Gene finding and gene structure prediction

Lorenzo Cerutti
Swiss Institute of Bioinformatics

EMBnet course, 2004
Outline

- Introduction
- Ab initio methods
  - Principles: signal detection and coding statistics
  - Methods to integrate signal detection and coding statistics
  - Examples of software
- Homology methods
  - Principles
  - An overview of the homology methods
  - Examples of software
- Evaluating performances of gene predictors and limitations
Introduction: gene structure
The Central Dogma of molecular biology

Replication

DNA

Transcription

RNA

Translation

Protein
What is gene finding?

• From a genomic DNA sequence we want to predict the regions that will encode for a protein: the genes.

• Gene finding is about detecting these coding regions and infer the gene structure starting from genomic DNA sequences.

• We need to distinguish coding from non-coding regions using properties specific to each type of DNA region.

• Gene finding is not an easy task!
  • DNA sequence signals have low information content (small alphabet and short sequences);
  • It is difficult to discriminate real signals from noise (degenerated and highly unspecific signals);
  • Gene structure can be complex (sparse exons, alternative splicing, ...);
  • DNA signals may vary in different organisms;
  • Sequencing errors (frame shifts, ...).
Gene structure in prokaryotes

- High gene density and simple gene structure.
- Short genes have little information.
- Overlapping genes.
Gene structure in eukaryotes

- Low gene density and complex gene structure.
- Alternative splicing.
- Pseudo-genes.
Gene finding strategies

- *Ab initio methods:*
  - Based on statistical signals within the DNA:
    - **Signals:** short DNA motifs (promoters, start/stop codons, splice sites, ...)
    - **Coding statistics:** nucleotide compositional bias in coding and non-coding regions
  - **Strengths:**
    - easy to run and fast execution time
    - only require the DNA sequence as input
  - **Weaknesses:**
    - prior knowledge is required (training sets)
    - high number of mispredicted gene structures
Gene finding strategies

- **Homology methods:**
  - Gene structure is deduced using homologous sequences (EST, mRNA, protein).
  - Very accurate results when using homologous sequences with high similarity.
  - **Strengths:**
    - accurate
  - **Weaknesses:**
    - need of good homologous sequences
    - execution is slow
Gene finding: Ab initio methods
Ab initio methods: a simple view

Gene of unknown structure

Find signals and probable coding regions

Coding region probability

ATG {TAA, TGA, TAG} GT AG

AAAAA

Promoter signal PolyA signal
Methods for signal detection

- Detect short DNA motifs (promoters, start/stop codons, splice sites, intron branching point, ...).

- A number of methods are used for signal detection:
  - **Consensus string**: based on most frequently observed residues at a given position.
  - **Pattern recognition**: flexible consensus strings.
  - **Weight matrices**: based on observed frequencies of residues at a given position. Uses standard alignment algorithms.
  - **Weight array matrices**: weight matrices based on dinucleotides frequencies. Takes into account the non-independence of adjacent positions in the sites.
  - **Maximal dependence decomposition (MDD)**: MDD generates a model which captures significant dependencies between non-adjacent as well adjacent positions, starting from an aligned set of signals.
Methods for signal detection

- Methods for signal detection:
  - *Hidden Markov Models* (HMMs):
    - HMMs use a probabilistic framework to infer the probability that a sequence correspond to a real signal.
  - *Neural Networks* (NNs):
    - NNs are trained with positive and negative examples. NNs ”discover” the features that distinguish the two sets.

Example: NN for acceptor sites, the *perceptron*, *(Horton and Kanehisa, 1992):*
Signal detection limitations

- Problems with signal detection:
  - DNA sequence signals have low information content.
  - Signals are highly unspecific and degenerated.
  - Difficult to distinguish between true and false positive.

- How to improve signal detection:
  - Take context into consideration (ex. acceptor site must be flanked by an intron and an exon).
  - Combine with coding statistics (compositional bias).
Types of coding statistics

- Inter-genic regions, introns, and exons have different nucleotides contents.
- This compositional differences can be used to infer gene structure.

