BIOINFORMATICS APPLICATIONS NOTE Vol. 19 no. 10 2003, pages 1296–1297
DOI: 10.1093/bioinformatics/btg139

GeneComber: combining outputs of gene prediction programs for improved results

Sohrab P. Shah[1], Graham P. McVicker[1,†], Alan K. Mackworth[2], Sanja Rogic[2] and B. F. Francis Ouellette[1,*]

[1]UBC Bioinformatics Centre, University of British Columbia, 950 28th Ave W, Vancouver BC, V5Z4H4 Canada and [2]Department of Computer Science, University of Brtish Columbia, 201-2366 Main Mall, Vancouver BC , V6T1Z4 Canada

Received on November 20, 2002; revised on January 15, 2003; accepted on January 20, 2003

ABSTRACT

Summary: We recently demonstrated that combining the output from Genscan and HMMgene can provide increased accuracy of gene predictions. We have created a robust software system that runs algorithms previously described on DNA sequences and provides a public web interface to the system for use by the biological community worldwide. The GeneComber system performs *ab initio* gene prediction by first taking a user inputted DNA sequence and running Genscan and HMMgene. The outputs of Genscan and HMMgene are then integrated using the EUI, GI and EUI_frame algorithms. All results are then stored into a relational database management system (RDBMS) and can then be retrieved through a web interface. The web interface provides a unified view of the GeneComber predictions by graphically overlaying outputs from Genscan, HMMgene, EUI, GI and EUI_frame. Outputs can also be retrieved in general feature format (GFF) or FASTA format. The software is written in the Perl programming language and is both dependent on and interoperable with the Bioperl toolkit. It includes high-level application programming interfaces (APIs) to run Genscan, HMMgene and a database API to insert prediction results into an RDBMS. The APIs are assembled into the *genecomber* script which is executed by the web interface or can be run directly from the Unix command line. The web interface is written in PHP and is structured so as to be easily modified for viewing data from any database that stores gene structures.

Availability: The GeneComber public web interface and supplementary information is located at http://bioinformatics.ubc.ca/genecomber. The source code is released under the GNU General Public License and is available at ftp://ftp.bioinformatics.ubc.ca/pub/genecomber/software.

*To whom correspondence should be addressed.
† Present address: EMBL - European Bioinformatics Institute, Cambridge CB10 1SD, UK.

Contact: francis@bioinformatics.ubc.ca

ALGORITHMS

The GeneComber system contains implementations of three novel algorithms: EUI, GI and EUI_frame, all of which combine results of Genscan (Burge and Karlin, 1997) and HMMgene (Krogh, 1997) and are described in previous work (Rogic *et al.*, 2002).

SYSTEM ORGANIZATION

The GeneComber system is built from three modular parts: (1) gene prediction modules; (2) a back-end database system (genecomber_db); and (3) a front-end web interface system. The parts are connected together with a script that executes data flow through the system. The *genecomber* script receives a sequence as input, runs Genscan and HMMgene, parses their outputs and integrates these outputs using EUI, GI and EUI_frame. The script then inserts the Genscan, HMMgene, EUI, GI and EUI_frame results into genecomber_db. Results can then be retrieved and displayed graphically, or textually to the user through the web interface.

Gene prediction modules

We developed gene prediction modules in the Perl programming language that are derived classes from Bioperl's (Stajich *et al.*, 2002) Bio::Tools::AnalysisResult package. We extended the Bio::Tools::Genscan module to allow for repeated use of the object in memory, created a module for parsing HMMgene output, and created modules for EUI, GI, EUI_frame, all three of which extend a generic interface module (EUII) that contains the common functions for its three implementing classes. The EUI, GI and EUI_frame modules have the internal methods for running their respective algorithms. All five modules extend Bioperl's Bio::Tools::AnalysisResult and therefore have a common interface including a method for outputting results in GFF. The modular design of the GeneComber system

allows for the addition of other gene prediction programs' results to be included in the output with little modification of the system.

