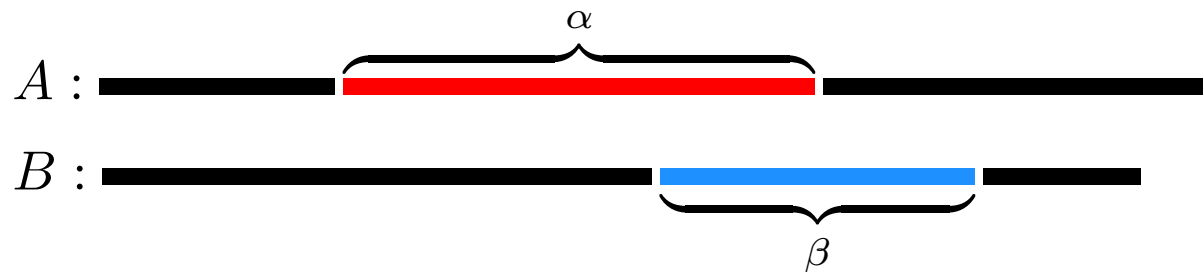


Comparative gene finding

Input: Let $A[1..n]$ and $B[1..m]$ be two DNA sequences which are known to contain a homologous gene ...

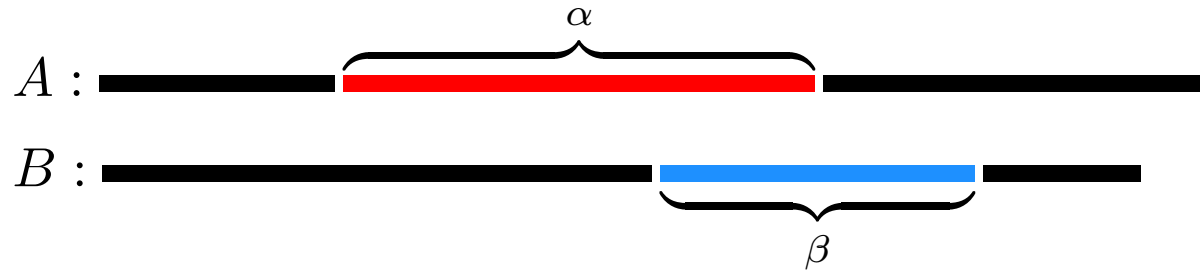


Problem: find the gene structure of A and B ...

Idea: use pairwise sequence comparison, recall local alignment ...

Local alignment, part 1

Searching for region of two strings which are highly similar ...



Local alignment:

Given two strings $A[1..n]$ and $B[1..m]$ and a similarity measure $Sim(A, B)$. Find a pair of substrings α of A and β of B , such that $Sim(\alpha, \beta)$ is maximal for all choices of α and β , i.e.:

$$LocalSim(A, B) = \max_{\alpha, \beta} Sim(\alpha, \beta)$$

If $Sim(A, B)$ is an alignment score, then $LocalSim(A, B)$ can be computed in time $O(|A| \cdot |B|)$ [Smith and Waterman, 1981] ...

Local alignment, part 2

$$LocalSim(A, B) = \max_{i,j} \max_{h \leq i, k \leq j} Sim(A[h+1..i], B[k+1..j]) = \max_{i,j} v(i, j)$$

We can compute $v(i, j)$ recursively ...

$$v(i, j) = \max \begin{cases} v(i-1, j-1) + s(A[i], B[j]) & i > 0 \text{ and } j > 0 \\ v(i-1, j) - \alpha & i > 0 \text{ and } j \geq 0 \\ v(i, j-1) - \alpha & i \geq 0 \text{ and } j > 0 \\ 0 & i \geq 0 \text{ and } j \geq 0 \end{cases}$$

Algorithm: Compute v row by row; find (i', j') s.t. $v(i', j') = \max v(i, j)$; back-track from (i', j') to (h, k) where $v(h, k) = 0$...