Examples of coding statistics:

- ORF length:
  - Assuming an uniform random distribution, stop codons are present every $64/3$ codons ($\approx 21$ codons) in average.
  - In coding regions stop codon average decrease.
  - This measure is sensitive to frame shift errors.
  - Can’t detect short coding regions.

- Bias in nucleotide content in coding regions:
  - Generally coding regions are G+C rich.
  - There are exceptions! For example coding regions of *P. falciparum* are A+T rich.
Types of coding statistics

• Examples of coding statistics:
  
  • Periodicity: The number of residues separating a pair of adenines (A) shows a periodicity in coding regions, but not in non-coding regions. This arises because of the asymmetry in base composition at the third codon position ($3^{rd}$ codon position: 90% are A/T; 10% are G/C).

Coding statistics: codon frequencies

- Codon frequencies:
  Assume \( S = a_1b_1c_1, a_2b_2c_2, \ldots, a_{n+1}b_{n+1}c_{n+1} \) is a coding sequence with unknown reading frame. Let \( f_{abc} \) denote the appearance frequency of codon \( abc \) in a coding sequence. The probabilities \( p_1, p_2, p_3 \) of observing the sequence of \( n \) codons in the 1\(^{st}\), 2\(^{nd}\), and 3\(^{rd}\) frame respectively are:

\[
P_1 = f_{a_1b_1c_1} \times f_{a_2b_2c_2} \times \cdots \times f_{a_nb_nc_n} \tag{1}
\]

\[
P_2 = f_{b_1c_1a_2} \times f_{b_2c_2a_3} \times \cdots \times f_{b_nc_na_{n+1}} \tag{2}
\]

\[
P_3 = f_{c_1a_2b_2} \times f_{c_2a_3b_3} \times \cdots \times f_{c_na_{n+1}b_{n+1}} \tag{3}
\]

The probability \( P_i \) of the \( i \)th reading frame for being the coding region is \((i = 1, 2, 3)\):

\[
P_i = \frac{p_i}{p_1 + p_2 + p_3} \tag{4}
\]
Coding statistics: codon frequencies

- In practice we use these computations in a search algorithm with a *sliding window*:
  - Select a window of size $n$ (for example $n = 30$).
  - Slide the window along the sequence and calculate $P_i$ for each start position of the window.

- A variation of the codon frequency method is to use *6-tuple frequencies* instead of 3-tuple (codon) frequencies. This method was found to be the best single property to predict whether a region of vertebrate genomic sequence was coding or non-coding (*Claverie and Bougueleret, 1986*).

- The usage of hexamers frequencies has been integrated in a number of gene predictors.
Integrating signal and compositional information for gene structure prediction

• A number of methods exists for gene structure prediction which integrate different techniques to detect signals (splicing sites, promoters, etc.) and coding statistics.

• All these methods are classifiers based on machine learning theory.

• Training sets are required to train the algorithms.
Ab initio methods: Generalized HMMs

Genomic DNA

Exon
Intron
Begin
End

Predicted gene structure
Ab initio methods: Generalized HMMs

- Phase 0 intron
  - GT/GC
  - Py tract

- Phase 1 intron
  - GT/GC
  - central

- Phase 2 intron
  - GT/GC
  - central spacer

- 5' UTR
- 3' UTR
- promoter signal
- poly-A signal
- intragenic region
Ab initio methods: GENSCAN

- The underlying (hidden) model of GENSCAN:
## GENSCAN output

- **WEB server:** [http://genes.mit.edu/GENSCAN.html](http://genes.mit.edu/GENSCAN.html)

