Search Browse (/directory/) Blog (/blog/) Help (/support) Jobs (/jobs?source=header) Log In (/account/login.php)
(/)
SOLUTION CENTERS Smarter Commerce (http://ibmsmartercommerce.sourceforge.net/) Go Parallel (http://goparallel.sourceforge.net/) HTML5

Flexbar
(/p/flexbar/)

Summary (/projects/flexbar/) Files (/projects/flexbar/files/) Requests (/p/flexbar/support/) Manual (/p/flexbar/wiki/) Code (/p/flexbar/code/)

Search Wiki

Wiki Home (/p/flexbar/wiki/)
Browse Pages (/p/flexbar/wiki/browse
Browse Labels (/p/flexbar/wiki/browse
Formatting Help (/p/flexbar/wiki/markdo

Manual (history) (feed) (../search)

Flexbar — flexible barcode and adapter removal

Flexbar is a software to preprocess high-throughput sequencing data efficiently. It demultiplexes barcoded runs and removes adapter sequences. Moreover, trimming and filtering features are provided. Flexbar increases mapping rates and improves genome and transcriptome assemblies. It supports next-generation sequencing data from Illumina, Roche 454, and the SOLiD platform. Recognition is based on exact overlap sequence alignment.

Processing steps

The following series of preprocessing steps is conducted by Flexbar.
In standard settings, only steps that are marked with a star (*) are active.

1. Filtering reads with uncalled bases (*)
2. Trimming of bases on left or right end
3. Quality based trimming from right side
4. Barcode detection, removal and read separation
5. Adapter detection and removal
6. Trimming of reads to certain length from right
7. Filtering short sequencing reads (*)

Contents

Program usage

Usage of program on command line:

```
flexbar -t target -f format -r reads [-b barcodes] [-a adapters] [options]
```

Required parameters

Mandatory parameters:

- input file with reads
- input format of reads
- target name for output prefix

Format of reads

Choose the format that matches your sequencing data. The fastq format has different quality scalings, which should be specified for quality based trimming of reads. The "cs" is an abbreviation for color-space. In case of csfastq, Flexbar assumes sanger quality scaling. The output format is the same as the format of input.

Supported read formats:

- fasta
- fastq
- fastq-sanger
- fastq-solexa
- fastq-illumina1.3
- fastq-illumina1.5
- fastq-illumina1.8
- csfasta
- csfastq

Barcodes and adapters

To detect barcodes or remove adapter sequences from sequencing reads, specify barcodes and adapter(s) in addition. Create a fasta file containing barcodes or adapter sequences to be detected. Tag names (e.g. adapter1) can be chosen freely:

```
>adapter1
TCGATTACGT
>adapter2
GGTAGTACGCTA
```

Except for reads in color space format, the wildcard character 'N' can be used in barcodes and adapters to virtually match all characters in letter space. Run Flexbar, for example with 4 threads for parallel computation:

```
    flexbar -t output -f fastq -r reads.fastq -a adapters.fasta -n 4
```

Flexbar aligns each read from file.fastq to the adapter sequences found in adapters.fasta one by one based on global alignment. For each read, the highest scoring valid alignment is employed for adapter removal. In case of reads in color-space format, barcodes and adapters are automatically converted from letter-space. Color-space reads typically begin with an artificial starting reference base and a first transition T1. The first two characters are not taken into account during alignment. The output will contain the first two characters again. This is necessary to allow an overlap of the adapter sequence and the read start.

Paired reads

Flexbar natively processes paired reads and outputs consistent read files after processing. If single reads are left over, they are written to an extra file. To use paired reads, specify two input files that are in correct order for paired reads:

```
    flexbar -r s_1_1.fastq -p s_1_2.fastq -f fastq -t output [options]
```

Output files output_1.fastq and output_2.fastq that contain processed reads are created.

