

821

分子模拟+dl

分尺度

增强抽样算法

基于物理和数据的混合模型

- 小蛋白/小分子

统一在框架

前景

物理数据驱动

大模型

场景化2B应用 (覆盖各行业)

mindx SDK

原生加速网络库

- 1 大算子融合
- 2 整图下沉
- 3 自适应梯度切分

mindspore2.0新特性

回顾

物理驱动PINNs, PDE正向求解, 通过物理约束实现无监督

数据驱动alphafold2

物理加数据DeePMD, google流体力学, 小数据集训练AI力场再推理

编程新范式

函数式与面向对象融合编程

DualCore框架, AI和函数式编程原生融合, 共用同一套微分逻辑

即时编译

ms.jit修饰器

- 一行切换动静态图
- 即时编译, 被修饰函数转为整图

functorch

functorch需要手动将Module转换为function， mindspore直接支持Cell进行函数内调用和函数变换

numpy

mnp.

scipy

[ref](#)

Vmap

自动向量化特性，批处理逻辑从函数中脱离

ModelArts

AI落地难（开发算力人才）

科学计算基础操作

2023Summer

密钥

SHA256:senqZo6hCXlkeQtOmVxSPM/CGZZOz00b4+HFE/CyK98

```
1 git clone 地址
2 cd application
3 cd mindsponge
4 mkdir community
5 git status
6 cd applications
7 git add .
8 git commit -m "homework"
9
10 git push
11
12 bash build.sh -e ascend -j 128
```

822

业界趋势及实践

deepmind加速排序

[论文地址](#)

二度回顾PINNs代表的物理及数据驱动

布局加套件落地

流体Flow

大模型

- 数据预处理
- vit
- 小波变换

SOTA模型套件SciAI

待发布

分子动力学

模拟原理拓展和应用

模拟原理

回顾发展

- 1971年Lee和Richards提出了可及表面的概念；
- 1971年Stillinger和Rahman模拟了液态水的分子动力学过程；
- 1976年Warshel和Levitt第一次采用QM/MM的方法研究了溶菌酶的催化机制；
- 1977年McCammon, JA等人报道了第一类蛋白质分子动力学模拟；
- 1977年约束算法SHAKE被引入分子动力学的模拟；
- 1978年Streett W. B., Tildesley D. J.以及Saville G.引入Multiple-time step的积分方法；
- 1979年Levitt等人采用了分子动力学方法揭示了X射线晶体衍射B因子的起源；
- 1980年Journal of Computational Chemistry杂志创刊；
- 量子力学方法

薛定谔方程采用近似，精度越高计算越难

- 分子力学方法

经典力学描述等价量子效应

各原子笛卡尔坐标

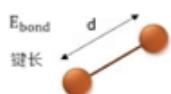
AMBER force field model: two body additive model

Coulombic In Nature

- Atoms in ionic materials behave like point charges
- Simple to describe by Coulomb potential
- Very long-range interactions

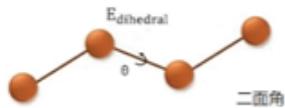
$$E_{coul} = \frac{1}{4\pi\epsilon_0} \frac{\overbrace{q_i q_j}^{\text{Charge on atom } i \text{ and } j}}{\underbrace{r_{ij}}_{\text{Distance between atom } i \text{ and } j}}$$

传统分子力场的数学形式

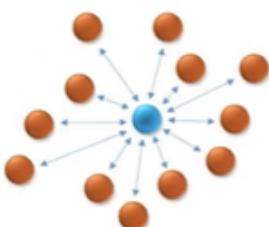


键长 : $E_b(l) = \frac{1}{2} k_l (l - l_0)^2$

键角 : $E_\theta(\theta) = \frac{1}{2} k_\theta (\theta - \theta_0)^2$



二面角 : $E_t(\omega) = \sum_n \frac{V_n}{2} [1 - \cos(n\omega - \gamma_n)]$

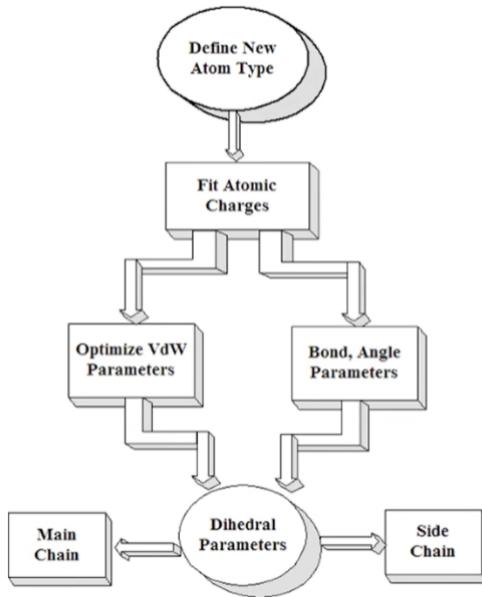


静电相互作用 : $E_{ele} = \sum_{i < j} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$

范德华 (van der Waals) 相互作用 : $E_{vdW} = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$

$E_{non-bonded} = E_{elec} + E_{vdW}$
非键 (静电+范德华) 相互作用

分子力场开发流程



过场繁琐，尝试ml替代

突破分子模拟局限性

模拟尺度小

- no special condition - simulation in vacuum (bad idea)
- keeping the outer molecules rigid or constraint (ice-like conditions)
- applying additional forces when molecules try to escape from a certain container region
- periodic boundary conditions

未来展望

分子动力学模拟-时间平均

实验观测-系综平均

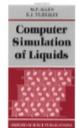
各态历经Ergodicity

- 推荐书单

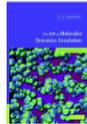
- Frenkel & Smit, Understanding Molecular Simulation:
From Algorithms to Applications. Probably most through text on MD.



- Allen & Tildesley, Computer Simulation of Liquids.
Great MD intro and algorithms.



- Rapaport, The Art of Molecular Dynamics Simulation.
MD cookbook text, ideal for developing your own code.



智能分子模拟套件

体系模板Template

速度Verlet积分器

约束控制器 (linear constraint solver) 约束体系键长

温度的生成与计算，温度控制器Thermostat Controller

积分器与控温器的协同工作

mindspunge主程序是sponge，将体系、势能和优化器三模块组装起来

近邻表Neighbour List

基于偏置势

埋拓动力学Metadynamics

基于能量包装器的增强采样方法

运行模拟与回调函数Callback

集成变量Colvar与度量函数Metrics

H5MD模拟轨迹文件

QM/MM

半经验方法为主

hartreefock与DFT

AM1半经验方法

- 单中心积分近似
- 双中心积分近似

精确刻画势能面

启发其他

上机 | 建模与模拟

tutorial_c01.ipynb

tutorial_c02.ipynb

作业

Homework

1. Build a PBC system for case1.pdb, run a normal product simulation of 10 ps, plot the phi-psi distribution.

Hints: mainly base on tutorial_p03, and add CVs output for phi and psi.

2. Use the same system, run a MetaD simulation of 10 ps, plot the phi-psi distribution.

Hints: add MetaD modules, and modify the WithEnergyCell as tutorial_p04 done.

3. compare the results, and analysis the effort of MetaD.

Upload attachments contains:

homework.py

homework.ipynb

results.jpg or .png): plot the phi-psi distribution results of (a) normal MD and (b) MetaD.

1. 为 case1.pdb 构建一个 PBC 系统，运行10ps 的正常产品模拟，绘制 phi-psi 分布图。

提示: 主要基于 p03，并添加 phi 和 psi 的CVs输出。

2. 使用相同的系统，运行10ps 的 MetaD 模拟，绘制 phi-psi 分布图。

提示: 添加 MetaD 模块，并将 WithEnergyCell 作为 p04 的done。

3. 比较结果，并分析 MetaD 作用。

1

tutorial_p02.py: 读取1个pdb蛋白质分子，构建周期性水盒子，做能量极小化。

tutorial_p03.py: 读取p02保存的水盒子，进行蛋白模拟流程示例：NVT模拟—NPT模拟—成品模拟。

p02保存的水盒子是p02.pdb

phi-psi

Barostat cannot be used for the system without periodic boundary condition.

823

药物设计

药物发现途径

- 基于现象发现
- 基于靶标发现

高通量筛选

药物发挥作用

锁钥模型，药物与作用靶标结合

非共价结合

化合物对蛋白质活性的控制

直接占据活性部位（抑制）

别构调控（激活或抑制on/off state） Allosteric regulation

在临床实验上筛选部分

药物设计分类

- 基于靶标三维结构
- 不依赖靶标三维结构

计算机辅助蛋白质结合化合物发现

- 搜索已知化合物数据库
- 片段连接方法
- 从头生长de novo

分子对接具体筛选

蛋白质结合口袋探测算法

存在问题

可靠有效算法

准确打分函数（精确结合自由能计算）

靶标结构柔性的处理

基于ms图神经网络的应用

图学习任务与方法

- 边任务、节点任务、整图任务
- 深度学习、矩阵分解法、统计方法、随机游走

生物组学的网络推理

降本增效

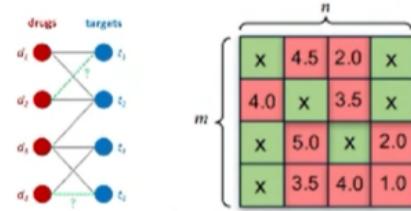
分子性质预测

矩阵分解法



矩阵分解法 (Matrix Factorization)

- 以老药新用 (Drug Repurposing) 为例:

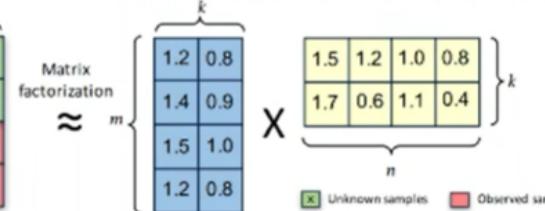


药物-靶点互作的邻接矩阵为Y，其中对应的每个元素的值为 N_{ij} ，即：

$$N_{ij} = \begin{cases} 1 & j \in \mathcal{N}_p(i) \& i \in \mathcal{N}_p(j) \\ 0 & j \notin \mathcal{N}_p(i) \& i \notin \mathcal{N}_p(j) \\ 0.5 & \text{otherwise} \end{cases}$$

- 将Y分解为A和B的乘积，即：

估计出规模分别为 $m \times k$ 的药物特征矩阵A和 $n \times k$ 的靶点矩阵B，



使得 AB^T 的值尽可能逼近矩阵D，即：

$$\begin{aligned} \min_{A,B} \quad & \|Y - AB^T\|_F^2 \\ & + \lambda_f(\|A\|_F^2 + \|B\|_F^2) \\ & + \lambda_d Tr(A^T \mathcal{L}_d A) \\ & + \lambda_t Tr(B^T \mathcal{L}_t B) \end{aligned}$$

通过分别对A, B求导，并且 $\frac{\partial L}{\partial A} = \frac{\partial L}{\partial B} = 0$ ，可得：

$$\begin{aligned} A &= (YB - \lambda_d \mathcal{L}_d A)(B^T B + \lambda_t I_k)^{-1} \\ B &= (Y^T A - \lambda_t \mathcal{L}_t B)(A^T A + \lambda_d I_k)^{-1} \end{aligned}$$

随机游走 (偏dfs)

走不同结点概率衡量



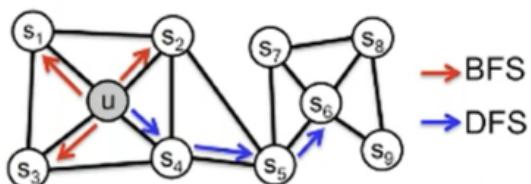
随机游走(Random Walk)

核心概念：

- 任何无规则行走者所带的守恒量都各自对应着一个扩散运输定律。

基本思想：

1. 从一个或一系列顶点开始遍历一张图。
2. 在任意一个顶点，遍历者将以概率 $1-a$ 游走到这个顶点的邻居顶点。
3. 以概率 a 随机跳跃到图中的任何一个顶点，称 a 为跳转发生概率。
4. 每次游走后得出一个概率分布，该概率分布刻画了图中每一个顶点被访问到的概率。
5. 用这个概率分布作为下一次游走的输入并反复迭代这一过程。
6. 当满足一定前提条件时，这个概率分布会趋于收敛，收敛后，即可以得到一个平稳的概率分布。



