

PyMOL 完全学习手册：从入门到分子可视化大师

前言

PyMOL 作为一款功能强大的开源分子可视化软件，已成为生物信息学、结构生物学、药物研发等领域不可或缺的工具。它能够将复杂的分子结构以直观、精美的形式呈现，支持分子模型的构建、编辑、分析与渲染，无论是科研数据展示、学术论文插图制作，还是教学演示，都能发挥重要作用。

本手册专为零基础学习者及有进阶需求的科研人员编写，遵循“理论 + 实操”的原则，从软件安装到高级技巧，层层递进、逐步深入。每个章节都配备了详细的操作步骤、示例代码和常见问题解答，帮助读者快速上手并灵活运用 PyMOL 解决实际问题。无论你是学生、教师，还是科研工作者，都能通过本手册掌握 PyMOL 的核心技能，开启分子可视化的探索之旅。

第 1 章：PyMOL 简介与安装

1.1 PyMOL 的核心特点

PyMOL 由 Warren Lyford DeLano 开发，兼具开源性与专业性，其核心特点包括：

- 跨平台兼容性：支持 Windows、macOS、Linux 三大操作系统，满足不同用户的使用需求；
- 强大的可视化能力：支持多种分子模型显示模式（如球棍模型、丝带模型、表面模型等），可自定义颜色、透明度、光照效果，生成高质量图像和动画；
- 灵活的操作方式：提供图形化界面（GUI）和命令行两种操作模式，GUI 适合快速上手，命令行适合批量处理和高级定制；
- 丰富的功能拓展：支持 Python 脚本编程，可通过插件扩展功能，如分子对接结果分析、蛋白质结构比对、药效团模型构建等；
- 广泛的文件格式支持：兼容 PDB、XYZ、SDF、FASTA 等多种分子结构文件格式，方便与其他科研软件（如 VMD、Schrödinger、Discovery Studio）协同工作。

1.2 安装准备与系统要求

- Windows 系统：Windows 7 及以上版本，64 位操作系统，至少 2GB 内存，100MB 以上空闲磁盘空间；
- macOS 系统：macOS 10.10 及以上版本，64 位处理器，至少 2GB 内存，100MB 以上空闲磁盘空间；

- Linux 系统: Ubuntu 16.04 及以上、CentOS 7 及以上等主流发行版, 64 位操作系统, 至少 2GB 内存, 100MB 以上空闲磁盘空间。

1.3 不同系统的安装步骤

1.3.1 Windows 系统安装

1. 访问 PyMOL 官方网站 (<https://pymol.org/2/>) , 下载适合 Windows 系统的最新版本安装包 (建议选择开源免费版) ;
2. 双击下载的.exe 安装文件, 启动安装向导, 点击 “Next”;
3. 阅读许可协议, 勾选 “I accept the agreement”, 点击 “Next”;
4. 选择安装路径 (默认路径即可, 也可自定义, 注意路径中不要包含中文字符) , 点击 “Next”;
5. 选择创建桌面快捷方式, 点击 “Next”;
6. 点击 “Install” 开始安装, 等待安装完成 (约 1-2 分钟) ;
7. 安装完成后, 勾选 “Launch PyMOL”, 点击 “Finish” 启动软件。

1.3.2 macOS 系统安装

1. 访问 PyMOL 官方网站, 下载 macOS 版本的安装包 (.dmg 格式) ;
2. 双击.dmg 安装文件, 将 PyMOL 图标拖拽到 “Applications” 文件夹中;
3. 打开 “Applications” 文件夹, 找到 PyMOL 图标, 双击启动 (首次启动可能需要授权, 按照系统提示操作即可) 。

1.3.3 Linux 系统安装 (以 Ubuntu 为例)

1. 打开终端 (Ctrl+Alt+T) , 更新软件源:

```
sudo apt-get update
```

1. 安装 PyMOL 依赖包:

```
sudo apt-get install pymol
```

1. 等待安装完成后, 在终端中输入 “pymol” 即可启动软件。

1.4 安装后的验证与问题排查

1.4.1 验证安装成功

启动 PyMOL 后，若出现主界面（包含视图窗口、命令行、菜单等），且能正常加载示例分子（如默认显示的 PyMOL 图标分子），则说明安装成功。

1.4.2 常见安装问题排查

- 问题 1: Windows 系统启动 PyMOL 时提示“缺少.dll 文件”?

