

Table of Contents

1	Brief overview	4
1.1	<i>Glossary</i>	4
1.2	<i>Data types handled by EDENetworks</i>	6
1.2.1	Genotype matrix for individual samples:	6
1.2.2	Presence-absence and presence-abundance data:	7
1.2.3	Distance matrix:	7
1.2.4	Network data:	7
1.3	<i>System requirements</i>	7
1.4	<i>Installing and uninstalling EDENetworks</i>	8
	<i>How cite EDENetworks</i>	8
	<i>Acknowledgements</i>	8
2	Getting started	9
2.1	<i>Input files</i>	10
2.1.1	Genotype matrix	10
2.1.2	Presence-absence and abundance data	10
2.1.3	Distance/dissimilarity matrix	11
2.1.4	Network data	13
2.1.5	Auxiliary data for nodes (samples or sampling sites)	13
2.1.6	Spatial coordinates for drawing the network	15
2.2	<i>Interface</i>	17
2.2.1	Begin a new analysis project	17
2.2.2	Genotype matrix : Individual-centred or location-based	17
2.2.3	Distance matrix	18
2.2.4	Network data	19
2.3	<i>Data analysis</i>	19
2.3.1	Distance matrix	19
2.3.2	Schematic overview of data input and analysis	21
2.3.3	Deriving, Analyzing and Visualizing Networks	22
2.3.4	Analyzing network structure	23
2.3.5	Visualizing networks	24
2.3.6	Randomizations	25
2.3.6.1	Statistical analysis for betweenness centrality	25
2.3.6.2	Statistical analysis for clustering coefficient	27
2.3.7	Output files	28
3	Methodological outline: Details and references for the calculations	29
3.1	<i>Ecological distances</i>	29
	Presence-absence and presence-abundance data	29
3.2	<i>Genetic distances implemented</i>	29
3.3.1	Individual-centred	29
3.3.2	Presence-absence data	30
3.3.3	Population-centred	30

3.4 <i>Network descriptors</i>	31
3.4.1 Paths and components.....	31
3.4.2 Minimum Spanning Tree	31
3.4.3 Thresholded Network	31
3.4.4 Percolation.....	31
3.4.5 Clustering	32
3.4.6 Assortativity	32
3.5 <i>Node descriptors</i>	32
3.5.1 Degree.....	32
3.5.2 Clustering	32
3.5.3 Shortest paths and diameter	33
3.5.4 Betweenness centrality.....	33
4 Brief guidelines for interpretation of results, and some warnings	34
4.1 <i>Global analysis: Network topology and threshold choice</i>	34
4.2 <i>Clustering</i>	37
4.3 <i>Node-level analysis: specific properties of chosen geographic locations</i>	40
4.4 <i>Benchmarking and performance</i>	41
5 References	45

1 Brief overview

The goal of EDENetworks is to provide researchers with a set of methods for studying population-genetic or ecological data sets in the form of networks built from genetic distance matrices. The built-in network analysis tools allow extracting information on the structure of a system of individuals, populations, or other genetic groups.

EDENetworks has been designed for visualizing and analysing networks in order to study genetic relationships in an entire dataset, without *a priori* assumptions on the clustering of individuals, populations or genetic groups. The only underlying assumptions are linked to the genetic distance measure chosen by the user, and therefore its careful and accurate selection is crucial.

Network analysis is a fairly new but very promising method in ecology and evolution (Bascompte, et al., 2003; Hernández-García, et al., 2007; Proulx, et al., 2005). We recommend a number of articles dealing with this methodology in order to understand both the usefulness and limitations of this holistic approach. There are examples of population-genetics analysis based on networks at the level of individuals (Becheler, et al., 2010; Hernandez-Garcia, et al., 2006; Rozenfeld, et al., 2007; Moalic et al., 2011) and at the level of populations (Fortuna, et al., 2009; Rozenfeld, et al., 2008). Most applications in population genetics are based on the methodology implemented in EDENetworks, where networks are derived from genetic distance matrices by thresholding out the largest distances. Note that there are alternatives, such as the method based on link correlations proposed by Fortuna et al., which is inspired by earlier population genetics graph analysis methods (Dyer and Nason, 2004). This method may be implemented in a future version.

Ecological networks based on the distance-thresholding approach have also been recently proposed for illustrating and quantitatively analyzing relationships between communities (genetic groups) and defining biogeographic provinces, based on ecological distances describing their taxonomic composition (Moalic, et al., 2012).

In addition to population-genetic data, EDENetworks allows for studies of any weighted network; however, its set of analysis methods has been selected for their usefulness in genetic analysis.

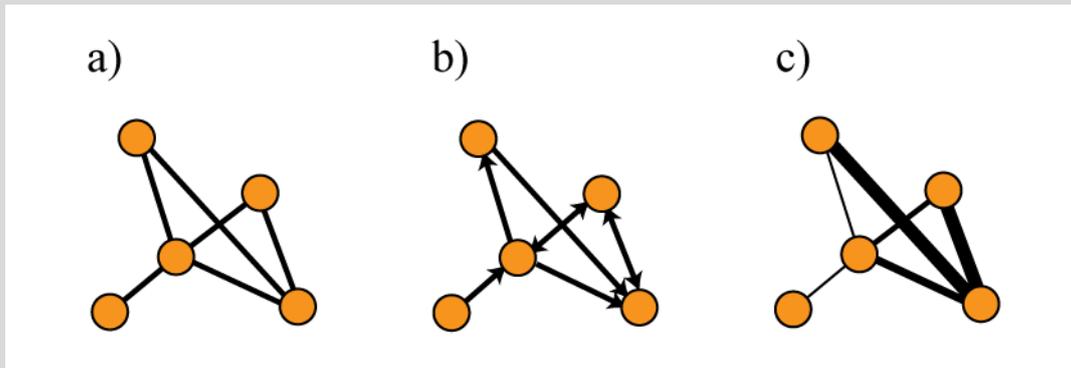
1.1 *Glossary*

In this section, we provide a brief overview of the terminology and concepts of network analysis. This section is by no means a substitute to complex networks literature, and we encourage the users of EDENetworks to get familiar with the fundamentals of network science. As an introduction we may recommend some reviews on network theory (Albert and Barabasi, 2002; Albert, et al., 2000; Newman, 2003; Watts, 2004; Watts and Strogatz, 1998; Newman 2010).

The following glossary may however be useful in order to understand both the manual and analysis methods implemented in the software.

Introductory definitions

Networks consist of **nodes** (or **vertices**) linked by **links** (or **edges**). The nodes represent the fundamental units of the system, such as individuals or populations, and links represent their interactions or relationships. The strength of such relationships can be taken into account in the form of edge weights; in the case of EDENetworks, a genetic distance is associated with every edge.



Examples of possible networks (a) simple and undirected, (b) directed (c) weighted. EDENetworks deals with networks of type (c).

Properties of individual nodes

Degree (also called '**connectivity degree**'): number of edges connected to a node.

Betweenness Centrality: number of shortest paths between other nodes passing through a node.

Average Nearest-Neighbour Degree: the average degree of the nodes to which the node is connected

Clustering Coefficient: the ratio of existing connections between a node's neighbours to the maximum possible number of such connections.

Component : the component to which a node belongs is the set of nodes that can be reached from it by following the links of the network.

Descriptors of network topology

Degree distribution: probability density distribution of node degrees, i.e. the probability of a randomly sampled node to have a given degree.

Average shortest path length (or **geodesic distance**): average number of links on the shortest paths between all pairs of nodes.

Clustering coefficient (or **transitivity**): either the network average of the clustering coefficients of individual nodes, or the ratio of interconnected nodes triplets compared to the total of possible triplets in the network. As a high clustering coefficient value indicates non-randomness in the network structure, this index can be used as a measure of substructure (also understood as hierarchical structure in population genetics, but bearing a very precise and slightly different meaning in physics for hierarchical networks) in the network.

Diameter: The diameter of a network is the length (number of edges) of the longest path between any two nodes.

Small World: Small-world networks that have a low diameter/average path length and high clustering. Such a topology is typical for most natural of networks, including biological, social, and technological networks, as well as some population-genetic networks (Hernandez-Garcia et al., 2006; Rozenfeld et al., 2007).

Fully connected network: a network in which every node is directly connected to every other node.

Connected network: a network which consists of a single component, i.e. all nodes can reach every other via some path.

Percolation threshold: When links are removed from a connected network, it eventually fragments into small components. The point where this happens is called the percolation threshold. More accurately, this is the point where the so-called giant component (whose size is of the order of the network size) disappears and there is no long-range connectivity; even before the percolation threshold small disconnected fragments will appear, yet a substantial fraction of nodes belongs to the giant component.

1.2 *Data types handled by EDENetworks*

EDENetworks can handle a wide range of genetic data types. These data can be input at the level of individuals and used to construct networks of genetic distances, either between the individuals or between the populations they have been sampled from. In the first case, the nodes of a network then represent individuals, and in the second case they represent populations. As an alternative to individual-level genetic data, EDENetworks can also read pre-calculated genetic distance matrices or network files.

1.2.1 **Genotype matrix for individual samples:**

For individual samples, the genotypes can be input as

- Allozymes
- Microsatellites
- AFLP
- RFLP
- SNPs

Based on this data, the software can be used to construct and analyze either a network of individual samples, where the nodes represent samples and edges their genetic distances, or a population-level network, if the input data is augmented with population labels for each individual.

For individual-centred analysis, the following genetic distance measures can be chosen:

- ✓ the Allele Sharing distance or
- ✓ the Linear Manhattan distance (only applicable to microsatellites under assumption of stepwise mutation model)

For population-centred analysis, the available distance measures are:

- ✓ the Goldstein Distance
- ✓ FST Based Distance of Reynolds (D)

1.2.2 Presence-absence and presence-abundance data:

Presence-absence data and presence/abundance data can be analyzed with EDENetworks similarly to individual-centred genetic data. The distance measure is Bray-Curtis dissimilarity for presence/abundance data, and Sorensen or Dice index for presence/absence data.

1.2.3 Distance matrix:

Instead of inputting genotypes of every individual, the user can pre-construct a genetic or ecological distance matrix outside EDENetworks, and input this matrix for network analysis. Such a matrix can represent either distances between individuals or distances between populations, such that rows/columns correspond to individuals or populations, respectively. The user is free to choose any distance measure, as long as it yields a pairwise distance matrix.

A non-exhaustive list of examples includes:

- ✓ Genetic distance: Fst, Rst, Goldstein, Nei...
- ✓ Ecological distance: Jaccard, Bray Curtis, Manhattan...

1.2.4 Network data:

Genetic or ecological distance data may also be input directly in the form of a network, where nodes represent individuals or populations and edges their distances. This network may or may not be pre-thresholded; however, for fully connected networks where an edge links every node, we recommend inputting a distance matrix as above. Network data can either be stored as a GML file (Graph Markup Language, .gml), or an edge file (.edg) where one row denotes an edge, so that there are three entries per row: (vertex1 vertex2 edge_weight).

1.3 System requirements

EDENetworks can be installed on any computer running Windows, and has been tested on Windows XP and 7. At least 1 GB of memory is recommended for analyzing larger networks. A Linux version is also available as a .deb package, which has been tested on Ubuntu Linux. The program can also be run from the source code in any system that has Python, NumPy, Matplotlib, SciPy and PyLab installed.

1.4 Installing and uninstalling EDENetworks

To get the latest version of EDENetworks, go to <http://becs.aalto.fi/edenetworks/> and download the version suitable for your operating system.

For Windows users there is an installer that you need to execute. It will guide you through the rest of the installation process. After installation you should have a folder in your start menu containing short cuts for launching EDENetworks and for uninstalling it.

For other operating systems, you need to use the Python source code to run EDENetworks. Unzip the EDENetworks.zip to a directory of your choice and then execute the script file *eden_launcher.py*. To uninstall, simply delete the entire folder. Note that EDENetworks requires some software packages which need to be installed in order for it to function. These packages are: Python, NumPy, Matplotlib, SciPy and PyLab. You will also need to compile Himmeli version 3.0.1 and place the executable in a subdirectory *himmeli_3.0.1*. The correct version of Himmeli can be downloaded from the EDENetworks website.