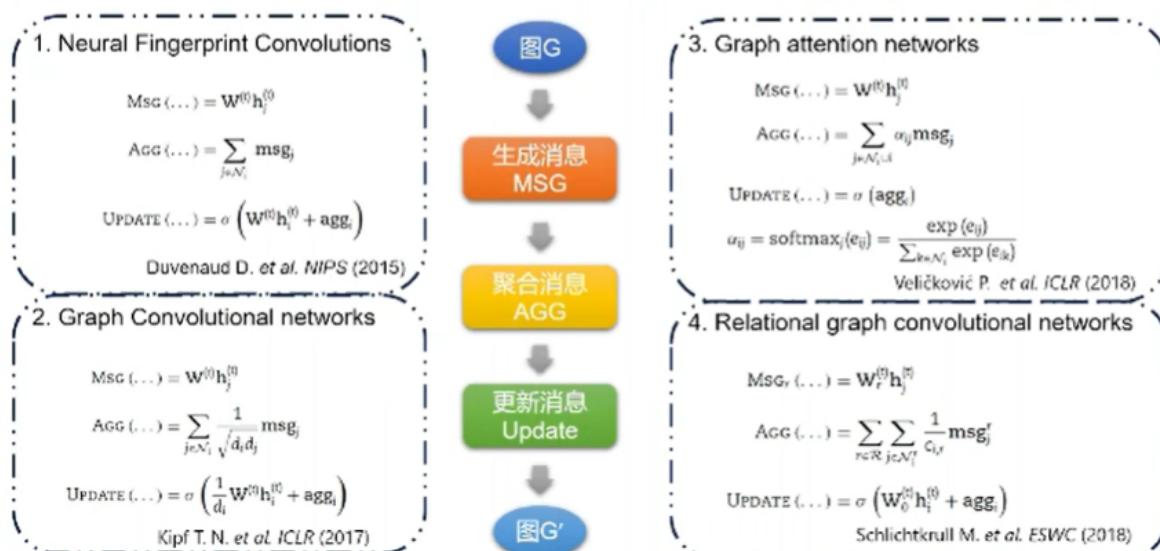
Input: the similarity network $G = (V, E)$;
a start node s_i ;
restart probability c_i ;
Output: the proximity vector $p_s^*(V)$;

Let $r_s^*(V)$ be the restart vector with 0 for all its entries except a 1 for the entry denoted by node s_i ;
Let \mathbf{A} be the column normalized adjacency matrix defined by E ;

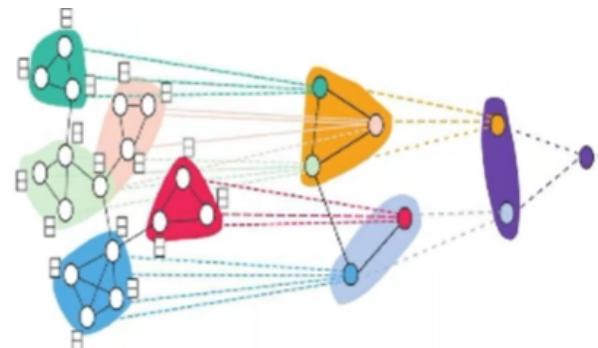
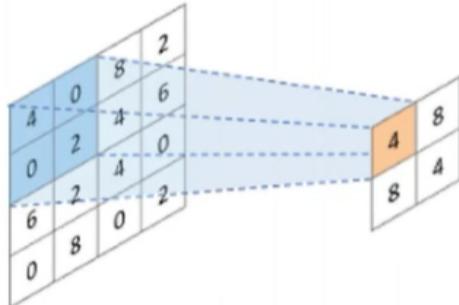
Initialize $p_s^*(V) := r_s^*(V)$;
while ($p_s^*(V)$ has not converged)
 $p_s^*(V) := (1 - c_i)\mathbf{A}p_s^*(V) + cr_s^*(V)$;

红色箭头走领域，类卷积想法，ReLU

消息传递架构



池化前后传播



最大池化 (Max Pooling)

- 前向传播：选图像区域的最大值作为该区域池化后的值。
- 反向传播：梯度通过最大值的位置传播，其它位置梯度为0

平均池化 (Mean pooling)

- 前向传播：计算图像区域的平均值作为该区域池化后的值
- 反向传播：梯度取均值后分给每个位置

图的池化

挑战性：

- 图的结构没有确定的拓扑结构（节点数和边的数量不固定）
- 没有明确的局部空间的概念（不能用卷积核去对周围元素操作）

任务：

- 建立一个通用的、端到端的、可微的多个层次化模型

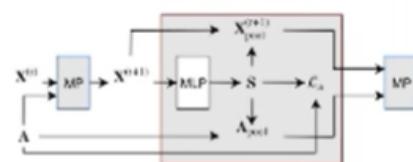
1. 学习分配矩阵 (DiffPool) :

$$Z^{(l)} = \text{GNN}_{l, \text{tent}}(A^{(l)}, X^{(l)})$$

$$S^{(l)} = \text{softmax} \left(\text{GNN}_{l, \text{pool}}(A^{(l)}, X^{(l)}) \right)$$

2. 分割图(MinCUT) :

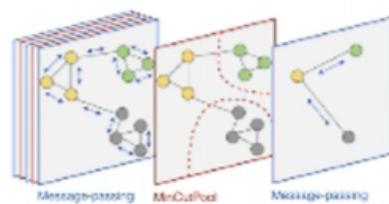
$$\frac{1}{K} \sum_{k=1}^K \frac{\text{links}(V_k)}{\text{degree}(V_k)} = \frac{1}{K} \sum_{k=1}^K \frac{\sum_{i,j \in V_k} E_{i,j}}{\sum_{i \in V_k, j \in V \setminus V_k} E_{i,j}}$$



3. 用分配矩阵来逐层池化

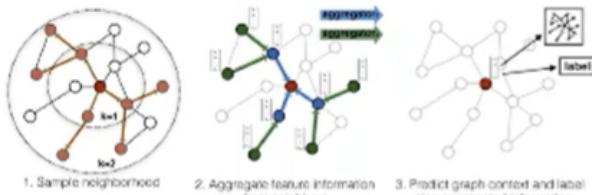
$$X^{(l+1)} = S^{(l)}{}^T Z^{(l)} \in \mathbb{R}^{m_{l+1} \times d},$$

$$A^{(l+1)} = S^{(l)}{}^T A^{(l)} S^{(l)} \in \mathbb{R}^{m_{l+1} \times m_{l+1}}$$



图卷积的表征学习

大模型的前身



Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

```

Input : Graph  $\mathcal{G}(V, E)$ ; input features  $\{\mathbf{x}_v, \forall v \in V\}$ ; depth  $K$ ; weight matrices  $\mathbf{W}^k, \forall k \in \{1, \dots, K\}$ ; non-linearity  $\sigma$ ; differentiable aggregator functions  $\text{AGGREGATE}_k, \forall k \in \{1, \dots, K\}$ ; neighborhood function  $N : v \rightarrow 2^V$ 
Output : Vector representations  $\mathbf{z}_v$  for all  $v \in V$ 
1.  $\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in V$ ;
2. for  $k = 1..K$  do
3.   for  $v \in \mathcal{V}$  do
4.      $\mathbf{h}_{N(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in N(v)\})$ ;
5.      $\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{N(v)}^k))$ 
6.   end
7.    $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in V$ 
8. end
9.  $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in V$ 
```

23-Aug-23

Hamilton W. L. et al. NIPS (2017) 18

word2vec邻域

基于随机游走的表征学习

Deepwalk v.s. Node2vec

Algorithm 1 DEEPWALK(G, w, d, γ, t)

```

Input: graph  $G(V, E)$ 
  window size  $w$ 
  embedding size  $d$ 
  walks per vertex  $\gamma$ 
  walk length  $t$ 
Output: matrix of vertex representations  $\Phi \in \mathbb{R}^{|V| \times d}$ 
1: Initialization: Sample  $\Phi$  from  $\mathcal{U}^{|V| \times d}$ 
2: Build a binary Tree  $T$  from  $V$ 
3: for  $i = 0$  to  $\gamma$  do
4:    $\mathcal{O} = \text{Shuffle}(V)$ 
5:   for each  $v_i \in \mathcal{O}$  do
6:      $\mathcal{W}_{v_i} = \text{RandomWalk}(G, v_i, t)$ 
7:     SkipGram( $\Phi, \mathcal{W}_{v_i}, w$ )
8:   end for
9: end for

Algorithm 2 SkipGram( $\Phi, \mathcal{W}_{v_i}, w$ )
1: for each  $v_j \in \mathcal{W}_{v_i}$  do
2:   for each  $u_k \in \mathcal{W}_{v_i}[j - w : j + w]$  do
3:      $J(\Phi) = -\log \Pr(u_k | \Phi(v_j))$ 
4:      $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$ 
5:   end for
6: end for
```

23-Aug-23 Perozzi B. et al. doi:10.1145/2623330.2623732 (2014)

共同点:



不同点:

- DeepWalk中是将图转换为二叉树随机抽取邻接点
 - Node2vec采用的是一种有偏的随机游走，采用Alias算法而是按概率进行节点采样进行采样：
- $$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{d_x} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$
- $$\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx} \quad \alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$
- 若 p 较高，则访问刚刚访问过的顶点的概率会变低，反之变高。
 - 若 $q > 1$ ，随机游走倾向于访问和接近的顶点(偏向BFS)。
 - 若 $q < 1$ ，倾向于访问远离的顶点(偏向DFS)。

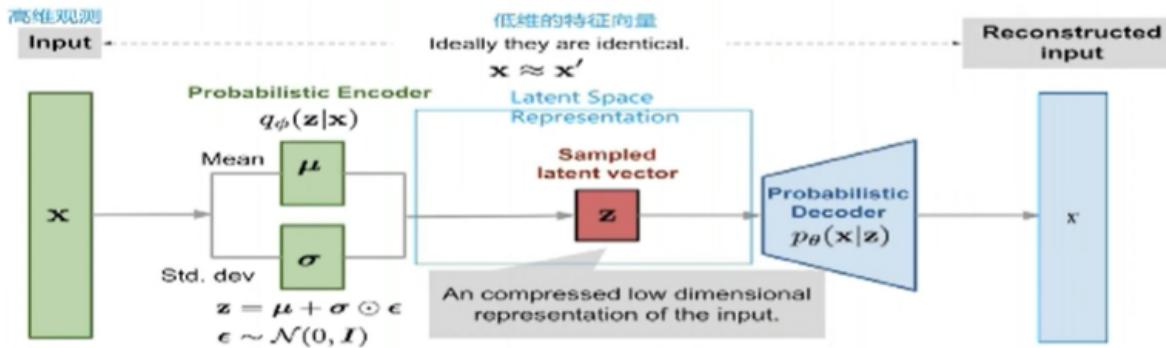
Grover A. & Leskovec J. KDD (2016)

19

借鉴思想，算力有限追求速度

变分自编码 (VAE)

正态分布空间中采样



在VAE网络中，对于所有样本，我们希望最大化对数似然率，即：

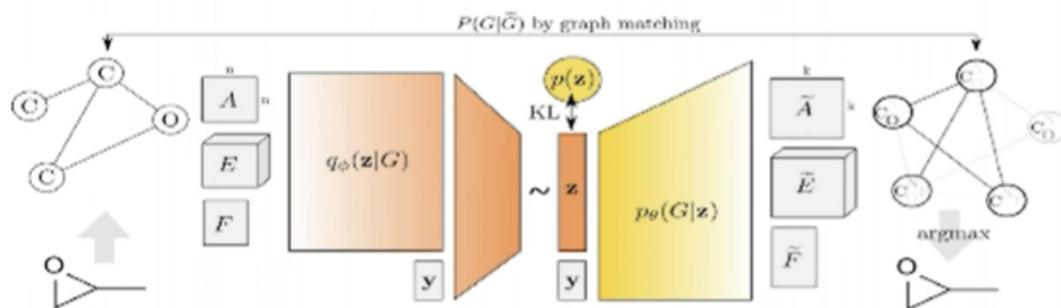
$$\log p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \sum_{i=1}^N \log p(\mathbf{x}_i),$$

为了优化该目标，可以优化下式：

$$\log p(\mathbf{x}) = KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) + L(\Phi, \Theta; \mathbf{x})$$

$$L(\Phi, \Theta; \mathbf{x}) = E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \frac{p_\theta(\mathbf{z}, \mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}]$$

图变分自编码 (Graph VAE)

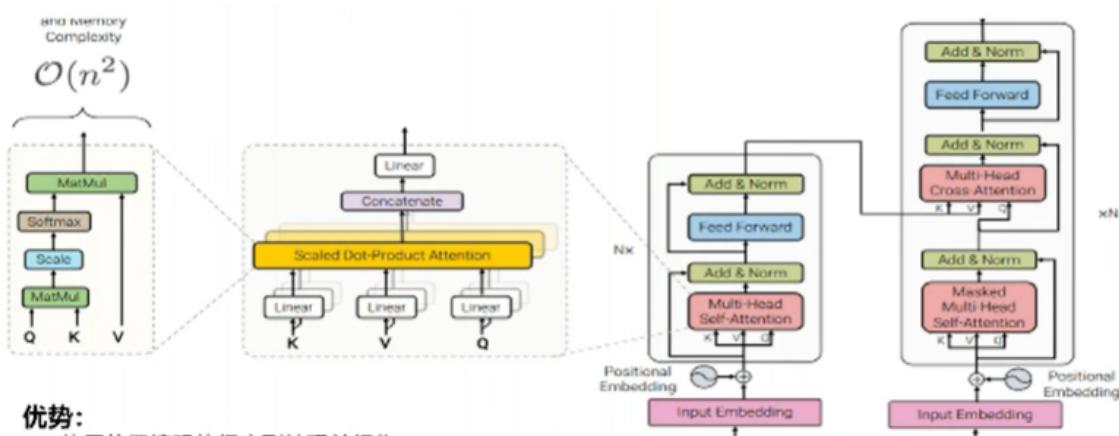


Kipf (2016) 的做法：

$$\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \quad p(\mathbf{A} | \mathbf{Z}) = \prod_{i=1}^N \prod_{j=1}^N p(A_{ij} | \mathbf{z}_i, \mathbf{z}_j) \quad \tilde{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^\top), \text{ with } \mathbf{Z} = \text{GCN}(\mathbf{X}, \mathbf{A})$$

