Large multiple sequence alignments with a root-to-leaf regressive method
Content

1. Background
2. Introduction
3. Results
4. Conclusions
Content

1. Background
2. Introduction
3. Results
4. Conclusions
Background & Challenges

- Multiple sequence alignment (MSA) is an important step in many biological problems.
- Time consuming for large datasets
- Large volumes of sequences and the complexity of large datasets make it hard to construct accurate alignment
- Approximate solution
Content

1. Background
2. Introduction
3. Results
4. Conclusions
Main Idea

- Generally, methods that are executable on large datasets are progressive and many of them make use of the guide tree.

- Usually, progressive methods would start by aligning the most similar sequences and subsequently incorporate the remaining sequences, from leaf to root (Garriga et al. 2019).

- However, this paper presents a regressive method that works in the opposite way - it tries to align the most diverse sequences at first.
Pipeline

Sequences → Selection → Tree-shaped clusters → Tree-shaped sub-MSAs → Merge by transitivity → Final MSA
1. A guide tree is generated by any proper third-party software (e.g. Mafft) and each node of the guide tree is marked as the **longest** sequence below/at it.

2. Then the tree-shaped clusters would be constructed by walking through the tree by BFS, given a user-specified number $N$, which is the max number of children and the max size of each cluster.
Pipeline

Sequences → Tree-shaped clusters → Tree-shaped sub-MSAs → Final MSA

Selection

Merge by transitivity
Because there is a common sequence between each pair of parent and its child, all the indels can be projected so that the parent sub-MSA and the child sub-MSA can always be joint.

Garriga et al. 2019
Pipeline

Sequences → Selection → Tree-shaped clusters → Tree-shaped sub-MSAs → Merge by transitivity → Final MSA
Content

1. Background

2. Introduction

3. Results

4. Conclusions
Performance on 20 HomFam datasets

- Run time: Regressive < Nonregressive except (default+Sparsecore)
- TC score: Regressive > Nonregressive except (mBed+Fftns1)
- SoP: Regressive > Nonregressive except (mBed+Fftns1)

<table>
<thead>
<tr>
<th>Tree Method</th>
<th>MSA Algorithm</th>
<th>A</th>
<th>Over 10,000 sequences - 20 Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sums of Pairs (SoP)</td>
<td>Total Column Score (TC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nonregressive</td>
<td>Regressive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Score (%)</td>
<td>Score (%)</td>
</tr>
<tr>
<td>PartTree</td>
<td>Ffts1</td>
<td>56.34</td>
<td>59.74</td>
</tr>
<tr>
<td>mBed</td>
<td>Ffts1</td>
<td>68.71</td>
<td>64.00</td>
</tr>
<tr>
<td>PartTree</td>
<td>ClustalO</td>
<td>50.13</td>
<td>66.58</td>
</tr>
<tr>
<td>mBed</td>
<td>ClustalO</td>
<td>64.97</td>
<td>71.11</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>60.04</td>
<td>65.36</td>
</tr>
</tbody>
</table>

Garriga et al. 2019
Performance on 55 HomFam datasets

- Average run time: Regressive < Nonregressive
- Average TC score: Regressive > Nonregressive
- Average SoP: Regressive > Nonregressive
Performance on 94 HomFam datasets

- Average run time: Regressive < Nonregressive
- Average TC score: Regressive > Nonregressive
- Average SoP: Regressive > Nonregressive

<table>
<thead>
<tr>
<th>Tree Method</th>
<th>MSA Algorithm</th>
<th>Sums of Pairs (SoP)</th>
<th>Total Column Score (TC)</th>
<th>CPU (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Nonregressive</td>
<td>Regressive</td>
<td>Reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Score (%)</td>
<td>Score (%)</td>
<td>Score (%)</td>
</tr>
<tr>
<td>PartTree</td>
<td>Fhns1</td>
<td>70.95</td>
<td>71.71</td>
<td>83.64</td>
</tr>
<tr>
<td>mBed</td>
<td>Fhns1</td>
<td>78.46</td>
<td>75.99</td>
<td>84.94</td>
</tr>
<tr>
<td>PartTree</td>
<td>ClustalO</td>
<td>71.28</td>
<td>77.89</td>
<td>85.77</td>
</tr>
<tr>
<td>mBed</td>
<td>ClustalO</td>
<td>80.41</td>
<td>81.73</td>
<td>86.67</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td><strong>75.28</strong></td>
<td><strong>76.83</strong></td>
<td><strong>85.26</strong></td>
</tr>
</tbody>
</table>

Garriga et al. 2019
As the number of sequences increases, the average differential TC scores decreases (except an increase at the end).

Generally, TC scores decreases for all 3 conditions, where regressive methods decreases slower than nonregressive methods.

Garriga et al. 2019
Constrained Correspondence Analysis

- Above horizontal axis: Component constrained to be the TC score accuracy.
- Vertical axis: The best unconstrained component.

The projection onto the upper horizontal axis quantifies the contribution to variance of overall accuracy (Garriga et al. 2019).

- The major factor is MSA algorithm.
- Regressive methods have a definite positive effect on both axises.

Garriga et al. 2019
Except Ffns1-mBed and Sparsecore-mBed, regressive methods have less run time and better performance (almost same as the previous table).

Garriga et al. 2019
Run Time

A linear regression on the tuples of run time of Homfam datasets.

The slope of the curve is less than 1, which indicates regressive methods generally has less run time.

Garriga et al. 2019
Content

1. Background

2. Introduction

3. Results

4. Conclusions
Conclusions

- Generally, regressive methods have less run time and better performance than nonregressive* methods.

- Regressive operations have the main contributions to the improvement of the performance.

- However, the experimental results are not compared to other methods such as UPP, which weakens the conclusions’ strength.

- Further research is needed to detect the performance and the effect of parent identity.
Thanks