An Improved Search Algorithm for Optimal Multiple-Sequence Alignment

Paper by: Stefan Schroedl
Presentation by: Bryan Franklin
Outline

- Multiple-Sequence Alignment (MSA)
- Graph Representation
- Computing Path Costs
- Heuristics
- Experimental Results
- Other Optimizations
Multiple-Sequence-Alignment

• Sequence:
  • DNA: String over alphabet \{A,C,G,T\}
  • Protein: String with \(|\Sigma|=20\) (one symbol for each amino acid)

• Alignment:
  • Insert gaps (\_) into sequences to line up matching characters.
Multiple-Sequence-Alignment

Sequences: ABCB, BCD, DB

Alignment:

A B C _ B
_ B C D _
_ _ _ D B
Multiple-Sequence-Alignment

- Indel: Insertion, Deletion, Point mutation (single symbol replacement)
- Find a minimum set of indels between two or more sequences.
- NP-Hard for an arbitrary number of sequences
Multiple-Sequence-Alignment

• Applications
  • Common ancestry between species
  • Locating useful portions of DNA
  • Predicting structure of folded proteins
Graph Representation
Computing $G(n)$

- Considerations
  - Biological Meaning
  - Cost of computation
- Sum-of-pairs
  - Substitution matrix $((|\Sigma|+1)^2)$
  - Sum alignment costs for all pairs
  - Costs can depend on neighbors
Computing $G(n)$

**Sequences:** ABCB, BCD, DB

**Alignment:**

- A B C _ B
- _ B C D _
- _ _ _ D B

**Cost:** $6+7+8+7+7 = 35$

**Substitution matrix:**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

---

A number of improvements can be integrated into the sum of pairs cost like associating weights with sequences and using different substitution matrices for sequences of varying evolutionary distance. A major issue in multiple sequence alignment algorithms is their ability to handle gaps. Gap penalties can be made dependent on the neighbor letters. Moreover, it has been found that assigning a fixed score for each indel sometimes does not produce the biologically most plausible alignment. Since the insertion of a sequence of $x$ letters is more likely than $x$ separate insertions of a single letter, gap cost functions have been introduced that depend on the length of a gap. A useful approximation are gap costs which distinguish between opening and extension of a gap and charge $a x b$ for a gap of length $x$ for appropriate $a$ and $b$. Another frequently used modification is to waive the penalties for gaps at the beginning or end of a sequence.

Technically, in order to deal with fine gap costs we can no longer identify nodes in the search graph with lattice vertices, since the cost associated with an edge depends on the preceding edge in the path. Therefore, it is more suitable to store lattice edges in the priority...
Computing $G(n)$
Computing $G(n)$

\[
\begin{array}{cccc}
A & B & C & D \\
A & & & \\
B & & & \\
A & & & \\
D & & & \\
\end{array}
\]

\[
g = 53
\]

\[
cost(A, \_ ) = 3
\]
\[
gap \text{ penalty} = 4
\]
\[
g = 60
\]

\[
cost(\_, C) = 3
\]
\[
gap \text{ penalty} = 4
\]
\[
g = 60
\]

\[
cost(A, \_) = 3
\]
\[
g = 60
\]

\[
cost(A, C) = 4
\]
\[
gap \text{ penalty} = 0
\]
\[
g = 57
\]
Heuristics

- Methods Examined
  - 2-fold ($h_{\text{pair}}$)
  - divide-conquer
  - 3-fold ($h_{3,\text{all}}$)
  - 4-fold ($h_{4,\text{all}}$)
Heuristic Comparison

An Improved Search Algorithm for Optimal Multiple-Sequence Alignment

Figure 17: Combined maximum size of Open and Closed, for different heuristics (elongation factors EF-TU and EF-α).

