

## Using *MZEF* to Find Internal Coding Exons

Michael Q. Zhang, Ph.D.

Cold Spring Harbor Laboratory

1 Bungtown Road

Cold Spring Harbor, NY 11724

USA

Tel: (516)-367-8393

Fax: (516)367-8461

Email: [mzhang@cshl.org](mailto:mzhang@cshl.org)

<http://www.cshl.org/mzhanglab/>

Key terms: MZEF, QDA, ORF, exon, intron, Statistical pattern recognition, gene finding

### **Abstract:**

A simple step-wise protocol of how to download, use and interpret *MZEF* program and its result is described by using a real example.

## II. BASIC PROTOCOL

### A. Protocol Introduction

*MZEF* (Michael Zhang's Exon Finder, Zhang 1997) was developed as a direct extension of *Hexon* (Solovyev et al. 1994) – the early version of *Fgene* (UNIT 4.4), as *MZEF* is based on QDA and *Hexon* is based on LDA (see IV. for their definitions and the relation). It was designed to help identifying one of the most important classes of exons, *i.e.* the internal coding exons, in human genomic DNA sequences (Zhang 1998c). It is neither for predicting intronless genes, nor for assembling predicted exons into complete gene models. There is also a mouse version (*mMZEF*) and an *Arabidopsis* version (*aMZEF*) and they can all be found at <http://www.cshl.edu/genefinder/>. Since they all have the same interface, this Protocol will only describe how to use the human version and the users should be able to run other versions similarly.

### USING *MZEF* TO ANALYZE GENOMIC DNA SEQUENCES

There are two ways in which a user can analyze sequence data using *MZEF*:

*Web interface.* *MZEF* may be accessed through the Web at <http://www.cshl.edu/genefinder/>. A user can select “Human”, “Mouse” and “*Arabidopsis*” (“Fission Yeast” would lead to a different algorithm – *POMBE* which is based on LDA, see Chen and Zhang 1998) and get a brief description (README file) by clicking the link at the bottom of the page. Once the selection is made, a request form will be generated through which the prediction can be submitted.

*UNIX command-line.* This is the most powerful way of using *MZEF*. The software can be downloaded by anonymous FTP at <ftp://cshl.edu/pub/science/mzhanglab/mzef/>. A README file and three folders containing *MZEF*, *mMZEF* and *aMZEF* may be found respectively.

### B. Necessary Resources

#### i. Hardware

For command-line execution, any *UNIX* or Linux workstation. For Web access, any internet-connected computer with a Web browser.

#### ii. Software

The executable codes for *MZEF* are free for academic users. To obtain source codes (written in Fortran 77) or for commercial users, one should contact CSHL licensing office (Dr. Carol Dempster, 516-367-6885, [dempster@cshl.org](mailto:dempster@cshl.org)).

To download a *MZEF* executable file, run an FTP session as follows:

```
%ftp cshl.org
```

```

Name: anonymous
Password: [your internet address]
ftp> cd pub/science/mzhanglab/mzef
ftp> get README
ftp> cd human
ftp> binary
ftp> get mzef_new mzef
      (for the interactive version, or
      ftp> get mzef_cmd_lmb_sun mzef_cmd
      for the command-line version)
ftp> mget *.
...[answer "yes" to all the files]
ftp> quit

```

The software has evolved into many different versions to meet the demands from different users. The default platform is SUN (Solaris) unless indicated explicitly at end of an executable file name. "lmb" means the maximum input sequence size is 1 Mb, the default is 200 Kb. "cmd" means all the parameters must be entered from the command-line, the default is interactive (i.e. the program will prompt users for each parameter one line at a time during execution). "static" means it does not require a run-time Fortran library, the default requires libF77.so.x libraries. "new" or any versions after that (1997 or later) will not require files and data being in the current directory to run. Other version may also be compiled at a special request to [mzhang@cshl.org](mailto:mzhang@cshl.org). One may want to move *mzef* into another directory (folder), say:

```
%mv mzef ~/bin
```

### iii. Data Files

During the FTP session described above, all the program data files should have also been downloaded. One may want to move them into a special directory, say:

```
%mkdir ~/MZEF
%mv *.dat ~/MZEF
```

The required data files are:

```

as1.dat
as2.dat
br1.dat
br2.dat
ds1.dat
ds2.dat
h6ex1.dat
h6ex2.dat
h6exc1.dat
h6exc2.dat
h6exi1.dat
h6exi2.dat

