Description of Supplementary Data

Katharina Hoff and Peter Meinicke

January 18, 2008

This document contains a description of data set files that were used for training and testing of the method described in

Hoff KJ, Tech M, Lingner L, Daniel R, Morgenstern B, Meinicke P: **Gene prediction on metagenomic fragments: a large scale machine learning approach**. *Manuscript submitted to BMC Bioinformatics* 2007

The data sets are available for download at
`http://orphelia.gobics.de/datasets/`

1 Training Data

Two types of training data sets were compiled: data for training of the feature preprocessing discriminants and data for training of the neural network. Details about sampling of the different data sets are given in our manuscript. The files for both types of data sets are described in the following.

1.1 Feature Preprocessing

Feature preprocessing training was performed on `orf` and `tis` examples that were extracted from complete genomes (see manuscript).

1.1.1 Mono- and Dicodon Usage

Examples that were used for training discriminants for mono- and dicodon usage are contained in the matlab file TrainingPart1_Hexa50perc.mat. This file contains the sequences of open reading frames (ORFs) that were sampled as described in our manuscript. The data structure `HexaTraining` has four fields:

- `OrfSeqs` contains the ORF sequences,
- `SpeciesLabel` indicates for each ORF the GenBank accession number of the respective species,
- `NumSpeciesLabel` is similar to `SpeciesLabel` but refers to each species with a 'running number', e.g. the species NC_007644 corresponds to the numerical species label 139.
- `CandLabel` indicates for each candidate whether it is a true gene (`1`) or a random ORF which serves as a negative example (`-1`).

1.1.2 Translation Initiation Site

Training of the discriminant for translation initiation sites (TIS) was performed with the examples stored in matlab file TrainingPart1_Tis50perc.mat. The data structure `TisTraining` contains the following fields:

- TIS sequences were extracted as a symmetric 60 bp window around start codons and are stored in `TisSeqs`.
- Each of the TIS sequences is associated with other sequences in an ORF-set (defined in our manuscript). The `OrfLabel` in combination with `SpeciesLabel` or `NumSpeciesLabel`, description see section 1.1.1, gives the ORF-set of a corresponding TIS sequence.
- The data structure also contains `SpeciesLabel`, `NumSpeciesLabel` and `CandLabel` as described in section 1.1.1.

1.2 Neural Network

The neural network was trained on `orf` examples that were extracted from artificial 700 basepair fragments that were randomly sampled from annotated genomes (see manuscript). The matlab file TrainingPart2_131207.mat contains these training examples. The following fields are stored in this data structure:

- `OrfSeqs`, `SpeciesLabel`, `NumSpeciesLabel` and `CandLabel` correspond to the description given in section 1.1.1.
- `TisSeqs` corresponds to the description in section 1.1.2.
- `TisAvailableLabel` indicates for each ORF whether a TIS candidate is available (`1`) or not (`0`).
- It is stored in `IncomplLabel` whether an ORF is complete (`0`) or incomplete (`1`).
- `SeqLengths` gives the length of each ORF (with Ns added at incomplete ends until the correct reading frame is maintained).
- `GcContent` is the GC-content of the complete fragment from which an ORF originates.

2 Prediction Model

The prediction model that resulted from training discriminants and the neural network is stored in the matlab file Model25n_090108. The model data contains four matlab structs `ORF`,`TIS`,`LEN`,`NET` which are used for scoring and classification of gene candidates.

`ORF` contains the fields:
`name: 'TrainingPart1_Hexa50perc'` (name of training data file)
`WvecContHex: [4096x1 double]` (dicodon discriminant weight vector)
`WvecContTri: [64x1 double]` (monocodon discriminant weight vector)

The weight vector dimensions are organized according to an alphabetic order of the corresponding hexanucleotides and trinucleotides, respectively. E.g. monocodon dimension 1 weights relative frequencies of AAA, monocodon dimension 2 weights relative frequencies of AAC, ...

`TIS` contains the fields:
`name: 'TrainingPart1_Tis50perc'` (name of training data file)
`WvecSignalUp: [3712x1 double]` (tis discriminant weight vector)
`Mus: [-0.9434 -0.5165]` (estimated means of discriminant scores for negative and positive examples)
`Sigs: [0.1196 0.2410]` (standard deviations of discriminant scores for negative and positive examples)
`Pis: [0.9674 0.0326]` (a priori "mixture" weights of negative and positive score distributions)
`Labels: [-1 1]` (labels indicating the order [negative, positive] of the above parameters)

The weight vector dimensions are organized according to 64 trinucleotides x 58 window positions, e.g.
dimension 1 weights occurence of AAA at window position 1,
dimension 2 weights occurence of AAA at window position 2, ...,
dimension 59 weights occurence of AAC at window position 1, ...
Thereby window position 31 corresponds to the beginning of the (putative) start codon. The above parameters of the gaussian score densities are used to compute posterior probabilities (see manuscript).

`LEN` only contains the parameter `maxLengthBP` which denotes the maximal ORF length occurring in the training set.

`NET` contains the neural network model according to the struct used by NETLAB functions for training and evaluation of the neural net (see `http://www.ncrg.aston.ac.uk/netlab/`).

3 Test Data

The algorithm was tested on artificial fragments that were excised from complete genomes (see manuscript). We provide the test fragments and gene predictions of our method in the archive test_fragments.tgz. The archive contains on a first level subdirectories for different fragment lengths (frag100, ..., frag1200). Within each of these directories, you find subdirectories named by the GenBank accession number of each species that was used for testing our method. The species subdirectories contain three files:

- `frag.seq` holds the fragment sequences,
- `frag.cuts` contains the coordinates on complete genomes from which the corresponding fragment in `frag.seq` was excised.
- The predictions of our method are stored in `orphelia25.coords`.

The 100 bp fragments for *Bacillus subtilis* were split into five stacks of fragments for computational reasons. The directories rep1,...,rep5 contain the same files as all other species subdirectories on first level.

Folder frag700 contains additional folders for 9 fragment sampling repetitions. Those folders also contain species directories with the same files as above.

3.1 The Coords Format

The predictions of our new algorithm are stored as `*.coords`. Each `*.coords` file contains one predicted gene per line in the following format:

```
>FragNo,GeneNo,Coord1_Coord2_Strand_Frame_Completeness
```

- `FragNo` indicates the fragment number corresponding to a line in `frag.seq`.
- `GeneNo` gives the number a gene within the predicted genes on a fragment.
- `Coord1` and `Coord2` indicate the positions of the predicted gene on the fragment, where the first nucleotide of a fragment has the position `1`.
- `Strand` indicates whether the predicted gene is located on the `+` strand or whether it is located on the complementary strand (`-`).
- `Frame` gives the reading frame of an ORF counted from the 5'-end of the entered DNA fragment sequence. Reading frame 1 begins at the first position of the input sequence, frame 2 at the 2^{nd} nucleotide position and frame 3 and the third position.
- `Completeness` is a label which indicates whether a candidate is complete (`C`) or incomplete (`I`).