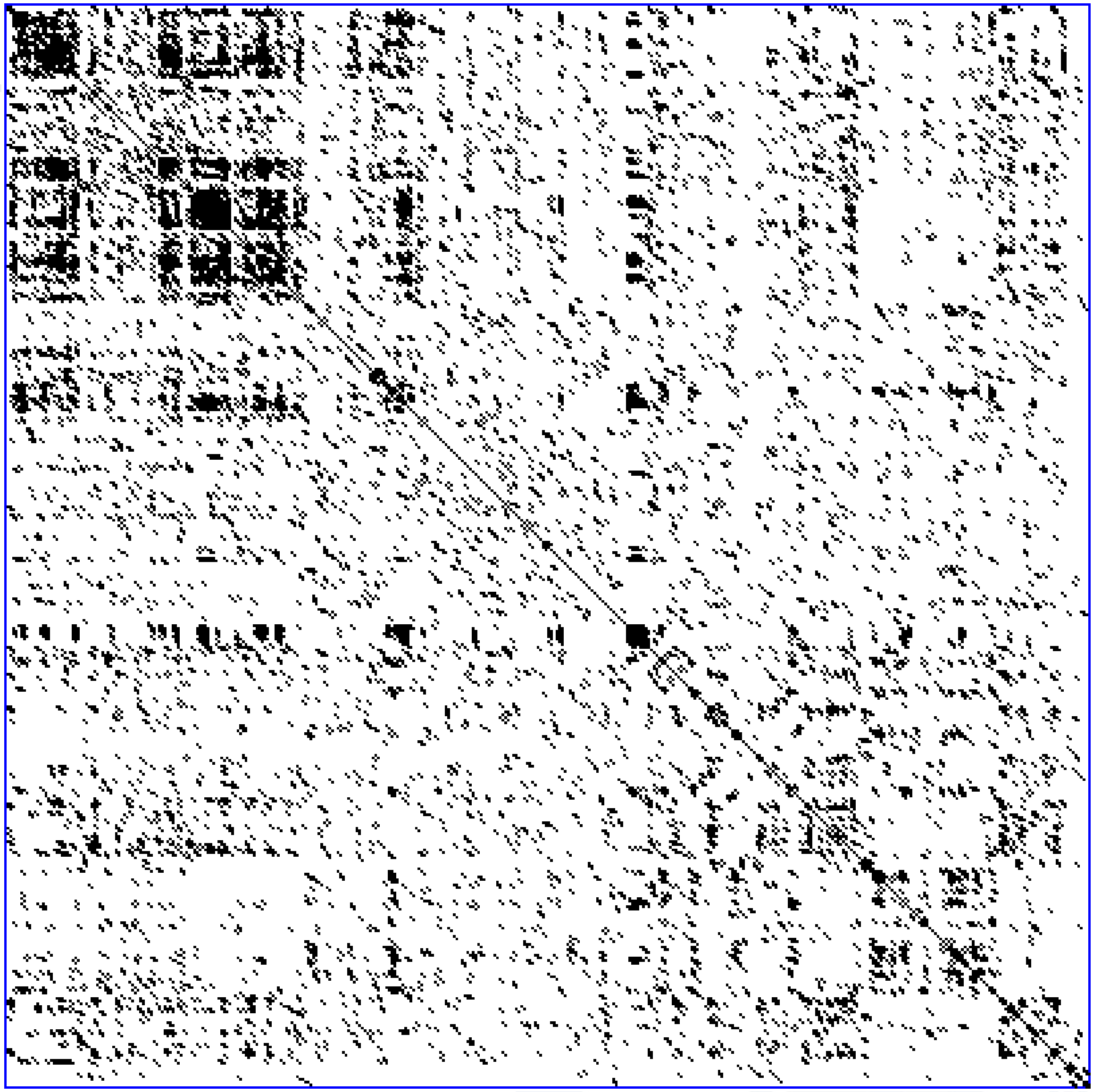
Course organization

- Introduction (Week 1-2)
  - Course introduction
  - A brief introduction to molecular biology
  - A brief introduction to sequence comparison
- Part I: Algorithms for Sequence Analysis (Week 3 - 11)
  - Chapter 1-3, Models and theories
    - » Probability theory and Statistics (Week 3)
    - » Algorithm complexity analysis (Week 4)
    - » Classic algorithms (Week 5)
    - » Lab: Linux and Perl
  - Chapter 4, Sequence alignment (week 6)
  - Chapter 5, Hidden Markov Models (week 8)
  - Chapter 6. Multiple sequence alignment (week 10)
  - Chapter 7. Motif finding (week 11)
  - Chapter 8. Sequence binning (week 11)
- Part II: Algorithms for Network Biology (Week 12 - 16)

