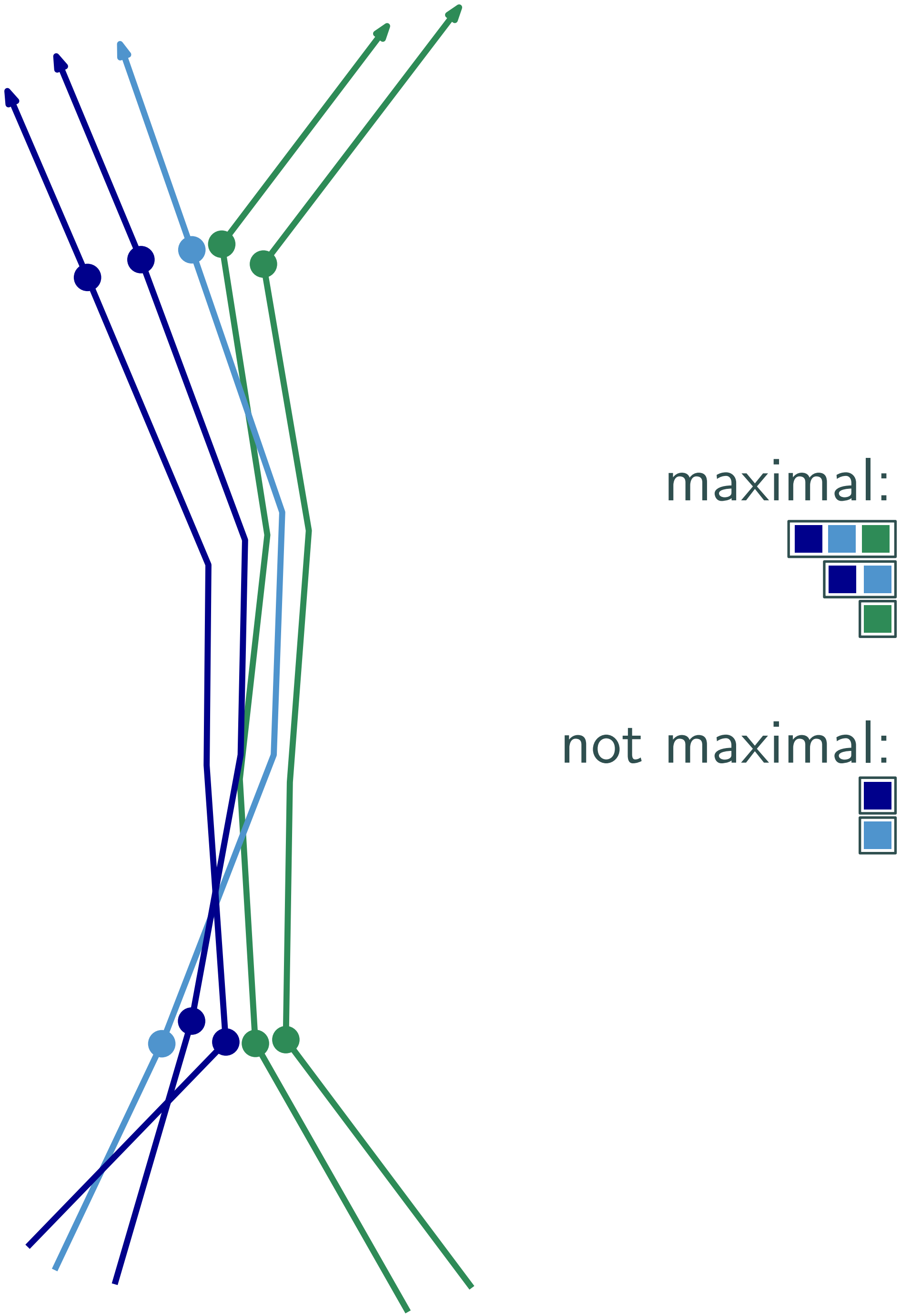
Computing all Groups

How many groups?

Computing all Groups



How many maximal groups?



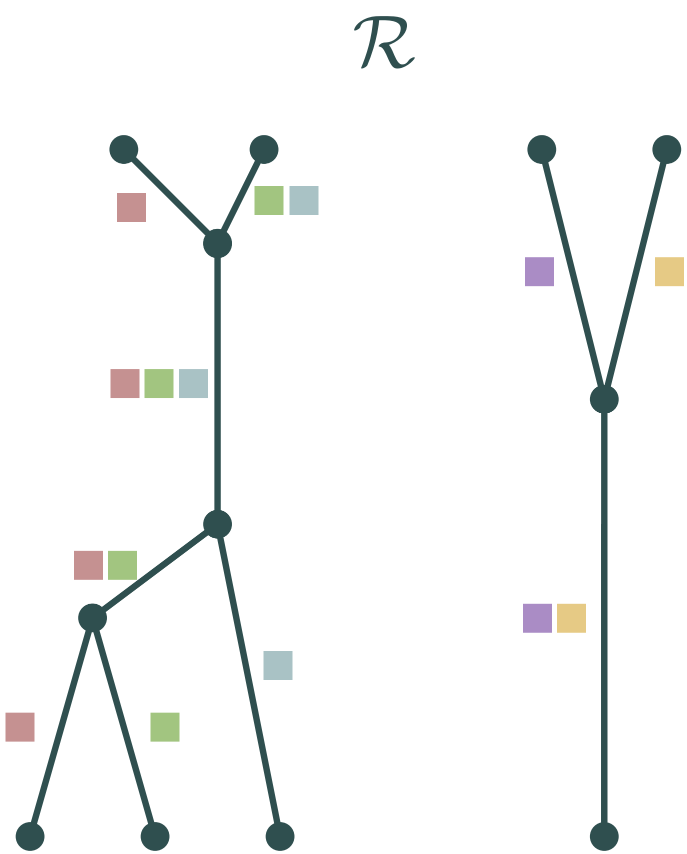
Computing all Groups

How many maximal groups?

At most n per vertex

$\implies \Theta(\tau n^3)$ in total.

