CLT optimization

Maxwell Spivakovsky

August 18, 2023

1 CLT

In [1] an algorithm for aligning two cell lineage trees is proposed. The algorithm is based on dynamic programming, choosing an alignment that optimizes the score between two trees. There are two external inputs to the algorithm: (1) the score matrix between the cell type of any terminal node of tree A and the cell type of any terminal node of tree B and (2) the pruning cost of leaving out a tree node of either tree A or tree B from the aligned tree. As mentioned in the [1] both of these are crucial to the results of alignment. Yet there is no theoretical model that selects these parameters; in the examples in the paper the scores used are either 2 or 0 and the pruning cost is chosen to be 1. There is a section that describes a greedy algorithm for choosing each score from a small number of available scores, but the algorithm relies on the assumption of the pruning cost that is fixed at 10, 20, 40, and 80 at each of the four rounds of the algorithm.

The main difficulty in relating the score matrix and the pruning cost to the output alignment score is that the dynamic programming algorithm is recursive by nature. To align two trees, each on the order of a thousand nodes requires the order of a million evaluations, where each evaluation may depend on the results of four other evaluations. This recursion makes the dependence on the score matrix and the pruning cost highly non-linear. But each evaluation on its own is a continuous function of the parameters, more specifically, the maximum of the results of several continuous evaluations, where the maximum corresponds to the dynamic programming nature of the solution. Deep learning packages like pyTorch are well suited for both recursive evaluations and for optimizing a piecewise linear objective function. The approach taken below treats the alignment score as a function of the score matrix and the tuning cost parameter and optimizes an objective function of these parameters. This not only optimizes the score matrix and the tuning cost separately but also accounts for how these parameters influence each other. The objective function that is optimized is not simply the alignment score between the two trees, the score matrix that produces a high alignment score could be aligning completely unrelated cells. Instead, the objective function maximizes the alignment score between two trees relative to the alignment scores of the random trees, constructed by randomly permuting the terminal nodes of each tree while keeping the rest of the structure unchanged. Let the alignment score between two trees, A and B be denoted as

$$A(a, b)$$

Let a_{σ^a} denote a tree that has the same structure as tree a but whose terminal nodes have been permuted according to the permutation σ^a. Suppose N such permutations are carried out and their mean and standard deviation are computed:

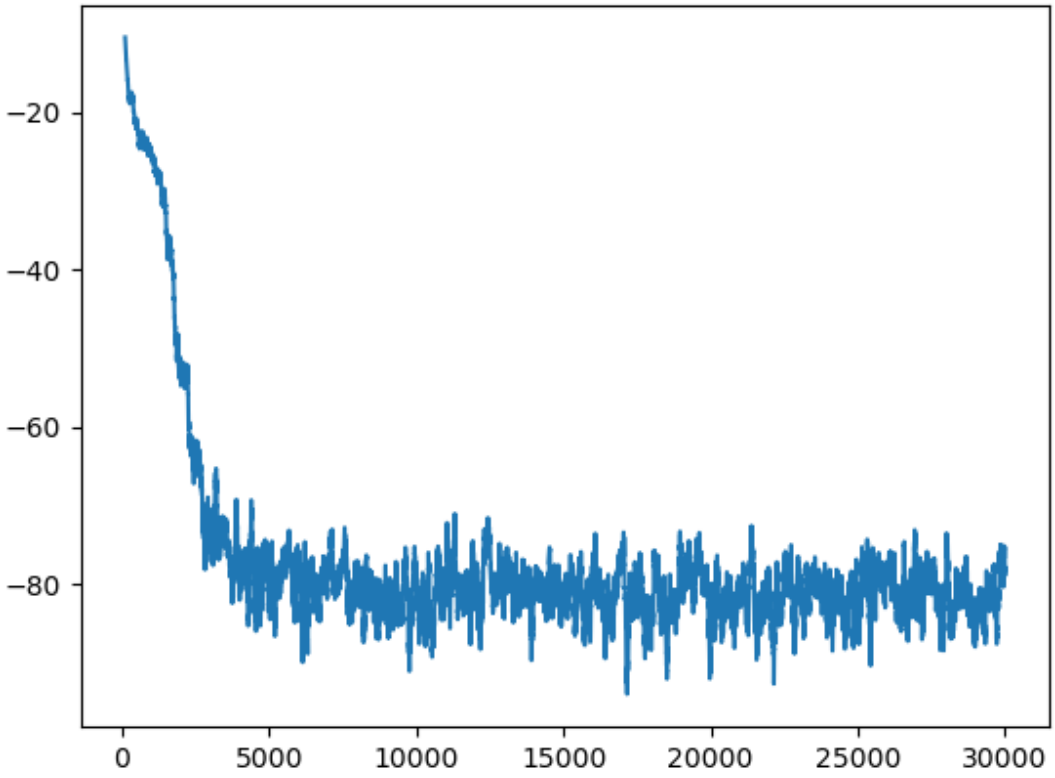
$$\mu = \frac{1}{N}\sum_{i=1}^{N} A(a_{\sigma_i^a}, b_{\sigma_i^b})$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\Big(A(a_{\sigma_i^a}, b_{\sigma_i^b}) - \mu\Big)^2$$