Barcoded reads

To detect barcodes within reads, specify barcodes file when running Flexbar:

```
    flexbar -r barcoded_reads.fastq -b barcodes.fasta -f fastq -t output [options]
```

This generates several output files for each barcode being contained in the barcode fasta file, where fasta tags are used in filename:

```
output_barcodeA.fastq
output_barcodeB.fastq ...
```

Separate barcode reads

Illumina produces two output files for single barcoded and three for paired-end barcoded runs:

```
s_1.fastq - containing sequencing reads (1st read of the pair)
s_2.fastq - separate barcode reads
s_3.fastq - containing 2nd read of the pair (optional)
```

Run Flexbar with these files:

```
    flexbar -r s_1.fastq -p s_3.fastq -b barcodes.fasta -br s_2.fastq -f fastq -t output
```

For each barcode several output files are generated:

```
output_barcodeA_1.fastq (s_1.fastq reads with barcode A)
output_barcodeA_2.fastq (s_3.fastq reads with barcode A)
output_barcodeB_1.fastq ...
```

Detection parameters

Parameters for detection and removal of barcodes as well as adapters:

Trim-end modes

Further, parameters --barcode-trim-end and --adapter-trim-end are important. These options refer to the barcode or adapter position within the read and specify which part of the read gets removed in barcode or adapter removal.

The following illustrations explain the five different modes. When a barcode or adapter aligns in agreement with the threshold and min-overlap, it could still be rejected due to restriction by trim-end mode. Red indicates a non-valid match and no removal of sequence from the read. Blue indicates removal of corresponding read sequence. Green highlights the remaining sequence of the read after removal. The aligned barcode or adapter sequence is depicted by Xs.

ANY mode: longer part of read remains

```
   ACGTAGCCGTACT
   ---------XXXXXX
   -------XXXXXX
   ----XXXXXX---
   XXXXXX-------
 XXXXXX---------
```

RIGHT mode: align >= read start, left part remains

```
   ACGTAGCCGTACT
   ---------XXXXXX
   -------XXXXXX
   ----XXXXXX---
   XXXXXX-------
 XXXXXX---------
```

LEFT mode: align <= read end, right part remains

```
   ACGTAGCCGTACT
   ---------XXXXXX
   -------XXXXXX
   ----XXXXXX---
   XXXXXX-------
 XXXXXX---------
```

RIGHT_TAIL mode: considers last n bases of read (default: barcode or adapter length)

```
   ACGTAGCCGTACT
   ---------XXXXXX
   -------XXXXXX
   ----XXXXXX---
   XXXXXX-------
 XXXXXX---------
```

LEFT_TAIL mode: considers first n bases of read (default: barcode or adapter length)

```
   ACGTAGCCGTACT
   ---------XXXXXX
   -------XXXXXX
   ----XXXXXX---
   XXXXXX-------
 XXXXXX---------
```

Options --barcode-tail-length and --adapter-tail-length can be used to modify the length of the tail mode region to be considered.

Min-overlap

The minimum required overlap for barcodes is set to their length per default and to one base-pair for adapters. It can be a drawback to set the min-overlap parameter to high, if adapter sequences are mostly located at the end of reads. Therefore Flexbar uses a very low default value. Run for example:

```
flexbar -r test.fasta -f fasta -t result -a adapters.fasta -ao 5 -l ALL
```

Assume a simple adapter like AAAAAA and a min-overlap of 5. The adapter would be recognized in the following read:

```
TCGAAAAAAGCGTGTTT
   ||||||
---AAAAAA--------
```

If the adapter is located at the 3' end, but only with 4 bases the adapter would not be removed, because we specified a min-overlap of 5 and only 4 bases match the read. With a lower min-overlap value adapters can be better detected at read ends. In default behaviour min-overlap is set to a low value and the adapter will be removed.

```
TCGGCGTGTTTAAAA
           ||||
-----------AAAAAA
```

Threshold

The threshold parameters specifies how many mismatches and indels are allowed for an adapter or barcode sequence to be removed. Consider the following alignment:

```
ACGTAGCCGTACTGT
       |||| |
-------CGTATT--
```

We have 1 mismatches in 6 bases. If we select an adapter-threshold of 1 error per 10 bases, only 0.6 errors are allowed per 6 bases (the overlap). The adapter is not removed. By increasing the threshold to 2 we allow 1.2 errors per 6 bases and therefore the adapter gets removed. For min-overlap we choose 6 or lower. Otherwise the adapter never gets removed.

```
    flexbar -t result -f fasta -r test.fasta -a adapters.fasta -at 2 -ao 6 -l ALL
```

Alignment scoring

The alignment scoring scheme can be adjusted for detection of barcodes and adapters separately. This includes match, mismatch and gap scores. For example, it could make sense to specify a larger score for gaps, when the data's sequencing platform has high indel error rates. See program options section for information on how to set them for barcodes or adapters. If the gap score is set to a value of -4 for example, it means that a gap's score corresponds to 4 mismatches and can be compensated by 4 matches:

```
    flexbar -t result -f fasta -r test.fasta -a adapters.fasta --adapter-gap -4
```

Filtering and trimming

Flexbar provides several basic read filtering and trimming features.

