TBseq: A computational pipeline for whole genome sequencing analysis of *Mycobacterium tuberculosis* complex isolates

[DESCRIPTION]

TBseq is a semi-automated pipeline for variant calling and detection of resistance mediating SNPs in user provided *Mycobacterium tuberculosis* samples. The pipeline consists of open source software for read processing, mapping and base call recalibration. TBseq has a modular architecture. Logical workflows are separated from each other in order to ensure skipping or repeating of specific tasks. Variant calling is carried out by SAMTOOLS mpileup files. Joined variant analysis from user provided samples can be performed for transmission chain analysis. Phylogenetic SNP files in FastA format are generated for phylogenetic tree reconstruction. All output is written in a .tab delimited file format and readable for most tabular viewers like Excel. TBseq uses multi-threading in time consuming parts of the pipeline.

[SYNOPSIS]

Use TBseq within a folder where reads are stored!

```
TBseq [OPTIONS] [PARAMETER]
```

[OPTIONS & PARAMETER]

[OPTIONS]

[PARAMETER]

--step

The pipeline is modular. You can either execute the full pipeline or only sub logics. This is an essential option!

TBfull

Parameter of **--step** for executing the whole pipeline. If you choose this parameter, make sure that you set all options to your appropriate parameter!

TBreads

Parameter of **--step** for merging .fastq.gz files. Depending on your sequencing machine, two, four or eight .fastq.gz files are generated for one sample. This logic merges and renames multiple .fastq.gz files from the same sample and sequencing library into one (single-end) or two (paired- end) file(s), depending on your input. It is very essential for the pipeline that your reads follow a six field naming scheme. Fields are separated by underscores. Therefore, no underscores are allowed within fields: `[SampleID]_[LibID]_[Source]_[RunID]_[Readlength]_[Direction].fastq.gz`

The `[Direction]` field indicates forward (R1) or reverse (R2) orientation of read files, if paired-end sequencing was performed or unidirectional (R0) if single-end sequencing was performed.

Input: `*.fastq.gz`

Output: `*.fastq.gz`

TBbwa

Parameter of **--step** for mapping reads to a reference genome, using BWA. On default, TBseq uses the *M. tuberculosis* H37Rv genome as a reference. BWA is executed with a default parameter setting. After mapping, .sam files are converted into .bam files, the mapping is sorted, indexed and putative PCR duplicates are removed, using the program SAMTOOLS. Wherever possible, multi-threading is activated by the user provided **--threads** parameter.

Input: `*.fastq.gz`

Output:

```
Bam/*.bam
Bam/*.bam.bai
Bam/*.bamlog
```

TBmerge

Parameter of **--step** for merging multiple .bam files from the same sample and sequencing library, using the program SAMBAMBA. This logic merges mappings, if multiple sequencing runs were performed for the same sample and library. It improves mapping quality for samples. SAMBAMBA is executed with default values.

Input:

```
Bam/*.bam
Bam/*.bai
Bam/*.bamlog
```

Output:

```
Bam/*multi*.bam
Bam/*multi*.bai
Bam/*multi*.mergelog
Merged_Bam/*.bam
Merged_Bam/*.bai
```

TBstats

Parameter of **--step** for basic mapping statistics using SAMTOOLS flagstat. This logic calculates the total number of reads, total number of mapped reads, theoretical coverage, real coverage, the ratio between both coverages and the $\log_2$ of the ratio for each sample. SAMTOOLS flagstat is executed with default values.

Input: `Bam/*.bam`

Output: `Bam/mapping_statistics.tab`

TBrefine

Parameter of **--step** for realignment around indels and base call recalibration, using the program GATK. GATK parameter are default parameter with the exception of:

```
--downsample_to_coverage 10000
--defaultBaseQualities 12
--maximum_cycle_value 600
--noOriginalAlignmentTags
```

For the Base call recalibration, a set of known MTB resistance SNPs is used. The list is stored in var/res/Base_Calibration_List.vcf of the package.

