Faster and More Sensitive Homology Search

Ming Li
Visiting Professor
City University of Hong Kong

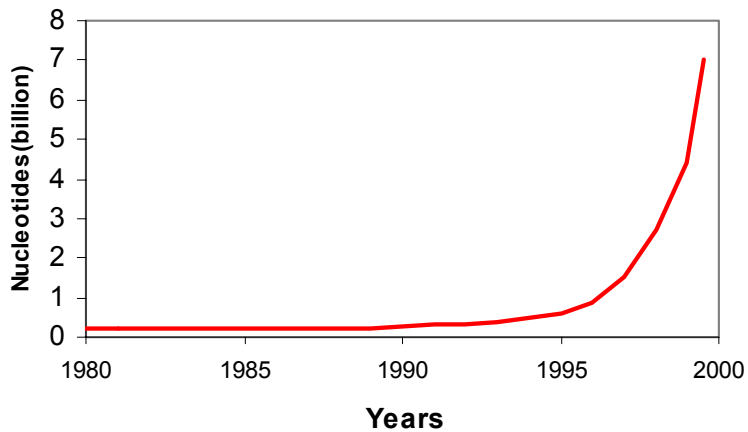
Canada Research Chair in Bioinformatics
Professor
University of Waterloo

Joint work with Bin Ma, John Tromp

I will present one simple idea
which is directly benefiting thousands of people daily

A gigantic gold mine

- The trend of genetic data growth

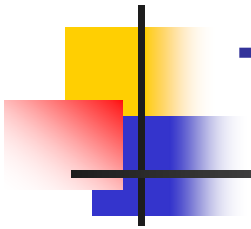


30 billion
in year 2005

- 400 Eukaryote genome projects underway
- GenBank doubles every 18 months
- Comparative genomics → all-against-all search

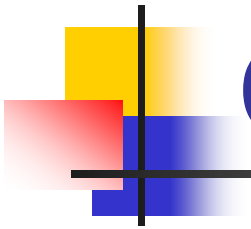
Comparing to internet search

- Internet search
  - Size limit: 5 billion people x homepage size
  - Supercomputing power used: ½ million CPU-hours/day
  - Query frequency: Google --- 112 million/day
  - Query type: exact keyword search --- easy to do
- Homology search
  - Size limit: 5 billion people x 3 billion basepairs + millions of species x billion bases
  - 10% (?) of world's supercomputing power
  - Query frequency: NCBI BLAST -- 150,000/day, 15% increase/month
  - Query type: approximate search --- topics today



Tremendous Cost

- Bioinformatics Companies living on BLAST:
  - Paracel (Celera)
  - TimeLogic
  - TurboGenomics (TurboWorx)
- NSF, NIH, pharmaceuticals *proudly* support many supercomputing centers for homology search
- However: hardware become obsolete in 2-3 years. Software solution is indispensable.



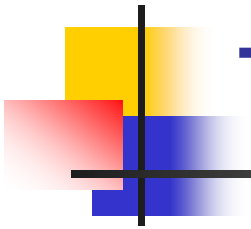
Outline

- **What is homology search**
- A simple (but profound) idea
- Its theory
- Its practical success

What is homology search

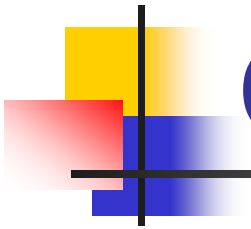
- Given two DNA sequences, find all local similar regions, using "edit distance" (match=1, mismatch=-1, gapopen=-5, gapext=-1).
- Example. Input:
  - E. coli genome: 5 million base pairs
  - H. influenza genome: 1.8 million base pairs

  Output: all local alignments.



Time Flies

- Dynamic programming (1970-1980)
  - Human vs mouse genomes: 10^4 CPU-years
- BLAST, FASTA heuristics (1980-1990)
  - Human vs mouse genomes: 19 CPU-years
  - BLAST paper was referenced 100000 times
- PatternHunter
  - Human vs mouse genomes: 20 CPU-days

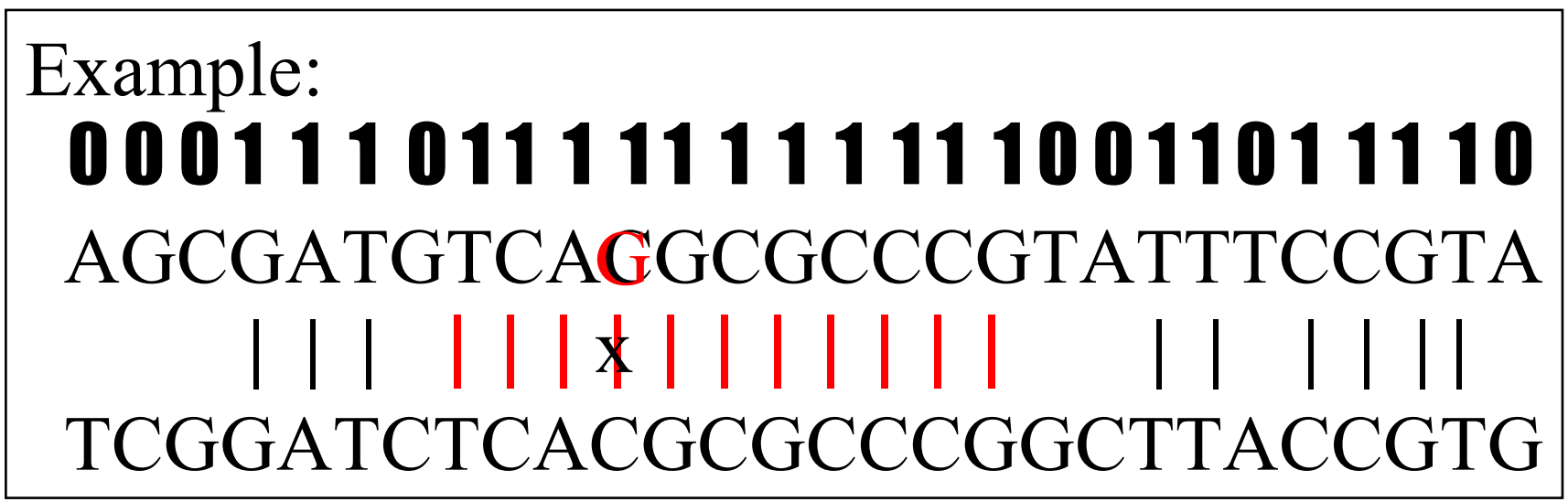


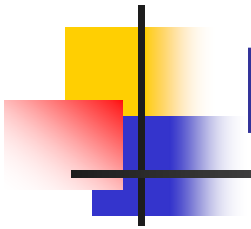
Outline

- What is homology search
- **A simple (but profound) idea**
- Its theory
- Its practical success

BLAST Algorithm & Example

- Find seeded matches of 11 base pairs
- Extend each match to right and left, until the scores drop too much, to form an alignment
- Report all local alignments





