PALM: Probabilistic Area Loss Minimization for Protein Sequence Alignment

Fan Ding[1,a], Nan Jiang[1,a], Jianzhu Ma[2], Jian Peng[3], Jinbo Xu[4], Yexiang Xue[1]

[1]Purdue University [2]Peking University [3]University of Illinois at Urbana-Champaign [4]Toyota Technological Institute at Chicago

[a]These authors contribute equally.

Pairwise Protein Sequence Alignment

- **Learning.** Given a training set of $\{(S^{(k)}, T^{(k)}, a^{*(k)})\}_{k=1}^{N}$, where $S^{(k)}, T^{(k)}$ is a pair of sequences, and $a^{*(k)}$ is the ground-truth alignment between the two sequences. We want to learn:

$$\max_{\theta} \prod_{k=1}^{N} Pr_{\theta}\left(a^{*(k)} | S^{(k)}, T^{(k)}\right). \tag{1}$$

  - θ is the model's parameter;
  - $Pr_{\theta}(a|S,T)$ is the probability of alignment a.
- **Inference.** After training, we use the model $Pr_{\theta}(a|S,T)$ to find the best alignment between two new sequences by:

$$\hat{a} = \arg\max_{a \in \mathcal{A}} Pr(a|S,T) \tag{2}$$

$\hat{a}$ is the predicted alignment.

Our Motivation

- Biology datasets contain **notable errors** from the real experiments.
- Existing approaches are **not robust to the noises** in the dataset. Because they lead to the minimization of the pointwise differences of the two alignments.
- We consider the area of two alignments, which is **robust to errors and offsets of alignments**.

Example

- The missed predictions of the green alignment is result of biological measurement noises, while the orange alignment completely misses the ground-truth.
- The area distance between the ground-truth (blue line) and the first predicted alignment (green line) is 1.5, and is 4.5 between the ground-truth and the second one (orange line).

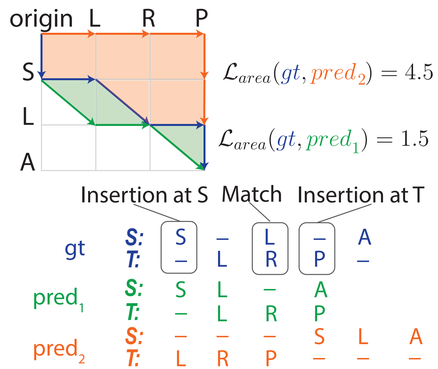

Figure 1. Given a pair of sequence (S,T), we can formulate an alignment matrix of shape $(|S|,|T|)$. Symbols M, I_S and I_T: represent a match, an insertion in S, and an insertion in T, respectively. Alignment a: a sequential symbols M, I_S and I_T.

Robust Learning via Area Loss Minimization

We define the area distance as:

$$Pr_{area}(a^*|a,S,T) = \frac{e^{-\lambda \mathcal{L}_{area}(a^*,a)}}{Z_{area}} \tag{3}$$

where the predicted alignment a that is similar to the observed one a^* has higher probability. It penalizes those predicted alignment that is far away from the observed alignment.

We maximize the likelihood of the observed alignment:

$$Pr(a^*|S,T) = \sum_{a} Pr_{area}(a^*|a,S,T) Pr_{\theta}(a|S,T). \tag{4}$$

which sums over the latent variable a.

Lower bound on $Pr_{\theta}(a^*|S,T)$

The objective needs to sum over all possible alignments, which is computationally intractable.

$$\log Pr(a^*|S,T) = \mathcal{L} = \log \sum_{a} \frac{e^{-\lambda \mathcal{L}_{area}(a^*,a)}}{Z_{area}} \frac{e^{\sum_{k=1}^{|a|} \phi_{\theta}(\pi_S(a,k), \pi_T(a,k), a_k)}}{Z_{\phi}}$$

So we use the lower bound $\mathcal{L}_{LB}$ as **learning objective**, based on the the principle of log-sum-exp function:

$$\mathcal{L}_{LB} = \max_{a} \{\sum_{k=1}^{|a|} \phi_{\theta}(\pi_S(a,k), \pi_T(a,k), a_k) - \lambda \mathcal{L}_{area}(a^*,a)\} - \log Z_{area} - \log Z_{\phi} \tag{5}$$

Gradient Estimation for $\mathcal{L}_{LB}$

The gradient is computed as:

$$\nabla \mathcal{L}_{LB} = \sum_{k=1}^{|\hat{a}|} \nabla \phi_{\theta}(\pi_S(\hat{a},k), \pi_T(\hat{a},k), \hat{a}_k) - \nabla \log Z_{\phi} \tag{6}$$

- $\nabla \phi_{\theta}(\pi_S(\hat{a},k), \pi_T(\hat{a},k), \hat{a}_k)$ is the gradient of function ϕ_{θ}.
- $\nabla \log Z_{\phi}$ is formulated as an expectation over $P(a|S,T)$ by contrastive divergence [1]:

$$\nabla \log Z_{\phi} \approx \frac{1}{M} \sum_{a^m \sim Pr_{\theta}(a|S,T)} \left[\sum_{k'=1}^{|a^m|} \nabla \phi_{\theta}(\pi_S(a^m,k'), \pi_T(a^m,k'), a^m_{k'}) \right] \tag{7}$$

We can approximate $\nabla \log Z_{\phi}$ by:

1. sampling M paths from $Pr_{\theta}(a|S,T)$
2. sum the gradients $\nabla \phi_{\theta}$ of all sampled path $\{a^m\}_{m=1}^{M}$.

