Learning protein affinity codes with dynamic residues selection and multidimensional feature interactions using graph transformer

---

Algorithm 1 Dynamic residues selection

Input:

   data: Shape as $(L,)$. Each element of L contains all the information characterizing each residue in that protein complex, e.g., atomic coordinates, residue name, corresponding chain, index on the residue sequence.

   pos: Shape as $(L, 14, 3)$. Each element of L records information on the 3D coordinates of all heavy atoms in the residue.

   core_index: Shape as $(N_{core},)$. An index of all the residues located in the core region of this protein complex was recorded.

   mut_index: Shape as $(N_{mut},)$. An index of all mutated residues in this protein complex was recorded.

   G: Expected CORE regional share.

   lr_d: Optimising the move step of the virtual mutation point at each iteration.

   $N_{Maxstep}$: Maximum number of iterations.

   N_{res}: Number of residues selected.

Output:

   select_data: Shape as $(N_{res},)$. Each element of N_{res} records all the characteristic information of the corresponding residue.

 1:   **FUNCTION** DynamicResiduesSelection(data, pos, core_index, mut_index, G , lr_d, $N_{Maxstep}$, N_{res}):

 2:       # Extract CA coordinates from pos

 3:       SET pos_CA TO pos[:, 1, :]

 4:       # Extract core and mutated CA coordinates

 5:       SET pos_core_CA TO pos_CA[core_index]

 6:       SET pos_mut_CA TO pos_CA[mut_index]

 7:       # Initialize selection

 8:       # Find the indices of the N_{res} residues closest to pos_mut_CA

 9:       SET select_index TO indices of the N_{res} closest residues to pos_mut_CA based on pos_CA

10:       # Calculate the geometric center of all points in pos_core_CA

11:       SET pos_core_CA_center TO the geometric center of all points in pos_core_CA

12:       SET rnum TO 0

13:       # Iteratively refine selection

14:       **WHILE** (LENGTH(SET(core_index) - SET(select_index)) / LENGTH(core_index)) > (1 - G) **DO**

15:           # Move the virtual mutation point towards the core center

16:           SET pos_mut_CA TO pos_mut_CA moved towards pos_core_CA_center by lr_d

17:           # Update select_index with the N_{res} closest residues to pos_mut_CA

18:           SET select_index TO indices of the N_{res} closest residues to pos_mut_CA based on pos_CA

19:           INCREMENT rnum BY 1

20:           **IF** rnum >= $N_{Maxstep}$ **THEN**

21:	**BREAK**
22:	**END IF**
23:	**END WHILE**
24:	SET select_data TO data[select_index]
25:	**RETURN** select_data
26:	**END FUNCTION**

Algorithm 2 Gating strongly affect coding

Input:

$\{f_i^{pos\_CB}\}$: Shape as $(N_{res}, 3)$. The coordinates of the CB atom corresponding to the residue are recorded for each element of N_{res}.

Output:

$\{d_{ij}^h\}$: Shape as (N_{res}, N_{res}, H).

1: **FUNCTION** GatingStronglyAffectCoding($\{f_i^{pos\_CB}\}$):

2:      $gamma = softplus(W)$ $gamma, W \in \mathbb{R}^H$, W is a learnable parameter with an initial value of $log(e - 1)$ for each element

3:      $d_{ij} = norm(f_i^{pos_{CB}} - f_j^{pos\_CB})$ $d_{ij} \in \mathbb{R}$, units: Å, $i, j \in \{1, ..., N_{res}\}$

4:      $g_{ij} = sigmoid(LinearNoBias(d_{ij}))$ $h \in \{1, ..., H\}$, The initial value of the learnable parameter in LinearNoBias is 10

5:      $d_{ij} = \frac{\sqrt{2}}{6} \cdot gamma \cdot d_{ij} \cdot g_{ij}$ $d_{ij} \in \mathbb{R}^H$

6:      **RETURN** $\{d_{ij}^h\}$

7: **END FUNCTION**

Algorithm 3 Structural fusion coding

Input:

$\{f_i^{chain}\}$: Shape as $(N_{res},)$. Each element of N_{res} corresponds to the chain to which the residue belongs.

$\{f_i^{seq}\}$: Shape as $(N_{res},)$. Each element of N_{res} corresponds to the ordinal number of the residue on the primary sequence.

$\{f_i^{res\_type}\}$: Shape as $(N_{res},)$. Each element of N_{res} corresponds to the name of the residue.

$\{f_i^{pos\_CB}\}$: Shape as $(N_{res}, 3)$. The coordinates of the CB atom corresponding to the residue are recorded for each element of N_{res}.

Output:

$\{p_{ij}^h\}$: Shape as (N_{res}, N_{res}, H).

1: **FUNCTION** FusionCoding($\{f_i^{chain}\}, \{f_i^{seq}\}, \{f_i^{res\_type}\}, \{f_i^{pos\_CB}\}$):

2:      # Encoding sequence, structure and type information

3:      $d_{ij}^{seq} = f_i^{seq} - f_j^{seq}$ $i, j \in \{1, ..., N_{res}\}$

4:      **IF** $f_i^{chain} \neq f_j^{chain}$ **THEN**

5:          $d_{ij}^{seq} = 33$

6:      **ELIF** $d_{ij}^{seq} < -32$ **THEN**

7:          $d_{ij}^{seq} = -32$

8:      **ELIF** $d_{ij}^{seq} > 32$ **THEN**

9:        $d_{ij}^{seq} = 32$

10:     **END IF**

11:     $d_{ij}^{seq} = Embedding(\ d_{ij}^{seq} + 32)$

12:     $f_i^{res\_type\_OneHot} = one\_hot(f_i^{res\_type})$ $f_i^{res\_type\_OneHot} \in \mathbb{R}^{N_{res} \times 21}$

