

# BLAST as a microcosm of all that is wrong with computational biology

Titus Brown

6/7/11



# (NCBI) BLAST

- You've all used it?
- Very popular! Fast, sensitive way to find sequence similarity => putative homology.
- Primary sequence comparison tool used by biologists, computational biologists.

# (NCBI) BLAST

- One or more query sequences...
- against a “subject” database.
- Finds core strong match, extends outwards.

```
SIRGGGVVDHGISDDDESQHSGDAGIS
S+RGGG++ G+S++          D+G
SVRGGGIEIGLSEE-----DSGAE
```



# Consideration #1

- BLAST *only* cares about sequence similarity.
- No positional information taken into account, for e.g. protein domains.

Query: 1628 AHLLVNSQKC-KQTSSECIDTTDNAASVISARAS----TGTLEAEFPINTVASTTNPTTP 1682  
N+Q C K C + S++ +A+ + TL F I++ ST T

Sbjct: 1537 MTFHANTQMCVKLDLQSCPTNVASVKSILGEKAAEFSTSSTLSRVFRIDSEGSTQTGT-- 1710

Query: 1683 PQDYTYXXXXXXXXXXXXXXXXXXXXXTHRKRKRETSTLWAPGFGNVTKKQRREPIGQDDLNG 1742  
Y +KRKRE LW PEGF + KK+R+E ++LN

Sbjct: 1711 TNYLVYIIAGGGIMVLIIVIAGVIVSQKRKRENGNLWVPEGFQLFKKRRKE----NELNL 1878

Query: 1743 LNGSIHPGELTQLDT-AGTPFLNRWENTSLPQKSNHYHVQYTPENITFLPNNGTVPXXXX 1801  
N L++ D A TPFL + + Q S + +L

Sbjct: 1879 NN-----LSKADMNAQTPFL---PHATEAQASKYSASSSDTPETDYL----- 1995

Query: 1802 XXXXXXXXXXXXXXXEPTDNRKWTPOHLEAADLSRAGSACTPVTDLTPPPHIDVDEDDVNAR 1861  
D R+WTP HLEAA+ S C + TPP + DD+NAR

Sbjct: 1996 -----HGSCASKEDKRQWTPHHLEAANNSNVN--CQIMN--TPPQSECPESEDINAR 2139

Query: 1862 GPDGVTPLMVASIRGGGVVDHGISDDESQHSGDAGISGEGSDSMIXXXXXXXXXXXXXXTDR 1921  
GPDG TPLM+AS+RGGG++ G+S++ D+G GEGSD+MI TDR

Sbjct: 2140 GPDGYTPLMIASVRGGGIEIGLSEE-----DSGAEGEGSDNMIAGLILQGASLSATTDR 2301

Query: 1922 SGETXXXXXXXXXXXXXXXXXXXXXXXXXNMKDHSGRTPPLHSAVAADAQGVFQILLRNRAT 1981  
+GET NMKD +GRTPPLH++VAADAQGVFQILLRNRAT

Sbjct: 2302 TGETALHLAARYARADAAKRLLDAGADANMKDQTGRTPPLHNSVAADAQGVFQILLRNRAT 2481

Query: 1982 DLDARTNDGTTPMILASRLAVEGMVEELISANADVNAVDDHGKSALHWAAAVNNVDAVST 2041  
DLDA+TNDGTTP+ILASRLAVEGMVE+LI+A+ADVNAVDD+HGKS+LHWAAAVNN DA+

Sbjct: 2482 DLDARTNDGTTPMILASRLAVEGMVEDLITAHADVNAVDDNHGKSSLHWAAAVNNNDIRA 2661

Query: 862 TCVCTPGFQGPTCANDINECMSPCKNGGKCRNREPGYFCECLDGYSGVNCEENVDDCAS 921  
TC QG T AN C G C N + C C +G++G CE ++ C  
Sbjct: 34 TCEVQAASQGTTVAN-----VCNGQGTCTCINSGNSHTCTCAEGFTGSYCETIINHCDP 189

