Bioinformatics

Applications in Molecular Biology problems

- **Multiple sequence alignment problem:** a multiple global sequence of k>2 strings S={ S_1, S_2,…., $S_κ$} is a physical generalization of aligning for two strings.

Όνομα Ακολουθίας	**Στοίχιση Ακολουθιών**																			
	1									10										20
P.falciparum	M	M	E	Q	V	C	D	V	F	D	I	Y	A	I	C	A	C	C	K	V
P.vivax	-	M	E	D	L	S	D	V	F	D	I	Y	A	I	C	A	C	C	K	V
P.chabaudi	-	M	E	D	I	S	E	I	F	D	I	Y	A	I	C	A	C	C	K	V
P.berghei	-	M	E	D	L	S	E	T	F	D	I	Y	A	I	C	A	C	C	K	V
P.vinckei	-	-	-	-	-	-	-	-	-	-	-	-	A	I	C	A	C	C	K	V
L.major	A	D	F	A	F	P	S	L	R	A	F	S	I	V	V	A	L	D	M	-
E.coli	-	-	-	-	-	-	-	-	-	M	I	S	L	I	A	A	L	A	V	-
L.casei	-	-	-	-	-	-	-	-	-	-	T	A	F	L	W	A	Q	N	R	-
H.sapiens	-	-	-	-	-	-	M	V	G	S	L	N	C	I	V	A	V	S	Q	-

Why we are interested in multiple sequence alignment

- Multiple sequence alignment is used:
  - to identify and represent protein families and super-families,
  - to represent characteristics that are transferred to DNA sequences or protein families,
  - To represent the evolution history (phylogenetic trees) in DNA or protein sequences.

Multiple sequence alignment

- Kind of alignments:
  - Extention of DP approach (too costly)
  - Use of pairwise alignment (center star algorithm)
- Algorithms of multiple sequence alignemnt:
  - FASTA
  - BLAST.

Sequence Database Searching

Steps to determine a protein sequence

1. Compare the new sequence with PROSITE and BLOCKS in order to locate well-characterized sequence motifs.
2. Search in DNA and protein sequence databases (Genbank, Swiss-Prot, etc.) for locating sequences that are locally similar (when using a local similarity criterion) – using FASTA and BLAST
3. If these searches provide interesting results, then use the technique of dynamic programming.
4. When we need to involve amino acid substitution matrices, we usually use a variant of the Dayhoff PAM matrix and BLOSUM matrix.

BLAST Algorithm

- BLAST: Basic Local Alignment Search Tool, Altschul et. Al. 1990
- Basic idea: locate common sub-sequences of the same length (segment pairs) that appear in the input query sequence and the set of data base sequences. In the sequel extend in order to locate maximal segment pairs

Algorithm	query	sequence
BLASTP	Protein	Protein
BLASTN	Nucleotide	Nucleotide
BLASTX	Nucleotide	Protein
TBLASTN	Protein	Nucleotide
TBLASTX	Nucleotide	Nucleotide

1° βήμα: τμηματοποίηση της δοσμένης ακολουθίας σε διαδοχικές υπο-λέξεις μεγέθους w=3																									
Query sequence:																									
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Words																									
a	b	c																							
	b	c	d																						
		c	d	e																					
			d	e	f																				
2° βήμα: Εντοπισμός των υπο-λέξεων με μέγιστη τιμή στοίχισης για το όλες τις ακολουθίες																									
High-scoring matching words:																									
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	b	c																							
d	s	k	o	w	j	j	d	f	k	s	l	m	n	k	d	k	j	d	f	k	k	j	d	f	f
											l	m	n												
m	s	l	z	m	s	o	w	u	r	n	f	k	s	a	d	e	f	a	q	m	a	z	m	s	l
															d	e	f								
3° βήμα: επέκταση των high-scoring words																									
a	b	c	d	w	f	h	h	f	j	s	l	m	n	k	d	k	j	d	e	h	k	k	j	f	f
a	b	c	⇒							⇐	l	m	n	⇒											

