****
****
**
** DOCUMENTATION FOR COMMAND-LINE VERSION OF GENSCAN **
**
**
** Christopher Burge, PhD **
**
** Department of Mathematics **
** Stanford University **
** Stanford, CA 94305 **
**
** cburge@mit.edu **
**
****
****

TABLE OF CONTENTS

---

1. OVERVIEW OF GENSCAN

GENSCAN is a general-purpose gene identification program which analyzes genomic DNA sequences from a variety of organisms including human, other vertebrates, invertebrates and plants. For each sequence, the program determines the most likely "parse" (gene structure) under a probabilistic model of the gene structural and compositional properties of the genomic DNA for the given organism. This set of exons/genes is then printed to an output file (the text output) together with the corresponding predicted peptide sequences. A graphical (PostScript) output may also be created which displays the location and DNA strand of each predicted exon. Unlike the majority of other currently available gene prediction programs, the model treats the most general case in which the sequence may contain no genes, one gene, or multiple genes on either or both DNA strands and partial genes as well as complete genes are considered. The most important restrictions are that only protein coding genes are considered (and not tRNA or rRNA genes, for example), and that transcription units are assumed to be non-overlapping.

The probabilistic model used by GENSCAN accounts for many of the essential gene structural properties of genomic sequences, e.g., typical gene density, the typical number of exons per gene, the distribution of exon sizes for different types of exon; and

also many of the important compositional properties of genes, e.g., the reading frame-specific hexamer composition of coding regions vs the (reading frame-independent) hexamer composition of introns and intergenic regions, and the position-specific composition of the translation initiation (Kozak) and termination signals, and of the TATA box, cap site and poly-adenylation signals. Importantly, novel models of the donor and acceptor splice sites are used which capture potentially important dependencies (interactions) between positions in these signals. For human and vertebrate sequences, separate sets of model parameters are used which account for the many substantial differences in gene density and structure observed in distinct C+G% compositional regions of the human genome and the genomes of other vertebrates.

The program and the model that underlies are described in Burge & Karlin, 1997 (see REFERENCES) and in greater detail in my thesis, which is available by anonymous ftp in PostScript format (see http://gnomic.stanford.edu/~chris).

INSTALLING GENSCAN

Make sure that you have requested the correct binary for the computer system on which you intend to install GENSCAN. Also make sure that the computer you have chosen has enough memory to run the types of sequences you want to run efficiently. As a rule, to run sequences of length N kilobases efficiently, your computer should have at least N/2 Megabytes of RAM. Longer sequences can be tried, but results may vary. For example, you might find that the program runs much more slowly or that you may get a "memory fault" error message for long sequences. In the latter case, it is probably best to break up the sequence into two or more smaller pieces and run each one separately.

Installing GENSCAN should be fairly easy. Here is a suggested sequence of steps - feel free to modify it depending on the setup of your system. The ">" symbol below represents the tcsh (or other shell) prompt.

1. First, if you have not done so already, make a directory to store files related to GENSCAN, insert the floppy disk provided into the drive and extract the files on it using the Unix bar command:

```
> mkdir GENSCAN

> cd GENSCAN

> bar -xvf /dev/rfd0 .
```

   (Here, /dev/rfd0 refers to the floppy disk drive: on some systems it may have a different name.)

2. List this directory (ls). The following files should be present:

```
README                  (This file: GENSCAN documentation)
HumanIso.smat           (Parameter file for human/vertebrates)
Arabidopsis.smat        (Parameter file for Arabidopsis)
```

```
        Maize.smat              (Parameter file for maize)
        HUMRASH                 (A sample human genomic sequence)
        HUMRASH.sample          (GENSCAN output for the sample sequence)
        genscan                 (The GENSCAN binary)
```

3. Make sure that you have permission to execute the binary
   and to read the parameter files:

```
        > chmod a+x genscan

        > chmod a+r *.smat
```

4. Install the binary in a directory which is in your path, e.g.,

```
        > mv genscan /usr/bin/genscan
```

   (Must be superuser to write to /usr/bin on most systems.)

5. Now make a directory for the GENSCAN parameter files and put
   them into this directory:

```
        > mkdir /usr/lib/GENSCAN

        > mv *.smat /usr/lib/GENSCAN
```

6. Type rehash and then try running the program (with no arguments):

```
        > rehash

        > genscan
```

What happened?

