MAPP.JAR - MULTIVARIATE ANALYSIS OF PROTEIN POLYMORPHISM

Readme. Last updated on 6/28/2005. Copyright Eric Stone and Arend Sidow, 2005.

OVERVIEW

MAPP reads an alignment of protein sequences and a tree relating the sequences. It then calculates the predicted impact of each potential SNP at each position. The predictions are based on a set of scales (of physicochemical properties) for which each amino acid has a numeric value.

MAPP requires Java 1.4 or higher. You must be familiar with running programs from the command line. Because MAPP is written in Java, it should work on any platform.

MAPP requires evolutionary variation. You cannot run MAPP without a reasonably large number of homologs, preferably orthologs.

Before you use MAPP, you need to do your homework. MAPP is a powerful tool, but if you put garbage in, garbage will come out. Here is the homework you need to do before you start using MAPP:

1. Read the MAPP paper and understand it. It is available at Genome Research or on the SidowLab web site.
2. Find orthologs of your protein of interest. MAPP assumes that the evolutionary variation present among the compared sequences is consistent with the protein's function. If you use paralogs whose function has drifted you will be introducing variation that specifies functional differences, which will compromise MAPP's predictive power (see paper).
3. Build a good alignment of your protein of interest with those orthologs. You need orthologs (or closely related paralogs whose functions have not changed) that give you substantial point variation in the alignment. We like ProbCons, but you can use ClustalW or other alignment programs if you want. Remember: it's alright to throw out sequences that poison your alignment. Once you have a good alignment you can build a tree from it.
4. Build a tree from the alignment. You can use a ClustalW tree, but it's better if you build a better tree using, for example, Semphy. The tree captures correlations among the sequences that MAPP needs to know about to calculate appropriate summary statistics about the evolutionary variation present in the alignment.

MAPP is open-source. Feel free to unpack the jar and tweak the code, but be aware that any tweaks may have unintended consequences. We are actively working on improving MAPP. MAPP.jar is bundled with classes from third-party libraries Jama (Java matrix library) and Jsci (math and statistics). Leave in all Copyright notices if you redistribute MAPP.

INPUT

General notes

MAPP requires a text file containing the alignment in Fasta format. You specify the file name using the -f option.

MAPP requires a text file containing the tree in parenthesis representation, with branch lengths. You specify the file name using the -t option.

Warning: The sequence identifiers (names) must be unique and match *exactly* in the two files. Make sure you have no trailing whitespace after the names and do not use special characters such as dots, brackets, parentheses, semicolons, etc. Underscores are ok.

Warning: The program is finnicky with input format. Please make sure to have no empty lines in the input files.

Alignment file

You must make sure that the sequences really are aligned. MAPP does not do alignments for you. The Fasta file should follow standard Fasta format:

```
>Seq_1_Human
aligned human sequence...
>Seq_2_Chick
aligned dog sequence...
...
```

Gaps in the alignment must be indicated by hyphens.

We apologize to those of you who are used to alignment formats other than Fasta. You must convert your alignment into Fasta format.

Tree file

The tree file must follow this format:

```
(Seq_1_Human:0.2,Seq_2_Chick:0.3,(Seq_3_Zfish:0.1,Seq_3_Fugu:0.1):0.35);
```

Note that this is an unrooted tree. If you give MAPP a rooted tree it will remove the root.

MAPP finds the first open parenthesis in a tree file and then finds the semicolon, which must be present after the last closing parenthesis. Everything between those landmarks it considers to belong to the tree. The tree can be on a single line (like in Semphy or Phylip output) or span multiple lines (like in ClustalW output). MAPP will count opening and closing parentheses, and if those numbers don't match it will throw an error. If there is anything else wrong with your tree, MAPP will simply bail out and say that it can't construct a tree from the supplied file.

The IDs in the tree must exactly match the IDs in the Fasta file. For example, Seq_1Human would not be the same as Seq_1_Human, and MAPP will bail. And if there's space between the Seq_1_Human and the ":" in the tree file, that means that there's trailing whitespace which may or may not be present in the alignment identifier.