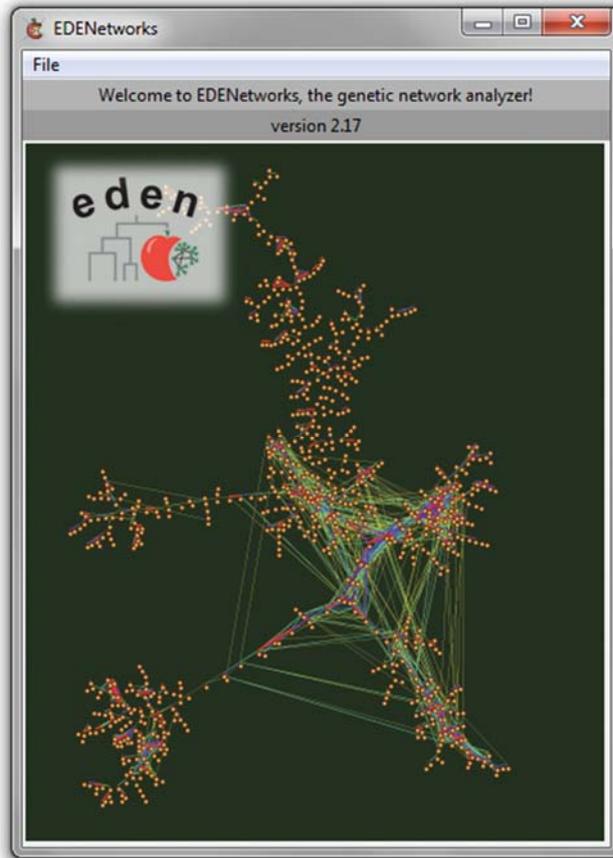
How cite EDENetworks

Instructions for citation can be found at <http://becs.aalto.fi/edenetworks/>.

Acknowledgements

This software had been made possible by the collaborations developed through the project EDEN: Ecological Diversity and Evolutionary Networks (<http://ifisc.uib.es/EDEN/>). This research project was supported by the New and Emerging Science and Technology programme (NEST) of the 6th Framework Programme of the European Commission, under the NEST-Pathfinder Complexity initiative. It started on 1st January 2007 and finished on December 2010. EDENetwork was tested on multi format datasets, completed and finalized in the framework of the ANR project Clonix (2011-2015; <http://www.ifremer.fr/clonix/>).

2 Getting started



As for any other software, we recommend to read this manual before using EDENetworks – it has been designed to provide you with all necessary information to perform your analysis. For assistance in interpreting the results, as well as recognizing the limitations of the analysis methods of EDENetworks, we recommend reading the references cited in this manual.

Examples of input files are provided with the installation package. They may be opened with any text editor such as Textpad (available on <http://www.textpad.com>) to check for the exact format (i.e. spaces and tabulations). Most input files can be prepared either in a text editor or Excel, and then saved as text files (.txt) with tabulator separators.

One to two input files should be prepared for each analysis. The first one is always required, containing the genetic data to be analyzed. Depending on what is being analyzed, the second optional file can contain auxiliary information either on sampled individuals or their populations.

Workflow in a nutshell

Typical use of EDENetworks consists of the following steps:

- Genetic data is imported
- Properties of the resulting distance matrix are inspected
- Distance matrix is thresholded, yielding a network
- The network is visually inspected for understanding structural features of the genetic system
- The network's properties are analyzed
- Key quantities such as betweenness centrality of nodes in population networks are inspected
- For population networks, the significance and robustness of key quantities is assessed
- Visualizations, graphs, data on nodes, and data on networks is saved.

Below, we will go through these procedures step by step.

2.1 Input files

EDENetworks accepts four kinds of input data: genotype data, distance matrices, network data and auxiliary data for the nodes. Any whitespace can be used as a separator.

2.1.1 Genotype matrix

A genotype matrix file should contain the genotypes of each individual. The file should be a text file where each row corresponds to the genotype of an individual, such that

- the first column contains the sampling location identifier
- the second column contains the identifier of the individual sample
- the genotype is described starting from the third column, with each **allele encoded as digits** and one column per allele. You can represent missing alleles by entering **999** as the allele length**.

In diploid data, the first two allele columns represent the first locus, the third and fourth columns the second locus and so on.

**WARNING ON MISSING DATA

Missing loci are discarded when computing distances and estimating the average upon the number of scored loci. Depending on the variability of the missing loci, missing data can therefore induce a higher variance of distance computed and may result in an artificially higher or lower distance among the individual with missing data and the others. We tested empirically the effect of missing loci and their prevalence consistently bias the individual-level network toward a higher density of links for samples with many missing data. We would therefore advise to discard samples with more than 5 to 10% missing loci and to seriously check the pattern of variability in the missing loci if included, to avoid misleading results, particularly at the individual level.

You can alternatively give the genotypes without the first and second column. In this case EDENetworks automatically detects that the file contains only numeric columns and asks for an additional file containing the first and the second column.

See the Figure below for an example with identifiers for locations and individual samples in the two first columns, followed by the genotypes with alleles in distinct columns for each locus.

```
Acqua_Azurra_3 s1 151 161 164 164 210 210 234 238 159 171 178 178 178 180
Acqua_Azurra_3 s2 147 151 182 182 210 220 234 252 163 163 170 170 178 180
Acqua_Azurra_3 s3 161 161 164 164 208 216 234 234 163 163 168 170 178 178
Acqua_Azurra_3 s4 151 161 164 164 210 210 234 238 159 171 170 170 178 178
Acqua_Azurra_3 s5 151 151 164 164 208 214 226 234 159 163 168 170 178 178
Acqua_Azurra_3 s6 151 151 164 182 208 218 234 236 163 167 168 172 178 178
Acqua_Azurra_3 s7 151 151 164 164 208 214 226 234 163 167 168 172 178 180
Acqua_Azurra_3 s8 147 151 182 182 210 220 234 252 159 165 168 168 178 178
Acqua_Azurra_3 s9 151 161 182 182 210 220 234 234 159 165 168 168 178 180
```

2.1.2 Presence-absence and abundance data

Presence-absence/abundance data can be given in same format as genotype matrices by replacing alleles by the presence-absence or abundance values. The presences and absences are presented as zeros and ones and abundances with integer values. Missing data is currently not supported by any distance/dissimilarity measures implemented in EDENetworks.

2.1.3 Distance/dissimilarity matrix

If the user wishes to work with pre-computed distance (or dissimilarity) matrices instead of individual genotypes, a text file containing the distance matrix is required. This file should only contain elements of the distance matrix, separated by white spaces and row changes. The distance matrix can be in square format or in upper or lower triangular format, where elements below or above the diagonal are not present. The program tries to automatically infer the form of the matrix, but can ask if the diagonal elements are present in the matrix or not for the triangular formats. The distance matrix can represent any level of distance among the entity the user wishes to analyze as agents/nodes (community, population, individuals...).

```
distancematrix.txt.mat | node_names_for_distance_matrix.txt | node_names_for_Fst_matrix.txt | PosiFst.txt
0 0.59.0894642857.72.4459459459.167.832678571.105.893035714.117.003214286.64.1591575092.88.8296130952.93.13684210
59.0894642857.0.0.76.8977799228.178.836785714.112.499821429.124.984107143.68.3721611722.95.8699404762.93.5.61.625
72.4459459459.76.8977799228.0.0.262.026737452.166.181949807.184.779633205.44.0074745075.64.964527027.57.094594594
167.832678571.178.836785714.262.026737452.0.0.91.8316071429.73.2551785714.187.895054945.228.092410714.224.9086466
105.893035714.112.499821429.166.181949807.91.8316071429.0.0.53.3244642857.127.585347985.164.654613095.162.7642857
117.003214286.124.984107143.184.779633205.73.2551785714.53.3244642857.0.0.137.986263736.176.019494048.175.8030075
64.1591575092.68.3721611722.44.0074745075.187.895054945.127.585347985.137.986263736.0.0.62.8493589744.35.11567379
88.8296130952.95.8699404762.64.964527027.228.092410714.164.654613095.176.019494048.62.8493589744.0.0.80.446271929
93.1368421053.93.5.57.0945945946.224.908646617.162.764285714.175.803007519.35.1156737999.80.4462719298.0.0.69.849
55.9979262673.61.6251152074.49.1231162038.198.722580645.129.27764977.142.793778802.47.985820631.70.2740015361.69.
62.7663690476.69.8522321429.37.4113577864.218.199255952.140.164583333.153.580952381.43.4795482295.61.2837301587.6
65.2481428571.74.1447142857.42.0888030888.198.496.139.803571429.149.224142857.40.0271062271.57.7560714286.63.6230
53.1728937729.60.4228937729.35.7727452727.176.299084249.121.027838828.130.401465201.28.1743214051.50.9409340659.4
60.8192857143.70.4457142857.45.8814671815.174.162857143.124.235892857.130.587857143.36.7844322344.56.9245535714.6
53.9094642857.59.6903571429.38.5951737452.186.674107143.119.553928571.131.2375.34.5783882784.58.6660714286.52.051
69.8735714286.76.2655357143.45.3114864865.232.226964286.151.815892857.166.223035714.50.7965201465.70.7010416667.7
57.8502040816.63.0516326531.50.7507446222.194.735306122.119.951428571.133.015714286.49.4595499738.73.2914965986.7
52.1385714286.58.5858928571.25.1725868726.191.626607143.125.267857143.137.413928571.25.1815018315.45.4165178571.3
63.0444642857.69.5592857143.48.7895752896.166.732142857.124.968035714.132.071071429.30.1146520147.59.4296130952.4
65.8538461538.74.1446886447.40.8327393327.205.572710623.140.74029304.151.405677656.42.1491499953.59.150030525.64.
60.6563909774.67.6571428571.33.9386303597.198.772180451.136.434774436.147.123120301.28.7029111124.52.4088345865.44
```

The user can input a file with node names as a single column. This file should contain a single name at each row. If this input file is not given by the user, the elements of the matrix, i.e. the nodes of the resulting network, are labelled with integer numbers (1..N).

2.1.4 Network data

The user may also choose to work with pre-computed network data, either prepared outside EDENetworks analyzer, or calculated from genetic distance data with EDENetworks and saved in its network window. In this case, there are two options for the type of the **first input file**:

- Graph Markup Language (GML, *.gml) – see <http://www.infosun.fim.uni-passau.de/Graphlet/GML/> for a full description
- Edge files (*.edg) which list all edges in the network.

For the latter, the format is as follows: every row in the text input file lists one edge with three columns: first vertex, second vertex, and distance value for their edge:

```
vertex1 vertex2 distance12
```

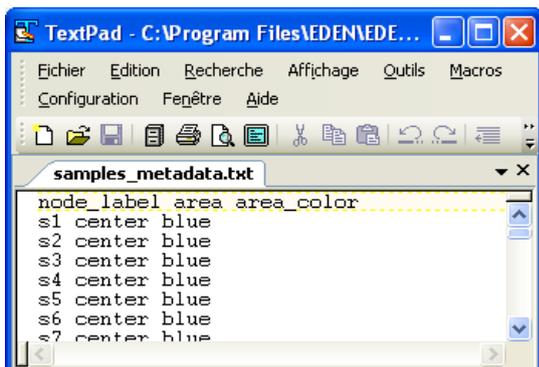
2.1.5 Auxiliary data for nodes (samples or sampling sites)

In addition, the user has the choice of adding auxiliary information for each node. The format of the auxiliary data file is the same for nodes that represent individual samples and for nodes which represent sampling sites. Furthermore, the node auxiliary data format does not depend on whether the genetic information is given in a genotype matrix, distance matrix, or as network file.

The node auxiliary data may contain numerical values, color codes or string data that can be used when visualizing the networks. Similarly to the other file formats, one row in the data file corresponds to one node. However, an additional header row is required. Each column corresponds to one auxiliary variable; however, the first column must contain identifiers of nodes labelled with the header 'node_label'.

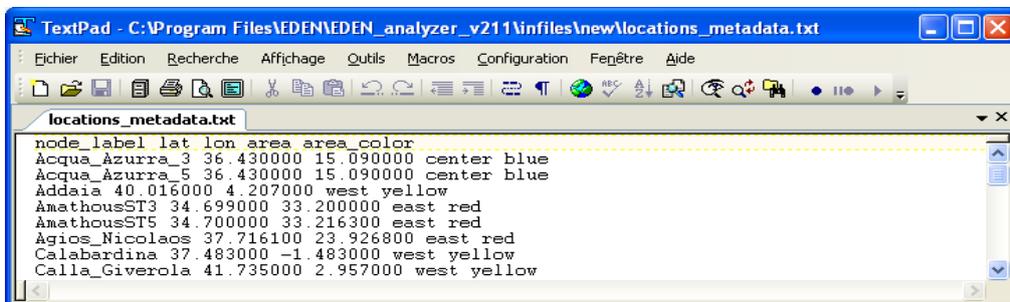
The auxiliary data file is constructed as follows:

- The first row contains the header
- The header contains labels for the column
- The first column contains node names with the header node_label
- The rest are user-specified auxiliary variables
- The rest of the rows should begin with sample identifier and then contain values/labels/colors for each auxiliary variable.



```
node_label area area_color
s1 center blue
s2 center blue
s3 center blue
s4 center blue
s5 center blue
s6 center blue
s7 center blue
```

The above figure contains an example auxiliary input file for nodes as samples. The samples are now enumerated and named with formula s[sample number].



```
node_label lat lon area area_color
Acqua_Azurra_3 36.430000 15.090000 center blue
Acqua_Azurra_5 36.430000 15.090000 center blue
Addaia 40.016000 4.207000 west yellow
AmathousST3 34.699000 33.200000 east red
AmathousST5 34.700000 33.216300 east red
Agios_Nicolaos 37.716100 23.926800 east red
Calabardina 37.483000 -1.483000 west yellow
Calla_Giverola 41.735000 2.957000 west yellow
```

The above figure contains an example auxiliary input file for nodes as sampling sites, then the latitude and longitude of nodes, the geographic group sampling sites belong to, and the colour chosen by the user to be used in visualization.