Query: 922 DPCMNGGTCLDDVNSYKCLCKRGFDGNQCQNDVNECENEPCCKNGATCTDYVNSYACTCPP 981  
+PC+N C +N Y+C C+ GF G+QCQ D++EC + PC NG TC + +N + C+CP  
Sbjct: 190 NPCINAVKCTSGINGYECDCEAGFQGSQCQLDIDECTSNPCMNGGTCFNAINGFQCSCPR 369

Query: 982 GFRGTTCMENIDE CNIGSCLNGGTCVDGINSYSCNCMAGFTGANCERDIDECVSSPC--K 1039  
G G C C+ C N G C GI S++C C G+ G C DI+EC S+PC +  
Sbjct: 370 GTLGVLCEVVSSLCDPNPCQNNGHCTSGIGSFTCQCKPGYGGYLCNGDINECASNPCSTE 549

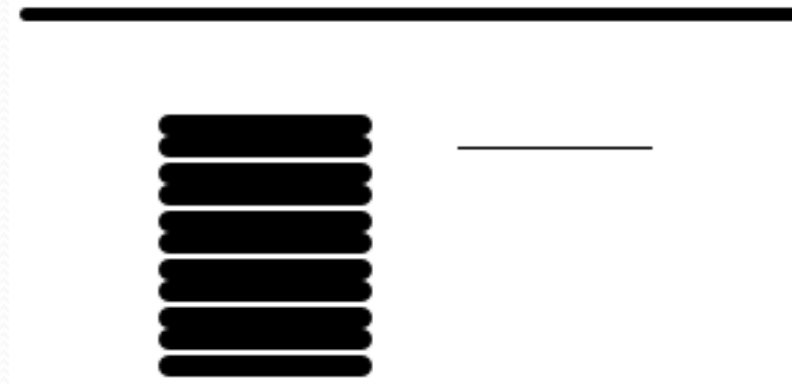
Query: 1040 NGAPCIHGINTFTCQCLTG YTGPTCAQMVDLCQNNPCRNGGQCSQTGTTSK---CLCTSS 1096  
C+ GIN F+C C GY G TC+ C NNPC NG C+ C CT+  
Sbjct: 550 GSLDCVQGINEFSLCKDGY YGDTCSNQASSCSNNPCLNGATCTDNSLEPLRYFC SCTND 729

Query: 1097 YSGVYCDVPRLSCSAAATWQGV EETSLCQHGGQCINSGSTHYCSCRAGYVGSYCETD--- 1153  
Y G C++ +C + +C + G+C++ GS YC C GY G+ C ++  
Sbjct: 730 YRGKNCMEFSTCPSLDM-----ICYNDGKCVD-GSAPYCKCPFGYTGTQCMSNTNT 882

Query: 1154 EDDCASY 1160  
E C+SY  
Sbjct: 883 EKQCSSY 903

# Consideration #2

- BLAST is a *local* alignment algorithm.
- Strong matches are reported first; multiple matches may be out of order between query, subject.



# Consideration #3

- BLAST creates gapped alignments.

```
SIRGGGVVDHGISDDESQHSGDAGIS
S+RGGG++ G+S++          D+G
SVRGGGIEIGLSEE-----DSGAE
```

- This means it's totally inappropriate for (for example) primer matching, unless you change the parameters.
- (Who here has actually changed BLAST parameters?)



# Consideration #4

- BLAST e-values are database-size dependent.
- BLAST bit scores are not.

Score = 87.0 bits (214), Expect = 5e-16

- You can't technically compare e-values from BLASTs against different databases!

