pandaseq

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
OUTPUT STATISTICS
LOGGING MESSAGES
EXAMPLES
VALIDATION MODULES
INCLUDED MODULES
SEE ALSO

NAME

pandaseq − PAired-eND Assembler for DNA sequences

SYNOPSIS

pandaseq **−f** *forward.fastq* **−r** *reverse.fastq* [**−6**] [**−A** *algorithm*] [**−a**] [**−B**] [**−C** *module*] [**−d** *flags*] [**−D** *penalty*] [**−F**] [**−g** *log.txt*] [**−G** *log.txt.bz2*] [**−i** *index.fastq*] [**−k** *kmers*] [**−l** *minlen*] [**−L** *maxlen*] [**−N**] [**−o** *minoverlap*] [**−O** *maxoverlap*] [**−p** *forwardprimer*] [**−q** *reverseprimer*] [**−t** *threshold*] [**−T** *threads*] [{ **−u** | **−U** } *unpaired.txt*] [**−w** *output.fasta*] [**−W** *output.fasta.bz2*]

DESCRIPTION

PANDASEQ assembles paired-end Illumina reads into sequences, trying to correct for errors and uncalled bases. The assembler reads two files in FASTQ format with quality information. If amplification primers were used (e.g., to isolate a variable region of the 16S gene, or the constant regions around zinc finger binding residues), they can be removed from the sequence during assembly. The final sequence will correct any uncalled bases in the overlapping region using the complementary strand. When mismatches occur in the overlapping region, the base with the better quality score is chosen.

The algorithm is as follows:

1. Find the positions where the forward and reverse primers match best above the threshold and discard the ends of the sequence, including the primer.
2. Pick and overlap to maximise the probability of the forward and reverse reads having come from a single piece of DNA.
3. Identify the masking of the end of the read with the quality score **B** or **#** as done by CASAVA and adjust the probabilities in this region.
4. Construct an assembled sequence between the primers and calculate the quality.
5. Check for various constraints, including quality, length, uncalled bases, and user-supplied modules.

OPTIONS

−6 Input files will have their quality encoded as PHRED + 64 instead of the PHRED + 33. PHRED + 64 was used originally in the CASAVA pipeline from version 1.3 through 1.7. In CASAVA 1.8, the score is encoded as PHRED + 33, the default.

−A algorithm

Set the algorithm used for assembly. Currently there are:

simple_bayesian[:*error_estimation*]

Uses the formula described in the original paper (Masella 2012), optionally with an error estimation (ε) provided.

ea_util

Uses the formula by FastqJoin in the ea-utils package (Aronesty 2013). No parameters are tunable.

flash Uses the formula by FLASH paper (Magoc 2011). No parameters are tunable.

pear[:*random_base*]

Uses the formula described in the PEAR paper (Zhang 2013), optionally with the probability of a random base (q) provided.

rdp_mle

Uses the formula by RDP paper (Cole 2013). No parameters are tunable.

stitch Uses the formula in the Sitch software, developed by Austin Richardson. No parameters are tunable.

uparse[:*error_estimation*]

Uses the scoring algorithm from UPARSE/USEARCH (Edgar 2013), optionally with an error estimation (ε) provided.

−a Strip the primers after assembly, rather than before. Stripping the primers first saves time, but if the overlap region is very large compared to the read, the read may have sequence from the other primer (i.e., the forward read ends with reverse primer, and/or the reverse read ends with forward primer). If the primers are stripped first, the reads will fail to assemble. This option attempts assembly first, then tries to strip the primers, so the heavily overlapping case will assemble. You should only need this if the region of interest is smaller than the whole read. It is undesirable, unless necessary, as it slows assembly down.

−B Allow input sequences to lack a barcode/tag. Normally, Illumina sequences have barcodes attached to the sequence. This allows the barcode to be missing. The tool **panda-checkid**(1) will determine if an Illumina identifier is undestood by PANDAseq and if the tag is included.

