Università degli studi di Pisa

Nicola Guido

PATTERNHUNTER: Faster and More Sensitive Homology Search

Seminario: Bioinformatica

a.a. 2008/2009

Contenuto della presentazione

- Introduzione
- Scenario
- PatternHunter
- Modello probabilistico semplice
- Risultati
- Conclusioni

Introduzione

Algoritmi per la ricerca di omologie nei db:

- per studi filogenetici
- per l'analisi di cambiamenti strutturali del genoma collegati a malattie
- ...

Algoritmi accurati tendono ad essere lenti.

Esistono tecniche di ricerca scalabili basate su euristiche che danno limitate garanzie sull'accuratezza risultato.

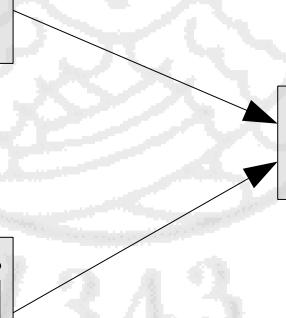
Introduzione

Paradismi esistenti

Obiettivo

Lenti, molto accurati

Veloci, meno accurati



Veloci, molto accurati

Contenuto della presentazione

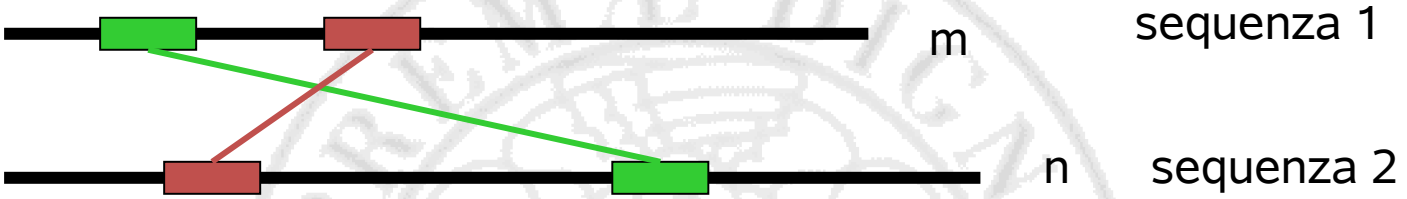
- Introduzione
- Scenario
- PatternHunter
- Modello probabilistico semplice
- Risultati
- Conclusioni

Scenario: Paradigmi esistenti

Algoritmo di Smith-Waterman.

Date due sequenze lunghe rispettivamente m ed n. La complessità in tempo e spazio è O(mn).

Scenario: Paradigmi esistenti



Vengono confrontate tutte le basi della prima sequenza con tutte le basi della seconda.

Sequenza Genoma Umano = $3x10^9$
Sequenza Genoma Topo = $3x10^9$

Umano vs Topo = $9x10^{18}$

Scenario: Paradigmi esistenti

Paradigmi esistenti più veloci sono basati su euristiche.

1) approccio basato sul concetto di 'seed'

2) approccio basato sui 'suffix tree'

Scenario: approccio 1

Blast family – FASTA – SIM – SENSEI

In questi programmi si cercano 'seed' nella sequenza target, che vengono successivamente estesi.

- Aumenta la dimensione del seed da ricercare, diminuisce l'accuratezza della ricerca.
- Diminuisce la dimensione del seed da ricercare, rallenta la computazione.

Scenario: approccio 2

MUMmer – REPuter – QUASAR

La ricerca si basa sugli alberi di suffissi. La struttura dati evidenzia la sequenza di basi per facilitare la ricerca di sottosequenze

- I suffix trees sono utilizzati per il match esatto di stringhe.
- La struttura dati richiede per la memorizzazione una quantità di spazio considerevole.

Scenario: ciò che verrà presentato

PATTERN HUNTER: basato sull'approccio seed

- Programma scritto in Java.