Input: `Bam/*.bam`

Output: `GATK_Bam/*.gatk.bam`

GATK_Bam/*.gatk.bai
GATK_Bam/*.gatk.bamlog
GATK_Bam/*.gatk.grp
GATK_Bam/*.gatk.intervals

TBpile

Parameter of **--step** for creating .mpileup file(s) from refined .bam file(s), using the program SAMTOOLS. Parameter are default parameter with the exception of:

-B
-A

see: http://www.htslib.org/doc/samtools.html for more information about this parameter setting.

Input: GATK_Bam/*.gatk.bam

Output: Mpileup/*.gatk.mpileup
Mpileup/*.gatk.mpileuplog

TBtable

Parameter of **--step** for creating a position table(s) from .mpileup file(s). The position table consists of 21 columns, representing the frequency of nucleotide compositions for each genomic position. The logic can be executed in low memory usage mode (**--lowmem**) or default mode. The columns are:

Pos	Genome position
Insindex	An index for reporting insertion sites. 0 means the level of the reference genome
RefBase	The nucleotide found for the reference genome at this position
As	forward read frequency of the nucleotide A within the mapping at this position
Cs	forward read frequency of the nucleotide C within the mapping at this position
Gs	forward read frequency of the nucleotide G within the mapping at this position
Ts	forward read frequency of the nucleotide T within the mapping at this position
Ns	forward read frequency of ambiguous nucleotides within the mapping at this position
GAPs	forward read frequency of a GAP within the mapping at this position
as	reverse read frequency of the nucleotide A within the mapping at this position
cs	reverse read frequency of the nucleotide C within the mapping at this position
gs	reverse read frequency of the nucleotide G within the mapping at this position

`ts` reverse read frequency of the nucleotide T within the mapping at this position

`ns` reverse read frequency of ambiguous nucleotides within the mapping at this position

`gaps` reverse read frequency of a gap within the mapping at this position

`Aqual` Number of A nucleotides having a higher or equal phred score of 20

`Cqual` Number of C nucleotides having a higher or equal phred score of 20

`Gqual` Number of G nucleotides having a higher or equal phred score of 20

`Tqual` Number of T nucleotides having a higher or equal phred score of 20

`Nqual` Number of ambiguous nucleotides having a higher or equal phred score of 20

`GAPqual` Number of GAPs having a higher or equal phred score of 20

Input: `Mpileup/*.gatk.mpileup`

Output: `Position_Tables/*.gatk_position_table.tab`

TBvariants

Parameter of **--step** for variant calling from position table(s). This logic is able to use five user provided thresholds for variant calling. On default, an allele is called based on majority rule. First, positions are only considered, if at least four forward reads (**--mincovf 4**) and second, four reverse reads (**--mincovr 4**) support an allele. Third, at least four reads must show a phred score ≥ 20 (**--miphred20 4**). Fourth, the majority allele must be present with a frequency of 75% or higher (**--mifreq 75**), otherwise the allele is indicated as ambiguous. Fifth, the module can be executed with the output modes 0 to 3 (**--outmode**). Output mode 0 is the default mode and ignores positions where only the reference is present. In addition, ambiguous variant calls and uncovered positions are skipped. Output mode 1 is the low frequency mode. If a position shows a minority allele competing with a majority reference allele, the low frequency allele is reported. Output mode 2 is the SNP mode and considers only SNP positions. Output mode 3 is the all output mode and reports every variant with the exception of low frequency variants.

Input: `Position_Tables/*.gatk_position_table.tab`

Output:
```
Called/*.gatk_position_statistics_*.tab
Called/*.gatk_position_uncovered_*.tab
Called/*.gatk_position_variants_*.tab
```

TBjoin

Parameter of **--step** for creating a joint SNP analysis of user provided samples. First, a scaffold of all variant positions is built from variant files of your provided samples (--

samples). You need to provide a grouping name (**--groupname**) for file naming. Second, variants are recalculated with output mode 3. The output is a table of concatenated variant files from user provided samples. The first line within the .tab delimited file is a sample header line. The second line describes the joint analysis for variant positions and is separated to global fields and sample specific fields.

Global fields:

`#Position` Genome position with a variant in at least one of the samples.

`Insindex` An index for reporting insertion sites. 0 means the level of the genome. If Insindex > 0, then at least one sample showed an insertion at this position.