```

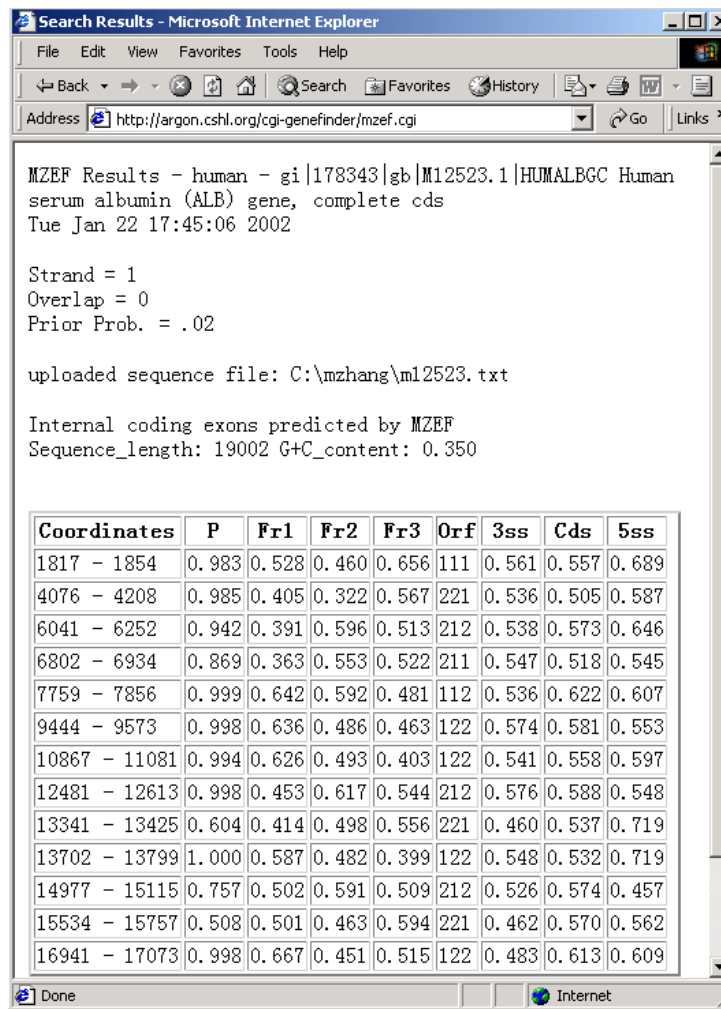
h6exl1.dat  
h6exl2.dat  
h6exr1.dat  
h6exr2.dat  
qda.dat

and test.dat is just a short input DNA sequence for a test run. The format of input sequence file is the standard FASTA format with no more than 80 characters per line. Before executing *MZEF*, one should tell the program where the program data files are stored by defining the environmental variable MZEFDATA to be the correct path:

```
%setenv MZEFDATA `~/MZEF`
```

The instructions on how to install *MZEF* are in the *README* file, which also has a brief description about the program and parameters.

The example used in the following is a 19kb human genomic DNA sequence containing the serum albumin (ALB) gene (Genbank accession number M12523, gi:178343. Minchiotti *et al.*, 1986). The sequence may also be found on the *Current*



**Figure 1.** The screen-dump from an example run, using M12523.seq as the input sequence with all the default parameters.

*Protocols in Bioinformatics* Web site at <http://www.currentprotocols.com/>. This gene has an alternative last exon, the CDS annotation is as follows:

CDS        join(1776..1854,2564..2621,4076..4208,6041..6252,  
6802..6934,7759..7856,9444..9573,10867..11081,  
12481..12613,13702..13799,14977..15115,15534..15757,  
16941..17073,18526..18555)

CDS        join(1776..1854,2564..2621,4076..4208,6041..6252,  
6802..6934,7759..7856,9444..9573,10867..11081,  
12481..12613,13702..13799,14977..15115,15534..15757,  
16941..17073,17688..17732)

that may be compared with various *MZEF* predictions below.

### C. Stepwise Procedure

#### i. Running the web version

Using a web browser and pointing to <http://www.cshl.org/genefinder>, select “human” and cut-and-paste the sequence into the input window. One can also type in the sequence file name or use the “Browse” button to upload the sequence. *MZEF* can only take the standard DNA/RNA character symbols (either in up or lower cases), ambiguous IUPAC symbols will be converted to the standard symbols by a random draw (e.g. “N” will be converted into “A”, “C”, “G”, “T” with equal probability). There are 3 parameters:

- “Strand” = 1 Watson strand (default), 2 Crick strand. Determine which strand should be used;
- “Overlap” = 0 (default), or an integer. Determine maximum number of overlapping exons per splice site allowed in the output;
- “Prior” = 0.02 (default). Determine how likely a randomly picked potential exon (AG + ORF + GT) is a real one.

If one pastes in *m12523.fasta* and clicks “submit”, one will see the result as shown in **Figure 1**. One can also ask the result to be sent back via email by typing in the email address before submitting.

#### ii. Running the interactive version

If one runs the interactive version of *MZEF* locally on a *UNIX/Linux* machine, the result will look similarly as follows:

```
%mzef
ENTER NAME OF THE SEQUENCE FILE (in single quotes)
'm12523.fasta'
ENTER 1 FOR FORWARD, 2 FOR REVERSE
1
```

```

ENTER PRIOR PROBABILITY (suggesting .04)
.04
ENTER OVER LAPPING NUMBER (suggesting 0)
0