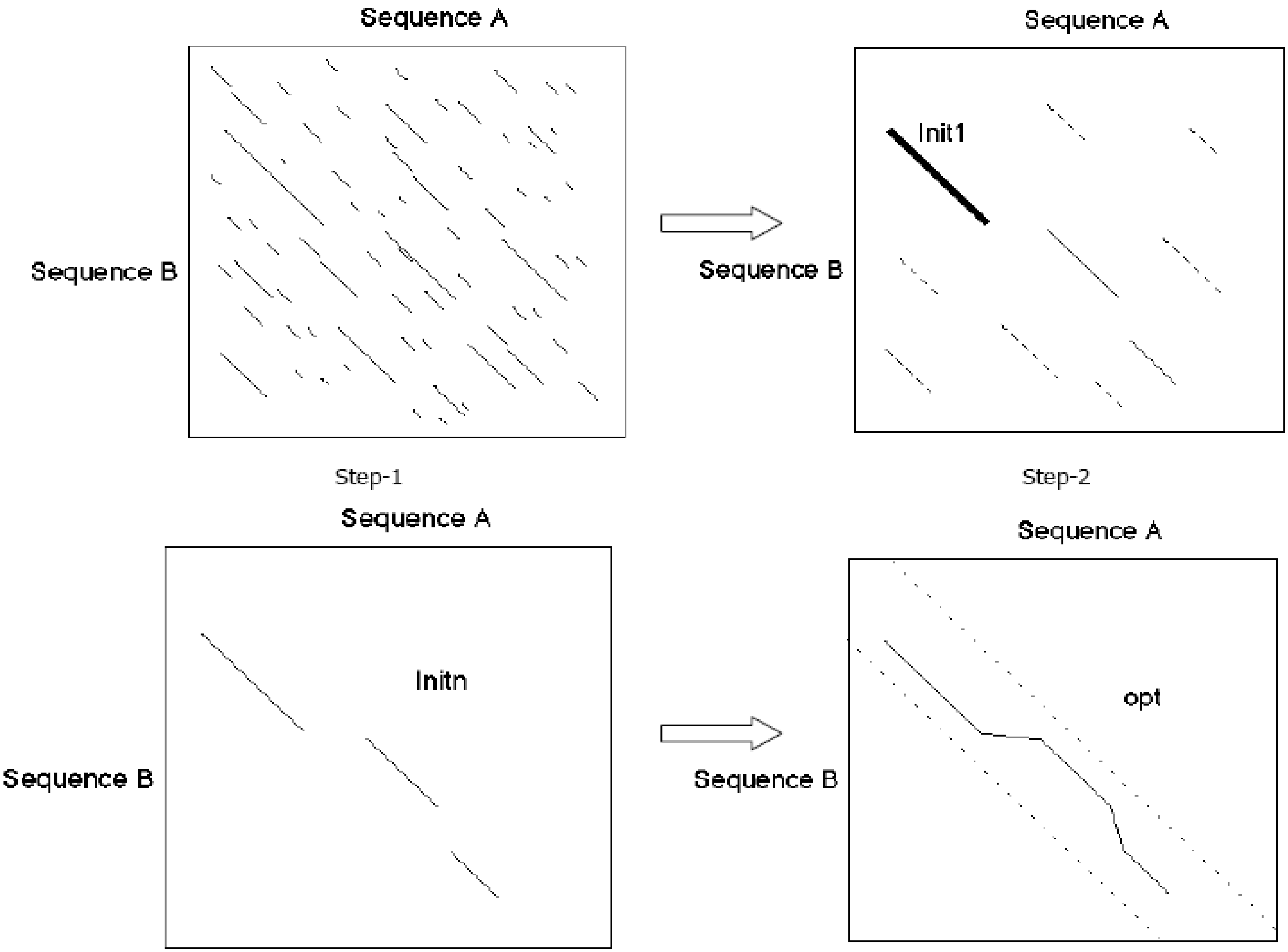
FASTA

- FAST: Fast – All, Lipman et al. 1985
- Central idea: use of small words (words ή k-tuples) that appear in both sequences. In the case of protein sequences the word length is το 1-2 residues while for DNA sequences the word length can reach 6 bases

FASTA

- 1o step: we search for words of length ktup in the dynamic programming table: ‘hot spots’ (pairs *(i,j)*)
- 2o step: we locate the ten better diagonal runs– diagonal runs από ‘hot-spots’ στον πίνακα

  (a hot spot define the (i-j)-diagonal, the score is the sum of scores of hot-spots plus weighted decreasing as the distance increases)
- 3o step: we align «good sub-alignments»
- 4o step: we produce the best path