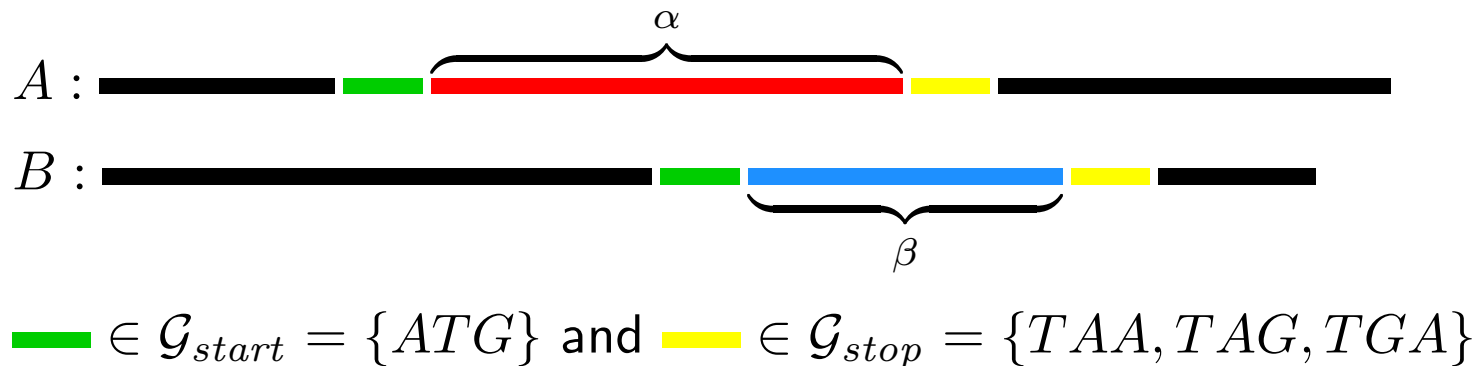
$$Sim(\underbrace{A[h+1..i']}_{\alpha}, \underbrace{B[k+1..j']}_{\beta}) = v(i', j') = \max_{i,j} v(i, j) = LocalSim(A, B)$$

Prokaryotic gene finding, part 1

Observation: coding regions evolve slower than non-coding regions, i.e. local sequence similarity can be used as a *gene finder*

... highly similar regions are probably coding for homologous genes

Idea: extend “local alignment algorithm” with “gene syntax” ...



Gene finding: find legal genes $\alpha = A[h + 1 .. i]$ and $\beta = B[k + 1 .. j]$ such that $Sim(\alpha, \beta)$ is maximal over all legal gene structures ...

Prokaryotic gene finding, part 2

$$\text{GeneSim}(A, B) = \max_{i,j} \max_{h \leq i, k \leq j} \text{Sim}(A[h+1..i], B[k+1..j]) = \max_{i,j} S(i, j)$$

where

- $A[h-2..h], B[k-2..k] \in \mathcal{G}_{start}$
- $A[i+1..i+3], B[j+1..j+3] \in \mathcal{G}_{stop}$

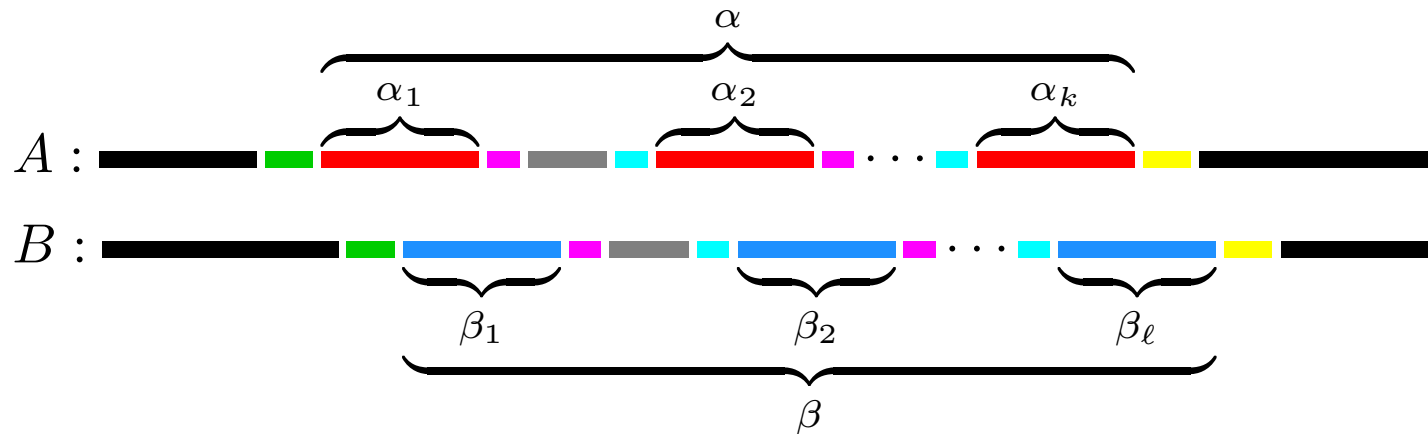
Compute $S(i, j)$ cf. local alignment recursion ...

