



AlphaFold2: 原理、应用和未来

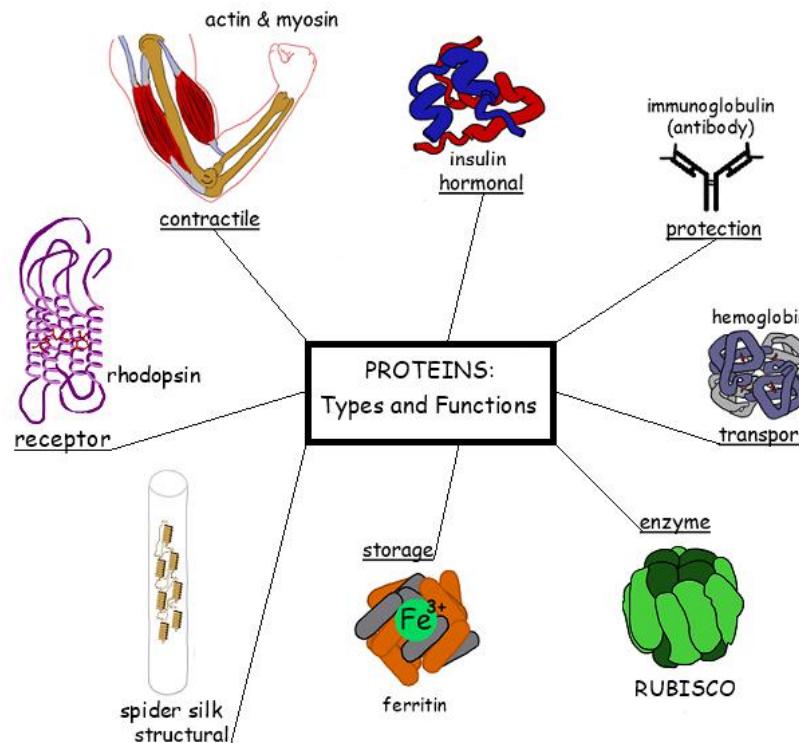
钟博子韬 上海交通大学

2021/09/15

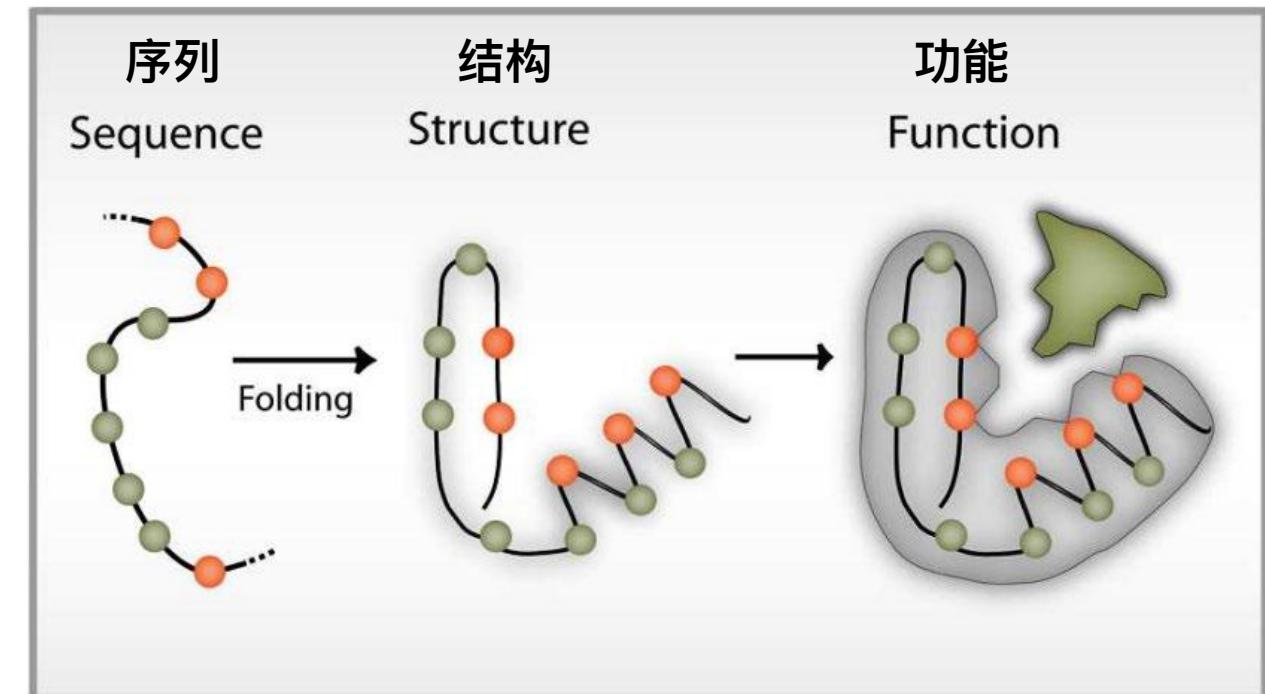
目录 Content

- I. 蛋白质结构预测
- II. AlphaFold模型架构
- III. 预测结果
- IV. AlphaFold复现与应用
- V. 高通量结构预测
- VI. 未来前景

蛋白质结构研究的重要性



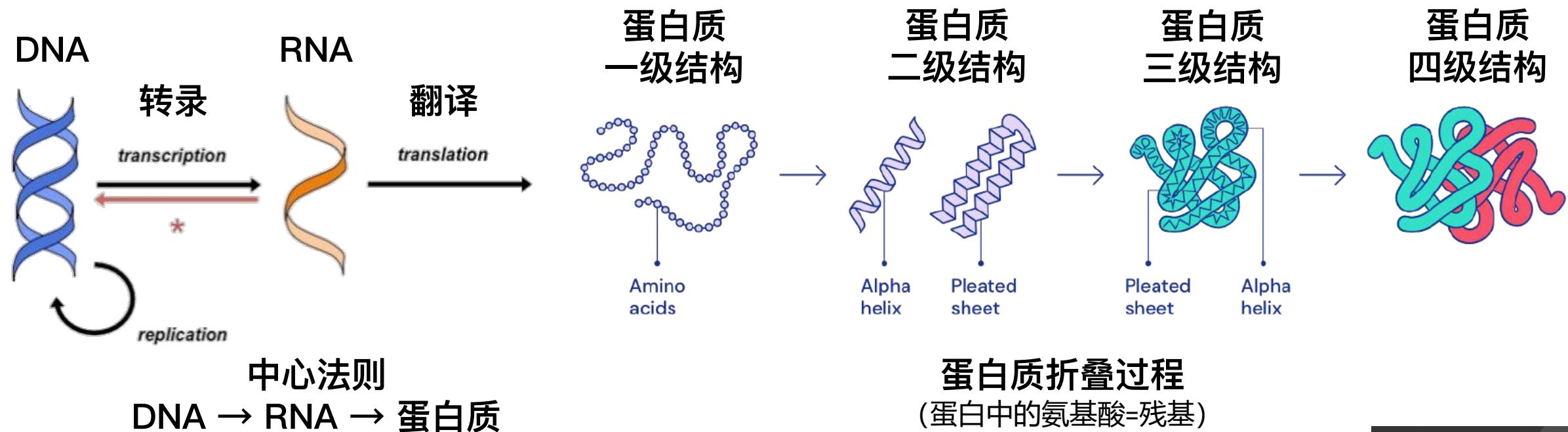
特定的三维结构产生对应的功能



蛋白是生命活动的主要承担者

序列决定结构，结构决定功能

蛋白质的折叠过程



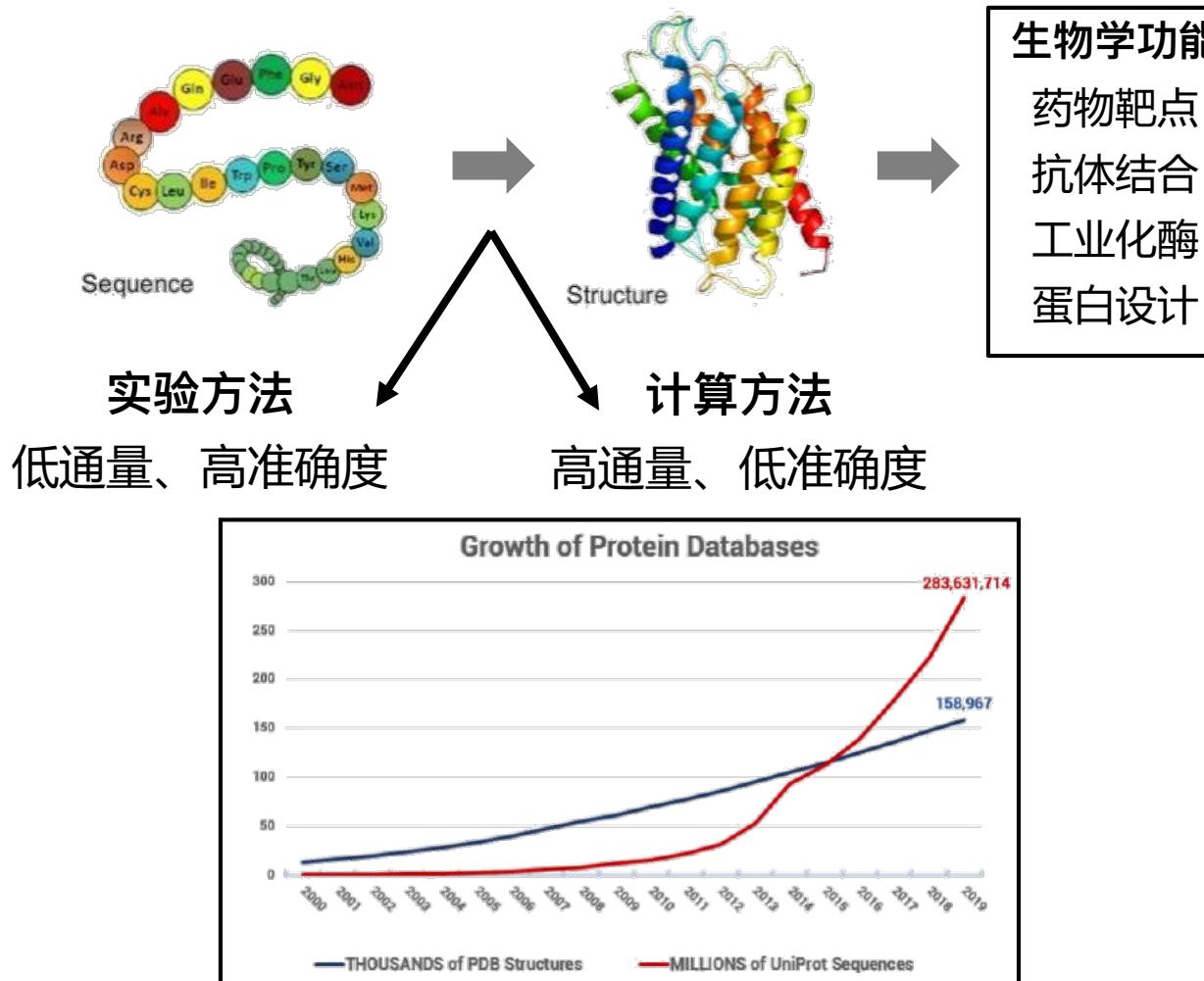
Anfinsen's Dogma

- 蛋白质折叠成原始结构所需的信息都已被编码在氨基酸序列中
- 蛋白质折叠到最小能量状态
- 大多数蛋白质会折叠成一个独特的构象

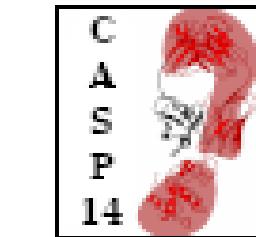
Christian B. Anfinsen
1972 Nobel Prize in Chemistry



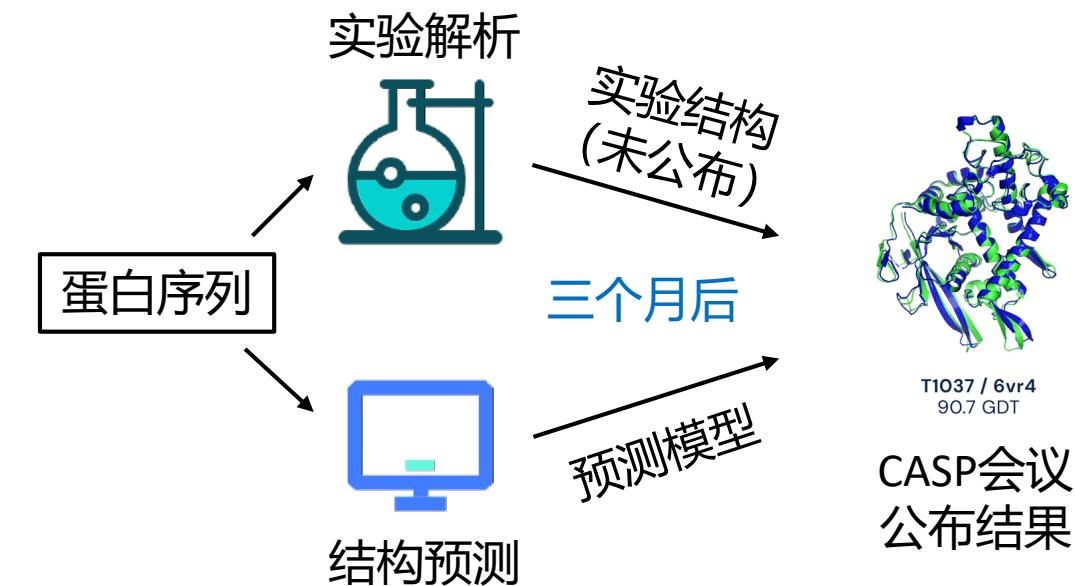
蛋白质结构预测：半个世纪的难题



海量的序列信息，少量的结构信息
需要高精度高通量发现蛋白质结构的方法

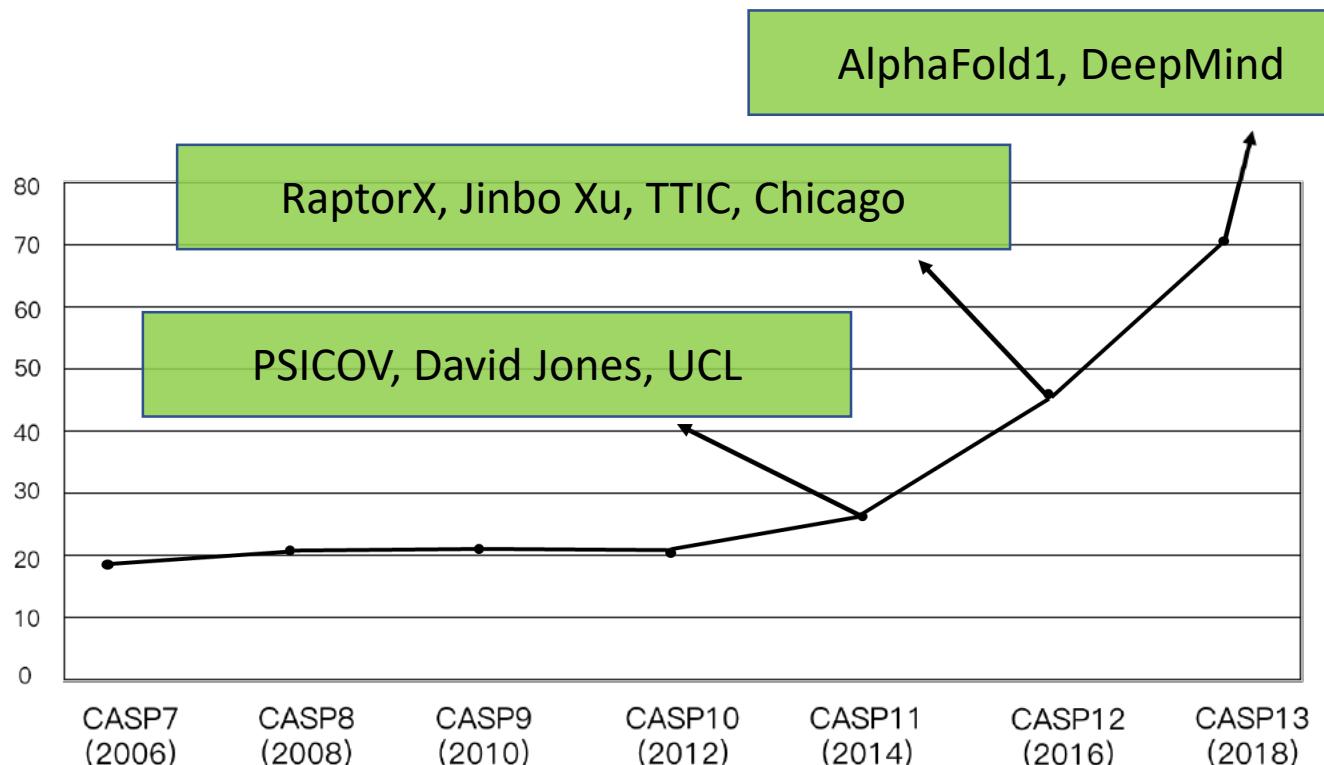


CASP
Critical Assessment of protein Structure Prediction, 1994

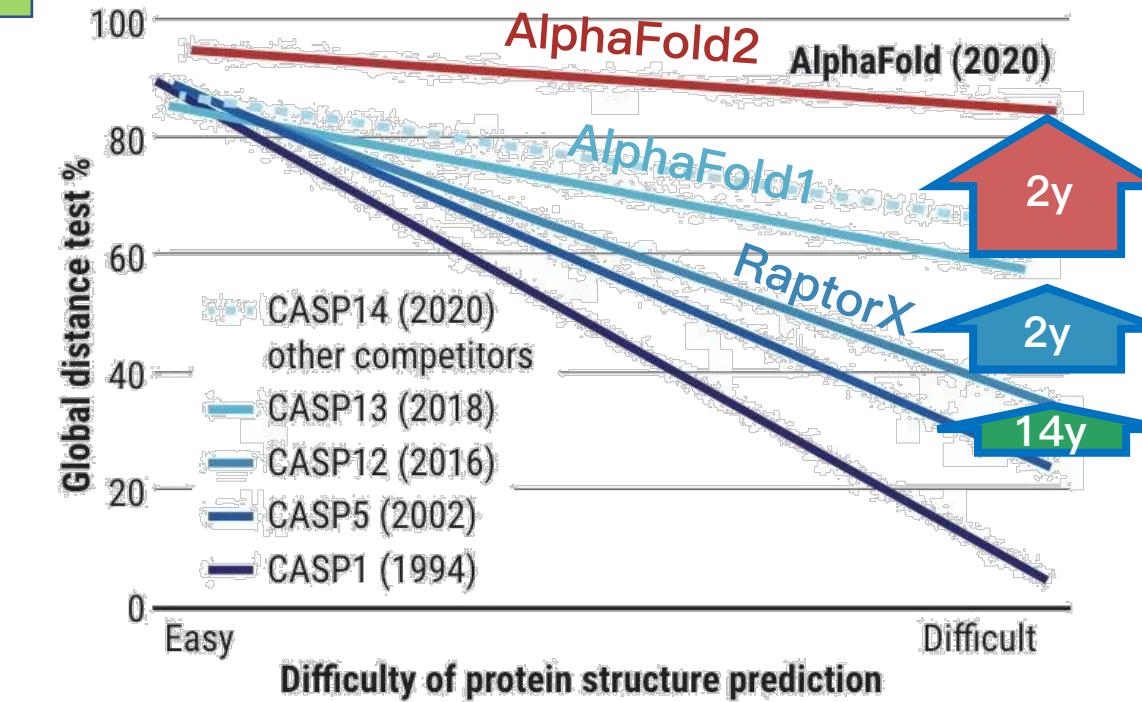


评估计算机预测蛋白质结构方法的准确度

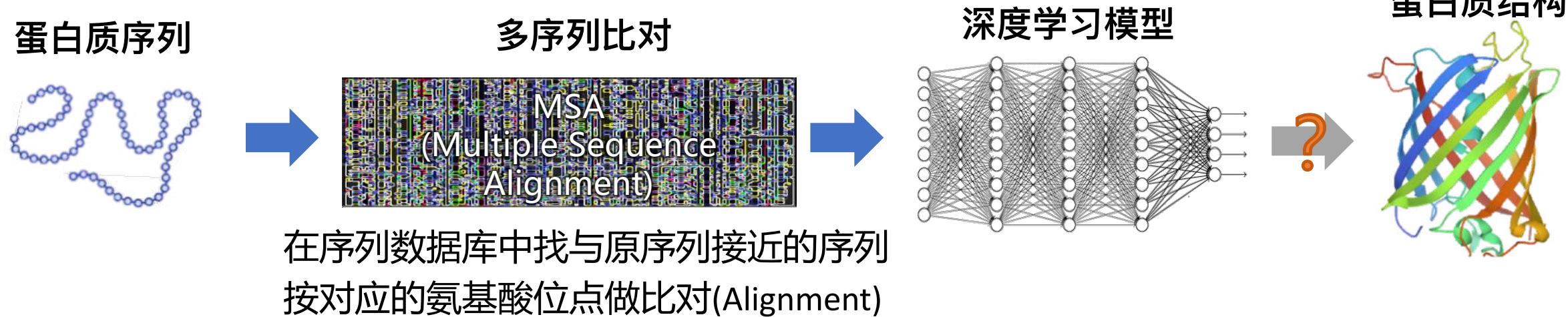
CASP中的关键提升



- 2014 (MSA): PSICOV, David Jones, UCL
- 2016 (Deep Learning/ResNet): RaptorX-Contact, Jinbo Xu, TTIC, Chicago
- 2020 (Transformer): AlphaFold2, DeepMind

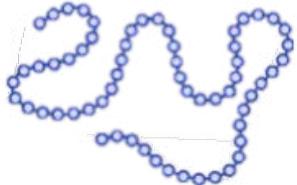


预测蛋白质的Contact Map: 间接预测蛋白质结构



预测蛋白质的Contact Map: 间接预测蛋白质结构

蛋白质序列



多序列比对

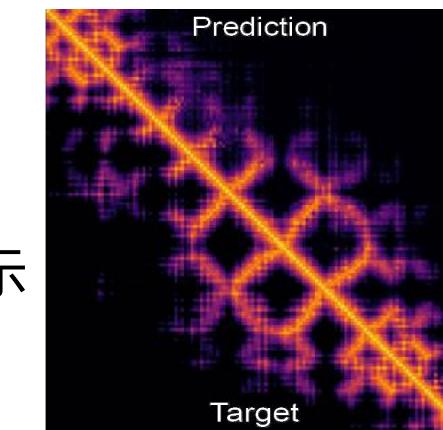


在序列数据库中找与原序列接近的序列
按对应的氨基酸位点做比对(Alignment)

为什么选用这种方法：

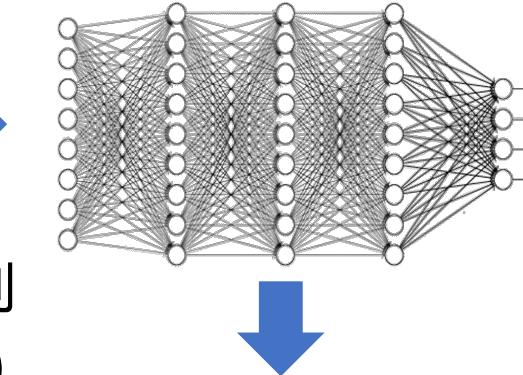
- 直接预测蛋白质结构3D坐标比较困难
- 先预测蛋白质的Contact map，然后作为限制来优化蛋白质折叠，相对来说更简单

蛋白质结构的二维表示
Contact Map
(Distance Map)

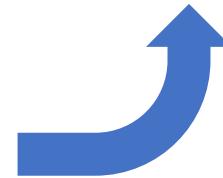
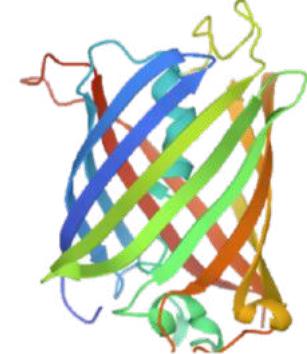