```

```

Internal coding exons predicted by MZEF
File_Name: m12523.fas Sequence_length: 19002 G+C_content: 0.350
Coordinates      P      Fr1    Fr2    Fr3    Orf    3ss    Cds    5ss
1817 - 1854 0.992 0.528 0.460 0.656 111 0.561 0.557 0.689
2564 - 2621 0.635 0.636 0.522 0.374 112 0.443 0.557 0.567
4076 - 4208 0.993 0.405 0.322 0.567 221 0.536 0.505 0.587
6041 - 6252 0.971 0.391 0.596 0.513 212 0.538 0.573 0.646
6802 - 6934 0.932 0.363 0.553 0.522 211 0.547 0.518 0.545
7759 - 7856 0.999 0.642 0.592 0.481 112 0.536 0.622 0.607
9444 - 9573 0.999 0.636 0.486 0.463 122 0.574 0.581 0.553
10867 - 11081 0.997 0.626 0.493 0.403 122 0.541 0.558 0.597
12481 - 12613 0.999 0.453 0.617 0.544 212 0.576 0.588 0.548
13341 - 13425 0.757 0.414 0.498 0.556 221 0.460 0.537 0.719
13702 - 13799 1.000 0.587 0.482 0.399 122 0.548 0.532 0.719
14977 - 15115 0.864 0.502 0.591 0.509 212 0.526 0.574 0.457
15534 - 15757 0.678 0.501 0.463 0.594 221 0.462 0.570 0.562
16941 - 17073 0.999 0.667 0.451 0.515 122 0.483 0.613 0.609
17812 - 17874 0.514 0.489 0.434 0.544 221 0.539 0.540 0.493

```

Here the new prior probability value (“Prior” = 0.04) was used instead of the Web default (0.02), and therefore one can see some additional exon predictions in the output.

### iii. *Running the command-line version*

One can also run the command-line version on the local computer:

```

%mfzef_cmd m12523.fasta 1 0.02 1
Internal coding exons predicted by MZEF
File_Name: m12523.fas Sequence_length: 19002 G+C_content: 0.350
Coordinates      P      Fr1    Fr2    Fr3    Orf    3ss    Cds    5ss
1817 - 1854 0.983 0.528 0.460 0.656 111 0.561 0.557 0.689
4076 - 4208 0.985 0.405 0.322 0.567 221 0.536 0.505 0.587
6041 - 6252 0.942 0.391 0.596 0.513 212 0.538 0.573 0.646
6072 - 6252 0.791 0.385 0.608 0.510 212 0.459 0.581 0.646
6802 - 6934 0.869 0.363 0.553 0.522 211 0.547 0.518 0.545
7759 - 7856 0.999 0.642 0.592 0.481 112 0.536 0.622 0.607
9444 - 9573 0.998 0.636 0.486 0.463 122 0.574 0.581 0.553
9449 - 9573 0.808 0.633 0.498 0.455 122 0.468 0.583 0.553
10867 - 11081 0.994 0.626 0.493 0.403 122 0.541 0.558 0.597
10914 - 11081 0.809 0.619 0.468 0.396 122 0.540 0.554 0.597
12481 - 12613 0.998 0.453 0.617 0.544 212 0.576 0.588 0.548
12505 - 12613 0.866 0.467 0.633 0.545 212 0.465 0.602 0.548
13341 - 13425 0.604 0.414 0.498 0.556 221 0.460 0.537 0.719
13357 - 13425 0.575 0.411 0.486 0.564 221 0.473 0.547 0.719
13702 - 13799 1.000 0.587 0.482 0.399 122 0.548 0.532 0.719
13730 - 13799 0.839 0.560 0.477 0.432 122 0.462 0.522 0.719
14977 - 15115 0.757 0.502 0.591 0.509 212 0.526 0.574 0.457
15534 - 15757 0.508 0.501 0.463 0.594 221 0.462 0.570 0.562
16941 - 17073 0.998 0.667 0.451 0.515 122 0.483 0.613 0.609
16969 - 17073 0.555 0.680 0.465 0.520 122 0.415 0.622 0.609

```

Here the user entered “Overlap” = 1, and therefore one can see there are several overlapping exons in the output. If one forgets the parameter order, one can simply type in the command-name by itself and MZEF will output a short usage snippet:

```
%mzef_cmd
Usage: mzef_cmd seqfile strand p0 overlap
sequence file in fasta format (required)
strand: 1 (default)- forward; 2 - reverse
p0: prior probability (default 0.04)
overlap: maximum exon overlap (default 0)
```

### III. DATA INTERPRETATION

The result output contains the following information: File\_Name (maybe truncated if too long), Sequence\_length (in basepair), G+C\_content and a table of internal coding exons predicted. There are nine columns in the table, they are

1. “Coordinates” – the exon coordinates in the input DNA sequence (if “Strand” = 2, one should reverse-complement each output region to get the sense strand segment);
2. “P” – the posterior probability ( $> 0.5$ ) for each exon;
3. “Fr1” – first-frame preference score;
4. “Fr2” – second-frame preference score;
5. “Fr3” – third-frame preference score;
6. “Orf” – open reading frames, *e.g.* “112” (or “110”) means the first and the second frames are open;
7. “3ss” – the acceptor site score;
8. “Cds” – the coding-potential score;
9. “5ss” – the donor site score.

In the Web example, the predicted exon in region (4076..4208) has only one ORF in the third frame which is also consistent with “Fr3” is relatively larger than both “Fr1” and “Fr2”. For the same reason, the predicted exon (7759..7856) has two ORFs (the first and the second because “Orf” = “112”), but the ORF in the first frame is more likely to be the real one because “Fr1” is the larger than “Fr2”..

