Dynamic Programming 2: Gene Prediction

Bioinformatics Programming - 2016

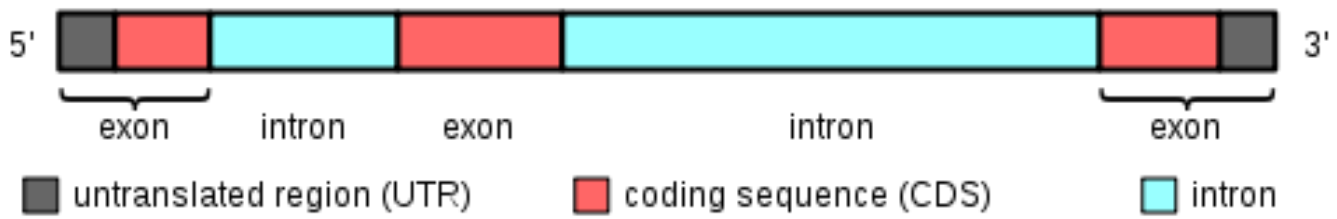
Computer Engineering, Chiang Mai University

Gene Prediction

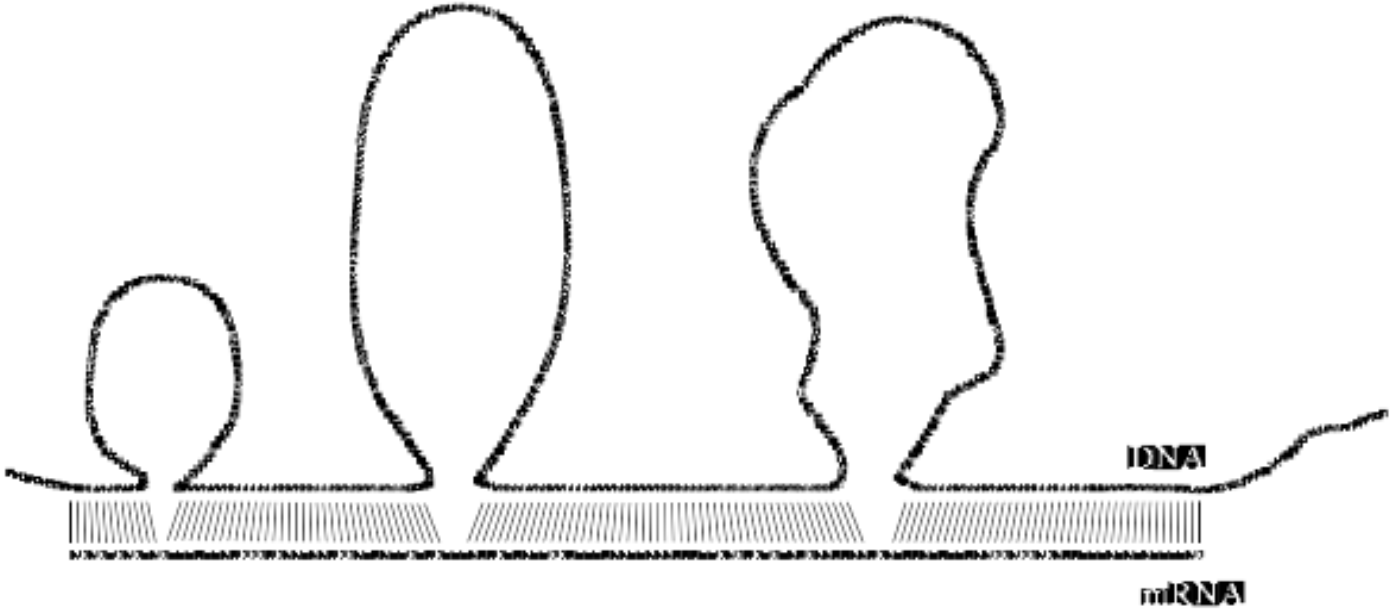
- The first steps in understanding the genome of a species once it has been sequenced – aka. Gene finding
- The process of identifying the regions of genomic DNA that encode genes [wikipedia]
  - mRNA genes
  - Protein coding genes
  - Regulatory regions
- A high degree of similarity to a known mRNA or protein product is strong evidence that a region of a target genome is a protein-coding gene

