This software is OSI Certified Open Source Software.
OSI Certified is a certification mark of the Open Source Initiative.

JIGSAW README file

JIGSAW is a software program designed to construct gene models from multiple sources of prediction evidence. It can take as input gene prediction programs, sequence alignment data and splice site prediction programs, which map to a supplied genomic sequence. The standard JIGSAW program is called "jigsaw" and is designed to operated on a single user supplied fasta formatted genomic sequence.

GETTING STARTED

In order to get JIGSAW up and running you first have to decide what gene structure evidence to collect from your annotation pipeline. At a minimum it is expected that two (preferably three) different gene prediction programs are used and two different sets of alignments will be used (alignments from a protein and transcript database respectively). Although it may be possible to use less evidence it is not recommended. Additional supplemental evidence may also be useful including custom transcript databases, additional gene predictions programs and splice site prediction programs. Once the set of evidence sources is determined, JIGSAW must be trained to evaluate the accuracy of the different combinations of evidence. This is done by providing examples of known genes and comparing the output from your gene structure annotation pipeline with the structure of the known genes. This is also referred to as, "training JIGSAW". During the training process, JIGSAW builds models of what it expects to be the combinations of evidence which correspond to different parts of the gene structure. These models will then be used to piece together new genes on previously unseen data. The next section describes the training process in more detail. In addition, there is a tutorial included with this distribution, which walks you through an example of training and running JIGSAW.

TRAINING THE JIGSAW

The perl script "bin/train_jigsaw.pl" provides a template for how to train JIGSAW, and assumes there are multiple directories, each with a single contig and evidence associated with that contig. The script uses the libraries stored in the lib directory, "oc1" (uses the "mktree" binary) and "TIGR" (perl library). The script assumes that the "mktree" binary is somewhere in your path, and the TIGR/Foundation.pm perl library is in the perl library search path. An important component of training and running JIGSAW is the creation of an evidence list file which lists each type of output generated by the annotation pipeline, and specifies the filenames containing the output of the gene structure annotation pipeline. The evidence list file must list each piece of evidence in the exact same order for both training and running. In other words if you are using two gene prediction files "gp1.txt" and "gp2.txt", if the evidence list file has an order like

- "gp1.txt default geneprediction acc don coding start stop intron"
- "gp2.txt default geneprediction acc don coding start stop intron"

then the listed order when running JIGSAW must match the order used in training. The format for the evidence list file is shown when you type "jigsaw" at the command line. The difference between the training evidence list file and the running evidence list file is there is an additional line in the training evidence list file, containing the filename of the true genes for training followed by file format and the string id, "curation".

ie.:

/data/answer.genes.txt default curation

The default file format for the "true" genes is equivalent to the default format for "geneprediction". See the section "FEEDING DATA TO JIGSAW" below. Training must take place in an area that can be "written" to, as new files are created (some temporarily). Running the "train_jigsaw.pl" will create a directory with a user specified name containing decision trees, and the example evidence vectors, plus a parameter file, “param.txt”, containing some potentially useful tuning parameters. The decision tree construction can take a long time to run. To speed things up use the "-a" option to the "mktree" binary, which uses single condition splits for nodes in the tree. When using the "train_jigsaw.pl" script -t turns on the -a option for mktree.

RUNNING JIGSAW

The script, "run_jigsaw.pl" is designed to run the JIGSAW on multiple sequence contigs using a template evidence list file. It is assumed that there is a separate directory for each sequence contig and the gene prediction evidence associated with the contig, is located in that directory. The basic command line options for running JIGSAW (after training) are:

jigsaw -f "fasta seq file" -d "training directory" -m "gene output file" -e "evidence file"

-f: Genomic sequence in fasta format associated with evidence

-d: directory where training data resides.

-m: outputs gene models to this file, in a simple gff format.

-e: A simple flat file that lists the locations and type of each piece of evidence.