- **Vertebrate, Arabidopsis, Maize**

<table>
<thead>
<tr>
<th>Gn.Ex</th>
<th>Type</th>
<th>S</th>
<th>Begin</th>
<th>...End</th>
<th>Len</th>
<th>Fr</th>
<th>Ph</th>
<th>I/Ac</th>
<th>Do/T</th>
<th>CodRg</th>
<th>P....</th>
<th>Tscr..</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Prom</td>
<td>+</td>
<td>1653</td>
<td>1692</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1.16</td>
</tr>
<tr>
<td>1.01</td>
<td>Init</td>
<td>+</td>
<td>5215</td>
<td>5266</td>
<td>52</td>
<td>0</td>
<td>1</td>
<td>83</td>
<td>75</td>
<td>151</td>
<td>0.925</td>
<td>12.64</td>
</tr>
<tr>
<td>1.02</td>
<td>Intr</td>
<td>+</td>
<td>5395</td>
<td>5562</td>
<td>168</td>
<td>2</td>
<td>0</td>
<td>89</td>
<td>75</td>
<td>163</td>
<td>0.895</td>
<td>15.02</td>
</tr>
<tr>
<td>1.03</td>
<td>Intr</td>
<td>+</td>
<td>11738</td>
<td>11899</td>
<td>162</td>
<td>0</td>
<td>0</td>
<td>74</td>
<td>113</td>
<td>101</td>
<td>0.990</td>
<td>11.15</td>
</tr>
<tr>
<td>1.04</td>
<td>Intr</td>
<td>+</td>
<td>12188</td>
<td>12424</td>
<td>237</td>
<td>0</td>
<td>0</td>
<td>71</td>
<td>86</td>
<td>197</td>
<td>0.662</td>
<td>15.39</td>
</tr>
<tr>
<td>1.05</td>
<td>Intr</td>
<td>+</td>
<td>14288</td>
<td>14623</td>
<td>336</td>
<td>0</td>
<td>0</td>
<td>82</td>
<td>98</td>
<td>263</td>
<td>0.986</td>
<td>22.19</td>
</tr>
<tr>
<td>1.06</td>
<td>Intr</td>
<td>+</td>
<td>17003</td>
<td>17203</td>
<td>201</td>
<td>0</td>
<td>0</td>
<td>116</td>
<td>86</td>
<td>102</td>
<td>0.976</td>
<td>12.06</td>
</tr>
<tr>
<td>1.07</td>
<td>Intr</td>
<td>+</td>
<td>17741</td>
<td>17859</td>
<td>119</td>
<td>0</td>
<td>2</td>
<td>78</td>
<td>109</td>
<td>51</td>
<td>0.984</td>
<td>6.38</td>
</tr>
<tr>
<td>1.08</td>
<td>Intr</td>
<td>+</td>
<td>18197</td>
<td>18264</td>
<td>68</td>
<td>1</td>
<td>2</td>
<td>103</td>
<td>72</td>
<td>81</td>
<td>0.541</td>
<td>5.70</td>
</tr>
</tbody>
</table>

>02:36:44|GENSCAN_predicted_peptide_1|448_aa
MCRAISLRLLLLQLSQLLAVTQGKTLVLGKEGESAELEEPCESSQQKITVFTKFSDQR KILGQHKGVLIRGGPSQFDRFSKKGAWEKGSFPLIINKLMEDSQTYICELENKEE...
GENSCAN output

Gn.Ex : gene number, exon number (for reference)
Type : Init = Initial exon (ATG to 5’ splice site)
       Intr = Internal exon (3’ splice site to 5’ splice site)
       Term = Terminal exon (3’ splice site to stop codon)
       Sngl = Single-exon gene (ATG to stop)
       Prom = Promoter (TATA box / initiation site)
       PlyA = poly-A signal (consensus: AATAAA)
S : DNA strand (+ = input strand; - = opposite strand)
Begin : beginning of exon or signal (numbered on input strand)
End : end point of exon or signal (numbered on input strand)
Len : length of exon or signal (bp)
Fr : reading frame (a forward strand codon ending at x has frame x mod 3)
Ph : net phase of exon (exon length modulo 3)
I/Ac : initiation signal or 3’ splice site score (tenth bit units)
Do/T : 5’ splice site or termination signal score (tenth bit units)
CodRg : coding region score (tenth bit units)
P : probability of exon (sum over all parses containing exon)
Tscr : exon score (depends on length, I/Ac, Do/T and CodRg scores)
GENSCAN output