Gene Prediction

- **Sydney Brenner** and **Francis Crick** showed that every triplet of nucleotides (codon) in gene codes for one amino acid
  - Deleting three consecutive nucleotides results in minor change in the protein
- Biologists believed that a protein was encoded by a long string of contiguous triplets
  - Many organisms contain large amount of "junk DNA" that does not code for proteins at all – introns
  - These introns break organism genome into pieces of coding gene - exons



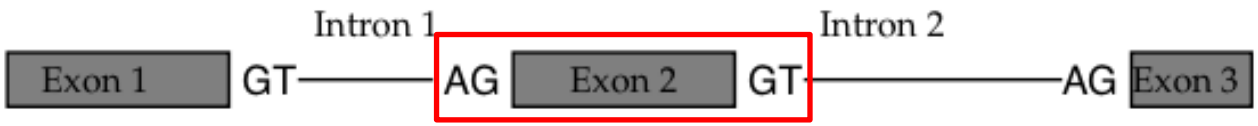
Gene Prediction



- The jump between different parts of split genes are inconsistent from species to species
- Number of exons may be different
  - While the genes are related (between species)
  - An exon in human genome may be broken into two in the mouse genome, or vice versa

Gene Prediction

- Human genes, exons, consist of only 3% of human genome
- Prokaryotic organisms do not have broken genes
  - Gene prediction algorithms tend to be simpler than those for eukaryotes
- Two approaches
  - Statistical approach – looks for features that appear frequently in gene: splicing signals (exon-intron junction)



  - Similarity-based approach – a newly sequenced gene has a good chance of being related to one that is already know

Gene Prediction

- We cannot simply look for similar sequence in one organism's genome based on the genes known in another:
  - Exon sequence and exon structure of the related gene in different species are different
- The commonality between related genes in both organisms is that they produce similar proteins
  - Suppose we know a human protein,
  - We want to discover the exon structure of the related gene in the genome that produce similar human protein

Statistical Approaches

- The simplest way to detect potential coding region is to look at open reading frames (ORFs)
  - The subsegments start with start codon and end with stop codon
- Start codon
  - The first codon of a mRNA that signals the a start of translation
  - Almost always codes for methionine (Met) – AUG (or ATG in DNA)
- Stop codon
  - Termination codon
  - A triplet within mRNA that signals a termination of translation
  - In RNA – UAG, UAA, UGA (TAG, TAA, TGA in DNA)

Statistical Approaches

- Example: three reading frames

```
1.  ATG CAA TGG GGA AAT GTT ACC AGG TCC GAA CTT ATT GAG GTA AGA CAG ATT TAA
2.  A TGC AAT GGG GAA ATG TTA CCA GGT CCG AAC TTA TTG AGG TAA GAC AGA TTT AA
3.  AT GCA ATG GGG AAA TGT TAC CAG GTC CGA ACT TAT TGA GGT AAG ACA GAT TTA A
```

- DNA has two anti-parallel strands, an additional three reading frames arise, giving possible six frame translations

```
Possible Amino Acid   R  S  R  A  F  W  S  P  M  S  A  A  D  S  S  *  K  A
Sequences (Forward)    D  L  G  R  S  G  R  R  C  R  R  P  T  H  L  E  R
                        I  S  G  V  L  V  A  D  V  G  G  R  L  I  L  K  G
Nucleotide Sequence   CGATCTCGGGCGTTCTGGTCGCCGATGTCGGCGGCCGACTCATCTTGAAAGG
                      ....:....|....:....|....:....|....:....|....:....|..
                      GCTAGAGCCCGCAAGACCAGCGGCTACAGCCGCCGGCTGAGTAGAACTTTCC
Possible Amino Acid    R  D  R  A  N  Q  D  G  I  D  A  A  S  E  D  Q  F
Sequences (Reverse)     S  R  P  R  E  P  R  R  H  R  R  G  V  *  R  S  L
                       A  I  E  P  T  R  T  A  S  T  P  P  R  S  M  K  F  P
```

Statistical Approaches

- Long ORFs are often used to initially identify candidates in DNA sequence
  - Longer than some threshold length
  - May fail to detect short genes or genes with short exons
  - The presence of an ORF does not mean that the region is ever translated
- Many statistical algorithm rely on statistical features in protein-coding regions
  - Frequency of occurrence – 64 codon usage array

