The BLASTER suite Documentation

Hadi Quesneville

Bioinformatics and genomics
Institut Jacques Monod, Paris, France
http://www.ijm.fr/ijm/recherche/equipes/Bioinformatique-genomique

Last modification: 05/09/06

The BLASTER programs suite are C++ programs developed for transposable element search and their annotation in large eukaryotic genome sequence. But we think that the tools are generic enough to be used in other contexts.

The suite is composed by three programs BLASTER, MATCHER and GROUPER. The results of BLASTER can be treated further by the MATCHER and GROUPER program, but these two last programs can be also used in conjunction with other programs such as RepeatMasker, CENSOR, BLAST, BLAT, ... if their outputs are correctly formated as a BLASTER output file (*i.e.* *.align).

BLASTER

BLASTER compares two sets of sequences: a query databank against a subject databank. For each sequence in the query databank, BLASTER launches one of the BLAST programs (blastn, tblastn, blastx, tblastx, blastp, megablast) (Altschul et al., 1990, 1997) to search the subject databank. BLASTER can use either WU-BLAST or NCBI-BLAST programs. It cuts long sequences before launching BLAST and re-assembles the results afterwards. Hence, it can work on whole genomes, in particular to compare a genome with itself to detect repeats. The results of BLASTER can then be treated by the MATCHER and GROUPER programs described below.

Commands

```
usage: blaster [option]
Options are:
-q:
        query bank name, default=<not set>
-s:
        subject bank name, default=<not set>
-a:
        all_by_all, default=FALSE
-W:
        use WU-BLAST, default=NCBI-BLAST
-n:
        blast name, default=blastn
-p:
        blast options, default=
-l:
        fragment cutter length, default=50000
-o:
        fragment cutter overlap, default=100
-w:
        minimum cutter word, default=11
-e:
        extention of cutted file, default=_cut
-S:
        sensitivity, default=0 (3 levels for blastn, 3 being the most
          sensitive)
-E:
        e-value filter, default=1e-10
-I:
        identity filter, default=0
-L:
        length filter, default=20
-b:
        output filename prefix, default=<not set>
-B:
        output filename prefix (no time stamp), default=<not set>
-r:
        force re-run all, default=Off
-P:
        only prepare the data default=Off
```

Outputs:

Several files are produced by BLASTER.

`*_cut` : preprocessed file for use with blast (cut in chunk and remove N stretches)

`*_cut.nhr, *_cut.nin, *_cut.nsq, ...` : bank preprocessed by the NCBI blast formatdb program

`*_cut.xnd, *_cut.xns, *_cut.xnt, ...` : bank preprocessed by the WU blast xdformat program

`*.Nstretch.map` : coordinates of the removed N stretches

`*.param` : dump of the used parameters

`*.raw` : raw BLAST results

`*.seq_treated`: sequences treated by BLASTER. Used to restart in case of failure.

`last_time_stamp.log`: a file containing the last_time stamp number used.

`*.align` : results of BLASTER in a tabular format. Columns are separated by tabulations

col1: query sequence name

col2: match start coordinate on the query sequence

col3: match end coordinate on the query sequence

col4: subject sequence name

col5: match start coordinate on the subject sequence

col6: match end coordinate on the subject sequence

col7: BLAST E-value

col8: BLAST score

col9: identity percentage

MATCHER

MATCHER has been developed to treat BLASTER results and to map the matches (HSPs) of the subject sequences on the queries. But it can be used in conjunction with other programs such as RepeatMasker or Censor.

Cross hits are filtered as overlapping hits. Here, overlapping means also included. So when two HSPs overlap on the genomic sequence, the one with the best alignment score is kept, the other is truncated such as non-overlapping region remains on the HSP. As a result of this procedure an HSPs is totally removed only if it is included in a longer one with a best score. By this way, nested elements are kept.

