BBAP

BLAST-Based Assembly Pipeline

User Manual

v1.0

You-Yu Lin

National Taiwan University

Introduction

BBAP is a novel assembly pipeline developed for the assembly of metagenomics sequence data sets and capable of performing both *de novo* and reference assembly. BBAP, which implements a BLAST-based greedy algorithm, provides a strong tool for the assembly of highly polymorphic metagenomics data sets with increased assembly efficiency and accuracy.

System requirements

BBAP requires Perl and basic Linux command lines. BBAP was developed with Perl v5.10.0 built for x86_64-linux-thread-multi on a Linux v2.6.29.6 platform.

Installation

The BBAP package includes individual standalone perl scripts along with pipeline perl scripts that call upon the individual scripts. In addition, BLAST is required for BBAP assembly. BLAST is available at ftp://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/ and instructions at http://www.ncbi.nlm.nih.gov/books/NBK279690/. Steps for installing BBAP are:

1. Download the BBAP package at http://homepage.ntu.edu.tw/~youylin/BBAP.rar.
2. Extract the downloaded BBAP package into a single directory.
3. Download and install BLAST accordingly from NCBI.

Assembly instructions

1. For *de novo* assembly:

a) perl -w QC_SB_AC_masterPipeline.pl -p *pipeline_directory* -F *sequence_format* -o *output_heading* -O *output_directory* -f *sequence_file* -b *BLAST_exec_directory* -q *quality_threshold* -Q *Illumina_format* -A trim_start -B *trim_end* -r *reverse_complement* -c *redundancy_threshold* -e *e-value_threshold* -a *num_CPU* -i *cluster_identity_threshold* -l *cluster_length_threshold* -C *cluster_size_threshold* -I *alignment_identity_threshold* -L *alignment_length_threshold* -R *degenerate_threshold*

b) Example 2: perl -w QC_SB_AC_masterPipeline.pl -p /BBAP/ -F 1 -o DenovoExample1 -O ./ DenovoExample1 -f Example_Fasta.fastq -b /BLAST/bin/

c) Example 2: perl -w QC_SB_AC_masterPipeline.pl -p /BBAP/ -F 1 -o DenovoExample2 -O ./ DenovoExample2 -f Example_Fasta.fastq -q 15 -Q 33 -A 0 -B 0 -r 2 -c 2 -e 1e-5 -b /BLAST/bin/ -a 1 -i 75 -l 75 -C 1 -I 75 -L 75 -R 0.4

2. For reference assembly:

a) perl -w QC_SB_AC_reference_assembly_masterPipeline.pl -p *pipeline_directory* -F *sequence_format* -o *output_heading* -O *output_directory* -f *sequence_file* -S *reference_sequence* -b *BLAST_exec_directory* -q *quality_threshold* -Q *Illumina_format* -A trim_start -B *trim_end* -r *reverse_complement* -c *redundancy_threshold* -e *e-value_threshold* -a *num_CPU* -i *cluster_identity_threshold* -l *cluster_length_threshold* -C *cluster_size_threshold* -I *alignment_identity_threshold* -L *alignment_length_threshold* -R *degenerate_threshold*

b) example: perl -w QC_SB_AC_reference_assembly_masterPipeline.pl -p /BBAP/ -F 2 -o ReferenceExample1 -O ./ ReferenceExample1 -f Example_Fasta.fas -S Example_ReferenceSequence.fas -b /BLAST/bin/

c) example: perl -w QC_SB_AC_reference_assembly_masterPipeline.pl -p /BBAP/ -F 2 -o ReferenceExample2 -O ./ ReferenceExample2 -f Example_Fasta.fas -q 20 -Q 33 -A 0 -B 0 -r 2 -c 2 -e 1e-5 -S Example_ReferenceSequence.fas -b /BLAST/bin/ -a 1 -i 75 -l 75 -C 1 -I 75 -L 75 -R 0.4

Parameters

1. –p [pipeline_directory]: Directory containing the BBAP perl scripts.
2. –F [1/2/3]: Specifies the sequence format of the sequence file. 1, fastq format; 2, fasta format; 3, unique fasta format.
3. –o [output_heading]: Output file heading for files generated, including unique fasta files, BLAST result files, cluster files, consensus sequence fasta file, statistics file, and log file.
4. –O [output_directory]: Output directory heading for directories containing alignment files, BLAST result files, consensus sequence fasta file and cluster sequence fasta files.
5. –f [sequence_file]: Sequence file to be assembled. File format specified by –F parameter.
6. –q [integer, default = 20]: Quality threshold. Reads with nucleotide quality score lower than the quality threshold will be excluded.
7. –Q [integer, default = 33]: Conversion constant for translating ASCII to Phred quality score. Required for fastq format sequence files only; input value will be ignored for other formats. For Illumina 1.8+, 33; Illumina 1.3+, 64.
8. –A [integer, default = 0]: Length to trim from the start of each raw read. Required for fastq format sequence files only; input value will be ignored for other formats.
9. –B [integer, default = 0]: Length to trim from the end of each raw read. Required

for fastq format sequence files only; input value will be ignored for other formats.