These quantities depend on the score matrix S and pruning cost p. The objective function is

$$F(S,p) = \frac{A(a,b) - \mu}{\sigma}$$

Maximizing $F(S,p)$ is choosing S, p that results in the highest alignment score of the two trees relative to the alignment scores of randomly permuted trees. The intention is that S, p are capturing the relationship between the two trees, both the geometry of each tree separately, as well as the geometry of their combination. Optimization is done with pyTorch in the typical deep learning fashion: a batch of random trees is created and $F(S,p)$ is evaluated with respect to that batch. pyTorch computes the gradient of $F(S,p)$ and modifies the values of S and p. Then another batch is created, different from the preceding one, and the process is repeated. It takes around 10,000 batches to converge to a solution. The matrix S is normalized to the unit norm at each step by applying $softmax$ to S. Below is a typical convergence profile of the model:



the x-axis is the number of batches. the y-axis shows by how many standard deviations the average batch's permuted trees alignment score is lower than the alignment score of the actual trees. This quantity is calculated on each batch before the score matrix and the pruning cost is changed after processing the batch so that the random trees in that batch have not been incorporated into optimization.

1.1 Optimal score matrix for a tree

When a tree is aligned with itself perfect alignment is expected. Intuitively, optimization should place high scores on matching a cell type from tree 1 to the same cell type of tree 2. There are 3 lineage trees in the GitHub that corresponds to the results in [1]. The names of the three trees are 'fun','pma', and 'hro'. 'fun' seems to correspond to *Caenorhabditis elegans*, 'pma' to *P. marina*. For each of the three trees, the optimizer selected the optimal score matrix when aligning the tree to itself. Since tree 1 and tree 2 correspond to the same organism the score matrix was constrained to be symmetric. The results are below.

	BLA	DEA	EPI	GER	GLA	INT	MUS	NEU	STR
BLA	**0.069614**	0.007390	0.000220	0.002148	0.002704	0.007579	0.000349	0.000673	0.000080
DEA	0.007390	**0.048439**	0.000558	0.022470	0.000884	0.012193	0.000153	0.001751	0.000141
EPI	0.000220	0.000558	**0.050339**	0.005938	0.001458	0.000376	0.000750	0.000016	0.003547
GER	0.002148	0.022470	0.005938	**0.157615**	0.000364	0.000605	0.016638	0.007050	0.015871
GLA	0.002704	0.000884	0.001458	0.000364	**0.089513**	0.004575	0.012991	0.008902	0.026936
INT	0.007579	0.012193	0.000376	0.000605	0.004575	**0.078844**	0.000169	0.000941	0.001295
MUS	0.000349	0.000153	0.000750	0.016638	0.012991	0.000169	**0.044478**	0.000404	0.006591
NEU	0.000673	0.001751	0.000016	0.007050	0.008902	0.000941	0.000404	**0.037532**	0.004220
STR	0.000080	0.000141	0.003547	0.015871	0.026936	0.001295	0.006591	0.004220	**0.065764**