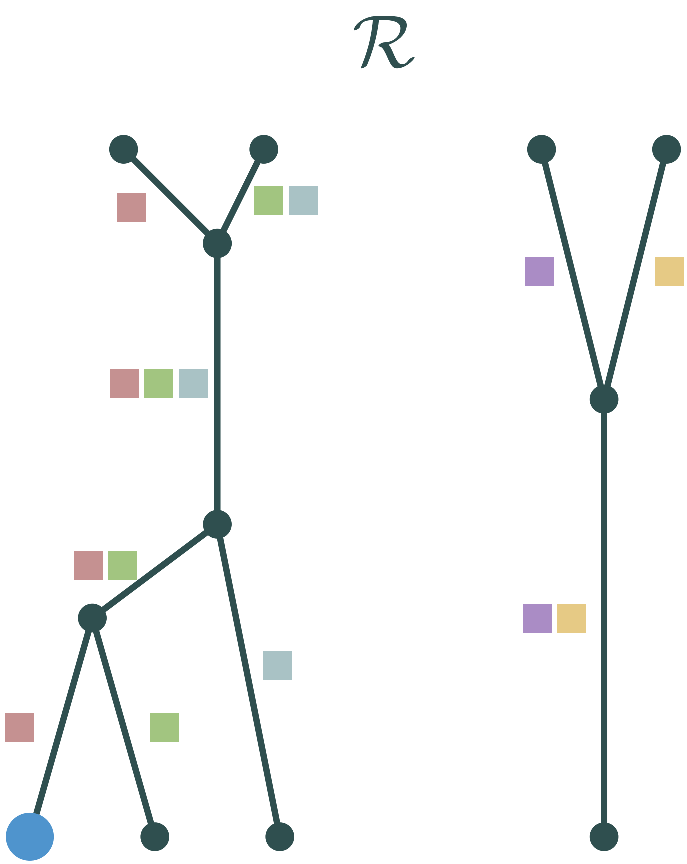
Computing all Groups



How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

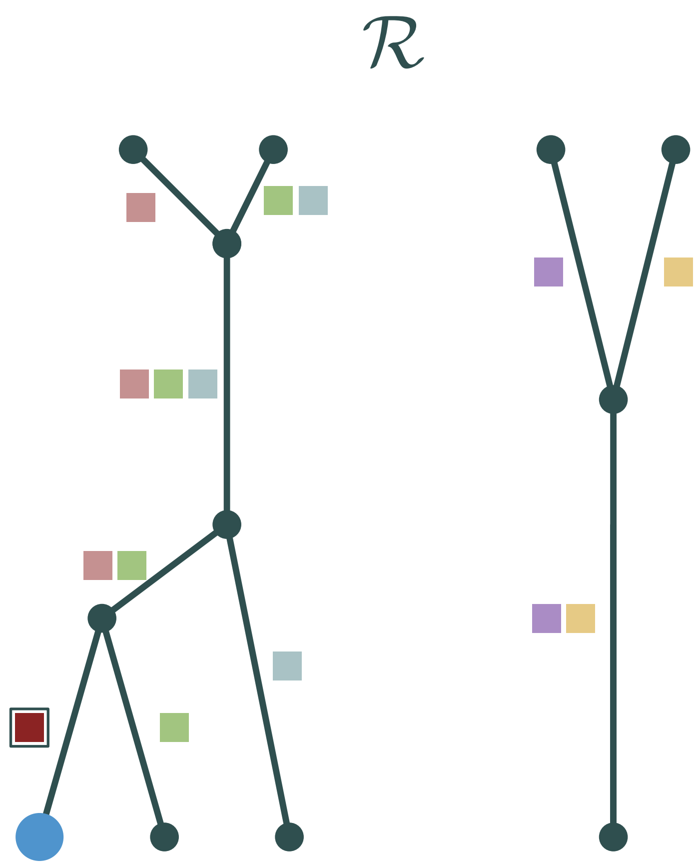


Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

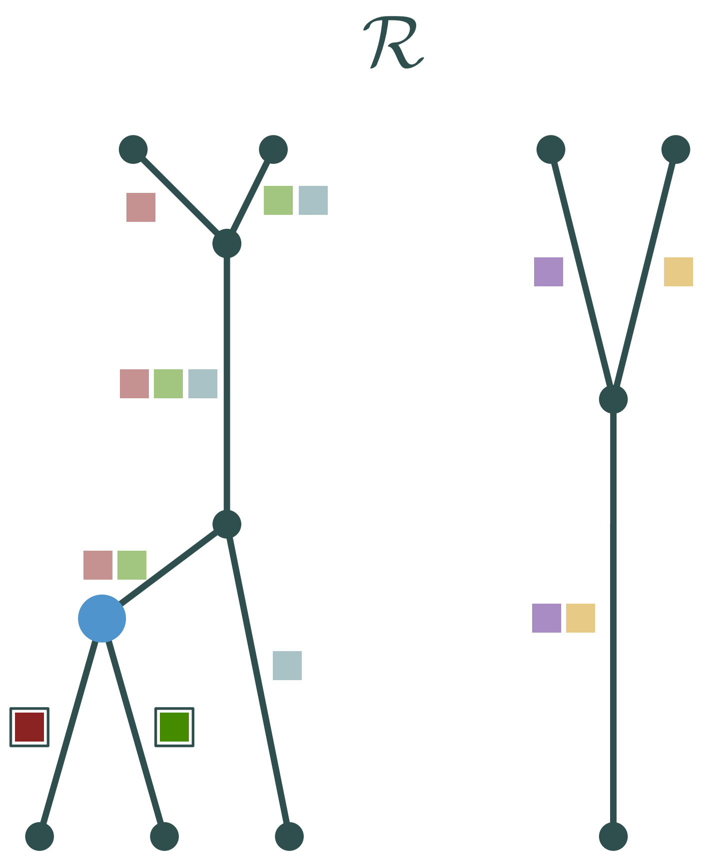


Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

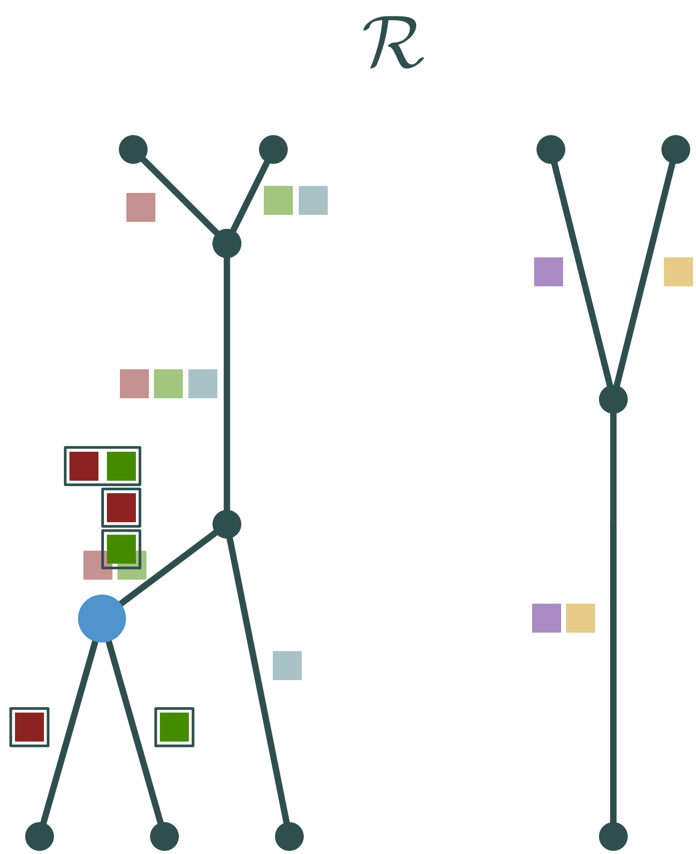


Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

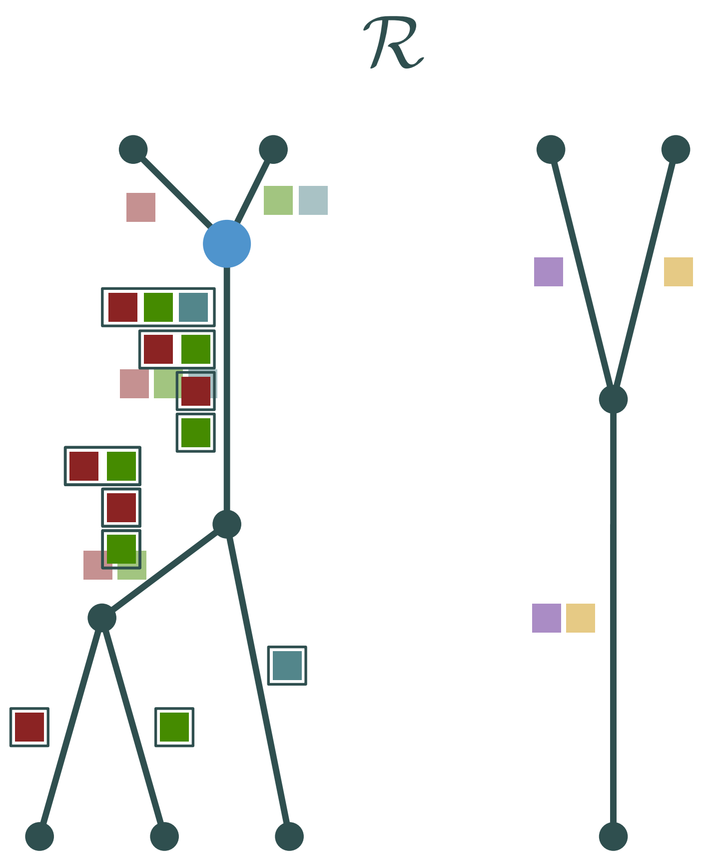


Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

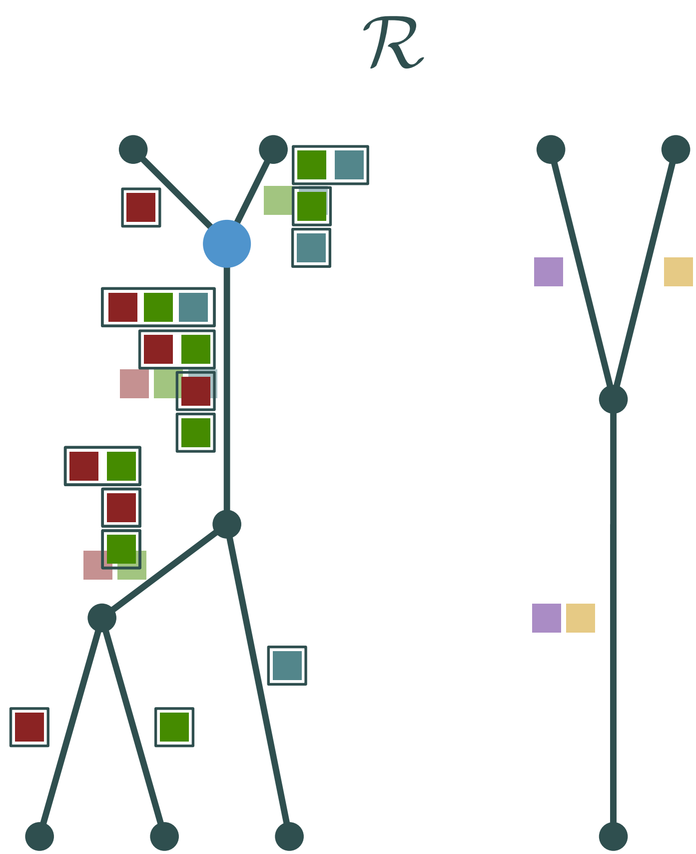


Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

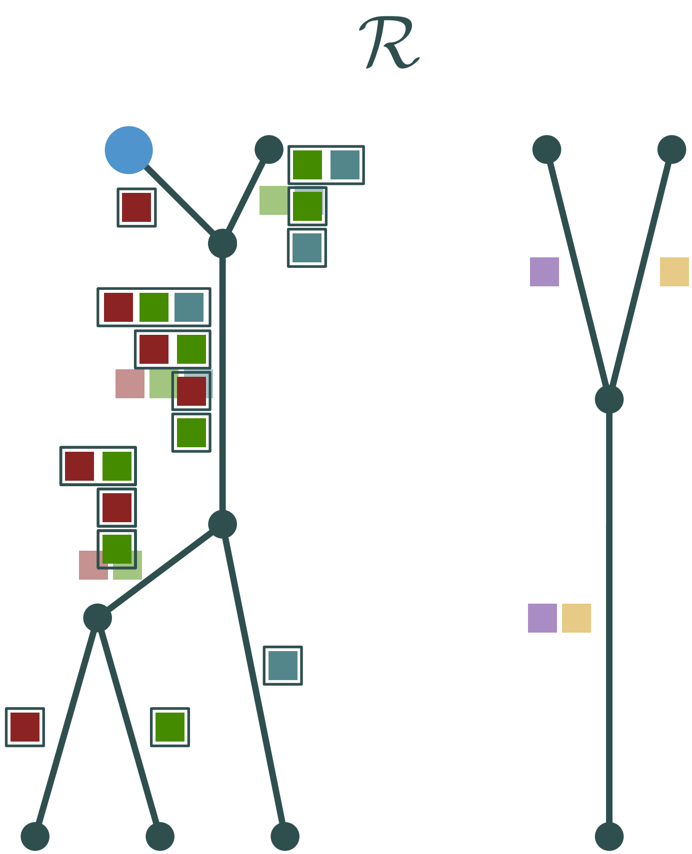


Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

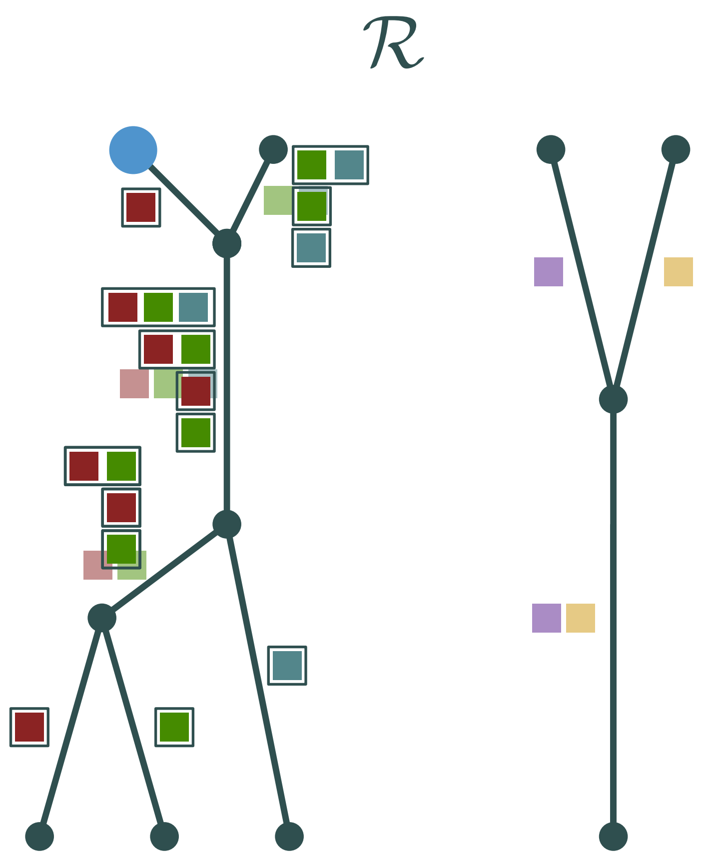


Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.