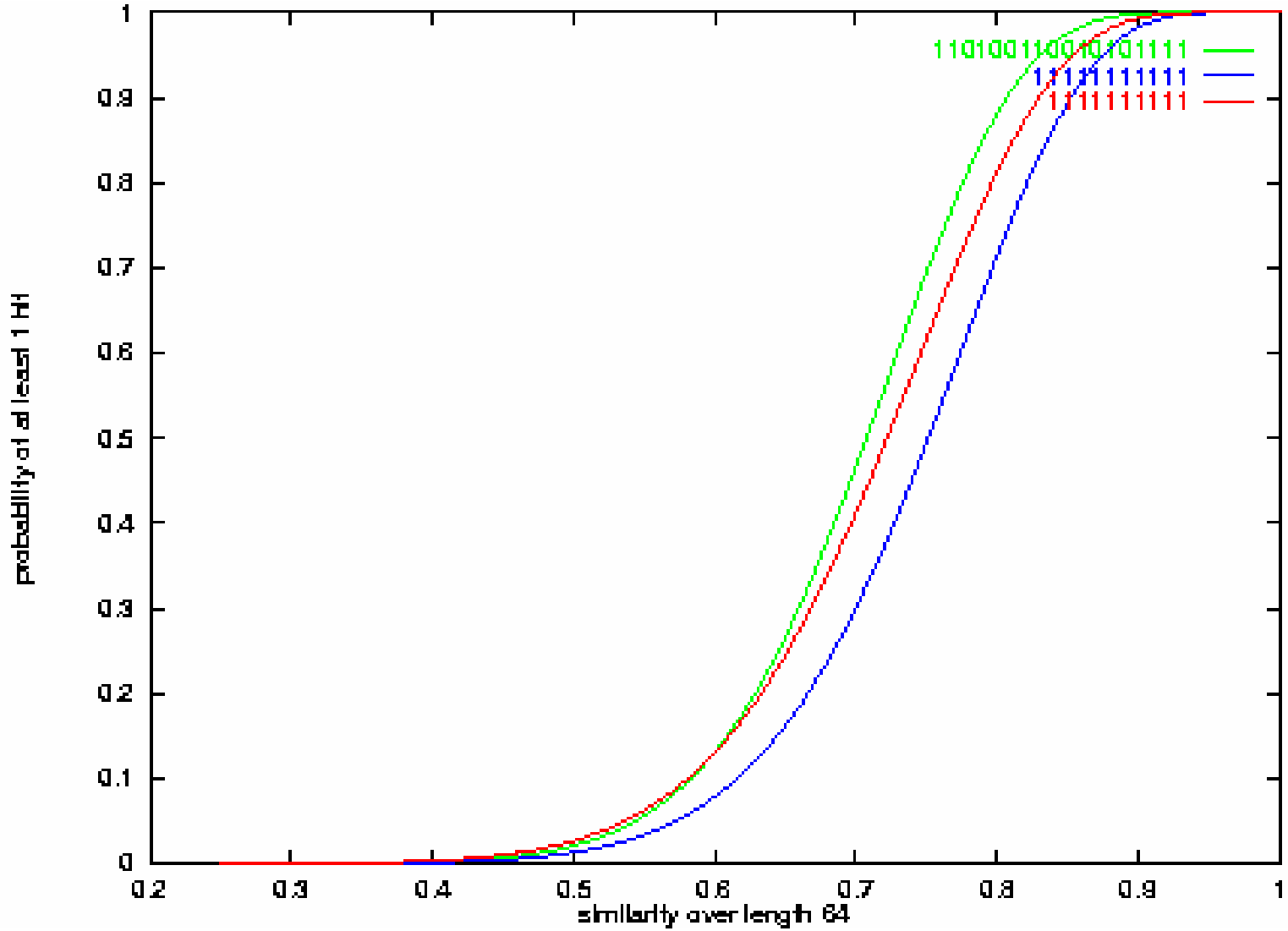
BLAST Dilemma:

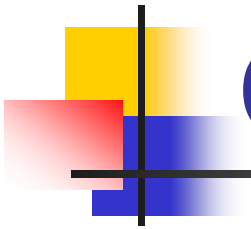
- If you want to speed up, have to use a longer seed. However, we now face a dilemma:
  - increasing seed size speeds up, but loses sensitivity;
  - decreasing seed size gains sensitivity, but loses speed.
- How do we increase sensitivity & speed simultaneously? For 20 years, many tried: suffix tree, better programming ..

New Idea: Spaced Seed

- Spaced Seed: nonconsecutive matches and optimize match positions.
- Represent BLAST seed by 11111111111
- Spaced seed: 111*1**1*1**11*111
  - 1 means a required match
  - * means "don't care" position
- This seemingly simple change makes a huge difference: significantly increases hit to homologous region while reducing bad hits.

Sensitivity: PH weight 11 seed vs BLAST 11 & 10





Outline

- What is homology search
- A simple (but profound) idea
- **Its theory**
- Its practical success

Formalize

- Given i.i.d. sequence (homology region) with Pr(1)=p and Pr(0)=1-p for each bit:

```
1100111011101101011101101011111011101
                              111*1**1*1**11*111
```

- Which seed is more likely to hit this region:
  - BLAST seed: 11111111111
  - Spaced seed: 111*1**1*1**11*111

Expect Less, Get More

- Lemma: The expected number of hits of a weight W length M seed model within a length L region with homology level p is

  $$(L-M+1)p^W$$

  Proof. $E(\#hits) = \Sigma_{i=1 \ldots L-M+1} p^W$ ■

- Example: In a region of length 64 with p=0.7
  - Pr(BLAST seed hits)=0.3
    E(# of hits by BLAST seed)=1.07
  - Pr(optimal spaced seed hits)=0.466, 50% more
    E(# of hits by spaced seed)=0.93, 14% less

Why Is Spaced Seed Better?

A wrong, but intuitive, proof: seed s, interval I, similarity p

E(#hits) = Pr(s hits) E(#hits | s hits)

Thus:

Pr(s hits) = Lp^w / E(#hits | s hits)

For optimized spaced seed, E(#hits | s hits)

111*1**1*1**11*111	Non overlap	Prob
111*1**1*1**11*111	6	p^6
111*1**1*1**11*111	6	p^6
111*1**1*1**11*111	6	p^6
111*1**1*1**11*111	7	p^7

.....

- For spaced seed: the divisor is $1+p^6+p^6+p^6+p^7+$ …
- For BLAST seed: the divisor is bigger: $1+ p + p^2 + p^3 +$ …

Complexity of finding the optimal spaced seed (Li, Ma, manuscript)

Theorem 1. Given a seed and it is NP-hard to find its sensitivity, even in a uniform region.

Theorem 2. The sensitivity of a given seed can be efficiently approximated with arbitrary accuracy, with high probability.

Theorem 3. Optimal seeds can be found in exponential time deterministically. Near optimal seed can be found in $O(n^{\log n})$ time probabilistically.

Computing Spaced Seeds

(Keich, Li, Ma, Tromp, *Discrete Appl. Math*)

Let $f(i,b)$ be the probability that seed s hits the length i prefix of R that ends with b.

Thus, if s matches b, then

$$f(i,b) = 1,$$

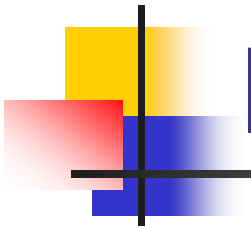
otherwise we have the recursive relationship:

$$f(i,b) = (1-p)f(i-1,0b') + pf(i-1,1b')$$

where b' is b deleting the last bit.