解决方案: 安装 Microsoft Visual C++ Redistributable（可从微软官网下载对应版本），重启电脑后重新启动 PyMOL。

- 问题 2: macOS 系统双击 PyMOL 图标无响应?

解决方案: 右键点击 PyMOL 图标，选择“打开”，在弹出的提示框中点击“打开”（绕过系统安全限制）；若仍无响应，检查系统版本是否符合要求，重新下载安装包重试。

- 问题 3: Linux 系统终端输入“pymol”提示“命令未找到”?

解决方案: 检查软件源是否更新成功，重新执行安装命令；若仍失败，可通过源码编译安装（参考官方文档）。

第 2 章: PyMOL 界面初识与基础操作

2.1 主界面组成与功能介绍

PyMOL 的主界面主要由以下几个部分组成，各部分功能明确，协同工作：

- 视图窗口 (Viewer Window)：核心区域，用于显示分子模型，支持鼠标操作旋转、平移、缩放分子；
- 命令行 (Command Line)：位于视图窗口下方，可输入 PyMOL 命令实现各种操作（如加载分子、调整显示模式等），支持命令自动补全（按 Tab 键）；
- 菜单栏 (Menu Bar)：位于主界面顶部，包含 File、Edit、View、Display 等多个菜单，提供图形化操作入口；
- 工具栏 (Toolbar)：位于菜单栏下方，包含常用操作的快捷按钮（如打开文件、保存图像、旋转分子等）；
- 对象管理器 (Object Manager)：位于主界面右侧，显示当前加载的分子对象和选择集，可通过勾选 / 取消勾选控制分子的显示 / 隐藏。

2.2 鼠标基础操作：旋转、平移、缩放

PyMOL 的视图操作主要通过鼠标完成，操作逻辑简单直观，需熟练掌握：

- 旋转分子：按住鼠标左键，在视图窗口中拖动，分子将围绕中心点旋转；
- 平移分子：按住鼠标中键（或 Ctrl + 鼠标左键），拖动鼠标可平移分子位置；
- 缩放分子：滚动鼠标滚轮（或 Shift + 鼠标左键拖动），可放大或缩小分子视图；
- 重置视图：双击鼠标左键，分子将自动居中并恢复默认缩放比例。

2.3 菜单栏核心功能详解

2.3.1 File 菜单（文件操作）

- New: 新建空白项目，清除当前所有分子对象；
- Open: 打开本地分子文件（支持 PDB、XYZ 等格式）；
- Fetch: 从 PDB 数据库直接下载分子结构（输入 PDB ID 即可，如“1AKE”）；
- Save: 保存当前分子对象为指定格式文件；
- Export Image: 导出视图窗口中的图像（支持 PNG、JPG、SVG 等格式）；
- Quit: 退出 PyMOL 软件。

2.3.2 Edit 菜单（编辑操作）

- Undo: 撤销上一步操作；
- Redo: 重做上一步操作；
- Select: 选择分子的特定部分（如原子、残基、链等）；
- Deselect: 取消当前选择；
- Delete: 删除选中的分子部分或整个分子对象。

2.3.3 View 菜单（视图设置）

- Reset View: 重置视图，分子居中显示；
- Zoom: 缩放视图（Zoom In 放大、Zoom Out 缩小）；
- Center: 将选中的分子部分设为视图中心；
- Orthographic: 切换正交投影 / 透视投影模式（正交投影适合精确测量，透视投影更具立体感）；
- Show/Hide: 显示或隐藏视图中的辅助元素（如坐标轴、网格、氢键等）。

2.3.4 Display 菜单（显示模式）

- Representation: 选择分子的显示模式 (核心功能, 后续章节详细介绍) ;
- Color: 设置分子的颜色 (可按原子类型、残基类型、链等分类着色) ;
- Transparency: 调整分子的透明度 (0 为不透明, 1 为完全透明) ;
- Quality: 设置显示质量 (High 高质量、Medium 中等质量、Low 低质量, 高质量显示更清晰但占用更多资源) 。