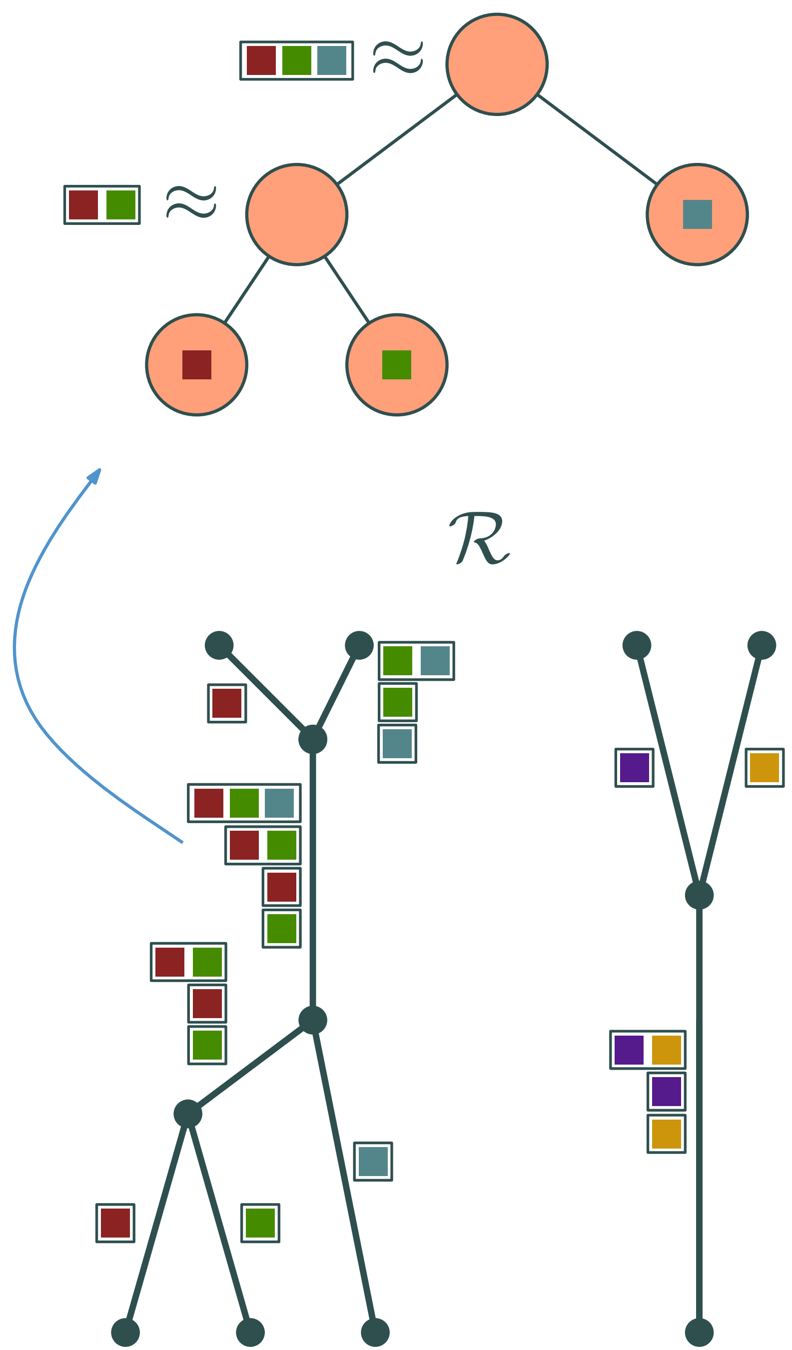
Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

How to represent the groups?



Computing all Groups

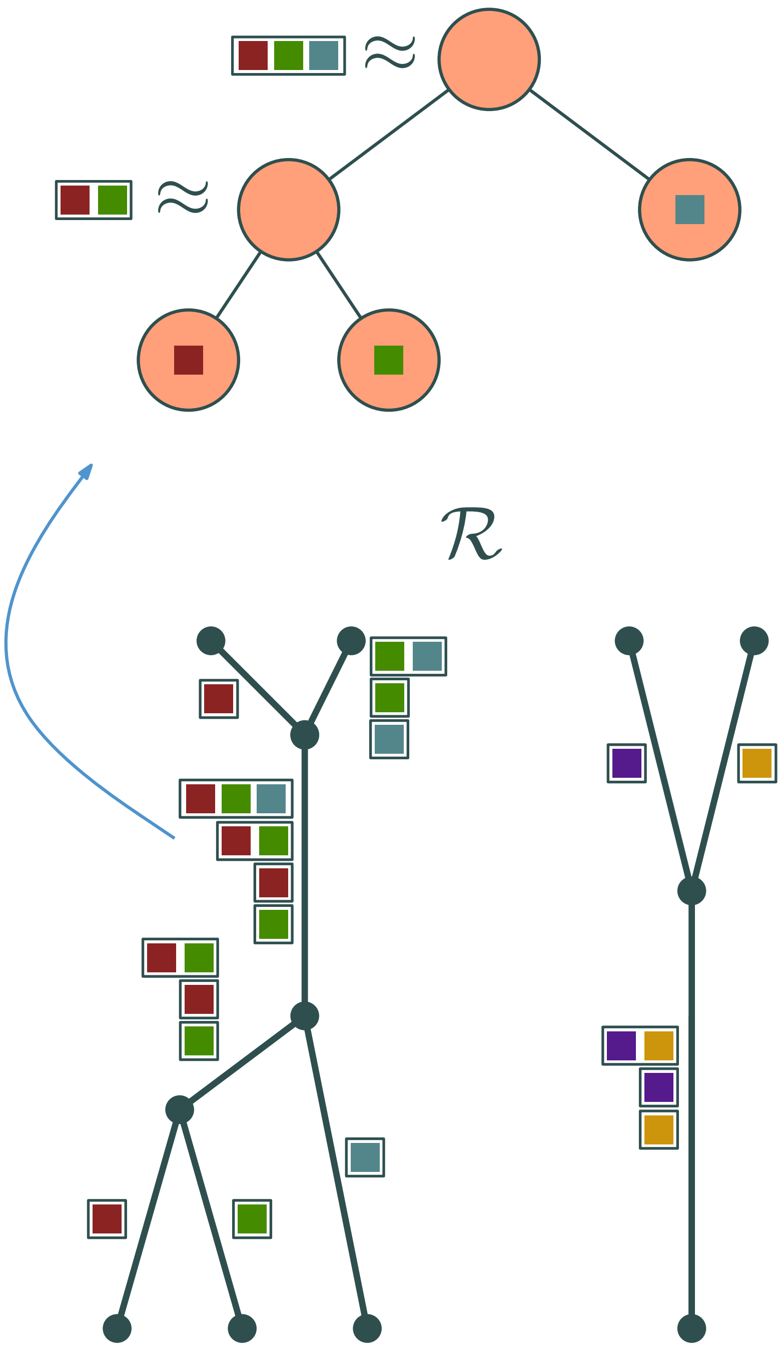
How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

How to represent the groups?

Use a binary tree.



Computing all Groups

How to compute all maximal groups?

with parameters ε, m, and δ

1. Compute $\mathcal{R}$ for the given ε.
2. Process the vertices in time order.
3. Label the outgoing edges with all known maximal groups, assuming $m = 1$ and $\delta = 0$.
4. Filter the groups on δ and m.

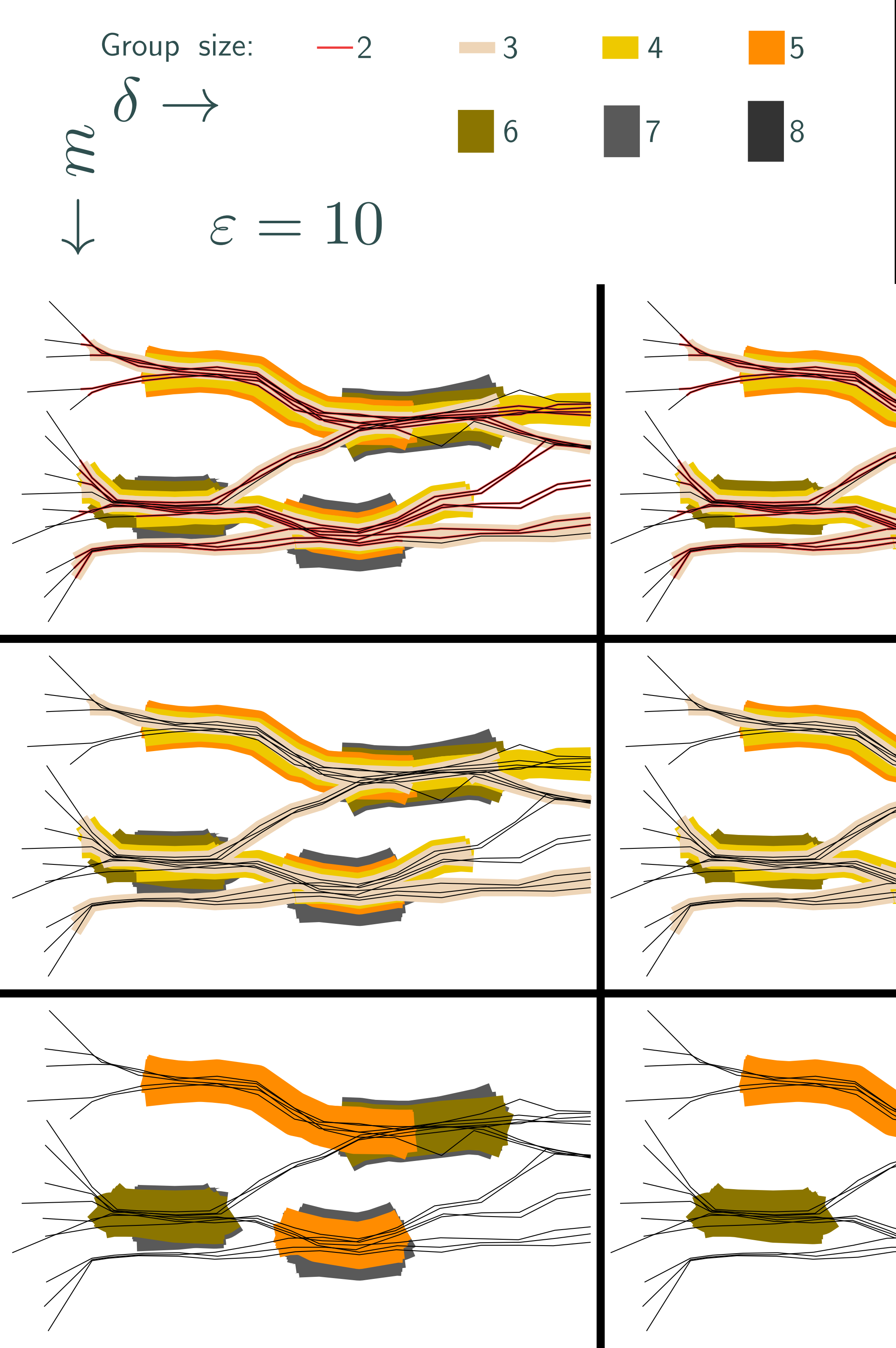
How to represent the groups?

Use a binary tree.

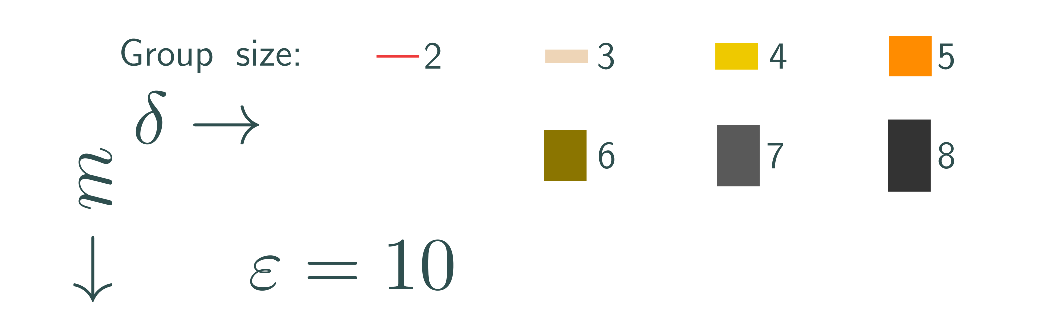
Running time: $O(\tau n^3 + N)$.

