1 Platforms and requirements

The ModSeq pipeline is designed to run on Linux and other POSIX-compatible platforms, including OSX.

1.1 Software packages

1. PANDAseq, and an additional module called 'qualString.c'
2. BWA-MEM
3. SAMtools
4. HTSlib (optional)
5. Picard
6. Genome Analysis Tooolkit (GATK)
7. R packages ggplot2, seqinr, parallel, Biostrings and ShortRead.
8. Tablet (optional)

PANDAseq

PANDAseq [3] assembles Illumina overlapping paired-end reads into a consensus sequences, trying to correct for errors and uncalled bases (e. g. , N). It is distributed under the GNU General Public License.

The program takes a set of forward and reverse reads in fastq format and outputs assembled reads in fasta or fastq format.

PANDAseq binaries can be installed in Ubuntu (Linux) entering in the terminal the following command:

```
sudo apt-add-repository ppa:neufeldlab/ppa && sudo apt-get update && sudo apt
   -get install pandaseq
```

Alternatively, PANDAseq can be built form source typing in a terminal the following command:

```
git clone http://github.com/neufeld/pandaseq.git
cd pandaseq
./autogen.sh && ./configure && make && sudo make install
```

An additional module has been implemented in order to preserve meaningful phred scores for posterior applications, such as read mapping or variant detection. In order to compile such module, you should use the tool pandaxs. Move to the directory where the additional module was saved (i. e. , 'modseq/src/qualString.c'), and invoke pandaxs with the file name to build as a command-line argument:

```
pandaxs qualString.c
```

For OSX, it may required additional flags. If the linker fails, try using:

```
pandaxs qualString.c -Lpandaseq -lpandaseq
```

For more information, refer to the PANDAseq project repository (`https://github.com/neufeld/pandaseq`) or to the manual,

```
man pandaseq
```

BWA-MEM

The BWA-MEM algorithm is implemented as a component of the BWA program [1], which is distributed under the GNU General Public License version 3.0 (GPLv3). The implementation takes a BWA index and a query fasta or fastq file as input and outputs the alignments in the SAM format [2]. The pipeline uses BWA version 0.7.12-r1044.

The latest version of the source code can be downloaded from the project repository (`https://github.com/lh3/bwa`). Alternatively, you can obtain the source code from the repository and build the binary, typing in the terminal:

```
git clone https://github.com/lh3/bwa.git
cd bwa; make
```

The compiled binary is called 'bwa'. The program can be executed invoking this compiled binary file, but you have to indicate the directory where it is located. In order to make it available on the command prompt regardless of its location, add the BWA compiled binary file to your path. To do so, you could add the directory where the executable file was saved to the PATH environment variable or move it to a directory that is currently on your path. You can view the list of directories where executable files (programs) are kept with the following command (Unix-like systems):

```
echo $PATH
```

To copy the BWA binary into a directory that is already on your path, use the following command:

```
sudo scp bwa \textit{path-to-directory}/bwa
```

where *path-to-directory* is the path to the directory where you want to copy the binary executable file.

Previous releases can also be found in the sourceforge download page (`http://sourceforge.net/projects/bio-bwa/files/`). For more information, refer to the BWA project repository (`https://github.com/lh3/bwa`) or to the manual pages (http://bio-bwa.sourceforge.net/bwa.shtml). Manual pages can also be viewed, typing in the terminal:

```
man ./bwa.1
```

Likewise, you should add the binary file 'bwa.1' to your path.

Genome Analysis Toolkit

The Genome Analysis Toolkit (GATK) is a software package for analysis of high-throughput sequencing (HTS) data. The GATK is designed to run on Linux and other POSIX-compatible platforms, including MacOS X. In order to run the GATK, Java needs to be installed (version 1.7 and newer). If you're not sure whether you have java installed type in the command prompt:

```
java -version
```

You should see something like:

```
java version "1.8.0_60"
Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)
```

The GATK software package can be downloaded from the website `https://www.broadinstitute.org/gatk/`, after opening an account. For the pipeline, we used version 3.6-0-g89b7209.

Picard tools

Picard is a software package comprised of a set of tools for analyzing HTS data. It has been written in Java (version 1.8), and a jar file containing all utilities can be downloaded from the website `http://broadinstitute.github.io/picard/`. The pipeline uses Picard tools version 2.4.1.

R and R packages

R version 3.2.3 (2015-12-10). The following R packages are required:

1. seqinr (version 3.1-3)
2. parallel (base package, version 3.2.3)
3. Biostrings (version 2.38.2)
4. ShortRead (version 1.28.0)
5. ggplot2 (version 1.0.1)
6. grid (version 3.2.3)
7. gridExtra (2.2.1)
8. scales (version 0.4.0)
9. Rcpp (version 0.12.2)

Tablet

Tablet is a graphical viewer for next-generation sequencing alignments. The most recent release of this viewer can be downloaded from the James Hutton Institute website (`http://ics.hutton.ac.uk/tablet/download-tablet/`).

