Practical Bioinformatics

Mark Voorhies

5/6/2011

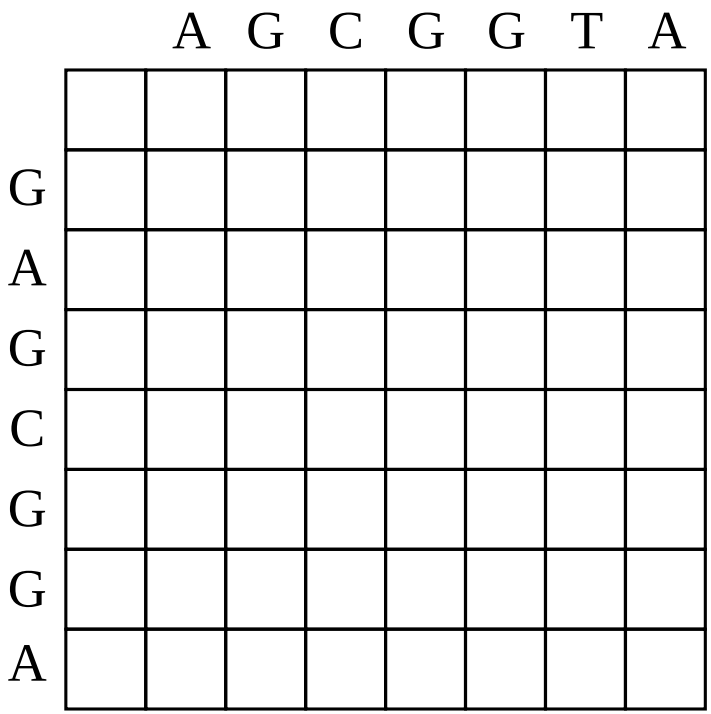
		C	O	E	L	A	C	A	N	T	H
	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8	← -9	← -10
P	↑ -1	↖ -1	↖ -2	↖ -3	↖ -4	↖ -5	↖ -6	↖ -7	↖ -8	↖ -9	↖ -10
E	↑ -2	↖ -2	↖ -2	↖ -1	← -0	← -3	← -4	← -5	← -6	← -7	← -8
L	↑ -3	↖ -3	↖ -3	← -2	↖ -2	← -1	← -2	← -3	← -4	← -5	← -6
I	↑ -4	↖ -4	↑ -4	↑ -3	↑ -1	↖ -1	↖ -2	↖ -1	↖ -4	↖ -5	↖ -6
C	↑ -5	↖ -3	← -4	↑ -4	↑ -2	↖ -2	↖ -0	← -1	← -2	← -3	← -4
A	↑ -6	↑ -4	↖ -4	↖ -5	↑ -3	↖ -1	↑ -1	↖ -1	← -0	← -1	← -2
N	↑ -7	↑ -5	↖ -5	↖ -5	↑ -4	↑ -2	↖ -2	← -0	↖ -2	← -1	← -0

-PELICAN--
COELACANTH

		C	O	E	L	A	C	A	N	T	H
	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8	← -9	← -10
P	↑ -1	↖ -1	↖ -2	↖ -3	↖ -4	↖ -5	↖ -6	↖ -7	↖ -8	↖ -9	↖ -10
E	↑ -2	↖ -2	↖ -2	↖ -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8
L	↑ -3	↖ -3	↖ -3	← -2	↖ 0	← -1	← -2	← -3	← -4	← -5	← -6
I	↑ -4	↖ -4	↑ -4	↑ -3	↑ -1	↖ -1	↖ -2	↖ -1	↖ -4	↖ -5	↖ -6
C	↑ -5	↖ -3	← -4	↑ -4	↑ -2	↖ -2	↖ 0	← -1	← -2	← -3	← -4
A	↑ -6	↑ -4	↖ -4	↖ -5	↑ -3	↖ -1	↑ -1	↖ 1	← -0	← -1	← -2
N	↑ -7	↑ -5	↖ -5	↖ -5	↑ -4	↑ -2	↖ -2	← -0	↖ -2	← -1	← -0