2.4 命令行基础使用

2.4.1 命令行的优势

命令行操作比图形化界面更高效, 尤其适合批量处理、重复操作和脚本编写。例如, 通过一条命令即可完成分子加载和显示模式设置, 无需多次点击菜单。

2.4.2 常用基础命令

- 加载分子: `load 文件名.pdb` (加载本地文件) 或 `fetch PDBID` (从 PDB 数据库下载) ;
- 显示模式设置: `show 显示模式, 对象名` (如 `show sticks, 1AKE`, 将 1AKE 分子设为球棍模型) ;
- 颜色设置: `color 颜色名, 选择对象` (如 `color red, chain A`, 将 A 链设为红色) ;
- 选择操作: `select 选择集名称, 选择条件` (如 `select active_site, resi 100-120`, 选择 100-120 号残基作为活性位点) ;
- 保存图像: `png 输出文件名.png, width=800, height=600` (导出 800×600 像素的 PNG 图像) ;
- 帮助命令: `help 命令名` (如 `help show`, 查看 show 命令的使用说明) 。

2.4.3 命令行快捷技巧

- 命令自动补全: 输入命令前缀后按 Tab 键, PyMOL 将自动补全命令或列出可选参数;
- 命令历史: 按上下箭头键可查看之前输入的命令, 方便重复使用;
- 批量执行命令: 将多条命令写入.txt 文件 (后缀改为.pml), 通过 `@文件名.pml` 命令可批量执行。

第 3 章: 分子模型的显示与美化

3.1 核心显示模式详解

PyMOL 提供多种分子显示模式，可根据需求选择单独使用或组合使用，以下是最常用的几种模式：

3.1.1 球棍模型 (Sticks)

- 特点：以“球”代表原子，“棍”代表化学键，清晰展示分子的原子连接方式和空间结构；
- 适用场景：小分子化合物、简单蛋白质结构的细节展示；
- 操作方式：菜单栏 **Display > Representation > Sticks**，或命令行 **show sticks, 对象名**；
- 自定义设置：通过 **set stick_radius, 0.15**（调整棍的半径）、**set stick_color, blue**（设置棍的颜色）修改样式。

3.1.2 球体模型 (Spheres)

- 特点：以球体代表原子，原子大小与范德华半径成正比，直观展示分子的空间占位；
- 适用场景：分子表面相互作用、原子堆积情况分析；
- 操作方式：菜单栏 **Display > Representation > Spheres**，或命令行 **show spheres, 对象名**；
- 自定义设置：通过 **set sphere_scale, 0.8**（调整球体大小比例）修改样式。

3.1.3 丝带模型 (Ribbons)

- 特点：简化蛋白质结构，以丝带形式展示肽链的二级结构（ α -螺旋显示为螺旋状， β -折叠显示为箭头状）；
- 适用场景：蛋白质整体结构展示、二级结构分析；
- 操作方式：菜单栏 **Display > Representation > Ribbons**，或命令行 **show ribbons, 对象名**；
- 自定义设置：通过 **set ribbon_width, 2.0**（调整丝带宽度）、**set ribbon_color, cyan**（设置丝带颜色）修改样式。

3.1.4 卡通模型 (Cartoon)

- 特点：丝带模型的进阶版本，更形象地展示蛋白质二级结构（ α -螺旋为螺旋带， β -折叠为宽箭头，无规则卷曲为细带）；
- 适用场景：学术论文插图、教学演示（最常用的蛋白质显示模式）；
- 操作方式：菜单栏 **Display > Representation > Cartoon**，或命令行 **show cartoon, 对象名**；

- 自定义设置：通过 `set cartoon_style, 1`（调整卡通样式）、`set cartoon_transparency, 0.3`（设置透明度）修改样式。

3.1.5 表面模型 (Surface)

- 特点：展示分子的溶剂可及表面，通过颜色深浅或不同颜色区分原子类型、电荷分布等；
- 适用场景：分子结合口袋分析、蛋白质 - 配体相互作用展示；
- 操作方式：菜单栏 `Display > Representation > Surface`，或命令行 `show surface, 对象名`；
- 自定义设置：通过 `set surface_transparency, 0.5`（调整透明度）、`set surface_color, gray`（设置表面颜色）修改样式。

