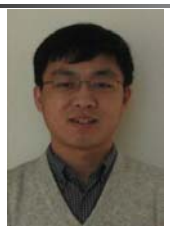
GIW Dec 1-3, 2008, Brisbane

Modern Homology Search

Ming Li

Canada Research Chair in Bioinformatics
University of Waterloo

Coauthors



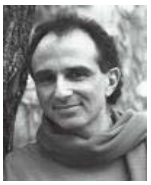
Bin Ma

John Tromp

Xuefeng Cui

Tomas Vinar

Brona



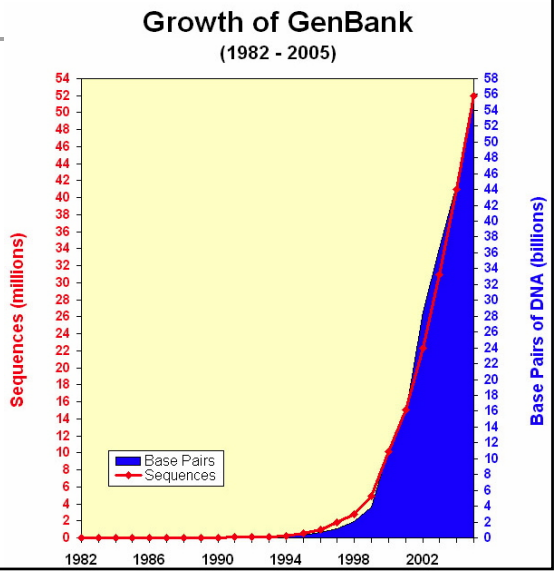
Dennis Shasha

I will present several simple ideas --

some worked, some didn't.

faster than we can search it

- GenBank doubles every 18 months
- 600 Eukaryote genome projects underway
- Solexa and 454: $1000-one day-genomes in 5 years



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 bases

  Output: all local alignments.

Comparing to internet search

- Internet search
  - Size limit: 5 billion people x homepage size
  - Supercomputing: ½ million CPU-hours/day
  - Query frequency: Google --- 112 million/day
  - Query type: exact keyword match --- easy to do
- Homology search
  - Size limit: 5 billion people x 3 billion basepairs + millions of species x billion bases
  - Query frequency: NCBI BLAST -- 150,000/day, 15% increase/month
  - Query type: approximate match.

Homology Search

Old paradigm

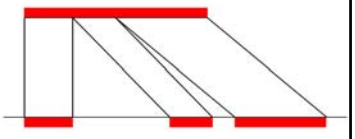
Sensitive, but slow

Fast, low sensitivity

Low specificity

We want

100% sensitivity
Fast



Old Homology Search

- Dynamic programming (1970-1980)
  - Human vs mouse genomes: 10^4 CPU-years
- BLAST, FASTA heuristics (1980-1990)
  - Trading sensitivity for speed,
  - Yet, still not fast enough -- Human vs mouse genomes: 3 CPU-years.
  - It takes years to map Illumina/Solexa reads, produced in 1 day, to a reference human genome

Modern Homology Search

- ~100% sensitivity, approaching to dynamic programming. Not sacrificing speed.
- Return proper gene matches: with intron/exon boundaries
- 1 day whole genome reads mapping.

Talk Outline

1. A simple idea: spaced seeds.
2. A trivial idea: multiple seeds.
3. An idea to make the search specific.
4. A bad idea: changing seeds.
5. The bad idea becomes good idea for a different application.

1. Optimal Spaced Seeds

BLAST Algorithm & Example

- Find seeded matches of eleven base pairs, represented as 1111111111.
- Extend each match to right and left, until the scores drop, to form an alignment.
- Report all local alignments.

Example:

0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0

AGCGATGTCAGGCGCCCGTATTTCCGTA

| | | | | | x | | | | | | | | | | | | | | | |

TCGGATCTCACGCGCCCGGCTTACCGTG

BLAST Dilemma:

- Speed & sensitivity have contradictory requirement for seed length:
  - 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: Optimal Spaced Seed

BLAST seed was:

11111111111

And this:

11111*11*11*11

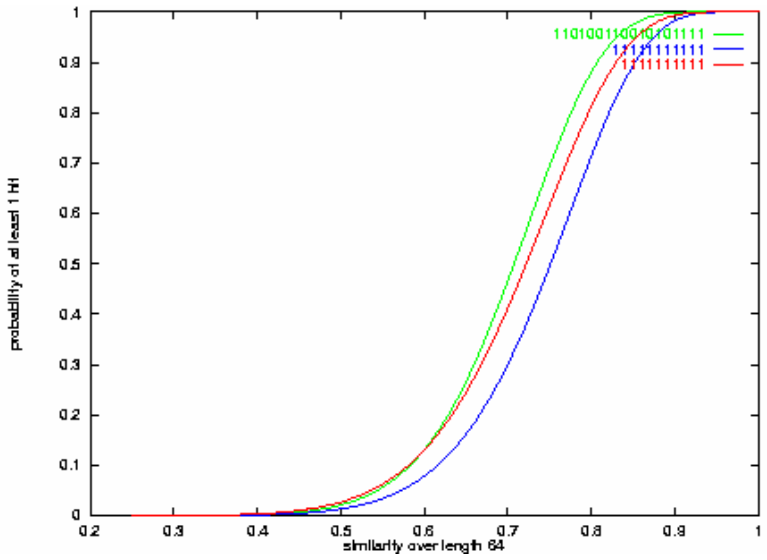
Optimizing gives: 111*1**1*1**11*111

- 1 means a required match
- * means "don't care" position

Optimal Spaced Seed

- Spaced Seed: nonconsecutive matches and optimize match positions.
- BLAST seed 11111111111 is the worst seed
- Spaced seed: 111*1**1*1**11*111 is optimal
  - 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



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

	Non overlap	Prob
111*1**1*1**11*111		
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, Zhang, SODA'2006)

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.

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

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

Related 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 on probe design
- Optimizing & further work
  - Buhler-Keich-Sun
  - Brejova-Bronw-Vinar
  - Choi-Zhang
  - Over 100 research papers.

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 & industry.

Comparison with BLAST

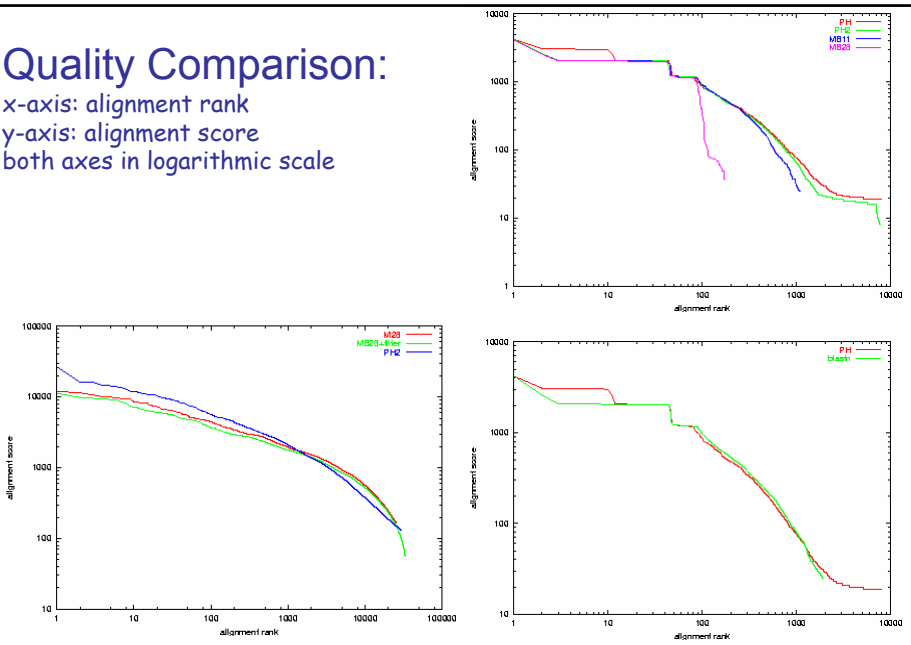
- On Pentium III 700MH, 1GB

	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
- Mouse genome against Human genome (*Nature*, 2002) 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



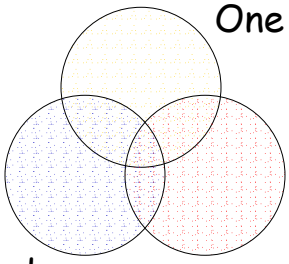
A. thaliana chr 2 vs 4

E. Coli vs *H. influenza*

2. Multiple Seeds: Full Sensitivity

More seeds, more sensitivity

Space of homologous regions



One seed

Two seeds

Three seeds

PattternHunter II:

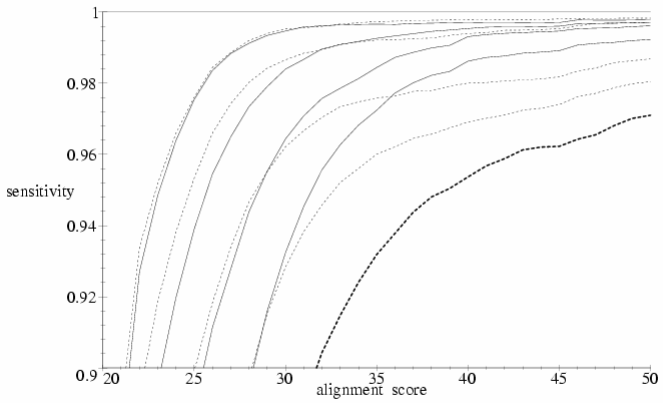
-- Smith-Waterman Sensitivity, BLAST Speed

(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.
- Using multiple optimal seeds. PH II approaches Smith-Waterman sensitivity & 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, resp.