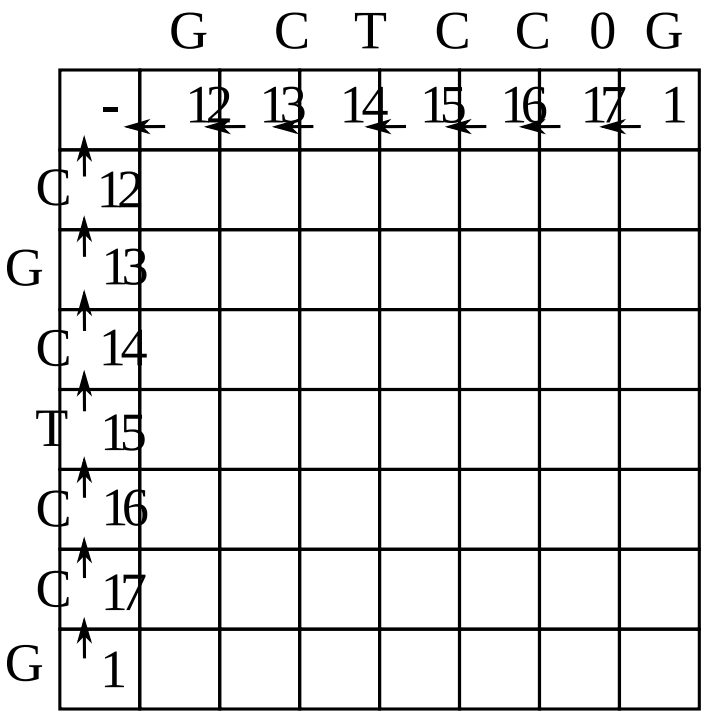
-PELICAN--
COELACANTH

dp.nw_ll: Needleman-Wunsch with g=0



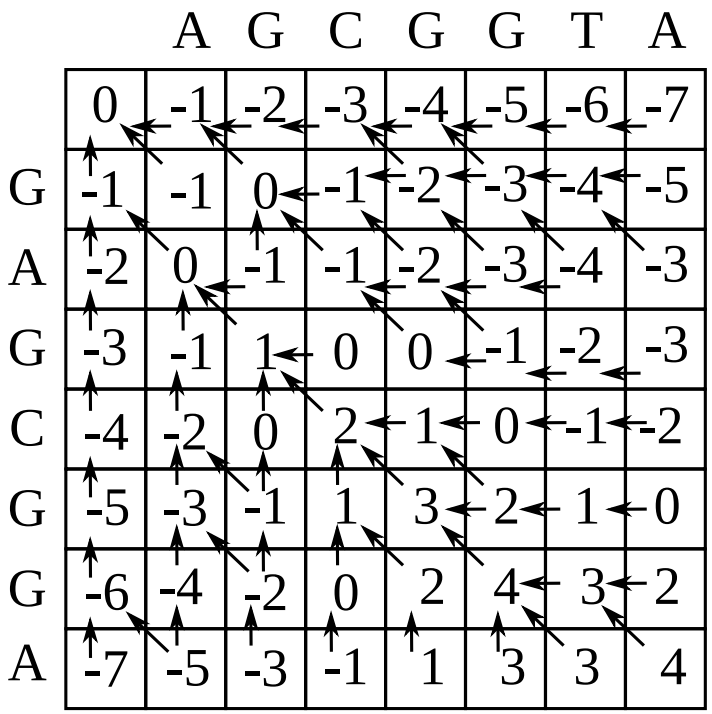

```
def nw_fill(seq1, seq2, s, e):
    # m[i][j] = best score for subalignment
    #   of seq1[:i+1],seq2[:j+1]
    m = [[0]]

    # p[i][j] = list of best paths from
    #   m[i][j], or None if the alignment
    #   stops here.
    p = [[None]]
```