- Nuovo modello di 'seed' per aumentare l'accuratezza della ricerca

Contenuto della presentazione

- Introduzione
- Scenario
- PatternHunter
- Modello probabilistico semplice
- Risultati
- Conclusioni

PatternHunter : Terminologia

Seed = piccolo “segment pair” tra due sequenze

Segment = sottostringa di una sequenza.

Segment pair = Date due sequenze, è una coppia di sottostringhe appartenenti ognuna ad una sequenza.

PATTERNHUNTER vs BLAST

BLAST = cerca seed di k=11 lettere consecutive.

Questi seed sono poi estesi in entrambe le direzioni, senza gap, fino al massimo possibile score ammesso per l'estensione del seed.

Avere K lettere consecutive limita l'accuratezza della ricerca

PATTERNHUNTER vs BLAST

Utilizzando un seed di k=11 lettere consecutive non trovo hits.

```
GAGTACTCAACACCAACATTAGTGGGCAATGGAAAAT
||  |||||||||| |||||| | ||||||   ||||||
GAATACTCAACAGCAACATCAATGGGCAGCAGAAAAT
```

Dilemma:

- Seed corti: più accurati, meno velocità
- Seed lunghi: meno accurati, più velocità

PatternHunter : spaced seed

Come aumentare contemporaneamente velocità e accuratezza?

PatternHunter = introduce seed di k lettere non consecutive (spaced seed).

Questo semplice cambiamento ha degli effetti sorprendenti sull'accuratezza della ricerca.

PatternHunter : spaced seed

MODELLO(stringa binaria) k = peso

111010010100110111 ∑ 1s = 11

dove:

1 = richiesto match

0 = don't care

PatternHunter : spaced seed

Supponiamo di avere due sequenze simili, s e t, di lunghezza L.

Definiamo una stringa binaria, *v = v (s,t)* tale che

v[i] = 1 sse s[i] = t[i]
v[i] = 0 altrimenti

Si dice che un modello m copre una stringa u sse m[i] ≤ u [i].

PatternHunter : Esempio di seed

MODELLO

k = peso

111010010100110111

∑ 1s = 11

```
GAGTACTCAACACCAACATTAGTGGCAATGGAAAAT…
||  |||||||||  |||||  ||  |||||    ||||||
GAATACTCAACAGCAACACTAATGGCAGCAGAAAAT…
        1110100101001101111
```

PatternHunter : Esempio di seed

MODELLO

k = peso

1110100101001101111

∑ 1s = 11

```
GAGTACTCAACACCAACATTAGTGGCAATGGAAAAT…
||  |||||||||  |||||  ||  |||||    ||||||
GAATACTCAACAGCAACACTAATGGCAGCAGAAAAT…
          1110100101001101111
```

Come già visto, seed 11111111111 non rileva nessuna omologia tra le due sequenze.
Spaced seed forniscono risultati più accurati.

Contenuto della presentazione

- Introduzione
- Scenario
- Pattern Hunter
- Modello probabilistico semplice
- Osservazioni

Modello Probabilistico semplice

Caratteristiche dei seed a confronto

```
TTGACCTCACC?
|||||||||||?
TTGACCTCACC?
11111111111
 11111111111
```

```
CAA?A??A?C??TA?TGG?
|||?|??|?|??||?|||?
CAA?A??A?C??TA?TGG?
11101001010011O111
 11101001010011O111
```

BLAST: eventi di match dipendenti

PH: eventi di match indipendenti

Modello Probabilistico semplice

PatternHunter ha poca sovrapposizione

```
1110100101001101111
 1110100101001101111
  1110100101001101111
   1110100101001101111
    1110100101001101111
     1110100101001101111
      1110100101001101111
     1110100101001101111
        ......
```

Modello probabilistico semplice

Consideriamo una regione di lunghezza L e consideriamo gli eventi, di avere un match in differenti posizioni, indipendenti

Formalmente, la regione può essere vista come una sequenza di L variabiali casuali di Bernulli
X_0 X_1 X_2 ..... X_{L-1}.

dove

$Pr(X_i = 1) = p$ $\quad\quad$ $i = 0.....N-1$

Modello Probabilistico semplice

Il numero di hits atteso è

$$(L - M + 1)\, p^{K}$$

dove:

L = la lunghezza della regione
M = la lunghezza del modello
p = la probabilità di avere un match
K = peso del modello

Modello Probabilistico semplice

Con L=64, p=0,70 e k=11

La probabilità di trovare almeno un hits è

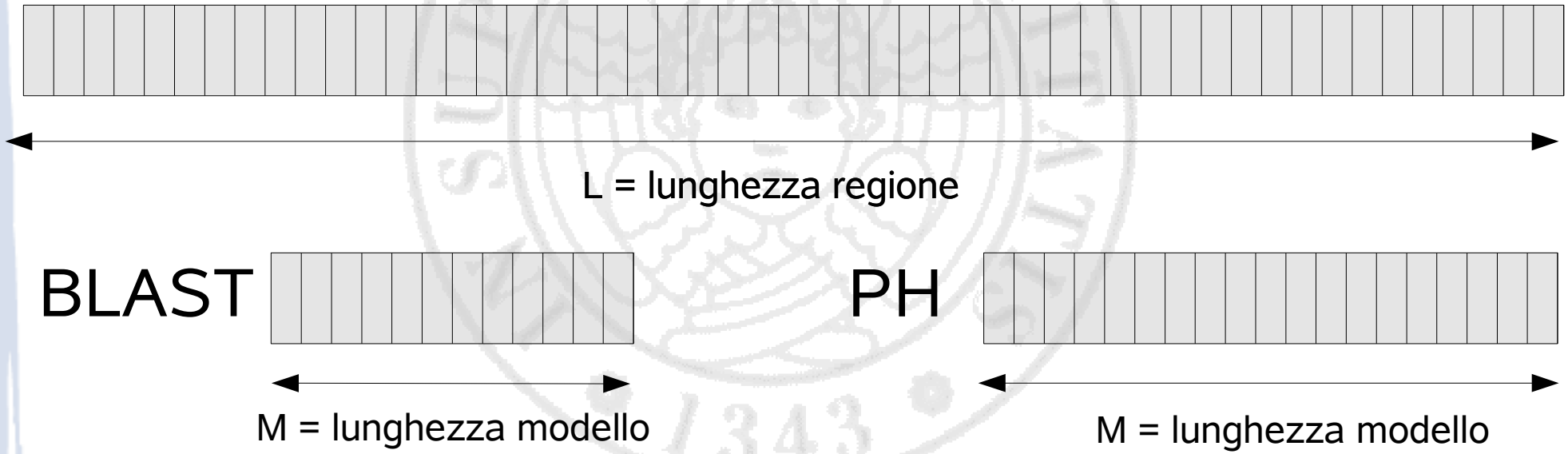
Blast's consecutive model --M=11--> 30,0%
PH's nonconsecutive model --M=28--> 46,6%

Il numero atteso di hits è

Blast's consecutive model --M=11--> 1,09
PH's nonconsecutive model --M=28--> 0,93

