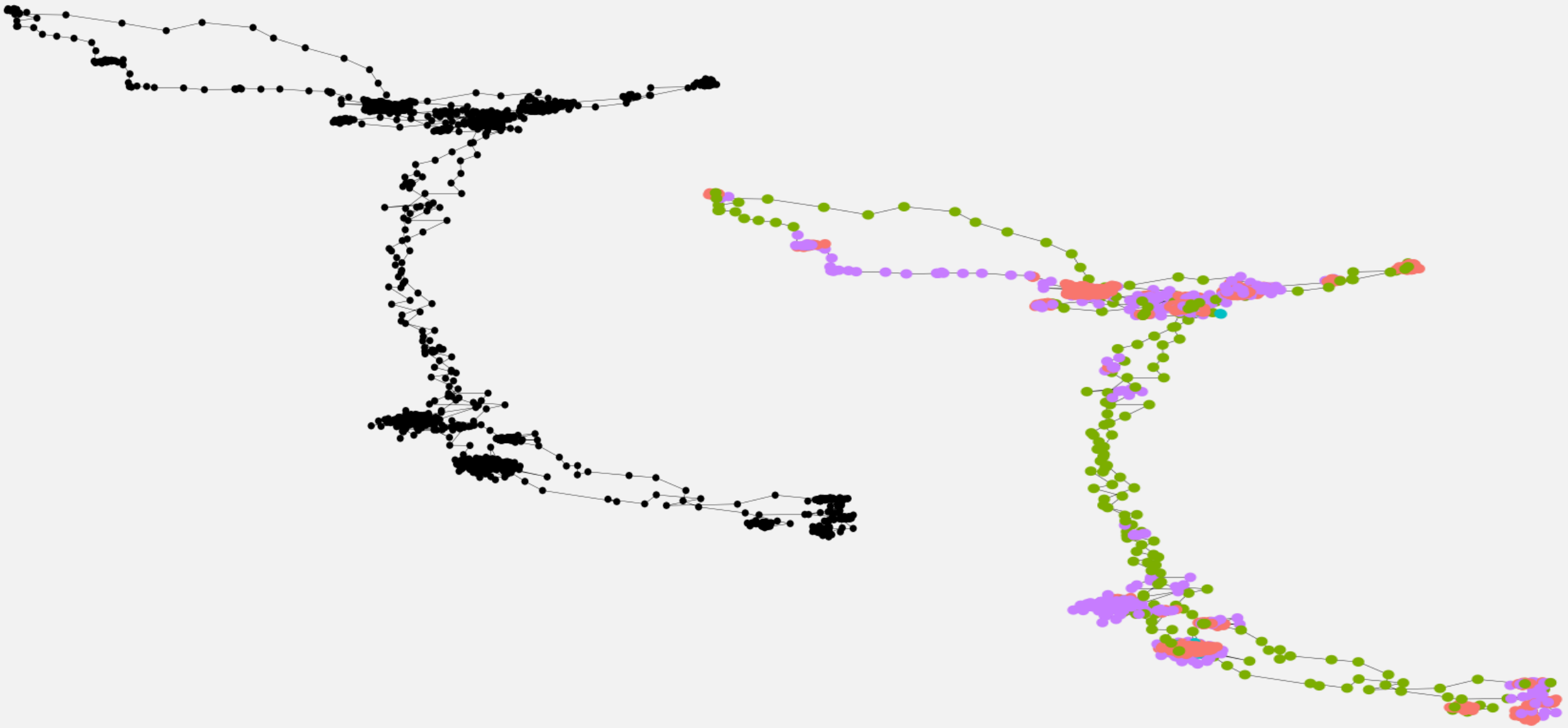
Movement Segmentation

Using Machine Learning

Python & R pipeline

Omer Zlotnick, Spiegel lab

Movement segmentation



How far do animals move?

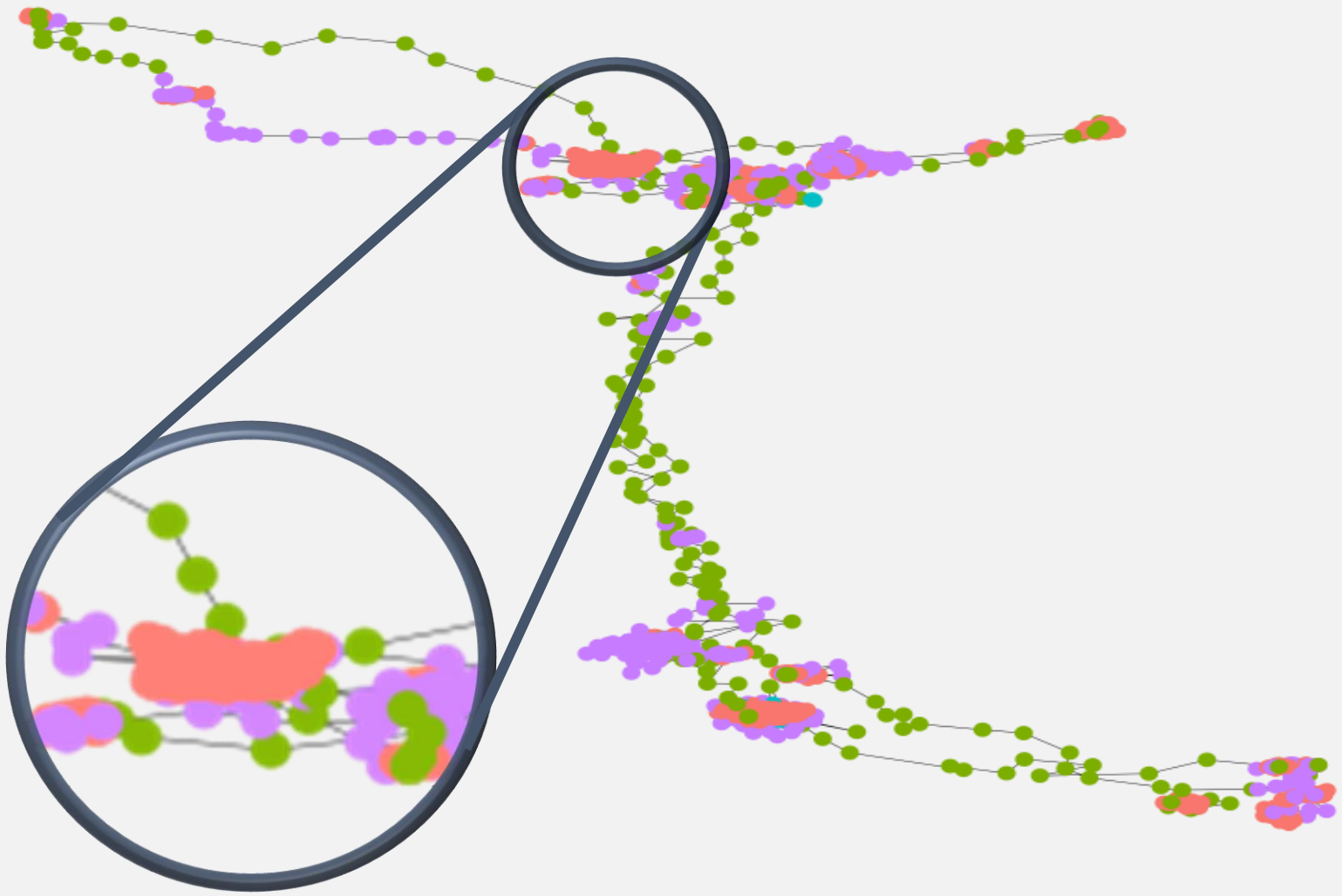
Where do animals spend their time?

Is there variance in movement between individuals?

How do animals move in each state?

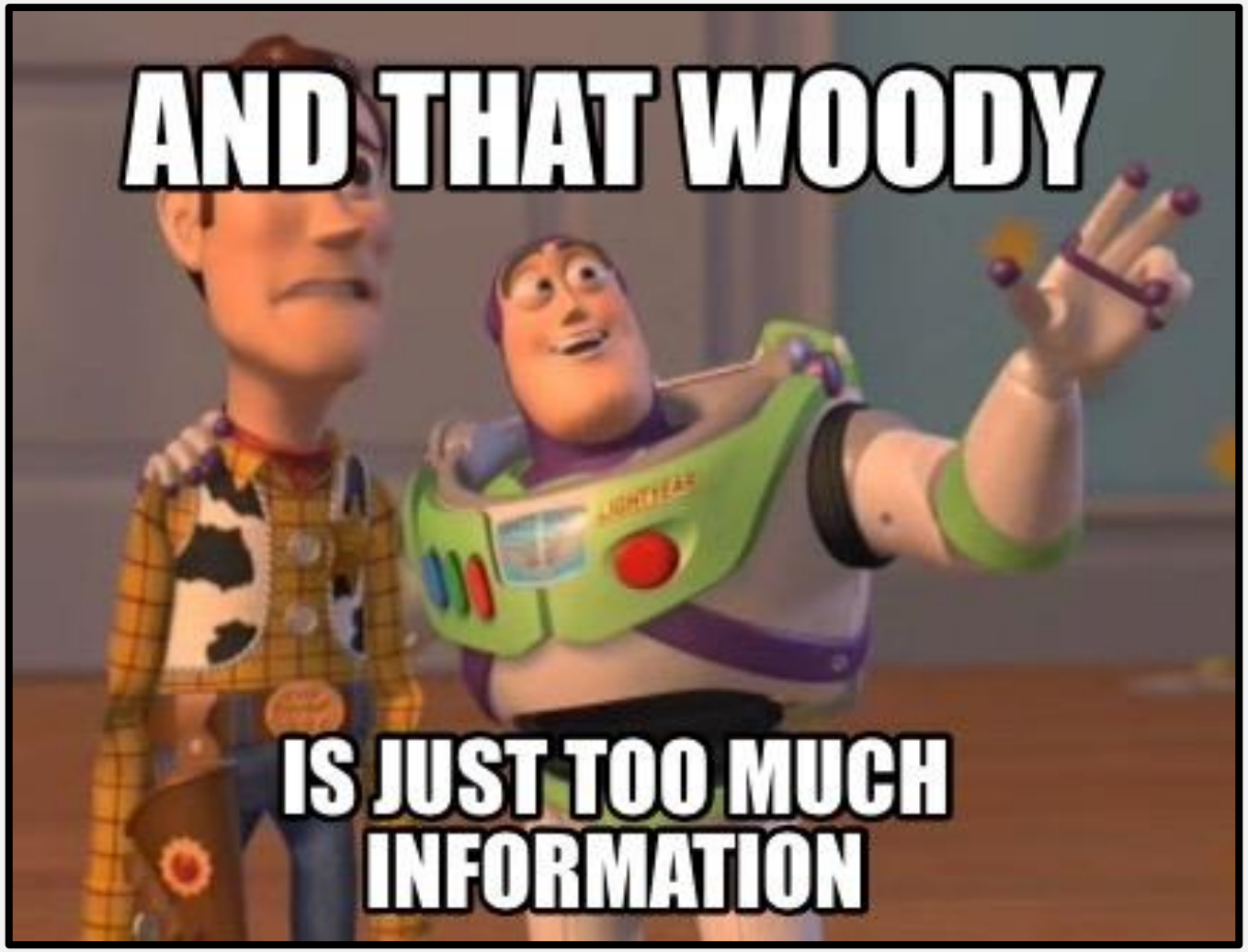
Where do animals prefer to perform each behavior?

Does the same state look different between individuals?

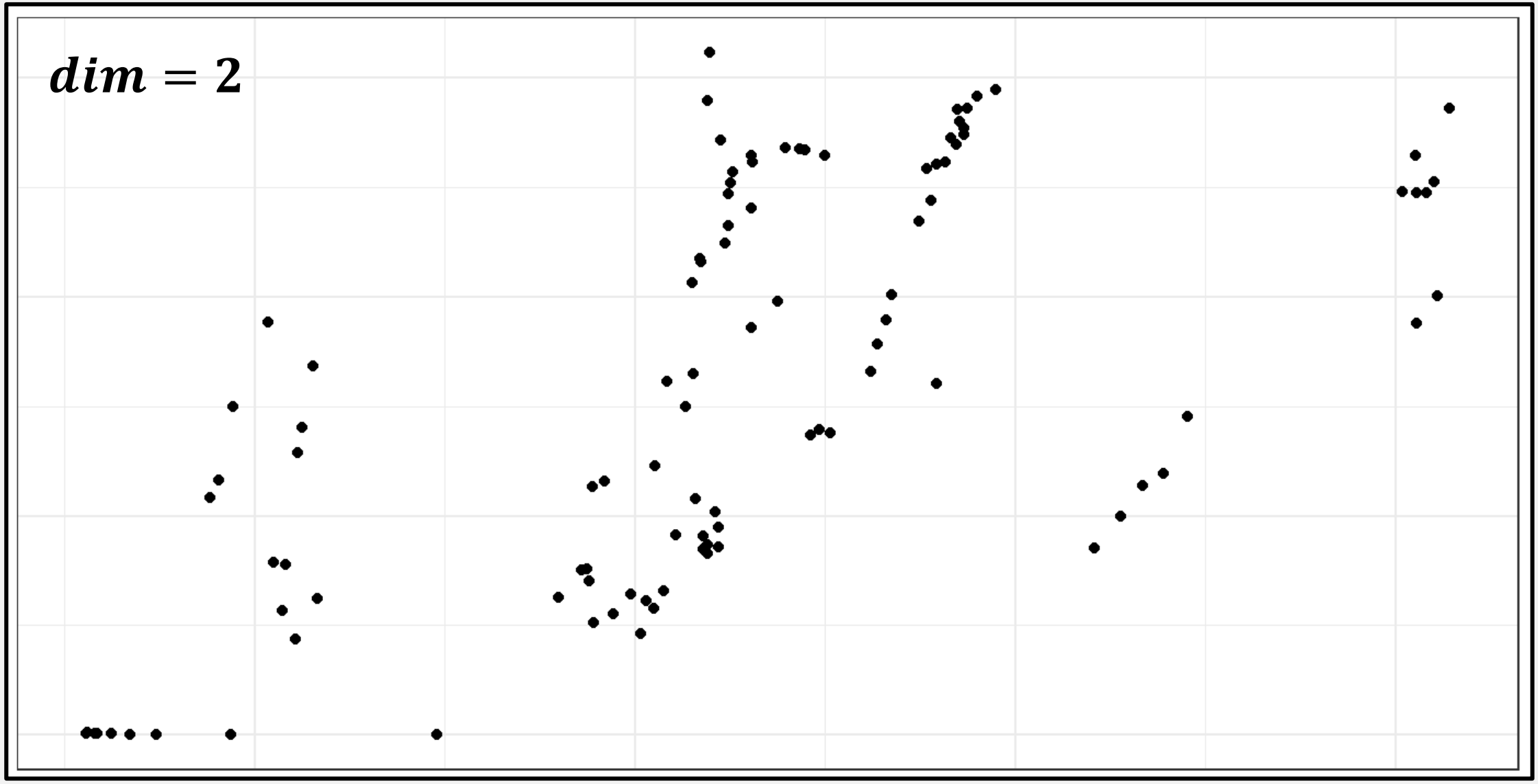


How to segment?