dp.nw_ll: Needleman-Wunsch with g=0



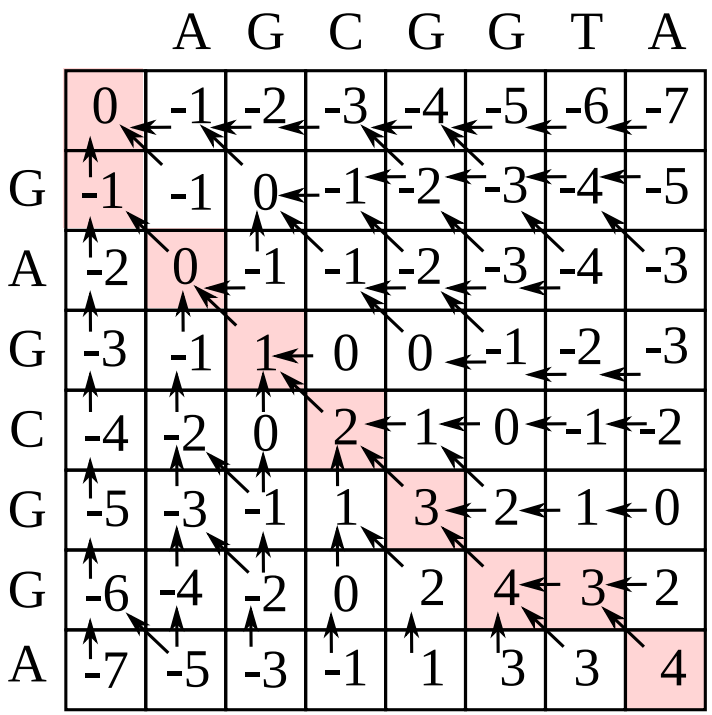

```
# Fill first row as leading gaps
for j in range(len(seq2)):
    m[-1].append(m[0][j]+e)
    p[-1].append([(0,j)])
```

dp.nw_ll: Needleman-Wunsch with g=0



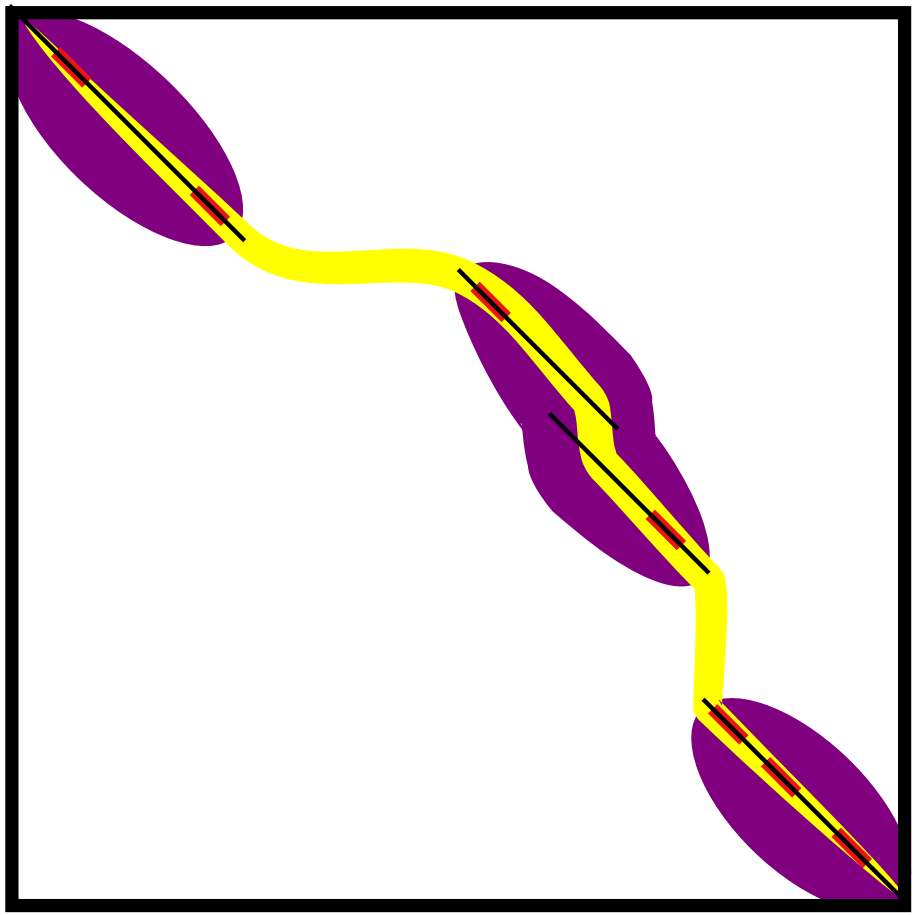

```
for i in range(len(seq1)):
    # First column is leading gaps
    m.append([m[i][0]+e])
    p.append([[(i,0)]])
    for j in range(len(seq2)):
        # Score for aligning seq1[i] with seq2[j]
        match = m[i][j]+s[seq1[i]][seq2[j]]
        # Score for aligning seq1[i] with a gap
        hgap = 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)
        p[-1].append([])

        if(match == best):
            p[-1][-1].append((i,j))
        if(hgap == best):
            p[-1][-1].append((i+1,j))
        if(vgap == best):
            p[-1][-1].append((i,j+1))
```