# Consideration #5

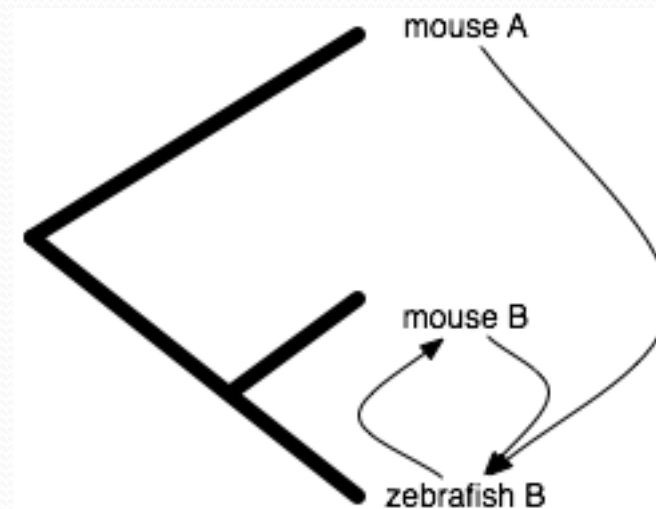
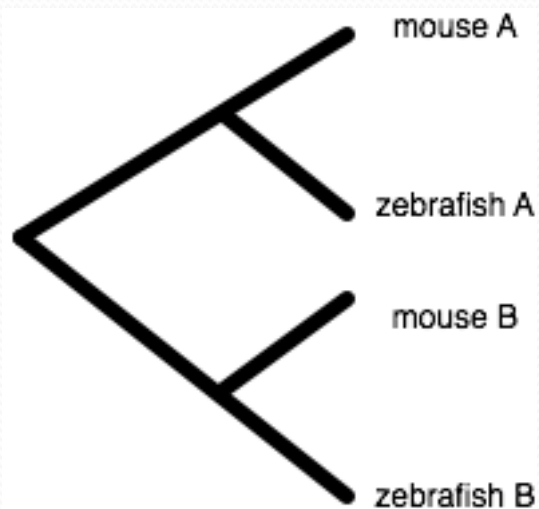
- BLAST uses an heuristic to speed things up: *requires* an **exact** match between 11 bases (DNA) or 3 amino acids in order to start an alignment.

```
Query: 862 TCVCTPGFQGPTCANDINECMSPPCKNGGKCRNREPGYFCECLDGYSGVNCEENVDDCAS 921
          TC      QG T AN      C   G C N      + C C +G++G   CE   ++ C
Sbjct: 34  TCEVQAASQGTTVAN-----VCNGQGTCINSGNSHTCTCAEGFTGSYCETIINHCDP 189

Query: 922 DPCMNGGTCLDDVNSYKCLCKRGFDGNqcqNDVNECENEPCCKNGATCTDYVNSYACTCPP 981
          +PC+N   C   +N Y+C C+ GF G+qcq D++EC + PC NG TC + +N + C+CP
Sbjct: 190 NPCINAVKCTSGINGYECDCEAGFQGSqcqLDIDECTSNPCMNGGTCTFNAINGFQCSCPR 369
```

# Consideration #6

- Reciprocal BLAST is a *horrible* (but frequently used) heuristic for “orthology”. Intended for:



...but local alignments cause trouble here!



# Considerations #7+

- BLAST implementation is (was?) impenetrable: completely inextensible, very optimized, built on a huge library.
  - Does it have bugs? Nobody knows...
  - V. difficult to embed => difficult to reuse
- BLAST text output format changes frequently and is designed for humans only to read; very hard for computers to parse.



# BLAST is also kind of inconvenient

- No good Web interface for uploading your own databases (that I know of).



# So, nobody uses BLAST, right?

- Absolutely wrong!
- Biologists love it: it's fast, sensitive, and has a nice Web interface at NCBI.
- Bioinformaticians love/hate it:
  - Biologists => programmers use it by default, and then spend a lot of time correcting for its problems.
  - Computer scientists => biologists often can't escape:
    - Lots of biology behind BLAST; tough to write your own.
    - Biologists *believe* in BLAST, and not your own dinky algorithm.