归一化再重构，向量到矩阵做外积

Transformer



优势：

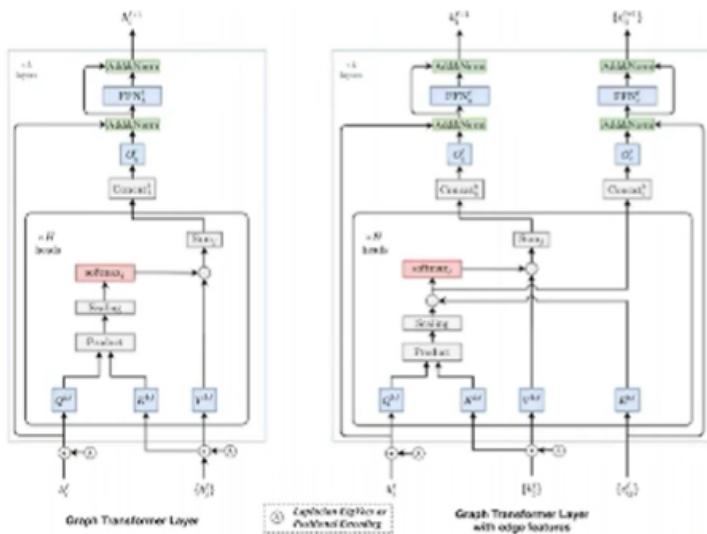
1. 使用位置编码使得序列处理并行化
2. 使用残差连接增加网络深度
3. 使用Multi-head 注意力机制

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

劣势：

1. 算法复杂度变为 $O(n^2)$
2. 对于长序列开销较大

节点的位置编码



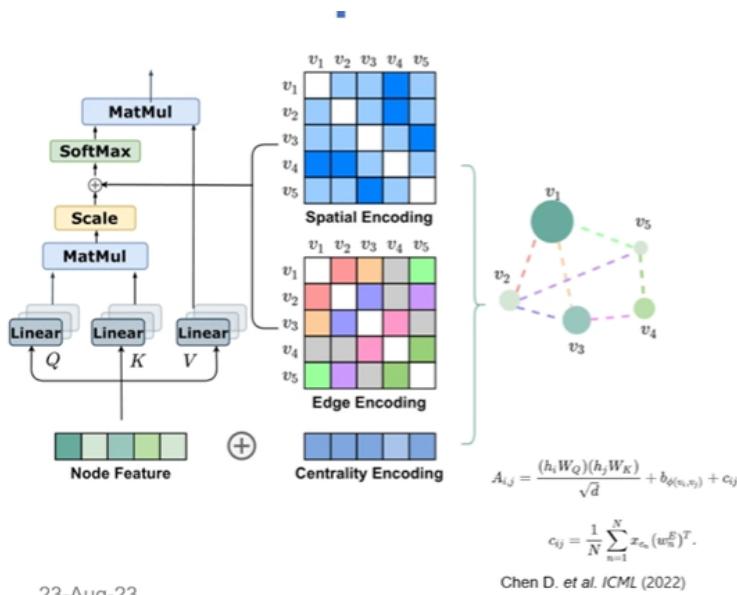
- GT:**
• 拉普拉斯矩阵的本征值

- GraphiT:**
• 基于正定核的注意力得分加权的相对位置编码策略
• 通过利用图卷积核网络 (GCKN) 将 small sub-structure 编码出来作为输入。

- SAN:**
• 使用 learnable PE, 对为什么 laplacian PE 比较有效进行了比较好的说明

<https://zhuanlan.zhihu.com/p/536489997>

Graphomer



23-Aug-23

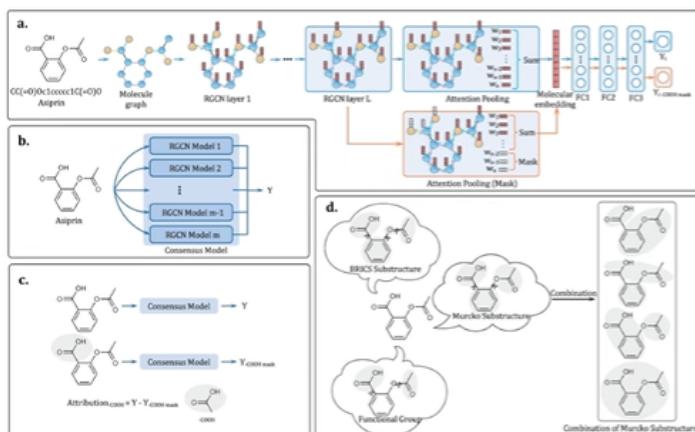
24

- 1. Centrality Encoding (中心性编码)。**
• 一个节点的“度” (degree) 越大, 代表这个节点与其他节点相连接的边越多, 那么往往这样的节点就会更重要

- 2. Spatial Encoding (空间编码)。**
• 实际上图结构信息不仅包含了每个节点上的重要性, 也包含了节点之间的相关性。
• 例如: 邻居节点或距离相近的节点之间往往相关性比距离较远的节点相关性高。
• 采取了无权的最短路径作为空间编码的距离度量。为其分配相应的编码向量。

- 3. Edge Encoding (边信息编码)。**
• 连边上的信息有非常重要的作用, 例如连边上的距离、流量等等。
• 将连边上的信息作为权重偏置 (Bias) 引入注意力机制中来在计算两个节点之间的相关性。
• 两个节点最短路径上的连边特征进行加权求和作为注意力偏置, 其中权重是可学习的。

可解释性



基本思路:

基于扰动的方法的关键是如何对输入进行扰动, 将某些原子/键/碎片从GNN模型中隐藏起来, 以发现哪些子结构在输入中不存在时可能会极大地影响模型的预测。

23-Aug-23

Wu Z. et al. *Nature Communications* (2023)

预测任务:

- 亲水性 (ESOL)
- 致突变性 (Mutagenicity)
- 心脏毒性 (hERG)
- 血脑屏障渗透性 (BBBP)

网络架构:

RGCN + Attention Pooling

归因值 (Attribution):

在分子图上应用子结构掩盖之前和之后进行两次预测, 预测值之间的差值即可作为该子结构的归因值。

$$Y = \sum_i^m Y_i$$

$$Y_{sub} = \sum_i^m Y_{i,sub}$$

$$\text{Attribution}_{sub} = Y - Y_{sub}$$

26

黑盒模型

任务:

- 预测针对多个靶点的药物小分子的 pIC_{50}

评价指标:

$$\text{MAE} (y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

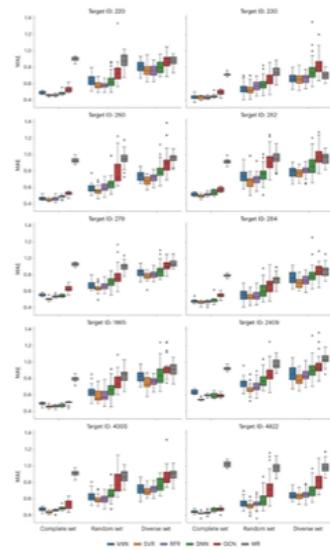
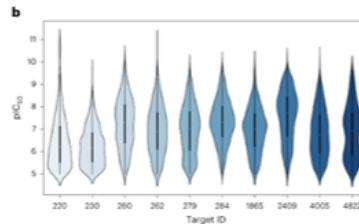
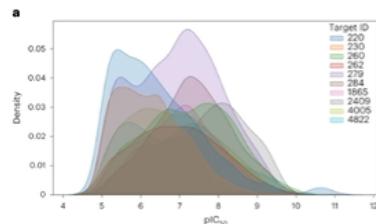
$$\text{RMSE} (y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

方法:

- 支持向量机 (SVR)
- 随机森林 (RFR)
- 全连接深度神经网络 (DNN)
- 图神经网络 (GCN)
- 均值回归 (MR, 对照)

结果:

SVR >= kNN > RFR >> DNN > GCN



23-Aug-23

Tiago Janela & Jürgen Bajorath *nature machine intelligence* (2022)

27

群论和等变性

1. 群 (group)

- 群G是一组具有满足这些的二元运算“.”属性的变换:
- 这些属性存在一个标识元素,
- 每个元素G存在一个逆,使得在associative composition下是闭集的。

2. 等变性 (Equivariance):

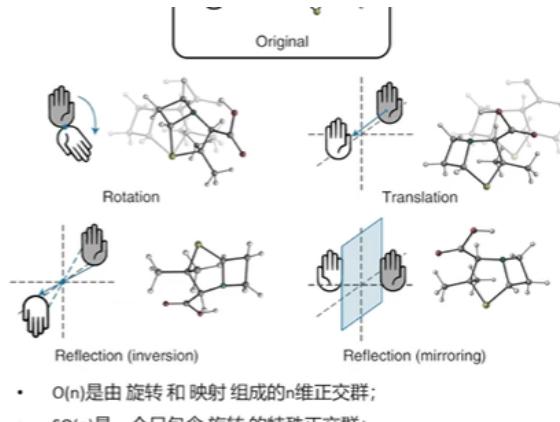
- 对于一个函数/特征以及一个变换,如果我们有:
 $f(g(x)) = g(f(x))$
- 则称变换为等变的。即对其输入施加的变换也会同样反应在输出上

3. 不变性 (invariance)

- 对于特征,进行变换:

$$f(x) = f(g(x))$$

- 则称变换 f 对变换 g 具有不变性。



aichemist算法包

不同场景训练不同模型, 图神经网络动态构建

构建Graph类

GNN实现

message\aggregate\combine

```

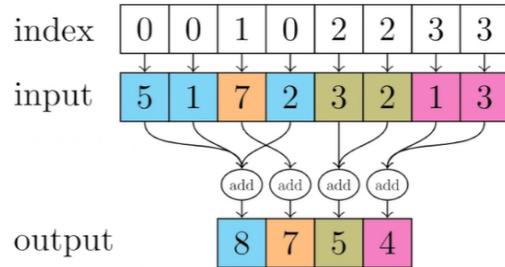
def message(self, graph, inputs):
    node_in = graph.edges[0]
    message = inputs[node_in]
    edge_weight = ops.expand_dims(graph.edge_weight, axis=-1)
    if self.edge_linear:
        message += self.edge_linear(graph.edge_feat)
    message *= edge_weight
    return message

def aggregate(self, graph, message):
    node_out = graph.edges[1]
    update = scatter_add(message, node_out, axis=0, n_axis=graph.n_node)
    return update

def combine(self, inputs, update):
    output = self.linear(inputs + update)
    if self.batch_norm:
        output = self.batch_norm(output)
    if self.activation:
        output = self.activation(output)
    return output

```

22 8.12 22



• Pytorch库 (PyG: pytorch_scatter)

```

from torch_scatter import scatter

src = torch.randn(10, 6, 64)
index = torch.tensor([0, 1, 0, 1, 2, 1])

# Broadcasting in the first and Last dim.
out = scatter(src, index, dim=1, reduce="sum")

print(out.size())

