WHAM

WISCONSIN'S HIGH-THROUGHPUT ALIGNMENT METHOD

Getting Started

You can download the latest version of WHAM. This tarball includes the source code and sample data. After downloading this file, move it to the appropriate directory and issue the following command to extract the code:

```
tar -zxvf wham.tar.gz
```

Then compile the code in the new directory, as follows:

```
cd wham
make
```

Throughout this tutorial, if you see an error "command not found", add `./` before the command. Further explanations regarding command options can be found in the manual.

Building a new index

Before performing any alignments, you need to build a new index. The index is stored on disk and will be loaded when performing alignments. The WHAM tarball comes with a sample sequence containing the first 100,000 bases of *Human Genome Chromosome 1*. As an example, we issue the following command to build an index on the sample sequence.

```
wham-build -l 60 -v 2 --mask sequences/chr1_100k.fa indexes/idx
```

The options `-l 60` and `-v 2` specify that the index is used for aligning 60bps reads with up to 2 mismatches. This command prints the message "Complete" if the index is built successfully. The directory indices should contain four new files: `idx.head.whm`, `idx.interval.whm`,`idx.sequence.whm`, and `idx.i0.whm`.

Computing alignments

With the pre-built index, we can use the WHAM aligner to align reads. The WHAM tarball includes a sample single-end read file `sample.fq` and a pair of paired-end read files. The following examples demonstrate how to use the WHAM aligner.

Example 1

```
wham reads/sample.fq indexes/idx output
```

This command aligns all reads in the file `reads/sample.fq` and prints out one alignment per read into the file output. WHAM finds three alignments as shown below. The first alignment is on the forward strand (+) and has 1 mismatch (`C⇒T`). The second alignment is an exact match on the forward strand. The last match is on the reverse strand (-).

```
+ chr1 51614 CCAGCACTTTGGGAGGCCGAGGCGGGTGGATCACGAGGTCAGGAGATCGAGACCATCCTG 26:C>T

+ chr1 83977 GAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAA

- chr1 17446 CACAGCGTGCACTGTGGGGTCCCAGGCCTCCCGAGCCGAGCCACCCGTCACCCCCTGGCT
```

Example 2

```
wham -a --best reads/sample.fq indexes/idx output
```

Specifying `-a` instructs WHAM to report all the valid alignments for each read. Option `--best` results in a best-to-worst order on the reported alignments. WHAM finds 2 alignments for the first read, 6 alignments for the second read, and 1 alignment for the third read. All alignments are printed in sorted order.

```
+ chr1 51614 CCAGCACTTTGGGAGGCCGAGGCGGGTGGATCACGAGGTCAGGAGATCGAGACCATCCTG 26:C>T

- chr1 62070 CAGGATGGTCTCGATCTCCTGACCTCGTGATCCACCCGCCTCGGCCTCCCAAAGTGCTGG 22:T>G,41:T>C

+ chr1 83977 GAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAA

+ chr1 83981 GAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAA

- chr1 54712 TTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTC

+ chr1 83973 GAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAA 0:A>G

- chr1 54716 TTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTC 1:C>T

- chr1 54708 TTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTC 56:T>C,58:A>T
```

```
- chr1 17446 CACAGCGTGCACTGTGGGGTCCCAGGCCTCCCGAGCCGAGCCACCCGTCACCCCCTGGCT
```

Example 3

```
wham -k 3 -m 5 reads/sample.fq indexes/idx output
```

Specifying `-k 3` instructs WHAM to report up to 3 valid alignments per read. Specifying `-m 5` instructs WHAM to refrain from reporting any alignments for reads that have more than 5 valid alignments. In this case, a total of 3 valid alignments exist. All alignments of the third read are discarded.

```
+ chr1 51614 CCAGCACTTTGGGAGGCCGAGGCGGGTGGATCACGAGGTCAGGAGATCGAGACCATCCTG 26:C>T
- chr1 62070 CAGGATGGTCTCGATCTCCTGACCTCGTGATCCACCCGCCTCGGCCTCCCAAAGTGCTGG 22:T>G,41:T>C
- chr1 17446 CACAGCGTGCACTGTGGGGTCCCAGGCCTCCCGAGCCGAGCCACCCGTCACCCCCTGGCT
```

Example 4

```
wham -v 3 reads/sample.fq indexes/idx output
```

Specifying `-v 3` instructs WHAM to report alignments with up to 3 mismatches. However, WHAM does not guarantee that it can find all valid alignments with 3 mismatches. Note that since the index `idx` is built with the option `-v 2` (see the command for building the index in the manual), all alignments with up to 2 mismatches can be found by WHAM. In this case, WHAM reports one more alignment for the fourth read with 3 mismatches.

```
+ chr1 51614 CCAGCACTTTGGGAGGCCGAGGCGGGTGGATCACGAGGTCAGGAGATCGAGACCATCCTG 26:C>T
+ chr1 83977 GAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAA
- chr1 17446 CACAGCGTGCACTGTGGGGTCCCAGGCCTCCCGAGCCGAGCCACCCGTCACCCCCTGGCT
- chr1 17962 GCGGGTGCGTCTATGCAGGCCAGGGTCCTGGGCGCCCGTGAAGATGGAGCCATAGTCCTG
5:T>G,47:C>A,54:G>T
```

Example 5

```
wham -a -e 5 reads/sample.fq indexes/idx output
```

Specifying `-e 100` instructs WHAM to report alignments with up to 2 mismatches. In addition, the sum of the Phred quality scores at all mismatched positions cannot exceed 100. Compared with the results of Example 2, most of the alignments with mismatches are filtered out by the option -

`e 100`.

```
+ chr1 51614 CCAGCACTTTGGGAGGCCGAGGCGGGTGGATCACGAGGTCAGGAGATCGAGACCATCCTG 26:C>T

- chr1 17446 CACAGCGTGCACTGTGGGGTCCCAGGCCTCCCGAGCCGAGCCACCCGTCACCCCCTGGCT
```

Example 6

```
wham -1 reads/sample_pair_1.fq -2 reads/sample_pair_2.fq indexes/idx output
```

This command takes paired-end reads from the two files, and outputs valid alignments to the file `output`.

```
+ chr1 42601 AAAAGTTAACCCATATGGAATGCAATGGAGGAAATCAATGACATATCAGATCTAGAAACT

- chr1 42735 AAATTATTGAGAATAAAAAAAAAGATTAGAATAGTTTTTTTAAAAAAAAAGCCCAGAAAC 49:C>G,51:C>G

+ chr1 89382 CTTATTCATTCAGAAAACATACTAAGTGCTGGCTCTTTTTCATGTCCTTTATCAAGTTTG

- chr1 89458 GTTTTCTTTCTGATGTAAACTCTCAAAGTTTGAAGGGTATTGTCTTTTCCTGATACATAC 6:C>T

+ chr1 47231 AACACATTTTCAGTGTTGAATGATAAATTTTGGAATAGTTAACAGATGATAAAAGTGTTG

- chr1 47410 TCTTGACACACATTAAGCTCACTGACCCCCACACCATGAATGAGGGCATCTTCAACAATG
```

Example 7

```
wham -t 16 reads/sample.fq indexes/idx output
```

Specifying `-t 16` instructs WHAM to align reads with 16 concurrent threads. The output is exactly same as that of Example 1.

WHAM

PAGES

Comparison using wgsim
Download
Main
Manual
Tutorial

THE LATEST

New release

A new version (v0.1.5) of WHAM with lots of improvements is available […]

MORE

Thanks for dropping by! Feel free to join the discussion by leaving comments, and stay updated by subscribing to the RSS feed.