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.

FUNCTION DynamicResiduesSelection(data, pos, core_index, mut_index,G , $\mathrm{lr_d}$, $\mathrm{N_{Maxstep}}$, $\mathrm{N_{res}}$):
    # Extract CA coordinates from pos
    SET pos_CA TO pos[:, 1, :]
    # Extract core and mutated CA coordinates
    SET pos_core_CA TO pos_CA[core_index]
    SET pos_mut_CA TO pos_CA[mut_index]
    # Initialize selection
    # Find the indices of the N_{res} residues closest to pos_mut_CA
    SET select_index TO indices of the $\mathrm{N_{res}}$ closest residues to pos_mut_CA based on pos_CA
    # Calculate the geometric center of all points in pos_core_CA
    SET pos_core_CA_center TO the geometric center of all points in pos_core_CA
    SET rnum TO 0
    # Iteratively refine selection
    WHILE (LENGTH(SET(core_index) - SET(select_index)) / LENGTH(core_index)) > (1 - G) **DO**
        # Move the virtual mutation point towards the core center
        SET pos_mut_CA TO pos_mut_CA moved towards pos_core_CA_center by $\mathrm{lr_d}$
        # Update select_index with the $\mathrm{N_{res}}$ closest residues to pos_mut_CA
        SET select_index TO indices of the $\mathrm{N_{res}}$ closest residues to pos_mut_CA based on pos_CA
        INCREMENT rnum BY 1
        IF rnum >= $\mathrm{N_{Maxstep}}$ **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, $i, j \in \{1, \dots, N_{res}\}$

4: $g_{ij} = sigmoid(LinearNoBias(d_{ij}))$ — $h \in \{1, \dots, H\}$, 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: $d_{ij}^{seq} = f_i^{seq} - f_j^{seq}$ — $i, j \in \{1, \dots, N_{res}\}$

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**

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

END IF

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

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

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

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

Output projection

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

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

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

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

RETURN $\{p_{ij}^h\}$

END FUNCTION

Algorithm 4 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.

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

Global node aggregation

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

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

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

Adaptive recalibration

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

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

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

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

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

$a = softplus(O_0)$

$b = softplus(O_1)$

$c = sigmoid(O_2)$

Output projection

$A_{ij}^h = softmax_j\left(a v_{ij}^h - c d_{ij}^h + b p_{ij}^h\right)$

RETURN $\{A_{ij}^h\}$

END FUNCTION

Algorithm 5 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.

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

Mapping to multidimensional space

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

Euclidean transformation

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

Direction, distance and position

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

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

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

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

RETURN $\{S_i\}$

END FUNCTION