

NETS2MAPS USER GUIDE

NETS2MAPS (<http://morfeo.ffn.ub.es/nets2maps/>) is a web platform for the geometric mapping, analysis, and visualization of complex networks and big data systems. Network geometry states that the architecture of real complex networks has a geometric origin. Under this approach, the elements of complex networks can be characterized by their positions in a latent underlying geometry so that the observable network topology is simply a reflection of their distances in this space.

The design of the platform makes it easy for any user to create mappings of real complex networks and carrying out different community detection algorithms. In addition, the platform allows the visualization of embedded networks and their communities with a high level of detail. In this document, we describe the main features of NETS2MAPS and how to use it. Specifically, we explain first the main features of the platform and then we explain step by step how to create a project, how to perform mappings and how to apply community detection algorithms and to obtain visualization of the results.

NETS2MAPS has been specifically designed for creating networks maps in the hyperbolic plane using our mapping tool Mercator [1]. It also includes web applications for the detection of geometric community structure, analysis and visualization of complex networks and big data systems. Throughout this document, we use the words embedding and mapping indistinctly.

1 Main Features

The web platform NETS2MAPS has the following main features:

- Mapping: A network mapping is a map from graph nodes to vectors of real numbers in a multidimensional space. To be useful, a good mapping should work with a low dimensionality and preserve the structure of the graph. The vectors can then be used as input to various network analysis tasks, such as link prediction. Geometric network models are able to explain in a very natural way highly non-trivial properties of real networks, like their self-similarity, community structure, or navigability properties.

In NETS2MAPS you can create your own network maps in the hyperbolic plane using our mapping tool Mercator.

- Community Detection: Community structure refers to the occurrence of groups of nodes in a network that are more densely connected internally

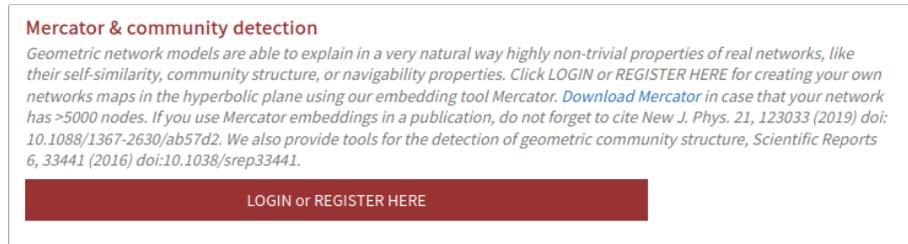


Fig. 1: Section of NETS2MAPS main page with the button to access to Mercator, Community Detection Algorithms and Visualizations.

than with the rest of the network. Communities are often defined in terms of the partition of the set of vertices, that is, each node is put into one and only one community. This is the case of community detection algorithms in NETS2MAPS.

In the platform, two types of community detection algorithms are available [2]:

- In the geometric detection of communities, once the nodes are located in the hyperbolic plane, the system calculates the critical gap and perform a partition between each pair of consecutive nodes whose angular distance is greater than this critical gap.
- In the topological detection of communities the system iterates over possible values of the gap, selects the one that maximizes modularity and uses it to perform a partition between each pair of consecutive nodes whose angular distance is greater than this gap.
- Visualization: NETS2MAPS also allows the visualization of the mappings obtained with Mercator, as well as the partitions obtained through the two available community detection methods. In addition, it allows to filter each representation according to the connection probability between each pair of linked nodes, thus allowing to represent only the most probable links of the network.

The visualization system also allows to configure different visual aspects such as the size of the nodes and the addition of labels that allow you to identify the represented elements.

2 User Creation and Login

The NETS2MAPS platform has a section called *Toolbox* from which you can access the Mercator mapping system and the community detection algorithms as well as the application related to the World Trade Atlas.

Fig. 2: Section of NETS2MAPS login page with the dialog to log in and the button to register a new user.



Fig. 3: Header of the NETS2MAPS toolbox with buttons related with the creation, listing and search of projects.

To access Mercator and the community detection algorithms the user must press the button *LOGIN or REGISTER HERE* which will take him to the login page (Fig. 1). If the user is not already registered, he must register by pressing the *Register* button (Fig. 2). If the user is already registered, he can access directly using his email address and password (Fig. 2).

3 Creating a Project and Uploading a Network

Once the user has accessed the NETS2MAPS toolbox, the system will list the projects he has previously created.