`Ref` The reference allele present at this position

`Gene` A gene ID is indicated, if the position is within a gene

`GeneName` A gene name is indicated, if available

`Annotation` The product of the gene

Sample specific fields:

`Type` The type of the variant. Possible values are "none, SNP, GAP"

`Allele` The allele that was present for a sample at this position

`CovFor` The forward coverage at this position

`CovRev` The reverse coverage at this position

`Qual20` The number of nucleotides having a phred score above 20

`Freq` The frequency of the allele

`Cov` The coverage at this position

`Subst` The quality of a non-Synonymous substitution if the SNP occurs in a gene

Input: `Called/*.gatk_position_variants_*.tab`
`Position_Tables/*.gatk_position_table.tab`

Output: `Joint/*_joint*.tab`
`Joint/*_joint*.log`

TBamend

Parameter of **--step** for amending joint variant tables. This module is able to use three user provided thresholds. First, you need to provide a grouping name with (**--groupname**) to find the joint variant file. Second, only variants that are unambiguous in 95% of all samples are reported (**--unamig 95**). Third, SNPs that occur in a distance of 12 nucleotides are excluded in order to reduce false positive calls **(--window 12**). Fourth, phylogenetic SNPs need to pass the following criteria: 95% of all samples need to be unambiguous. Only SNPs are considered. SNPs in repetitive regions or nested within a resistance gene are excluded. Positions not passing these criteria are reported in the removed output.

Input: `Position_Tables/*_joint*.tab`

Output: `Amend/*_joint*_amended.tab`
`Amend/*_joint*_amended_u*_phylo.tab`
`Amend/*_joint*_amended_u*_phylo_w*.tab`

`Amend/*_joint*_amended_u*_phylo_w*_removed.tab`
`Amend/*_joint*_amended_u*_phylo_plainIDs.fasta`
`Amend/*_joint*_amended_u*_phylo_w*_plainIDs.fasta`
`Amend/*_joint*_amended_u*_phylo.fasta`
`Amend/*_joint*_amended_u*_phylo_w*.fasta`

TBstrains

Parameter of **--step** for lineage classification based on phylogenetic SNP maps from {Homolka:2012hd} and {Coll:2014by} publications. This module creates a .tab delimited file within the Classification folder. The file reports the majority lineage within a sample.

Input: `Joint/*.gatk_position_table.tab`

Output `Classification/strain_classification.tab`

TBgroups

Grouping of samples based on SNP distances. This module is able to use one user provided threshold for SNP distances. Strains grouped together, if they are not more than 12 SNPs apart from each other (**--distance 12**). If they are more than 12 SNPs apart from each other, a new group is formed. This module enables to distinguish between groups of samples via SNP distance patterns.

Input: `Amend/joint*_amended_u*_phylo_w*.tab`

Output: `Groups/*.matrix`
`Groups/*.groups`

--continue

If a module was chosen with the **--step** option, the **--continue** option ensures that the pipeline will continue with downstream modules. You do not need to ste this option, if you start the analysis with **--step TBfull**.

--samples

This option needs a user sample file (e.g. `samples.txt`). The file must be a two column file. Column 1 should be your `[SampleID]`. Column 2 should be your `[LibID]`. **TBjoin** depends on this file!

--groupname

This option takes a group name for the logics **TBjoin**, **TBamend** and **TBgroups**. These logics depend on the group name. If you don't support a group name, `[NONE]` is used as a default parameter.

--ref

This option sets the reference genome on which the reads will be mapped. You can choose between *M. abscessus* CIP-104536T, *M. chimaera* DSM44623, *M. fortuitum* CT6 and *M. tuberculosis* H37Rv. On default, *M. tuberculosis* H37Rv will be used for the mapping.

--machine

This option sets the field `[Source]` for the read file naming scheme. On default, the option is set to the parameter `[NGS-MACHINE]`. The parameter is ignored, if your read files have a six field naming scheme.

--run

This option sets the field `[RunID]` for the read file naming scheme. On default, the option is set to `[nXXXX]`. The parameter is ignored, if your read files have already a six field naming scheme.