−C module

Loads an optional validation module to verify sequences are valid before emitting them. See below for more information. You may repeat this option to use multiple validation modules.

−d flags

Set debugging/output flags to provide more details about what PANDAseq is doing. To enable a flag, capitalise it; to disable, include a uncapitalise it. Provide information about the **b**uilding of a sequence. Show excruciating detail about **r**econstruction. Show some optional **s**tatistics. Show information about building the **k**-mer table. Provide error about the **f**ile parsing. Show every **m**ismatch. The default is **BFSrk**.

−D penalty

Sometimes, with repetitive sequence, the primer aligns further down the sequence. To avoid this, a primer penalty can be applied. For each base further down the sequence, *penalty* is subtracted from the proability that the primer aligns to this location. By default, the value is zero, and if used, the value should be rather small; 0.01 seesm to be sufficient in most cases.

−f forward.fastq

The location of the forward reads in FASTQ format. The file may be plain FASTQ, or compressed with **gzip**(1) or **bzip2**(1). File compression is automatically detected.

−F Normally, output will be as a FASTA even though per-base quality information is available. To retain this quality information, this option will output the sequence and the quality information in FASTQ format with quality scores encoded as PHRED + 33 (even if the input scores are PHRED + 64). The meaning of the quality score is conceptually different from the input quality scores for the overlap region, but this may not matter depending on your downstream application. If you intend to use this

information for further quality filtering, especially by a program expecting Illumina reads, you are not using this data correctly.

−g log.txt

Log all output to a plain text file, *log.txt*, instead of standard error.

−i index.fastq

If the index/barcode reads are in a separate FASTQ file, read them an apply them to the input reads.

−G log.txt.bz2

Log all output to a **bzip2**(1) compressed text file, *log.txt.bz2*, instead of standard error.

−j This option is ignored. It used to indicate that input files specified by **-f** and **-r** are compressed by **bzip2**(1). This is automatically detected now.

−k kmers

Sets the number of sequence locations for a particular *k*-mer. When attempting to align the sequences, the assembler will store the location of every *k*-mer in a table. If the same *k*-mer is present multiple times, only the first ones will be stored until the table is full; when this occurs, an **FML** error is emitted. If the sequences are highly repetitive, lost positions can prevent good alignments; this can be alleviated by increasing this amount. This should be small (no more than 10; the default is 2), or the *k*-mer table will be extremely large, using a large amount of RAM per thread. Try increasing the value until **FML** errors go away.

−l minlen

Sets the minimum length for a sequence, after primers are removed. By default, all sequences are kept. With this option, sequences shorter than desired can be discarded.

−L maxlen

Sets maximum length for a sequence, after primers are removed. By default, all sequences are kept. With this option, sequences longer than desired can be discarded.

−N Eliminate all sequences with uncalled nucleotides in the output. Otherwise, during assembly, uncalled bases (Ns) from unpaired regions may be emitted.

−o minoverlap

Sets the minimum overlap between forward and reverse reads. By default, this is at least one nucleotide of overlap. Raising this number does not generally increase the quality of the output as alignments with small overlaps tend to score poorly and are discarded anyway.

−O maxoverlap

Sets the maximum overlap between forward and reverse reads. By default, this is the read length. In highly overlapping sequences (i.e., those where the end of one read precede the start of the other), this parameter should be set to the sum of the input reads, or a value larger than that.

−p forwardprimer

Strip out primers from the start of the sequence. If the data contains a forward primer (e.g., a conserved region to amplify a 16S variable region), specifying it here will cause the primer to be located in the read and the primer, and any sequence before it, will be discarded. It is also possible to specify a number and the same number of leading bases will be stripped from the sequence. It may be useful to user a number if the sequence has many uncalled bases in the primer region, preventing a nucleotide primer from matching.

−q reverseprimer

Strip out primers from the end of the sequence. The primer is specified as it appears in the reverse read (i.e., it is a reverse complement of what it would be in the alignment).

−r reverse.fastq

FASTQ file containing the reverse reads. See **-f** for more information.

