Gap Scores

Stephen Altschul

National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health

Length-Dependent Gap Scores

The Biological Issue: A single mutational event can insert or delete multiple letters at one time. Therefore, an alignment containing a single gap of length 6, for example, may be more biologically plausible than one containing three gaps of length 1. Assigning a score of g to each indel does not capture this fact.

Fitch, W.M. & Smith, T.F. (1983) *Proc. Natl. Acad. Sci. USA* **80**:1382-1386.

Definitional change: A *gap* is a run of nulls in one sequence aligned with letters in the other.

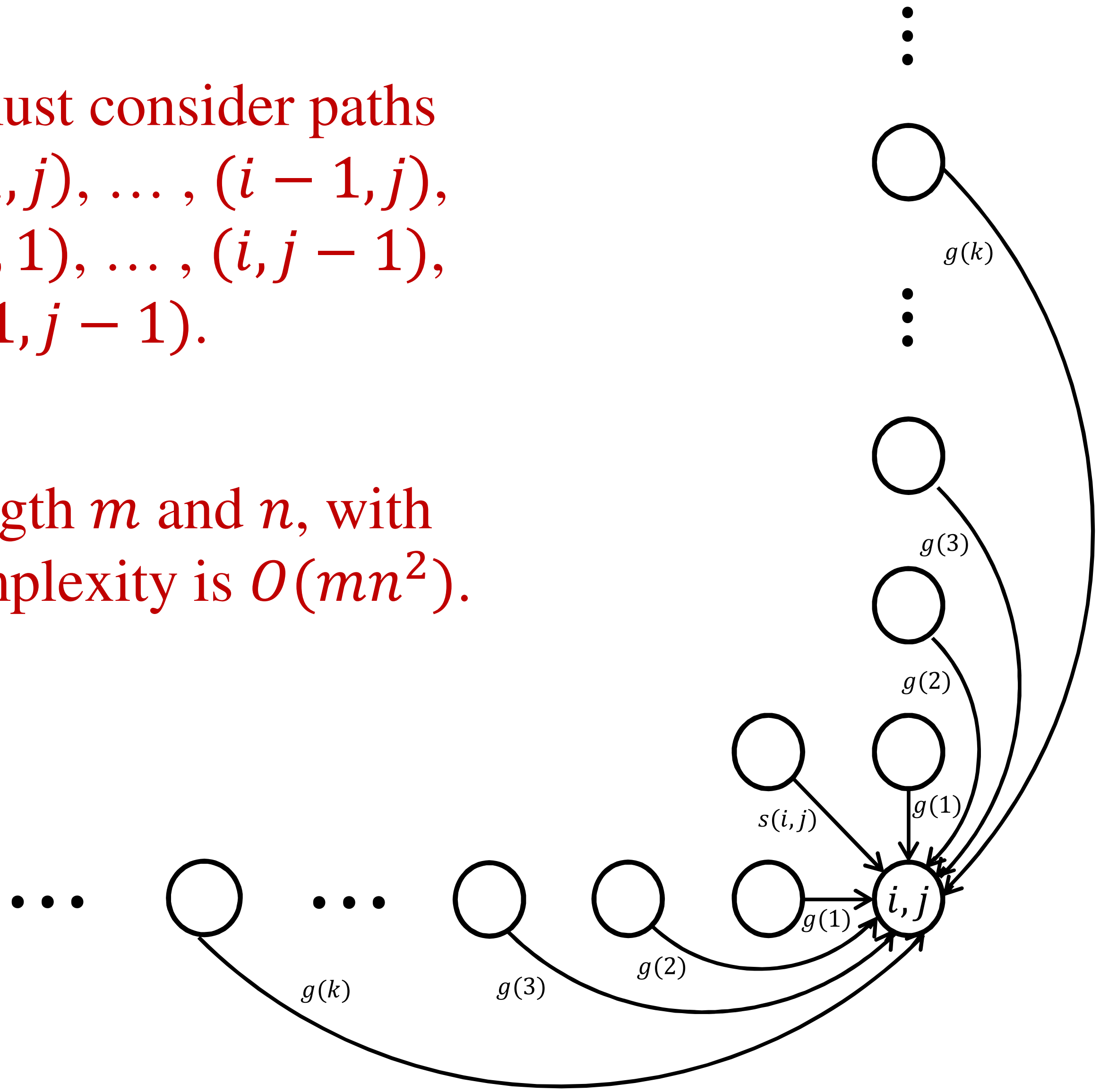
Scoring system change: The score for a gap of length k is $g(k)$.