Introduction to Sequence Comparison

Chaochun Wei

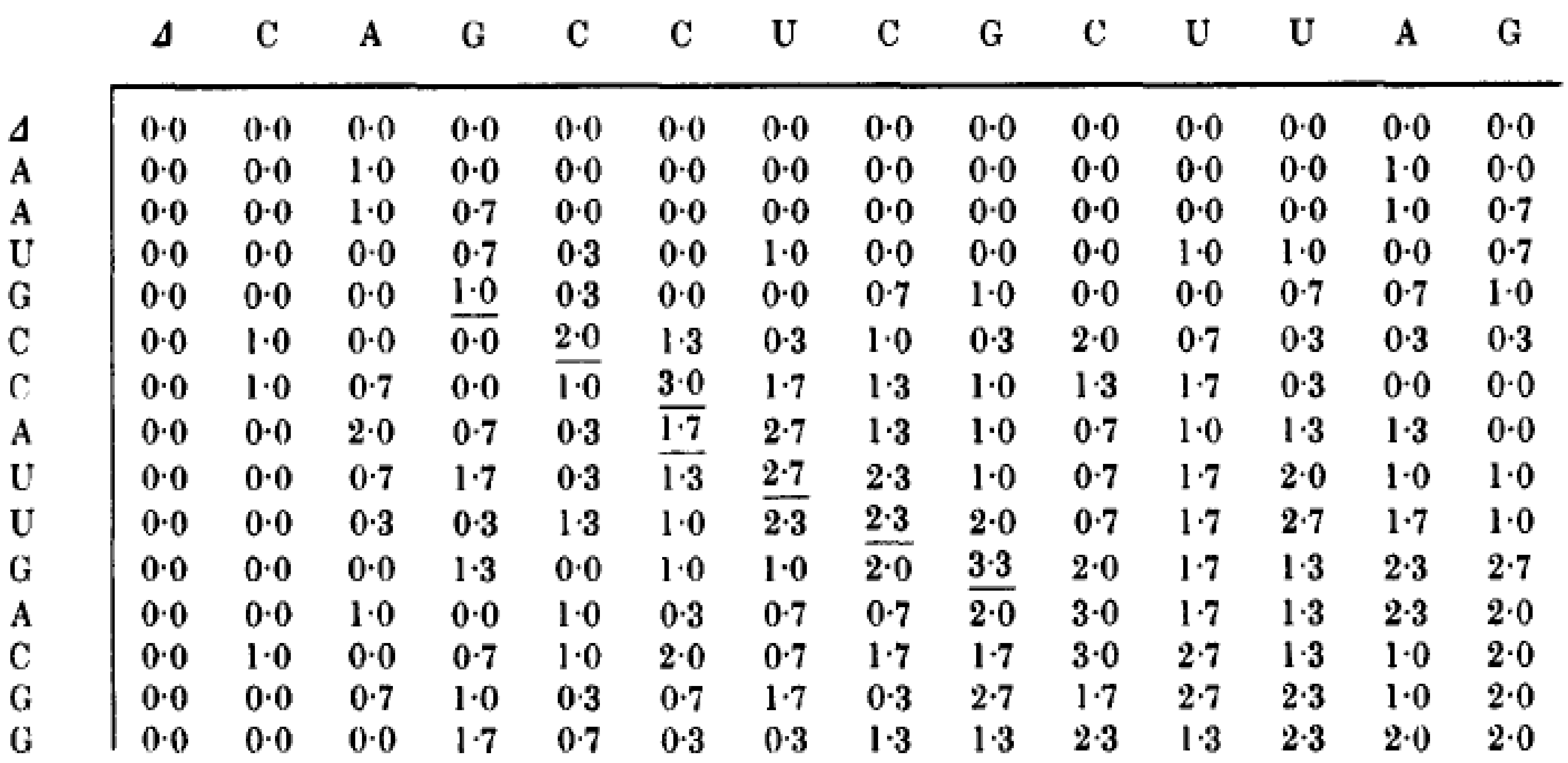
The simple but powerful dot plot


A DNA dot plot of a human zinc finger transcription factor (GenBank ID NM_002383), showing regional self-similarity

Sequence comparison algorithms

- Simple identity (as in C's strcmp())
- Hashing
- Longest common substring

Longest common substring



	Δ	C	A	G	C	C	U	C	G	C	U	U	A	G
Δ	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0
A	0·0	0·0	1·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	1·0	0·0
A	0·0	0·0	1·0	0·7	0·0	0·0	0·0	0·0	0·0	0·0	0·0	0·0	1·0	0·7
U	0·0	0·0	0·0	0·7	0·3	0·0	1·0	0·0	0·0	0·0	1·0	1·0	0·0	0·7
G	0·0	0·0	0·0	1·0	0·3	0·0	0·0	0·7	1·0	0·0	0·0	0·7	0·7	1·0
C	0·0	1·0	0·0	0·0	2·0	1·3	0·3	1·0	0·3	2·0	0·7	0·3	0·3	0·3
C	0·0	1·0	0·7	0·0	1·0	3·0	1·7	1·3	1·0	1·3	1·7	0·3	0·0	0·0
A	0·0	0·0	2·0	0·7	0·3	1·7	2·7	1·3	1·0	0·7	1·0	1·3	1·3	0·0
U	0·0	0·0	0·7	1·7	0·3	1·3	2·7	2·3	1·0	0·7	1·7	2·0	1·0	1·0
U	0·0	0·0	0·3	0·3	1·3	1·0	2·3	2·3	2·0	0·7	1·7	2·7	1·7	1·0
