NAME

GeneSeqer − Gene Identification by Spliced Alignment

SYNOPSIS

GeneSeqer [**−hv**] [**−s** *species*] [**−dD** *dbest(s)*] [**−eE** *estdbn*] [**−k**] [**−m** *maxnest*] [**−M** *sgmntsze*] [**−x** *wsize*] [**−y** *minqHSP*] [**−z** *minqHSPc*] [**−w** *minESTc*] [**−p** *prmfile*] [**−qQ** *qpdbn(s)*] [**−I** *matname*] [**−a** *from*] [**−b** *to*] [**−frR**] [**−oO** *outfname*] [**−c** *fcountA*] [**−C** *fcountB*] [**−lL** *libfname*] [**−g** *gbfname(s)*]

DESCRIPTION

GeneSeqer is a gene identification tool based on spliced alignment or "spliced threading" of ESTs (cDNAs; Reference 1) or target protein sequences (Reference 2) with a genomic query sequence. In a spliced alignment, aligned residues in the genomic sequence are assigned exon status. Introns are identfied as large gaps in the alignment, typically (but not necessarily) flanked by the consensus GT and AG dinucleotides at the donor and acceptor sites, respectively. The optimal alignment is derived by scoring for both sequence similarity and potential splice site strength. The program is designed to handle alignment of a large number of ESTs on a long genomic query sequence (BAC size). Therefore, the ESTs are pre-screened, and only ESTs with sufficient significant matching are fully aligned. Optionally, target protein sequences are optimally aligned directly to implied translation products of the genomic DNA. This mode does not involve prescreening and is limited to short genomic sequence segments.

The **GeneSeqer** program was developed in the group of Prof. V. Brendel and is freely available under the GNU General Public Licence at http://brendelgroup.org/bioinformatics2go/GeneSeqer.php/. Correspondence relating to **GeneSeqer** should be addressed to

Volker Brendel
Indiana University
Department of Biology
212 South Hawthorne Drive
Simon Hall 205C
Bloomington, IN 47405
U.S.A.
phone: (812) 855-7074
email: vbrendel@indiana.edu

REFERENCES

1. Usuka, J., Zhu, W. & Brendel, V. (2000)
Optimal spliced alignment of homologous cDNA to a genomic DNA template.
Bioinformatics 16, 203-211.

2. Usuka, J. & Brendel, V. (2000)
Gene structure prediction by spliced alignment of genomic DNA with protein sequences: Increased accuracy by differential splice site scoring.
J. Mol. Biol. 297, 1075-1085.

3. Brendel, V., Xing, L. & Zhu, W. (2004)
Gene structure prediction from consensus spliced alignment of multiple ESTs matching the same genomic locus.
Bioinformatics 20, 1157-1169.

4. Sparks, M.E. & Brendel, V. (2005)
Incorporation of splice site probability models for non-canonical introns improves gene structure prediction in plants.
Bioinformatics 21 Suppl.3, iii1-iii11.

5. Zhu, W. & Buell, C.R. (2007) *Improvement of whole-genome annotation of cereals through comparative analyses.*
Genome Res. 17, 299-310.

OPTIONS

−h Generate HTML output [default: simple text]. The HTML output should be viewed with an HTML browser. It provides scrolling within the text and links to the NCBI databases and BLAST server.

−v Generate verbose output (display the genomic DNA sequence and its base composition statistics).

−s *species*
Set *species* to select the most appropriate splice site models. This parameter must be specified. Options: "human", "mouse", "rat", "chicken", "Drosophila", "Daphnia", "nematode", "yeast", "Aspergillus", "Arabidopsis", "maize", "rice", "Medicago", "generic".

−dD *dbest(s)*
Read EST sequence data from pre-processed library file(s) *dbest(s)*.

−eE *estdbn*
Read EST sequence data from library file *estdbn*.

−k Assume fixed 5' to 3' transcript orientation of supplied EST sequences.

−m *maxnest*
Display at most *maxnest* spliced EST alignments per genomic DNA input [default: 500].