GENSCAN predicted genes in sequence 02:36:44

Key:
- Initial exon
- Internal exon
- Terminal exon
- Single-exon gene
- Optimal exon
- Suboptimal exon
Ab initio methods: HMMgene

- Designed to predict complete gene structures
- Uses HMMs with a criterion called *Conditional Maximum Likelihood* which maximize the probability of correct predictions
- Can return sub-optimal prediction to help identifying alternative splicing
- Regions of the sequence can be locked as coding and non-coding by the user
- [http://genome.cbs.dtu.dk/services/HMMgene](http://genome.cbs.dtu.dk/services/HMMgene)
- Human and worm
### HMMgene output

<table>
<thead>
<tr>
<th># SEQ: Sequence 20000 (−) A:5406 C:4748 G:4754 T:5092</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
<tr>
<td><strong>Sequence</strong></td>
</tr>
</tbody>
</table>

...  

| position | prob | strand and frame |

Symbols: firstex = first exon; exon\_n = internal exon; lastex = last exon; singleex = single exon gene; CDS = coding region
Ab initio methods: Linear and quadratic discrimination analysis

- **Linear discrimination analysis** is a standard technique in multivariate analysis.

- Linear discrimination analysis is used to *linearly combine* several measures (e.g. signals and coding statistics) in order to perform the best discrimination between coding and non-coding sequences.

- **Quadratic discriminant analysis.** Similar to linear discrimination analysis, but uses a quadratic discriminant function.

- Dynamic programming is used to combine the inferred exons.
Ab initio methods: Integrate signal detection and coding statistics

Ab initio methods: Linear and quadratic discrimination analysis
Ab initio methods: FGENES

- Combine several measures of pattern recognition using a linear discriminant analysis
  - Donor and acceptor splice sites
  - Putative coding regions
  - 5’ and 3’ intronic regions of the putative exon

- Pass the previous results to a dynamic programming algorithm to find a coherent gene model

- [http://www.softberry.com/berry.phtml](http://www.softberry.com/berry.phtml)

- Can combine homology method with ab initio results

- Human, Drosophila, Worm, Yeasts, Plants
### FGEnES output

Length of sequence: 20000 GC content: 0.48 Zone: 2

Number of predicted genes: 2 In +chain: 2 In -chain: 0
Number of predicted exons: 12 In +chain: 12 In -chain: 0

Predicted genes and exons in var: 2 Max var= 15 GENE WEIGHT: 27.3

<table>
<thead>
<tr>
<th>G</th>
<th>Str</th>
<th>Feature</th>
<th>Start</th>
<th>End</th>
<th>Weight</th>
<th>ORF-start</th>
<th>ORF-end</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>1 CDSf</td>
<td>990</td>
<td>1032</td>
<td>1.84</td>
<td>990 - 1031</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>+</td>
<td>2 CDSi</td>
<td>1576</td>
<td>1835</td>
<td>0.89</td>
<td>1578 - 1832</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>+</td>
<td>PolA</td>
<td>3106</td>
<td></td>
<td>4.64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>1 CDSf</td>
<td>5215</td>
<td>5266</td>
<td>5.25</td>
<td>5215 - 5265</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>2 CDSi</td>
<td>5395</td>
<td>5562</td>
<td>3.08</td>
<td>5397 - 5561</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>3 CDSi</td>
<td>11464</td>
<td>11490</td>
<td>0.76</td>
<td>11466 - 11489</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>4 CDSi</td>
<td>11738</td>
<td>11899</td>
<td>3.28</td>
<td>11740 - 11898</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>5 CDSi</td>
<td>12188</td>
<td>12424</td>
<td>2.48</td>
<td>12190 - 12423</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>6 CDSi</td>
<td>14288</td>
<td>14623</td>
<td>3.26</td>
<td>14290 - 14622</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>7 CDSi</td>
<td>17003</td>
<td>17203</td>
<td>2.79</td>
<td>17005 - 17202</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>8 CDSi</td>
<td>17741</td>
<td>17859</td>
<td>1.62</td>
<td>17741 - 17857</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>9 CDSi</td>
<td>18197</td>
<td>18264</td>
<td>2.53</td>
<td>18196 - 18264</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>10 CDSi</td>
<td>18324</td>
<td>18630</td>
<td>0.87</td>
<td>18325 - 18627</td>
<td></td>
</tr>
</tbody>
</table>