−t threshold

The score, between zero and one, that a sequence must meet to be kept in the output. Any alignments lower than this will be discarded as low quality. Increasing this number will not necessarily prevent uncalled bases (Ns) from appearing in the final

sequence. It is also used as the threshold to match primers, if primers are supplied. The default value is 0.6.

−T threads

The number of threads to spawn. This will only be avilable if PANDAseq was compiled with **pthreads**(7). In most cases, PANDAseq is IO-bound, not CPU-bound; therefore, adding more CPU capacity would have no effect. Try monitoring a running copy of PANDAseq with **top**(1); watch the CPU% for the PANDAseq process and the overall system CPU waiting time (*%wa* in the banner at the top). If waiting time is low and CPU% is very high, then multi-threading may increase speed. If the CPU waiting time is high, threading will simply not help.

Note that using multiple threads prevents sequences from being output in the same order as the original file. If you a filtering reads downstream, consider using the **filter** validation module as matching them up may be difficult.

−[U|u] unpaired.txt

Write sequences for which the optimal alignment cannot be computed to a file as concatenated pairs. For downstream processing or to stare at wistfully. If **-U** is used, the quality scores will be included.

−w output.fasta

Write all assembled sequences to a FASTA (or FASTQ) file, *output.fasta*, instead of standard output.

−W output.fasta.bz2

Write all assembled sequences to a **bzip2**(1) compressed FASTA (or FASTQ) file, *output.fasta*, instead of standard output.

OUTPUT STATISTICS

At the end of reconstruction, several statistics are output on lines beginning with **STAT**.

READS The number of reads in the input files.

NOALGN The number of sequences where there exists no overlap with a probability above the threshold.

BADR The number of sequences where the reads are unsatisfactory (too short to assemble).

SLOW The number of sequences where the fast hashing algorithm could not figure out the optimal overlap, and so every possible overlap had to be considered. Nothing is necessarily wrong with these sequences; they just take longer to assemble. Very repetitive patterns can cause PANDAseq to spend more time investigating overlaps that are likely wrong, resulting the processing time of the file to be quite long if there are many sequences in this category. If they are a significant percentage of the input data, try increasing the size of the *k*-mer table, using the **-k** option; this will cause PANDAseq to use more memory, but it may be faster.

NOFP The number of sequences where the forward primer could not be aligned. This is only done when **-p** is supplied and a nucleotide sequence.

NORP The number of sequences where the reverse primer could not be aligned. This is only done when **-q** is supplied and a nucleotide sequence.

LOWQ The number of sequences where the quality score of the reconstruction is below the threshold. This says nothing about the quality scores of the individual bases in the forward and reverse reads.

DEGENERATE

The number of sequences containing uncalled/degenerate/N bases in the final reconstruction (it is immaterial if there are uncalled bases in the reads.) This is only done when **-N** is provided.

SHORT The number of sequences where the final reconstructed sequence is too short. This is only done when **-l** is provided.

LONG The number of sequences where the final reconstructed sequence is too long. This is only done when **-L** is provided.

OK The number of sequences output.

OVERLAPS

The number of sequences assembled for each possible overlapping length. The first number is the number of sequences with only one overlapping base, the second with two overlapping bases, and so on.

LOGGING MESSAGES

During output, the assembler may output any of the following errors.

ERR BADID
: The name of the input read did not follow the known Illumina standard formats. Older versions of CASAVA produce sequences with IDs that look like **HWUSI-EAS1661_9323_FC619KG:7:1:1190:15190#ATCACG/1**, where the fields are *instrument:lane:tile:x:y#tag/direction*. Newer version of CASAVA produce IDs that look like **HWI-ST822:85:C05C3ACXX:1:1101:1171:2104 3:N:0:TAGACA**, where the fields are *instrument:run:flowcell:lane:tile:x:y direction:filtered:flags:tag*. If your sequence headers do not look like either of these, either Illumina has created yet-another header format or, more likely, your sequence headers have been manipulated by some upstream processing, possibly at your sequencing centre. PANDAseq needs the original Illumina probabilities; not ones manipulated by other programs. We're very picky about that. Sometimes, for mysterious reasons, the sequences lack the barcoding tag. The **-B** option will cause the lack of barcode to be ignored. This will obviously invalidate the use of validation modules that depend on the barcode.