Statistical Approaches

- The codon usage array in ***E.COLI*** genes

	Codon	Amino acid[2]	%[3]	Ratio[4]	Codon	Amino acid	%	Ratio	Codon	Amino acid	%	Ratio	Codon	Amino acid	%	Ratio	
U	UUU	Phe (F)	1.9	0.51	UCU	Ser (S)	1.1	0.19	UAU	Tyr (Y)	1.6	0.53	UGU	Cys (C)	0.4	0.43	U
	UUC	Phe (F)	1.8	0.49	UCC	Ser (S)	1.0	0.17	UAC	Tyr (Y)	1.4	0.47	UGC	Cys (C)	0.6	0.57	C
	UUA	Leu (L)	1.0	0.11	UCA	Ser (S)	0.7	0.12	UAA	STOP	0.2	0.62	UGA	STOP	0.1	0.30	A
	UUG	Leu (L)	1.1	0.11	UCG	Ser (S)	0.8	0.13	UAG	STOP	0.03	0.09	UGG	Trp (W)	1.4	1.00	G
C	CUU	Leu (L)	1.0	0.10	CCU	Pro (P)	0.7	0.16	CAU	His (H)	1.2	0.52	CGU	Arg (R)	2.4	0.42	U
	CUC	Leu (L)	0.9	0.10	CCC	Pro (P)	0.4	0.10	CAC	His (H)	1.1	0.48	CGC	Arg (R)	2.2	0.37	C
	CUA	Leu (L)	0.3	0.03	CCA	Pro (P)	0.8	0.20	CAA	Gln (Q)	1.3	0.31	CGA	Arg (R)	0.3	0.05	A
	CUG	Leu (L)	5.2	0.55	CCG	Pro (P)	2.4	0.55	CAG	Gln (Q)	2.9	0.69	CGG	Arg (R)	0.5	0.08	G
A	AUU	Ile (I)	2.7	0.47	ACU	Thr (T)	1.2	0.21	AAU	Asn (N)	1.6	0.39	AGU	Ser (S)	0.7	0.13	U
	AUC	Ile (I)	2.7	0.46	ACC	Thr (T)	2.4	0.43	AAC	Asn (N)	2.6	0.61	AGC	Ser (S)	1.5	0.27	C
	AUA	Ile (I)	0.4	0.07	ACA	Thr (T)	0.1	0.30	AAA	Lys (K)	3.8	0.76	AGA	Arg (R)	0.2	0.04	A
	AUG	Met (M)	2.6	1.00	ACG	Thr (T)	1.3	0.23	AAG	Lys (K)	1.2	0.24	AGG	Arg (R)	0.2	0.03	G
G	GUU	Val (V)	2.0	0.29	GCU	Ala (A)	1.8	0.19	GAU	Asp (D)	3.3	0.59	GGU	Gly (G)	2.8	0.38	U
	GUC	Val (V)	1.4	0.20	GCC	Ala (A)	2.3	0.25	GAC	Asp (D)	2.3	0.41	GGC	Gly (G)	3.0	0.40	C
	GUA	Val (V)	1.2	0.17	GCA	Ala (A)	2.1	0.22	GAA	Glu (E)	4.4	0.70	GGA	Gly (G)	0.7	0.09	A
	GUG	Val (V)	2.4	0.34	GCG	Ala (A)	3.2	0.34	GAG	Glu (E)	1.9	0.30	GGG	Gly (G)	0.9	0.13	G
	U				C				A				G				

Statistical Approaches

- The codon usage array
  - The arrays for coding regions and for non-coding regions are different – enabling one to use them for gene prediction
- In human, CGC and AGG code for the same amino acid (Arg) but have very different frequencies
  - GCG is 12x more likely to be used in genes than AGG
  - ORF that prefers CGC over AGG while coding for Arg is likely candidate gene

