# FastQ Screen

# Contamination screening for NGS data

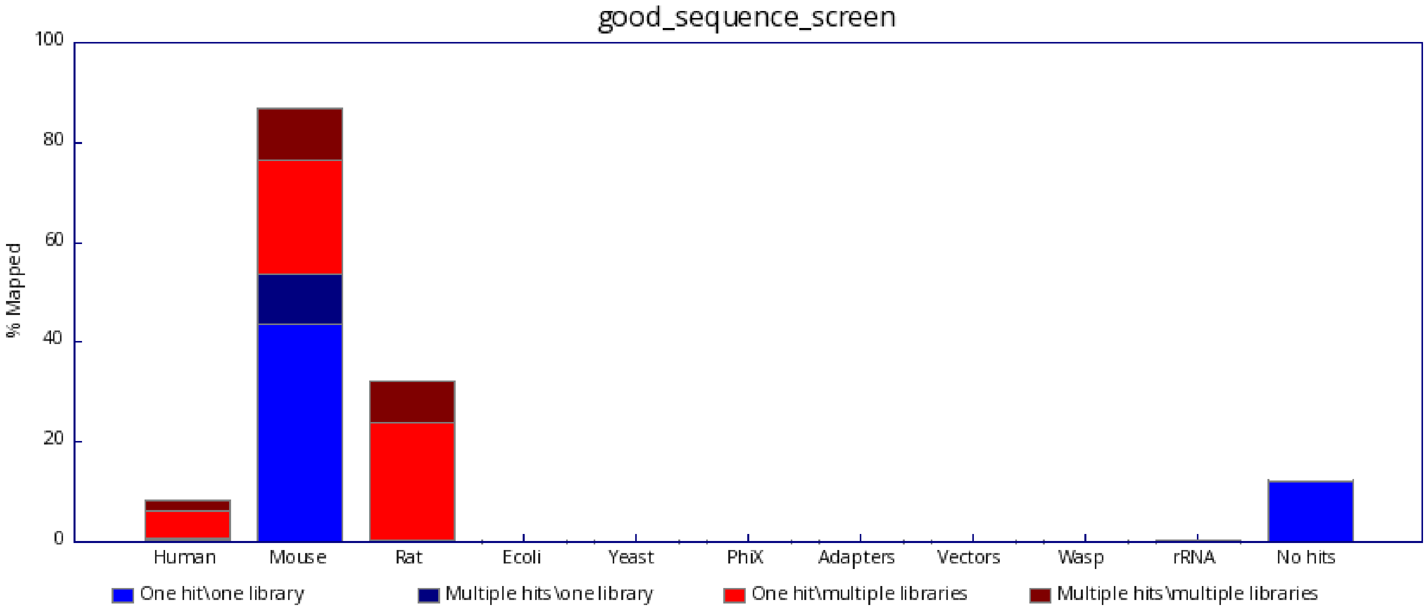
## Introduction

FastQ Screen is a simple application which allows you to search a large sequence dataset against a panel of different genomes to determine from where the sequences in your data originate. It was built as a QC check for sequencing pipelines but may also be useful in characterising metagenomic samples. When running a sequencing pipeline it is useful to know that your sequencing runs contain the types of sequence they're supposed to. Your search libraries might contain the genomes of all of the organisms you work on, along with PhiX, Vectors or other contaminants commonly seen in sequencing experiments.

