Structure- and Function-Aware Substitution Matrices via Learnable Graph Matching

Paolo Pellizzoni Carlos Oliver Karsten Borgwardt

Max Planck Institute of Biochemistry

Recomb 2024

What is a Substitution Matrix?

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

[a] Source: Wikipedia

- Given an alphabet $\mathcal{A} = \{a_1, ..., a_k\}$ of k elements.
- A substitution matrix is a function $c(a_i, a_j)$ which estimates the cost of replacing a_i for a_j on pairs of sequences $s \in \mathcal{A}^+$.
- Usually used to generate good alignments on sequences that have no ground-truth alignments.

Limitations

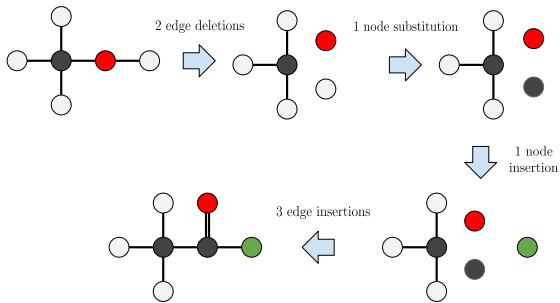
Typically the cost matrix is estimated as a substitution frequency in a set of ground-truth pre-aligned sequences. Recent work *[Llinares-López et al., Nature Methods 2023]* proposed a method to learn the cost matrix from data.

- Substitution matrices are designed to work mostly on **sequences**. Not applicable to non-sequence data and does not exploit **structural information** (higher-order signals).

- They often require ground-truth **alignments**, which might be hard to get.

Generalized Structure Alignment: Graph Edit Distance

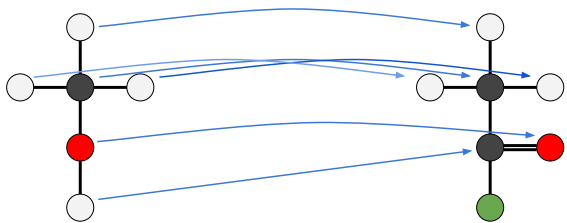
- Transform one graph into another by node and edge insertions, removal and substitutions.
- Given edit costs: find the minimum-cost edit path.



Generalized Structure Alignment: Graph Edit Distance

- Alternative formulation: find node matching $\pi : V_1 \to V_2$ that minimizes sum of node and edge edit costs.

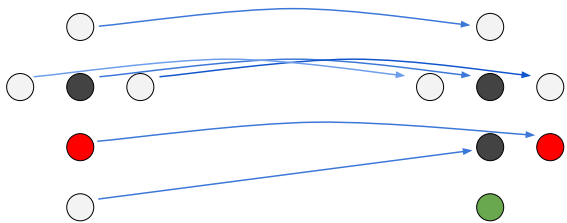
$$GED(G_1, G_2) = \min_{\pi \in \Pi} \sum_{v_i \in V_1^+} c_v(v_i, \pi(v_i)) + \sum_{v_i, v_j \in V_1} c_e(v_i, v_j, \pi(v_i), \pi(v_j)),$$



Approximations to GED

- **Naive strategy**: drop the edge cost and match just nodes. Polynomial-time solvable.

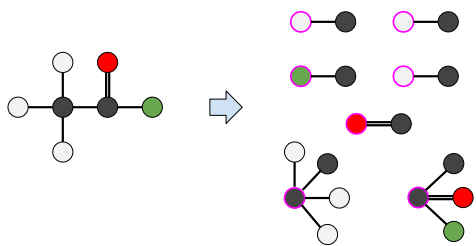
$$d(G_1, G_2) = \min_{\pi \in \Pi} \sum_{v_i \in V_1^+} c_v(v_i, \pi(v_i)),$$



Approximations to GED

- **Matching heuristic**: match substructures around nodes (e.g. Weisfeiler-Lehman unfolding trees). One needs to define an edit cost $c_{v,w}$ between such structures.

$$d_C(G_1, G_2) = \min_{\pi \in \Pi} \sum_{v_i \in V_1^+} c_{v_i, \pi(v_i)},$$



Learnable encodings of local substructures

- Graph Neural Networks (GNNs) are functions ψ_θ which embed node-centered substructures into $\mathbb{R}^d$.

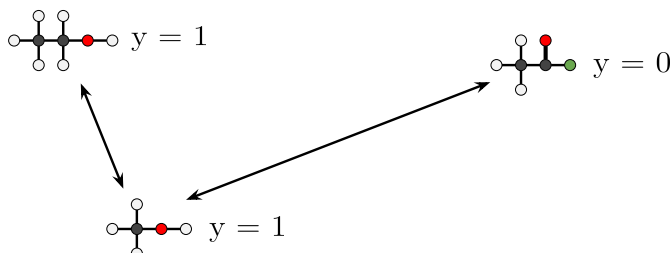
- We can obtain a **learnable** edit cost between substructures centered in nodes u, v, e.g. $c_{v,w} = \|\psi_\theta(v) - \psi_\theta(w)\|$.