To create a new project, the user must press the *Create new project* button (Fig. 3) which will take him to a screen where he can define the name of the project. Pressing the *Save* button the system will create the project and the menu related to network files, mappings, community detection and visualizations will be available.

To upload his own network into the project, the user must press the *File* tab that will take him to a dialog box where to upload the corresponding file. The format in which the networks must be coded to be used in NETS2MAPS is the adjacent list format and the file extension must be *.net* or *.dat*. The maximum number of nodes per network is limited to 5000. For larger systems, we advise to download MERCATOR from <https://github.com/networkgeometry/mercator> and run it locally.

Once the file is selected, the user must press the *Save* button and after a short check the file will be added to the project and the system will be ready

| N | File | File name | File size (Kb) | |
|---|---------------------------|---------------------------|----------------|--------------------------|
| 1 | Geometric CGM network | Geometric CGM network | 58056 | <input type="checkbox"/> |
| 2 | Topological CGM network | Topological CGM network | 58905 | <input type="checkbox"/> |
| 3 | Geometric CGM-community | Geometric CGM-community | 691 | <input type="checkbox"/> |
| 4 | Plain view | Plain view | 57846 | <input type="checkbox"/> |
| 5 | Topological CGM-community | Topological CGM-community | 744 | <input type="checkbox"/> |
| 6 | mercator44.inf_log | mercator44.inf_log | 6118 | <input type="checkbox"/> |
| 7 | mercator44.inf_coord | mercator44.inf_coord | 14668 | <input type="checkbox"/> |
| 8 | edge_list | WTW_2013.net | 3518 | <input type="checkbox"/> |

Fig. 4: Snapshot of the list of files of a project in NETS2MAPS.



Fig. 5: Available tabs for a project in NETS2MAPS.

to perform a mapping of the network.

4 Project Files

In this section we describe the files that can belong to a project in NETS2MAPS (Fig. 4).

As we have already described, the network input file format must be an edge-list (one link per line, two digit columns) with *.dat* or *.net* extension.

Only one edge list per project is allowed (uploading a new edge list will overwrite the current one). Any input file will automatically named *edge_list*, to view your file or download it click on the file name.

The platform outputs two files after the mapping procedure is finished:

- *inf_coord*: Contains the inferred parameters and the inferred positions.
- *inf_log*: Contains detailed information about the mapping procedure execution.

In addition, the platform generates a file each time a community detection algorithm is applied or each time a type of display is generated.

5 Using Mercator

Once we have added our network to the project, to carry out a mapping using the Mercator tool, the user should click on the *Embedding* tab (Fig. 5) where there are three available options: *Compute*, *Embedding Options* and *Cancel Computing*.

The second option *Embedding Options* allows the user to configure mapping parameters. Several options are provided to adjust the mapping procedure to specific needs and can be combined:

- Custom value for β value. This parameter controls the temperature (randomness) in the S1/H2 model [3]. By default a value for β is inferred by Mercator to reproduce the average local clustering coefficient of the network.
- Custom value for the seed of the random number generator (must be a integer number). This parameter is useful when we want to reproduce a mapping (we should use the same seed).
- Clean output mode. If false, this option outputs a file with extension *.inf_coord_raw* containing the columns 2, 3 and 4 of the file with extension *.inf_coord*. The global parameters (i.e., β , μ , etc.) are not included in the file. Default is false.
- Fast mode. Skip the likelihood maximization step (i.e., only infers the positions using the EigenMap method). Default is false.
- Skip post-processing of inferred values of the radial positions. The inferred radial positions are updated based on the inferred angular positions. When deactivated, nodes with the same degree have the same radial position in the hyperbolic disk. Default is true.
- Quiet mode: When activated, the program does not output details about the progress of the mapping procedure (i.e., the file *.inf_log* is not generated). This mode supersedes the verbose mode (i.e., no output on screen). Default is false.
- Refine mode. When a file containing the previously inferred coordinates is provided (*.inf_coord*), the program uses the already inferred positions and parameters as a starting point and perform another round of the likelihood maximization step to refine the inferred positions. The use of a different output filename is recommended to keep track of the changes. Default is false.
- Validation mode. Validates and characterizes the inferred random network ensemble. This is done by generating a large number of networks based on the inferred parameters and positions. The following files are generated:
 - *inf_inf_pconn*: Compares the inferred probability of connection with the theoretical one based on the inferred parameters.
 - *inf_theta_density*: Density of the angular distribution.
 - *inf_vprop*: Contains the following topological properties for every nodes in the inferred ensemble and in the original network:
 - * *degree*.