Statistical Approaches

- The codon usage in Homo sapiens

	U			C			A			G		
U	UUU	Phe	57	UCU	Ser	16	UAU	Tyr	58	UGU	Cys	45
	UUC	Phe	43	UCC	Ser	15	UAC	Tyr	42	UGC	Cys	55
	UUA	Leu	13	UCA	Ser	13	UAA	Stp	62	UGA	Stp	30
	UUG	Leu	13	UCG	Ser	15	UAG	Stp	8	UGG	Trp	100
C	CUU	Leu	11	CCU	Pro	17	CAU	His	57	CGU	Arg	37
	CUC	Leu	10	CCC	Pro	17	CAC	His	43	CGC	Arg	38
	CUA	Leu	4	CCA	Pro	20	CAA	Gln	45	CGA	Arg	7
	CUG	Leu	49	CCG	Pro	51	CAG	Gln	66	CGG	Arg	10
A	AUU	Ile	50	ACU	Thr	18	AAU	Asn	46	AGU	Ser	15
	AUC	Ile	41	ACC	Thr	42	AAC	Asn	54	AGC	Ser	26
	AUA	Ile	9	ACA	Thr	15	AAA	Lys	75	AGA	Arg	5
	AUG	Met	100	ACG	Thr	26	AAG	Lys	25	AGG	Arg	3
G	GUU	Val	27	GCU	Ala	17	GAU	Asp	63	GGU	Gly	34
	GUC	Val	21	GCC	Ala	27	GAC	Asp	37	GGC	Gly	39
	GUA	Val	16	GCA	Ala	22	GAA	Glu	68	GGA	Gly	12
	GUG	Val	36	GCG	Ala	34	GAG	Glu	32	GGG	Gly	15

Statistical Approaches

- The codon usage array in ***E.COLI*** genes

	Codon	Amino acid[2]	%[3]	Ratio[4]	Codon	Amino acid	%	Ratio	Codon	Amino acid	%	Ratio	Codon	Amino acid	%	Ratio	
U	UUU	Phe (F)	1.9	0.51	UCU	Ser (S)	1.1	0.19	UAU	Tyr (Y)	1.6	0.53	UGU	Cys (C)	0.4	0.43	U
	UUC	Phe (F)	1.8	0.49	UCC	Ser (S)	1.0	0.17	UAC	Tyr (Y)	1.4	0.47	UGC	Cys (C)	0.6	0.57	C
	UUA	Leu (L)	1.0	0.11	UCA	Ser (S)	0.7	0.12	UAA	STOP	0.2	0.62	UGA	STOP	0.1	0.30	A
	UUG	Leu (L)	1.1	0.11	UCG	Ser (S)	0.8	0.13	UAG	STOP	0.03	0.09	UGG	Trp (W)	1.4	1.00	G
C	CUU	Leu (L)	1.0	0.10	CCU	Pro (P)	0.7	0.16	CAU	His (H)	1.2	0.52	CGU	Arg (R)	2.4	0.42	U
	CUC	Leu (L)	0.9	0.10	CCC	Pro (P)	0.4	0.10	CAC	His (H)	1.1	0.48	CGC	Arg (R)	2.2	0.37	C
	CUA	Leu (L)	0.3	0.03	CCA	Pro (P)	0.8	0.20	CAA	Gln (Q)	1.3	0.31	CGA	Arg (R)	0.3	0.05	A
	CUG	Leu (L)	5.2	0.55	CCG	Pro (P)	2.4	0.55	CAG	Gln (Q)	2.9	0.69	CGG	Arg (R)	0.5	0.08	G
A	AUU	Ile (I)	2.7	0.47	ACU	Thr (T)	1.2	0.21	AAU	Asn (N)	1.6	0.39	AGU	Ser (S)	0.7	0.13	U
	AUC	Ile (I)	2.7	0.46	ACC	Thr (T)	2.4	0.43	AAC	Asn (N)	2.6	0.61	AGC	Ser (S)	1.5	0.27	C
	AUA	Ile (I)	0.4	0.07	ACA	Thr (T)	0.1	0.30	AAA	Lys (K)	3.8	0.76	AGA	Arg (R)	0.2	0.04	A
	AUG	Met (M)	2.6	1.00	ACG	Thr (T)	1.3	0.23	AAG	Lys (K)	1.2	0.24	AGG	Arg (R)	0.2	0.03	G
G	GUU	Val (V)	2.0	0.29	GCU	Ala (A)	1.8	0.19	GAU	Asp (D)	3.3	0.59	GGU	Gly (G)	2.8	0.38	U
	GUC	Val (V)	1.4	0.20	GCC	Ala (A)	2.3	0.25	GAC	Asp (D)	2.3	0.41	GGC	Gly (G)	3.0	0.40	C
	GUA	Val (V)	1.2	0.17	GCA	Ala (A)	2.1	0.22	GAA	Glu (E)	4.4	0.70	GGA	Gly (G)	0.7	0.09	A
	GUG	Val (V)	2.4	0.34	GCG	Ala (A)	3.2	0.34	GAG	Glu (E)	1.9	0.30	GGG	Gly (G)	0.9	0.13	G
	U				C				A				G				