−M *sgmntsze*
Process input sequence in segments of size *sgmntsze* [default: 240000; increased by *np* * 20000 for the MPI version with *np* processors]. The entire input sequence is tiled with overlapping segments, thus all significant alignments will be reported independent of the setting. Decrease or increase the segment size depending on how much memory is available. The maximal value of *sgmntsze* is set to *MAXSGMNTSZE* in sahmt.h and can be changed at compile time (see below).

−x *wsize*
Specify the word size for the initial exact match search step. Default: *wsize* = 12 for Arabidopsis and maize, 16 otherwise. Must be set to >= 12. Increasing *wsize* will improve search speed and selectivity but reduce sensitivity.

−y *minqHSP*
Specify the minimum quality value of alignment HSPs to be pursued. Default: *minqHSP* = 12 for Arabidopsis and maize, 16 otherwise. Must be set to >= 'wsize'. Increasing *minqHSP* will improve search speed and selectivity but reduce sensitivity.

−z *minqHSPc*
Specify the minimum quality value of alignment HSP-chains to be pursued. Default: *minqHSPc* = 30 for Arabidopsis and maize, 40 otherwise. Must be set to >= 'minqHSP'. Increasing *minqHSPc* will improve search speed and selectivity but reduce sensitivity.

−w *minESTc*
Specify the minimum EST coverage for an alignment HSP-chain to be pursued. Default: *minESTc* = 0.0 (no restriction). Must be set to a value between 0.0 and 1.0. Increasing *minESTc* will improve search speed and selectivity but reduce sensitivity. Alignments missed by setting *minESTc* to high coverage would include those with sequence errors (or short exons) at either 5'-

or 3'-end such that these EST parts do not participate in high-quality matches making up the HSP-chain that triggers the dynamic programming alignment. Note that the values of *MinWidthOfEST* and *MinWidthOfGDNA* (set in *include/mytype.h*) exclude short HSP-chains based on absolute length (both parameters set to 50 by default; for specialty applications such as matching short tags across exon boundaries, **GeneSeqer** would have to be recompiled with smaller values for these parameters). The idea for *minESTc* was kindly provided by W. Zhu (Reference 5).

−p *prmfile*
Read parameters for EST matching from file *prmfile*.

−qQ *qpdbn(s)*
Read target protein sequence data from library file(s) *qpdbn(s)* (FASTA-format).

−I *matname*
Read amino acid substitution scoring matrix from file *matname*.

−a *from*
Analyze genomic sequence from position *from* [default: 1].

−b *to* Analyze genomic sequence up to position *to* [default: end of sequence].

−f Analyze forward strand.

−r Analyze reverse strand.

−R Analyze both strands [default].

−oO *outfname*
Redirect output to file *outfname* [default: stdout; if 'outfname' is set with the -o option, then on-the-fly output will be directed to stdout].

−c *fcountA*
If multiple genomic input sequences are supplied with the -lLg options, skip the first sequences and process starting from sequence number *fcountA*.

−C *fcountB*
If multiple genomic input sequences are supplied with the -lLg options, skip any sequences after the sequence numbered *fcountB*.

−lL *libfname*
Read (multiple) sequence data from library file *libfname* (FASTA-format).

−g *gbfname(s)*
Read nucleic acid sequence data from GenBank file(s) *gbfname(s)*. If specified, the -g option must be last.

USAGE

Input file format

Genomic DNA input: Sequences should be in the one-letter-code ({a,b,c,d,g,h,i,k,m,n,q,r,s,t,u,v,w,y}), upper or lower case; all other characters are ignored during input. Multiple sequence input is accepted in *library (FASTA) file format* or in *GenBank format*.

