**中图分类号： 单位代号：10280**

**密 级： 学 号：**

**硕 士 学 位 论 文**

**SHANGHAI UNIVERSITY**

**MASTER’S DISSERTATION**

|  |  |
| --- | --- |
| **题**  **目** |  |

**作 者 王文松**

**学科专业 凝聚态物理**

**导 师** 吴伟

**完成日期** 2021年4月

姓 名： 学号：

论文题目：

上海大学

本论文经答辩委员会全体委员审查,确认符合上海大学硕/博士学位论文质量要求。

答辩委员会签名：

主任：

委员：

导 师：

答辩日期：

姓 名： 学号：

论文题目：

**原 创 性 声 明**

本人声明：所呈交的论文是本人在导师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名： 日 期：

**本论文使用授权说明**

本人完全了解上海大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（**保密的论文在解密后应遵守此规定**）

签 名： 导师签名： 日期：

上海大学理学硕士学位论文

**关于提高蛋白质结构精修**

**的探索研究**

姓 名：王文松

导 师：吴伟

学科专业：凝聚态物理

上海大学理学院

2021年4月

A Dissertation Submitted to Shanghai University for the Degree of Master in Science

**Exploration on improving**

**protein structure refinement**

MA Candidate：Wensong Wang

Supervisor：Wei Wu

Major：Optics

**College of Science, Shanghai University**

**April, 2021**

摘 要

在蛋白质结构三维模型重建领域冷冻电镜(cryo-EM)技术正在迅速成为X射线衍射晶体学的主要对手，尤其是一些很难或者不可能结晶的大分子结构。尽管近年来科技的进步有力的促进了蛋白质结构三维重建模型的分辨率，但是跟X射线衍射结晶技术相比，冷冻电镜测得的蛋白质结构大部分仍然还处于低分辨率的范围。

目前对于蛋白质结构精修的方法主要有经典精修和量子精修两种，经典精修一般是基于经验参数化约束或分子动力学对蛋白质结构精修，量子精修是利用基于量子化学方法衍生出来的化学约束（比如，能量或能量梯度）用于蛋白质结构精修，但都有一定的局限性。经典精修有计算速度快的优势，冷冻电镜蛋白质结构精修的一种很有效的方法就是参考蛋白质结构的电子密度图（map）,但是很多被参考的map本身就存在问题。在最新的量子精修项目Q|R中已被证明量子精修相对于经典精修有可能更准确而且可以更多很容易应用于新的分子，如药物或新的辅助因子，但是该方法对于初始蛋白质结构的要求过于苛刻，比如蛋白质结构分辨率不能太低，而且分子结构不能太大，精修结果有可能停留在局部最优而不是全局最优，而且计算资源消耗巨大。

众所周知蛋白质由氨基酸残基组成，本文首先基于开源软件PHENIX进行二次开发，对低分辨率的冷冻电镜蛋白质结构寻找序列相同序列的高分辨率蛋白质链并标注提取生成用于优化初始结构的约束模型，然后得到经过经典精修或结构优化过的初始模型然后进行经典精修和量子精修，这样经典精修与量子精修相结合的方式既可以解决经典计算效果不太好的瓶颈，又可以解决量子精修对初始模型要求苛刻、改善提高精修结果更易全局最优的几率，而且可以减少计算资源和计算时长的浪费，该方法不仅可以应用于冷冻电镜蛋白质精修，X射线衍射结晶等方法获得的蛋白质模型也可进行尝试。具体内容如下：

1. 基于phenix软件接口编写代码对蛋白质pdb结构进行加氢、去杂质等准备工作，然后遍历统计蛋白质数据库寻找并统计拥有相同蛋白质序列的高分辨率和低分辨率冷冻电镜蛋白质模型对，构建hierary蛋白质模型结构，然后编写相同序列蛋白质链的对齐和转换算法，根据对应关系及之前实现的标注功能提取氢键、盐桥等非共价键及氨基酸残基内与这些非共价键相关联的共价键，然后将其自动生成phil约束模型和用于pymol展示的氢键模型。
2. 在之前统计好的高低分辨率蛋白质对中选取部分冷冻电镜蛋白质结构，直接进行使用phenix.refine 和phenix.real\_space\_refine 进行经典精修，使用XTB进行量子精修，并用phenix.molprobity 和Z-score进行模型参数计算统计。
3. 将第二步中蛋白质对中的高分辨率蛋白质结构运用之前编写的代码进行约束文件自动生成，然后运用phenix.geometry\_minimization 进行结构优化，然后将优化之后的模型代替原始模型重复第二步的过程，并将相对应的结果进行比较分析发现经过优化之后的模型经过经典精修和量子精修结果都要好，而且量子精修所消耗的资源和时间都相应的减少。

**关键词：**量子精修、经典精修、蛋白质序列、冷冻电镜、X涉嫌衍射

**ABSTRACT**

The cryo-electron microscopy (CRYO-EM) technique is rapidly emerging as the main rival of X-ray diffraction crystallography in the field of three-dimensional reconstruction of protein structures, especially for large molecules that are difficult or impossible to crystallize. Although scientific and technological advances in recent years have greatly improved the resolution of three-dimensional reconstruction models of protein structures, most of the protein structures measured by cryo-electron microscopy are still in the low resolution range compared with X-ray diffraction crystallization techniques.

At present the main method for protein structure refinement are classical and quantum fine trim two, classic refinement is generally based on experience parametric constraints or molecular dynamics of protein structure refinement, quantum refinement is based on quantum chemistry method derived chemical constraints (such as energy or energy gradient) for protein structure refinement, but there are some limitations. Classical refining has the advantage of fast computing speed, and a very effective method for refining protein structures in cryo-electron microscopy is to refer to the MAP of protein structures (phenix.real\_space\_refining), but many of the referenced maps have problems in themselves. Quantum refinement projects in the latest Q | R quantum refinement has been shown in the relative to the classic refinement may be more accurate and can be more easily applied to new molecules, such as drugs or new cofactors, but this method is too harsh to the requirement of initial protein structure, such as protein structure resolution cannot be too low, cannot too big, and molecular structure refinement results are likely to remain in local optimum, rather than the global optimal, and huge computational resources consumption.

Known protein is composed of amino acid residues, PHENIX we based on open source software for secondary development, the low resolution of frozen electron microscopy (sem) protein structure for sequences of the same high resolution protein chain and getting generated annotation is used to optimize the initial structure of the constraints of the model, get the classic refinement or structure optimized initial model and quantum refinement, such classic refinement with quantum finishing a combination of both can solve the classical computational bottleneck effect is not very good, and can solve the quantum refinement to the demanding of initial model, and improve chances of finishing results are more likely to be the global optimal, This method can not only be used for protein refinement in cryo-electron microscopy, but also for protein models obtained by X-ray diffraction crystallization and other methods. The main content is as follows:

(1) Based on phenix software interface code for protein structure of PDB hydrogenation, to impurities, such as preparation, then traverse statistics protein database search and statistics have the same protein sequences of high resolution and low resolution electron microscopy (sem) protein frozen model to build hierary protein model structure, and then write the same sequence alignment of protein chain and transform algorithm, and according to the relation between before implementation of tagging function to extract the hydrogen bonds, salt bridge non covalent bond and the amino acid residues in a covalent bond is associated with the non covalent bond, Then it automatically generates the Phil constraint model and the hydrogen bond model for the PyMol demonstration.

(2) Some protein structures of cryo-electron microscopy were selected from the previously calculated high-resolution protein pairs, and classical refinement was directly performed by phenix.refine and phenix.real\_space\_refine, quantum refinement was performed by XTB, and model parameters were calculated and counted by phenix.molprobity and Z-score.

(3) The protein in the second step in the high resolution protein structure using code written before constraint file generated automatically, and then using the phenix. Geometry\_minimization structure optimization, and then repeat step 2 after optimization model instead of the original model of the process, and the comparative analysis and the corresponding results are found after optimization model through classical refinement and quantum finishing result is better, and quantum refinement consume resources and time are decreased accordingly.

**Keywords:** Quantum refinement, classical refinement, protein sequence, cryo-electron microscopy, X diffraction

**目 录**

[摘 要 V](#_Toc68881017)

[ABSTRACT VII](#_Toc68881018)

[目 录 X](#_Toc68881019)

[第一章 绪论 1](#_Toc68881020)

[1.1 课题研究的目的和意义 1](#_Toc68881021)

[1.2 蛋白质结构经典精修 3](#_Toc68881022)

[1.2.1蛋白质经典精修原理和方法 3](#_Toc68881023)

[1.2.2蛋白质经典精修方法 5](#_Toc68881024)

[参数化(精修策略) 6](#_Toc68881025)

[目标和约束 6](#_Toc68881026)

[优化方法和其他选项 7](#_Toc68881027)

[1.2.3 蛋白质经典精修优势与劣势 8](#_Toc68881028)

[1.3 蛋白质结构量子精修 9](#_Toc68881029)

[1.3.1量子精修方法及原理 9](#_Toc68881030)

[Hybrid QM/MM 9](#_Toc68881031)

[Semi-empirical 10](#_Toc68881032)

[Linear-scaling density functional theory 10](#_Toc68881033)

[1.4.3蛋白质量子精修的优势和劣 11](#_Toc68881034)

[1.4 论文的主要研究内容 11](#_Toc68881035)

[第二章 精修约束模型及参数 13](#_Toc68881036)

[2.1 蛋白质结构 13](#_Toc68881037)

[2.1.1 蛋白质分级结构 14](#_Toc68881038)

[2.1.2 蛋白质序列 14](#_Toc68881039)

[2.1.3 蛋白质分子间作用力 14](#_Toc68881040)

[2.1.4约束模型标注及提取 15](#_Toc68881041)

[第三章 蛋白质初始结构模型优化 19](#_Toc68881042)

[3.1 优化方法 19](#_Toc68881043)

[3.2 几何结构优化结果 20](#_Toc68881044)

[3.3 蛋白质优化结果验证 21](#_Toc68881045)

[第四章 优化后的初始模型进行经典精修 23](#_Toc68881046)

[4.1 运用经典精修CCTBX进行精修 23](#_Toc68881047)

[4.2 精修结果验证及分析 25](#_Toc68881048)

[第五章 优化后的结果进行量子精修 27](#_Toc68881049)

[5.1 运行开源量子精修方法XTB进行精修 27](#_Toc68881050)

[5.2 精修结果验证及分析 29](#_Toc68881051)

[第六章 总结与展望 31](#_Toc68881052)

[6.1 总结 31](#_Toc68881053)

[6.2 展望 32](#_Toc68881054)

[参考文献 33](#_Toc68881055)

[致 谢 38](#_Toc68881056)

[附录 40](#_Toc68881057)

