Practical Bioinformatics

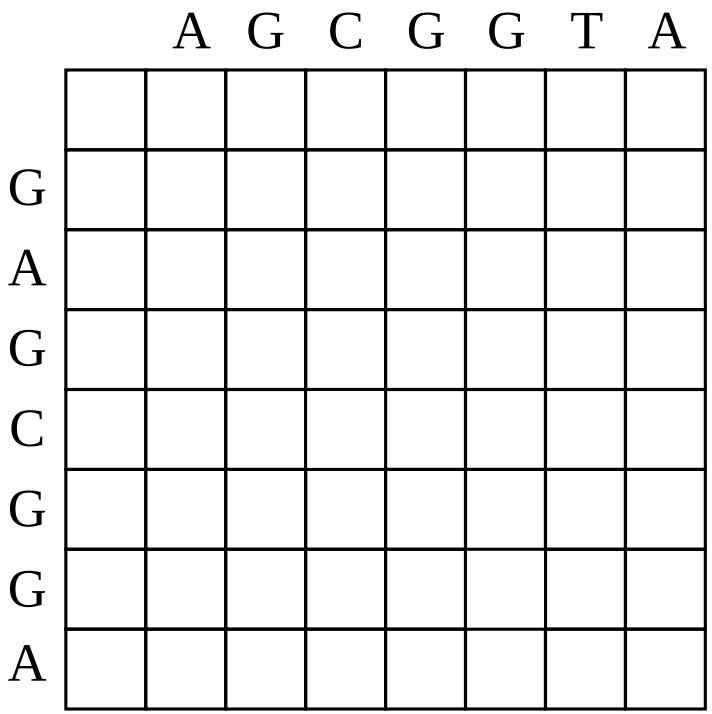
Mark Voorhies

6/3/2013

Needleman-Wunsch with g=0

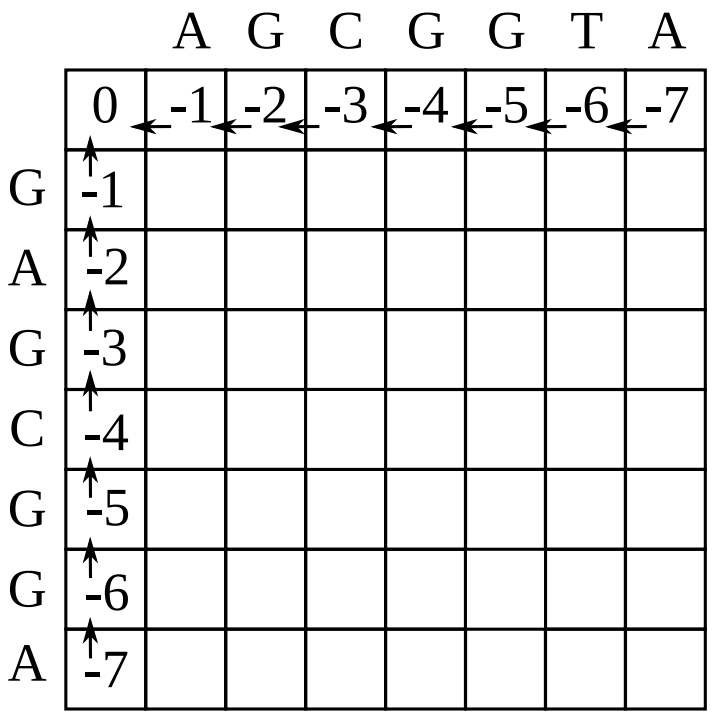
		A	G	C	G	G	T	A
G								
A								
G								
C								
G								
G								
A								

Needleman-Wunsch with g=0



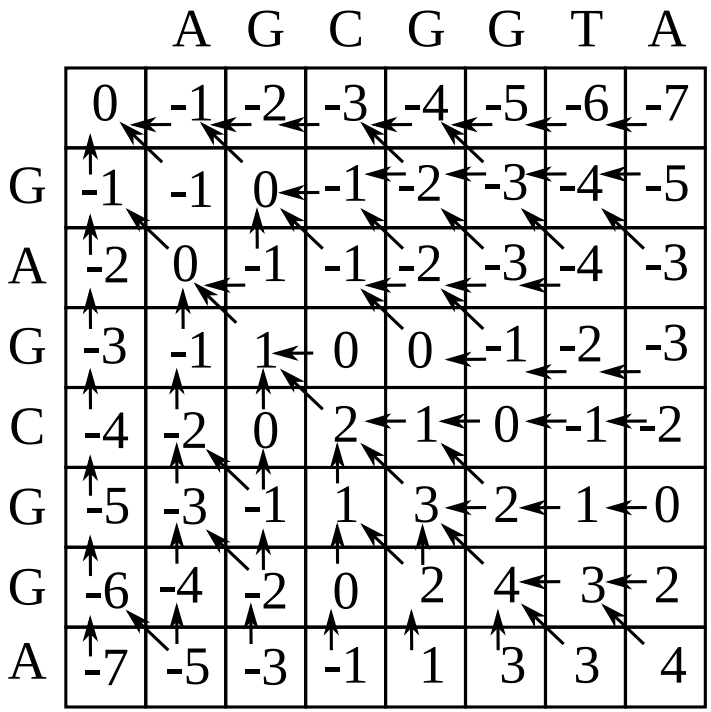

```
def nw_fill(seq1, seq2, s, e):
    # m[i][j] = best score for subalignment
    #    of seq1[:i],seq2[:j]
    m = [[]]
```

Needleman-Wunsch with g=0




```
# Top left corner
m[-1].append(0)
# Fill first row as leading gaps
for j in range(len(seq2)):
    m[-1].append(m[0][j]+e)
# Fill first column as leading gaps
for i in range(len(seq1)):
    m.append([m[i][0]+e])
```