--readlen

This option sets the field `[Readlength]` for the read file naming scheme. **IMPORTANT:** the logic **TBstats** depend on this parameter! Make sure you provide the correct read length, otherwise statistics will be calculated wrong. On, default the option is set to 150.

--outmode

This option is used in the logic(s): **TBvariants**. On default, the option is set to 0. The option has influence on variant output of the **TBvariants** logic. See **TBvariants** for more details.

--minbqual

This option is used in the logic(s): **TBtable**. On default, the option is set to 13. The option sets a threshold for the mapping quality. Bases covering a position are only considered, if the quality is greater or equal this value.

--mincovf

This option is used in the logic(s): **TBvariants**, **TBjoin**, **TBamend**, **TBstrains**. On default, the option is set to 4. The option sets a minimum forward read coverage threshold. Alleles must have a forward coverage of this value or higher to be considered.

--mincovr

This option is used in the logic(s): **TBvariants**, **TBjoin**, **TBamend**, **TBstrains**. On default, the option is set to 4. The option sets a minimum reverse read coverage threshold. Alleles must have a reverse coverage of this value or higher to be considered.

--minphred20

This option is used in the logic(s): **TBvariants**, **TBjoin**, **TBamend**, **TBstrains**. On default, the option is set to 4. The option sets a minimum read coverage with a specific phred quality score. A user provided number of reads must show a phred quality above or equal 20 for a certain position to be considered.

--minfreq

This option is used in the logic(s): **TBvariants**, **TBjoin**, **TBamend**, **TBstrains**. On default, the option is set to 75. The option sets a minimum frequency for the majority allele. Only majority alleles with this frequency are indicated as unambiguous. Within the module **TBstrains**, this option will have an effect on lineage classification quality. If all phylogenetic SNPs have a frequency of this value and are covered 10-fold, then lineage classification will result in a *good* quality, otherwise in a *bad* quality for a sample.

--unambig

This option is used in the logic(s): **TBamend**. On default, the option is set to 95. The option sets a minimum percentage of samples that need to show the called variant as unambiguous. If less than this percentage of samples have an unambiguous variant call at a certain position, the position will not be reported in the amended joint variant table.

--window

This option is used in the logic(s): **TBamend**. On default, the option is set to 12. The option sets a windows size in which the algorithm scans for multiple SNPs. If more than one SNP occurs within this window, SNP positions will not be reported in the amended joint variant table.

--distance

This option is used in the logic(s): **TBgroups**. On default, the option is set to 12. The option sets a SNP distance that is used to classify samples into groups of samples. If SNP distances of samples are less or equal this value, then they are grouped together, otherwise a new group is formed. We recommend to execute this module twice, with **--distance 5** and **--distance 12**.

--lowmem

This option sets a low memory environment for the logic **TBtable**. In some system situations it could be better to calculated the position tables with lower memory usage. The runtime of this logic will increase significantly, if you set this option.

--threads

This option is used in the logic(s) **TBbwa**, **TBmerge**, **TBrefine**, **TBpile**, **TBtable**. On default, the option is set to 1. The option sets the maximum number of CPUs to use within the pipeline. You can use more than one core in order to execute the pipeline faster.

[EXAMPLES]

```
TBseq --step Tbfull > Tbseq_run.log
```

Will execute the whole pipeline with default values. All log output is written into a file called TBseq_run.log. The log file is located were you started the pipeline.

```
TBseq --step TBbwa --continue --threads 8 > TBseq_run.log
```

Recommended, if read file naming scheme is correct. The pipeline starts with the read mapping module and continues after finishing this module. The system will use 8 cores whenever possible. All log output is written into a file called TBseq_run.log.

```
TBseq --step TBtable --threads 8
```

This example calls the logic **TBtable** with 8 threads and display the log output on screen. To execute this module, you need to have a finished **TBpile** output.

```
TBseq --step TBjoin
```

This example calls the logic **TBjoin** with default parameter setting and display the log output on screen. To execute this module, you need to have a finished **TBtable** and **TBvariants** output.

```
TBseq --step TBstrains > TBseq_run.log
```

This example calls the logic **TBstrains** with default parameter setting. To execute this module, you need to have a finished **TBtable** output. Log output is written into a file called TBseq_run.log.

```
TBseq --step TBvariants --mincovf 10 --mincovr 10 --minfreq 80 --minphred20 10 --
outmode 2
```

This example calls the logic **TBvariants** with a modified parameter setting for variant calling. **TBvariants** will output only SNP positions and display the log output on screen. To execute this module, you need to have a finished **TBtable** output.