[氢键标注源码片段 40](#_Toc68881058)

[氢键统计源码片段 45](#_Toc68881059)

[非共价键约束文件提取源码片段 47](#_Toc68881060)

[蛋白质序列对其源码片段 49](#_Toc68881061)

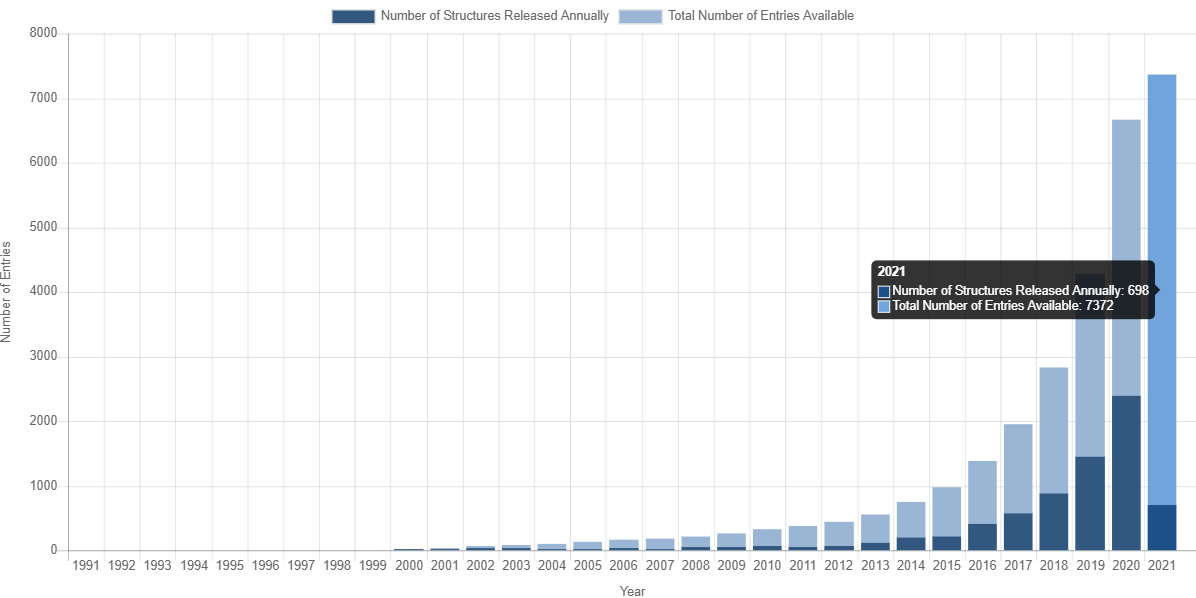
[X射线蛋白质链与低分辨率冷冻电镜蛋白质链相同序列匹配源码片段 52](#_Toc68881062)

# 第一章 绪论

## 课题研究的目的和意义

在蛋白质数据库Protein Data Bank[1][2]中，X射线衍射晶体结构约占所有结构的90% ;因此，它是获取生物大分子三维结构的主要工具。但是冷冻电镜(cryo-EM) 所测定的蛋白质结构的数目正以爆炸式的速度增长，正迅速成为X涉嫌衍射晶体结构主要竞争对手[3]。该技术对蛋白质或其他生物分子的迅速溶液冷冻，然后用电子轰击它们，以产生单个分子的显微镜图像。这些用于重建分子的3D形状或结构。这种结构有助于揭示蛋白质是如何工作的，它们是如何在疾病中发生故障的，以及如何用药物瞄准它们。

几十年来，结构生物学家们更喜欢使用X射线晶体学，这种技术使蛋白质结晶，用X射线照射蛋白质，并从由此产生的衍射光模式中重建它们的形状[4][5]。X射线晶体学产生高质量的结构，但要应用与所有蛋白质并不容易——有些蛋白质可能需要几个月或几年的时间才能结晶，而另一些则根本不能结晶。Cryo-EM不需要蛋白质晶体，开发者赢得了2017年诺贝尔化学奖，使得冷冻电镜蛋白质结构在蛋白质数据库中增长迅猛，如下图1.1所示。



1. **PDB Statistics: Growth of Structures from 3DEM Experiments Released per Year()**

虽然冷冻电镜技术的飞速进步带来了明显更高分辨率的蛋白质结构三维重建，但与X射线衍射晶体学相比，冷冻电镜蛋白质结构的平均质量仍然处于低分辨率的末端。长期以来，改进原子模型的一个挑战是为这个分辨率体系生成有立体化学意义的模型。现有的改进原子模型的方法主要分两大类：经典精修（如 ：基于分子动力学的CCTBX）[6]和量子精修（如基于量子化学的XTB[7]、TeraChem[9]）。但是两者在精修冷冻电镜蛋白质时各有弊端，正如CCTBX官方说明的CCTBX适合小分子结构[8]，但冷冻电镜蛋白质大部分趋于大分子蛋白质[1]，如果是分辨率低的大分子冷冻电镜结构，cctbx计算起来可能更加困难；量子精修则对于初始模型有较高要求，初始模型分辨率太低或者结构太差可能使量子精修蛋白质结构过程中消耗资源过大且计算时间过长[11]，但是冷冻电镜蛋白质分辨率相较于X涉嫌衍射蛋白质大部分还处在低分辨率阶段，而且会有可能精修到达局部最优后便停止无法到达最稳定、能量最低的结构状态，解决这个问题我们可以调大精修梯度使其更快收敛，但这又会导致另外一个问题：精修在最稳定结构状态无限震荡下去，永远无法达到最优状态。这是一个长期困扰蛋白质结构精修研究者的问题。

本文尝试寻找一种方案解决或者改善上述面临的问题。基于开源软件phenix[10]进行二次开发，实现非共价键及氨基酸残基内部与非共价键相关的共价键标注及约束文件自动生成，用于初始模型的几何优化，然后用于经典精修和量子精修。

## 蛋白质结构经典精修

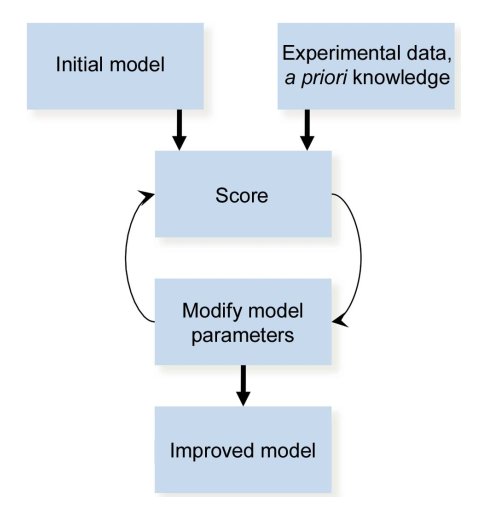


图2 A general model refinement workflow

### 1.2.1蛋白质经典精修原理和方法

一般的精修示意图见图2。给定原始原子模型和实验数据，精修算法根据原子参数计算精修目标及其导数，然后将其发送给优化器(通常是最小化器)。最小化器更新模型参数，然后将它们发送回精修算法，精修算法然后计算一个新的目标值和一组导数，并将它们返回给最小化器。这个过程迭代进行，直到达到收敛。更精细的细节和具体的实现取决于特定的软件和实验数据(例如x射线、中子衍射或冷冻电镜)。针对实验数据的模型精修是改变描述模型的参数以满足目标函数的优化过程。一个目标函数将模型参数与实验数据联系起来，如果需要，还需要一个先验知识。生物大分子的的结构数据的质量总是不满足直接用来精修的条件，所以精修之前总是需要准备相应的先验知识，而且整个蛋白质数据库中令人满意的超高分辨率数据不足0.5%。

先验知识通常作为约束或加权项(wTrestraints)引入精修目标函数，

(1-1)

以后被称为“restraints”。Tdata被称为实验术语或数据术语，是对数据拟合的模型评分的术语，w是平衡实验数据和约束贡献的相对权重。

最流行的经典精修软件，如REFMAC[12], SHELXL[13], CNS[14], BUSTER-TNT[15]和phenix.refine[16]，使用一系列的势能来限制原子模型的特定特征，如键长或角度，或二面角等。一般来说，它是六项之和，

(1-2)

其中每一项都有一个特定的特征:共价键、角度、平面、手性体积、扭转角以及防止无意义的空间碰撞。在2.5-3Å或更高分辨率的数据下，这种约束在大多数情况下就足够了。然而，对于较低的分辨率，或者对于通常在冷冻电镜中发现的蛋白质分辨率，这些限制是不够的。事实上，在典型的大分子分辨率(大约2Å)下，没有足够的信息来确定原子层面的细节，但它包含了二级和更高阶结构的信息。然而，低分辨率的数据甚至可能不包含足够的信息来准确地描述次级结构。需要像(1-2)式中所述的限制来弥补信息的缺乏。在对低分辨率数据进行精修时，(1-2)中约束如果表现不佳则产生的不利影响至少增加一倍。首先，精修模型的几何形状可能不合理;例如，helices和sheets在使用相同的满足(1-2)式中的约束时可能会被扭曲。然后模型参数的数量过多可能会导致严重地过拟合。

经典精修目标的约束是(1-2)的函数;虽然这些额外的限制措施显然是一种改进，之前它们需要手动注释(用户需要告诉程序非共价键是什么)，而且它们仍然是简单的势函。这些电位被拟合以再现从汇编库中获得的平均值，而不考虑诸如原子周围环境和附近的电荷等更精细的细节。使用此类参数化约束的细化今后称为经典细化。我们基于phenix进行二次开发编写脚本可以自动化注释非共价键和非共价键相关的共价键、二面角等约束参数。用于几何优化初始模型，作为后面精修的初始模型使用。

### 1.2.2蛋白质经典精修方法

前面提到了集中经典精修软件如REFMAC[12], SHELXL[13], CNS[14], BUSTER-TNT[15]和phenix. real\_space\_refine[16]。其中phenix.real\_space\_refine是专门用作于冷冻电镜蛋白质结构精修的，计算晶体学工具箱(cctbx)作为Phenix项目的开源组件正在开发中。Phenix项目的目标是推进分子结构测定的自动化。phenix取决于cctbx，但反之亦然。这种分层的方法使清晰的设计成为一个可重用的库。因此，cctbx对于小分子晶体学甚至一般科学应用也有用。phenix. real\_space\_refine程序将模型参考map进行精修。map可以通过x射线、中子晶体学或冷冻电镜技术得到，其质量可以从好到坏不等。该软件的目标是获得一个模型，尽可能地拟合map，同时拥有有意义的几何形状(no validation outliers, such as Ramachandran plot[17] or rotamer outliers)。