Although *MZEF* does not assemble the exons into a gene model, by requiring frame compatibility between adjacent coding exons, sometime one could resolve the frame ambiguity or eliminate the false-positive exon predictions. In the Web example above, the predicted exon (6802..6934) had two ORFs (*i.e.* “Orf” = “211”) with “Fr2” (0.553) ~ “Fr3” (0.522), but in order for it to be compatible with the adjacent coding exons, the second ORF would have to be used. For the similar reason, the predicted exon (13341..13425) may be a false-positive, because its ORF is not compatible with others and its “P” score is relative low comparing to the adjacent ones. It must be careful when using frame-compatibility, because it has to assume the adjacent ones are correct and there is no missing (false-negative) one next to it. Sometimes a true exon’s frame is not compatible to the next predicted one because of alternative splicing (*i.e.* it may be compatible with another one further downstream).

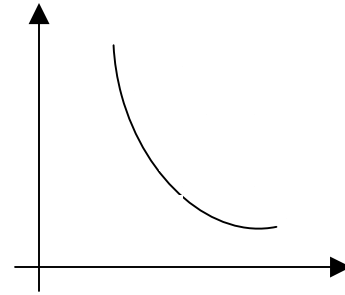
## IV. COMMENTARY

### A. Background Information

#### i. Theory

##### A. Discriminant analysis and Bayes error

*MZEF* is based on a classical discrimination method QDA (Quadratic Discriminant Analysis) which is a direct generation of LDA (Linear Discriminant Analysis, see Fgeneh in UNIT 4.4). Discriminant analysis belongs to general statistical pattern recognition methods and has been used widely in many fields for optimal classification (e.g. Fukunaga 1990). Discriminant analysis is used to answer the following question: given  $N$  objects, how can one assign each object into  $K$  known classes with minimum error? For simplicity, we only consider the case of  $K=2$ , although the theory can be easily generalized to  $K>2$ . In order to distinguish one class object from another, one needs two things: a set of feature variables  $\mathbf{x} = \{x_\alpha : \alpha=1, \dots, p\}$  and a decision rule (*i.e.* classifier)  $C$  such that given the measured values  $\mathbf{x}^i$  for the  $i$ th object,  $C$  would be able to map it into either class I (denoted by “+”) or class II (denoted by “-“, see **Figure 2**). In practice, choosing the set of feature variables that is most discriminative with respect to the two classes is the key to success. For example, the sex hormone level is a much better discriminative feature variable than the weight when classifying people into males and females. Although there are many systematic methods for selecting better feature variables, it is still more or less like a black art, which depends heavily on the master’s insight to the nature of the subject. Once the set of feature variables is decided (or given) and hence one can represent the  $N$  objects to be classified as  $N$  sample points  $\mathbf{x}^i$  in the  $p$ -dimensional feature space. Discriminant theory will offer the mathematical tools for finding the optimal classifier in the sense of minimizing the classification errors.



In general, the (Bayesian) theory assumes the samples points were drawn from two distinct distributions  $p(\mathbf{x}|+) = f_+(\mathbf{x})$  and  $p(\mathbf{x}|-) = f_-(\mathbf{x})$ . If these conditional distributions and the *a priori* probabilities  $\pi_+$  and  $\pi_-$  (for a randomly chosen sample being in class + or -, respectively) are known, then the *a posteriori* probability  $q_+(\mathbf{x})$  of seeing the data  $\mathbf{x}$  and it belonging to class + is given by the Bayes formula

$$q_+(\mathbf{x}) = \pi_+ f_+(\mathbf{x}) / [\pi_+ f_+(\mathbf{x}) + \pi_- f_-(\mathbf{x})],$$

this is because

$$q_+(\mathbf{x}) = p(+|\mathbf{x}) = p(+, \mathbf{x}) / p(\mathbf{x}) = p(\mathbf{x}|+) \pi_+ / p(\mathbf{x}) = p(\mathbf{x}|+) \pi_+ / [p(\mathbf{x}|+) \pi_+ + p(\mathbf{x}|-) \pi_-].$$

A discriminant function  $h(\mathbf{x})$  is defined as the log likelihood ratio

$$h(\mathbf{x}) = \ln [q_+(\mathbf{x}) / q_-(\mathbf{x})].$$

One can choose the decision boundary  $C_B$  (the *Bayes decision rule*) as the hyper-surface  $h(\mathbf{x}) = 0$ , because for any given sample point  $\mathbf{x}^i$ , it would be more likely to belonging to class + if  $h(\mathbf{x}^i) > 0$ . By assigning  $\mathbf{x}^i$  to class +, one would make an error with probability  $q_-(\mathbf{x}^i) < q_+(\mathbf{x}^i)$ . Similarly,

by assigning  $\mathbf{x}^j$  to class  $-$  when  $h(\mathbf{x}^j) < 0$ , one would make an error with probability  $q_+(\mathbf{x}^j) < q_-(\mathbf{x}^j)$ . In general for any decision rule  $C$ , the total error (the *Bayes error*)

$$\varepsilon = \text{probability of misclassification} = \int_{R_+} q_-(\mathbf{x}) d\mathbf{x} + \int_{R_-} q_+(\mathbf{x}) d\mathbf{x},$$

where the regions  $R_+$  and  $R_-$  are classified to  $+$  and  $-$  by  $C$ , respectively.