The type of data that is stored in each column of the auxiliary data file is recognized automatically; white space is used as the separator. The possible types of fields are integers (int), floating point numbers (float), color codes (string/color) and strings (string). Note that if, for example, a column contains both integers and strings, the whole column would be recognized as being a mix of different types (mixed).

Any names for the auxiliary data variables except *node_label* may be used (see Table here below). However, there are few reserved label names: Variable names *x* and *y* can be used as layout coordinates in visualization. Also, when importing a genotype matrix for individual-based analysis, *location* is automatically derived from the sampling site column. When a genotype matrix for sampling-site-based analysis is imported, *population_size*, *genotypes* and *clonal_diversity* are automatically calculated and added as auxiliary data.

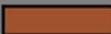
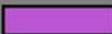
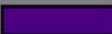
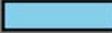
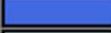
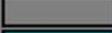
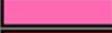
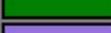
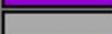
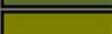
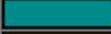
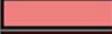
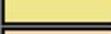
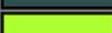
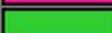
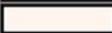
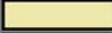
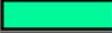
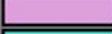
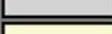
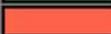
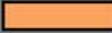
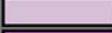
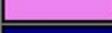
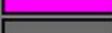
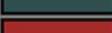
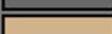
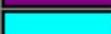
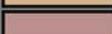
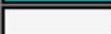
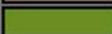
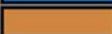
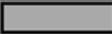
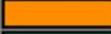
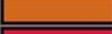
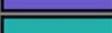
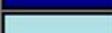
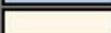
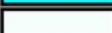
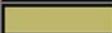
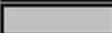
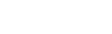
Node colors can be represented in following formats:

1. letters from the set 'rgbcmykw'
2. hex color strings, like '#00FFFF'
3. standard names, see the table on the next page
4. floats, like '0.4', corresponding to levels of gray on a 0-1 scale

2.1.6 Spatial coordinates for drawing the network

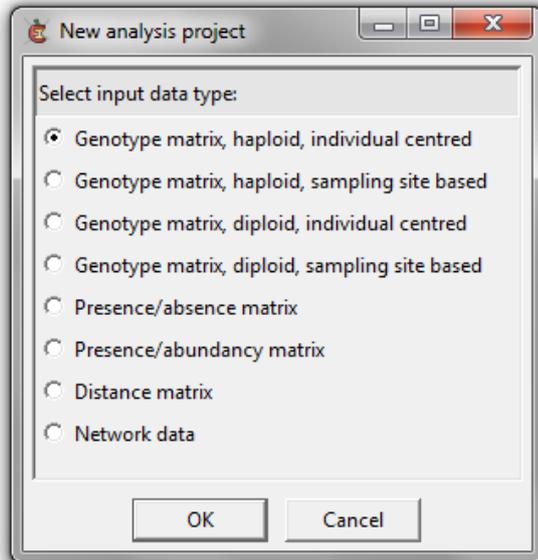
Once the window "View network" is opened, click on Options → Layout → Load coordinates file. This allows loading a file with pre-calculated (x,y)- coordinates of all nodes. These pre-calculated coordinates can be used to visualize the network e.g. so that the layout corresponds to some map area (and projection) chosen by the user.

Table of Colors

	sienna		mediumorchid		indigo
	mediumblue		skyblue		gold
	royalblue		gray		hotpink
	lightcyan		darkturquoise		firebrick
	green		goldenrod		indianred
	mediumpurple		darkgreen		yellow
	midnightblue		floralwhite		mistyrose
	cornsilk		darkviolet		darkolivegreen
	red		darkgray		olive
	bisque		moccasin		darkseagreen
	slategray		saddlebrown		pink
	darkcyan		grey		lightcoral
	khaki		darkslateblue		orangered
	wheat		lightskyblue		navajowhite
	teal		lightpink		lime
	darkorchid		mediumvioletred		palegreen
	deepskyblue		slategrey		darkslategrey
	salmon		deeppink		greenyellow
	darkred		limegreen		burlywood
	steelblue		darkmagenta		seashell
	palevioletred		palegoldenrod		mediumspringgreen
	lightslategray		plum		fuchsia
	aliceblue		turquoise		papayawhip
	lightslategray		lightgrey		blanchedalmon
	lightgreen		lightgoldenrodyellow		chartreuse
	orchid		darkgoldenrod		dimgray
	gainsboro		lavender		black
	mediumseagreen		maroon		peachpuff
	tomato		yellowgreen		springgreen
	mediumturquoise		sandybrown		aquamarine
	lemonchiffon		thistle		white
	cadetblue		violet		orange
	lightyellow		navy		lightsalmon
	lavenderblush		magenta		darkslategrey
	coral		dimgray		brown
	purple		tan		ivory
	aqua		rosybrown		dodgerblue
	whitesmoke		olivedrab		peru
	mediumslateblue		blue		darkgrey
	darkorange		lightblue		lawngreen
	mediumaquamarine		ghostwhite		chocolate
	darksalmon		honeydew		crimson
	beige		cornflowerblue		forestgreen
	blueviolet		linen		slateblue
	azure		darkblue		lightseagreen
	lightsteelblue		powderblue		cyan
	oldlace		seagreen		mintcream
			darkkhaki		silver
			snow		antiquewhite

2.2 Interface

2.2.1 Begin a new analysis project



Genotype matrix

For analysis based on genotype matrix (see section 2.1.1), to build either individual-centred (nodes are individuals) or population-based (nodes correspond to sampling sites) networks.

Presence/absence/abundance matrix

A matrix of present features or counts for samples/populations. See section 2.1.2.

Distance matrix

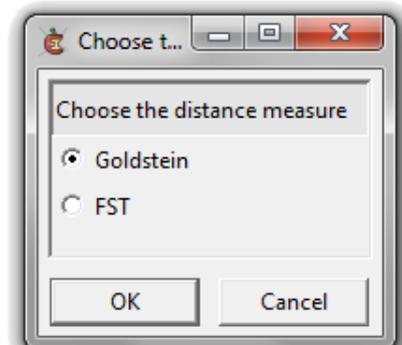
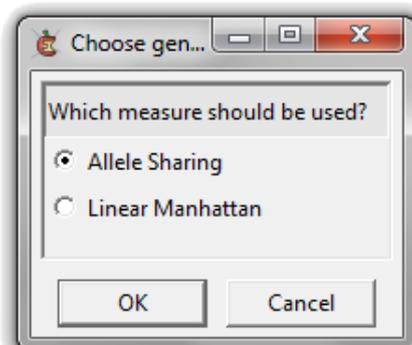
Allows entering an input file with a pre-computed pairwise distance matrix with any chosen distance (ecological, genetic, or any other pairwise distance) and at any level (individual, population, or location centred).

Network data

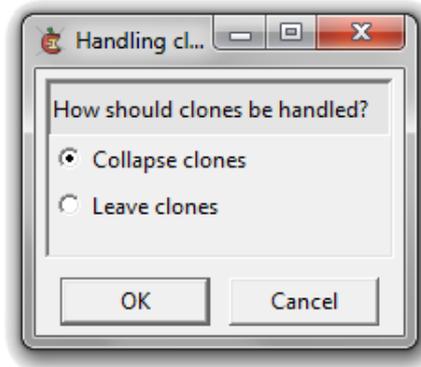
Allows entering an input file with a pre-computed network generated with EDENetworks or any other software (individual, population, or location centred).

2.2.2 Genotype matrix : Individual-centred or location-based

First choose the distance measure to be used. The choice of distance measure depends on the type of data (individual-centred or sampling-site-based) :

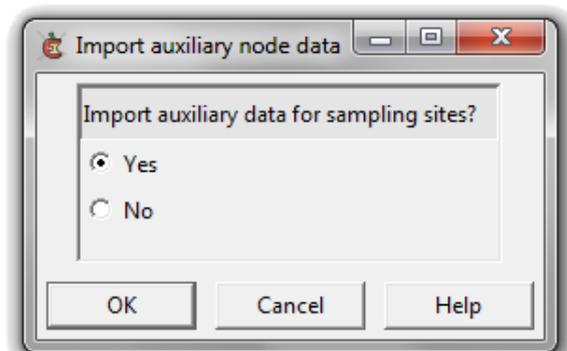
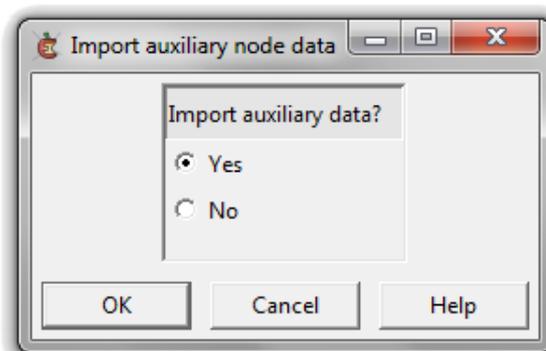


When opening a genotype matrix for individual-centred analysis, the program will in addition ask how possible clones should be handled:



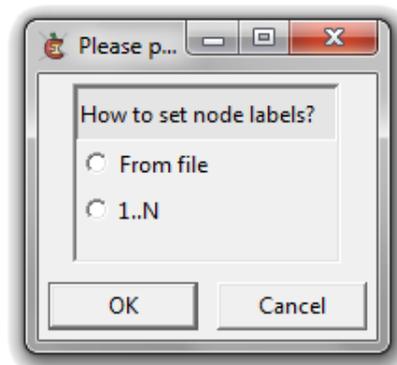
This option is only meaningful if you study a clonal organisms. For non-clonal organisms, choose "leave clones". For clonal organisms, it is recommended that you ascertain clonal membership (Arnaud-Haond et al., 2007; Arnaud-Haond & Belkhir, 2007) before choosing to collapse clones – when clones are collapsed, a single node will represent all samples with exactly identical multi-locus genotypes. Note that if there are no identical genotypes, collapsing clones will have no effect on the data.

Finally, for both individual-centred or location-based analysis, the program will propose the option to import auxiliary data (see 2.1.5):



2.2.3 Distance matrix

When opening a distance matrix the user must first input the file containing the elements of the matrix (see section 2.1.3). After that the program will ask how the



nodes should be labelled. There are two choices: labelling the nodes with numbers starting from 1 or giving an extra input file which contains the node labels:

If the input file is a triangular format matrix and the node labels are not given as a separate file, the program asks if the diagonal elements are present in the input matrix.

Finally, the user can import node auxiliary data (see section 2.1.5).

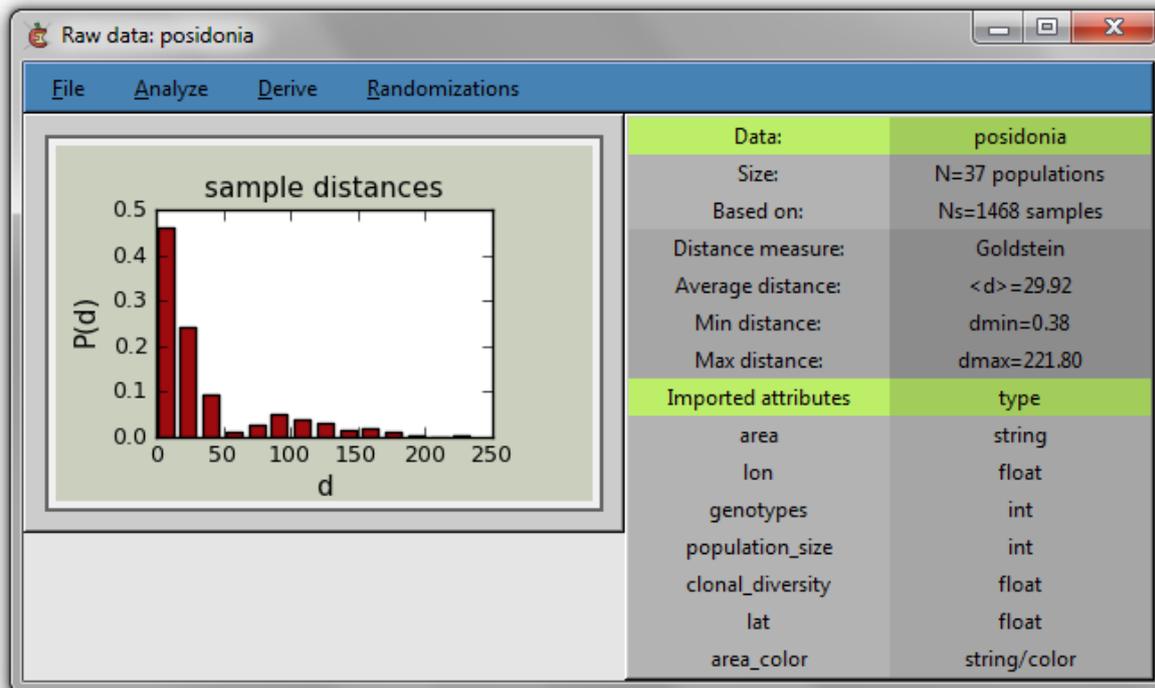
2.2.4 Network data

The user may want to input the data directly in the network format. In this case the analysis starts directly from the network data window (see 2.4.4 and 2.4.5). The data must be given in .edg or in GML format (see 2.1.4). Auxiliary node data can also be included as a separate file (see 2.1.5).

