Guide for using convert2nex

Introduction

This document aims to explain how to use the “convert_to_nex.py” script can be used in order to convert an Excel spreadsheet (.xlsx) into a .nex file that contains relevant commands for analysis by MrBayes.

It should be noted that the relevant commands included in the converted file are *not* sufficient (in the vast majority of cases) to produce good quality results and that following the conversion of the data, *adjustments should be made* to the resulting file. The *main aim of this program is to save time* from being spent on manually transcribing the .xlsx file into a .nex one.

Any issues or bugs with the code can be directed to timforrer16@gmail.com. Alternatively, you can raise issues in the Github repository for this code found here: https://github.com/tim-forrer/Convert2Nex. Please note also that this code is not currently being developed upon and therefore feature requests or any other suggestions for improvement of the code beyond the fixing of bugs are unlikely to be considered.

Requirements

The requirements for the script are not high. It was developed on a system that had Python 3.8.2. To my knowledge, the script should run on *Python 3.6+* but this has not been tested and as such if issues are found with running the script, the first port of call should be to ensure your version of Python is updated to version 3.8.2 or higher.

This code also requires the installation of the “xlrd” module. Provided that Python is installed this can be done by entering the following command into the terminal (Linux) or Command Prompt (Windows):

```
python -m pip install xlrd
```

The script has been tested to work on both *Windows 10* and *Ubuntu 20.04*. I struggle to imagine that alternative operating systems will cause this script to fail but I mention these as tested operating systems for completeness.

Please ensure that “convert2nex.py” is placed in the *same directory* as “script.py”.

The target .xlsx file must conform to a certain standard in order for this script to work. *Each taxon must be on a new row*, with *no rows left blank* between any two taxa. It is also assumed that that the *final two columns of the spreadsheet are LAD and FAD in that order.* It is also required that *nothing is entered* into the spreadsheet beyond the FAD column. This is so that the script can correctly determine the position of the final column of the target data.

Obtaining the Python scripts

Two files are needed for this conversion: “script.py” and “convert2nex.py”. These files should be available in the same location that this guide is found, but if they cannot be located then they can be downloaded from:

Please ensure that these two scripts are place in the *same directory.*

Setting up the script

Before the script can be run, the user is required to open and edit the “script.py” file and input the following data:

Variable	Description
file_loc	The absolute location of the file on the user’s system. Must be surrounded by “” or ‘’. The name of the file must be included as well. Please note that if using a windows system, copying the file location will result in a string that looks like “C:\Users\timfo\Desktop\data.xlsx” – all the backslashes must be converted to forward slashes like so: “C:/Users/timfo/Desktop/data.xlsx”
sheet_index	The index of the desired sheet. Please note that index counting starts from 0 i.e. the first sheet has index 0, the second sheet index 1 etc.
row_start_index	The index of the first row that contains taxon data. Again, index counting starts from 0.
row_end_index	The index of the final row of the sheet that contains taxon data. Index counting starts from 0.

An example of how to set up the script is given below.

```
import convert2nex

# Edit these lines of code in order to convert the desired spreadsheet into a .nex file.
file_loc = "/home/tim/Desktop/Plectambonitoidea (2)/data.xlsx"
sheet_index = 1
row_start_index = 3
row_end_index = 118

# Nothing beyond here needs to be changed.
sheet = convert2nex.load_data(file_loc, sheet_index)
convert2nex.generate_nexus_file(sheet, row_start_index, row_end_index + 1)
```

In this example, the location of the file is at “home/tim/Desktop/Plectambonitoidea (2)/data.xlsx”.

The spreadsheet with the coded data is the second spreadsheet and therefore has index 1.

The taxon data starts from row 4 which therefore has an index of 3.

Similarly, the taxon data ends on row 119 which corresponds to index 118.

Running the script

Once the script has been setup correctly you are ready to convert the file to a .nex file format. How you do this will depend upon your operating system. Included here are instructions for both Linux and Windows 10.

Monospaced text represents what is displayed on the terminal with bold script indicating manual input by the user.

On Linux

Open the terminal (Ctrl + Alt + T on many Linux based operating systems) and navigate inside terminal to the folder containing “script.py”:

```
$ cd location/of/folder/containing/script
```

Then, enter the following command:

```
$ python3 script.py
```

The script should run as required and a file named “generated_file.nex” should appear in the same location as “script.py” is found.

On Windows
It is advised that the user first navigates to the location of “script.py”, right clicks the file and selects “Properties”. In the window that appears, the location of the folder containing “script.py” will be displayed which can be copied and then pasted as appropriate into the Command Prompt.

Open the Windows Command Prompt by pressing the Windows key on your keyboard and typing “cmd” then press Enter.

Navigate inside the command prompt to the folder containing “script.py”:

```
C:Users\CurrentUser> cd location\of\folder\containing\script
```

Enter the following command:

```
location\of\folder\containing\script> python script.py
```

The script should run as required and a file named “generated_file.nex” should appear in the same location as “script.py” is found.

Finished!

The .nex file has now been generated and is ready for further modification before running it through MrBayes.