Hopefully, you got a message like:

```
   **********************************************************************
   **********************************************************************
   **                                                                  **
   **                         N O T I C E                              **
   **                                                                  **
   **   This program is made available ...
   ...
                          yadda yadda yadda
   ...
   **                                                                  **
   **********************************************************************
   **********************************************************************


usage: genscan parfname seqfname [-v] [-cds] [-subopt cutoff] [-ps psfname scale]

       parfname : full pathname of parameter file
                  (for appropriate organism)

       seqfname : full pathname of sequence file
                  (FastA or minimal GenBank format)

       -v       : verbose output (extra explanatory info)
```

```
       -cds     : print predicted coding sequences (nucleic acid)

       -subopt  : display suboptimal exons with P > cutoff (optional)
       cutoff   : suboptimal exon probability cutoff (minimum: 0.01)

       -ps      : create Postscript output (optional)
       psfname  : filename for PostScript output
       scale    : scale for PostScript output (bp per line)
```

If not, you might double-check your path to make sure it includes
the directory where you put genscan and make sure that all of the
permissions are set properly.

You have now successfully installed genscan.

---

3. GENSCAN INPUT

Typing "genscan" at the prompt displays the usage information
for the program (see above). In brief, the program takes two
essential command-line arguments, as well as one or more optional
arguments (the ones in square brackets above).
These arguments are described below.

OPTIONAL ARGUMENTS

-v Add some extra explanatory information to the text output.
This information may be helpful the first few times you run
the program but will soon become tiresome (that's why its optional).

-cds Print predicted CDS (coding sequences) as well as predicted
peptides.

-subopt Identify suboptimal exons. The default output of the program
is the optimal "parse" of the sequence, i.e. the highest
probability gene structure(s) which is present: the exons
in this optimal parse are referred to as "optimal exons" and
are always printed out by GENSCAN. Suboptimal exons, on the
other hand, are defined as potential exons which have probability
above a certian threshold but which are not contained in the
optimal parse of the sequence. Suboptimal exons have a variety
of potential uses. First, suboptimal exons sometimes correspond
to real exons which were missed for whatever reason by the optimal
parse of the sequence. Second, regions of a prediction which
contain multiple overlapping and/or incompatible optimal and
suboptimal exons may in some cases indicate alternatively spliced
regions of a gene (Burge & Karlin, in preparation).

The argument "cutoff" is the probability cutoff used to
determine which potential exons qualify as suboptimal exons.
This argument should be a number between 0.01 and 0.99.
For most applications, a cutoff value of about 0.10 is
recommended. Setting the value much lower than 0.10 will often

lead to an explosion in the number of suboptimal exons, most of which will probably not be useful. On the other hand, if the value is set much higher than 0.10, then potentially interesting suboptimal exons may be missed.

-ps Create the PostScript (graphical) output, which is a diagram of the locations and DNA strand of all predicted exons/genes. Exons on the "forward" (input) strand of the sequence are displayed above the sequence line; exons on the other strand are displayed below this line. Optimal exons are displayed as solid blue blocks (black on a black-and-white monitor or printer); suboptimal exons are outlined in blue. (If many overlapping suboptimal exons are present, the graphical display may become somewhat cluttered.)

The argument "psfname" is the name of the file where you want the PostScript output to go (should end in ".ps"). This argument is required whenever the "-ps" flag is used.

The "scale" argument tells the program what scale to make the PostScript image, i.e. how many base pairs to represent per line. This number must be no greater than one fourth of the length of the sequence (because only four lines fit on a page). If this argument is omitted, the program will choose a reasonable scale for the image.

The PostScript output may be printed on any PostScript printer: it can be viewed using any of several PostScript interpreters such as ghostscript/ghostview, pageview, xpsview, etc. The PostScript language was developed by Adobe Systems Inc. and the name PostScript is a registered trademark of this company.

ESSENTIAL ARGUMENTS

1. The Parameter File

The parameter file must follow a very specific format to be read correctly by GENSCAN and SHOULD NEVER BE MODIFIED. If you really have a good reason for wanting to change some of the parameters, please send email to:

chris@gnomic.stanford.edu

and we can discuss the proposed changes.

Note that separate parameter files are provided for different organisms. Currently available parameter files are:

HumanIso.smat	for human/vertebrate sequences (also Drosophila)
Arabidopsis.smat	for Arabidopsis thaliana sequences
Maize.smat	for Zea mays sequences

The full path of the parameter file is the first (mandatory) command line argument. You can either type the full path every time you run the program or save some typing by using aliases. For example, assuming that the parameter files have been installed in the directory /usr/lib/GENSCAN, you could put the following aliases in your .cshrc file (normally located in your home directory):

```
alias genvert genscan /usr/lib/GENSCAN/HumanIso.smat
alias genarab genscan /usr/lib/GENSCAN/Arabidopsis.smat
alias genmaiz genscan /usr/lib/GENSCAN/Maize.smat
```

That way (after you source .cshrc) you can simply type, for example,

```
        > genvert SEQFILE
```

to run the program on SEQFILE with the human/vertebrate parameters.

2. The Sequence File

The sequence file may be in either FastA or minimal GenBank format.
These formats are described below, with examples of each.