Figure 18: Maximum size of Open list, dependent on the final search threshold (BAliBASE).
### Resource Usage

#### Table 8: Long sequences, low similarity.

<table>
<thead>
<tr>
<th>S</th>
<th>δ</th>
<th>g*(t)</th>
<th>h(s)</th>
<th>#Exp</th>
<th>#Op</th>
<th>#Op+Cl</th>
<th>#Heu</th>
<th>Time</th>
<th>Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ajsA</td>
<td>4</td>
<td>38382</td>
<td>38173</td>
<td>318460012</td>
<td>1126697</td>
<td>2310632</td>
<td>27102589</td>
<td>9827.233</td>
<td>208951</td>
</tr>
<tr>
<td>1cpt</td>
<td>4</td>
<td>39745</td>
<td>39628</td>
<td>873548</td>
<td>5260</td>
<td>12954</td>
<td>10494564</td>
<td>223.926</td>
<td>32119</td>
</tr>
<tr>
<td>1lvl</td>
<td>4</td>
<td>43997</td>
<td>43775</td>
<td>537914936</td>
<td>1335670</td>
<td>2706940</td>
<td>37491416</td>
<td>16473.420</td>
<td>255123</td>
</tr>
<tr>
<td>1ped</td>
<td>3</td>
<td>15351</td>
<td>15207</td>
<td>2566052</td>
<td>7986</td>
<td>27718</td>
<td>0</td>
<td>20.035</td>
<td>4447</td>
</tr>
<tr>
<td>2myr</td>
<td>4</td>
<td>43414</td>
<td>43084</td>
<td>374001764</td>
<td>275256</td>
<td>544800</td>
<td>118747184</td>
<td>136874.980</td>
<td>927735</td>
</tr>
<tr>
<td>4enl</td>
<td>3</td>
<td>16146</td>
<td>16011</td>
<td>5169296</td>
<td>42620</td>
<td>72502</td>
<td>0</td>
<td>41.716</td>
<td>5589</td>
</tr>
</tbody>
</table>

#### Table 9: Long sequences, medium similarity.

<table>
<thead>
<tr>
<th>S</th>
<th>δ</th>
<th>g*(t)</th>
<th>h(s)</th>
<th>#Exp</th>
<th>#Op</th>
<th>#Op+Cl</th>
<th>#Heu</th>
<th>Time</th>
<th>Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ac5</td>
<td>4</td>
<td>37147</td>
<td>37020</td>
<td>169779871</td>
<td>732333</td>
<td>1513853</td>
<td>18464119</td>
<td>6815.760</td>
<td>124877</td>
</tr>
<tr>
<td>1adj</td>
<td>4</td>
<td>32815</td>
<td>32785</td>
<td>207072</td>
<td>3106</td>
<td>5145</td>
<td>96176</td>
<td>7.829</td>
<td>4595</td>
</tr>
<tr>
<td>1bgl</td>
<td>4</td>
<td>78366</td>
<td>78215</td>
<td>188429118</td>
<td>857008</td>
<td>1744149</td>
<td>101816849</td>
<td>8795.000</td>
<td>291618</td>
</tr>
<tr>
<td>1dlc</td>
<td>4</td>
<td>47430</td>
<td>47337</td>
<td>14993317</td>
<td>65288</td>
<td>126608</td>
<td>12801019</td>
<td>843.402</td>
<td>43158</td>
</tr>
<tr>
<td>1eft</td>
<td>4</td>
<td>31377</td>
<td>31301</td>
<td>9379999</td>
<td>42620</td>
<td>72502</td>
<td>1476154</td>
<td>334.475</td>
<td>13115</td>
</tr>
<tr>
<td>1fieA</td>
<td>4</td>
<td>53321</td>
<td>53241</td>
<td>6905957</td>
<td>46779</td>
<td>90937</td>
<td>6040375</td>
<td>348.134</td>
<td>26884</td>
</tr>
<tr>
<td>1gowA</td>
<td>4</td>
<td>38784</td>
<td>38632</td>
<td>45590739</td>
<td>275256</td>
<td>544800</td>
<td>31318364</td>
<td>2251.190</td>
<td>99537</td>
</tr>
<tr>
<td>1pkm</td>
<td>4</td>
<td>36356</td>
<td>36256</td>
<td>11197890</td>
<td>75144</td>
<td>140472</td>
<td>5962640</td>
<td>505.778</td>
<td>27244</td>
</tr>
<tr>
<td>1sesA</td>
<td>5</td>
<td>57670</td>
<td>57557</td>
<td>4755983</td>
<td>96014</td>
<td>136677</td>
<td>3585721</td>
<td>463.962</td>
<td>27452</td>
</tr>
<tr>
<td>2ack</td>
<td>5</td>
<td>76937</td>
<td>76466</td>
<td>994225856</td>
<td>8077412</td>
<td>12436928</td>
<td>75819994</td>
<td>32965.522</td>
<td>765715</td>
</tr>
<tr>
<td>2arp</td>
<td>5</td>
<td>54939</td>
<td>54696</td>
<td>182635167</td>
<td>1291185</td>
<td>2160263</td>
<td>38368530</td>
<td>15972.000</td>
<td>193364</td>
</tr>
<tr>
<td>2glg</td>
<td>5</td>
<td>74282</td>
<td>74059</td>
<td>9251905</td>
<td>87916</td>
<td>120180</td>
<td>22622910</td>
<td>733.202</td>
<td>72148</td>
</tr>
</tbody>
</table>

