Gene Prediction: Similarity-Based Approaches
Outline

• The idea of similarity-based approach to gene prediction
• Exon Chaining Problem
• Spliced Alignment Problem
• Gene prediction tools
Using Known Genes to Predict New Genes

• Some genomes may be very well-studied, with many genes having been experimentally verified.
• Closely-related organisms may have similar genes
• Unknown genes in one species may be compared to genes in some closely-related species
Similarity-Based Approach to Gene Prediction

- Genes in different organisms are similar
- The similarity-based approach uses known genes in one genome to predict (unknown) genes in another genome
- **Problem:** Given a known gene and an unannotated genome sequence, find a set of substrings of the genomic sequence whose concatenation best fits the gene
Comparing Genes in Two Genomes

- Small islands of similarity corresponding to similarities between exons
Reverse Translation

• Given a known protein, find a gene in the genome which codes for it
• One might infer the coding DNA of the given protein by reversing the translation process
  • Inexact: amino acids map to > 1 codon
  • This problem is essentially reduced to an alignment problem
Reverse Translation (cont’d)

- This reverse translation problem can be modeled as traveling in Manhattan grid with free horizontal jumps
  - Complexity of Manhattan is $n^3$
- Every horizontal jump models an insertion of an intron
- Problem with this approach: would match nucleotides pointwise and use horizontal jumps at every opportunity
Comparing Genomic DNA Against mRNA

Portion of genome

mRNA (codon sequence)

exon1

intron1

exon2

intron2

exon3
Using Similarities to Find the Exon Structure

- The known frog gene is aligned to different locations in the human genome
- Find the “best” path to reveal the exon structure of human gene
Finding Local Alignments

Use local alignments to find all islands of similarity

Frog Genes (known)  Human Genome
Chaining Local Alignments

• Find substrings that match a given gene sequence (candidate exons)
• Define a candidate exons as $(l, r, w)$ ($l$, $r$, $w$ defined as score of local alignment)
• Look for a maximum chain of substrings
  • Chain: a set of non-overlapping nonadjacent intervals.
Exon Chaining Problem

- Locate the beginning and end of each interval (2n points)
- Find the “best” path
Exon Chaining Problem: Formulation

- **Exon Chaining Problem**: Given a set of putative exons, find a maximum set of non-overlapping putative exons

- **Input**: a set of weighted intervals (putative exons)

- **Output**: A maximum chain of intervals from this set
Exon Chaining Problem: Formulation

- **Exon Chaining Problem**: Given a set of putative exons, find a maximum set of non-overlapping putative exons

- **Input**: a set of weighted intervals (putative exons)

- **Output**: A maximum chain of intervals from this set

Would a greedy algorithm solve this problem?
This problem can be solved with dynamic programming in $O(n)$ time.
Exon Chaining Algorithm

`ExonChaining(G, n) //Graph, number of intervals
2     for i ← to 2n
3         s_j ← 0
4     for i ← 1 to 2n
5         if vertex v_i in G corresponds to right end of the interval I
6         j ← index of vertex for left end of the interval I
7         w ← weight of the interval I
8         s_j ← max{ s_j + w, s_{i-1} }
9     else
10         s_i ← s_{i-1}
11     return s_{2n}`
Exon Chaining: Deficiencies

- Poor definition of the putative exon endpoints
- Optimal chain of intervals may not correspond to any valid alignment
  - First interval may correspond to a suffix, whereas second interval may correspond to a prefix
  - Combination of such intervals is not a valid alignment
Infeasible Chains

Red local similarities form two non-overlapping intervals but do not form a valid global alignment.
Gene Prediction Analogy: Selecting Putative Exons

The cell carries DNA as a blueprint for producing proteins, like a manufacturer carries a blueprint for producing a car.
Using Blueprint
Assembling Putative Exons
Still Assembling Putative Exons
Spliced Alignment

- Mikhail Gelfand and colleagues proposed a **spliced alignment** approach of using a protein within one genome to reconstruct the exon-intron structure of a (related) gene in another genome.
- Begins by selecting either all putative exons between potential acceptor and donor sites or by finding all substrings similar to the target protein (as in the Exon Chaining Problem).
- This set is further filtered in a way that attempt to retain all true exons, with some false ones.
Spliced Alignment Problem: Formulation

- **Goal**: Find a chain of blocks in a genomic sequence that best fits a target sequence

- **Input**: Genomic sequences $G$, target sequence $T$, and a set of candidate exons $B$.

- **Output**: A chain of exons $\Gamma$ such that the global alignment score between $\Gamma^*$ and $T$ is maximum among all chains of blocks from $B$.

$\Gamma^*$ - concatenation of all exons from chain $\Gamma$
Lewis Carroll Example
Spliced Alignment: Idea

- Compute the best alignment between $i$-prefix of genomic sequence $G$ and $j$-prefix of target $T$: $S(i,j)$

- But what is “$i$-prefix” of $G$?