### B. QDA and its relation to LDA

When samples are assumed to be drawn from two different normal distributions

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right\} = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\left\{-\frac{1}{2} \Delta^2(\mathbf{x}, \boldsymbol{\mu}_k)\right\}$$

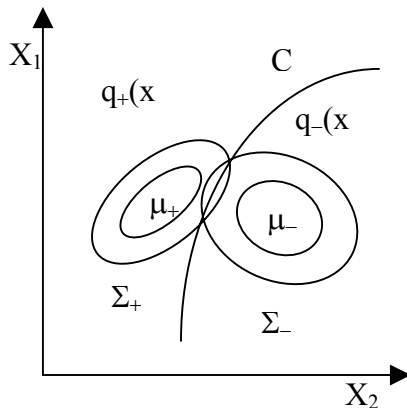
where  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_k$  are the mean and the covariance matrix for the class  $k$  ( $k = +$  or  $-$ ,  $|\boldsymbol{\Sigma}_k|$  is the determinant of the  $p \times p$  matrix and  $\Delta_k(\mathbf{x}, \mathbf{y})$  is called Mahalanobis distance between two vectors  $\mathbf{x}, \mathbf{y}$  within class  $k$ ), the discriminant function will be a quadratic function of  $\mathbf{x}$  (through  $\Delta^2$  defined in the above formula):

$$h(\mathbf{x}) = -\frac{1}{2} \left[ \Delta^2(\mathbf{x}, \boldsymbol{\mu}_+) - \Delta^2(\mathbf{x}, \boldsymbol{\mu}_-) + \ln \frac{|\boldsymbol{\Sigma}_+|}{|\boldsymbol{\Sigma}_-|} \right] + \gamma_{\pm} \quad (1),$$

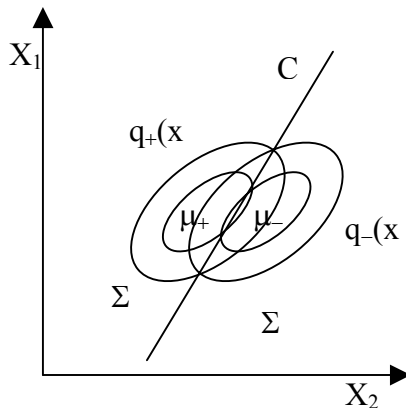
where  $\gamma_{\pm} = \ln(\pi_{\pm}/\pi)$ . Geometrically, the decision boundary is a quadratic hyper-surface in  $p$ -dimensions (**Figure 3**) when  $\boldsymbol{\Sigma}_+ \neq \boldsymbol{\Sigma}_-$ . Using such a quadratic discriminant function for classification is called QDA (quadratic discriminant analysis). When  $\boldsymbol{\Sigma}_+ = \boldsymbol{\Sigma}_- = \boldsymbol{\Sigma}$ , the quadratic terms in  $h(\mathbf{x})$  will be cancelled out:

$$h(\mathbf{x}) = (\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} (\boldsymbol{\mu}_+^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_-) + \gamma_{\pm} \quad (2),$$

the Bayes decision boundary will become linear (hyper-plane as seen in **Figure 4**). Although linear decision boundaries are optimal (in the Bayes sense) only for normal distributions with equal covariance matrices, because of its simplicity, one may always want to know how well one can do with just a linear discriminant function for arbitrary class of distributions. A general linear



**Figure 3** Quadratic decision boundary for normal distributions.



**Figure 4** Linear decision boundary for normal distributions when  $\boldsymbol{\Sigma}_+ = \boldsymbol{\Sigma}_-$ .

discriminant function can be written as  $h(\mathbf{x}) = \mathbf{V}^T \mathbf{x} + v$ , this means that  $\mathbf{x}$  is projected onto a vector  $\mathbf{V}$  and the variable  $y = \mathbf{V}^T \mathbf{x}$  in the projected linear space is classified according whether  $y > v$  or  $y < v$ . Suppose the means and variances in the projected subspace are  $\eta_{\pm} = \mathbf{E}\{h(\mathbf{x})|\pm\} = \mathbf{V}^T \boldsymbol{\mu}_{\pm} + v$  and  $\sigma_{\pm}^2 = \mathbf{Var}\{h(\mathbf{x})|\pm\} = \mathbf{V}^T \boldsymbol{\Sigma}_{\pm} \mathbf{V}$ , the most popular choice for the optimal  $\mathbf{V}$  is

$$\mathbf{V} = \left( \frac{1}{2} \boldsymbol{\Sigma}_+ + \frac{1}{2} \boldsymbol{\Sigma}_- \right)^{-1} (\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) \quad (3)$$

which maximizes the *Fisher criterion*  $(\eta_+ - \eta_-)^2 / (\sigma_+^2 + \sigma_-^2)$  (Fisher 1936). One notices that the Fisher coefficient (3) will reduce to that of (2) when  $\boldsymbol{\Sigma}_+ = \boldsymbol{\Sigma}_-$ , although minimization of the Fisher criterion cannot provide an optimal value for the constant threshold  $\nu$  which may be chosen by minimizing the classification errors in the linear subspace. Using a linear discriminant function (often the Fisher discriminant function) for classification is called LDA (linear discriminant analysis).

In real applications, one normally does not know the distributions. One should always try to transform variables so that they are approximately normal (there are many techniques for doing this, for instance, the Box-Cox transformation 1964). Even if one assumes some parametric distributions, one still has to estimate the parameters using the training data. LDA is more robust, because it does not require normality of the distributions and it has less parameter to be estimated. But if one has sufficient data and the decision boundary is intrinsically nonlinear (two class distributions have very different shapes as indicated by  $\boldsymbol{\Sigma}_+ \neq \boldsymbol{\Sigma}_-$ ), QDA may be superior. Of course, there are also other nonparametric methods that are beyond the scope of this paper. Discriminant analysis can be done equally well by neural networks or machine learning approaches, where the decision boundary or the distribution parameters are estimated by iteration algorithms (Bishop 1996); here we only focus on the multivariate statistical approach for its analytical clarity.