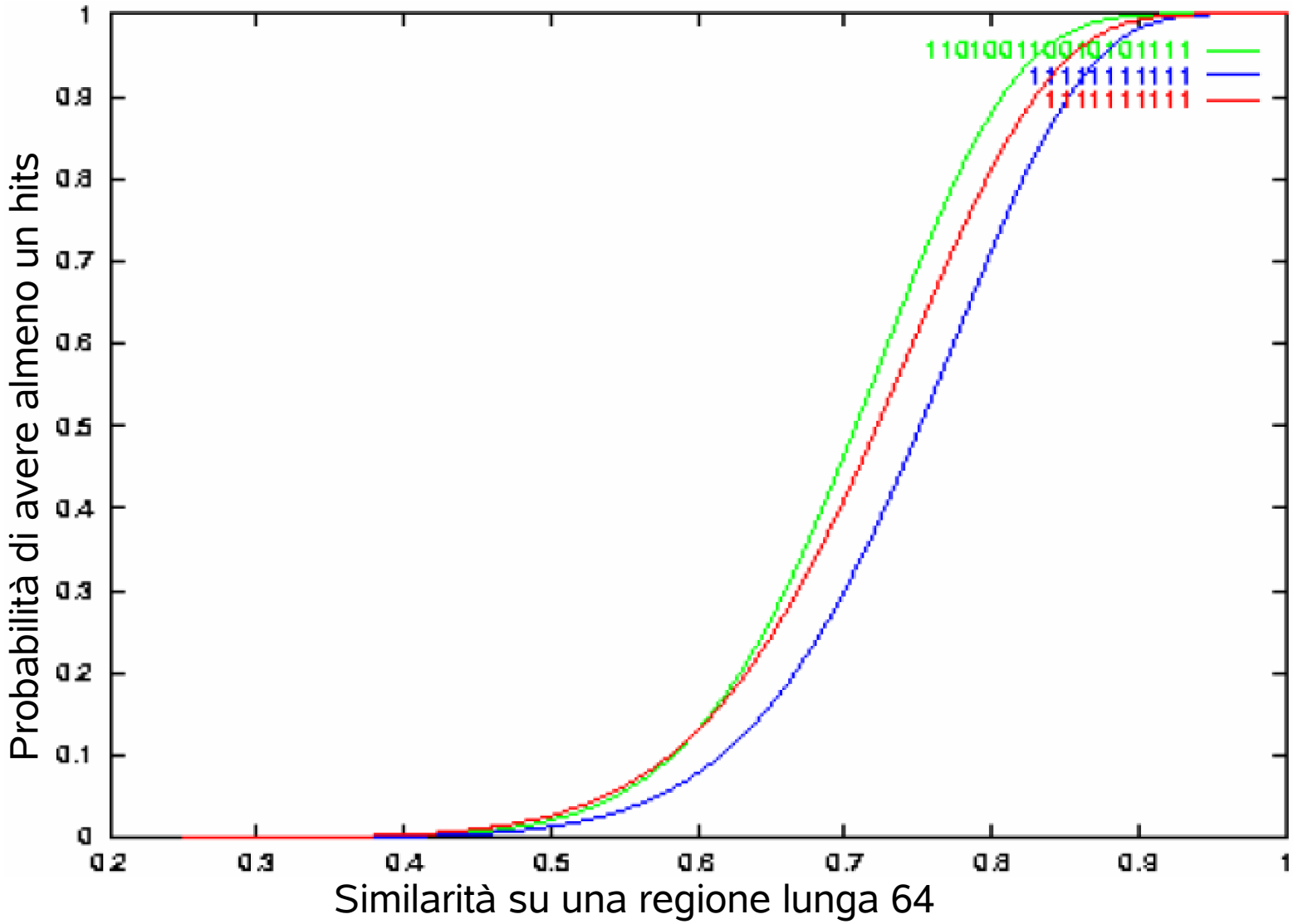
Modello Probabilistico semplice

Posso effettuare (L – M + 1) shift diversi



Con BLAST posso effettuare più shift rispetto PH

Modello Probabilistico semplice



Modello Probabilistico semplice

Theorema.
Trovare il seed ottimo, data la lunghezza e il peso, è un problema np-arduo

Idea di dimostrazione
(viene effettuata una riduzione)

$$\text{SeedOttimo} <_{\text{logspace}} \text{3-SAT}$$

Contenuto della presentazione

- Introduzione
- Scenario
- Pattern Hunter
- Modello Probabilistico semplice
- Risultati
- Conclusioni

