Sequence alignment ... in parallel!!

Diego Puppin

Oggi parliamo di...

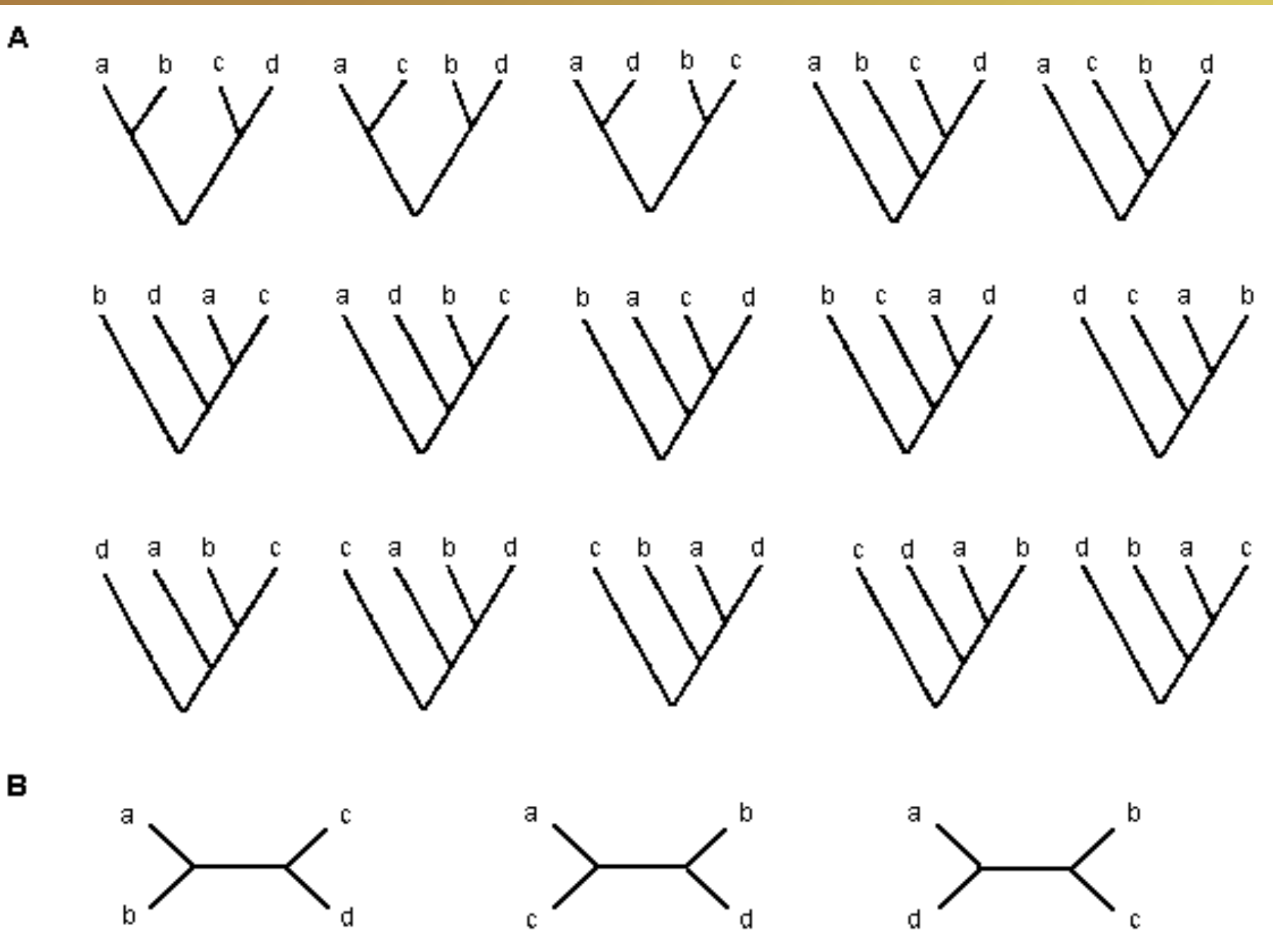
- Introduzione: Allineamento di sequenze
  - Algoritmi basati su Programmazione Dinamica
  - Algoritmo Smith-Waterman
- Micro-parallelismo
  - Parallelismo intra-sequenza / inter-sequenza
- Soluzioni parallele / distribuite
  - Multithreaded / Memoria Condivisa
- Conclusioni

OUR FEATURE PRESENTATION

- Matrici di profilo
- Scheduling a fronte d'onda
- Funzione *max* senza salti
- Approccio ottimistico ai gap
- Interi calcolati come floating point
- Scheduling multi-threaded interleaved

L'allineamento di sequenze

- Strumento essenziale in bio-informatica
  - Alcune scoperte sono risultato di allineamenti
- Al cuore di molti problemi
  - Ricostruzione di alberi filo-genetici
  - Scoperta di geni
  - Somiglianza tra proteine


A
a b c d
a c b d
a d b c
a b c d
a c b d
b d a c
a d b c
b a c d
b c a d
d c a b
d a b c
c a b d
c b a d
c d a b
d b a c
B
a c b d
a b c d
a b d c

Applicazioni

- Comparative analysis of gene clusters
- Looking for missing genes
- Studying horizontal gene transfer
- Studying evolution of metabolism
- Designing new organisms

Crescente disponibilita' di dati

- Trascritto il DNA di interi organismi
  - Incluso l'uomo!!
- Archivi come GenBank/EMBL/DDDBJ raddoppiano ogni 15 mesi
  - Alcuni liberamente accessibili
- Sfidano la legge di Moore!
  - Servono migliori algoritmi

... i dati!