Remark

The whole optimization process of each iteration is $\mathcal{O}(|S||T| + (|S|+|T|)M)$.

- computing gradient via dynamic sampling is $\mathcal{O}(|S||T| + (|S|+|T|)M)$.
- computing Z is $\mathcal{O}(|S||T|)$
- computing the probability distribution for sampling is $\mathcal{O}((|S|+|T|)M)$

Learning Effectiveness for PALM

	$\|S\| \in [1,100], \|T\| \in [100,200]$		$\|S\| \in [100,200], \|T\| \in [1,100]$	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
	exact/ 4-offset/10-offset	exact/ 4-offset /10-offset	exact/4-offset/10-offset	exact/4-offset/10-offset
DP	7.8/**31.3**/**51.2**	20.4/39.0/56.3	20.2/40.4/59.4	6.1/26.3/**45.1**
PALM	**9.9**/29.8/48.7	**23.5**/**43.1**/**62.3**	**26.8**/**44.6**/**63.2**	**6.4**/**26.6**/43.1
	$\|S\| \in [1,100], \|T\| \in [200,400]$		$\|S\| \in [200,400], \|T\| \in [1,100]$	
	exact/ 4-offset/10-offset	exact/ 4-offset/10-offset	exact/4-offset/10-offset	exact/4-offset/10-offset
DP	5.2/**27.6**/**46.1**	32.0/39.8/46.7	30.0/37.5/44.7	**3.8**/**19.8**/**34.4**
PALM	**6.5**/26.9/43.3	**51.4**/**62.5**/**73.3**	**52.7**/**63.5**/**73.8**	3.3/18.7/31.0

Table 1. Comparison of precision and recall between our method and dynamic programming (DP) over different lengths of protein sequences on PDB [2] dataset. 4-offset/10-offset are the relaxed measures. PALM gets better results especially on longer sequences and remote homologies than the competing approach.

Ablation study on hyper-parameter λ

	$\|S\| \in [1,100], \|T\| \in [400,+\infty)$		$\|S\| \in [400,+\infty), \|T\| \in [1,100]$	
	Precision	Recall	Precision	Recall
	exact/4-offset/10-offset	exact/4-offset/10-offset	exact/4-offset/10-offset	exact/4-offset/10-offset
$\lambda = 50$	**5.1%**/**22.6%**/**36.4%**	75.3%/81.1%/86.3%	75.9%/**81.1%**/86.0%	2.6%/17.0%/27.2%
$\lambda = 100$	4.6%/21.3%/35.2%	75.3%/81.1%/86.3%	**76.0%**/**81.1%**/**86.1%**	3.1%/**18.1%**/**29.1%**
$\lambda = 500$	4.5%/20.9%/34.0%	**75.4%**/**81.2%**/**86.4%**	75.9%/81.0%/85.9%	3.1%/17.4%/28.3%
$\lambda \to +\infty$	4.2%/20.8%/35.7%	75.1%/80.9%/85.0 %	75.0%/80.7%/85.0%	**3.5%**/16.8%/27.8%

Table 2. When λ approaches infinity, area distance becomes more important in the inference of $\hat{a}$ during training, which leads to $\hat{a}$ more similar to the ground-truth alignment a^*. It can be seen that when we select a suitable λ that strikes a balance between the area distance and the score function, we can learn a better model than pure maximum likelihood learning (when $\lambda \to +\infty$).

Time efficiency of computing the gradient

$\|S\|$	$\|T\|$	PALM (Ours)	Autograd
$[1,100]$	$[1,100]$	$\mathbf{0.7 \pm 0.2s}$	$2.5 \pm 0.8s$
$[100,200]$	$[100,200]$	$\mathbf{2.7 \pm 0.9s}$	$9.4 \pm 3.3s$
$[100,200]$	$[200,400]$	$\mathbf{6.6 \pm 2.3s}$	$25.4 \pm 9.4s$
$[200,400]$	$[100,200]$	$\mathbf{6.2 \pm 2.0s}$	$23.2 \pm 8.1s$
$[200,400]$	$[200,400]$	$\mathbf{12.5 \pm 2.3s}$	$51.7 \pm 11.2s$
$[400,+\infty)$	$[400,+\infty)$	$\mathbf{63.4 \pm 32.0s}$	$297.6 \pm 282.2s$

Table 3. PALM is much time efficient than the competing method Autograd, which computes the exact gradient by automatically back-propagation, among all length intervals of two protein sequences.

Reference

[1] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[2] Fandi Wu and Jinbo Xu. Deep template-based protein structure prediction. *bioRxiv*, 2020.

Acknowledgements

This research was supported by NSF grants IIS-1850243, CCF-1918327.

code: https://github.com/jiangnanhugo/PALM

{fan274, jiang631, yexiang}@purdue.edu, majianzhu@pku.edu.cn, jianpeng@illinois.edu, jinboxu@gmail.com