A brief tutorial for using `Transmic`

Identification of putative disease transmission links using hierarchical clustering on phylogenetic trees

Kaveh Pouran Yousef
kaveh.yousef@fu-berlin.de

November 2, 2016

1 What is `Transmic`?

`Transmic` is a Python program for identifying clusters in phylogenetic trees by combining inter-taxon patristic distances with statstical support-values of the corresponding branch nodes. The node support may be given either by posterior values (used in the Bayesian phylogenies) or bootstrap values (usually used in the Maximum Likelihood setting). The corresponding clades of phylogenetic trees indicate sets of evolutionary close species.

`Transmic` is customized for the identification of putative disease transmission clusters. In particular, it allows for multiple follow-up viral sequences extracted from the same individual (for a detailed description of the underlying method see supplementary information in [2]).

`Transmic` provides some visualization options for plotting basic statistics of the clustering outcome.

2 How to use `Transmic`?

2.1 Required Python libraries

The program requires Biopython [1] and Dendropy version >4.0 [3].

2.2 Files required by the program

Besides the Python program transmic.py itself, three additional files are needed for running it:

1. `transmiclib.py`: a library file containing the necessary python functions (included in the download).
2. The control file of the program called `controlfile.tbsc` (included in the download).
3. A file containing the phylogenetic tree in Newick or Nexus format.

As an additional option an .xls file can be included in order to incorporate meta-information for each taxon of the tree. The library and the control files must be placed in the same directory as the program. The other two files may also be located in other directories than the program specified in the control file.

2.3 Format of genetic sequence identifiers

Since the program allows for two different modes (individual- and sequence-based clustering, see subsection 2.4) the format of taxa in the phylogenetic tree needs to be adopted accordingly. In the individual-based mode, the name of the taxa must contain the sequence ID and the individual/patient ID separated by an underscore. For instance, a taxon with the name `12-1023_0321` indicates that the sequence ID is `12-1023` and this sequence belongs to the individual/patient ID is `0321`.

In contrast, in the sequence-based mode the sequence IDs may have any format as defined in the phylogenetic tree (i.e. in particular, an underscore is not required).

2.4 Configuring the file `controlfile.tbsc`

The program configuration is conducted through a control file called `controlfile.tbsc`. It must be placed in the same directory as the transmic.py file. A sample control file is contained in the program package. In order to customize the program for your needs, the parameters in the double quotes have to be modified. Consider the following example of a control file:

```
# ------------------------------------------------------
# TransmicBS control file

# Transmic control file
# Please only edit parameter values within double quotes.
# ------------------------------------------------------
# Clustering parameters

# 1.Name of the tree file
# (or absolute path if file not in same directory as the programm)
tree_file="/home/user/myfoder/RAxML_bipartitions_WithFP.BS_TREE"

# 2.Tree format (newick/nexus)
tree_format = "nexus"

# 3.Node support (bootstrap/posterior)
node_support = "posterior"

# 4.Cut-off value for the mean pairwise patristic distances (mppd)
# i.e. a clade is a transmission cluster if  mppd =< cutoff_mppd
#type here one numeric value or a range between x and y in z-steps by typing x:z:y
```

```
#cutoff_mppd="0.001:0.0005:0.003"
cutoff_mppd="5"

# 5.Cut-off for the node support value (bootstrap or posterior, in percent)
# i.e. a clade is a transmission cluster if  value >= cutoff_node_support
#type here one numeric value or a range between x and y in z-steps by typing x:z:y
#cutoff_node_support="80:5:100"
cutoff_node_support="95"

# 6.Patient-wise clustering (true/false)
# This option only works with sequence identifiers consisting
# of a unique sequence ID followed by a patient ID separated by
# an underscore e.g. seqID_patID. In this case set it to true.
# In all other cases (e.g. simple sequence-based clustering) set this
# option to false.
patWiseClust="false"

# 7.Name of excel file with patient meta-information
# The first column of this excel file must contain the sequence-IDs
# of the taxa in the tree file. The following columns contain additional
# information for each sequence. If
# If no file with meta-information available, leave this empty: "".
database_file="metainfo_database.xls"

# 8.Identifier of the outgroup sequence
outgroupSeqId="SIVcpz_ant"

# ------------------------------------------------------
# Visualization
#
# 9. Figure showing distribution of patristic distances between taxa (true/false)
# Set to true to generate a plot.
figure_distances="true"

# 10. Figure showing number of clusters  depending
# on distance and support thresholds (true/false)
# Set to true to generate a plot.
figure_nr_clusters="true"

# 11. Figure showing mean size of clusters depending
# on distance and support thresholds (true/false)
# Set to true to generate a plot.
figure_sizeof_clusters="true"
# ------------------------------------------------------------------
```

All the lines beginning with a hash symbol # are ignored by the program as comments and all control parameters are entered within quotation marks "...". In the following we explain the control parameters step by step:

1. `tree_file`: this parameter determines the path to the file containing the phylogenetic tree. If the file is located in another directory, in Linux you may either type the absolute path to it e.g. `/home/somedirectory/ExampleTree` or the relative path (depending on the location of the program).