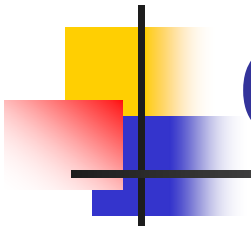
Then the probability of s hitting R is

$$\Sigma_{|b|=M} \, Prob(b) \, f(L-M,b)$$



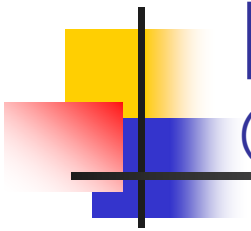
Prior Literature

- Random or multiple spaced q-grams were used in the following work:
  - FLASH by Califano & Rigoutsos
  - Multiple filtration by Pevzner & Waterman
  - LSH of Buhler
  - Praparata et al



Outline

- What is homology search
- A simple (but profound) idea
- Its theory
- **Its practical success**



PatternHunter

(Ma, Tromp, Li: *Bioinformatics*, 18:3, 2002, 440-445)

- PH used optimal spaced seeds, novel usage of data structures: red-black tree, queues, stacks, hashtables, new gapped alignment algorithm.
- Written in Java.
- Used in Mouse Genome Consortium (*Nature*, Dec. 5, 2002), as well as in hundreds of institutions and industry.

Comparison with BLAST

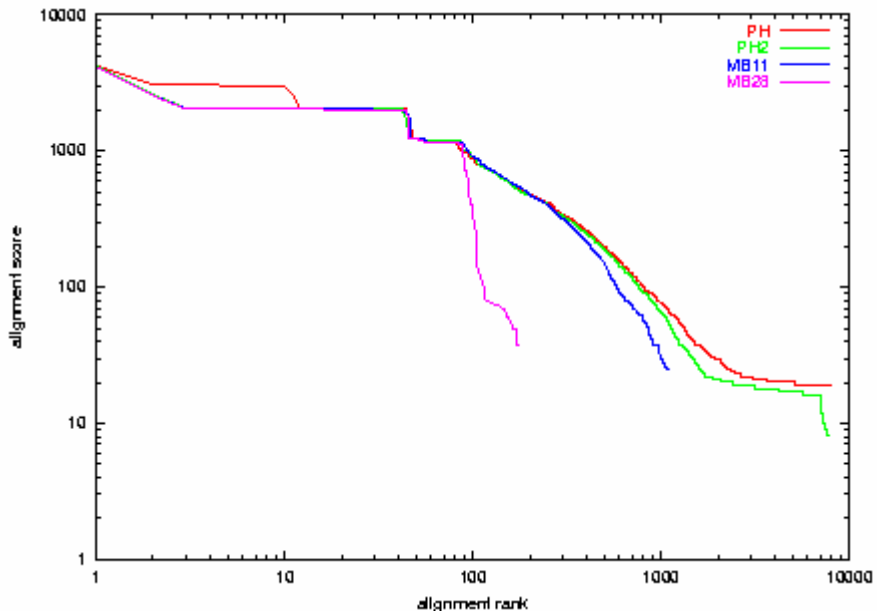
- On Pentium III 700MH, 1GB

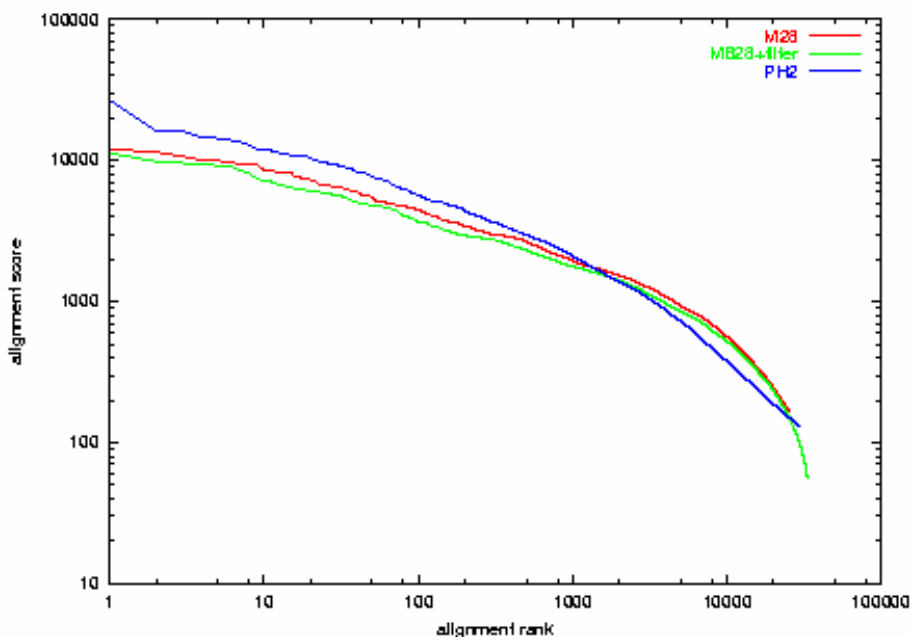
	BLAST	PatternHunter
E.coli vs H.inf	*716s*	*14s/68M*
Arabidopsis 2 vs 4	--	*498s/280M*
Human 21 vs 22	--	*5250s/417M*
Human(3G) vs Mouse(x3=9G)*	*19 years*	*20 days*

- All with filter off and identical parameters
- 16M reads of Mouse genome against Human genome for MIT Whitehead. Best BLAST program takes 19 years at the same sensitivity

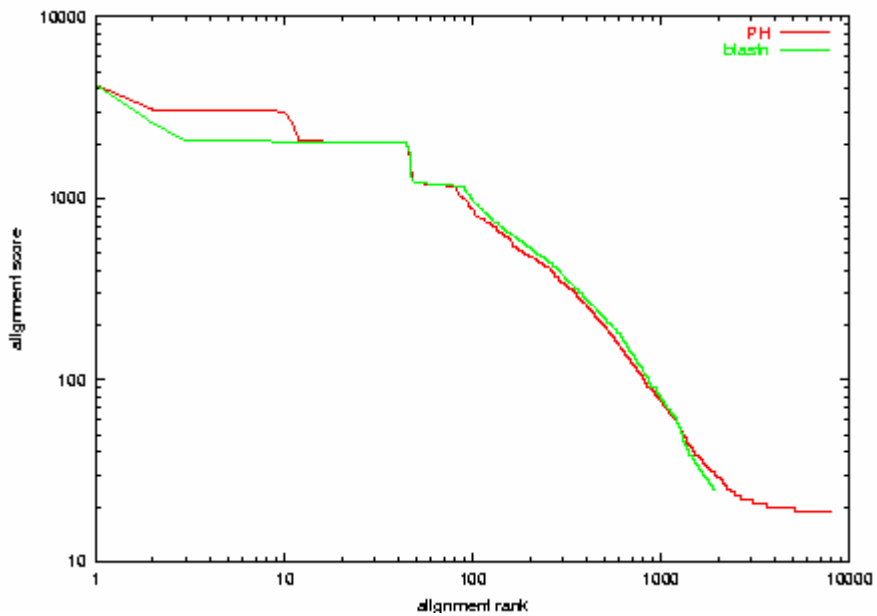
Quality Comparison:

x-axis: alignment rank
y-axis: alignment score
both axes in logarithmic scale

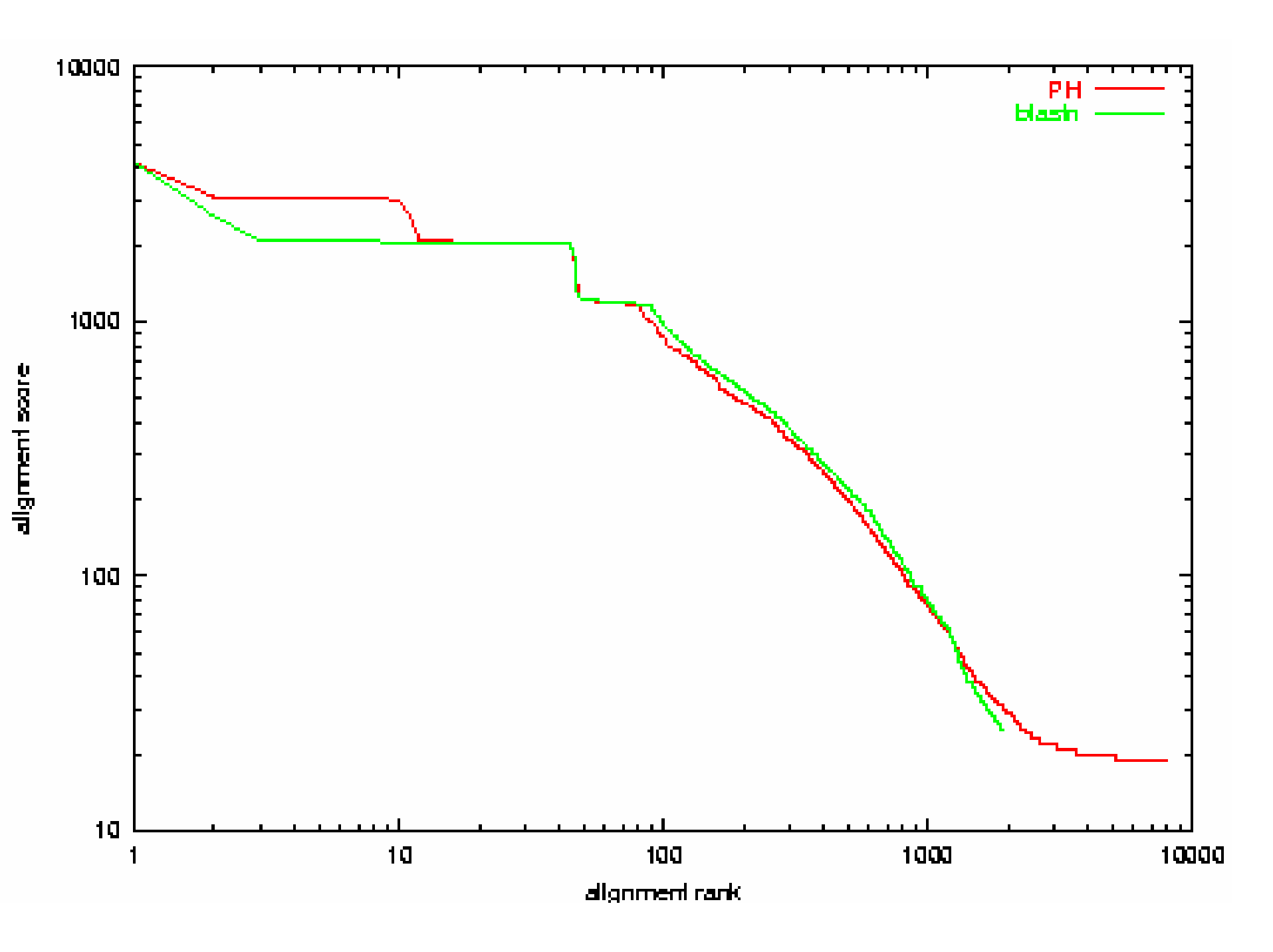


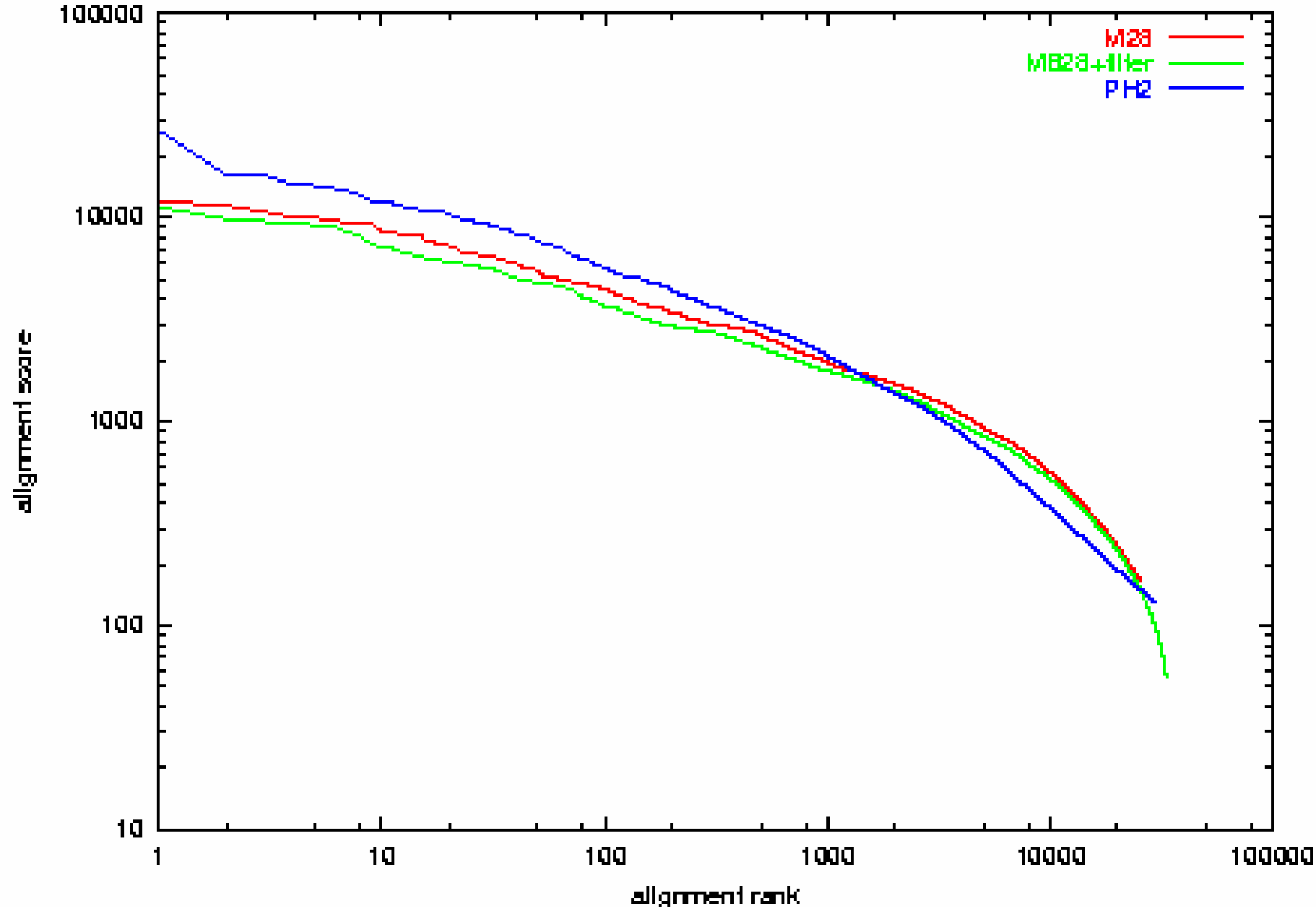


A. thaliana chr 2 vs 4

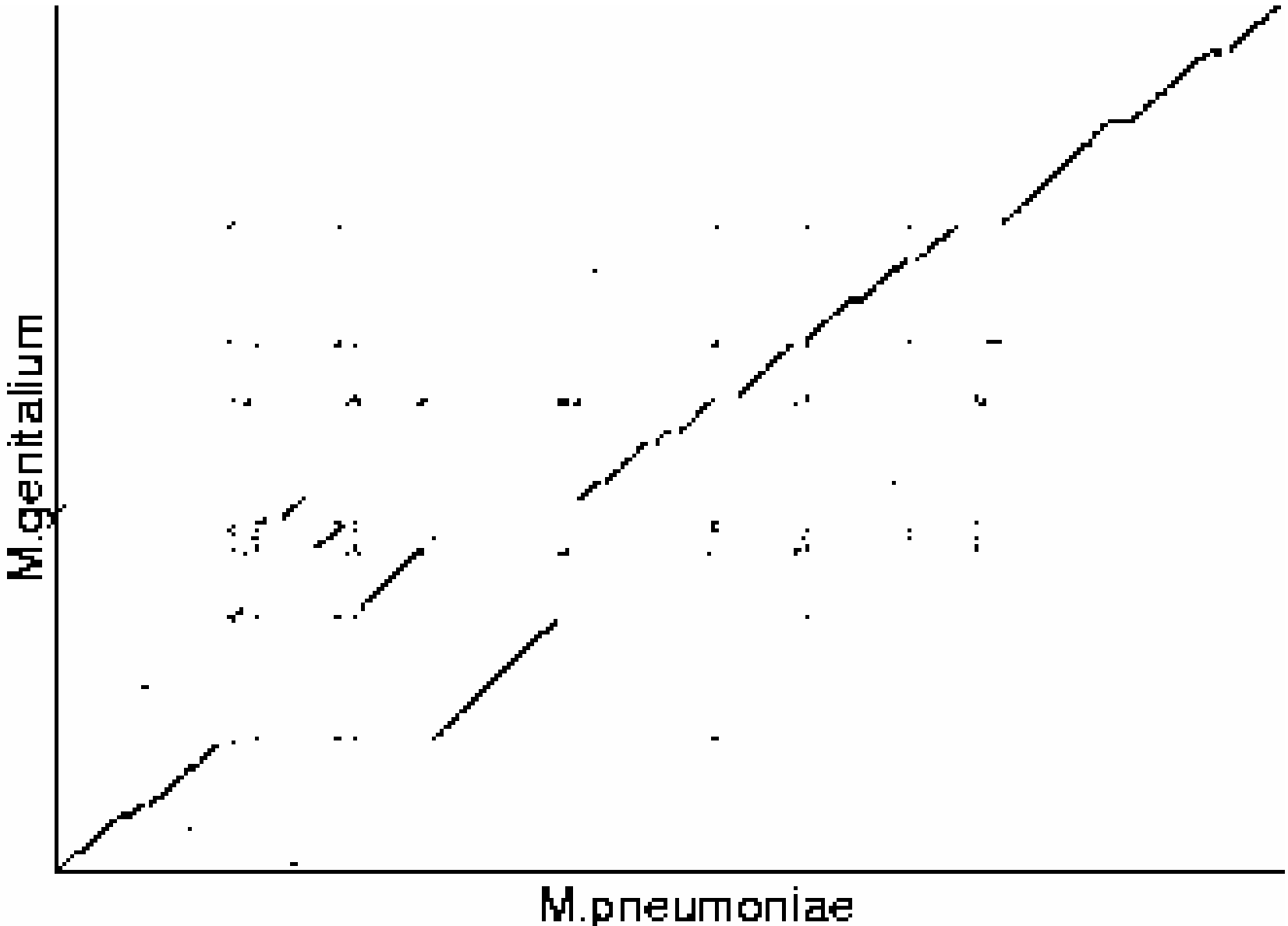


E. Coli vs *H. influenza*


10000
1000
100
10
alignment score
1
10