Library (FASTA) file format refers to raw sequence data separated by identifier lines of the form starting with ">" followed by the sequence name. For options **-d**, **-e**, **-q**, and **-l**, the name of the sequence is taken to be the first string on the ">" line delimited by space, tab, |, or : starting from position 5. For example, ">gi|idnumber|something-else" is given the name "idnumber". For options **-D**, **-E**, **-Q**, and **-L**, the name of the sequence is taken to be the first string on the ">" line delimited by space, tab, |, or : starting from position 2. In the above example, the name would be "gi". Typically, this option is appropriate for sequences supplied by the user in the format ">my-sequence-name comments". The **-k** option is appropriate when all

the EST sequences are represented in correct 5' to 3' transcript orientation. In this case, no alignment with the complementary strand is ever attempted.

Examples (**-d**, **-e**, and **-l** options):

```
>gi|sequence1 - upper case
ACGATTGGATCAAAATCCATGAAAGAGGGGAATCTATAGGCGGAATTGAG
CGCCAGCGACTGGCTGCCTTGGCGGGGGAGGCCTTGGCGGA

>SQ;sequence2 - upper case with numbering
       1  ACGATTGGAT CAAAATCCAT GAAAGAGGGG AATCTATAGG CGGAATTGAG
      51  CGCCAGCGAC TGGCTGCCTT GGCGGGGGAG GCCTTGGCGG A

>vb:sequence3 - lower case
acgattggatcaaaatccatgaaagaggggaatctataggcggaattgagcgccagcgac
tggctgccttggcgggggaggccttggcgga

>vb:sequence4 - mixed format
       1  ACGATTGGAT CAAAATCCAT GAAAGAGGGG AATCTATAGG GGGGGGATCT
cgccagcgac
        tggctgcct       tggcggggg       AGGCCTTGGCGGA
```

GenBank format refers to raw sequence data with possible annotations as in standard GenBank files. Minimal requirements are the LOCUS and ORIGIN lines. Multiple sequences must be separated by // lines. Note that you can use the **-c** and **-C** options to specify only certain sequences from the input files to be processed. These options are handy if you want to split the processing of a multi-sequence input file over several CPUs, for example.

EST database input: The EST database for spliced alignment may be supplied in pre-processed form with the *-dD dbest(s)* option or as a sequence file in library format with the *-eE estdbn* option. Pre-processing must be performed prior to the **GeneSeqer** application with the accompanying program **MakeArray** (command [for each file dbest]: MakeArray dbest).

Query protein input: Query protein sequences for spliced alignment may be supplied with the *-qQ qpdbn(s)* option, where *qpdbn(s)* are sequence files in library format. Spliced alignment on the protein level will only be performed for genomic DNA sequences of lengths not exceeding the parameter MAXGLGTH (default: 15000).

Output format

Output is directed to standard output (default) or to the file specified with the *-o outfname* command line argument. For each significantly matching EST, the predicted gene structure based on an optimal spliced alignment is displayed. The upper line gives the genomic DNA and the lower line gives the EST sequence. Identities are indicated by vertical bars in the center line. Introns are indicated by dots, gaps in the exons by '_'. For protein spliced alignments, the alignment gives the genomic DNA sequence, its inferred protein translation (one-letter-code), and the matching parts of the target protein sequence. Identical residues are linked by "|", positively scoring substitutions by "+", and zero scoring substitutions by "." according to the amino acid substitution scoring matrix used in the alignment (BLOSUM62 by default). Coordinates for the predicted exons and introns are given in the list preceding the alignment. Exons are assigned a normalized similarity score (1.000 represents 100% identity). Per position alignment scores can be changed by changing the parameters PDG, IDS, MMS, NNS, and DLS in 'prmfile' (Reference 1). For introns, the list gives

adjusted P-values of the donor and acceptor sites (References 3 and 4) as well as a similarity score (s) based on the sequence similarity in the adjacent 50 bases of exon. Introns shorter than a specified minimal length (parameter MIN_INTRON_LENGTH in 'prmfile') are penalized in the optimal alignment (predicted introns of size MIN_INTRON_LENGTH or less are flagged by '??' in the output). Similarly, a minimal exon size is set by the parameter MIN_EXON_LENGTH. The alignment ends are forced to terminate with MIN_NBR_ENDMATCHES identities (default value: 2) and to be indel-free for MIN_EXON_LENGTH positions to avoid display of poor quality alignment ends.