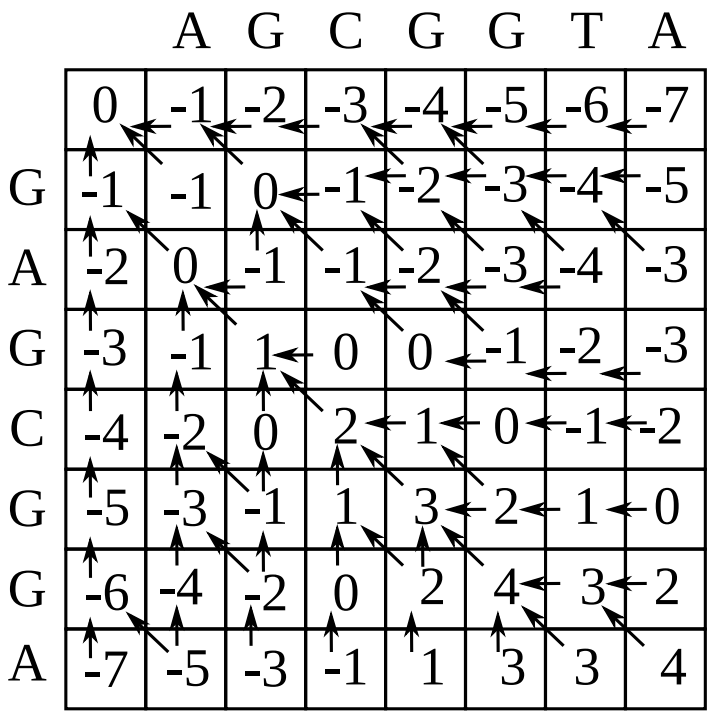
Needleman-Wunsch with g=0



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

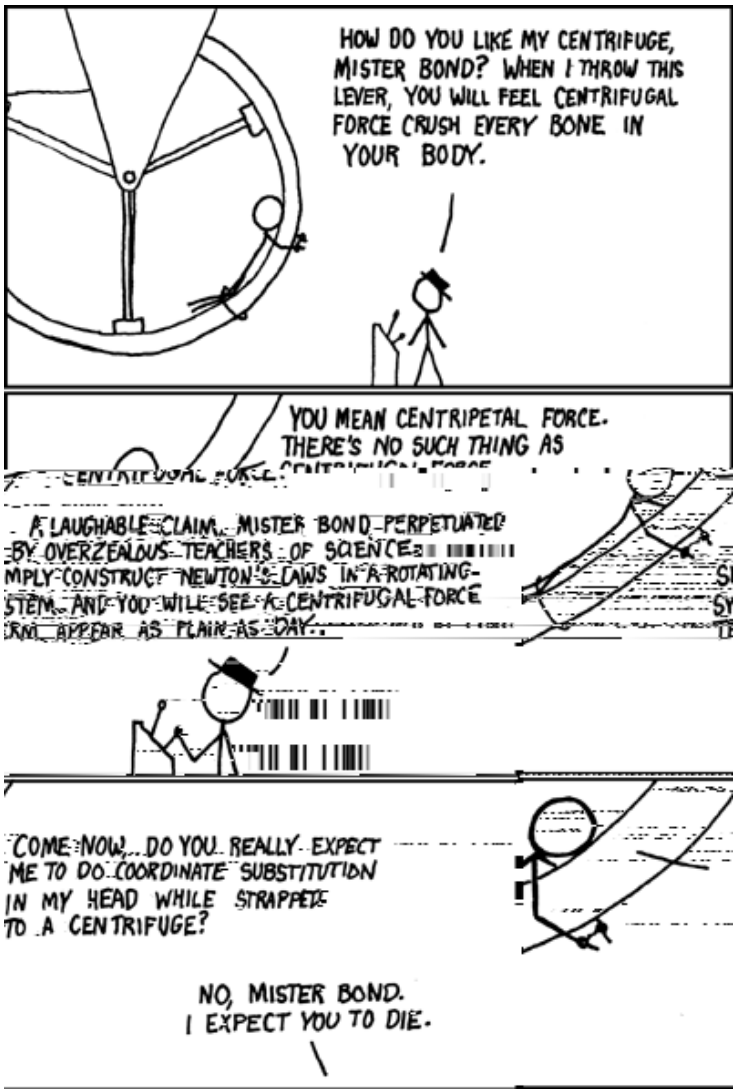
```
for i in range(len(seq1)):
    for j in range(len(seq2)):
        # Score for aligning seq1[i] with seq2[j]
        match = m[i][j]+s[seq1[i]][seq2[j]]
        # Score for aligning seq1[i] with a gap
        hgap = m[i+1][j]+e
        # Score for aligning seq2[i] with a gap
        vgap = m[i][j+1]+e

        best = max(match, vgap, hgap)
        m[-1].append(best)
```

Needleman-Wunsch with g=0



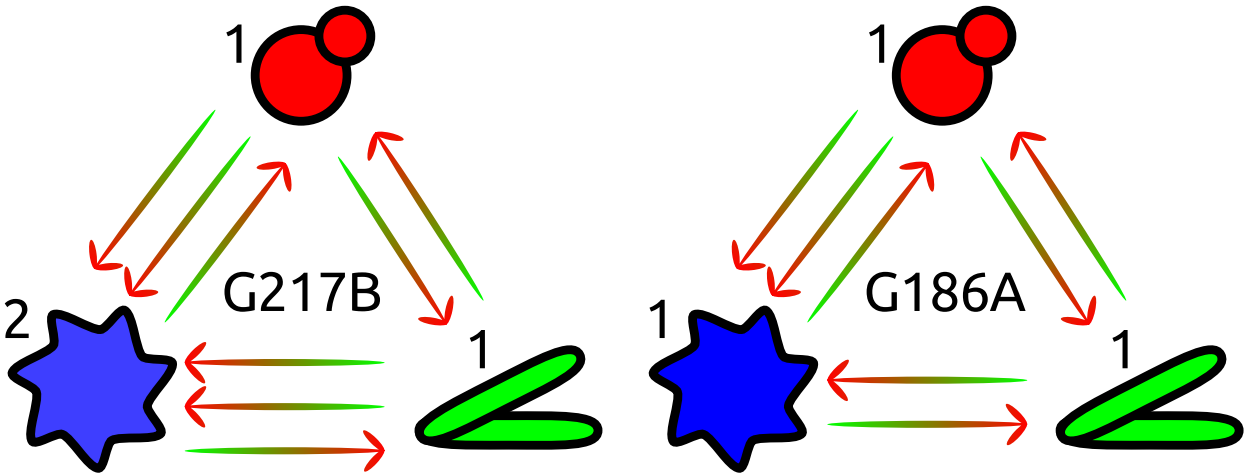

```
# Start at bottom right corner
curpos = (len(seq1),len(seq2))
aligned1 = ""
aligned2 = ""

exitFlag = False
for i in range(len(seq1)+len(seq2)):
    plist = p[curpos[0]][curpos[1]]
    if(plist is None):
        exitFlag = True
        break
    nextpos = plist[0]
    # Check for vgap
    if(nextpos[0] == curpos[0]):
        aligned1 = "-"+aligned1
    else:
        aligned1 = seq1[nextpos[0]]+aligned1
    # Check for hgap
    if(nextpos[1] == curpos[1]):
        aligned2 = "-"+aligned2
    else:
        aligned2 = seq2[nextpos[1]]+aligned2

    curpos = nextpos

if(exitFlag == False):
    print "WARNING: Unexpected exit from traceback"
```

Change of Coordinates



http://xkcd.com/123/

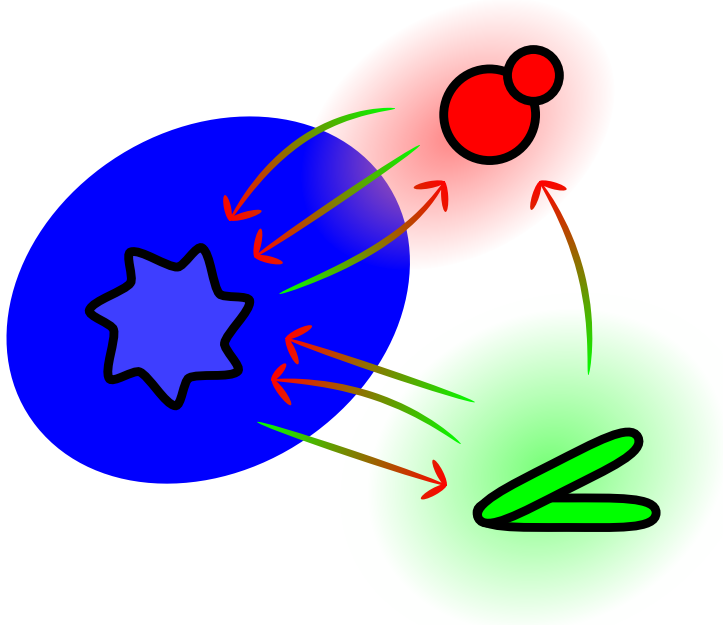
Phase-specific gene expression in *Histoplasma capsulatum*



