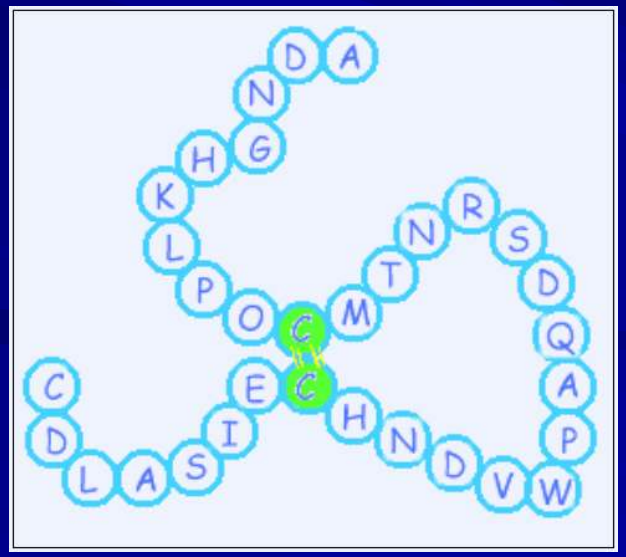
On Compressibility of Protein Sequences

Donald Adjeroh and Fei Nan
Data Compression Conference, 2006. Proceedings. DCC 2006 pages 422- 431

Presented by Chen LIN

Protein

- String of amino acids
- 20 types amino acids – symbols
  (log2(20)=4.32 bps)
- Protein sequence – the entire collection of all the proteins in an organism



Content

- Why interest in compressing protein
- Why compression is difficult
- Long-Range Correlation
- Compressing Protein Sequences

Why interest?

- Exponential growth of biological sequences →

  Efficient storage and data transport

- Compression is a means for analyzing protein sequences

Why difficult?

- Difficult to model

  Due to the apparent randomness of symbols, model derived for proteins is different from models derived for text.

  Classical compression methods – expansion.

  "Protein is incompressible"

  -- Nevill-Manning C.G & Witten I.H.

  The IEEE Data Compression Conference, 257- 299, 1999

In studying the SCP statistics of genomic sequences, the authors bumped unto an unusual observation:

an unprecedented redundancy in protein sequences – long range correlation

Sorted Common Prefix (SCP)

- Using the relationship between the BWT and the suffix tree, we can obtain all the sorted suffixes of an input sequence in linear time.
- A table of sorted suffixes

Example: Sequence S = “BRATATBAT…”

Index	Sorted Suffixes		Sorted Common Prefix
1	SS1 = ATATBAT…		AT
2	SS2 = ATBAT…	→	AT
3	SS3 = AT…		AT
4	SS4 = BAT….		B
5	SS5 = BRATATBAT…		B
…	…		…
ith	SSi		
jth	SSj		

Result of Sorted Common Prefix (SCP)

Kmax – the maximum common prefix for a given sequence

Table 3: SCP Statistics for Concatenated Protein Sequences

Seq	size, *u*	# of genes	Kmax	Kmax/*u*	Start Index1	Start Index2	Diff
HI	448770	1740	220685	0.492	53200	8	53192
MJ	509508	1680	343105	0.673	34899	3	34896
SC	2900346	8220	886531	0.306	480296	29	480267
HS	3295749	5733	392004	0.119	358676	24	358652