G	0·0	0·0	0·0	1·3	0·0	1·0	1·0	2·0	3·3	2·0	1·7	1·3	2·3	2·7
A	0·0	0·0	1·0	0·0	1·0	0·3	0·7	0·7	2·0	3·0	1·7	1·3	2·3	2·0
C	0·0	1·0	0·0	0·7	1·0	2·0	0·7	1·7	1·7	3·0	2·7	1·3	1·0	2·0
G	0·0	0·0	0·7	1·0	0·3	0·7	1·7	0·3	2·7	1·7	2·7	2·3	1·0	2·0
G	0·0	0·0	0·0	1·7	0·7	0·3	0·3	1·3	1·3	2·3	1·3	2·3	2·0	2·0

FIG. 1. H_{ij} matrix generated from the application of eqn (1) to the sequences A-A-U-G-C-C-A-U-U-G-A-C-G-G and C-A-G-C-C-U-C-G-C-U-U-A-G. The underlined elements indicate the trackback path from the maximal element 3·30.

Analysis of algorithms and big-O notation

Measure the Complexity of an algorithm: O()

- strcmp: O(n)
- longest common substring: O(nm)

Pattern matching algorithms

- Brute force
- Knuth/Morris/Pratt: a finite state automata solution
- Regular expressions and nondeterministic finite state automata

Dynamic programming sequence alignment algorithms

- Needleman/Wunsch global alignment
- Smith/Waterman local alignment
- Linear and affine gap penalties