# Digression: it's not BLAST's fault, really.

- Most of the “considerations” I presented are completely obvious and stated clearly all over the place.
- Everybody uses BLAST because it's there, it (mostly) works, and it's trusted by (almost) everyone.
- BLAST use may be starting to break down, though:
  - Doesn't scale to volume of data
  - Default gapping model is inappropriate for short-read mapping
  - Has significant false positive rate on very divergent proteins (metagenomics, “evolutionarily interesting” organisms)



# This course & BLAST

- We'll be (mis)using BLAST just like everyone else.
- We'll show you how to run BLAST at the command line:
  - Run long jobs on some other computer
  - Make your own BLAST databases
- We'll show you BLAST output “parsing”
  - Make your own spreadsheet of matches
  - Your very own reciprocal BLAST script...





# The UNIX command line

- Many computer folk, and most bioinformaticians, work with a text interface to their computers: “the command line”.
- Sort of the granddaddy of all interfaces... think back to teletypes.
- Why?
  - Writing *new* programs is much easier if you write them for the command line (text, no graphics)
  - Simple & flexible (not nec. *good*) user interface design: **none**
  - Simple “pipelining” ability (Th or Fri)
- Almost all bioinformatics programs work at the command line, or via a Web interface.



# The UNIX command line, part 2

- This software can be installed on your computer (Windows) or already exists (Mac OS X).
- ...but we really, really, really don't want you to use your own computer to do analyses!
  - Laptops are sloooow
  - Data files are big
  - *Your* computer is here at KBS, and we don't want you transferring 50 gb+ of data here!
  - You'd much rather use your laptop as an interface!
- **Dilemma.** But we have a solution...



# Part II: Cloud computing



# What is cloud computing?

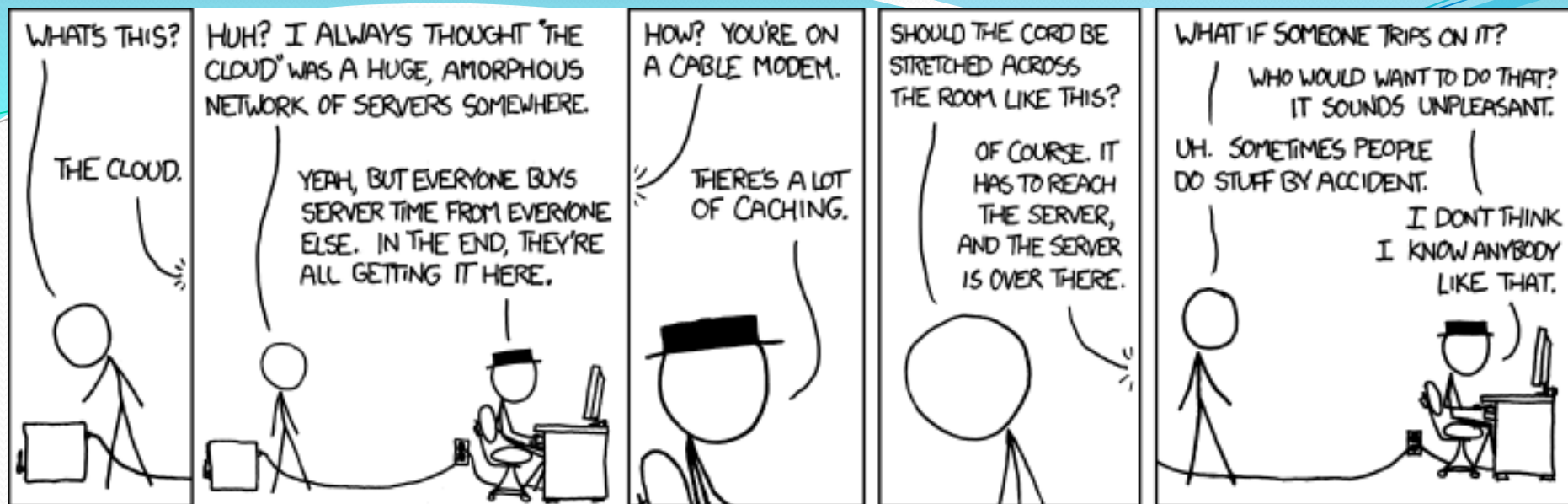
(for scientists)

- You can “rent” access to computers and disk space from a commercial provider of same.
- This provides you with a way to scale your computation for “burst” periods, without investing in hardware.
- Or you can just use a bigger, faster computer.
- (I will demonstrate.)