- Major Seq. Repositories (7)
- Comparative Genomics (7)
- Gene Expression (19)
- Gene ID & Structure (31)
- Genetic & Physical Maps (9)
- Genomic (49)
- Intermolecular Interactions (5)
- Metabolic Pathways & Cellular Regulation (12)
- Mutation (34)
- Pathology (8)
- Protein (51)
- Protein Sequence Motifs (18)
- Proteome Resources (8)
- Retrieval Systems & DB Structure (3)
- RNA Sequences (26)
- Structure (32)
- Transgenics (2)
- Varied Biomedical (18)

Numero di basi di dati per settore, da:

Baxevanis, A.D. 2002. *Nucleic Acids Research* 30: 1-12.

Algoritmi esatti di allineamento

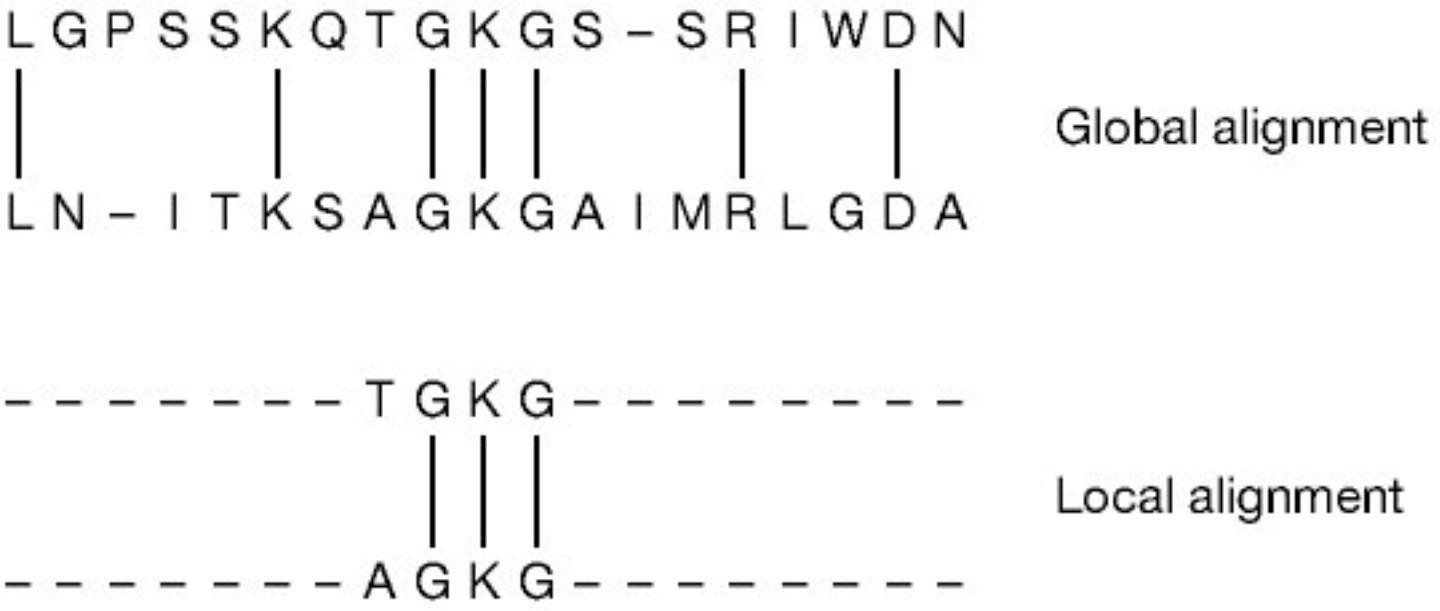
- Allineamento = determinare una sequenza (la piu' probabile) di sostituzioni, inserimenti ed eliminazioni che ha causato la trasformazione di una sequenza in un'altra
- Indicare per ogni base in A, quale le corrisponde in B (oppure GAP)
- Ogni operazione ha un costo (~probabilita') codificate nell'esperienza del ricercatore

Modelli di costo

- Incorporano in maniera sistematica l'esperienza:
  - Matrice di sostituzione (x, y)
    - Standard: BLOSUM, PAM
  - Penalita' di gap
    - Vari modelli (1-1, affine)
- L'algoritmo e' independente dal modello di costo
  - Ma cambia la qualita' del risultato