- * sum of the degree of the neighbors.
- * average degree of the neighbors.
- * number of triangles.
- * local clustering coefficient.
- *inf_vstat*: Contains the following statistics of the inferred ensemble.
 - * degree distribution.
 - * spectrum of the average degree of neighbors.
 - * clustering spectrum.
 - * *obs_vstat*: Contains the same statistics as above but for the original network.

The first button *Compute* allows us to carry out the mapping of our network using Mercator, having defined or not the mapping options.

Depending on the network structure, the process carried out by Mercator can sometimes take a long time. To stop the computation of the mapping process the user must use the third available option *Cancel Computing*. In case no computation is taking place in the current project, pressing this button will have no effect.

Once the mapping process has been completed, the user will see the information regarding the details of the mapping, such as the inferred β , μ and radius. Now the mapping is ready to be viewed.

To observe the mapping that the user has just done, he should first return to the project screen, using the corresponding button. Then the user must click on the *Visualization* tab and press the *Plain View* button. The *Plain View* button will take the user to the network mapping display screen, where each node occupies the position inferred by Mercator (Fig. 6).

On the embedding display screen the user has a number of tools available to make the network visualization more effective. Specifically, the user has a slider bar that allows to indicate the number of edges to be filtered (hidden). Since in the S1/H2 model used by Mercator [1], each edge has a probability associated with it, this filter allows to indicate the probability threshold required for an edge to be displayed in the representation. In addition, the user has a tick to indicate whether he wants to show the node labels in the network. The user has also a couple of text boxes where to enter the minimum and maximum size values of the nodes in the representation and a button that allows you to apply this new definition of the node sizes. Finally, the user can also download a snapshot of the network mapping visualization in *.png* format clicking the button *Convert to PNG*.

6 Detecting Communities

Once the network mapping has been carried out using Mercator, the user is ready to apply the community detection algorithms available in NETS2MAPS.

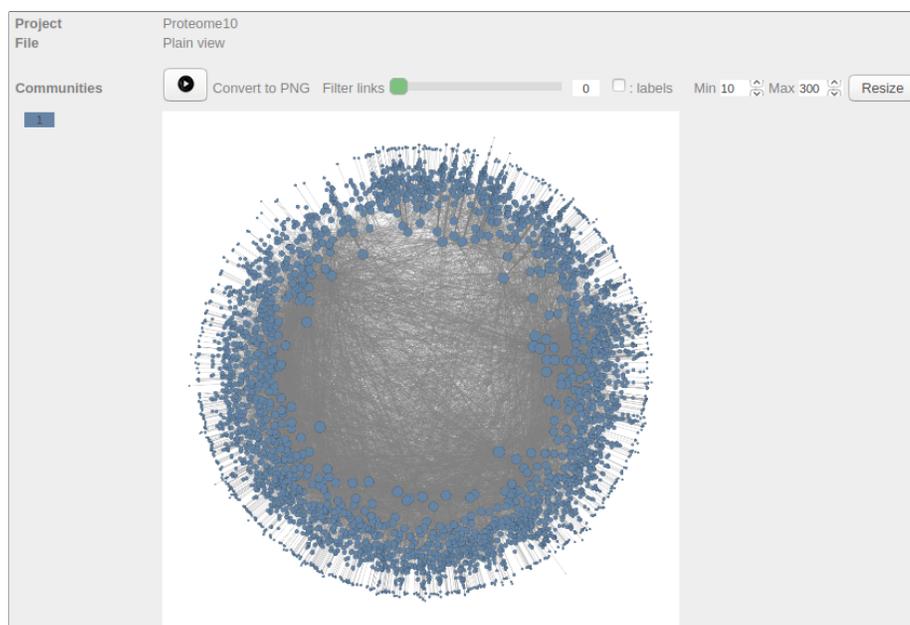


Fig. 6: Example of a plain view of a map performed by Mercator.