- In turn, we can obtain a **learnable** distance $d_{\psi_\theta}(G_1, G_2)$ between graphs G_1, G_2 using the matching heuristic.

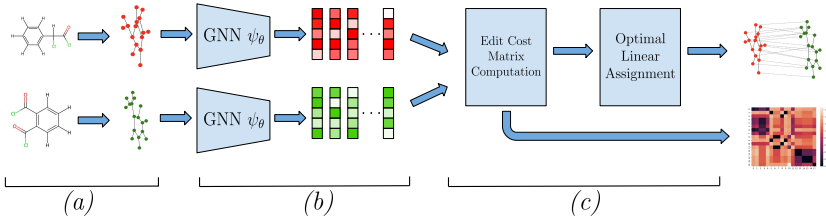
Metric Learning

- **Metric learning** setting: graphs that have the same labels should have low distance and graphs with different labels should have high distance.

- This allows to train a model without ground truth alignments: we just need class labels.



GMSM: Graph Matching Substitution Matrices



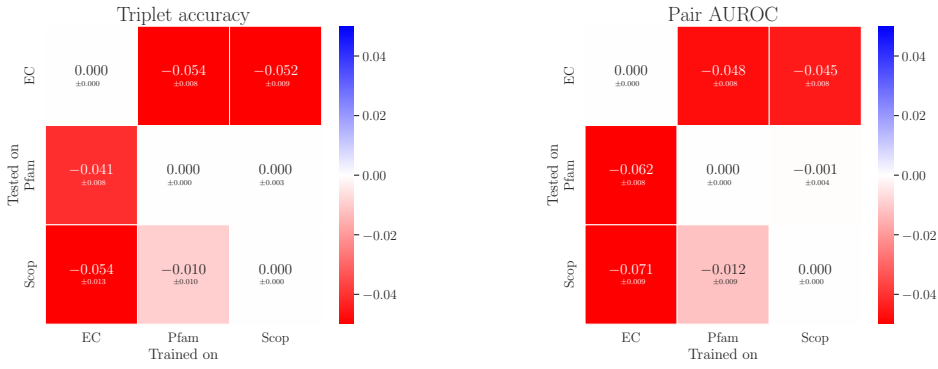
- Distance between graphs is a function of a shared and learnable edit cost function between substructures: strong **inductive bias**.
- Formally proven to yield a valid **metric space** (applicable to clustering and search).
- Trained with metric learning objective backpropagating through the edit costs.

GMSM outperforms structure-only and sequence-only methods

Similarity-based classification on macromolecules

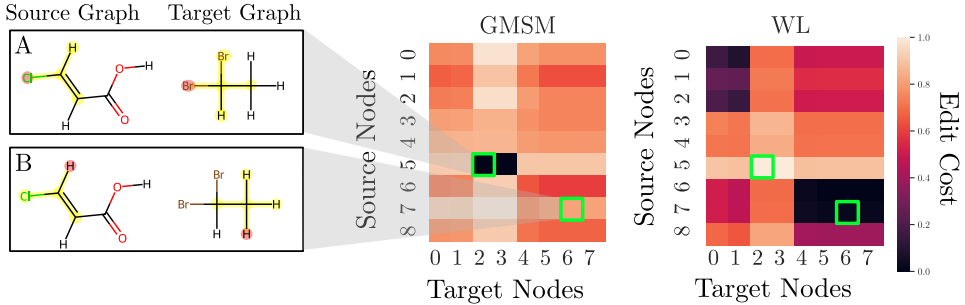
Method	EC		Pfam		SCOP		RNA	
	Trip. acc.	AUROC	Trip. acc.	AUROC	Trip. acc.	AUROC	Trip. acc.	AUROC
WL kernel	0.503	0.505	0.771	0.708	0.652	0.604	0.579	0.575
Seq. alignment (BLOSUM64)	0.470	0.478	0.587	0.541	0.470	0.478	-	-
Siamese-GNN	**0.643**	**0.622**	0.869	0.855	0.800	0.794	0.672	**0.663**
GMSM (node level)	0.520	0.518	0.864	0.855	0.799	0.789	0.606	0.603
GMSM (GNN)	0.573 ± 0.01	0.562 ± 0.01	**0.913** ± 0.01	**0.906** ± 0.00	**0.861** ± 0.01	**0.862** ± 0.01	**0.683** ± 0.02	0.659 ± 0.02

Task-Specificity of Learned Substitution Matrices



Transfer learning classification performance on proteins.

Learned substitution costs reflect functional substructures



Learned substitution matrices from GMSM vs structure-only WL kernel.

Conclusions

- We developed a new machine learning framework to automatically learn substitution costs over **structural alphabets** based solely on class labels (e.g. functional knowledge).

- **Future work:**
  - Incorporate domain-specific knowledge in feature extractors.
  - Develop theoretically sound framework to interpret the obtained substitution matrices.

Thank you!