If the *-o outfname* option is specified, EST alignments in the order in which they are produced are piped to the standard output. The *outfname* file displays the sorted and quality-screened EST alignments that make up the consensus gene predictions (see below).

Special lines:
MATCH gDNAx cDNAy scr lgth cvrg Z

where gDNA = name of genomic DNA sequence; x = + (forward strand) or - (reverse strand); cDNA = name of cDNA sequence; y = + (forward strand) or - (reverse strand); scr = alignment score; lgth = cumulative length of scored exons; cvrg = coverage of genomic DNA segment (Z = G) or cDNA (Z = C) or target protein (Z = P), whichever is highest.

PGS_gDNAx_cDNAy (a b,c d, ...)
or
PGS_gDNAx_qp (a b,c d, ...)

where gDNA = name of genomic DNA sequence; x = + (forward strand) or - (reverse strand); cDNA = name of cDNA sequence; y = + (forward strand) or - (reverse strand); qp = name of target protein; a, b, c, d, ... = exon coordinates.

The MATCH and PGS lines are useful for summarizing the search results for an application involving multiple genomic DNA sequences and multiple ESTs or target proteins (use a combination of 'egrep' and 'sort'). PGS = Predicted Gene Structure (GenBank CDS-styled exon coordinates). A "hqPGS" line following the alignment gives the coordinates of the high-quality alignment parts used to build consensus gene predictions (see below).

Consensus gene predictions

For EST matching, the overall gene predictions are summarized at the end of the output file in a section labeled "Predicted gene locations". In brief, individual EST alignments are culled to remove weak terminal exon predictions and then assembled into groups of overlapping alignments with respect to the genomic DNA coordinates (the maximal gap within a cluster is set by the parameter JOIN_LENGTH in 'prmfile'). This quality-adjustment may result in complete removal of weak EST alignments. If the adjustment only removes terminal exons, then the shortened alignment is used for generating the consensus gene predictions, although the complete alignment is still displayed for each EST for reference. Criteria for culling weak terminal exons are governed by the parameters TINY_EXON, SHORT_EXON, LONG_INTRON, POOR_EXON_SCORE, POOR_DONOR_SCORE, and POOR_ACPTR_SCORE specified in 'prmfile'.

Each overlapping cluster of alignments is indicated as a PGL (Predicted Gene Location). Within each PGL, alternative exon/intron assignments are indicated by labels AGS (Alternative Gene Structure), followed by a summary of the predicted gene structure and scores and the individual PGS lines. Details of the consensus building procedure are discussed in Reference 3.

COMPILATION OPTIONS

The following parameters are set in the file
GENESEQER/include/sahmt.h (change and re-compile depending on need and available memory):

MAXSGMNTSZE - maximum length of genomic DNA segment as a unit of processing (upper limit of argument *sgmntsze* to the -M option); default: 750000
MAXGLGTH - maximum length of genomic DNA segment for spliced alignment; default: 40000
MAXCLGTH - maximum length of cDNA/EST for spliced alignment; default: 16000
MAXPLGTH - maximum length of protein sequence for spliced alignment; default: 5000

For large applications, memory requirements may become limiting. In that case, first try to split individual EST database files into smaller files representing subsets.

FILES

GENESEQER/README
GENESEQER/bin
GENESEQER/data (examples)
GENESEQER/doc/GeneSeqer.1 (this file)
GENESEQER/include
GENESEQER/src

SEE ALSO

MakeArray(1), SplicePredictor(1).

NOTES

A hardcopy of this manual page is obtained by ‘man -t ./GeneSeqer.1 | lpr’.

GeneSeqer output can be graphically displayed with **MyGV**; see http://brendelgroup.org/bioinformatics2go/MyGV.php/.

AUTHOR

Volker Brendel <vbrendel@indiana.edu>