dp.nw_traceback: Needleman-Wunsch with g=0

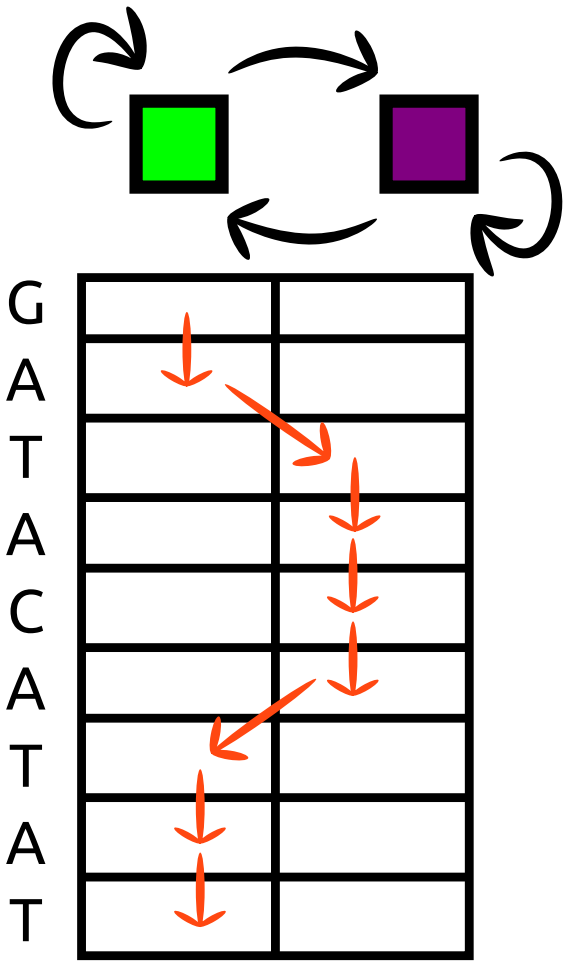



```
# Start at bottom right corner
curpos = (len(seq1),len(seq2))
aligned1 = ""
aligned2 = ""

plist = p[curpos[0]][curpos[1]]
while(plist is not None):
    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
    plist = p[curpos[0]][curpos[1]]
```

