M-Atlas Documentation DRAFT v.1.0

System Requirements

Minimum requirements for the system (actual version):
- Java 1.6 update 20 or higher
- A Postgres database 8.4.1 or higher extended with PostGis 1.5
- 100 MB disk space
- 1 GB Ram (Better 2 GB)

Installation

Types installation

The first step is to compile the new types in the database. To do it, connect to a new PostGis database and compile the SQL files in the folder *Postgis_types*:

- Cluster.sql
- Flock.sql
- Period.sql
- TPattern.sql
- Functions.sql
- CPattern.sql

Database connection

Now the Database is ready for the system. Change the database.properties file in the main folder of M-Atlas to set up the connection. The format is the following:

```
jdbc.drivers=org.postgresql.Driver
jdbc.url=jdbc:postgresql://<hostname>:<port (5432)>/<dbname>
jdbc.username=<username>
jdbc.password=<password>
```

Batch Files

The system is a java application, in the main folder it is possible to find the batch files: Matlas.bat (Windows based) and Matlas.sh (Linux based). They can be used directly if the java path is in the global scope, otherwise open it and add the complete path of the java installation.

Example: Java -> C:\Program Files\Java\jdk1.6.0_14\bin\Java

How to start

Data preparation

To build a dataset of moving points a table of raw observation is needed. It should be in the following format:

ID	LON	LAT	DATETIME
Text	*Numeric*	*Numeric*	*Timestamp*

Where *ID* is the identifier of the user which produces the observation, Lon and Lat are the coordinates and Time is the temporal reference. Starting from a set of these observations we create the dataset executing the following DMQL query:

```
CREATE OBJECT trajectories_table AS BUILD MOVING_POINT
FROM (

    SELECT id,lon,lat,datetime
    FROM <observation_table>
    ORDER BY id, datetime
    )
```

The system will create a new table called *trajectories_table* which contains the set of trajectories.

Example:

Having the table **observation_table**:

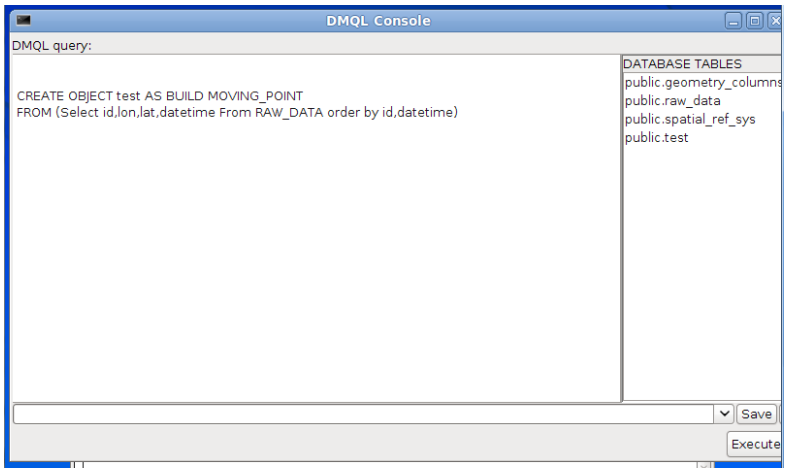
ID	LON	LAT	TIME
1	*49.512*	*5.345*	*2009-08-08 10:02:10*
1	*49.528*	*5.367*	*2009-08-08 10:03:00*
1	*49.530*	*5.401*	*2009-08-08 10:03:50*
1	*49.535*	*5.435*	*2009-08-08 10:05:03*
2	*49.365*	*5.260*	*2009-08-08 09:15:23*
2	*49.490*	*5.279*	*2009-08-08 09:16:02*

The result is a new table called **trajectories_table** which contains the two objects representing respectively a moving point composed by four points and a moving point composed by two points:

ID	OBJECT
1	*[MOVING_POINT]*
2	*[MOVING_POINT]*

Building your first dataset

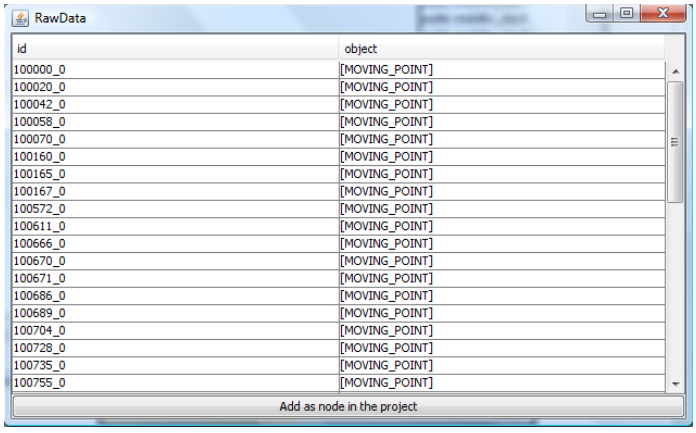
Open the M-Atlas system and Click on “DMQL Console”.



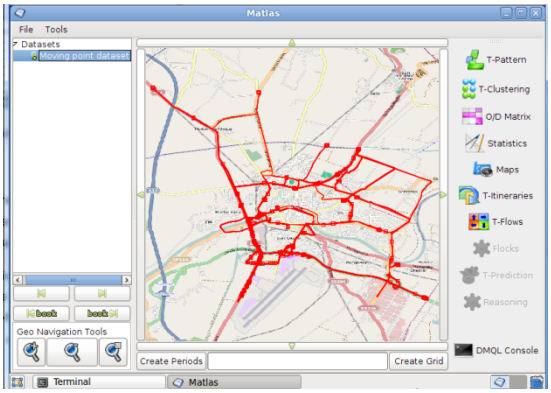
In the “DMQL query textbox” copy the following query and the press “execute”:

```
CREATE OBJECT test AS BUILD MOVING_POINT
FROM (Select id,lon,lat,datetime From RAW_DATA order by id,datetime)
```

After the computation the result in a relational format is shown:



Click on “Add as node in the project” to obtain a graphical representation of the trajectories:

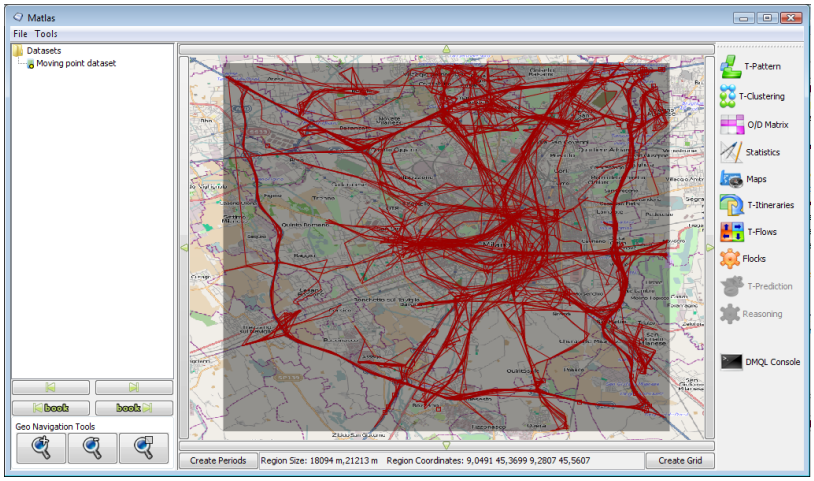


Now the dataset is ready to be used for all the analysis. The node in the tree is only a link to the table in the database. To add to the system an already created dataset it is possible to perform the following operations obtaining the same result:

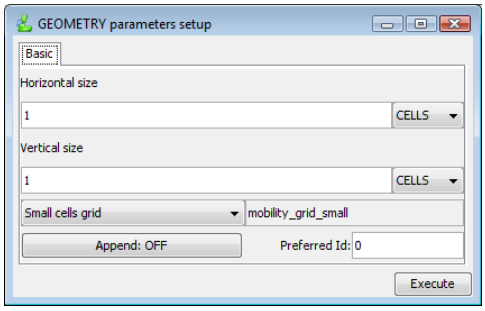
- Execute in the DMQL console the following query: *Select * from <trajectories_table>* and then click on "Add as node in the project" in the window showing the results.
- From the Main window click on File->New Dataset, and then in the first textbox insert the query *Select * from <trajectories_table>* and in the second textbox insert a name for this new dataset, then click "Create". In the Project tree a new node will be added with the name specified. Right click on this new node and select Visualize to graphically represent the new dataset created.