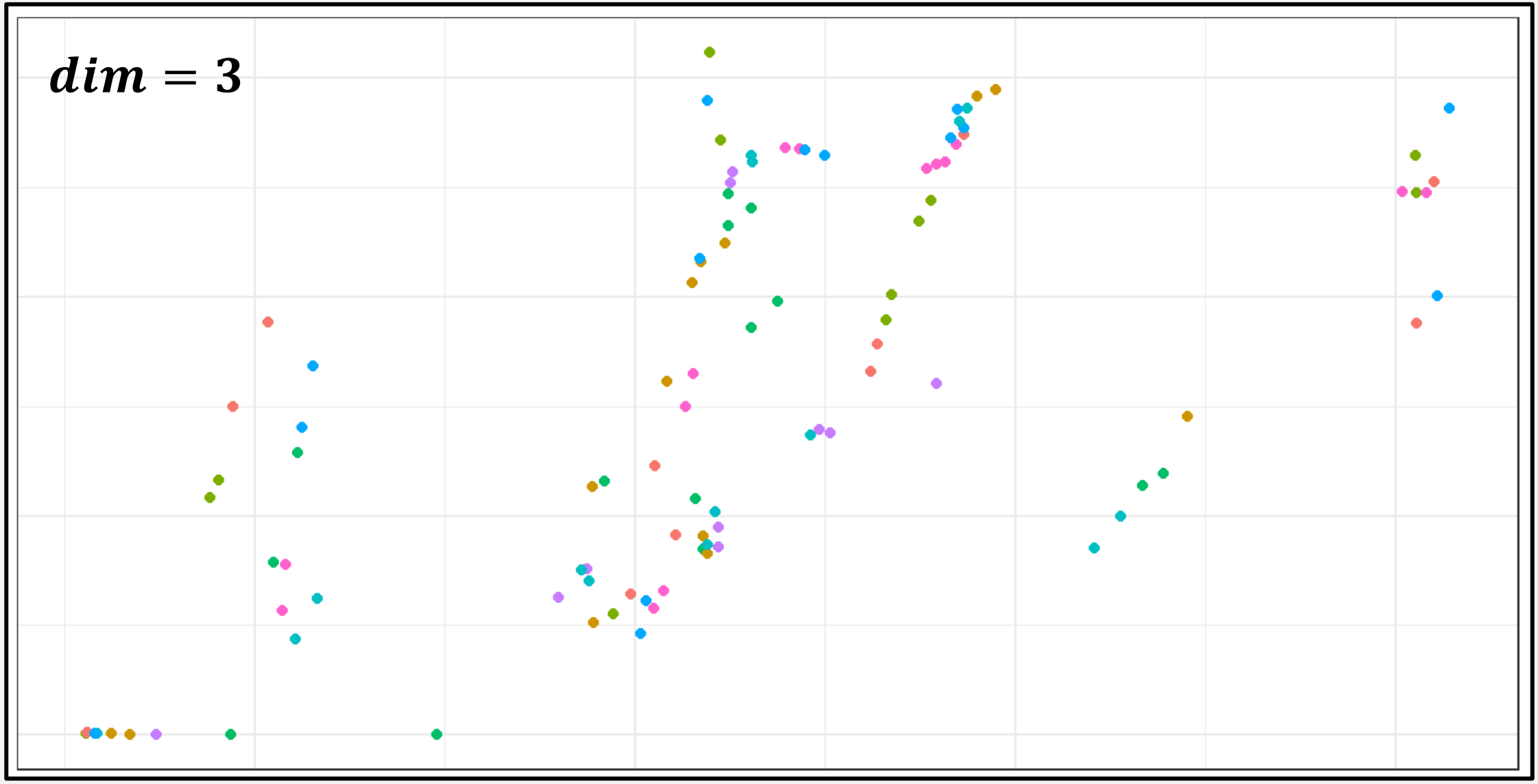
The problem of quantity



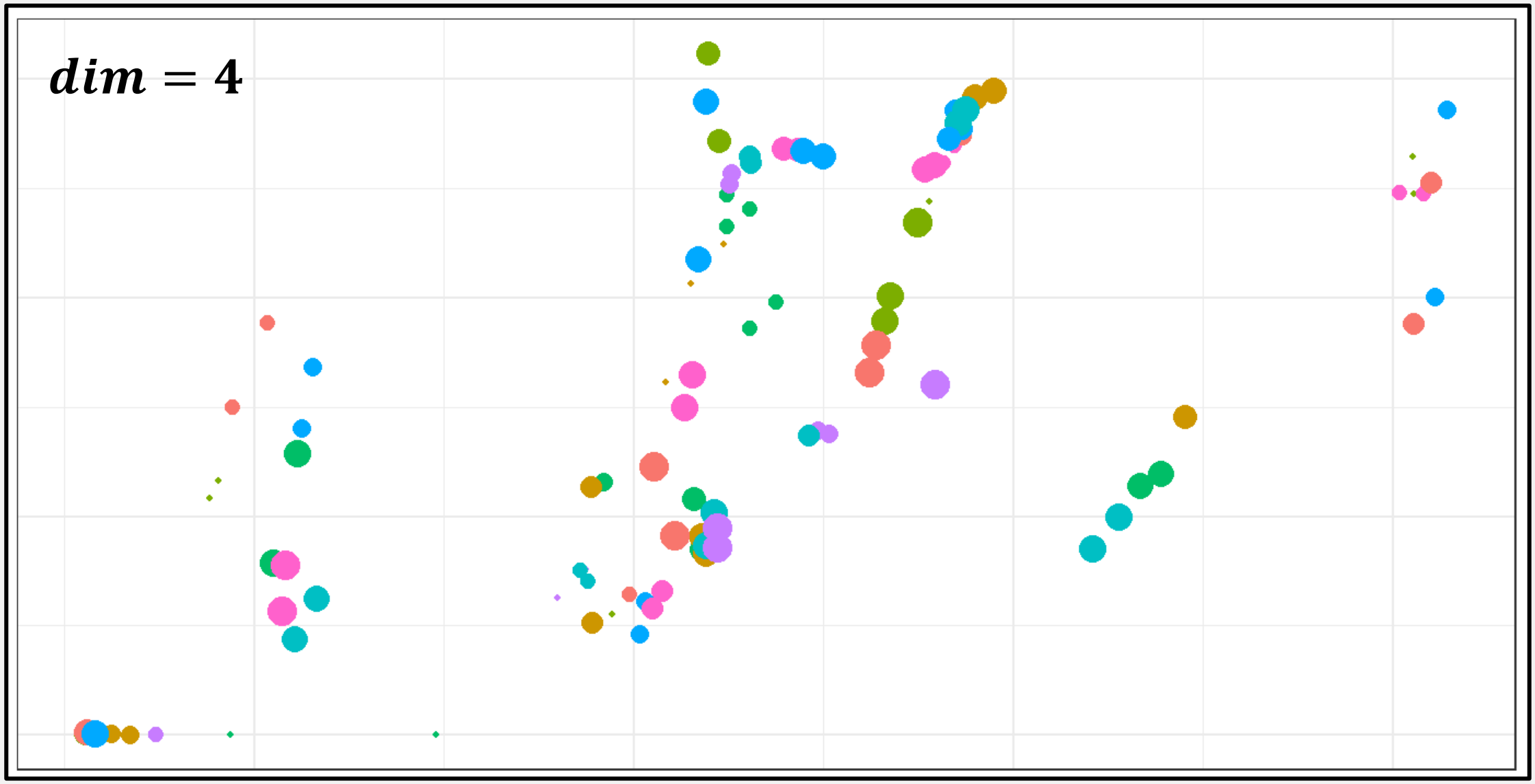
The problem of dimensionality



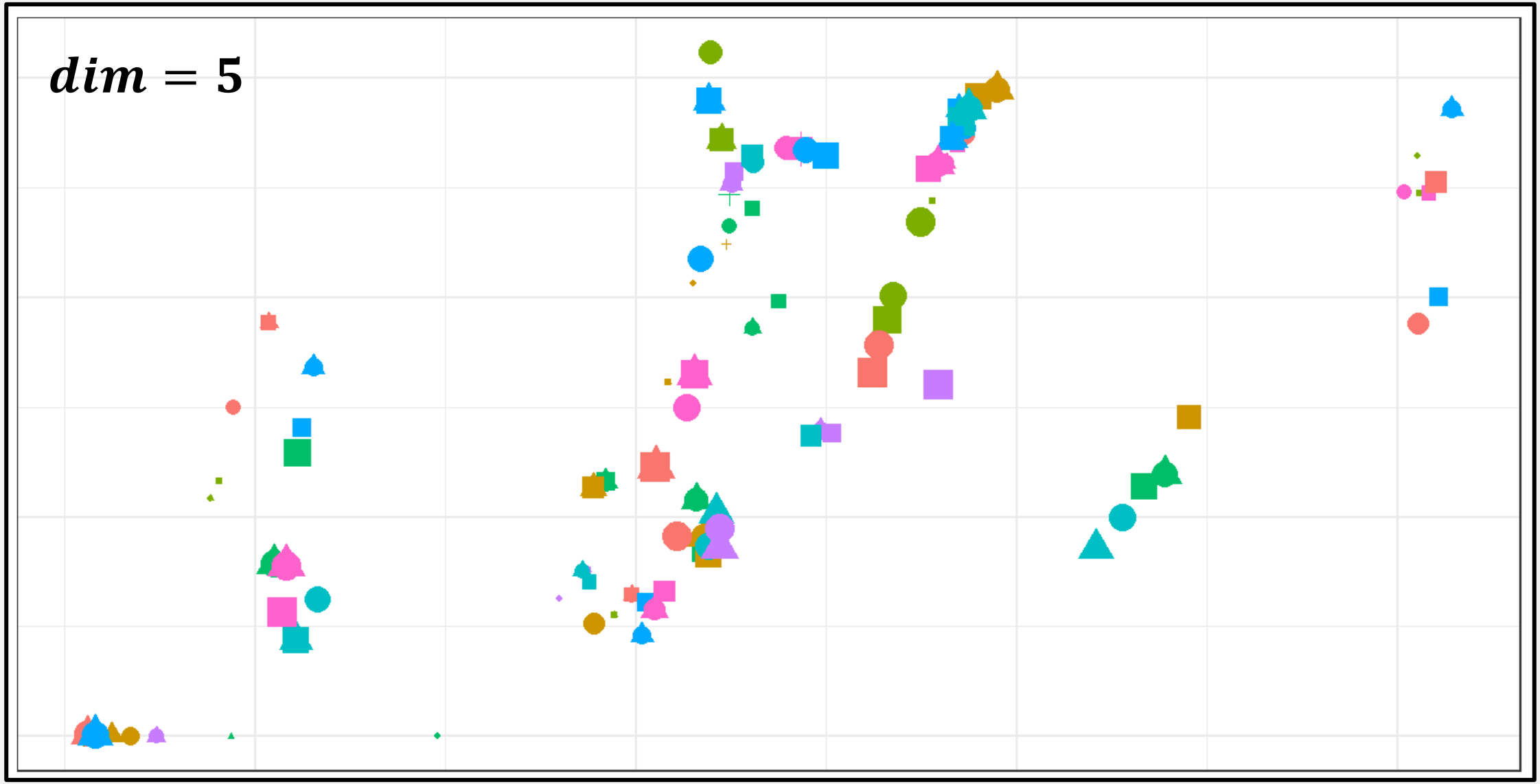
The problem of dimensionality



The problem of dimensionality



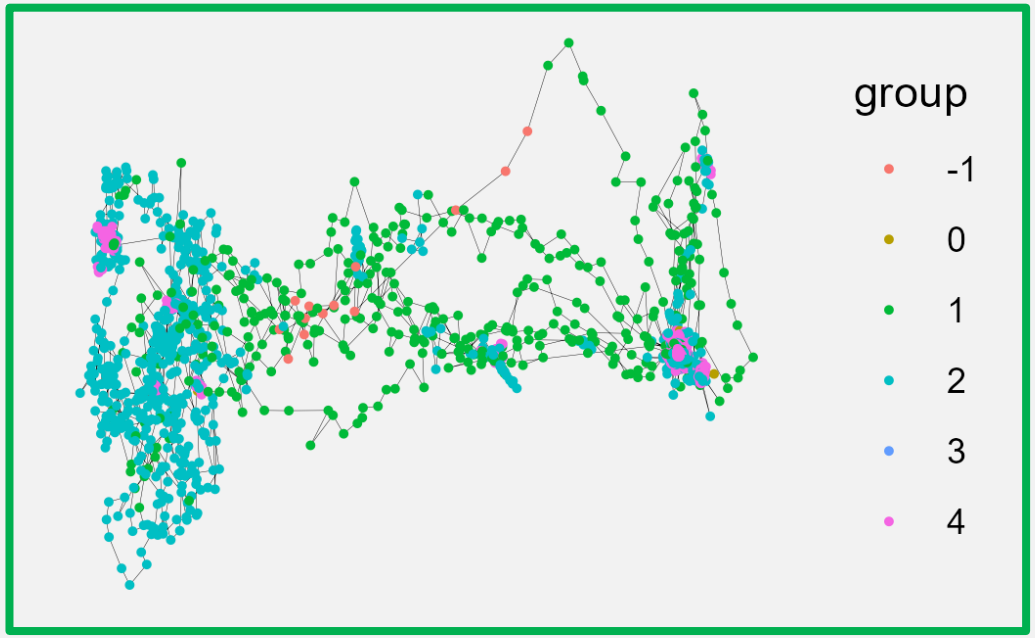
The problem of dimensionality



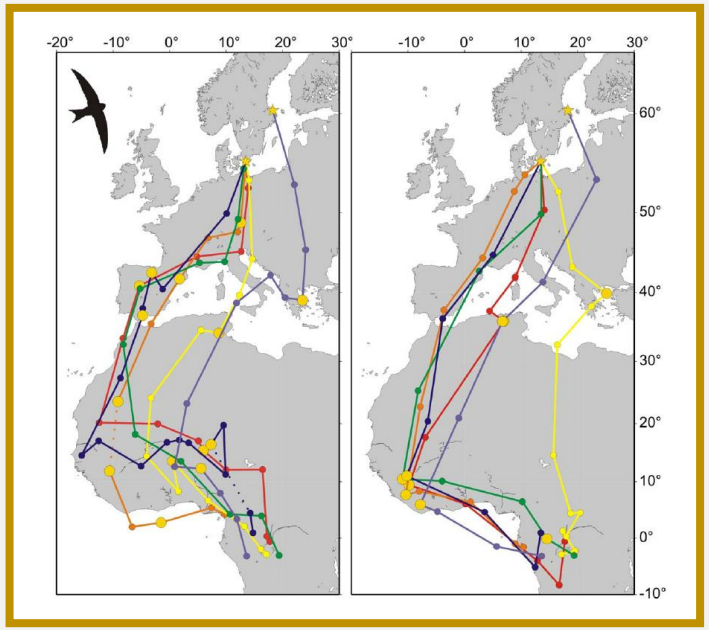
Previous studies

Maximum likelihood

$$Y = \exp(\beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots + \beta_n \cdot x_n)$$



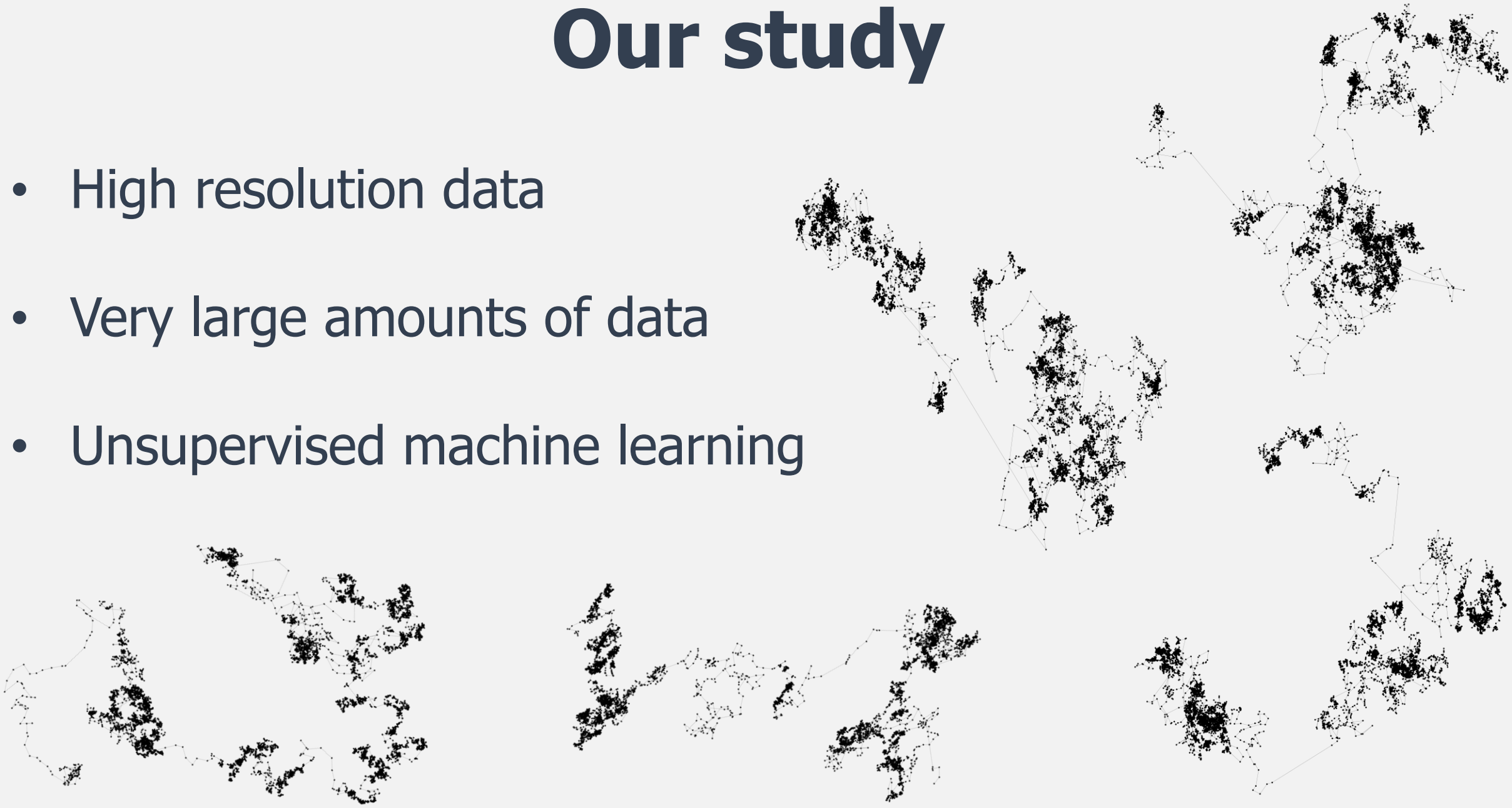
Supervised machine learning



Lower resolution

Our study

- High resolution data
- Very large amounts of data
- Unsupervised machine learning

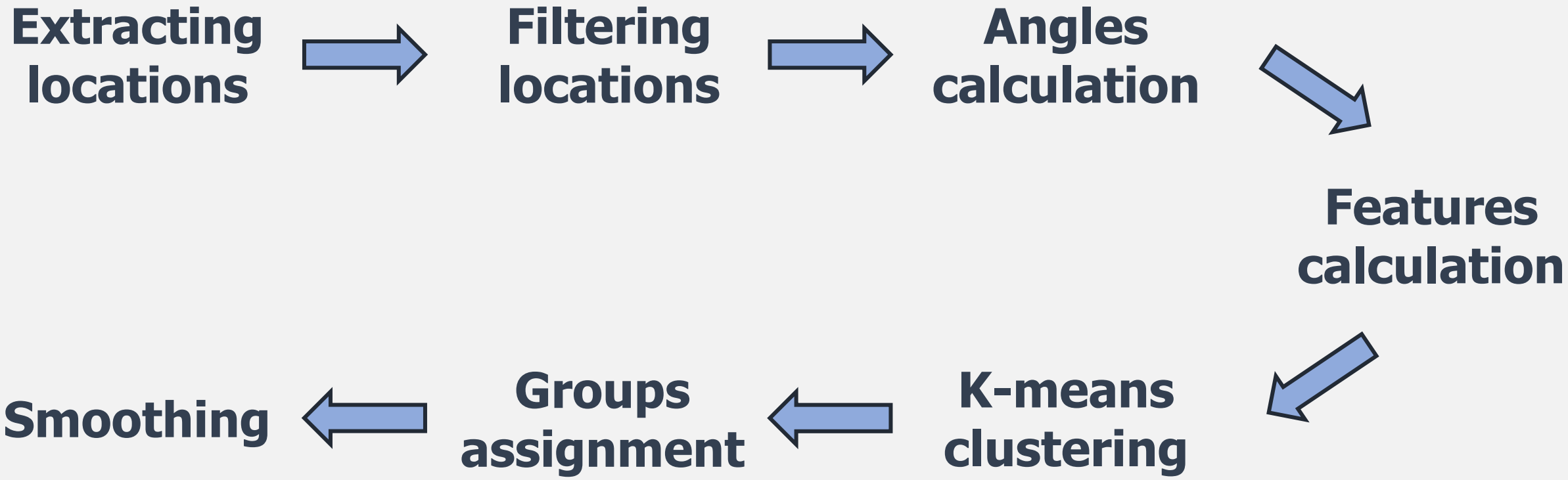


Objectives

- Segmentation of trajectories
- Creating artificial trajectories*

* More work should be done

Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

Directory structure

Previously developed functions for working with ATLAS

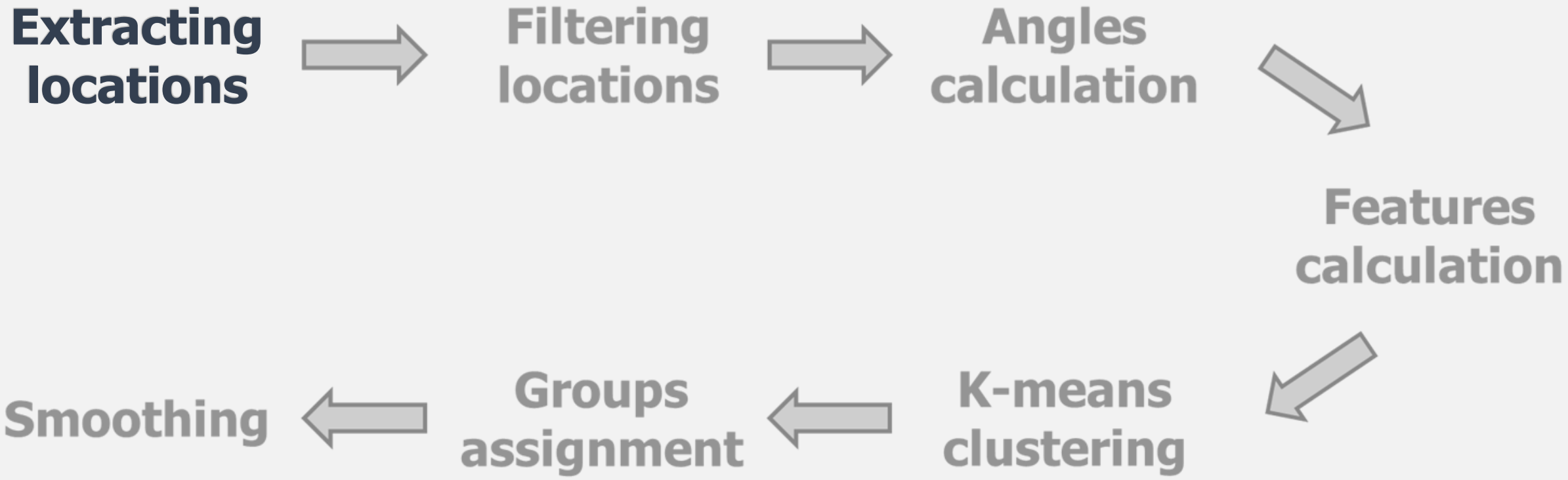
Input file for the data extraction

All codes written for this project

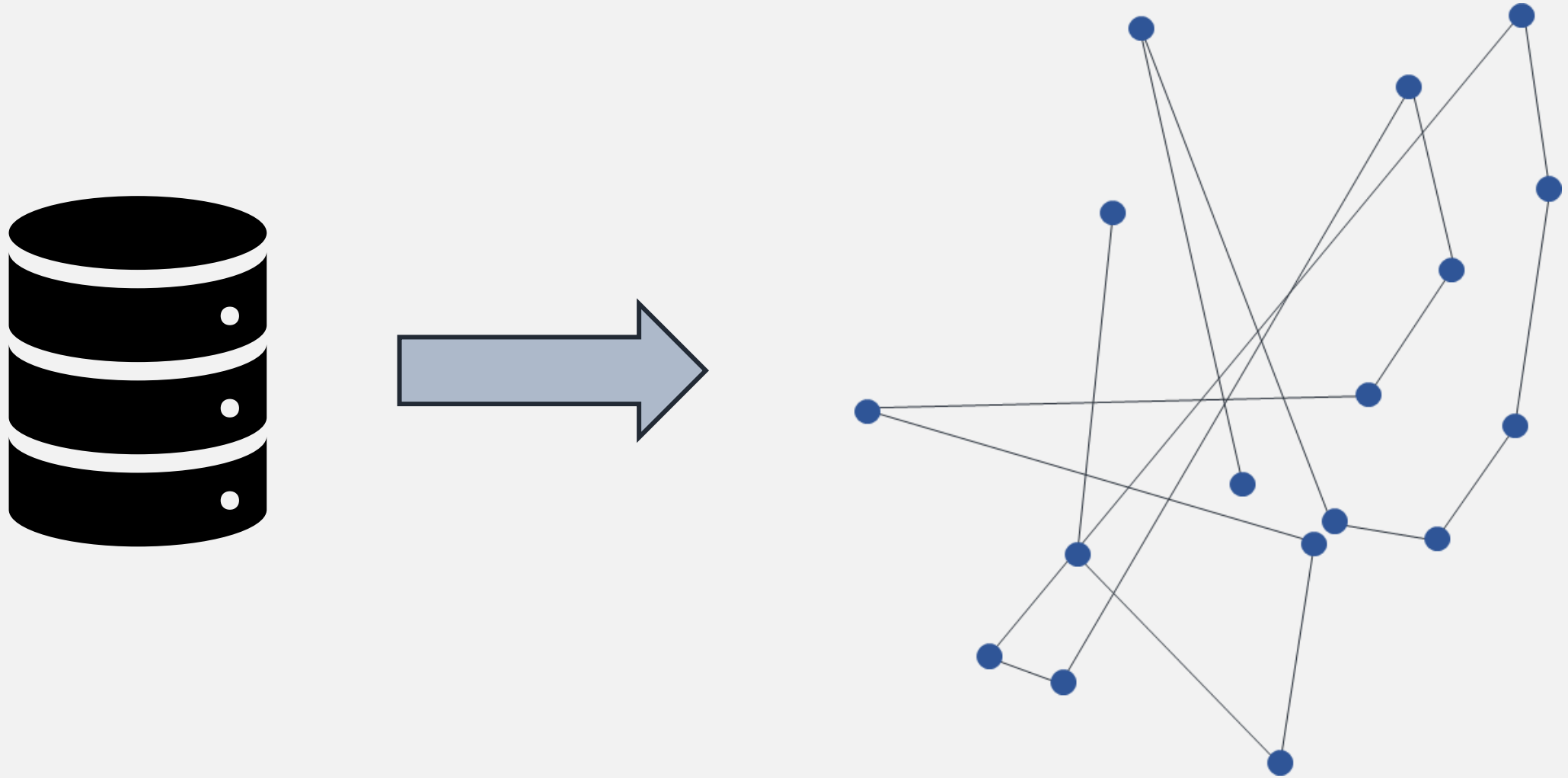
All plots produced in the pipeline

All output files produced in the pipeline

Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

Intuition – **Extracting locations**



Code – **Extracting locations**

my_codes

File name: 1 data_extraction.R

Needed input: movement_segmentation/inputs/TAG_dates.csv

Input format:

Tag ID	Date capture	Start hour	status_date
207	02/07/2020	23:59:59	01/11/2020

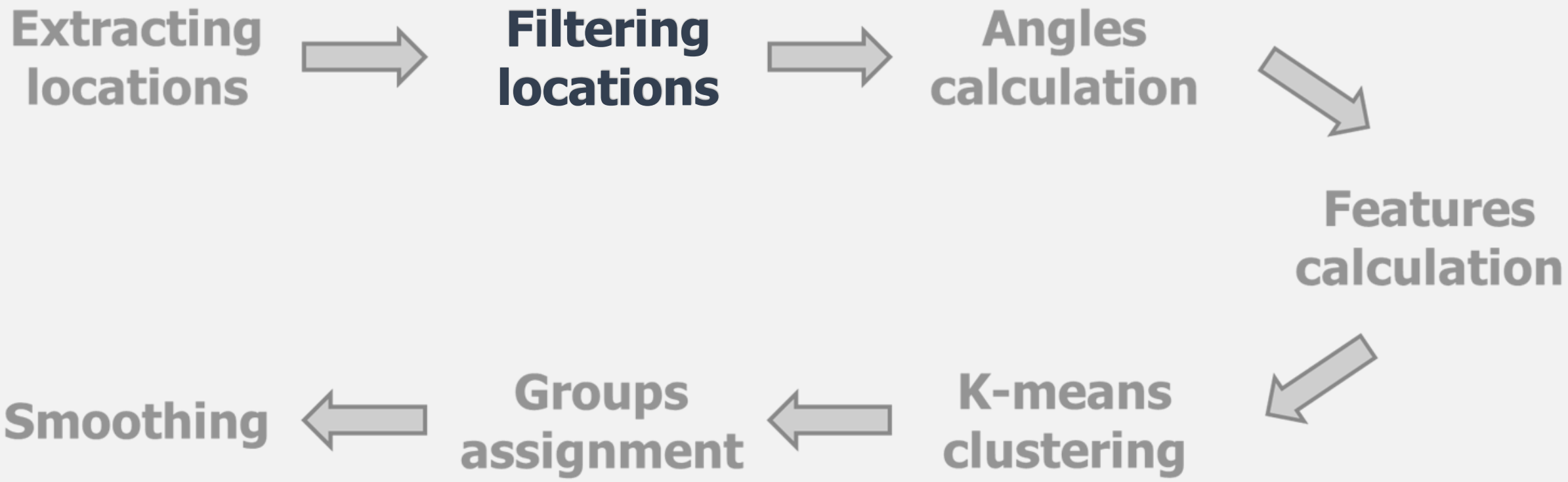
Date and hour from which locations are extracted

The last day for locations extraction

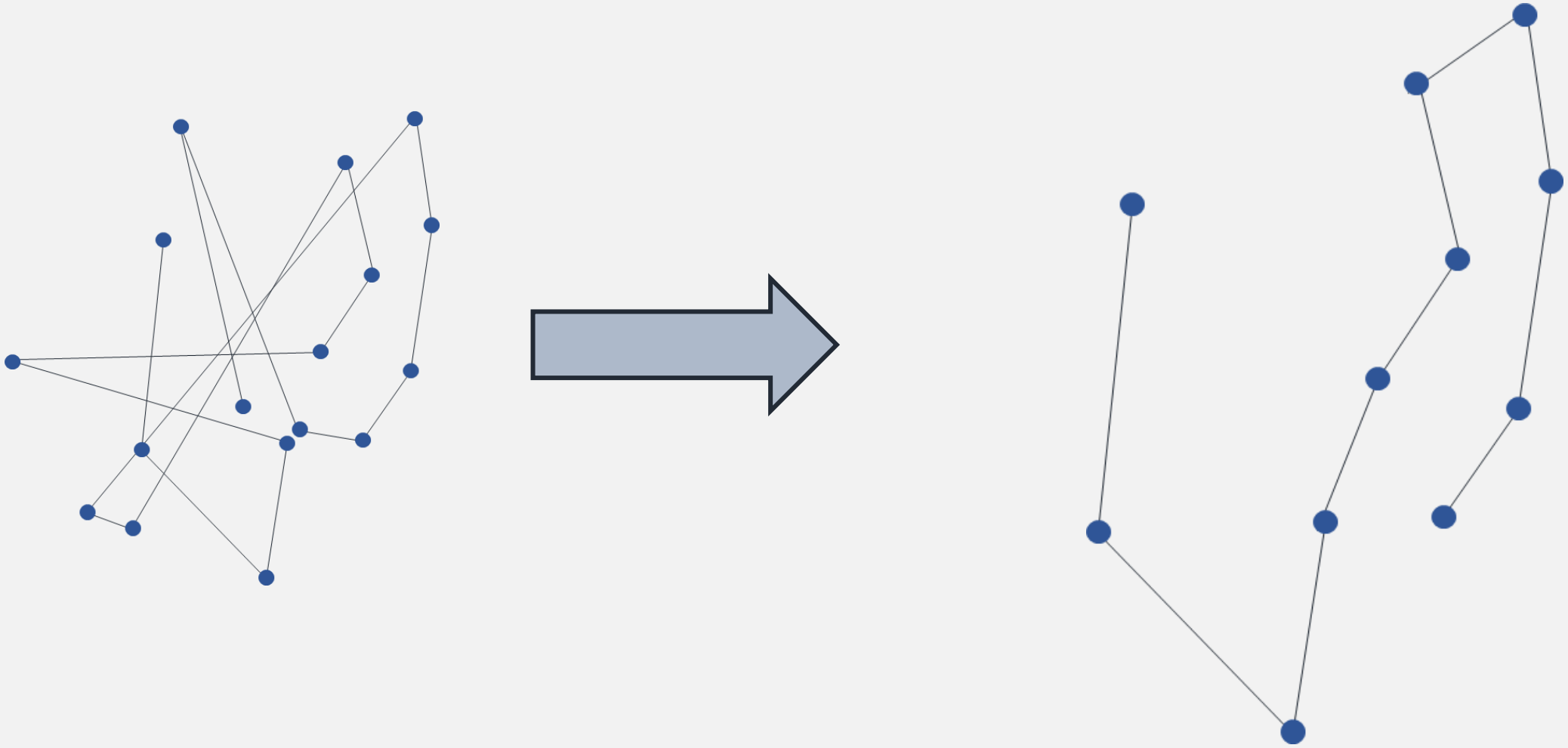
Notes:

- All tags requested should be with the same fix rate
- User must be connected to the university network
- The output – csv file for each tag

Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

Intuition – **Filtering locations**



Code – **Filtering locations**

my_codes

File name: 2 data_filtration.R

Input from: movement_segmentation/outputs/1 locations_data/

2 filters are used:

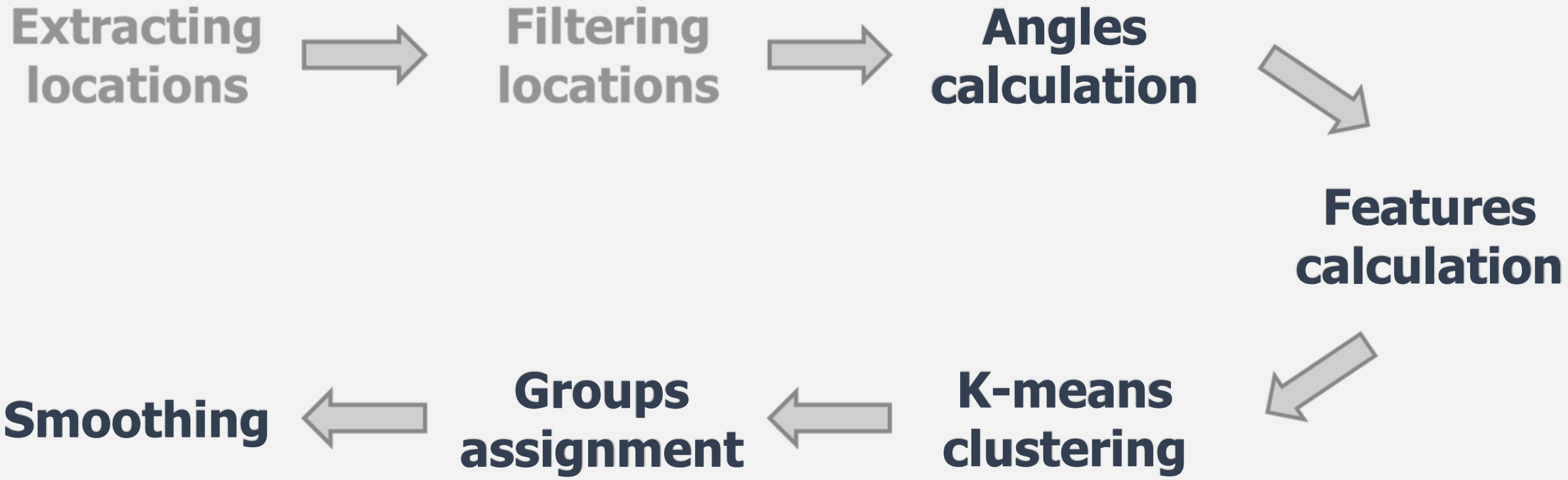
- atl_unifiedFilter (its attributes should be changed according to the species)
- Eitam's confidence filter

Notes:

- Other filtering methods can be used, as long as their output are in the format that appears below

TAG	X	Y	LON	LAT	dateTime	date	distance	dT	spd
442	239943.6	718322.3	35.42265	32.55901	25/05/2021 11:15	25/05/2021	3.182874	4	0.795718

Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

Code – **python pipeline**

my_codes

File name: 3 main.py

Uses 8 functions*:

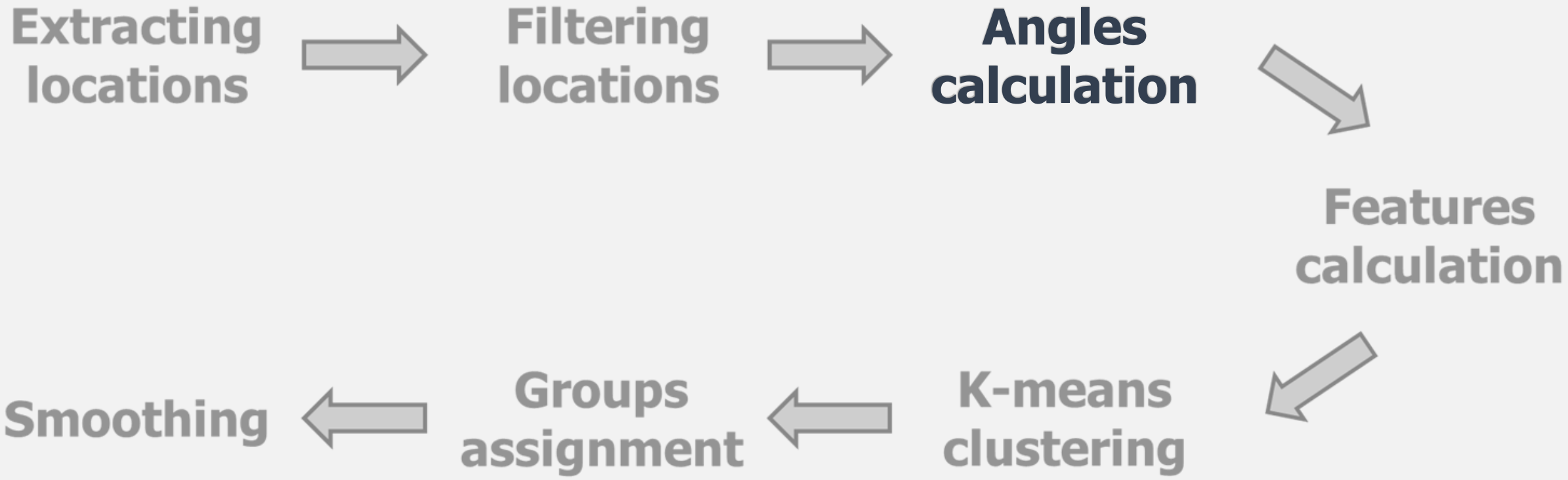
- calculate_angles
- calculate_features
- calculate_kmeans
- groups_to_locations
- smooth
- transitions
- get_examples
- distributions

Notes:

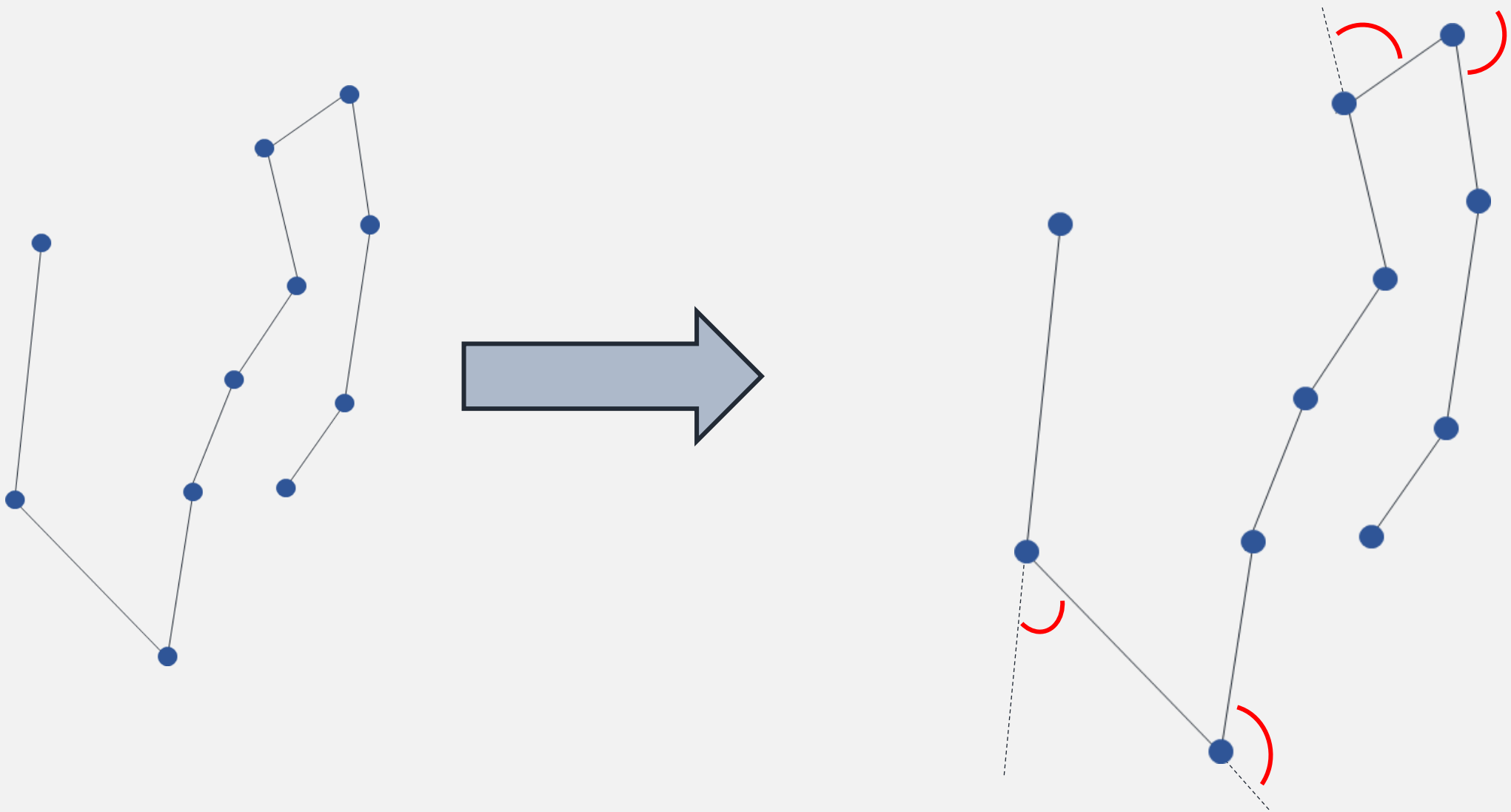
- Each function will be described separately
- Switch to comment functions you don't want to run

* The functions can be found at movement_segmentation/my_codes/omer_functions/__init__.py

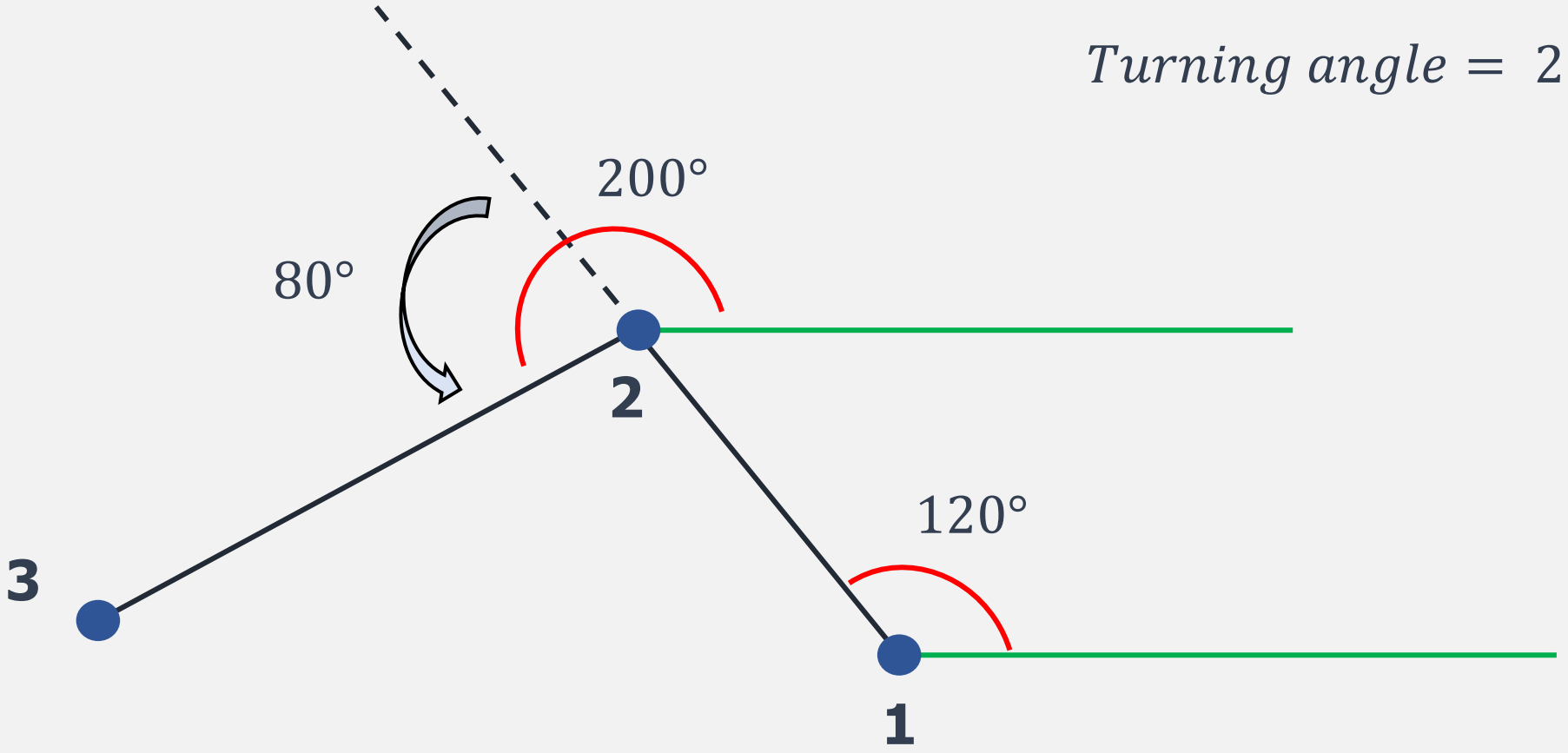
Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

Intuition – **Angles calculation**

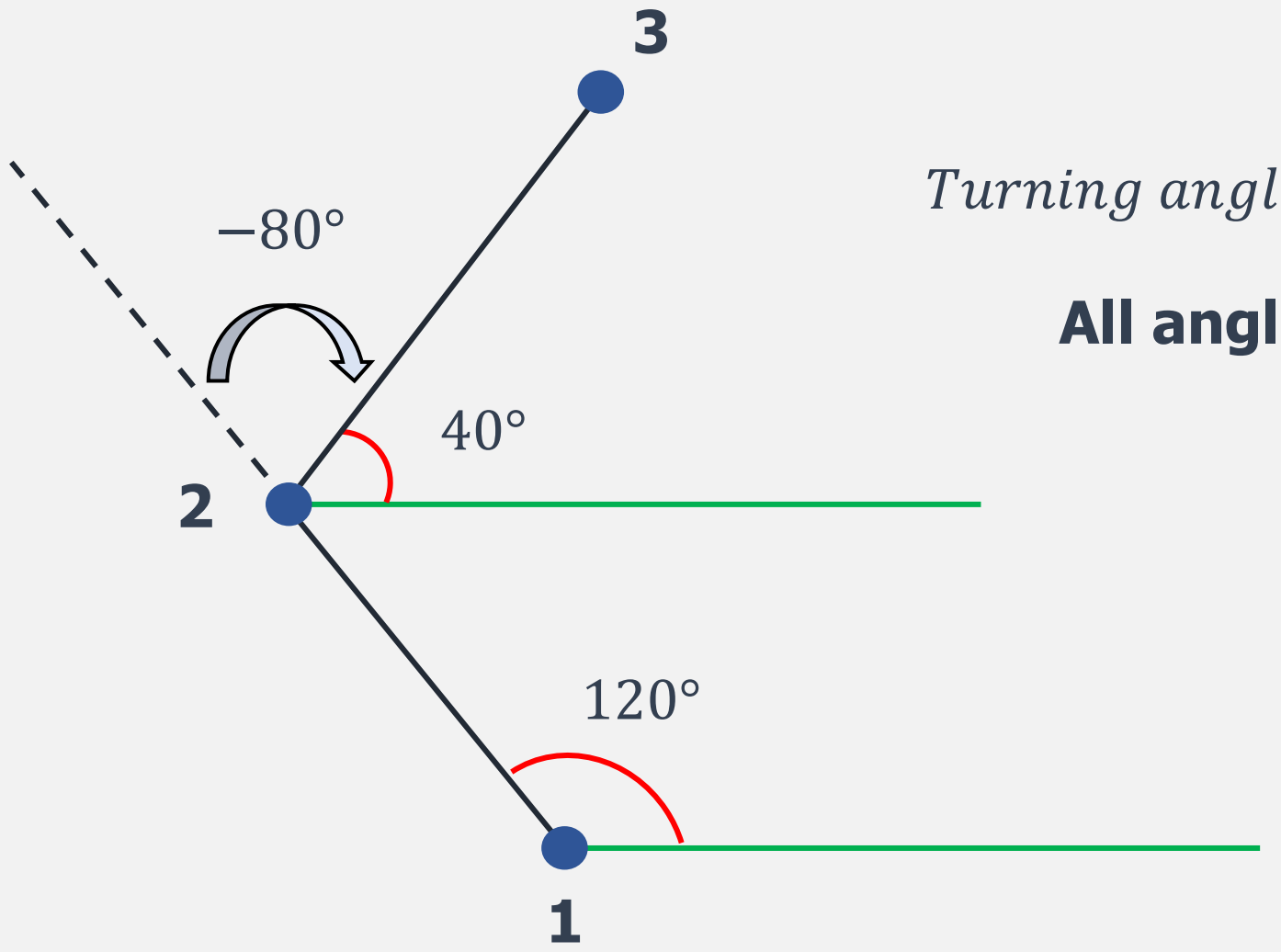


Intuition – **Angles calculation**



$$Turning\ angle = 200° - 120° = 80°$$

Intuition – **Angles calculation**



$$Turning\ angle = 40° - 120° = -80° = 280°$$

All angles between 0 and 360!