Gapped BLAST: Merge neighboring HSPs

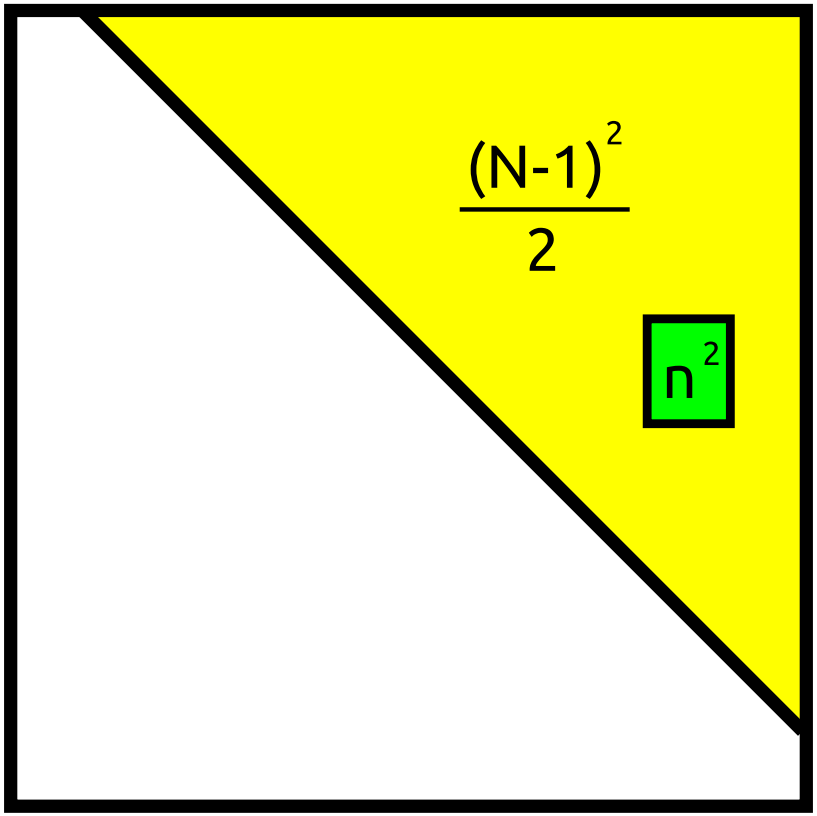


The Viterbi algorithm: Alignment

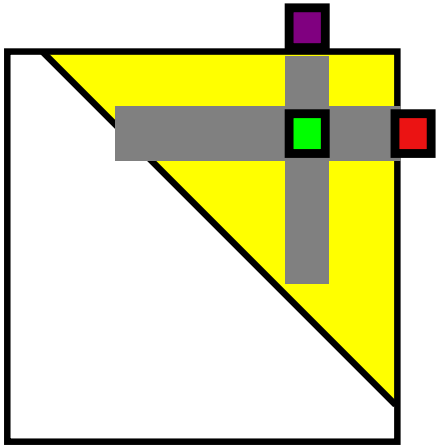


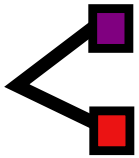
- Dynamic programming, like Smith-Waterman
- Sums *best* log probabilities of emissions and transitions (*i.e.*, multiplying independent probabilities)
- Result is most likely annotation of the target with hidden states

Measure all pairwise distances by dynamic programmi

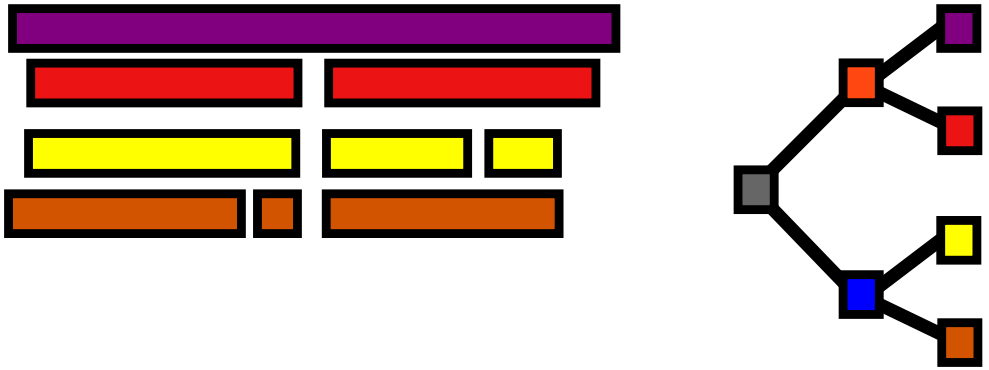


Generate a guide tree by UPGMA





Progressive alignment following the guide tree



Groovy Packages

- Python modules
  - numpy (numeric)
  - matplotlib
  - scipy
  - rpy
  - Pycluster (Biopython)
  - MySQLdb
- Distributions
  - Enthought Python Distribution
  - Ubuntu Linux