Allineamento

- L'algoritmo cerca l'allineamento che da' il minimo costo di trasformazione



Algoritmo di programmazione dinamica

- Matrice di somiglianza
  - Per (i,j) da' il costo del miglior allineamento tra A[1..i-1] e B[1..j-1]
- Programmazione dinamica a partire da (0, 0)
- Massimo tra tre opzioni:
  - Allinea A[i] con B[j] (costo sostituzione)
  - Allinea A[i] con GAP in B
  - Allinea B[j] con GAP in A

Programmazione dinamica (2)

$$SM[i,j] = max = \begin{cases} SM[i,j-1] + gap\_cost \\ SM[i-1,j-1] + matching\_cost(A[i], B[j]) \\ SM[i-1,j] + gap\_cost \end{cases}$$

- Estendibile a piu' dimensioni
  - ...ed ottimizzato

Storia

- 1970, presentato da Needleman-Wunsch (ricerca globale)
- 1981, Smith-Waterman per ricerca locale
- 1982, ottimizzazioni di Gotoh

Altre soluzioni

- Algoritmi basati su euristiche
  - FASTA, BLAST, velocita' 40x
  - Soluzioni approx. a volte imprecise
  - Non riconoscono allineamenti deboli
- Hardware ad-hoc
  - Molto costoso, poco diffuso

Micro-Parallelismo

Micro-Parallelismo

- Idea: sfruttare il parallelismo all'interno di un processore (nativo o simulato)
- Necessario codice di backup a volte!

