NetStruct: python pipeline

User Manual

The python version of NetStruct was coded by Omer Tzuk
cliffon@gmail.com

It is based on the Mathematica version of NetStruct coded by Gili Greenbaum
gili.greenbaum@gmail.com

Contents

1 Introduction

NetStruct is a software package for inference and analysis of population structure from genetic data, based on network theory methodologies. It implements analyses detailed in the following publication:

> Greenbaum G., Templeton A.R., Bar-David S. (2016). Inference and analysis of population structure from genetic data using network theory. *Genetics*, 202, 1299-1312.

This python pipeline performs the following procedures:

1. Build genetic similarity matrix from genotype data
2. Perform community detection analysis for a range of edge-removal thresholds
3. Perform SAD analysis for a specific edge-removal threshold

Each procedure is run independently and creates its own output files, which can then be used as input for the following procedure

1.1 Tips and recommendations

- The first procedure, building the similarity matrix, is usually the most time-intensive procedure (and the only one that depends on the number of loci; see publication). It is therefore recommended to run this procedure once, and use the output file, which contains the genetic similarity matrix, to run exploratory analyses with different thresholds etc.
- It is possible to give the output files different names than the default names. To do that, when typing the command line (for any procedure), add the desired file name as an additional parameter at the end.

2 Installation

The python implementation of NetStruct requires, beside python 2.7, several other packages.

2.1 Instructions for non python-users

The instructions below are intended for Windows 7 operating systems, but are similar with other operating systems (these instructions can be found online). They are intended for users without any experience in python, and for computers without any previous python installations.

2.1.1 Installing Anaconda

Anaconda is a platform which, besides the basic python platform, has many useful packages for science application.
In order to install the Anaconda platform:

1. Download the Anaconda installer on `www.continuum.io/downloads#windows`.
2. Run the installer. Be sure to check the "add to PATH" box.

After installing Anaconda, python is installed and so are most packages required by NetStruct.

2.1.2 Installing python-igraph and deepdish packages

Besides the packages in Anaconda, NetStruct requires two more packages: `deepdish` and `python-igraph`.

- In order to install the `python-igraph` package:
  1. Download the igraph-python *.whl file from `www.igraph.org/python` or `www.lfd.uci.edu/~gohlke/pythonlibs/#python-igraph`. Make sure to choose the python 2.7 ("cp27") and the correct version (32-bit or 64-bit, depending on the Anaconda version installed).
  2. In the command prompt, move to the folder where you have downloaded the *.whl file. Type

     ```
     python -m pip install SomePackage
     ```

     with the *.whl file you have downloaded instead of `SomePackage`. This should install the igraph package.

- In order to install the `deepdish` package, in the command prompt type:

  ```
  python -m pip install deepdish
  ```

2.2 Instructions for experienced python-users

These instructions are intended for users with some experience in installing python packages.

1. Make sure that you have the latest version of python 2.7.
2. Before running NetStruct, install the following packages (or verify that they are already installed): `numpy`, `scipy`, `matplotlib`, `deepdish`, `python-igraph`

2.3 Downloading NetStruct

NetStruct can be downloaded here `github.com/Omer80/NetStruct_python`. Download the ZIP file to a work folder (click "Clone or download" and then click "Download ZIP"), where you would like to run the analyses. Extract the ZIP file.

Note - only the `netstruct.py` file is neccesary for running netstruct. If you are running NetStruct on multiple analyses in several folders, simply copy this file to each work folder.

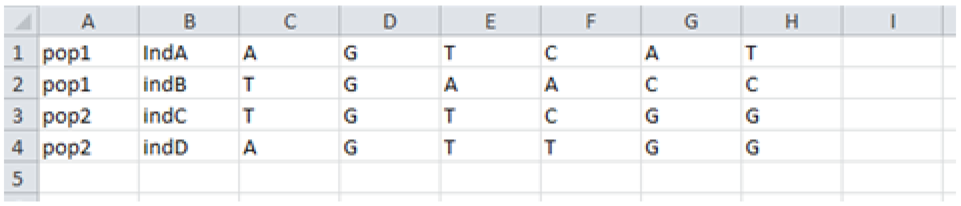
3 Preparing the data

The input for the first procedure in the pipeline, building genetic similarity matrix, is a CSV file. Each row contains data of a single individual in the following manner (Fig. 1):