蛋白质中两两氨基酸的距离形成的map
Contact是指氨基酸之间距离小于某个阈值

深度学习模型

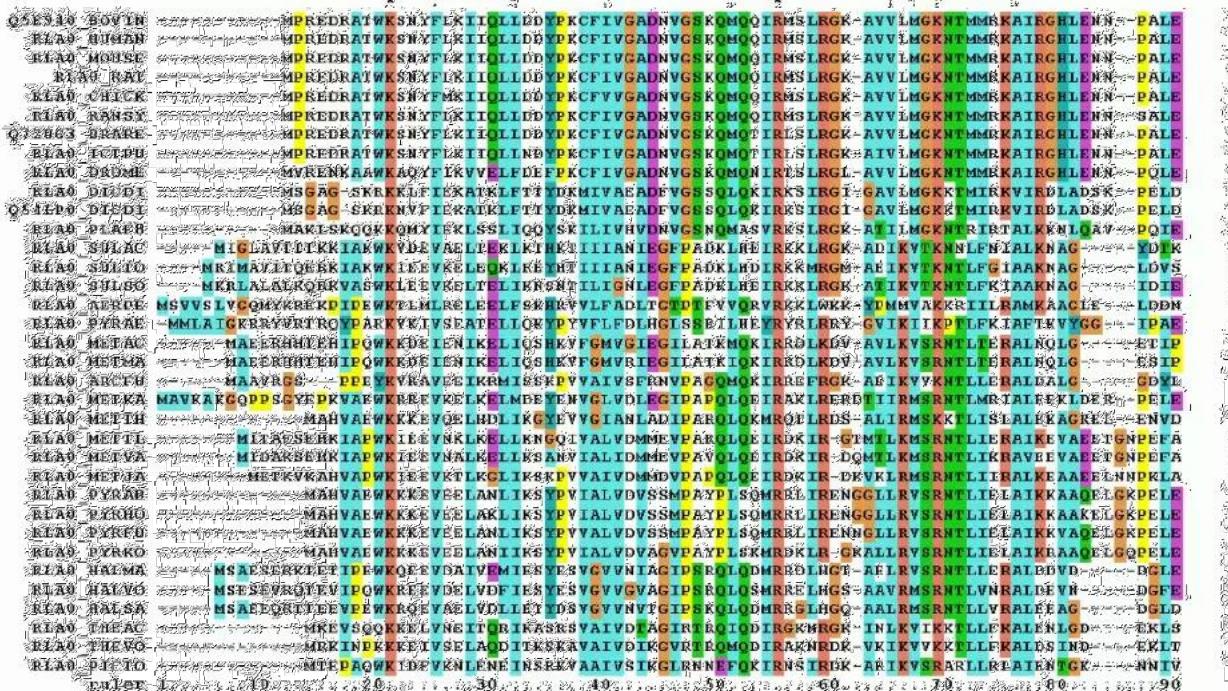


蛋白质结构



预测蛋白质的Contact Map: 理论基础

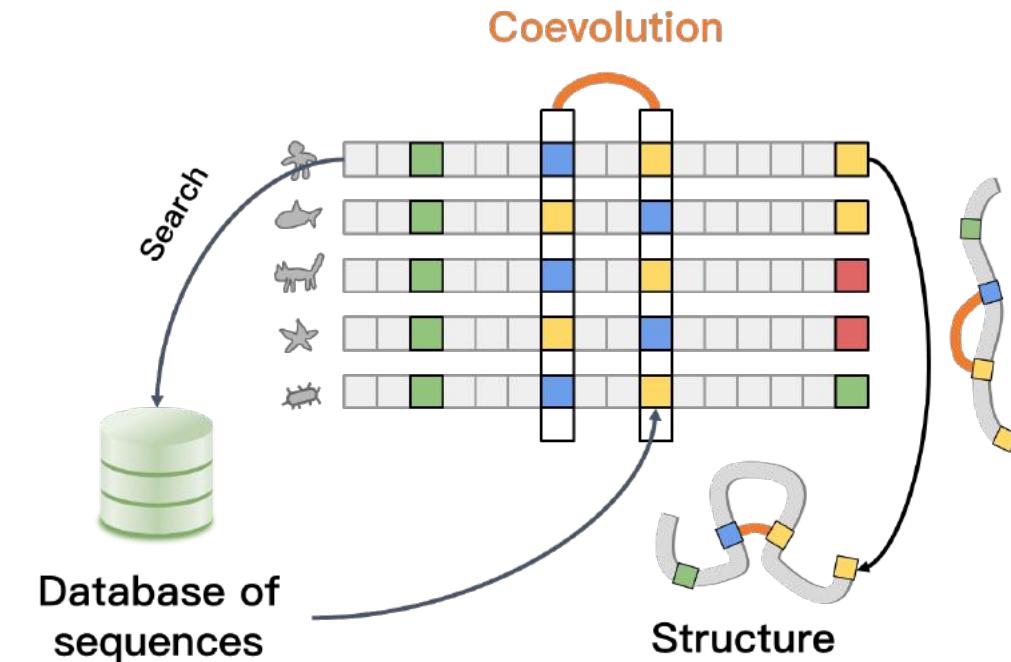
多序列比对 (MSA)



序列保守性信息：在不同物种中同一个位置的氨基酸不变

序列共进化信息：两个不同位置的氨基酸同步变化

共进化信息预测蛋白质Contact



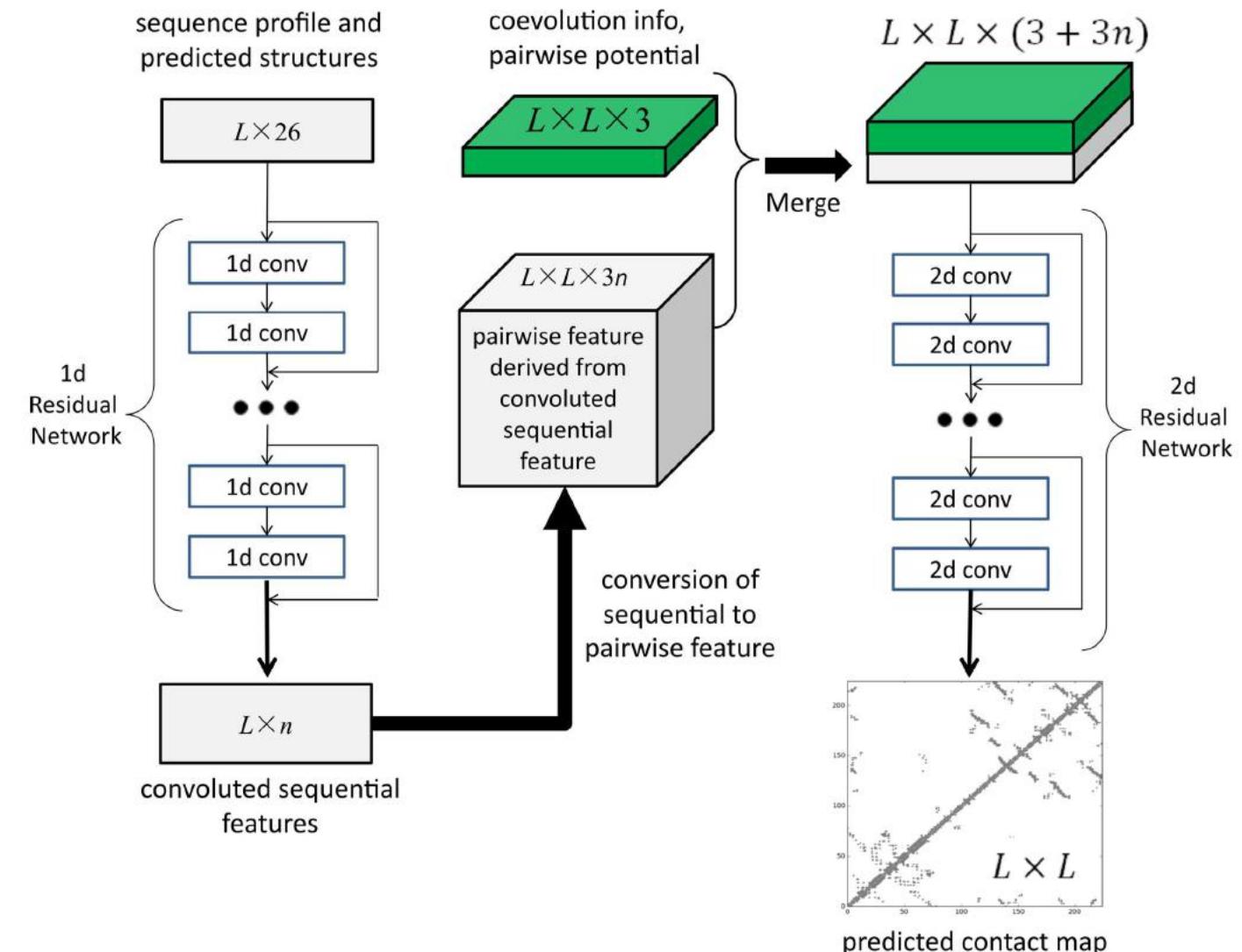
RaptorX 2016

主要特点：

- ResNet用于contact map预测

成功原因：

- Contact map接近于图像，架构成熟
- 应用ResNet能够达到足够的深度



AlphaFold 2018

主要特点：

- 用ResNet从MSA预测distance map
- 同时预测了蛋白质中的二面角

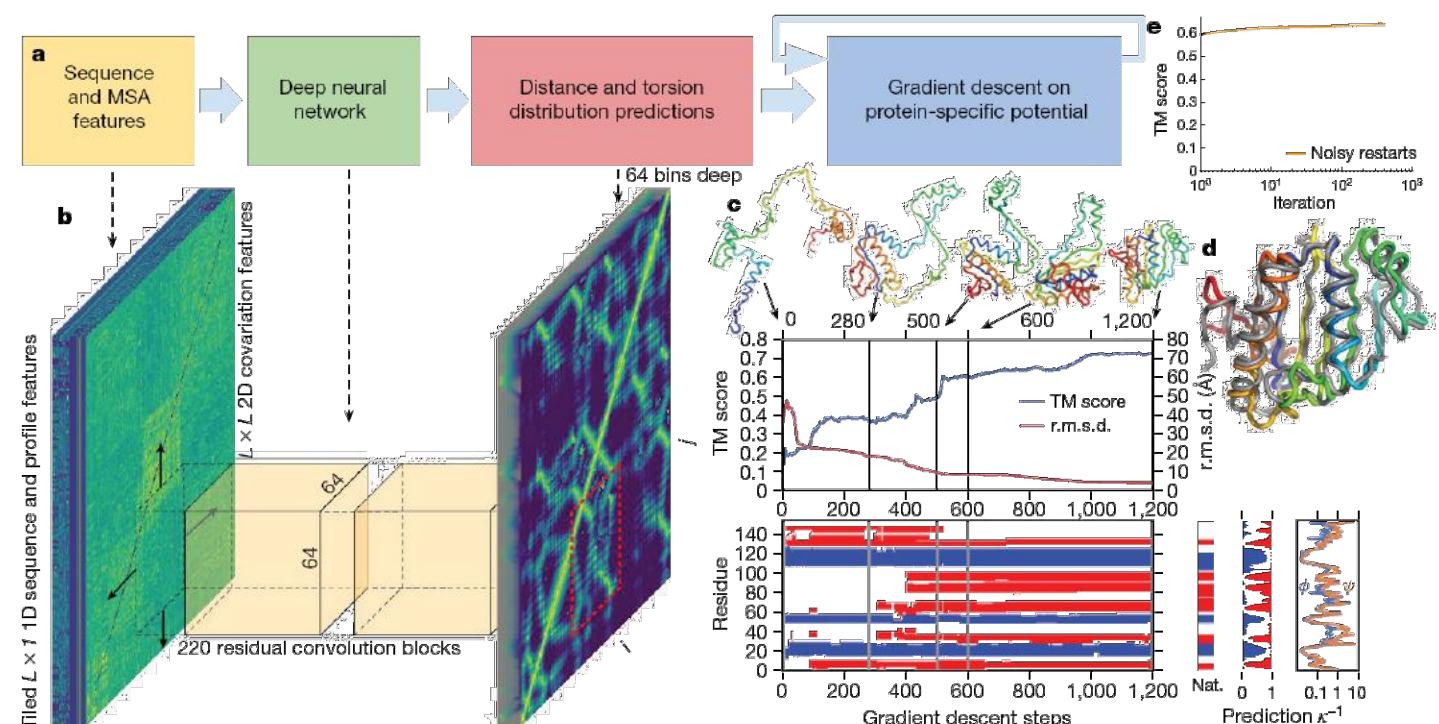
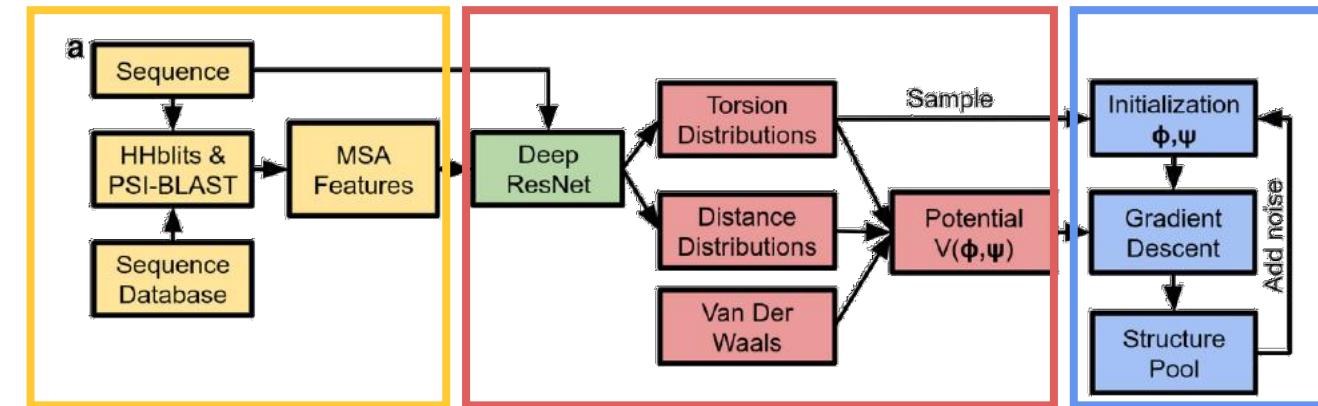
成功原因：

- 强有力的硬件优势
- 架构层面的优势对比其他预测 contact map 的方法并不显著

预测距离分布和
二面角分布

结构优化

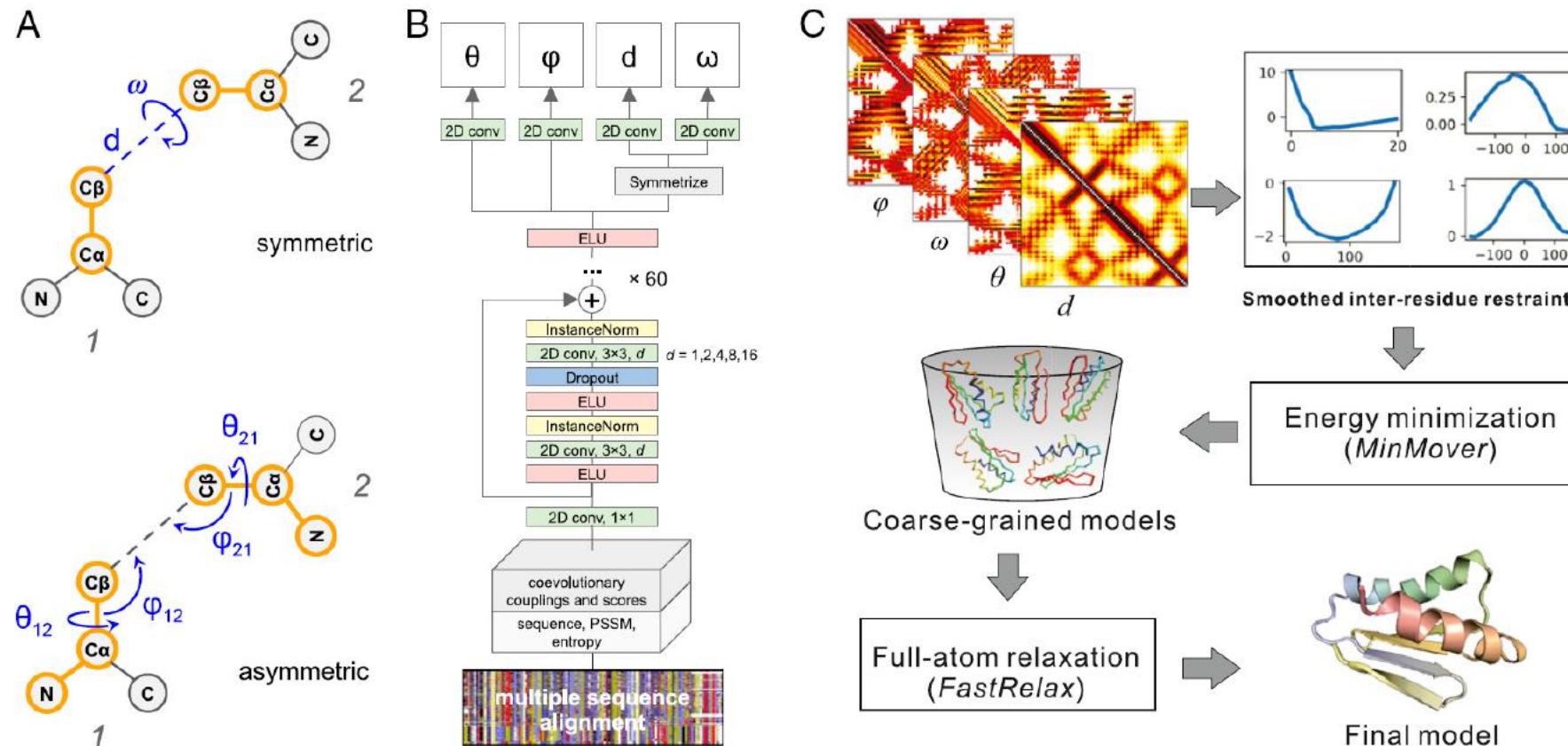
多序列比对



trRosetta (2018, 2020)

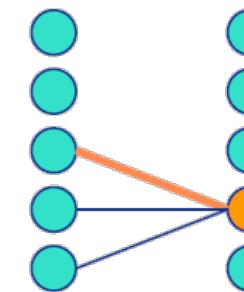
预测输出多个map(包括contact, torsion等), 获取更多的结构信息用于后续优化

预测中使用 $C\beta$ 作为每个氨基酸的中心, 相比于 $C\alpha$ 有更多的侧链信息



预测模型架构：如何抉择？

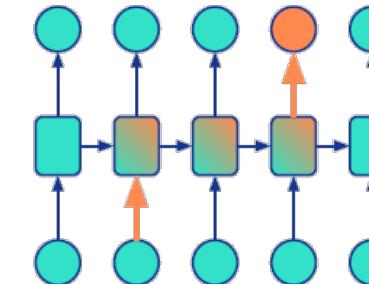
- CASP13 in 2018:
- 主流方法均选用预测contact map再后续优化的方法
 - AlphaFold (contact map, ResNet)
 - I-TASSER (contact map, Threading, ResNet)
- CASP14 in 2020:
- 大部分主流方法仍在预测contact map
 - BAKER (contact map, ResNet)
 - tFold (contact map, ResNet)
- AlphaFold则使用了End-to-end的架构
 - AlphaFold2 (end to end)



**Convolutional Networks
(e.g. computer vision)**

- data in regular grid
- information flow to local neighbours

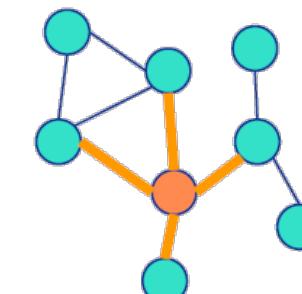
大多数主流方法 (ResNet)



**Recurrent Networks
(e.g. language)**

- data in ordered sequence
- information flow sequentially

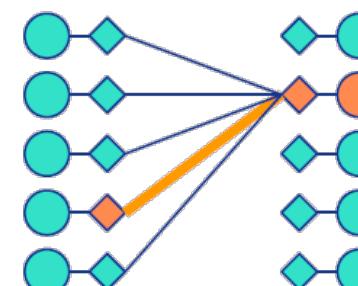
蛋白质功能预测: UniRep



Graph Networks (e.g. recommender systems or molecules)

- data in fixed graph structure
- information flow along fixed edges

SE(3)-Transformer等



Attention Module (e.g. language)

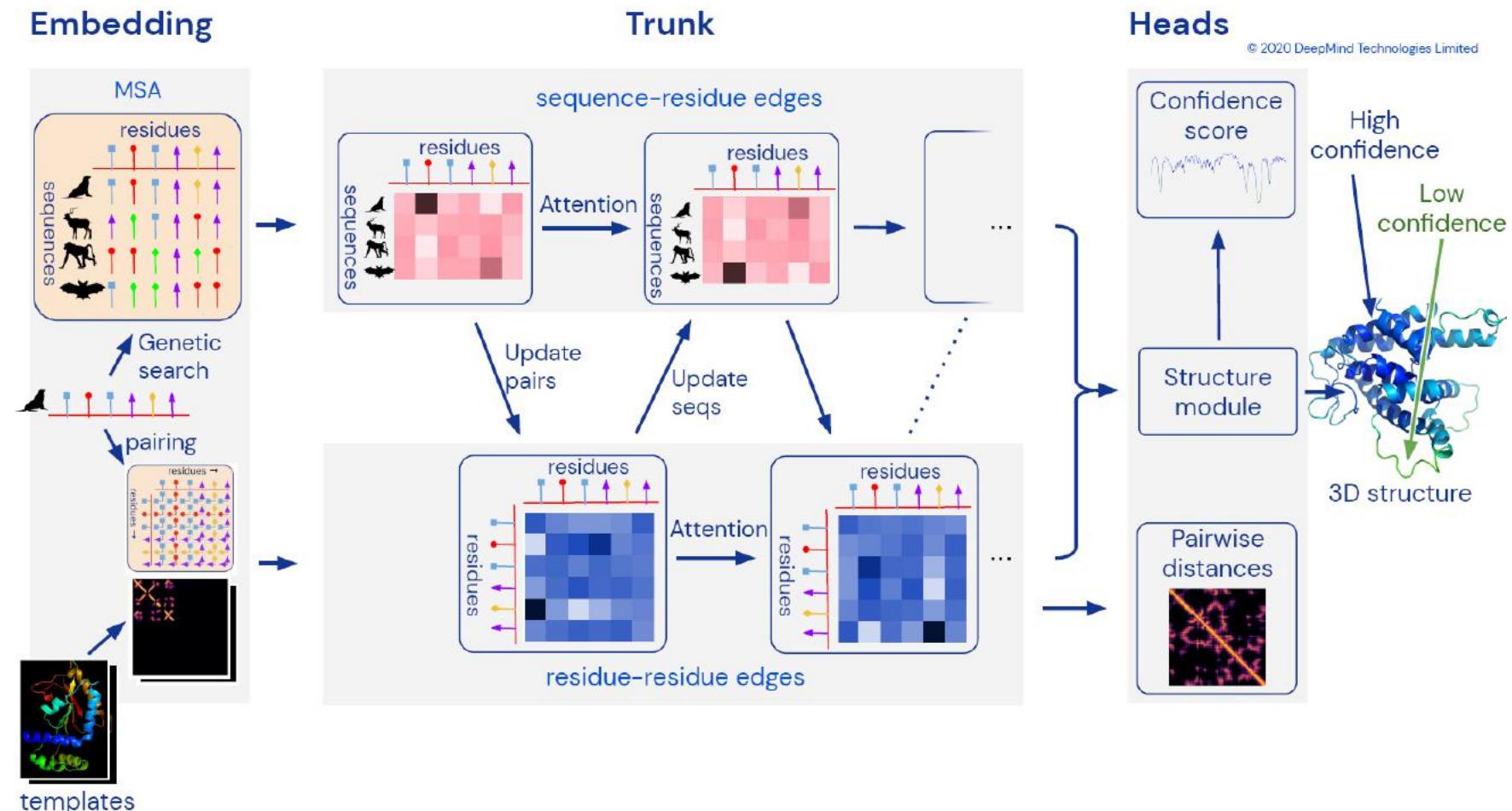
- data in unordered set
- information flow dynamically controlled by the network (via keys and queries)

AlphaFold, ProtBERT等

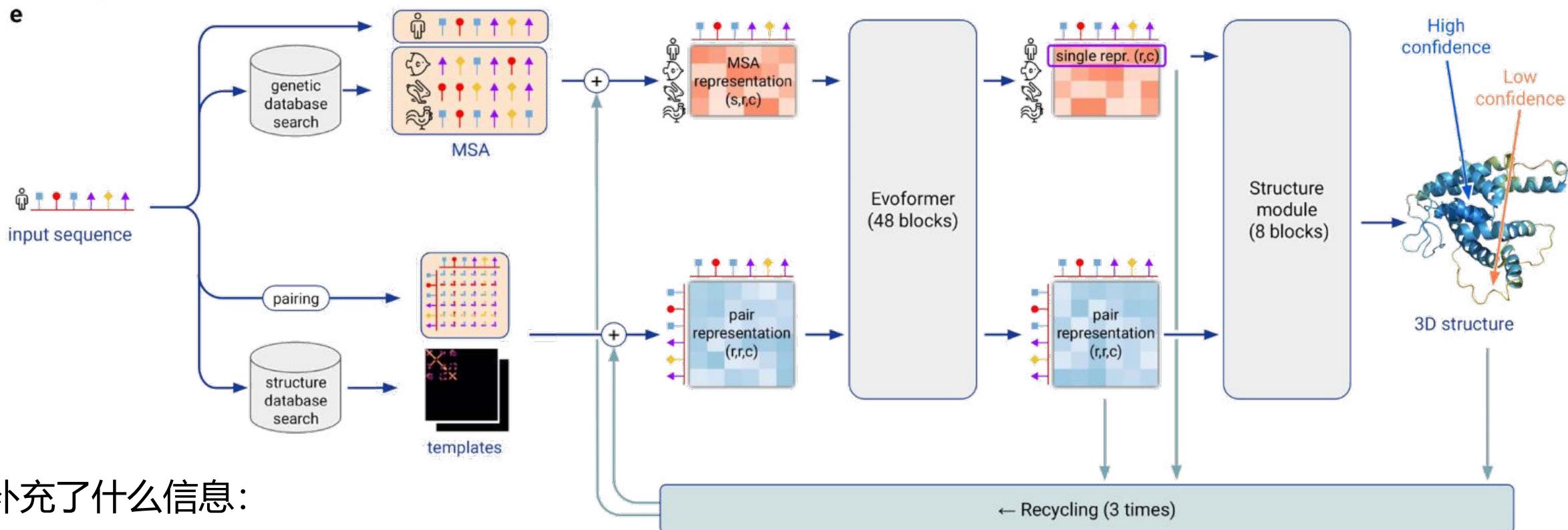
AlphaFold2的整体架构: 在2020公布的信息

主要特点:

- End-to-end 架构
- 1D与2D信息之间使用了Attention
- 3D Equivariant (等变) Structure Module



AlphaFold2的整体架构：2021年发表的Nature论文



补充了什么信息：

- 架构上补充了Recycling
- Attention模块的细节：Evoformer
- 公布了Structure module的细节：IPA, residue gas

