PREQUAL: the manual

FAQs

What is PREQUAL for?

PREQUAL (PRE-alignment QUALity filter) provides a command line tool for examining a group of homologous sequences and filter (remove or mask) characters or stretches within individual sequences that are unlikely to share a common ancestor (homology) with any other character in any other sequence in the group. In practice this filtering targets stretches of the sequences that could be considered a sequence specific insertion, including sequencing errors, inversions, frameshifts, and anything else that means that any stretch of a sequence has no simple homology with a stretch in any other sequence. PREQUAL works on unaligned sequences, in other words it filters sequences ***before*** multiple sequence alignment. It uses a sophisticated probabilistic modelling approach that does not assume any fixed sequence alignment. PREQUAL works with amino acid sequences, but can also handle DNA sequences of protein-coding genes.

What is PREQUAL not for?

PREQUAL does not work with aligned sequences. PREQUAL is not a tool to filter unambiguously aligned columns from multiple sequence alignments. It will not remove regions of alignment uncertainty, which can be the result of poor alignment, if there is evidence of shared homology somewhere else in the sequence. There are many other good programs available for this problem and we recommend looking at our related divvier program (https://github.com/simonwhelan/Divvier) if this is the problem you want solved. By removing likely non-homologous sequence stretches we believe PREQUAL eases subsequent multiple sequence alignment problem.

Does that mean PREQUAL helps multiple sequence alignment?

We have found that filtering with PREQUAL prior to making alignments seems to improve the quality of the resultant alignments, but we have no hard evidence for this effect. This improvement could occur because individual sequence errors are not part of the underlying model used in alignment programs, such as Clustal or MAFFT, so removing them could make the remaining homologies easier to identify. If you find PREQUAL helps in this way please let us know and share your findings!

Why might PREQUAL filter genuinely homologous stretches or miss an error?

There are many reasons why PREQUAL might make a mistake. PREQUAL is a type of classifier that divides characters in a sequence into real homologies and errors. These types of classifiers require a trade-off between true positives (TPs: errors) and true negatives (TNs: keeping real sequence). It's easy to get 100% TPs by removing all the sequence or to get 100% TNs by not removing any sequence, but what we want is the best possible balance between them by choosing an appropriate threshold. We have chosen what we believe to be a

useful threshold based on real data and some simulations, but real errors will inevitably slip through and some real sequence will be labelled as an error. ROC curves from simulations (see Supplementary Material for the paper) suggest PREQUAL does very well in this trade-off, but remember mistakes are inevitable.

Other reasons for mistakes might include a violation of some of the underlying assumptions in PREQUAL and we note two cases here. The first occurs when an error introduces the same (or very similar) sequence at similar relative locations in different sequences, which could occur when primers are incorporated in the same place or when the same intron is mistakenly included the gene model of different species. PREQUAL assumes such high similarity can only arise due to shared descent so will infer that these errors as homologous. The second case occurs in very fast evolving sequences, which are difficult to distinguish from random and therefore difficult to align. PREQUAL may not be able to find shared descent, so could remove them from your analyses.

Should I change the default threshold in PREQUAL?

We derived our default recommended threshold of 0.994 from looking at lots of phylogenomic data and simulations. We found that at this point the overwhelming majority of errors identifiable by eye appear to have been removed and simulations suggest >90% of errors will be removed while retaining >95% of the original data. Although our tests suggest this is a good value, if it's not working to your satisfaction you can change the thresholds to match your needs. We have found in real data that 0.99 might remove fewer true homologies while capturing most errors, and in some cases even 0.95 seems to work rather well. Please ensure you report changes to the threshold so others can replicate your research.

Should I use PREQUAL on repetitive elements?