(CDSf = first exon; CDSi = internal exon; CDSl = last exon; CDSo = only one exon; PolA = PolyA signal)
FGENES output

Predicted proteins:
>FGENES-M 1.5 >MySeq 1 Multiexon gene 990 - 1835 100 a Ch+
MSSAFSDPFKEQNPVISLITRTLNNSSSLPVRIVYCQPPNMFLYIAPCAVLVLSTSSTPRR
TENPLRMLNSRFPSFYYLCLRDYQYTPPQLGPLHGRCS
>FGENES-M 1.5 >MySeq 2 Multiexon gene 5215 - 18630 558 a Ch+
MCRAISLRRLLLLLQLSQLLLAVTQKTLVLGKEGESAEIPCESSQKITVFTWKFDQ
Ab initio methods: MZEF

• Designed to predict only internal coding exons

• Uses *quadratic discriminant analysis* of different measures
  
  • Exon length
  • Intron-exon transition/Exon-intron transition
  • Branch-site scores
  • 5’ and 3’ splice sites scores
  • Exon score
  • Strand score

• http://www.cshl.org/genefinder

• Human, Mouse, Arabidopsis, Fission yeast
MZEF output

Internal coding exons predicted by MZEF
Sequence_length: 19920  G+C_content: 0.475

<table>
<thead>
<tr>
<th>Coordinates</th>
<th>P</th>
<th>Fr1</th>
<th>Fr2</th>
<th>Fr3</th>
<th>Orf</th>
<th>3ss</th>
<th>Cds</th>
<th>5ss</th>
</tr>
</thead>
<tbody>
<tr>
<td>5315 - 5482</td>
<td>0.580</td>
<td>0.623</td>
<td>0.528</td>
<td>0.585</td>
<td>122</td>
<td>0.506</td>
<td>0.608</td>
<td>0.552</td>
</tr>
<tr>
<td>6475 - 6582</td>
<td>0.752</td>
<td>0.482</td>
<td>0.563</td>
<td>0.558</td>
<td>221</td>
<td>0.505</td>
<td>0.567</td>
<td>0.598</td>
</tr>
<tr>
<td>11658 - 11819</td>
<td>0.822</td>
<td>0.476</td>
<td>0.569</td>
<td>0.497</td>
<td>211</td>
<td>0.554</td>
<td>0.560</td>
<td>0.651</td>
</tr>
<tr>
<td>14208 - 14543</td>
<td>0.903</td>
<td>0.593</td>
<td>0.619</td>
<td>0.469</td>
<td>212</td>
<td>0.497</td>
<td>0.603</td>
<td>0.575</td>
</tr>
</tbody>
</table>

• Description of the symbols
  • P: Posterior probability (between .5 to 1.)
  • Fr$i$: Frame preference score for the $i$th frame of the genomic sequence
  • Orf: ORF indicator,”011” (or ”211”) means 2nd and 3rd frames are open
  • 3ss: Acceptor score
  • Cds: Coding preference score
  • 5ss: Donor score
Ab initio methods: Decision trees

- *Decision trees* can be automatically build using training algorithms.

- Internal nodes of a decision tree are *property values* tested for each subsequence passed to the tree.

- Properties can be various coding statistics (e.g. hexamers frequencies), signal strength.

- Bottom nodes (*leaves*) of the tree contains *class labels* to be associated with the subsequences.

- Dynamic programming can be used to deduce the complete gene structure.
Ab initio methods: Decision trees

- Example: from *MORGAN* (a decision tree system for finding genes in vertebrate DNA) (*Salzberg et al. 1998*).

```
\begin{align*}
d + a &< 3.4? \\
d + a &< 1.3? \\
d + a &< 5.3? \\
\text{hex} &< 10.3? \\
\text{donor} &< 0.09? \\
\text{hex} &< 0.1? \\
asym &< 4.6? \\
\text{hex} &< 5.6? \\
d: \text{donor site score} \\
a: \text{acceptor site score} \\
\text{hex}: \text{in-frame hexamer frequency} \\
asym: \text{Fickett’s position asymmetry statistic} \\
\text{donor}: \text{donor site score} \\
\text{leaf nodes}: \text{exon, pseudo-exon distribution in the training set}
\end{align*}
```
**Ab initio methods: Neural network**

- A *neural network* is trained with a set of true positives and true negatives examples (set of true exons/false exons, ...).