Table 1: fun aligned with fun

	?	EPI	GER	INT	MUS	NER	PHA	X
?	**0.107569**	0.000359	0.030688	0.017442	0.000731	0.000040	0.002285	0.000146
EPI	0.000359	**0.063155**	0.000767	0.022299	0.000057	0.000278	0.000054	0.005833
GER	0.030688	0.000767	**0.216988**	0.001329	0.002331	0.001100	0.002389	0.001176
INT	0.017442	0.022299	0.001329	**0.125486**	0.001374	0.000019	0.003746	0.000202
MUS	0.000731	0.000057	0.002331	0.001374	**0.080111**	0.000002	0.000031	0.000006
NER	0.000040	0.000278	0.001100	0.000019	0.000002	**0.063512**	0.001003	0.000761
PHA	0.002285	0.000054	0.002389	0.003746	0.000031	0.001003	**0.068355**	0.000151
X	0.000146	0.005833	0.001176	0.000202	0.000006	0.000761	0.000151	**0.081624**

Table 2: pma aligned with pma

	END	EPI	MES	MUS	NER	NOT	UND
END	**0.114746**	0.023429	0.037622	0.000288	0.000104	0.034231	0.026823
EPI	0.023429	**0.061407**	0.007034	0.000108	0.012182	0.001714	0.002341
MES	0.037622	0.007034	**0.108711**	0.006061	0.019805	0.001054	0.000820
MUS	0.000288	0.000108	0.006061	**0.081251**	0.000019	0.000141	0.000361
NER	0.000104	0.012182	0.019805	0.000019	**0.080426**	0.002290	0.005659
NOT	0.034231	0.001714	0.001054	0.000141	0.002290	**0.095617**	0.000435
UND	0.026823	0.002341	0.000820	0.000361	0.005659	0.000435	**0.092802**

Table 3: hro aligned with hro

The resulting score matrix follows a pattern that makes sense: cells of the same cell type have high scores. These are the cells on the diagonals. What is interesting is the differences in the values of different types. They are inversely correlated to the relative frequency of that type in the tree. The table below shows this relationship

	score	freq
GER	0.216988	0.003135
INT	0.125486	0.031348
?	0.107569	0.047022
X	0.081624	0.105016
MUS	0.080111	0.126959
PHA	0.068355	0.175549
EPI	0.063155	0.205329
NER	0.063512	0.305643

Table 4: pma

	score	freq
GER	0.157615	0.002981
GLA	0.089513	0.019374
INT	0.078844	0.029806
BLA	0.069614	0.058122
STR	0.065764	0.068554
EPI	0.050339	0.138599
DEA	0.048439	0.168405
MUS	0.044478	0.183308
NEU	0.037532	0.330849

Table 5: fun

	score	freq
MES	0.108711	0.054545
UND	0.092802	0.054545
MUS	0.081251	0.090909
NOT	0.095617	0.090909
END	0.114746	0.109091
NER	0.080426	0.145455
EPI	0.061407	0.454545

Table 6: hro

Interestingly, not all 100% of weights go to the diagonal but only about $\frac{2}{3}$, except for *p.marina* where the number is 80%. The rest are distributed off-diagonal and show the

relevance of the structure of the tree.
The structure becomes more important when the score matrix is optimized between two trees of, potentially, different species that may have almost no common cell types. Optimized score matrices reveal a shared structure:

	BLA	DEA	EPI	GER	GLA	INT	MUS	NEU	STR
?	0.024275	0.006415	0.027885	0.011482	0.021846	0.006801	0.017833	0.000940	0.001354
EPI	0.026921	0.001013	0.024802	0.021039	0.011732	0.001623	0.001116	0.018367	0.019453
GER	0.003457	0.004396	0.004243	0.114557	0.025875	0.006862	0.023546	0.010042	0.029298
INT	0.008157	0.006220	0.002835	0.005442	0.016921	0.053291	0.004206	0.005393	0.005459
MUS	0.002488	0.001362	0.003693	0.010061	0.004952	0.005915	0.034438	0.002265	0.003737
NER	0.002552	0.013031	0.001001	0.011818	0.005530	0.003976	0.006895	0.023576	0.030240
PHA	0.005753	0.007883	0.025624	0.014449	0.031280	0.005207	0.024075	0.019766	0.019413
X	0.002249	0.031382	0.010304	0.024894	0.005800	0.004921	0.011795	0.007727	0.004855

Table 7: pma aligned with fun

	BLA	DEA	EPI	GER	GLA	INT	MUS	NEU	STR
END	0.044961	0.047796	0.004321	0.018675	0.006088	0.006202	0.003622	0.004312	0.005678
EPI	0.008040	0.007596	0.000922	0.019303	0.014485	0.007654	0.007545	0.109376	0.008933
MES	0.024814	0.002101	0.001278	0.012823	0.025432	0.005811	0.025439	0.001166	0.001437
MUS	0.003692	0.002611	0.122409	0.038071	0.014373	0.010864	0.023734	0.016389	0.009897
NER	0.000385	0.000383	0.025789	0.040149	0.001510	0.000447	0.000374	0.000374	0.024223
NOT	0.011779	0.000938	0.011064	0.005534	0.002853	0.001401	0.000944	0.009909	0.001262
UND	0.014220	0.102719	0.005293	0.005924	0.026061	0.002589	0.011367	0.013813	0.010840

Table 8: hro aligned with fun

	?	EPI	GER	INT	MUS	NER	PHA	X
END	0.010317	0.082711	0.052107	0.027220	0.006975	0.026307	0.013094	0.036335
EPI	0.004391	0.004606	0.009331	0.004602	0.000648	0.033138	0.032584	0.004593
MES	0.004132	0.002755	0.009353	0.005197	0.004156	0.029742	0.004702	0.029706
MUS	0.030624	0.008573	0.034671	0.002886	0.011293	0.069257	0.011446	0.013647
NER	0.003268	0.002049	0.004210	0.002221	0.002047	0.032818	0.002147	0.026344
NOT	0.007108	0.011502	0.008691	0.010627	0.006071	0.057660	0.016311	0.010566
UND	0.020827	0.003862	0.015333	0.019328	0.020456	0.002516	0.020043	0.072894

Table 9: hro aligned with pma

pma and fun share many of the same cell types and the mutual alignment of three of them, GER, INT, and MUS, correspond to the three highest scores in the matrix but they still add up to under 20% of the total score. The other two cases have even less correspondence between cell types, with cell types labeled as the same type sometimes getting 0 weight

	fun	hro	pma
fun	0.000000	0.155655	0.009906
hro	0.155655	0.000000	0.098156
pma	0.009906	0.098156	0.000000

Table 10: Optimal pruning costs

(EPI in hro aligned with fun). The conditional distributions of a sibling conditional on the sibling, or of a child conditional on parents or grandparents become the primary drivers of alignment. These conditional distributions are complicated objects because they depend on many recursive layers. Score matrices reflect these conditional distributions through pairwise scores.

Optimal pruning costs vary quite a lot from scenario to scenario:

Since it is easy to get full alignment when the two trees represent the same organism, pruning costs are zero in that case. Pruning costs when aligning fun and hro or pma and hro are of the same order of magnitude as the hand-picked pruning costs in [1]. For aligning fun and pma the optimal pruning cost is an order of magnitude smaller.

References

[1] Yuan M, Yang X, Lin J, Cao X, Chen F, Zhang X, Li Z, Zheng G, Wang X, Chen X, Yang JR. Alignment of Cell Lineage Trees Elucidates Genetic Programs for the Development and Evolution of Cell Types. iScience. 2020 Jul 24;23(7):101273. doi: 10.1016/j.isci.2020.101273. Epub 2020 Jun 16. PMID: 32599560; PMCID: PMC7327887.