$$S(i, j) = \max \begin{cases} S(i-1, j-1) + \delta(a_i, b_j) \\ S(i-1, j) + \lambda \\ S(i, j-1) + \lambda \\ 0 \end{cases} \quad \text{if } A[i-2..i], B[j-2..j] \in \mathcal{G}_{start}$$

Find $S(i', j') = \max S(i, j)$ s.t. $A[i'+1..i'+3], B[j'+1..j'+3] \in \mathcal{G}_{stop}$;
back-track from (i', j') to (h, k) where $S(h, k) = 0$...

Eukaryotic gene finding, part 1

Recall that eukaryotic genes consist of introns and exons ...



■ $\in \mathcal{G}_{start} = \{ATG\}$ and ■ $\in \mathcal{G}_{stop} = \{TAA, TAG, TGA\}$

■ $\in \mathcal{D} = \{GT\}$ and ■ $\in \mathcal{A} = \{AG\}$

Gene finding: find legal genes $\alpha = A[h + 1 .. i]$ and $\beta = B[k + 1 .. j]$ with exons $\alpha_1, \alpha_2, \dots, \alpha_k$ and $\beta_1, \beta_2, \dots, \beta_l$ such that $Sim(\alpha, \beta)$ is maximal over all legal gene structures ...

Eukaryotic gene finding, part 2

... same idea as before cf. local alignment but use a similarity function which ignores introns by treating them as “free gaps” ...

$$\text{GeneSim}(A, B) = \max_{i,j} \max_{h \leq i, k \leq j} \text{Sim}'(A[h+1..i], B[k+1..j]) = \max_{i,j} S(i, j)$$

where

- $A[h-2..h], B[k-2..k] \in \mathcal{G}_{start}$
- $A[i+1..i+3], B[j+1..j+3] \in \mathcal{G}_{stop}$

Keep track of $S(i, j)$ as the maximum similarity of any $A[h+1..i]$ and $B[k+1..j]$ with legal exon structures and ending in an exon (possibly of length zero), where $A[h-2..h], B[k-2..k] \in \mathcal{G}_{start}$...



— Extending the gene finder —

We can make the gene finder more sensitive in various ways . . .

“signal sensors”:

- use a dedicated tool to define \mathcal{A} , \mathcal{D} , \mathcal{G}_{start} , and \mathcal{G}_{stop}

“context sensors”:

- take the codons of the predicted exons into account, e.g. stop codons should not be allowed within exons
- consider the encoded proteins of the predicted exons when defining the similarity of exons
- gap lengths in exons are usually a multiple of three, otherwise they induce a frame shift, i.e. change the tail of the encoded protein
- use an affine gap cost

— Evaluating the gene finder, part 1 —

... examine eukaryotic gene finding on simulated and real data. Simulated data is obtained by simulating the evolution of homologues pair of genes from an ancestor gene cf. the Jukes-Cantor model ...