- For each training example, the neurons are tuned to return the right answer.

- Dynamic programming can be used to deduce the complete gene structure.

(Uberbacher et al., 1996)
Ab initio methods: GRAIL

• Neural network recognizing coding potential

• Incorporates genomic context information (splice junctions, start and stop codons, poly-A signals)

• Not appropriate for sequences without genomic context

• http://compbio.ornl.gov

• Human, Mouse, Drosophila, Arabidopsis, and E. coli
GRAIL

[grail2exons -> Exons]

<table>
<thead>
<tr>
<th>St Fr</th>
<th>Start</th>
<th>End</th>
<th>ORFstart</th>
<th>ORFend</th>
<th>Score</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1- f 1</td>
<td>479</td>
<td>666</td>
<td>452</td>
<td>670</td>
<td>52.000</td>
<td>good</td>
</tr>
<tr>
<td>2- f 0</td>
<td>5176</td>
<td>5290</td>
<td>5176</td>
<td>5370</td>
<td>82.000</td>
<td>excellent</td>
</tr>
<tr>
<td>3- f 2</td>
<td>5395</td>
<td>5562</td>
<td>5364</td>
<td>5618</td>
<td>99.000</td>
<td>excellent</td>
</tr>
<tr>
<td>4- f 0</td>
<td>7063</td>
<td>7113</td>
<td>7063</td>
<td>7113</td>
<td>53.000</td>
<td>good</td>
</tr>
<tr>
<td>5- f 0</td>
<td>11827</td>
<td>11899</td>
<td>11590</td>
<td>11925</td>
<td>74.000</td>
<td>good</td>
</tr>
<tr>
<td>6- f 0</td>
<td>12188</td>
<td>12424</td>
<td>12163</td>
<td>12633</td>
<td>88.000</td>
<td>excellent</td>
</tr>
<tr>
<td>7- f 0</td>
<td>14288</td>
<td>14623</td>
<td>14194</td>
<td>14640</td>
<td>94.000</td>
<td>excellent</td>
</tr>
<tr>
<td>8- f 0</td>
<td>17003</td>
<td>17203</td>
<td>16957</td>
<td>17235</td>
<td>100.000</td>
<td>excellent</td>
</tr>
<tr>
<td>9- f 0</td>
<td>17751</td>
<td>17859</td>
<td>17659</td>
<td>17988</td>
<td>50.000</td>
<td>good</td>
</tr>
<tr>
<td>10- f 1</td>
<td>18212</td>
<td>18264</td>
<td>18071</td>
<td>18268</td>
<td>61.000</td>
<td>good</td>
</tr>
</tbody>
</table>

[grail2exons -> Exon Translations]

11- MLRGTDasnnNevFKkAKIMFLEVRKSLTcgQgPttGSSCNGAGQRESGHA
AFGIKHTQSVDR

12- AQIPNQQELKETTCRAISLRRLLLLLLQLCKFSDLGT

13- AQLLAVTQGKLVLGKEGESAEPEESSQKKITVFTWKFSDQKRILGQHG
KGVLIR
Gene finding: Homology methods
Homology methods: a simple view

Genomic DNA sequence → mRNA, EST, protein homologous

Pairwise comparison →

Find DNA signals →

Infer gene structure
Homology methods: Procrustes

- *Procrustes* predicts gene structure using protein homology (*Gelfand et al., 1996*).

Find all possible blocks (exons) in the query sequence (based on the acceptor/donor sites)

Find optimal alignments between blocks and homologous sequences

Find best alignment between concatenations of the blocks and the homologous sequences
Homology methods: Genewise

- Uses HMMs to compare DNA sequences to protein sequences at the level of its conceptual translation, regardless of sequencing errors and introns.