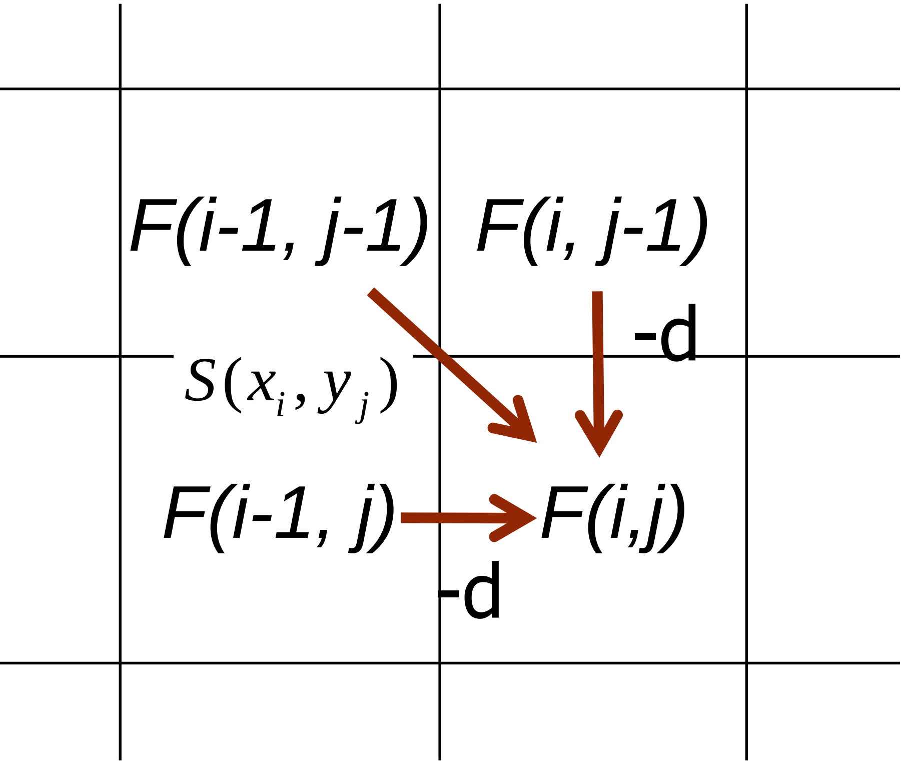
Needleman/Wunsch global alignment (1970)

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to x_i and $Y_{1\ldots j}$ of Y up to Y_j $(0 \le i \le n,\ 0 \le j \le m)$*, then we have*

$$F(0,0) = 0$$

$$F(i,j) = \max\begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Needleman/Wunsch global alignment (1970)



$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Smith/Waterman local alignment (1981)

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to x_i and $Y_{1\ldots j}$ of Y up to Y_j $(0 \le i \le n, 0 \le j \le m)$*, then we have*

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} 0 \\ F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Linear and affine gap penalties

- Linear: $w(k) = k\,d$
- Affine: $w(k) = d + (k-1)\,e$
- Let $M(i,j)$, $I_x(i,j)$, $I_y(i,j)$ be the best scores up to (i,j):
  - $M(i,j)$: x_i is aligned to y_j;
  - $I_x(i,j)$: x_i is aligned to a gap;
  - $I_y(i,j)$ y_j is aligned to a gap

then we have

$$M(i,j) = \max\begin{cases} M(i-1, j-1) + s(x_i, y_j), \\ I_x(i-1, j-1) + s(x_i, y_j), \\ I_y(i-1, j-1) + s(x_i, y_j); \end{cases}$$

$$I_x(i,j) = \max\begin{cases} M(i-1, j) - d, \\ I_x(i-1, j) - e; \end{cases}$$

$$I_y(i,j) = \max\begin{cases} M(i, j-1) - d, \\ I_y(i, j-1) - e. \end{cases}$$

Reading materials

Required

1. "A general method applicable to the search for similarities in the amino acid sequence of two proteins", Needleman, SB and Wunsch, CD. J. Mol. Biol. 48:443-453, 1970
2. "Identification of Common Molecular Subsequences", Smith, TF and Waterman, MS. J. Mol. Biol. 147: 195-197, 1981

   The Smith/Waterman algorithm

Other recommended background:

1. "An improved algorithm for matching biological sequences", Gotoh, O. J. Mol. Biol. 162:705-708, 1982

   The efficient form of the Needleman/Wunsch and Smith/Waterman algorithms.
2. "Optimal alignment in linear space", Myers, E. W. and Miller, W. CABIOS 4: 11-17, 1988.