A sequence in FastA format begins with a single-line description,
followed by lines of sequence data. The description line is
distinguished from the sequence data by a greater-than (">") symbol
in the first column. The sequence data may be upper or lower case.
All spaces, tabs, numbers or other non-alphabet characters are
ignored by GENSCAN, with the exception of asterisks ("*"), which
are treated as unknown nucleotides (N's). GENSCAN does not
distinguish between special symbols indicating purine or pyrimidine
nucleotides such as R, Y, etc., but treats all letters other than
A, C, G, T as unknowns (N). It is usually an easy task to convert
sequences stored in other formats such as Intelligenetics, EMBL, etc.
to FastA format either by hand or using any of several standard
utitilities. A sample FastA format sequence is shown below:

```
>HC2667A cosmid clone from human chromosome 5q22
GGATCCCAGCCTTTCCCCAGCCCGTAGCCCCGGGACCTCCGCGGTGGGCGGCGCCGCGCT
GCCGGCGCAGGGAGGGCCTCTGGTGCACCGGCACCGCTGAGTCGGGTTCTCTCGCCGGCC
TGTTCCCGGGAGAGCCCGGGGCCCTGCTCGGAGATGCCGCCCCGGGCCCCCAGACACCGG
......
```

GENSCAN may also be run on files in "minimal GenBank" format.
This includes files in proper GenBank format as well as files
(perhaps constructed by hand) which contain only partial GenBank
annotation (see below). The main reason why GENSCAN has been written
so as to accept GenBank annotated as well as unannotated (FastA) files
is so that the predictive accuracy of the program can be easily measured
on sequences with known gene locations. Thus, when run on a GenBank
file containing a feature table, the program automatically compares its
predictions to the annotated CDS (coding sequence) features, displays
a summary of the annotated as well as predicted exons, and calculates
some standard measures of predictive accuracy such as nucleotide- and
exon-level sensitivity, specificity and so on. (The conventions used
to calculate these statistics are those described in Burset and Guigo,
1996 - see REFERENCES). This makes it relatively easy to check the
program's accuracy for any particular set of sequences for which the
annotated CDS features are deemed to be reliable and complete.
Of course, the program does not actually use the annotation in any
way to make its predictions: that would be silly. Therefore, in
particular, the set of predicted genes/exons will be identical for
a sequence in GenBank format as for the same sequence converted to
FastA format, i.e. without the feature annotation.

In general, GenBank format files must follow an elaborate set of formatting conventions devised by the National Center for Biotechnology Information (NCBI) in Washington. However, GENSCAN needs only a fraction (described below) of the complete GenBank annotation for its purposes, so only these lines need be present (in the proper order) in an input file which is to be read by the program. This "minimal GenBank" format is as follows.

The LOCUS and ORIGIN lines must be present. The first line of the file must be the LOCUS line and this line must be in proper GenBank format (see below). The ORIGIN line must begin with the word ORIGIN but has no other special restrictions. The sequence must begin on the line immediately after the ORIGIN line. All other lines normally present in GenBank files (e.g., ACCESSION, KEYWORDS, etc.) are optional. However, if a feature table is present, it must begin with a line:

```
FEATURES             Location/Qualifiers
```

(with exactly the same spacing/capitalization/etc. as in a GenBank file) and must end with a BASE COUNT line (as in a real GenBank file) followed by an ORIGIN line, and then the sequence. The feature table must be in correct GenBank feature table format (including spacing) -- consult the NCBI GenBank format description or look at some real GenBank files if in doubt. All features other than those labeled "CDS" are ignored. All CDS features are read and the complete set of annotated coding exons are compared to the complete set of predicted coding exons.

The format for the LOCUS line is:

```
LOCUS SEQNAME seqlen bp seqtype taxgroup date
```

where SEQNAME is the name of the sequence (any string), seqlen is the length of the sequence in base pairs (must match the actual number of sequence characters in the file), and the last three strings describe the type of sequence (e.g., ds-DNA, cDNA), the taxonomic group code, and the date. Again, the sequence may be upper or lower case and all spaces, tabs, numbers, etc. occurring after the ORIGIN line are ignored. An example of a sequence in minimal GenBank format is given below:

```
LOCUS       HUMRASH      6453 bp ds-DNA             PRI       15-MAR-1988
DEFINITION  Human c-Ha-ras1 proto-oncogene, complete coding sequence.
ACCESSION   J00277 J00206 J00276 K00954
FEATURES             Location/Qualifiers
     prim_transcript <1664..3744
                     /note="c-Ha-ras1 mRNA"
     CDS             join(1664..1774,2042..2220,2374..2533,3231..3350)
                     /note="c-Ha-ras1 p21 protein;  NCBI gi: 190891."
                     /codon_start=1
                     /translation="MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYRKQV
                     VIDGETCLLDILDTAGQEEYSAMRDQYMRTGEGFLCVFAINNTKSFEDIHQYREQIKR
                     VKDSDDVPMVLVGNKCDLAARTVESRQAQDLARSYGIPYIETSAKTRQGVEDAFYTLV
                     REIRQHKLRKLNPPDESGPGCMSCKCVLS"
     source          1..6453
                     /organism="Homo sapiens"
BASE COUNT      946 a   2287 c   2113 g   1107 t
ORIGIN      1 bp upstream of BamHI site.
        1 ggatcccagc ctttccccag cccgtagccc cgggacctcc gcggtgggcg gcgccgcgct
       61 gccggcgcag ggagggcctc tggtgcaccg gcaccgctga gtcgggttct ctcgccggcc
```

```
      121 tgttcccggg agagcccggg gccctgctcg gagatgccgc cccgggcccc cagacaccgg
      ... ..........
```