Diane Inglis

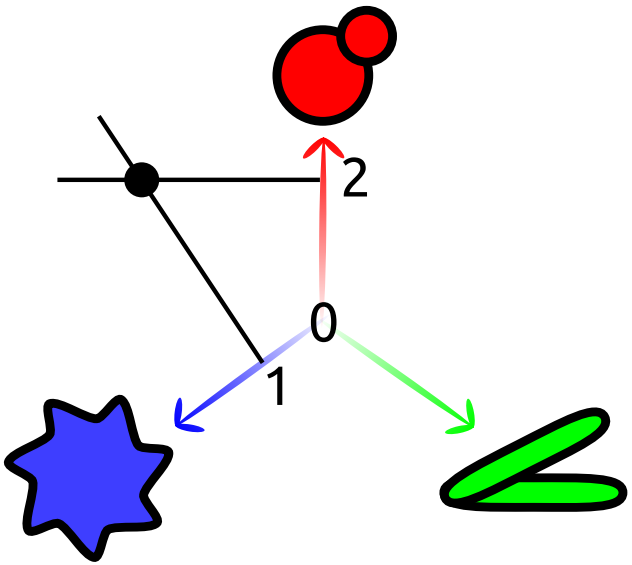
Phase-specific gene expression in

Phase-specific gene expression in *Histoplasma capsulatum*

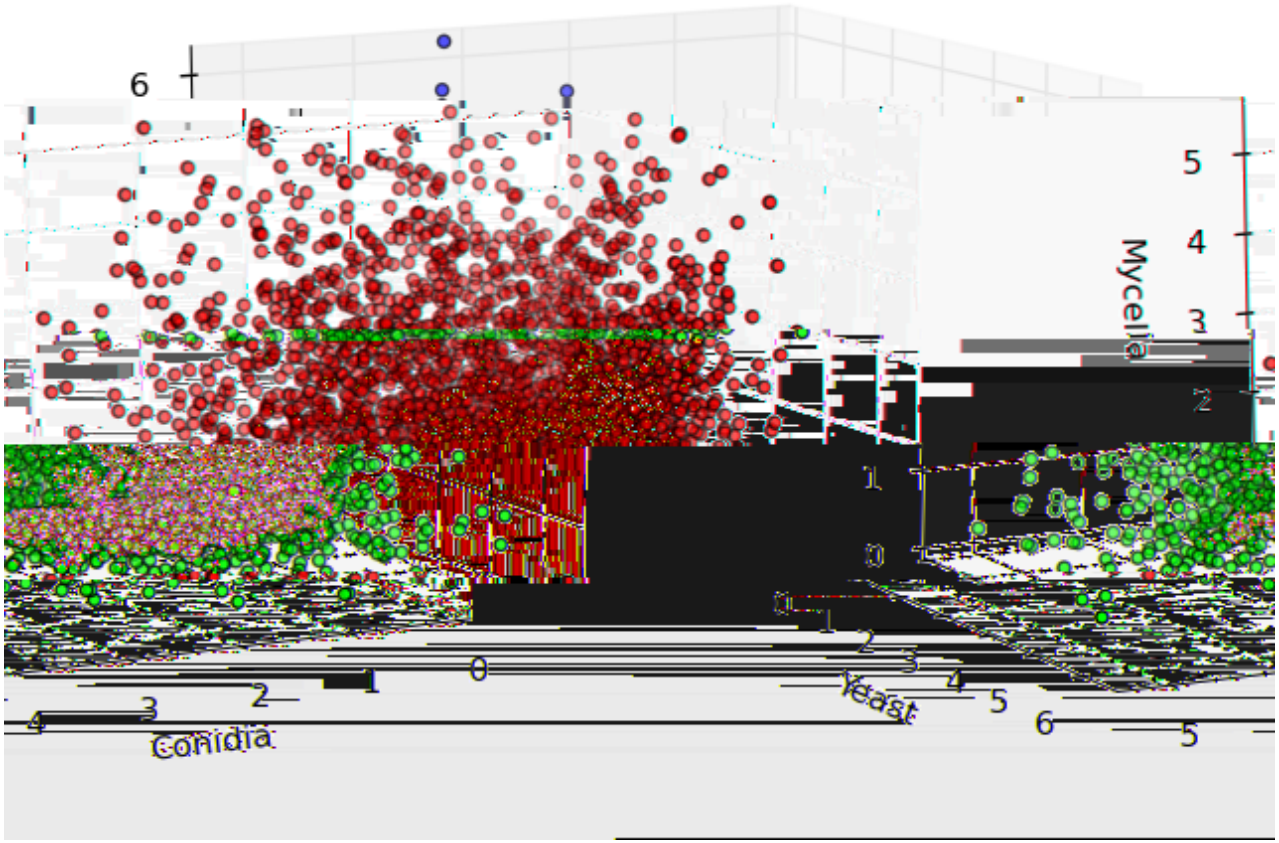


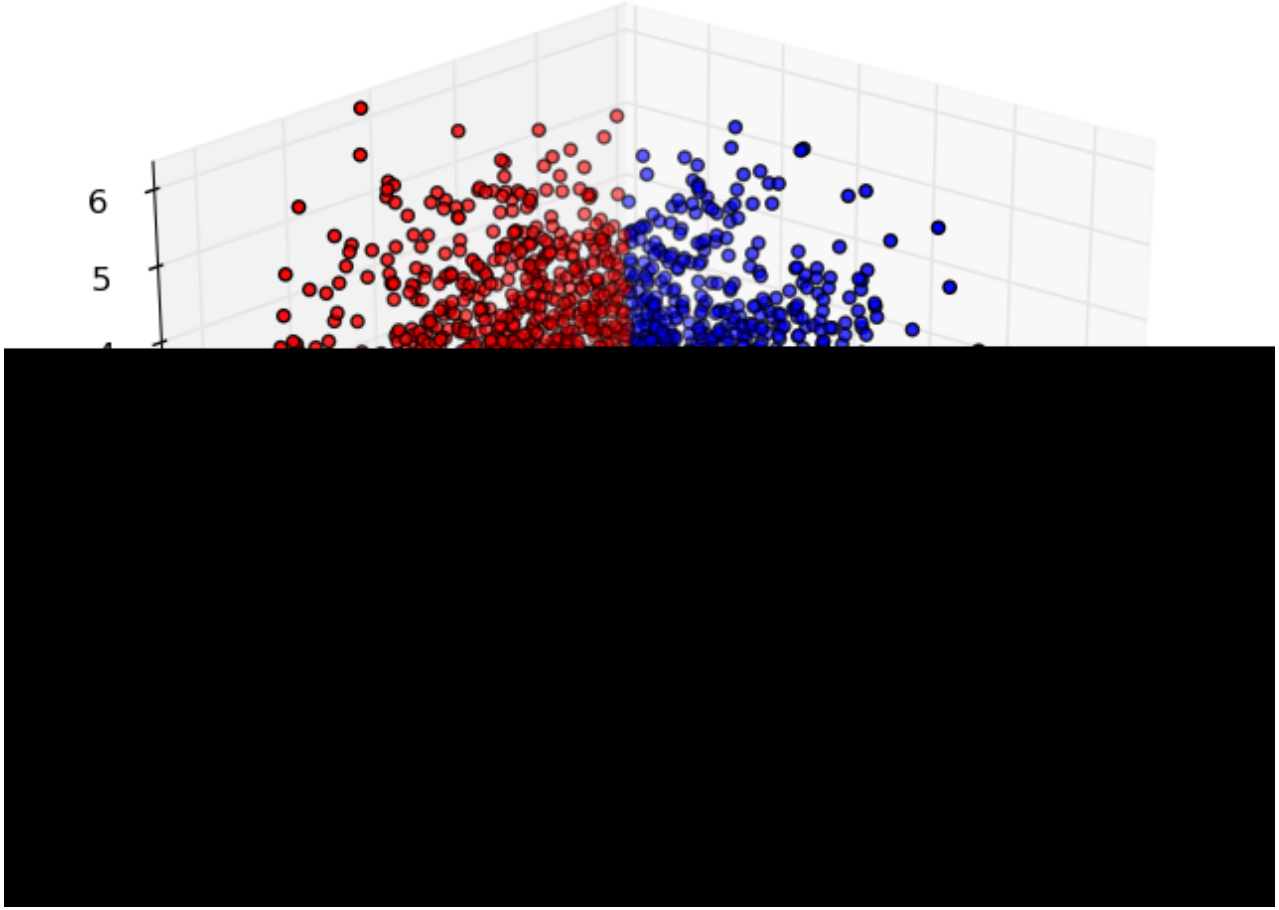
Diane Inglis

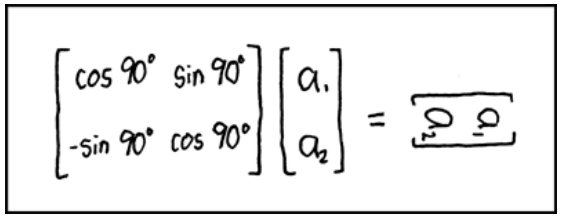
Phase-specific gene expression in *Histoplasma capsulatum*