Short answer is no. Long answer is maybe, but be warned that PREQUAL is designed to work on simple protein-coding homologous sequences and presence of repetitive sequences might affect its efficacy. Whenever homologous regions ***within*** an individual sequence exist, such as repeated domains and other tandem repeats, establishing the homology between the repeats becomes more challenging. Any evolutionary analysis based on sequences with repetitive elements is risky and we leave it to the user to decide whether PREQUAL adequately works in these cases. Some basic measures are put in place to capture the presence identical repeats within a sequence, which can also occur due to poor gene models or other errors.

How do I report a bug in the program?

Contact simon.whelan@ebc.uu.se or fabien.burki@ebc.uu.se with the details of the bug, including any screen text and the input file on which the bug occurred. We'll try and fix it, but it might take some time.

I've used the program, but how do I cite it?

Thanks for using PREQUAL! Please cite:
PREQUAL: a program for detecting errors in sets of unaligned homologous sequences. Whelan, Irisarri & Burki. *Submitted*.

Basic usage of PREQUAL

PREQUAL is designed to be as simple and lightweight to use as possible, but also provide the user with many options for fine-tuning analyses if required. Basic usage is first described, followed by the description of all of the available options.

Installation

PREQUAL can be build from source ('make'). Alternatively, binaries are available for Linux, OS X and Windows. Please select the binary that best fits your operating system and place it in a place you can access from the command line. If there is no suitable binary available, please contact us.

The command line

PREQUAL runs on the command line. The majority of users will find the basic usage sufficient. Assuming that PREQUAL is in the current directory, type:

```
./prequal my_sequence_data.fasta
```

This will run PREQUAL on the input file `my_sequence_data.fasta` and provide basic output and error checking. By default, PREQUAL will create two output files:

`my_sequence_data.fasta.filtered:` main output: filtered sequences
`my_sequence_data.fasta.PP:` internal file containing calculated posterior probabilities. The PP file can be used to re-run PREQUAL (e.g., varying its settings) without the need of re-computing posterior probabilities.

Sometimes, PREQUAL will output warnings to:
`my_sequence_data.fasta.warning`

Input files

The sequence file should be in simple Fasta format, which for each sequence includes a line specifying the name prefixed by '>' and a number of lines defining the sequence, which should not contain stop codons (*). Below is an example of a Fasta format file:

```
>Sceloporus
MHTTLSTTTMLTAMVILAAPIILTTQNSPNYTKNVKFALQTTCLITTIP
AMLFINYGEELALTNFSLLLISNFNIKISFIMDMYSLSFTPIALFVTWSILEFSTWYMAA
DPHINKFFKYLLIFLIAMITLITANNLFQFFIGWEGVGIMSFLLIGWWYSRADANSSALQ
AIIYNRIGDIGLILTIAWLTANTPTWQIQENFLYNINNLVPMLGLILAATGKSAQFGLHP
WLPAAMEGPTPVSALLHSSTMVVAGVFLLIRIHPIIQTNQTALTICLCLGSMTTAFTAIC
ALTQNDIKKIVAFSTSSQLGLMMVTIGLNQPLLAFMHISTHAFFKAMLFLCSGSIIHNLS
NEQDIRMMGGIANTMPTTSSCLTIGSLALMGTPFLAGFYSKDTIIETLNNSHLNAWALLM
TILATMLTAAYSLRIMFYVQMKSMRHKPLINIDENTKPLINPIIRLALGTLISGLLITTA
ILPTKTIQMTMPISTKLMAISITVIGLMLALDISNQTTTLMPTKQTTYYQFSNQLGFFN
LLHRDVPNMMLKTSQKTATQMNDLLWLEKLGPKGLSTSQLPMINLSSSQKGLIKNYLLTF
ITTTALFAILSIX
>Pogona
VTMMILTLQLTTIFIMTTPLLPKWGGRAPISVKTAVKMAFITSLIP
STLMLKYQAQPLSTLFPISTKPDMCLTITLNHFSALLLPVVLFVAWSIMEFATWYISP
TPLTKTFTSALLIFLLAMVILICAGNLFQLFIGWEGVGIMSFILINWWTSRTTTNSAALQ
AMIYNRIGDIGLILAISILAMHYSTWDLVQSTAQQTKDMLLAMGLTLAAVGKSAQFFMHM
WLPTAMEGPTPVSALLHSSTMVVAGIYMLAQLHPLLNASKHILTLCLYLGATTSLFTASC
ALAQNDIKKVIAFSTSSQLGLMMTAIGIGSPGLAIFHMATHATFKATLFLSAGSIIHCMQ
NEQDIRKMGNSSTTMPITTTCLTINSLTLAGIPFLSGFYSKDAILETMTNSHLSSWALLM
TIMATTMTSAYTLRMLIYTTAASPRHKPYTHLHESESNQVSPILRPTILTIILGLTLST
IFPAQPTTPLPTTLKLIPILTILIGTTLTIDLTNTSLTPTPQKYAPYKTWNQLAFYGI
MLHRSFSFMALKLSQASTQLIDLIWLEKSGPKYLHSTNTEISKLTTAQTGLLKNYLIIF
LIFTSTLITIYHLTKYYX
```