- **First column** - The first column contains the name of the sampled population. This could be a sampling site or some other putative partition of individuals to populations. This information is not used in any of the analyses, since NetStruct does not rely on *a prioiri* information, and is intended only for further analysis outside of NetStruct. If there is no information on putative populations, it is best just to enter "1" (or some other text) for all individuals in this column.
- **Second column** - The second column contains the individual's ID. This could be any text that identifies the individual (e.g. name, identification number, etc.)
- **Columns 3 to $(2 * \#\text{loci} + 2)$** - Starting from column 3 , each pair of columns contain a genotype from one locus (although the procedure itself is not limited to diploid organisms, this NetStruct pipeline currently only supports analysis of diploids). This data is treated as text, even if numerical data (e.g. for microsattelites) is entered. *For missing data, enter 0 at both columns of the relevant locus*

Note that this is the same data format used for the Mathematica version of NetStruct.

The input for the other procedures are the outputs of previous procedures, and are detailed in the next section.



	A	B	C	D	E	F	G	H	I
1	pop1	IndA	A	G	T	C	A	T	
2	pop1	indB	T	G	A	A	C	C	
3	pop2	indC	T	G	T	C	G	G	
4	pop2	indD	A	G	T	T	G	G	
5									

Figure 1: An example of a CSV file to be used as input for NetStruct. The data set contains 4 individuals, sampled at two sampling sites (pop1 and pop2), with 3 SNPs analyzed

4 Running NetStruct procedures

4.1 Building genetic similarity matrix

4.1.1 Running the procedure

The first procedure in the NetStruct pipeline is to build the genetic similarity matrix from the genotype data. In order to run the procedure, you should prepare a work folder which contains the `NetStruct.py` file. In the same folder place the prepared CSV file with individual's genotypes. Next, open the command prompt and navigate to the work folder. In order to build the genetic similarity matrix type:

```
python netstruct.py -b filename.csv
```

with the prepared file name instead of `filename`. You should see the following text:

```
Constructing genetic similarity matrix
```

4.1.2 Output file

After the procedure is terminated you should see a *.hdf5 file with the same name as your data file in the work folder. This file contains the genetic similarity matrix in a packed format, and can be used as input for the community detection analysis procedure.

4.2 Community detection analysis

4.2.1 Running the procedure

After constructing the genetic similarity matrix, it is now possible to apply an edge-removal threshold to obtain a network, and apply the community detection procedure on this network. The procedure in this pipeline analyzes a range of community detection thresholds in set intervals.
In the work folder type:

```
python netstruct.py -c filename.hdf5 min_th max_th int_th alg
```

where filename is the name of your file, min_th is the minimum threshold to be analyzed, max_th is the maximum threshold to be analyzed, int_th is the threshold interval, and alg is the number of algorithm to be used. NetStruct currently allows application of six weighted community detection algorithms (details on these algorithms can be found in the documentation of the igraph package). The algorithms are:

- 1 - Label propagation algorithm (`community_label_propagation` in igraph)
- 2 - Girvan-Newman algorithm (`community_edge_betweenness` in igraph)
- 3 - FastGreedy algorithm (`community_fastgreedy` in igraph)
- 4 - Walktrap algorithm (`community_walktrap` in igraph)
- 5 - Spinglass algorithm (`community_spinglass` in igraph)
- 6 - Spectral analysis algorithm (`community_leading_eigenvector` in igraph)

All these algorithms are applied to the networks as weighted networks.
For example the procedure could be run with this command:

```
python netstruct.py -c data.hdf5 0.1 0.2 0.05 6
```

Which will analyze 21 networks from edge removal 0.1 to 0.2 in intervals of 0.05, applying the Spectral analysis community detection algorithm to each network. When the procedure initiates, you should see the following line:

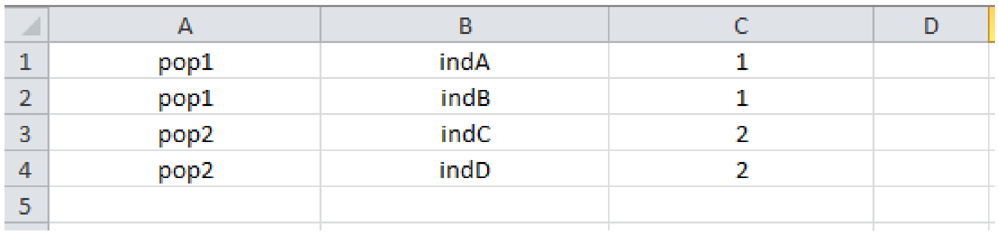
```
Processing command:  data.hdf5 0.1 0.2 0.05 6 None
```

4.2.2 Output files

After the procedure is terminated you should see two additional file in the work folder: a file named Community_Detection_Results.zip and a file named piecharts_Community_Detection_Results.pdf.

