AlphaFold 3

`/fs00/software/alphafold/3.0.0/AlphaFold-v3.0.0.sif`

0. ????

alphafold3 Google DeepMind json

- data pipeline:

  (multi-sequence alginment, msa) (templates)

  json
- inference: msa templates json mmcif

  confidence score

github apptainer alphafold3

——

term of use

Google DeepMind

1. ????

```
path_to_af3db="/fsb/data/alphafold/3"
path_to_af3container="/fs00/software/alphafold/3.0.0/AlphaFold-v3.0.0.sif "

io_dir=</path/to/your/input/and/output>
weights_dir=<path/to/directory/containing/your/af3-weights>

RUN_ALPHAFOLD="apptainer run --nv --bind
```

```
${path_to_af3db}:/databases,${io_dir}:/host_iopath,${weights_dir}:/af3_weights
${path_to_af3container} python run_alphafold.py"
```

其中 `${path_to_af3db}` 是集群上 alphafold3数据库地址， `${io_dir}` 是输入文件所在的、及希望结果输出的目录， `${weights_dir}` 是自己申请的 alphafold3权重文件所在目录。 `run_alphafold.py` 是 官方 github库 提供的接口文件。本页面后续出现的指令均默认这些变量存在。

输出帮助信息 （包括参数说明 ）

```
${RUN_ALPHAFOLD} --help
```

将输入文件 `input.json` 放在 `${io_dir}` 目录下，如下指令可使用 alphafold3预测此文件描述体系的结构

```
${RUN_ALPHAFOLD} --db_dir=/databases/ --model_dir=/af3_weights/ \
  --json_path=/host_iopath/input.json --output_dir=/host_iopath/
```

程序将根据 `input.json` 中的 name字段值，在 `${io_dir}` 下创建同名子目录， data pipeline结束时补全的 json 文件、预测结构文件和 confidence score都会被保存在这个子目录中。运行时 log信息通过 stderr输出 （这是 python logging模块的行为 ）。关于更详细的输入输出文件说明，请参考官方 input documentation和 output documentation。

2. ????data pipeline?inference

如前所述，程序运行主要分为 data pipeline和 inference两步。其中， data pipeline只使用 cpu 计算且耗时很长，如果在 gpu队列上运行会导致 gpu资源长期空置，可能导致高额费用和 gpu资源浪费。因此， 推荐先在 **cpu**队列上运行 **data pipeline**，得到中间结果 **(**也就是补全的 **json)** 退出，然后再到 **gpu** 队列上进行结构预测 。

为满足上述需求，可以用 bool型命令行参数 `--run_data_pipeline` 和 `--run_inference` 控制模型只运行其中一步，两参数默认值均为 True.

分步运行指令示例 ：

- data pipeline (cpu job)

```
${RUN_ALPHAFOLD} --db_dir=/databases/ --model_dir=/af3_weights/ \
  --json_path=/host_iopath/input.json --output_dir=/host_iopath/ \
  --run_inference=False
```

- inference (gpu job)

```
${RUN_ALPHAFOLD} --db_dir=/databases/ --model_dir=/af3_weights/ \
  --json_path=/host_iopath/input.json --output_dir=/host_iopath/ \
  --run_data_pipeline=False
```

另外，出于研究或加速计算的目的，用户可能希望使用自行准备的 msa或 templates，因而需要跳过 alphafold3搜索步骤，此时需要手动将 msa和 templates写进输入 json文件，然后带 `--run_data_pipeline=False` 参数运行模型，进行结构预测。关于如何将 msa和 templates写入 json 文件，请自行参考 DeepMind-input documentation。注意， msa和 templates会以文本形式写入 json 中，因此写入了 msa和 templates的 json文件大小会达到数十到百 Mb量级。

3. ?????????

除去以上参数外，容器还可接受很多其他参数，可以用 `${RUN_ALPHAFOLD} --help` 查看，或是在 `run_alphafold.py` 中查看。我认为有 3点值得注意：

- `--input_dir` 和 `--json_path` 都可以指定输入文件，使用时只应指定其一。当指定 input_dir 时，模型会从 input_dir中寻找一个 json格式文件，当作输入文件
- `--jackhmmer_n_cpu` 和 `--nhmmer_n_cpu` 指定生信搜索中使用 cpu cores数目的上限
- 机器学习程序通常允许设置 random seed来保证运行的可重复性， alphafold3的 random seed 不能以命令行参数指定，而是在 input.json中

4. ?????????????

data pipeline对设备要求不高， inference则可能对 gpu设备有些要求。目前测试了两例：

- 2PV7，蛋白，双链 homomer，单链长度 298
- 1AKE，蛋白，双链 homomer，单链长度 214

在队列 83a100ib和 734090d上测试结果显示: 以上体系在单张 A100卡 (memory 40 G) 或单张 4090d (memory 24 G) 上均可完成 inference. 在队列 722080tiib和 72rtxib上无法成功运行 . 其他 GPU 队列未做测试。注意，官方 performance documentation文档指出官方仅测试过 alphafold3在 A100(80G), A100(40 G)和 H100设备上推理的准确性，因此使用如 4090等其他设备做推理时，请自行校验结果准确性。

测试中 data pipeline和 inference计算所需时间如下:

- 使用 8 cpu cores对一条长约 300的蛋白序列进行 msa和 templates搜索耗时为 1.5-2 h; 其中，搜出 msa后，根据部分匹配片段信息挑选出 templates耗时为 10 s量级，相对于总时间可以忽略
- 对每个随机种子， alphafold3会预测 5个结构并按置信度打分挑选最好的作为最终结果，此过程耗时 100 s左右; inference步总耗时与设置随机种子数有关

更多 performance信息可参考官方 performance documentation。

5. ?????????

比较 2PV7预测结构与实验结构，全原子 RMSD=4.410 (单位 : Angstrom，下同)。为衡量单链水平结构误差，计算预测 A:B链 -实验 A:B链间 RMSD， 4个 RMSD值分别为 3.012, 2.759, 2.971, 2.740. 此单链上， alphafold3预测准确度低于 alphafold2. 为衡量 2个 subunits间相对位置误差，计算 DockQ 分数为 0.499>0.23，预测结构 docking成功。

1AKE双链结构预测有较大错误，预测结构与 1AKE实验结构 RMSD=18.176，与 4AKE实验结构 RMSD=26.791 ，该结果远不如 alphafold2-multimer。subunits间 DockQ=0.019<<0.23 ，说明结构相对位置也预测错误。尚不清楚是 Alphafold3本身原因，还是部署或使用不当导致。使用 4090d和 A100推理的结果无明显差异。

> “ dockQ (Mirabello & Wallner, 2024, Bioinformatics) 打分用于衡量模型对 subunits 间位置关系预测准确度。此分数取值范围是 0-1，通常 <0.23认为错误， >0.8认为准确。

Revision #15
Created 2024-11-18 19:13:54 CST by LadderOperator
Updated 2025-03-13 23:54:58 CST by Yao Ge