ERR BADNT
: An invalid letter was found in a nucleotide read. Likely caused by incorrect or corrupt input files.

ERR BADSEQ
: The an unexpected character or end of the input file was detected. Likely caused by incorrect or corrupt input files.

ERR EOF
: The end of the input file was detected before it was expected. Likely caused by incorrect or corrupt input files.

ERR KLNG
: The *k*-mer table is too small to hold a read of the size requested. This is a bug or platform-dependent behaviour. Please file a ticket either way.

ERR LOWQ
: The sequence is discarded because the quality is too low given the supplied threshold.

ERR NEGS
: The reconstruction parameters do not produce a valid sequence. Instead, they produce a negative-length sequence. This read pair is discarded.

ERR NODATA
: A FASTQ record has no sequence data. Likely caused by incorrect or corrupt input files.

ERR NOFILE
: The input file was not found or could not be read.

ERR NOFP
: The forward primer could not be matched to the forward read. Either the primer is incorrect or the read is low quality or the sequence provided is not the correct original molecule.

ERR NOQUAL
: Quality information is missing from the FASTQ file. This data is required to reconstruct the sequence.

ERR NORP
: The reverse primer could not be matched to the reverse read. See **NOFP**.

ERR NOTPAIRED
: Sequences from FASTQ files are not pairing correctly given their sequence names. Likely, the files are mismatched.

ERR OOM
: An out of memory condition has occurred. Given the memory available, assembly of this sequence is not possible. As Illumina sequencing gets longer, the amount of memory needed can be adjusted. Please file a ticket.

ERR READLEN
: The read length is too long for this version of PANDAseq. PANDAseq needs to be recompiled with a longer allowable seqeuence length; this length is kept short to improve performance.

INFO ARG[*n*]
: The *n*th command line argument that generated this output, for posterity.

INFO BESTOLP
: The best overlap parameter for a sequence.

INFO BUILD
: The parameters of a reconstructed base.

INFO MISM
: A mismatch has been identified in the reconstruction.

INFO MOD
: Information about a module.

INFO OLD
: An overlap possibility, with probability, as been identified.

INFO RECR
: The proposed reconstruction parameters.

INFO VER
: The version of PANDAseq that generated this output, for posterity.

STAT
: Some information about the assembly process. See above.

DBG FMER
: A *k*-mer has been identified in the forward read.

DBG FML
: A duplicate *k*-mer has been identified in the forward read and discarded. This might cause failure to assemble a sequence if repeated too often. See the **-k** option to correct this.

DBG RMER
: A *k*-mer has been identified in the reverse read.

ERR UNKNOWN ERROR
: Something truly unexpected has happened. This probably involves an validation module.

EXAMPLES

This will assemble a data from a run in lane 7:

pandaseq -f s_7_1.fastq.bz2 -r s_7_2.fastq.bz2 > s_7.fasta

This will assemble data from lane 7, stripping conserved regions around the prokaryotic 16S V3 region and store the results in **s_7.fasta.bz2** and store the logging output **s_7.log.bz2.**

pandaseq -f s_7_1.fastq.bz2 -r s_7_2.fastq.bz2 -p CCTACGGGAGGCAGCAG -q ATTACCGCGGCTGCTGG -G s_7.log.bz2 | bzip2 > s_7.fasta.bz2