Statistical Approaches

- The likelihood ratio approach
  - Compute conditional probabilities of the DNA sequence in a window, under the hypotheses:
    - The window contains a coding sequence
    - The window contains a noncoding sequence
  - Slide the window along the DNA sequence and calculate the likelihood
    - Genes are often showed as peaks in the likelihood ratio plot

Statistical Approaches

- These approaches are successful in prokaryotes, but using them with eukaryotes is complicated by exon-intron structure
  - Average length of exon in vertebrates – 130 nucleotides
  - 130 nucleotides is too short to produce reliable peaks because they are not different enough from random variations
- Many researchers have used a more biologically oriented approach to recognize the splicing signals at the exon-intron junctions
  - Profiles for splice sites are weak and thereby limited success
  - Replaced by hidden Markov model (HMM) approaches

Similarity-Based Approaches

- Uses previously sequenced genes, ***G***, and their protein products as a template for the recognition of unknown target genes, ***T***
- Combinatorial puzzle:
  - Given a known target protein and a genomic sequence
  - Find a set of substrings (candidate exons) whose concatenation (splicing) best fits the target

Similarity-Based Approaches

- Naive Brute Force - The spliced alignment problem
  - Find all local similarities between the genomic sequence, G, and the target protein sequence, T
  - Each substring from G that exhibits sufficient similarity to T could be considered a putative exon (possible exon)
    - a putative may not be flanked by **AG** and **GT** dinucleotide
  - The resulting set may contain overlapping substrings
  - Choose the best subset of nonoverlapping substrings as a putative exon structure
    - Exon in real genes do not overlap

Similarity-Based Approaches

- Modeling a putative exon
  - Weighted interval – (l, r, w)
    - l : left-hand position
    - r : right-hand position
    - w : weight, reflects the likelihood that this interval is an exon
  - Chain – a set of nonoverlapping weighted intervals
  - Total weight of a chain
    - The sum of the weights of the intervals in the chain
  - Maximum chain
    - A chain with maximum total weight among all possible chains

Similarity-Based Approaches

- Exon Chaining Problem

Exon Chaining Problem:
Given a set of putative exons, find a maximum set of nonoverlapping putative exons.

Input: A set of weighted intervals (putative exons).

Output: A maximum chain of intervals from this set.

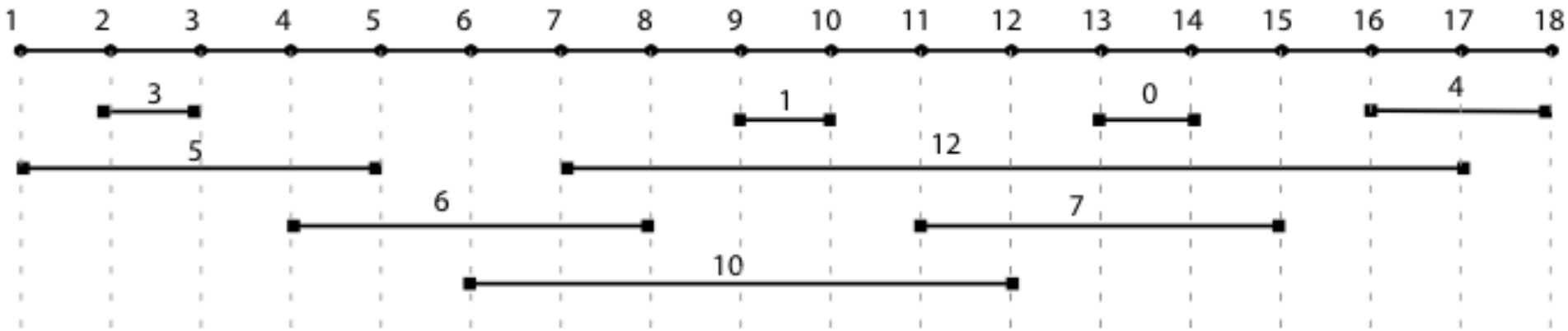
- Problem of n intervals can be solved by dynamic programming in a graph G on $2n$ vertices: for *left* and *right* positions
- Assuming that the set of vertices are sorted into increasing order

$$(v_1, v_2, \ldots, v_{2n})$$

Similarity-Based Approaches

- Exon Chaining Problem

Sorted vertex: $(v_1, v_2, \ldots v_{2n})$



Similarity-Based Approaches

- Exon Chaining Problem
  - There are $(3n - 1)$ edges in the graph:
    - An edge for each interval, between l_i and r_i, with weight w_i
    - $(2n - 1)$ edges of weight 0 which connect adjacent vertices
  - S_i – the length of the longest path in the graph ending with v_i
  - S_{2n} – the solution to the problem