Ramachandran plot是一个多肽中构成氨基酸所有可能骨架构型的图谱。图的轴由两个可以自由旋转的主键(分别为φ和ψ)的旋转角度组成;因此，图上的每一点φ，ψ都代表一个可以想象的氨基酸主链构型。

rotamers被定义为末端阻塞氨基酸侧链对应的局部最小势能(范德瓦尔斯力和扭转项)的侧链扭转(chi-angle)组合。如果至少有一个chi-angle与旋转体相差20度以上，则认为侧链在能量(势能增加不小于1 ~ 2 kcal/mol)和几何(原子定位精度低于0.5 ANG)两方面都存在偏差。

结合phenix.real\_space\_refine和phenix.refine的cctbx相对于其他软件具有许多优势：快速梯度驱动最小化组合map和constraints ，

引入基于局部网格搜索的拟合，可以修正旋转体离群值或较差的map到模型的匹配度；使用map引导的连续小模型碎片刚体进行移动；可选模拟退火优化；引入刚体精修；标准约束:键，角度，平面性，手性，二面体。附加约束：拉式图(“oldfield”和“emsley”两个选项)、C-beta偏差、旋转器、二级结构、参考模型、起始位置[18]；多线程运行。

使用cctbx进行经典精修时需要注意一下三个方面：

1，模型参数化。这决定了模型的哪些属性将被精修，以及如何对它们进行分组。选项包括坐标细化(单独针对对等空间数据或实空间映射;或作为刚体群对往复式空间数据);b因子或ADP(原子位移参数)改进，有几个选择分组和各向同性与各向异性;原子占据率(即原子出现在晶体细化位置的时间比例);和异常组。

2，优化目标。这定义了要最小化的函数;默认情况下，这至少是模型振幅与实验测量振幅的一致。对于几乎所有大分子结构来说，共价几何和附近原子的b因子的约束也是必要的，而且几何约束的各种定制是可能的[19]。

3，优化方法。在phenix.refine中使用的标准精修方法和类似的程序是梯度驱动的最小化算法，但也有其他选项可用。这些方法的收敛半径通常明显更大，但代价是增加了运行时长。

#### 参数化(精修策略)

所有策略都有相关的原子选择，它们控制模型的哪些部分被精修。下面是一些比较重要的参数，指定的分辨率限制仅是近似的[20]，在实践中可能会根据模型质量、数据质量和观测参数比而有很大差异。

* **XYZ coordinates** 是模型几何的标准笛卡尔坐标，并与x射线数据一致。这几乎总是合适的，但在低分辨率(通常是3.0 Å或更糟)额外的限制(NCS，参考模型等)通常是必要的，以防止过拟合。
* **Real-space** 也改变了原子位置，但使用电子密度图作为目标，而不是F(obs)。它将在局部运行(以适合密度较低的单个残基或旋转体异常值)和全局运行。这通常对中到高分辨率结构的早期精修最有帮助;在比3Å更糟糕**的情况下，效率会迅速下降[21]。**
* **Rigid-body 在不改变几何形状的情况下，改进了大的、用户定义的模型区域的位置。通常在分子替换后立即使用，r因子通常很高，分辨率很低(4.5**Å**或更低)。如果刚体和单个站点的细化都被检查(推荐)，在第一个精修循环中，首先执行刚体步骤。**
* **Individual B-factors 是受限制的原子b因子精修参数，通常是合适的，除非在非常低的分辨率下(有一些关于截止应该是什么，但我们通常发现PDB中的绝大多数结构通过这个选项细化得很好)。原子是各向同性还是各向异性是单独定义的[21]。**
* **Group B-factors** 为一次多个原子精炼b因子，大多数情况下单个残基可以作为一个整体，也可以分为主链和侧链。
* **TLS parameters 改进了大原子团的各向异性位移，通常是单个域或链。由于每组只引入10个参数，而且使用的组相对较少，所以除非单个各向异性b因子得到精修，否则几乎在任何分辨率下都是合适的。**
* **Occupancies 改进原子占有率。在高分辨率下(通常为1.7**Å**或更高)，配体、离子或溶剂原子不是结合在每个单元格中的情况下。默认情况下，它只会对部分占据的原子执行，因此即使没有部分占据的原子，也可以让它保持打开状态。如果可能，分组是自动确定的，但也可以手动指定它们。**

#### 目标和约束

* **Target function 描述了基于模型的数据项的计算;“ML”指“最大可能性”，通常是适当的;“MLHL”在添加实验相约束时使用;“ML-SAD”不适合一般使用。最小二乘(LS)目标只对孪生细化或非常小的数据集有用[22]。**
* **Weight optimization 使用网格搜索来确定目标函数和几何或b因素约束的最佳相对权重。在原子分辨率下，这很少是必要的，但在低分辨率下，这将非常有帮助，**
* **Automatic covalent bonds 补充标准的共价几何约束为特殊情况，如修饰氨基酸，糖蛋白。但是这可能会导致创建虚假键。所以本文定义自定义几何约束来进行几何优化。**
* **Reference model restraints 利用现有结构的二面角作为输入模型构象的约束，例如，提取具有高分辨率结构的分子的低分辨率结构，也就是本文脚本自动生成的从相同序列的高分辨率蛋白质链提取约束模型的一部分。这在低分辨率(通常低于3.0**Å**)下最有用[23]，这正好是冷冻电镜大部分蛋白质结构分辨率所在的区间。**
* **Secondary structure restraints** 增加α螺旋、β表和沃森-克里克碱基对中氢键的距离限制。这些有助于在较低分辨率(2.5 Å或更低)下保持正确的几何图形。适当的atom选择将被自动检测到，但是本文自动化标注并提取了它们。

#### 优化方法和其他选项

* **Automatically add hydrogens to model 本文通过python进一步调用phenix.ready\_set接口实现在适当的地方添加氢原子。这通常只影响在高分辨率中的r因子，但可以在任何分辨率下都非常有助于改善几何结构。在整个精炼过程中在蛋白质、核酸和配体分子上使用明确的氢。但是水会影响结构的分辨率，在水中添加氢后会使冷冻电镜蛋白质的结构分辨率变得更低。**
* **Simulated annealing** 使用分子动力学与额外的数据项作为额外的优化方法对于消除相位偏置和克服能量障碍非常有帮助，通常会产生一个显著的改进，而不是简单的最小化早期的精修。主要的缺点是速度。两种类型的参数化是可用的，笛卡儿角和扭转角;后者更适合于低分辨率[24]。
* **Automatically correct N/Q/H errors 使用程序Reduce向后翻转侧链，当显式氢不存在时，侧链显得对称。**
* **Twin law 使成双成对的细化。精修目标是最小二乘(LS)而不是最大似然，r因子总是较低，孪生映射将适当减少模型偏差。**
* **Number of processors 决定几个可选步骤的并行化，最重要的是权重优化，它可以在大型多核系统上将运行时间减半。使用默认设置phenix.refine几乎完全以串行方式运行，由于操作系统的限制。**

s

### 1.2.3 蛋白质经典精修优势与劣势

相对于量子精修来说经典精修对初始模型的要求没有那么苛刻，而且经典精修应用的蛋白质结构的多样性也大大多余量子精修，对于某些结构特别差的低分辨率冷冻电镜蛋白质结构可能直接没有精修的意义，因为可能在浪费了无数计算资源之后还是不能收敛到理想状态结构，因为初始模型太差了，但是较低分辨率的冷冻电镜模型是可以用来经典精修的，虽然精修效果有待提高，但是可以手动添加约束模型的方法来提高经典精修的效果和稳定性，而且经典精修消耗计算资源和时间较少。而且经典精修如cctbx等一般都带有相应的结构验证工具，我们可以利用经典精修对初始模型进行快速几何结构等检验，如果可以，我们可以进一步进行量子精修，起到一个筛选作用，如果检验结果显示模型很差，那我们可以想其他办法，比如本文自动标注并提取相同序列对高分辨率蛋白质链约束模型的办法，先对初始模型进行几何优化，是初始模型达到一个较高的状态然后进行量子精修。我们优化好初始模型选择进行量子精修而不是经典精修的原因也是因为经典精修的弊端，虽然在小分子蛋白质结构中经典精修表现良好，但冷冻电镜技术获取的蛋白质结构趋向于大分子结构，在本文后面的实验中也证明了这一点，量子精修在冷冻电镜蛋白质结构精修的结果是略优于经典精修的。

## 蛋白质结构量子精修

量子精修是对基于分子力场机制(MM)精修的改进。在后者中,数据补充额外的化学信息通过MM力场进行精修,而量子精修描述至关重要的大分子的通过量子力学(QM)而不是分子力学。[25]量子化学方法有潜力通过以更少的特殊方式提供约束，在精修过程中发挥转型作用，这可能会导致更有化学意义的结构[26]。基于量子的精修不包括任何参数化约束，如(1-2)公式，这意味着当遇到新的配体时，不需要创建特定于配体的参数。

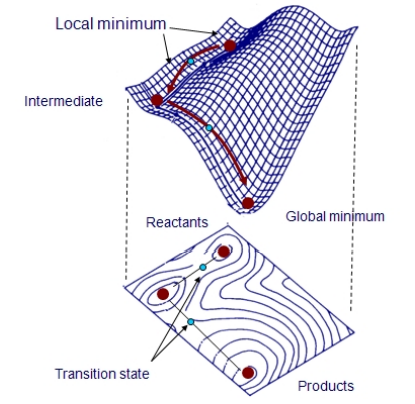


图3 量子精修能量梯度路径

### 1.3.1量子精修方法及原理

对于一个给定的分子系统，适当的量子化学方法需要在几何形状的精度和计算成本之间进行权衡[27][28]。对大分子进行精确高效的量子化学计算仍然是计算化学中的一个挑战[29][30]。然而，以前曾报道过利用量子化学计算作为精修限制源的几次尝试，它们可以分为以下几类。

#### Hybrid QM/MM

精修过程可以集中在分子的“活跃”区域上。这样做的优点是不必浪费计算资源来更好地描述(可能没有意义的)环境区域。Ryde[31][32][33]和Nilsson[34]等人是这一领域的先驱。ComQum软件包[35]是Ryde等人为这项任务开发的。在此之后，ChemShell [36]作为一个用于QM/MM模拟的模块化软件包，被修改以执行基于QM/MM的蛋白x射线结构的精修[37]。混合QM/ MM方法的挑战是需要仔细选择活跃的QM区域，确保采取足够大的区域。进行收敛性研究是费时费力的，而且，寻找一个平衡的力场和从头开始计算的组合仍然是QM/MM模型的探索领域一个具有挑战性的任务。