The numbers after the colons are branch lengths, usually in substitutions per site. The program you use to build a tree should have an option that gives you branch lengths, and a tree format that should be compatible with MAPP. Without branch lengths, MAPP does not work. MAPP needs branch lengths.

Sample files

You will find two sample files on the SidowLab web site, off the same page where you found MAPP.jar and this Readme:

LacI_Alignment.fa
LacI.tree

We suggest you first run MAPP on those files to make sure it works before you use your own files.

PHYSICOCHEMICAL PROPERTIES

The physicochemical properties used by MAPP are:

1. Hydropathy
2. Polarity
3. Charge
4. Volume
5. Free energy in alpha-helix conformation
6. Free energy in beta-strand conformation

You can choose to run MAPP using only a subset of these properties. To do this, use the -s option followed by the numbers of the desired properties, separated by colons. For example, to use only Hydropathy (first), Charge (third), and Volume (fourth), specify "-s 1:3:4". If you do not specify the -s option, the program will use all scales.

We note that MAPP properly corrects for the correlations among these properties and that you should not have to drop properties. We also caution that dropping properties to go fishing for "better results" is statistically inappropriate.

OUTPUT

MAPP produces an output table in which each row corresponds to a position (column) in the alignment. Output goes to screen unless a file is specified with the -o option. The following data are provided for each position:

Header	Excel Column	Description
Position	A	Position of column in the alignment
Column score	B	Median MAPP score for this alignment column.
Column p-value	C	P-value interpretation of the column's median MAPP score. Note: P values are output in scientific notation, with a trailing "E-x". Excel will correctly parse this.
Alignment	D	Amino acids observed in that column, in alignment order. Note: we put a single tick (') as the first character so that Excel does not interpret a leading dash as a minus sign.
Gap weight	E	Weighted fraction of gaps in column.
Over gap weight threshold	F	Flag indicating whether the column is over gap weight threshold. MAPP will not calculate scores if the gap weight of a column is greater than 50%
Hydropathy, Polarity, ..., Free Energy Beta	G-L	P-values corresponding to the significance of each physicochemical property. A low P-value means that the column exhibits strong contraint for this property.
A, C, D, ... , Y	M-AF	MAPP scores for each possible amino acid variant.
A, C, D, ... , Y	AG-AZ	P-value interpretations of the MAPP scores, predicting the impact of each amino acid variant.
Good amino acids	BA	List of amino acids predicted to not substantially impair protein function at this position.
Bad amino acids	BB	List of amino acids predicted to be deleterious at this position.

We recommend importing the output into Excel, which will allow straight-forward manipulation of the data. Alternatively, writing a parser in Perl or another language is straight-forward as well, as the output is tab-delimited.

If you drop properties with the -s option the amino acid columns will shift to the left relative to the layout described here.

If the column gap weight is greater than 50%, MAPP will not produce predictions for this position in the alignment and output "na" in all fields except Position, Alignment, Gap weight, and Over gap weight threshold.

COMMAND-LINE OPTIONS AND HOW TO RUN

To run the program on the test files, execute the following command:

java -jar MAPP.jar -f LacI_Alignment.fa -t LacI.tree -o LacI_output.xls

In red is the execution of the jar. Blue are the required options. Specifying -o is not necessary.

Valid command-line options:

Option and parameter	Required?	Explanation
-f <path_and_alignment_file_name>	Yes	path to text file containing sequence alignment in Fasta format
-t <path_and_tree_file_name>	Yes	path to text file containing tree with branch lengths
-s <scales_to_be_used>	No	column numbers of scales to use, separated by colons (see above)
-o <output_file_name>	No	path to output file
-h	No	display help

It is convenient to add the extension .xls to the output file so that the file opens in Excel, which will import the tab-delimited data and correctly interpret the scientific notation of the P-values. Note the presence of a tickmark as the first character in the alignment column to prevent excel from interpreting a leading gap character as a minus sign.