```

基于传统和机器学习方法的分子对接程序DSDP的介绍与应用

每年住院人次1亿/y > 一款药物研发十几年

CADD->AIDD计算机到AI辅助药物设计

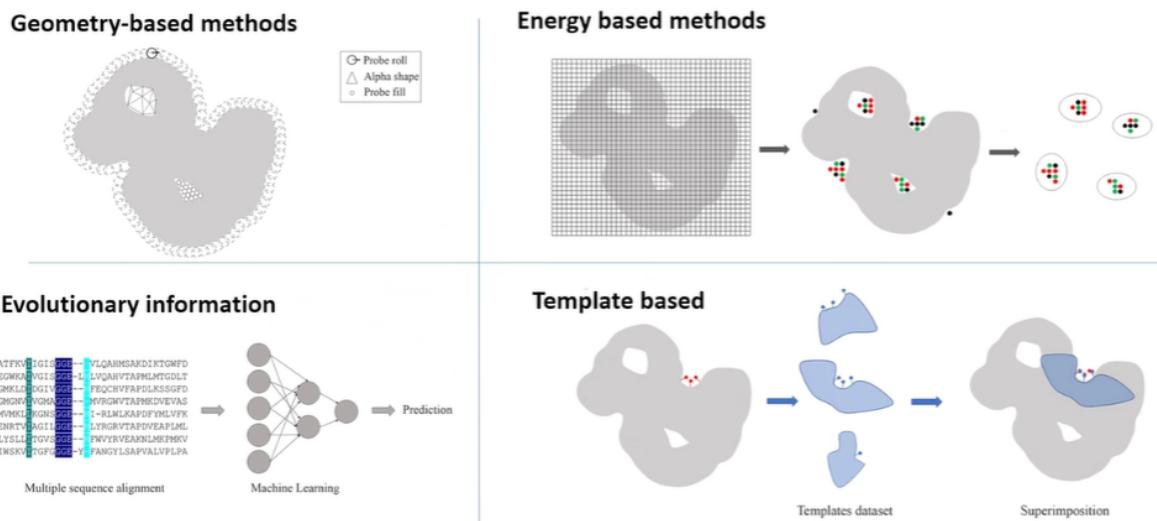
分子对接流程概述



去重排序/成药性分析 improve sr

```
1 from rdkit import Chem  
2 ...  
3 AllChem.GetMorganFingerprint()
```

蛋白质结合口袋预测



机器学习引入位点预测的两种策略

- 与传统方法相结合提高传统方法的精度
 - 端到端机器学习方法

构象采样

刚性对接，半柔性对接，柔性对接

传统随机采样算法：蒙特卡洛，遗传算法

EquiBind

机器学习：EquiBind

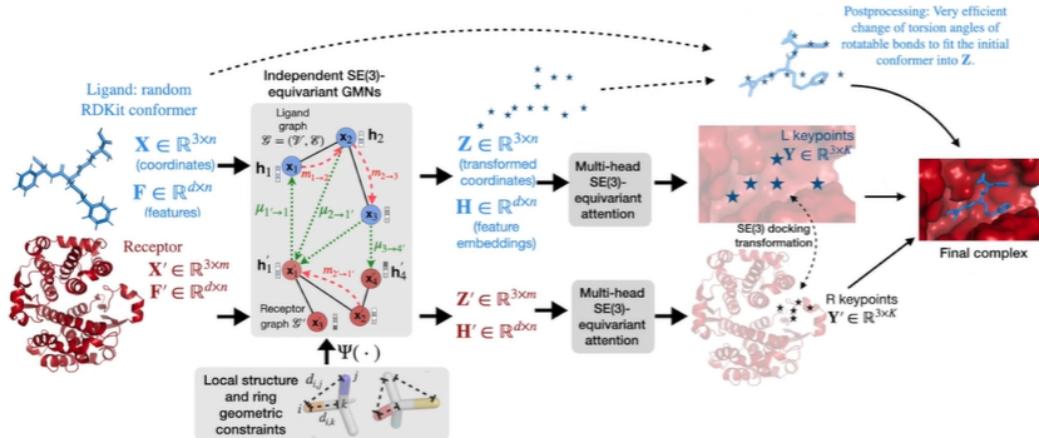
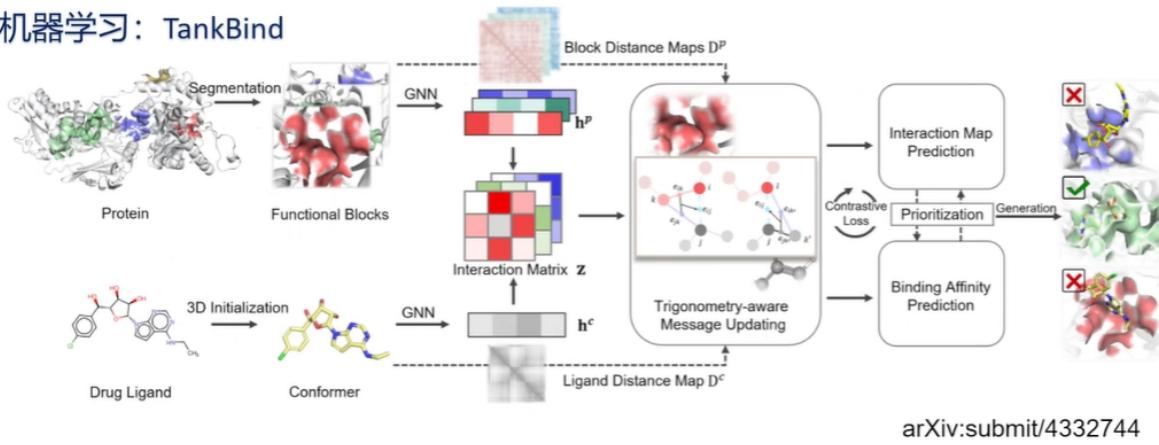


Figure 2. EQUIBIND model architecture.

arXiv:2202.05146v1

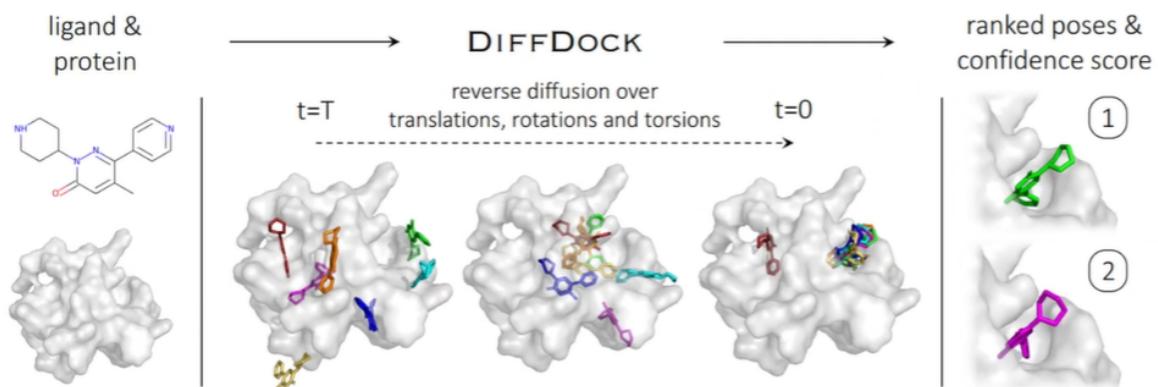
TankBind

机器学习：TankBind



DiffDock

机器学习：DiffDock

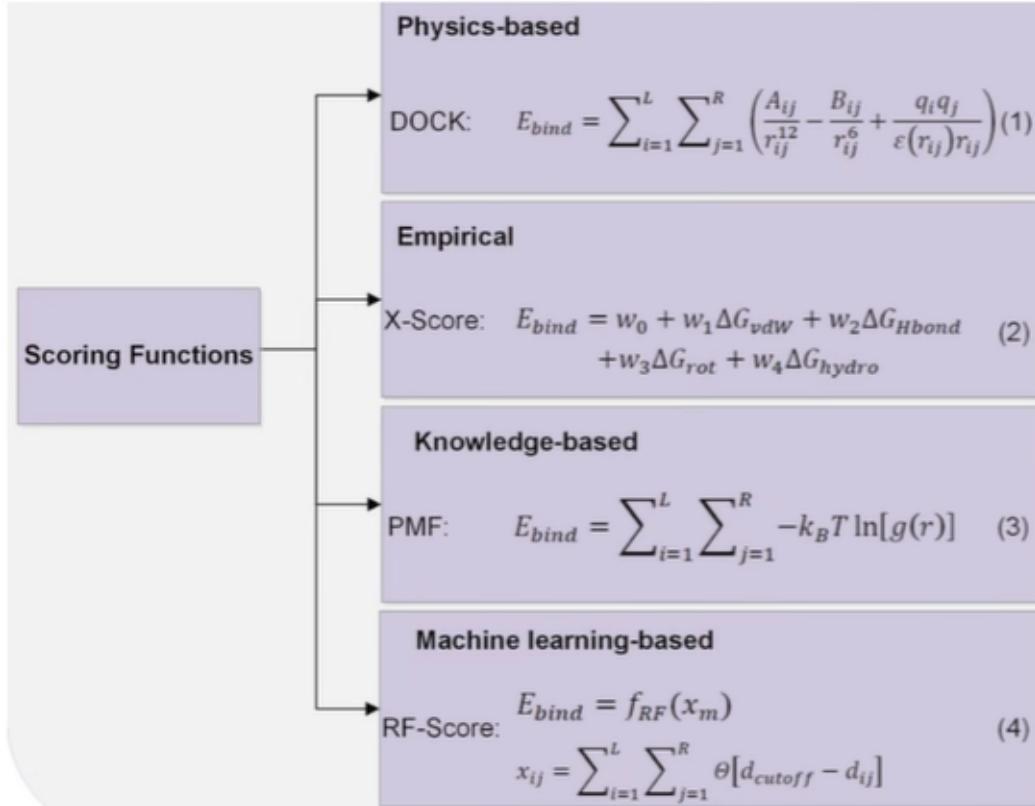


Method	Top-1 RMSD (Å)		Top-5 RMSD (Å)		Average Runtime (s)
	%<2	Med.	%<2	Med.	
QVINAW	20.9	7.7			49*
GNINA	22.9	7.7	32.9	4.5	127
SMINA	18.7	7.1	29.3	4.6	126*
GLIDE	21.8	9.3			1405*
EQUIBIND	5.5	6.2	-	-	0.04
TANKBIND	20.4	4.0	24.5	3.4	0.7/2.5
P2RANK+SMINA	20.4	6.9	33.2	4.4	126*
P2RANK+GNINA	28.8	5.5	38.3	3.4	127
EQUIBIND+SMINA	23.2	6.5	38.6	3.4	126*
EQUIBIND+GNINA	28.8	4.9	39.1	3.1	127
DIFFDOCK (10)	35.0 ± 1.4	3.56 ± 0.05	40.7 ± 1.6	2.65 ± 0.10	10
DIFFDOCK (40)	38.2 ± 1.0	3.30 ± 0.11	44.7 ± 1.7	2.40 ± 0.12	40

arXiv:2210.01776v1

蛋白质小分子相互作用评估

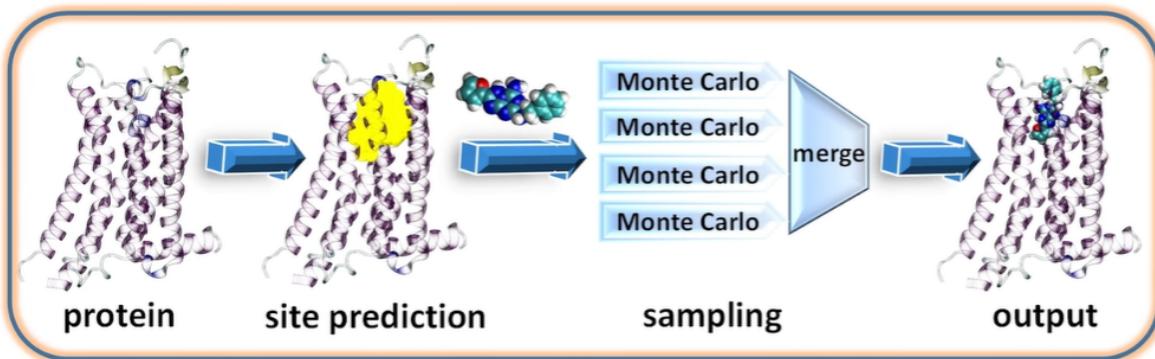
打分函数



Interdiscip. Sci.: Comput. Life Sci. 2019, 11, 320–328.