   More advanced reading: a divide and conquer method to reduce the memory cost from O(n^2) to O(n)

BLAT: Blast-Like Alignment Tool

- Not BLAST
- Indexed on database (BLAST indexed on the query)
  - Need ~1G memory for human genome
- Need some extra time for database initialization (index)
- Can be 500 times faster than BLAST
- Can display results in the UCSC genome browser

Kent, WJ (2002), “BLAT– the BLAST-like alignment tool”, Genome Research, 12(4):656-64
Blat FAQ: http://genome.ucsc.edu/FAQ/FAQblat.html

BLAT

- Designed to quickly find
  - DNA sequences of 95% and greater similarity of length 25 bases or more.
  - Protein sequences of 80% and greater similarity of length 20 amino acids or more.
- In practice
  - DNA BLAT works well on primates, and
  - protein blat on land vertebrates

BLAT—The BLAST-Like Alignment Tool

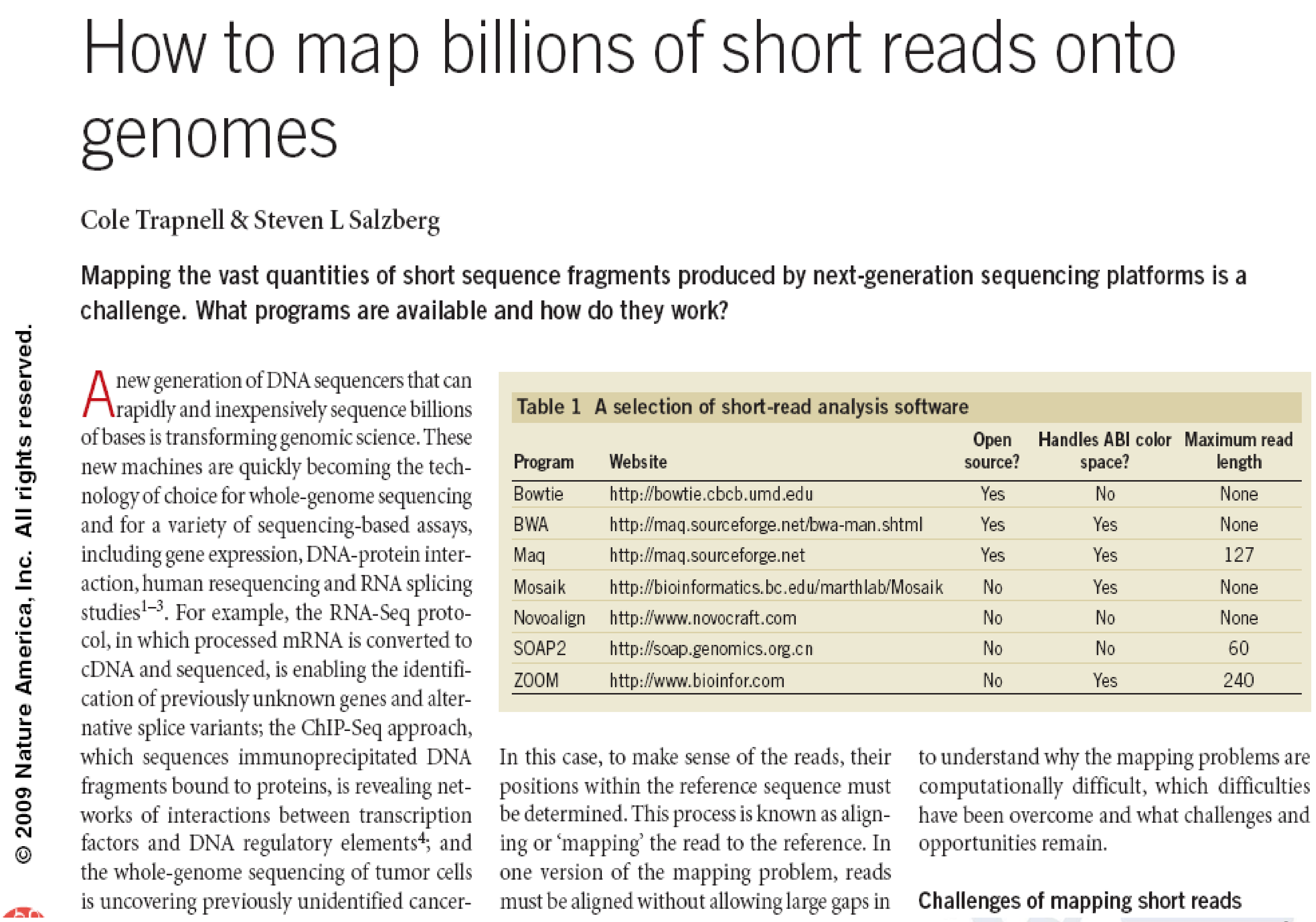
Timing of BLAT vs.WU-TBLASTX on a Data Set of 1000 Mouse Reads against a RepeatMasked Human Chromosome 22