Algorithmic change: The basic Needleman-Wunsch and Smith-Waterman algorithms need to be modified to deal with length-dependent gap scores.

Dynamic Programming with Length-Dependent Gap Scores

At node (i, j), one must consider paths from nodes $(0, j)$, $(1, j)$, … , $(i-1, j)$, from nodes $(i, 0)$, $(i, 1)$, … , $(i, j-1)$, and from node $(i-1, j-1)$.

For sequences of length m and n, with $m \leq n$, the time complexity is $O(mn^2)$.



Affine Gap Scores

Time complexity $O(mn^2)$ is very burdensome even for two sequences of relatively moderate length, and is completely impractical when many sequences must be compared, such as in a database search.

However, consider gap scores of the form $\boldsymbol{g(k) = -(a + bk)}$. When $a > 0$, these *affine* gap scores favor alignments with fewer total gaps, not just alignments with a smaller number of indels.

Fortunately, a relatively simple modification of the Needleman-Wunsch and Smith-Waterman algorithms permits one to use affine gap costs with only a constant factor more computation, so that the algorithms remain $O(mn)$.

Gotoh Algorithm for Affine Gap Scores

The key idea is to store at each node (i, j) the best score $H(i, j)$ for entering the node from the left, the best score $V(i, j)$ for entering the node from above, as well as the best score $SIM(i, j)$ for entering the node by any path. For the Needleman-Wunsch algorithm, with affine gap scores of the form $g(k) = -(a + bk)$, one then has these recursions:

$$\boldsymbol{H(i,j) = \max[\, H(i,j-1) - b \;,\; SIM(i,j-1) - a - b]}$$

$$\boldsymbol{V(i,j) = \max[\, V(i-1,j) - b \;,\; SIM(i-1,j) - a - b]}$$

$$\boldsymbol{SIM(i,j) = \max[\, H(i,j) \;,\; V(i,j) \,,\; SIM(i-1,j-1) + s\left(x_i, y_j\right)]}$$

Gotoh, O. (1982) “An improved algorithm for matching biological sequences.” *J. Mol. Biol.* **162**:705-708.

Note: The traceback procedure has a few subtleties, and Gotoh’s published procedure is incorrect. A correct procedure is described in: Altschul S.F. & Erickson, B.W. (1986) *Bull. Math. Biol.* **48**:603-616.

Affine Gap Score Nomenclature

Sometimes a set of affine gap costs are described as having an "opening cost" a and an "extension cost" b, or by the ordered pair (a, b).

Unfortunately, both descriptions can be ambiguous because they can mean either that a gap of length k has score $\boldsymbol{g(k) = -(a + bk)}$, or that it has score $\boldsymbol{g(k) = -[a + b(k - 1)]}$. In the former case, a gap of length 1 has score $-(a + b)$, and in the latter case it has score $-a$.

Be sure to read a paper carefully to understand which convention is being used, and be explicit when describing affine gap scores. We will use the convention that $g(k) = -(a + bk)$.

Concave Gap Scores

Gap scores that are not as simple as affine, but still not completely general, have been proposed. For example, logarithmic gap scores of the form

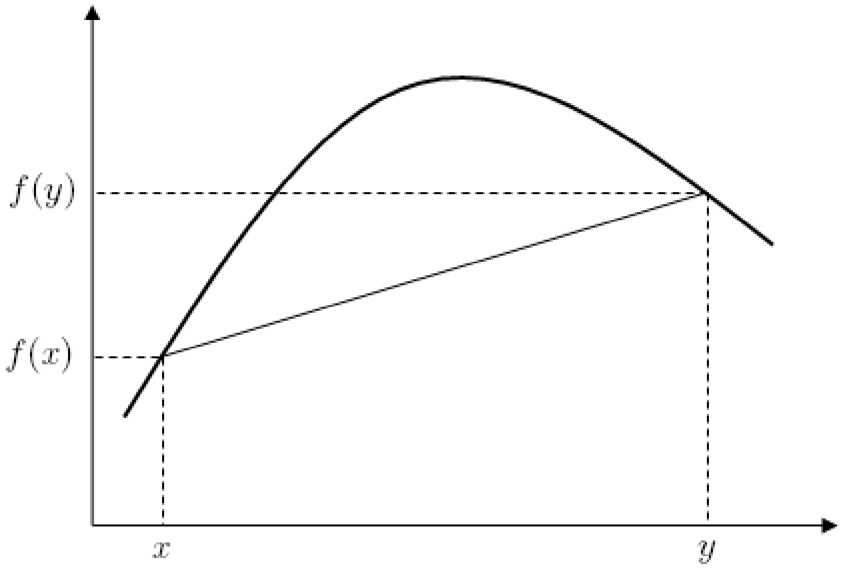
$$\boldsymbol{g(k) = -[a + b\log(k)]}$$

have sometimes been advocated. Gap scores of this form, and of many others, are called *concave,* because $|g|$ is a *concave function*.