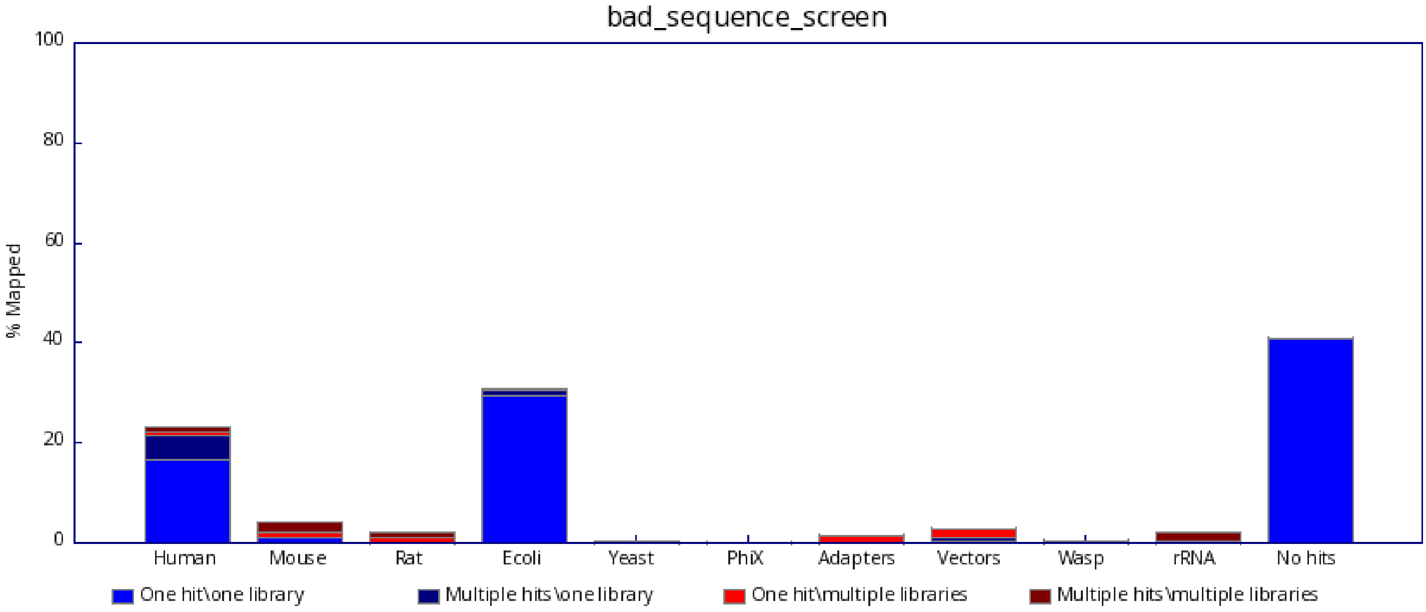
Although the program wasn't built with any particular technology in mind it is probably only really suitable for processing short reads due to the use of either Bowtie, Bowtie2 or BWA as the searching application.

The program generates both text and graphical output to inform you what proportion of your library was able to map, either uniquely or to more than one location, against each of your specified reference genomes. The user should therefore be able to identify a clean sequencing experiment in which the overwhelming majority of reads are probably derived from a single genomic origin.

***(Please note, in version 0.9.4 the graphs colour scheme changed from that shown below to a similar, but colour-blind safe colour scheme.)***



In contrast, poor sequencing results will include results from one or more unexpected species. Identifying such reads may help the user discover the source of the contamination.



**The FastQ Screen Homepage is at: http://www.bioinformatics.babraham.ac.uk/projects/fastq_screen**

# FastQ Screen online tutorials

To assist your understanding of FastQ Screen and how it should be used, we have prepared a series of short training videos.

Training video 1: Introduction to FastQ Screen

Training video 2: Downloading, configuring and running FastQ Screen

Training video 3: Interpreting FastQ Screen results

**We recommend watching these before using FastQ Screen for the first time.** In total the videos take no longer than 20 minutes to watch, and should could cover everything you need to get started with the software.

# Download

FastQ Screen may be obtained from the Babraham Bioinformatics download page.

# Installation

Before running FastQ Screen there are a few prerequisites that will need to be installed:

1. A sequence aligner. FastQ Screen is compatible with Bowtie, Bowtie2 or BWA. It's easier if you put the chosen aligner in your path, but if not you can configure its location in the config file.

2. We recommend running FastQ Screen in a Linux system, on which the programming language Perl should already be installed. Perl should also be pre-installed on OSX systems, or if trying to run FastQ Screen on a Windows system you may obtain Perl from ActiveState.

3. GD::Graph FastQ Screen uses the GD::Graph module to draw PNG format graphs summarising the mapping results. FastQ Screen will still produce both text and HTML format summaries of the results if GD::Graph is not installed.

   Windows ActivePerl users can install this using;

   ```
   ppm install GD-Graph
   ```

   Other platforms can use the built in CPAN shell to install this:

   ```
   perl -MCPAN -e "install GD"
   ```

   Because GD graph uses GD this will be brought in as a dependency. GD may be easier to install using a package manager on many linux distributions. On Fedora for example you can install GD using:

   ```
   yum install perl-GD
   ```

   ..before doing the CPAN install of GD::Graph

4. IO::Uncompress::Gunzip is also required by Fastq Screen. While this is currently a standard module, previous version of Perl may not have this module installed by default.

Actually installing Fastq Screen is very simple. Download the tar.gz distribution file and then do:

```
tar -xzf fastq_screen_v0.x.x.tar.gz
```

You will see a folder called fastq_screen_v0.x.x has been created and the program is inside that. You can add the program to your path either by linking the program into: `usr/local/bin` or by adding the program installation directory to your search path.