#### Semi-empirical

Merz和同事的开创性工作已经成功地解决了量子细化中的许多问题[38][39][40][41][42][43]。使用半经验计算或多或少地缓解了计算尺度的问题。DivCon软件[44]用于此目的，并与PHENIX进行了交互[45]。采用半经验方法是有吸引力的，因为其固有的计算效率更高的尺度[46]。然而，从长远来看，准确性和稳健性问题可能被证明是一个太大的缺点。

#### Linear-scaling density functional theory

Canfield等人[47]的工作采用了基于分治和征服的QM/MM优化方法来研究15万个原子的光系统I三聚体。整个蛋白质被分成单独的优化区域，每个区域(及其直接环境)由密度泛函理论(DFT)处理，其余的蛋白质由分子力学处理。本研究利用DFT计算得到的力对结构进行优化。这个计算发现了将三聚体连接在一起的结构特征。叶绿素“特殊对”的坐标存在严重的误差。对35个残基侧链的取向进行了优化，形成了改进的氢键网络。量子方法如半经验、harte - fock或DFT可以用来计算辅助因子、共晶体、药物结合活性位点等的约束。基于量子的方法主要关注的是所需的大量计算资源。新软件包Q|R[48]对于一个完整的量子描述比一个混合的基于QM/ mm的方法有很多好处;例如，可以减轻准备配体的力场参数的费力步骤，可以避免伪QM/MM边界效应。新软件包的另一个设计目标是与许多不同的量子化学软件包建立通用接口。这意味着没有明确地耦合到单个量子化学包。通用的接口将有助于从许多量子化学软件套件中快速整合任何新实现的方法。

Q|R使用cctbx开源库[51]构建一个标准的细化协议，非常类似于phenix。改进和大多数其他凤凰工具。cctbx用于计算(1)中的数据项及其衍生物，进行缩放和计算大体积溶剂，并使用L-BFGS驱动细化[52]最小化。Q|R中的标准和量子细化都在条件(1-1)下进行了最小化。这两种细化的区别是约束的来源。cctbx用于使用标准细化中的表达式(1-2)计算约束及其导数。然而,限制(Trestraints)及其在量子精修中的衍生物是由外部量子化学软件计算的电子能量和梯度。

### 1.4.3蛋白质量子精修的优势和劣

量子精修以为不需要手动标注添加额外的约束参数模型，而是通过计算化学软件自动计算分子体系的整体能量和能量梯度，所以在面对新的蛋白质或者药物精修时候会变得更加灵活，而且量子精修在精修是对于分子体系的大小和精度方面都大大优于经典精修，但是量子精修也有一定的局限性，比如对于初始分子模型的要求过于苛刻，计算资源和计算时长消耗太大，计算化学软件繁杂，虽然都有默认参数，如果初始模型不是很理性地情况下，计算参数的设定需要一定的经验等。量子精修简化路径如图3中描述[53]，先达到局部最优然后达到全局最优，但是会由于精修激活函数的梯度差太小使其在局部最优状态停止，虽然可以选择较大的梯度差激活，使其达到全局最优附近，但是会造成在全局最优附近无限震荡无法收敛。如果可以使用经典精修或者几何优之后将的蛋白质结构跨国局部最优状态，然后选择合适的能量梯度就可以改善这一问题。

## 论文的主要研究内容

本论文是以作者攻读硕士学位期间承担课题的工作为基础，研究了国内外现有的蛋白质经典精修及量子精修的优劣势，发现其中的关联性，探索一个针对低分辨率冷冻电镜蛋白质结构精修的可行的方案，旨在结合经典精修和量子精修，扬长避短，即达到能够普遍适用于大分子低分辨率冷冻电镜蛋白质结构精修，又节省计算资源，改善量子精修对初始模型要求过高的弊端。下面图4中简单示意了低分辨率冷冻电镜蛋白质结构的精修流程图。

在第一章中阐述了课题研究的目的、意义以及国内外研究蛋白质经典精修和量子精修的现状及计算方法和策略。了解两种方法的优势和劣势，探索一套方案将经典精修和量子精修相结合用于低分辨率冷冻电镜蛋白质结构的精修。

第二章阐述了蛋白质结构中对于经典精修比较重要的分级结构、分子作用里等，根据相同序列的蛋白质链，可以标注提取高分辨率共价键等一级结构和螺旋或折叠中的氢键、盐桥等非共价键二级结构参数模型用于后面的低分辨率冷冻电镜蛋白质结构几何结构优化和精修。

第三章阐述了非共价键二级结构级相关一级结构参数约束模型的标注提取方法及将其用于初始低分辨率冷冻电镜蛋白质结构的几何优化过程，并利用经典精修携带的验证工具进行结果验证。

第四章主要是将前面生成的参数约束模型用经典精修方法cctbx进行快速验证，对比原始模型及几何优化过的模型进行经典精修的结果差异并分析原因。

第五章主要是将前面自动化生成的约束参数模型用量子精修方法xtb进行精修，对比原始模型及几何优化过的模型进行量子精修的结果差异并分析原因。

第六章主要对经典精修及量子精修前后的模型结果进行横向对比，并进行总结和展望。可以尝试更加复杂的过程：先对初始低分辨率冷冻电镜蛋白质模型进行几何优化，然后进行简单的经典精修，最后再将经典精修过的模型进一步进行量子精修，达到更好的效果。

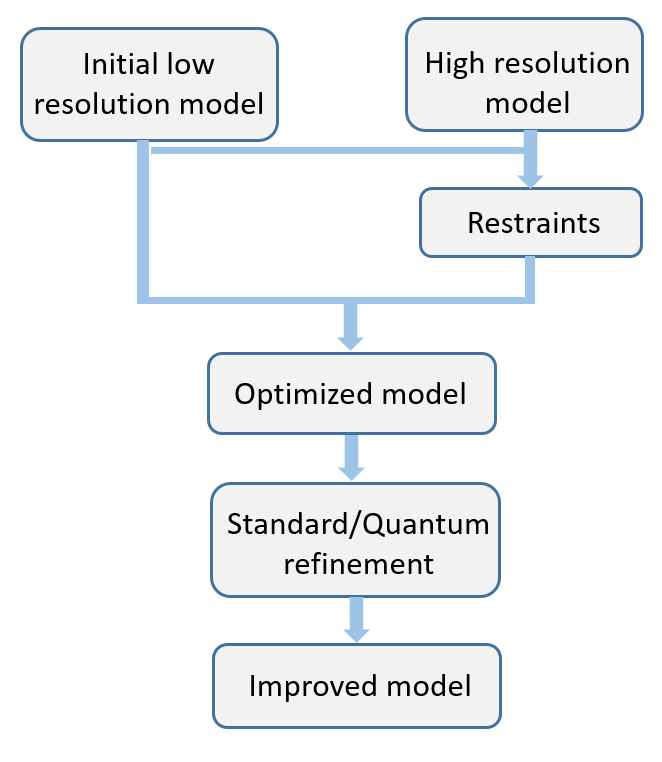


图4 低分辨率冷冻电镜蛋白质结构精修方案简易流程图

# 第二章 精修约束模型及参数

## 蛋白质结构

蛋白质是由氨基酸组成的生物聚合物。氨基酸由肽键连接在一起形成多肽链。一条或多条多肽链扭曲成三维形状形成蛋白质。蛋白质具有复杂的形状，包括各种折叠、环和曲线。蛋白质的折叠是自发发生的。多肽链各部分之间的化学结合有助于将蛋白质结合在一起，并使其形成形状。蛋白质结构的四个层次是由多肽链的复杂程度来区分的。如下面图5所示单个蛋白质分子可以包含一种或多种蛋白质结构类型:一级结构、二级结构、三级结构和四级结构。蛋白质的三维形状是由其一级结构决定的。氨基酸的顺序决定了蛋白质的结构和特定功能[54]。

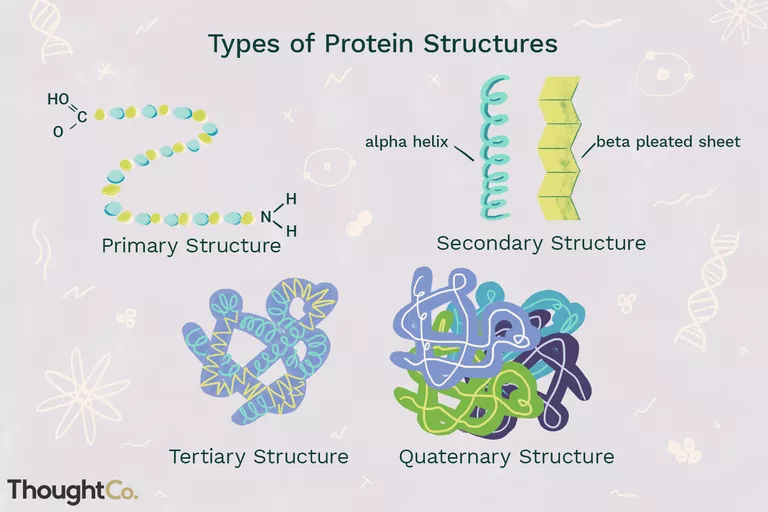


图5 蛋白质的结构类型

（https://www.thoughtco.com/protein-structure-373563）

### 2.1.1 蛋白质分级结构

蛋白质的结构取决于其氨基酸序列和多肽主链和氨基酸侧链中原子间的局部低能化学键。蛋白质结构在其功能中起关键作用;如果一个蛋白质在任何结构层面上失去了它的形状，它可能就不再具有功能。一级结构是氨基酸的序列。二级结构是多肽链之间的局部相互作用，包括α-螺旋和β-折叠片结构。三级结构是主要由R基团相互作用驱动的整体三维折叠结构。

第四系结构是多亚基蛋白质中亚基的取向和排列。

### 2.1.2 蛋白质序列

蛋白质的一级结构是组成蛋白质的每个多肽链中氨基酸的独特序列。实际上，这只是一个氨基酸在多肽链中以什么顺序出现的列表，并不是一个真正的结构。但是，因为最终的蛋白质结构最终取决于这个序列，这被称为多肽链的一级结构。例如下面图6展示胰岛素有两条多肽链，A和B和他们各自的一节结构序列。

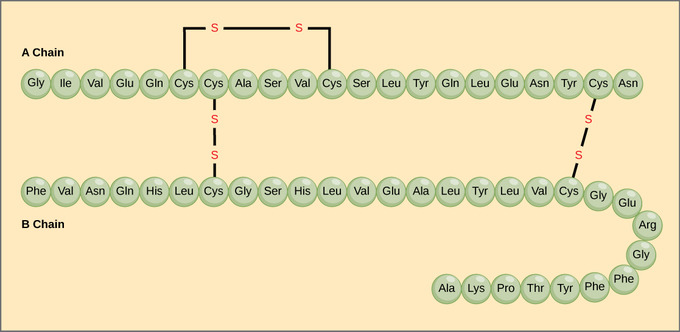


图6 胰岛素的一级蛋白质序列

（https://courses.lumenlearning.com/introchem/chapter/protein-structure/）

### 2.1.3 蛋白质分子间作用力

氢键在二级结构中非常重要，二级结构是指赋予蛋白质三维形状的多肽链的卷曲或折叠。在蛋白质中观察到两种类型的二级结构。一种是α螺旋结构。这种结构类似于一个卷曲的弹簧，由多肽链上的氢键固定。蛋白质的第二种二级结构是β折叠片。这种结构似乎是折叠的，并通过折叠链上相邻的多肽单元之间的氢键连接在一起。