Filtering uncalled bases

In the first step, reads that contain more uncalled bases than specified are discarded. These reads are not included in further processing steps and the output. Per default not a single uncalled base is allowed. For example, to allow not more than 2 uncalled bases per read, run:

```
    flexbar -t target -f fastq -r reads.fastq --max-uncalled 2
```

Trimming of read ends

Trimming a fixed number of bases at left and right read ends is the next step, which is not performed with standard settings. For example, to trim 5 bases at the left side of reads:

```
    flexbar -t target -f fastq -r reads.fastq --pre-trim-left 5
```

Quality based trimming

The trimming step based on phred quality values helps to deal with higher error rates towards the end of reads. For example, to trim the 3' end until quality offset value 30 (corresponding to 63 in sanger format) or higher is reached, specify:

```
    flexbar -t target -f fastq-sanger -r reads.fastq --pre-trim-phred 30
```

Trimming to read length

Reads can be trimmed to a certain length by cutting the right side. This step comes after barcoding and adapter removal. It is suited for cases where tools in the downstream analysis require a maximal or even an exact length of reads. For example, to cut the rigth side of reads such that reads are not longer than 50 bases, issue the following command:

```
    flexbar -t target -f fastq -r reads.fastq --post-trim-length 50
```

Filtering short reads

There are many applications of sequencing data in which reads cannot be processed if they are too short. Therefore, we support filtering of reads based on their length. For example to discard reads that are shorter than 50 base pairs, set the minimal read length to this number. To make sure that reads have exactly a certain length, use the post-trim-length option in addition:

```
    flexbar -t target -f fastq -r reads.fastq --post-trim-length 50 --min-readlength 50
```

Logging and tagging

Flexbar serves various logging and read tagging features for inspection of alignments, and to facilitate downstream data analysis. For example, random sequence tags which allows to recognize artifacts that stem from library amplification are supported.

Logging alignments

Print the optimal sequence alignment for each read and the barcode or adapter with maximal score if it is valid. Choose either to view all such valid alignmnets (ALL) or only those being used for read modification by sequence removal (MOD). A third option (TAB) can be selected for tabular output of alignment statistics. Usage example:

```
    flexbar -t target -f fastq -r reads.fastq --log-level ALL
```

Single read output

While processing paired reads, it is possible to run into the situation that only one of the two reads of a pair is shorter than the specified minimal read length. In this case Flexbar discards also the single read that is not too short in order to keep the two files of paired reads in sync. The single-reads option can be used to write these long enough single reads to separate files without loosing consistency for paired read output.

Random sequence tags

Random sequence tags in reads allow to recognize artifacts, e.g. stemming from amplification by PCR during preparation of the sequencing library. Such random tags can be captured with Flexbar by specifying the random-tags option, and the character N at barcode or adapter positions for which the read is supposed to contain the random sequence tag. The characters at these positions with N are extracted and get subsequently appended to the read name, separated by underscore.

Miscellaneous options

The option fasta-output forces non-quality file formats fasta and csfasta for output. For example, this option is suited if the input file format is fastq, and fasta output is preferred. Furthermore, one can inspect the length ditribution of read output files by setting the option flag length-dist, which leads to the generation of a length distribution file for read output file. The number-tags option triggers replacement of read name tags by an ascending number to save space. The removal-tags option can be employed to tag reads if adapter or barcode removal takes place.