### C. Feature variables used in MZEF

If  $f_A$  is some frequency found in class  $A$ , we define a preference for  $A$  vs.  $B$  (say, exons vs. pseudoexons) to be the ratio  $p_{A,B} = f_A / (f_A + f_B)$ . It is clear that if  $f_A \ll f_B$ , the preference for  $A$  would be close to zero; if  $f_A \sim f_B$ , the preference for  $A$  would be close to  $1/2$  (no preference). There are 9 feature variables used in MZEF and they are computed for high or low (0.48 being the cutoff) G+C query sequences separately. Suppose  $f_{\text{exon}}$  and  $f_{\text{intron}}$  are frequencies for 6mers (or 3ners) in the exon and intron regions pre-computed from the training data, then these 9 feature variables computed on-the-flight are:

- 1) Exon\_length score,  $x_1 = \log_{10}(bp)$ ;
- 2) Intron-exon\_transition score,  $x_2$   
 = average [(Intron\_preference to the left) – (Exon\_preference to the right)]  
 = [(sum of  $p_{\text{Intron,Exon}}$  over all overlapping 6mers in the 54 bp window to the left of 3'ss) – (sum of  $p_{\text{Exon,Intron}}$  over all overlapping 6mers in the 54 bp window to the right of 3'ss)]/49;
- 3) Branch-site score,  $x_3$  = maximum log likelihood branch-site score found in the window (-54,-3) relative to 3'ss using the pre-computed weight matrix;
- 4) 3'ss splice-site score,  $x_4$  = position-dependent triplet preference for true\_acceptor vs pseudo\_acceptor in the window (-24,+3) using pre-computed 3mer weight matrices;
- 5) Exon\_score,  $x_5$   
 = average [ 6mer preference for exon vs. intron ]  
 = (sum of  $p_{\text{Exon,Intron}}$  over all overlapping 6mers in the exon window) / (exon length – 5);
- 6) Strand\_score,  $x_6$   
 = average [ 6mer exon preference for the forward strand vs. the reverse ]  
 = [sum of  $f_{\text{exon}}(w) / (f_{\text{exon}}(w) + f_{\text{exon}}(w'))$  over all overlapping 6mers  $w$  in the exon window ] / (exon length – 5), where the 6mer  $w'$  is the reverse complement of  $w$ ;
- 7) Frame\_score,  $x_7 = \max_{i=0,1,2}$  (frame-specific 6mer preference for exon vs. intron in frame  $i$  in the exon window);

- 8) 5'ss splice-site score,  $x_8$  = position-dependent triplet preference for true\_doner vs pseudo\_doner in the window (-3,+8) using pre-computed 3mer weight matrices;
- 9) Exon-intron\_transition,  $x_9$   
 = average [(Exon\_preference to the left) – (Intron\_preference to the right)]  
 = [(sum of  $p_{Exon,Intron}$  over all overlapping 6mers in the 54 bp window to the left of 5'ss) – (sum of  $p_{Intron,Exon}$  over all overlapping 6mers in the 54 bp window to the right of 5'ss)]/49.

For the *Arabidopsis\_MZEF* (Zhang 1998a), a 60pb flanking intron window is used (instead of 54bp). Since no-isochores are found in *Arabidopsis* genome, no G+C specific feature variables are necessary. But because of the G+C content feature itself had been recognized as the important variable, *Arabidopsis\_MZEF* introduced one additional feature variable – GC\_ratio score,  $x_{10}$  = (G+C content in the exon) / (G+C content in the flanking introns).

## ii. Advantages and limitations

### Advantages

- *MZEF* is simple and fast. It is easily portable and may be incorporated into other programs readily;
- It can find internal coding exons in a short DNA sequence fragment which may not contain the full gene (it only requires 54 bp flanking intron sequence);
- It can also output exons with alternative splice sites by allowing overlaps;
- It can handle very short exons (> 18 bp) and tends to give better accuracy on exon-level statistics.

### Limitations

- Since *MZEF* is only designed to identify one class (albeit the most important class) of exons – internal coding exons, one would need other tools for identifying other eleven classes (Zhang 1998c) of exons (see C. Suggestions for Further Analysis).
- *MZEF* does not produce gene model, one has to assemble a gene model by hand (this may not be regarded as a limitation when one is facing alternative splicing that occurs in nearly 50% human genes).
- User cannot adjust various threshold values other than the few input parameters;
- User has to run reverse strand separately.