多肽链上的氢键和氨基酸“R”基团之间的氢键通过使蛋白质保持疏水相互作用所建立的形状，有助于稳定蛋白质结构。盐桥和范德华力相互作用也有助于蛋白质结构的稳定。众所周知，原子模型精修在分辨率低于3-3.5 Å时具有挑战性[55]。这是因为超出这个分辨率，实验数据中的细节量不足以在原子水平上可视化蛋白质的结构特征，如二级结构或氨基酸残基的旋转结构。这些相互作用是指发生在极化分子之间的引力和排斥力。这些力有助于分子间的成键，而且对于分子保持稳定的结构状态至关重要。所以这些做作用力都可以成为我们标注提取用于优化初始结构模型的参数模型。

# 第三章 蛋白质初始结构模型优化

## 3.1.约束模型标注及提取

首先我们需要自动化标注非共价键，比如氢键，我们选取以下图7模型[56]作为标注参考。

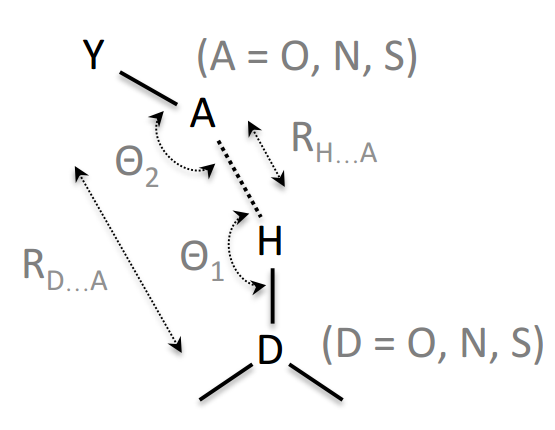


图7 氢键模型

我们在统计了蛋白质数据库中的不同分辨率的氢键参数后，Θ2角度分布如图8所示，其他参数不一一展示，最后决定我们使用的氢键参数模型中的几个参数的限定范围：A原子与氢原子之间的距离在1.4Å和2.4Å之间，Θ1的角度尽量大于120°，D原子与A原子之间的距离尽量在2.5 Å和3.5 Å之间，Θ2尽量是钝角。

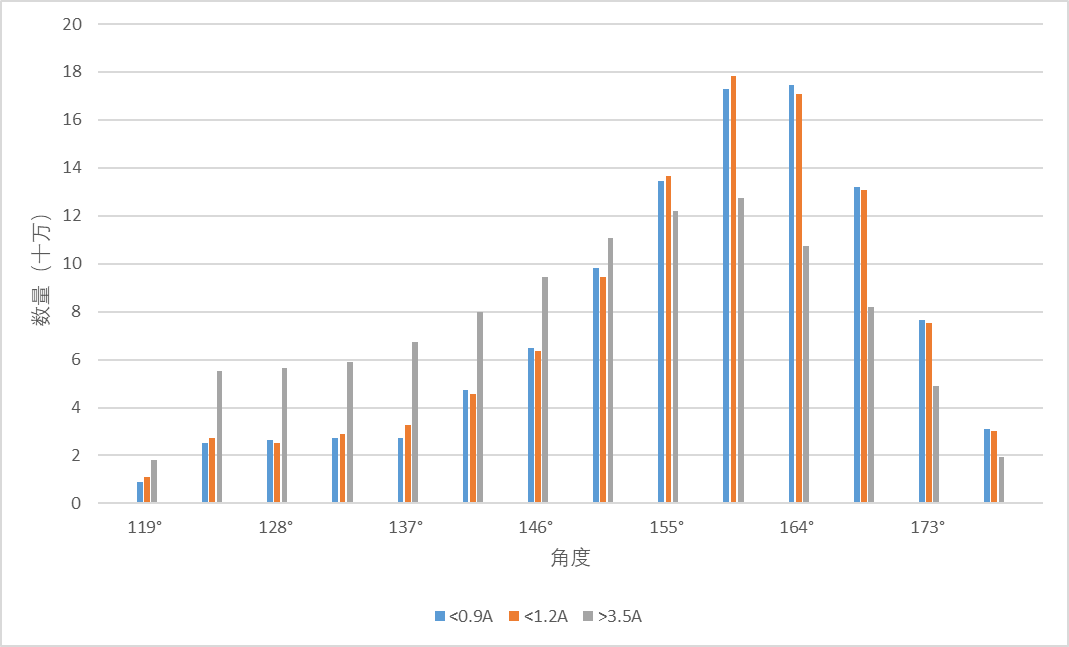


图8 氢键模型中Θ2在不同分辨率下的分布

由统计结果可以看出高分辨率大于0.9A和1.2A时的蛋白质模型氢键参数呈现基本相同的类似正态分布，大部分集中在140度到180度之间，低分辨率的蛋白质结构氢键参数有些许偏差，小于140度的区域还占有很大一部分。这说明低分辨率的蛋白质模型普遍存在问题，需要进一步精修优化，尤其是冷冻电镜蛋白质分子结构的分辨率普遍在3.5Å以下，所以低分辨率冷冻电镜蛋白质结构的精修需要更多的突破。

## 3.2 优化方法

## 3.3 几何结构优化结果

加工一个圆柱零件… …

… …

… …

H6={±(0,0,1), ±(0,1,0), ±(1,0,0)}………………(3.1)