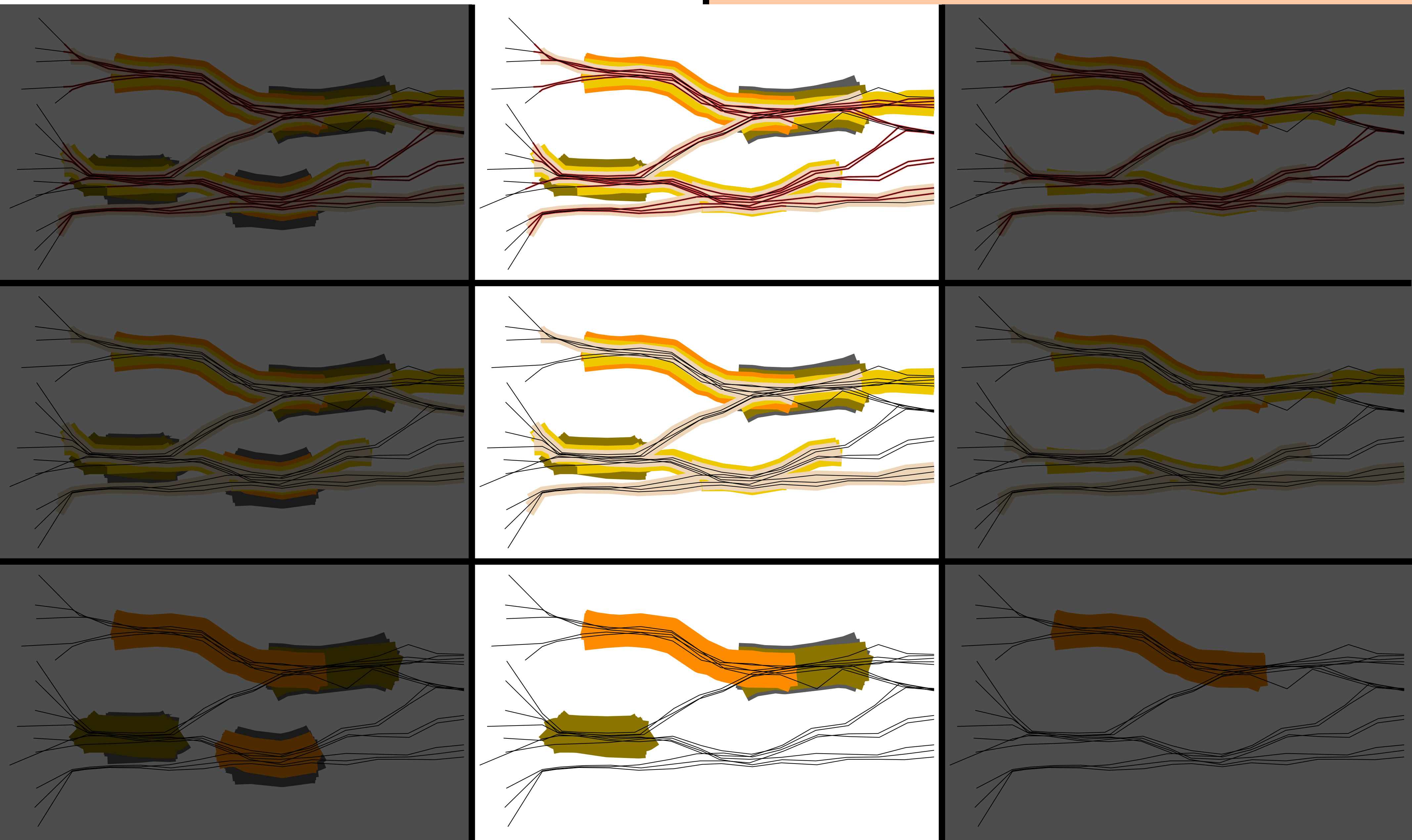
N = total group size

Different Scales

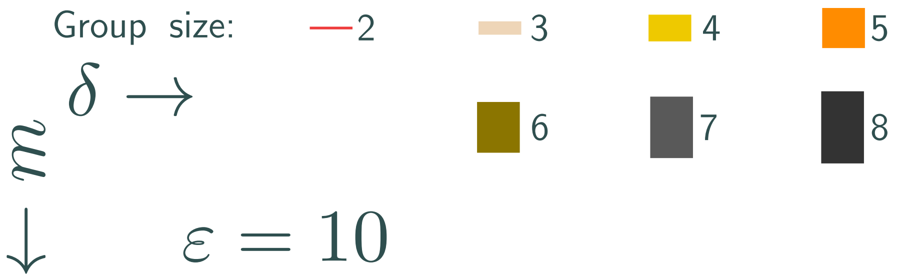


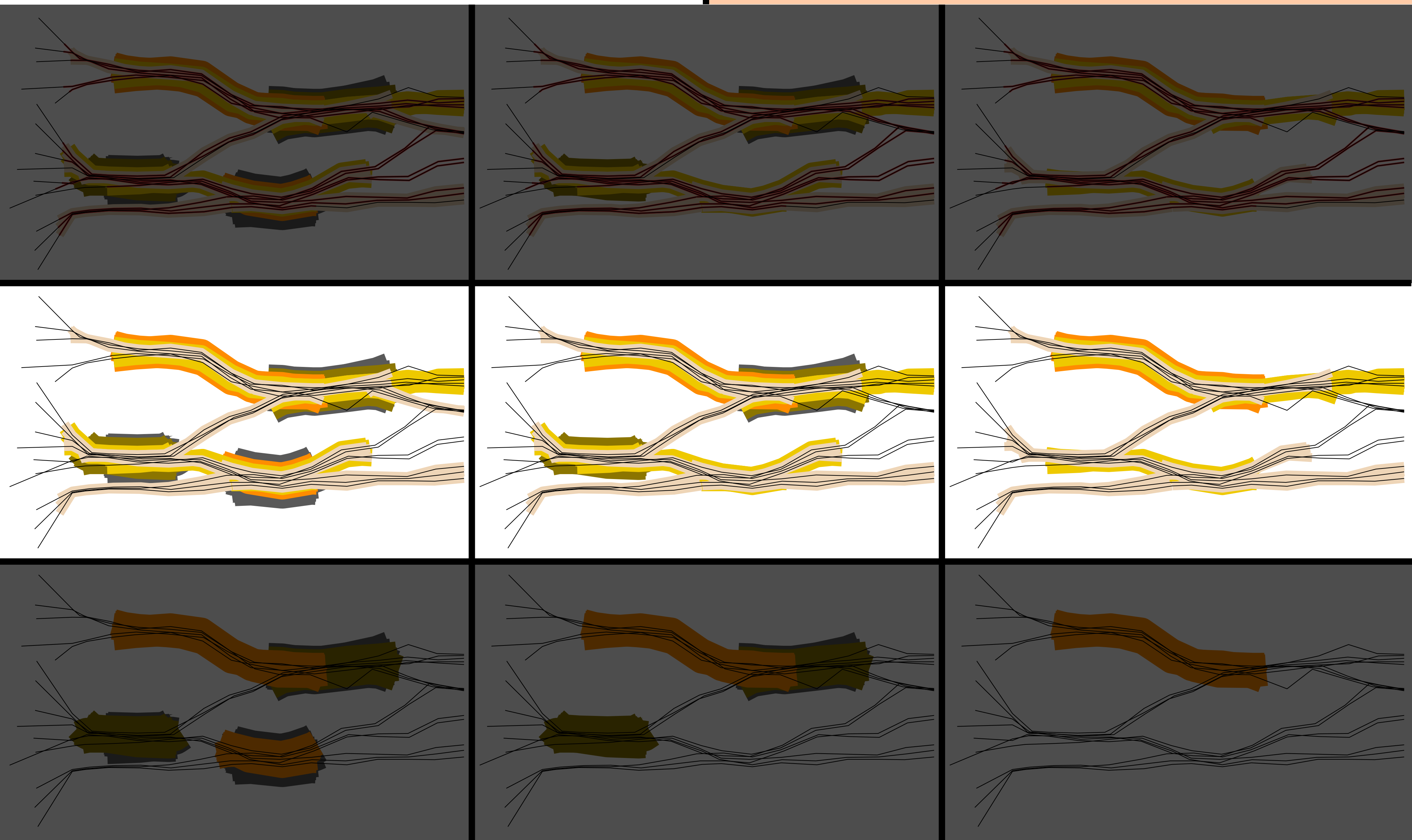
Different Scales





Different Scales





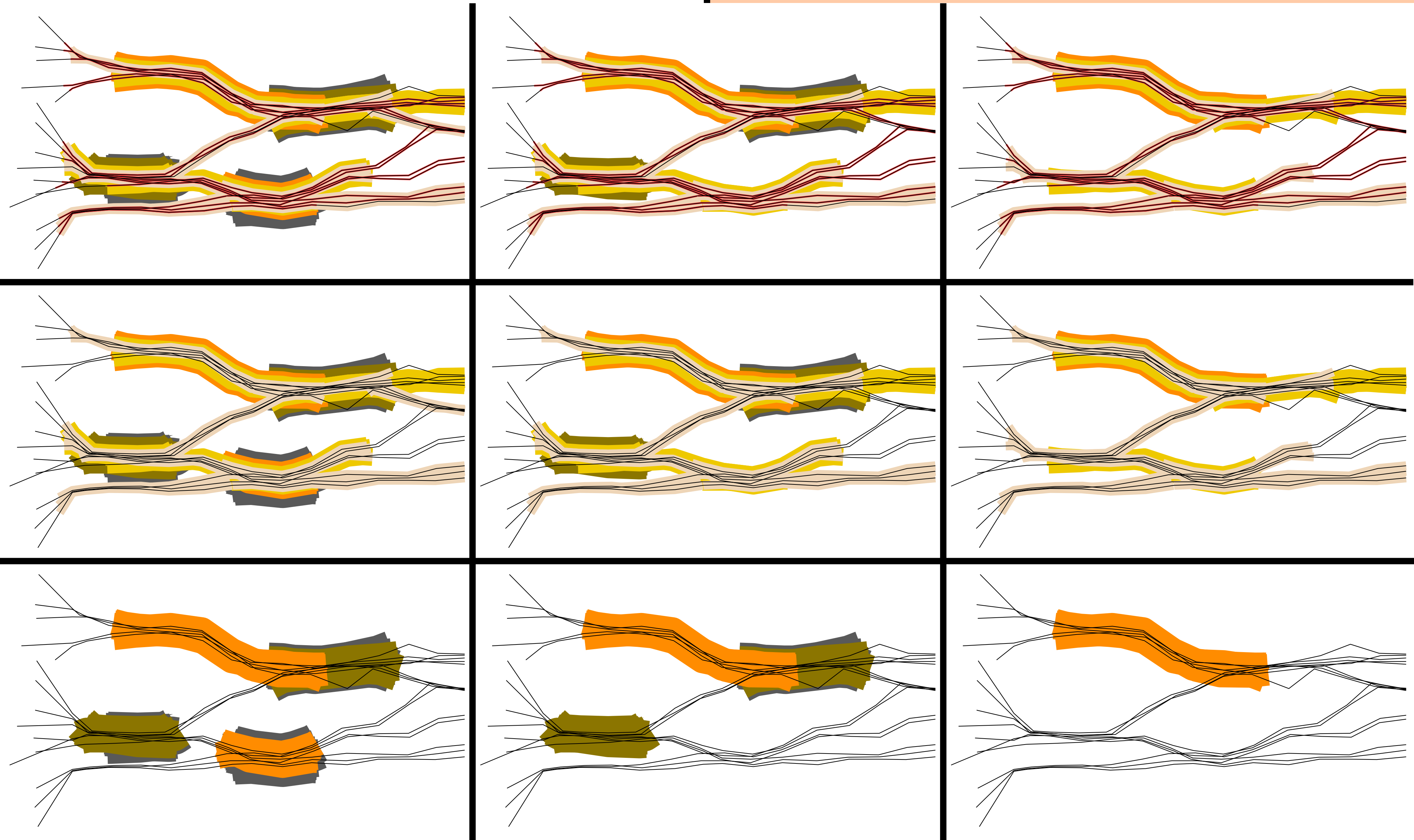
Different Scales

Group size: 2 3 4 5 6 7 8

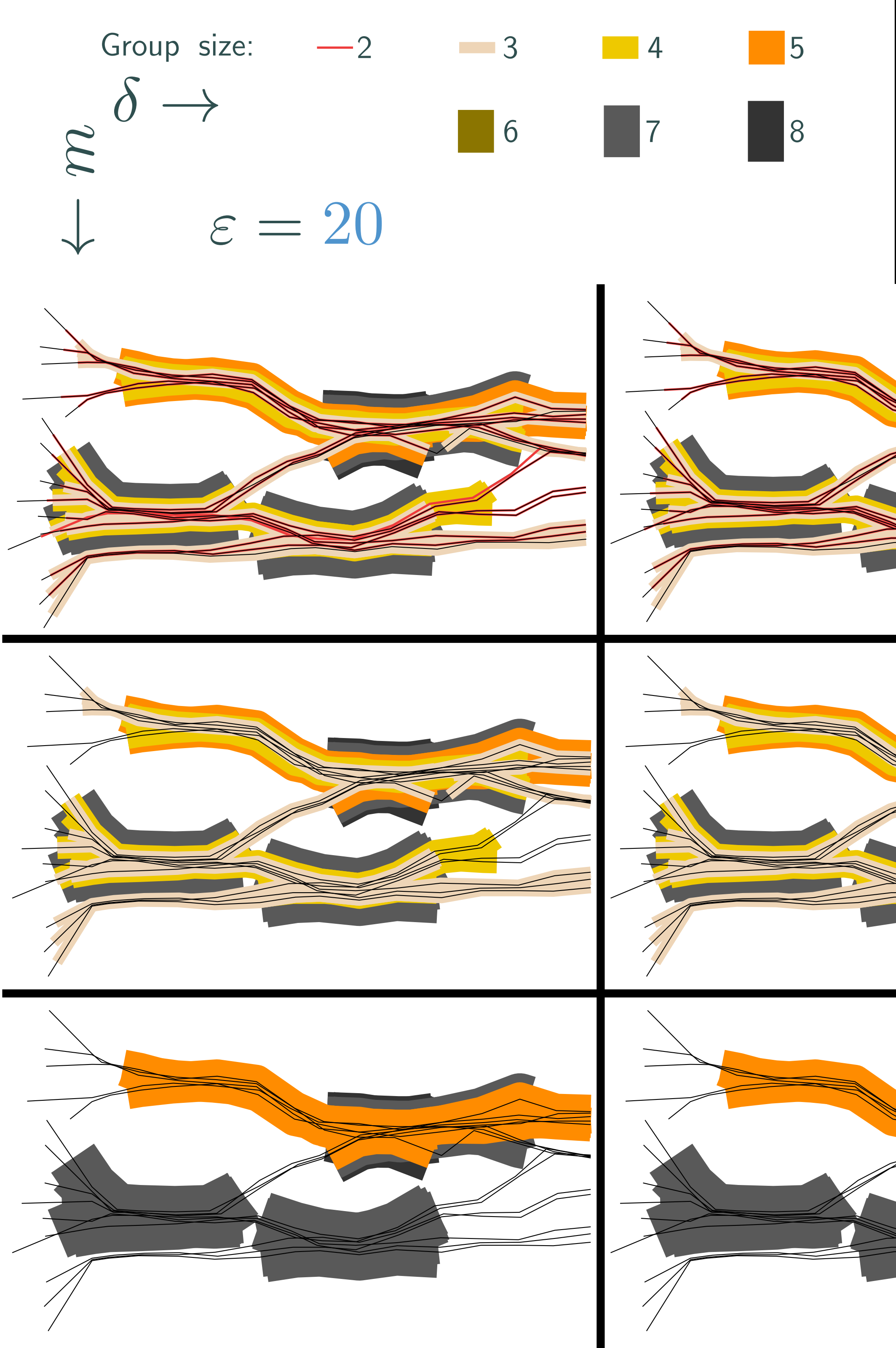