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.

3. Homology search for genes

Meaningful Match?

- Given a gene sequence, BLAST or PH simply returns a bunch of alignments.
- Can we return a complete gene match?
- Idea: Combine PH with ExonHunter (Brejova, Brown, Li, Vinar, ISMB'2005): take the ab initio gene-finder (HMM) trained for the database genome, further train/bias it with the query gene model (its splice sites etc). Use PH to find possible hot regions and use this HMM to do extension, deciding on introns/exons.

Example:

- Given a human gene [GI:35560], want a homologous gene in mouse genome [GI:293767]

BLAST Result

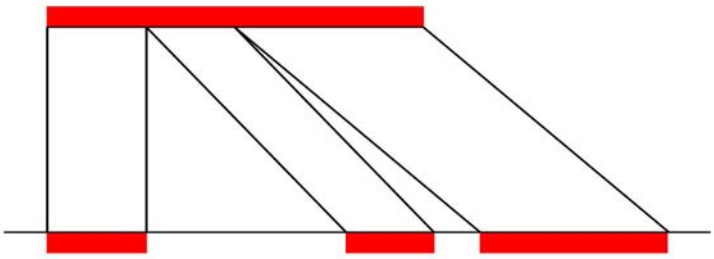
- 249 alignments are returned
- Only 3 alignments are relevant
- Exons / Splice sites are not detected

New gPH results

- Fully correct homologous gene-match is returned. Just one alignment!

human

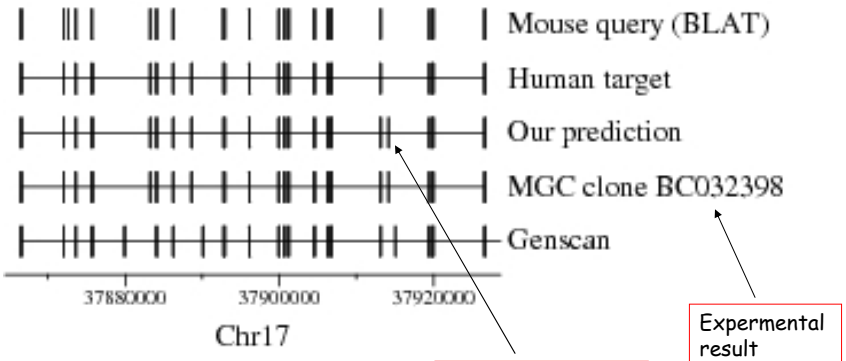
Mouse genome



An experiment

- 400 one-to-one orthologous gene pairs of human-mouse from NCBI HomoloGene database.
- At Exon level, gPH achieves
  - 79% sensitivity,
  - 80% specificity
- Compared to GenScan
  - 71% sensitivity
  - 50% specificity.
- And plain TBLASTN
  - 7% sensitivity
  - 5% specificity
- Found 50 (12%) human genes with better alignment.

One gene with better alignment: aligning a mouse gene to human seq.



Expermental result

Extra exon

4. An idea that did not work

- The optimal spaced seed has the least self correlation.
- Idea: can we further improve this by using different (or alternating) spaced seeds as we scan through the sequences?

```
111*1**1*1**11*111
 1*111**1*11*11**11
  111*1**1*1**11*111
   1*111**1*11*11**11
        …
```

- But this was no good!

5. The "same idea" now works

- Illumina/Solexa 1G Sequencing System: 1 billion bases per run.
- In a few years: 1 day-$1k-5x-genomes for personalized medicine.
- Key computational task: map all reads to a reference genome, and identify all SNP's.
- PatternHunter, BLAST, BLAT all need from CPU-days to CPU years for human genome.

Short Reads Mapping

- Around 35 bases
- Allow 2 mismatches
- Different from PH vs BLAST case. Here, the extension is very cheap, high homology. Competing goals:
  - Seed weight low → too many false positive, cost time in extension.
  - Seed weight high → too many seeds, cost time in seed mapping. In PH case, this did not matter

Picking up the previous idea

- Goal:
  - Minimum number of seed maps
  - 100% sensitivity allowing 2 mismatches
- What did not work in the PH, works now:
  - Use different seed each time
  - Optimize them.
- Example: Reads length 33, seed weight 13, 2 mismatches, these 4 seeds work. But are they optimal – can we use fewer seeds?

```
111111111111100000000000000000000
000000011111111111110000000000000
000000000000000000001111111111111
111111000000111111000000000000000
```

Designing the seeds.