Code – **Angles calculation**

my_codes

File name: 3 main.py

Function name: calculate_angles

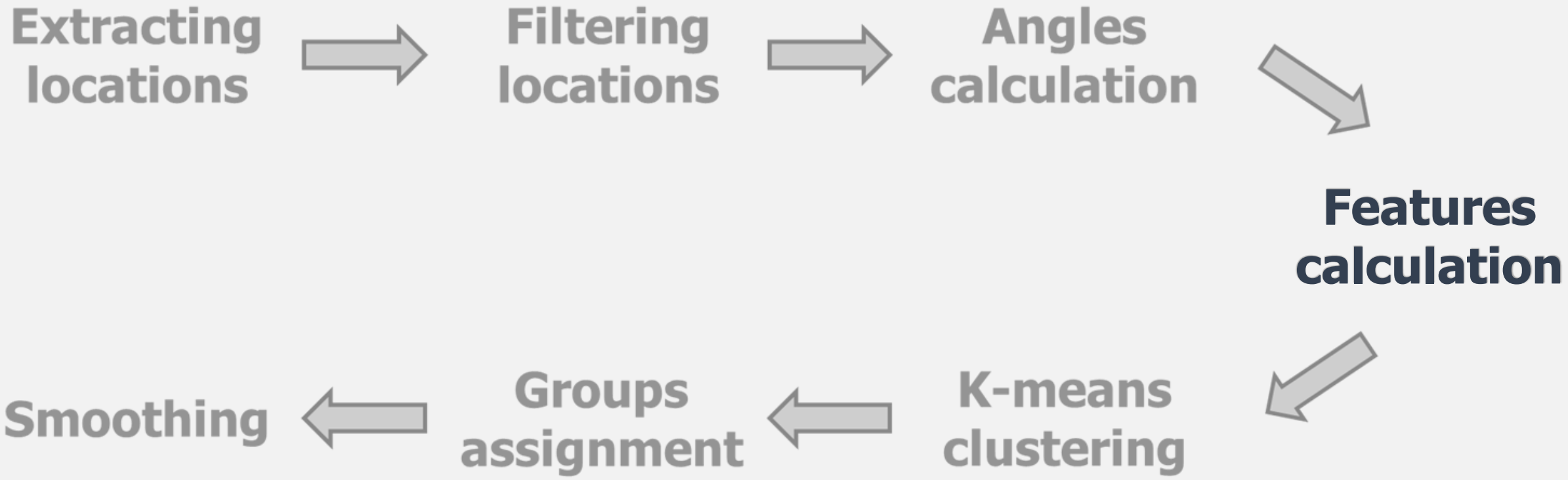
Input: movement_segmentation/outputs/2 filtered_data/

Output: movement_segmentation/outputs/3 data_with_angles/

Parameters:

- input_path string path to directory with input files (separated to tags)
- output_path string path to directory for output files

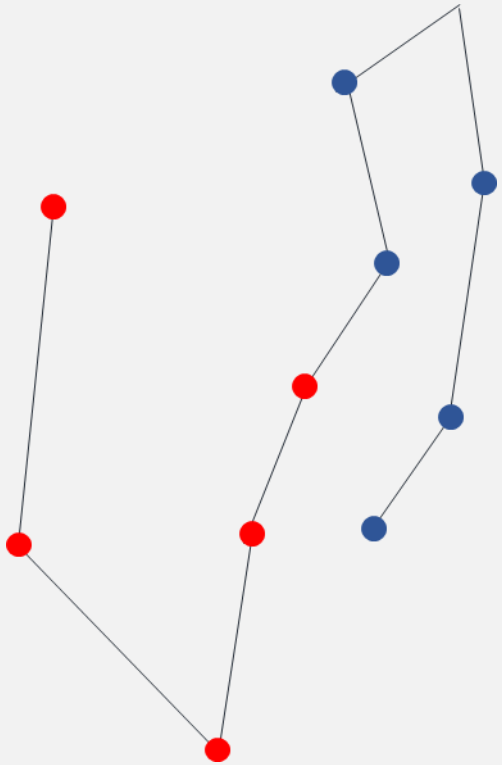
Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

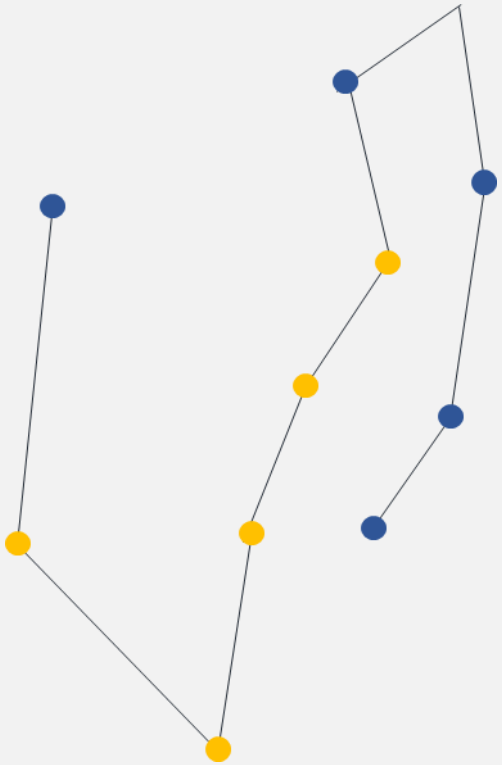
Intuition – **Features calculation**

# of series	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5

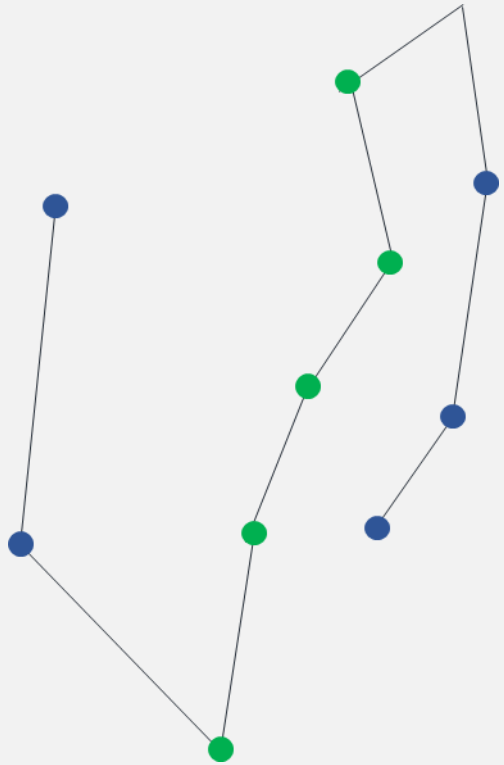
Intuition – **Features calculation**



Series 1

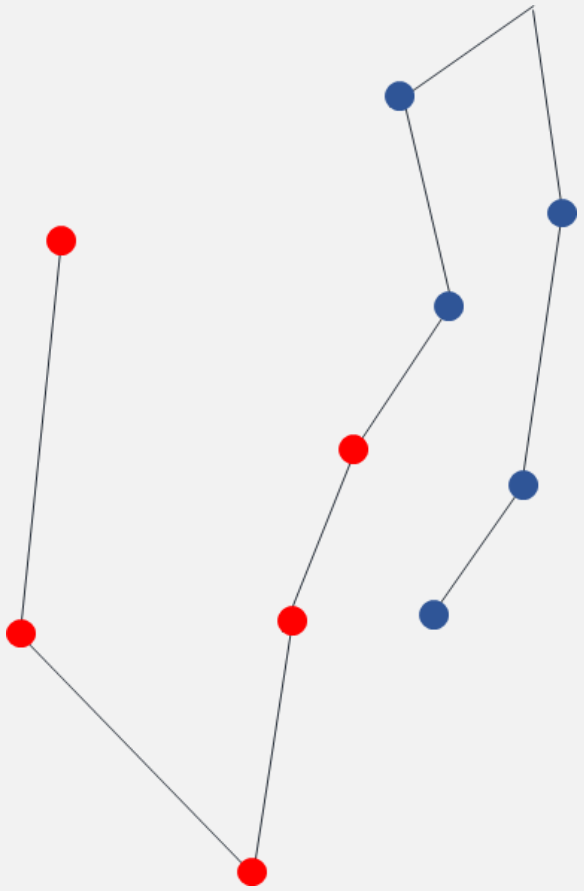


Series 2



Series 3

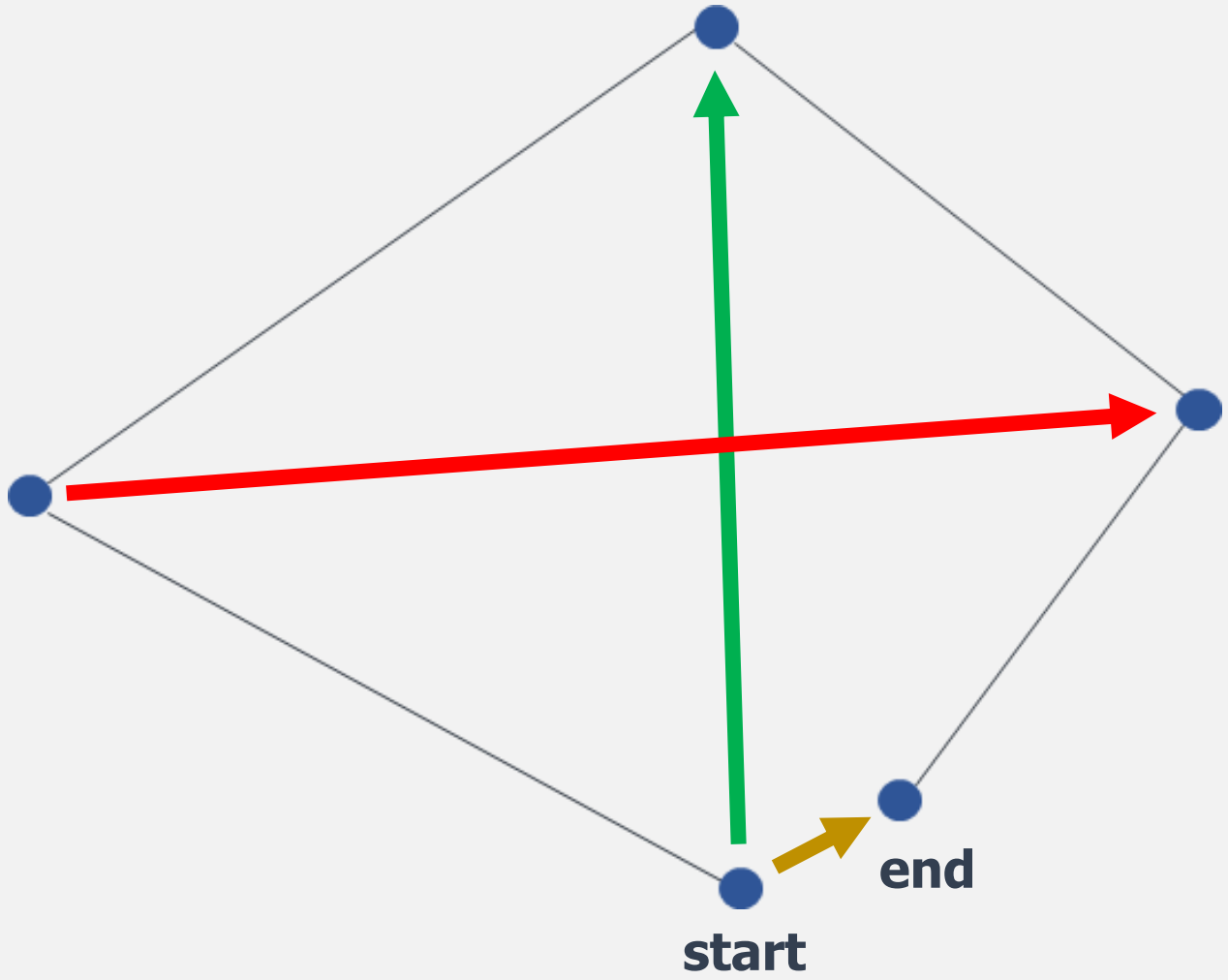
Intuition – **Features calculation**



Optional features:

- total distance
- mean speed
- standard deviation of speeds
- mean turning angle
- standard deviation of turning angles
- net displacement
- max displacement
- absolute max displacement
- tortuosity

Intuition – **Features calculation**



net displacement
Distance between first and last point

max displacement
Distance between first point and its furthest point

absolute max displacement
Distance between the two furthest points

$$Tortuosity = \frac{total\ distance}{net\ displacement}$$

Code – **Features calculation**

my_codes

File name: 3 main.py

Function name: calculate_features

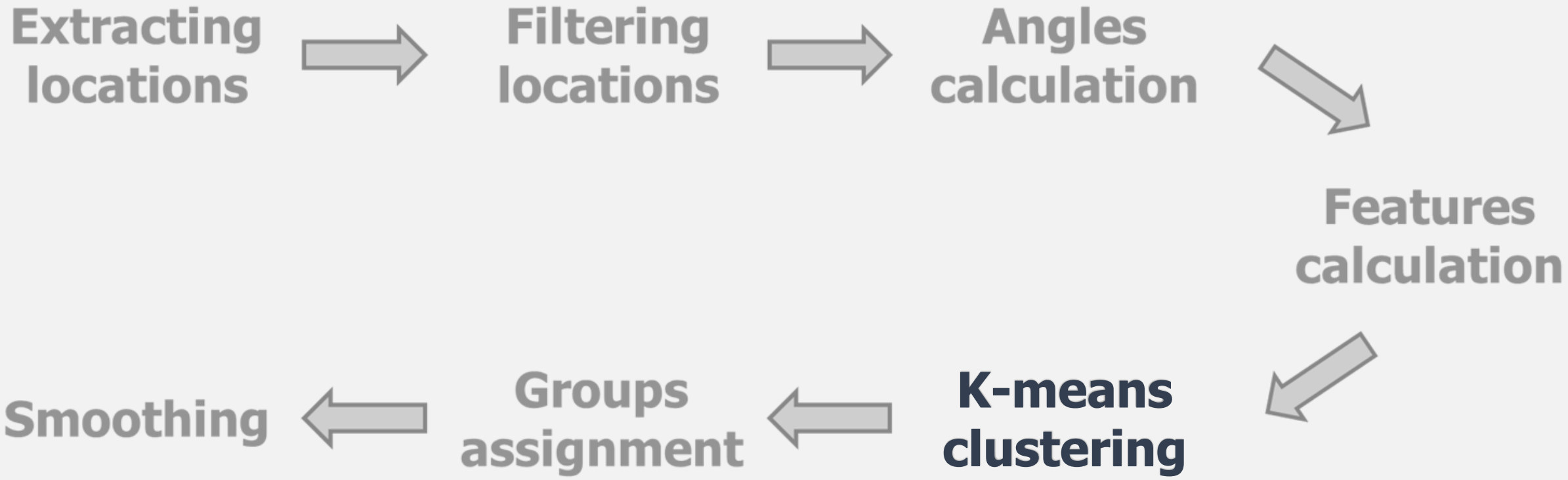
Input: movement_segmentation/outputs/3 data_with_angles/

Output: movement_segmentation/outputs/4 features/

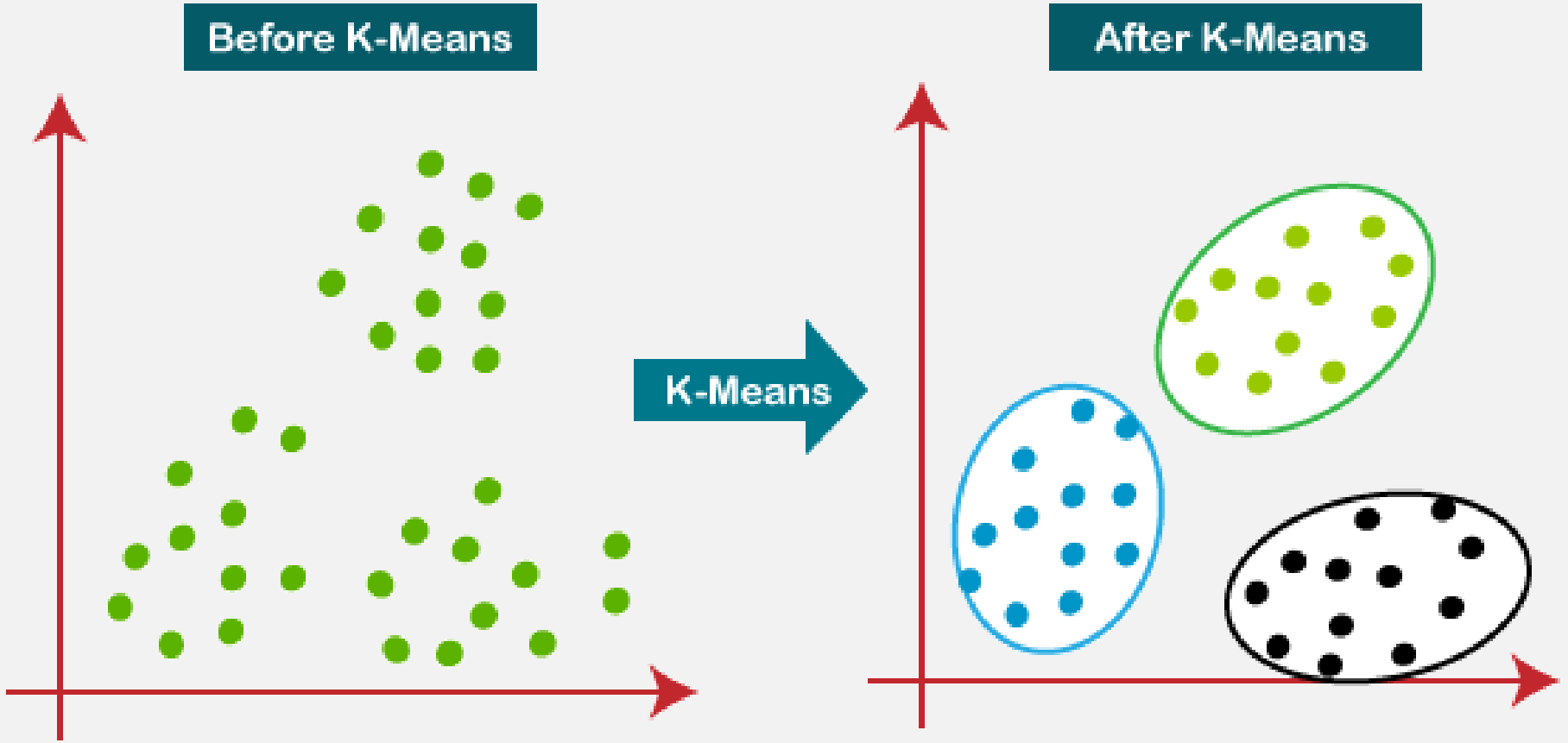
Parameters:

• input_path	string	path to directory with input files (separated to tags)
• output_path	string	path to directory for output file
• tw	int	time window, number of locations in each series
• normalized	boolian	does the user want the features to be normalized?
• features_list	list of strings	which features should be calculated? Full list in comment
• sep_files	boolian	export each tag to a separate file (in addition to the united)