10. –r [1/2, default = 2]: Whether to collapse reverse complement reads into a single unique read. 1, collapse; 2 (or any non 0 non 1 integer), do not collapse and view reverse complement reads as separate reads.
11. –c [integer, default = 1]: Redundancy threshold for unique reads. Unique reads with redundancy lower than the redundancy threshold are excluded from further analyses.
12. –b [BLAST_exec_directory]: Directory containing BLAST executable *blastall*
13. –a [integer, default = 1]: BLAST parameter designating the number of processors to use on of processors.
14. –i [real number, default = 85]: BLAST identity threshold for BBAP clustering. Reads with BLAST identity lower than the cluster identity threshold are not clustered together.
15. –l [real number, default = 85% of read length]: BLAST length threshold for BBAP clustering. Reads with BLAST length lower than the cluster length threshold are not cluster together.
16. –C [integer, default = 1]: Cluster size threshold. Clusters with number of unique reads lower than the cluster size threshold are excluded from further analyses.
17. –I [real number, default = 85]: Alignment identity threshold. Cluster reads with BLAST identity lower than the alignment identity threshold are not aligned together.
18. –L [real number, default = 85% of read length]: Alignment length threshold. Cluster reads with BLAST length lower than the alignment length threshold are not aligned together.
19. –R [0..1, default = 0.2]: Degenerate threshold. Nucleotides with frequency lower than the degenerate threshold are excluded from the consensus nucleotide.
20. –S [reference_sequence]: Reference sequence file.

FAQ

1. **How to determine an appropriate redundancy threshold?**

   The redundancy threshold is used to exclude low frequency unique reads, which represent low frequency polymorphism or low frequency sequencing errors. For BBAP *de novo* assembly, the computation time required for the self-BLAST step increases exponentially as the size of the sequence file increases linearly. Therefore, the optimal redundancy threshold is dependent of the size and polymorphic level of the sequence file, sequencing error rate, and most importantly computational capacity. Generally, plotting the redundancy distribution of the data set is an informative method to provide insight towards determination of the optimal threshold value.

2. **How to determine BLAST identity and/or length thresholds?**
   Higher identity and/or length thresholds increase the assembly accuracy, but also decrease the amount of polymorphism for each cluster and/or scaffold, which in turn increases the number of resulting assembled scaffolds. Lower identity and/or length thresholds result in longer and less fragmentized scaffolds, but also increased polymorphism and probability of inaccurate alignments. Specifically, identity thresholds are more related to the tolerance of SNPs and nucleotide polymorphism levels, whereas length thresholds have more influence on the clustering and alignment of structural variations.
   Generally, the assembly results with an identity threshold of 85 (i.e. 85%) and a length threshold equal to 85% of the read length are not too fragmentized but also not too stringent. However, the optimal setting differs for each data set and requires further adjustments.
3. **What is the difference between cluster identity/length thresholds and alignment identity/length thresholds?**
   BBAP assembles data sets by clustering reads into clusters, and then aligns the reads of each cluster into contigs/scaffolds. The alignment phase serves as a second clustering phase but with more detailed outputs. In addition, the clustering phase processes all BLAST results that meet the cluster identity/length thresholds, whereas the alignment phase only processes the top BLAST result (that also meets the alignment identity/length thresholds) for each read. Overall, the clustering and alignment phases were split apart for computational reasons. Therefore, the alignment identity/length thresholds are influential on the accuracy, length, and diversity of the final assembly results, whereas the cluster identity/length thresholds are more related to computational efficiency. Additionally, the alignment identity/length thresholds should be equal to or more stringent than the cluster identity/length thresholds.
4. **Does BBAP output nucleotide frequency data?**
   BBAP assembly does not directly provide nucleotide frequency data. However, the BBAP package includes two standalone perl scripts (*NTfreq* and *NTfreq_excludeSINGLETON*) that can calculate the nucleotide frequency data of BBAP assembly generated alignment files. *NTfreq* calculates nucleotide frequency straightforward, whereas *NTfreq_excludeSINGLETON* performs an additional quality filter by excluding singleton nucleotide alleles (i.e. nucleotide alleles only sequenced by a single unique read) prior to calculating nucleotide frequencies.