$\delta \rightarrow$

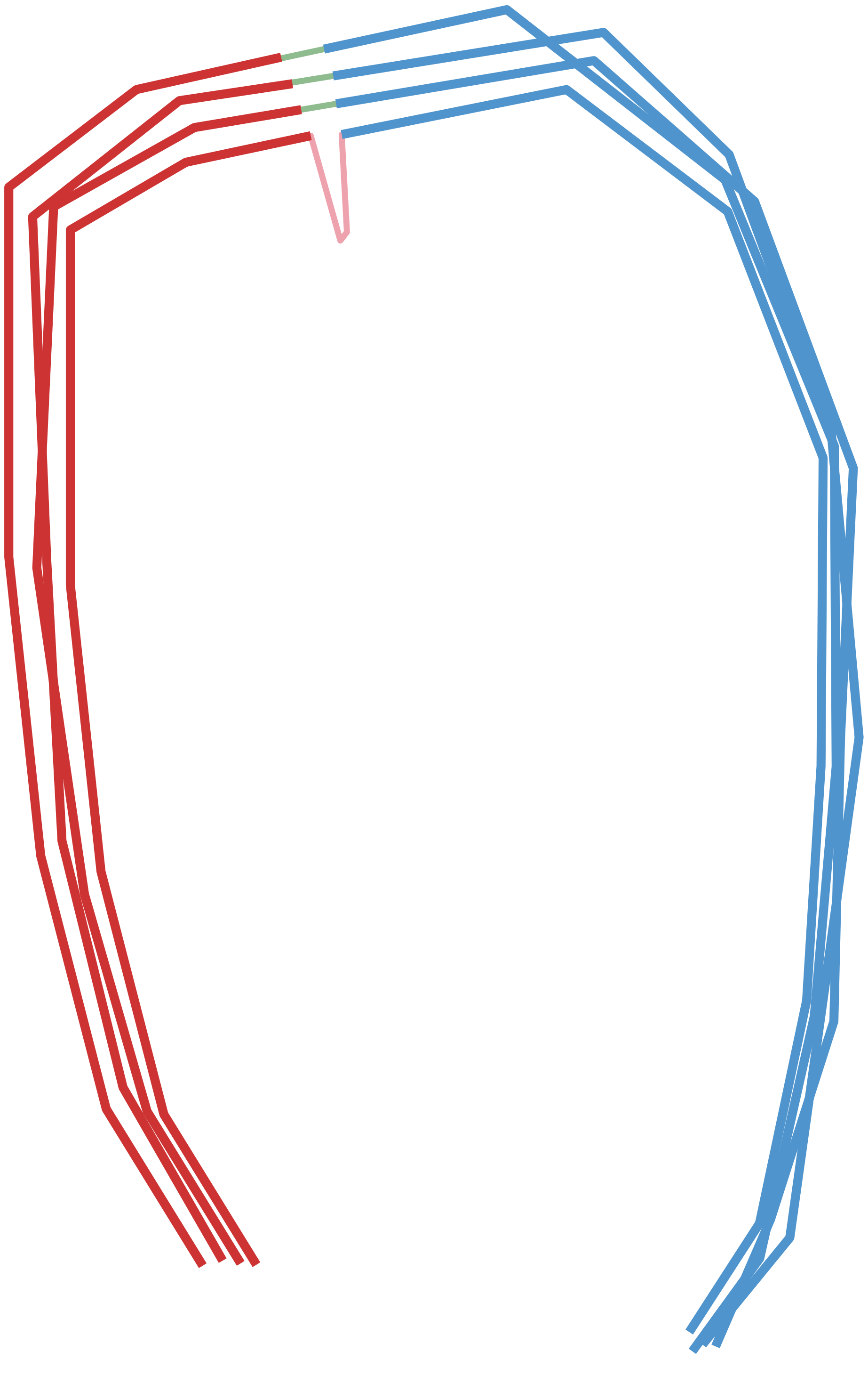
$m \downarrow$

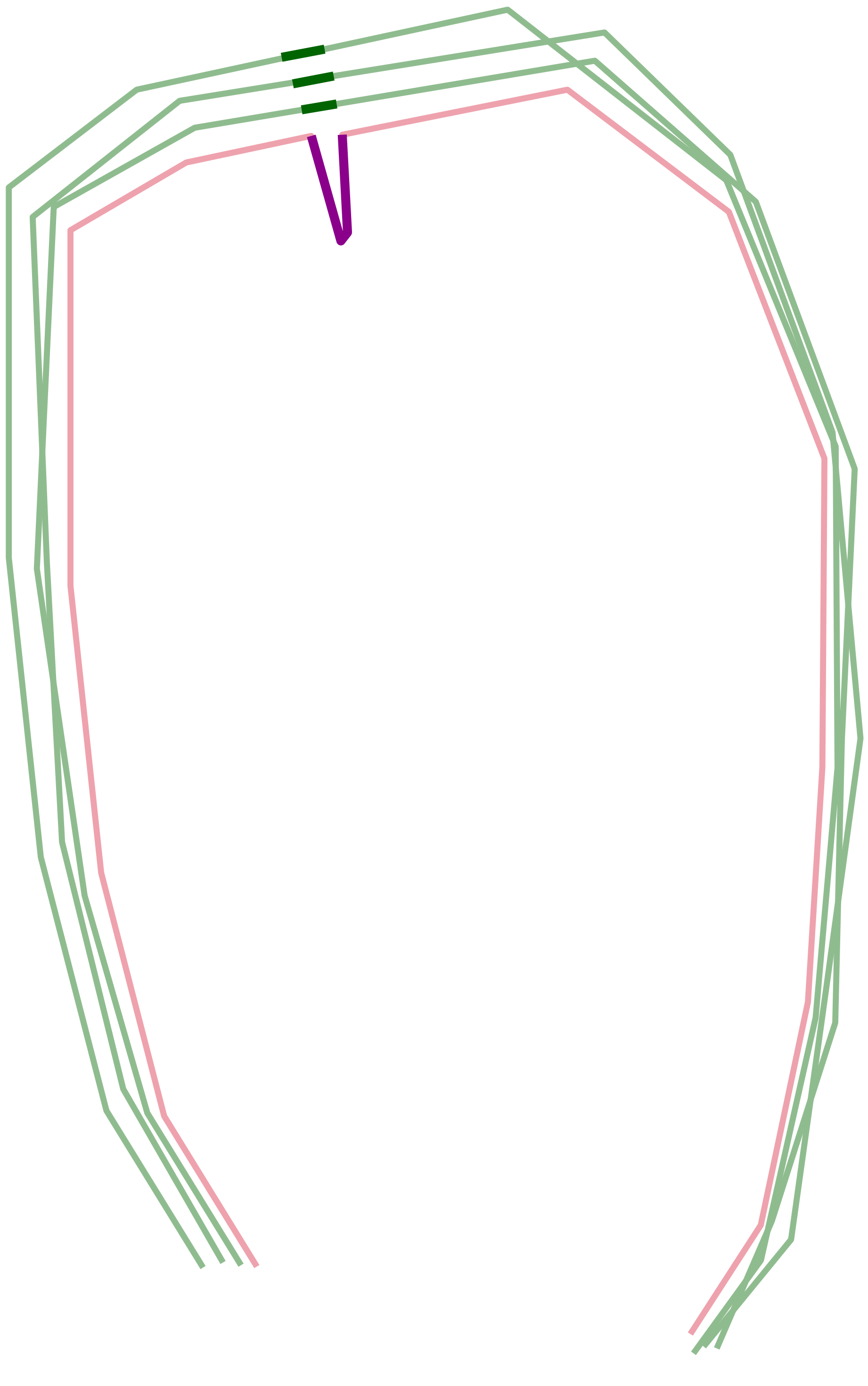
$\varepsilon = 10$

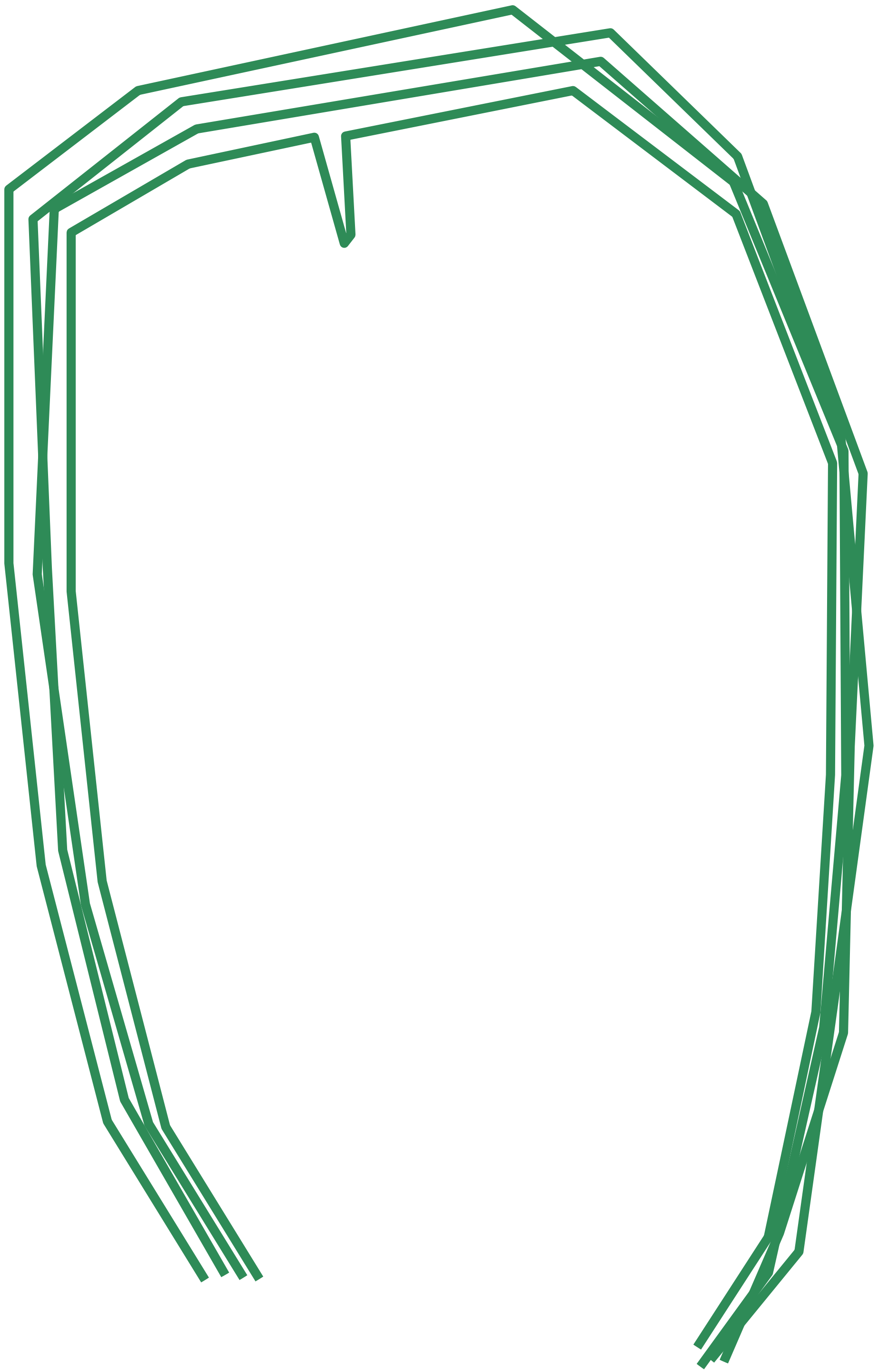


Different Scales

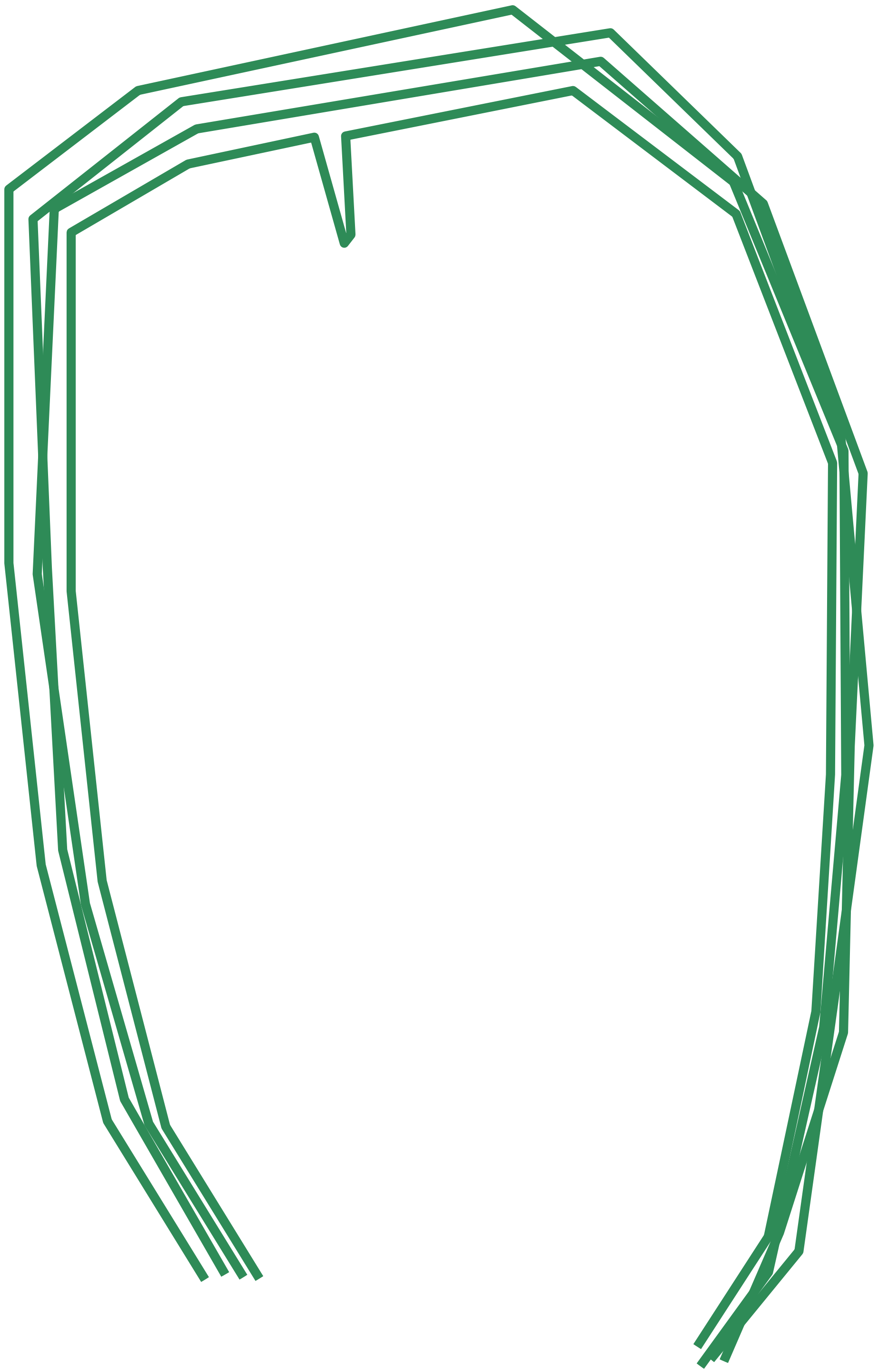








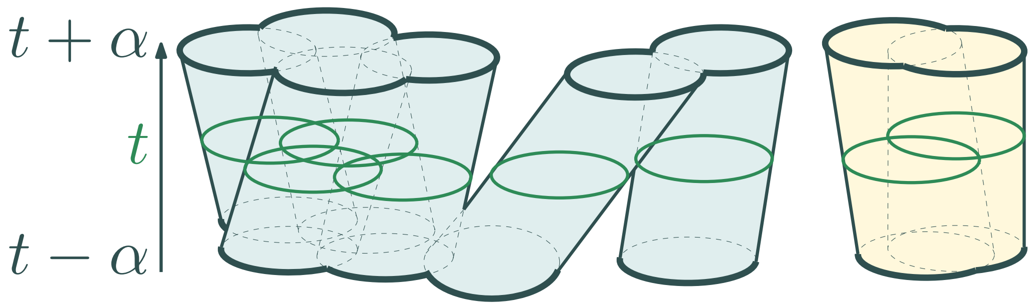
Robustness



Robustness