Long insertions (or deletions) in one of two homologous sequences result in two HSPs, instead of one with a long gap. Thus the remaining HSP are linked by dynamic programming as in FASTA algorithm. A score is calculated by summing HSP scores and substracting a gap penalty (x times the gap length) and a mismatch penalty (x times the mismatch length region).

The algorithm is inspired by the first part of the algorithm of sim2 described in Chao et al. (1995) CABIOS, Vol. 11, pages 147-153. But it is modified to make local alignment as follows.

1) An HSP is chained with a chain of others only if its score is less than the score of the resulting chain that links it. Thus, the chaining is stopped if the score of the resulting chain of HSP is less than the HSP not chained. The best scoring chain is kept.

2) Then to identify other HSP chains, the chain previously found is removed, and we search again for the next best HSP chain. This is done iteratively until no chain is found.

3) This algorithm is repeated independently for HSP matching in strand +/+, ±, and -/+

A maximum of x% of overlap between the HSP is authorized.

Commands

```
usage: matcher [option]
Options are:
-m:
        matches text file (in BLASTER *.align format)
-q:
        query bank name, default=<not set>
-s:
        subject bank name, default=<not set>
-j:
        join matches, default=false
-i:
        identity tolerance to join matches, default=2
-g:
        gap penalty to join matches, default=0.05
-d:
        distance penalty to join matches, default=0.2
-c:
        authorized overlap to join matches, default=20
-E:
        e-value filter, default=1e-10
-I:
        identity filter, default=0
-L:
        length filter, default=20
-b:
        output filename prefix, default=<not set>
-B:
        output filename prefix (no time stamp), default=<not set>
-a,
        all conflicting subject default=FALSE
```

Outputs

`*.clean_match.fa` : sequences of the matching region. This is a concatenation of the joined fragments

`*.clean_match.map` : coordinates of the whole matching region (boundaries of the joined fragments) on the query in the following tabulated format (the separator is a tabulation):

col1: the subject sequence name

col2: the query sequence name

col3: start coordinate

col4: end coordinate

`*.clean_match.param` : dump of the used parameters for MATCHER

`*.clean_match.path` : results of MATCHER in a tabular format. Columns are separated by tabulations.

col1: path number. Joined fragments have the same path number

col2: query sequence name

col3: match start coordinate on the query sequence

col4: match end coordinate on the query sequence

col5: subject sequence name

col6: match start coordinate on the subject sequence

col7: match end coordinate on the subject sequence

col8: BLAST E-value

col9: BLAST score

col10: identity percentage

`*.clean_match.tab` : summary results of MATCHER in a tabular format. One line per joined fragment. Columns are separated by tabulations.

col1: query sequence name

col2: whole match start coordinate on the query sequence

col3: whole match end coordinate on the query sequence

col4: length on the query sequence

col5: length in percentage of the query sequence

col6: length on the query relative to the subject length in percentage

col7: subject sequence name

col8: whole match start coordinate on the subject sequence

col9: whole match end coordinate on the subject sequence

col10: BLAST E-value

col11: BLAST score

col12: identity percentage

col13: path number

GROUPER

GROUPER has been developed to treat BLASTER results. It uses HSPs to gather similar sequences into groups by simple link clustering. An alignment belongs to a group if one of the two aligned sequences already belongs to this group over 95% of its length. Groups that share sequence locations are regrouped into what we called a cluster. As a result of these procedures, each group contains sequences that are homogeneous in length. A given region may belong to several groups, but all of these groups belong to the same cluster.