- The Community_Detection_Results.zip file contains for each network analyzed (i.e. for each edge-removal threshold in the selected range) two files. The first is a CSV file, named according to the corresponding threshold (e.g. 0_180.csv), which contains three columns for each individual. The first column contains the sampled population, the second the individual's ID, and the third indicates the community this individual is assigned to. The communities are numbered from 1 upwards, ordered by the size of the community (lower number correspond to larger subpopulations). The second type of file is a *.hdf5, which contains the data necessary for further analyses.

- The piecharts_Community_Detection_Results.pdf. is a PDF illustrating how the communities are distributed according to the sampled population. For each threshold and each sampled population, a piechart depicts how individuals are assigned to communities (for a given threshold, each community is assigned a color; colors may be assigned differently for different thresholds, as community detection results change). This PDF can be viewed to scan the results and select thresholds for further analyses or to rerun the procedure with different parameters.



	A	B	C	D
1	pop1	indA	1	
2	pop1	indB	1	
3	pop2	indC	2	
4	pop2	indD	2	
5				

Figure 2: An example of a CSV output file for a specific threhsold from the community detection procedure.

4.3 Strength of association distribution (SAD) analysis

4.3.1 Running the procedure

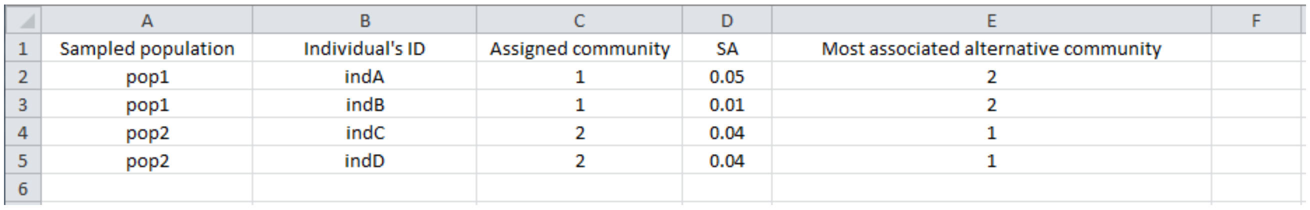
After community detection is performed and population structure is identified, it is possible to run an additional analysis to explore the patterns with which individuals are associated with subpopulations. The procedure analyzes a single network, after community detection was performed, identified by the edge-removal threshold applied for its derivation. As input, this procedure requires the Community_Detection_Results.zip file and a threshold to be analyzed.
In the work folder, type:

```
python netstruct.py -s Community_Detection_Results.zip th
```

with `th` being the threshold chosen to be analyzed (e.g. `0.198`).

4.3.2 Output file

After the procedure is terminated you should see an additional file in the work folder named `SADresults.csv`. This file contains a list, where for each individual the following details are given: sampled population, individual's ID, assigned community, Strength of Association, and the most associated alternative community. For the latter, see the paper (it is the term that maximizes equation three in the paper). Most SA values should be positive, negative values should be taken to mean that this individual is very weakly assigned, and could just as well be assigned to the alternative community.



	A	B	C	D	E	F
1	Sampled population	Individual's ID	Assigned community	SA	Most associated alternative community	
2	pop1	indA	1	0.05	2	
3	pop1	indB	1	0.01	2	
4	pop2	indC	2	0.04	1	
5	pop2	indD	2	0.04	1	
6						

Figure 3: An example of a CSV output file from the SAD analysis procedure.

5 Citing NetStruct

Please cite:

Greenbaum G., Templeton A.R., Bar-David S. (2016). Inference and analysis of population structure from genetic data using network theory. *Genetics*, 202, 1299-1312.