- Principle:
  - The exon model used in genewise is a HMM with 3 base states (match, insert, delete) with the addition of more transitions between states to consider frame-shifts.
  - Intron states have been added to the base model.
  - Genewise directly compare HMM-profiles of proteins or domains to the gene structure HMM model.

- Genewise is a powerful tool, but time consuming.

- Requires strong similarities (>70% identity) to produce good predictions.

- Genewise is part of the Wise2 package: http://www.ebi.ac.uk/Wise2/.
Homology methods: Genewise

Homology methods EMBnet 2004
Genewise output: perfect match

...
Genewise output: frame shift

...  

seq1  249  TDRRIGCLLS  GLDSSLVAATLLK  
     TDRRIGCL S  GLDSSLVAATLLK  
     TDRRIGCL!S  G:G[ggg]  GLDSSLVAATLLK  

seq1  12930  agaaagtc2tGGTGAAGT  Intron 4  TAGGGgtgtatgggacta  
     caggttgt  c <1-----[12960:13407]-1>  gtacgtccctta  
     acagtccct a  cgcccgttctggg  

...  

Gene 1  
Gene 2979 12953  
Exon 2979 3227 phase 0  
Exon 7315 7552 phase 0  
Exon 12416 12601 phase 1  
Exon 12859 12953 phase 1  

Gene 2  
Gene 12956 19553  
Exon 12956 12959 phase 0  
Exon 13408 13535 phase 1  
Exon 14998 15124 phase 0  
Exon 16355 16461 phase 1  
Exon 18600 18755 phase 0  
Exon 19347 19553 phase 0
Genewise output: mismatches

... seq1 249 TDRR--CLLS TDRR  CLLS TDRRIGCLLS G:G[ggg] GLDSSLVAATLLK
seq1 12930 agaaagtcttGGTGAAGT Intron 4 TAGGGgtgtatgggacta caggtggttc <1------[12961:13408]-1> gtacgttcctta acagtcctaa cgcccgttctggg
...

Gene 1
Gene 2979 19554
Exon 2979 3227 phase 0
Exon 7315 7552 phase 0
Exon 12416 12601 phase 1
Exon 12859 12960 phase 1
Exon 13409 13536 phase 1
Exon 14999 15125 phase 0
Exon 16356 16462 phase 1
Exon 18601 18756 phase 0
Exon 19348 19554 phase 0
Homology methods: sim4

- Align cDNA to genomic sequences.

- sim4 performs standard dynamic programming:
  - models splice sites
  - introns are treated as special kind of gaps with low penalties

- sim4 performs very well, but needs strong similarity between the sequences.
Homology methods

sim4 output

1-249 (1-249) 100% -> (GT/AG)
4337-4574 (250-487) 100% -> (GT/AG)
9438-9623 (488-673) 100% -> (GT/AG)
9881-9982 (674-775) 100% -> (GT/AG)
10431-10558 (776-903) 100% -> (GT/AG)
12021-12135 (904-1018) 100% -> (GT/AG)
13425-13484 (1019-1077) 98% -> (GT/AG)
15623-15778 (1078-1233) 100% -> (GT/AG)
16370-16576 (1234-1440) 100%
Homology methods: BLAST

- BLAST can be used to find genomic sequences similar to proteins, ESTs, cDNAs.

- A BLAST hit doesn't mean necessarily an exon. Some post-processing is required.

- BLAST can indicate the rough position of exons, but nothing about the gene structure.