2.3 *Data analysis*

2.3.1 Distance matrix

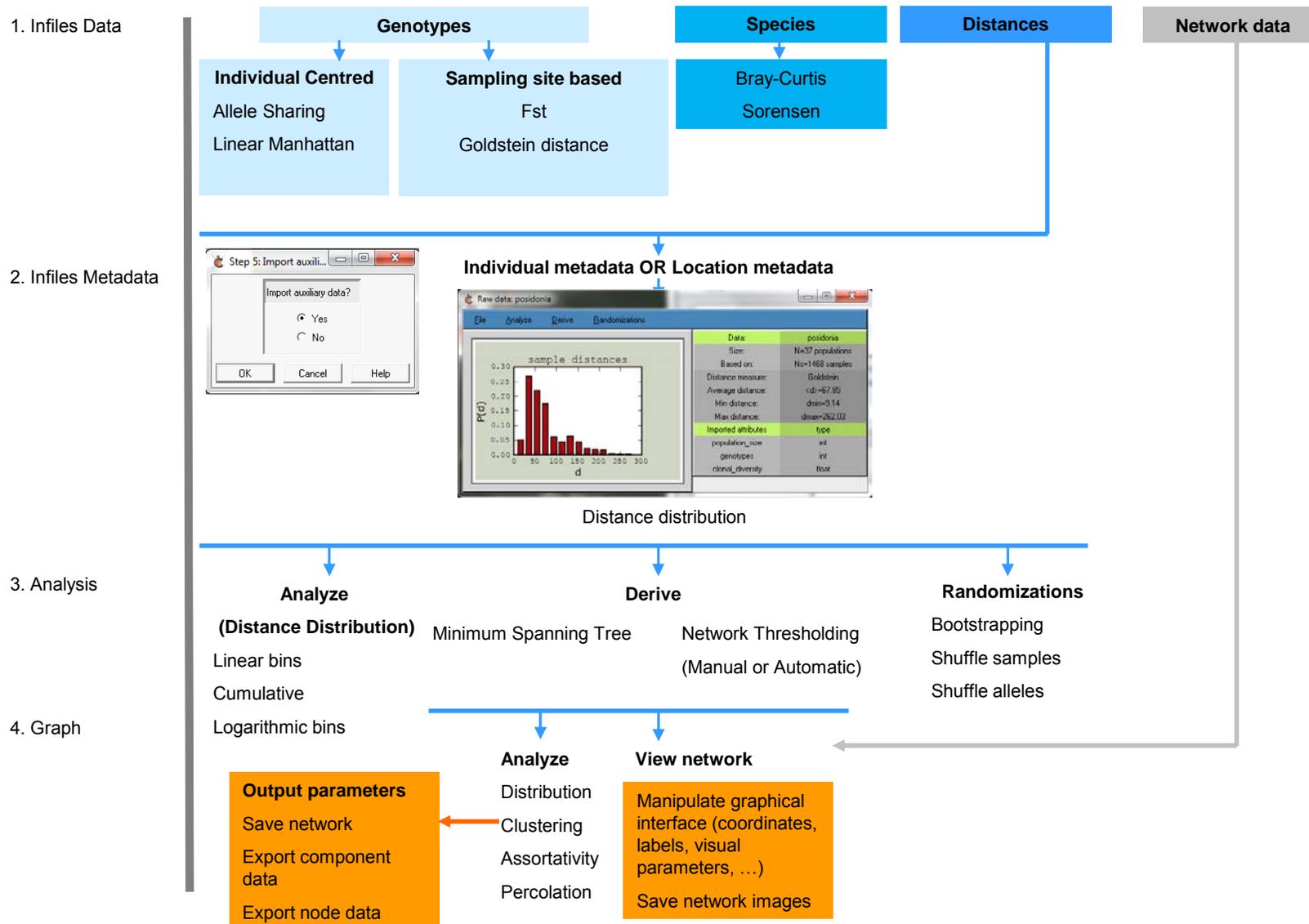
After inputting genotype, presence-absence/abundance or distance matrix data, you will have a distance/dissimilarity matrix between the nodes which represent either sampling sites or sampling locations. The analysis pipeline for both types of nodes is almost identical, with the exception of randomizations (see 2.4.6), which are only available for sampling sites.



The distance data is presented to the user as a window where some basic statistics and other information about the data set is shown (see the figure above). From this window, the user can

- save the distance matrix (File -> Save distance matrix)
- analyze the distance distribution using linear or logarithmic axes, or presented as a cumulative distribution. From the distance distribution window, one can save the plot or the distribution as numbers (File -> Save graph, File->Export data).
- derive network data (see section 2.4.3)
- randomize the data ((option only available for sampling-site analysis; see section 2.4.6)

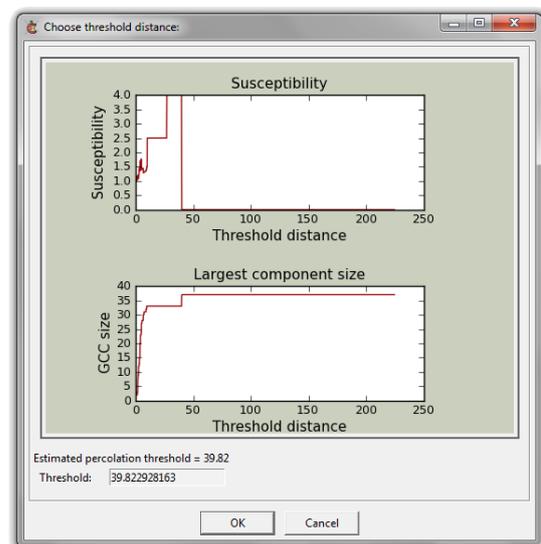
2.3.2 Schematic overview of data input and analysis



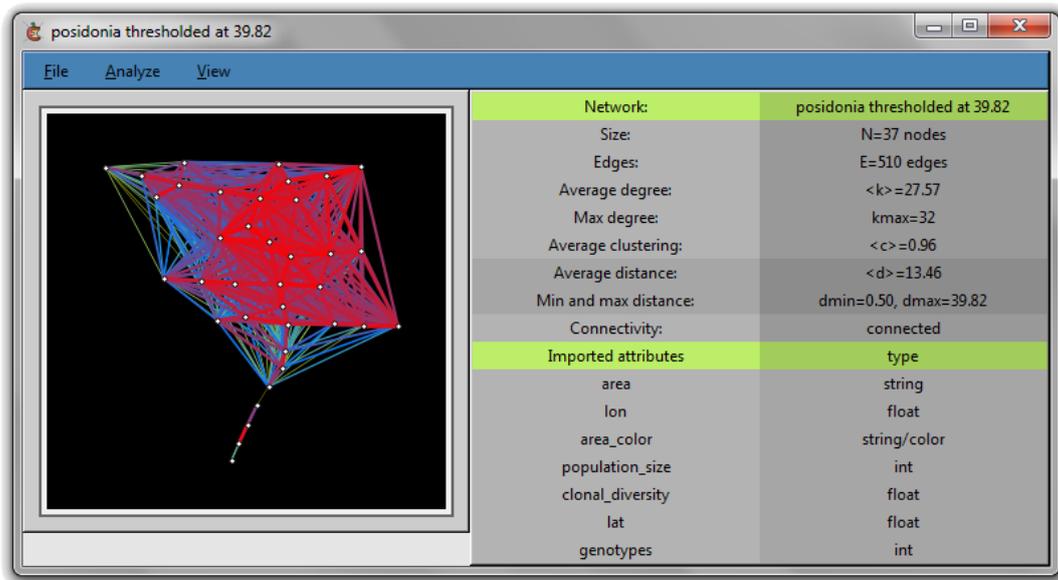
2.3.3 Deriving, Analyzing and Visualizing Networks

From the distance matrix window, choosing Menu -> Derive allows deriving two kinds of networks:

- A **Minimum Spanning Tree**: Given a connected, undirected graph, a spanning tree of that graph is a subgraph without cycles that connects all vertices. Provided each edge is labeled with a cost (in the analysis in EDENetworks the chosen distances among the connected nodes) each spanning tree can be characterized by the sum of the cost of its edges. A minimum spanning tree is then a spanning tree with minimal total cost: the minimum-cost subgraph connecting all vertices, since subgraphs containing cycles necessarily have more total cost.
- A **Thresholded Network**: the threshold is the maximum distance considered as a link in the network. All links corresponding to distances above that threshold are removed. There are two options:
 - In **Automatic thresholding**, the percolation threshold of the network will be automatically detected. The threshold value used by the software is set slightly below the percolation threshold (see manual thresholding), such that the network still remains connected.
 - In **Manual Thresholding**, results of the percolation analysis are shown and the user is free to choose the threshold manually either based on this analysis or other considerations. Two graphs are displayed: the relative size of the largest connected component, which should shrink for thresholds below the percolation point, and the so-called susceptibility, which should display a peak at the percolation point. The software automatically suggests a threshold based on this peak; note, however, that these graphs are accurate only for very large networks and may thus be noisy or hard to interpret. We recommend the user to experiment using different thresholds. This additionally allows analyzing sub-structured systems at different, decreasing structure scales.

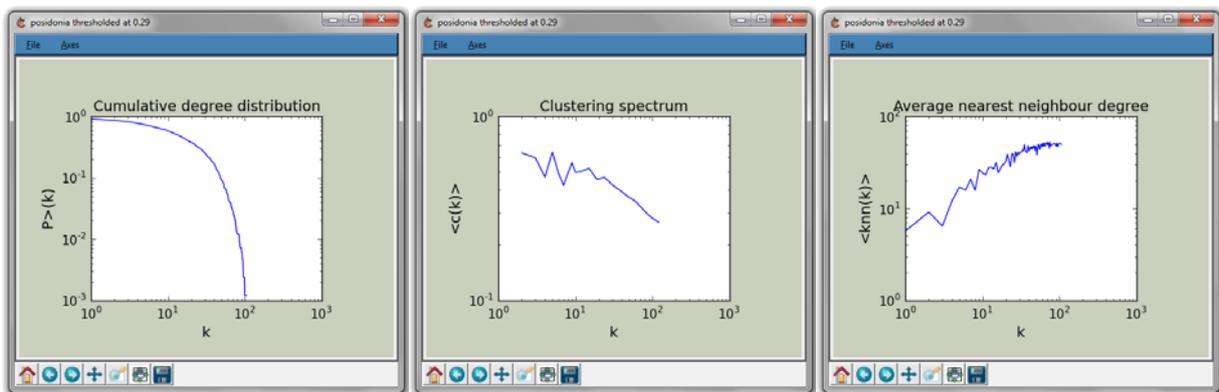


The network derived using either of these options is then displayed in the Network view, which also displays the main properties of the derived network. Within this window, the user can choose to analyze the properties of the network, or launch a visualization window with an enlarged view on the network and access to graphical options for defining the colours and legends of nodes and edges and manipulating the network visual picture.



2.3.4 Analyzing network structure

The Analyze menu can be used to produce several statistical distributions and graphs. For the majority, the user can select between different data binning methods and axis choices. From the resulting graph window, the results can be exported either as numbers in a text file or as a figure file (File->Save graph, File->Export data).