100
1000
10000
alignment rank
PH
blastn


M2B
MB2B+filter
PH2
100000
10000
1000
100
10
1
10
100
1000
10000
100000
alignment score
alignment rank

Genome Alignment by PatternHunter (4 seconds)



PattternHunter II:

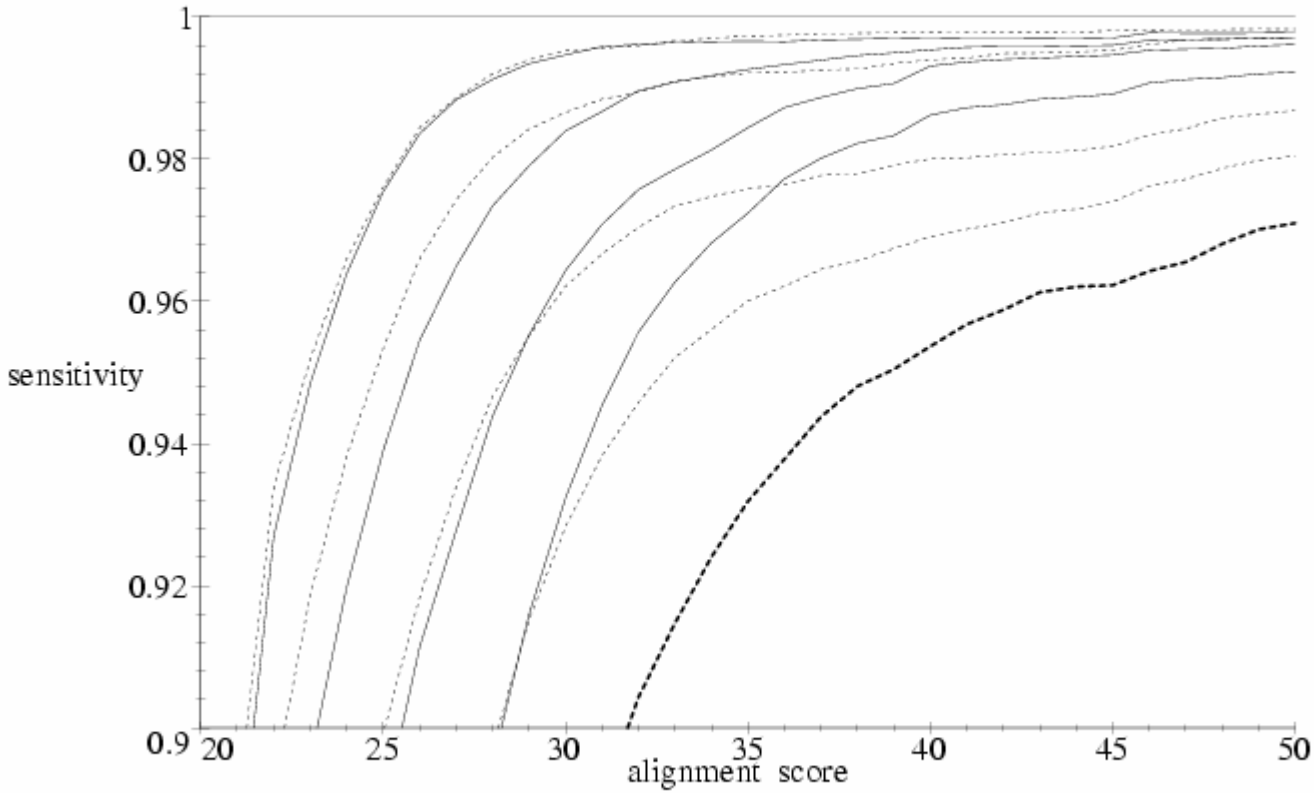
-- Smith-Waterman Sensitivity, BLAST Speed