- There may be a few $i$-prefixes of $G$ depending on which block $B$ we are in.
Spliced Alignment: Idea

- Compute the best alignment between $i$-prefix of genomic sequence $G$ and $j$-prefix of target $T$: $S(i,j)$

- But what is “$i$-prefix” of $G$?
- There may be a few $i$-prefixes of $G$ depending on which block $B$ we are in.
- Compute the best alignment between $i$-prefix of genomic sequence $G$ and $j$-prefix of target $T$ under the assumption that the alignment uses the block $B$ at position $i$: $S(i,j,B)$
Spliced Alignment Recurrence

If $i$ is not the starting vertex of block $B$:

$S(i, j, B) = \max \{ S(i - 1, j, B) - \text{indel penalty} \}
S(i, j - 1, B) - \text{indel penalty} \}
S(i - 1, j - 1, B) + \delta(g_i, t_j) \}$

If $i$ is the starting vertex of block $B$:

$S(i, j, B) = \max \{ S(i, j - 1, B) - \text{indel penalty} \}
\max \text{all blocks } B' \text{ preceding block } B \ S(\text{end}(B'), j, B') - \text{indel penalty} \}
\max \text{all blocks } B' \text{ preceding block } B \ S(\text{end}(B'), j - 1, B') + \delta(g_i, t_j) \}$
Spliced Alignment Solution

- After computing the three-dimensional table $S(i, j, B)$, the score of the optimal spliced alignment is:

$$\max_{\text{all blocks } B} S(\text{end}(B), \text{length}(T), B)$$
Spliced Alignment: Complications

- Considering multiple $i$-prefixes leads to slow down.
  
  \[ O(mn^2 |B|) \]
  
  where $m$ is the target length, $n$ is the genomic sequence length and $|B|$ is the number of blocks.

- A **mosaic effect**: short exons are easily combined to fit any target protein
Spliced Alignment: Speedup
Spliced Alignment: Speedup
Spliced Alignment: Speedup

\[ P(i,j) = \max_{\text{all blocks } B \text{ preceding position } i} S(\text{end}(B), j, B) \]
Exon Chaining vs Spliced Alignment

• In Spliced Alignment, every path spells out string obtained by concatenation of labels of its edges. The weight of the path is defined as optimal alignment score between concatenated labels (blocks) and target sequence.

• Defines weight of entire path in graph, but not the weights for individual edges.

• Exon Chaining assumes the positions and weights of exons are pre-defined.
Gene Prediction: Aligning Genome vs. Genome

- Align entire human and mouse genomes
- Predict genes in both sequences simultaneously as chains of aligned blocks (exons)
- This approach does not assume any annotation of either human or mouse genes.
Gene Prediction Tools

- GENSCAN/Genome Scan
- TwinScan
- Glimmer
- GenMark
The GENSCAN Algorithm

- Algorithm is based on probabilistic model of gene structure similar to *Hidden Markov Models* (HMMs).
- GENSCAN uses a training set in order to estimate the HMM parameters, then the algorithm returns the exon structure using maximum likelihood approach standard to many HMM algorithms (*Viterbi* algorithm).
- Biological input: Codon bias in coding regions, gene structure (start and stop codons, typical exon and intron length, presence of promoters, presence of genes on both strands, etc)
- Covers cases where input sequence contains no gene, partial gene, complete gene, multiple genes.
GENSCAN Limitations

• Does not use similarity search to predict genes.
• Does not address alternative splicing.
• Could combine two exons from consecutive genes together.
GenomeScan

- Incorporates similarity information into GENSCAN: predicts gene structure which corresponds to maximum probability conditional on similarity information
- Algorithm is a combination of two sources of information
  - Probabilistic models of exons-introns
  - Sequence similarity information
TwinScan

- Run Viterbi algorithm using emissions $e_k(b)$ where $b \in \{A-, A: , A|, ..., T| \}$.
TwinScan (cont’d)

- The emission probabilities are estimated from human/mouse gene pairs.
  - Ex. $e_I(x|) < e_E(x|)$ since matches are favored in exons, and $e_I(x-) > e_E(x-)$ since gaps (as well as mismatches) are favored in introns.
  - Compensates for dominant occurrence of poly-A region in introns
Glimmer

- **Gene Locator and Interpolated Markov ModelER**
- Finds genes in bacterial DNA
- Uses interpolated Markov Models
The Glimmer Algorithm

• Made of 2 programs
  • **BuildIMM**
    • Takes sequences as input and outputs the Interpolated Markov Models (IMMs)
  • **Glimmer**
    • Takes IMMs and outputs all candidate genes
    • Automatically resolves overlapping genes by choosing one, hence limited
    • Marks “suspected to truly overlap” genes for closer inspection by user
GenMark

- Based on non-stationary Markov chain models
- Results displayed graphically with coding vs. noncoding probability dependent on position in nucleotide sequence