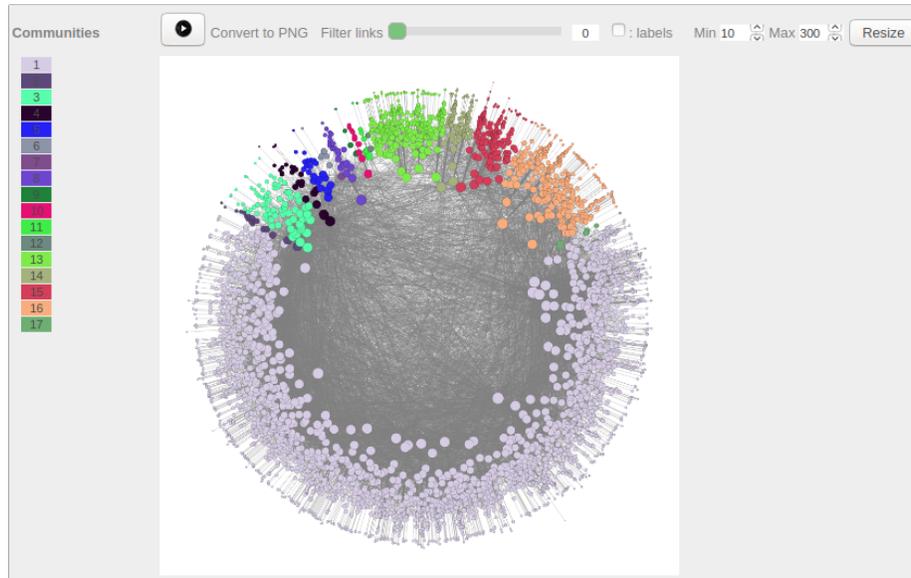


Fig. 7: Example of geometric communities of a map in NETS2MAPS.

As indicated earlier in this document, the platform offers two community detection algorithms that work with the mapping obtained through Mercator: geometric and topological community detection.

NETS2MAPS allows us to apply these two community detection algorithms in two different ways: visualizing the result or without visualizing it.

To carry out one of the communities algorithms visualizing graphically the result (through colors in the nodes representation), the user must click on the *Visualization* tab and select one of the two available methods with the label *network*. In this case, the graphic representation and controls described in the previous section for the visualization of the network will be shown (Fig. 7).

To carry out one of the community algorithms without visualizing it graphically, the user must click on the *Communities* tab and select one of the two available methods.

Finally, there is a third option when obtaining the result of the community detection on a network, and that is the *pie* representation. In this case we will observe the proportion of nodes in each of the detected communities as a portion in a representation of this type. To do this, the user must click on the *Visualization* tab and select one of the two methods available with the *pie* label (Fig. 8).

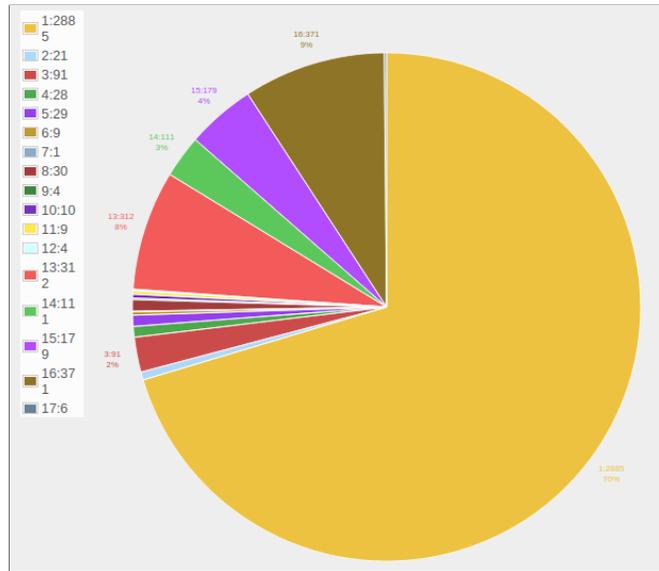


Fig. 8: Example of a pie view in NETS2MAPS.

7 Citation

If you use Mercator mapping in a publication, do not forget to cite New J. Phys. 21, 123033 (2019) doi: 10.1088/1367-2630/ab57d2 [1]. And the detection of geometric community structure, Scientific Reports 6, 33441 (2016) doi:10.1038/srep33441 [2].

References

- [1] Guillermo García-Pérez, Antoine Allard, M Ángeles Serrano, and Marián Boguñá. Mercator: uncovering faithful hyperbolic embeddings of complex networks. *New Journal of Physics*, 21(12):123033, 2019.
- [2] Guillermo García-Pérez, Marián Boguñá, Antoine Allard, and M Ángeles Serrano. The hidden hyperbolic geometry of international trade: World trade atlas 1870–2013. *Scientific reports*, 6:33441, 2016.
- [3] M Angeles Serrano, Dmitri Krioukov, and Marián Boguñá. Self-similarity of complex networks and hidden metric spaces. *Physical review letters*, 100(7):078701, 2008.