3.2 颜色设置技巧

颜色是分子可视化的重要元素，合理的颜色搭配能让分子结构更清晰、更具表现力。

3.2.1 预设颜色方案

- 按原子类型着色（默认）：C（灰色）、H（白色）、O（红色）、N（蓝色）、S（黄色）、P（橙色）；
- 按残基类型着色：不同氨基酸残基显示为不同颜色（如丙氨酸为粉色、亮氨酸为黄色），适合蛋白质结构分析；
- 按链着色：多链蛋白质中，每条链显示为不同颜色，方便区分链的分布；
- 操作方式：菜单栏 `Display > Color`，选择对应的着色方案，或命令行 `color 方案名, 对象名`（如 `color chain, 1AKE`）。

3.2.2 自定义颜色

- 使用颜色名称：PyMOL 支持常见颜色名称（如 red、blue、green、cyan、magenta 等），例如 `color magenta, resi 50`（将 50 号残基设为品红色）；
- 使用 RGB 值：通过 RGB 值精确设置颜色，格式为 `color [r, g, b], 选择对象`（如 `color [0.2, 0.8, 0.5], ligand`，将配体设为淡绿色）；
- 渐变颜色：通过 `spectrum` 命令设置渐变颜色，例如 `spectrum b, blue_white_red, 1AKE`（按 B 因子从蓝到红渐变着色）。

3.3 透明度与光照调整

3.3.1 透明度设置

通过调整透明度，可以突出重点区域，同时展示背景结构，常用场景包括：

- 蛋白质 - 配体复合物：将蛋白质设为半透明 (`set transparency, 0.5, protein`)，配体设为不透明，清晰展示配体在结合口袋中的位置；
- 多分子复合物：不同分子设置不同透明度，区分主次关系；
- 操作方式：菜单栏 `Display > Transparency`，拖动滑块调整整体透明度；或命令行 `set transparency, 数值, 选择对象`（数值 0-1，0 为不透明）。

3.3.2 光照效果调整

光照效果能增强分子的立体感和质感，PyMOL 提供多种光照设置：

- 启用阴影：`set shadow, on`（默认关闭，开启后分子边缘将产生阴影，更具立体感）；
- 调整光源位置：`set light_position, [x, y, z]`（通过坐标调整光源方向，如 `set light_position, [1, 1, 1]`）；
- 启用环境光：`set ambient, 0.3`（调整环境光强度，0 为无环境光，1 为最强）；
- 操作方式：也可通过菜单栏 `View > Lighting` 进行可视化调整。

3.4 高级美化：背景与坐标轴设置

3.4.1 背景设置

- 纯色背景：默认背景为黑色，可改为白色 (`set background, white`)、灰色 (`set background, gray`) 等，适合不同应用场景（如白色背景适合论文插图，黑色背景适合演示）；
- 渐变背景：`set gradient_background, on`（启用渐变背景），并通过 `set background_top, blue`、`set background_bottom, white` 设置上下渐变颜色；
- 操作方式：命令行输入对应命令，或通过 `Edit > Preferences > Display` 进行图形化设置。

3.4.2 坐标轴与网格设置

- 显示 / 隐藏坐标轴：`show axes`（显示）、`hide axes`（隐藏），坐标轴默认显示在视图左下角，可通过 `set axes_position, [0.1, 0.1]` 调整位置；
- 显示 / 隐藏网格：`show grid`（显示）、`hide grid`（隐藏），网格可辅助判断分子的空间位置，通过 `set grid_color, gray` 设置网格颜色；
- 适用场景：教学演示、精确测量分子距离时启用，论文插图中通常隐藏以保持画面简洁。

第 4 章：分子选择与测量分析

4.1 分子选择的常用方法

精确选择分子的特定部分是后续分析操作的基础，PyMOL 提供多种选择方式，满足不同需求：

4.1.1 图形化选择

- 鼠标选择：在视图窗口中，按住 Ctrl 键 + 鼠标左键点击原子、残基或链，即可选中（选中部分将高亮显示）；
- 框选：按住 Ctrl 键 + 鼠标左键拖动，绘制矩形框，框内的分子部分将被选中；
- 菜单选择：通过 **Edit > Select**，选择对应的选择类型（如 Atom、Residue、Chain、Object 等），然后在视图中点击目标对象。