Diane Inglis


6
5
4
3
2
1
0
Mycelia
Yeast
Conidia


6
5

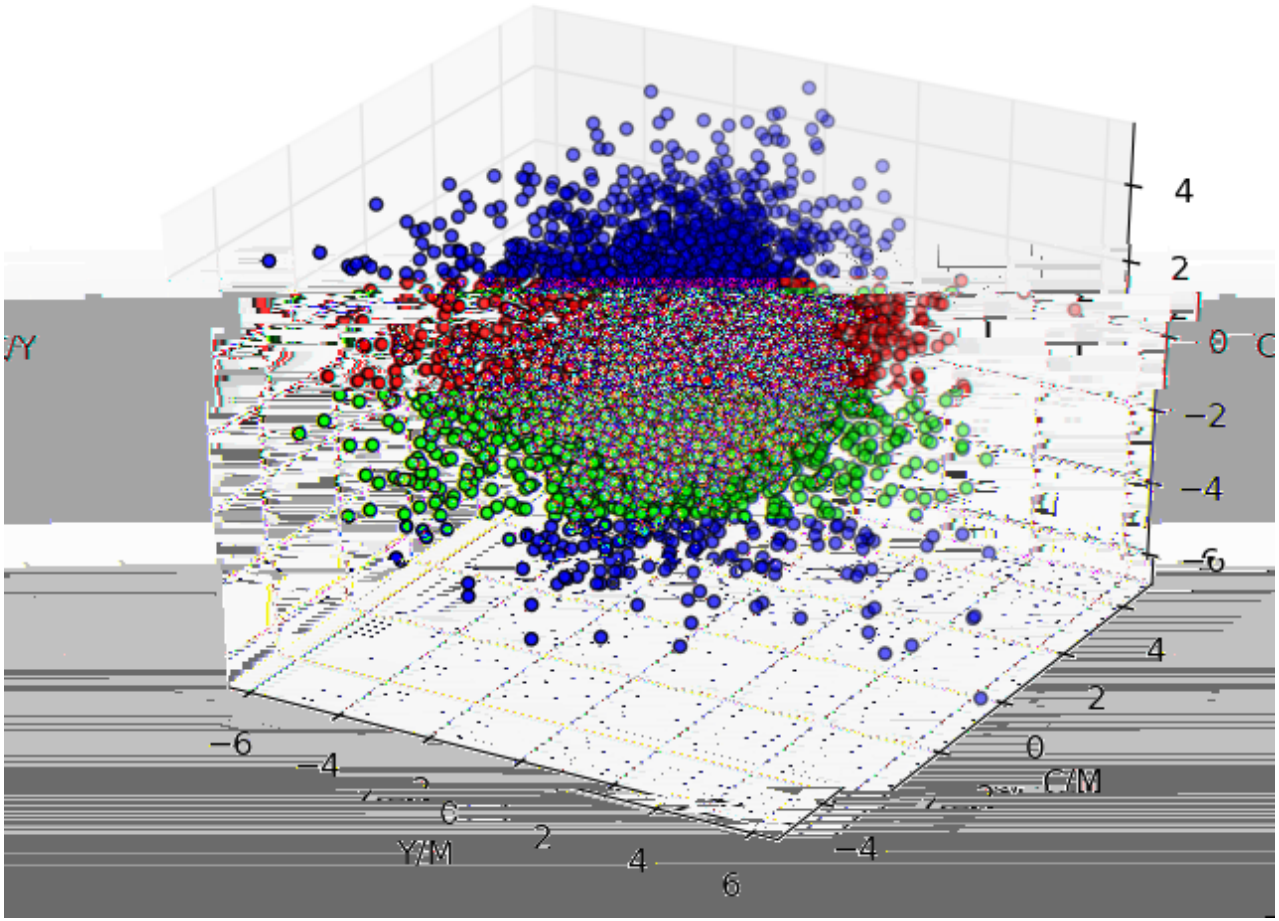


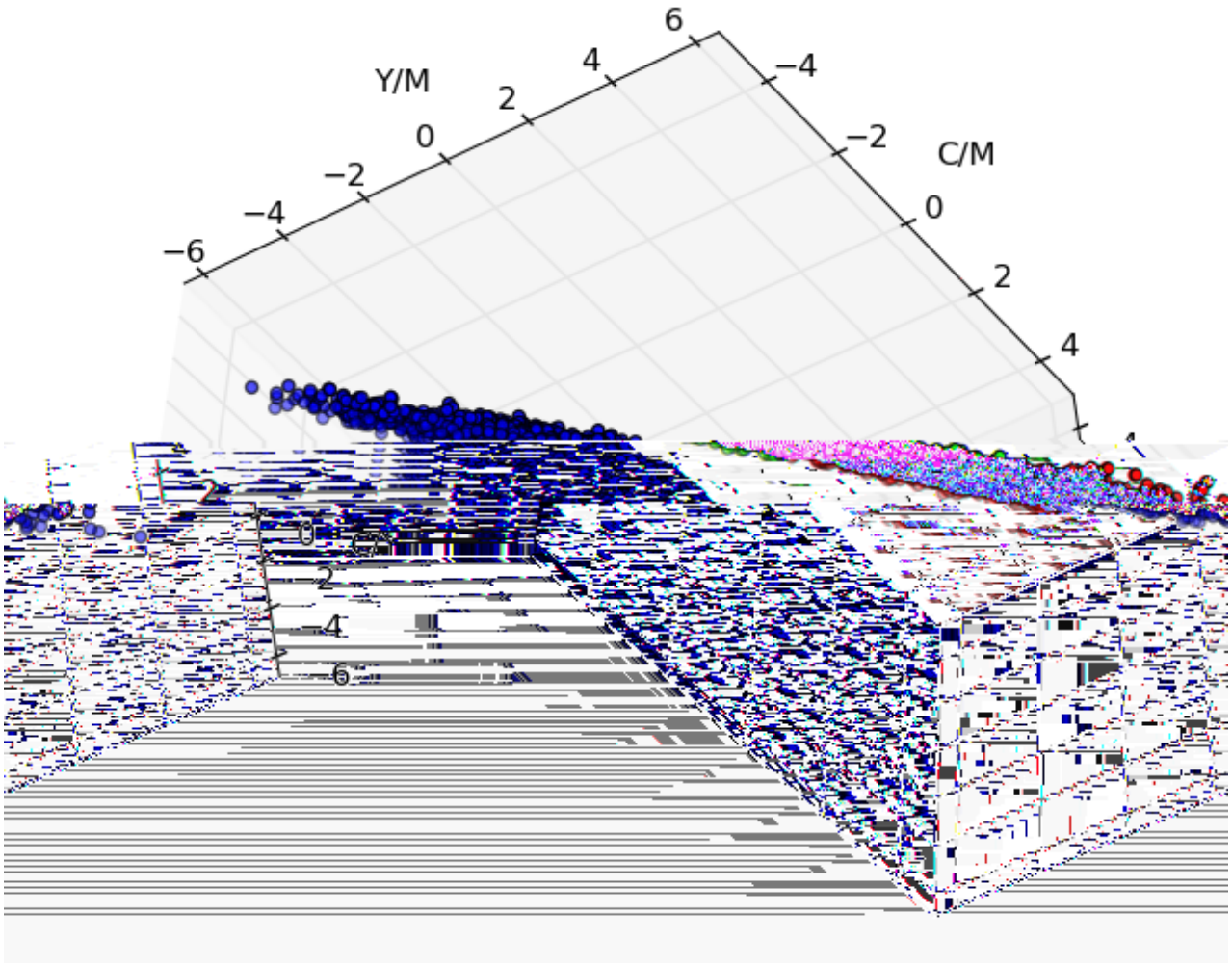
$$E = \begin{pmatrix} C_i & Y_i & M_i \\ \vdots & \vdots & \vdots \end{pmatrix}$$