## iii. Other Options for Similar Analysis

There are two related programs which extend *MZEF* to improving performance under certain conditions. One is called GSA2 (Huang X.Q., unpublished), which has combined *MZEF* with EST database search results. It may be accessed at the AAT (Analysis and Annotational Tool) Website <http://genome.cs.mtu.edu/aat/aat.html>. If one uses the same sequence and parameters as the example of the interactive *MZEF* run, one will obtain the following result from the AAT server:

Prediction Results

Sequence: >GI|178343|GB|M12523.1|HUMALBGC HUMAN SERUM ALBUMIN (ALB)  
 GENE, COMPLETE CDS

Length: 19002 bp C+G Content: 35%

Type	End5	End3	Leng	Fr	St/Ac	Do/Te	FrCod	Prob	Score
Intr	1817	1854	38	2	0.561	0.689	0.656	0.992	10.02
Intr	2564	2621	58	0	0.443	0.567	0.155	0.635	1.41
Intr	4076	4208	133	2	0.536	0.587	0.567	0.993	10.15
Intr	6041	6252	212	1	0.538	0.646	0.691	0.971	7.38
Intr	6802	6934	133	1	0.547	0.545	0.553	0.932	5.51
Intr	7759	7856	98	0	0.536	0.607	0.965	0.999	14.92
Intr	9444	9573	130	0	0.574	0.553	0.636	0.999	14.18
Intr	10867	11081	215	0	0.541	0.597	0.991	0.997	12.02
Intr	12481	12613	133	1	0.576	0.548	0.617	0.999	13.75
Intr	13702	13799	98	0	0.548	0.719	0.976	1.000	19.63
Intr	14977	15115	139	1	0.526	0.457	0.591	0.864	3.90
Intr	15534	15757	224	2	0.462	0.562	0.724	0.678	1.79
Intr	16941	17073	133	0	0.483	0.609	0.667	0.999	14.48

Reverse Strand

Notations:

Star, initial exon; Intr, internal exon; Term, terminal exon;  
 End5, 5' exon coordinate; End3, 3' exon coordinate;  
 Leng, exon length; Fr, frame number (0, 1, or 2);  
 St/Ac, start or acceptor site score; Do/Te, donor or stop site score;  
 FrCod, in-frame coding score; Prob, exon probability; Score, exon score.

Coding cDNA and protein for each coding region:

.....

It can be seen that the two false-positive internal coding exons, (13341..13425) and (17812..17874) have been eliminated due to the lack of EST matches. There is a danger when a novel exon may also be eliminated.

Another related program is called *MZEF-SPC* (Thanaraj and Robinson 2000), which is an integrated system for exon finding with SpliceProximalCheck as a front-end for *MZEF*. It may be accessed at the EBI Website <http://industry.ebi.ac.uk/~thanaraj/MZEF-SPC.html>. If one uses the same sequence and parameters as the example of the command-line *MZEF* run, one will obtain the following result from the *MZEF-SPC* server:

Results

The predicted exon boundaries by MZEF are further characterised by Splice Proximal Check.

SEQ	EXON	ACCEPTOR SITE	DONOR SITE
gi 178343 gb M11817-1854		Possibly TRUE	Possibly TRUE
gi 178343 gb M14076-4208		Possibly TRUE	Possibly TRUE
gi 178343 gb M16041-6252		Possibly TRUE	Possibly TRUE
gi 178343 gb M16072-6252		FALSE	Possibly TRUE
gi 178343 gb M16802-6934		Possibly TRUE	Possibly TRUE

gi 178343 gb M17759-7856	Possibly TRUE	Possibly TRUE
gi 178343 gb M19444-9573	Possibly TRUE	Possibly TRUE
gi 178343 gb M19449-9573	FALSE	Possibly TRUE
gi 178343 gb M110867-11081	Possibly TRUE	Possibly TRUE
gi 178343 gb M110914-11081	Possibly TRUE	Possibly TRUE
gi 178343 gb M112481-12613	Possibly TRUE	Possibly TRUE
gi 178343 gb M112505-12613	FALSE	Possibly TRUE
gi 178343 gb M113341-13425	Possibly TRUE	Possibly TRUE
gi 178343 gb M113357-13425	Possibly TRUE	Possibly TRUE
gi 178343 gb M113702-13799	Possibly TRUE	Possibly TRUE
gi 178343 gb M113730-13799	FALSE	Possibly TRUE
gi 178343 gb M114977-15115	Possibly TRUE	Possibly TRUE
gi 178343 gb M115534-15757	Possibly TRUE	Possibly TRUE
gi 178343 gb M116941-17073	FALSE	Possibly TRUE
gi 178343 gb M116969-17073	FALSE	Possibly TRUE

Since “Overlap” was set to “1” (the default “Overlap” = “10” in the *MZEF-SPC* server!) among overlapping *MZEF* predicted exons, *MZEF-SPC* was able to pick out most of the exons correctly except the last one. But according to the test, in average, *MZEF-SPC* should pick out more true exons among overlapping ones than *MZEF* non-overlapping predictions. But when selecting true exons among possible ones, frame compatibility should also be considered.

## B. Critical Parameters/Troubleshooting

As mentioned above, *MZEF* requires three input parameters (other than the sequence file itself):

- “Strand” = 1 or 2, the meaning is obvious. One should try both strands if the coding strand information is unknown.
- “P0” or “Prior probability”. It reflects the a priori belief on the coding exon density in the genomic region. As one can see from the above examples, when P0 was changed from 0.02 to 0.04, *MZEF* predicted two more exons that include one true exon (2564..2621) and another false-positive exon (17812..17874). So the effect of increasing P0 is to have more putative exons predicted. The default value is 0.02 for the Web version and is 0.04 for the local version.
- “Overlap” can allow overlapping exons predicted. The default is “0”, namely, no overlapping is allowed. As shown in the command-line version example above, when we set “Overlap” = 1, at most one overlapping exon was allowed to output for each exon region. This would allow the user to choose an exon with alternative splice site, especially when one is looking for an exon that has a compatible frame with other adjacent exons during gene model building.