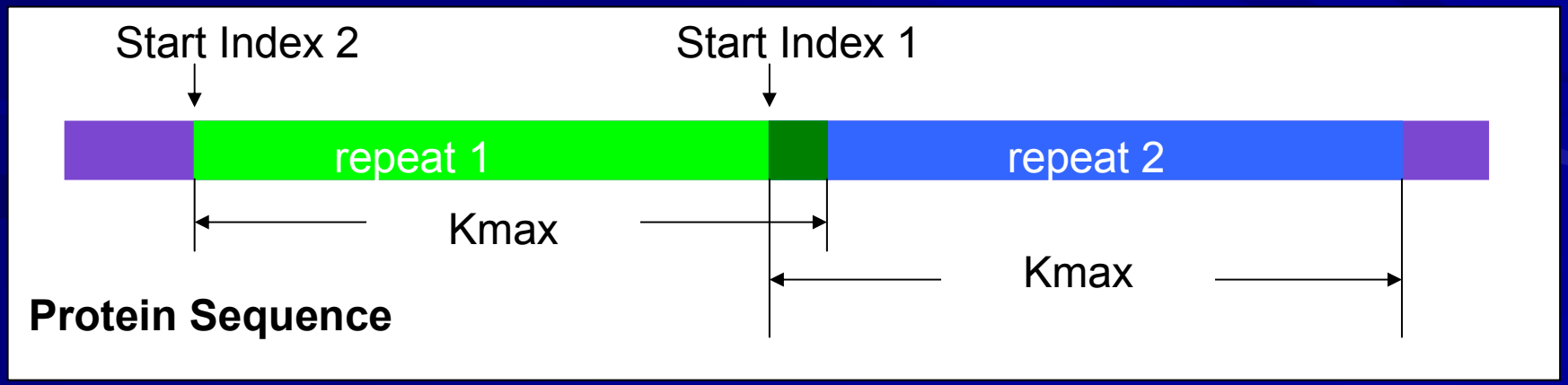
HI: *H. influenzae*; MJ: *M. jannaschii*; HS: *H. sapiens*; SC: *S. cerevisiae*

Diff=|Index1-Index2|

Long-Range Correlation

- Unusually high values of Kmax
- Long range of separation between the repeats (common prefix)

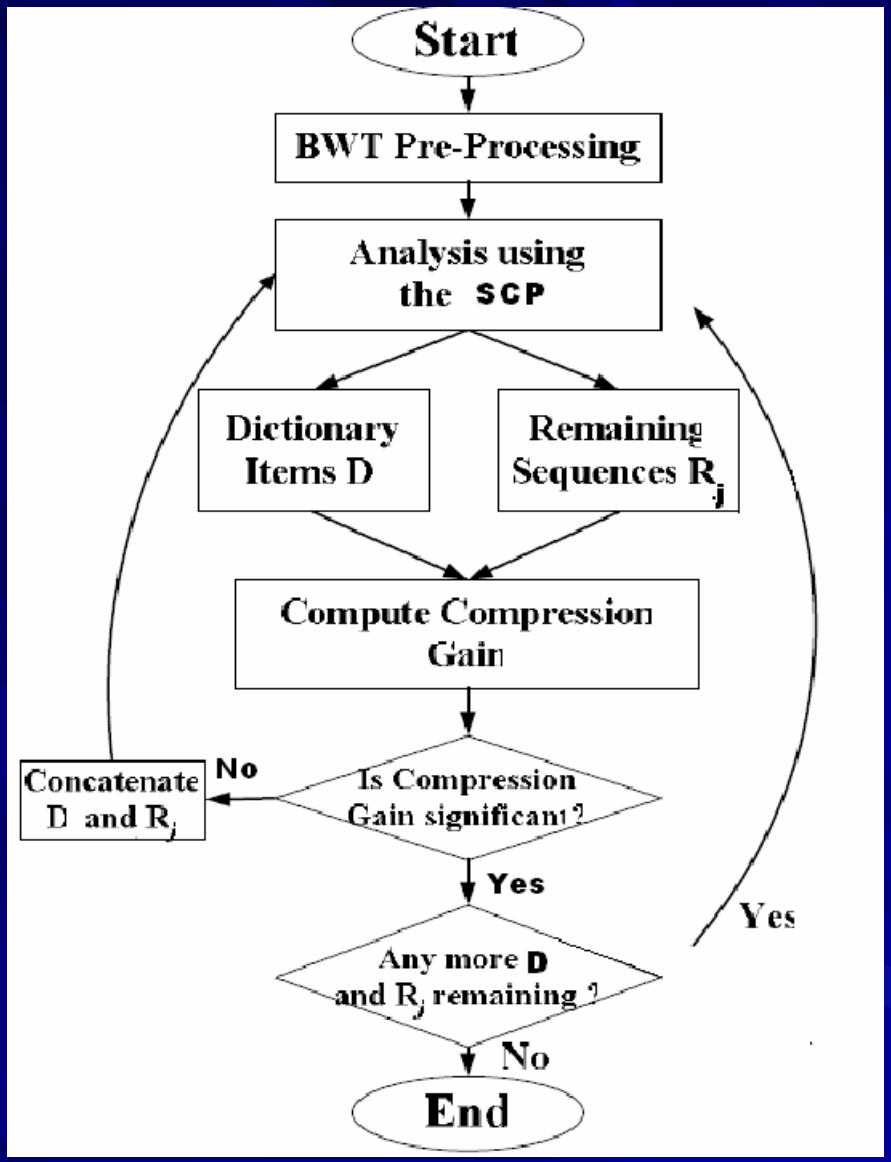
  separation > 350,000 protein symbols
- Kmax > |Index2 – Index1|