Similarity-Based Approaches

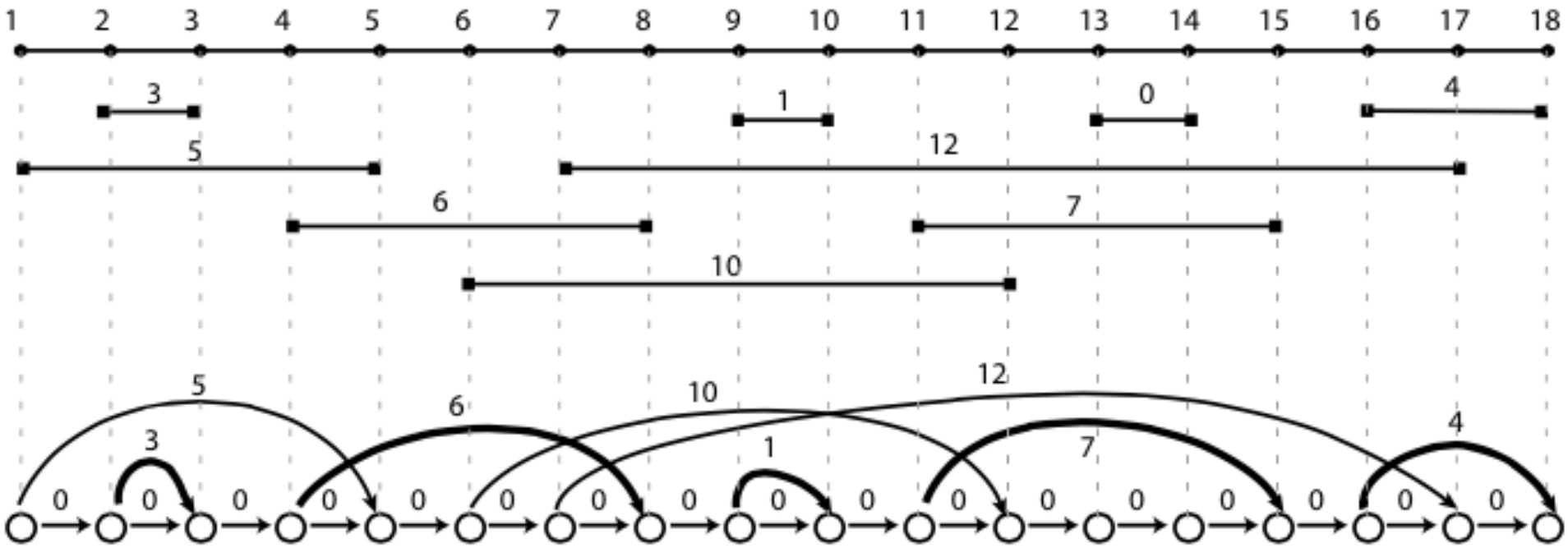
- EXONCHAINING Algorithm

```
EXONCHAINING(G, n)
for i ← 1 to 2n
    s_i ← 0
for i ← 1 to 2n
    if vertex v_i in G corresponds to the right end of an interval I
        j ← index of vertex for left end of the interval I
        w ← weight of the interval I
        s_i ← max {s_j + w, s_{i-1}}
    else
        s_i ← s_{i-1}
return s_{2n}
```