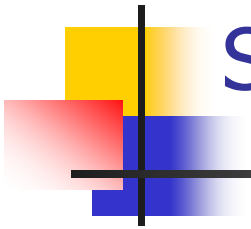
(Li, Ma, Kisman, Tromp, *J. Bioinfo Comput. Biol*. 2004)

- The biggest problem for BLAST was low sensitivity (and low speed). Massive parallel machines are built to do S-W exhaustive dynamic programming.
- Spaced seeds give PH a *unique* opportunity of using several optimal seeds to achieve optimal sensitivity, this was not possible by BLAST technology.
- We have designed PH II, with multiple optimal seeds.
- PH II approaches Smith-Waterman sensitivity, and 3000 times faster.
- Experiment: 29715 mouse EST, 4407 human EST.

Sensitivity Comparison with Smith-Waterman (at 100%)

The thick dashed curve is the sensitivity of BLAST, seed weight 11. From low to high, the solid curves are the sensitivity of PH II using 1, 2, 4, 8 weight 11 coding region seeds, and the thin dashed curves are the sensitivity 1, 2, 4, 8 weight 11 general purpose seeds, respectively





Speed Comparison with Smith-Waterman

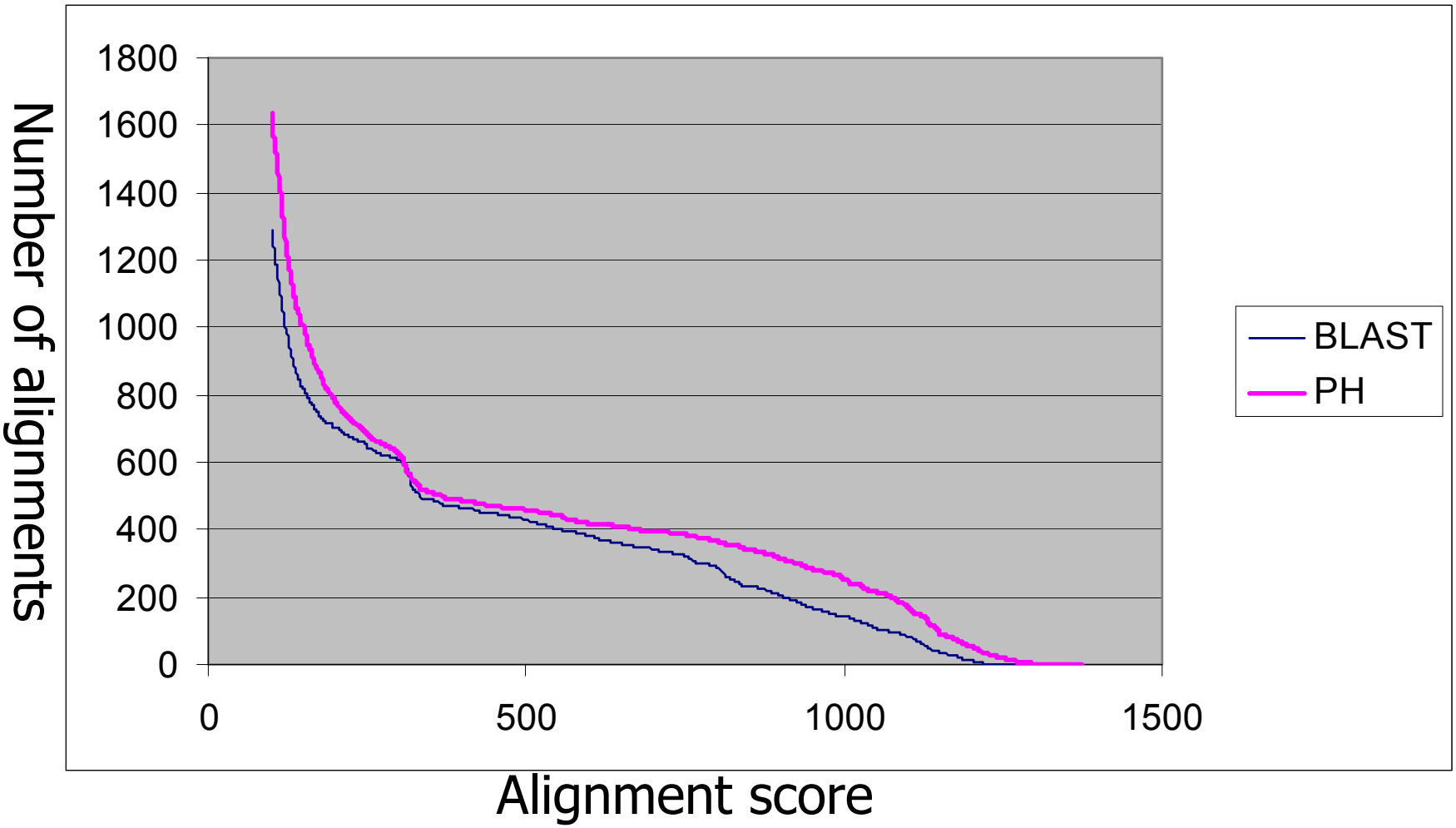
- Smith-Waterman (SSearch): 20 CPU-days.
- PatternHunter II with 4 seeds: 475 CPU-seconds. 3638 times faster than Smith-Waterman dynamic programming at the same sensitivity.