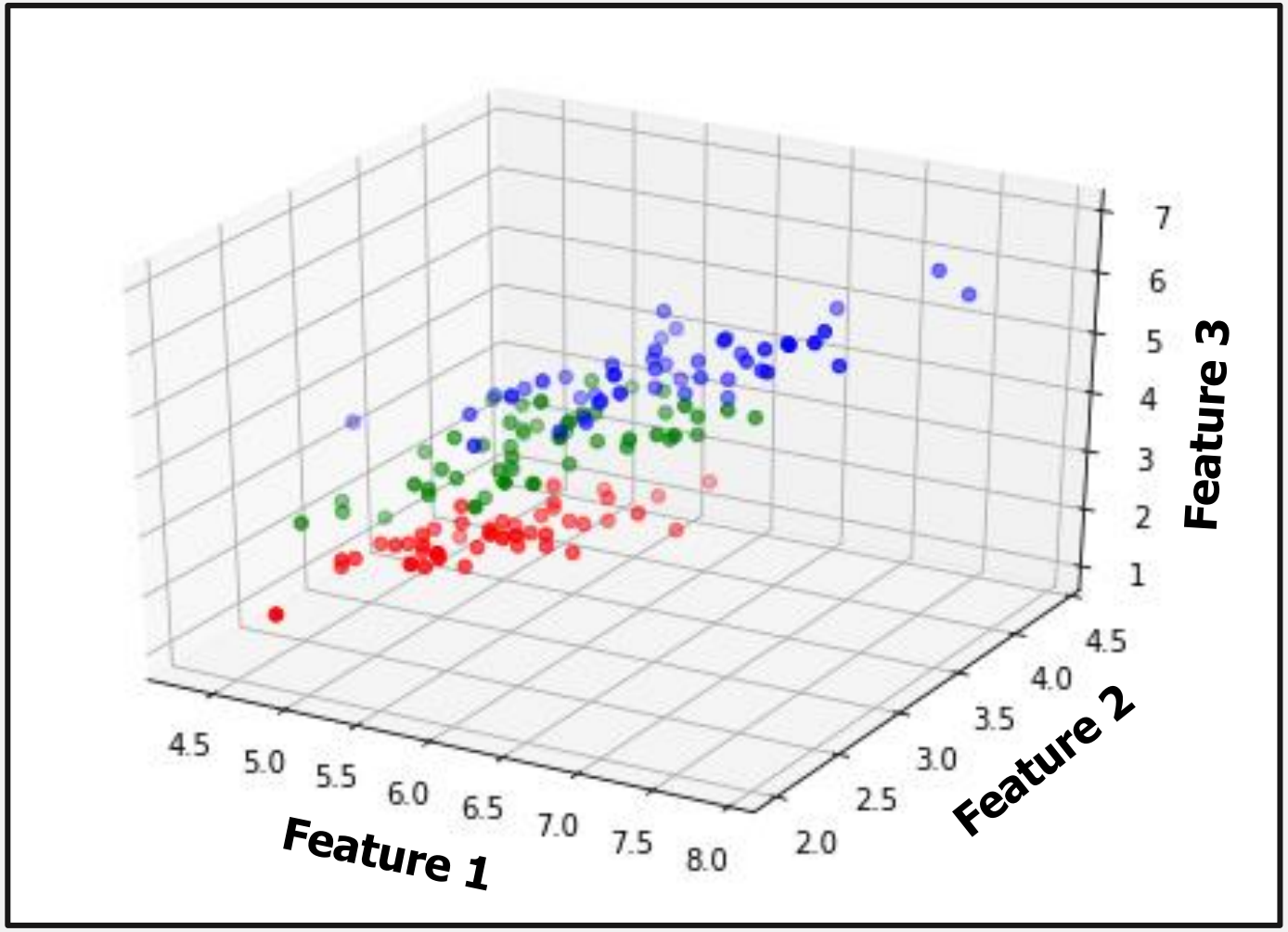
Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

Intuition – **K-means clustering**



Intuition – **K-means clustering**



# of series	group

Code – **K-means clustering**

my_codes

File name: 3 main.py

Function name: calculate_kmeans

Input: movement_segmentation/outputs/4 features/

Output: movement_segmentation/outputs/5 kmeans/

Parameters:

• input_path	string	path to directory with input file
• output_path	string	path to directory for output file
• tw	int	time window, number of locations in each series
• normalized	boolian	does the user want the features to be normalized?
• k_range	list of ints	how many groups the user want to cluster to? (one or more)

Code – **K-means clustering**

my_codes

File name: 3 main.py

Function name: calculate_kmeans

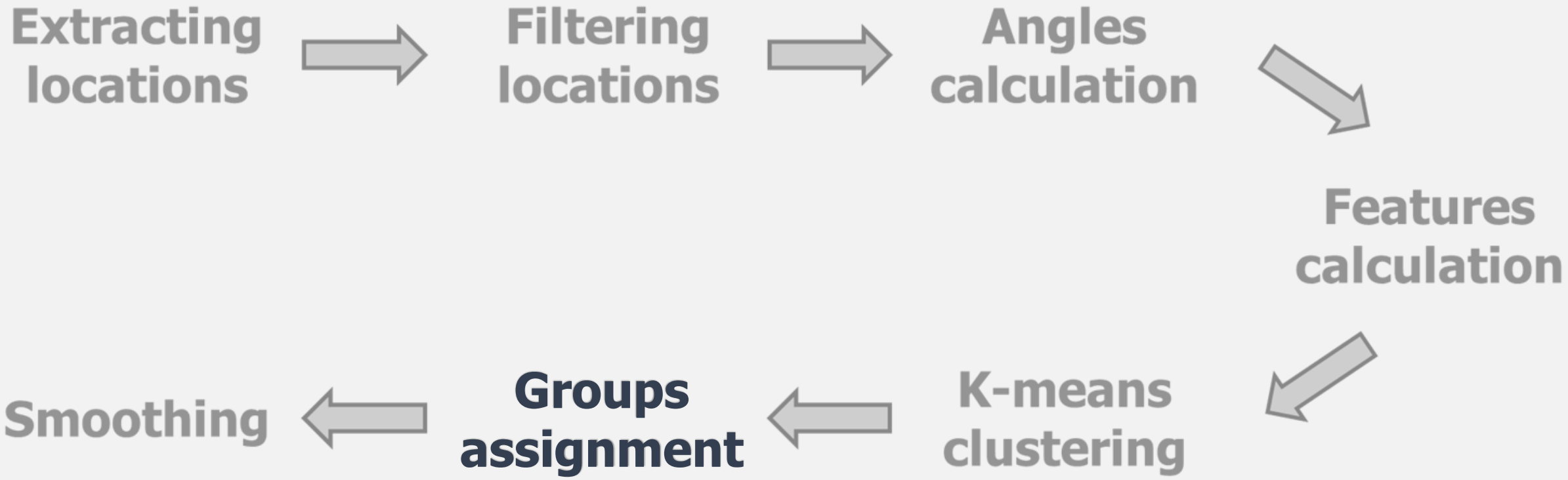
Input: movement_segmentation/outputs/4 features/

Output: movement_segmentation/outputs/5 kmeans/

Parameters:

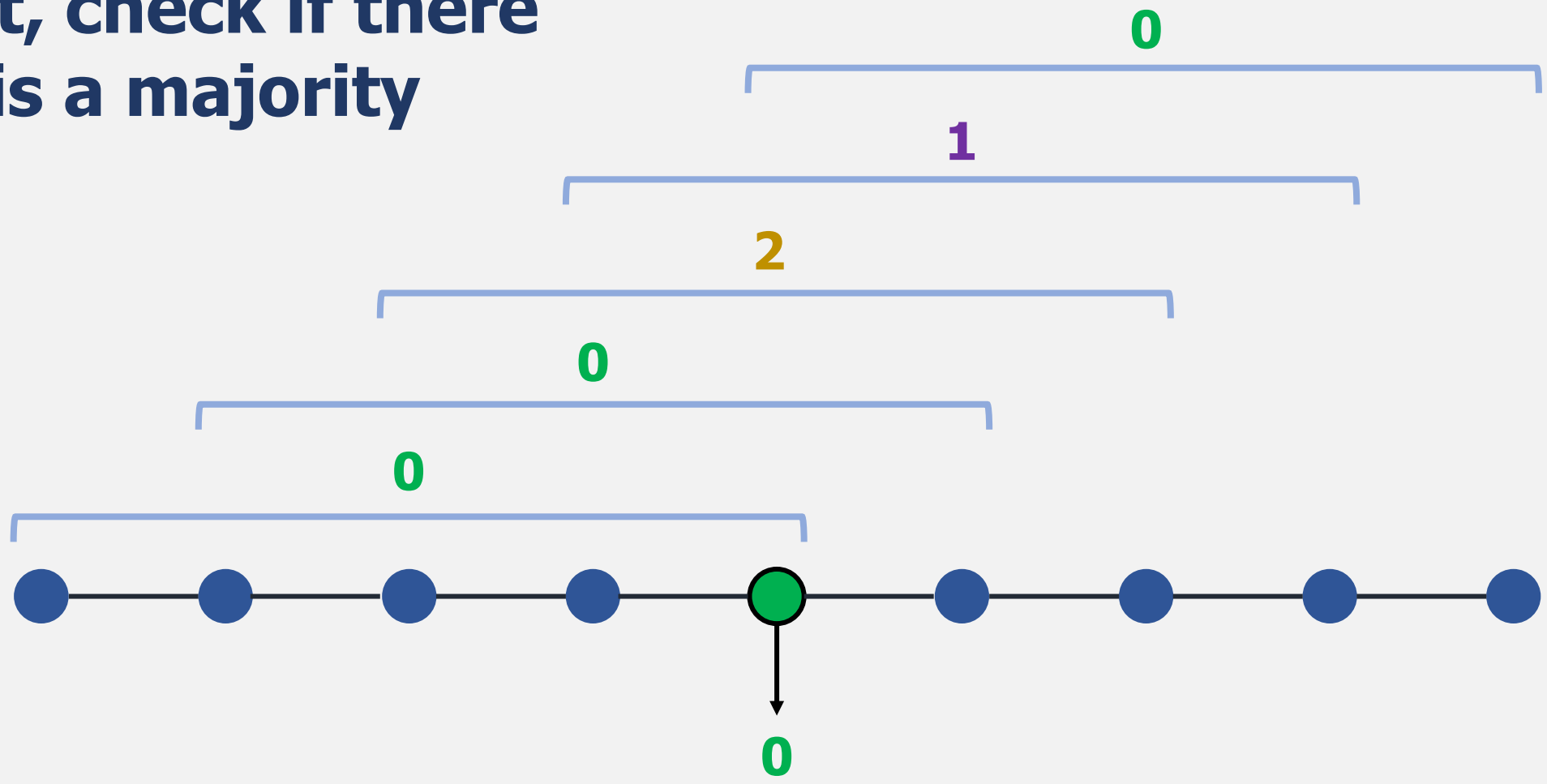
• rel_features_list	list of strings	which features should be considered?
• filt_lim	float	series with one or more features larger or smaller than $\pm sd \cdot filt\_lim$ will be omitted
• sep_files	boolian	export each tag to a separate file (in addition to the united)
• plot	boolian	create PCA plot of the clustering

Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

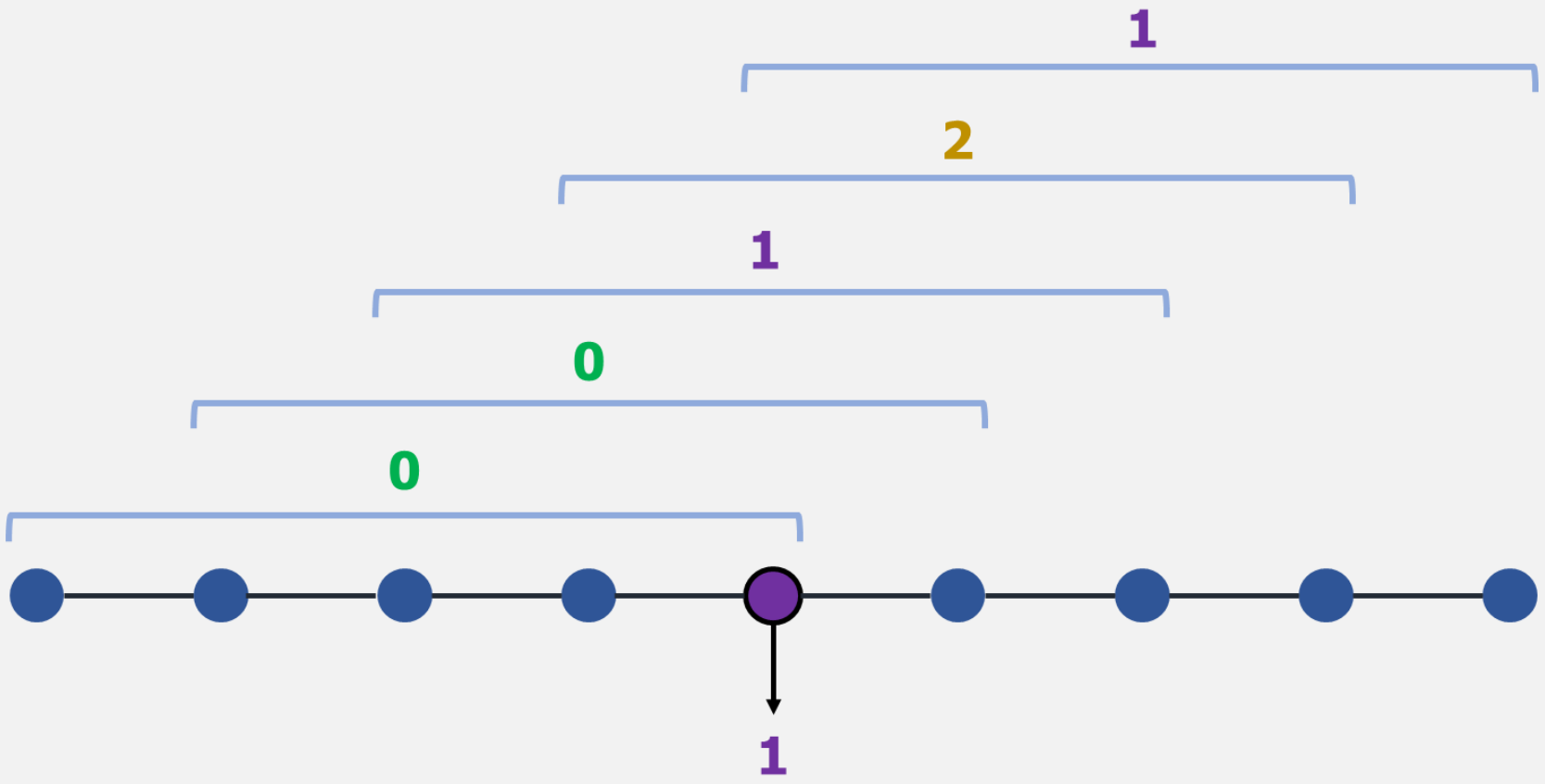
Intuition – **Groups assignment**

First, check if there is a majority



Intuition – **Groups assignment**

If not, calculate weights for each series



Intuition – **Groups assignment**

If not, calculate weights for each series

Coefficients defined by the user

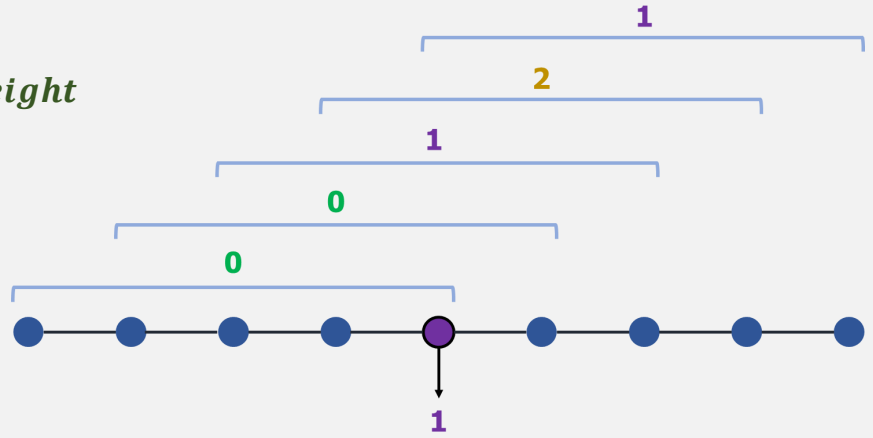
$$weight = \left(\frac{1}{mean\_time}\right)^{time\_weight} \cdot \left(\frac{1}{distance\ from\ middle}\right)^{dist\_weight}$$

Series with low mean time (high fix rate) are preferred

Series in which the current point is closer to the middle are preferred

Intuition – **Groups assignment**

$$weight = \left(\frac{1}{mean\_time}\right)^{time\_weight} \cdot \left(\frac{1}{distance\ from\ middle}\right)^{dist\_weight}$$



# of series	group	Mean time	Distance from the middle
1	0	8	3
2	0	16	2
3	1	8	1
4	2	8	2
5	1	24	3

$$weight = \left(\frac{1}{8}\right)^1 \cdot \left(\frac{1}{3}\right)^1 = 0.042$$

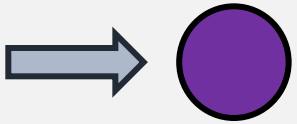
$$weight = \left(\frac{1}{16}\right)^1 \cdot \left(\frac{1}{2}\right)^1 = 0.031$$

$$weight = \left(\frac{1}{8}\right)^1 \cdot \left(\frac{1}{1}\right)^1 = 0.125$$

$$weight = \left(\frac{1}{8}\right)^1 \cdot \left(\frac{1}{2}\right)^1 = 0.062$$

$$weight = \left(\frac{1}{24}\right)^1 \cdot \left(\frac{1}{3}\right)^1 = 0.014$$

group	weight
0	$0.042 + 0.031 = 0.073$
1	$0.125 + 0.014 = 0.139$
2	0.062



Code – **Groups assignment**

my_codes

File name: 3 main.py

Function name: groups_to_locations

Input: movement_segmentation/outputs/5 kmeans/

Output: movement_segmentation/outputs/6 groups_to_locations/

Parameters:

• input_path	string	path to directory with input file
• locations_path	string	path to directory with the filtered locations (with angles)
• output_path	string	path to directory for output file
• tw	int	time window, number of locations in each series
• k_range	list of ints	which clustering the user want to use? (one or more)

Code – **Groups assignment**

my_codes

File name: 3 main.py

Function name: groups_to_locations

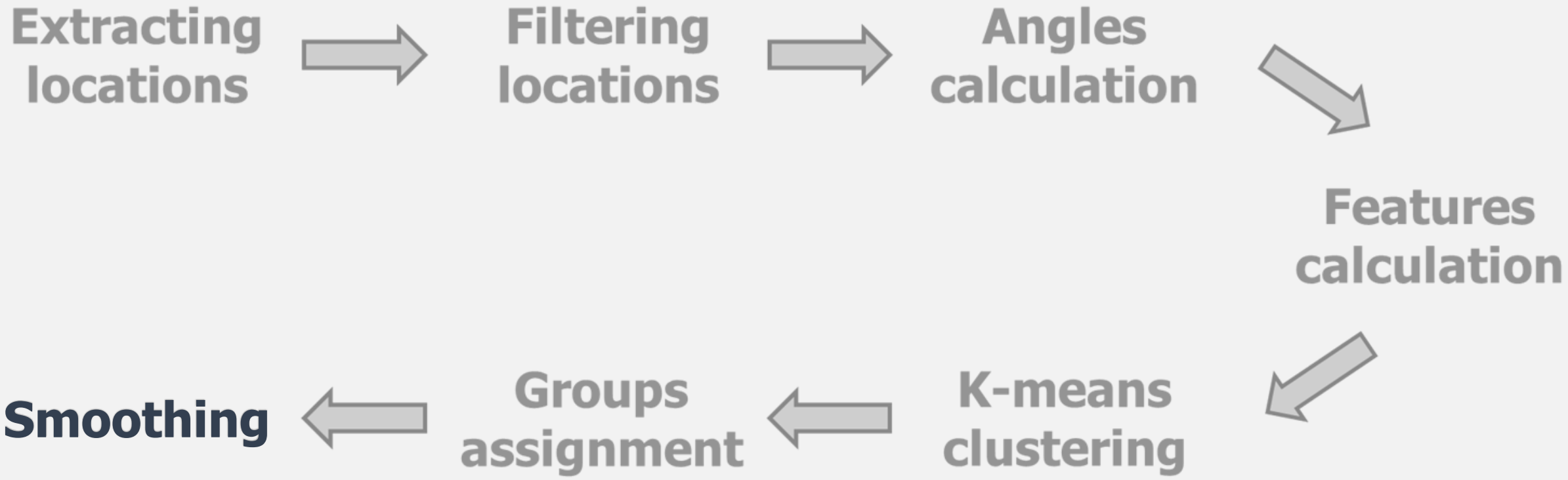
Input: movement_segmentation/outputs/5 kmeans/

Output: movement_segmentation/outputs/6 groups_to_locations/

Parameters:

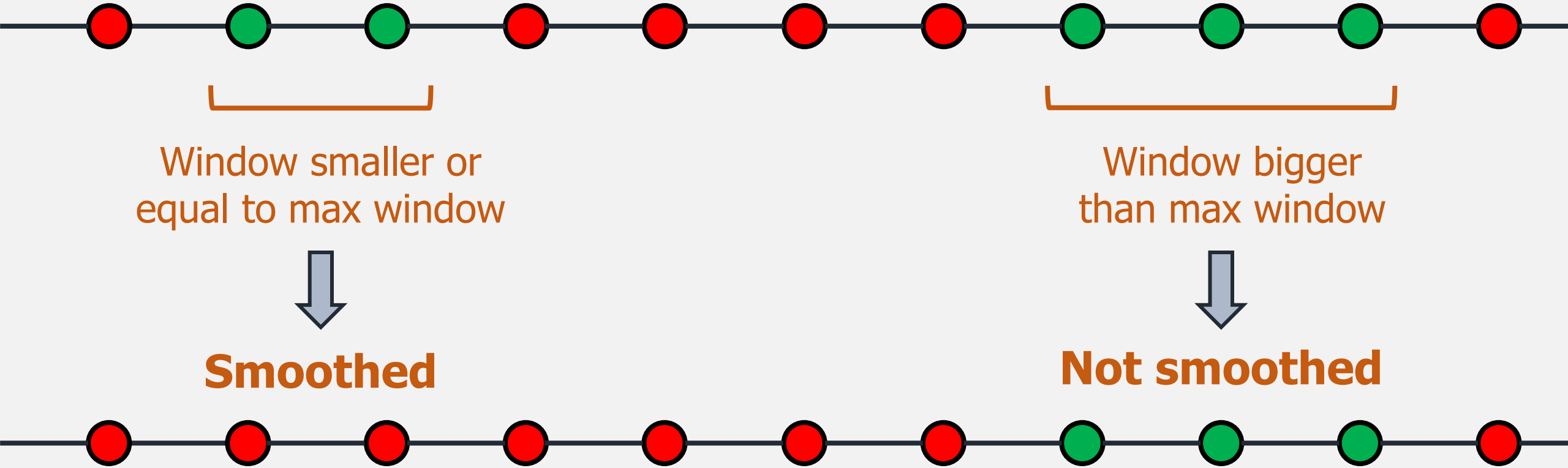
• dist_weight	float	how much weight the user want to give to the distance from the middle? (as explained in the intuition part)
• time_weight	float	how much weight the user want to give to the mean time (dT) of the series? (as explained in the intuition part)
• sep_files	boolian	export each tag to a separate file (in addition to the united)

Pipeline


Extracting locations
Filtering locations
Angles calculation
Features calculation
K-means clustering
Groups assignment
Smoothing

Intuition – **Smoothing**

$$max\ window = 2$$



Window smaller or
equal to max window

Window bigger
than max window

Smoothed

Not smoothed

Code – **Smoothing**

my_codes

File name: 3 main.py

Function name: smooth

Input: movement_segmentation/outputs/ 6 groups_to_locations/

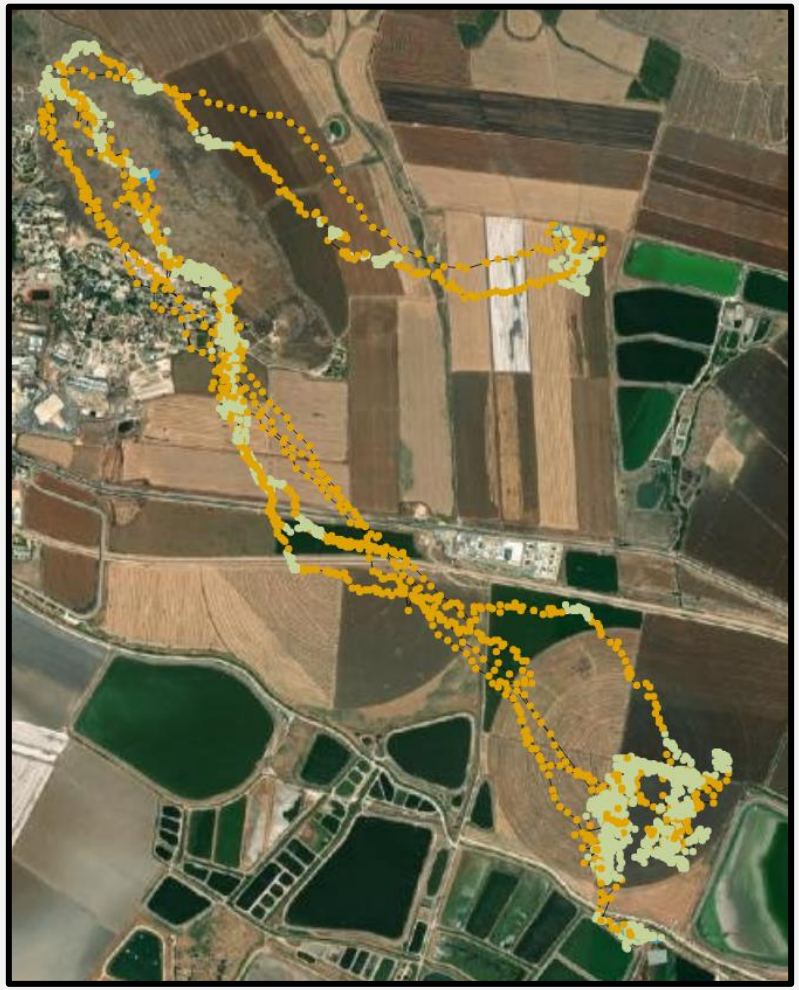
Output: movement_segmentation/outputs/7 smoothed_groups/

Parameters:

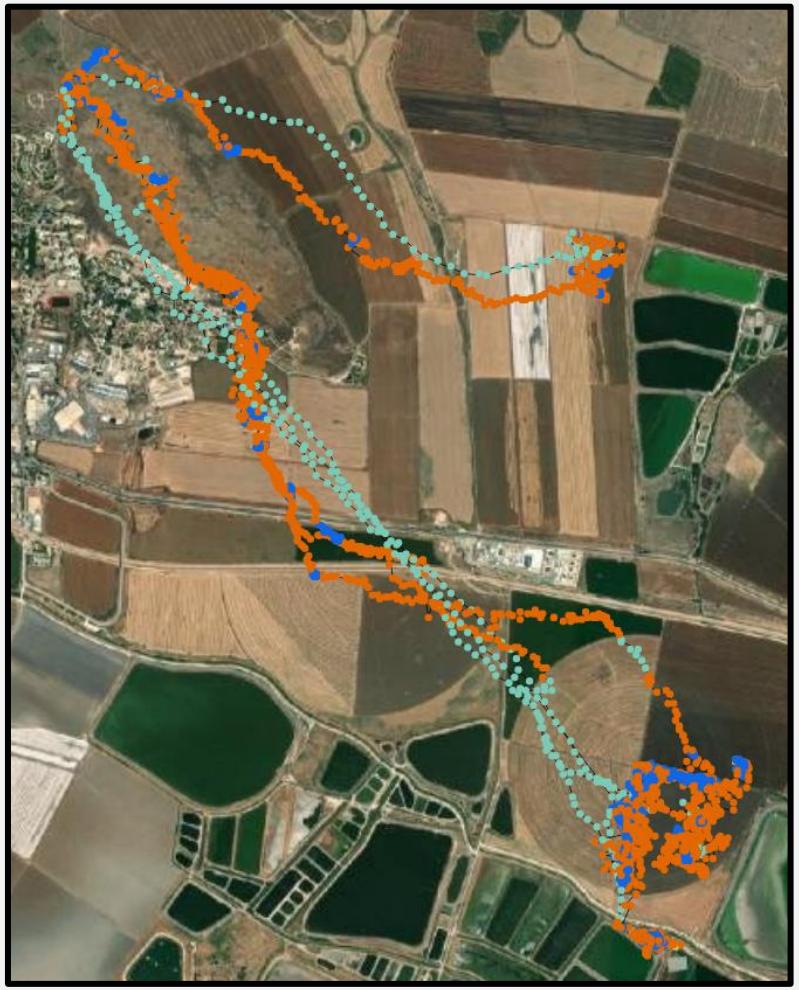
• input_path	string	path to directory with input file
• output_path	string	path to directory for output file
• tw	int	time window, number of locations in each series
• k_range	list of ints	which clustering the user want to use? (one or more)
• max_window	int	gaps equal or shorter than this value will be smoothed
• sep_files	boolian	export each tag to a separate file (in addition to the united)

Examples

k = 2



k = 3



k = 4

Extra codes – analysis and plots

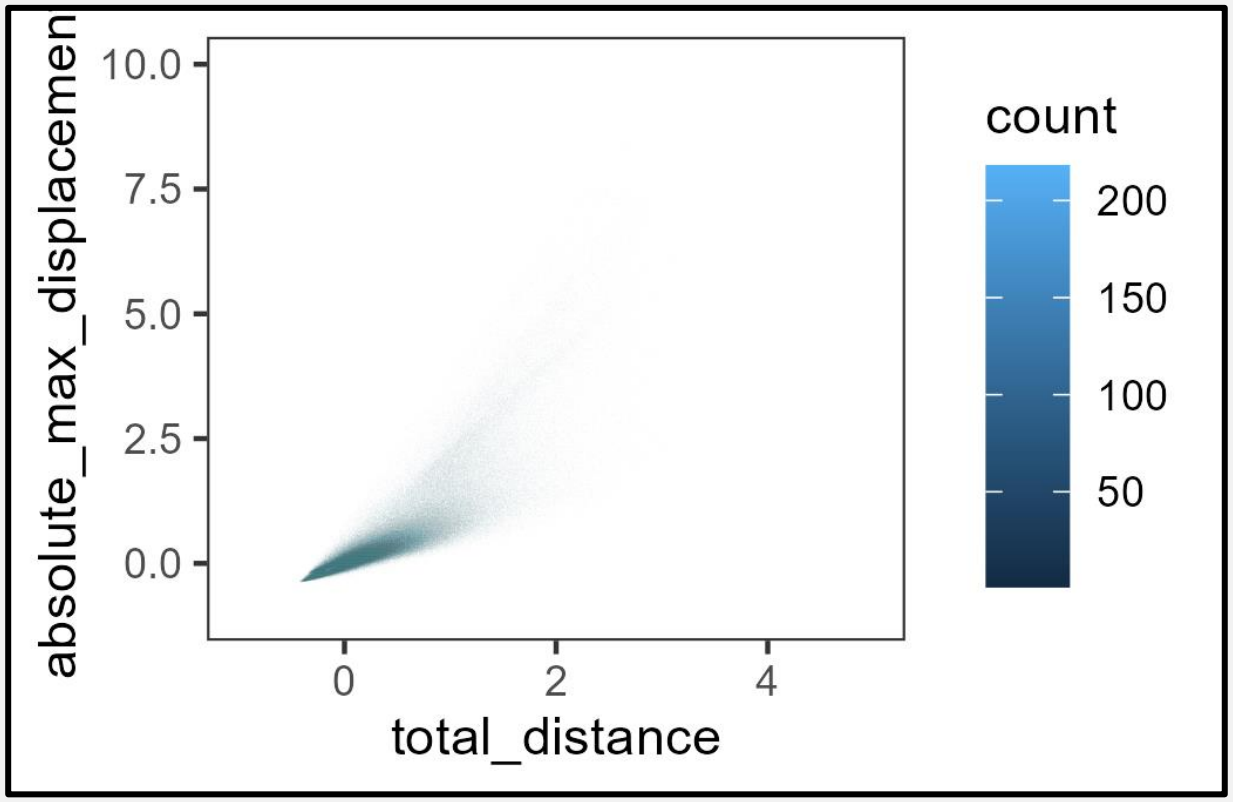
my_codes

Features statistics → **Exploring groups by features** → **Exploring groups by attributes** → **ploting**

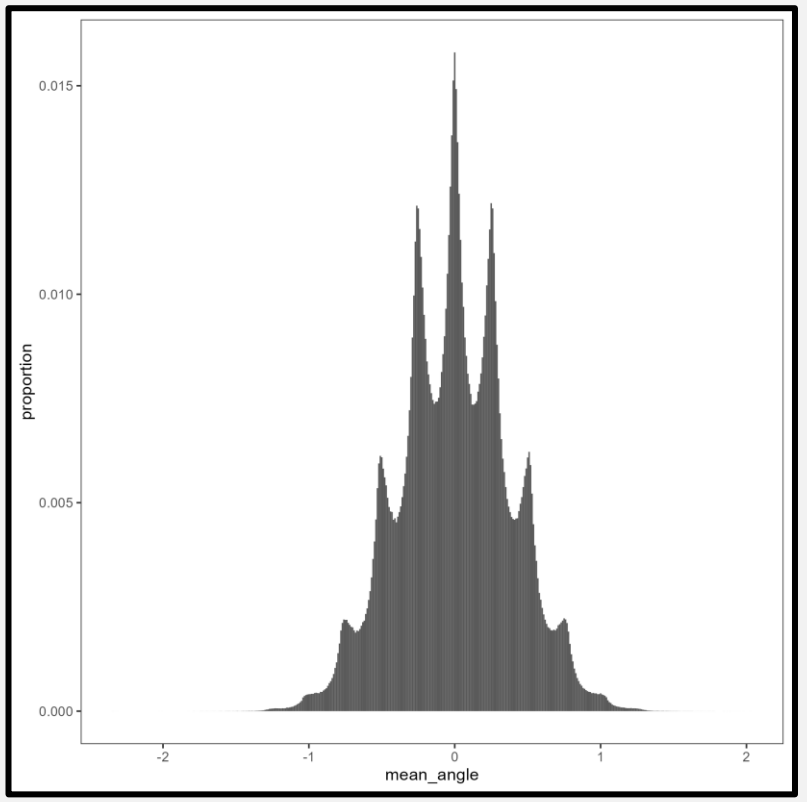
4 analysis.R

5 ploting.R

Code – **Features statistics**



correlations



histograms

Code – **Features statistics**

my_codes

File name: 4 analysis.R

Function name: features_statistics

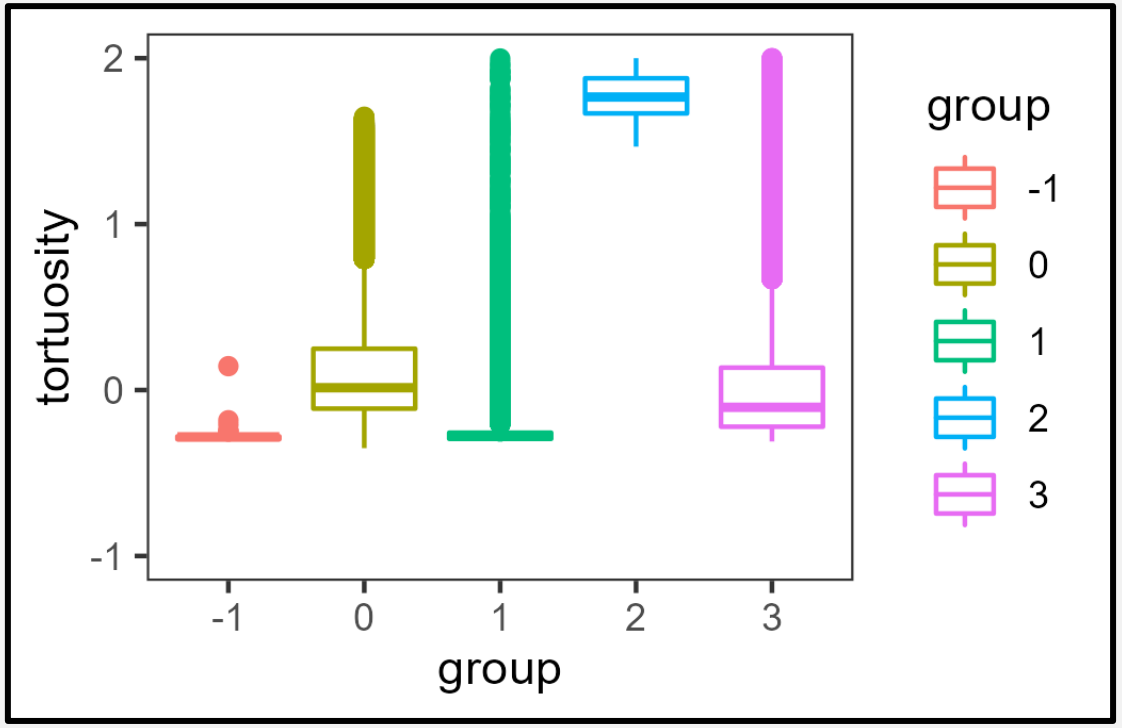
Input: movement_segmentation/outputs/4 features/

Output: movement_segmentation/plots/features_statistics/

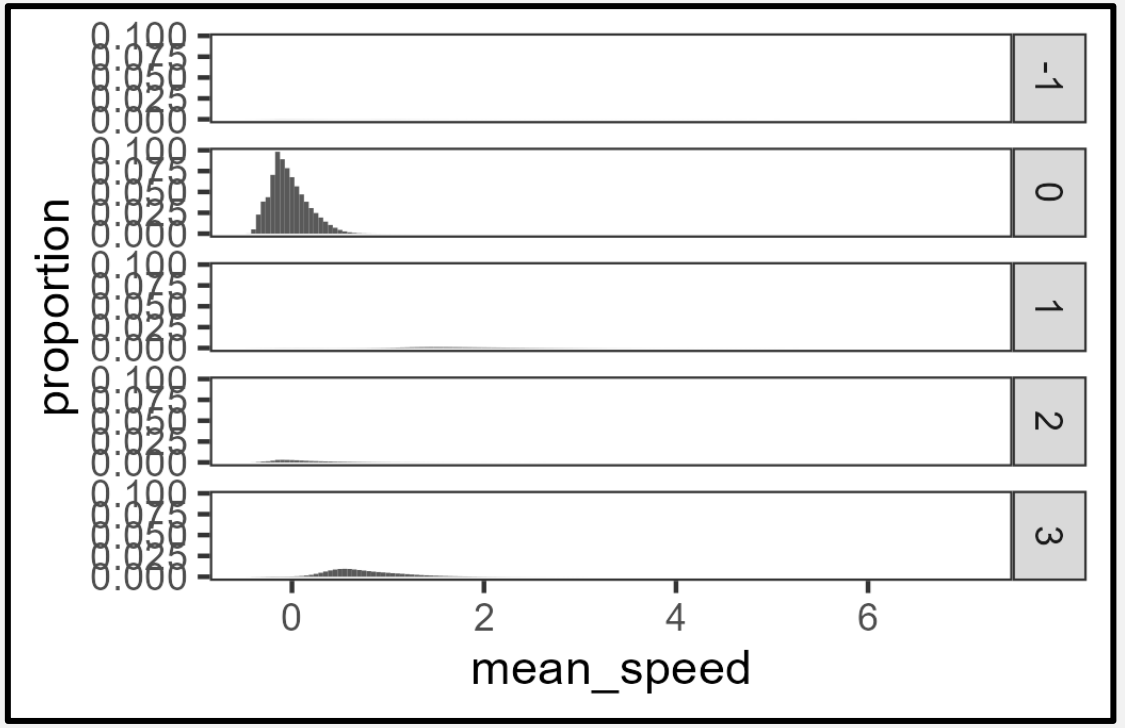
Parameters:

- tw int time window, number of locations in each series

Code – **Exploring groups by features**



boxplots



histograms

Code – **Exploring groups by features**

my_codes

File name: 4 analysis.R

Function name: exploring_groups_by_features

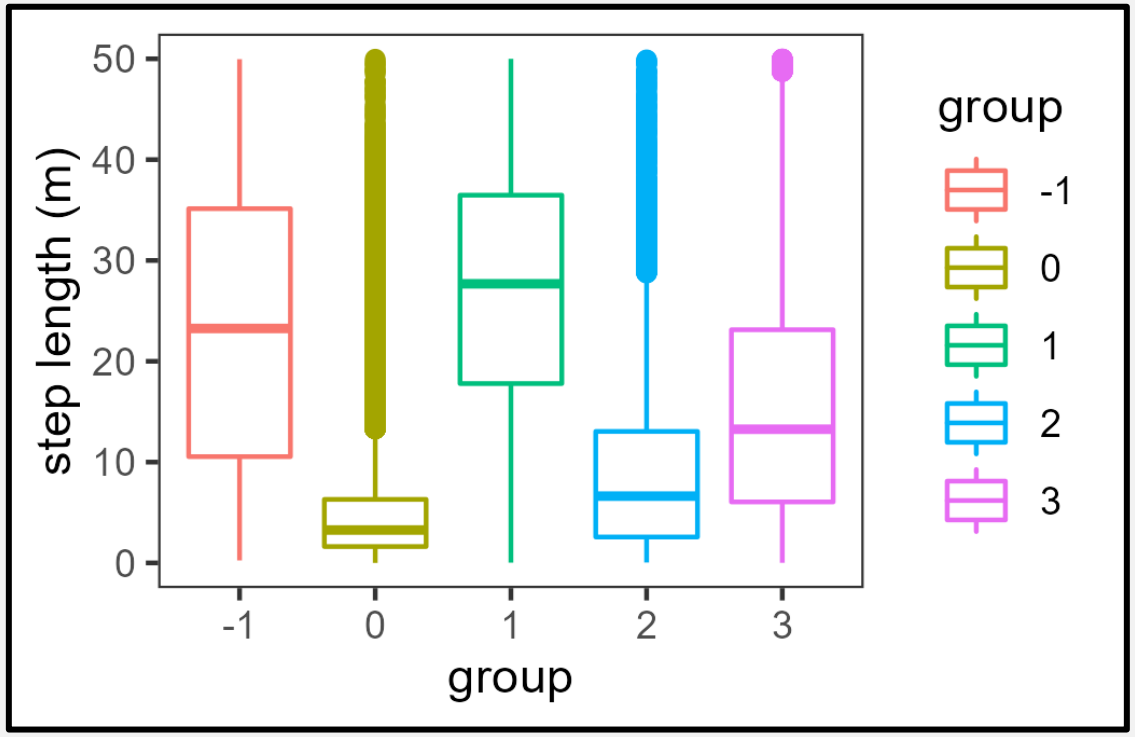
Input: movement_segmentation/outputs/5 kmeans/