整体架构的精彩之一： 模型输入——更强大的MSA & Templates

序列数据库

- UniRef90 (JackHMMER) 来自UniProt
- BFD (HHblits) 自建数据库
- MGnify clusters (JackHMMER) 宏基因组

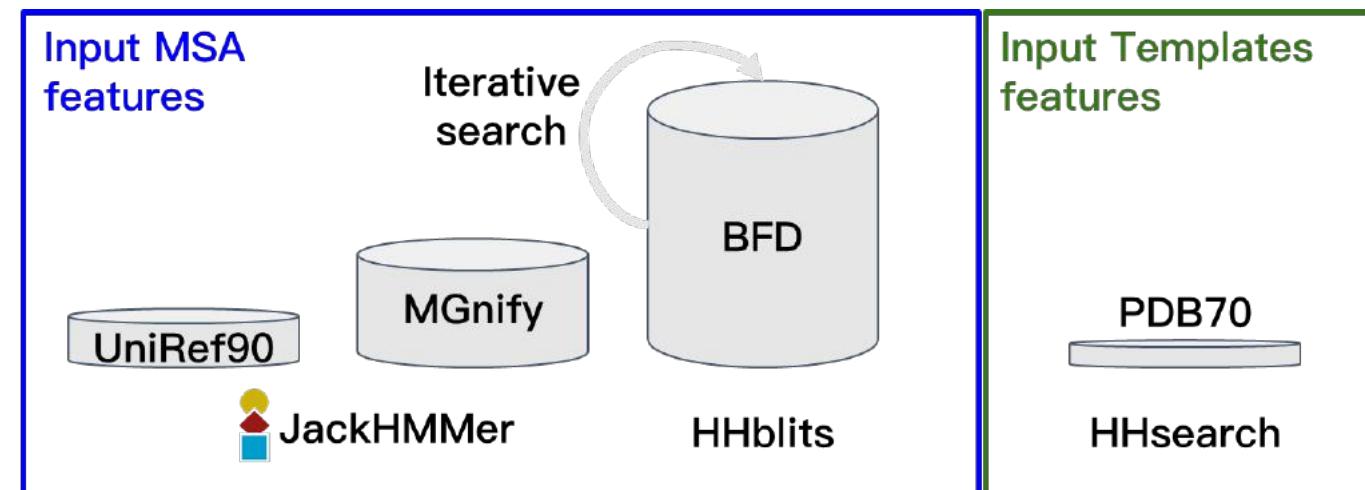
结构数据库

- PDB (用于训练)
- PDB70聚类 (HHsearch)

所有序列数据均来自公开数据库

MSA决定结构预测准确度的上限

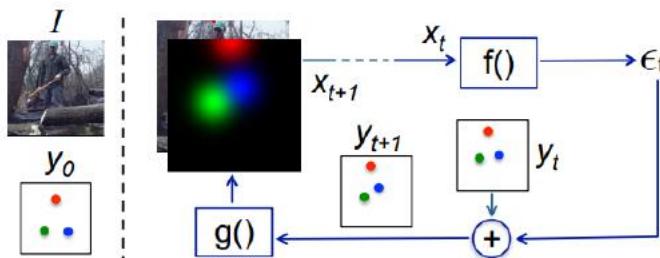
AlphaFold的序列数据库足够大，MSA结果有一定的保证



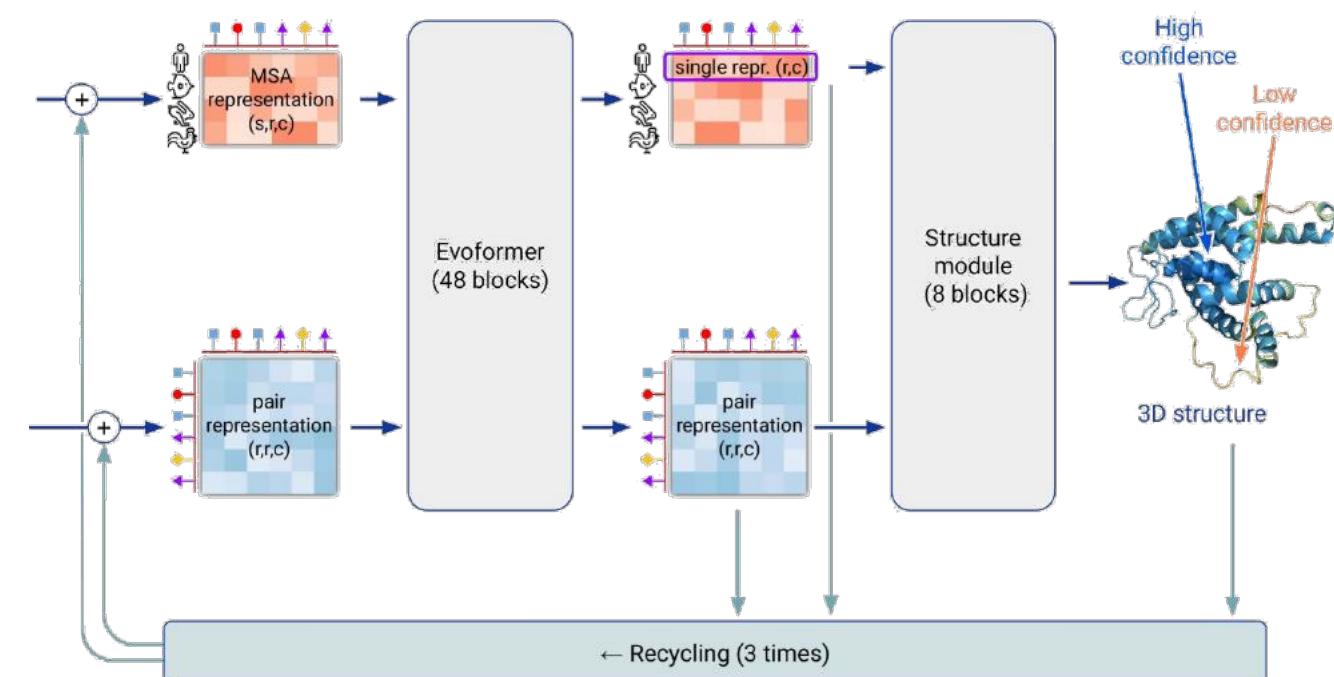
整体架构的精彩之二：

使用Recycling进行多轮迭代训练和测试

“We find it helpful to execute the network multiple times, each time embedding the previous outputs as additional inputs.”



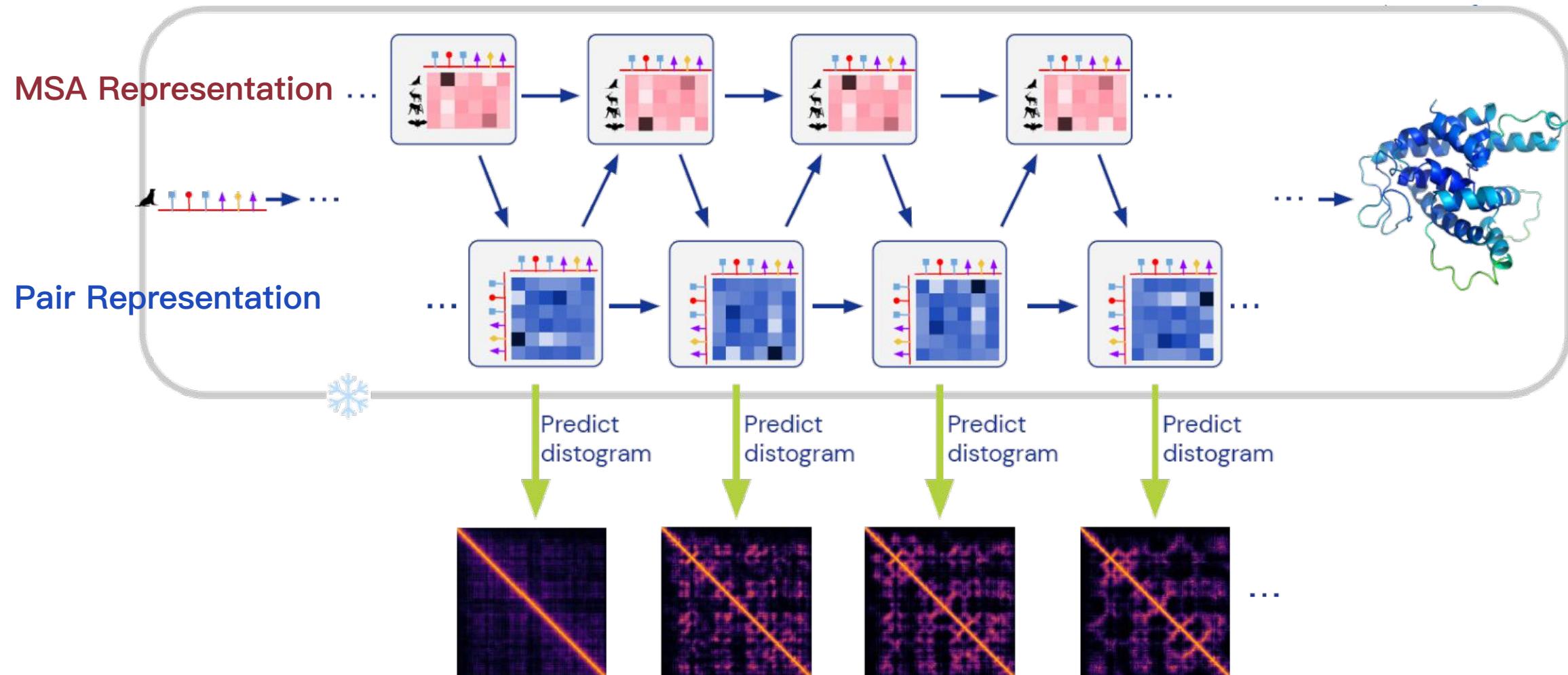
Recycling最初用于计算机视觉中的姿态估计(post estimation)问题，将训练的结果返回输入继续迭代训练



多轮迭代让模型的深度更深，能够通过迭代让结构预测更精确，预测到更复杂的结构

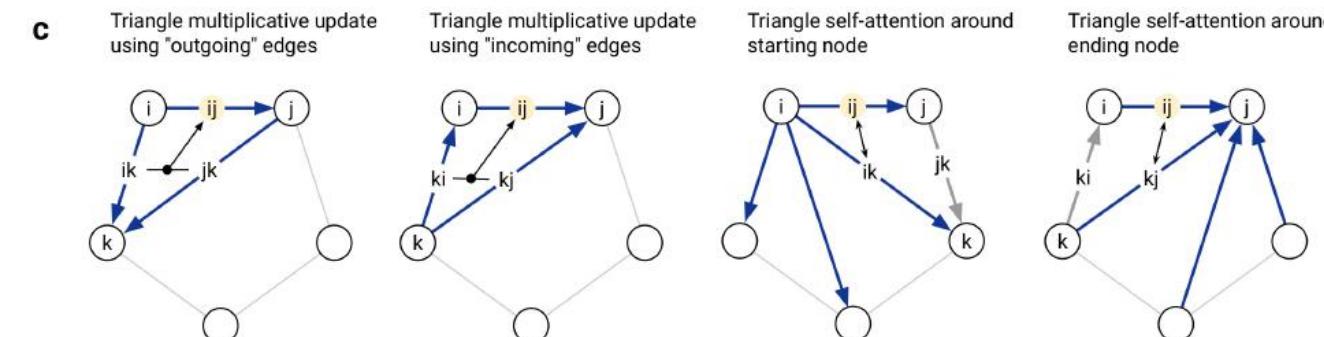
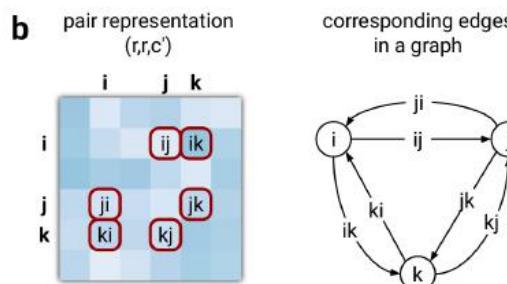
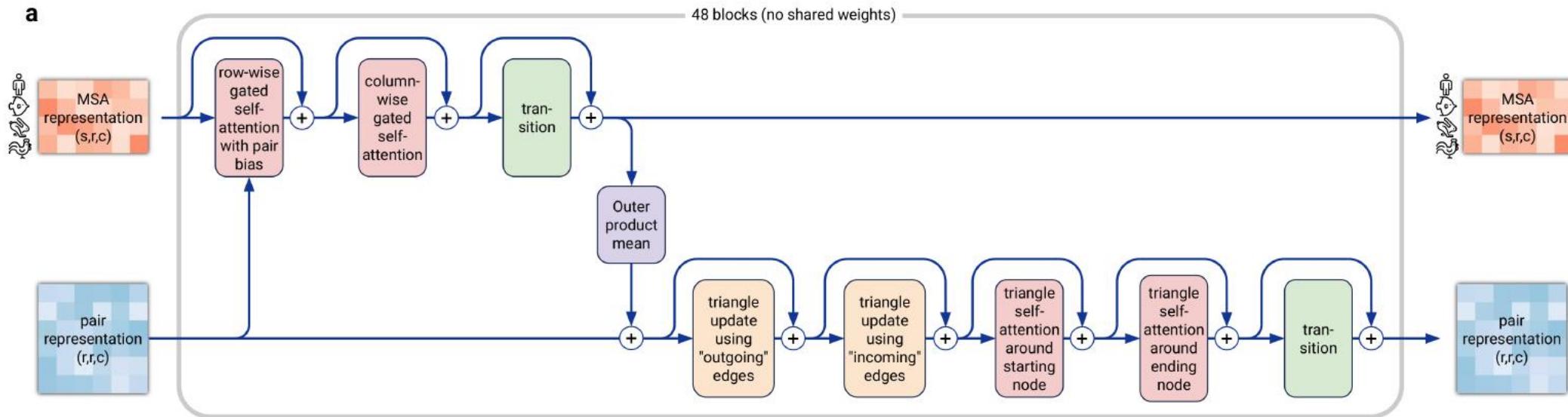
整体架构的精彩之三：

Evoformer: 用于结构预测的Attention架构



整体架构的精彩之三：

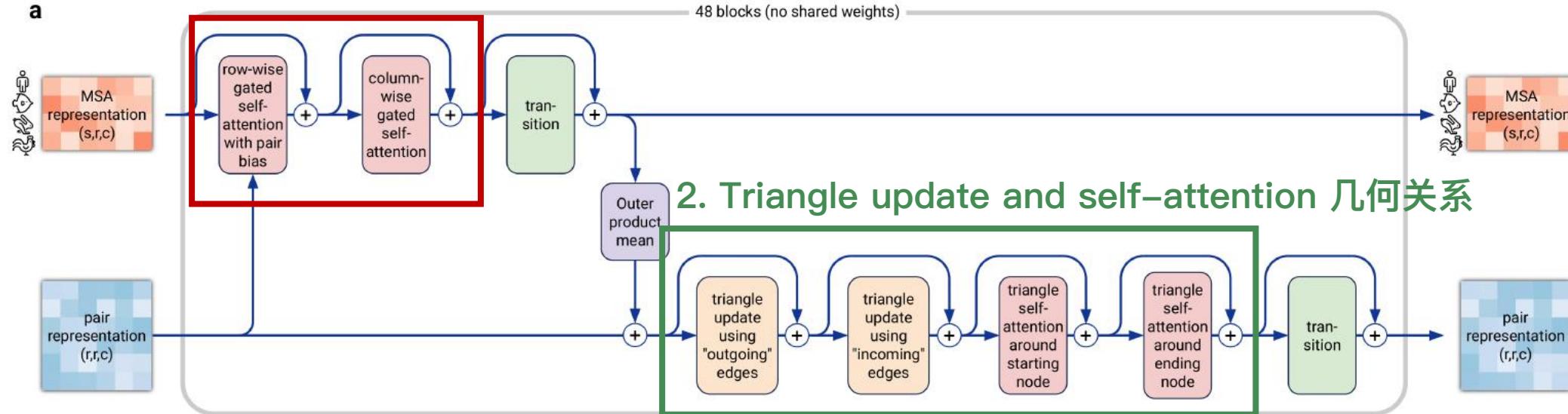
Evoformer: 实现细节



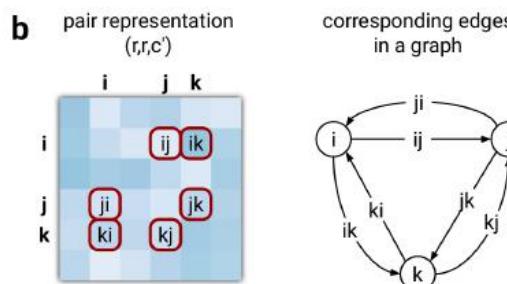
整体架构的精彩之三：

Evoformer: 实现细节

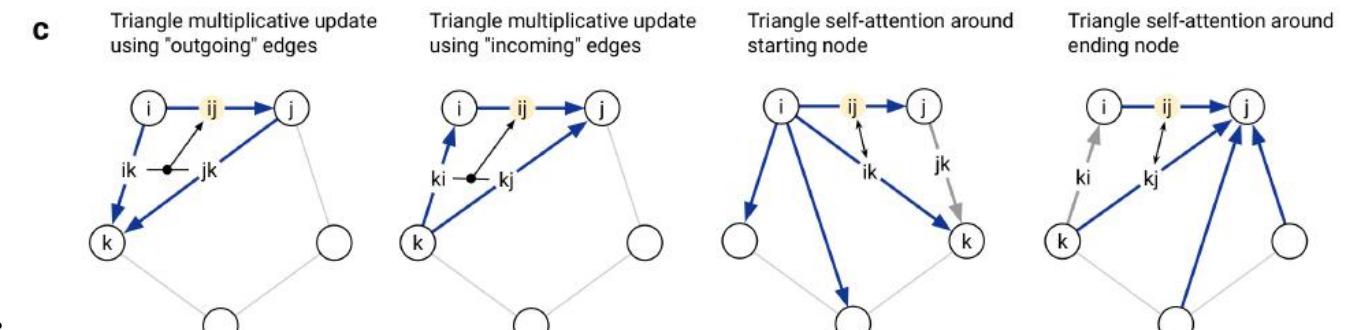
1. Row/column-wise self-attention 逐行逐列相关性



Evoformer的整体架构



Triangle update and self-attention
利用边之间的三角形关系中互相推断



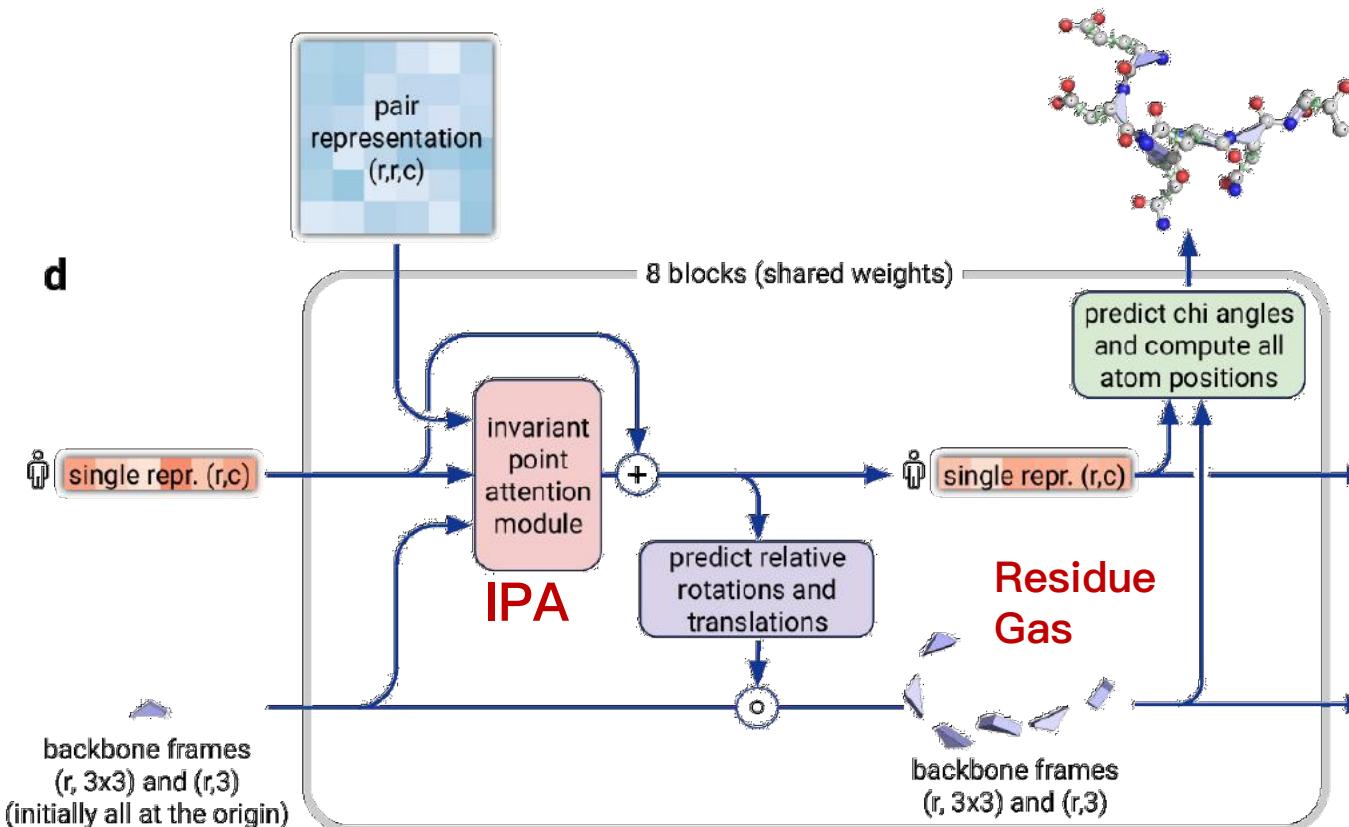
用ik, jk推断ij的信息

用ik推ji的信息，是否接受更新取决于jk边

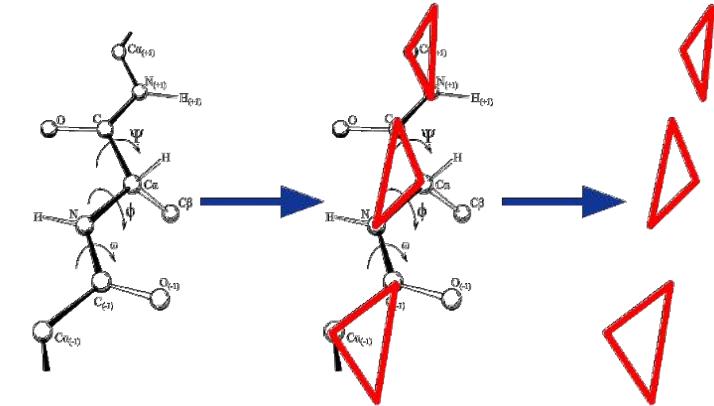
整体架构的精彩之四：

Structure Module的关键——Equivariant

重要架构：IPA (Invariant Point Attention) 和 Residue Gas



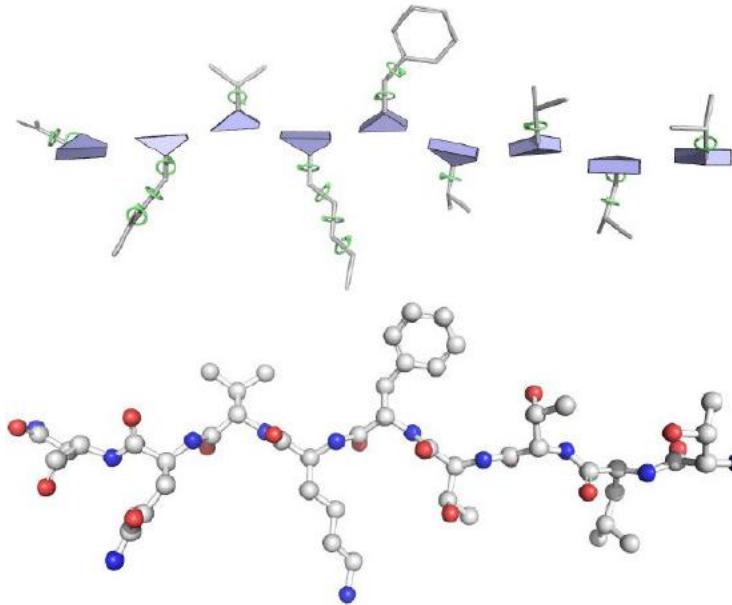
Protein backbone = gas of 3-D rigid bodies



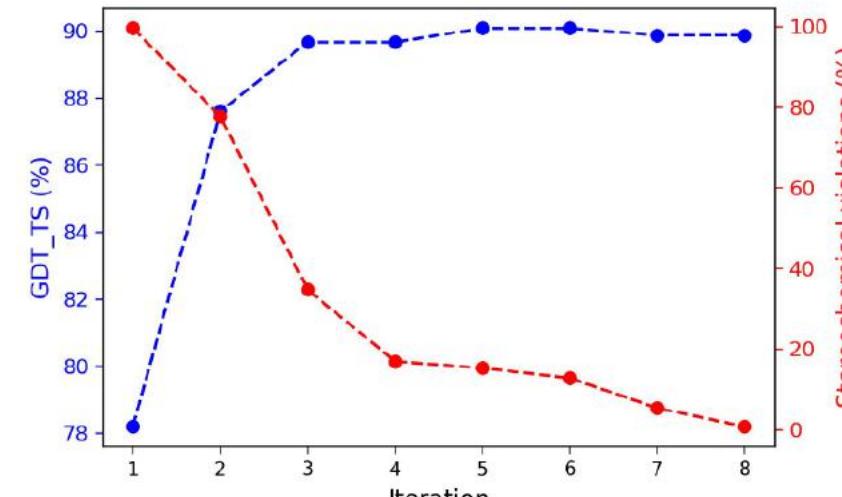
- IPA用于实现3D Equivariant（平移旋转等变性）
- Residue Gas用于表示蛋白质结构
- 输入：
 - 序列信息（目标蛋白）
 - Distance Map信息
 - 蛋白质骨架初始Residue Gas
- 输出：
 - 全原子的位置坐标
 - IDDT-C α （评估建模精度）