（注：公式置中）

…　…

## 3.4 蛋白质优化结果验证

# 第四章 优化后的初始模型进行经典精修

## 4.1 运用经典精修CCTBX进行精修

考虑到与G代码的兼容性，以及当今绝大部分数控机床… …

## 4.2 精修结果验证及分析

… …。CVS能有效的将G代码转换为CNCS代码。… …

# 第五章 优化后的结果进行量子精修

在第三章的软硬件环境中，… …

## 5.1 运行开源量子精修方法XTB进行精修

## 5.2 精修结果验证及分析

# 第六章 总结与展望

## 6.1 总结

## 6.2 展望

参考文献

1. Berman, H.M. et al. *Nucleic Acids Res.* **28**, 235–242 (2000).
2. Bernstein, F.C. et al. *J. Mol. Biol.* **112**, 535–542 (1977).
3. Ewen Callaway,Nature | Vol 578 | 13 February 2020 | 201
4. Drenth J. Principles of Protein X-Ray Crystallography. Springer, Apr 5, 2007
5. Chernov AA (2003). "Protein crystals and their growth". J. Struct. Biol. 142 (1): 3–21
6. P. V. Afonine, B. P. Klaholz, N. W. Moriarty, B. K. Poon, O. V. Sobolev, T. C. Terwilliger, P. D. Adams and A. Urzhumtse.，New tools for the analysis and validation of cryo-EM maps and atomic models. Acta Cryst. (2018). D74, 814-840
7. Liam Wilbraham, Enrico Berardo, Lukas Turcani, Kim E. Jelfs, and Martijn A. Zwijnenburg. High-Throughput Screening Approach for the Optoelectronic Properties of Conjugated Polymers. Journal of Chemical Information and Modeling 2018 58 (12), 2450-2459
8. Grosse-Kunstleve R W , Afonine P V , Adams P D . cctbx news: Geometry restraints and other new features. office of scientific & technical information technical reports, 2004.
9. Seritan S , Bannwarth C , Fales B S , et al. TeraChem: A graphical processing unit‐accelerated electronic structure package for large‐scale ab initio molecular dynamics[J]. Wiley Interdisciplinary Reviews: Computational Molecular Science, 2020.
10. Adams P D , Afonine P V , G Bunkóczi, et al. PHENIX: a comprehensive Python‐based system for macromolecular structure solution[M].
11. Ching W Y , Adhikari P , Jawad B , et al. Ultra-large-scale ab initio quantum chemical computation of bio-molecular systems: the case of Spike protein of SARS-CoV-2 virus[J]. Computational and Structural Biotechnology Journal, 2021.
12. Murshudov G N , P Skubák, Lebedev A A , et al. REFMAC5 for the refinement of macromolecular crystal structures[J]. Acta Crystallographica, 2011, 67(Pt 4):355-367.
13. Sheldrick G M , Schneider T R . [16] SHELXL: High-resolution refinement[J]. Methods in Enzymology, 1997, 277(277):319-343.
14. Hickey W F . Perivascular microglial cells of the CNS are bone marrow-derived and present antigen in vivo[J]. Science, 1988, 239(4837):290-292.
15. E B lanc, Roversi P , Vonrhein C , et al. Refinement of severely incomplete structures with maximum likelihood in BUSTER-TNT[J]. Acta Crystallographica Section D Biological Crystallography, 2005, 60(Pt 12 Pt 1):2210-2221.
16. Afonine P V , Poon B K , Read R J , et al. Real-space refinement in PHENIX for cryo-EM and crystallography[J]. Acta Crystallographica Section D Structural Biology, 2018, 74(6).
17. Gooch J W . Ramachandran Plot[M]. Springer New York, 2011.
18. Towards automated crystallographic structure refinement with phenix.refine. P.V. Afonine, R.W. Grosse-Kunstleve, N. Echols, J.J. Headd, N.W. Moriarty, M. Mustyakimov, T.C. Terwilliger, A. Urzhumtsev, P.H. Zwart, and P.D. Adams. Acta Crystallogr D Biol Crystallogr 68, 352-67 (2012).
19. Conformation dependence of backbone geometry in proteins. D.S. Berkholz, M.V. Shapovalov, R.L. Dunbrack, and P.A. Karplus. [Structure 17, 1316-25 (2009)](https://doi.org/10.1016/j.str.2009.08.012).
20. MolProbity: More and better reference data for improved all-atom structure validation. C.J. Williams, B.J. Hintze, J.J. Headd, N.W. Moriarty, V.B. Chen, S. Jain, S.M. Prisant MG Lewis, L.L. Videau, D.A. Keedy, L.N. Deis, I.I.I. Arendall WB, V. Verma, J.S. Snoeyink, P.D. Adams, S.C. Lovell, J.S. Richardson, and D.C. Richardson. [Protein Science 27, 293-315 (2018)](https://doi.org/doi:10.1002/pro.3330).
21. Use of knowledge-based restraints in phenix.refine to improve macromolecular refinement at low resolution. J.J. Headd, N. Echols, P.V. Afonine, R.W. Grosse-Kunstleve, V.B. Chen, N.W. Moriarty, D.C. Richardson, J.S. Richardson, and P.D. Adams. [Acta Cryst. D68, 381-390 (2012)](https://doi.org/10.1107/S0907444911047834).
22. electronic Ligand Builder and Optimization Workbench (eLBOW): a tool for ligand coordinate and restraint generation. N.W. Moriarty, R.W. Grosse-Kunstleve, and P.D. Adams. [Acta Crystallogr D Biol Crystallogr 65, 1074-80 (2009)](https://doi.org/10.1107/S0907444909029436).
23. Using a conformation-dependent stereochemical library improves crystallographic refinement of proteins. D.E. Tronrud, D.S. Berkholz, and P.A. Karplus. [Acta Crystallogr D Biol Crystallogr 66, 834-42 (2010)](https://doi.org/10.1107/S0907444910019207).
24. Crystallographic model quality at a glance. L. Urzhumtseva, P.V. Afonine, P.D. Adams, and A. Urzhumtsev. [Acta Cryst. D65, 297-300 (2009)](https://doi.org/doi:10.1107/S0907444908044296).
25. Hsiao Y W ,Sanchez-Garcia E ,Doerr M , et al. Quantum Refinement of Protein Structures: Implementation and Application to the Red Fluorescent Protein DsRed.M1[J]. Journal of Physical Chemistry B, 2010, 114(46):15413-15423.
26. Carlsen, M. & Røgen, P. (2015). Proteins, 83, 1616–1624
27. Goerigk, L., Collyer, C. A. & Reimers, J. R. (2014). J. Phys. Chem. B,118, 14612–14626
28. Goerigk, L. & Reimers, J. R. (2013). J. Chem. Theory Comput. 9,3240–3251
29. Borbulevych, O. Y., Plumley, J. A., Martin, R. I., Merz, K. M. Jr &Westerhoff, L. M. (2014). Acta Cryst. D70, 1233–1247.
30. Borbulevych, O., Martin, R. I., Tickle, I. J. & Westerhoff, L. M. (2016).Acta Cryst. D72, 586–598.
31. Ryde, U. (2003). Curr. Opin. Chem. Biol. 7, 136–142.
32. Ryde, U. & Nilsson, K. (2003a). J. Mol. Struct. 632, 259–275
33. Ryde, U. & Nilsson, K. (2003b). J. Am. Chem. Soc. 125, 14232–14233
34. Nilsson, K., Hersleth, H.-P., Rod, T. H., Andersson, K. K. & Ryde, U.(2004). Biophys. J. 87, 3437–3447.
35. Ryde, U. (1996). J. Comput. Aided Mol. Des. 10, 153–164.
36. Sherwood, P. et al. (2003). J. Mol. Struct. 632, 1–28.
37. Hsiao, Y. W., Sanchez-Garcia, E., Doerr, M. & Thiel, W. (2010). J.Phys. Chem. B, 114, 15413–15423
38. Yu, N., Hayik, S. A., Wang, B., Liao, N., Reynolds, C. H. & Merz,K. M. Jr (2006). J. Chem. Theory Comput. 2, 1057–1069
39. Yu, N., Li, X., Cui, G., Hayik, S. A. & Merz, K. M. Jr (2006). Protein Sci. 15, 2773–2784.
40. Yu, N., Yennawar, H. P. & Merz, K. M. Jr (2005). Acta Cryst. D61,322–332.
41. Li, X., Fu, Z. & Merz, K. M. Jr (2012). J. Comput. Chem. 33, 301–310
42. Fu, Z., Li, X., Miao, Y. & Merz, K. M. Jr (2013). J. Chem. Theory Comput. 9, 1686–1693
43. Borbulevych, O. Y., Plumley, J. A., Martin, R. I., Merz, K. M. Jr &Westerhoff, L. M. (2014). Acta Cryst. D70, 1233–1247
44. Dixon, S. L. & Merz, K. M. Jr (1996). J. Chem. Phys. 104, 6643–6649
45. Adams, P. D. et al. (2010). Acta Cryst. D66, 213–221
46. Korth, M. & Thiel, W. (2011). J. Chem. Theory Comput. 7, 2929–2936
47. Canfield, P., Dahlbom, M. G., Hush, N. S. & Reimers, J. R. (2006). J.Chem. Phys. 124, 024301
48. Zheng M,Moriarty NW , XuY , etal. Solving the scalability issue in quantum-based refinement: Q|R#1[J]. Acta Crystallographica, 2017, 73(12):1020–1028.
49. Grosse-Kunstleve, R. W. & Adams, P. D. (2002). J. Appl. Cryst. 35,477–480
50. Grosse-Kunstleve, R. W., Sauter, N. K., Moriarty, N. W. & Adams,P. D. (2002). J. Appl. Cryst. 35, 126–136
51. Liu, D. C. & Nocedal, J. (1989). Math. Program. 45, 503–528
52. Byrd, R. H., Lu, P., Nocedal, J. & Zhu, C. (1995). SIAM J. Sci.Comput. 16, 1190–1208.
53. Gaussian accelerated molecular dynamics: Principles and applications[J]. Wiley Interdisciplinary Reviews Computational Molecular Science, 2021.
54. Bailey, Regina. "Learn About the 4 Types of Protein Structure." ThoughtCo, Aug. 28, 2020, thoughtco.com/protein-structure-373563.
55. Jane S , Richardson, Christopher J , et al. Assessment of detailed conformations suggests strategies for improving cryoEM models: Helix at lower resolution, ensembles, pre-refinement fixups, and validation at multi-residue length scale.[J]. Journal of structural biology, 2018.
56. Voth&Ho Shing. Current Topic in Medicinal Chemistry, Volume 7, Number 14, 2007,pp.1336-1348(13)

致 谢

本文是在导师×××教授的悉心指导下完成的。承蒙×老师的亲切关怀和精心指导，虽然有繁忙的工作，但仍抽出时间给予我学术上的指导和帮助，特别是给我提供了良好的学习环境，使我从中获益不浅。×老师对学生认真负责的态度、严谨的科学研究方法、敏锐的学术洞察力、勤勉的工作作风以及勇于创新、勇于开拓的精神是我永远学习的榜样。在此，谨向×老师致以深深的敬意和由衷的感谢。

还要感谢我的父母，他们在生活上给予我很大的支持和鼓励，是他们给予我努力学习的信心和力量。

最后，感谢所有关心我、支持我和帮助过我的同学、朋友、老师和亲人。在这里，我仅用一句话来表明我无法言语的心情：感谢你们！

# 附录

#### 氢键标注源码片段

def get\_D\_H\_A\_Y(i, j, Hs, fsc0, rt\_mx\_ji, fm, om, atoms):

"""

Get atom objects for donor and acceptor atoms

Apply symmetry op if necessary, so that correct geometry can be calculated

"""

Y = []

if(atoms[i].element in Hs):

H = atoms[i]

D = atoms[fsc0[i][0]]

A = atoms[j]

Y\_iseqs = fsc0[j]

if(len(Y\_iseqs)>0):

Y = [atoms[k] for k in fsc0[j]]

atom\_H = make\_atom\_id(atom = H, index = i)

atom\_A = make\_atom\_id(atom = A, index = j)

atom\_D = make\_atom\_id(atom = D, index = D.i\_seq)

if(rt\_mx\_ji is not None and str(rt\_mx\_ji) != "x,y,z"):

A = apply\_symop\_to\_copy(A, rt\_mx\_ji, fm, om)

if(len(Y\_iseqs)>0):

Y = [apply\_symop\_to\_copy(y, rt\_mx\_ji, fm, om) for y in Y]

if(atoms[j].element in Hs):

H = atoms[j]

D = atoms[fsc0[j][0]]

A = atoms[i]

Y\_iseqs = fsc0[i]

if(len(Y\_iseqs)>0):

Y = [atoms[k] for k in fsc0[i]]

atom\_A = make\_atom\_id(atom = A, index = i)

atom\_H = make\_atom\_id(atom = H, index = j)

atom\_D = make\_atom\_id(atom = D, index = D.i\_seq)

if(rt\_mx\_ji is not None and str(rt\_mx\_ji) != "x,y,z"):

H = apply\_symop\_to\_copy(H, rt\_mx\_ji, fm, om)

D = apply\_symop\_to\_copy(D, rt\_mx\_ji, fm, om)

return D, H, A, Y, atom\_A, atom\_H, atom\_D

class find(object):

def \_\_init\_\_(self,

model,

h\_bond\_params = None,

protein\_only = False,

pair\_proxies = None,

write\_eff\_file = True):

if(h\_bond\_params is None):

import cctbx.geometry\_restraints.process\_nonbonded\_proxies as pnp

h\_bond\_params = pnp.h\_bond()

Hs = h\_bond\_params.Hs

As = h\_bond\_params.As

Ds = h\_bond\_params.Ds

d\_HA\_cutoff = h\_bond\_params.d\_HA\_cutoff

d\_DA\_cutoff = h\_bond\_params.d\_DA\_cutoff

a\_DHA\_cutoff = h\_bond\_params.a\_DHA\_cutoff

a\_YAH\_cutoff = h\_bond\_params.a\_YAH\_cutoff

#

self.result = []

self.model = model

self.pair\_proxies = pair\_proxies

self.external\_proxies = False

if(self.pair\_proxies is not None):

self.external\_proxies = True

atoms = self.model.get\_hierarchy().atoms()

geometry = self.model.get\_restraints\_manager()

fsc0 = geometry.geometry.shell\_sym\_tables[0].full\_simple\_connectivity()

bond\_proxies\_simple, asu = geometry.geometry.get\_all\_bond\_proxies(

sites\_cart = self.model.get\_sites\_cart())

sites\_cart = self.model.get\_sites\_cart()

crystal\_symmetry = self.model.crystal\_symmetry()

fm = crystal\_symmetry.unit\_cell().fractionalization\_matrix()

om = crystal\_symmetry.unit\_cell().orthogonalization\_matrix()

pg = get\_pair\_generator(

crystal\_symmetry = crystal\_symmetry,

buffer\_thickness = d\_HA\_cutoff[1],

sites\_cart = sites\_cart)

get\_class = iotbx.pdb.common\_residue\_names\_get\_class

# find proxies if not provided

if(self.pair\_proxies is None):

pp = []

self.pair\_proxies = []

pp = [p for p in pg.pair\_generator]

else:

pp = self.pair\_proxies

# now loop over proxies

for p in pp:

i, j = p.i\_seq, p.j\_seq

if(self.external\_proxies): # making sure proxies point to same atoms

a\_i = make\_atom\_id(atom = atoms[i], index = i).id\_str

a\_j = make\_atom\_id(atom = atoms[j], index = j).id\_str

assert a\_i == p.atom\_A.id\_str, [a\_i, p.atom\_A.id\_str]

assert a\_j == p.atom\_H.id\_str, [a\_j, p.atom\_H.id\_str]