VALIDATION MODULES

Validation modules are capable of making decisions about whether or not to keep output sequences. For example, one could write a module to check secondary structure of a RNA, or that a coding sequence contains no stop codons. To create a module, please see **pandaxs**(1). Invoking a module can be done using the **-C** option on the command line. As many modules as desired may be added. The path to the module may be followed by a colon (on Windows, a

semicolon) and arguments. For example, the following will include all sequences after **HWI-ST822:85:C05C3ACXX:1:1101:1171:2104 3:N:0:TAGACA** in the input file:

pandaseq -f s_7_1.fastq.bz2 -r s_7_2.fastq.bz2 -C "after:HWI-ST822:85:C05C3ACXX:1:1101:1171:2104 3:N:0:TAGACA" > s_7.fasta

INCLUDED MODULES

There are some included modules:

"after:*identifer*"
: Assemble only the sequences after (and including) the sequence specified. This is done in file order.

"before:*identifer*"
: Assemble only the sequences before (and excluding) the sequence specified. This is done in file order.

completely_miss_the_point
: This can be used to only include sequences with perfect overlap regions. You shouldn't want to do it. The whole point is to fix sequences which are probably good. Moreover, assuming that the sequencer is right in the overlap region and in the non-overlapping regions requires an unsound leap in statistics. My dislike has been appropriately embodied in the name of this validation module.

empty
: Sometimes, an assembled sequence can have zero length. Some downstream applications do not like this, so this filter allows removing any such sequences.

filter:*file*
: Output only the sequences whose identifiers match those in the file specified, one per line. If the file is missing, sequences are read from standard input.

min_phred:*value*
: Check the PHRED score of every base in the output sequence and make sure it is at least *value*. The threshold is based on the sequence as a whole, but this is based on the individual base scores, as they would be seen with the **-F** option.

min_overlapbits:*value*
: Check that the number of "bits saved" (Cole, et al. 2013) is above the provided value.

other_primer:*direction*:*primer*
: Sometimes, libraries are constructed with a mix of primer sequences. The allows separating the primer mix. To do this given primers, run PANDAseq twice: once with the each primer given to **-p** or **-q** and the other primer given as the *primer* option to this module. The *direction* must be specified as either **f** or **p** for the forward primer or **r** or **q** for the reverse primer. This module will reject and reads that match the primer, eliminating them from the output.

overlap_stat
: Produces a histogram of the number of overlaps that were examined for each of the sequences that assembled. This does not indicate the number of overlaps that were examined for discarded sequences.

pear
: Perform the false-positive test described in section 2.2 of Zhang 2013.

validtag:*tag1*:*tag2*:...
: Only include sequences in the output with one of the tags specified. This can be used to demultiplex sequences. This will not work well with **-B** option.

SEE ALSO

pandaseq-checkid(1), **pandaxs**(1), **gzip**(1), **bzip2**(1).

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 2012, 13:31. http://www.biomedcentral.com/1471-2105/13/31

E. Aronesty. *Comparison of Sequencing Utility Programs* TOBioiJ (2013); doi:10.2174/1875036201307010001 http://benthamscience.com/open/openaccess.php?tobioij/articles/V007/1TOBIOIJ.htm

J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis. *PEAR: A fast and accurate Illumina Paired-End reAd mergeR* Bioinformatics 2013 : btt593v1-btt593. http://bioinformatics.oxfordjournals.org/content/early/2013/10/18/bioinformatics.btt593.short

J. R. Cole, Q. Wang, J. A. Fish, B. Chai, D. M. McGarrell, Y. Sun, C. T. Brown, A. Porras-Alfaro, C. R. Kuske, and J. M. Tiedje. *Ribosomal Database Project: data and tools for high throughput rRNA analysis* Nucl. Acids Res. Database issue: first published online 27 Nov 2013; doi: 10.1093/nar/gkt1244 http://nar.oxfordjournals.org/content/early/2013/11/26/nar.gkt1244.full

T. Magoc, and S. Salzberg. *FLASH: Fast length adjustment of short reads to improve genome assemblies.* Bioinformatics 27:21 (2011), 2957-63.a http://ccb.jhu.edu/software/FLASH/FLASH-reprint.pdf

R. C. Edgar. *personal communication*