整体架构的精彩之四：

Structure Module中的优化过程——原子水平的优化



同时对主链结构和支链结构的优化，实现了原子水平的end-to-end三维结构预测和优化。

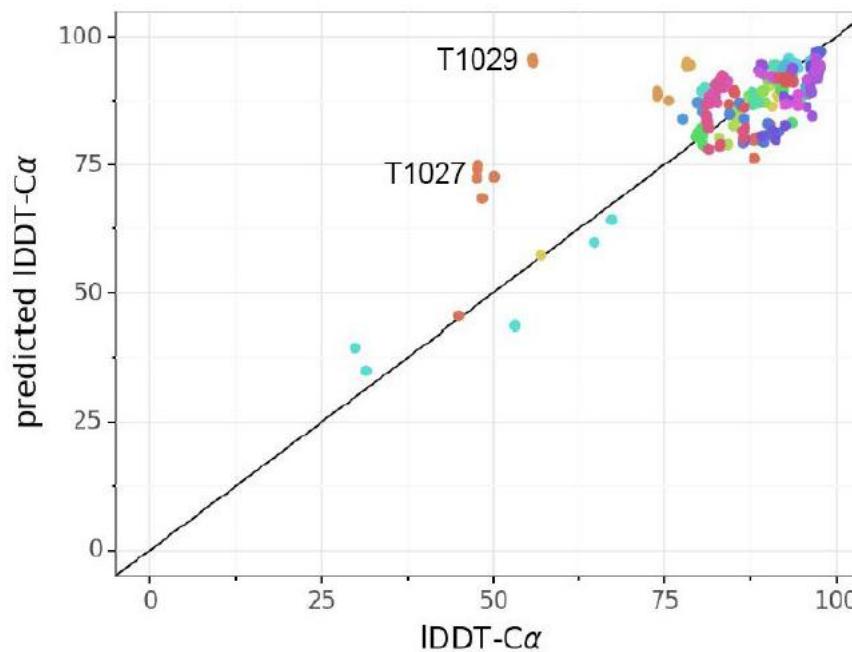


Target: T1O41

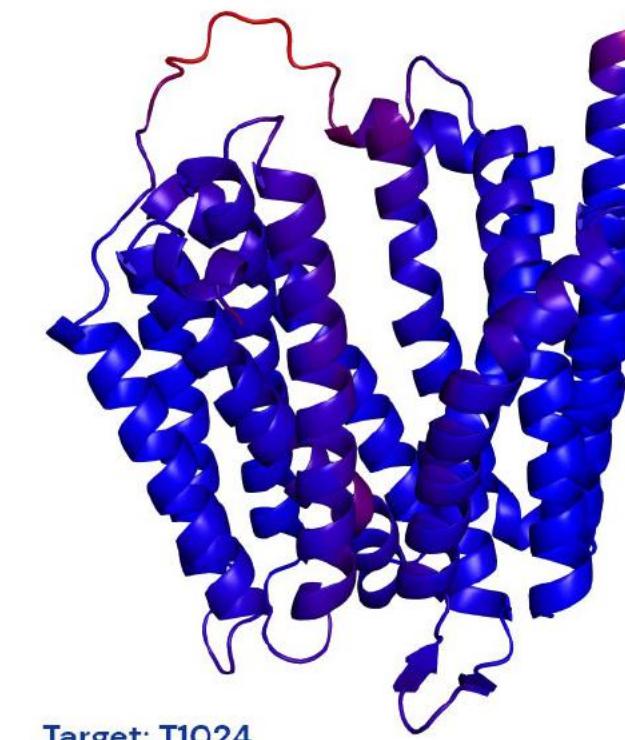
在建模准确性提升的同时，结构中的不合理成分也逐步降低

整体架构的精彩之五： 多输出——如何知道预测的结构的精确度

模型预测的IDDT-C α 与实际值十分接近
MAE=3.3

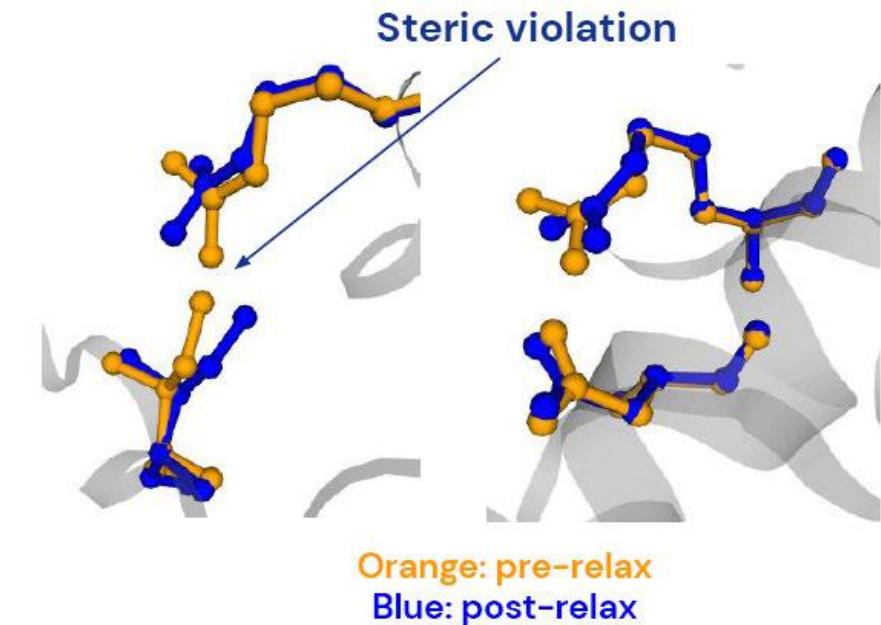


预测的IDDT-C α 能反应预测结果的准确度
(蓝色：准确度高，红色：准确度低)

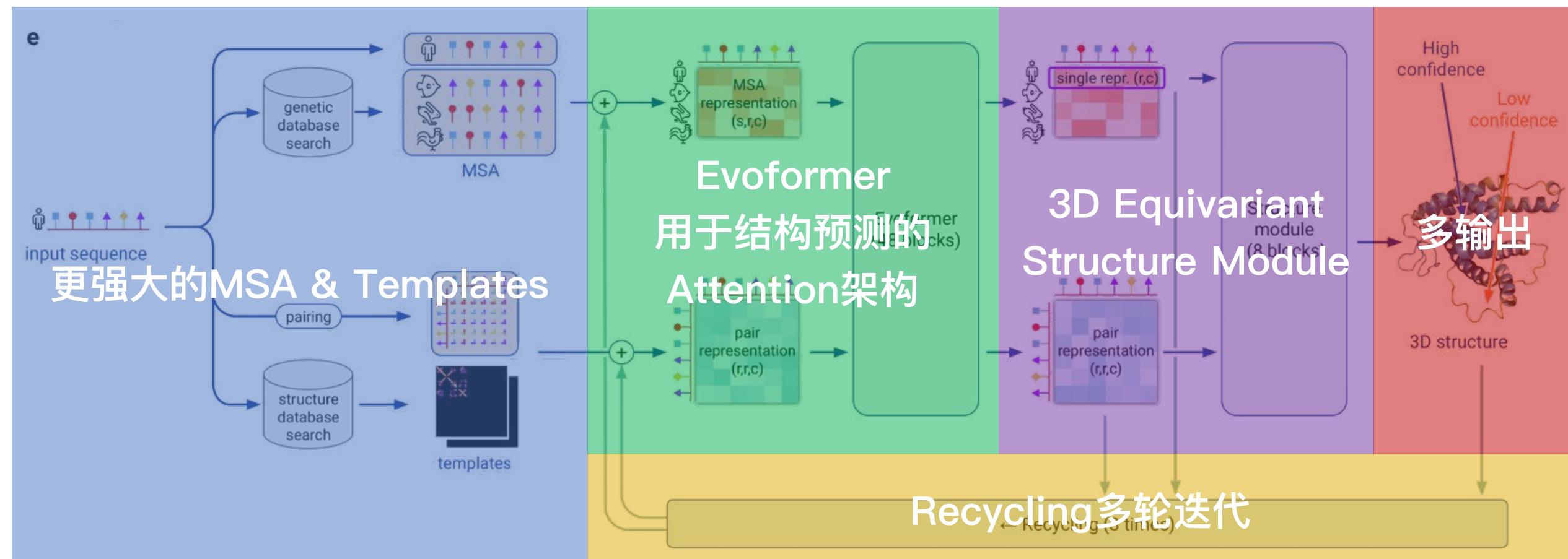


最后一步：Relaxation：AMBER

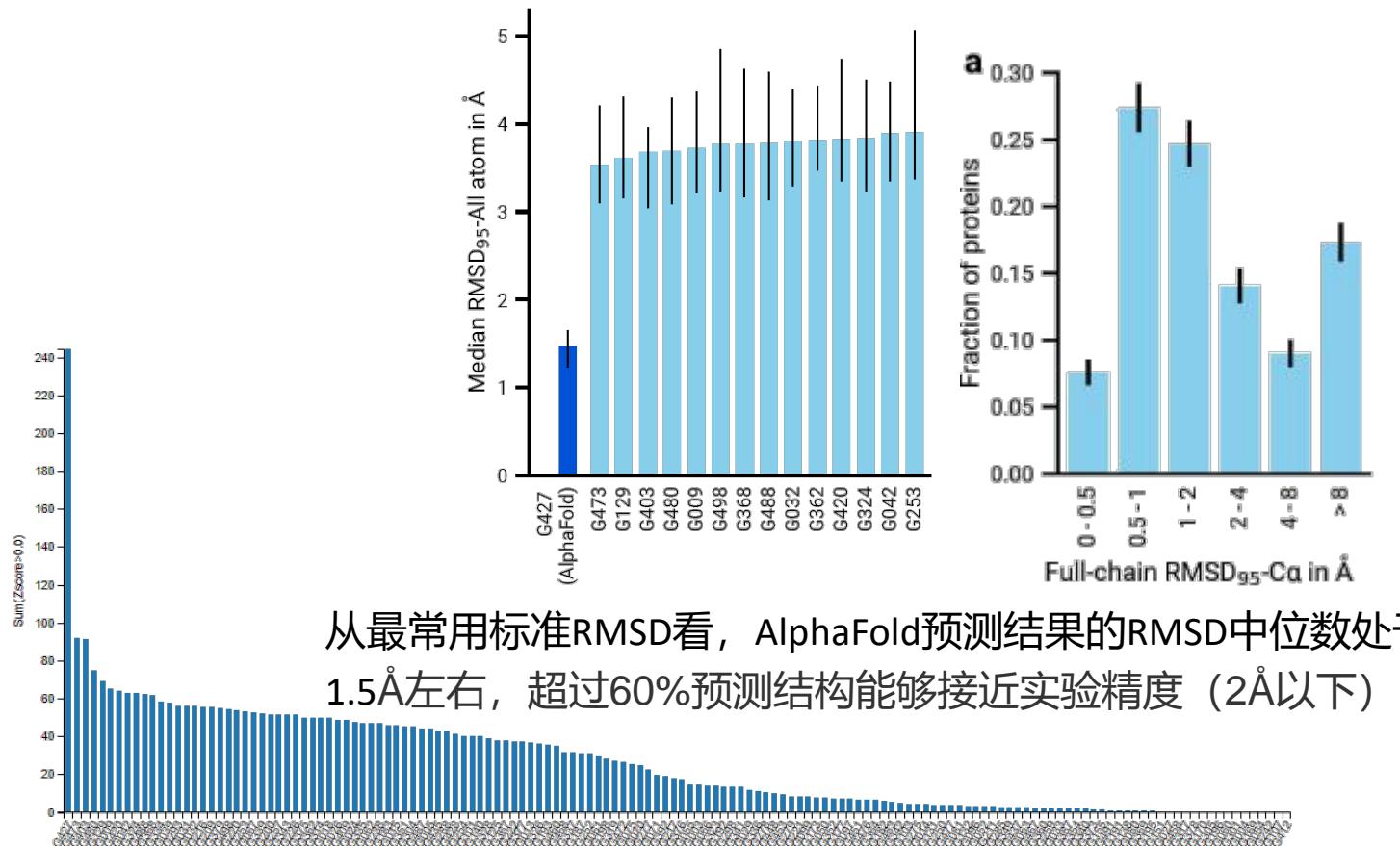
- 最终输出的结果并不一定满足所有的立体化学约束 (stereochemical constraints)
- 使用限制主链坐标的 gradient descent 来优化这些冲突
- 使用了 OpenMM 中的 AMBER ff99SB 力场用于优化
- 正如论文中指出，力场优化并不能够提升模型的pLDDT打分（即模型预测精度），只是对结构细节做小幅度的优化



再回顾AlphaFold2的整体架构



CASP14中AlphaFold2的碾压性优势

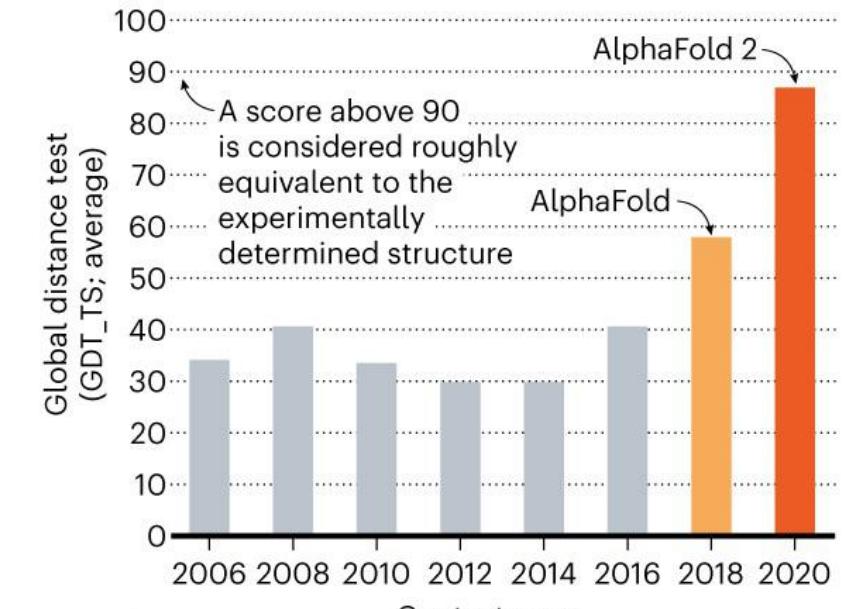


本届CASP14比赛比赛中，Alphafold的GDT-TS总分取得了碾压性的优势，远超第二名的BAKER团队

精确度的显著提升让AlphaFold2成为了突破性的成果，让科学家有足够的信心接受Alphafold的预测结果

STRUCTURE SOLVER

DeepMind's AlphaFold 2 algorithm significantly outperformed other teams at the CASP14 protein-folding contest — and its previous version's performance at the last CASP.



©nature

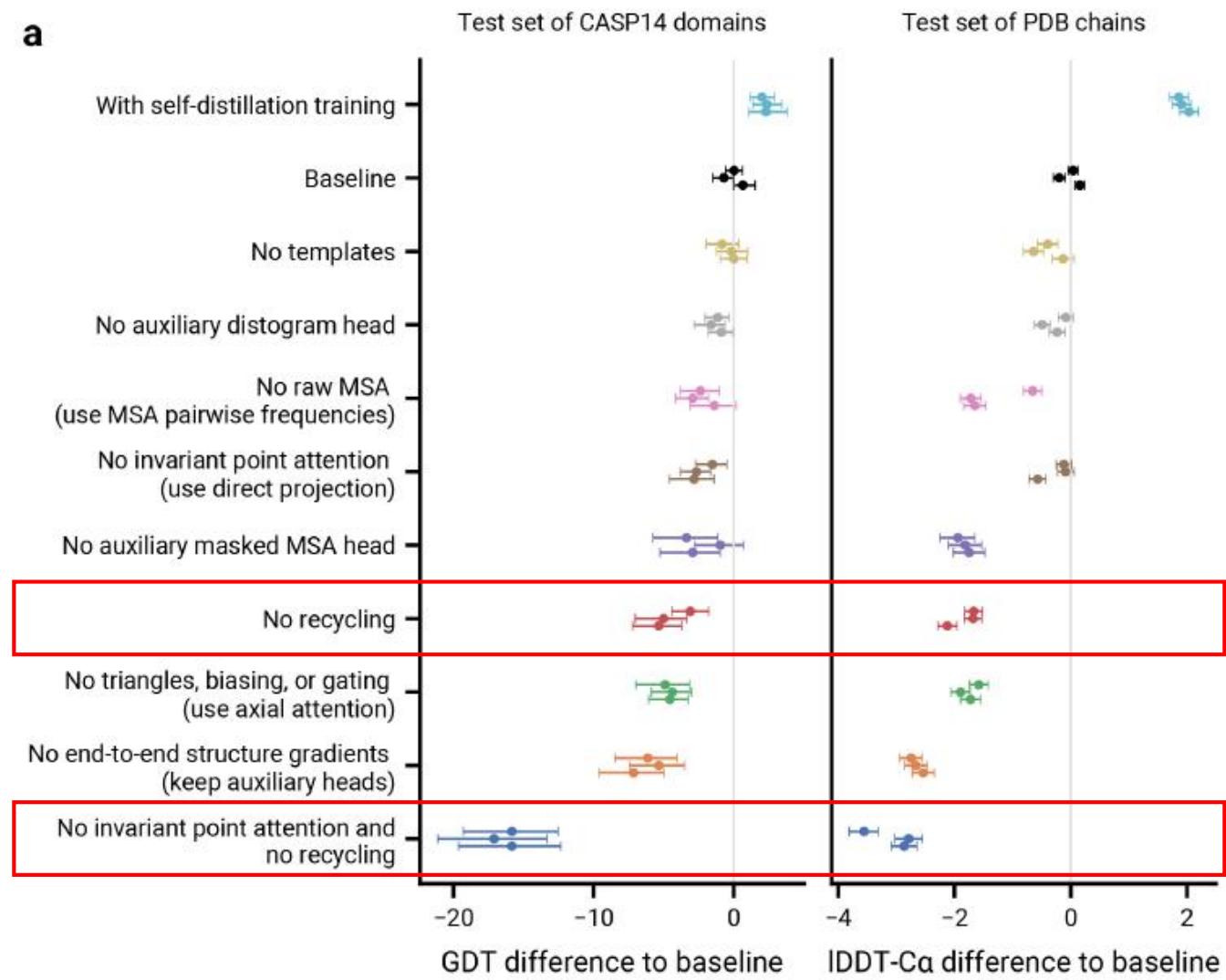
对比历届CASP的结果，AlphaFold的提升是非常显著的

AlphaFold架构中每个部分的重要性

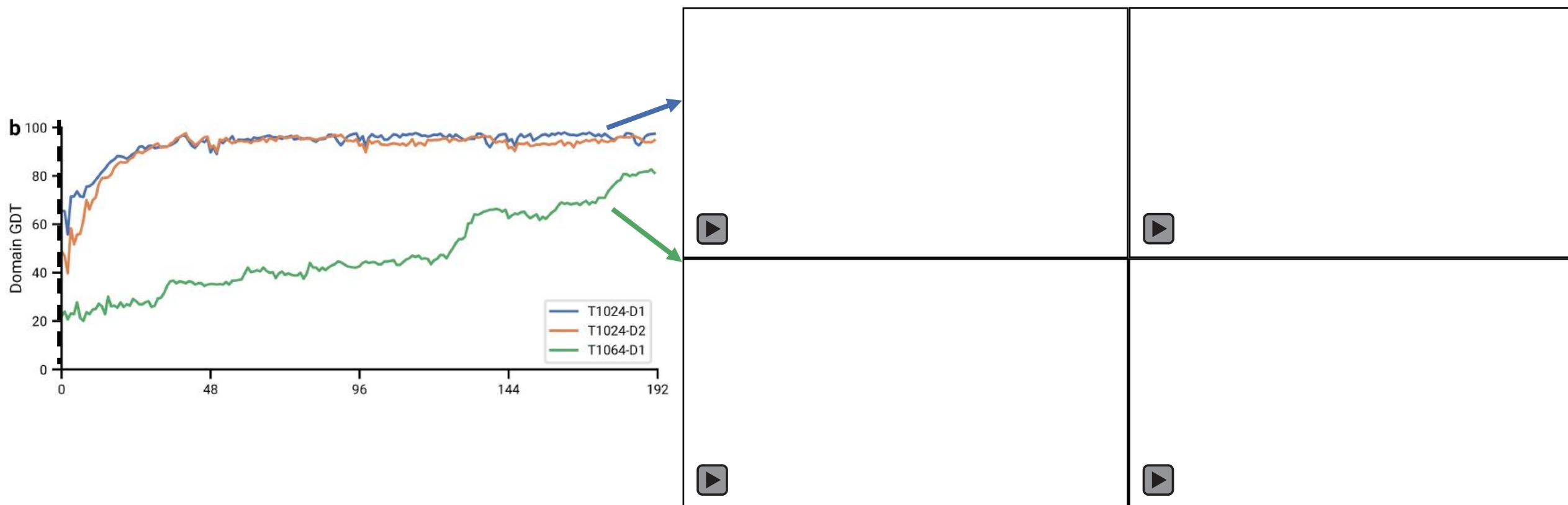
Ablation Results of AlphaFold

对于CASP14测试集

- AlphaFold本身在CASP表现为Baseline
- 不使用模板对模型损失较小
- 没有Recycling迭代会对模型的精确度有约10%的损失



有的蛋白很快就能够折叠，有的蛋白很慢



多轮迭代优化有一定的必要性，较为复杂的蛋白可能在优化流程最后
(4轮优化) 才能折叠到正确的结构

MSA深度和模板的选择

我们需要何种质量的MSA?