$$D = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

$$R = ED$$

http://xkcd.com/184/

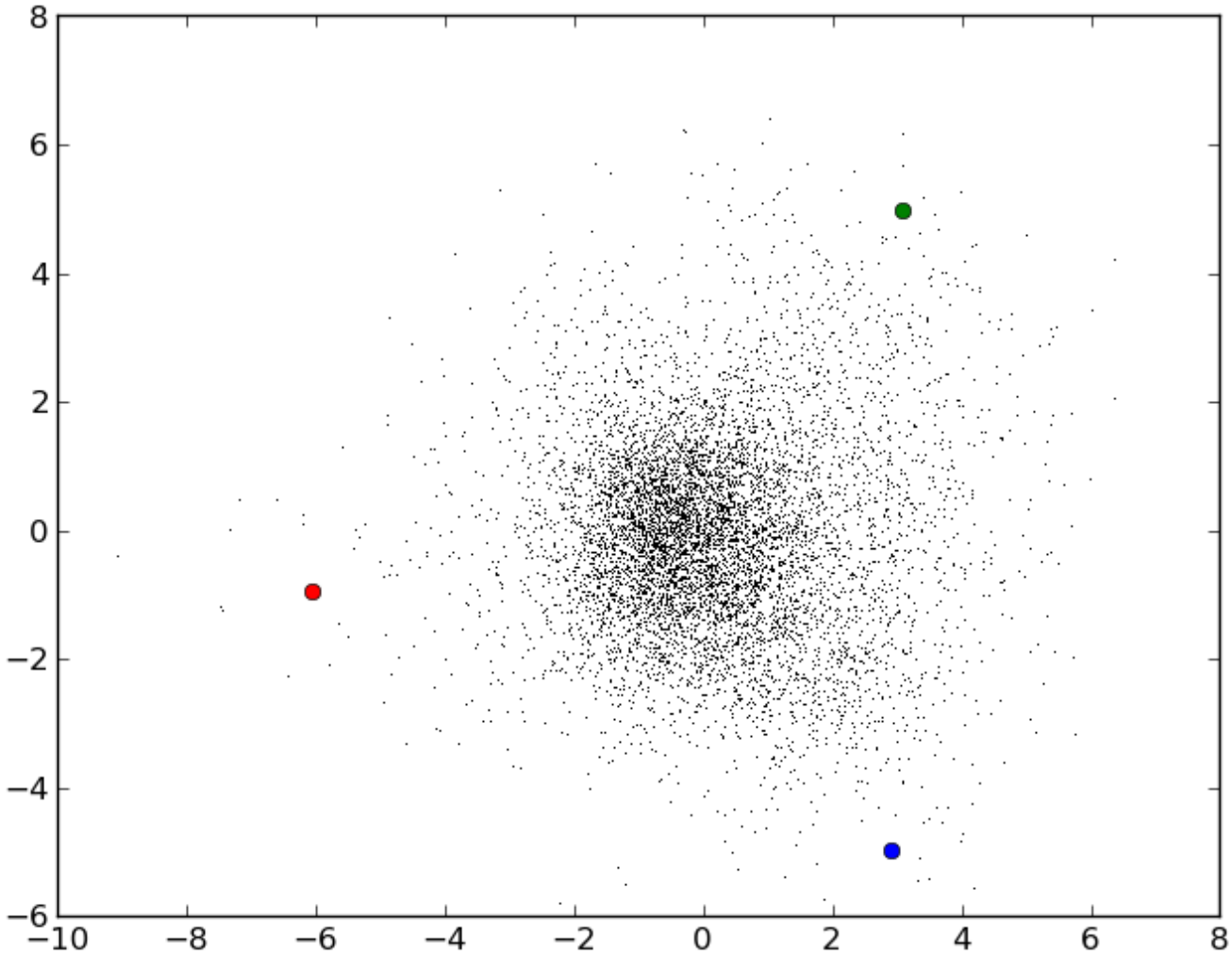

4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
Y/M
C/M


Y/M
6
4
2
0
−2
−4
−6
C/M
−4
−2
0
2
4

$$
\begin{aligned}
R &= U\Sigma V^T \\
\Sigma &= diag\left(140, 138, 2.53 \quad 10^{-13}\right) \\
U' &= U_{0,1} \\
P &= RU' \\
&= EDU' \\
&= EU'' \\
U'' &= DU' \\
&= \begin{pmatrix} 0.81 & 1.16 \\ 1.41 & 0.12 \\ 0.60 & 1.28 \end{pmatrix}
\end{aligned}
$$

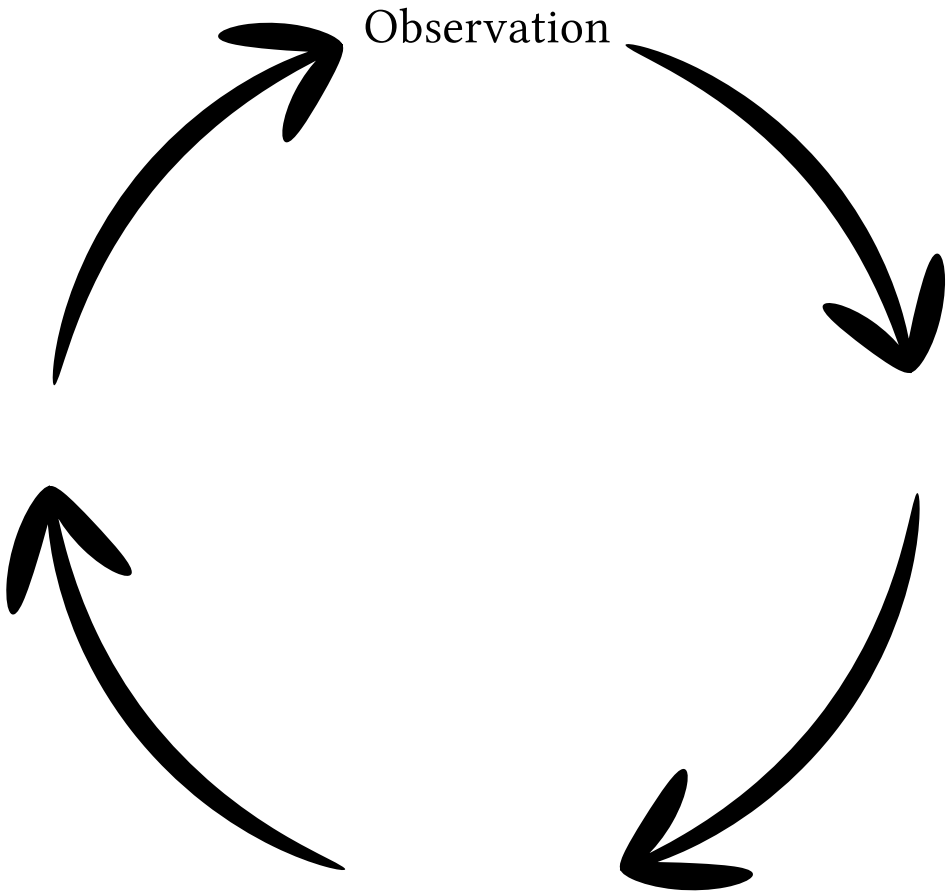
$$
\begin{aligned}
U'' &= \begin{pmatrix} 0.81 & 1.16 \\ 1.41 & 0.12 \\ 0.60 & 1.28 \end{pmatrix} \\
T &= \begin{pmatrix} \sqrt{\frac{3}{2}} & .5 \\ 0 & 1 \\ \sqrt{\frac{3}{2}} & .5 \end{pmatrix} \\
\frac{U''_0 \quad U''_1}{kU''_0 \quad U''_1 k} &= \begin{pmatrix} \frac{-1}{\sqrt{3}} \\ \frac{-1}{\sqrt{3}} \\ \frac{-1}{\sqrt{3}} \end{pmatrix} = \frac{T_0 \quad T_1}{kT_0 \quad T_1 k}
\end{aligned}
$$

SVD matches YMC transform

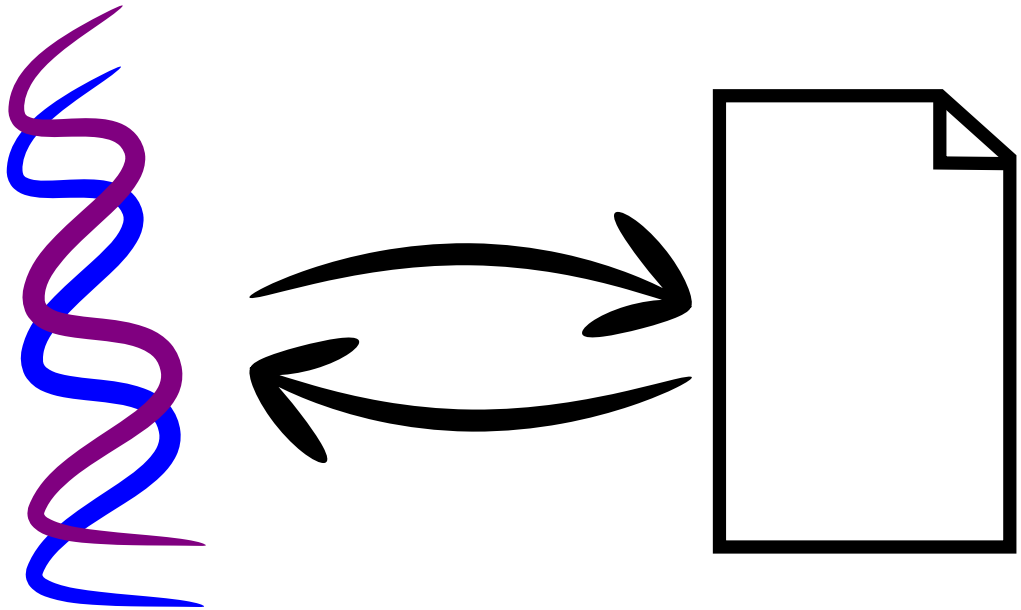


RTFM

- For tools:
  - Read the manual
  - Read the paper
- Good general references:
  - The O'Reilly BLAST book
  - Durbin, Eddy, Krogh, and Mitcheson (HMMs)
  - Numerical Recipes
  - Branden & Tooze