Step-1

Step-2

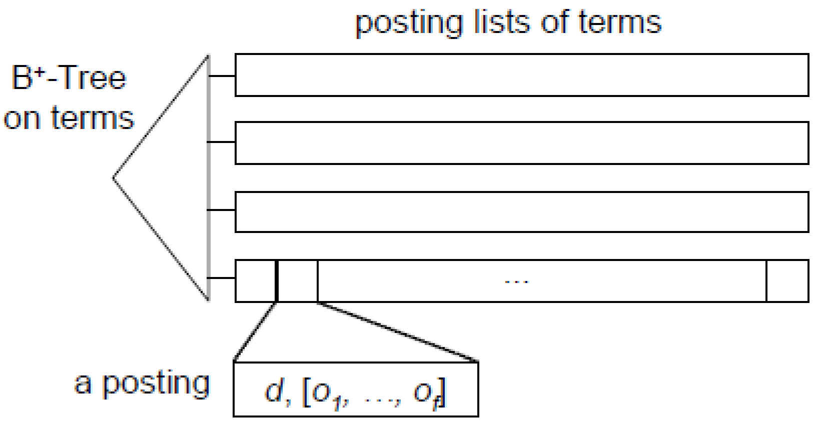
Step-3

Step-4

n-Gram/2L: A Space and Time Efficient Two-Level n-Gram Inverted Index Structure

Min-Soo Kim, Kyu-Young Whang, Jae-Gil Lee, Min-Jae Lee

Department of Computer Science and Advanced Information Technology Research Center (AITrc)
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
{mskim, kywhang, jglee, mjlee}@mozart.kaist.ac.kr



d: document identifier
o_i: offset where term *t* occurs in document *d*
f: frequency of occurrence of term *t* in document *d*

Figure 1: The structure of the inverted index.

Indexing Approaches

- k-gram indexing
- direct indexing
- vector space indexing

k-gram Indexing

- Hash tables (FASTP, FASTA, BLAST, BL2SEQ, PSI-BLAST, MegaBLAST, BLASTZ, WU-BLAST, BLAT, SSAHA, SENSEI)

- ed-tree (the gram size *k*, the skip interval *Δ*, the segment length vector $H=[h_1, \ldots, h_t]$, where $\text{sum}_i(h_i)=k$)

-- For each sequence, the algorithm generates all k-grams with Δ skips

-- Each gram is partitioned according to H, and the partitions are inserted into the ed-tree.

The search algorithm partitions accordingly the query string, and initiates a search procedure, beginning from the root. Some important properties: (1) the k-grams are usually longer than that of BLAST, (2) they allow inexact k-gram matches, (3) only one k-gram out of Δ k-grams is indexed,

Ed-tree is used by CHAOS, LAGAN, DIALIGN.

Direct Indexing

- Suffix trees

  -- MUMmer, AVID, REPuter, MGA, QUASAR

- VP-trees (Vantage Point tree)

MUMmer

- *Detecting MUM* (Maximal Unique Matches), that is pair of subsequences (x’,y’) that exactly match and there is no other matching pair that contains *x*’, *y*’ simultaneously (just use the GST(x,y))

- Find the backbone of the alignment: all the pairs (*x*’,*y*’) are sorted in increasing order of the position of *x*’. Next the longest sequence of MUMs whose subsequences from *x*, *y* are in sorted order is found. These form the backbone

- Closing gaps. The gaps between consecutive MUMs are aligned with the help of Smith-Waterman

Similar tools are AVID, REPuter, MGA.

VP-trees (Vantage Point tree)

The VP-tree has been adapted to sequence databases where the distance is the edit distance or the block edit distance. It can be applied to other distance functions as long as they are almost metric.

The algorithm takes a database, $D=\{s_1,\ldots,s_n\}$, and chooses a sequence s as the root, while the median of the distances to *s* is computed. The two sets are: (i) the sequences that are closer to s than the median, (ii) the rest of the sequences.

Given a query *q*, the sequences at distance *r* are found as follows: First, *q* is compared to the vantage sequence *s*, at the root node. Let M be the median distance.

1. If $d(q,s) \leq r$, *s* is inserted to the result set
2. If $d(q,s) \leq r+M$, then the left child is searched recursively.
3. If $d(q,s) \geq M-r$, then the right child is searched recursively.

Vector Space Indexing

These index structures, map sequences or subsequences to vectors in a vector space.

There exist two important index structures:

(i) SST (Sequence Search Tree)

(i) MRS (Multi Resolution String) index

SST

- Vector Space Mapping (window size *w*, shift amount *Δ*, tuple size *k*)

The parameter *k*, determines the size of the computed vector (for alphabet size *σ* this vector has size σ^{κ}

- The produced vectors are stored in a so called centroid structure.

- A query is performed as follows: a query sequence is first divided into subsequences of window size *w*, using a shift amount of *Δ=w/2*. Each of the produced query vectors is then searched on the index structure starting from the root node.