A concave function $f(x)$ is one for which

$$\boldsymbol{f[tx + (1-t)y] \geq tf(x) + (1-t)f(y)}$$

for all x, y in f's domain, and all $t \in [0,1]$.



Algorithms with time complexity $O(mn)$ have been described for concave gap scores, although the algorithms themselves are somewhat complicated. In practice, almost all alignment programs in common use employ affine gap costs.

Miller, W. & Myers, E.W. (1988) “Sequence comparison with concave weighting functions.” *Bull. Math. Biol.* **50**:97-120.

Other Generalizations

Letter-dependent gap scores

Most of the algorithms we have considered allow the score for inserting or deleting a letter to depend on the letter, with no change in time complexity.

Generalized affine gap scores

Once a gap has been opened, one may allow diagonal "gap steps" through the path graph, as well as horizontal or vertical ones. Such diagonal steps may be interpreted as leaving letters inside the alignment unaligned. This can be useful for distantly related proteins, where certain regions may have diverged beyond reliable alignment. There is no increase in time complexity.

Altschul, S.F. (1998) "Generalized affine gap costs
for protein sequence alignment." *Proteins* **32**:88-96.

The Effect of Changing Gap Scores

Three optimal local protein alignments produced using BLOSUM-62 substitution scores in conjunction with various gap scores.

Gap scores

(0, 6)

```
 49 CERTLKYFLGIAGGKWVVSYFWVTQSIKERKMLNEHDFEVRGDVVNGRNHQGPKRARESQDRK-IFRGLEICCYG  122
    C RT KYFL +A G   VS+ WV  S    ++ N  ++ +   +  G + +  +R  + Q R+  F+ L++
865 C-RTRKYFLCLASGIPCVSHVWVHDSCHANQLQNYRNY-L---LPAGYSLE-EQRILDWQPRENPFQNLKVLLVS  933

123 PFTNMPTDQLEWM-VQLC-GASVVKE-LSS-FT--LGTGVHPIVVVQPDAWTEDNGFHAIGQMCEAPVVTREWVL  191
            +L W  + +  GA+ VK+  SS     +  GV  +VV  P      +      +  + PVV++EWV+
934 D-QQQNFLEL-WSEILMTGGAASVKQhHSSAHNKDIALGVFDVVVTDPSC-PA-SVLKC-AEALQLPVVSQEWVI 1003
```

(11, 1)

```
 51 RTLKYFLGIAGGKWVVSYFWVTQSIKERKMLNEHDFEVRGDVVNGRNHQGPKRARESQDRK-IFRGLEICCYG  122
    RT KYFL +A G   VS+ WV  S    ++ N  ++     ++        +R  + Q R+  F+ L++
866 RTRKYFLCLASGIPCVSHVWVHDSCHANQLQNYRNY-----LLPAGYSLEEQRILDWQPRENPFQNLKVLLVS  933

123 PFTNMPTDQLEWMVQLCGASVVKELSSFT----LGTGVHPIVVVQPDAWTEDNGFHAIGQMCEAPVVTREWVL  191
           +    ++   GA+ VK+  S      +  GV  +VV  P             +  + PVV++EWV+
934 DQQQNFLELWSEILMTGGAASVKQHHSSAHNKDIALGVFDVVVTDPSC---PASVLKCAEALQLPVVSQEWVI 1003
```

(12, 1, 0.3)

```
 51 RTLKYFLGIAGGKWVVSYFWVTQSIKERKMLNEHDFevrgdvvngrnhqgpKRARESQDRKi-FRGLEIccyg  122
    RT KYFL +A G   VS+ WV  S    ++ N  ++               +R  + Q R+  F+ L++
866 RTRKYFLCLASGIPCVSHVWVHDSCHANQLQNYRNYllpagyslee-----QRILDWQPREnpFQNLKVllvs  933

123 pftnmptdqlewmvqlcGASVVKELSSft----LGTGVHPIVVVQPDawtedngfhaiGQMCEAPVVTREWVL  191
                     GA+ VK+  S      ++ GV  +VV  P            ++  + PVV++EWV+
934 dqqqnflelwseilmtgGAASVKQHHSsahnkdIALGVFDVVVTDPScpasvlkc---AEALQLPVVSQEWVI 1003
```