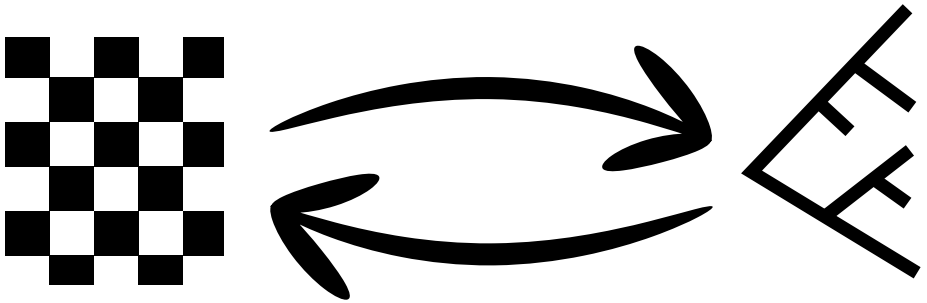

Observation

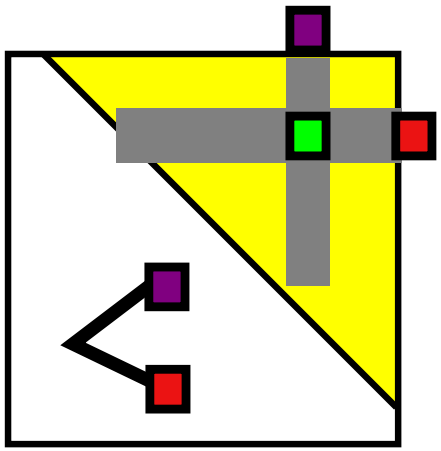
Every object should have an isomorphism to a file

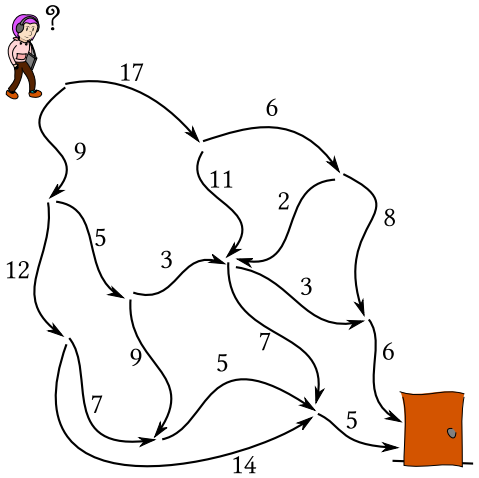


- Export, audit, edit, and import *independent* of a given program.
- Standard file formats for portability.
- Don't be afraid to look inside and hack on *your* data files.

A few techniques can solve many problems

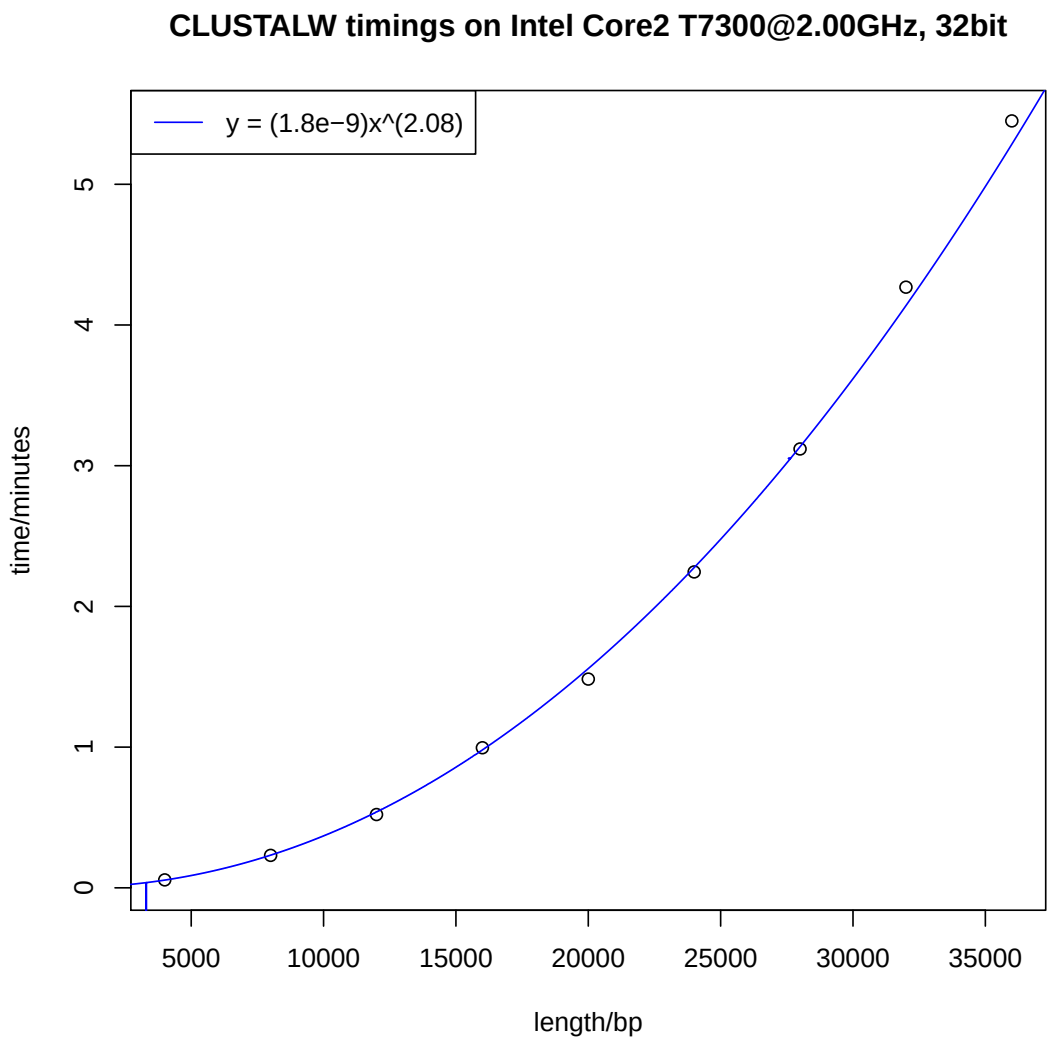




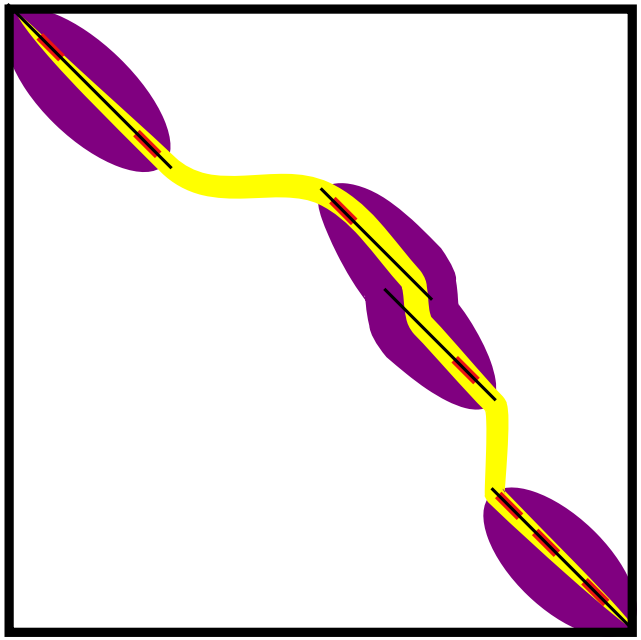


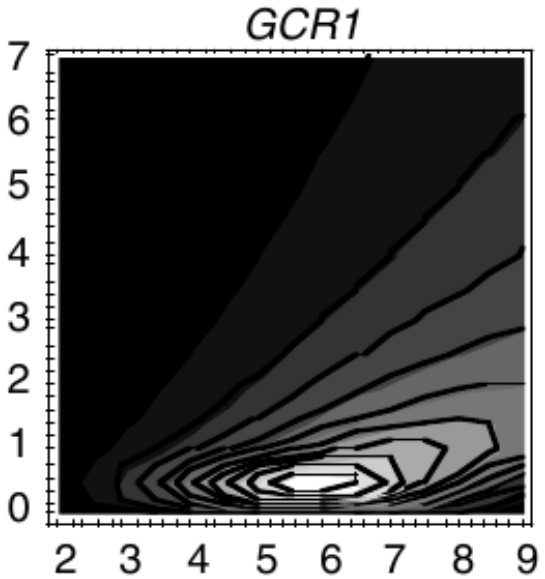
Iteration, clustering, dynamic programming, ...