Introduce a new parameter α

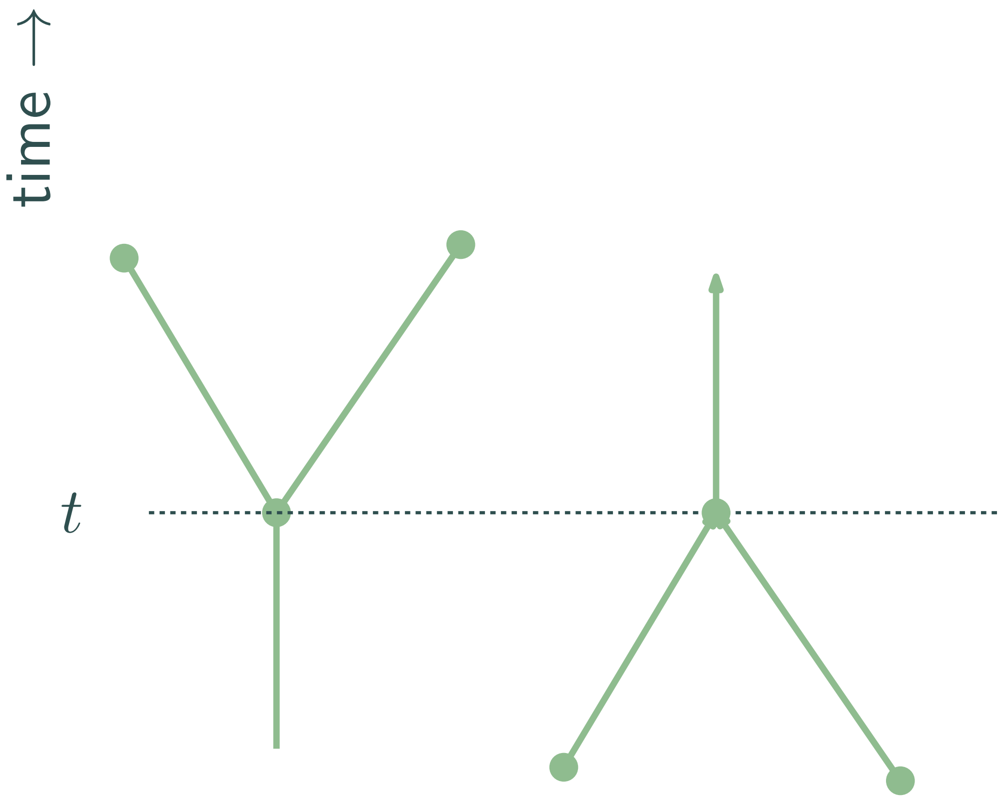
x and y are in one group at time t if they are ε-connected at a time $t' \in [t-\alpha, t+\alpha]$



Robustness

Introduce a new parameter α

x and y are in one group at time t if they are ε-connected at a time $t' \in [t - \alpha, t + \alpha]$