- We proved 84 lower bound theorems, and constructed 84 upper bounds

Weight	Read Length											
	25	26	27	28	29	30	31	32	33	34	35	36
9	4	4	3	3	3	3	3	3	3	3	3	3
10	4	4	4	4	4	3	3	3	3	3	3	3
11	5	5	5	4	4	4	4	4	3	3	3	3
12	6	6	5	5	5	4	4	4	4	4	4	3
13	7	6	6	6	6	5	5	5	4	4	4	4
14			7	6	6	6	6	5	5	5	4	4
15					7	6	6	6	6	5	5	5
16							7	6	6	6	6	5

Tight bounds: # of seeds needed

ZOOM! Zillions Of Oligos Mapped

Joint work with Z. Zhang, H. Lin, B. Ma

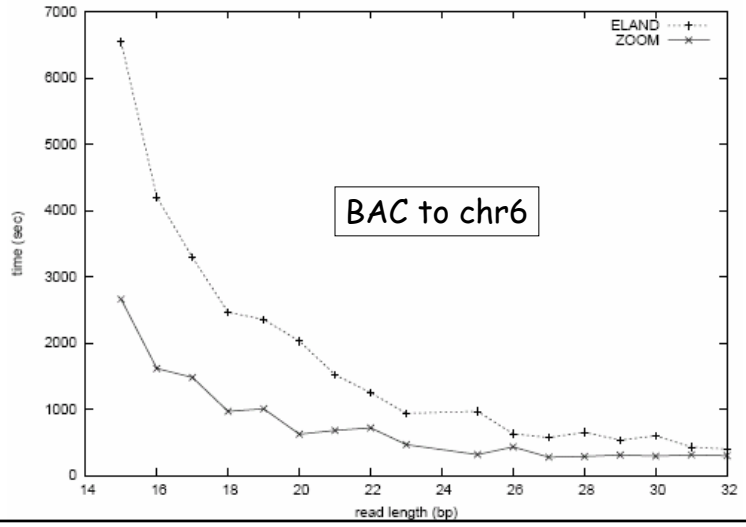
- We implemented these ideas in ZOOM

3.4 M reads, length=36

Human genome

program	BAC on MHC-162k	BAC on chr6	BAC on all
BLAST	06:56:11 (51M)	> 5 days	> 8 days
BLAT	00:04:06 (32M)	06:33:03 (32M)	7days+22:47:16(32M)
RMAP	00:00:51 (1.9G)	00:27:54 (1.9G)	10:09:03 (1.9G)
Mosaik	00:05:33 (214M)	00:07:41 (3.4G)	02:11:15 (3.5G)
ZOOM	00:00:37 (1.1G)	00:06:09 (1.1G)	01:33:03 (1.1G)

ZOOM vs ELAND (0.2.2.5)



ZOOM vs ELAND

ZOOM uses 6 weight 13 seeds. Reads length: 27

Data set		Reads cnt	ZOOM	ELAND
STAT1-stimulated on hg18	part1	12,471,522	03:24:13 (2.9G)	04:29:57
	part2	11,508,843	03:19:59 (2.9G)	03:41:53
	all	23,980,365	04:49:29 (5.1G)	-
STAT1-unstimulated on hg18	part1	7,667,108	02:48:03 (1.9G)	03:21:10
	part2	14,508,477	03:29:27 (3.4G)	04:28:34
	all	22,175,585	04:21:01 (4.8G)	-

Human genome

Chr6, 5x, 2 errors 24M reads

Experiment run	ZOOM
chr6.2X.e2 on chr6	00:09:48 (2.9G)
chr6.2X.e2 on the human genome	02:37:04 (2.9G)
chr6.5X.e2 on chr6	00:17:17 (6.5G)
chr6.5X.e2 on the human genome	04:48:05 (6.5G)
all.0.2X.e2 on the human genome	04:25:40 (4.5G)

Conclusion

Simple ideas are often the better ones.

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, Brown, Zhang.
- gPH is joint work with X.F. Cui, T. Vinar, B. Brejova, D. Shasha, ISMB'2007.
- ZOOM is joint work with Z. Zhang, H. Lin, B. Ma.
- Financial support: NSERC, Killam Fellowship, Steacie Fellowship, CRC chair program, Bioinformatics Solutions Inc.