#### Table 10: Long sequences, high similarity.

<table>
<thead>
<tr>
<th>S</th>
<th>δ</th>
<th>g*(t)</th>
<th>h(s)</th>
<th>#Exp</th>
<th>#Op</th>
<th>#Op+Cl</th>
<th>#Heu</th>
<th>Time</th>
<th>Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ad3</td>
<td>4</td>
<td>33641</td>
<td>33604</td>
<td>104627</td>
<td>2218</td>
<td>3461</td>
<td>34539</td>
<td>4.196</td>
<td>3968</td>
</tr>
<tr>
<td>1gpb</td>
<td>5</td>
<td>101296</td>
<td>101231</td>
<td>1232707</td>
<td>62184</td>
<td>98476</td>
<td>2702949</td>
<td>178.610</td>
<td>25698</td>
</tr>
<tr>
<td>1gtr</td>
<td>5</td>
<td>55242</td>
<td>55133</td>
<td>2037633</td>
<td>54496</td>
<td>91656</td>
<td>1916127</td>
<td>226.791</td>
<td>18050</td>
</tr>
<tr>
<td>1lcf</td>
<td>6</td>
<td>149249</td>
<td>148854</td>
<td>181810148</td>
<td>3235312</td>
<td>3824010</td>
<td>28614215</td>
<td>15363.051</td>
<td>294688</td>
</tr>
<tr>
<td>1rthA</td>
<td>5</td>
<td>69296</td>
<td>69133</td>
<td>14891538</td>
<td>71081</td>
<td>105082</td>
<td>24587882</td>
<td>1721.070</td>
<td>70569</td>
</tr>
<tr>
<td>1taq</td>
<td>5</td>
<td>133723</td>
<td>133321</td>
<td>1693501628</td>
<td>9384718</td>
<td>17298456</td>
<td>145223167</td>
<td>5713.240</td>
<td>1170673</td>
</tr>
<tr>
<td>3pmg</td>
<td>4</td>
<td>42193</td>
<td>42133</td>
<td>1036943</td>
<td>8511</td>
<td>15540</td>
<td>777639</td>
<td>50.796</td>
<td>8133</td>
</tr>
<tr>
<td>actin</td>
<td>5</td>
<td>48924</td>
<td>48826</td>
<td>824295</td>
<td>35283</td>
<td>53009</td>
<td>777058</td>
<td>96.147</td>
<td>11198</td>
</tr>
</tbody>
</table>
Optimizations

- Sparse Path Representation
- Curve fitting for predicting threshold values
- Sub-optimal paths periodically deleted
The End

Any Questions?