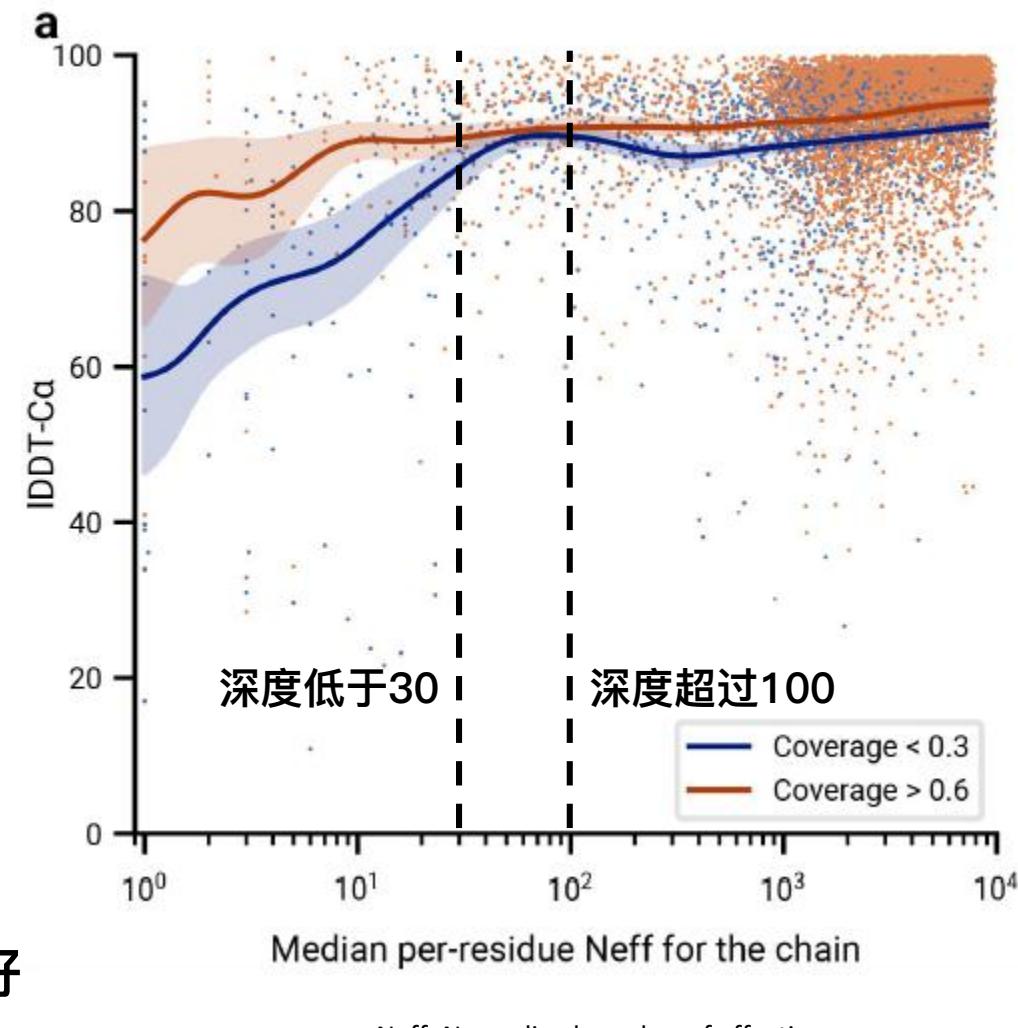
对于MSA的深度

- MSA平均深度需超过30才能取得较好的预测效果
- MSA深度超过100的则提升并不显著

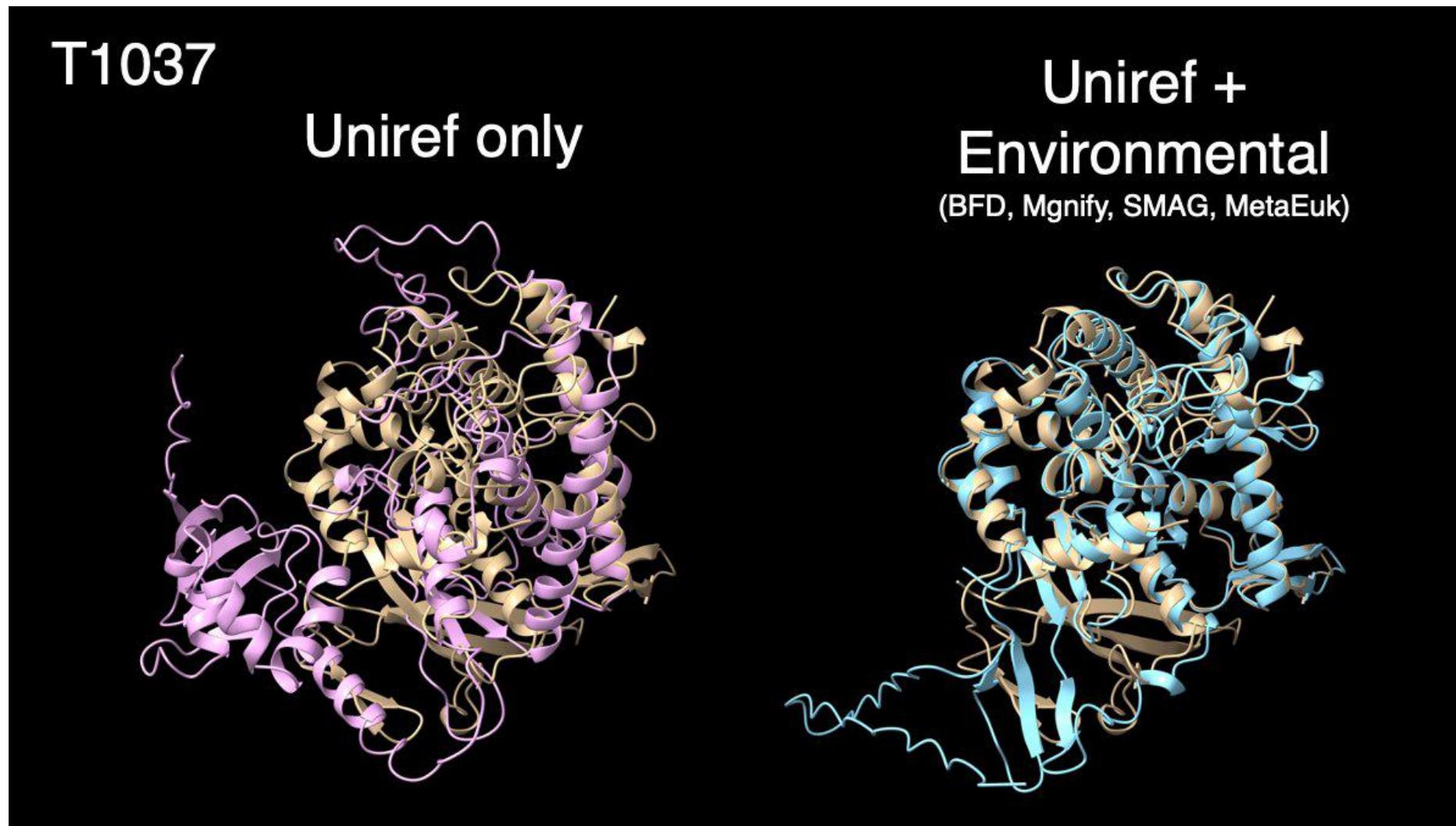
对于Template的覆盖率

- MSA深度低于30时，模板的相似度才会有比较大的对准确度的影响（高相似度模板优于低相似度模板）

MSA做的够好的话，没有很好的模板也能跑的很好

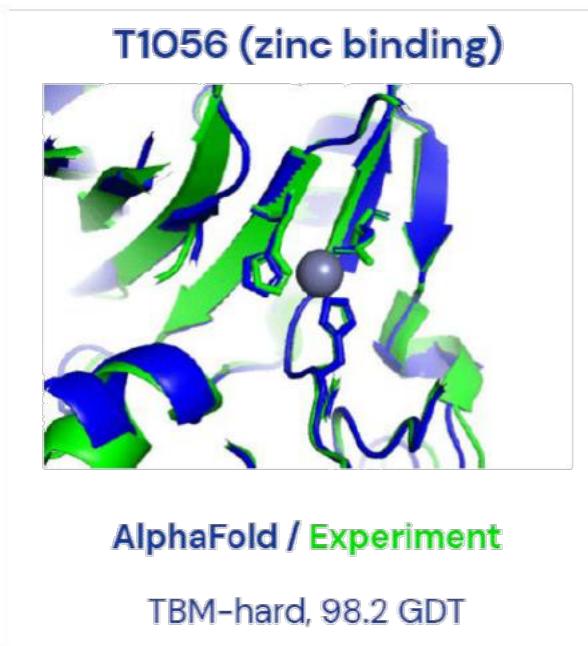


低质量的MSA确实会影响结构预测的精度

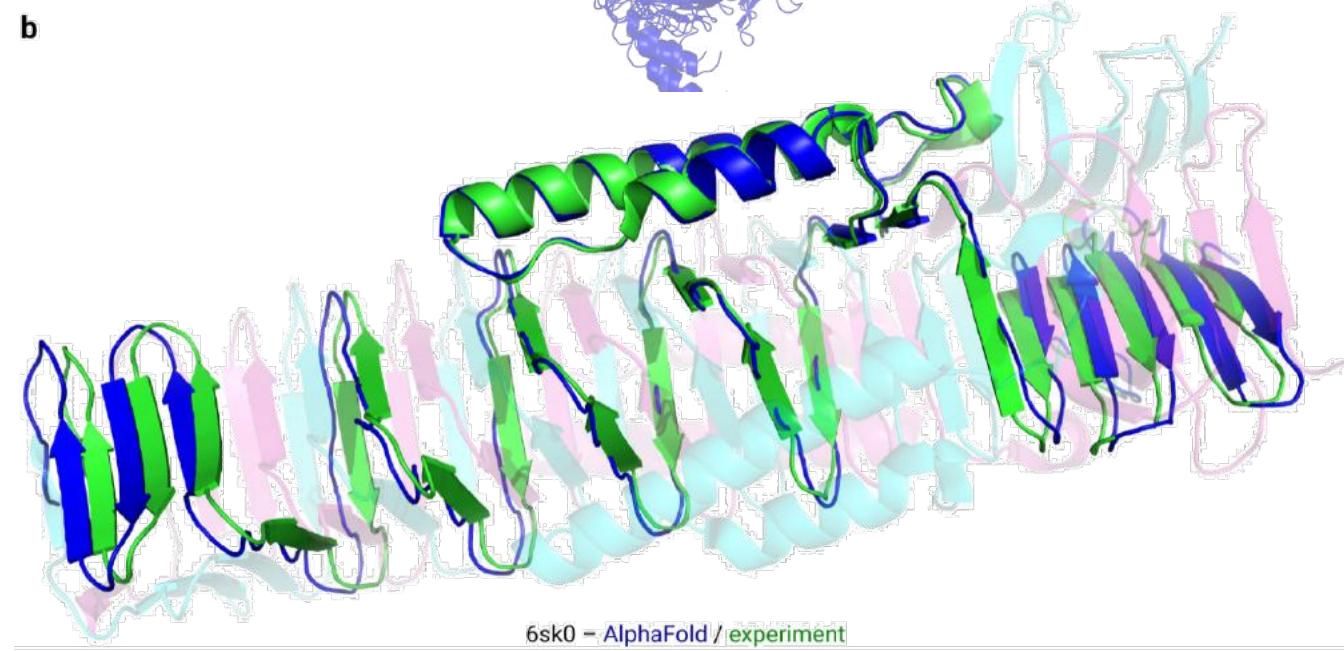


复杂的蛋白也是可以预测的

- 计算结构预测通常是不明确的
 - 低聚态，配体，DNA结合，实验条件，多种构象等情况
- 我们的网络使用了各种物理和进化信息，隐含地建模了缺失的部分，使预测结果仍然十分准确



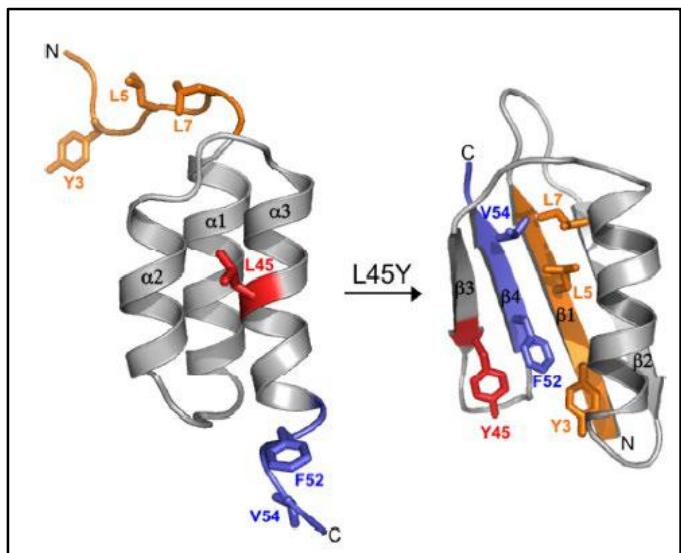
含有金属离子的体系



寡聚体蛋白质体系

Concern: AlphaFold学习了从序列到晶体结构的映射，而晶体结构并不能代表真正的蛋白质构象

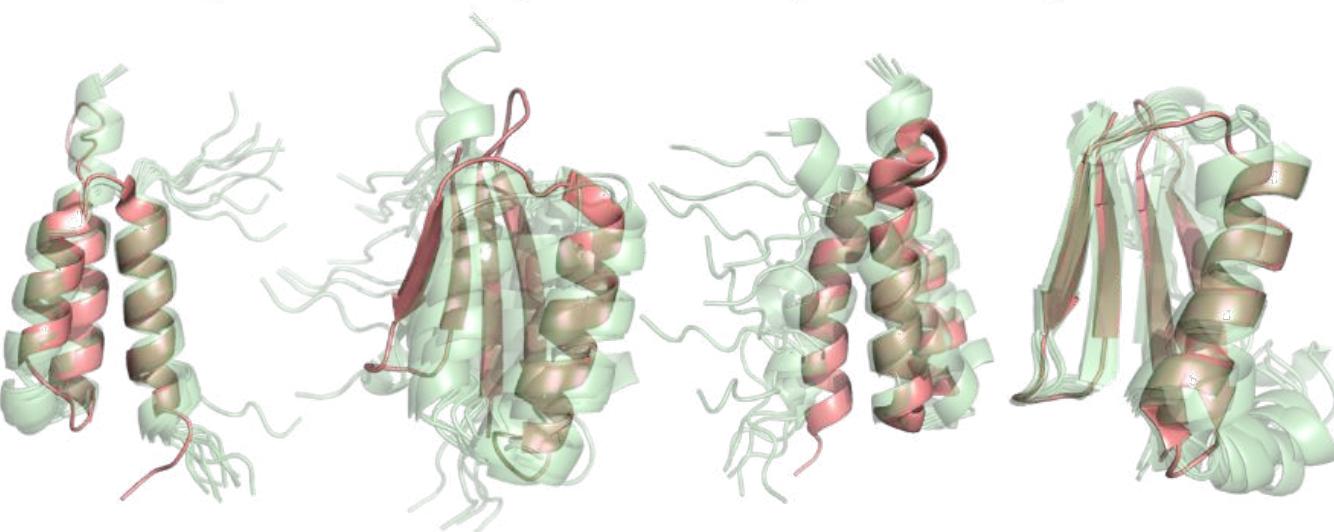
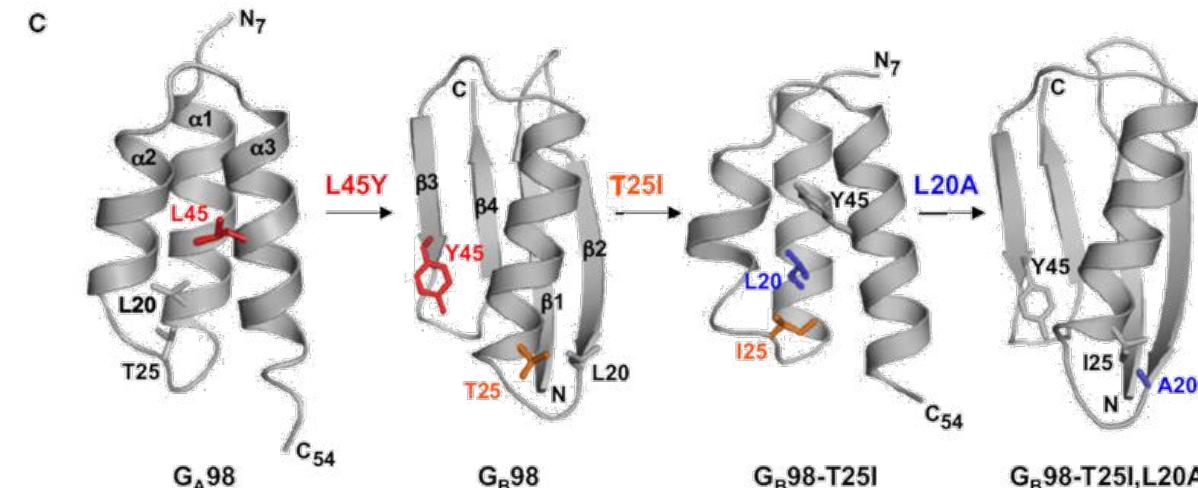
单氨基酸替换的测试：GA98 & GB98



α₁ α₂ α₃
 0 0 0 0 0 0 0 0 0 0 0 0 0
 10 20 30 40 50
 G_A77 TTYKLILNLKQAKEEAIKELVDACTAEKYIKLIANAKTVEGVWTLKDEI_{KT}FTVTE
 G_A88 TTYKLILNLKQAKEEAIKELVDACTAEKYIKLIANAKTVEGVWTLKDEI_{LT}FTVTE
 G_A91 TTYKLILNLKQAKEEAIKE_LVDAGTAEKYIKLIANAKTVEGVWTLKDEI_{LT}FTVTE
 G_A95 TTYKLILNLKQAKEEAIKE_LVDAGTAEKY_IKLIANAKTVEGVWTLKDEI_{KT}FTVTE
 G_A98 TTYKLILNLKQAKEEAIKELVDACTAEKYFKLIANAKTVEGVWTLKDEI_{KT}FTVTE

 G_B98 TTYKLILNLKQAKEEAIKELVDACTAEKYFKLIANAKTVEGVWT_YKDEI_{KT}FTVTE
 G_B95 TTYKLILNLKQAKEEAIKE_AVDAGTAEKYFKLIANAKTVEGVWT_YKDEI_{KT}FTVTE
 G_B91 TTYKLILNLKQAKEEAIKE_AVDAGTAEKY_EFKLIANAKTVEGVWT_YKDEI_{KT}FTVTE
 G_B88b TTYKLILNLKQAKEEAI_TEVDACTAEKYFKLIANAKTVEGVWT_YKDEI_{KT}FTVTE
 G_B77 TTYKLILNLKQAKEEAI_TEVDACTAEKYFKLIANAKTVEGVWT_YKDET_{KT}FTVTE

 β1 → β2 → α₁ → β3 → β4 →



在一个典型的单氨基酸突变导致结构变化的例子中，我们测试结果显示AlphaFold仍然能预测到正确的结构，但也有时预测错误（使用MSA，未使用结构模板）

AlphaFold 优点总结和补充

- 基于recycling的迭代优化。这一点在很多领域已经得到过应用，比如计算机视觉中的姿态估计 (post estimation)
- 广泛应用的Attention架构。将二维的表横着做Attention、再竖着做Attention，对于图可以在局部做Attention，不断精化了Embedding过程；Structure module中也继续用到了Attention
- 半监督学习拓展训练集 (Self Distillation)。用带标签的数据先训练一遍，再用无标签的数据预测一遍形成新的数据集，然后再混合继续训练。这种方法曾经在Google Brain的noisy student使用过，在这里再次得到了应用
- 类似BERT的mask结构。Mask对各种输入添加噪音以增加模型的鲁棒性，这在BERT类模型中非常的常见

Self Distillation策略

Self-training是最简单的半监督方法之一，其主要思想是找到一种方法，用未标记的数据集来扩充已标记的数据集。算法流程如下：

1. 利用已标记的数据来训练一个好的模型，然后使用这个模型对未标记的数据(Uniclust30)进行标记
2. 进行伪标签的生成，因为我们知道，已训练好的模型对未标记数据的所有预测都不可能都是好的，因此通常使用分数阈值过滤部分预测。这里选择了高可信度的预测结果的子集。
3. 将生成的伪标签与原始的标记数据相结合，并在合并后数据上进行联合训练
4. 整个过程不断重复，直到达到较好的预测结果

Mask策略

遮挡或者突变部分MSA中的氨基酸。

这个目标鼓励网络学习解释系统发育和共变关系，而不把特定的相关统计量编码到特征中。BERT目标与正常的PDB结构损失在相同的训练实例上共同训练，与最近的工作不同，它不是预先训练的。

AlphaFold 的成就与不足

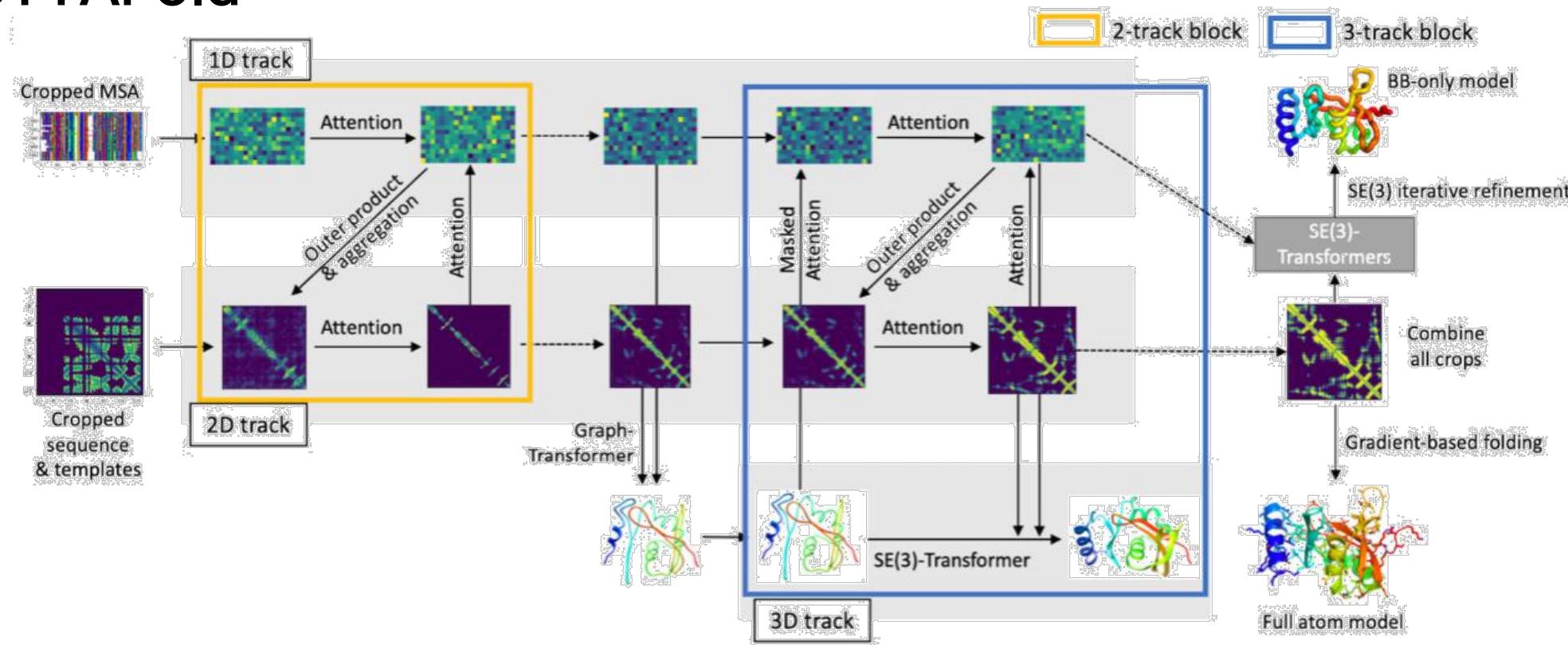
成就

- 完整建立了用于蛋白质结构预测的端到端(end-to-end)架构
- 将物理信息和几何信息融入模型，而不是使用搜索方式找到结构
- 模型能够预测自己的准确性，可以用于建模打分和排序
- 实现了计算机蛋白质建模极高的精确度

不足

- 建模输入限制于单链
- 只能建模蛋白（20种常见氨基酸），不能识别修饰、核酸、小分子、金属离子
- 本质上来说，得到的结果是晶体结构，但晶体结构并不一定能够代表真实结构

RoseTTAFold

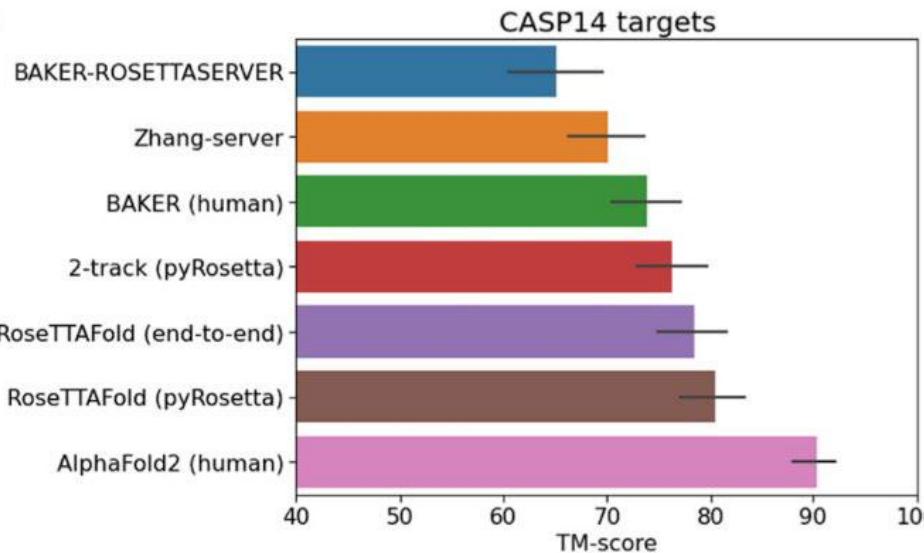


比较明显的区别：没有Recycling（2020年AlphaFold并未提到）

细节上的区别：Attention模块差异巨大，SE(3)-Transformer与Structure Module有差异

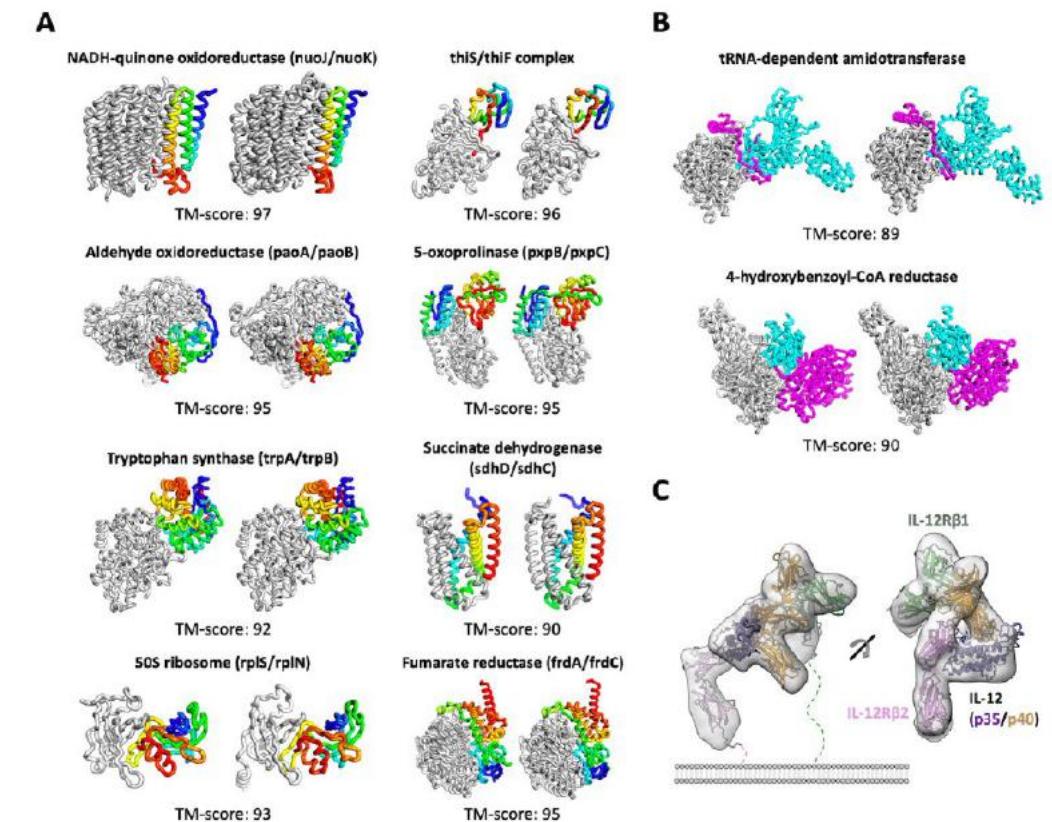
- AlphaFold中的Attention模块有很多的小trick，比如三角形优化等
- Structure Module也有很多专为蛋白质特化的设计，如Residue Gas