Commands

```
usage: grouper [option]
Options are:
-m:
        match text file(in BLASTER *.align format)
-q:
        query bank name, default=<not set>
-s:
        subject bank name, default=<not set>
-j:
        join matches, default=false
-i:
        identity tolerance to join matches, default=2
-g:
        gap penalty to join matches, default=0.05
-d:
        distance penalty to join matches, default=0.2
-c:
        authorized overlap to join matches, default=20
-E:
        e-value filter, default=1e-10
-I:
        identity filter, default=0
-L:
        length filter, default=20
-b:
        output filename prefix, default=<not set>
-B:
        output filename prefix (no time stamp), default=<not set>
-G:
         min coverage length for connecting groups in a cluster,default=100
-S:
         connecting groups in a cluster if all members of one group overlap
            members of another group, default=FALSE
-C:
         coverage for group construction default=0.95
```

Outputs

Output files have the following prefix `*.align.group.c0.[0-9]+` . The number after `*.align.group.c` indicates the coverage threshold used.

`*.align.group.c0.95.fa`: repeated sequences found in fasta format. Each repeated sequence is named according to the following nomenclature: Mb[SQ][0-9]+Gr[0-9]+Cl[0-9]+

- "Mb" followed by "S"or "Q" according to the fact that this sequence was found in a query (Q) or a subject (S) sequence and a number identifying the sequence
- "Gr" followed by the group number
- "Cl" followed by the cluster number

After this name, follows the "fasta header" of the source sequence and the coordinates on it.

`*.align.group.c0.95.map`: coordinates of the sequences found. Columns are separated by tabulations.

col1: name of the repeated sequence fragment (as for fasta)

col2: query sequence name

col3: start coordinate on the query sequence

col4: end coordinate on the sequence

`*.align.group.c0.95.set`: coordinates in set format, indicating connected fragments. Columns are separated by tabulations.

col1: path number. Joined fragments have the same path number

col2: name of the repeated sequence fragment (as for fasta)

col3: query sequence name

col4: start coordinate on the query sequence

col5: end coordinate on the sequence

`*.align.group.c0.95.param` : parameters used for grouper

`*.align.group.c0.95.txt`: detailed description of the groups and clusters

`*.align.group.c0.95.cluster.dot`: Graph representation of groups links within clusters in dot format (use the dot program of graphviz package to visualize http://www.graphviz.org/)

`*.align.group.c0.95.overlap.dot`: Graph representation of sequences links in dot format (useful only for small example)

References

Publications where BLASTER is used:

1. Quesneville H, Nouaud D, Anxolabehere D. P elements and MITE relatives in the whole genome sequence of Anopheles gambiae. BMC Genomics. 2006 Aug 18;7(1):214 [Epub ahead of print] PMID: 16919158 [PubMed - as supplied by publisher]
2. Drouaud J, Camilleri C, Bourguignon PY, Canaguier A, Berard A, Vezon D, Giancola S, Brunel D, Colot V, Prum B, Quesneville H, Mezard C. Variation in crossing-over rates across chromosome 4 of Arabidopsis thaliana reveals the presence of meiotic recombination "hot spots". Genome Res. 2006 Jan;16(1):106-14. Epub 2005 Dec 12. PMID: 16344568 [PubMed - indexed for MEDLINE]
3. Quesneville H, Bergman CM, Andrieu O, Autard D, Nouaud D, Ashburner M, Anxolabehere D. Combined evidence annotation of transposable elements in genome sequences. PLoS Comput Biol. 2005 Jul;1(2):166-75. Epub 2005 Jul 29. PMID: 16110336 [PubMed]
4. Eiglmeier K, Wincker P, Cattolico L, Anthouard V, Holm I, Eckenberg R, Quesneville H, Jaillon O, Collins FH, Weissenbach J, Brey PT, Roth CW. Comparative analysis of BAC and whole genome shotgun sequences from an Anopheles gambiae region related to Plasmodium encapsulation. Insect Biochem Mol Biol. 2005 Aug;35(8):799-814. Epub 2005 Apr 12. PMID: 15944077 [PubMed - indexed for MEDLINE]
5. Quesneville H, Nouaud D, Anxolabehere D. Detection of new transposable element families in Drosophila melanogaster and Anopheles gambiae genomes. J Mol Evol. 2003;57 Suppl 1:S50-9. PMID: 15008403 [PubMed - indexed for MEDLINE]