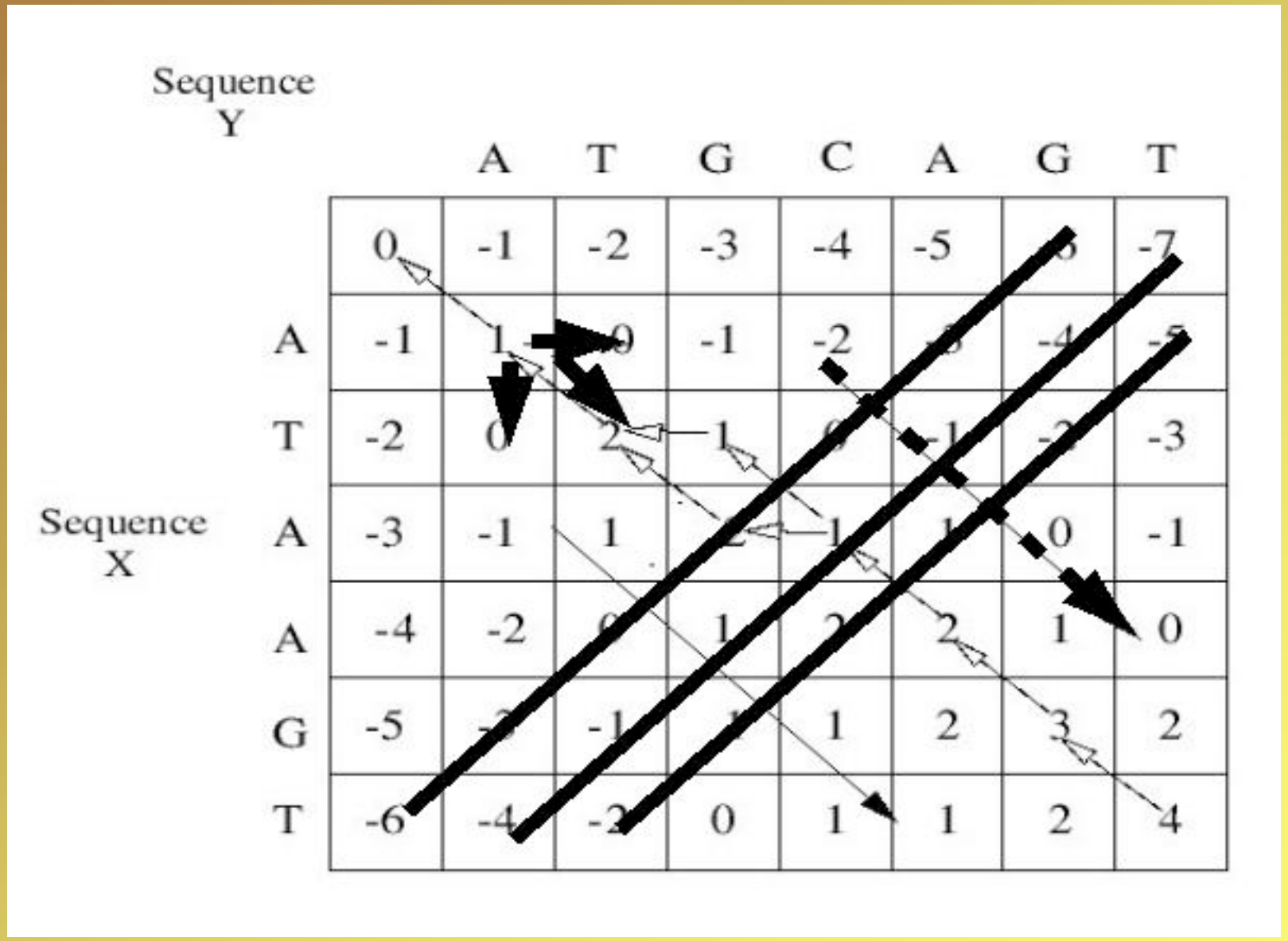
MSB	LSB
5	7
PLUS	
3	9
=	
8	16

MSB LSB
327,687
PLUS
196,617
=
524,304

$$(x, y) \rightarrow x \times 2^{16} + y$$

Ordinamento a fronte d'onda

- Wavefront scheduling
- Riorganizza il calcolo mettendo in evidenza dati independenti
- Ristruttura il ciclo
  - Migliora l'uso della cache
- Permette di ridurre la richiesta di memoria
  - Due sole righe se non dobbiamo ricostruire l'allineamento


Sequence Y
A T G C A G T
Sequence X
A T A A G T
0 -1 -2 -3 -4 -5 -6 -7
-1 1 0 -1 -2 -3 -4 -5
-2 0 2 1 0 -1 -2 -3
-3 -1 1 2 1 1 0 -1
-4 -2 0 1 2 2 1 0
-5 -3 -1 0 1 2 3 2
-6 -4 -2 0 1 1 2 4

Parallelismo Inter-Sequenza

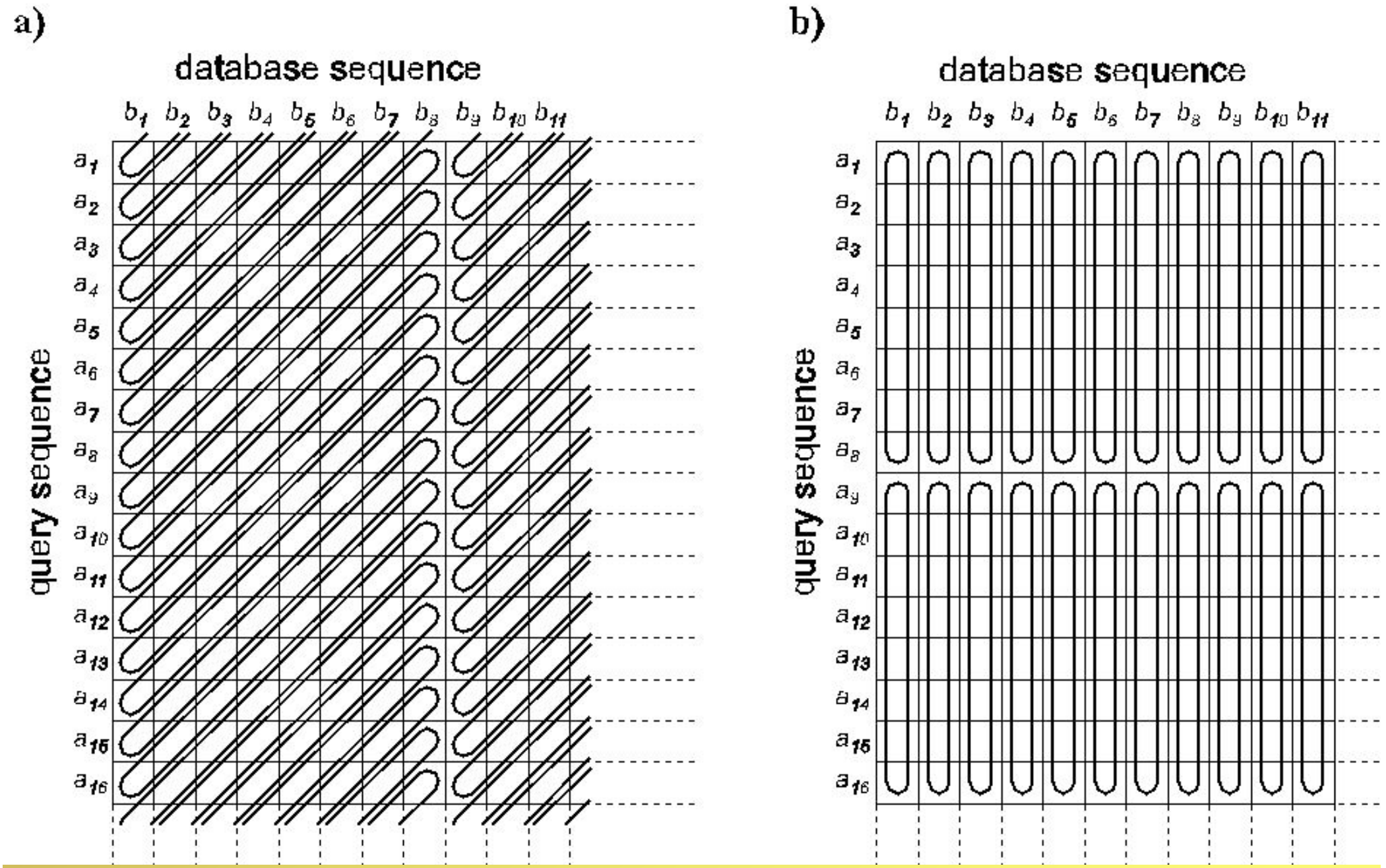
- Alpen et al.
- Eseguono in parallelo 2 - 4 allineamenti
- IBM Power2
  - 2 allineamenti, 20% piu' veloce
- Intel i860
  - istruzioni grafiche su registri lunghi,
  - fornisce “min”
  - 4 allineamenti, 6.41 speed-up