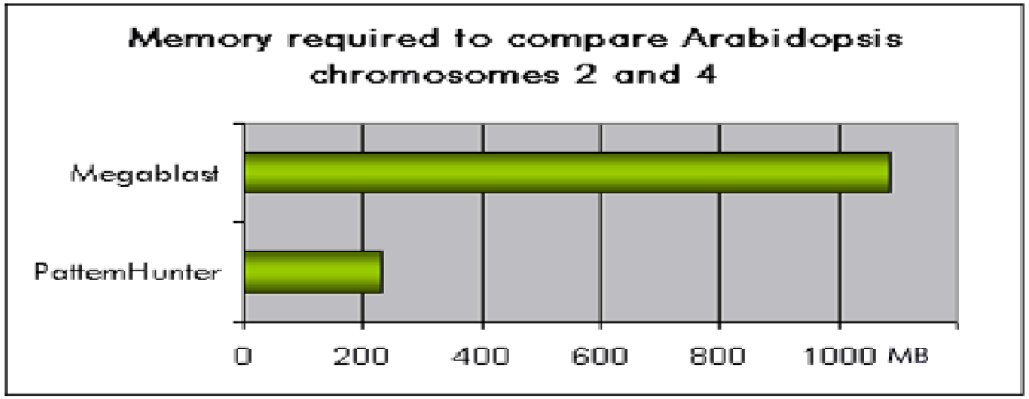
Risultati

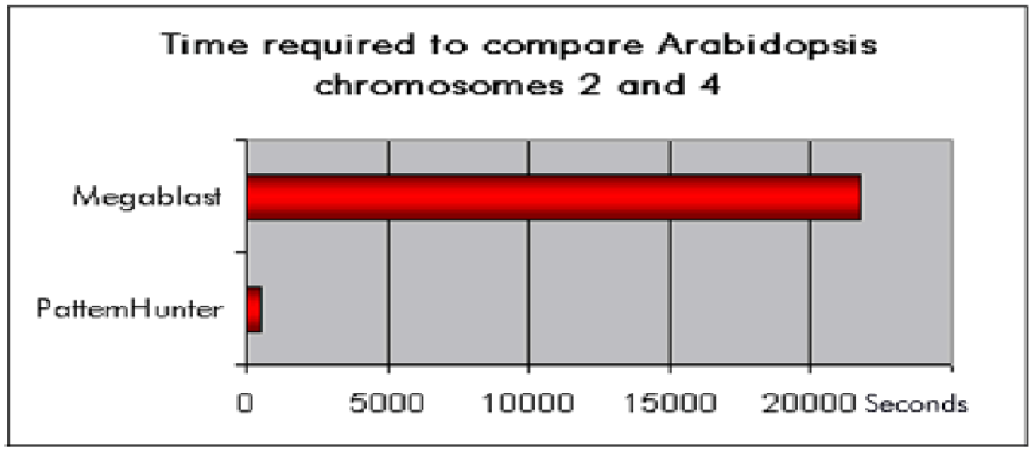
Consideriamo la ricerca di omologie tra Arabidopsis Chromosome 2 (20Mb) e Chromosome 4 (18Mb)

Lunghezza sequenza	Blastn	PatternHunter
20Mb vs 18Mb	Out of memory	13 min

Considerando Megablast...

Risultati


Memory required to compare Arabidopsis chromosomes 2 and 4
Megablast
PatternHunter
0
200
400
600
800
1000 MB


Time required to compare Arabidopsis chromosomes 2 and 4
Megablast
PatternHunter
0
5000
10000
15000
20000 Seconds

Risultati

Sequenza genoma UMANO vs TOPO

Pentium IV 3GHz Linux PC

SSearch	Blastn	PatternHunter				
20 Days	575 s	Seeds	1	2	4	8
		General	242 s	381 s	647 s	1027 s
		Specific	214 s	357 s	575 s	996 s

Contenuto della presentazione

- Introduzione
- Scenario
- Pattern Hunter
- Modello Probabilistico semplice
- Risultati
- Conclusioni

Conclusioni

PatternHunter è un algoritmo basato su euristiche, che presenta un'accuratezza paragonabile all'algoritmo di Smith-Waterman, ma veloce tanto quanto Blast.

Riferimenti

[1] PatternHunter: faster and more sensitive homology search.

[2] Patternhunter II: highly sensitive and fast homology search.

[3] Optimal spaced seed for homologous coding regions

[4] A Tutoria of recent developments in the seeding of local alignment.