Method I: find a best scoring sub-alignment of legal exon structures of an existing alignment (obtained by standard global alignment)

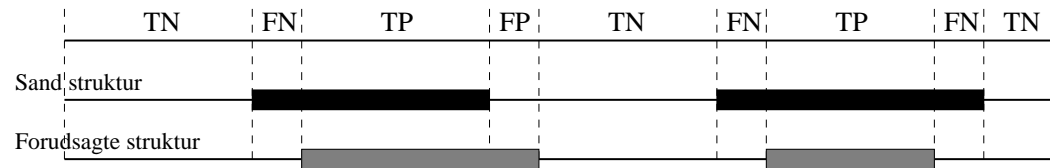
Method II: construct a best scoring local alignment of legal exon structures (as presented in this talk)

... incorporate the extensions on the previous slide, and test both methods using a “DNA level score” and a “DNA and protein level score” ...

<http://www.birc.dk/Software/GenePair/>

Quality measures, part 1

Counting nucleotides: True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN):



Sensitivity: Fraction of coding nucleoties being predicted as being coding:

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

Specificity: Fraction of predicted coding nucleotides which are coding

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

Quality measures, part 2

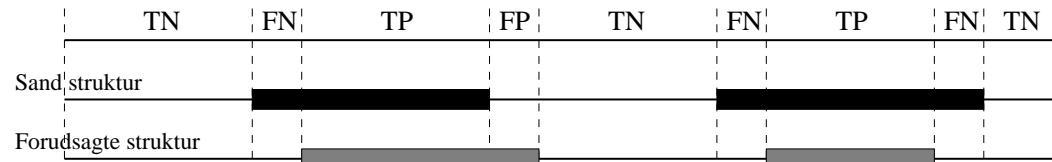
Correlation coefficient:

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

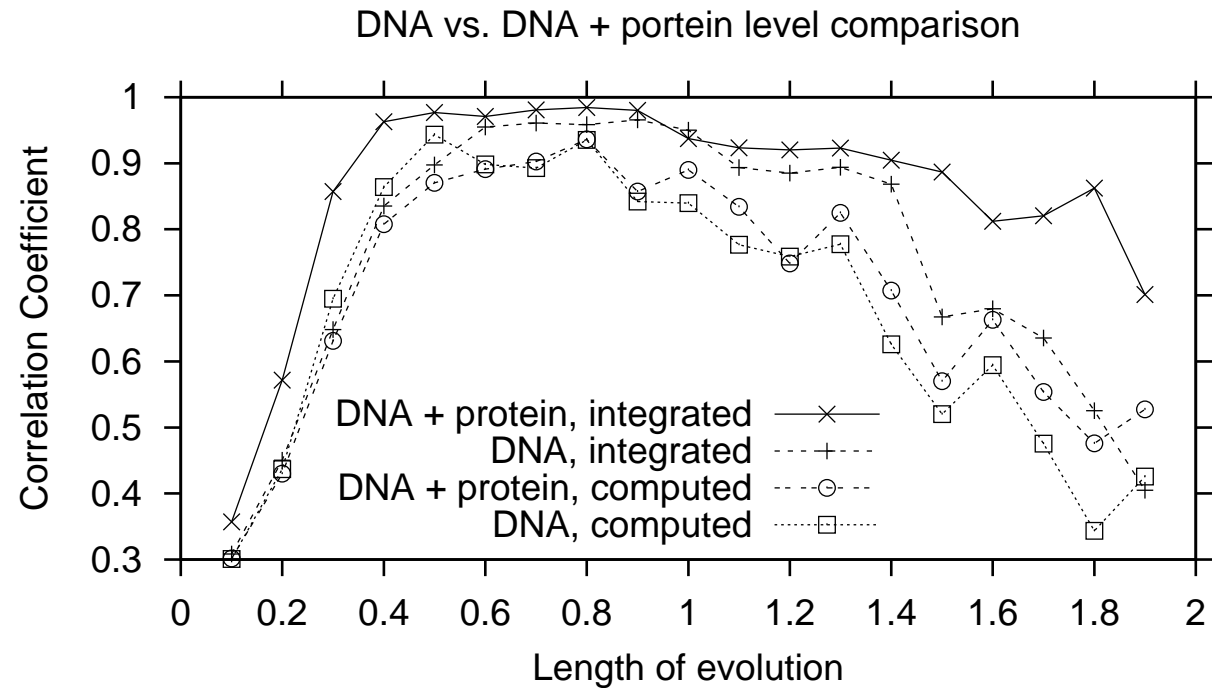
Approximate correlation:

$$AC = \left(\frac{1}{4} \cdot \left(\frac{TP}{TP + FN} + \frac{TP}{TP + FP} + \frac{TN}{TN + FP} + \frac{TN}{TN + FN} \right) - \frac{1}{2} \right)$$