Menu item	Submenu item	Description
Distribution	Degree distribution	Probability density distribution of node degrees
	Distance distribution	Probability density distribution of distances associated with edges
	Shortest path	Probability density distribution of the shortest paths in the network
Clustering		Clustering coefficients of nodes averaged over each degree
Assortativity	Average neighbour degree	Average neighbour degrees of nodes averaged over each degree

algorithms, there is a stochastic component in the algorithm and hence the results will vary even when visualizing the same network. The user may choose to re-visualize the network from the Options->Layout menu by selecting Calculate new layout. Furthermore, if the user wishes to use own coordinates for visualization and these have been imported within the auxiliary file input within the project launch wizard, these may be selected for visualization here.

The network is also interactive, which means the user can arrange nodes in space as wished for a better representations, and their edges will automatically follow. Once arranged manually, the coordinates of nodes can be saved, in order to re-apply the exact same scheme to network at different thresholds or network with the same agents based on different distances.

2.3.6 Randomizations

In order to assess the significance of results obtained with the software and the effects of noise or incomplete statistics, several randomization procedures have been implemented to generate networks or statistics corresponding to a selection of null hypotheses for population-based networks. The randomization procedures can be accessed in the Raw data view, displaying the distance distribution.

For randomizations, there are two main categories: either to generate single networks, for visual and numerical inspection, or to generate statistics over a larger number of networks obtained with the randomization procedures.

For single networks, the options are to:

- Bootstrap, or resample, the samples in each population such that for each population only a random subset of the original samples is kept. Note that this sampling is done without replacement and the resampled populations are smaller than the original ones.
- Shuffle samples. Each individual sample is randomly assigned to a new population such that the total number of samples in each population is kept unchanged. This removes all spatial structure in genetic distances.
- Shuffle alleles. For each locus, alleles in all of the samples are shuffled.

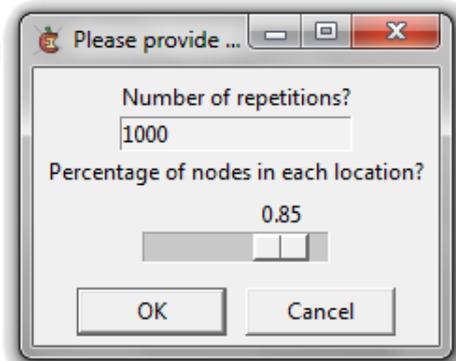
These options can also be used for statistical analysis as discussed next. Note that when the data input is a distance matrix or a network, randomization procedures are disabled.

2.3.6.1 Statistical analysis for betweenness centrality

The user can analyze the significance and robustness of the betweenness centrality values of the populations with the bootstrap procedure described above. The resampled data are then turned into full network using the chosen distance measure. These networks are automatically thresholded, and betweenness centralities for all nodes are calculated. The BC value distributions are shown for nodes which have, in total, the highest betweenness centralities considering all generated networks. Also, histograms displaying how many times nodes appeared in top 5 and top 1 lists of BC values are shown. If a node appears several times in the top 5 and top 1 lists, its

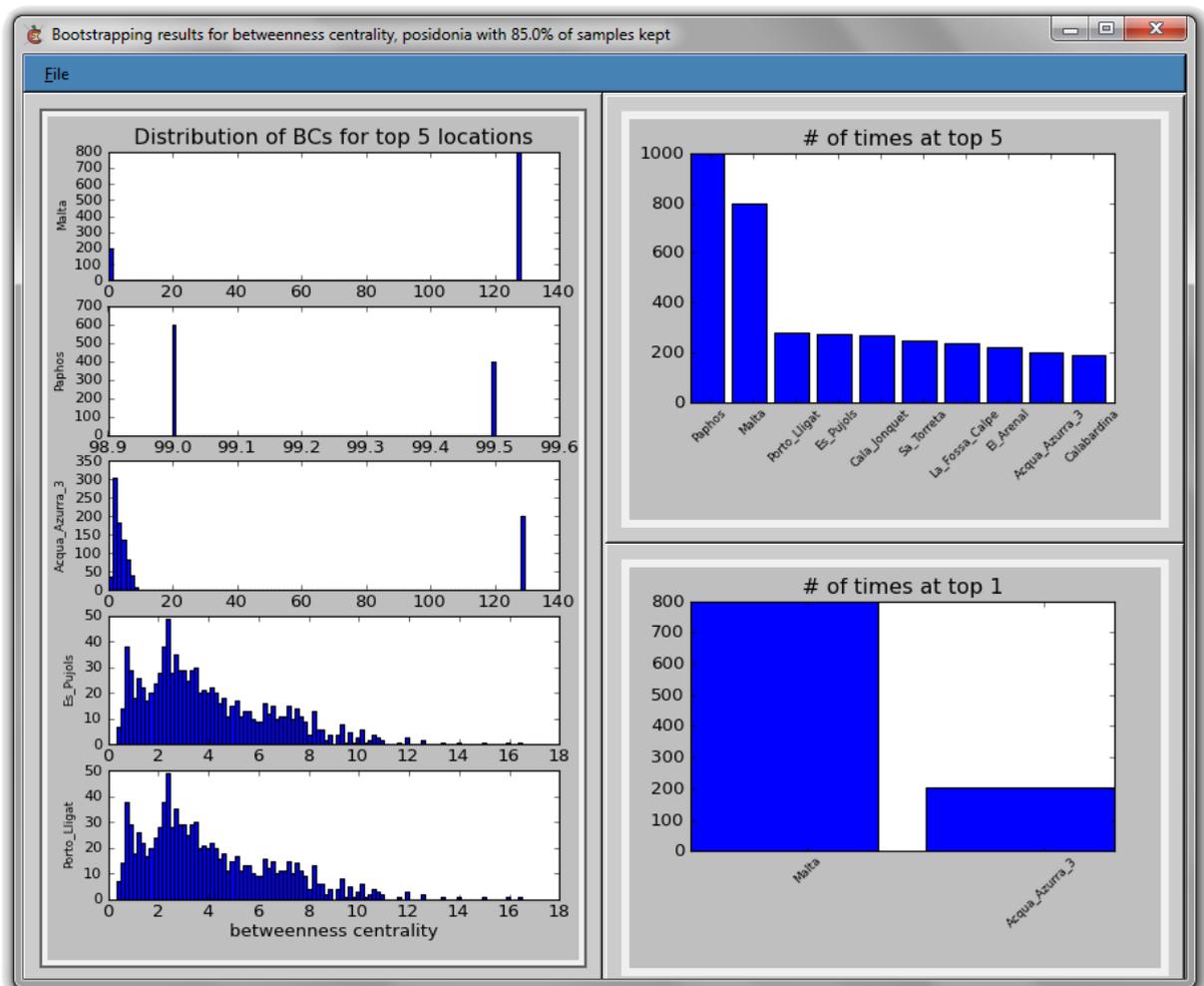
betweenness centrality can be viewed as statistically significant and insensitive to the bootstrapping procedure.

Start by selecting



Randomizations->Statistics->Betweenness:Bootstrapping in the distance data window. You are then asked to provide the number of times you want the bootstrapping process to be repeated and for the percentage of samples that you want to be randomly selected at each round from each population. These randomly resampled data are then transformed into automatically thresholded networks which are then used for calculating the betweenness centralities.

After the calculations are finished, the results are presented in a figure as shown below.

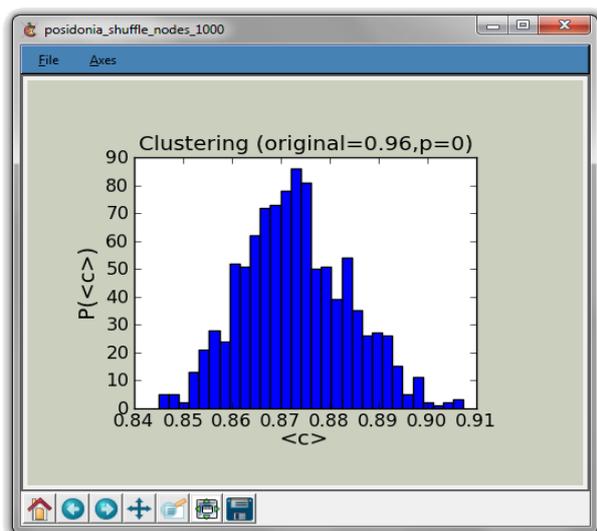


Comparing this figure to the original data you are able to see if the nodes with high betweenness centrality values also had high values of betweenness centrality in the random realizations produced by the bootstrapping process. For example, in the below figure, the two nodes *Aqua_Azurra_3* and *Paphos* have high betweenness centrality values in all the randomization rounds. The betweenness centrality values of other nodes are much lower than for the two top nodes. However, there is some large variance in the betweenness centrality values of the other nodes, and *Aqua_Azurra_5* has even replaced *Aqua_Azurra_3* in one out of thousand realizations.

2.3.6.2 Statistical analysis for clustering coefficient

The statistical significance of the global clustering coefficient at a chosen threshold can be calculated for either sample or allele shuffling (see above) null models. These randomizations are performed for any number of times requested by the user, and statistics of the clustering coefficient across these runs are displayed and compared to the original clustering coefficient at the given threshold. For the shuffled networks, instead of using an automated percolation threshold (as with the statistics for betweenness centrality), the same number of lowest-distance links as in the original network at the chosen threshold is used and the rest of the links are discarded. This strategy is chosen since the value of the clustering coefficient can be expected to have strong positive correlation with the link density.

Start by selecting Randomizations->Statistics->Clustering:Shuffle samples/alleles in the distance data window. You are then asked to give the threshold in the original network that you want to study. The default value is the same as produced by the automatic thresholding algorithm. After the calculations are done, the histogram of the average clustering coefficient values is shown (see figure below).



In the figure title, the average clustering coefficient of the original network with the corresponding threshold value is given, and the probability of getting a value as large or larger in the randomizations (estimate for the p -value). For the example figure, we could deduce that the clustering coefficient with the chosen threshold level would be very high even for random assignment of samples to sampling locations, but still smaller than in the observed data with a high probability.

2.3.7 Output files

The following files can be generated with EDENetworks using its various menus.

Item	Window/Menu	File/Save	Description	Format
Dist. Distr.	Raw data/Analyze	Save graph	Figure file	multiple
		Export data	Distribution as numbers	.txt
MST or Thr. Network	Network/File	Save Network	Network in a file	.edg .gml
		Export component data	Describes each component (size, identity and properties of nodes forming it)	.txt .asc
		Export node data	List of nodes together with their properties (degree, betweenness, clustering, user- imported auxiliary data)	.txt .asc
	Network/ Analyze	Save graph	Figure file for any network analysis graph	multiple
		Export data	Data for any network analysis results	.txt
	Visualized network/File	Save figure	Network visualization in figure file	multiple
		Display coordinates	File with (x,y) coordinates of the visualization	.txt

3 Methodological outline: Details and references for the calculations

3.1 Ecological distances

Presence-absence and presence-abundance data

EDENetworks uses Bray-Curtis (1957) dissimilarity for the presence-absence and presence-abundance data on taxa:

$$D(A,B) = \frac{2 \sum_{i=1}^k \min(A_i, B_i)}{\sum_{i=1}^k (A_i + B_i)},$$

where A and B are the two nodes represented with feature vector of length k , where each i th element contains the count of the feature (or binary value in the case of presence-absence data).

It should be noted that in case of presence/absence data this distance is identical to Sorensen or Dice coefficient (Somerfield, 2008)

3.2 Genetic distances implemented

3.2.1 Individual-centred

These are measures of genetic similarity between individuals represented by genotypes in the genotype matrix. For two particular individuals, A and B , genotyped for k locus in a diploid organism, are represented as:

$$A = (a_1, A_1)(a_2, A_2) \dots (a_k, A_k), \text{ and } B = (b_1, B_1)(b_2, B_2) \dots (b_k, B_k),$$

where a_i and A_i , and are the allele length b_i and B_i (in number of nucleotides) in both chromosomes at locus i for individuals A and B , respectively.

Allele Shared Distance

Allele Shared distance (ASD). This genetic distance is based on the proportion of shared alleles (Chakraborty & Jin 1993; Bowcock, 1994). For individual pairwise comparisons the proportion of shared alleles is estimated by :

$$P_{SA_i} = \frac{\sum uS}{2u}$$

where the number of shared alleles S is summed over all loci u . The distance between individuals is then

$$D_{SA_i} = 1 - P_{SA_i}$$

Manhattan distance

The dissimilarity degree between A and B at locus i is:

$$d_i(A,B) = \min(|A_i - B_i|, |a_i - b_i|, |A_i - b_i|, |a_i - B_i|),$$

providing a parsimonious (i.e., minimal) representation of the genetic distance, in allele length, between samples A and B . The *genetic distance* among individuals is obtained by averaging the contributions from all loci:

$$D(A,B) = \frac{1}{k} \sum_{i=1}^k d_i,$$

which provides the degree of global dissimilarity between A and B .

Missing data

The loci with missing data are disregarded in the pairwise comparisons. This means that the average value of the distances between loci are only taken over those loci which do not have any missing data.

