anagallis, a program for hierarchic character optimization
v1.03, 8 September 2020
Jan De Laet

This `anagallis_v1.03_08SEP2020.readme.1st.pdf` file contains some basic information to get started with the program.

Other files in the distribution tarfile:

- Linux executable `anagallis_v1.03_08SEP2020.linux` is a dynamically linked 64-bit executable compiled with gcc 9.3.0 (the program is written in C), extensively tested using valgrind.
- MacOS executable `anagallis_v1.03_08SEP2020.macos` is a dynamically linked High Sierra executable
- File `anagallis_v1.03_08SEP2020.changelist.pdf` lists the differences with the previous versions.
- File `anagallis_v1.03_08SEP2020.documentation.pdf` is a text dump of the built-in documentation of the program that was converted to pdf (adding page numbers and some line breaks in the process).
- The `examples` subdir contains some commented example inputfiles.
- Some relevant pdfs can be found in the `pdfs` subdir.

The program provides an implementation of the ideas that are developed in De Laet (2005, especially pp. 110-111; see also De Laet 2015, pp. 552-556): the problems with inapplicable data that were first pointed out by Maddison (1991) disappear when maximizing similarity that can be interpreted as homology. As such, they provide a solution to those problems that makes sense from a biological point of view. Similarity is used in a well-defined technical meaning here: an observed point of similarity amounts to a prior hypothesis of homology that is ultimately rooted in empirical observation.

A paper illustrating the approach and the program using some case studies is still in preparation. In the meantime, the built-in help contains an extensive discussion of the theory behind the program, an outline of the algorithm, and several examples that illustrate the concept of hierarchic optimization. The built-in documentation also has a brief comment on how the approach of anagallis differs from Brazeau et al.'s 2017 algorithm for inapplicable data (see De Laet 2017 for some first comments; a version of the dataset discussed in that report that works with this version is included here as test example 5).

Some old notes on anagallis' algorithm to calculate the tree score of a character hierarchy can be found in the 2012 presentation that is included in the pdf subdirectory. The path flips that are required to guarantee optimality that are mentioned in that presentation are implemented in this version up to a degree: optimality of reported tree scores is guaranteed as long as no regular unconstrained Fitch optimization of an absence/presence character in a character hierarchy has a region of absence that has more than 11 (default; can be increased to 21) neighbouring regions of presence. This is a situtation that is highly unlikely with empirical data on optimal or near-optimal trees. Whenever that constraint is not met, tree scores reported are heuristic approximations and the program prints a warning to that effect.

Comments or bug reports are highly appreciated. They can be sent to `jan.de.laet@anagallis.be`. A list of bugs and known issues will be maintained at www.anagallis.be.

.

To get started, copy the gzipped tarfile file in some directory, then open a terminal and go that directory. There, execute the following command:

```
tar -xvf anagallis_v1.03_08SEP2020.tar.gz
```

That will unpack the file into a subdirectory `anagallis_v1.03_08SEP2020` that gets created. Move to that subdirectory and there you find the files and subdirectories of the distribution.

For ease, you can copy the executable of choice to file `anagallis`. For the macOS executable that would be

```
cp anagallis_v1.03_08SEP2020.macos anagallis
```

To start the program, then just enter:

```
./anagallis
```

The core functionality of this program - definition and analysis of character hierarchies - is relatively well tested. That said, the program still performs a number of consistency checks during the calculation of the score of a character hierarchy on a tree. If any such test fails, the program quits with an appropriate message. Please let me know when this happens, and with what data and commands.

The main check is this one. At any level in a character hierarchy, let nsubc be the number of subcharacters for this level on a given tree on which the hierarchy is optimized. Let nzr and nor then be the number of regions within those subcharacters in which the a/p character for that level is either absent or present. The number of indel events in that character must then be equal to nzr + nor – nsubc. This relationship can be used to get the value of the fourth variable given the values of the three other ones. But, as in the beta versions, the number of indels is still calculated independently of the values of nsubc, nzr, and nor. This allows the above relationship to be used as a dynamic consistency check if all calculations are correct. The drawback is that it comes with some overhead. Future versions will start to focus on speed and may include a switch to turn of such consistency checks.

Some of the more obvious omissions in this version:

- special keys such as arrows don't work
- no support for nexus files
- no command history
- no executable for windows

Here is an overview of the included example files (several of them contain small corrections and additions compared to the beta distribution):

- `example_1.after_Maddison_1991.ana`
  An analysis of one of Maddsion's (1993) classic examples.
- `example_2.Strong_Lipscomb_1999_fig1.ana`
  An analysis of a first example discussed by Strong and Lipscomb (1999).
- `example_3.Strong_Lipscomb_1999_figs4-5.ana`
  An analysis of another example from Strong and Lipscomb (1999).
  Here, the problems with inapplicables get confounded by problems of collapsing zero-lentgh branches.

- `example_4.after_De_Laet_2005_pg113.ana`
  An example of how alignments can be analyzed as nested character hierarchies.
  A discussion of the same example in the context of alignment free-analysis can be found in De Laet (2005).

- `example_5.Brazeau_et_al_2017_fig3.ana`
  An analysis of Brazeau et al. 's (2017) Fig. 5 under a first set of biological assumptions.
  This analysis is also discussed in De Laet (2017), using an anagallis development version with slightly different interface.