Creating Grids and Periods

In various analysis a grid or a set of periods are requested. To create a Grid visualize your data (or a portion of it) and draw a rectangle on the map holding the right button of the mouse. You will obtain a gray rectangle similar to the following figure:



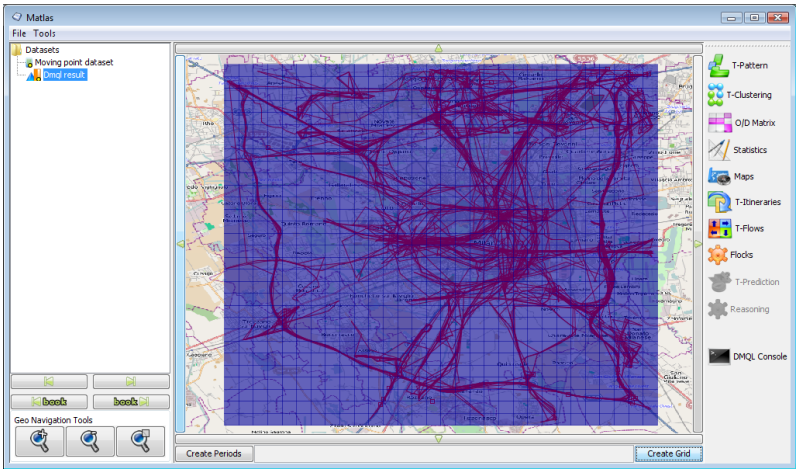
Then click on the button *Create Grid* in the bottom of the window, the following window appears:



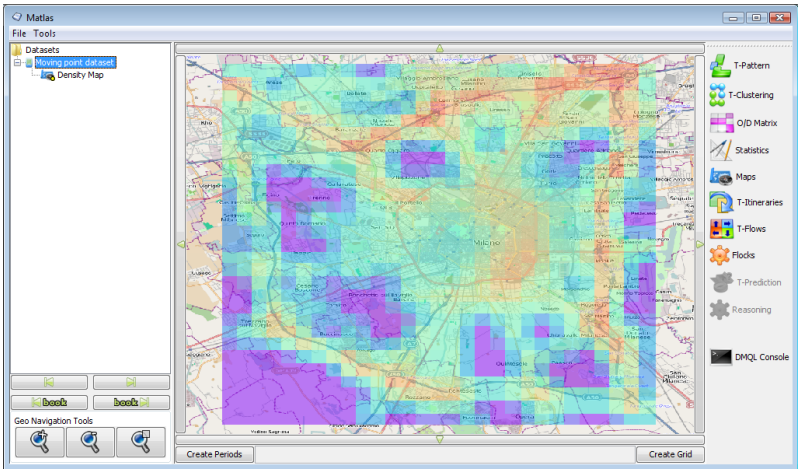
here it is possible to specify the parameters for the creation of the grid, they are:

- the number of cells of the grid in the horizontal and vertical dimensions,
- the name of the table where the cells will be materialized
- if the result must create or replace the table or append it to an already created table of cells.
- The preferred id of the region if the result is a single region (1x1)

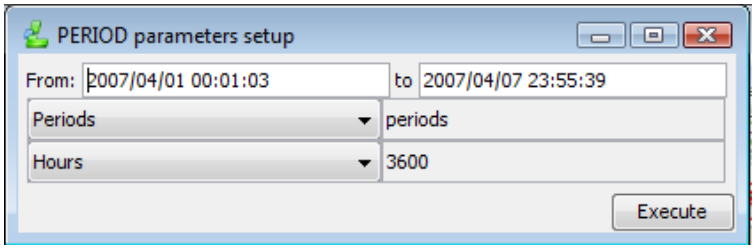
For example specifying a 30x30 grid the result is the following:



This grid can be used for all the analysis which need it, such as the Density Map analysis:



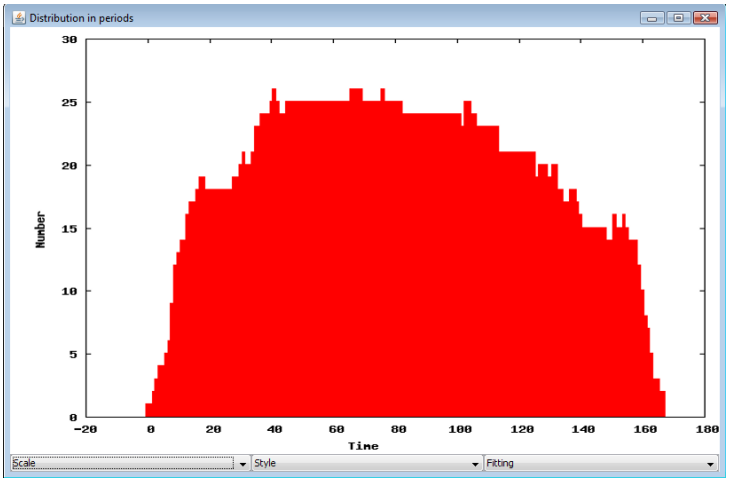
Similarly for the Periods the user must create a table containing them for some of the analysis. Select a dataset of trajectories and click *Create Periods*, the following window appears:



In this case the requested parameters are:

- the initial (from) and final (to) timepoints,
- the name of the table where the periods will be stored
- the length of a single period expressed in seconds

It is not necessary to add the results in the project: the table of intervals is stored in the DB and can be used in all the analysis, moreover the intervals have no space dimension and they can't be visualized in the map. If the selected dataset of trajectories is visualized, the *from* and *to* fields correspond to its minimum and its maximum timepoints. An example of analysis which uses the periods is the *Distribution in Periods* (Statistic).



The axes of the plot are the *number of the trajectories* (y) and the id of the periods (x).

Visual Navigation

Clicking with the right button on a node, the user can visualize the trajectories or the patterns. To navigate visually the results the user can:

- Use the arrow buttons around the map
- Zoom in and zoom out with the first and second button in the navigation tools box.
- Create a region with the right button of the mouse on the map and then zoom in that region clicking on the third button in the navigation tools box.

Cache System

M-Atlas is equipped with a cache system in order to avoid useless computations. When the user asks for an analysis which is already computed, the computation in avoided and the previous result is retrieved from the database. If the user want to recomputed the analysis from scratch (i.e. because the content of the original tables are changed) simply click with the right button of the mouse on the node of the analysis and select “delete cached result”, this means that the result is completely removed from the database.

System Tools

Data construction

The query presented above build a set of trajectories from the raw observations: in the query you can specify a set of parameters using the WHERE statement:

```
CREATE OBJECT trajectories_table AS BUILD MOVING_POINT
FROM (

    SELECT userid,lon,lat,datetime
    FROM observation_table
    WHERE datetime is not null
    ORDER BY userid, datetime
    )
```

```
WHERE MOVING_POINT.MAX_TIME_GAP = 1800
```

In this example the trajectories are cut when the difference in time between two consecutive points are more than half-an-hour (the time is expressed in second).

The complete list of constraints is the following:

- Source data needed: userid,lon,lat,datetime (ordered by userid and datetime)
- MAX_POINTS: the maximum points of a trajectory. If the points are more than this threshold they will be sampled to reach this limit (Infinity)
- MAX_TIME_GAP: the maximum period of time between two consecutive points, otherwise the trajectories is cut and a new one will start from here. (infinity)
- MAX_SPACE_GAP: the maximum length (meters) allowed between two consecutive points, otherwise the point is removed (Infinity)
- MIN_SPACE_GAP: the minimum length (meters) allowed between two consecutive points, otherwise the point is removed (0)
- MAX_SPEED: the maximum speed of an object, the points which not satisfy this threshold will be considered noise. (Infinity)
- REMOVE_REDUNDANCY: Given three consecutive points P1, P2, P3, if the distance between the segment P1-P3 and the point P2 is less than the value of this constraint, the point P2 is removed. The value is expressed in meters. If the value is -1 the constraint is disabled (-1)
- SOURCE_PATTERN: the format of the datetime (yyyy-MM-dd HH:mm:ss);
- IGNORE_ONE_POINT: trajectories formed by a single point (both in space and time) is ignored (true)
- IGNORE_NOISE: if the constraints computation ignores the previously skipped points (true)
- BUILD_STOP: instead creating the moving point of movement it creates the stops (false)
- DISAPPEAR_TOLERANCE: in case of stops creation is the time period considered for the unknown period of stay before and after the first and last observation (86400)