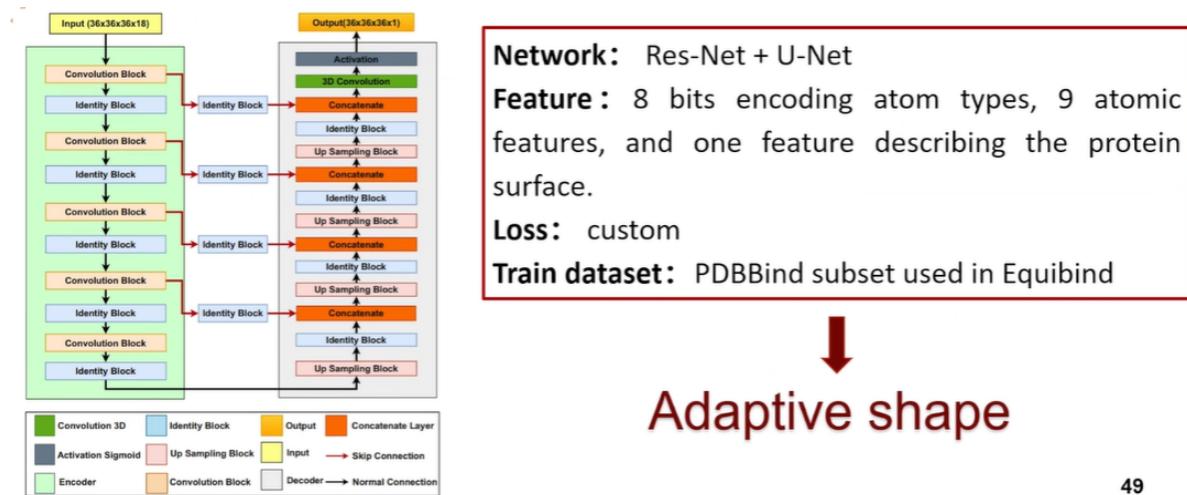
多个数据进行打分

DSDP传统方法与ml优势结合



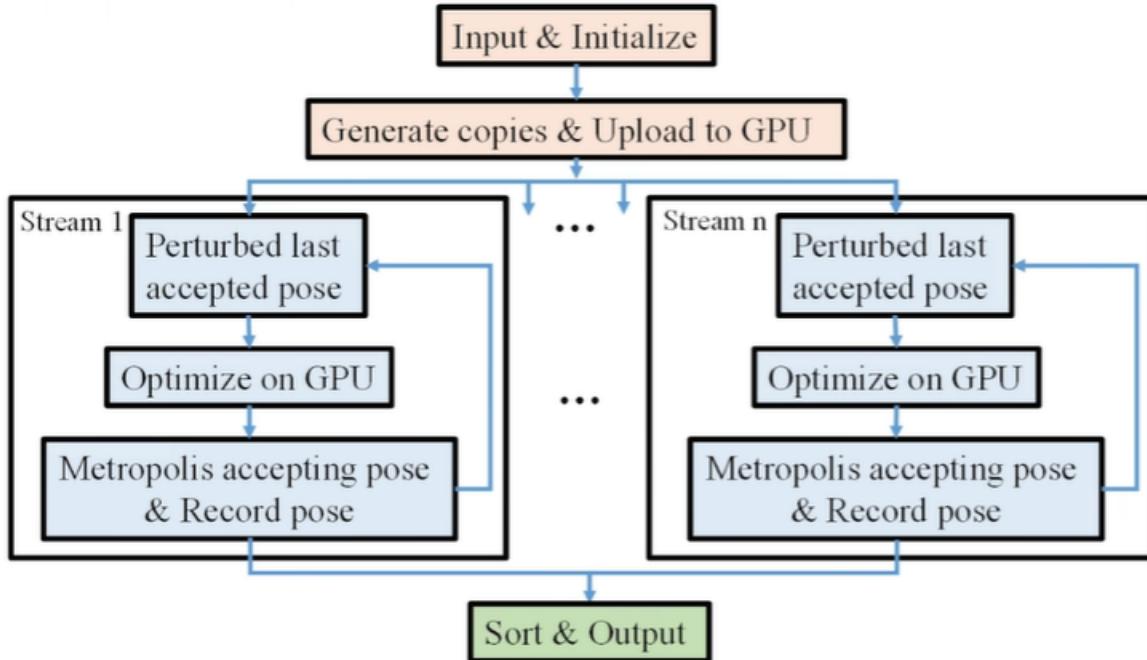
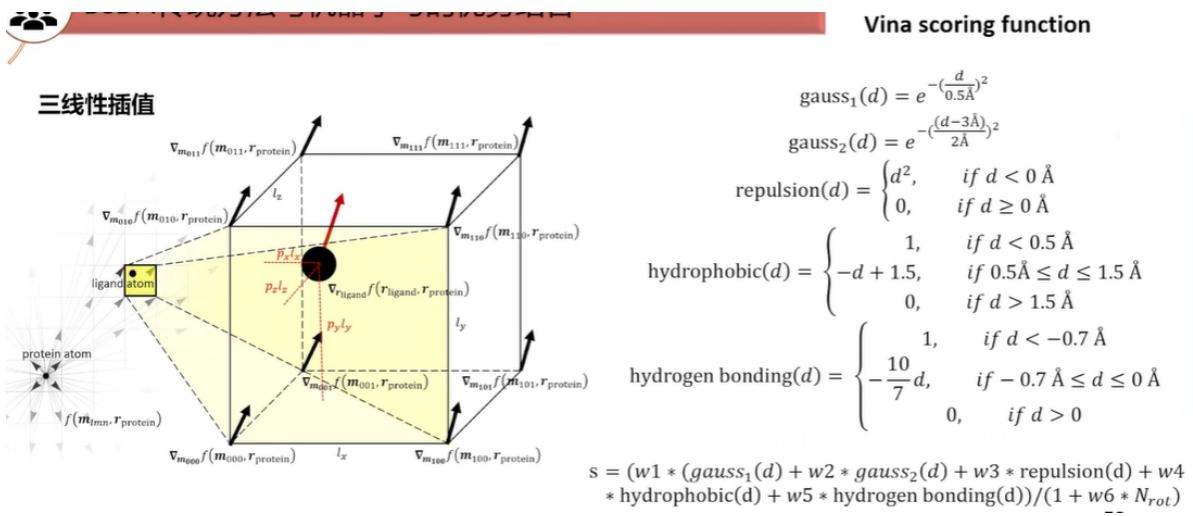
DSDP: A Blind Docking Strategy Accelerated by GPUs.

48



49

三线性插值



conclusion

1. 传统方法与机器学习在分子对接中都表现出各自的优势，其中机器学习在位点预测与蛋白-小分子相互作用预测方面表现较好
2. DSDP结合了机器学习与传统算法的优势，减少盲对接的采样范围，能快速准确的实现redocking, blind docking, virtual screening等任务，在多个测试集中都表现出不错的效果
3. 在后续工作中，将更准确的打分函数引入到对接过程中是提高 docking 精度的关键

MolEdit基于生成式学习的分子编辑

功能分子设计

基团锚定再link

AI时代

生成式学习

比如给蛋白口袋/分子片段/结构/分子性质

- 分子式化学空间
 - 离散，表示简洁，优化难，要近似
 - 数据多
 - 分子图有稀疏性
- 坐标空间
 - 物理上完备表示，良定义梯度优化
 - 数据少
 - 构象采样额外复杂度

选取合适模型

MolGAN：分子图+GAN

分子图是稀疏的

EDM：3D结构+扩散模型

药物分子的性质预测及分子对接

作业

1. 图卷积网络（动态图）

- 网络结构：NeuralFP
- 数据集：Tox21数据集预测

• 几何神经网络（静态图）

- 网络结构：MolCT
- 数据集QM9

• 3D点云卷积网络（静态图）

- 网络结构：3d convolution + Residual Connection
- 数据集：PDBBind（子集）
- 任务：预测蛋白质空腔（结合口袋）

作业：扩展上机用例1

1. 实现在训练过程中，将loss值最小的模型保存
2. 加载最好的模型，并输出在验证集上的表现结果；如损失值，准确率等
3. 加载最好的模型，并对测试集进行预测，画出AUC曲线

进阶作业

利用图VAE来构建图表示学习

1. 输入: 分子图
2. 输出: 标准化的邻接矩阵
3. 数据集: Zinc250k
4. 编码器: NeuralFP (忽略原子坐标)
5. 解码器: MLP
6. 损失函数:

1. 交叉熵
2. KL散度

$$\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$$

$$p(\mathbf{A} | \mathbf{Z}) = \prod_{i=1}^N \prod_{j=1}^N p(A_{ij} | \mathbf{z}_i, \mathbf{z}_j)$$

$$\hat{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^\top), \text{ with } \mathbf{Z} = \text{GCN}(\mathbf{X}, \mathbf{A})$$

代码
样例

```
class Decoder(nn.Cell):
    def __init__(self, in_dim, out_dim):
        super().__init__()

    def construct(self, inputs):
        ...

class GVAE(nn.Cell):
    def __init__(self, input_dim, output_dim, hidden_dims, adj_dim):
        super().__init__()
        self.adj_dim = adj_dim
        self.gcn_mean = NeuralFingerprint(input_dim, output_dim, hidden_dims)
        self.gcn_logstd = NeuralFingerprint(input_dim, output_dim, hidden_dims)
        self.decoder = Decoder(output_dim, output_dim, adj_dim)

    def construct(self, graph):
        ...

    def loss_fn(self, **kwargs):
        ...
```

提示:

1. 用scatter函数将边矩阵转换为邻接矩阵
2. 用value_and_grad函数进行自动微分
3. 使用动态图, 忽略分子对齐 (padding)

参考文献: Kipf N. T. et al. NIPS (2016)

参考代码: <https://zhuanlan.zhihu.com/p/371142471>

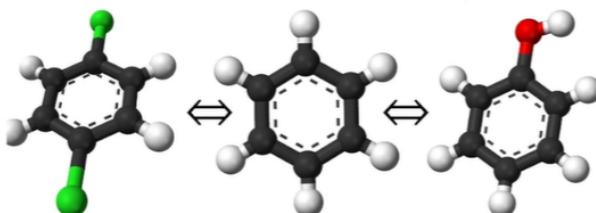
824

自由能微扰计算的理论与实践: SPONGE的应用

自由能微扰 (FEP) 计算

nn, dock初筛

- 此处FEP为广义的自由能微扰, 也即使用炼金术化学 (Alchemy) 的自由能计算方法



- 自由能变化 ΔG 是热力学中重要的概念, 可以用来描述变化的方向性和状态的稳定性, 因此在化学和生物学研究中具有重要意义
- FEP计算在化学和生物学研究中有广泛的应用, 可以用于预测药物分子的亲和性、疾病相关蛋白结合亲和性、化学反应的热力学性质等。通过FEP计算, 可以更好地理解和设计分子结构和相互作用, 为药物设计和化学反应优化提供有价值的信息。
- 相较于分子对接 (docking) 方法: 精确, 耗时

2

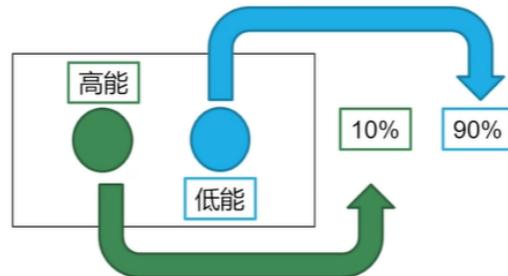
分子动力学

- 在NVT或NPT系综下，进行分子动力学模拟，模拟时间足够长时，获得任意一个构象的概率遵从玻尔兹曼分布

$$p(\mathbf{r}) = \frac{1}{Q} \exp\left(-\frac{U(\mathbf{r})}{k_B T}\right) \quad \text{NVT}$$

$$p(\mathbf{r}) = \frac{1}{Q} \exp\left(-\frac{U(\mathbf{r}) + pV}{k_B T}\right) \quad \text{NPT}$$

- 也即，在热力学研究中，分子动力学是一种基于能量的重要性抽样



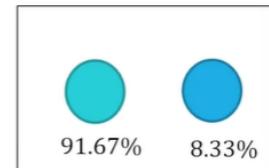
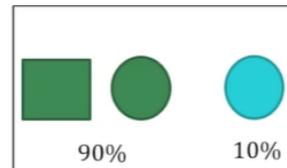
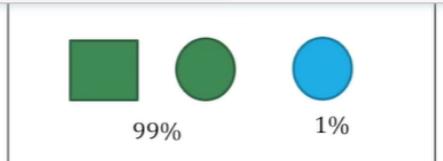
熵

增强采样

增强抽样

- ✓ 低概率事件难以抽到
- 抽10次，绿色出现10次，蓝色出现0次
- 以频率（抽样结果）估计概率： $\Delta G_{\text{绿蓝}} = -k_B T \ln 10/0 = -\infty$, 错误
- 解决方法

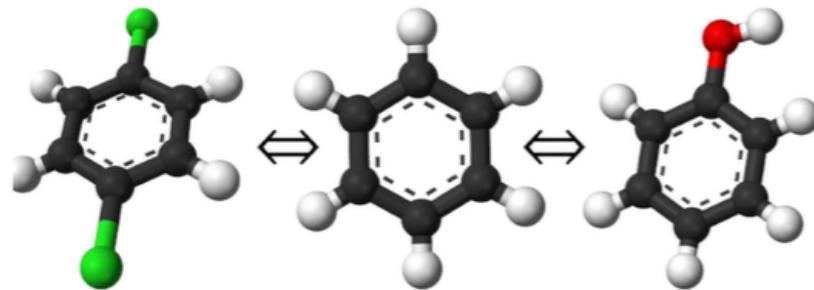
1. 抽100次，抽1000次，抽10000次
2. 构建1个中间态，构建2个中间态，构建3个中间态



3. 抽取的时候把绿色概率减小10倍，在最后计算的时候乘上10倍；
- 抽取的时候把绿色概率减小100倍，在最后计算的时候乘上100倍；
- 抽取的时候把绿色概率减小1000倍，在最后计算的时候乘上1000倍；