Similarity-Based Approaches

- Exon Chaining Problem

Sorted vertex: $(v_1, v_2, \ldots v_{2n})$

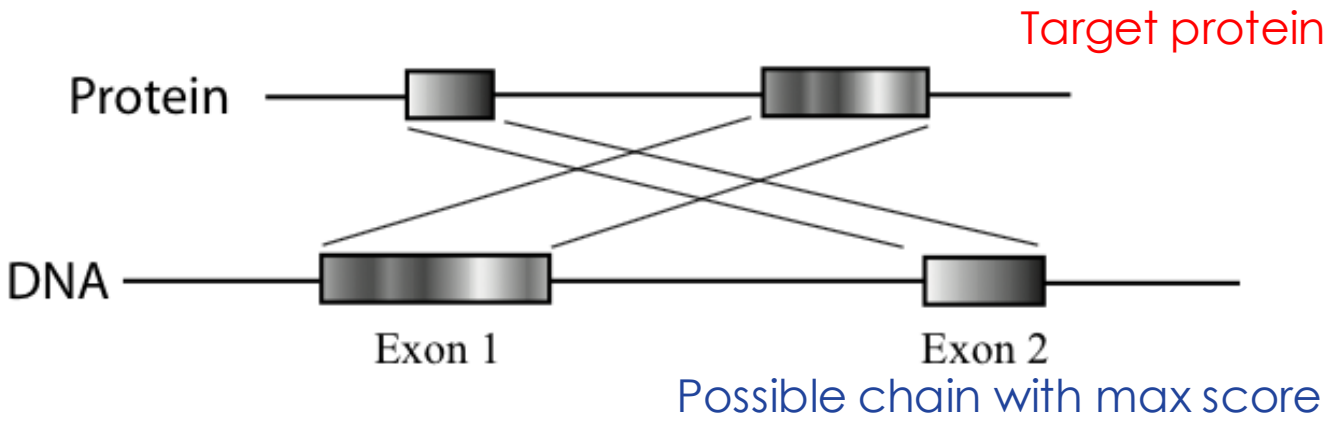


0	0	3	3	5	5	5	9	9	10	10	15	15	15	17	17	17	20

S_i – length of the longest path to vertex v_i

Similarity-Based Approaches

- Exon Chaining Problem
  - Disadvantages
    - The endpoints of putative exons are not well defined
    - Optimal chain of intervals may not correspond to any valid alignment



Spliced Alignment

- In 1996, Mikhail Gelfand and colleagues proposed the spliced alignment approach to find genes in eukaryotes
  - Given a genomic sequence and a set of candidate exons
  - Explore all possible exon assemblies and find a chain of exons which best fits a related target protein
- A set of candidate exons - block
  - All putative exons between potential **AG** and **GT**, or
  - All substrings similar to target protein (local similarities)

Spliced Alignment

- Next step is to filter the set of candidate exons very gentle
  - This left a set of candidate exons that may contains many false exons, but definitely contains all the true ones
- Given the set of (filtered) candidate exons (aka blocks) and a target protein sequence
  - Explore all possible chains (assemblies)
  - Find the assembly with the highest similarity score to the target

Spliced Alignment

- Spliced Alignment Problem
  - Genomic sequence: $G = g_1 \ldots g_n$
  - Target sequence: $T = t_1 \ldots t_m$
  - Chain Γ : a sequence of nonoverlapping blocks
  - String Γ^* : a string formed by the chain Γ
    - We are looking for a string with highest similarity to the target sequence (global alignment), $s(\Gamma^*, T)$