To install the viewer, open the terminal and go to the folder where the installer was saved. For a linux (64 bit) operating systmem, type in the terminal:

```
sh tablet_linux_x64_<version>.sh
```

2 Input files

As input, the ModSeq pipeline is designed to take the raw sequencing data in fastq format and the modules from the combinatorial library in a comma separated values (csv) format. Additionally, a input file containing user-configurable options should be also provided.

2.1 Sequencing data

The pipeline was tailored to analyze Illumina paired-end reads. In particular, it can process relatively long reads, as opposed to shorter reads (¡100 bp) initially generated by sequencing platforms.

2.2 Library modules

The csv file should be formatted as a table, with the modules as fields/columns and the variants as records/rows. The number of fields per line should correspond to the number of modules the library was designed to have, and columns must be provided in the order in which the modules are to be concatenated. The number of rows can correspond to the total number of variants (i. e. , assigning a row per modular variant) or indicate a common feature among variants of the different modules, e. g. , the organism or sequence of origin of the variant (cf. tab. 1). Empty fields should be filled as not applicable, i. e. , 'NA'. In addition, the csv file should include a header, or first line, that defines the fields in the table, and an extra column containing the row identifiers. Spreadsheet programs like Microsoft Excel make it easy to create and edit csv files.

As an example, a library of novel lantiobiotics is shown (tab. 1). This library is comprised of 5 modules, modules `A`, `B`, `C`, `D` and `E` on table 1, and two flanking consensus regions, named as `CON1` and `CON2`. Rows correspond to the type of lantibiotic from which the codons (i. e. , triplets of nucleotides that made up the variant sequence) have been taken.

Table 1: Sample input file: module-table

```
CON1,A,B,C,D,E,CON2
Constant1,GGTGCTAGCCCACGT,NA,NA,NA,NA,NA,NA
Gallidermin,NA,ATTGCATCAAAATTTCTTTGT,ACTCCAGGTTGTGCA,NA,AAAACAGGT,NA,NA
Nisin,NA,ATTACTTCGATCTCATTGTGT,ACACCTGGTTGTAAA,ACAGGTGCACTTATGGGTTGT,AATATGAAA,ACAGCTACTTGTAATTGTTCAATTCACGTTTCAAAA,NA
MutacinBNy266,NA,NA,NA,TTTAAATCTTGGAGCTTTTGT,NA,NA,NA
Pep5,NA,NA,TCAGTAAAACAATGTCAA,AAAACATTAAAAGCTACA,CGCCTTTTT,ACAGTTTCATGTAAAGGTAAAAACGGATGTAAA,NA
Epicidin280,NA,NA,ACACGTCAAGTTTGTCCA,NA,NA,NA,NA
EpilancinK7,NA,NA,NA,TCAAAAAAATATTGC,AAAGGTGTT,ACATTGACATGTGGATGCAACATCACAGGAGGTAAA,NA
Actagardine,NA,NA,NA,TCATCAGGTTGGGTTTGC,NA,NA,NA
LactocinS,NA,NA,NA,TCTGATGTGGCTGGCTGT,TTCAAATAT,TCAGCTAAACATCACTGC,NA
Subtilin,NA,NA,NA,TGGAAATCAGAATCTCTTTGC,TTCCTTCAA,NA,NA
Mutacin1140,NA,NA,NA,AAAGGTGGTTCTGGAGTTATTCAT,NA,NA,NA
HaloduracinA,NA,NA,ACTTTGACAGTTGAGTGT,NA,NA,NA,NA
Paenibacillin,NA,NA,ACTACAATTAAAGTT,TCAAAAGCAGTTTGT,NA,ACATTGACATGTATTTGTACAGGTTCATGTTCAAATTGTAAG,NA
Empty,NA,NA,NA,,NA,NA,NA
Synthetic1,NA,NA,NA,NA,AATATGAAAAAA,NA,NA
Synthetic2,NA,NA,NA,NA,AACATGAAAGTC,NA,NA
Synthetic3,NA,NA,NA,NA,AATATGTCA,NA,NA
Synthetic4,NA,NA,NA,NA,GGT,NA,NA
Constant2,NA,NA,NA,NA,NA,NA,TGATAAGCTTTCTTTGAAC
```

2.3 User-configurable options

Script is divided into six components: quality trimming, paired-end read assembly, read mapping, library composition, variant calling and error correction.