# presecreen candidates

ei, ej = atoms[i].element, atoms[j].element

is\_candidate = precheck(

atoms = atoms,

i = i,

j = j,

Hs = Hs,

As = As,

Ds = Ds,

fsc0 = fsc0)

if(protein\_only):

for it in [i,j]:

resname = atoms[it].parent().resname

is\_candidate &= get\_class(name=resname) == "common\_amino\_acid"

if(not is\_candidate): continue

# pre-screen candidates end

# symop tp map onto symmetry related

rt\_mx\_ji = None

if(not self.external\_proxies):

rt\_mx\_i = pg.conn\_asu\_mappings.get\_rt\_mx\_i(p)

rt\_mx\_j = pg.conn\_asu\_mappings.get\_rt\_mx\_j(p)

rt\_mx\_ji = rt\_mx\_i.inverse().multiply(rt\_mx\_j)

else:

rt\_mx\_ji = p.rt\_mx\_ji

#

D, H, A, Y, atom\_A, atom\_H, atom\_D = get\_D\_H\_A\_Y(

i = i,

j = j,

Hs = Hs,

fsc0 = fsc0,

rt\_mx\_ji = rt\_mx\_ji,

fm = fm,

om = om,

atoms = atoms)

if(len(Y) == 0): continue # don't use 'lone' acceptors

#

d\_DA = D.distance(A)

if(not self.external\_proxies):

if(d\_DA < d\_DA\_cutoff[0] or d\_DA > d\_DA\_cutoff[1]):

continue

#

d\_HA = A.distance(H)

if(not self.external\_proxies):

assert d\_HA <= d\_HA\_cutoff[1]

assert approx\_equal(math.sqrt(p.dist\_sq), d\_HA, 1.e-3)

if(d\_HA < d\_HA\_cutoff[0]): continue

# assert H.distance(D) < 1.15, [H.distance(D), H.name, D.name]

# filter by a\_DHA

a\_DHA = H.angle(A, D, deg=True)

if(not self.external\_proxies):

if(a\_DHA < a\_DHA\_cutoff): continue

# filter by a\_YAH

a\_YAH = []

if(len(Y)>0):

for Y\_ in Y:

a\_YAH\_ = A.angle(Y\_, H, deg=True)

a\_YAH.append(a\_YAH\_)

if(not self.external\_proxies):

flags = []

for a\_YAH\_ in a\_YAH:

flags.append(

not (a\_YAH\_ >= a\_YAH\_cutoff[0] and a\_YAH\_ <= a\_YAH\_cutoff[1]))

flags = list(set(flags))

if(len(flags)>1 or (len(flags)==1 and flags[0])): continue

#

assert approx\_equal(d\_HA, H.distance(A), 1.e-3)

#a\_YAD = []

#if(len(Y)>0):

# for Y\_ in Y:

# a\_YAD\_ = A.angle(Y\_, D, deg=True)

# a\_YAD.append(a\_YAD\_)

self.result.append(group\_args(

i = i,

j = j,

atom\_H = atom\_H,

atom\_A = atom\_A,

atom\_D = atom\_D,

symop = rt\_mx\_ji,

d\_HA = d\_HA,

a\_DHA = a\_DHA,

a\_YAH = a\_YAH,

#a\_YAD = a\_YAD,

d\_AD = A.distance(D)

))

if(not self.external\_proxies):

proxy\_custom = group\_args(i\_seq = i, j\_seq = j, rt\_mx\_ji = rt\_mx\_ji,

atom\_H = atom\_H, atom\_A = atom\_A)

self.pair\_proxies.append(proxy\_custom)

#

if(write\_eff\_file):

self.as\_restraints()

氢键统计源码片段

def get\_hierarchy(pdb\_inp):

hierarchy = pdb\_inp.construct\_hierarchy()

asc = hierarchy.atom\_selection\_cache()

ss = " or ".join(aa\_codes)

ss = "(%s) and not (resname UNX or resname UNK or resname UNL)"%ss

sel = asc.selection(ss)

hierarchy = hierarchy.select(sel)

xray\_structure = hierarchy.extract\_xray\_structure()

xray\_structure.convert\_to\_isotropic()

hierarchy.adopt\_xray\_structure(xray\_structure)

if(hierarchy.atoms().size()==0):

return None

if(len(hierarchy.models())>1):

return None

return hierarchy

def run(file\_name):

pdb\_code = os.path.basename(file\_name)[3:7]

pdb\_inp = iotbx.pdb.input(file\_name=file\_name)

data\_type = pdb\_inp.get\_experiment\_type()

resolution = pdb\_inp.resolution()

hierarchy = get\_hierarchy(pdb\_inp)

list\_p=None

if hierarchy is not None:

rr = run\_reduce\_with\_timeout(

stdin\_lines=hierarchy.as\_pdb\_string().splitlines(),

file\_name=None, # "model.pdb.gz",

parameters="-oh -his -flip -keep -allalt -pen9999 -",

override\_auto\_timeout\_with=None)

pdb\_inp = iotbx.pdb.input(source\_info=None, lines=rr.stdout\_lines)

hierarchy = pdb\_inp.construct\_hierarchy()

model = mmtbx.model.manager(

model\_input = None,

pdb\_hierarchy = hierarchy,

crystal\_symmetry = pdb\_inp.crystal\_symmetry(),

process\_input = True,

log = null\_out())

result = mmtbx.nci.hbond.find(model=model).result

for r in result:

list\_p = [r.d\_AD,r.d\_HA,r.a\_YAH,r.a\_DHA,data\_type,resolution,pdb\_code]

return list\_p

def foo(f):

list\_p =None

pdb\_inp = iotbx.pdb.input(file\_name=f)

try:

list\_p = run(f)

except Exception as e :

errorFile = open('log.txt','a')

errorFile.write(f + '\n')

errorFile.write(traceback.format\_exc())

errorFile.close()

return list\_p

logging.basicConfig(filename='log.txt',level=logging.DEBUG,

format='%(asctime)s - %(levelname)s - %(message)s')

path = '/home/wensong/PDB/pdb/'

of = open("".join([path,"INDEX"]),"r")

files = ["".join([path,f]).strip() for f in of.readlines()]

for f in files:

if not os.path.exists(f):

files.remove(f)

of.close()

i=0

list\_pdbcode\_result = pool\_map(

processes=10,

func=foo,

iterable=files)

easy\_pickle.dump("hydrogen\_bond\_statistics.pkl",list\_pdbcode\_result)

非共价键约束文件提取源码片段

def add\_hydrogen\_for\_hierarchy(hierarchy\_old):

xray\_structure = hierarchy\_old.extract\_xray\_structure()

xray\_structure.convert\_to\_isotropic()

hierarchy\_old.adopt\_xray\_structure(xray\_structure)

rr = run\_reduce\_with\_timeout(

stdin\_lines=hierarchy\_old.as\_pdb\_string().splitlines(),

file\_name=None, # "model.pdb.gz",

parameters="-oh -his -flip -keep -allalt -pen9999 -",

override\_auto\_timeout\_with=None)

pdb\_inp = iotbx.pdb.input(source\_info=None, lines=rr.stdout\_lines)

hierarchy\_new = pdb\_inp.construct\_hierarchy()

return hierarchy\_new

def prepare\_hydrogen\_restraints(hierarchy,hierarchy\_pair=None,pdb\_id=None,

fetch\_pdb=False,dump\_phil\_files=True,SS\_percent=True):

params = homology.get\_default\_params()

params.num\_of\_best\_pdb = 1

if hierarchy\_pair == None:

he\_h = add\_hydrogen\_for\_hierarchy(hierarchy)

atoms = he\_h.atoms()

atom\_total = atoms.size()

phils = []

fetch\_list=[]

num\_sum = 0

for chain in hierarchy.chains():

if not chain.is\_protein(): continue

sequence = chain.as\_padded\_sequence()

res = homology.perfect\_pair(sequence, params)

phil\_result= {}

if res is None:continue

for r in res :

pdb\_code = r.pdb\_code.lower()

chain\_id = r.chain\_id.lower()

match\_info = pdb\_id+"\_"+chain.id.lower()+"\_"+pdb\_code+"\_"+chain\_id

if fetch\_pdb == True:

easy\_run.call("phenix.fetch\_pdb {0}".format(pdb\_code))

pdb\_inp = iotbx.pdb.input(pdb\_code + ".pdb")

data\_type\_X = pdb\_inp.get\_experiment\_type()

data\_resolution = pdb\_inp.resolution()

if data\_resolution > 2.01:continue

if data\_type\_X != "X-RAY DIFFRACTION": continue

fetch\_list.append(pdb\_code)

# there are two conformers in chian B 233 5kdo.pdb,

# so cann't keep only one conformer situation in hierarchy

#easy\_run.call("phenix.pdbtools {0} remove\_alt\_confs=True".format(pdb\_code + ".pdb"))

#pdb\_inp = iotbx.pdb.input(pdb\_code + ".pdb\_modified.pdb")

else:

pdb\_file\_local = "/home/wangwensong/PDB/pdb/%s"%pdb\_code[1:3] + "/pdb%s"%pdb\_code + ".ent.gz"

#if not os.path.exists(pdb\_file\_local): continue

pdb\_inp = iotbx.pdb.input(pdb\_file\_local)

data\_type\_X = pdb\_inp.get\_experiment\_type()

data\_resolution = pdb\_inp.resolution()

if data\_resolution > 2.0:continue

if data\_type\_X != "X-RAY DIFFRACTION": continue

#easy\_run.call("phenix.pdbtools {0} remove\_alt\_confs=True".format(pdb\_file\_local))

#pdb\_inp = iotbx.pdb.input("pdb"+pdb\_code + ".ent.gz\_modified.pdb")

hx = pdb\_inp.construct\_hierarchy()

sel\_x = hx.atom\_selection\_cache().selection("protein and chain %s" % (chain\_id))

hx = hx.select(sel\_x)

hx\_h = add\_hydrogen\_for\_hierarchy(hx)

chain\_E = chain

chain\_X = (hx.only\_chain())

hierarchy\_E\_str = chain\_E.parent().parent().as\_pdb\_string()

hierarchy\_X\_str = hx\_h.as\_pdb\_string()