3.2.2 Presence-absence data

EDENetworks can also be used to compute Bray-Curtis dissimilarity (1957) for molecular data based on presence-absence:

$$D(A,B) = \frac{2 \sum_{i=1}^k \min(A_i, B_i)}{\sum_{i=1}^k (A_i + B_i)},$$

where A and B are the two nodes represented with feature vector of length k , where each i th element contains the binary value indicating the presence or absence of bands/alleles.

3.2.3 Population-centred

Goldstein distance

The Goldstein distance (Goldstein, et al., 1995) is implemented in EDENetworks; it is defined only for microsatellites and relies on the assumption of a strict –or almost strict – Stepwise Mutation Model (SMM). It is defined as:

$$(\delta\mu)^2 = (m_x - m_y)^2$$

where m_x and m_y are the means of allele sizes in population x and y respectively. Averages are always averages of non-missing alleles.

FST based distance

The FST based Reynolds Distance D (1983) is estimated, $D = -\ln(1 - \theta_w)$, using the weighted estimate of the pairwise coancestry coefficient:

$$\theta_w = \frac{\sum_{i=1}^k a_i}{\sum_{i=1}^k a_i + b_i},$$

where a_i and b_i are the estimates of variances of interest between populations and within populations for locus i .

3.3 Network descriptors

3.3.1 Paths and components

There is a path between two nodes i and j in the network if there is a sequence of edges which can be followed to get from i to j . A component in a network is a maximal set of nodes that are all joined by paths, where maximality means that one cannot add any further nodes to the set such that they would still all have paths between each other. Any undirected network can be uniquely divided into components such that each node belongs to a single component. If there is only one component in a network, i.e. all its node pairs are joined by paths, it is said to be connected.

3.3.2 Minimum Spanning Tree

Given a connected, undirected graph, a spanning tree of that graph is a subgraph without cycles which connects all the vertices together. Provided each edge is labelled with a cost (in the analysis in EDENetworks the chosen distances among the connected nodes) each spanning tree can be characterized by the sum of the cost of its edges. A minimum spanning tree is then a spanning tree with minimal total cost: the minimum-cost subgraph connecting all vertices, since subgraphs containing cycles necessarily have more total cost.

3.3.3 Thresholded Network

The threshold is the maximum distance considered as generating a link in the network. All links corresponding to distances beyond that threshold are removed. The network can be analyzed at various meaningful thresholds, particularly when hierarchical substructures occur through the network and different clusters are revealed at different thresholds.

One meaningful distance is the one corresponding to the percolation threshold, above which there is a giant component containing almost all the nodes in the networks and below which the network is fragmented into small disconnected components and the system therefore loses its ability to transport resources/information/genes/species across the whole system.

3.3.4 Percolation

The precise location of the percolation point is searched using the definition classically proposed for finite systems (Stauffer and Aharony, 1994) by calculating the average size of the clusters excluding the largest one:

$$\langle S \rangle = \sum_{s < S_{max}} s^2 n_s,$$

as a function of the last distance value removed, thr , and identifying the critical distance with the one at which $\langle S \rangle^*$ has a maximum. N is the total number of nodes not included in the largest cluster and n_s is the number of clusters containing s nodes. Note that for a particular network a well-defined threshold value may not exist at all.

3.3.5 Clustering

The clustering coefficient of the whole network $\langle C \rangle$ is defined as the average of the clustering coefficients of all nodes in the network. The degree-dependent clustering spectrum $C(k)$ is obtained after averaging C_i for nodes with degree k (see in section 3.4.2).

3.3.6 Assortativity

Assortativity is usually studied by determining the properties of the average degree $\langle k_{nn} \rangle$ of neighbors of a node as a function of its degree k (Lee, et al., 2006; Newman, 2002; Pastor-Satorras, et al., 2001). If this function is increasing, nodes of high degree connect, on average, to nodes of high degree and the network is assortative. Alternatively, if the function is decreasing nodes of high degree tend to connect to nodes of lower degree revealing that the network is disassortative.

3.4 Node descriptors

3.4.1 Degree

The degree k_i of a given node i is the number of other nodes linked to it (i.e., the number of neighbor nodes). The *distribution* $P(k)$ gives the fraction of nodes in the network having degree k .

3.4.2 Clustering

We denote by E_i the number of links existing among the neighbors of node i . This quantity takes values between 0 and $E_i^{(max)} = \frac{k_i(k_i - 1)}{2}$, which is the case of a fully connected neighborhood. The clustering coefficient C_i of node i is defined as:

$$C_i = \frac{2E_i}{k_i(k_i - 1)}$$

3.4.3 Shortest paths and diameter

The shortest path between two nodes, i and j , is the minimum number of edges that need to be traveled in order to get from i to j . The diameter of the network is the length of the longest of the shortest paths.

3.4.4 Betweenness centrality

The betweenness centrality of node i , $bc(i)$, counts the fraction of shortest paths between pairs of nodes which pass through node i . Let σ_{st} denote the number of shortest paths connecting nodes s and t and $\sigma_{st}(i)$ the number of those passing through the node i . Then,

$$bc(i) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}} .$$

The degree-dependent betweenness, $bc(k)$, is the average betweenness value of nodes having degree k .

4 Brief guidelines for interpretation of results, and some warnings

The use of network analysis to illustrate and understand historical and contemporary patterns of gene flow is a recent idea. So far, only few network-related methods have been developed, to the best of our knowledge. The method presented here is similar to “divisive hierarchical clustering”-like processes: we start scanning the network from the fully connected state and lower the threshold distance to observe the emergence of clusters at different steps. This process has been the basis of some articles dealing with ancient divergence among hydrothermal communities for ecological networks, and attempts to understand hybridization history in algae and gene flow at an intra-specific scale among populations of seagrasses. The use and interpretation of such tools derived from physics in ecology and evolution will likely increase with the help of the methods implemented in this software package and other similar methods. However, such methods are still relatively unknown for most ecologists and evolutionary biologists.

Here, we don't pretend to propose a ‘*how to*’ method, but rather some guidelines for interpreting results of network analysis in ecological terms. This brief set of guidelines aims at shedding light on the parallels that can be made between network properties and the usual population-genetics parameters we aim at studying. Here we chose to use as an example the analysis of a set of populations aiming at unravelling contemporary (*i.e.* present-day or recent) gene flow; however the results can be extrapolated to communities or individuals in a rather straightforward way by formulating each analysis procedure in terms of its own spatial and temporal scale & context. We hope these guidelines can be of help and will welcome any suggestion to improve them in the future.

4.1 ***Global analysis: Network topology and threshold choice***

Q1: How is gene flow distributed among my populations? How do I choose threshold(s)?

The method we propose here is based on the choice of a distance above which the connections among populations may be considered as negligible compared to the connections reflected by lower distances (*i.e.* increased similarity). The network at each threshold is illustrated and analyzed only considering connections at the distances below the threshold.

This threshold can be considered as somewhat arbitrary, yet setting the threshold value can be based on two rationales:

- 1)** A mathematical one based on the intrinsic properties of the dataset: the percolation threshold. As explained before, the fully connected network (*i.e.* including all distances) is a giant cluster, and the percolation threshold corresponds to the point below which the giant cluster dissolves, *i.e.* to the first sub-division of the system into sub-groups of populations. In order to explore the gene flow at the scale of the entire system, analyzing the network just above the percolation threshold is

usually a good starting point. At this stage, secondary cluster(s) emerge illustrating the existence of a sub-structure and a significant clustering (see Q2). However if most analyzed populations show similar pairwise distances, and therefore a lack of hierarchical clustering, the network will rather lose most connections successively within a very small distance interval. In this case, the percolation process would be similar to the process observed for random data, and significant clustering will likely not be detected at the population level but may rather exist, if the system is structured, at the individual level.

2) User-defined exploration, which consists of scanning the network across different thresholds. Besides checking for the stability of the pattern revealed at the percolation threshold, this scanning methodology may possibly reveal the emergence of clusters at different genetic distances, illustrating different levels of gene flow across different spatial/temporal scales, and possibly bio-geographical regions with variable times and levels of divergence. Starting the exploration from the entire set of populations and getting from global to regional and local scales at decreasing thresholds may successively illustrate the organization of populations in sub-clusters, provided that such sub-clusters exist. The threshold method is based on a relative ranking of distances among populations that depends on all elements of the data set. While sweeping the threshold provides a high-level view on structure at several scales, further analysis focusing on lower scale, *i.e.* identified sub-clusters, may be done preferably on the basis of a new network construction including only the members of these sub-clusters in order to estimate and interpret network parameters such as clustering, degree, betweenness centrality or assortativity for the set of populations focused on.

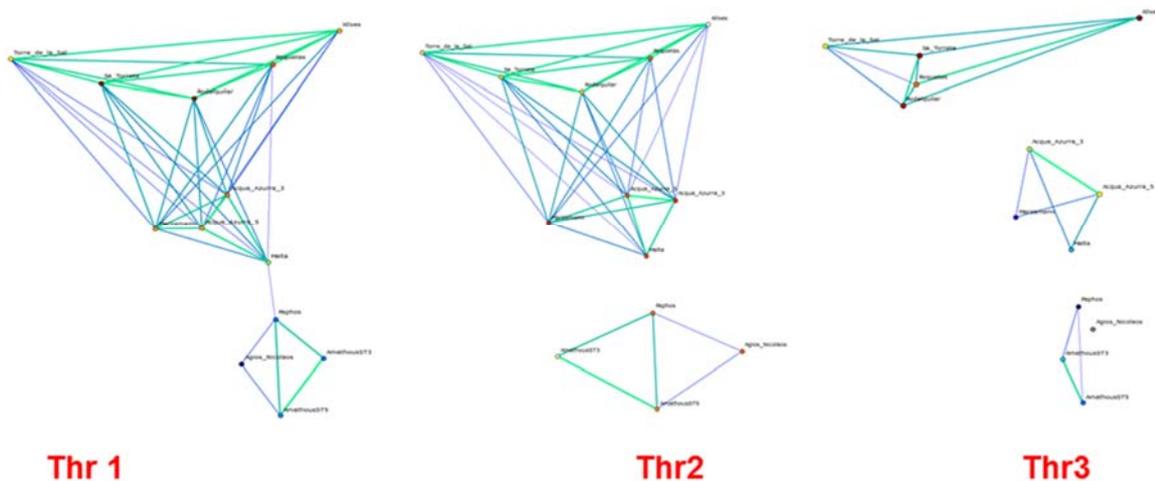


Fig. 4.1.a: Example of sub-structured system (subset of populations from Arnaud-Haond et al., 2007), in which the percolation threshold (a) illustrates the emergence of a set of populations from the Eastern Mediterranean, and the scanning at decreasing threshold shows the hierarchical subdivision of population from the Eastern, Central and Western Mediterranean.

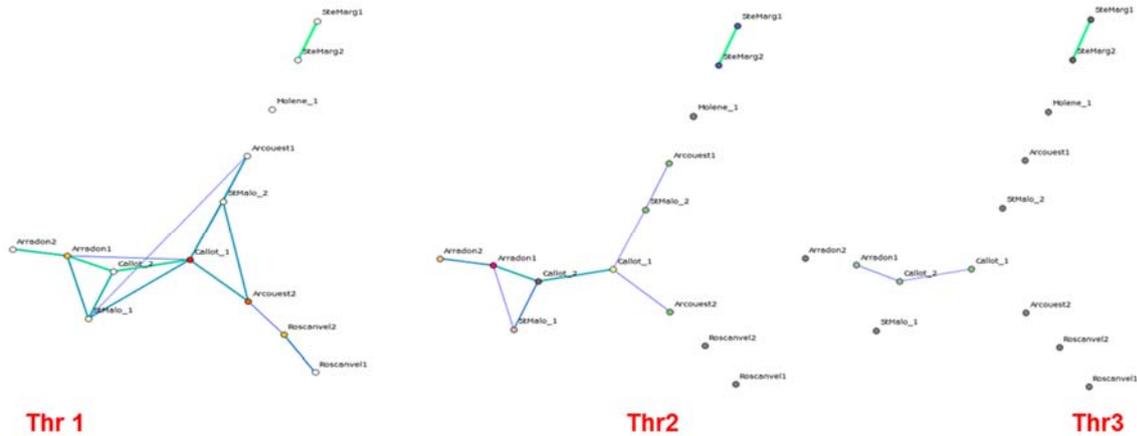


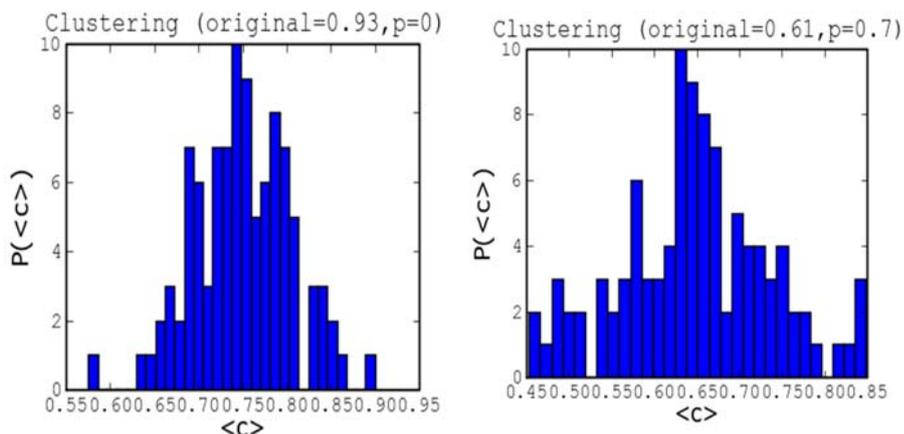
Fig 4.1.b: Example of a significantly differentiated, but (almost) non sub-structured system of *Zostera marina* meadows of Brittany (Becheler et al., 2010), one may observe that meadows are serially disconnected mostly one by one within a small range of distance, without the emergence of a secondary cluster.