# Configuration

In order to use FastQ Screen you will need to configure some genome databases for the program to search. This will involve downloading the sequences for the databases in FASTA format and then using either Bowtie, Bowtie2 or BWA to build the relevant index files. Please note: the aligner used to build the index files must be used to map the reads

Once you have built your index you can configure the FastQ Screen program. You do this by editing the fastq_screen.conf.example file which is distributed with the program. This shows an example set of database configurations which you will need to change to reflect the actual databases you have set up. Rename the file to fastq_screen.conf after you have finished editing.

The other options you can set in the config file are the location of the aligner binary (if it's not in your path),and the number of threads you want to allocate to the aligner when performing your screen. The number of threads will be the number of CPU cores the code will run on so you shouldn't set this value higher than the number of physical cores you have in your machine. The more threads you can allow the faster the searching part of the screen will run.

## Running the program

An example command is shown below. This would process two FASTQ files and would create the screen output in the same directory as the original files.

```
fastq_screen sample1.fastq sample2.fastq
```

By default the program looks for a configuration file named "fastq_screen.conf" in the folder where the FastQ Screen script it is located. If you wish to specify a different configuration file, which may be placed in different folder, then use the --conf option:

```
fastq_screen -conf /home/myConfig.conf sample1.fastq sample2.fastq
```

Full documentation for the FastQ Screen options can be obtained by running:

```
fastq_screen --help
```

## Test Dataset

To confirm FastQ Screen functions correctly on your system please download the Test Dataset. The file 'fastq_screen_test_dataset.fastq.gz' contains reads in Sanger FASTQ format.

1. Extract the tar archive before processing: `tar xvzf fastq_screen_test_dataset.tar.gz`
2. If not present already, create index files of recent versions of the Mouse and Human genomes (how the index files are generated will depend on the aligner used for the mapping i.e. refer to either the Bowtie, Bowtie2 or BWA documentation for further details).
3. Create a configuration file tailored to your system.
4. Run FastQ Screen

## Screening Bisulfite Samples

Mapping bisulfite converted sequences is possible with FastQ Screen, which uses the tool Bismark to process the FASTQ files. After downloading and setting-up Bismark, provide the path to Bismark in the configuration file and run FastQ Screen in bisulfite mode.

```
fastq_screen --bisulfite sample3.fastq
```

FastQ Screen, when run in Bisulfite mode, reports to which strand the reads aligned (original top strand, complementary to original top strand, complementary to original bottom strand, or original bottom strand). Refer to the Bismark documentation for more details on these bisulfite strand definitions.

# Filtering FastQ Files

You may want to filter your data to remove reads mapping to a certain species. With FastQ Screen it is possible to generate a new FASTQ file in which each FASTQ read is tagged, listing to which genomes the read did, or did not align. This file may then be processed as required to select for, or filter out, reads aligning to given species. By default, selecting --tag will result in the whole file being processed, unless over-ridden by the --subset option.

To create a tagged FASTQ file, enter on the command line something similar to that below:

```
fastq_screen --tag sample4.fastq
```

To filter the tag file, enter on the command line something similar to that below:

```
fastq_screen --filter 1000 sample5.fastq
```

This instructs FastQ Screen to extract from the FASTQ file reads that map uniquely to genome 1, but not to genomes 2, 3 or 4 (genome order set by the ordered entered in the configuration file). See the table in the FastQ Screen Option Summmary for further details of the --filter options.

It is also possible to tag and filter a file in a single operation:

```
fastq_screen --tag --filter 0001 sample6.fastq
```

In this example the file is tagged and reads mapping to a single location on genome 4, but do not align to any of the other three genomes, are written to the output file.

Adjust the filter options as required:

```
fastq_screen --tag --filter 5555 --pass 1 sample7.fastq
```

The --pass command allows the user to specify how many filters need to be passed for a read to be written to the output file. By default, all the filters should be passed. Consequently the example above will remove reads that map uniquely to any of the genomes.

It is also possible to extract reads mapping to none of the reference genomes with the option --nohits:

```
fastq_screen --nohits sample7.fastq
```

The option --nohits is equivalent to --tag --filter 0000 (zero for every genome screened).

By adjusting the filters and, if necessary, undergoing several rounds of filtering it should be possible for a user to extract the desired reads.

## FastQ Screen Options Summary

**aligner <func> :** Specify the aligner to use for the mapping. Valid arguments are 'bowtie', bowtie2' (default) or 'bwa'. Bowtie maps with parameters -k 2, Bowtie 2 with parameters -k 2 --very-fast-local and BWA with mem -a. Local aligners such as BWA or Bowtie2 will be better at detecting the origin of chimeric reads.

**bisulfite :** Process bisulfite libraries. Bismark runs in non-directional mode. The path to the bisulfite aligner (Bismark) may be specified in the configuration file. Either conventional or bisulfite libraries may be processed, but not both simultaneously. The --bisulfite option cannot be used in conjunction with --bwa.

