**NETMAGE User Guide**

***Availability and implementation:*** Our service runs at <https://hdpm.biomedinfolab.com>. Source code can be downloaded at https://github.com/dokyoonkimlab/netmage

***Contact***: [dokyoon.kim@pennmedicine.upenn.edu](mailto:dokyoon.kim@pennmedicine.upenn.edu)

**1. Creating “data.json” given PheWAS summary data:**

This first section of the guide provides a description of how to take your input PheWAS data and convert it into ***edge*** and ***node maps***, and then apply ***Gephi*** to generate ***data.json*** from these maps. ***Data.json*** can then be uploaded into NETMAGE to create a corresponding disease-disease network.

* 1. **Edge Map:** a comma-separated-value file including all established edges within our disease-disease network. This file should include a column for “Sources” and a column for “Targets” respectively, specifying the start and end of each edge in our map. Other columns, such as edge weight or shared SNPs, can optionally also be included in the file.

Table

Description automatically generated

Figure 1. An example edge map

* 1. **Node Map:** a comma-separated-value file of nodes permitted within our disease-disease network. This dataset should include columns for node names as well as any desired attributes to be displayed in the right side of the visualization.

A picture containing graphical user interface, application, table

Description automatically generated

Figure 2. An example node map

* 1. **Generating the Node and Edge Maps:**

The script ***prepDataForGephi.py*** can be used to generate node and edge maps from input summary PheWAS data.

The script requires:

* A directory of summary PheWAS input data (each file should correspond to a different phenotype, and each row of each file should correspond to a SNP found to be associated with the phenotype)
* The name of the column corresponding to SNP name in the input data files
* The delimiter used in the input data files

Note that the input files need to include at least a column corresponding to SNP ID in order for this script to function. No other parameters are essential for the script to run. If the column name for phenotype ID is not provided in input, the script will use the file names themselves as phenotype names.

The script will start off by filtering data if desired: p-value, minor allele frequency, and case count thresholds can all be applied. To include these filters, simply include the column name as well as the desired threshold as parameters of the script. These filtered data are then processed to create a dictionary mapping each phenotype to a list of its associated SNPs. Finally, these SNPs are used to make node maps. If a node map is provided, the associated SNPs (sorted by p-value, if desired), will be appended to it as a last column. If a node map is not provided, a new CSV file will be generated with phenotype and associated SNPs as its two columns. Data are also automatically converted into an adjacency matrix for nodes (the edge map).

Filtration, as well as creation of the node map and the edge map are fully optional. Users can choose to generate these files based upon their inclusion of input and output parameters when calling the python script: 1) if the variable name and threshold value are not specified, data will not be filtered by that variable 2) if an edge map output path is not specified, the edge map will not be generated 3) if a node map output path isn’t specified, the node map will not be generated. For Phecode-based PheWAS data, an input node map that can be used is the “Phecode Definitions 1.2” file: <https://phewascatalog.org/phecodes>. Note that in this case, phenotype IDs in the input data must match the phenotype ID of the input node map file in order for a node map to be generated.

If all functionality is applied, the output of this script includes two CSV files: the edge map and the node map. Here are a couple examples of how to call “prepDataForGephi.py”:

**Example Usages**

1. **Basic (no filtration, input node map, or parameter names beside SNP ID):** python prepDataForGephi.py --input-data "./subset/" --edgemap-output "./subsetEdgeMap.csv" --nodemap-output "./subsetNodeMap.csv" --snp-name "ID" --delim " "
2. **Including all possible parameters:** python prepDataForGephi.py --input-data "./subset/" --edgemap-output "./subsetEdgeMap.csv" --nodemap-input "./phecodeDefinitions.csv" --nodemap-output "./subsetNodeMap.csv" --maf-threshold "0.01" --casecount-threshold "200" --pvalue-threshold "1e-8" --phenotype-name "phenotypeID" --snp-name "ID" --maf-name "af" --casecount-name "num\_cases" --pvalue-name "pval" --delim " "
   1. **Creating a Network in Gephi**

Graphical user interface, application

Description automatically generated

Figure 3. Homepage for Gephi upon initialization. Click on “New Project” to start a new map.

1. Import nodeMap.csv as a Node table. Make sure it is undirected, make sure to add it to the existing workspace, and avoid including extraneous variables

Graphical user interface, text, application

Description automatically generated

Figure 4. Use the “Import Spreadsheet” button to upload both your node and edge maps.

Graphical user interface, table

Description automatically generated Graphical user interface, text, application

Description automatically generated

Figure 5. Layout for Node Map upload.

Graphical user interface, text, application

Description automatically generated

Figure 6. Results of uploading a node map into Gephi

1. Import edgeMap.csv as an Edge table. Make sure to add it to the existing workspace.

Table

Description automatically generated Graphical user interface, text, application

Description automatically generated

Figure 7. Steps to upload an edge map into Gephi

1. Click on overview to see the current network.

A picture containing diagram

Description automatically generated

Figure 8. The map can be visualized by clicking “Overview” after uploading the node and edge maps in the Data Laboratory.

1. Using the Statistics menu at the right of the screen, calculate the degrees of all nodes. If you desire, you can navigate back to the Data Laboratory, select nodes that have degree 0, and delete them so that your map looks cleaner.
2. Calculate the rest of the node statistics (and re-calculate degree if you deleted nodes) by clicking “Run” next to each network statistic in the Statistics menu.

Diagram

Description automatically generated

Figure 9. Use the statistics menu on your map by clicking run for each of the options at the right of the screen

1. Fit your desired layout to your network. Points can also be dragged around to yield different shapes.

Table

Description automatically generatedPolygon

Description automatically generated

Figure 10. An example of using the Fruchterman Reingold layout to reorganize your network.