The value inside the parenthesis is the default value. To specify more than a constraint you can use the AND keyword:

MOVING_POINT.<constraint_name> = <value> AND
MOVING_POINT.<constraint_name> = <value>

The trajectories built from the raw observations can be linked to them using the IDs. Due the possibility of having more than a trajectory for the same user the ID generated by the system is composed by two parts:

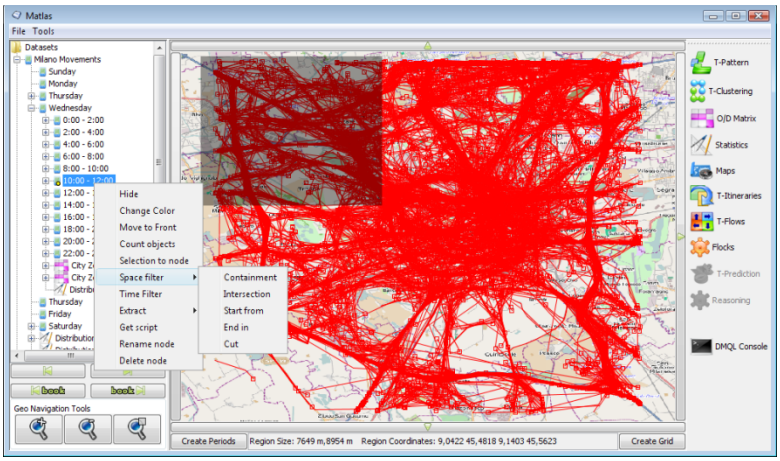
<Original user ID>_<Progressive>

For this reason there is a function in the DB which can be used to split the ID parts: *get_token(ID, index, delim)* where *Index* is the position of the desired part and the *delim* is the special character _. The index can be also negative, in that case the function use it starting from the end of the ID.

Space filters

Creating a region with right button of the mouse on the map, the system enables a set of space filters on every node representing a dataset of trajectories:

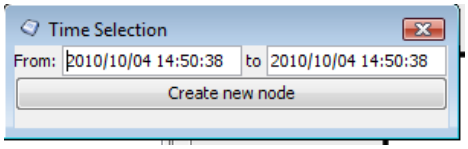
- Containment: Select all the trajectories which are completely contained in the region
- Intersection: Select all the trajectories intersecting the region
- Start from: Select all the trajectories starting from the region
- End in: Select all the trajectories ending in the region
- Cut: Cut the trajectories in the region obtaining the segments of them which intersect the region.



It's important to notice that it is not necessary to visualize all the data to apply these filters; in fact the user can visualize a sample of the data, create the region and then apply the filter on the original one.

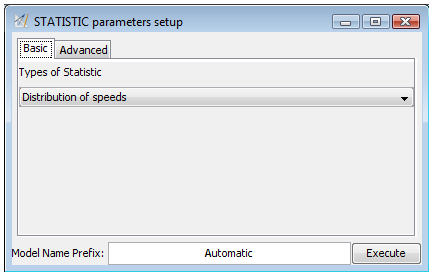
Time filter

On every node representing a dataset of trajectories the user can apply a filter which projects the data in a specified interval of time. To do it, click with the right button on a node and select *Time Filter* from the menu.



Statistics

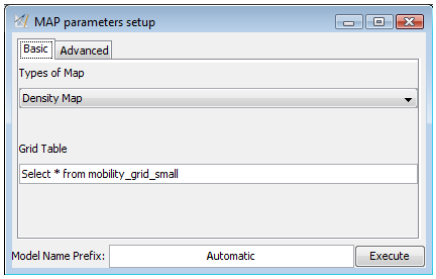
Clicking on the Statistic button in the right toolbar the following panel appears:



It contains the list of statistics which can be computed on a dataset of trajectories. Some of them require additional information such as the table of periods in the example shown above.

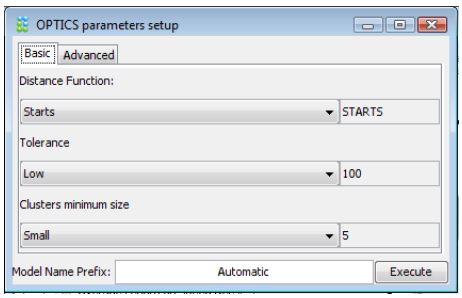
Maps

There are two types of Maps that can be computed: the Density map or CO2 Map. Both of them require the grid to be used.