Compressing Protein Sequences

A dictionary-based compression:

- 1) remove the repeated substring from the input sequence, and move it to an external dictionary. → D + R
- 2) In the dictionary, record the positions in the sequence where each repetition occurred, along with the repetition type.
- 3) Pass D and R to the core algorithm again until no compression could be achieved



Flowchart of proposed compression algorithm

- Feed dictionary entries D & remaining sequence R for further decomposition and Parsing
- Stops when the compression gain G(S) is negative or less than a threshold.
- Each leaf in the tree is part of the final output

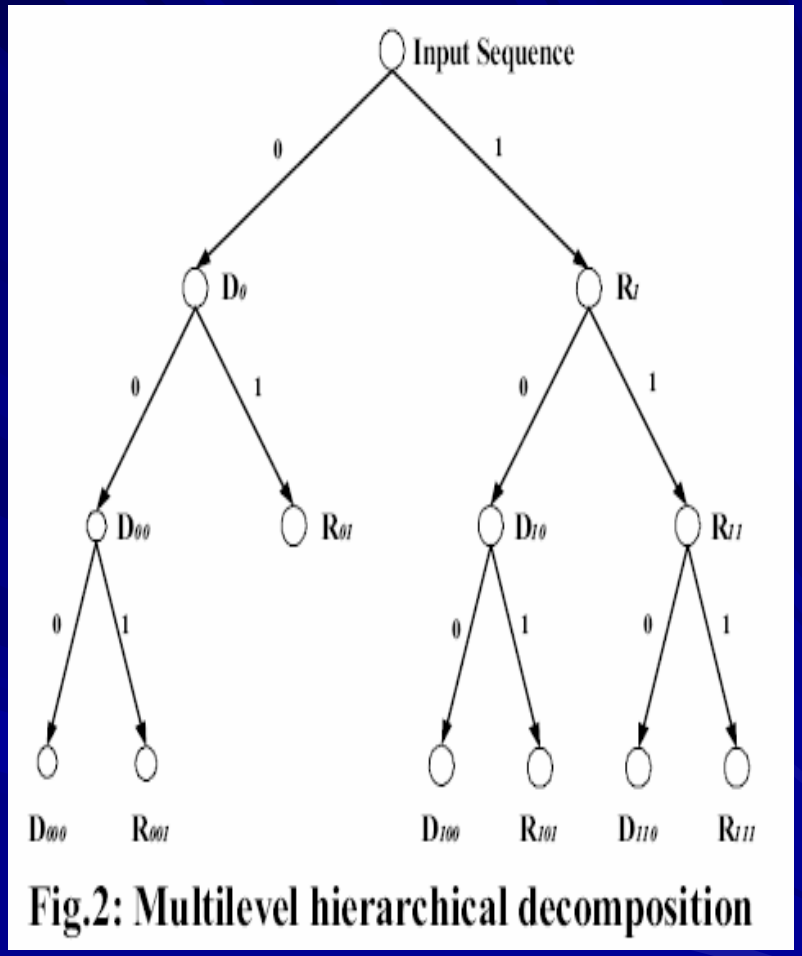
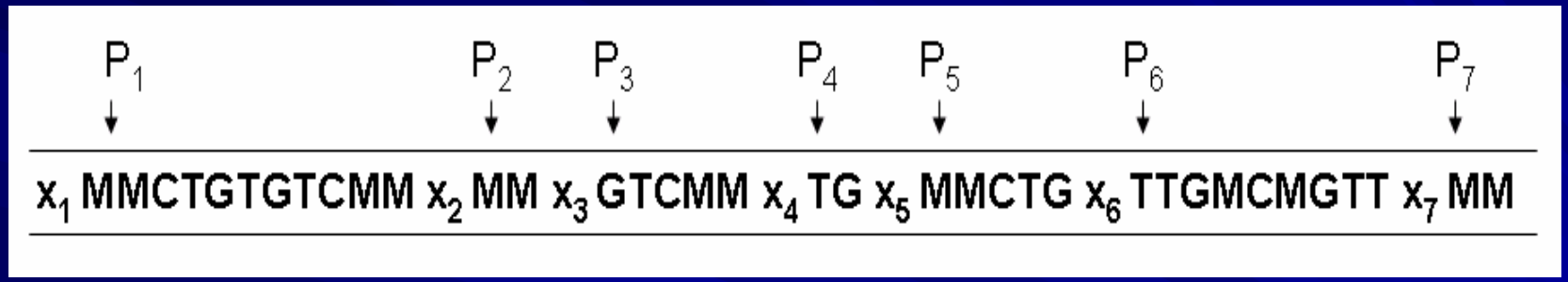