Translated PatternHunter

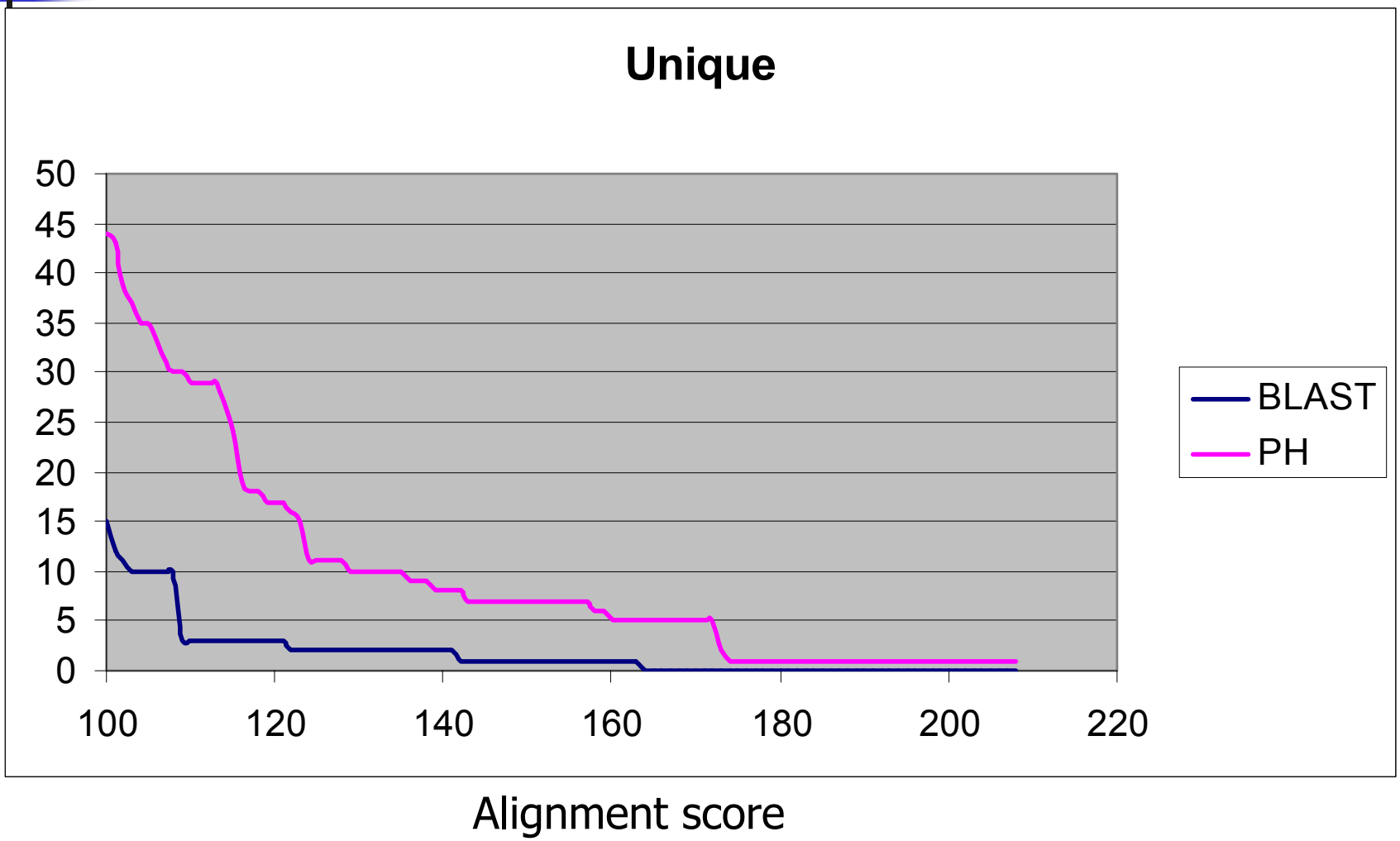
- Has all the functionalities of
  - Blastp
  - tBlastx – with gapped alignments
  - tBlastn, Blastx – with gapped alignments
- More sensitive and faster – new algorithm replacing 6-frame translation

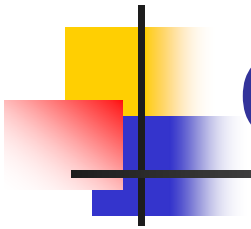
Alignment comparison: tBLASTx vs tPH

tPH: 253 seconds
tBLASTx: 807 seconds



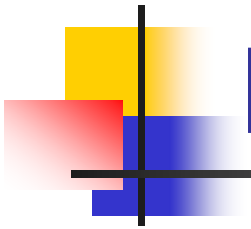
Unique Alignments: tBLASTx vs tPH





Old field, new trend

- Research trend
  - Over 30 papers on spaced seeds have appeared since our original paper, in 2 years.
  - Many more have used PH in their work.
  - Most modern alignment programs (including BLAST) have now adopted spaced seeds
  - Spaced seeds are serving thousands of users/day
- PatternHunter direct users
  - Pharmaceutical/biotech firms.
  - Mouse Genome Consortium, *Nature, Dec. 5, 2002.*
  - Hundreds of academic institutions.

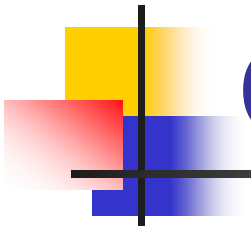


Running PH

Available at: www.BioinformaticsSolutions.com

Java –Xmx512m –jar ph.jar –i query.fna –j subject.fna –o out.txt

-Xmx512m --- for large files
-j missing: query.fna self-comparison
-db: multiple sequence input, 0,1,2,3 (no, query, subject, both)
-W: seed weight
-G: open gap penalty (default 5)
-E: gap extension (default 1)
-q: mismatch penalty (default 1)
-r: reward for match (default 1)
-model: specify model in binary
-H: hits before extension
-P: show progress
-multi 4: use 4 seeds

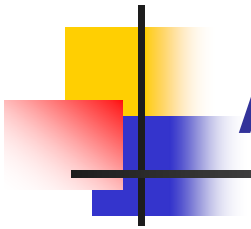


Conclusion

Best ideas are simple ones. I hope I have presented one such idea today.

Open questions:

- Polynomial time probabilistic algorithm for finding (near) optimal seed, multiple seeds.
- Tighter bounds on why spaced seeds are better.
- Applications to other areas.