RoseTTAFold结果



相较于AlphaFold仍有差距 (80→90)

相比于BAKER之前的方案提升并不高 (75→80)



RoseTTAFold已经能够预测蛋白质复合物的结构

AlphaFold Github

<https://github.com/deepmind/alphafold>

硬件软件需求

- Download **genetic databases** - Total: ~ 2.2 TB (download: 428 GB)
- Download **model parameters** (3.5 GB)

论文公布的一些训练和测试的条件

- 训练使用了128张TPUv3（较大的算力），初步训练用了约一周，进一步调试用了4天
- 测试蛋白所需的时间取决于蛋白长度：（测试体系均基于1张V100，2500残基是4张V100）
 - DeepMind: 256个残基需4.8分钟；384个残基需9.2分钟；2500个残基需18小时
 - 我们的测试(不含MSA、JAX编译时间): 56个残基需6秒/模型；841残基需要5分钟/模型
- 大蛋白的预测很容易超出显存，对于16G的V100来说，上限是约1300个残基。2500残基的蛋白用了4张V100（论文数据）

AlphaFold 本地实现可能遇到的问题

<https://github.com/deepmind/alphafold/issues>

硬件要求并不高

- 运行AlphaFold只需要一张A100或者1~2张V100即可
 - 显存大小是限制模型运行的因素，显存较小的显卡在运行大蛋白时可能报错
 - 多卡：只能共享显存，不支持加速，by CUDA unified memory
- MSA数据的存储空间：3T
 - 可以通过其他MSA方法如MMseq2来减少存储空间，不过这样会损失MSA数据量上的优势
- Model本身占用空间很小（3.5G）

我想试试AlphaFold，有打开网页就能用的方法吗？

Martin Steinegger 和 Sergey Ovchinnikov在 Google Colab 复现了 AlphaFold

- Google Colab是基于Google云端硬盘的应用，使用共享的计算资源在云端运行深度学习模型
- 只需访问notebook，输入想建模的蛋白序列，再全部运行即可
- 已经测试过一个长度为79aa的小蛋白，用时约7分钟完成建模
- 缺点：MSA方法使用的是MMseq2，相比于Alphafold会有一定差距；Colab提供的硬件难以保证；Colab资源有限（Colab会员制度）

```
▶ Input protein sequence here before you "Run all"  
query_sequence: "MAKTIKITQTRSAIGRLPKHKATLLGLGLRRIGHTVEREDTPAIRGMINAVSFMVKVEE  
jobname: "RL30_ECOLI  
  
Advanced settings  
num_models: 5  
msa_mode: MMseqs2  
use_amber:   
use_templates: 
```

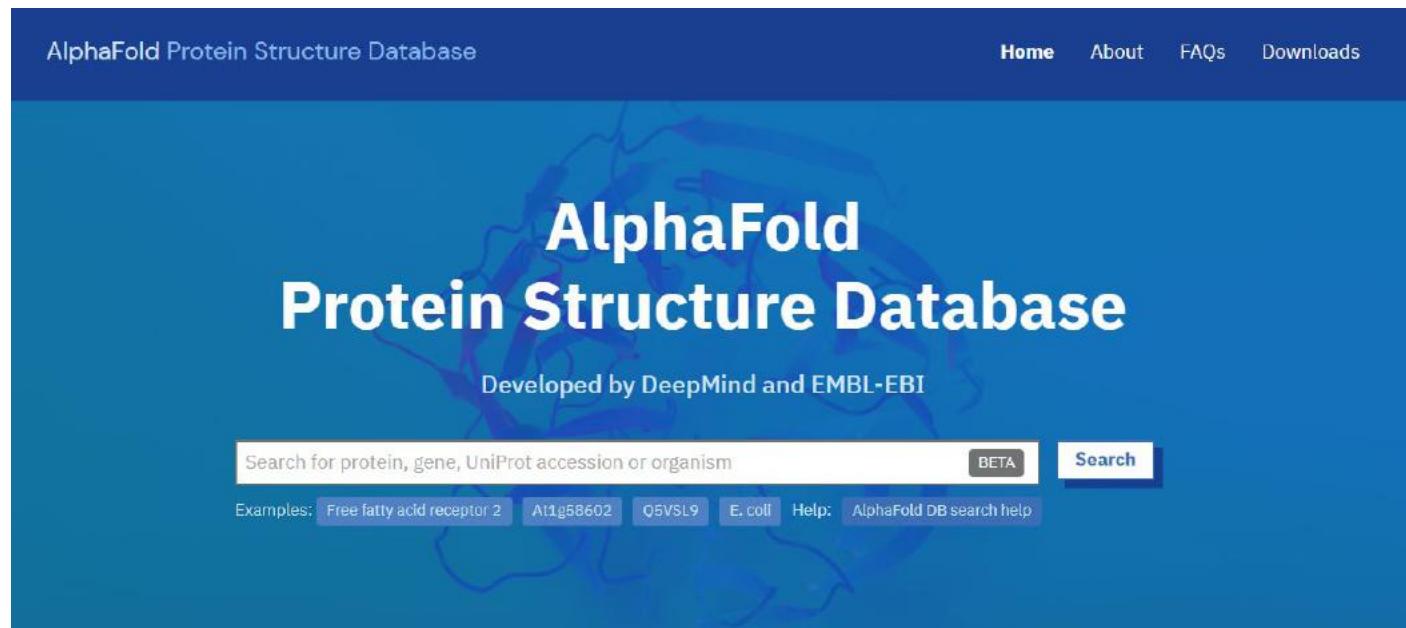


Google Colab 需要能访问外网

<https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb>

AlphaFold Database

- DeepMind 使用 AlphaFold 预测了很多常见生物的蛋白质组结构，包括人类、小鼠、大肠杆菌等
- 很多常用蛋白的结构都可以直接到AlphaFold DB下载，而且AlphaFold DB也已经接入蛋白质数据库UniProt



Species	Common Name	Predicted Structures
<i>Arabidopsis thaliana</i>	Arabidopsis	27,434
<i>Caenorhabditis elegans</i>	Nematode worm	19,694
<i>Candida albicans</i>	<i>C. albicans</i>	5,974
<i>Danio rerio</i>	Zebrafish	24,664
<i>Dictyostelium discoideum</i>	<i>Dictyostelium</i>	12,622
<i>Drosophila melanogaster</i>	Fruit fly	13,458
<i>Escherichia coli</i>	<i>E. coli</i>	4,363
<i>Glycine max</i>	Soybean	55,799
<i>Homo sapiens</i>	Human	23,391
<i>Leishmania infantum</i>	<i>L. infantum</i>	7,924
<i>Methanocaldococcus jannaschii</i>	<i>M. jannaschii</i>	1,773
<i>Mus musculus</i>	Mouse	21,615
<i>Mycobacterium tuberculosis</i>	<i>M. tuberculosis</i>	3,988
<i>Oryza sativa</i>	Asian rice	43,649
<i>Plasmodium falciparum</i>	<i>P. falciparum</i>	5,187
<i>Rattus norvegicus</i>	Rat	21,272
<i>Saccharomyces cerevisiae</i>	Budding yeast	6,040
<i>Schizosaccharomyces pombe</i>	Fission yeast	5,128
<i>Staphylococcus aureus</i>	<i>S. aureus</i>	2,888
<i>Trypanosoma cruzi</i>	<i>T. cruzi</i>	19,036
<i>Zea mays</i>	Maize	39,299

AlphaFold DB 中包含的蛋白质组

如何评估 AlphaFold 预测结果

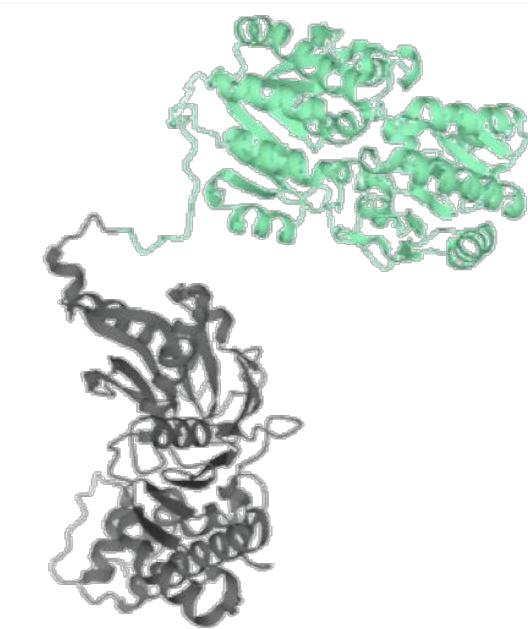
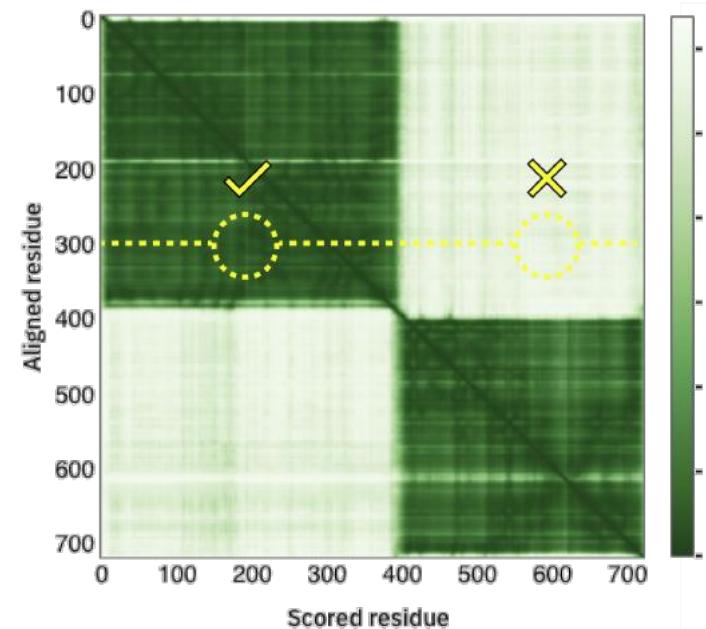
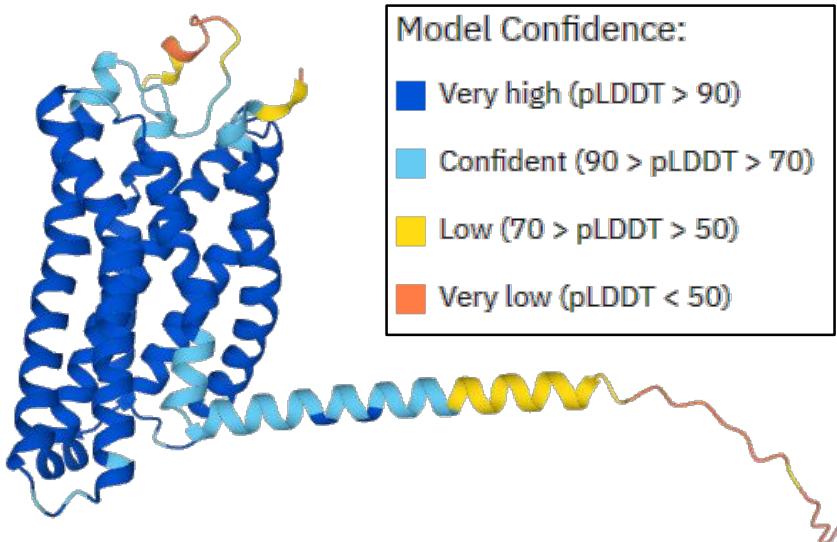
最常用的用于评估AlphaFold建模精度的指标：

pLDDT：反应每个氨基酸的预测精确度：

- 低于50表示非常不准确或为无规蛋白
- 低于70表示可信度低
- 高于90可信度极高

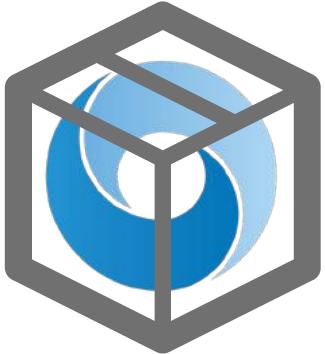
PAE：反应每对氨基酸距离与真实值的预测差值：

- 越低越好，单位为Angstrom
- 在反应多Domain或Complex结构精度是效果很好
- 只有pTM模型才有PAE打分



在 SJTU π2.0 上使用 AlphaFold

在 SJTU π2.0 上我们提供了四种不同的使用 AlphaFold 的方法



Module 版

使用方便
高度封装
支持多GPU



Conda 版

需要安装环境
源代码可调
自由度高



Colab 版

MSA较快
预测蛋白质复合物
调用比Colab更多资源



ParallelFold

高通量预测
运算效率提升
更多功能更新中

Module 版本和 Conda 版本

Module 版本

- 高度封装的版本，加载即用，免除安装困难
- 可调参数比较少
 - 输入序列
 - 预设方案：casp14/full_dbs/reduced_dbs
 - 模板时间：设置结构模板搜索范围

- reduced_dbs: 使用了较少的BFD，没有ensembling，只需要8核CPU，8G内存，600G存储
- full_dbs: 比casp14快8倍，取消了ensembling，GDT只降低了0.1
- casp14: CASP14比赛中用的设定，8轮ensembling
- Ensembling: the trunk of the network is run multiple times with different random choices for the MSA cluster centres

Conda 版本

- 相当于本地安装的版本，包含完整的AlphaFold代码和运行环境
- 需要安装AlphaFold环境，请参考<https://docs.hpc.sjtu.edu.cn/app/bioinformatics/alphafold2.html>
- 支持自定义，如选取计算 5 CASP14 models 和 5 pTM models 的全部或部分、修改 Recycling 次数、选择是否 Amber 优化、设定 data 数据集位置等，均可通过修改代码实现

Colab 版本

如果你听过ColabFold，那我们的Colab版本即相当于本地实现的ColabFold

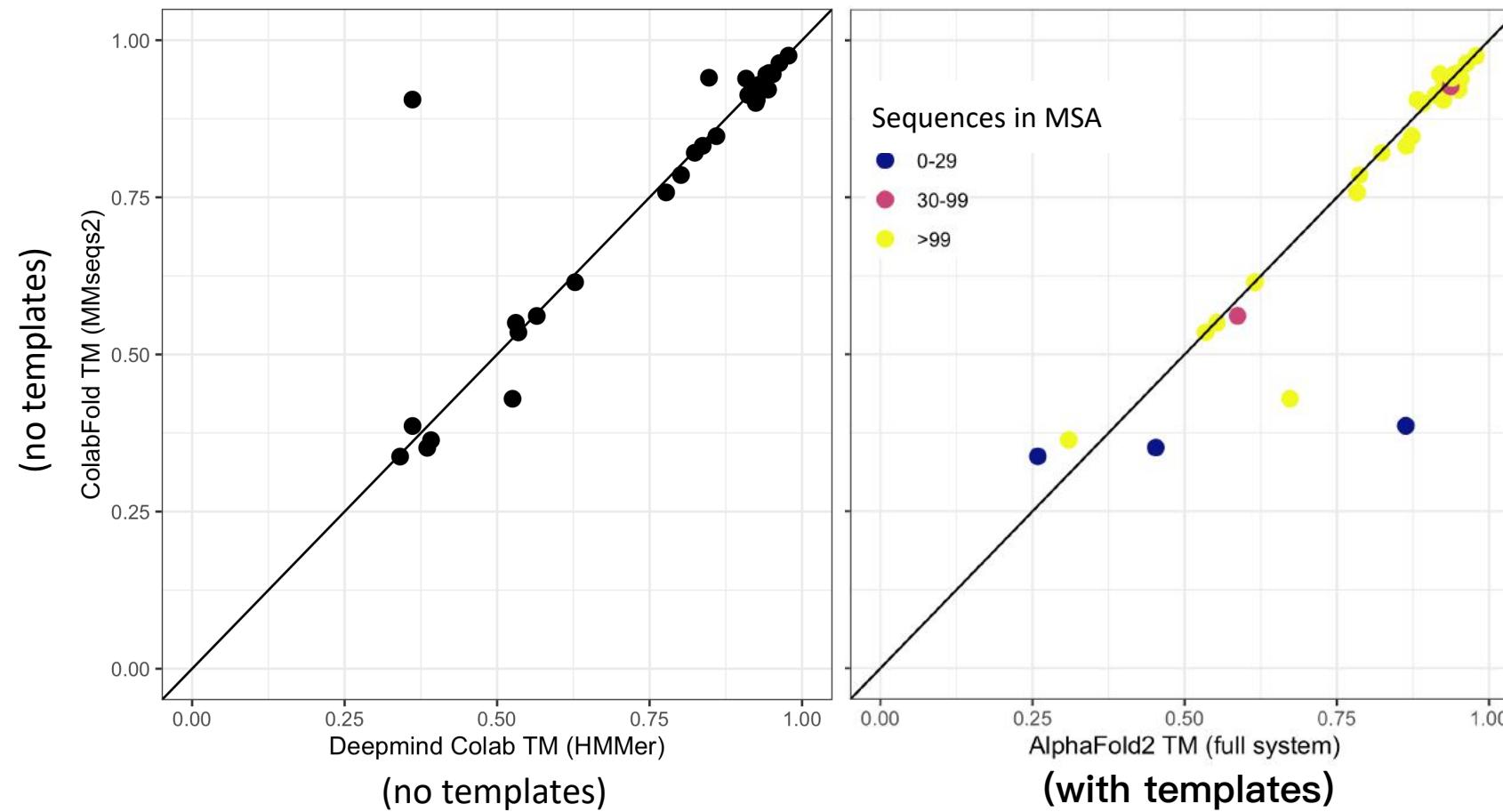
ColabFold 本身的优点

- 使用极度方便，打开网页即可使用，不需要配置资源
- 支持很多功能：调整recycling；选择是否需要模板、AMBER优化；选择更快的序列比对算法MMseq2；建模蛋白质异源复合物(hetero oligomer)；建模蛋白质多聚体(homo oligomer)
- Jupyter Notebook形式，运行方便
- 优秀的可视化方法

ColabFold 本身的缺点

- Colab现在已经限制GPU资源，没有GPU不能运行AlphaFold
- ColabFold本身不支持高通量结构预测：MMseq2在远程服务器运行，Colab仍然使用串行流程

Colab 版本：预测结果可达接近精度，但速度更快

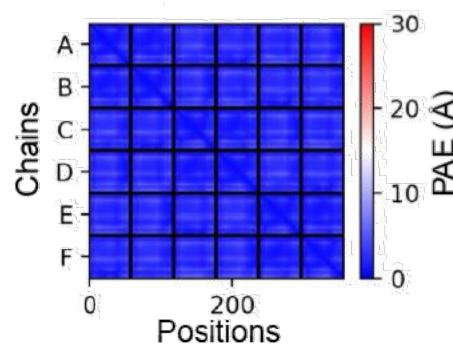
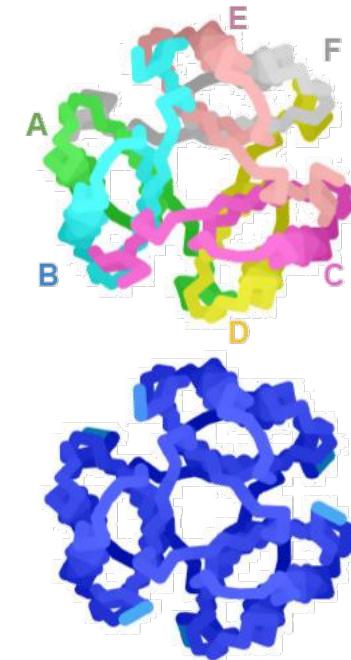


MMseq2: 单核CPU只需4分钟即可生成MSA结果

Colab 版本：预测蛋白质复合体

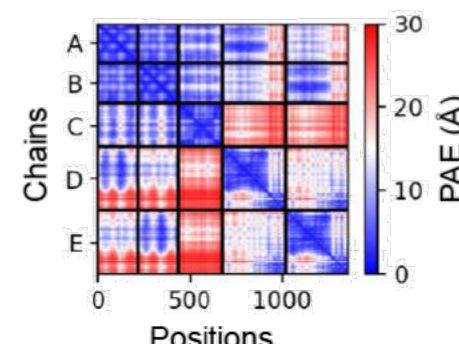
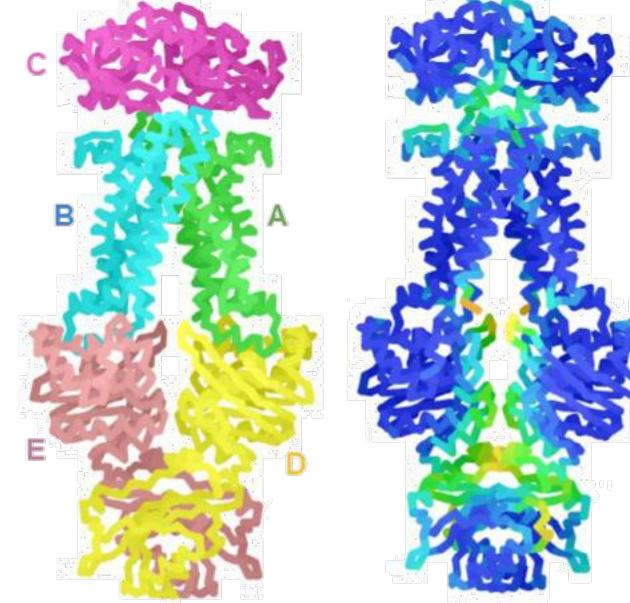
同源多聚体

A - Homo-oligomer (6)



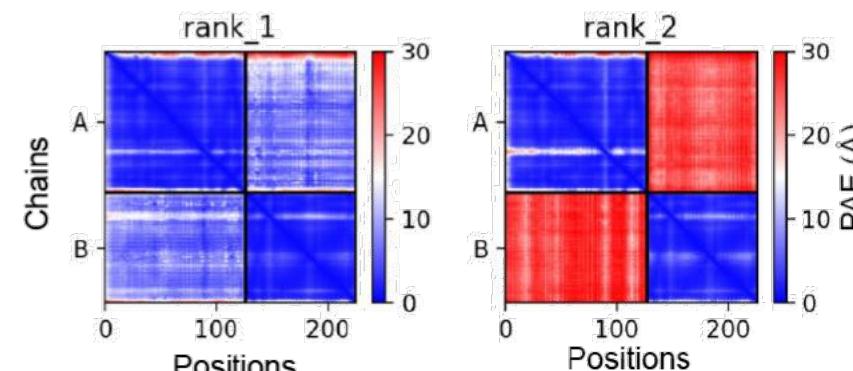
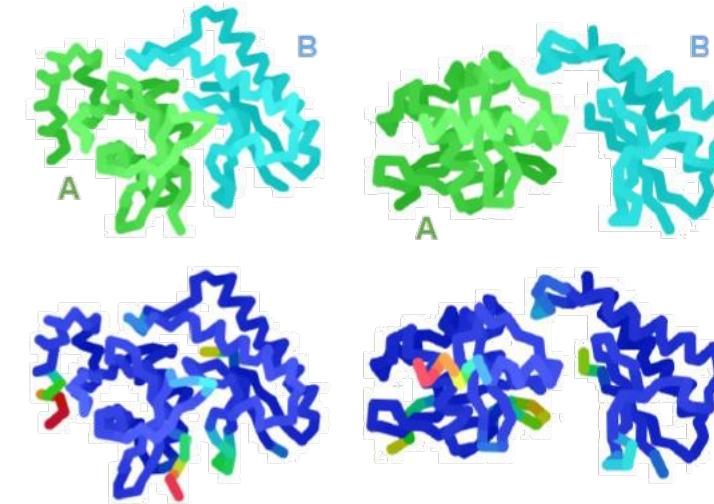
异源多聚体

B - Homo/hetero-oligomer (2:1:2)