4.2 Clustering

Q2: Is the system hierarchically structured?

The clustering coefficient of the network at each threshold should be compared with the ones obtained by using appropriate null models to see if there is more structure in the network than one would expect by pure chance. When the global clustering coefficient gets significantly higher values than the randomized networks, it indicates the existence of sub-structures in the data set that should be observable on the network graph.

As an example in the two figures showed here-below illustrating sub-structured and non sub-structured systems, the clustering coefficients are respectively $\langle C \rangle = 0.93$ ($p < 0.001$) and $\langle C \rangle = 0.61$ ($p = 0.70$):



Q3: Can I identify and describe the sub-clusters of populations?

When observing the network topology at each threshold chosen (view network), one can visualize the different components/clusters of populations and the nodes/populations constituting them, and by using the menu export the 'Component data' to obtain a description of the components. The component data indicates the number and nature of dissociated clusters, and lists the node (populations) clustered

in each of them with their properties (also found in 'Node data') in terms of betweenness centrality, clustering and degree, as well as number of samples and genotypes in the populations.

Table 4.2: Example of Table showing component data at the three successive thresholds used to illustrate the structure of the subset of *P. oceanica* populations illustrated here above.

Node	Degree	Clustering	betweenness	population_size	genotypes
Thr 1					
Component 0: Size = 13 nodes					
Agios_Nicolaos	2	1.00	0.00	40	28
Marzamemi	8	0.96	0.71	38	31
Acqua_Azurra_3	8	0.96	0.71	40	31
Malta	8	0.75	32.00	39	29
Acqua_Azurra_5	8	0.96	0.71	40	29
Roquetas	8	0.96	0.71	50	35
Paphos	4	0.33	27.50	38	26
Rodalquilar	8	0.96	0.71	50	27
AmathousST5	3	0.67	0.50	40	25
Torre_de_la_Sal	8	0.96	0.71	29	15
AmathousST3	2	1.00	0.00	40	18
Sa_Torreta	8	0.96	0.71	40	21
Xilxes	7	1.00	0.00	32	12
Thr 2					
Component 0: Size = 9 nodes					
Marzamemi	8	0.93	0.33	38	31
Acqua_Azurra_3	8	0.93	0.33	40	31
Malta	6	1.00	0.00	39	29
Acqua_Azurra_5	8	0.93	0.33	40	29
Roquetas	7	1.00	0.00	50	35
Rodalquilar	8	0.93	0.33	50	27
Torre_de_la_Sal	8	0.93	0.33	29	15
Sa_Torreta	8	0.93	0.33	40	21
Xilxes	7	1.00	0	32	12
Component 1: Size = 4 nodes					
Paphos	3	0.67	0.5	38	26
Agios_Nicolaos	2	1.00	0	40	28
AmathousST3	2	1.00	0	40	18
AmathousST5	3	0.67	0.5	40	25
Thr3					
Component 0: Size = 5 nodes					
Roquetas	4	1.00	0	50	35
Sa_Torreta	4	1.00	0	40	21
Xilxes	4	1.00	0	32	12
Torre_de_la_Sal	4	1.00	0	29	15
Rodalquilar	4	1.00	0	50	27
Component 1: Size = 4 nodes					
Acqua_Azurra_3	3	0.67	0.5	40	31
Malta	2	1.00	0	39	29
Marzamemi	2	1.00	0	38	31
Acqua_Azurra_5	3	0.67	0.5	40	29
Component 2: Size = 3 nodes					
Paphos	2	1.00	0	38	26
AmathousST3	2	1.00	0	40	18
AmathousST5	2	1.00	0	40	25

4.3 Node-level analysis: specific properties of chosen geographic locations

Studying network measures of nodes may allow identifying “source” regions (expected to exhibit dominant degree and possibly betweenness centrality) where populations contribute predominantly to supply the whole system (or sinks, see Rozenfeld et al., 2008 for a discussion on their properties), or essential pathways (high betweenness centrality) maintaining the connectivity across distinct groups of populations, or biogeographic areas. However, when interpreting network properties of nodes it is important to keep in mind that, unless an exhaustive sampling has been performed, populations emerging as important sources (or sinks) or essential pathways may, rather than being important by themselves, reflect, for example, uneven geographic sampling for the system.

Many network measures for nodes can be plotted on the graph figures by choosing options allowing to assign a size or color to nodes depending on these properties.

Q4: Are some populations exchanging more migrants than others?

Degree

The degree is defined as the number of connections a node has in the network. That is, it summarizes how strongly a population (or an individual, or community) is connected to the other populations (individuals, communities) in the system.

Ideally, the degree of a node should give us an idea about the relative level of *connectivity* of a given population: how much it is exchanging migrants with other populations in the system, if it dominates the exchanges and how important input (source, pathway...or generic sink) it is in the system.

Although the degree is the simplest and the most used node characteristic in network theory, and likely one of the most important ones also from the ecology perspective, one must acknowledge the risks involved when applying it into ecology. The degrees of nodes are highly influenced by the homogeneity of sampling. For example, any data set where a region will be more densely sampled than others (unfortunately this applies to the majority of datasets in ecology) will result in higher degrees for the populations located in the more densely studied area even when these nodes do not *in situ* exchange more migrants than average with the other populations of the system. Remember that the use of percolation threshold is based on a relative ranking of distances among populations and these strongly depend on sampling.

Q5: Are some populations more important than others in maintaining gene flow?

Betweenness centrality

In the Table 4.2, right above the percolation threshold, one can see that two nodes have a much larger (about 40 to 60 times, in green in Table 4.2) betweenness centrality than the others: these populations are connecting the Eastern and Western Mediterranean cluster, and reveal that the region they belong to is an essential

pathways between the two main geographical areas. This observation is confirmed by analysis of a larger data set displayed in Figure 4.2. However, betweenness centrality may be sensitive to noisy data, and we strongly recommend double-checking findings e.g. with the help of the bootstrapping procedure.

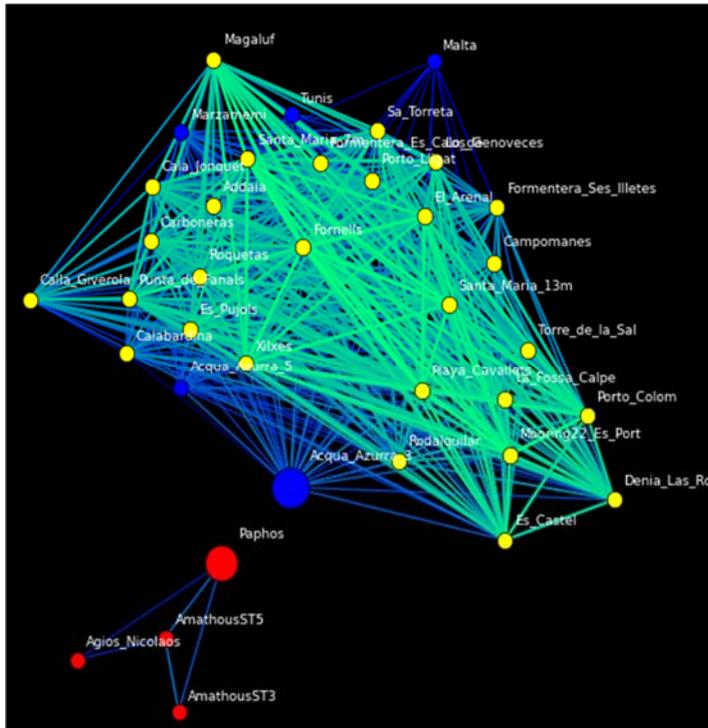


Figure 4.3 In the *Posidonia oceanica* dataset (Arnaud-Haond et al., 2007; Rozenfeld et al., 2008) the use of degree as a measure of gene flow on the global set of populations would indicate a much higher degree among Spanish (yellow nodes) than Eastern Mediterranean (red dots) meadows. Yet, rather than illustrating a higher gene flow among Spanish meadows this is clearly driven by the much higher sampling efforts along Spanish coasts than in the Eastern Mediterranean. When reducing the data set to the Spanish populations only, a more homogeneous sampling is obtained, allowing to some extent the –still cautious- use of degree as an estimator of connectivity. Populations exhibiting high betweenness centrality, reflecting areas of importance to relay gene flow between groups of populations, here the Eastern –red- and Western –yellow- Mediterranean, are illustrated by larger nodes

Assortativity

A network is said to show assortative mixing if the nodes in the network that have many connections tend to be connected to other nodes with many connections.

4.4 **Benchmarking and performance**

Next we give benchmarks of typical use cases with EDENetworks, and analyze the scaling of the computational time as a function of the input data size.

All the algorithms for EDENetworks are developed keeping the scaling of computational time and memory usage in mind. For example, there are two data structures for representing networks internally depending on the density of the network. Sparse networks are represented as adjacency dictionaries, such that for each node there is a dictionary object (i.e., a hash map) containing names of the neighboring nodes as keys and weights of the corresponding edges as values. In this approach the average cost of adding new edges and testing for existence of an edge is constant, and the average time it takes to iterate through all neighbors scales linearly with the number of neighbors a node has. Furthermore, the memory requirements also grow linearly with then number of neighbors. Dense networks are stored as weighted adjacency matrices, i.e., 2-dimensional arrays, where value of element i,j corresponds to a weight between i -th and j -th node, or zero if there is no edge between them. Note that the dense representation instead of the sparse one only yields a constant decrease in the time and memory complexity.

Measuring computational efficiency is, in general, not straightforward, since there are various factors which affect the speed of execution of any computer program. These factors include details of the computer hardware, the operating system, other programs interfering with the execution, a Python interpreter that is used, and so on. These factors should be kept in mind when interpreting the benchmarking results unless you are running your program in a setup identical to the one that was used to get those results.

We start by considering a typical use case of EDENetworks, and record the computational time for completing various tasks for the data set of *Posidonia oceanica* which was used as an example earlier in this section and is distributed with the program. The data set consists of 37 groups and 1468 samples each having 7 diploid loci sequenced. That is, in total there are 20 552 data points. The results are displayed in Table 1. We also run the test suite (excluding the group-based tests) for a dataset of environmental metagenomics comprising the 16S massive characterization of bacterial Operational Taxonomic Units (OTUs) associated to the green algae *Caulerpa racemosa* (Aires et al., 2013). This data set comprises 55 *C. racemosa* samples and the distribution of about 30.000 bacterial OTUs for a total of around 1.7 million data points. These results are in Table 2. A typical desktop computer (with AMD FX 4350 processor, 4GB of memory, and running Ubuntu Linux 12.04) was used for running EDENetworks. The times are reported for the computation of the results only and don't account for the time to update the GUI or other user interface elements. All the functions were run several times and fastest result selected in order to eliminate the effect of any background processes interfering with the computation. All the network statistics are calculated after automatic thresholding procedure is run for the distance matrix. Note also that the distance matrix calculations don't include the time to read the data from the disk, but includes the time to parse the data just after reading it.

All the computation times are within reasonable limits in order to guarantee smooth user experience when analyzing the data. Note that although the groups centric operations are fast, they are needed to repeat multiple times if null models are used.

For example, you need to run the whole analysis pipeline large number of times when estimating if a clustering coefficient values are statistically significant or if they are something that could be expected to be produced by a null model. From Table1 we can see that in PyPy 2.1.0 the whole pipeline takes around 50ms to run. To get sufficient statistics we could run permutation procedure that implies the pipeline is repeated by the program 1000 times, which would then amount to 50 seconds of processing time for the whole procedure of estimating the significance of the clustering coefficient results.