---

4. GENSCAN OUTPUT

By default, the text output of the program is directed to stdout, which means that if you simply run the program without redirecting or piping the output, it will be printed to the screen. The purpose of printing the text output to stdout is so that you (the user) have the option of either redirecting the output to a file, e.g.,

```
> genscan HumanIso.smat SEQFILE > SEQFILE.out
```

or piping the text output through some sort of filtering program to put it in a form which is more convenient for your purposes, e.g.,

```
> genscan HumanIso.smat SEQFILE | gsfilter > SEQFILE.fil_out
```

Of course, it is your responsibility to create the filtering program.

At this point, you may wish to test GENSCAN by running the program on the sample sequence HUMRASH which was provided and redirecting the output to a file, e.g. (assuming you set up the recommended aliases) type:

```
> genvert HUMRASH > HUMRASH.out
```

While running, the program should print out the "notice" message and then the following (everything between the rows of stars):

```
*********************************************************************

reading sequence file HUMRASH...   6453 bp

reading parameter file /usr/lib/GENSCAN/HumanIso.smat... 14 matrices

scoring sequence... done

running Viterbi/forward algorithms...  done

running backward algorithm... done

printing results... 1 predicted genes

processing features...

done

Run time 5 seconds :  0.8 s / kb

*********************************************************************
```

Of course, the run time may vary depending on what type of computer you are using and what other processes were running at the same time.

Other than that, the output should be similar to that shown above.
If not, then there may have been a problem with the installation.
(Check the steps listed above.) These messages are designed to give you
an idea of what the program is doing at each particular time so that if
a problem occurs you have some idea of what might have gone wrong.
They should be fairly self-explanatory (see next section).

Now compare the output HUMRASH.out to the output HUMRASH.sample
which was provided using the Unix diff command:

```
> diff HUMRASH.out HUMRASH.sample
```

Only one line should differ between the two files: the line
beginning GENSCAN 1.0 and containing the date and time the
program was run.

If any problems occurred up to this point (or in subsequent use of
the program), send email to Chris Burge at:

        chris@gnomic.stanford.edu

Include in the email the type of computer system you have, the exact
syntax of the command used to invoke the program, a description
of the problem which occurred while running the program, and any
error messages which appeared.

In addition, if any bugs are found in GENSCAN, please notify me
promptly so that I can fix them as soon as possible.

Happy gene hunting!!!

---

5. HOW GENSCAN WORKS

This section gives an indication of the sequence of steps which the
program carries out when run on a sequence.

1) The sequence and parameters are read and stored in dynamically
   allocated arrays.

2) The sequence is scored using the probabilistic models of
   coding/non-coding regions, donor and acceptor splice sites, etc.
   given in the parameter file and the resulting scores are also
   stored in dynamically allocated arrays.

3) Modified Viterbi, forward and backward recursions are performed
   which allow determination of the most likely gene structure in
   the sequence, probability of each predicted exon, and (optionally)
   all suboptimal exons with probability above a given cutoff.

4) The predicted gene structure(s), suboptimal exons, associated score
   information and the corresponding predicted peptides are then printed
   to stdout (the text output). Optionally, a PostScript output file
   which displays the locations of all predicted exons is also created
   (the graphical output).

5) The program indicates the amount of time it took to run and exits.

---

6. WEB PAGES

GENSCAN web server

http://gnomic.stanford.edu/GENSCANW.html

(Contains links to other GENSCAN-related pages)

GENSCAN email server instructions

http://gnomic.stanford.edu/GENSCANM.html

My home page:

http://gnomic.stanford.edu/~chris/index.html

---

7. REFERENCES

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

Burge, C. & Karlin, S. (1997) Gene structure, exon prediction, and alternative splicing. (in preparation).

Burge, C. (1997) Identification of genes in human genomic DNA. PhD thesis, Stanford University, Stanford, CA.

Burset, M. & Guigo, R. (1996) Evaluation of gene structure prediction programs. Genomics 34, 353-367.