Program options

Usage of Flexbar version 2.33:

```
    flexbar -t target -f format -r reads [-b barcodes] [-a adapters] [options]
```

Information on software:

```
    -h, --help
          Display this help message
        --version
          Display program version
    -c, --cite
          View citation information
```

Basic parameters

```
    -n, --threads NUM
          Number of threads, default: 1
    -t, --target STR
          Prefix for output file names
    -r, --reads FILE
          Input file with reads, that may contain barcodes
    -p, --reads2 FILE
          Second input file for paired read scenario
    -f, --format STR
          Input format of reads: csfasta, csfastq, fasta, fastq, fastq-sanger,
          fastq-solexa, fastq-i1.3, fastq-i1.5, fastq-i1.8 (illumina 1.8+)
```

Barcode detection

```
    -b,  --barcodes FILE
           Fasta file with barcodes, specify (br) to use separate barcode reads
    -br, --barcode-reads FILE
           Fasta or fastq file with separate barcode reads, if not within reads
    -be, --barcode-trim-end STR
           Type of barcoding within reads, see section trim-end modes, default: ANY
    -bn, --barcode-tail-length NUM
           Number of bases for tail trim-end types, default: barcode length
    -bo, --barcode-min-overlap NUM
           Minimum overlap of barcode and read, default: barcode length
    -bt, --barcode-threshold NUM
           Allowed mismatches and indels per 10 bases for barcode, default: 1.0
    -bk, --barcode-keep
           Keep barcodes within reads instead of removal
    -bm, --barcode-match NUM
           Match score, default: 1
    -bi, --barcode-mismatch NUM
           Mismatch score, default: -1
    -bg, --barcode-gap NUM
           Gap score, default: -7
```

Adapter removal

```
    -a,  --adapters FILE
           Fasta file with adapters, or barcodes to remove within reads
    -as, --adapter-seq STR
           Single adapter sequence as alternative to adapters option
    -ae, --adapter-trim-end STR
           Type of alignment for removal, see section trim-end modes, default: RIGHT
    -an, --adapter-tail-length NUM
           Number of bases for tail trim-end types, default: adapter length
    -ao, --adapter-min-overlap NUM
           Minimum overlap of adapter and read, default: 1
    -at, --adapter-threshold NUM
           Allowed mismatches and indels per 10 bases for adapter, default: 3.0
    -am, --adapter-match NUM
           Match score, default: 1
    -ai, --adapter-mismatch NUM
           Mismatch score, default: -1
    -ag, --adapter-gap NUM
           Gap score, default: -7
```

Filtering and trimming

You can filter and trim reads before and after processing of barcodes and adapters:

```
    -u, --max-uncalled NUM
          Allowed uncalled bases (N or .) in reads, default: 0
    -x, --pre-trim-left NUM
          Trim specified number of bases on 5' end of reads before detection
    -y, --pre-trim-right NUM
          Trim specified number of bases on 3' end of reads before detection
    -q, --pre-trim-phred NUM
          Trim 3' end until specified or higher quality reached, precise fastq format
    -z, --post-trim-length NUM
          Trim to specified read length from 3' end after removal
    -m, --min-readlength NUM
          Minimum read length to remain after removal, default: 18
```

Logging and tagging

Invoke Flexbar with --log-level ALL to print alignments or choose to tag modified reads:

```
    -l, --log-level STR
          Print valid sequence alignments of reads. One of ALL, MOD, and TAB.
    -s, --single-reads
          Output single reads for partially too short paired reads.
    -o, --fasta-output
          Prefer non-quality formats fasta and csfasta for output.
    -i, --length-dist
          Write length distribution for read output files.
    -e, --number-tags
          Replace read tags by ascending number to save space.
    -g, --removal-tags
          Tag reads that are subject to adapter or barcode removal.
    -d, --random-tags
          Random read tags at barcode or adapter positions with N.
```

SourceForge

About (/about)

Site Status (/blog/category/sitestatus/)

@sfnet_ops (http://twitter.com/sfnet_ops)

Find and Develop Software

Create a Project (/create/)

Software Directory (/directory/)

Top Downloaded Projects (/top)

Community

Blog (/blog/)

@sourceforge (http://twitter.com/sourceforge)

Job Board (/jobs?source=footer)

Help

Site Documentation (http://p.sf.net/sourceforge/doc

Support Request (/support)

Real-Time Support (http://p.sf.net/sourceforge/irc)

Copyright © 2013 SourceForge. All Rights Reserved.
SourceForge is a Dice Holdings, Inc.
(http://www.diceholdingsinc.com/phoenix.zhtml?

Terms (http://geeknetmedia.com/terms-of-use) Privacy (http://geeknetmedia.com/privacy-statement/)
Cookies/Opt Out (http://geeknetmedia.com/opt-out-choices) Advertise (http://geeknetmedia.com)