PREQUAL typically works with amino acid sequences, but it can also take DNA sequences of protein-coding genes as input (provided that all codons are complete and in the right reading frame). To use PREQUAL on DNA please read the 'Using PREQUAL on DNA sequences' section below and pay special attention to any warnings in the output files.

Typical screen output

Below is an example of the kind of information you might see on screen after running PREQUAL:

```
------------------------------------------------
        PREQUAL v.1.00  by Simon Whelan
------------------------------------------------
A There are 91 sequences of max length 718
  Prepping pairHMM ... done
  Collecting subset of posterior probabilities based on closest 10 sequences determined by Kmers
          This may take time for larger data sets:                B1
B
  Creating collection sets of PPs based on Kmer distances
  \ 91 / 91 ... done                  B2
  Getting posterior probabilities:
   ... Done Get Posteriors
  Outputting posterior probabilities to EXAMPLE1_PAPER.fasta.filtered.PP... done B3

  Performing filtering:
          Applying standard threshold 0.99 resulting in 1530 residues removed C1
C         Extending filtered regions with width of 10  ... 11 additional regions removed C2
          Applying front/back trimming for runs of 3 resulting in 33 sections removed C3

  Outputting results:
    D Outputting filtered amino acid sequences to EXAMPLE1_PAPER.fasta.filtered

  Computation complete

  ====================== Summary =======================
                    Original    Filtered   %Retained
E #Sequences              91          91        100%
  #Residues            50257       48666       96.8%

F Analysis may have some problems. Warnings output to EXAMPLE1_PAPER.fasta.warning

  Filtering complete!
```

A This line provides a summary of your data. Check this matches your input.
B This section provides information about the computation of posterior probabilities that are core to the PREQUAL methodology
B1 PREQUAL uses a heuristic that compares each sequence only to similar sequences with adequate coverage. The information here confirms the settings of that heuristic.
B2 The majority of the run time for PREQUAL will be spent calculating Kmer distances and posterior probabilities. Progress spinners will give you some indication of how long the program will take to run.
B3 Confirmation of output of the posterior probabilities. The .PP file can be used to re-run PREQUAL without the need of re-computing posterior probabilities in B2, in order to fine-tune its behaviour.
C The actual filtering step that results in the removal or masking of sequence characters.
C1 The filtering threshold and the number of residues that fall below this threshold.
C2 Multiple residues falling below the threshold in close proximity indicate a poor stretch. This section indicates how many of those stretches are joined together and filtered.