(phil\_obj, num\_sub) = align\_tow\_chain(chain\_E=chain\_E, chain\_X=chain\_X,

str\_chain\_E=hierarchy\_E\_str, str\_chain\_X=hierarchy\_X\_str)

if phil\_obj is None: continue

if num\_sub == 0: continue

num\_sum = num\_sum + num\_sub

phils.extend(phil\_obj)

if hierarchy\_pair is not None:

he\_h = add\_hydrogen\_for\_hierarchy(hierarchy)

atoms = he\_h.atoms()

atom\_total = atoms.size()

phils = []

num\_sum = 0

for chain in hierarchy.chains():

if not chain.is\_protein(): continue

chain\_E = chain

ss = " or ".join(aa\_codes)

ss = "(%s) and not (resname UNX or resname UNK or resname UNL)" % ss

sel\_x = hierarchy\_pair .atom\_selection\_cache().selection(ss)

hx = hierarchy\_pair .select(sel\_x)

chain\_X = (hx.only\_chain())

hx\_h = add\_hydrogen\_for\_hierarchy(hx)

hierarchy\_E\_str = chain\_E.parent().parent().as\_pdb\_string()

hierarchy\_X\_str = hx\_h.as\_pdb\_string()

(phil\_obj, num\_sub) = align\_tow\_chain(chain\_E=chain\_E, chain\_X=chain\_X,

str\_chain\_E=hierarchy\_E\_str,str\_chain\_X=hierarchy\_X\_str)

if phil\_obj is None: continue

if num\_sub == 0: continue

num\_sum = num\_sum + num\_sub

phils.extend(phil\_obj)

phils = list(set(phils))

if dump\_phil\_files == True:

phils = ("".join(phils))

dump\_phil\_file(phils, pdb\_id)

if SS\_percent == True:

p = (float('%.4f' % (float(num\_sum) / atom\_total))) \* 100

print("\nRestraint file is %s.eff; "%pdb\_id)

print("Percentage of {0} pdb atoms matching is :{1}".format(pdb\_id,p))

easy\_run.call("rm -f {0}".format("\*modified\*"))

for i in fetch\_list :

if i is None:continue

easy\_run.call("rm -f \*{0}\*".format(i))

if os.path.exists("hbond.eff"):

easy\_run.call("rm -r {0}".format("hbond.eff"))

if os.path.exists("myprotein.fasta"):

easy\_run.call("rm -r {0}".format("myprotein.fasta"))

return (p)

#### 蛋白质序列对其源码片段

def align\_tow\_chain(chain\_E,str\_chain\_E,chain\_X,str\_chain\_X,

distance\_ideal=None, sigma\_dist=0.1,angle\_ideal = None,

sigma\_angle=2, use\_actual=True):

se = chain\_E.as\_padded\_sequence()

sx = chain\_X.as\_padded\_sequence()

align = mmtbx.alignment.align(

seq\_a = se,

seq\_b = sx)

alignment = align.extract\_alignment()

chain\_E\_id = chain\_E.id

#chain\_X\_id = chain\_X.id

(i\_seqs,j\_seqs) = alignment.exact\_match\_selections()

match\_X\_E = {}

for (i,j) in zip(i\_seqs,j\_seqs):

match\_X\_E[j+1]=i

reses\_e=[]

for conformer in (chain\_E.conformers()):

reses\_e.append(conformer.residues())

for reses in reses\_e[1:]:

reses\_e[0].extend(reses)

reses\_E = reses\_e[0]

num\_sub = 0

for r in reses\_E:

if (int(r.resseq.strip()) - 1) in i\_seqs:

num\_sub = num\_sub + r.atoms\_size()

# get hydrogen bond informations of model cryo-EM

pdb\_inp\_E = iotbx.pdb.input(source\_info=None, lines=str\_chain\_E)

model\_E = mmtbx.model.manager(

model\_input=pdb\_inp\_E,

process\_input=True,

log=null\_out())

result\_E = mmtbx.nci.hbond.find(model=model\_E).result

# get hydrogen bond informations of model X-ray

pdb\_inp = iotbx.pdb.input(source\_info=None, lines=str\_chain\_X)

model = mmtbx.model.manager(

model\_input=pdb\_inp,

process\_input=True,

log=null\_out())

result\_X = mmtbx.nci.hbond.find(model=model).result

# matching restraints

f = "chain %s and resseq %s and name %s"

top=[]

dis =None

ang =None

base=None

for r\_e in result\_E:

for r\_x in result\_X:

if r\_e is None:continue

if r\_x is None:continue

# get h bond atoms in model cryo-EM

atom\_A\_e = r\_e.atom\_A

atom\_D\_e = r\_e.atom\_D

atom\_H\_e = r\_e.atom\_H

# get h bond atoms in model X-ray

atom\_A\_x = r\_x.atom\_A

atom\_D\_x = r\_x.atom\_D

atom\_H\_x = r\_x.atom\_H

#selection matching resseq in model cryo-EM

if not match\_X\_E.has\_key(int(atom\_A\_x.resseq.strip())):continue

if not match\_X\_E.has\_key(int(atom\_D\_x.resseq.strip())):continue

if not match\_X\_E.has\_key(int(atom\_H\_x.resseq.strip())):continue

# make sure matching reses

if match\_X\_E[int(atom\_A\_x.resseq.strip())] !=(int(atom\_A\_e.resseq.strip())):continue

if match\_X\_E[int(atom\_D\_x.resseq.strip())] !=(int(atom\_D\_e.resseq.strip())):continue

if match\_X\_E[int(atom\_H\_x.resseq.strip())] !=(int(atom\_H\_e.resseq.strip())):continue

if atom\_A\_e.name == atom\_A\_x.name:

if atom\_D\_e.name == atom\_D\_x.name:

if atom\_D\_e.name == atom\_D\_x.name:

a = f % (atom\_A\_e.chain,atom\_A\_e.resseq,atom\_A\_e.name)

d = f % (atom\_D\_e.chain,atom\_D\_e.resseq,atom\_D\_e.name)

h = f % (atom\_H\_e.chain,atom\_H\_e.resseq,atom\_H\_e.name)

if (not use\_actual):

if (r\_e.d\_HA < 2.5):

dt = 2.05

else:

dt = 2.8

if (r\_e.a\_DHA < 130):

at = 115

else:

at = 160

else:

dt = r\_x.d\_HA

at = r\_x.a\_DHA

e\_d = "%.3f" % (r\_e.d\_HA)

x\_d = "%.3f" % (r\_x.d\_HA)

print (e\_d,x\_d)

if e\_d != x\_d:

dis = """bond {

atom\_selection\_1 = %s

atom\_selection\_2 = %s

symmetry\_operation = %s

distance\_ideal = %f

sigma = 0.05

}

""" % (h, a, str(r\_e.symop), dt)

if (str(r\_e.symop) != "x,y,z"): continue

e\_a = "%.2f" % (r\_e.a\_DHA)

x\_a = "%.2f" % (r\_x.a\_DHA)

print (e\_a,x\_a)

if e\_a != x\_a :

ang = """angle {

atom\_selection\_1 = %s

atom\_selection\_2 = %s

atom\_selection\_3 = %s

angle\_ideal = %f

sigma = 5

}

""" % (a, h, d, at)

base = dis + ang

if base is not None:

top.append(base)

phil\_obj = list(set(top))

return (phil\_obj,num\_sub)

X射线蛋白质链与低分辨率冷冻电镜蛋白质链相同序列匹配源码片段

def get\_perfect\_pair(res,data\_type,model,file\_path, chain\_ids=None):

pdb\_info = iotbx.bioinformatics.pdb\_info.pdb\_info\_local()

pdb\_id = os.path.basename(file\_path).strip("pdb")[:4].lower()

pdb\_inp = iotbx.pdb.input(file\_path)

h = model.get\_hierarchy()

low\_res = res

tup = None

for chain in h.only\_model().chains():

if not chain.is\_protein():

continue

if chain\_ids is not None and chain.id not in chain\_ids():

continue

sequence = chain.as\_padded\_sequence()

l\_blast = local\_blast.pdbaa(seq=sequence)

blast\_xml\_result = l\_blast.run()

blast\_summary = summarize\_blast\_output("\n".join(blast\_xml\_result))

pdb\_ids\_to\_study = {}

for hit in blast\_summary:

ali\_identity = 100\*hit.length/chain.residue\_groups\_size()

if ali\_identity<100:continue

hit.show(out=sys.stdout)

if hit.identity < 100: continue

pdb\_ids\_to\_study[hit.pdb\_id] = hit.chain\_id

print pdb\_ids\_to\_study,"\*"\*39

pdb\_pair\_id = hit.pdb\_id.lower()

pdb\_pair\_chain\_id = hit.chain\_id

work\_dir = "/home/wensong/PDB/pdb/"

pdb\_path = work\_dir +str(pdb\_pair\_id[1:3])+"/pdb"+ pdb\_pair\_id + ".ent.gz"

pdb\_inp = iotbx.pdb.input(pdb\_path)

high\_res = pdb\_inp.resolution()

pdb\_pair\_data\_type = pdb\_inp.get\_experiment\_type()

if(0<=high\_res<=2.0)and("X-RAY DIFFRACTION" in pdb\_pair\_data\_type):

tup = (pdb\_id,chain.id,low\_res,data\_type,

pdb\_pair\_id,pdb\_pair\_chain\_id,high\_res,pdb\_pair\_data\_type,hit.identity)

if tup is not None:

print tup,"tup"\*40

return tup

def run(file\_path):

result = None

if os.path.isfile(file\_path):

pdb\_inp = iotbx.pdb.input(file\_path)

data\_type = pdb\_inp.get\_experiment\_type()

res = pdb\_inp.resolution()

if (data\_type == "ELECTRON MICROSCOPY") and (3.0 <= res <=6.0):

try:

model = mmtbx.model.manager(model\_input=iotbx.pdb.input(file\_path))

if model.size()<=11000:

tup = get\_perfect\_pair(res,data\_type,model,file\_path)

if tup:

result=tup

except Exception, e:

print "%s: %s"%(file\_path, e)

return result

files=[]

work\_dir = "/home/wensong/PDB/pdb/"

pdb\_info = easy\_pickle.load(file\_name='pdb\_dict.pickle')

for key,value in pdb\_info.items():

file\_name = "pdb"+key.lower()+".ent.gz"

file\_path = os.path.join(work\_dir,key.lower()[1:3],file\_name)

if os.path.exists(file\_path):

files.append(file\_path)

pair\_list=[]

for f in files:

try:

result = run(f)

if result:

pair\_list.append(result)

except Exception,e:

print e

if pair\_list is None: print "None pairList"\*20

if pair\_list is not None:

for pair in pair\_list:

if pair is None:

pair\_list.remove(pair)

if pair\_list:

easy\_pickle.dump("X\_RAY\_CYRO\_EM\_PAIR\_100\_whole.pkl",pair\_list)

else:print "why None?end"\*15