Method	K	N	Matrix	Time
WU-TBLASTX	5	1	+15/−12	2736 s
WU-TBLASTX	5	1	BLOSUM62	2714 s
BLAT	5	1	+2/−1	61 s
BLAT	4	2	+2/−1	37 s

K: the size of the perfectly matching as a seed for an alignment
N: the number of hits in a gapless 100-aa window required to trigger a detailed alignment.
Matrix: column describes the match/mismatch scores or the substitution score matrix used.

Comparison of NGSs vs. traditional technology

Platforms	Sanger	454	Solexa	SOLiD
Read Length (bps)	650-1100	150-250	35-150	25-50
Capacity (reads/run)	96	400,000	200,000,000	2,000,000,000
Error Rate	10^-3	<10^-2	~10^-2	~10^-2
Cost ($/Mbp)	5000	~5	~0.6	~0.2
Time/run	~3h	~7h	2-10d	3-14d
Throughput	100Kb	~1Gb	~600Gb	100-300Gb



How to map billions of short reads onto genomes

Cole Trapnell & Steven L Salzberg

Mapping the vast quantities of short sequence fragments produced by next-generation sequencing platforms is a challenge. What programs are available and how do they work?

A new generation of DNA sequencers that can rapidly and inexpensively sequence billions of bases is transforming genomic science. These new machines are quickly becoming the technology of choice for whole-genome sequencing and for a variety of sequencing-based assays, including gene expression, DNA-protein interaction, human resequencing and RNA splicing studies[1–3]. For example, the RNA-Seq protocol, in which processed mRNA is converted to cDNA and sequenced, is enabling the identification of previously unknown genes and alternative splice variants; the ChIP-Seq approach, which sequences immunoprecipitated DNA fragments bound to proteins, is revealing networks of interactions between transcription factors and DNA regulatory elements[4]; and the whole-genome sequencing of tumor cells is uncovering previously unidentified cancer-

Table 1 A selection of short-read analysis software

Program	Website	Open source?	Handles ABI color space?	Maximum read length
Bowtie	http://bowtie.cbcb.umd.edu	Yes	No	None
BWA	http://maq.sourceforge.net/bwa-man.shtml	Yes	Yes	None
Maq	http://maq.sourceforge.net	Yes	Yes	127
Mosaik	http://bioinformatics.bc.edu/marthlab/Mosaik	No	Yes	None
Novoalign	http://www.novocraft.com	No	No	None
SOAP2	http://soap.genomics.org.cn	No	No	60
ZOOM	http://www.bioinfor.com	No	Yes	240