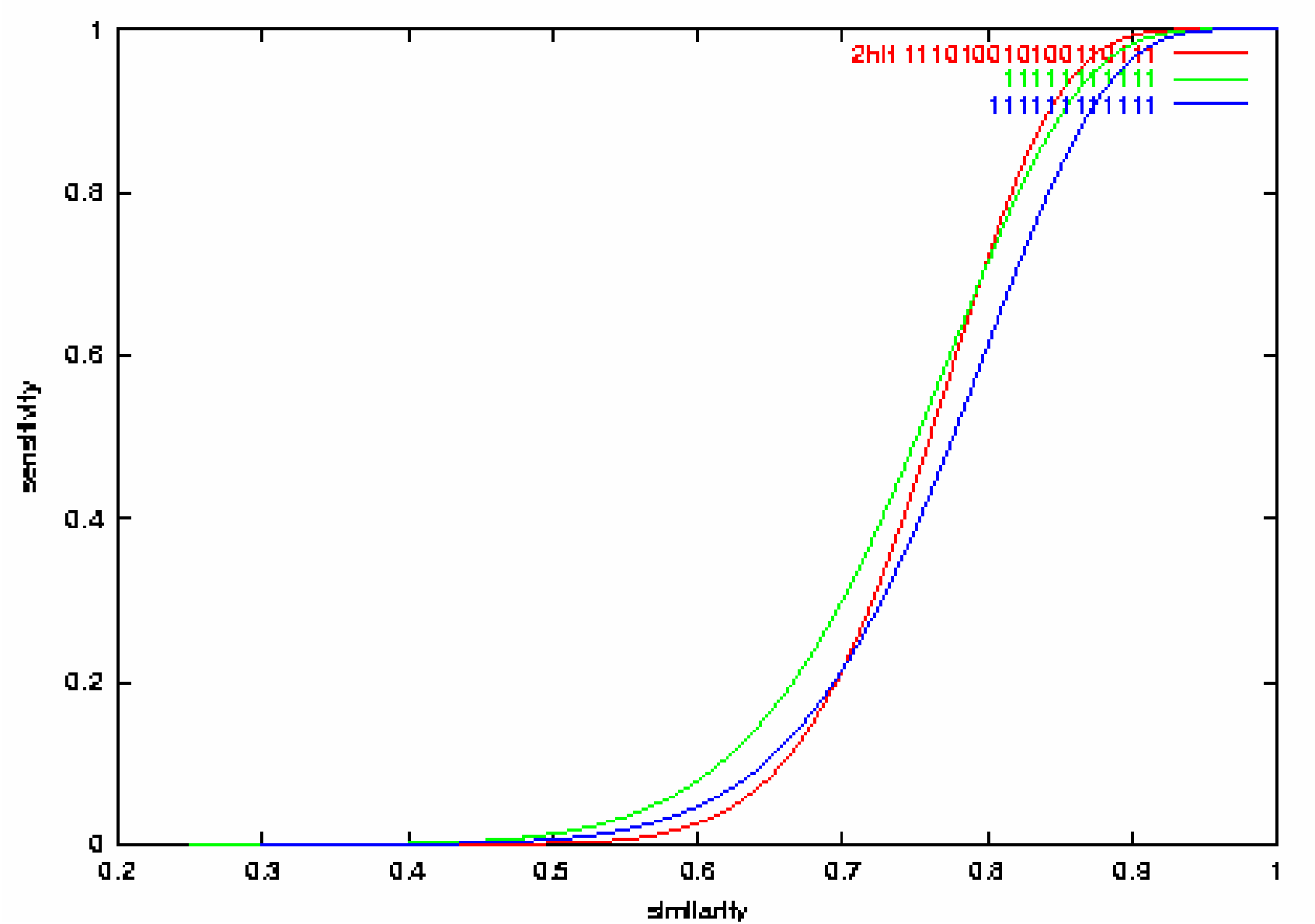
Acknowledgement

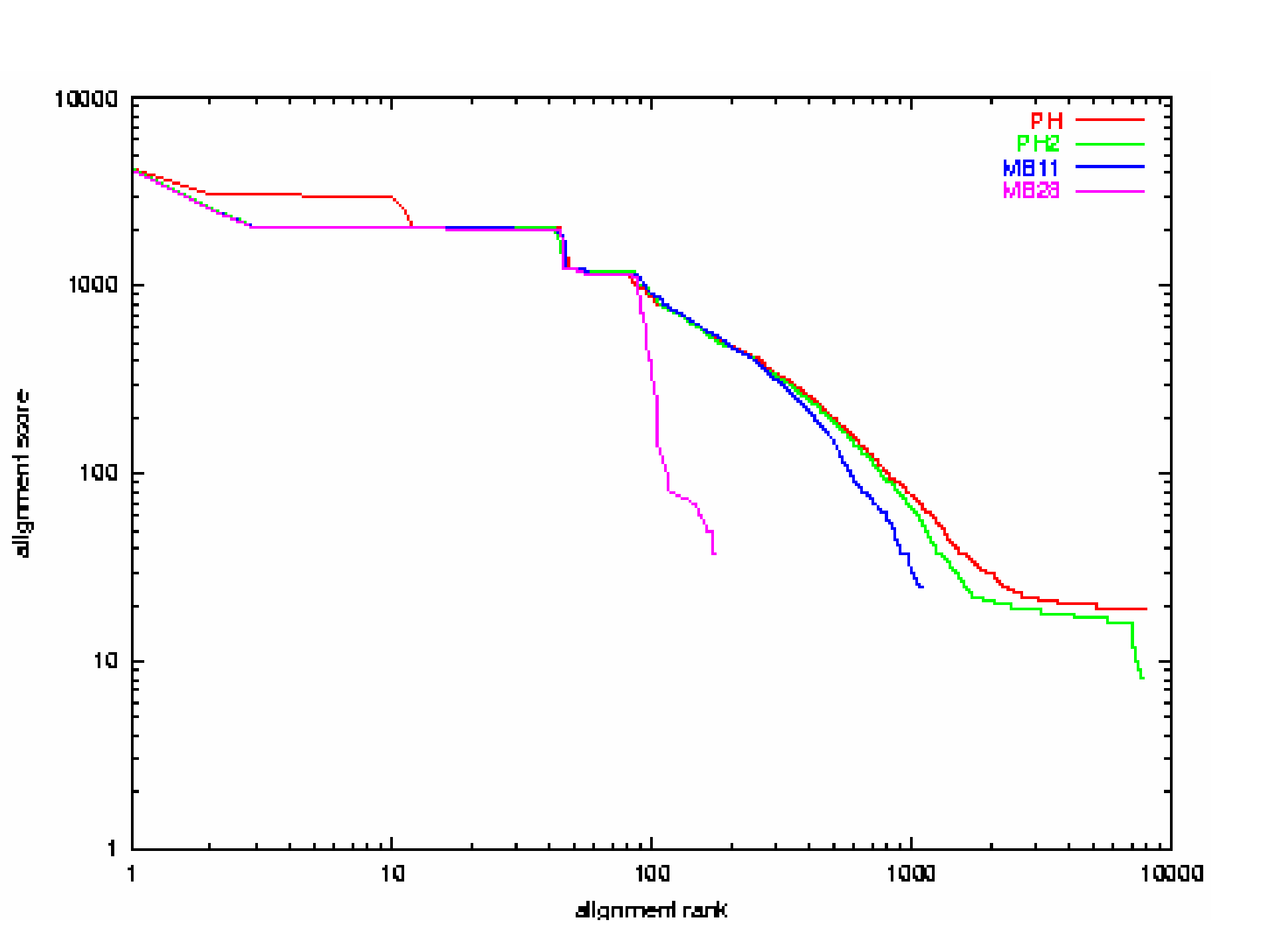
- PH is joint work with Bin Ma and John Tromp
- PH II is joint work with Ma, Kisman, and Tromp
- Some joint theoretical work with Ma, Keich, Tromp, Xu, and Brown.
- Financial support: Bioinformatics Solutions Inc, NSERC, Killam Fellowship, Steacie Fellowship, CRC chair program.

Conclusion – continued

- Another simple idea applied to data mining: from irreversibly computing 1 bit requires 1kT energy (von Neumann, Landauer), we derived shared information d(x,y) between x,y, to classify
  - Species & genomes, Li *et al*, in *Bioinformatics*, 2001
  - Chain letters, Bennett, Li, Ma, *Scientific American*, 2003
  - Languages, Benedeto,Caglioti,Loreto, *Phy. Rev. Let.*'02
  - Music, Cilibrasi, Vitanyi, de Wolf, *New Scientist*, 2003
  - Time series/anomaly detection, Keogh, Lonardi, Ratanamahatana, KDD'04. They compared d(x,y) with 51 methods/measures from SIGKDD, SIGMOD, ICDM, ICDE, SSDB, VLDB, PKDD, PAKDD and concluded our method the simplest & best --- Keogh tutorial ICDM'04.

PH 2-hit sensitivity vs BLAST 11, 12 1-hit




PH
PH2
MB11
MB23
alignment score
alignment rank
10000
1000
100
10
1
1
10
100
1000
10000

Natura enim simplex est,
et rerum causis superfluis non luxuriat.
I. Newton