2. `tree_format`: the format of the tree file. The two alternative options are newick and nexus.

3. `node_support`: type of support statistic assigned to the tree nodes. The two alternative options are bootstrap and posterior.

4. `cutoff_mppd`: distance threshold (mean pairwise patristic distance, MPPD) for the hierarchical clustering method. This should either be a scalar value or a sequence of values entered in the format $x : z : y$. This yields a list of clustering thresholds starting from the value x and going to y in z-steps.

5. `cutoff_bootstrap`: bootstrap support threshold which determines the candidate clades as potential clusters (units: %). This should either be a scalar value or a sequence of values entered in the format $x : z : y$. This yields a list of clustering thresholds starting from the value x and going to y in z-steps.

6. `patWiseClust`: boolean parameter which determines if the clustering should be based on individuals (e.g. virus hosts). Setting this parameter to `"true"` invokes a clustering method were the distance between individuals is computed as the minimal tree distance between any pair of viral sequences belonging to the two individuals (for details see Pouran Yousef *et al.* [2] and the corresponding supplementary information). This method is useful if multiple sequences per individual are allowed. Otherwise, if clustering should be conducted based on a single sequence per individual then this parameter should be set to `"false"`.

7. `database_file`: (optional) Name of the .xls file containing the meta-information about the genetic sequences. The first column of this file has to contain the sequence IDs which are equal to the sequence IDs representing in the phylogenetic tree (see subsection 2.3). The following columns may contain further information corresponding to each genetic sequence, such as collection year and subtype of the viral sequence or the characteristics of the individuals carrying the virus etc. Note that currently only the .xls format. Therefore, in order to use a table file containing meta-information for the genetic sequences, simply export your data into .xls format. This parameter empty if no meta-information is available (i.e. write `""`).

8. `outgroupSeqId` Sequence ID of the sequence in the tree used as an outgroup (e. g. for the purpose of rooting the tree). Type the name of this sequence here. Leave this parameter empty if no outgroup is used.

9. `figure_distances` a boolean parameter determining if a distribution plot of tree distances should be made.

10. `figure_nr_clusters` a boolean parameter determining if a plot of the number of clusters depending on the clustering threshold should be made. This parameter should be only set to `"true"` if a range of clustering thresholds is given by the parameter `cutoff_mppd`.

11. `figure_sizeof_clusters` a boolean parameter determining if a plot of the size of clusters depending on the clustering threshold should be made. This parameter should be only set to `"true"` if a range of clustering thresholds is given by the parameter `cutoff_mppd`.

2.5 Running the program

After ensuring that Python is installed on your system, all required files are available in the program folder and the configuration of the control file is finished, the program can be started in the Linux shell or Wondows command line by:

```
me@mycomputer:~$ python transmicBS.py
```

2.6 Table output

The clustering output is available in table format. The first column of the output table denotes the number of the cluster that the genetic sequence is associated with. This number is zero if the sequence was not associated with any cluster. The second column of the cluster contains the sequence ID. From the column three on the output file contains meta-information provided by the (optional) additional file using the parameter `database_file`. If this parameter is empty the corrsponding columns in the output file remain empty.

2.7 Text output

The clustering output is also available in text format. As an example, if three clusters have been identified, the text output file in the directory `text_out` may have the following form:

```
This is an output file of TransmicBS.

Transmission clusters computed according to the distance  threshold of 0.013
and a bootstrap threshold of 95.0 %.
Number of sequences:  17
Number of identified clusters:  3
Average number of sequences in a cluster:  2.6
Number of sequences not included in clusters:  9

1.
```

```
Mean value of pairwise patristic distances 0.0097
Node support (%): 96
['F', 'G']

2.
Mean value of pairwise patristic distances 0.0118
Node support (%): 100
['M', 'N', 'O']

3.
Mean value of pairwise patristic distances 0.0081
Node support (%): 99
['B', 'C', 'D']
```

2.8 Acknowledgement

Credits and thanks go out to Silvana Gromöller who wrote an earlier version of this program.

References

[1] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics (Oxford, England)*, 25(11):1422–1423, June 2009.

[2] K. Pouran Yousef, K. Meixenberger, M. Smith, S. Somogyi, S. Gromöller, D. Schmidt, B. Gunsenheimer-Bartmeyer, O. Hamouda, C. Kücherer, , and M. von Kleist. Inferring HIV-1 Transmission Dynamics in Germany From Recently Transmitted Viruses. *JAIDS, accepted.*, 2016.

[3] J. Sukumaran and M. T. Holder. The DendroPy Phylogenetic Computing Library Documentation. Retrieved June 23, 2016 from http://dendropy.org/.