PQ user manual

Sergey Spirin

September 13, 2016

PQ (positions-quartets) is a program for reconstruction of phylogeny of biological sequences. It inputs a multiple sequence alignment in fasta format and outputs an unrooted phylogenetic tree without branch lenghts, in Newick format.

1 Downloading and usage

1.1 For Windows users

The executable file `PQ.exe` and a number of scoring matrices are available in the zip archive `http://mouse.belozersky.msu.ru/software/pq/pq-windows.zip` .

PQ for Windows is a command-line application. To run it, copy the file `PQ.exe` to your working folder or to a folder listed in PATH environmental variable. Open a command line window (`cmd`), activate your working folder and type a command.

For example, to reconstruct phylogeny of sequences whose alignment is in file `cyb5.fasta` you may execute the following command

```
PQ -alignment cyb5.fasta
```

As a result of execution, the file `pq_out.tre` will be created with a phylogenetic tree in Newick format. The tree may be visualized by any tree visualization software, i.e., MEGA.

The recommended variant for protein sequences is as follows

```
PQ -alignment cyb5.fasta -pwm BLOSUM62.txt -out cyb5.tre
```

that means use BLOSUM62 scoring matrix and put the results into the file `cyb5.tre`. Note that the file `BLOSUM62.txt` should be in the working folder.

1.2 For Linux, Cygwin, etc.

Download `http://mouse.belozersky.msu.ru/software/pq/pq.tar.gz` . Unzip, extract and compile:

```
gunzip pq.tar.gz
tar xf pq.tar
make
```

Copy the executable file `pq` to your working folder. For protein alignments it is recommended to use the scoring matrix `BLOSUM62.txt`. To do this, copy this file from the folder `matrices` to your working folder, too.

Prepare alignment of your sequences in fasta format. Let the name of the file with alignment be, for example, `myprot.fasta`. Then to reconstruct phylogeny with PQ, you may run the following command:

```
./pq -alignment myprots.fasta -pwm BLOSUM62.txt -out myprots.tre
```

the result will appear in the file `myprots.tre`.

Available parameters of the program see in the section 3.

2 Tree score

The reconstructed by PQ phylogenetic tree is a result of optimization of the tree score calculated at the base of input multiple alignment.

The formula for the tree score is as follows:

$$Q = \sum_{c,q} Q_{cq}$$

where c runs through all positions of the alignment and q over all quartets of sequences (PQ is for "positions-quartets"). A quartet is a four-element subset $q = \{i, j, k, l\}$ of sequences, thus it does not depend on the order of these four sequences. In the following formulas the order is presumed to be consistent with the topology of the tested tree, namely i and j are separated from k and l by at least one branch (split) of the tree.

The formula for Q_{cq} is as follows:

$$Q_{cq} = \begin{cases} 0 & \text{if } S(a_{ic}, a_{jc}) < X_{cq} \text{ and } S(a_{kc}, a_{lc}) < X_{cq} \\ S(a_{ic}, a_{jc}) - X_{cq} & \text{if } S(a_{ic}, a_{jc}) > X_{cq} \text{ and } S(a_{kc}, a_{lc}) < X_{cq} \\ S(a_{kc}, a_{lc}) - X_{cq} & \text{if } S(a_{ic}, a_{jc}) < X_{cq} \text{ and } S(a_{kc}, a_{lc}) > X_{cq} \\ \alpha(S(a_{ic}, a_{jc}) + S(a_{kc}, a_{lc}) - 2X_{cq}) & \text{if } S(a_{ic}, a_{jc}) > X_{cq} \text{ and } S(a_{kc}, a_{lc}) > X_{cq} \end{cases}$$

where a_{ic} is the letter of the sequence i in the position c of the input alignment, S is the scoring matrix, $\alpha > 0$ is a parameter of the program and

$$X_{cq} = \max\left(S(a_{ic}, a_{kc}), S(a_{ic}, a_{lc}), S(a_{jc}, a_{kc}), S(a_{jc}, a_{lc})\right)$$

3 Description of parameters

`-h` or `-help` or `--help`

Produce the list of parameters with short descriptions and exit.

`-alignment <FileName>`

Input alignment, required. The file called `<FileName>` must contain a sequence alignment in fasta format.

`-out <FileName>`

Output file: phylogenetic tree in Newick format. Default value: `pq_out.tre`.

`-iniTree <FileName>`

Initial tree, *i.e.* a starting point to search a tree with best PQ score. Must be in Newick format with leaves labeled by sequence names of the input alignment. An optional parameter. By default, the initial tree is formed by growing (stepwise addition). See `-grType` below.