1. Color nodes by desired variable (i.e., disease category). Size nodes by desired variable (i.e., node degree)

Graphical user interface, text, application

Description automatically generated

Figure 11. The appearance menu can be used to size and color nodes. To color a node by a variable, click “Node”, click the painter’s palette, click “Ranking,” and then select your desired variable. To size a node, click “node,” click the concentric circles, click “Ranking,” and then select your desired variable.

1. Scale edges up to their full capacity to make sure they are visible after exporting

Graphical user interface

Description automatically generated

Figure 12. Scale the edges all the way up so that they will remain visible after exporting the map.

1. Export the graph to “Graph file” format. The output of export will be a single “data.json” file. Make sure that the disease category column name is labeled as “Category” (this is case-sensitive). “data.json” can now be plugged into the NETMAGE software for the generation of a corresponding disease-disease network. Data can also be saved in GEXF or XML format from Gephi.

Graphical user interface, text, application, chat or text message

Description automatically generated

A picture containing text

Description automatically generated

Figure 13. Exporting the map as a “Graph file” will automatically safe the information into a file called “data.json.” This output can be plugged into NETMAGE for the visualization of the network.

**2. Interacting with the network online:**

This second section of the guide provides a brief description of how to interact with the resulting disease-disease network produced by NETMAGE.

* Use the drop-down menu to select a variable by which to search the network map
* If the variable selected is continuous, a 2-input slider will appear, allowing users to specify a range of values they want to be permitted. The values allowed in this slider will vary depending on the range of values for the chosen variable in the dataset
* If the variable selected is categorical, a text-box input will appear, allowing users to search for nodes that match the search term
  + The Group Selector can be used to display nodes within each category. Nodes are colored as well by the variable chosen for category.
  + Any node within the graph can be selected – hovering over a node will show the node’s name
* Selection of a node will display the chosen node and all first-degree neighbors.
* The right side of the visualization will then display a variety of network statistics, such as Modularity Class, Eccentricity, and Degree, as well as other variables included by the user (i.e associated SNPs)

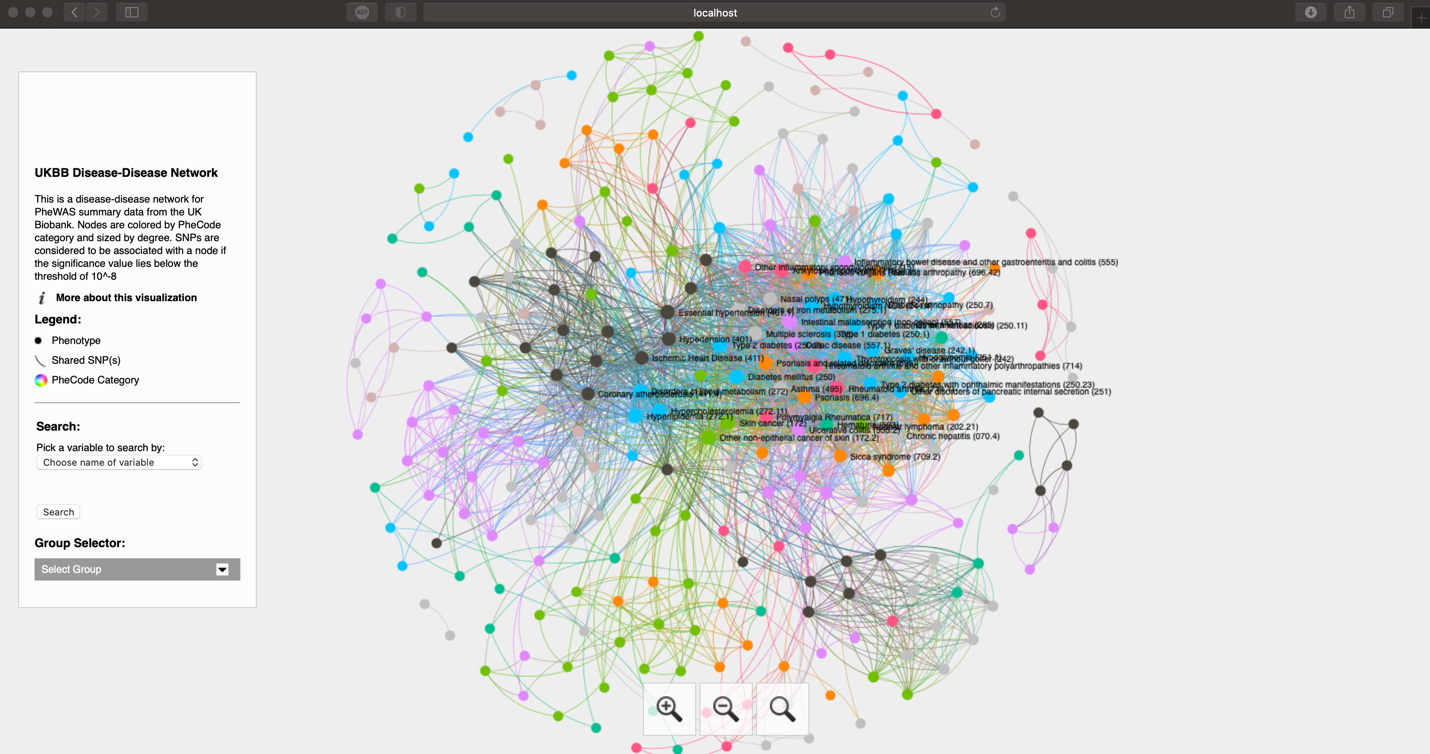
****

Figure 14. A video demonstration of how to interact with networks produced by NETMAGE.