Relational database

The GeneComber system uses a MySQL database system for storage and retrieval of the predicted sequence features. A relational schema was developed to hierarchically model sequences and their predicted gene structures, associated protein and mRNA sequences, and exon features. In addition, a Perl module to interface with the database was created to provide strict separation of database-dependent functionality from the rest of the system. This allows for clean substitution of relational database management systems (RDBMS), without having to modify any other parts of the system. GeneComber uses MySQL for its RDBMS for its open nature, free distribution, retrieval speed and the availability of both Perl and PHP APIs for database communication.

genecomber script

The execution of data flow is carried out by the *genecomber* script, which can be run directly from the Unix command line, or is seamlessly called by the web front end. The script has several input options, including a DNA sequence FASTA file, a GenBank accession number for a DNA sequence, or pre-computed HMMgene and Genscan reports. The script will optionally execute EUI, GI and/or EUI_frame and deposit the results in genecomber_db. Results can then be viewed as text output in GFF or GenBank flat file format in the command-line console, or through the web viewer in GFF, GenBank, FASTA, or graphical representation.

Web viewer

The web viewer is made up of an object-oriented hierarchy of PHP classes that abstracts the concept of a sequence and its associated features. This abstraction allows for easy porting of the viewer to different annotation databases and data sources and allows for an arbitrary number of analyses to be viewed simultaneously. The viewer contains distinguishable glyphs for initial, internal and terminal exons and single exon genes. Each glyph is clickable and allows the user to access extra information about the feature the glyph represents such as the corresponding sequence, probability score and the algorithm from which it was predicted. Predictions from the different programs are displayed on separate, colour coded tracks that are vertically overlayed to easily discern the predictions from Genscan, HMMgene, EUI, GI and EUI_frame. The viewer also supports full sequence navigation and zooming capabilities. In addition, the mRNA, protein and individual exon sequences of predicted genes for all the algorithms can be easily retrieved from the viewer in FASTA format for further analyses.

External dependencies

The GeneComber system requires the following external software to be installed on the Unix or Linux platform: Genscan, HMMgene1.1, Perl5.x, Bioperl 1.x, MySQL 3.2x, Apache and PHP 4.0. All external dependencies are freely available on the web, or by contacting the authors.

Performance and reporting

GeneComber is run on a Sun 4-way E-450 server (Solaris 5.7) with 450 MHz processors and 2 GB of RAM, but can easily be run on a commodity PC running the Linux operating system. Typical execution time is 60 s/100 kb of DNA sequence submitted, but is nearly instantaneous if Genscan and HMMgene reports are submitted. Sequences longer than 1 Mb are currently not accepted. Upon completion of a job, users are notified by email with a URL pointing to their results page.

Future work

We are developing an automated system to run GeneComber on NCBI's assembly of the human genome. We will make the results available for public consumption through a Distributed Annotation Server (Dowell *et al.*, 2001) track for Ensembl (Hubbard *et al.*, 2002). We are also developing tools to validate and improve the GeneComber system for genome-scale analyses.

REFERENCES

Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol*, **268**, 78–94.

Dowell,R., Jokerst,R., Day,A., Eddy,S. and Stein,L. (2001) The distributed annotation system. *BMC Bioinformatics*, **2**, 7.

Hubbard,T., Barker,D., Birney,E., Cameron,G., Chen,Y., Clark,L., Cox,T., Cuff,J., Curwen,V., Down,T. *et al.* (2002) The Ensembl genome database project. *Nucleic Acids Res.*, **30**, 38–41.

Krogh,A. (1997) Two methods for improving performance of an HMM and their application for gene finding. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **5**, 179–186.

Rogic,S., Ouellette,B. and Mackworth,A. (2002) Improving gene recognition accuracy by combining predictions from two gene-finding programs. *Bioinformatics*, **18**, 1034–1045.

Stajich,J., Block,D., Boulez,K., Brenner,S., Chervitz,S., Dagdigian,C., Fuellen,G., Gilbert,J., Korf,I. and Lapp,H., *et al.* (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.