Interi -> Virgola mobile

- Su IBM Power2, usano virgola mobile (veloce) per implementare l'algoritmo
  - Piccoli problemi di arrotondamento
  - 3.46 speed-up

Micro-Parallelismo su un Pentium!!

- SWMMX, di Rognes e Seeberg
- Usano MMX di Pentium III
- Varie ottimizzazioni:
  - **Approccio ottimistico al gap** (SWAT2)
  - Uso di partizioni verticali
  - 8 caselle calcolate insieme
  - **Matrice di profilo**
  - Ottimizzazioni in genere


a)
database sequence
b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11
query sequence
a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16
b)
database sequence
b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11
query sequence
a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16

Matrice di profilo

- Invece di consultare una tabella
  - cost (x, y)
- si accede una tabella
  - cost (posx, y)
- Il profilo si accede in sequenza (cache)
  - prefetching della colonna cost(pos x+1)
- E' riutilizzabile per fare analisi multiple

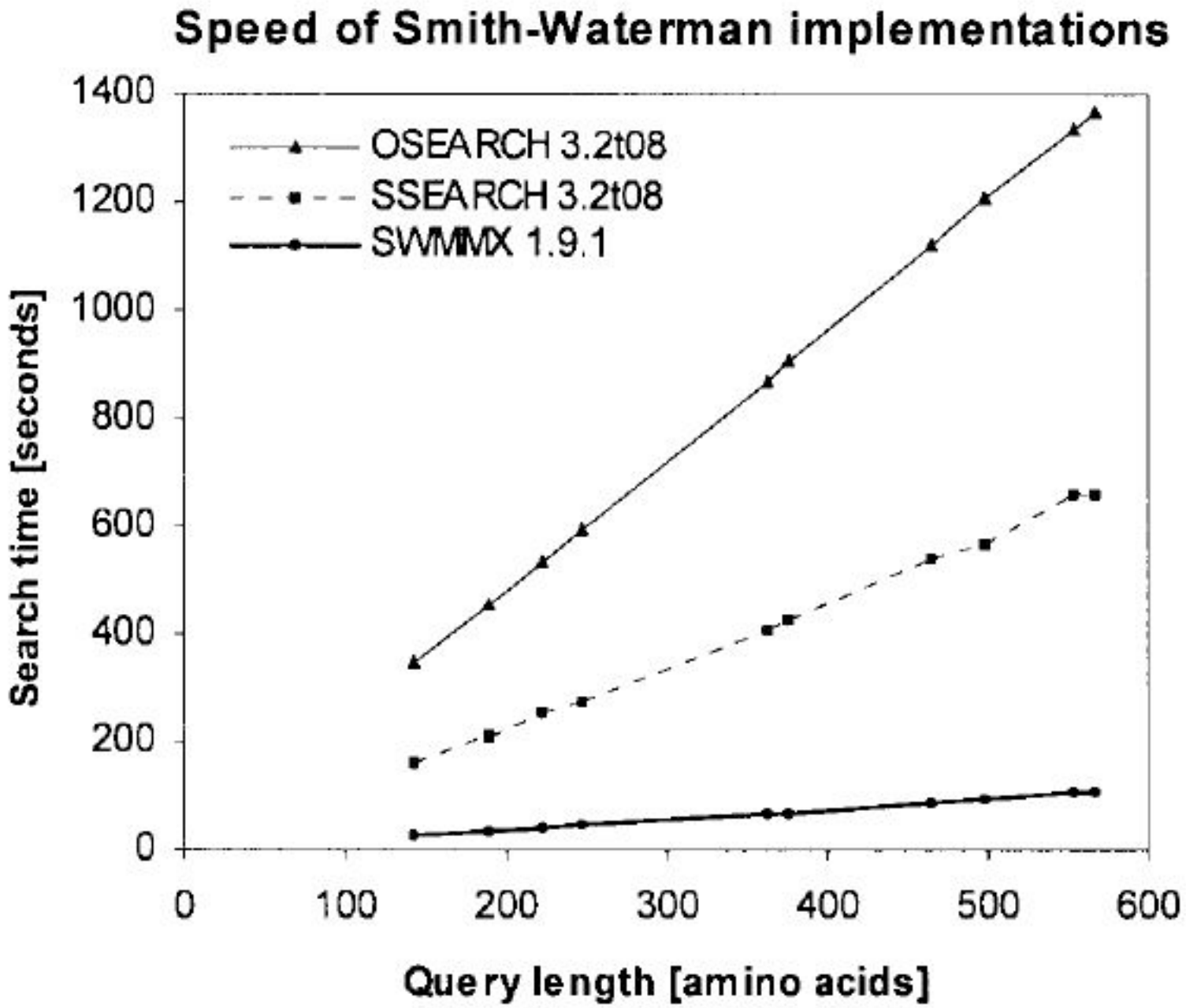
Approccio ott. al gap

- Se il costo del gap e' alto, i gap saranno rari
- Con buona probabilita', i valori N e W non contribuiscono al massimo valore
- Le righe sono indipendenti!
- **Serve codice di backup per verificare e correggere**

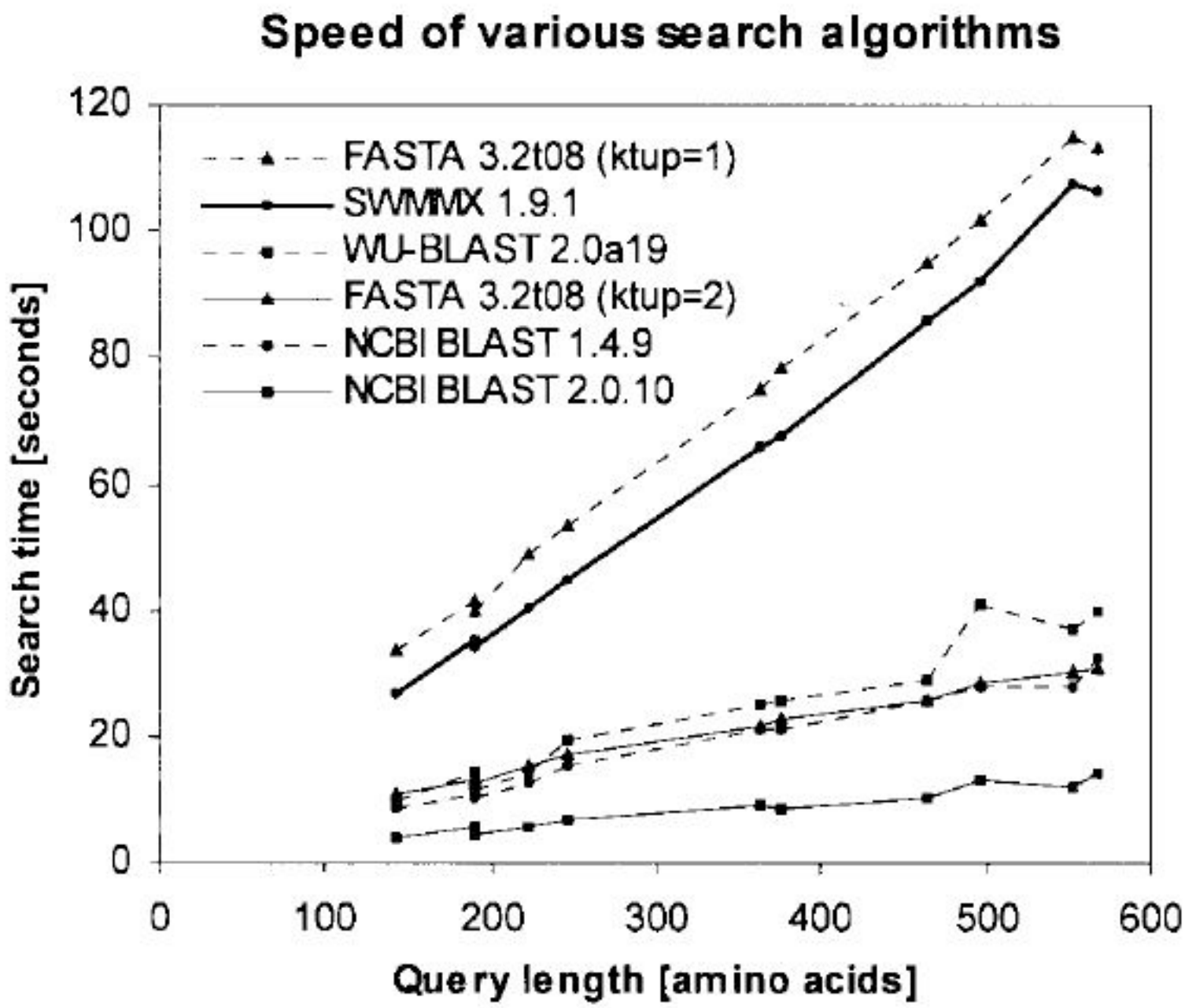
Risultati

- 13x rispetto SW non-ottimizzato
- 6x rispetto SW ottimizzato
- Confrontabile con FASTA/BLAST a massima precisione (ancora approssimati)

a)