Run times are predictable and measurable



Heuristics and stochastic sampling for hard problems



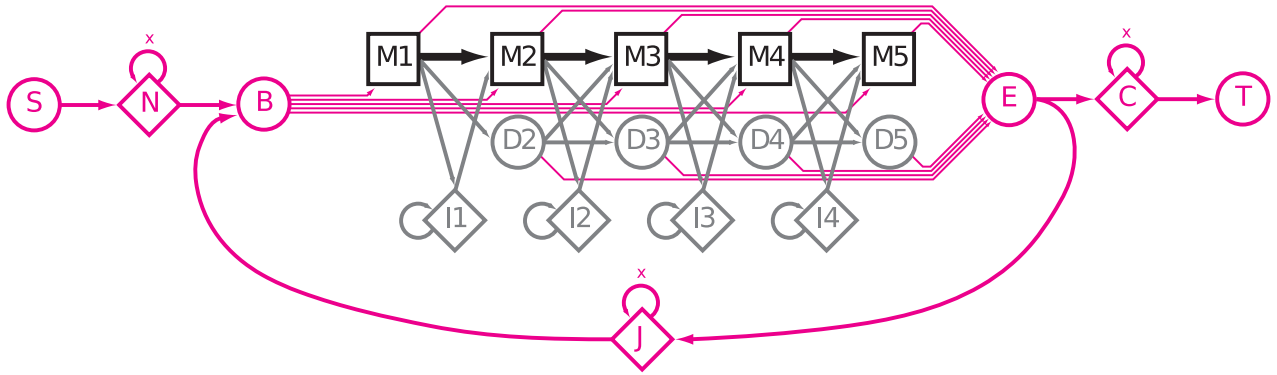


BLAST, HMMer3, MrBayes, ...

Evolution is a rich source of information

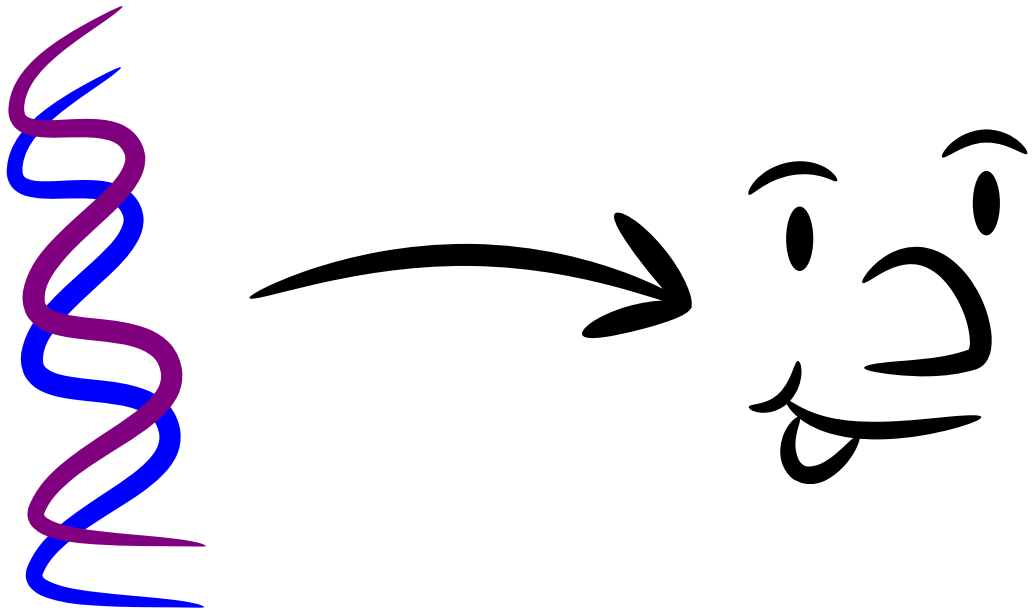
- Infer homology from sequence similarity
- More sequences provide more information

HMMs capture position and gap information

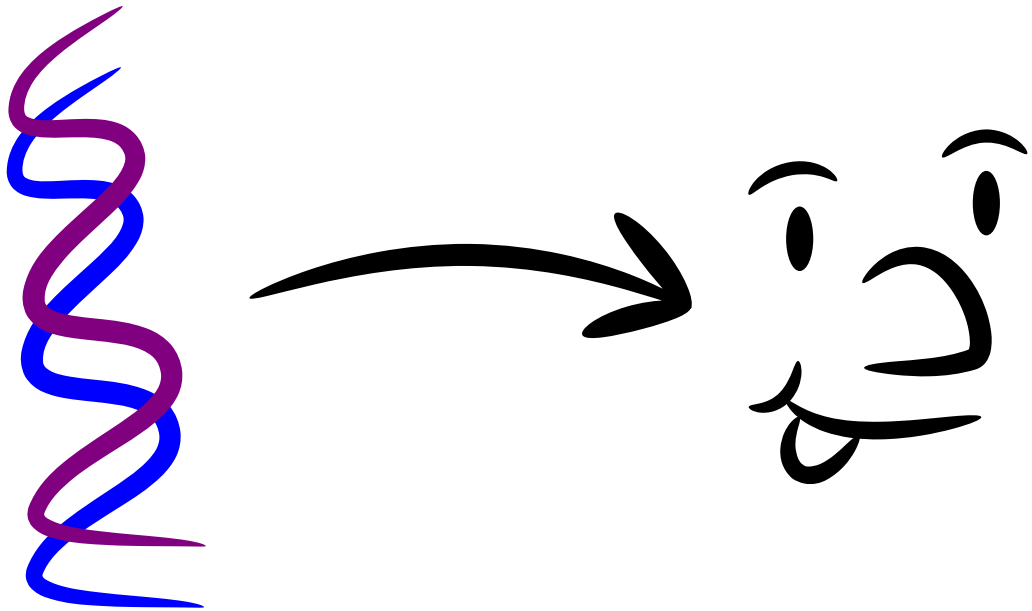


(Image from Sean Eddy, PLoS Comp. Biol. 4:e1000069)