  ![Diagram showing "ideal" and "real" BLAST]

- However, BLAST is fast! and can reduce the search space for others programs.
Homology methods: Trimming with BLAST

Protein sequence

BLAST vs genomic

Get best BLAST HSPs (trimming)

GeneWise

cDNA sequence

sim4
Evaluation of performances
Evaluating performances

• Measures (*Burset and Guigo, 1996; Snyder and Stormo, 1997*):
  
  • **Sensitivity** $S_n$ is the proportion of coding nucleotides that are correctly predicted as coding:

  $$S_n = \frac{TP}{TP + FN}$$

  • **Specificity** $S_p$ is the proportion of nucleotides predicted as coding that are actually coding:

  $$S_p = \frac{TP}{TP + FP}$$
Evaluating performances

• Measures (contd.):

  • **Correlation coefficient** $CC$ is a single measure that captures both specificity and sensitivity:

    $$ CC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP+FN) \times (TN+FP) \times (TP+FP) \times (TN+FN)}} $$

  • **Approximate correlation** $AC$ is similar to $CC$, but defined under any circumstances:

    $$ AC = (ACP - 0.5) \times 2 $$

    where

    $$ ACP = \frac{1}{4} \left( \frac{TP}{TP+FN} + \frac{TP}{TP+FP} + \frac{TN}{TN+FP} + \frac{TN}{TN+FN} \right) $$
# Accuracy of the different methods

- Evaluation of the different programs *(Rogic et al., 2001)*

<table>
<thead>
<tr>
<th>Programs</th>
<th>No. of sequences</th>
<th>$S_n$</th>
<th>$S_p$</th>
<th>AC</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGENES</td>
<td>195</td>
<td>0.86</td>
<td>0.88</td>
<td>0.84 ± 0.19</td>
<td>0.83</td>
</tr>
<tr>
<td>GeneMark.hmm</td>
<td>195</td>
<td>0.87</td>
<td>0.89</td>
<td>0.84 ± 0.18</td>
<td>0.83</td>
</tr>
<tr>
<td>GENSCAN</td>
<td>195</td>
<td>0.95</td>
<td>0.90</td>
<td>0.91 ± 0.12</td>
<td>0.91</td>
</tr>
<tr>
<td>HMMgene</td>
<td>195</td>
<td>0.93</td>
<td>0.93</td>
<td>0.91 ± 0.13</td>
<td>0.91</td>
</tr>
<tr>
<td>MZEF</td>
<td>119</td>
<td>0.70</td>
<td>0.73</td>
<td>0.68 ± 0.21</td>
<td>0.66</td>
</tr>
</tbody>
</table>
Accuracy of the different methods

- Overall performances are the best for HMMgene and GENSCAN.

- Some program’s accuracy depends on the G+C content, except for HMMgene and GENSCAN, which use different parameters sets for different G+C contents.

- For almost all the tested programs, ”medium” exons (70-200 nucleotides long), are most accurately predicted. Accuracy decrease for shorter and longer exons, except for HMMgene.

- Internal exons are much more likely to be correctly predicted (weakness of the start/stop codon detection).

- Initial and terminal exons are most likely to be missed completely.

- Only HMMgene and GENSCAN have reliable scores for exon prediction.
Accurac of the different methods

- Recently a new benchmark has been published by Makarov (2002), with similar results, but other predictors have been included.

<table>
<thead>
<tr>
<th>Programs</th>
<th>$S^a_n$</th>
<th>$S^a_p$</th>
<th>$S^b_n$</th>
<th>$S^b_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMMgene</td>
<td>97</td>
<td>91</td>
<td>93</td>
<td>93</td>
</tr>
<tr>
<td>GenScan</td>
<td></td>
<td>95</td>
<td>90</td>
<td>93</td>
</tr>
<tr>
<td>Geneid</td>
<td>86</td>
<td>83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Genie</td>
<td>96</td>
<td>92</td>
<td>91</td>
<td>90</td>
</tr>
<tr>
<td>FGENES</td>
<td>89</td>
<td>77</td>
<td>86</td>
<td>88</td>
</tr>
</tbody>
</table>

- $^a$ Adh region of Drosophila.

- $^b$ 195 high-quality mammalian sequences (human, mouse, and rat).
Gene prediction limits

• Existing predictors are for protein coding regions
  • Non-coding areas are not detected (5' and 3' UTR)
  • Non-coding RNA genes are missed

• Predictions are for "typical" genes
  • Partial genes are often missed
  • Training sets may be biased
  • Atypical genes use other grammars
The end