Fig.2: Multilevel hierarchical decomposition

Example of Parsing and Encoding

- Sample sequence S:

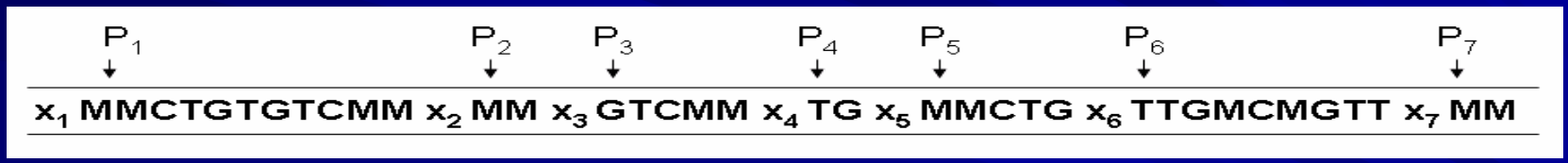


- Remaining parsed sequence:

$$\textbf{Parse(S)} : x_1x_2x_3x_4x_5x_6x_7$$

Example -- Dictionary

index	Repeat Pattern	*l(r)*	*t(r)*	*η(r)*	positions
1	MMCTGTCMM	9	1	1	P_1
			3	1	P_6
2	GTCMM	5	1	1	P_3
			2	1	P_5
3	MM	2	1	2	P_2, P_7



- Notation:

r = a repetition pattern; l(r) = |r| = length of r;

η(r) = total number of occurrences of r

t(r) = repetition type for current occurrence of r. (1:direct repeat; 2:reverse repeat; 3:complementary palindrome)

Results

Table 4: Statistics of maximal repeats and compression results using the observed long-range correlations.

	Size	Kmax	repeat length $l(r)$	number of occurrence $\eta(r)$	Compression Results (bps)
HI	509508	220685	34896	7	2.546
MJ	448770	343105	53192	5	2.273
SC	2900346	886531	406239	3	3.111
HS	3295749	392004	338359	3	3.435

Table 5: Comparative compression performance using the observed long-range correlations. Compression results in bits/symbol (smaller values imply better performance).

	Gen Compress	CP(0)	CP(1)	CP(2)	CP(3)	lzaCTW (8)	Block Code	Proposed Method
HI	4.156	4.156	4.149	4.146	4.143	4.118	3.665	2.546
MJ	4.062	4.068	4.06	4.056	4.051	4.028	5.102	2.273
SC	3.97	4.163	4.158	4.152	4.146	3.952	5.175	3.111
HS	3.972	4.133	4.126	4.12	4.112	3.920	5.087	3.435

20 symbols → log2(20)=4.32 bps

Conclusion

- Identify the correlated protein sequences based on the sorted common prefix
- Using a dictionary-based parsing and encoding scheme to provide compression
- Can provide consistent compression, at times down to less than 2.3 bps