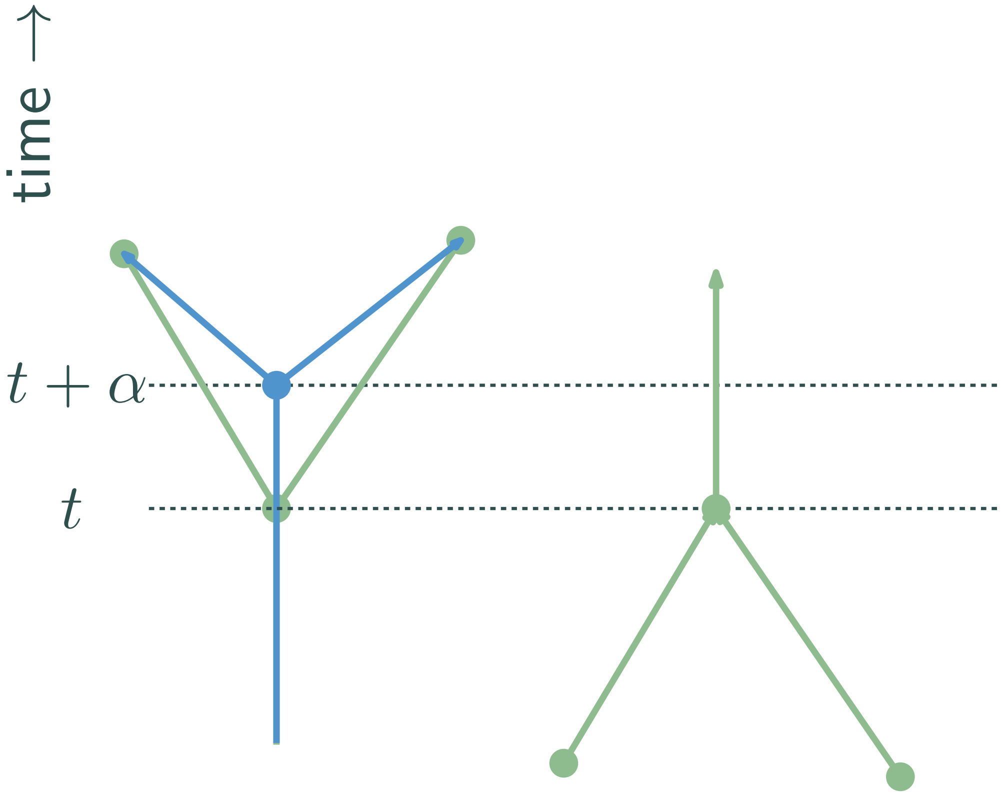
Robustness

Introduce a new parameter α

x and y are in one group at time t if they are ε-connected at a time $t' \in [t - \alpha, t + \alpha]$

$\Longrightarrow$

The vertices in $\mathcal{R}$ move in time.



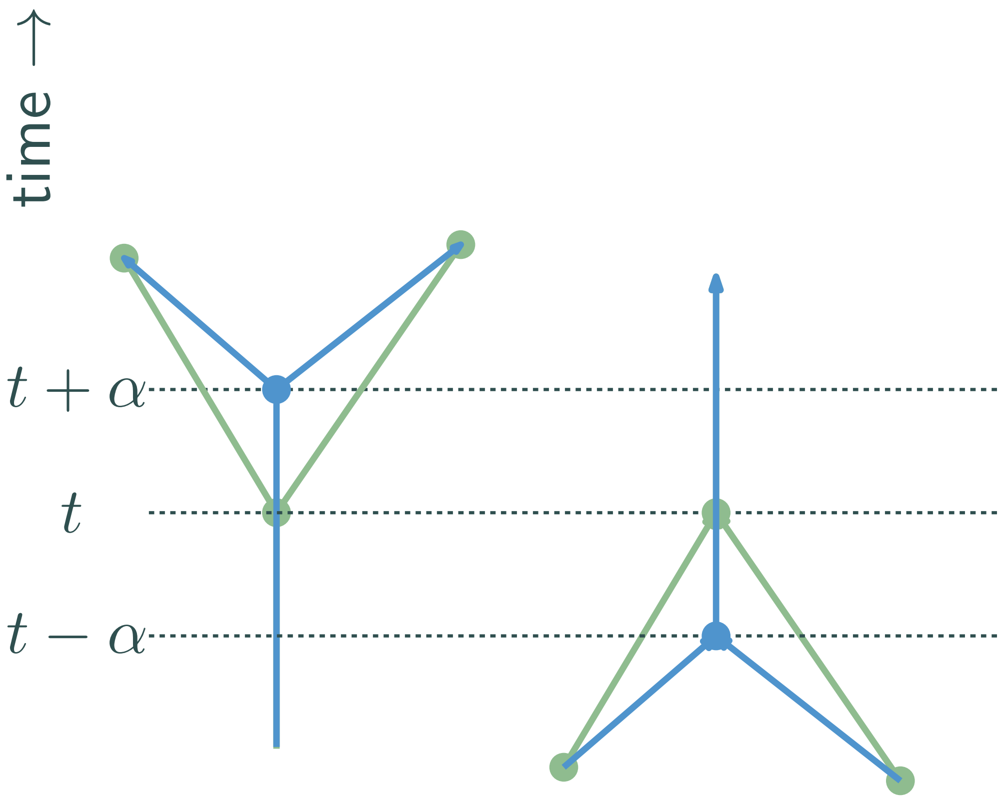
Robustness

Introduce a new parameter α

x and y are in one group at time t if they are ε-connected at a time $t' \in [t-\alpha, t+\alpha]$

$\Longrightarrow$

The vertices in $\mathcal{R}$ move in time.



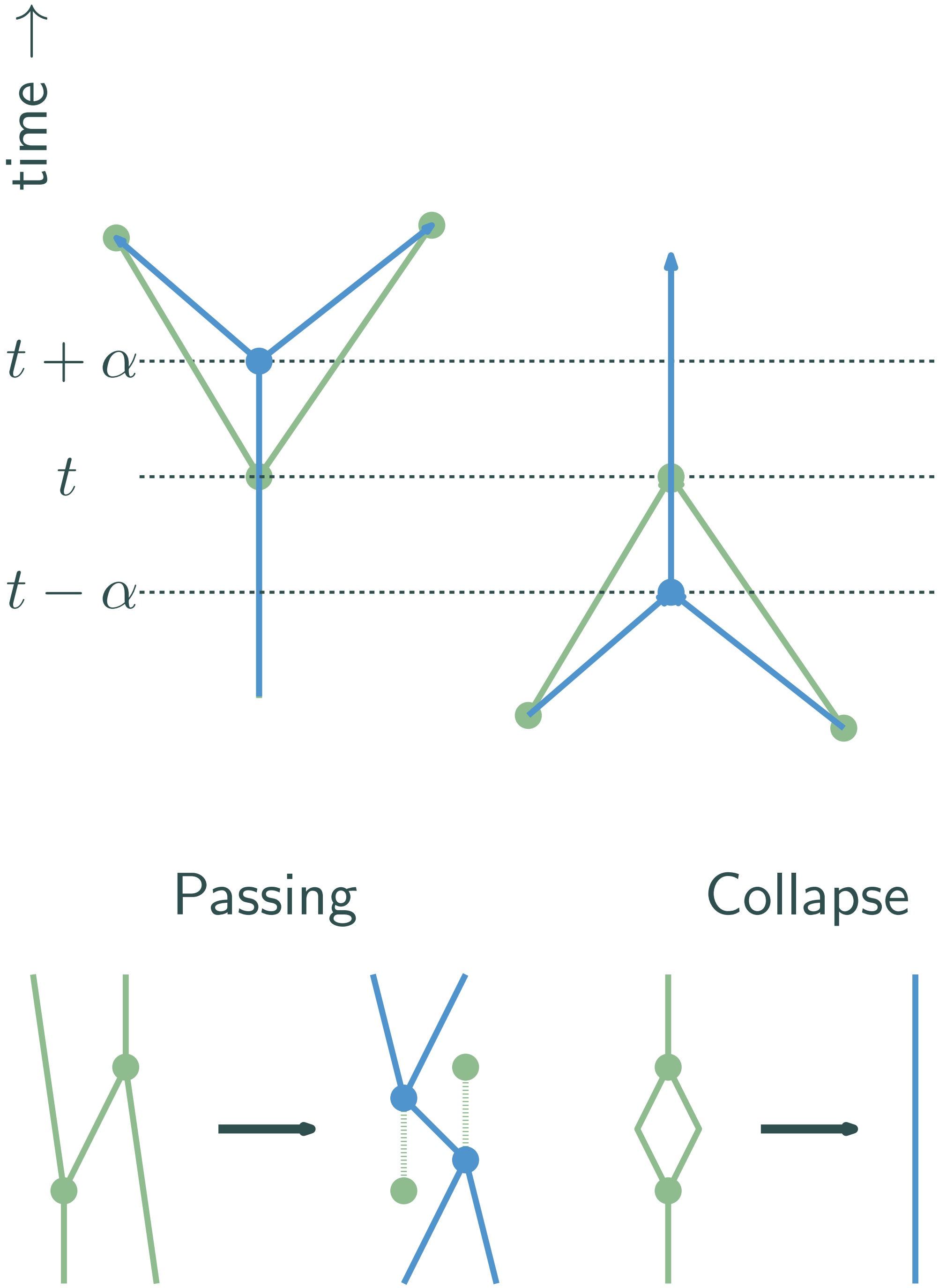
Robustness

Introduce a new parameter α

x and y are in one group at time t if they are ε-connected at a time $t' \in [t-\alpha, t+\alpha]$

$\implies$

The vertices in $\mathcal{R}$ move in time.



Robustness

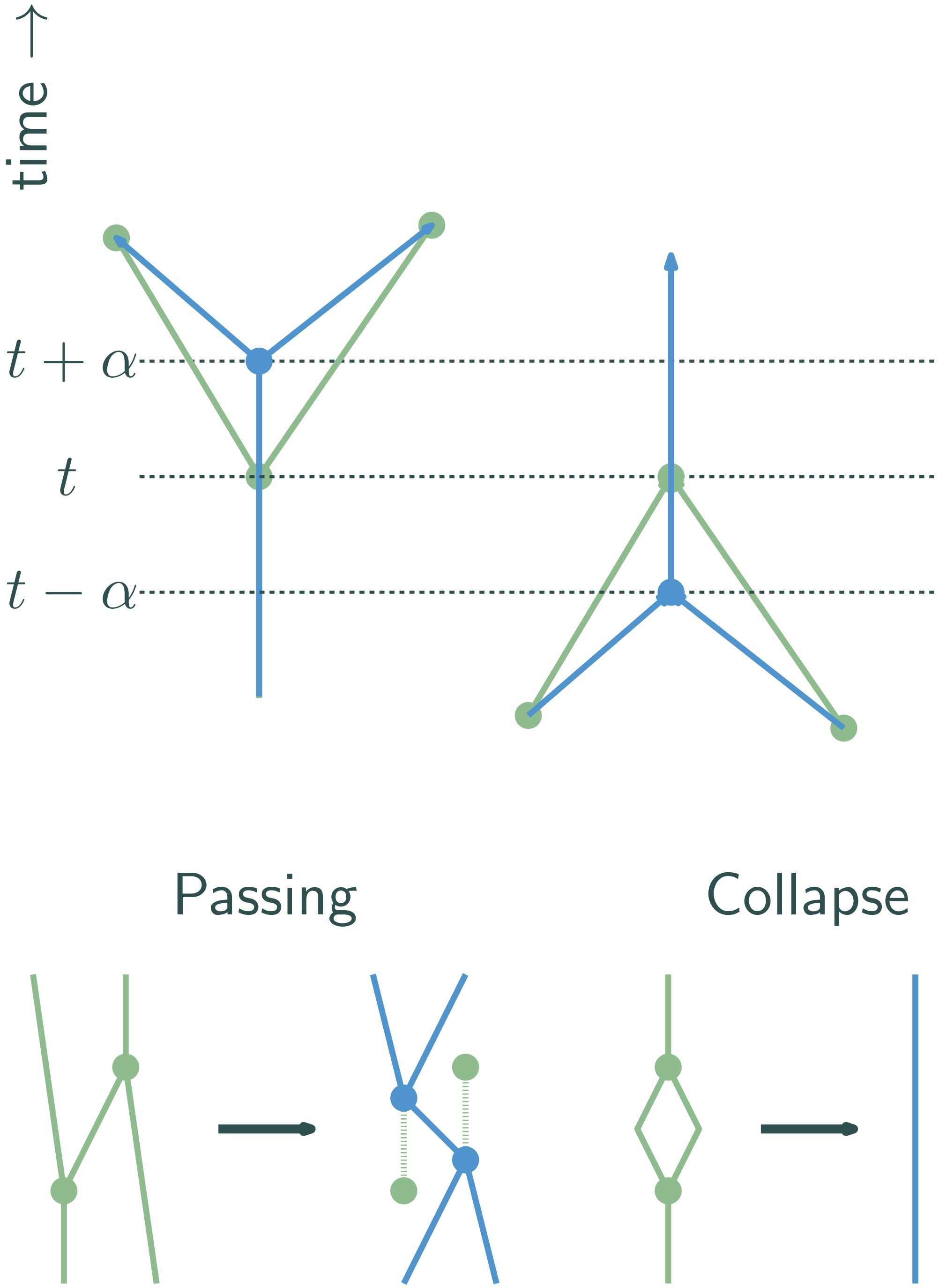
Introduce a new parameter α

x and y are in one group at time t if they are ε-connected at a time $t' \in [t-\alpha, t+\alpha]$

$\implies$

The vertices in $\mathcal{R}$ move in time.