- `example_6.Brazeau_et_al_2017.fig3-alt.ana`
  An analysis of Brazeau et al.'s (2017) Fig. 5 under an alternative set of biological assumptions.

In combination, examples 5 and 6 illustrate that, in general, the problems with inapplicable data cannot be solved by single-character algorithms. This is so because different sets of plausible biological assumptions (that may reflect different empirical findings) can lead to different multicharacter hierarchies. These, in turn, as is the case here, can lead to different results.

- `example_7.Brazeau_et_al_2017.fig3-alt2.ana`
  An analysis of Brazeau et al.'s (2017) Fig. 5 under the same assumptions as in example 5 but with an additional character hierarchy added. This illustrates another reason why, in general, single character algorithms are theoretically insufficient to analyze datasets with inapplicable data. Single-character algorithms are by definition unable to find out if 'absence' in two characters with the same distribution of 'absence' refers to absence of the same structure or to absence of two different structures. This is empirical information that can influence tree scores. It is readily incorporated in an algorithm that explicitly optimizes complete user-defined character hierarchies, as is the case in anagallis, but not in single character algorithms.

- `example_8.a_more_complex_optimization.ana`
  A simple example where a single character in a character hierarchy has multiple regions of absence that can be optimized independently. This poses some problems for vizualization of results. Several similar examples are now included in the built-in documentation.

Assuming that the executable has been renamed '`anagallis`', example file '`scriptfile`' can be executed using

```
./anagallis sestaf scriptfile
```

this is short for

```
./anagallis scriptfile execute start f scriptfile
```

Some example files contain statement

```
scriptfile execute pauze
```

This can variously be abbreviated as '`sep`', '`scr ex p`', '`scep`', ... and returns control to the console.

To resume execution of the script file, enter

```
ser
```

This is short for

```
scriptfile execute resume
```

Some example files include statement

```
pq
```

(short for '`program quit`').

This quits the program and by including it in a script file the program can be run in full batch mode. There are several ways to produce outputfiles for that purpose as well.

There are two commands that can be used to document script files: '#' and '"'.

'#' basically is just a comment that is skipped during execution.

'"' is similar, but the text following the double quote command is echoed to the output during execution. It is useful to put comments in the that output.

Anagallis can read multiple anagallis statements from the command line, using '`;`' as command separator.
So an alternative way to run the program in full batch mode is to include '`pq`' on the commandline.
The '`;`' that separates anagallis commands must somehow be escaped then. For example:

```
./anagallis sestaf scriptfile\; pq
```

or

```
./anagallis 'sestaf scriptfile; pq'
```

To get an overview of available commands, enter

```
help summary
```

To obtain information about command `xyz`, enter

```
? xyz
```

To get more information about the '?' command itself, for example, use

```
??
```

Command '?' is a synonym for '`help command -`', so the same result is obtained with

```
help command - ?
```

With the full form of the command, a number of options are available that are not present for '?'.

Commands are hierarchically structured. For example

```
? trees
```

will give more information about command '`trees`', including a list of subcommands such as '`trees search`'. More information about command '`trees search`', including a further list of subcommands, can be obained like this:

```
? trees search
```

And more information about command '`trees search mult`', including a description of its options, like this:

```
? trees search mult
```

An advantage of the long form of the help command is that it has a number of options.

```
hc r - trees
```

for example, will recursively show the documentation of command '`trees`' and all its subcommands.

All built-in docs can be dumped to a file `abc` (overwrite mode) using

```
heduofabc
```

This is short for

```
help dump o f abc
```

Different modes of collapsing zero-length branches can be set using command '`trees set zerocollapse`'. Default mode is 2, corresponding to Nona's `amb-`.
More info with

```
?trsetz
```

Tree plotting by default uses UTF-8 encoded multibyte unicode characters.
This can be problematic in consoles that do not support unicode.
Command

```
program set unicode –
```

can be used to switch to ascii graphics in such cases.

Brazeau, M. D., Guillerme T., Smith, M.R. 2017. Morphological phylogenetic analysis with inapplicable data. BioRxiv preprint first posted online October 26, 2017. doi: http://dx.doi.org/10.1101/209775.

De Laet, J. 2005. Parsimony and the problem of inapplicables in sequence data. Pp. 81-116 in Albert, V.A. (ed.). Parsimony, phylogeny and genomics. Oxford University Press, ISBN 0-19-856493-7.

De Laet, J. 2012. Theory and practice of parsimony analysis when some characters are inapplicable in some terminals. Presentation at the XXXIth Meeting of the Willi Hennig Society, Riverside, California, USA. DOI10.13140/RG.2.2.12159.51363.

De Laet, J. 2015. Parsimony analysis of unaligned sequence data: maximization of homology and minimization of homoplasy, not minimization of operationally defined total cost or minimization of equally weighted transformations. Cladistics 31: 550-567.

De Laet, J. 2017. A note on Brazeau et al.'s (2017) algorithm for characters with inapplicable data, illustrated with an analysis of their Fig. 3d using anagallis, a program for parsimony analysis of character hierarchies. Technical report first posted online November 5 2017. DOI10.13140/RG.2.2.31309.54245.

Strong, E. E., Lipscomb, D. 1999. Character coding and inapplicable data, Cladistics 15, 363–371.

Maddison, W. P. 1993. Missing data versus missing characters in phylogenetic analysis. Syst. Biol. 42, 576-581.

Enjoy!

Jan De Laet
Veltem-Beisem
8 September 2020