BlastGraph

Intensive approximate pattern matching in a de-Bruijn graph or a sequence graph. Ver 2.0.

User's guide, Ver 2.0 – May 2013

By Holley Guillaume and Peterlongo Pierre.
Contact: pierre.peterlongo@inria.fr

Licence

BlastGraph is a new tool for computing intensive approximate pattern matching in a sequence graph or a de-Bruijn graph. Given an oriented graph and a set of query sequences, it detects paths in the graph on which query sequences align at most at the given edit distance.

Usage

BlastGraph may be used for (not limited to) :

- Detect patterns in a de-Bruijn graph or a sequence graph
- Check the presence of homologs of a set of sequences in a de-Bruijn graph or a sequence graph
- ...

Please refer to the original article for more details :
Guillaume Holley, Pierre Peterlongo. BlastGraph: intensive approximate pattern matching in string graphs and de-Bruijn graphs. PSC 2012, Aug 2012, Prague, Czech Republic

IN/OUT in a few words

Initially, BlastGraph takes two files as input :

- *graph.xgmml* : contains a de-Bruijn graph or a sequence graph in XGMML format. XGMML format is an XML format specialized for graphs
- *sequence.fasta* : contains a set of query sequences in FASTA format.

You can find examples of syntax in the /bin/Debug folder of the BlastGraph package

Note that these files are not limited in size but :

- The memory used and the execution time grow with the size and the number of graph nodes. During the alignment step, in the worst case, the memory used is proportional with the size of the biggest sequence of the nodes in the graph.
- The execution time increases with the number of query sequence.

In output, Blastree displays for each query sequence, the paths (nodes ID) where the query is matching.

Download

http://alcovna.genouest.org/blastgraph/

Installation

BlastGraph runs on every Linux OS with GCC installed. It has been tested on Fedora-14 Laughlin 64bits and Ubuntu 12.10 Quantal Quetzal 64bits.

- « *tar xvzf BlastGraph.tar.gz* » to decompress the file
- « *cd ./BlastGraph* » to enter in the newly decompress BlastGraph folder
- « *sudo make* » to compile BlastGraph (can take several minutes)

Run

The command to launch BlastGraph is :

BlastGraph <name_file_graph.xgmml> <name_query_file.fasta> <node_attribute> <edge_attribute> <seed_size> <overlap_size> [-options]

- *<node_attribute>* : the name of the attribute in the XGMML file (in the <node> marker) that contains the sequence of each node

- *<edge_attribute>* : the name of the attribute in the XGMML file (in the <node> marker) that contains the two letters indicating if the sequences of the source node and target node have to be read in forward (F) or reverse-complement (R)
- *<seed_size>* is the seed size used to anchor query sequences on the graph
- *<overlap_size>* is the size of the overlaps between the sequences source and target nodes

[-options] :
-r reverse_complement_attribute : take into account the reverse-complement identified by reverse_complement_attribute in the graph. By default, compute the reverse-complement of each sequence in each node identified by node_attribute.
-x value : use this value as treshold for the X-DROP heuristic. 0 by default.
-l value : use this value as the maximum number of characters per line in the query file. 120 by default.

Use case

The graph file used for this use case is *graph10000.xgmml* situed in /bin/Debug in the BlastGraph package. The query file is a FASTA file that contains the two first lines (one sequence) of the file *queries10000.fasta* in /bin/Debug in the BlastGraph package.

./BlastGraph graph10000.xgmml query1.fasta label label 20 30

The result displayed is :

Number of nucleotides in the graph : 1849460

Number of nodes in the graph : 59660

Number of edges in the graph : 99516

Parsing XGMML file done

Parsing FASTA file done

Indexing graph done

Time after indexing graph :
Real time : 96 cs
User time : 92 cs
System time : 2 cs

One or more alignments found for query 0

In node 1753
In node 12831
In node 45228

In node 47776
In node 35548
In node 4986

Number of queries found : 1
Number of queries not found : 0

Time after aligning graph :
Real time : 97 cs
User time : 92 cs
System time : 2 cs