In addition to three user-controllable parameters, there are also a few hard-coded *MZEF* parameters:

- Minimum ORF size = 18 bp, because shorter exons are extremely rare;
- Maximum ORF size = 999 bp, which was chosen according to the longest internal coding exon in the training set;
- Minimum acceptor site score = 0.38;
- Minimum donor site score = 0.26;
- Minimum total splice site score (acceptor site score + donor site score) = 0.79.

The purpose of setting such thresholds is to reduce a mount of false-positives and to cut down CPU time, perhaps at a reasonable expense of a few false-negatives.

Finally, *MZEF* can only output exons which have a “P” value larger than 0.5.

Most often, the trouble-shooting should start by checking if the input sequence file format is correct (*FASTA* format). One should always check the sequence length in the output report and see if it is correct. If it is not correct, it is most likely caused by extra plank spaces or more than 80 character per line in the sequence file. One should always test the program with a gene of known structure. If the number of predicted exons is too small, try to increase “P0” and vice versa. Normally, if G+C\_content is low, the exon density may also be low.

### C. Suggestions for Further Analysis

One should always run several gene-finding programs, such as Genscan, Fgene, Grail, *etc.* Extensive research has shown that an exon predicted with high score from more than two programs is most likely to be real, even if there is no cDNA support, because the exon may only expressed under special conditions. Homology search against known gene databases is also indispensable.

*MZEF* should also be run in conjunction with other programs that can prediction different type of exons and/or different part of the gene structure. Often the results from these programs can reinforce each other. For example, one could run *CorePromoter* (Zhang 1998b), *CpG\_Promoter* (Ioshikhes and Zhang 2000), *FirstEF* (A first exon finder, Davuluri *et al.* 2001), *JTEF* (A last exon finder, Tabaska *et al* 2001) and *Polyadq* (A polyA site finder, Tabaska and Zhang 1999). All these programs can be accessed from <http://www.cshl.org/mzhanglab/>. Examples of how one can combine some of these programs for gene-finding may be found in Zhang 2000.

### D. Internet Resources

<http://www.cshl.org/genefinder> *MZEF* web server.

<http://www.cshl.org/mzhanglab> Papers and other related information.

<ftp://cshl.org/pub/science/mzhanglab> FTP site.

### E. Literature Cited

- Bishop, C. M. 1996. *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.
- Box, G.E.P. and Cox, D.R. 1964. An analysis of transformations. *J. R. Statist. Soc. B*, 26:211-252.
- Chen, T. and Zhang, M.Q. 1998. *POMBE*: a fission yeast gene-finding and exon-intron structure prediction system. *Yeast* 14:701-710.
- Davuluri R., Grosse I. and Zhang M.Q. 2001. Computational identification of promoters and first exons in the human genome. *Nature Genet.*, 29:412-417.
- Fisher, R.A. 1936. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* 7:179-188.
- Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition* (2<sup>nd</sup>. Ed.) San Diego: Academic Press.

- Ioshikhes I. and Zhang M.Q. 2000. Large-scale human promoter mapping using CpG islands. discrimination. *Nature Genet.*, 26:61-63.
- Minghetti,P.P., Ruffner,D.E., Kuang,W.J., Dennison,O.E., Hawkins,J.W., Beattie,W.G. and Dugaiczky,A. 1986. Molecular structure of the human albumin gene is revealed by nucleotide sequence within q11-22 of chromosome 4. *J. Biol. Chem.* 261:6747-6757.
- Solovyev, V.V., Salamov, A.A. and Lawrence C.B.1994. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucl. Acid. Res.* 22:5156-5163.
- Tabaska, J.E. and Zhang, M.Q.1999. Detection of polyadenylation signals in human DNA sequences. *Gene*, 231:77-86.
- Tabaska J.E., Davuluri R. and Zhang M.Q. (2001). A novel 3'-Terminal exon recognition algorithm. *Bioinformatics* 17:602-607.
- Thanaraj, T.A. and Robinson A.J. 2000. Prediction of exact boundaries of exons. *Briefings in Bioinformatics.* 1(4):343-56.
- Zhang, M.Q. 1997. Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proc. Natl. Acad. Sci. USA.* 94:565-568.
- Zhang, M.Q. 1998a. Identification of protein-coding regions in Arabidopsis thaliana genome based on quadratic discriminant analysis. *Plant Mol. Biol.* 37:803-806.
- Zhang, M.Q. 1998b. Identification of Human Gene Core-promoters In Silico. *Genome Res.*, 8:319-326.
- Zhang, M.Q. 1998c. Statistical Features of Human Exons and Their Flanking Regions. *Hum. Mol. Genet.*7:919-932.
- Zhang, M.Q. 2000. Discriminant analysis and its application in DNA sequence motif recognition. *Briefings in Bioinformatics.* 1(4):331-342.

## F. Key References

- Zhang 1997. See above. This is the original *MZEF* paper.
- Zhang 1998c. See above. This has human exon classification and feature statistics.
- Zhang 2000. See above. This is a tutorial on discriminant analysis and has examples on how to combine *MZEF* with other programs.