Benchmark	CPython 2.7.3	PyPy 2.1.0
Null model: shuffle alleles	3.9 milliseconds	0.61 milliseconds
Remove clones	3.1 milliseconds	1.5 milliseconds
Distance matrix (groups)	0.30 seconds	24 milliseconds
Automatic thresholding (groups)	53 milliseconds	18 milliseconds
Minimal spanning tree (groups)	6.1 milliseconds	1.3 milliseconds
Clustering coefficient (groups)	0.13 seconds	3.7 milliseconds
Betweenness centrality (groups)	28 milliseconds	2.1 milliseconds
Distance matrix (samples)	14 seconds	0.62 seconds
Automatic thresholding (samples)	2.9 seconds	0.97 seconds
Minimal spanning tree (samples)	0.13 seconds	27 milliseconds
Clustering coefficient (samples)	4.9 seconds	0.13 seconds

Table 1: Computational time for various tasks for EDENetworks using the *Posidonia oceanica* data set (Arnaud-Haond et al., 2007; Rozenfeld et al., 2008).

Next we consider the scaling properties of algorithms implemented in EDENetworks. That is, we look how fast the computation times increase when the size of the data increases. For example, if the scaling is linear, doubling the size of the data will also double the computation time, and if scaling is quadratic, same operation takes four times longer for a data set that is two times larger. This type of analysis is more robust against implementation details such as the computer hardware or the Python interpreter that is used. Scaling analysis is also useful if the execution time for a small data set is known and when one wants to analyze a larger version of similar data set.

Benchmark	CPython 2.7.3	PyPy 2.1.0
Null model: shuffle OTUs	0.60 seconds	92 milliseconds
Remove clones	0.32 seconds	0.16 seconds
Distance matrix (samples)	20 seconds	7.0 seconds
Automatic thresholding (samples)	65 milliseconds	24 milliseconds
Minimal spanning tree (samples)	3.2 milliseconds	0.53 milliseconds
Clustering coefficient (samples)	19 milliseconds	0.43 milliseconds

Table 2: Computational time for various tasks for EDENetworks using the *Caulerpa racemosa* data set (Aires et al., 2013). The Czekanowski dissimilarity was used for the distance matrix.

We will show empirically how the computational procedures reported in Table 1 scale for randomly generated data sets. In our random data sets we vary the number of samples but always keep the groups size to 10 samples, number of loci to 10, and generate uniformly random alleles as numbers between 0 and 200. The number of groups is varied between 10 and 300 for sample centered analysis and between 100 and 300 for group centered analysis (except for the distance matrix generation for which the sample centric interval is used). We leave out the network diagnostics for groups centric networks, since the algorithms are identical to the ones used for sample centric networks. Results are displayed in Figure 1. Note that in practice there are limits for extrapolating these scaling results. For example, the memory usage for storing distance matrices grows quadratically, and exceeding the amount of physical memory in a computer can slow down the processing speed severely.

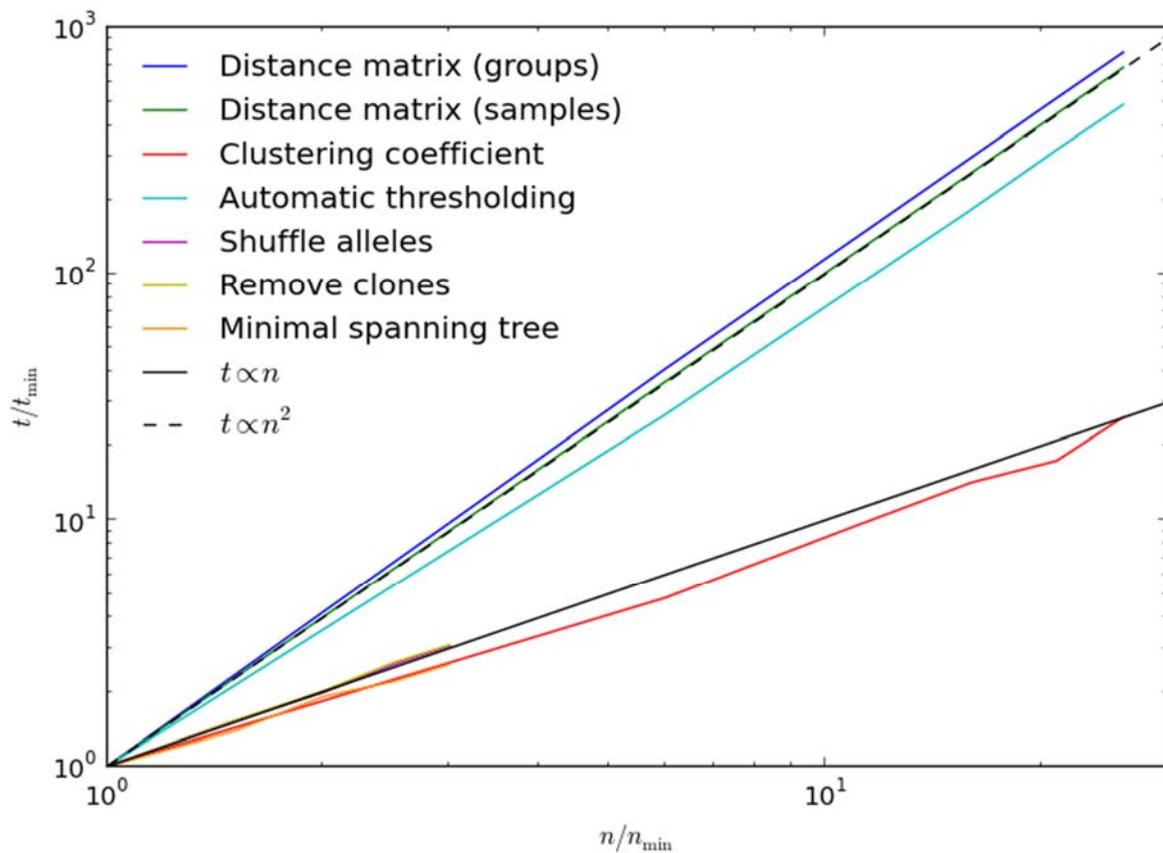


Figure 1: Empirical scaling of computational time of various data analysis steps in EDENetworks. Computations were done for a random data described in the main text. Horizontal axis represents the number of groups divided by the smallest number of groups, and vertical axis is represents the computational time divided by the minimum time. The two black lines correspond to linear (solid line) and quadratic (dotted line) scaling.

It is clear from Figure 1 that the scaling of the various procedures fall into two categories: ones that are roughly linear and ones that are roughly quadratic. Roughly quadratic procedures include distance matrix calculations, for which a quadratic scaling is optimal since the number of distance matrix elements grows quadratically with the size of the data. Automatic thresholding should scale roughly quadratically

since it takes a distance matrix as an input and sorts all its elements (sorting is a log-linear operation). The remaining operations that scale roughly linearly, and cannot be faster than that since they require inspecting each of the samples at least once.

It was shown above that the computation times for a typical data set are extremely fast (i.e., fast enough to be unnoticeable) for most of the operations or reasonable (i.e., few seconds) for the more demanding procedures such as distance matrix generation. Furthermore, in our empirical scaling benchmarks EDENetworks performs as would be expected from a program implementing optimal algorithms.

5 References

- Aires T, Serrao EA, Kendrick G, Duarte CM, Arnaud-Haond S (2013) Invasion is a community affair: Clandestine followers in the bacterial community associated to green algae, *Caulerpa racemosa*, track the invasion source. *PloS ONE* 8, e68429-e68429.
- Albert, R. and Barabasi, A.L. (2002) Statistical mechanics of complex networks, *Reviews of Modern Physics*, 74, 47-97.
- Albert, R., Jeong, H. and Barabasi, A.L. (2000) Error and attack tolerance of complex networks, *Nature*, 406, 378-382.
- Arnaud-Haond, S. and K. Belkhir (2007). "GENCLONE: a computer program to analyze genotypic data, test for clonality and describe spatial clonal organization." *Molecular Ecology Notes* 7(1): 15-17.
- Arnaud-Haond, S., C. M. Duarte, et al. (2007). "Standardizing methods to address clonality in population studies." *Molecular Ecology* 16(24): 5115-5139.
- Bascompte, J., Jordano, P., Melian, C.J. and Olesen, J.M. (2003) The nested assembly of plant-animal mutualistic networks, *Proceedings of the National Academy of Sciences of the United States of America*, 100, 9383-9387.
- Becheler, R., Diekmann, O.E., Hily, C., Moalic, Y. and Arnaud-Haond, S. (2010) The concept of population in clonal organisms: mosaics of temporally colonized patches are forming highly diverse meadows of *Zostera marina* in Brittany, *Molecular Ecology*, 12:2394-2407.
- Bray JR, Curtis JT (1957) An ordination of the upland forest communities of southern Wisconsin. *Ecological Monographs* 27: 325-349
- Dyer, R.J. and Nason, J.D. (2004) Population Graphs: the graph theoretic shape of genetic structure, *Molecular Ecology*, 13, 1713-1727.
- Fortuna, M.A., Albaladejo, R.G., Fernández, L., Aparicio, A. and Bascompte, J. (2009) Networks of spatial genetic variation across species, *Proceedings of the National Academy of Sciences*, 106, 19044-19049.
- Goldstein, D.B., Linares, A.R., Cavallisforza, L.L. and Feldman, M.W. (1995) Genetic Absolute Dating Based on Microsatellites and the Origin of Modern Humans, *Proceedings of the National Academy of Sciences of the United States of America*, 92, 6723-6727.
- Hernández-García, E., Herrada, E.A., Rozenfeld, A.F., Tessone, C.J., Eguíluz, V.M., Duarte, C.M., Arnaud-Haond, S. and Serrão, E. (2007) Evolutionary and Ecological Trees and Networks. *Nonequilibrium Statistical Mechanics and Nonlinear Physics In*, AIP Conference Proceedings. American Institute of Physics, New York, 2007 pp. 78-83.
- Hernandez-Garcia, E., Rozenfeld, A.F., Eguiluz, V.M., Arnaud-Haond, S. and Duarte, C.M. (2006) Clone size distributions in networks of genetic similarity, *Physica D*, 224, 166-173
- Lee, S.H., Kim, P.J. and Jeong, H. (2006) Statistical properties of sampled networks, *Physical Review E*, 73, -.
- Moalic, Y., Arnaud-Haond, S., Perrin, C., Pearson, G.A., Serrão, E.A (2011). Traveling in time with networks: Revealing present day hybridization versus ancestral polymorphism between two species of brown algae, *Fucus vesiculosus* and *F. spiralis*. *BMC Evolutionary Biology*, 11.
- Moalic, Y., Desbruyères, D., Rozenfeld, A.F., Bachraty, C., Duarte, C.M. and Arnaud-Haond, S. (2012) Worldwide biogeography of hydrothermal vents revisited with network analysis. *Systematic Biology*, 61:121-137.

- Newman, M.E.J. (2002) Assortative mixing in networks, *Physical Review Letters*, 89, -.
- Newman, M.E.J. (2003) The structure and function of complex networks, *Siam Review*, 45, 167-256.
- Newman, M.E.J. (2010) *Networks, An Introduction*, Oxford University Press
- Pastor-Satorras, R., Vazquez, A. and Vespignani, A. (2001) Dynamical and correlation properties of the Internet, *Physical Review Letters*, 87, -.
- Proulx, S.R., Promislow, D.E.L. and Phillips, P.C. (2005) Network thinking in ecology and evolution, *Trends in Ecology & Evolution*, 20, 345-353.
- Reynolds, J, Weir, B.S., Cockerham C.C (1983) Estimation of the Coancestry Coefficient: Basis for a Short-term Genetic Distance, *Genetics* 105: 767-779.
- Rozenfeld, A.F., Arnaud-Haond, S., Hernández-García, E., Eguíluz, V.M., Matías, M.A., Serrão, E.A. and Duarte, C.M. (2007) Spectrum of genetic diversity and networks of clonal populations, *Journal of the Royal Society Interface*, 4, 1093-1102.
- Rozenfeld, A.F., Arnaud-Haond, S., Hernandez-Garcia, E., Eguiluz, V.M., Serrao, E.A. and Duarte, C.M. (2008) Network analysis identifies weak and strong links in a metapopulation system, *Proceedings of the National Academy of Sciences of the United States of America*, 105, 18824-18829.
- Stauffer, D. and Aharony, A. (1994) *Introduction to Percolation Theory*. London.
- Somerfield, P.J. Identification of the Bray-Curtis similarity index: Comment on Yoshioka (2008). *Marine Ecology Progress Series* 372:303-306.
- Watts, D.J. (2004) The "new" science of networks, *Annual Review of Sociology*, 30, 243-270.
- Watts, D.J. and Strogatz, S.H. (1998) Collective dynamics of 'small-world' networks, *Nature*, 393, 440-442.