13:     $d_{ij}^{res\_type} = Linear\left(f_i^{res\_type\_OneHot}\right) + Linear\left(f_j^{res\_type\_OneHot}\right)$

14:     $d_{ij}^{distance} = Linear(one\_hot(norm(f_i^{pos\_CB} - f_j^{pos\_CB})))$ $d_{ij}^{seq}, d_{ij}^{res\_type}, d_{ij}^{distance} \in \mathbb{R}^{64}$

15:     # Output projection

16:     $d_{ij}^{seq} \leftarrow LayerNorm(d_{ij}^{seq})$

17:     $d_{ij}^{res\_type} \leftarrow LayerNorm(d_{ij}^{res\_type})$

18:     $d_{ij}^{distance} \leftarrow LayerNorm(d_{ij}^{distance})$

19:     $p_{ij} = Linear(d_{ij}^{seq} + d_{ij}^{res\_type} + d_{ij}^{distance})$ $p_{ij} \in \mathbb{R}^{H}$

20:     **RETURN** $\{p_{ij}^h\}$

21: **END FUNCTION**

Algorithm 3 Global node aggregation excitation attention

Input:

    $\{v_{ij}^h\}$: Shape as (N_{res}, N_{res}, H). Correlation matrix between residue pairs obtained using node feature encoding.

    $\{d_{ij}^h\}$: Shape as (N_{res}, N_{res}, H). Gating strongly affects coding matrix.

    $\{p_{ij}^h\}$: Shape as (N_{res}, N_{res}, H). Structural fusion coding matrix.

Output:

    $\{A_{ij}^h\}$: Shape as (N_{res}, N_{res}, H). Correlation matrix after modelling dependencies between multi-scale features.

1:    **FUNCTION** GlobalNodeAggregationExcitationAttention($\{v_{ij}^h\}, \{d_{ij}^h\}, \{p_{ij}^h\}$):

2:     # Global node aggregation

3:     $v_j = \sum_i \sum_h v_{ij}^h$

4:     $d_j = \sum_i \sum_h d_{ij}^h$

5:     $p_j = \sum_i \sum_h p_{ij}^h$ $j \in \{1, \dots, N_{res}\}, h \in \{1, \dots, H\}$

6:     # Adaptive recalibration

7:     $x_m \leftarrow \{v_j\}, \{d_j\}, \{p_j\}$ $x_m \in \mathbb{R}^{N_{res}}, m \in \{1, 2, 3\}$

8:     $Q_m^h, K_m^h, V_m^h = LinearNoBias(x_m)$ $Q_m^h, K_m^h, V_m^h \in \mathbb{R}^c$

9:     $a_{mn}^h = softmax_n\left(\frac{1}{\sqrt{c}} Q_m^{h\,T} K_n^h\right)$

10:     $O_m^h = \sum_n a_{mn}^h V_n^h$

11:     $O_m = Linear\left(concat_h(O_m^h)\right)$

12:     $a = softplus(O_0)$

13:     $b = softplus(O_1)$

14:     $c = sigmoid(O_2)$

15:     # Output projection

16:     $A_{ij}^h = softmax_j\left(av_{ij}^h - cd_{ij}^h + bp_{ij}^h\right)$

17:     **RETURN** $\{A_{ij}^h\}$

18: **END FUNCTION**

Algorithm 4 The Side-chain Structure Modelling and Information Interaction

Input:

$\{R_i\}$: Shape as $(N_{res}, 3, 3)$. Each element in N_{res} records the Euclidean transformation matrix of the corresponding residue.

$\{A_{ij}^h\}$: Shape as (N_{res}, N_{res}, H). Correlation matrix after modelling the dependencies between multi-scale features at the residue level.

$\{f_i^{pos\_CB}\}$: Shape as $(N_{res}, 3)$. The coordinates of the CB atom corresponding to the residue are recorded for each element of N_{res}.

$\{f_i^{pos\_CA}\}$: Shape as $(N_{res}, 3)$. The coordinates of the CA atom corresponding to the residue are recorded for each element of N_{res}.

Output:

$\{S_i\}$: Shape as $(N_{res}, 7H)$. Encoding of side-chain geometry features at the atomic level after establishing informative interactions with the residue level.

1:　　**FUNCTION** SideChainStructureModellingAndInformationInteraction($\{R_i\}, \{A_{ij}^h\}, \{f_i^{pos\_CB}\}, \{f_i^{pos\_CA}\}$):

2:　　　　# Mapping to multidimensional space

3:　　　　$f_i^{h\,pos\_CB} = \sum_j A_{ij}^h f_j^{pos\_CB}$

4:　　　　# Euclidean transformation

5:　　　　$f_i^{h\,pos\_CB} = R_i^T \left(f_i^{h\,pos\_CB} - f_i^{pos\_CA} \right)$

6:　　　　# Direction, distance and position

7:　　　　$\xi_i^h = f_i^{h\,pos\_CB}$　　　　　　　　　　　　$\xi_i^h \in \mathbb{R}^3$

8:　　　　$\zeta_i^h = norm\left(f_i^{h\,pos\_CB} \right)$　　　　　　　　$\zeta_i^h \in \mathbb{R}$

9:　　　　$\psi_i^h = \dfrac{f_i^{h\,pos\_CB}}{norm\left(f_i^{h\,pos\_CB} \right)}$　　　　　　　$\psi_i^h \in \mathbb{R}^3$

10:　　　　$S_i = concat_{\chi \in \{\xi, \zeta, \psi\}} \left(concat_h(\chi_i^h) \right)$　　　　$S_i \in \mathbb{R}^{7H}$

11:　　　　**RETURN** $\{S_i\}$

12:　**END FUNCTION**