炼金术自由能计算

- 中世纪炼金术：炼铅制金
- 炼金术自由能计算：氢原子变氯原子；一个氢原子变羟基两个原子

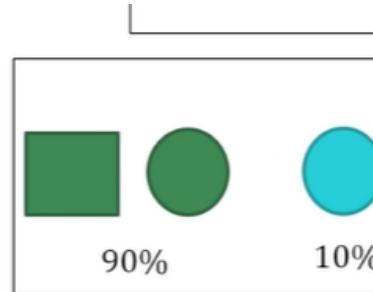


- 计算AB两个分子的自由能的方法：

$$\Delta G_{AB} = -k_B T \ln \int_A \exp\left(-\frac{U_A(\mathbf{r})}{k_B T}\right) d\mathbf{r} / \int_B \exp\left(-\frac{U_B(\mathbf{r})}{k_B T}\right) d\mathbf{r}$$

不断加中间态，线性插值

- 将分子变化变为连续的若干个中间体
 - 中间体由n个λ {λ_i}调控势能函数生成
- $$U_i(\mathbf{r}) = (1 - \lambda_i)U_0(\mathbf{r}) + \lambda_i U_n(\mathbf{r})$$
- λ₀ = 0, 此时U = U₀(r), 为A分子的势能函数
 - λ_n = 1, 此时U = U_n(r), 为B分子的势能函数
 - 当0 < λ_i < 1时, 分子介于AB之间, 非物理状态



$$\Delta G_{\lambda_{i+1}\lambda_i} = -k_B T \ln \int_{\lambda_{i+1}} \exp\left(-\frac{U_{i+1}(\mathbf{r})}{k_B T}\right) d\mathbf{r} / \int_{\lambda_i} \exp\left(-\frac{U_i(\mathbf{r})}{k_B T}\right) d\mathbf{r}$$

$$\Delta G_{AB} = \Delta G_{\lambda_1\lambda_0} + \Delta G_{\lambda_2\lambda_1} + \cdots + \Delta G_{\lambda_n\lambda_{n-1}}$$

FEP计算

FEP计算

热力学积分TI:

$$\int_{\lambda_{i+1}} \exp\left(-\frac{U_{i+1}(\mathbf{r})}{k_B T}\right) d\mathbf{r} / \int_{\lambda_i} \exp\left(-\frac{U_i(\mathbf{r})}{k_B T}\right) d\mathbf{r} = -\frac{1}{k_B T} \left(\frac{\partial U(\mathbf{r}, \lambda)}{\partial \lambda} \right),$$

$$\int_{\lambda_{i+1}} \exp\left(-\frac{U_{i+1}(\mathbf{r})}{k_B T}\right) d\mathbf{r} / \int_{\lambda_i} \exp\left(-\frac{U_i(\mathbf{r})}{k_B T}\right) d\mathbf{r}$$

- 如何用分子动力学模拟的抽样结果估计该概率之比?

✓ λ_{i+1} 与 λ_i 积分空间近似相同

$$\int_{\lambda_{i+1}} \exp\left(-\frac{U_{i+1}(\mathbf{r})}{k_B T}\right) d\mathbf{r} / \int_{\lambda_i} \exp\left(-\frac{U_i(\mathbf{r})}{k_B T}\right) d\mathbf{r} = \frac{\int \exp\left(-\frac{U_{i+1}(\mathbf{r}) - U_i(\mathbf{r})}{k_B T}\right) \exp\left(-\frac{U_i(\mathbf{r})}{k_B T}\right) d\mathbf{r}}{\int \exp\left(-\frac{U_i(\mathbf{r})}{k_B T}\right) d\mathbf{r}}$$

$$= \int \exp\left(-\frac{U_{i+1}(\mathbf{r}) - U_i(\mathbf{r})}{k_B T}\right) p_i(\mathbf{r}) d\mathbf{r}$$

$$= \left\langle \exp\left(-\frac{U_{i+1}(\mathbf{r}) - U_i(\mathbf{r})}{k_B T}\right) \right\rangle_i$$

$$\langle A \rangle_i = \int_i A(\mathbf{r}) p(\mathbf{r}) d\mathbf{r} = \int_i A(\mathbf{r}) \frac{1}{Q} \exp\left(-\frac{U(\mathbf{r})}{k_B T}\right) d\mathbf{r}$$

- Zwanzig方程, 狹义FEP, 指数平均

模拟退火算法[退火算法\(Annealing\)简介与详解](#)

12

温度积分增强抽样 (ITS)

- 改变温度

温度提高, 能量高的状态更容易抽到, 但能量低数量少的状态更不易抽到
温度降低, 能量低但是数量少的状态更容易抽到, 但能量低数量少的状态更不易抽到

- 温度积分增强抽样

通过数学方法构建有效势能:

$$U' = -k_B T \ln \sum_k n_k \exp\left(-\frac{U}{k_B T_k}\right)$$

此时的概率为

$$p'(\mathbf{r}) = \frac{1}{Q} \exp\left(-\frac{U'(\mathbf{r})}{k_B T}\right) = \frac{1}{Q} \sum_k n_k \exp\left(-\frac{U}{k_B T_k}\right)$$

无限交换频率的温度交换方法, 没有额外的副本交换计算

SPONGE: 独立的MD程序
Xponge: 独立的前后处理

1. 需要改动MD程序

2. 后处理需要额外处理

抽取的时候把绿色概率减小10倍, 在最后计算的时候乘上10倍;

实践部分

最终结构处理

- 处理结构中缺失的loop区、氢原子等
- 蛋白质封端。
- 溶剂化
- 添加离子, 使得体系电中性
- 保存处理好的结合态和游离态的结构至PDB

```
Xponge.source("mindsponge.toolkits.forcefield.amber.ff14sb")
Xponge.source("mindsponge.toolkits.forcefield.amber.tip3p")
protein = load_pdb("protein.pdb")
protein.Add_Missing_Atoms()
ligand = load_pdb("ligand_renamed.pdb")
ligand.Add_Missing_Atoms()
protein_ligand = ligand | protein

add_solvent_box(protein_ligand, WAT, 10)
c1 = int(round(protein_ligand.charge))
Solvent_Replace(protein_ligand, WAT, {CL:20 + c1, K:20})
Save_PDB(protein_ligand, "protein_ligand_water.pdb")

add_solvent_box(ligand, WAT, 10)
c2 = int(round(ligand.charge))
Solvent_Replace(ligand, WAT, {CL:10 + c2, K:10})
Save_PDB(ligand, "ligand_water.pdb")
```

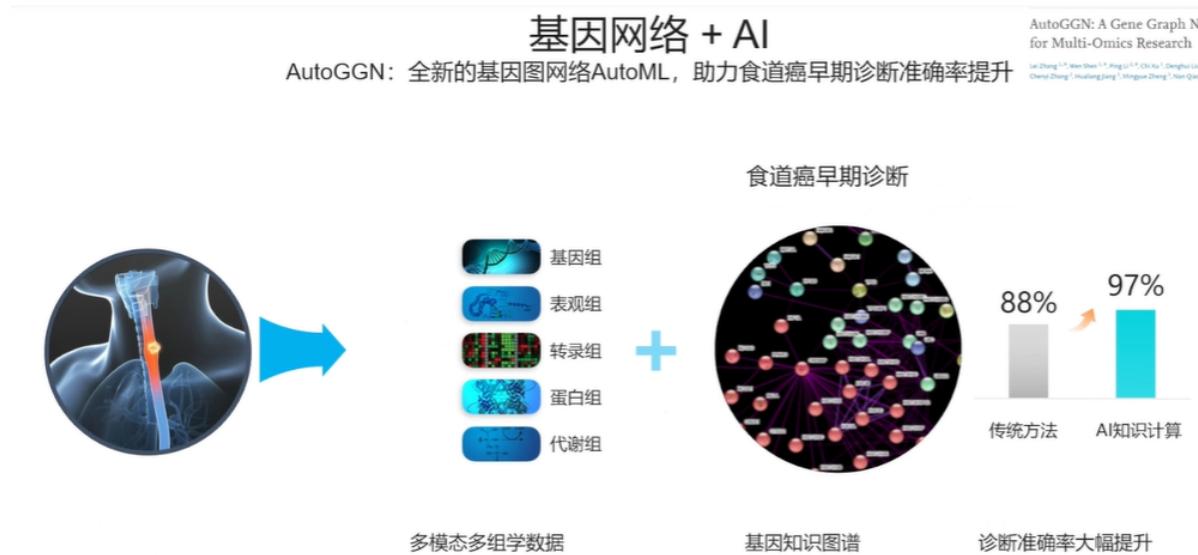
华为云AI在生物医药领域的探索和应用

缩短70%时间, 提升10%sr

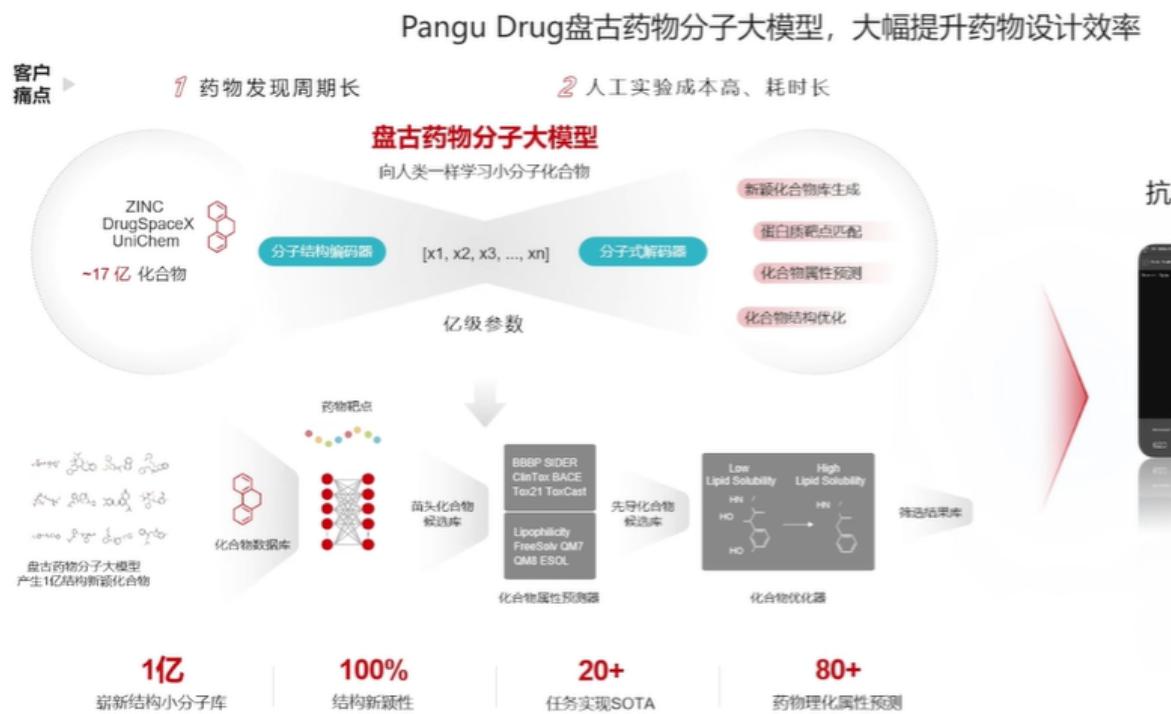
- 靶点发现/个性化药物
- 靶点模拟，精准对接

残差全连接网络 (RFCN)