Phenotype is more diverse than Genotype

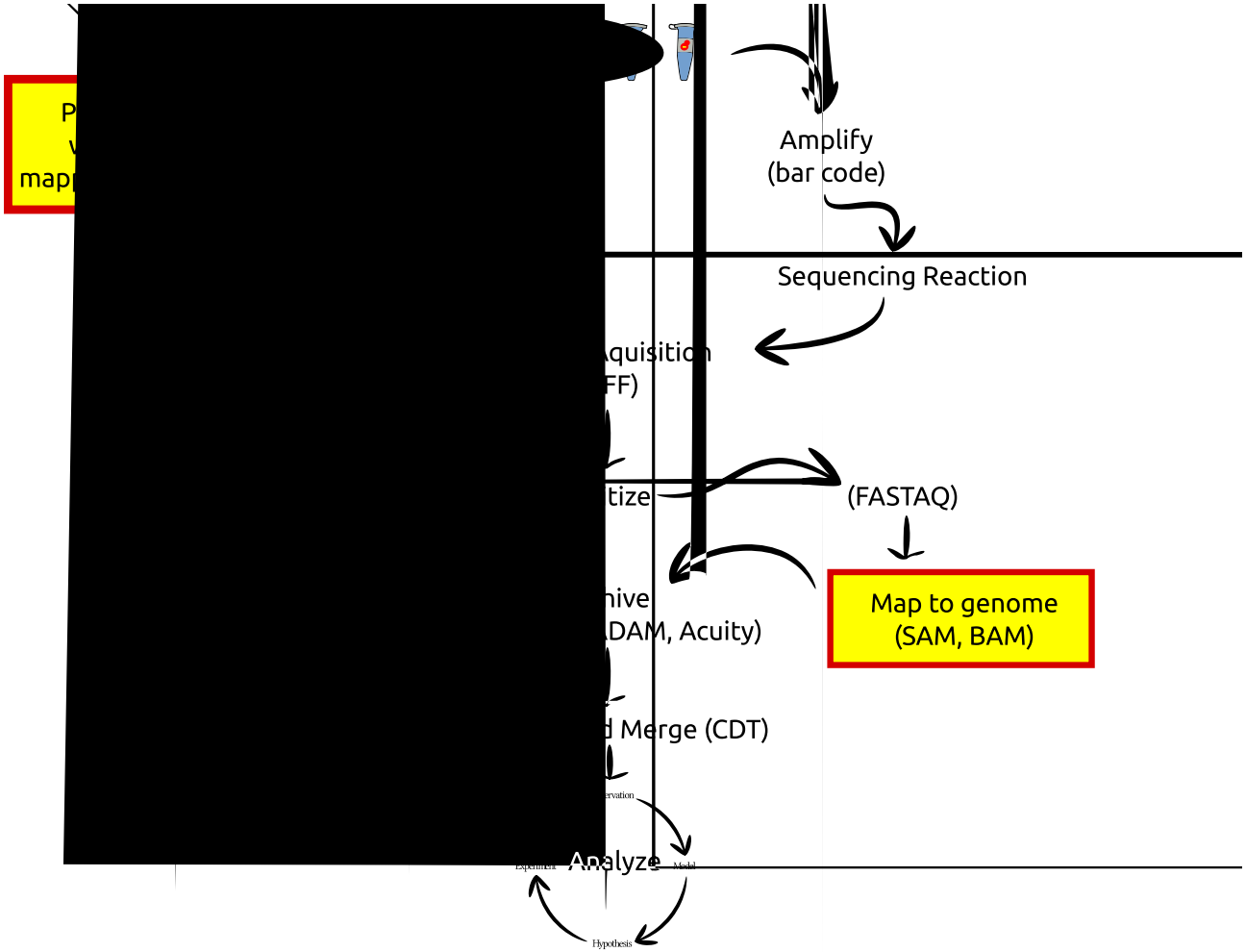


Phenotype is more diverse than Genotype

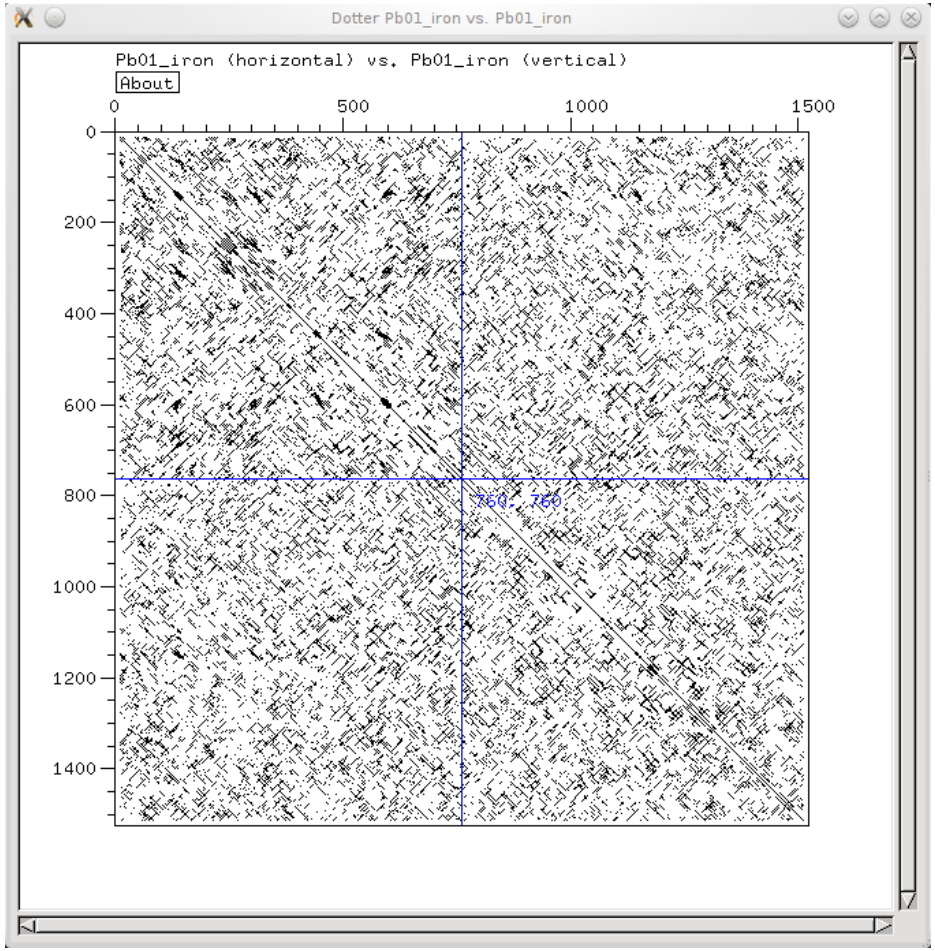


- Make sure you know what you are measuring
- Nucleic acid sequences are especially easy to address
- Many phenotypes can be analyzed by common numerical methods

Sequencing methods sidestep (some) design-time decisions



Start from an unbiased view



Tools should support aggregation and annotation

Groovy Packages

- numpy
- matplotlib
- scipy
- networkx
- rpy
- pandas
- Pycluster
- MySQLdb
- scikits.image
- scikits.learn

Science is a Conversation

- Follow computational methods as they evolve (Web of Science, PubMed RSS...)

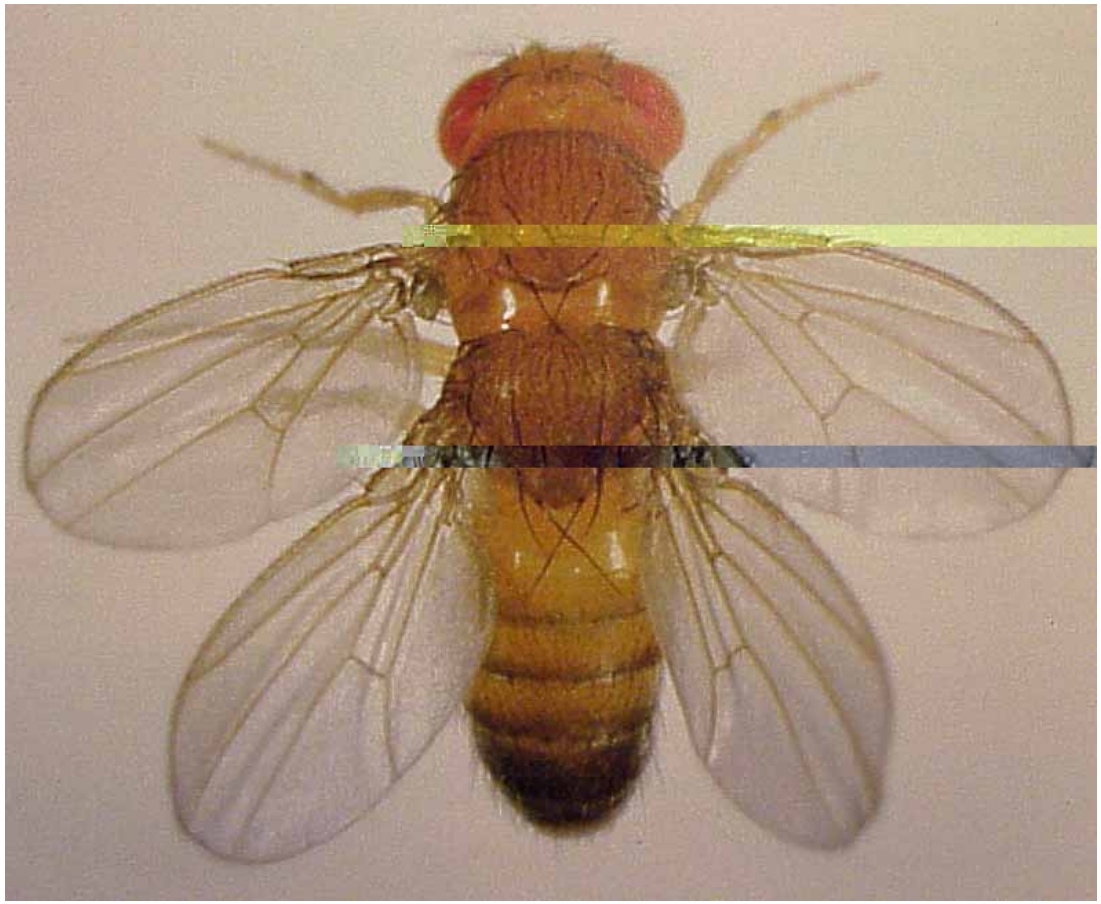
Science is a Conversation

- Follow computational methods as they evolve (Web of Science, PubMed RSS...)
- As a reviewer, insist on availability of source code

Science is a Conversation

- Follow computational methods as they evolve (Web of Science, PubMed RSS...)
- As a reviewer, insist on availability of source code
- Draw on your classmates' expertise

We understand systems by breaking them



Source: Peter A. Lawrence via http://www.bio.davidson.edu/courses/molbio/ubx/ubx.html