Output: movement_segmentation/plots/exploring_groups_by_features/

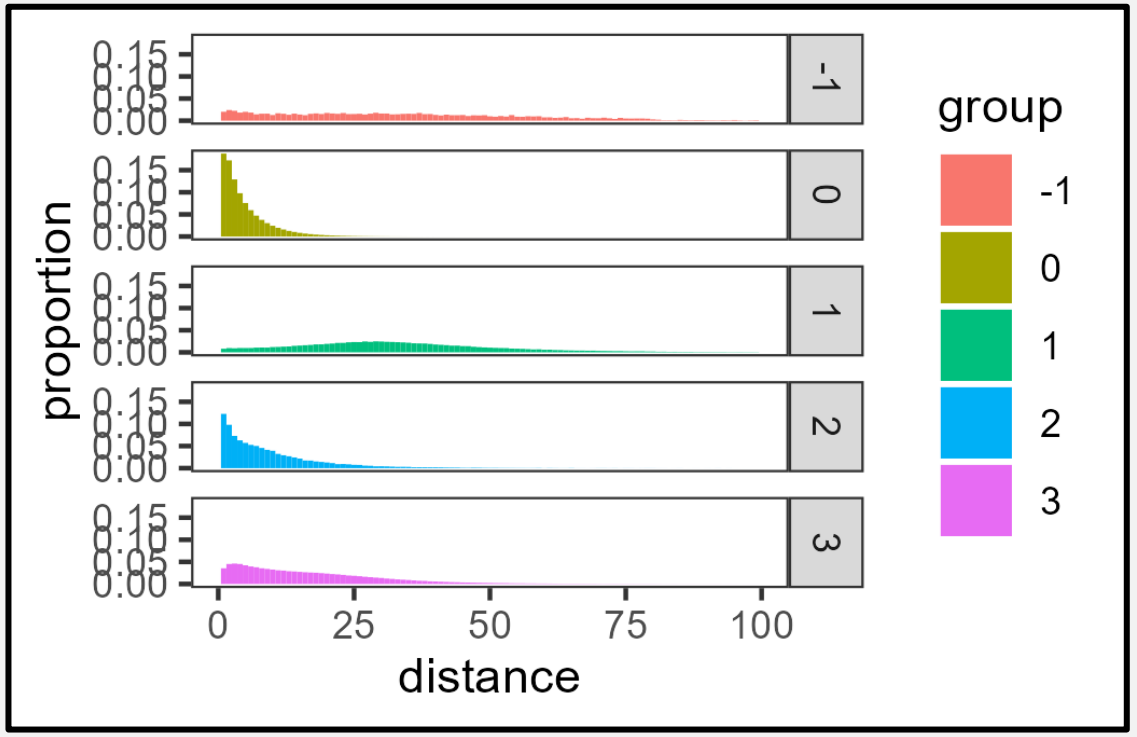
Parameters:

- tw — int — time window, number of locations in each series
- ks — list of ints — which clustering the user want to analyze? (one or more)

Code – **Exploring groups by attributes**



boxplots



histograms

Code – **Exploring groups by attributes**

my_codes

File name: 4 analysis.R

Function name: exploring_groups_by_attributes

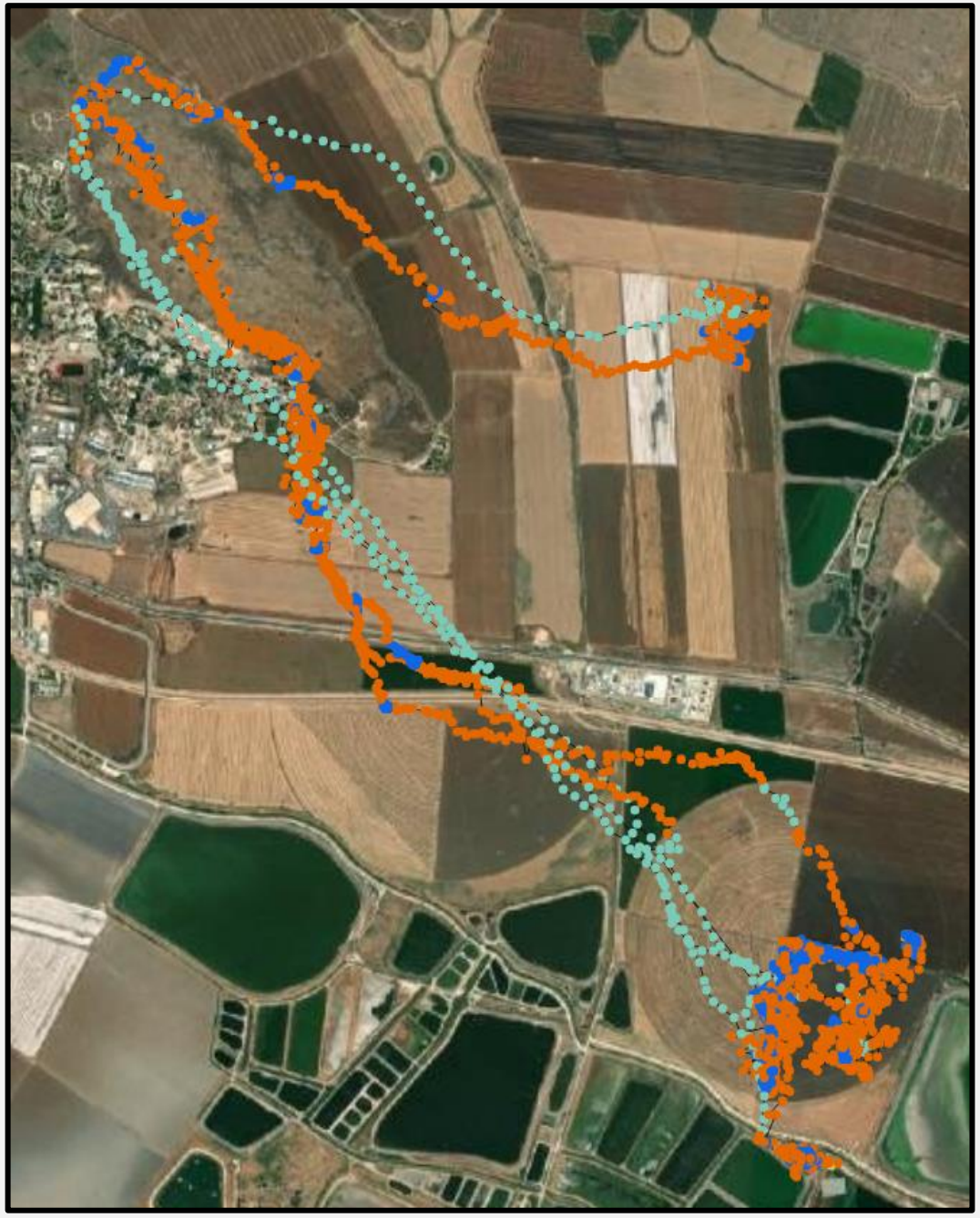
Input: movement_segmentation/outputs/7 smoothed_groups/

Output: movement_segmentation/plots/exploring_groups_by_attributes/

Parameters:

- tw int time window, number of locations in each series
- ks list of ints which clustering the user want to analyze? (one or more)

Code – **ploting**



Code – **ploting**

my_codes

File name: 5 ploting.R

Input: any file from movement_segmentation/outputs/9 examples/

Output: interactive map with the segmentation

Variables:

• tw	int	time window, number of locations in each series
• k	int	which clustering the user want to analyze?
• to_plot	string	which file the user want to plot?

Objectives

- Segmentation of trajectories
- Creating artificial trajectories*

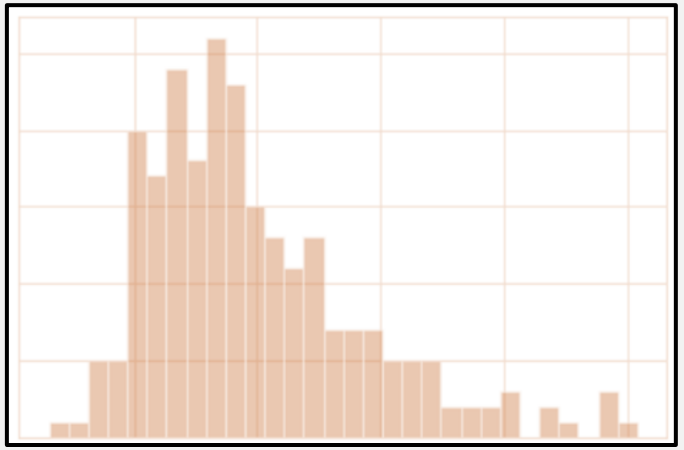
* More work should be done

Intuition – **multi-state random walk**

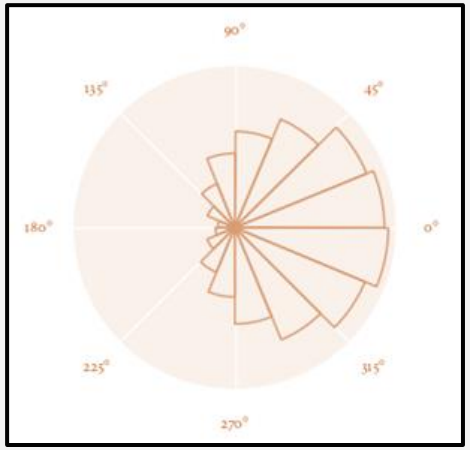
Probabilities for first step

state	0	1	2
probability			

For each state, distributions of:



Step lengths



Turning angles

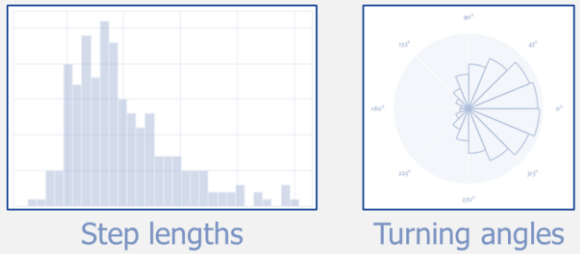
Transition matrix

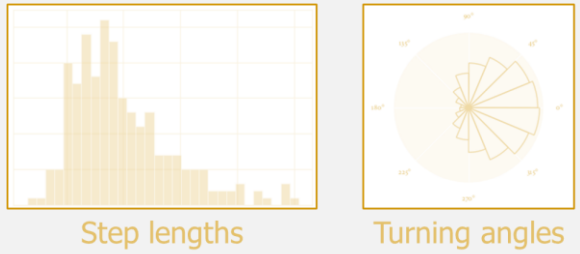
To state…

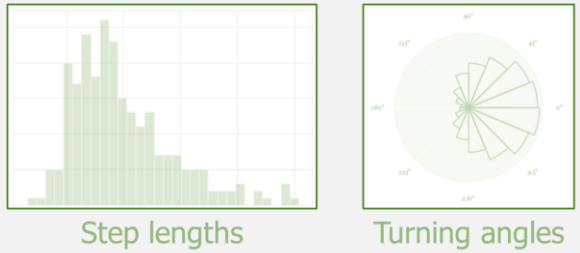
From state…

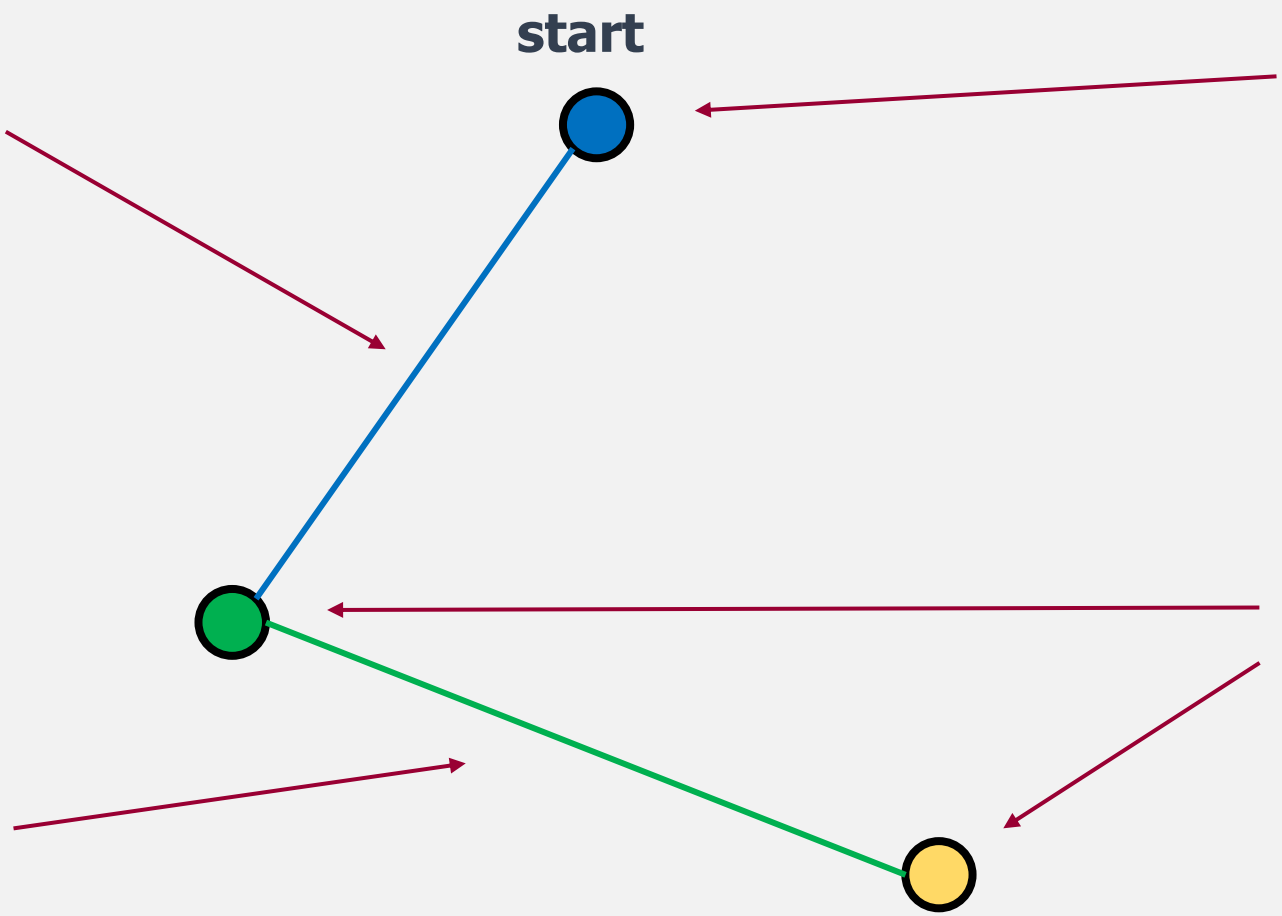
	0	1	2
0			
1			
2			

Intuition – **multi-state random walk**









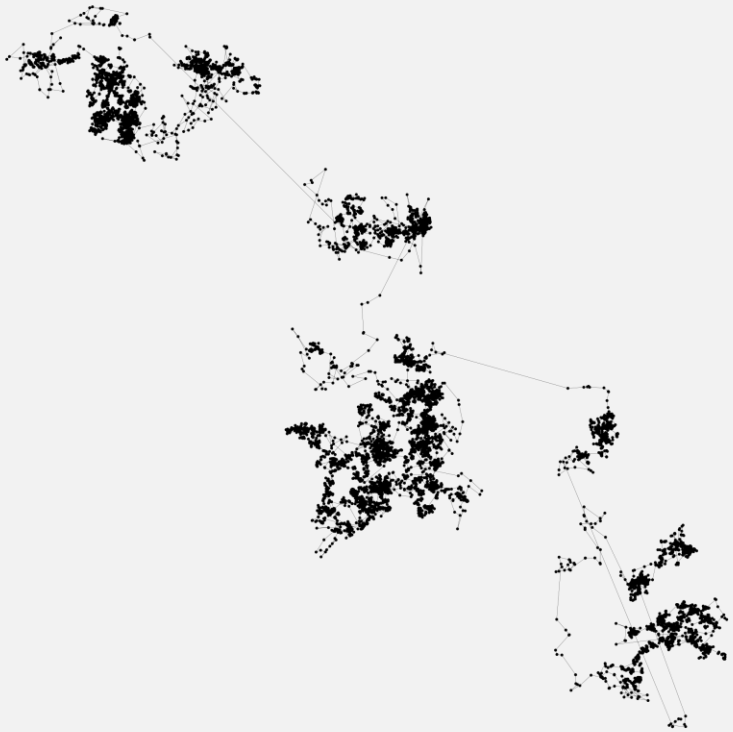
state	0	1	2
probability			

To state…

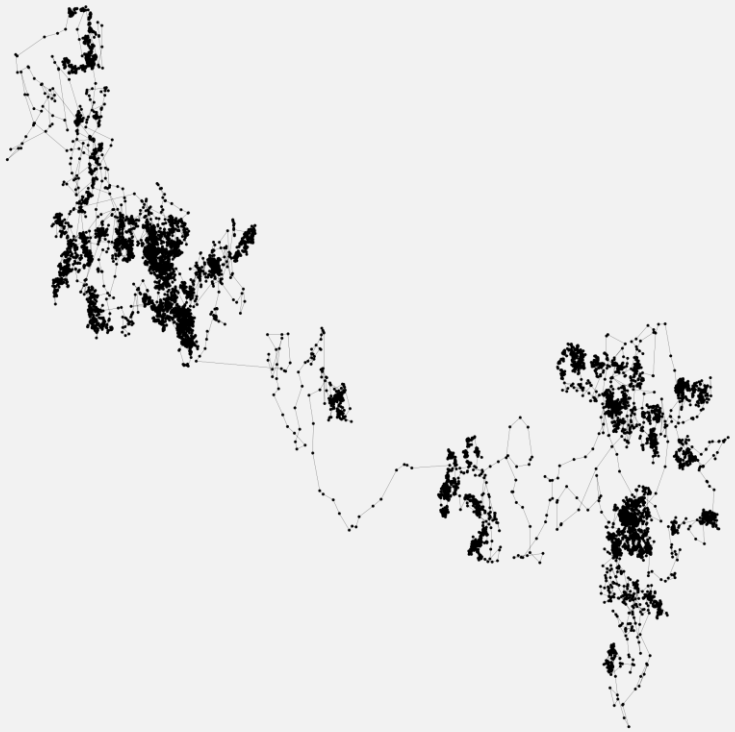
From state…	0	1	2
0			
1			
2			

Examples

k = 2



k = 3



k = 4

Code – **creating artificial trajectories**

my_codes

File name:	6 main_rw.py
Using the function:	multistate_random_walk
Input:	movement_segmentation/outputs/8 transitions/
	movement_segmentation/outputs/10 distributions/
Output:	movement_segmentation/plots/11 artificial_trajectories/

Variables:

• tw	int	the results of which time window to use? (choose the one you did the all the analysis for)
• K_range	list of ints	the results of which clustering (k) to use? (one or more)
• num_of_trajectories	int	number of artificial trajectories to create for each clustering (for each k)
• number_of_steps	int	number of steps per trajectory

Code – **ploting artificial trajectories**

my_codes

File name: 7 plot_artificial_trajectories.R

Input: movement_segmentation/outputs/11 artificial_trajectories/

movement_segmentation/outputs/1 locations_data/

Output: movement_segmentation/plots/artificial_trajectories/

Variables:

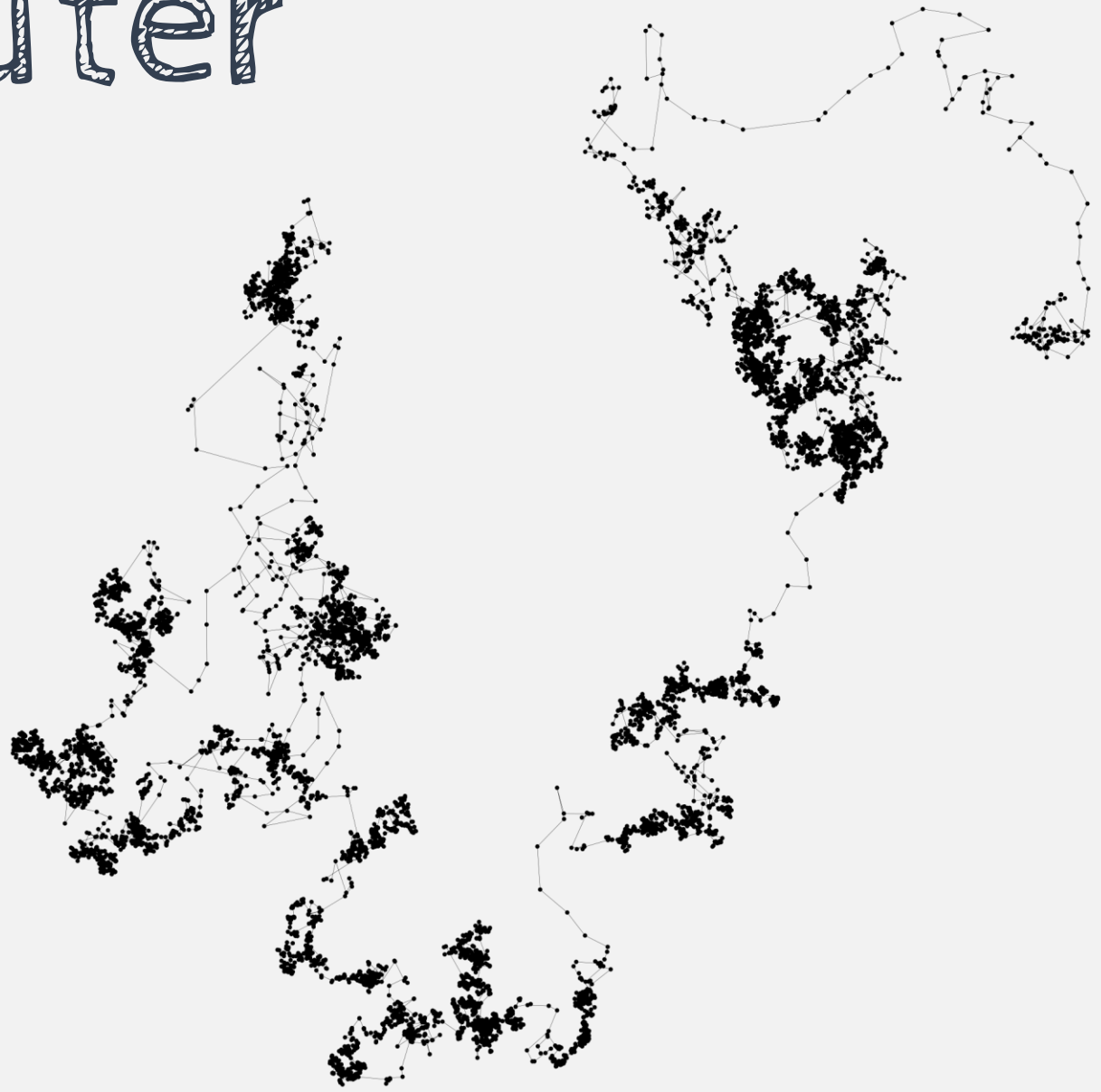
- tw — int — which time window the trajectories were based on?
- ks — vector of ints — which clustering the trajectories were based on? (one or more)
- number_of_traj — int — how many trajectories were created for each clustering (k)?