C3 PREQUAL defines N and C termini, which are to be fully removed from analyses, and 'core' regions that are masked with an 'X' character. This line summarises information about how those regions are extended.
D Name of the main output file.
E The summary of the output, showing the number of sequences and residues before ('Original') and after ('Filtered') filtering. Note that some sequences may be entirely removed! In which case a warning will be printed.
F In some analyses, like this one, you may see a warning file has been generated. This contains some important information and should be inspected. Some examples of what it may contain include:
1. When a large proportion of a sequence has been removed, for example:

```
WARNING: 86.94% of sequence removed/missing for [5] Seq1
```

2. When a entire sequence has been removed, for example:

```
WARNING: Fully removed sequence [5] Seq1
```

Using PREQUAL with DNA sequences

PREQUAL was designed for amino acid sequences. However, it can also take DNA sequences of protein coding genes, although it is your – the user's – responsibility to ensure the input contains complete codons, is in the correct reading frame and contains no internal stop codons. PREQUAL can be run directly on DNA sequences, which are first translated *in silico* upon automatic selection of the appropriate genetic code, filtered at the amino acid level, and the corresponding filtered DNA sequences generated. To avoid problems with the identification of non-homologous sequence stretches, DNA sequences are required to start and end with first and third codon positions, respectively. To run PREQUAL on DNA sequences:

```
./prequal my_sequence_DNA_data.fasta
```

The program will confirm DNA sequences by outputing to screen: "Found only DNA sequences. Doing translations." For output the program will generate the following files:

`my_sequence_DNA_data.fasta.dna.filtered` : this contains the filtered sequence data that you will want to use for your analyses

`my_sequence_DNA_data.fasta.filtered` : this contains the filtered sequence as amino acids.

`my_sequence_DNA_data.fasta.translation` : this file describes all the translation information, including the raw DNA sequence, the resultant amino acid sequence and the genetic code used to perform the translation.

`my_sequence_DNA_data.fasta.warning` : a warning file will also be likely generated, which may contain important information about your sequences. Please check this warning file and the translation file carefully to ensure that the filtering has been performed according to your expectations.

What PREQUAL does and does not do when working with DNA sequences

(+) PREQUAL searches only the first reading frame for each sequence, so it is your responsibility to ensure that all sequences are in correct frame.
(+) PREQUAL will test a range of genetic codes and pick a suitable one for translation. This choice is heavily guided by stop codons. Please check the translation file to ensure a reasonable genetic code it picked. Nevertheless, small differences in the genetic code should have little effect on sequence filtering.
(-) PREQUAL cannot deal with internal stop codons. You will need to remove these yourself.
(-) In case any ambiguous nucleotides are present (not A, C, G, T), the entire codon is will be ignored by PREQUAL and treated as gaps in the output.
(-) PREQUAL does limited checking of reading frames and internal stop codons, so please check any warnings.

PREQUAL options

Options affecting the core regions and filtering

A core region is defined as the central part of a protein that is relatively well conserved evolutionarily, and it can be flanked by less conserved regions attributable to N- and C- termini. By default PREQUAL defines the start and end of the core region as having three or more residues with high posterior probability (PP). Whenever a residue within the core region has low PP, it is masked with 'X'. Residues with low PP in the flanking (non-core) regions are simply removed. The following options affect this functionality:

`–corerun X`

Defines the number (X) of contiguous residues with high PP that are required to define a core region. Low values of X will make the program more generous at defining the core, whereas high values will make the program more conservative.

`–removeall`

Prevents the program from defining a core region and means that all low PPs residues will be removed (not masked with 'X', but completely removed from the sequence). We recommend using this option with caution. Masking, rather than simply removing, low PP residues within the core region facilitates the inference of the original positional homology among sequences. Complete removal can negatively affect multiple sequence alignment.

`–corefilter X`

Defines the character used to mask residues in the core region. By default this is 'X', but choosing another character might help with visualisation. Note that X it can only be a single character.

`–noremoverepeat`

By default PREQUAL will attempt to remove long identical repeats that can occur due to sequencing or assembly errors. When such a sequence is detected, a warning will be generated. Choose this option to stop this functionality.

Options dealing with protein coding DNA sequences

The following options affect how PREQUAL deals with protein coding DNA sequences.