Indexing Approaches

- k-gram indexing
- direct indexing
- vector space indexing

Index Structures for ApproximateMatching in Sequence Databases
Tamer Kahveci and Ambuj K. Singh in Srinivas Aluru,
Handbook of Computational Molecular Biology, CRC Press 2005.

k-gram Indexing

- Hash tables (FASTP, FASTA, BLAST, BL2SEQ, PSI-BLAST, MegaBLAST, BLASTZ, WU-BLAST, BLAT, SSAHA, SENSEI)

- ed-tree (the gram size *k*, the skip interval *Δ*, the segment length vector *H*=[h_1, …, h_t], where $\text{sum}_i(h_i)$=k)

-- For each sequence, the algorithm generates all k-grams with Δ skips

-- Each gram is partitioned according to H, and the partitions are inserted into the ed-tree.

The search algorithm partitions accordingly the query string, and initiates a search procedure, beginning from the root. Some important properties: (1) the k-grams are usually longer than that of BLAST, (2) they allow inexact k-gram matches, (3) only one k-gram out of Δ k-grams is indexed,

Ed-tree is used by CHAOS, LAGAN, DIALIGN.

Index Structures for ApproximateMatching in Sequence Databases

Tamer Kahveci and Ambuj K. Singh in Srinivas Aluru,

Handbook of Computational Molecular Biology, CRC Press 2005.

Direct Indexing

- Suffix trees

  -- MUMmer, AVID, REPuter, MGA, QUASAR

- VP-trees (Vantage Point tree)

Index Structures for ApproximateMatching in Sequence Databases
Tamer Kahveci and Ambuj K. Singh in Srinivas Aluru,
Handbook of Computational Molecular Biology, CRC Press 2005.

MUMmer

- *Detecting MUM* (Maximal Unique Matches), that is pair of subsequences (x’,y’) that exactly match and there is no other matching pair that contains *x*’, *y*’ simultaneously (just use the GST(x,y))

- Find the backbone of the alignment: all the pairs (*x*’,*y*’) are sorted in increasing order of the position of *x*’. Next the longest sequence of MUMs whose subsequences from *x*, *y* are in sorted order is found. These form the backbone

- Closing gaps. The gaps between consecutive MUMs are aligned with the help of Smith-Waterman

Similar tools are AVID, REPuter, MGA.

Index Structures for ApproximateMatching in Sequence Databases
Tamer Kahveci and Ambuj K. Singh in Srinivas Aluru,
Handbook of Computational Molecular Biology, CRC Press 2005.

VP-trees (Vantage Point tree)

The VP-tree has been adapted to sequence databases where the distance is the edit distance or the block edit distance. It can be applied to other distance functions as long as they are almost metric.

The algorithm takes a database, $D=\{s_1,\ldots,s_n\}$, and chooses a sequence s as the root, while the median of the distances to *s* is computed. The two sets are: (i) the sequences that are closer to s than the median, (ii) the rest of the sequences.

Given a query *q*, the sequences at distance *r* are found as follows: First, *q* is compared to the vantage sequence *s*, at the root node. Let M be the median distance.

1. If $d(q,s) \le r$, *s* is inserted to the result set
2. If $d(q,s) \le r+M$, then the left child is searched recursively.
3. If $d(q,s) \ge M-r$, then the right child is searched recursively.

Index Structures for ApproximateMatching in Sequence Databases
Tamer Kahveci and Ambuj K. Singh in Srinivas Aluru,
Handbook of Computational Molecular Biology, CRC Press 2005.

Vector Space Indexing

These index structures, map sequences or subsequences to vectors in a vector space.

There exist two important index structures:

(i) SST (Sequence Search Tree)

(i) MRS (Multi Resolution String) index

Index Structures for ApproximateMatching in Sequence Databases
Tamer Kahveci and Ambuj K. Singh in Srinivas Aluru,
Handbook of Computational Molecular Biology, CRC Press 2005.

SST

- Vector Space Mapping (window size *w*, shift amount *Δ*, tuple size *k*)

The parameter *k*, determines the size of the computed vector (for alphabet size *σ* this vector has size σ^k

- The produced vectors are stored in a so called centroid structure.

- A query is performed as follows: a query sequence is first divided into subsequences of window size *w*, using a shift amount of *Δ=w/2*. Each of the produced query vectors is then searched on the index structure starting from the root node.

Index Structures for Approximate Matching in Sequence Databases
Tamer Kahveci and Ambuj K. Singh in Srinivas Aluru,
Handbook of Computational Molecular Biology, CRC Press 2005.