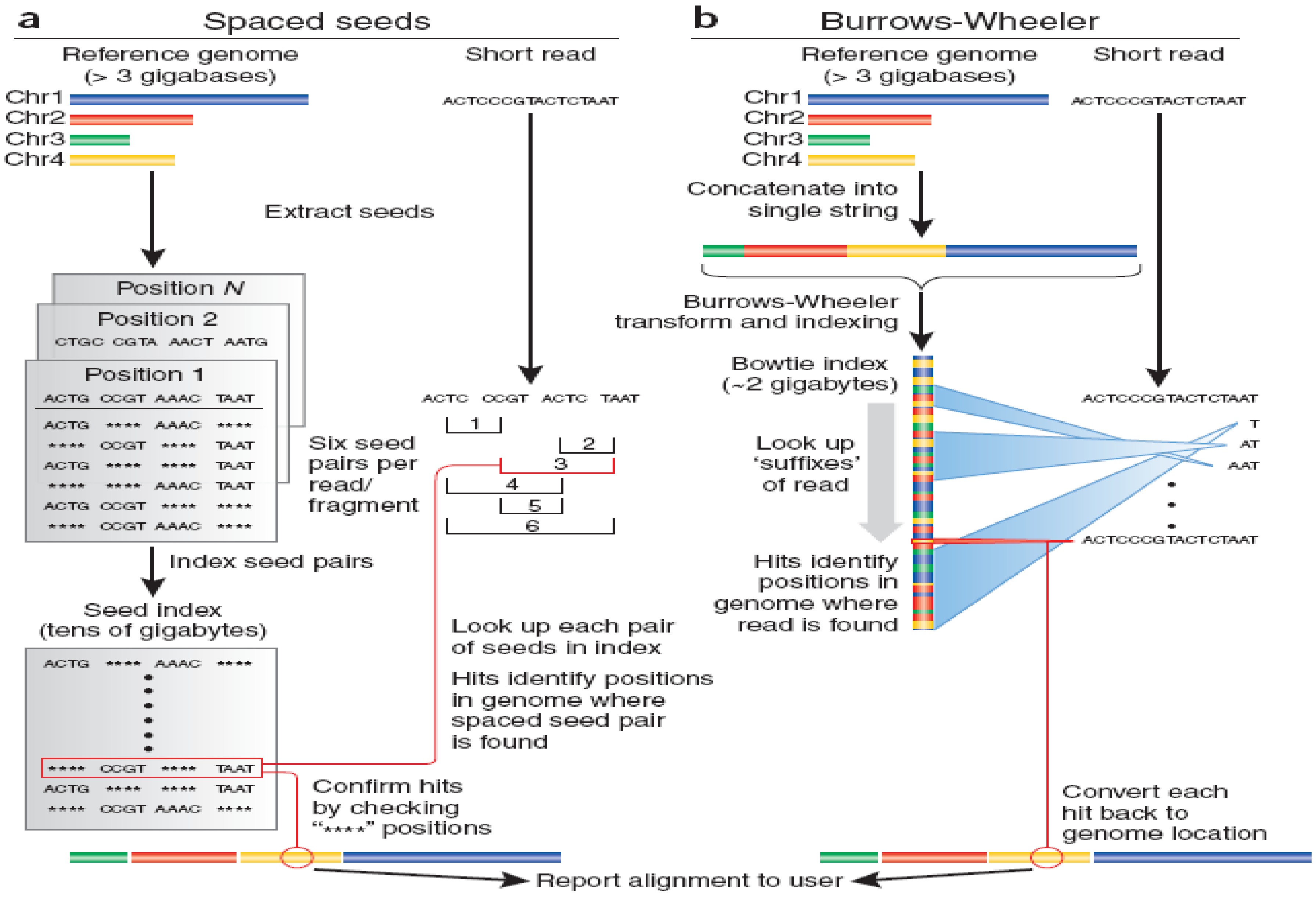
In this case, to make sense of the reads, their positions within the reference sequence must be determined. This process is known as aligning or 'mapping' the read to the reference. In one version of the mapping problem, reads must be aligned without allowing large gaps in

to understand why the mapping problems are computationally difficult, which difficulties have been overcome and what challenges and opportunities remain.

Challenges of mapping short reads

Latest progress of sequence alignment/mapping

- Aligning (mapping) billions of short reads
  - Bowtie
  - SOAP
  - BWA
  - Tophat



Algorithms (a) based on spaced-seed indexing; (b) based on Burrows-Wheeler transform