Each line in the file must contain the information in the following order:

1) file name containing evidence

2) file format of the file (see: FEEDING DATA TO THE JIGSAW, for supported file formats)

3) the type of evidence ("geneprediction","splicepredict ion" or "homology")

4) types of predictions the evidence makes, accepts any combination of the following:

"acc" (acceptor), "don" (donor), "start", "stop", "intron" and "coding"

A third column is a space separated list of the types of predictions the evidence

makes, any combination of acc (acceptor), don (donor), start, stop, intron and coding

are allowed.

example:

/data/genepredictor1 default geneprediction acc donor coding start stop intron

/data/proteins1 default homology acc donor coding start stop

/data/splicepredictions default splicepredict ion acc don

IMPORTANT

The evidence list order, must correspond to the evidence list order used in

training. This means that even if a particular directory does not have

all of the evidence, the file where the evidence would be in, must be listed in the

evidence file.

The same evidence file (with minor modifications) can be used for training and running.

FEEDING DATA TO JIGSAW

Input File Formats

JIGSAW reads several file formats: "default", "btab", "gff", "glimmerm" and "phat", "fgenesh", "genemarkhmm", "genscan","snap". Some formats are designed to read the output from specific programs, GlimmerM/GlimmerHMM, PHAT, FgenesH, Snap, Genemark and Genscan. These formats are compatible with the "geneprediction" evidence type. “jigsaw -S” lists the formats available for the latest version.

"btab" - is a tab delimited format which can be used to read "homology" data, only.

"gff" - (http://www.sanger.ac.uk/Software/formats/GFF/) is a more general format to specify sequence features, and can be used with any of the types "splicepredict ion", "homology" and "geneprediction". Using "gff" for "geneprediction" type predictions requires that the feature description describe exons in some format that distinguishes "Initial", "Internal", "Single" and "Terminal" exons. Any strings with the following will do: "final", "start", "stop", "end", "begin", "first", "last" If you have a program that only predicts start or stop codons, use the "geneprediction" type and gff format with a feature description of "start" and "stop".

Because data may come from any number of different sources, the hope is, it will be relatively easy for users to write their own perl scripts to parse and convert additional data formats if need be to one of the simple formats used by the JIGSAW.

"default" -

Below is a description of the "default" formats for the three expected prediction types:

1) Gene Prediction Program (string identifier is "geneprediction")

Two different formats, use whichever is easier....

"gene model #" 5’ 3’ score

example:

45 219690 220936 1.0

or

"gene model #" "exon model type" 5’ 3’ score

example:

1 single 7623 7958 1.0

"exon model type" is: internal, single, terminal or initial

2) Homology Data (string identifier is "homology")

5’ 3’ r5’ r3’ "STRING ID" "% IDENTITY" "% SIMILARITY"

example:

8853 8806 256 271 GP|7290595|gb|AAF46045.1||AE003434 18.750000 37.500000

r5' and r3' refers to the relative alignments of the homologous sequence
r5' and r3' are used only for reference
3) Splice Site Prediction (GeneSplicer format) (string identifier is "splicepre-diction")
5' 3' type score
where type is either "acceptor" or "donor"
example:
31906 31905 donor 1.0

USING THE PARAMETER FILE

The first time a jigsaw is run using a training directory, the program first checks to see if the training directory contains a "param.txt" file, if the file does not exist, it creates one using a default set of values. Contents of the default file are shown below.

```
# Intron Length Penalty
-1 10
# Intergenic Length Penalty
0 0
# Internal Exon Length Penalty
0 0
# Minimum Intron Length
10
# Minimum Intergenic Length
20
# Alignment Connection Cutoff
-1
# Donor Concensus Sequence
gt gc
# Maximum Sequence Length / Overlap Length
2000000 20000
```

The values for a given option are preceded by a comment describing what the option is. For example, the user can require JIGSAW predictions of intergenic sequence length to exceed a minimum value, the default minium is 20 bases. The donor census splice sites are defined by the user, the default is to allow both gt and gc splice sites to occur. For long sequences, it will be faster for JIGSAW to run on overlapping subsequences and merge the predictions. The user can define what a "long "sequence is, and what the overlap size should be. The default is to use 2 million base sequence windows, with ach window overlapping by 20,000 bases.

OUTPUT FORMAT

Predictions are output in the GFF format, where coordinates are 1 based (not 0). Associated with each gene is a score (column 6). The score is a log-odds ratio of the probability of the gene occurring versus the probability of the same region being intergenic. The score is normalized to correct for differences in gene length.

You can send comments, questions or problems to: jeallen@umiacs.umd.edu