Input files: options

`seq.mode`	Sequencing mode. The options are "SE" and "PE", the latter is set as the default value. "SE" stands for single-end reads which refers to unidirectional sequencing mode, and "PE" stands for paired-end reads which refers to bidirectional sequencing mode (Default: "PE").
`in.seqDir`	Path to raw sequencing data. If it is not specified, it is set by default to './data/raw/'. When modules are run independently, and output files from previous runs are not longer stored in the default directories, the user can make use of option `in.seqDir` in combination with `in.filename`, or `forward.filename` and `reverse.filename` to specify the path to the corresponding input files.
`in.filename`	File name of the raw sequencing reads file, without explicitly specifying the file extension. However, note that the sequencing data must be provided in fastq format. This option should be provided if the sequencing mode is set to "SE". In such case, options `forward.filename` and `reverse.filename` are ignored.
`forward.filename`	File name of the set of forward reads (i. e. , raw sequencing data), without explicitly specifying the file extension. However, note that the sequencing data must be in fastq format. This option should be provided if the sequencing mode is set to "PE". In such case, option `in.filename` is ignored.
`reverse.filename`	File name of the set of reverse reads (i. e. , raw sequencing data), without explicitly specifying the file extension. However, note that the sequencing data must be in fastq format. This option should be provided if the sequencing mode is set to "PE". In such case, option `in.filename` is ignored.
`in.modDir`	Path to module-table file. If it is not specified, it is set by default to './data/modules/'.
`mod.filename`	File name of module-table file, without explicitly specifying the file extension. However, note that the program is expecting a comma-separated-value file (i. e. , extension '.csv'). The program is expecting an input file containing a header, defining the fields in the table, as well as an extra column containing the row identifiers.

Output files: options

`out.dir`	Path to output directory. If it is not specified, it is set by default to './output'.
`out.filename`	Prefix for naming output files (Default: "out").
`out.ssplot`	Boolean variable indicating whether or not the summary- statistics plots should be outputted (Default: FALSE).

Quality trimming: options

`qtrim.thold`	Phred value at or below which a base is trimmed. This threshold must be an integer greater than zero, by default it is set at 10. Note that, the range of quality scores varies depending on the sequencing platform and the base caller. In the case of current Illumina platforms, it ranges from 0 up to 41.
`qtrim.3end`	Indicates whether the quality trimming is carried out at both termini or only at the 3'-end. The options are 0, for trimming at the 3'-end only, and 1, for trimming at both termini. By default, it is set to 1.
`qtrim.flag`	Indicates whether or not quality trimming has been applied to raw reads. The options are 0, for untrimmed reads, and 1, for trimmed reads; by default, it is set to 1. Note that this flag is changed to 1 after the module for quality trimming is executed.

Assembly of paired-end read: options

`paired.file` Indicates which set of reads should be taken for the read mapping step, i. e. , forward or reverse reads. The options are "f", for forward reads, and "r", for reverse reads. By default, it is set to "f". Note that this option is applicable when the option for the sequencing mode (i. e. , '`seq.mode`') is set to "PE" and the flag for paired-end reads (i. e. , '`paired.flag`') is set to 0.

`paired.flag` Indicates whether or not reads have been merged, i. e. , if paired-end read assembly has been performed. Alternatively, this option can be used to indicate which set of read should be taken for the read mapping step, i. e. , paired or unpaired reads. The options are 0, for unpaired reads, and 1, for paired reads; by default, it is set to 1, unless the sequencing mode (i. e. , '`seq.mode`') is set to "SE", in which case '`paired.flag`' is set to 0. Note that this flag is changed to 1 after execution of the paired-end read assembly component. In addition, it is applicable when the option for the sequencing mode (i. e. , '`seq.mode`') is set to "PE".

Read mapping: options

`map.mode` The options are "bwa", for bwa mem algorithm, "gls", for "grep-like' search algorithm, and "grPA", for greedy search strategy using pairwise-sequence alignments. By default, it is set to "bwa".

`gls.ambiguity` The options are TRUE or FALSE. If true (default), an ambiguous base (e. g. , "N") can be matched to any allowed base according to the IUPAC ambiguity code, i. e. , ambiguous matches are enabled. If false, an ambiguous base can only match the same character, i. e. , ambiguous matches are disabled.

3 Output files

3.1 Read mapping

gls

- Plot depicting the module-counts as the search progresses.
- Plot of the distribution of modular variants per search round.
- A table containing the reference sequence to which each read was mapped. This table has three columns, the read ID, reference-sequence ID and the reference sequence itself.
- Distribution of modular variants (plot and table).
- Distribution of module combinations.

References

[1] Heng. Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. 2013.

[2] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, June 2009.

[3] Andre P. Masella, Andrea K. Bartram, Jakub M. Truszkowski, Daniel G. Brown, and Josh D. Neufeld. PANDAseq: paired-end assembler for Illumina sequences. *BMC Bioinformatics*, 13:31, 2012.