resnet, densenet



pangu先导药研发周期压缩到1个月

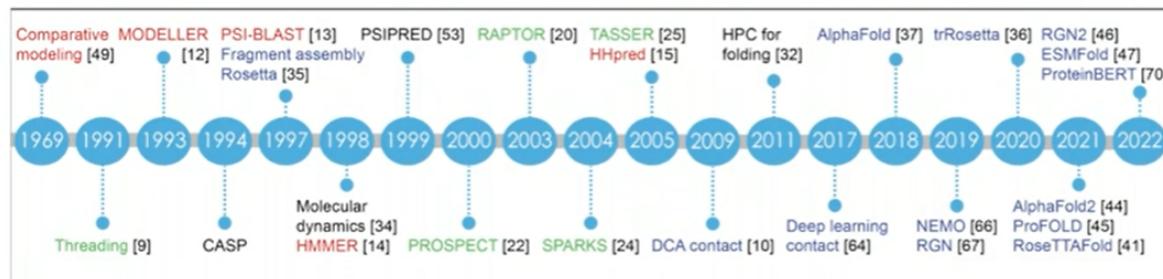


蛋白质结构预测：从生物学观察到算法

蛋白质及蛋白质折叠

anfinsen实验：去折叠的蛋白质在体外可自发再折叠，结构信息蕴含在序列之中

蛋白质结构预测方法发展史管窥



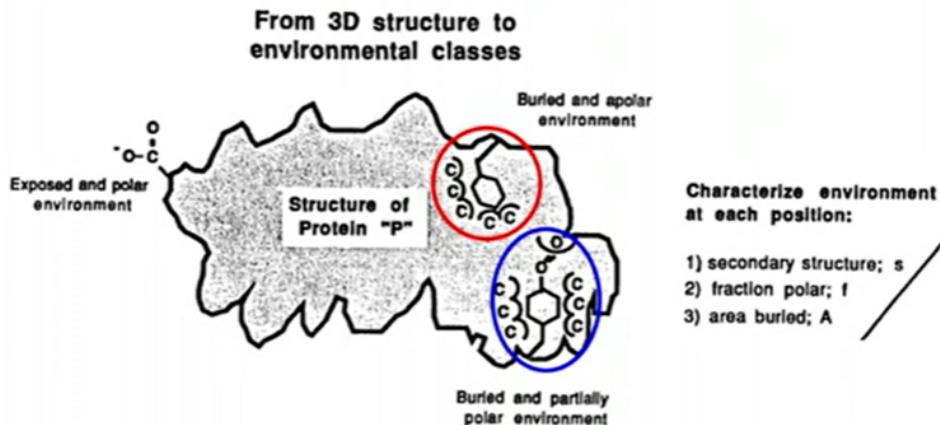
■ 核心：

- 对进化的认识与刻画（与NLP显著不同）
- 对折叠规律的认识与刻画

比较建模法

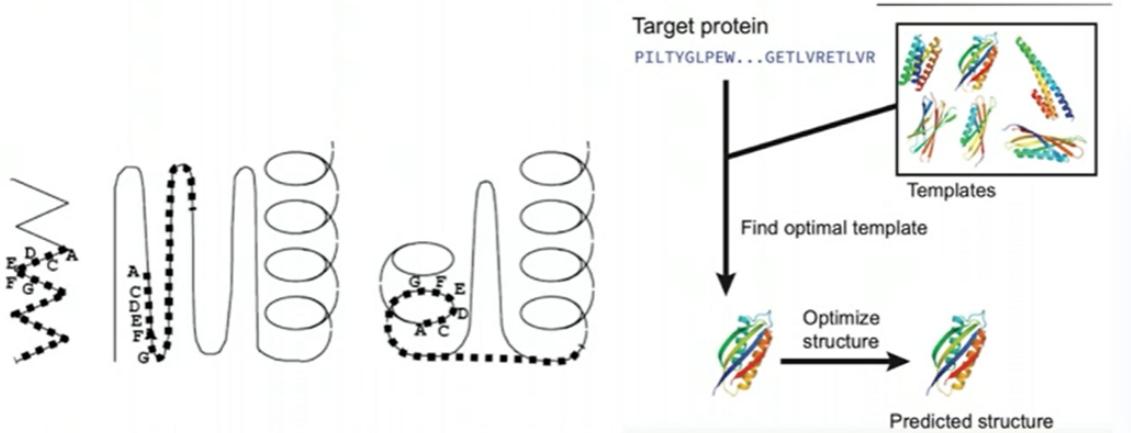
- 基本思想：找相似蛋白质（近同源），基于相似蛋白质的结构预测结构
- 核心操作：序列-序列比较
- 序列如何表示？单个序列、PSSM、HMM、CRF、大语言学模型

规范法



■ 观察与假设：

- 在天然态构象中，每个残基都fit其局部微环境



- 基本思想：把目标蛋白穿入已知结构中，残基fit其局部微环境者为最佳
- 核心操作：序列-结构比较

深度学习蛋白结构预测方法介绍

PSP protocols

accessing evolutionary information-Transformers

深度学习和实验信息在蛋白结构预测问题中的应用

深度学习和实验信息辅助的蛋白质结构预测上机演示

825

蛋白质设计：从能量函数优化到人工智能生成

怎样理解蛋白质结构和动力学？

(从物理概念上)

能量依赖于结构

量子力学处理电子 →

静态分子体系的能量(势能)依赖于其结构

两个原子



N个原子

$V(X_1, X_2, \dots, X_N)$

计算V的方法

第一性原理(非经验方法)

--量子力学

经验方法

--分子力学力场

--数据驱动(统计能量函数)

能量模型

$$V(x_1, x_2) = V(r)$$

统计热力学

所有微观结构都有可能(有非零概率存在)

概率分布遵从热力学定律

玻尔兹曼分布:

- 能量越低的微观结构，概率越大
- 温度越低，分布越往低能量的结构集中

概率分布依赖于能量

$$Q = \sum_j \exp(-E_j / k_B T),$$

$$p_i = \frac{\exp(-E_i / k_B T)}{Q}$$

玻尔兹曼因子

能量依赖于结构

多个微观结构的集合

配分函数：玻尔兹曼因子的加和

配分函数

$$Q = \sum_j \exp(-E_j / k_B T),$$

$$p_i = \frac{\exp(-E_i / k_B T)}{Q}$$

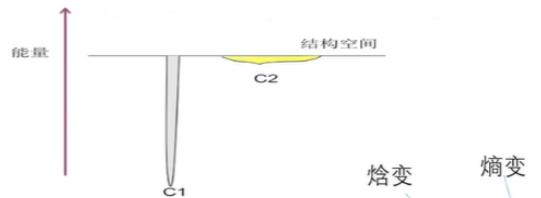
自由能

$$G = -k_B T \ln Q$$

平衡常数

$$\frac{P_{C1}}{P_{C2}} = \frac{Q_{C1}}{Q_{C2}} = \frac{e^{-\frac{G_{C1}}{k_B T}}}{e^{-\frac{G_{C2}}{k_B T}}} = e^{-\frac{\Delta G}{k_B T}}$$

$$\Delta G = G_{C1} - G_{C2} = \Delta H - T \Delta S$$

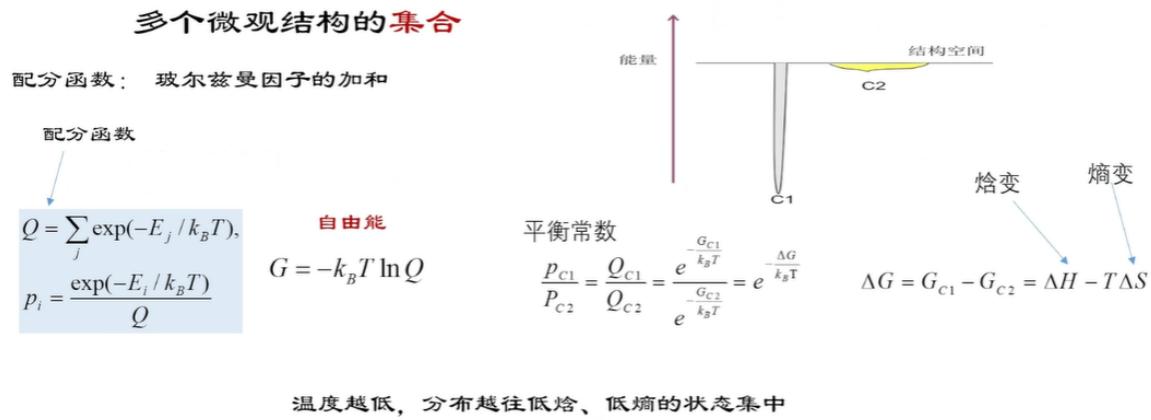


焓变

熵变

温度越低，分布越往低焓、低熵的状态集中

自由能、平衡常数、焓与熵

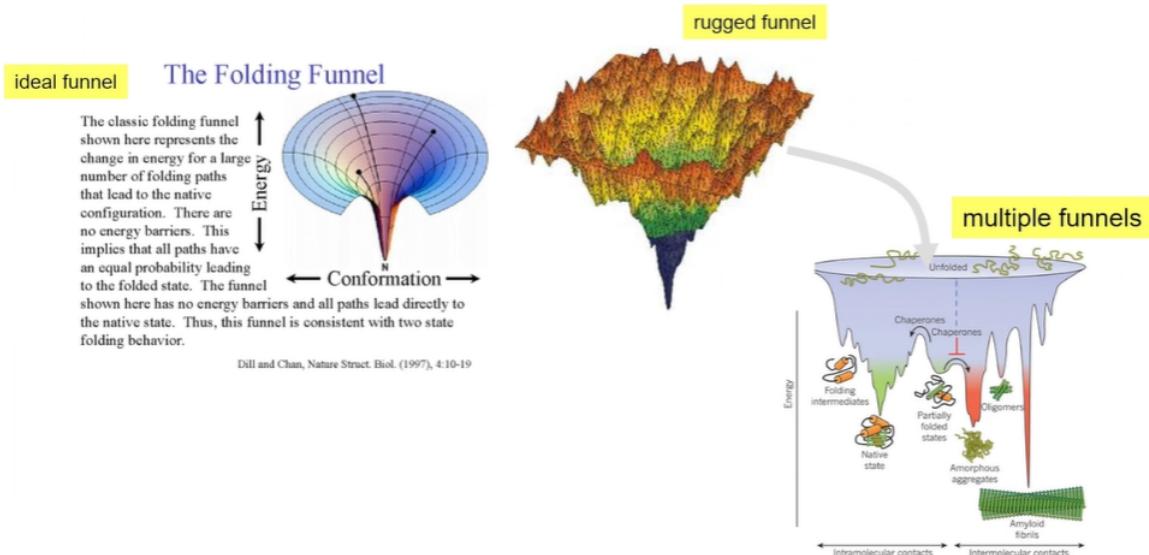


Levinthal's Paradox

关于蛋白质折叠速率：

- 蛋白质可能的构象状态是天文数字
- 如果蛋白质要通过构象的随机改变在这些构象中找到正确天然构象，需要的时间超过宇宙年龄
- 实际蛋白质折叠速率可快至毫秒

漏斗模型



决定蛋白质自由能地貌的相互作用

溶剂效应

溶剂效应 (solvent effects) :

疏水相互作用：碳骨架是疏水的。在水溶液中，对溶剂暴露的分子表面（溶剂可接近表面）的疏水性越强，相应构象态的自由能越高，稳定性越差。

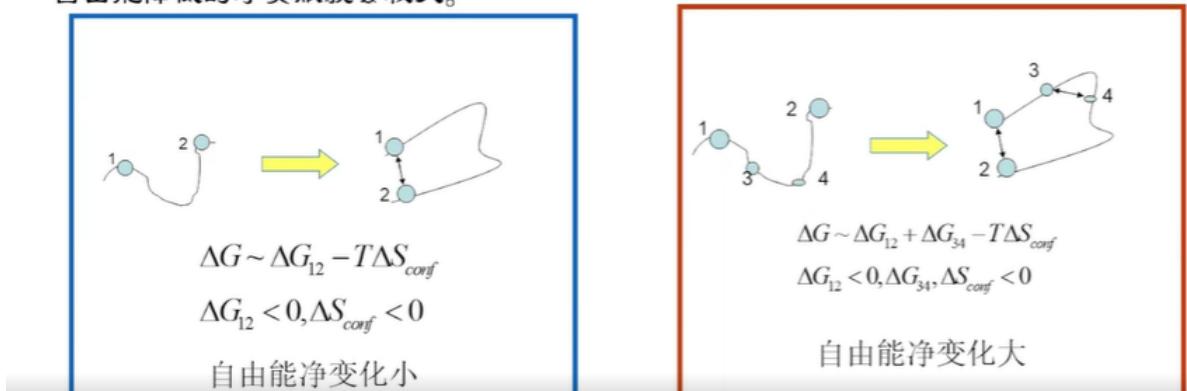
极性相互作用：包括极性官能团间的氢键、盐键。由于水分子是良好的氢键供体/受体，介电常数很高。一般而言，溶质极性官能团暴露在水溶液中的构象态自由能较低。除非溶质极性官能团之间能够互补地形成氢键或盐键，极性官能团包埋于溶质内部的构象态自由能较高，是不稳定的。



协同效应

自由能净变化 (大/小)

在生物大分子中，疏水的碳骨架和亲水的极性官能团通过共价键连接在一起。共价结构限制了这些基团在空间中可能的相对排列方式（可能的构象态）。如果某两个基团之间要形成有利的相互作用（如氢键），会进一步限制分子构象，从而带来构象熵的损失，二者对自由能贡献符号相反，带来的自由能总体变化可能并不大。然而，如果在受到限制的分子构象下，其他基团之间同时也能形成有利的相互作用，后者对自由能降低的净贡献就会较大。