# Why “cloud”?!



...because the diagram that CS people use to represent abstract compute resources looks like a cloud.



xkcd.com

Editor's note: Mr. Munroe has been missing for several days. We have received no submissions from him for some time, but we found this single panel on his desk in a folder labeled 'MY BEST IDEA EVER'. It is clearly part of a work in progress, but we have decided to post it in lieu of a complete comic.



# Amazon is a major cloud computing provider

- Did you know they rent computers!?
- Rumors are that it's more lucrative than their book selling division... not sure how that's measured, though.





# Terms

EC2 – Elastic Cloud Computing, computer rental from Amazon.

EBS – Elastic Block Storage, virtual hard drive rental from Amazon.

# Some quick calculations:

1 small machine, / yr:

1.7gb of RAM, a ~1.0 GHz single-core CPU, 160gb of local disk.

\$.085 / hr

8760 hrs / year

=> ~\$750 / year.

*Not an effective server replacement.*



1 high-memory quadruple extra-large instance / yr:

68.4 gb of RAM, 8 core @ ~3.2 GHz, 1.7tb of local disk.

\$2.40 / hr

8760 hrs / year

=> \$21,000 / year



20 high-CPU extra large machines, for a day:

7gb of RAM, 8 x 2.5 GHz CPUs, 1.7tb of local disk.

\$.68 / hr

24 hrs / day

20 machines

=> ~\$330/ day.



# Why is EC2 so expensive??

- They cover *all* hardware, power, air conditioning and network costs.
- That's actually way more expensive than you think. (Talk to your sysadmin or HPC person...)
- They do not operate at 100% capacity,
- They want to make \$\$.



# It's not really that bad...

- Used for last year's course
- ~25 people
- ~50gb of data each
- => \$800 cost.

We spent more on beer than on computation!

The only part that didn't work was using my own lab server  
at the very beginning.



# It's not really that bad #2...

- Using for med-memory algorithm testing
- 1 week / 100 gb of data
- Est \$250 total.
- So, if I need to compute in a hurry:
  - pay \$250
  - Or, buy a new computer (but no space! And I have to wait to get the computer!)
  - Or, block my lab from using our compute servers.
  - Or, use the HPC and wait a week for a bigmem machine.



# Renting disk space

- Elastic Block Storage lets you persist data local to EC2 instances.
- Think of it as a virtual hard drive.
- \$.10/gb/month.
- You can save, store, “snapshot”/backup, and share snapshots.





# What are **we** using it for?

- Teaching workshops and classes.
- Running our own analyses/data sets in a timely manner.
- Sharing data within the lab via EBS snapshots.
- Providing data to other people via S3.
- Automated testing on clean machines with known software install.



# Today's tutorials

1. Create (rent) a new machine from Amazon.
2. Install NCBI BLAST
3. Download & format some databases
4. Run BLAST
5. Produce an excel spreadsheet of best hits
- ...
1. Run 2-way BLAST (mouse x zfin, and vice versa)
2. Calculate reciprocal best hits
3. Produce an excel spreadsheet



# The real point...

- Renting machines
- Logging into them
- Transferring files to/from them **without** using your laptop.
- Transferring files to/from them from/to your laptop (dropbox!)



# The *other* real point...

- You can use all of this stuff at home, too.

# PLEASE DO NOT

- Try to transfer 100mb+ of files via Dropbox 😊
- Leave your EC2 machine running for no reason. (I'll explain how to check...)