T-Clustering

The panel shows the parameter used to compute the clustering on the selected dataset of trajectories.



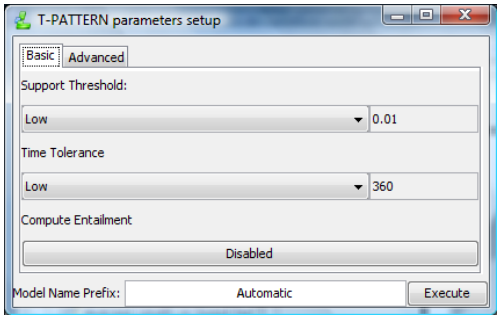
The listed distance functions are:

- Starts: compares the first point of the trajectories
- Ends: compares the last point of the trajectories
- Initial: compares the first k points of the trajectories where k is the minimum length (in points) of the trajectories
- Finish: compares the last k points of the trajectories where k is the minimum length (in points) of the trajectories
- Route Similarity: compares the best alignment of the two trajectories
- ERP Distance: (- Do not works -)
- Co-location: compares the locations visited by the trajectories (the order is not considered). The locations is defined as a buffer around the points, hence a spatial threshold is needed for the computation (in the advanced tab). The distance is in [0,1]: 0 = same locations, 1 = completely different locations.
- St-Co-location: similar to Co-location but consider also time, hence a temporal threshold is needed (in the advanced tab)

The Tolerance is expressed in meters and is used as threshold for the distance between two trajectories, there are some predefined values, but they are not significant for every application, in that cases use the User defined option.

T-Pattern

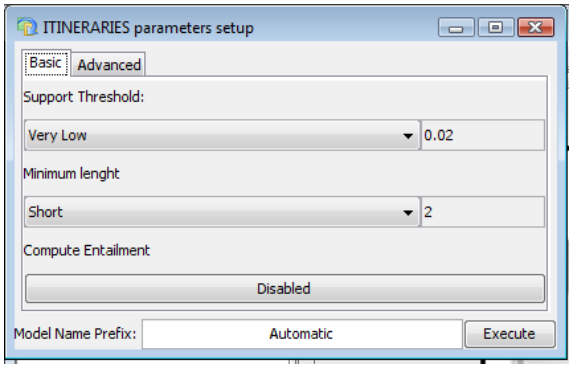
This algorithm discovers a set of sequential patterns describing frequent behaviors of trajectories. A pattern is considered frequent if the number of trajectories which satisfy it is greater than the specified support threshold. The computed pattern is a sequence of regions with transition times. A time tolerance is used in order to approximate the time intervals between two consecutive regions.



The compute Entailment option is used to compute the subset of trajectories of the original dataset which support the T-Patterns extracted. This operation can be very expensive in time and then can be disabled.

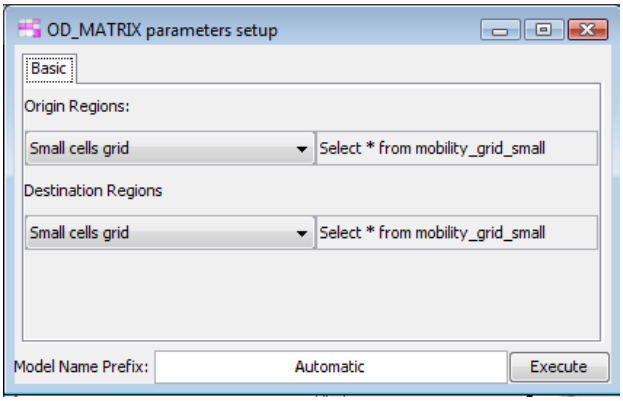
T-Itineraries

A T-Itinerary is a T-Pattern which considers only the space aspect. The minimum length is an additional parameter to filter only the itineraries which are composed by at least the number of regions specified.



T-OD Matrix

To compute the OD Matrix the system requires the tables of regions which define the origins and the destinations. An example of Origins and/or Destinations can be a grid generated on the data.



Roberto Trasarti: roberto.trasarti@isti.cnr.it
Salvo Rinzivillo: salvo.rinzivillo@isti.cnr.it
Fosca Giannotti: fosca.giannotti@isti.cnr.it