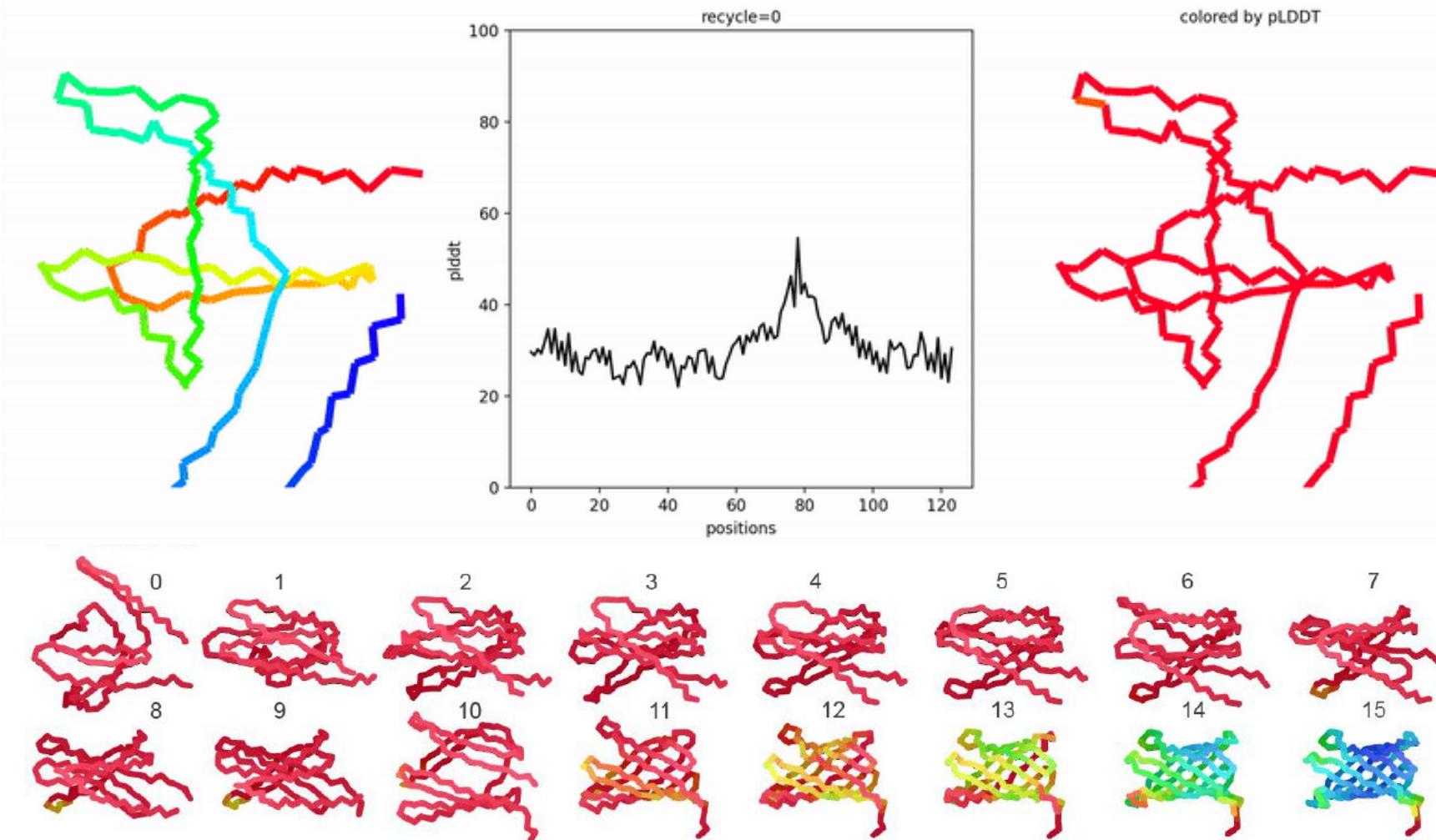
异源二聚体

C - Hetero-dimer (1:1)



PAE能够显示多聚体的预测准确度

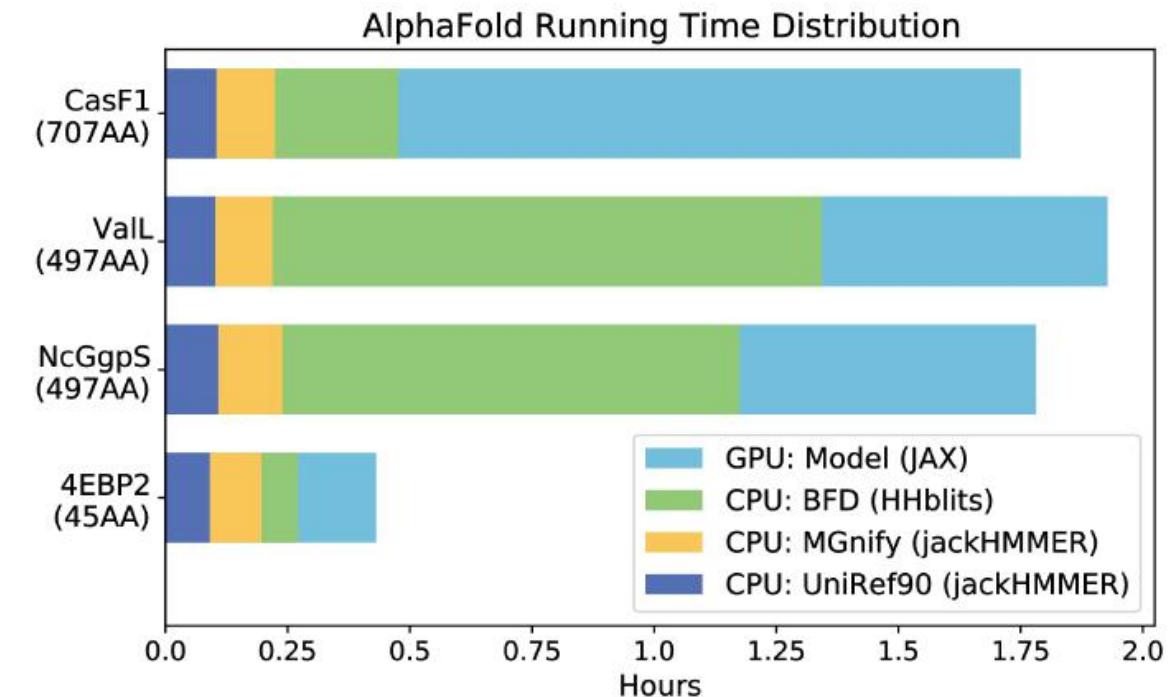
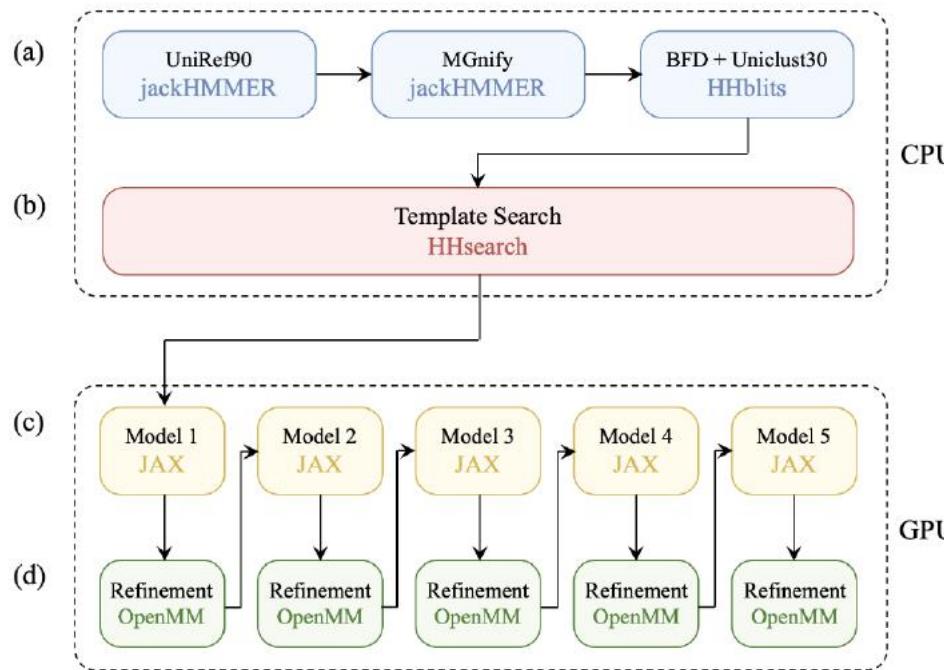
Colab 版本：更多轮的Recycling迭代



更多轮的迭代能够最大程度优化蛋白质结构到最稳定的构象

AlphaFold流程：子任务串行执行

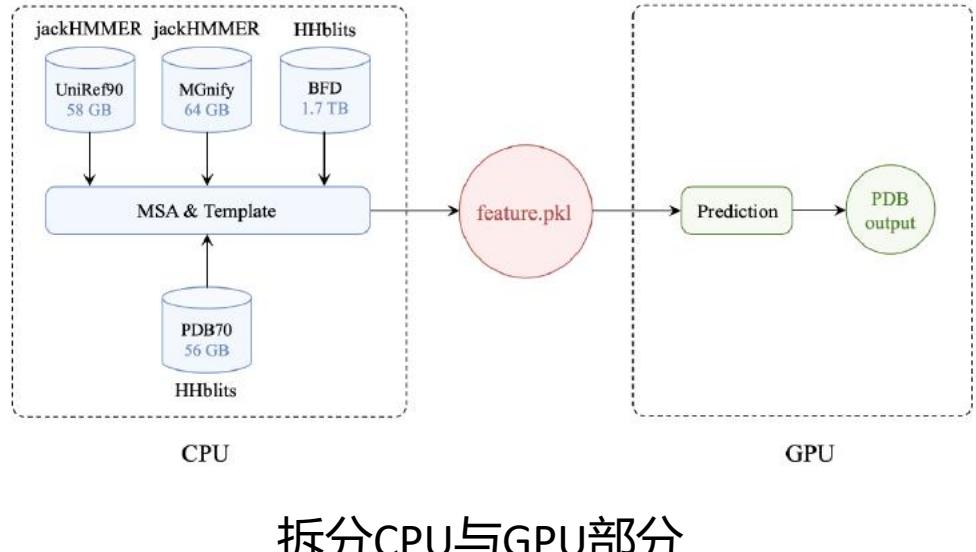
串行流程延缓预测速度，在GPU结点完成全部计算浪费GPU资源



AlphaFold结构预测采用串行流程

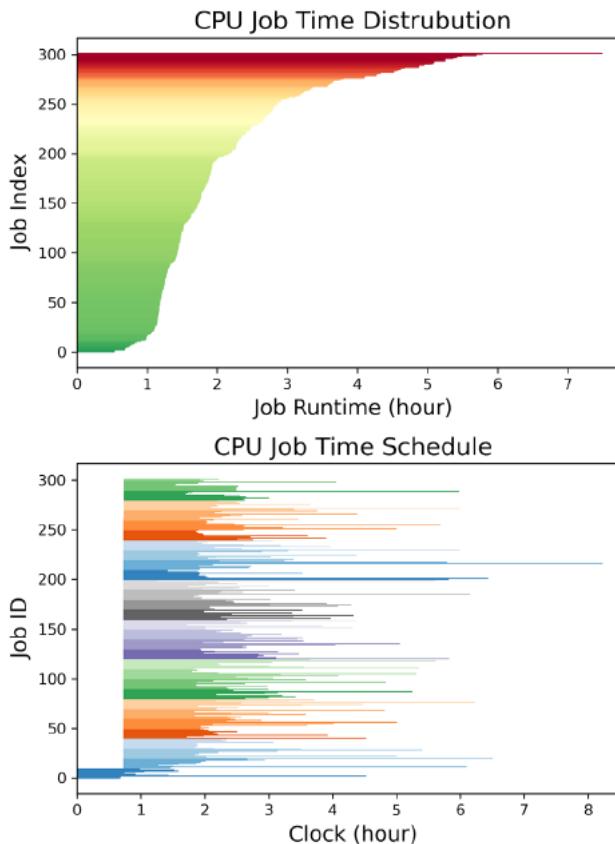
大部分预测所用时间浪费在CPU计算上
CPU每个蛋白算1~2小时， GPU一个模型5~10分钟

ParallelFold流程：CPU与GPU计算分离

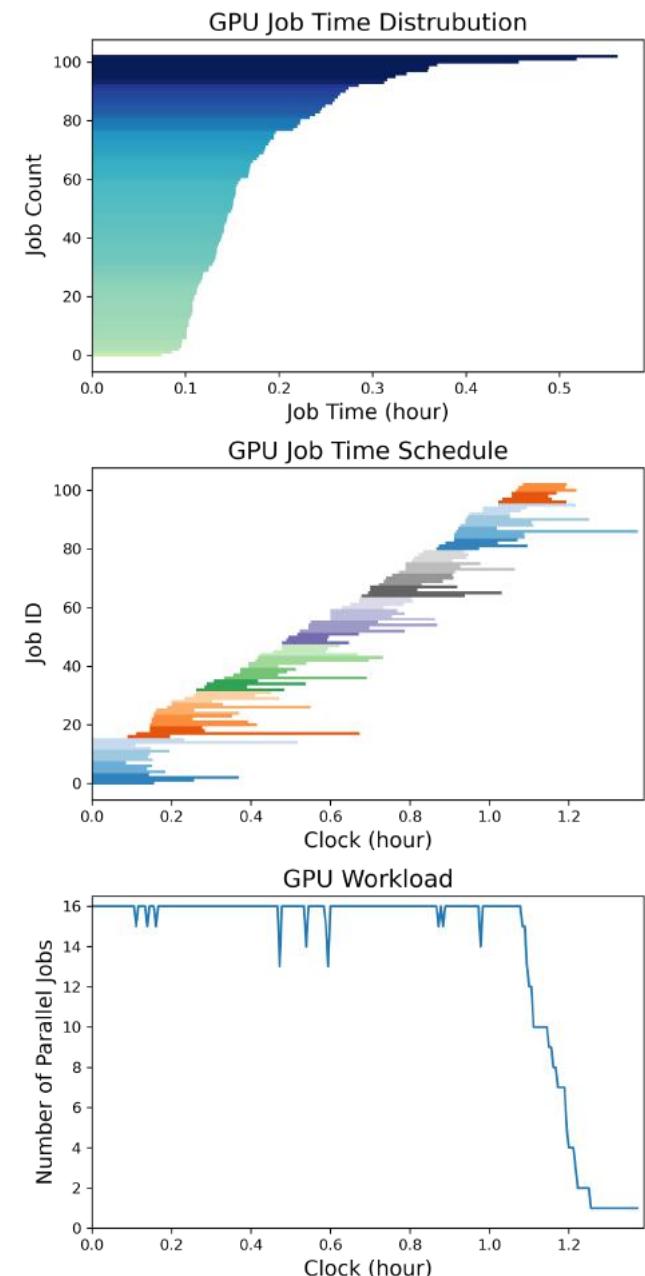


分拆CPU与GPU部分的计算

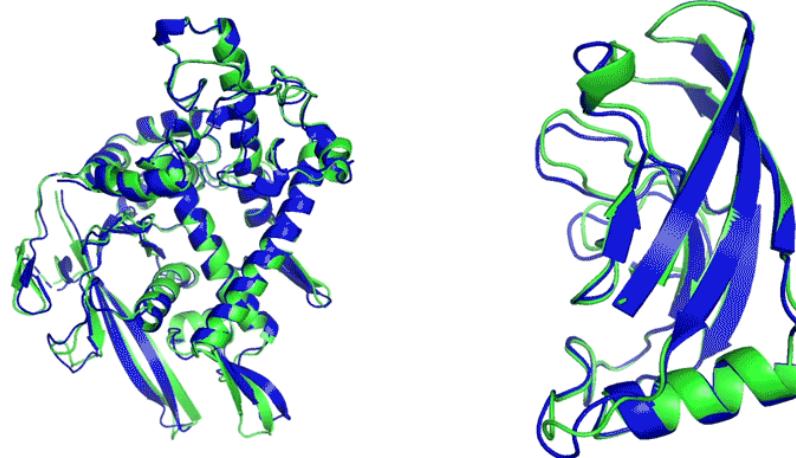
- CPU负责做MSA和模板搜索
- GPU负责神经网络部分



8小时完成300个CPU部分任务
任务平均CPU计算时间约2小时



1小时完成100个GPU部分任务
任务平均GPU计算时间约10分钟



T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

T1049 / 6y4f
93.3 GDT
(adhesin tip)

- Experimental result
- Computational prediction

谢谢聆听

Bozitao Zhong
2021/09/15

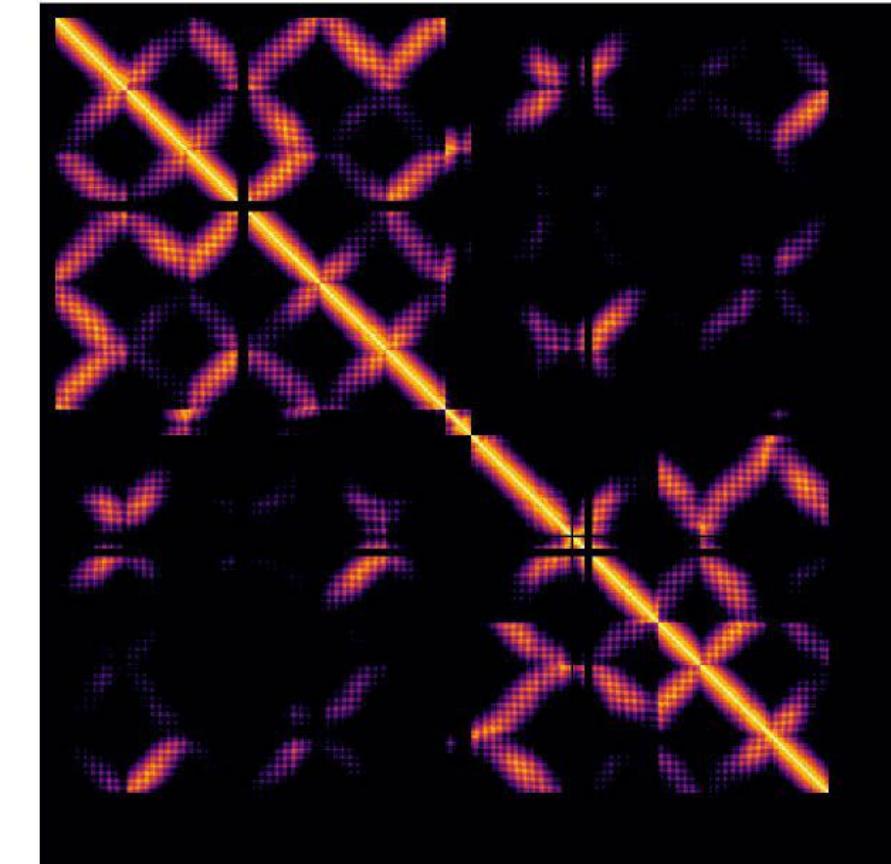
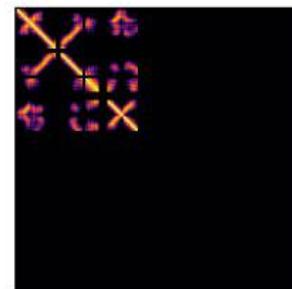


欢迎加入
AlphaFold 交大讨论群

整体架构的精彩之一： 模型输入——Templates Embedding

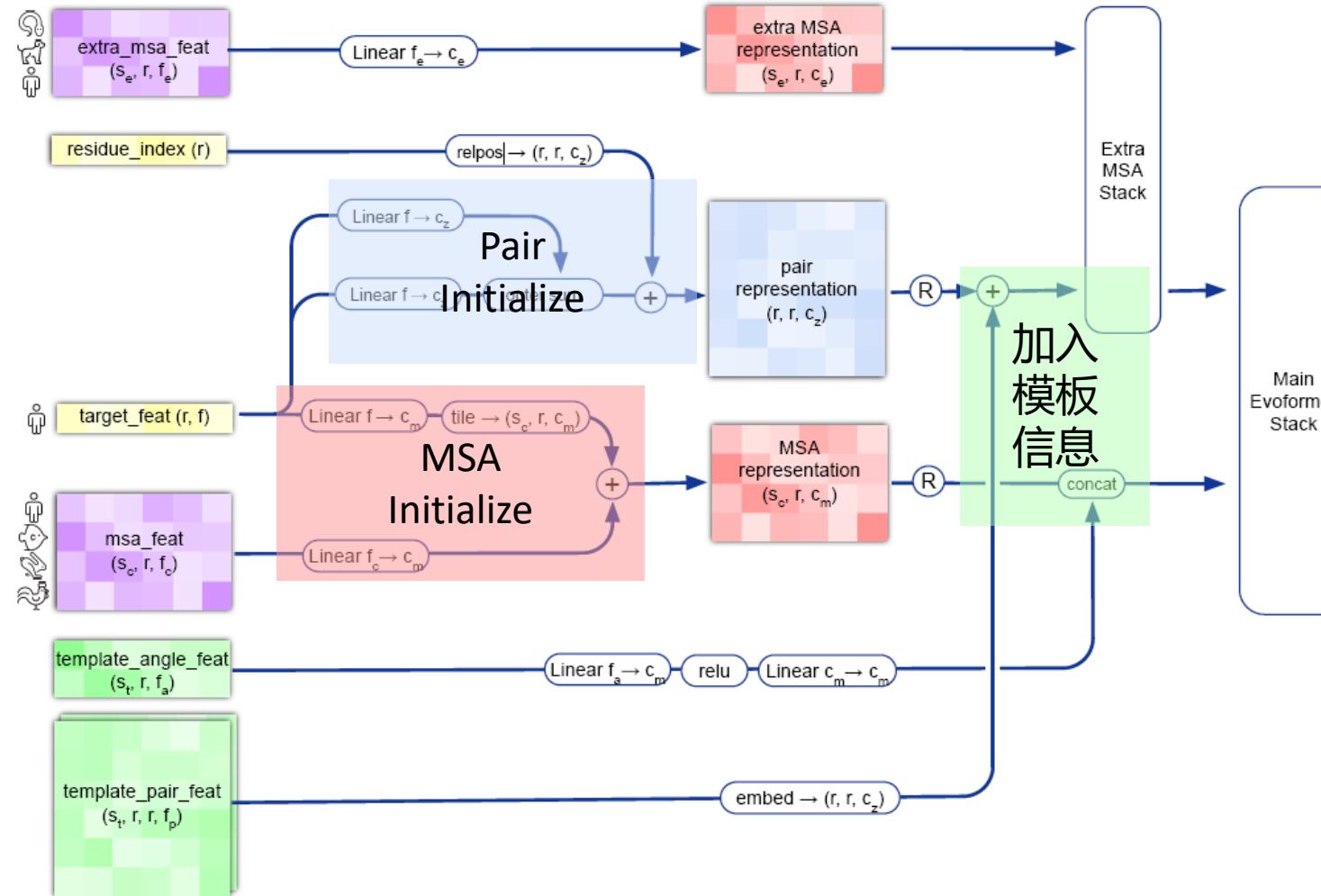
- 4 templates used (from PDB70 clusters, searched with HHsearch^{1,2})
- Input features are sequences, side chains, and distograms
- Templates are processed in the same way as the residue-residue representation

Partial
template:



整体架构的精彩之一：

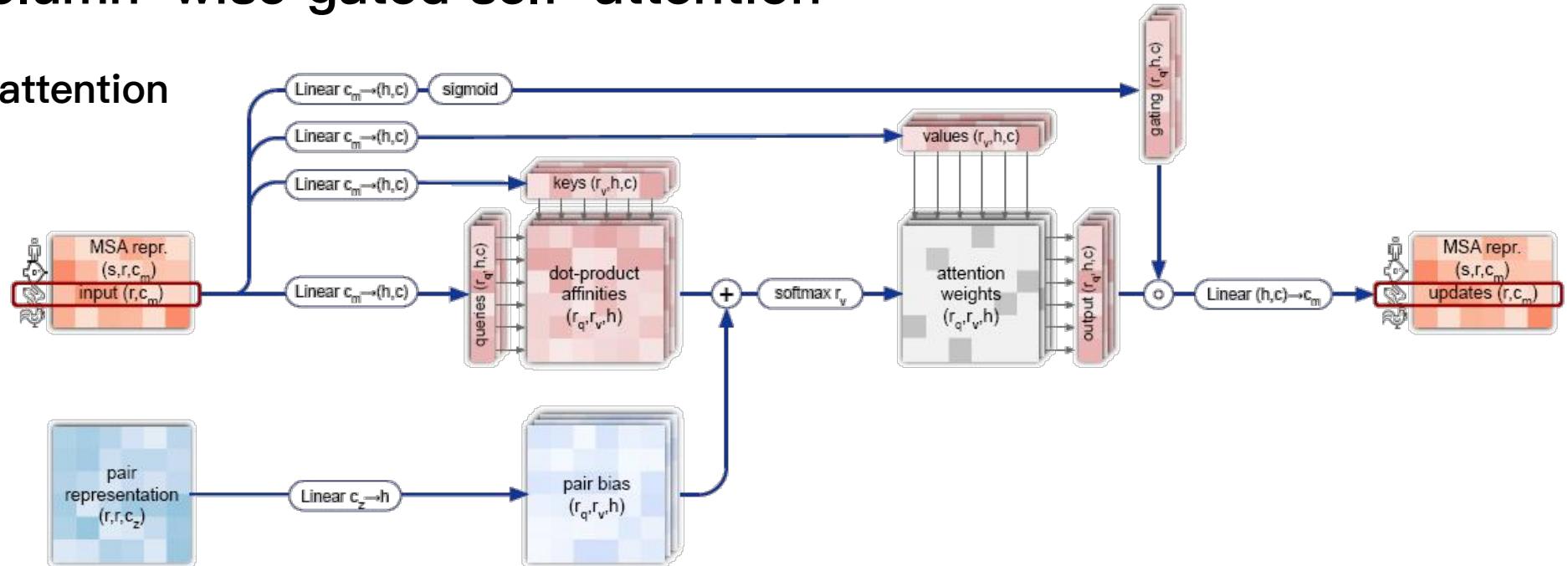
模型输入初始化细节



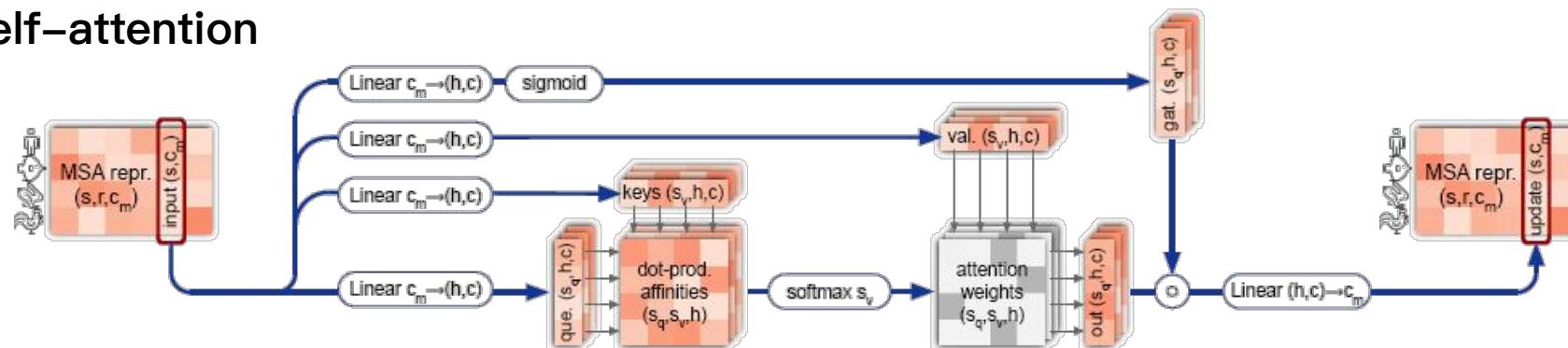
整体架构的精彩之三：

MSA row/column-wise gated self-attention

Row-wise self-attention

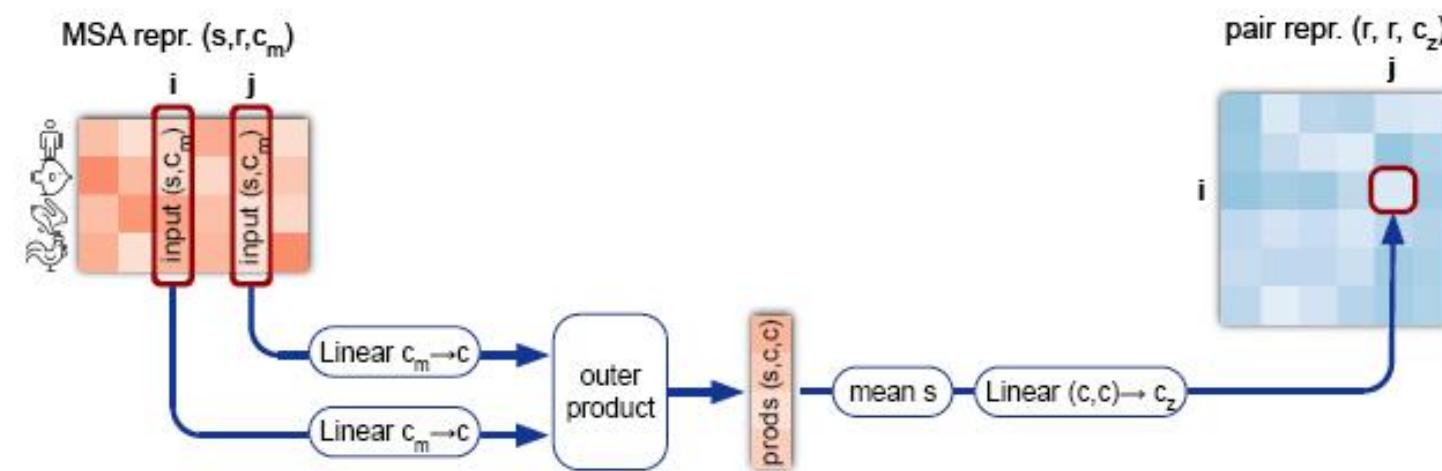


Column-wise self-attention



整体架构的精彩之三：

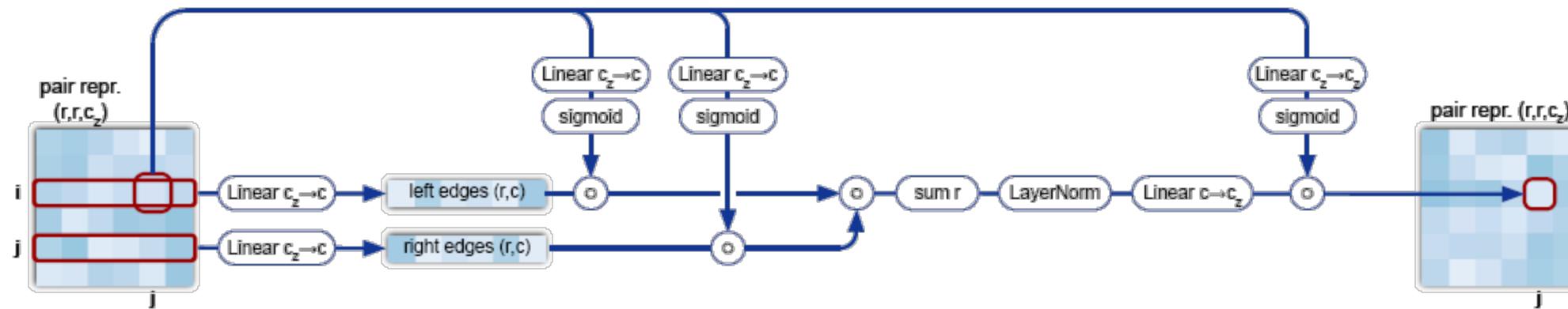
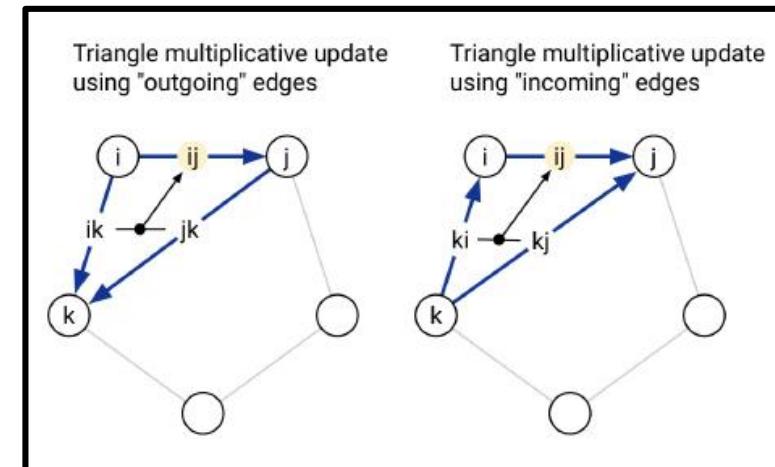
Outer product mean



使用self-attention更新后的MSA的信息来更新Pair端的信息

整体架构的精彩之三：

Triangular multiplicative update (独创)



Algorithm 11 Triangular multiplicative update using “outgoing” edges

```
def TriangleMultiplicationOutgoing({zij}, c = 128) :
    1: zij ← LayerNorm(zij)
    2: aij, bij = sigmoid (Linear(zij)) ⊙ Linear(zij)
    3: gij = sigmoid (Linear(zij))
    4: ūij = gij ⊙ Linear(LayerNorm(∑k aik ⊙ bjk))
    5: return {ūij}
```

$$\begin{aligned} a_{ij}, b_{ij} &\in \mathbb{R}^c \\ g_{ij} &\in \mathbb{R}^{c_z} \\ \tilde{z}_{ij} &\in \mathbb{R}^{c_z} \end{aligned}$$

Algorithm 12 Triangular multiplicative update using “incoming” edges

```
def TriangleMultiplicationIncoming({zij}, c = 128) :
    1: zij ← LayerNorm(zij)
    2: aij, bij = sigmoid (Linear(zij)) ⊙ Linear(zij)
    3: gij = sigmoid (Linear(zij))
    4: ūij = gij ⊙ Linear(LayerNorm(∑k aki ⊙ bkj))
    5: return {ūij}
```

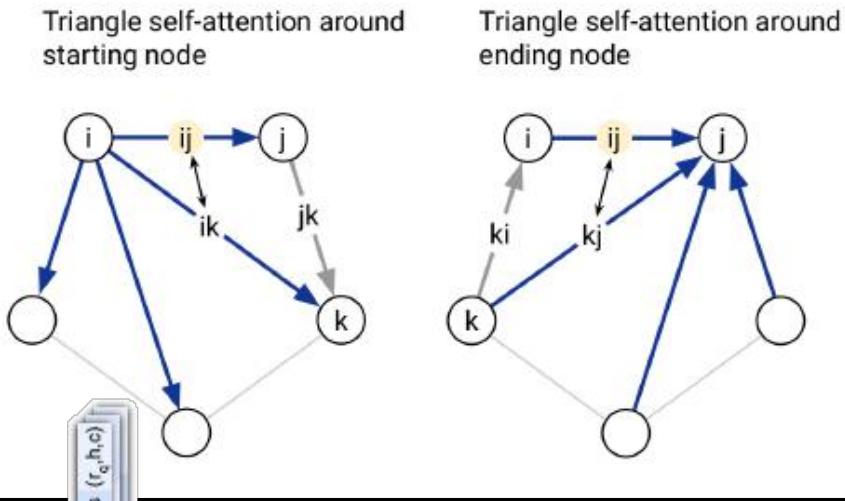
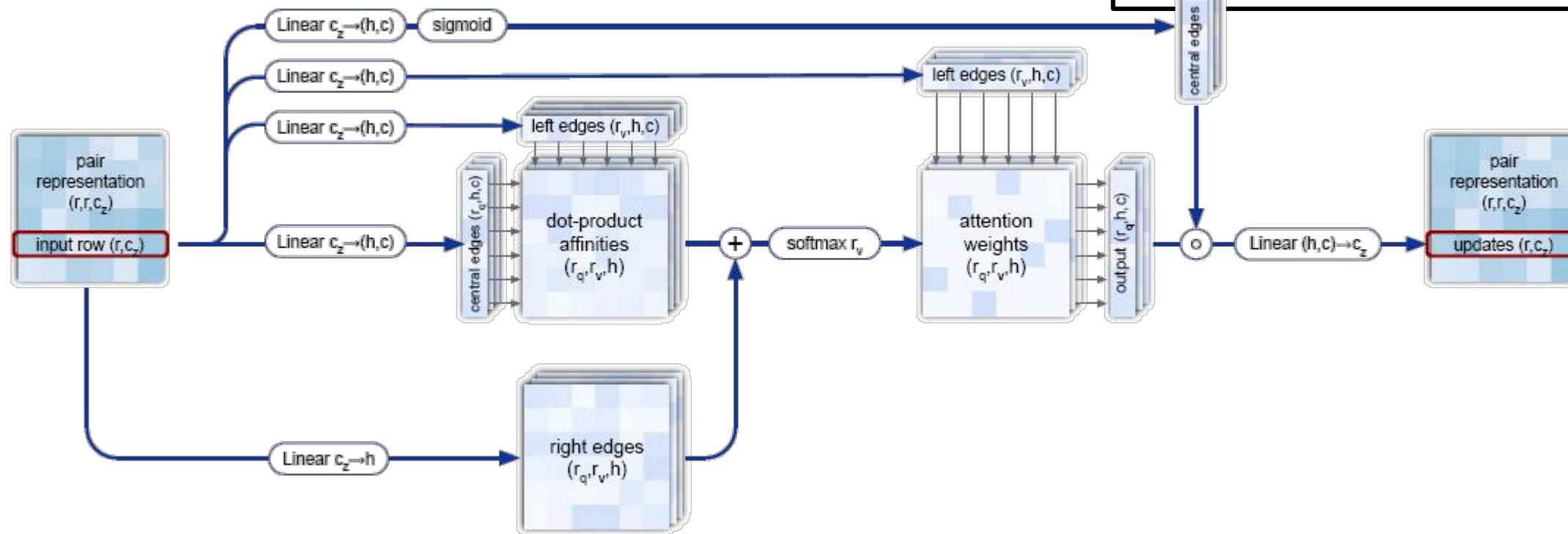
$$\begin{aligned} a_{ij}, b_{ij} &\in \mathbb{R}^c \\ g_{ij} &\in \mathbb{R}^{c_z} \\ \tilde{z}_{ij} &\in \mathbb{R}^{c_z} \end{aligned}$$

用三角形中两个边的相关性推第三条边的相关性

⊗: Hadamard乘积，表示对应位置元素相乘

整体架构的精彩之三：

Triangular self-attention (独创)



用从某一点出发的每个边来推某一个边，再用形成的三角形的第三条边判断是否接受

什么是 Equivariant? (计算机角度)

等变性 equivariant

- 通俗解释：对于一个函数，对其输入施加的变换会同样反应在输出上，那么这个函数就对该变换具有等变性
- 更严谨些：对于一个函数/特征 f 以及一个变换 g ，如果我们有： $f(g(x))=g(f(x))$ 则称 f 对变换 g 有等变性。举一个例子，假设我们的变换 g 是将图像向右平移一段距离，我们的函数 f 是检测一个人脸的位置（比如说输出坐标）， $f(g(x))$ 就是先将图片像右移，接着我们在新图较之原图偏右的位置检测到人脸； $g(f(x))$ 则是我们先检测到人脸，然后再将人脸往右移一点。这二者的输出是一样的，与我们施加变换的顺序无关。

不变性 invariant

- 通俗解释：对于一个函数，对其输入施加的变换不会影响到输出，那么这个函数就对该变换具有不变性。
- 更严谨些：假设我们的输入为 x ，函数为 f , 此时我们先对输入做变换 g : $g(x)=x'$ ，此时若有： $f(x)=f(x')=f(g(x))$ 则称 f 对变换 g 具有不变性。举个例子我们的函数是检测图像中是否有红色，此时如果我们的变换是旋转/平移，那么这些变换都不会对函数结果有任何影响，就可以说该函数对该变换具有不变性。

什么是 Equivariant? (数学角度)

在数学中，一个等变映射 (equivariant map) 是两个集合之间与群作用交换的一个函数。

具体地，设 G 是一个群， X 与 Y 是两个关联的 G -集合

一个函数 $f : X \rightarrow Y$ 称为等变，如果 $f(g \cdot x) = g \cdot f(x)$ 对所有 $g \in G$ 与 $x \in X$ 成立。

等变映射是 G -集合范畴（对一个取定的 G ）中的同态。从而它们也称为 G -映射或 G -同态。

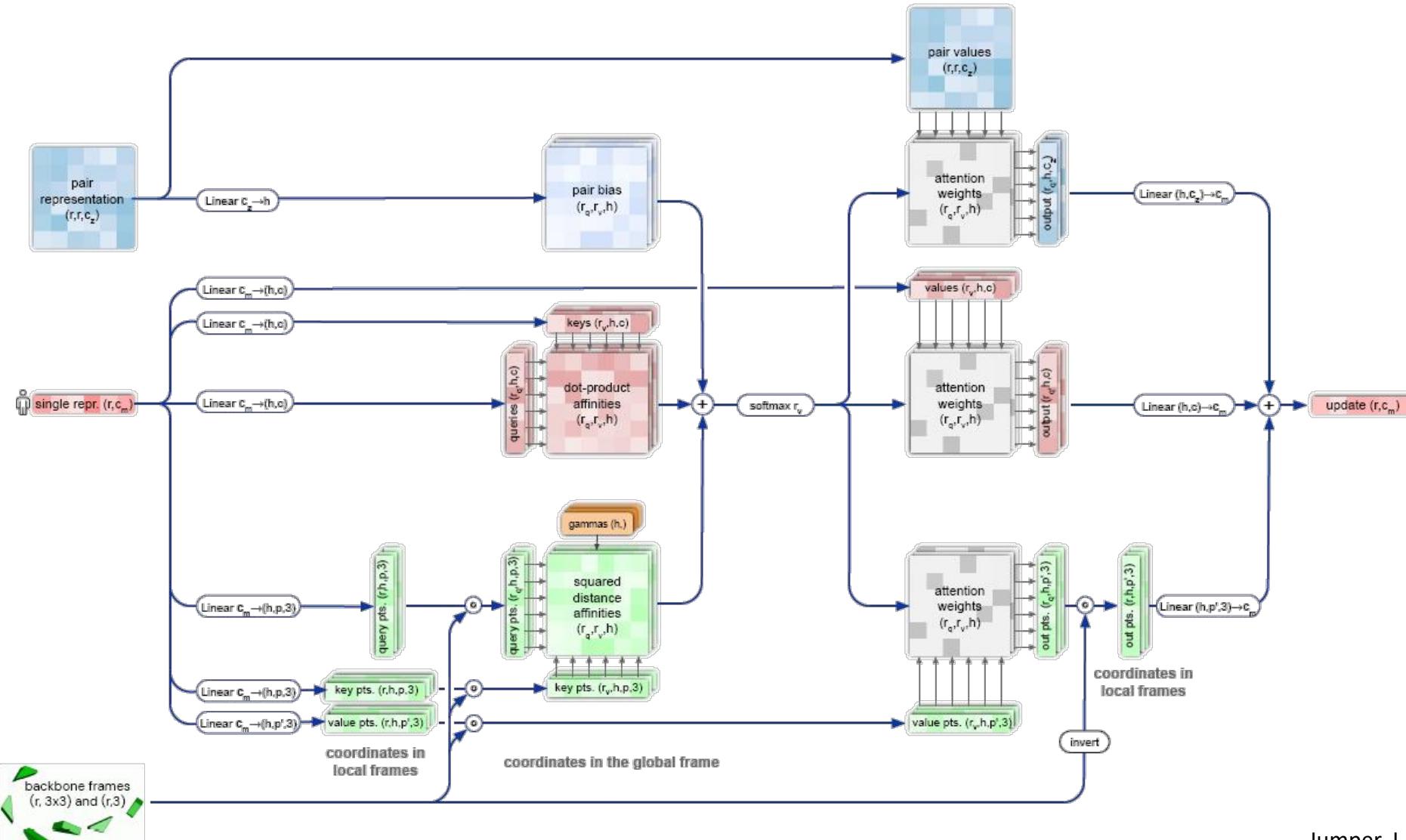
G -集合的同构就是等变双射。

$$\begin{array}{ccc} X & \xrightarrow{g \cdot} & X \\ f \downarrow & & \downarrow f \\ Y & \xrightarrow{g \cdot} & Y \end{array}$$

整体架构的精彩之四：

Structure Module的核心架构——Invariant point attention (IPA)

Invariant Point Attention (IPA) is a form of attention that acts on a set of frames (parametrized as Euclidean transforms) and is invariant under global Euclidean transformations on said frames.



整体架构的精彩之四：

Residue Gas的更多细节

将一个残基表示为 $T_i := (R_i, \vec{t}_i)$ 表示从local frame到global frame的欧几里得变换
也就是说，它将本地坐标中的一个位置转换为全局坐标中的一个位置

$$\begin{aligned} \text{也就是满足 } \vec{x}_{\text{global}} &= T_i \circ \vec{x}_{\text{local}} \\ &= R_i \vec{x}_{\text{local}} + \vec{t}_i . \end{aligned}$$

两个欧几里得变换的组合被表示为 $T_{\text{result}} = T_1 \circ T_2$

$$\begin{aligned} (R_{\text{result}}, \vec{t}_{\text{result}}) &= (R_1, \vec{t}_1) \circ (R_2, \vec{t}_2) \\ &= (R_1 R_2, R_1 \vec{t}_2 + \vec{t}_1) \end{aligned}$$

整体架构的精彩之四：

推断——Residue Gas与SE(3)特殊欧几里得群

将一个残基表示为 $T_i := (R_i, \vec{t}_i)$ 满足 $\vec{x}_{\text{global}} = T_i \circ \vec{x}_{\text{local}}$
 $= R_i \vec{x}_{\text{local}} + \vec{t}_i$.

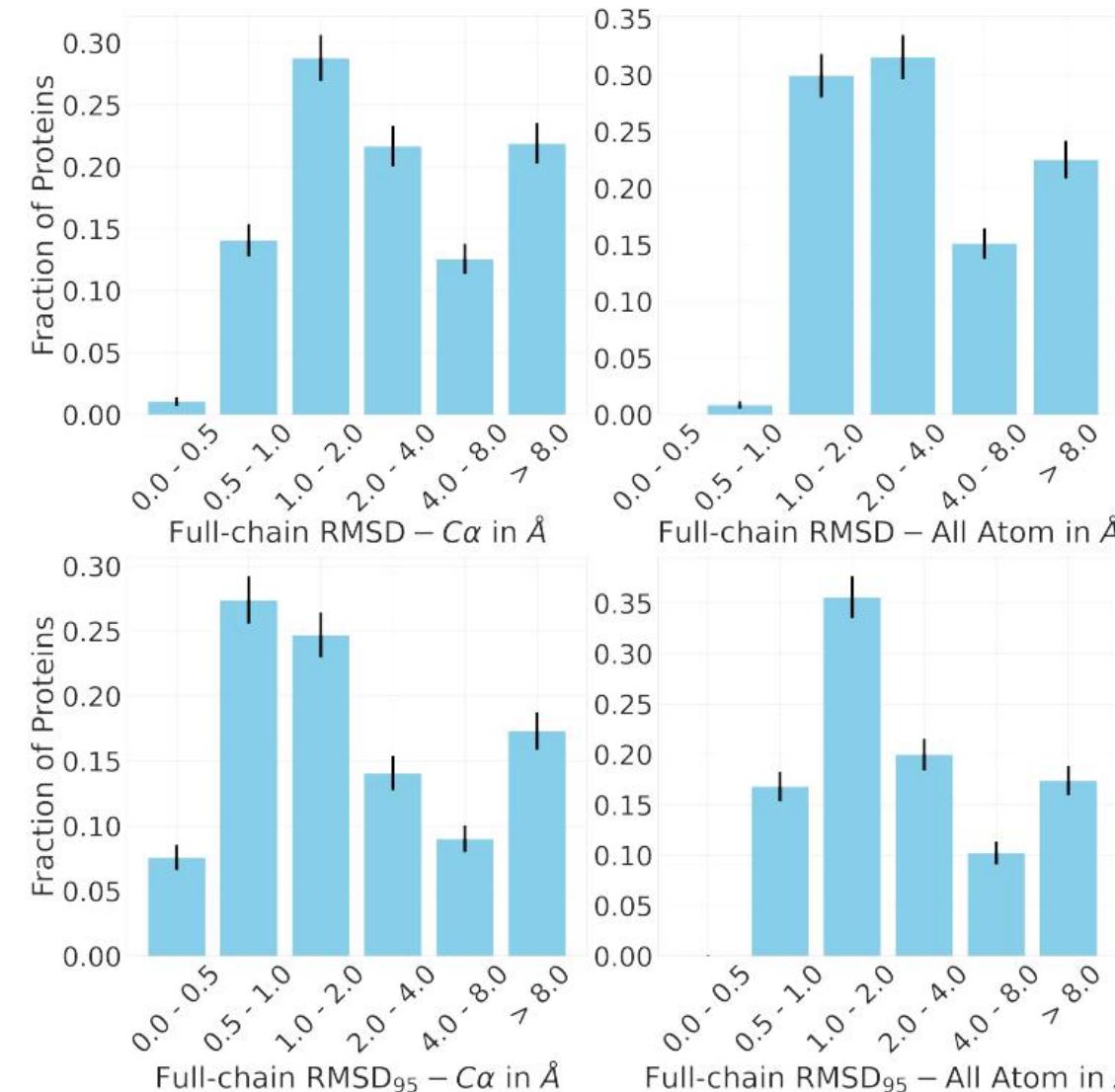
特殊正交群是有效的旋转矩阵的集合：

$$SO(3) = \{\mathbf{C} \in R^{3 \times 3} | \mathbf{C}\mathbf{C}^T = \mathbf{1}, \det \mathbf{C} = 1\}$$

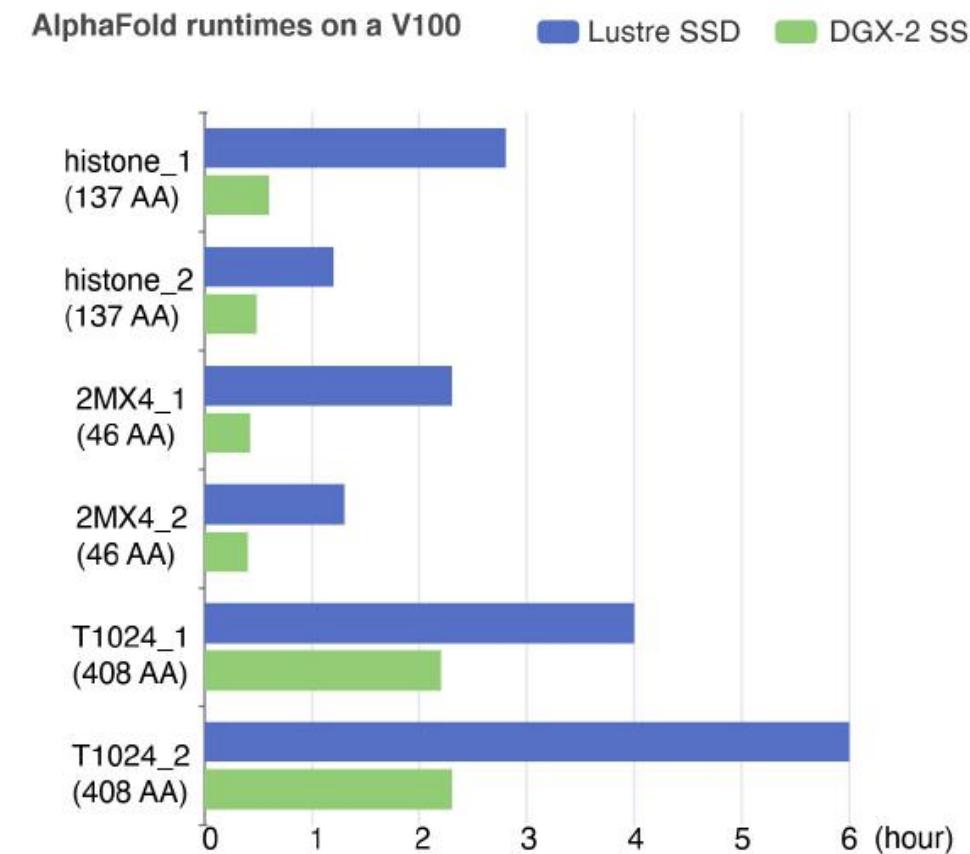
用于表示姿态的是特殊欧几里得群：

$$SE(3) = \{\mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} \in R^{4 \times 4} | \mathbf{C} \in SO(3), \mathbf{r} \in \mathbf{R}^3\}$$

结果：all-atom and backbone RMSD



ParallelFold优化：高速存储加速MSA过程



储存在计算节点DGX2上的数据能让MSA计算过程更快