`–nodna`

The program normally attempts to automatically detect whether the input contains DNA sequences and translate them. Choosing this option forces PREQUAL to read input as amino acid sequences.

`–forceuniversal`

By default PREQUAL attempts to choose the right genetic code for translating sequences by detecting codons that differ from the standard code. This option forces PREQUAL to always use the universal code for translation.

Options affecting output formats

These options affect the output files generated by the program and some naming conventions.

`–outsuffix X`

By default the program outputs filtered data to a file suffixed '.filtered'. Use this option to change that output suffix.

`–dosummary`

Output a broad summary file about your filtering, including information about the proportion of individual sequences removed and the amount of individual sequences defined as core region. This information will be output to a file suffixed '.summary'.

`–dodetail`

Generate a detailed summary file about your filtering. The output displays the PP for each residue and sequence, arranged into four columns:

> [Residue_number]Amino_acid : Indexing starts at 0
> Posterior_probability : Range (0,1)
> Whether_the_residue_is_removed : 0 = FALSE; 1 = TRUE
> Whether_the_residue_is_in_the_code : 0 = FALSE; 1 = TRUE

```
# [seq_pos]seq_character    maxPP   ToRemove    Inside
>Seq1
[0]F    0.5100  0   0
[1]G    0.8900  0   0
[2]D    0.9999  0   1
[3]N    1.0000  0   1
```

You could use this file to devise your own custom filtering scheme based on the PPs generated by PREQUAL.

`–noPP`

Do not output the posterior probability file. This will also force PREQUAL to recalculate PPs every run, in contrast to the default behaviour or reading in the corresponding .PP file if available.

Options affecting posterior probabilities and filtering

These options affect the heuristics used during the calculation of PPs and can affect the accuracy of the filtering. We have worked hard to choose values that work across a wide range of data sets, but inevitably some data sets will not work well for these values. (See FAQs.)

`–filterthresh X`

Specify a PP threshold for filtering. By default, every residue with PP<0.994 will be filtered. The most stringent threshold would be a threshold of 1.0, which will keep only the residues with absolutely only the highest confidence. The most liberal threshold of zero would not filter any residue. The default threshold of 0.994 was experimentally shown to perform well for a wide range of input data. This is the first of two main filtering approaches available in PREQUAL.

`–filterprop X`

Instead of filtering sequences by PP, the user can choose a proportion of the original data (X%) that is willing to loose in the filtering, and PREQUAL will adjust the filtering threshold accordingly. In practice, PREQUAL will often filter a little higher proportion of data because of the way regions of low confidence are joined together and how N- and C-termini are dealt with. This is the second of two main filtering approaches available in PREQUAL.

`–pptype X [Y]`

Specifies the algorithm to choose the subset of sequences to use when calculating PPs for each individual sequence. The default option is '–pptype closest', which compares each sequence against the 10 closest sequences defined by Kmer distance and also has mild coverage criteria. In some cases, you may wish to raise the default number of closest relatives Y to improve the accuracy of the PPs (e.g. '–pptype closest 20'). In other cases, '-pptype longest' might work better, which instead of evolutionary divergence choses the 10 longest sequences (the number of longest sequences may also be changed as above). We recommend to use this option with caution, because the longest sequences are often those containing the most errors. Finally, the option '-pptype all' will use all sequences. This option might be very slow, especially for data sets consisting of many sequences.

`–filterjoin X`

PREQUAL joins together low PP residues according to X in this threshold (default is 10). If there are fewer than X residues between two residues with low PPs, PREQUAL will filter all residues between them. Low values of X are more generous and will keep more data, whereas high values of X are more stringent and will filter more data.

`-nofilterlist X and -nofilterword X`

These two options define a file X that contains either a list of complete sequence names (i.e., corresponding to the Fasta headers) or a list of words that appear within these names that will not be masked during analysis. You may wish to consider this option if you have highly divergent sequences that might end up being heavily filtered because homology cannot be reliably assigned despite representing valid evolutionary data.