$\mathcal{R}$ changes at encounter events.



Robustness

Introduce a new parameter α

x and y are in one group at time t if they are ε-connected at a time $t' \in [t-\alpha, t+\alpha]$

$\Longrightarrow$

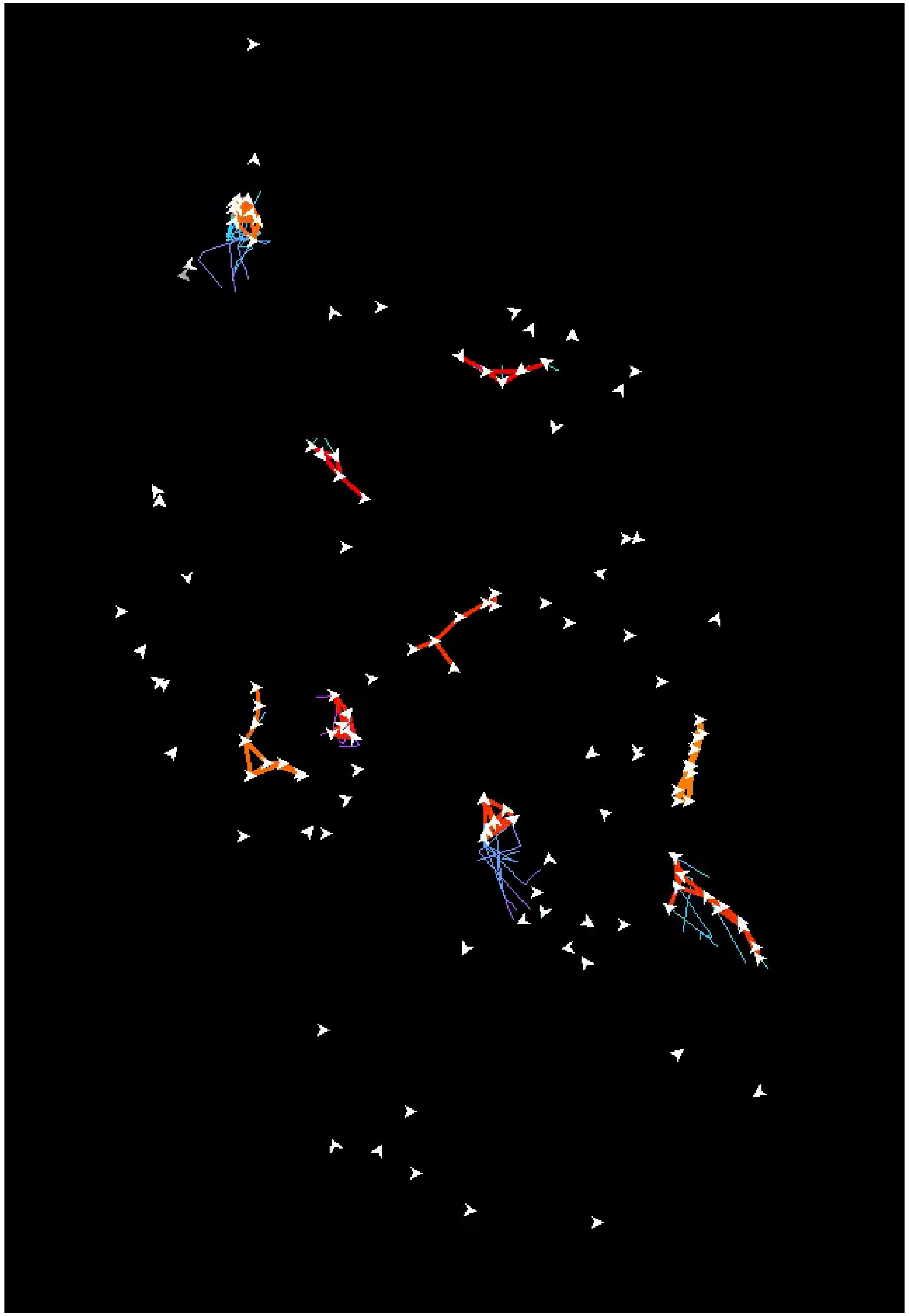
The vertices in $\mathcal{R}$ move in time.

$\mathcal{R}$ changes at encounter events.

#encounter events: $O(\tau n^3)$

we can compute them in $O(\tau n^3 \log n)$ time.

Starkey



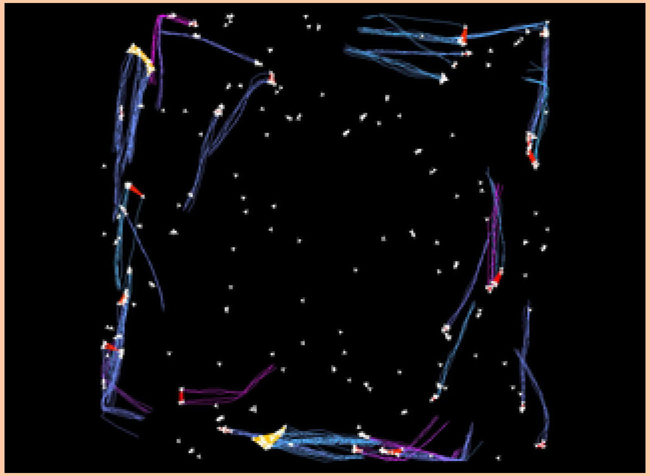
Elk (deer) tracked in Starkey (US).

$n = 126, \tau = 1264$

Evaluation

Evaluation on two data sets: Starkey and NetLogo.

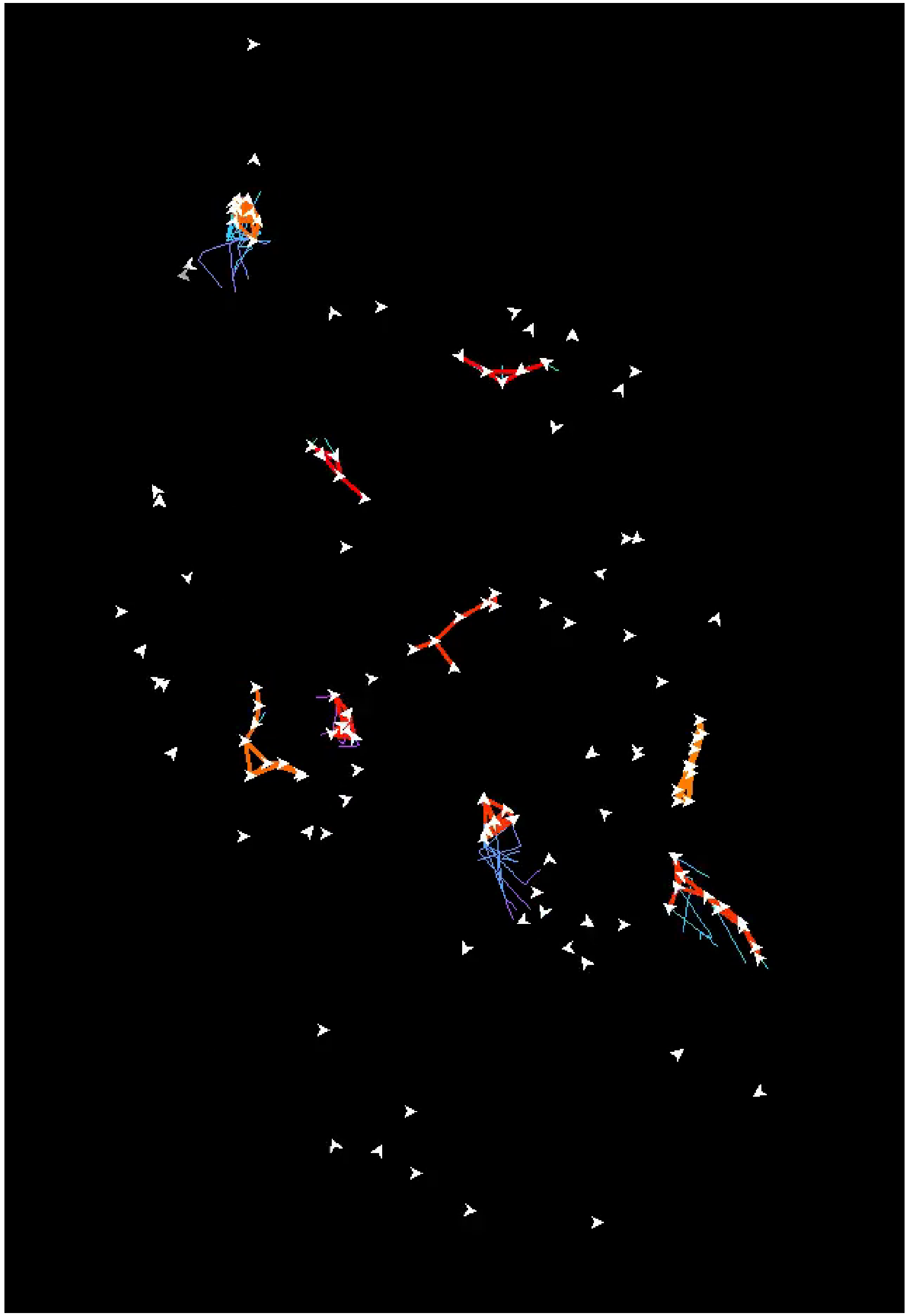
NetLogo



Based on the NetLogo flocking model.

$n = 400, \tau = 818$

Starkey



Elk (deer) tracked in Starkey (US).
$n = 126, \tau = 1264$

Evaluation

Evaluation on two data sets: Starkey and NetLogo.

Group size:

small large

Group duration:

short long

Videos on:

`www.staff.science.uu.nl/~staal006/grouping`

Starkey

Elk (deer) tracked in Starkey (US).
$n = 126, \tau = 1264$

Evaluation

Evaluation on two data sets:
Starkey and NetLogo.

Thank you!

small large

Group duration:

short long

Videos on:

`www.staff.science.uu.nl/~staal006/grouping`