Evaluating the gene finder, part 2



$$CC = (TP \cdot TN - FP \cdot FN) / \sqrt{(TP + FP) \cdot (TN + FN) \cdot (TP + FN) \cdot (TN + FP)}$$



Evaluating the gene finder, part 3

The real data consists of 117 pair of homologous genes from man and mouse which had Genbank entries in 1998 [Batzolou *et. al.*, 2000]

Sensitivity: The fraction of coding nucleotides which are predicted as coding.

Specificity: The fraction of prediction coding nucleotides which are indeed coding.

| Gene finding method | Specificity | Sensitivity | Correlation |
|---|-------------|-------------|-------------|
| DNA level, computed alignment | 0.89 | 0.92 | 0.88 |
| DNA level, glass alignment | 0.92 | 0.93 | 0.90 |
| DNA level, integrated alignment | 0.88 | 0.97 | 0.91 |
| DNA and protein level, computed alignment | 0.89 | 0.93 | 0.88 |
| DNA and protein level, glass alignment | 0.92 | 0.93 | 0.90 |
| DNA and protein level, integrated alignment | 0.92 | 0.98 | 0.94 |
| GLASS/ROSETTA | 0.97 | 0.95 | – |
| Genscan | 0.89 | 0.98 | – |

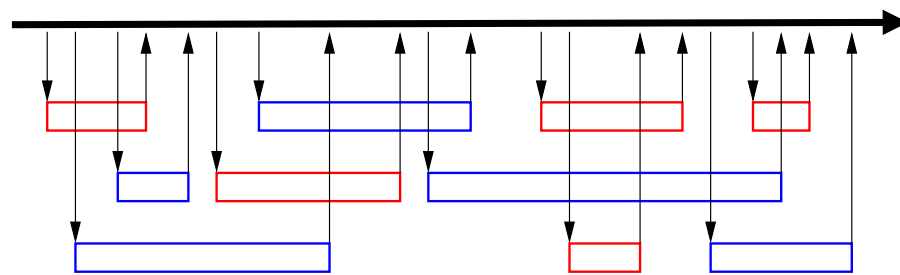
Simple method, good performance, but quadratic running time $O(|A| \cdot |B|)$ becomes a problem for long strings ...

Gene finding using Protein Templates

... instead of comparing two DNA strings for finding the best pair of gene structures, use a *target protein* to find a matching gene structure ...

Procrustes: from <http://www-hto.usc.edu/software/procrustes/>

... Given a genomic sequence and a set of candidate exons, the spliced alignment algorithm explores all possible exon assemblies and finds a chain of exons with the best fit to a related target protein. The set of candidate exons is constructed by selection of all blocks between candidate acceptor and donor sites ...



Target protein:

GSAQVKGHGKKVADALTN

Gene finding: given a DNA string S and a protein P , find legal exons $\alpha_1, \alpha_2, \dots, \alpha_k$ in S such that $Sim(P, \text{trans}(\alpha_1 \alpha_2 \dots \alpha_k))$ is maximal over all legal gene structures

Solvable in time $O(|S| \cdot |P|)$ [Gelfand *et al.*, 1996]

Summary

... gene finding is about detecting regions and deciding their functionality. A typical gene finder is constructed from *signal* and *context* sensors, and a method for combining regions into a legal gene structure ...

Employs summary statistics, sequence profiles, sequence homology, and probabilistic modeling, e.g. hidden Markov models ...



Interesting WWW pages:

A bibliography of gene finding literature:

<http://www.nslj-genetics.org/gene/>

Bioinformatics in general

<http://www.expasy.ch>