氨基酸序列决定自由能地貌

- 序列可折叠性：

存在热力学稳定的稳定折叠状态

- 结构可设计性

存在序列能稳定折叠成该结构

必要条件：

折叠态相对于去折叠态有足够的自由能能隙

蛋白质结构计算建模

基本范式

建立可计算的模型
刻画蛋白质

能量函数 and/or 分布函数

例如：基于物理的模型

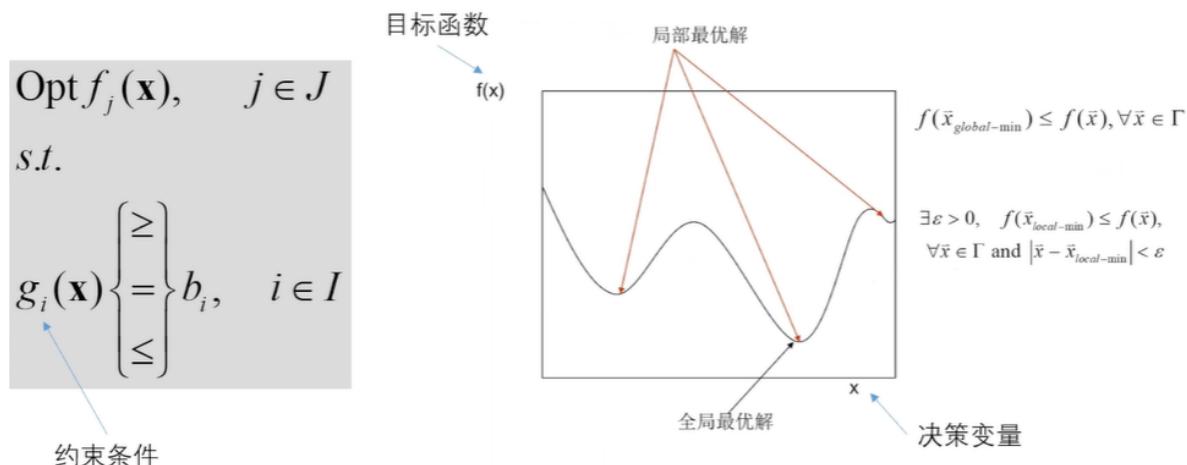
分子力场能量模型

玻尔兹曼分布

优化算法、采样算法

优化

优化就是找函数的最大值或最小值



抽样

抽样就是产生符合目标概率分布的样本集合

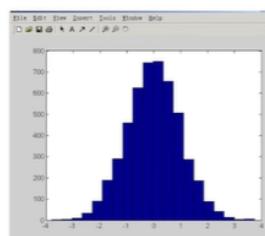
低维分布：简单

正态分布：

$$\rho_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- 正态分布
- `randn(n,m);` n X m matrix, each element a random variable of $N(0,1)$ distribution
- mu + sigma.*randn(n,m)

```
...
>> x=randn(5000,1);
>> hist(x);
...
```

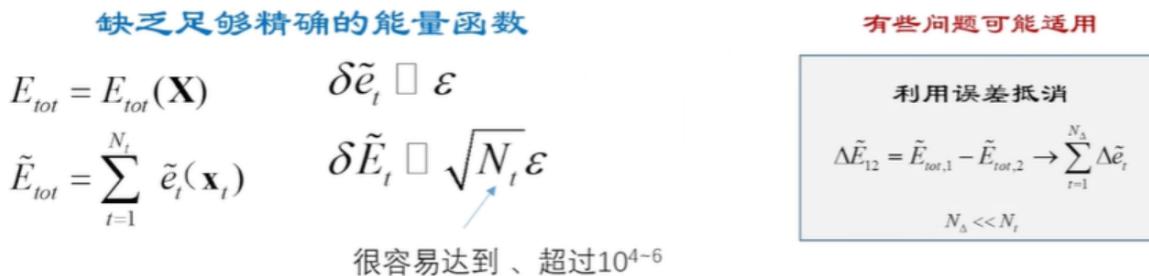


难点：高维分布

“解决”办法：
仿真（随机过程）

基本范式的困境

- 能量函数或分布函数困境
- 算法困境



分布函数：高维联合分布，直接建模困难

克服模型精度不足

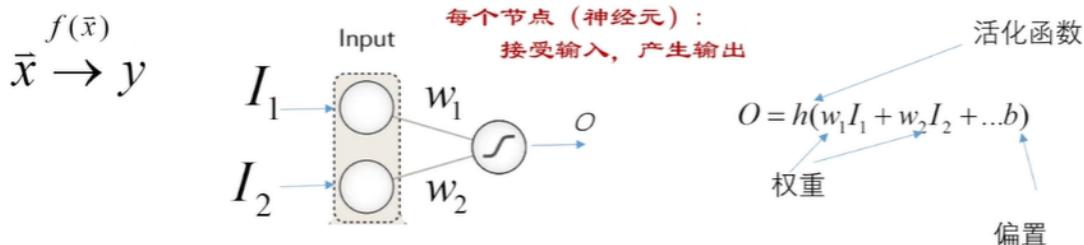
- 数据驱动
- AI

应用人工智能改进蛋白质结构计算

- 更好、更充分地利用数据改进模型
- 模型更少受到数学形式的限制

人工神经网络

用网络图的形式来表示的一种计算流程（或机制）
➢ 图由结点（node）和结点间的联接（connection）组成
➢ 上游结点的输出被传递给下游结点；
➢ 后者把多个上游结点的输入组合起来，经“活化函数”变换后，产生输出



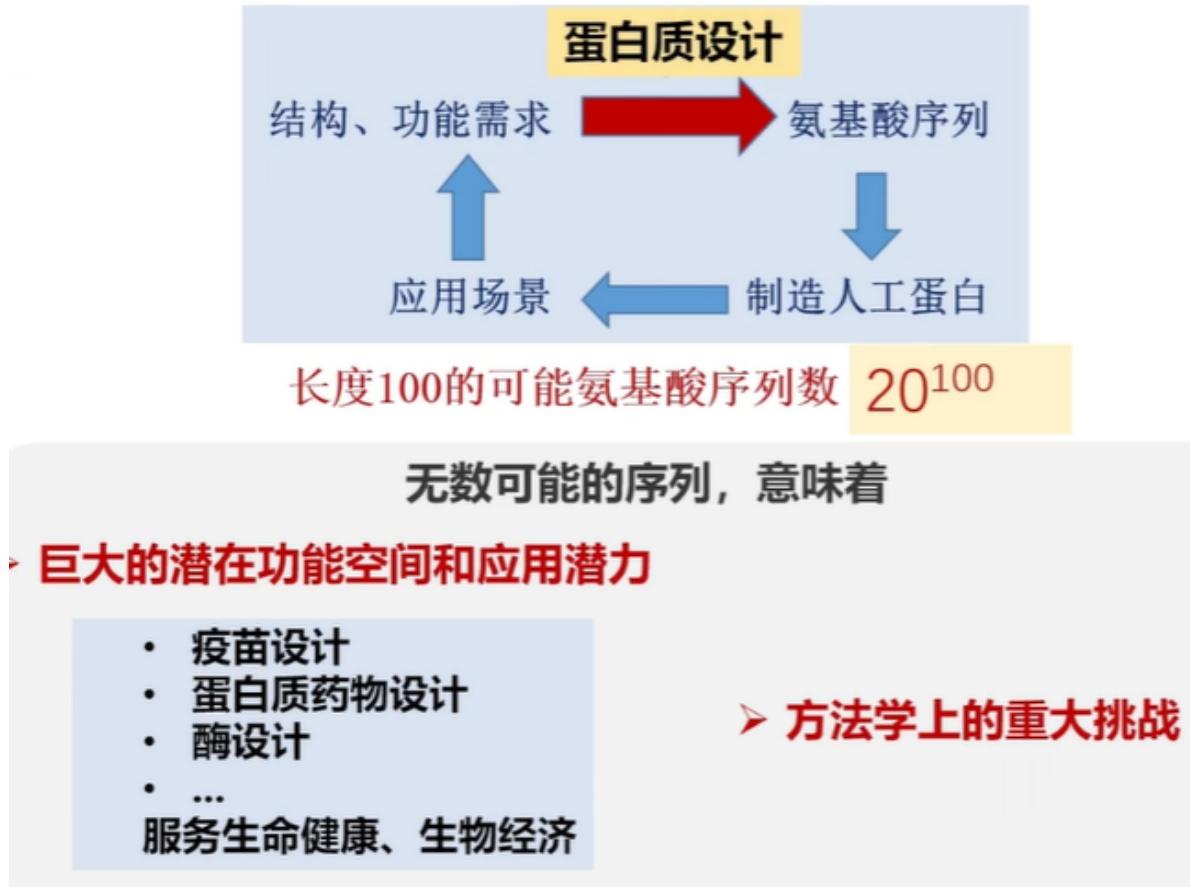
神经网络

- 神经网络的结构（有哪些结点、结点之间怎样联接、每个结点使用什么样的活化函数）一般是用户定义的；
- 待学习的参数：
 - 各个联接的权重、结点使用的偏置量

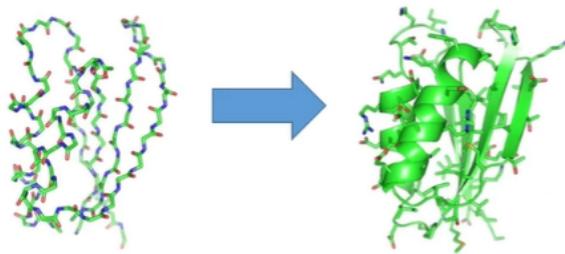
神经网络的优势：

- 简单、统一的局部结构，能够组合出任意复杂的非线性变换/映射
- 通用的训练算法

- 更好、更充分地利用数据改进模型
- 模型更少受到数学形式的限制



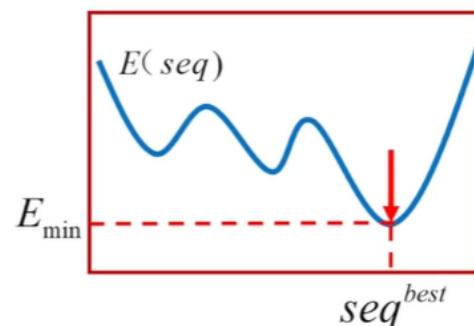
基于能量优化的蛋白质设计



$$E(\text{seq}) = \min_{\mathbf{r}} E(\mathbf{r}; \text{seq})$$

$$\begin{aligned} \mathbf{r} &\equiv (\mathbf{r}_{\text{backbone}}, \mathbf{r}_{\text{sidechain}}) \\ \mathbf{r}_{\text{backbone}} &\approx \mathbf{r}_{\text{backbone}}^0 \end{aligned}$$

$$E(\text{seq}; \mathbf{r}_{\text{backbone}}^0) = \min_{\mathbf{r}} E(\mathbf{r}; \text{seq}, \mathbf{r}_{\text{backbone}}^0)$$



大分子能量模型类型

- 第一性原理模型
 - 全原子分子力场
 - 连续介质模型
 - 粗粒度模型
 - 杂合模型
 - 统计能量模型
- } 一般不包含熵的贡献
} 包含熵的贡献

分子力场能量项 + 溶剂效应项 + 统计能量项

经验加权，参数集 θ

$$E_\theta(\text{seq})$$

怎样选 θ ？

$$\min_{\theta} [E_\theta(\text{seq}_{\text{native}}) - E_\theta(\text{seq}_{\text{non-native}})]$$

统计能量函数

统计热力学： Boltzmann 分布

微观状态

$$\rho_0(\mathbf{p}, \mathbf{q}) = \frac{e^{-\beta H(\mathbf{p}, \mathbf{q})}}{Z}$$

概率密度

$$Z = \int_{\Omega} e^{-\beta H(\mathbf{p}, \mathbf{q})} d\mathbf{p} d\mathbf{q}$$

$$H(\mathbf{p}, \mathbf{q}) = -\frac{1}{\beta} \ln \rho_0(\mathbf{p}, \mathbf{q}) + C$$

Boltzmann 分布
是在给定内能
(哈密顿量期望
值) 约束下的极
大熵分布

$$\begin{aligned}\rho_0 &= \arg \max_{\rho} - \int_{\Omega} \rho \ln \rho d\Omega \\ s.t. \quad & \langle H \rangle = \int_{\Omega} \rho H d\Omega = U \\ & \int_{\Omega} \rho d\Omega = 1\end{aligned}$$

困难： $\rho_0(\mathbf{p}, \mathbf{q})$ 维度很高，难以用常规方法估计

传统解决方法

把逆Boltzmann法应用于边缘分布

$$E_{total}(r_{1:N}) = \sum_m e_m(\Theta(r_{m_1:m_n}))$$

直接相互作用范围有限

联合分布 $\longrightarrow p(r_{1:N}) \propto \exp[-\beta E_{total}(r_{1:N})]$

依赖的自由度数目 = $3N-6$ $\approx \exp\left[-\beta \sum_m e_m(\Theta_m(r_{m_1:m_n}))\right]$

$\propto \prod_m p(\Theta_m)$

边缘分布 依赖的自由度数目远小于 $3N-6$

问题：可能忽略了非常重要的相关关系

改进方法

- 引入更高维的边缘分布

克服技术困难（维度灾难问题）

避免过拟合

SCUBA+ABACUS2 de novo

扩散生成采样好

检索增强的大语言模型

蛋白质属性建模以及突变预测

dl应用于蛋白质

Bert encoder

GPT decoder

注意力机制

蛋白质数据库

PDB, Uniprot

细胞调控图谱的计算解析