Real

or

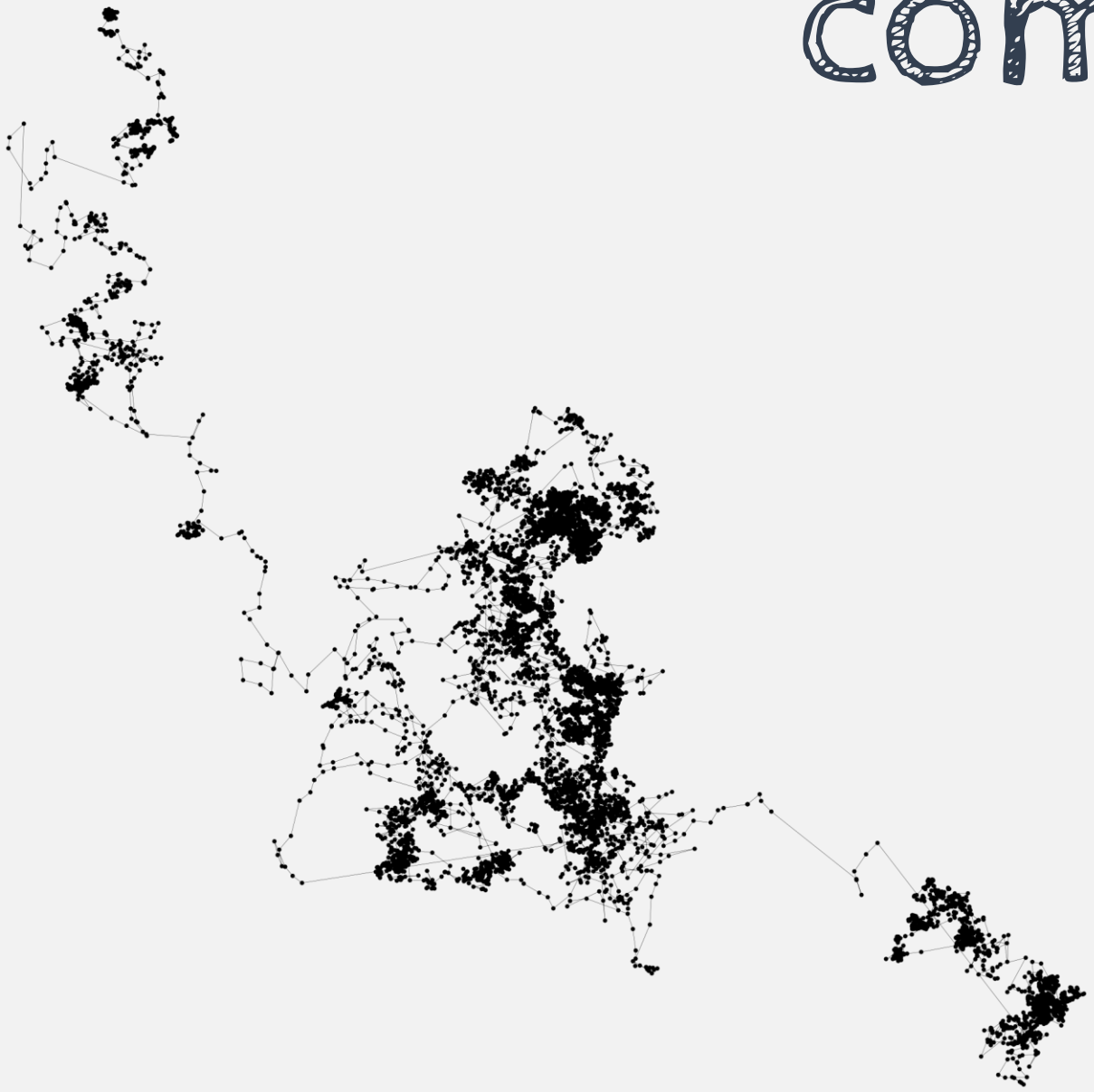
computer?

computer

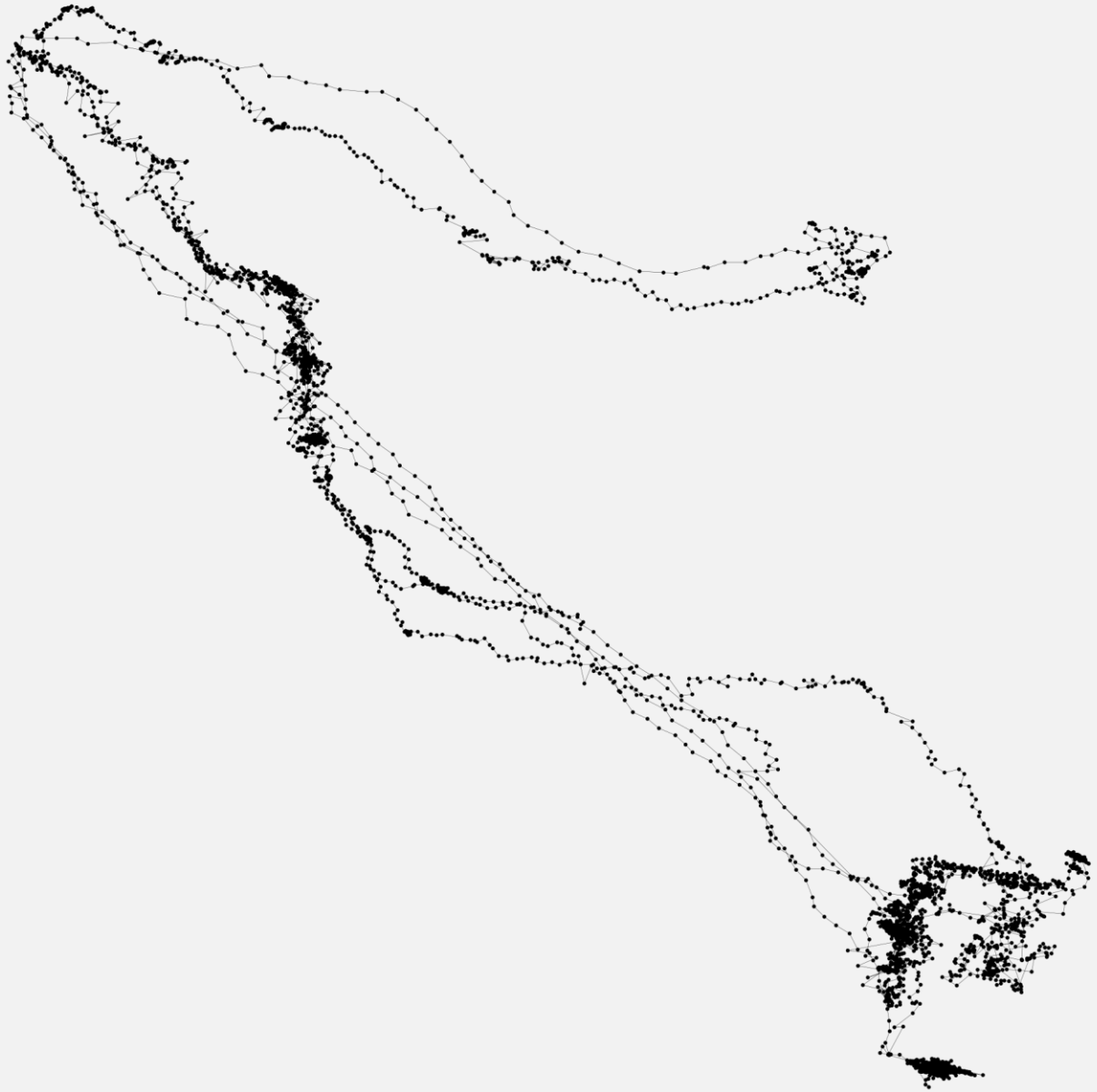


real

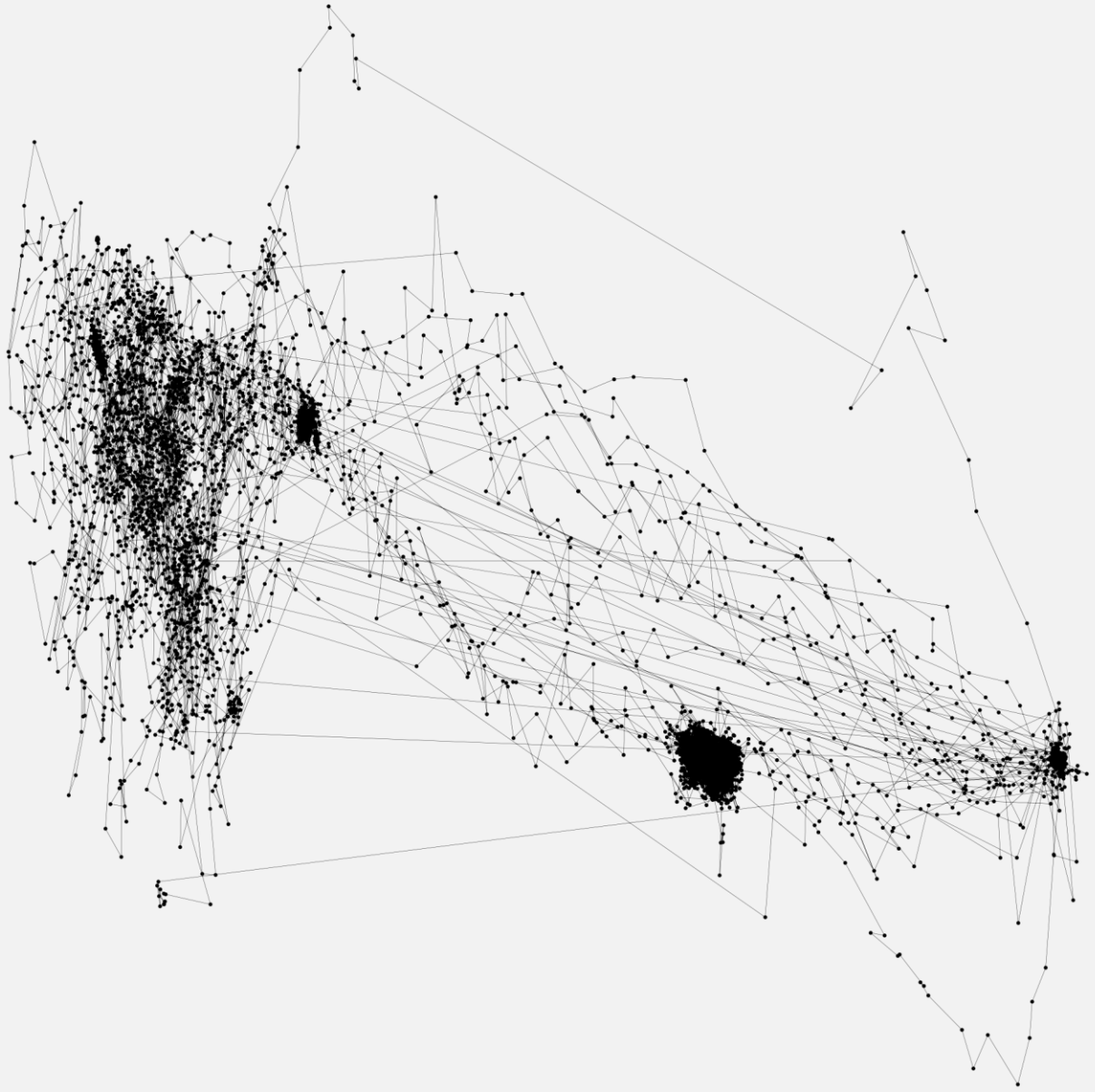
computer



real

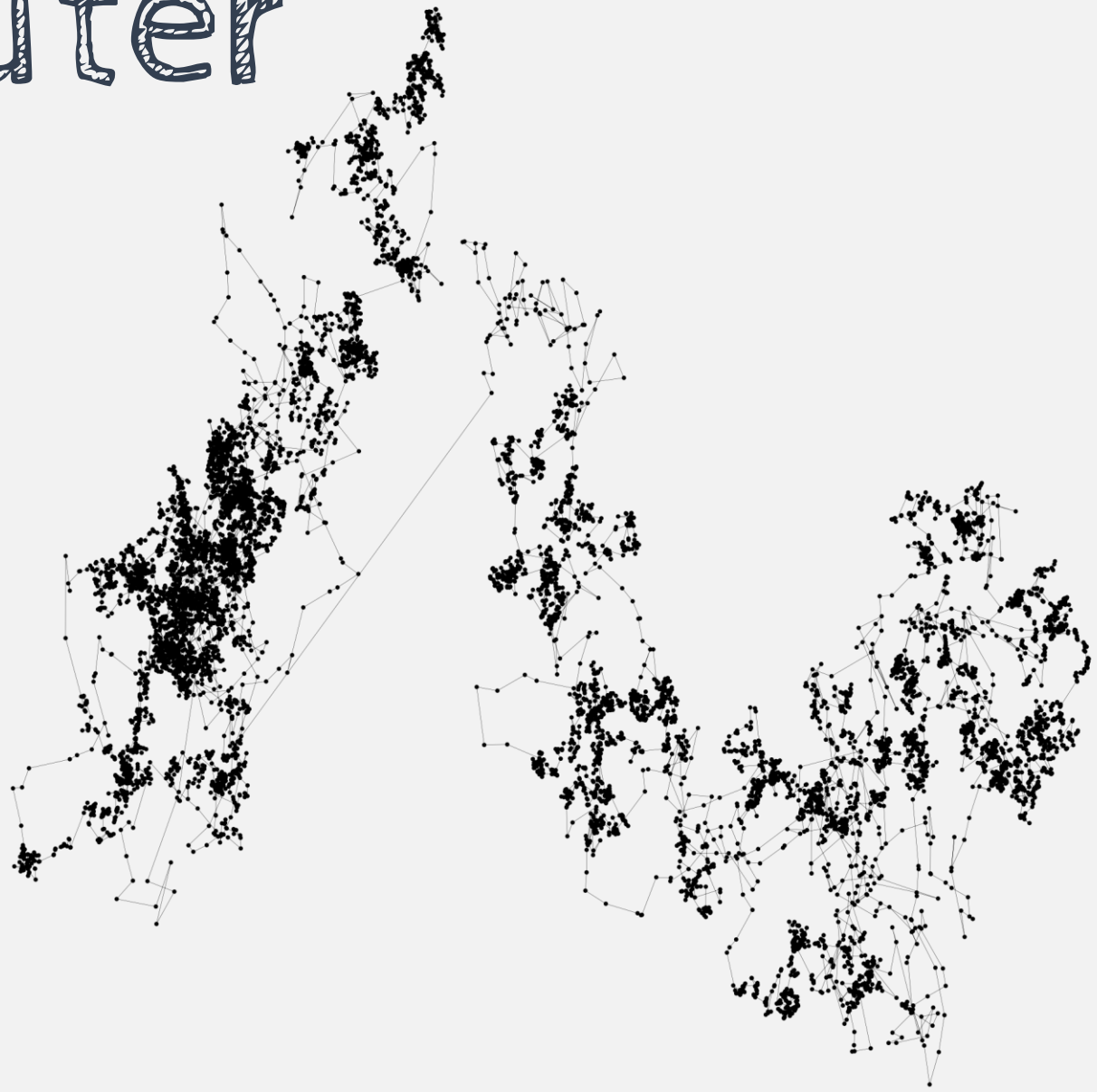


real

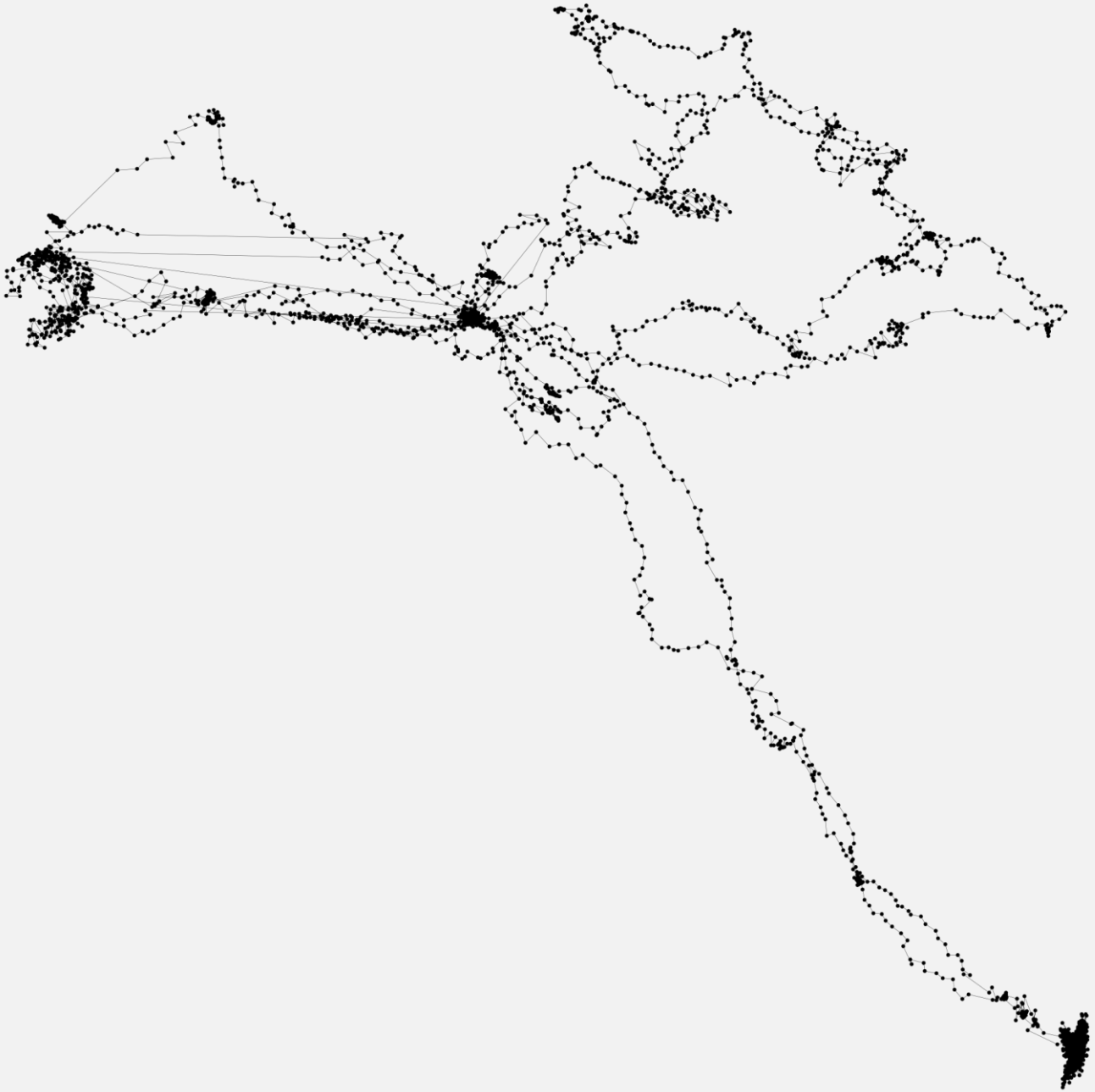


computer

computer



real



Future directions…?

- Exploring seasonality, personality etc. based on the segmentation
- Validating the artificial trajectories:
  - Against other models
  - Against real trajectories
- Improving the random-walk model

For any help contact me!

omerzlotnick@gmail.com