`-pwm <FileName>`

The scoring matrix. If this parameter is not given, all calculations are performed with the diagonal matrix. The scoring matrix must be square (not triangular) and contain integer numbers. In our tests, for prokaryotic protein sequences the best results are obtained with the matrix `BLOSUM62.txt`, for eukaryiotic sequences — with the matrix `FungiMatrix.txt`. The latter is calculated in a manner close to the algorithm of BLOSUM using a set of alignments of fungal proteins.

`-alpha <integer>`

Multiplying coefficient α for position-quartet pairs that support the given topology twice (see the previous section). By default, $\alpha = 1$. In this implementation, α can be any positive integer. Tests show that $\alpha = 2$ and $\alpha = 3$ give slightly different but in average equally accurate results comparing with $\alpha = 1$. Greater values of α give worse results.

`-gapOpt <0|1|2>`

How to score gaps.

0 means ignore gaps, this is the default value.

1 means score only position-quartet pairs that contain at most one gap.

2 means score all position-quartet pairs. For example if sequences i and j contain gaps in the position c and sequences k and l contain different letters, then (with the unity matrix $S(a, b) = \delta(a, b)$) the PQ score Q_{cq} will be equal to 1.

`-grType <one|multiple>`

If the initial tree is not given, then the starting point for the optimization of score is made by tree growing, or stepwise addition of sequences. This procedure can be performed once or several times, in the latter case each time the input order of sequences is randomly shuffled and the result is the tree with a maximal score. The default value is `multiple`.

`-randLeaves <0|1>`

This parameter has an effect only if `-grType one` is set. 0 means do not shuffle input order of sequnces before tree growing. The default value is 1.

`--treeNum <integer>`

This parameter has an effect only if `-grType multiple` is set. It means the number of times to perform tree growing. The default value is 10.

`-nniType <none|simple|direct|trajectory>`

This parameter regulates the type of optimization using nearest-neighbor interchange (NNI).

`none` means do not perform NNI optimization.

`simple` and `direct` are two closely related "aromats" of NNI hill climbing. `simple` means perform subsequently all possible NNIs of the current tree and change the current tree immediately when any neighboring tree with a higher score is found. `direct` means first try all possible NNIs then if there are neighbors with higher scores change the current tree to the neighbor with the highest score. The results of two procedures are usually the same, as well as the computation time.

`trajectory` means perform Monte Carlo search instead of hill climbing.

The default value is `direct`.

`--trTime <integer>`

This parameter has an effect only if `-nniType trajectory` is set. Its value is the number of steps in Monte Carlo search. The default value is 1000.

`--initTemp <integer>`

This parameter has an effect only if `-nniType trajectory` is set. Its value is the initial "temperature" T_{ini}. The temperature has slightly different meanings for different styles of Monte Carlo search (see `--mcStyle` below). After each step, the temperature is decreased by T_{ini}/N, where N is the number of steps (see `--trTime` above). The default value of initial temperature is 1000.

`--mcStyle <0|1|2>`

This parameter ("MC style") has an effect only if `-nniType trajectory` is set. It regulates the manner of Monte Carlo search.

If the MC style is set to 0, then the procedure of Monte Carlo search is as follows. At each step, there is a current tree and a current temperature T. Compute the score Q_{new} of the next NNI neighbor of the current tree. If this score is higher than the score Q_{old} of the current tree, then set this neighbor as the current tree and continue with the next NNI. If $Q_{\text{new}} < Q_{\text{new}}$, then with the probability

$$P = \exp\left(\frac{K}{T} \cdot \frac{Q_{\text{new}} - Q_{\text{old}}}{Q_{\text{old}}}\right)$$

where $K = 12000000$, set this neighbor as the current tree and with the probability $1 - P$ leave the old current tree. After each step reduce T by T_{ini}/N, where T_{ini} is the initial

temperature (see `--initTemp`) and N is the number of steps (see `--trTime`). Stop when T reaches zero. Output the tree with the highest score among all tested.

If the MC style is 1, the procedure is in general the same as for the case 0, but after each change of the current tree the order of possible NNIs is randomly shuffled and the testing of all NNIs is started from the beginning (for MC style 0 it is continued without shuffling, presuming that some cyclic order of all possible NNIs is fixed and any neighboring tree "inherits" this order in some natural manner).