**bowtie <text> :** Specify extra parameters to be passed to Bowtie. These parameters should be quoted to clearly delimit bowtie parameters from fastq_screen parameters. You should not try to use this option to override the normal search or reporting options for bowtie which are set automatically but it might be useful to allow reads to be trimmed before alignment etc.

**bowtie2 <text> :** Specify extra parameters to be passed to Bowtie 2. These parameters should be quoted to clearly delimit Bowtie 2 parameters from FastQ Screen parameters. You should not try to use this option to override the normal search or reporting options for bowtie which are set automatically but it might be useful to allow reads to be trimmed before alignment etc.

**bwa <text> :** Specify extra parameters to be passed to BWA. These parameters should be quoted to clearly delimit BWA parameters from FastQ Screen parameters. You should not try to use this option to override the normal search or reporting options for BWA which are set automatically but it might be useful to allow reads to be trimmed before alignment etc.

**conf <path> :** Manually specify a location for the configuration.

**filter <text> :** Produce a FASTQ file containing reads mapping to specified genomes. Pass the argument a string of characters (0, 1, 2, 3, -), in which each character corresponds to a reference genome (in the order the reference genome occurs in the configuration file).

Below gives an explanation of each character.

| Character | Explanation |
|---|---|
| 0 | Read does not map |
| 1 | Read maps uniquely |
| 2 | Read multi-maps |
| 3 | Read maps (one or more times) |
| 4 | Passes filter 0 or filter 1 |
| 5 | Passes filter 0 or filter 2 |
| - | Ignore whether a read maps to this genome |

Consider mapping to three genomes (A, B and C), the string '003' produces a file in which reads do not map to genomes A or B, but map (once or more) to genome C. The string '--1' would generate a file in which reads uniquely map to genome C. Whether reads map to genome A or B would be ignored.

When --filter is used in conjunction with --tag, FASTQ files shall be mapped, tagged and then filtered. If the --tag option is not selected however, the input FASTQ file should have been previously tagged.

**force :** Do not terminate if output files already exist, instead overwrite the files.

**help :** Print program help and exit.

**illumina1_3 :** Assume that the quality values are in encoded in Illumina v1.3 format. Defaults to Sanger format if this flag is not specified.

**nohits :** Writes to a file the sequences that did not map to any of the specified genomes. This option is equivalent to specifying --tag --filter 0000 (number of zeros corresponds to the number of genomes screened). By default the whole input file will be mapped, unless overridden by --subset.

**outdir <text> :** Specify a directory in which to save output files. If no directory is specified then output files are saved in the current working directory.

**pass <int> :** Used in conjuction with --filter. By default all genome filters must be passed for a read to pass the --filter option. However, a minimum number of genome filters may be specified that a read needs to pass to be considered to pass the --filter option. (--pass 1 effectively acts as an OR boolean operator for the genome filters.)

**quiet :** Suppress all progress reports on stderr and only report errors.

**subset <int> :** Don't use the whole sequence file, but create a temporary dataset of this specified number of reads. The dataset created will be of approximately (within a factor of 2) of this size. If the real dataset is smaller than twice the specified size then the whole dataset will be used. Subsets will be taken evenly from throughout the whole original dataset. By Default FastQ Screen runs with this parameter set to 100000. To process an entire dataset however, adjust --subset to 0.

**tag :** Label each FASTQ read header with a tag listing to which genomes the read did, or did not align. The first read in the output FASTQ file will list the full genome names along with a score denoting whether the read did not align (0), aligned uniquely to the specified genome (1), or aligned more than once (2). In subsequent reads the genome names are omitted and only the score is printed, in the same order as the first line.

This option results in the he whole file being processed unless overridden explicitly by the user with the --subset parameter

**threads <int> :** Specify across how many threads bowtie will be allowed to run. Overrides the default value set in the configuration file.

**top <int>/<int,int> :** Don't use the whole sequence file, but create a temporary dataset of the specified number of reads taken from the top of the original file. It is also possible to specify the number of lines to skip before beginning the selection e.g. --top 100000,5000000 skips the first five million reads and selects the subsequent one hundred thousand reads. While this option is usually faster than comparable --subset operations, it does not prevent biases arising from non-uniform distribution of reads in the original FastQ file. This option should only be used when minimising processing time is of highest priority.

**version :** Print the program version and exit.

# Terms of use

FastQ Screen is distributed under a "GNU General Public License", a copy of which is distributed with the software.

# Problems?

If you have any problems running this program you can either open them as bugs in our bug tracking system.

Or you can email them to: steven.wingett@babraham.ac.uk