[AUTHORS]

Thomas A. Kohl (core logic)

Robin Koch (core logic, package building)

Christian Utpatel (core logic)

Maria R. De Filippo (beta test)

Viola Schleusener (beta test)

Daniela M. Cirillo (Head)

Stefan Niemann (Head)

[COPYRIGHT AND LICENSE]

[REQUIREMENTS]

- Perl: perl 5, version 18, subversion 2 (v5.18.2) or higher

- Java: openjdk version "1.8.0_91" or higher

TBseq uses two non-standard CPAN modules:

- MCE (v1.808)
- Statistics::Basic (v1.6611)

TBseq uses standard CPAN modules:

- FindBin (v1.51)
- Cwd (v3.62)
- Getopt::Long (v 2.49)
- File::Copy (v2.31)
- List::Util (v1.46)
- Exporter (v5.72)
- vars (v1.03)

- lib (v0.63)
- strict (v1.11)
- warnings (v1.36)

[INSTALLATION]

1. Download TBseq from github:

   ```
   http://
   ```

2. Extract TBseq:
3. Move TBseq into /usr/local directory (needs sudo):

   ```
   mv [PATH_TO_YOUR_TBSEQ] /usr/local/
   ```

   Or move TBseq into a directory where you have write permission:

   ```
   mkdir -p /home/$USER/bin
   mv [PATH_TO_YOUR_TBSEQ] /home/$USER/bin/
   ```

4. Create a symbolic link within `/usr/local/bin`:

   ```
   ln -s /usr/local/TBseq/TBseq.pl /usr/local/bin/TBseq
   ```

   Or in the alternative directory:

   ```
   ln -s /home/$USER/bin/TBseq/TBseq.pl /home/$USER/bin/TBseq
   ```

5. Install modules via CPAN by typing in the command-line:

   ```
   cpan
   ```

6. Copy/paste the yellow block into the command line:

   ```
   install MCE
   install Statistics::Basic
   install FindBin
   install Cwd
   install Getopt::Long
   install File::Copy
   install List::Util
   install Exporter
   install vars
   install lib
   install strict
   install warnings
   ```

7. If third party programs (`bwa` and `samtools`) in `TBseq/opt` are not working, try to re-compile them. The re-compiled executables MUST be located within the appropriate folders. (e.g. change the PATH of the installation: `./configure --prefix = $PWD`). `$PWD` means:

   ```
   TBseq/opt/bwa_0.7.15/
   TBseq/opt/samtools_1.3.1/
   ```

Tested on ubuntu 16.04 LTS.

[TESTING]

You can test your installation with the read files in the directory `TBseq/tbtest`.
the test data set consists of two artificial samples. One sample has a native Illumina NextSeq naming scheme. The other sample has already an ideal naming scheme:

```
0001-01-lib0001_S01_L001_R1_001.fastq.gz
```

```
0001-01-lib0001_S01_L001_R2_001.fastq.gz
0001-01-lib0001_S01_L002_R1_001.fastq.gz
0001-01-lib0001_S01_L002_R2_001.fastq.gz
0001-01-lib0001_S01_L003_R1_001.fastq.gz
0001-01-lib0001_S01_L003_R2_001.fastq.gz
0001-01-lib0001_S01_L004_R1_001.fastq.gz
0001-01-lib0001_S01_L004_R2_001.fastq.gz
0002-01_lib0002_miseq_r0001_151bp_R1.fastq.gz
0002-01_lib0002_miseq_r0001_151bp_R2.fastq.gz
```

```
cp TBseq/tbtest /home/$USER/Desktop/
cd /home/$USER/Desktop/tbtest
TBseq --step TBfull
```

[HOMEPAGE AND SOURCE REPOSITORY]

TBseq on github:

`http://`

Research center Borstel:

`http://`