If the MC style is 2, the procedure is different. Also there is a current tree and a current temperature T. For the current tree and all its NNI neighbors compute the weight $w(t)$ (where t is a tree) equal to

$$w(t) = \exp\left(\frac{K}{T} \cdot \frac{Q_{\text{new}} - Q_{\text{old}}}{Q_{\text{old}}}\right)$$

where Q_{new} is the score of the tree t and Q_{old} is the score of the current tree. Thus for the current tree $w(t) = 1$, for neighbors with score higher than of the current tree $w(t) > 1$ and for neighbors with score lower than of the current tree $w(t) < 1$. Set $Z = \sum_t w(t)$ and for each tree compute the probablility $P(t) = w(t)/Z$. Randomly choose one tree using these probabilities and replace the current tree with the chosen tree (so the current tree remains the same with the probability $1/Z$). Reduce the temperature by nT_{ini}/N, where n is the number of neighbors. Repeat all procedure while $T > 0$.

Tests show approximately equal effectiveness of all three MC styles.

The default value of MC style is 0.

`-sprType <none|simple|direct|trajectory>`

This parameter regulates the type of optimization using subtree pruning and regrafting (SPR).

`none` means do not perform SPR optimization.

`simple` and `direct` have the same meanings as for `-nniType`, just replace NNI with SPR. If `-nniType` is not `none`, then the initial tree for SPR search is the result of NNI search.

Tests show that for relatively large alignments (30 sequences and more) SPR search gives in average more accurate reconstruction than NNI search, but requires a lot of time.

The default value is `none`.

`-neiZscore <0|1>`

Setting 1 switches on the calculation of Z-score of the result with respect to NNI neighbors. It means: calculate mean M and standart deviation σ of scores of all neighbors and set

$$Z = \frac{Q - M}{\sigma}$$

where Q is the score of the result tree. The value of Z is printed to the standard output.

The default value is 0.

`-randTreeZscore <0|1>`

Setting 1 switches on the calculation of Z-score of the result with respect to random trees. It means: calculate mean M and standart deviation σ of scores of n random trees, where n is the value of `--randTreeNum` and set

$$Z = \frac{Q - M}{\sigma}$$

where Q is the score of the result tree. The value of Z is printed to the standard output.

The default value is 0.

`--randTreeNum <integer>`

The number of random trees for Z-score calculation. The default value is 10.

`-distrFile <FileName>`

Setting this makes sence if either `-neiZscore` or `-randTreeZscore` is set to 1. To the file with name `<FileName>` the scores of neighbors and/or random trees are written.

Setting `-distrFile stdout` leads to adding the mentioned scores to the standard output.

4 Example and comments on output

Let the command be:

```
> PQ -alignment cyb5.fasta -pwm BLOSUM62.txt -out cyb5.tre
```

Because parameters `-grType`, `--treeNum`, `-nniType` and `-sprType` are not set, the default values are used: `multiple`, 10, `direct` and `none` respectively. That means: shuffle the order of sequences ten times, use each order for tree growing by stepwise addition, among the resulted trees choose one with the highest score, use this tree as initial in NNI hill climbing.

The output is as follows:

```
Growing 10 trees by stepwise addition
Tree number 1, score = 8032572
Tree number 2, score = 8062798
Tree number 3, score = 8041825
Tree number 4, score = 8047277
Tree number 5, score = 8041435
Tree number 6, score = 8040438
Tree number 7, score = 8061877
Tree number 8, score = 8048920
Tree number 9, score = 8053359
Tree number 10, score = 8060753
Performing gradient NNI hill climbing, initial score is 8062798
```

```
Tree with score 8062821 found
Tree with score 8063383 found
Tree with score 8063406 found
Tree with score 8063604 found
Tree with score 8063627 found
No better trees found
Score of result tree is 8063627
Relative score of result tree is 0.994713
```

The output consists of three parts: the first contains information about trees growed by stepwise addition, the second about the NNI optimization process, and the last line contains "relative score". This relative score is calculated as follows. For each quartet $q = \{i, j, k, l\}$ and each position of the alignment c calculate the maximal score $Q_{cq}^{\max}$ as maximum of three possible Q_{cq} obtained with three possible splitting of q onto two two-element subsets. Thus $Q_{cq}^{\max}$ does not depend on tree topology, it is the maximum contribution that this position of aligment can give with these four sequences. Set

$$Q^{\max} = \sum_{cq} Q_{cq}^{\max}$$

and the relative score is $R = Q/Q_{\max}$. With a fixed alignment, R is proportional to Q. However R is much more convinient for comparing reconstructions made with different alignments and thus for estimation of accuracy of the reconstruction. Tests show that R has a significant negative correlation with the distance between the reconstructed tree and the real tree of a set of sequences.

Licence and Disclaimer