Speed of Smith-Waterman implementations
OSEARCH 3.2t08
SSEARCH 3.2t08
SWIMMX 1.9.1
Search time [seconds]
Query length [amino acids]
0
200
400
600
800
1000
1200
1400
0
100
200
300
400
500
600

b)


Speed of various search algorithms
Search time [seconds]
Query length [amino acids]
FASTA 3.2t08 (ktup=1)
SWMMX 1.9.1
WU-BLAST 2.0a19
FASTA 3.2t08 (ktup=2)
NCBI BLAST 1.4.9
NCBI BLAST 2.0.10
0
20
40
60
80
100
120
0
100
200
300
400
500
600

Ogni trucco e' valido

- MAX(x,y) senza salto!!!!

```
MAX(x,y)
      x -= y
      t = (x >> (sizeof(x) – 1))
      x &= ~t
      x += y
```

MicroPar su Workstation

- Wozniak usa le istruzioni grafiche su ULTRA SPARC
  - riorganizza l'ordine del calcolo
  - esegue il calcolo su 8 righe alla volta
  - max senza salti!
  - 2x speed-up su un singolo processore
  - NON usa matrici di profilo, gap ottimistici etc.

Multi-threading e Memoria Condivisa

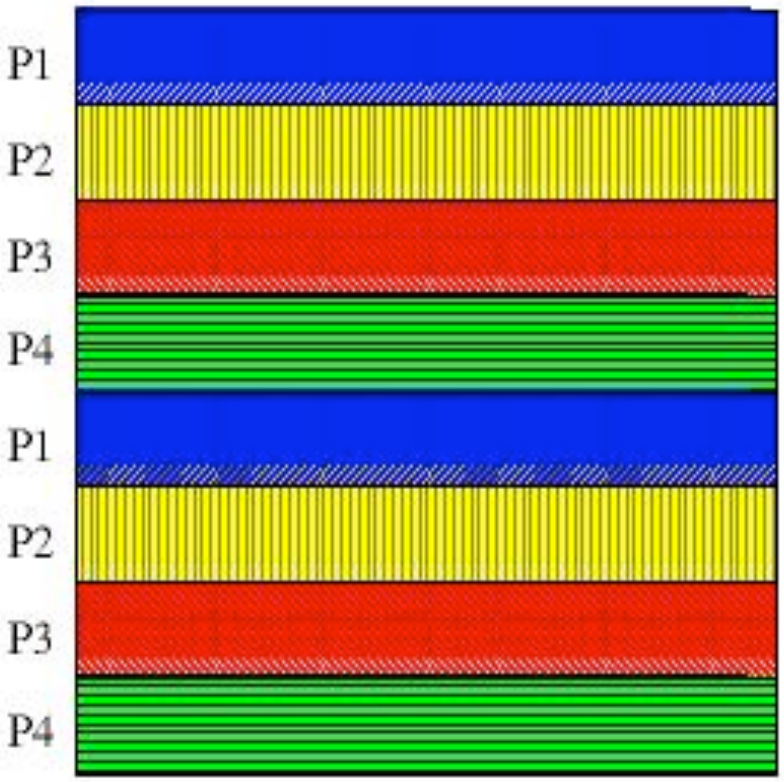
Soluzione multi-threaded

- Martins et al.
- similarity matrix (SM) e' divisa in blocchi
  - Riduce le comunicazioni
- assegnati a thread (ciclicamente)
  - Bilanciamento di carico
- EARTH system, implementato su MANNA, SP2, Sun SMP, Beowulf