Spliced Alignment

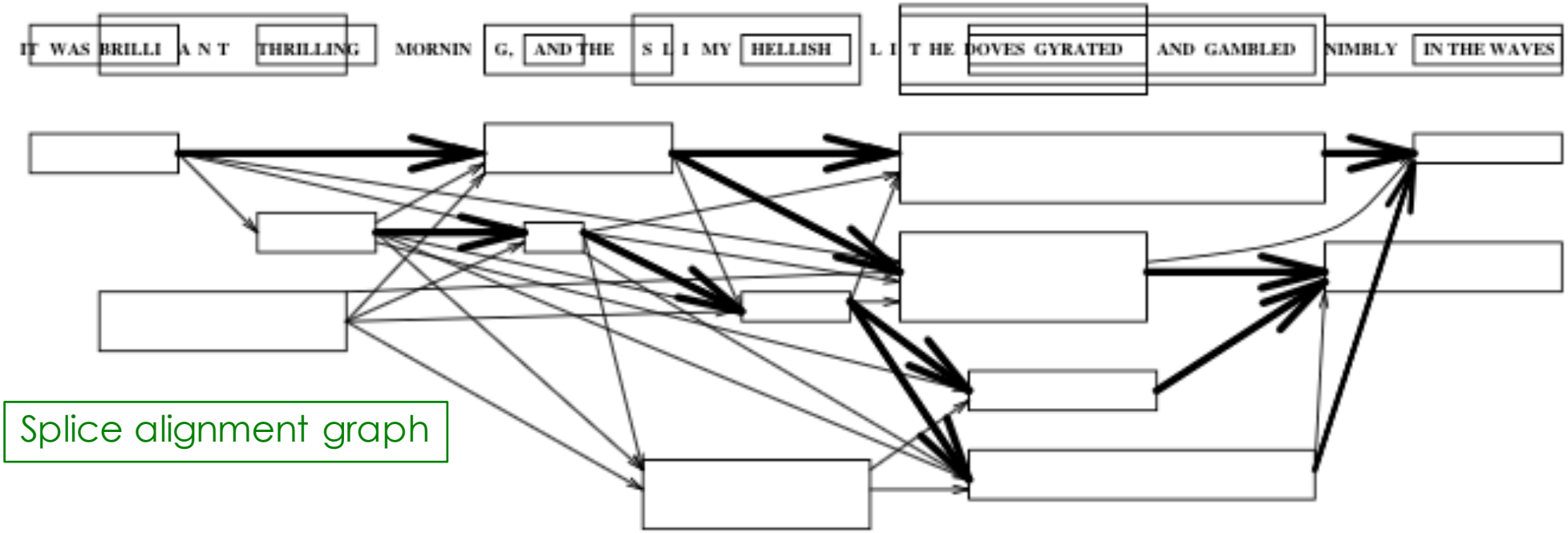
- Spliced Alignment Problem

Spliced Alignment Problem:
Find a chain of candidate exons in a genomic sequence that best fits a target sequence.

Input: Genomic sequence G, target sequence T, and a set of candidate exons (blocks) $\mathcal{B}$.

Output: A chain of candidate exons Γ such that the global alignment score $s(\Gamma^*, T)$ is maximum among all chains of candidate exons from $\mathcal{B}$.

Spliced Alignment



Splice alignment graph

Target: 'T WAS BR I LLI G, AND TH E SLI TH Y TOVES DI D GYRE AND GI M BLE I N TH E WABE

T WAS BR I LLI	G,	AND	TH E	SL	TH E	DOVES	GYRATED	AND	GAM BLED	I N	TH E	WAVE			
T WAS BR I LLI	G,	AND	TH E	SL	TH E	DOVES	GYRATED	NI M BLY	I N	TH E	WAVE				
T	HR I LLI NG	AND	H EL	LI S H	DOVES	GYRATED	AND	GAM BLED	I N	TH E	WAVE				
T	HR I LLI NG	AND	H EL	LI S H	DOVES	GYRATED	NI M BLY	I N	TH E	WAVE					

4 different block assemblies – best fit to the target is the first one

Spliced Alignment

- Spliced Alignment Problem
  - Similar to the problem of finding path in DAG
  - Vertices : blocks (candidate exons)
  - Edges : edges connect nonoverlapping blocks
  - Every path gives out a string obtained by concatenation of labels of its vertices
  - Weight of a path – the score of the optimal alignment between the concatenated blocks of a path and the target sequence
    - Not defined weights for individual edges

Spliced Alignment

- Similarity score between *i-prefix* of the blocks and *j-prefix* of the target sequence, T
  - $B = g_{left} \ldots g_i \ldots g_{right}$ (candidate exon containing position i)
  - $B(i) = g_{left} \ldots g_i$ (*i-prefix* of B)
  - $End(B) = right$ (the rightmost index of B)
- If the chain $\Gamma = (B1, B2, \ldots , B)$

$$\Gamma^*(i) = B_1 \circ B_2 \circ \cdots \circ B(i)$$

Spliced Alignment

- The score of the optimal spliced alignment between *i-prefix* of G and the *j-prefix* of T

$$S(i,j,B) = \max_{\text{all chains } \Gamma \text{ ending in } B} s(\Gamma^*(i), T(j)).$$

Assuming that this alignment ends in block B

- If i is NOT the starting vertex of block B:

$$S(i,j,B) = \max \begin{cases} S(i-1,j,B) - \sigma \\ S(i,j-1,B) - \sigma \\ S(i-1,j-1,B) + \delta(g_i, t_j) \end{cases}$$

Spliced Alignment

- If i is the starting vertex of block B:

$$S(i,j,B) = \max \begin{cases} S(i,j-1,B) - \sigma \\ \max_{\text{all blocks } B' \text{ preceding } B} S(end(B'), j-1, B') + \delta(g_i, t_j), \\ \max_{\text{all blocks } B' \text{ preceding } B} S(end(B'), j, B') - \sigma, \end{cases}$$

- After calculate the table $S(i, j, B)$, the score of the optimal spliced alignment is

$$\max_B S(end(B), m, B),$$

Spliced Alignment

- We can reduce the number of edges in the graph by making a transformation