Science is a Conversation

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

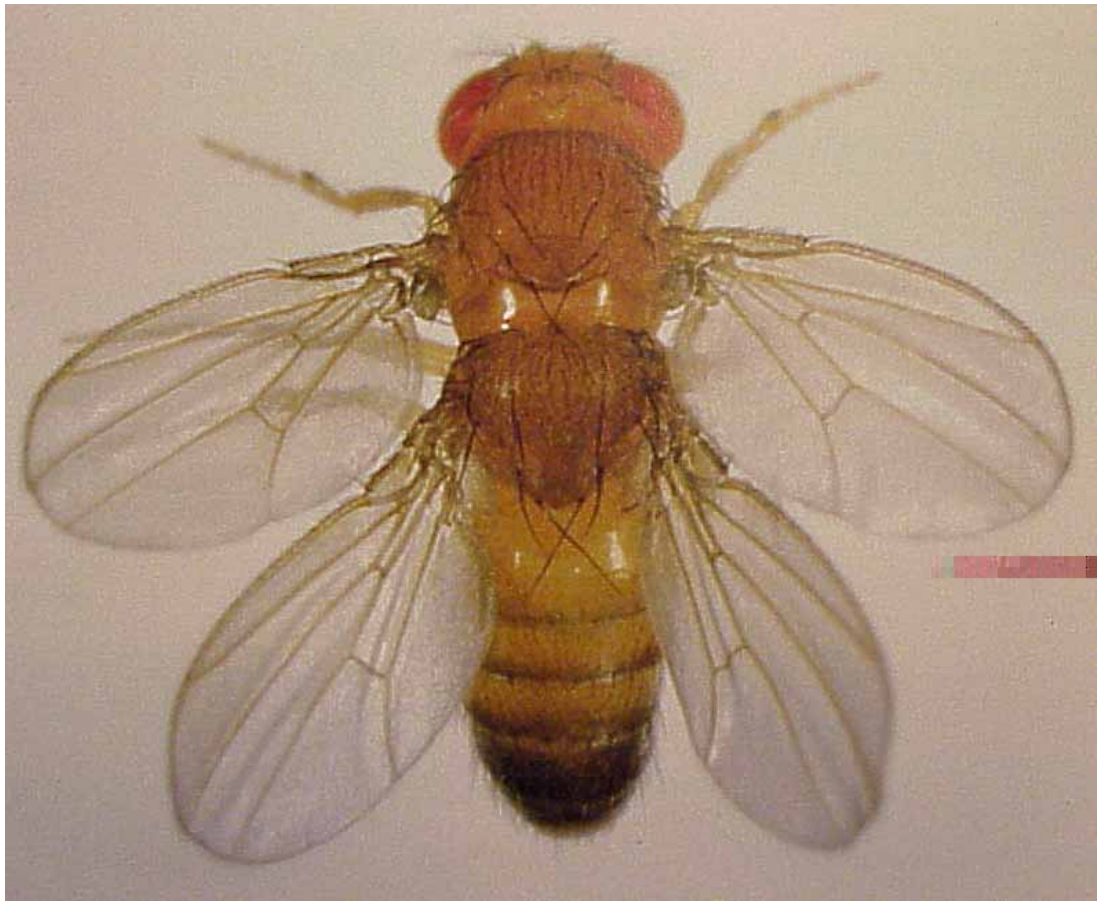
Science is a Conversation

- 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