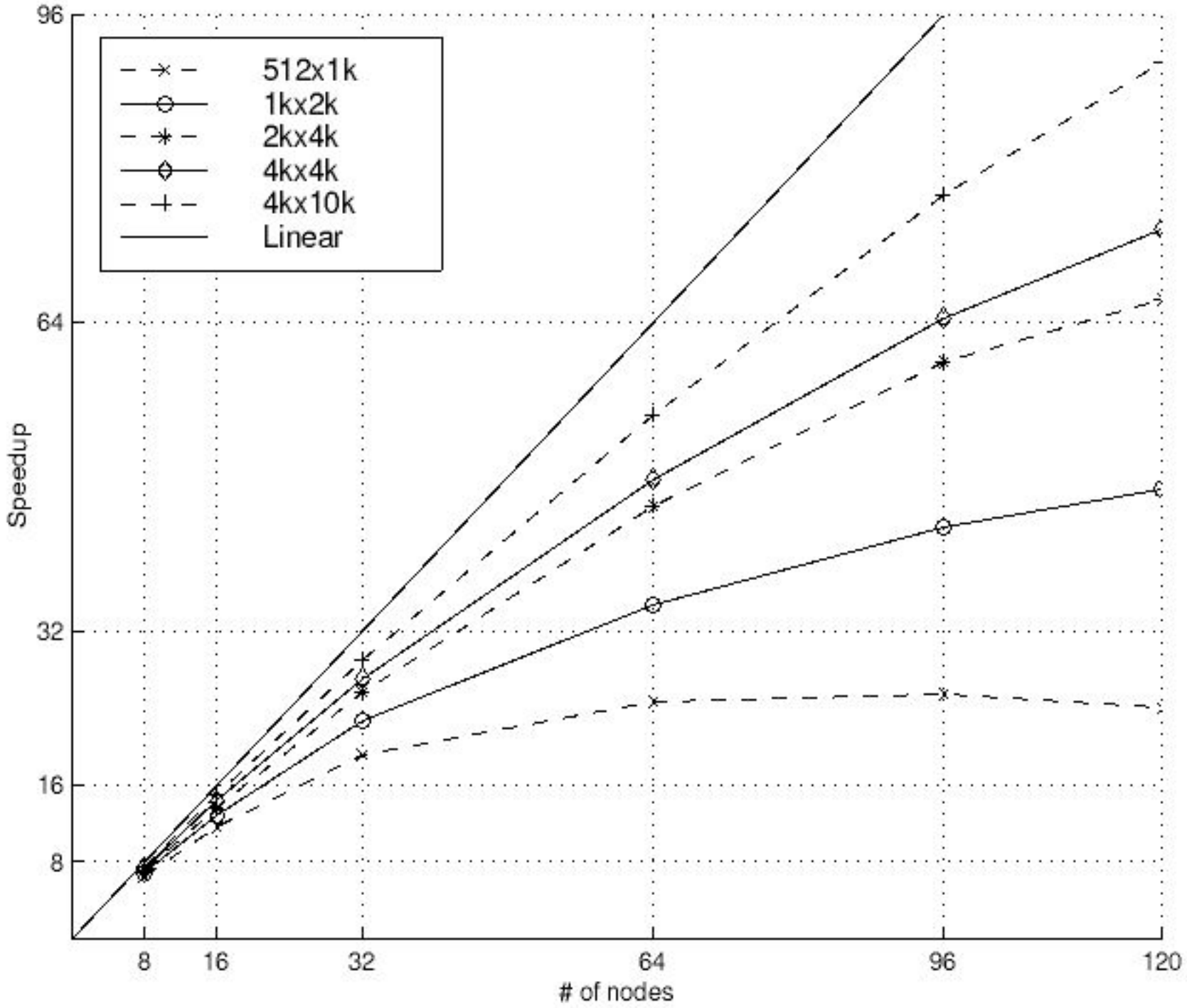
Distribuzione ciclica, a blocchi

P1	1	2	3	4
P2	5	6	7	8
P3	9	10	11	12
P4	13	14	15	16
P1	17	18	19	20
P2	21	22	23	24
P3	25	26	27	28
P4	29	30	31	32

(a) Block Division



(b) Processor Distribution


512x1k
1kx2k
2kx4k
4kx4k
4kx10k
Linear
Speedup
# of nodes
96
64
32
16
8
8
16
32
64
96
120

Soluzione su memoria condivisa (allineamento locale)

- Melo et al. usano DSM JIAJIA
- Usano ordinamento a ondate, bastano due righe di memoria
- Se il punteggio dell'allineamento aumenta, miglioro l'allineamento locale, altrimenti devo ricominciare

Risultati

- Speed-up di 4.58x su 8 macchine
- Efficienza simile a MT, ma modello piu' semplice e HW standard

Size	Serial Exec	2 proc Exec /Speedup	4 proc Exec /Speedup	8 proc Exec /Speedup
15K x 15K	296	283.18/1.04	202.18/1.46	181.29/1.63
50K x 50K	3461	2884.15/1.20	1669.53/2.07	1107.02/3.13
80K x 80K	7967	6094.19/1.31	3370.40/2.46	2162.82/3.68
150K x 150K	24107	19522.95/1.23	10377.89/2.32	5991.79/4.02
400K x 400K	175295	141840.98/1.23	72770.99/2.41	38206.84/4.58

Conclusioni

Conclusioni

- L'allineamento e' importante
- La quantita' di dati cresce piu' velocemente della velocita' dell'HW (crescita super-Moore!!!)

  => deve aumentare l'efficacia degli algoritmi
- Gli algoritmi approssimati spesso non sono sufficienti (soddisfacenti)

Conclusioni (2)

- Abbiamo mostrato come affrontare l'algoritmo “esatto”
  - Fare in parallo 2-4-8 allineamenti
  - Migliorare il singolo all. con matrice profilo, approccio ottimistico al gap, max senza salti...
  - Soluzioni multi-threaded o memoria condisiva
- Prestazioni confrontabili con gli alg. approssimati