4.1.2 命令行选择（精准高效）

命令行选择通过“选择表达式”实现，格式为 **select 选择集名称, 选择条件**，常用选择条件如下：

- 按对象名选择：**select prot, 1AKE**（选择名为 1AKE 的整个分子对象，命名为 prot）；
- 按链选择：**select chainA, chain A**（选择 A 链，命名为 chainA）；
- 按残基编号选择：**select res10-20, resi 10-20**（选择 10-20 号残基）；
- 按残基名称选择：**select lys, resn LYS**（选择所有赖氨酸残基，残基名称为三字母代码）；
- 按原子类型选择：**select oxygen, elem O**（选择所有氧原子，elem 后接原子符号）；
- 按距离选择：**select near_ligand, within 5 of ligand**（选择配体周围 5Å 范围内的所有原子）；
- 组合选择：**select interface, chain A and within 4 of chain B**（选择 A 链中距离 B 链 4Å 以内的原子，即链间界面）。

4.1.3 选择集的管理

- 查看选择集：选择集将显示在右侧对象管理器中，可通过勾选控制显示 / 隐藏；
- 激活选择集：在命令行中输入选择集名称，即可激活该选择集（后续操作默认针对该选择集）；
- 删除选择集：**delete 选择集名称**，或在对象管理器中右键点击选择集，选择“Delete”。

4.2 关键测量功能

PyMOL 支持分子结构的多种测量，为科研分析提供数据支持，常用测量功能如下：

4.2.1 距离测量 (Distance)

- 功能：测量两个原子之间的直线距离（单位：Å）；
- 操作方式：
 - a. 图形化：**Wizard > Measurement > Distance**，点击两个目标原子，视图中将显示距离数值；
 - b. 命令行：**distance dist1, atom1, atom2**（如 **distance dist_lys_arg, resi 50 and name NZ, resi 60 and name NH1**，测量 50 号赖氨酸 NZ 原子与 60 号精氨酸 NH1 原子的距离）；
- 自定义设置：**set label_size, 20**（调整距离标签字体大小）、**set label_color, red**（设置标签颜色）。

4.2.2 角度测量 (Angle)

- 功能：测量三个原子组成的角度（单位：度）；
- 操作方式：
 - c. 图形化：**Wizard > Measurement > Angle**，依次点击三个原子（中间原子为顶点）；
 - d. 命令行：**angle ang1, atom1, atom2, atom3**（atom2 为顶点）；
- 应用场景：分析化学键角、分子构象变化。

4.2.3 二面角测量 (Dihedral)

- 功能：测量四个原子组成的二面角（扭转角，单位：度）；
- 操作方式：
 - e. 图形化：**Wizard > Measurement > Dihedral**，依次点击四个原子；
 - f. 命令行：**dihedral dih1, atom1, atom2, atom3, atom4**；
- 应用场景：蛋白质主链构象（ ϕ 角、 ψ 角）分析、小分子构象分析。

4.2.4 氢键检测与显示

- 功能：自动检测分子内或分子间的氢键（默认判定标准：供体 - 氢键 - 受体角度 $> 120^\circ$ ，距离 Å）；
- 操作方式：
 - g. 图形化：**Actions > Find > Hydrogen Bonds**，选择氢键供体和受体对象（如蛋白质和配体）；

- h. 命令行: `distance hbonds, donor_object, acceptor_object, mode=2, cutoff=3.5` (mode=2 表示检测氢键, cutoff 为距离阈值) ;
- 自定义设置: `set dash_color, green` (设置氢键虚线颜色) 、 `set dash_radius, 0.05` (调整虚线粗细) 。

4.3 结构分析工具

4.3.1 氨基酸残基特性分析

- 显示残基性质: `label resi 1-50, resn + " " + name` (在 1-50 号残基上显示残基名称和原子名称) ;
- 疏水性分析: `color hydrophobic, resn ALA+VAL+LEU+ILE+PHE+TRP+MET` (将疏水性氨基酸设为特定颜色) , 或使用 `plugin > PymolWiki > Hydrophobicity` 插件生成疏水性图谱。

4.3.2 活性位点分析

- 基于配体筛选: 若已知配体, 通过 `select active_site, within 6 of ligand` 选择配体周围 6Å 的区域作为潜在活性位点;
- 基于口袋检测: 安装“SiteFinder”插件 (`plugin > Plugin Manager > Available Plugins > SiteFinder`) , 点击 `Plugins > SiteFinder > Run SiteFinder` , 自动检测蛋白质表面的口袋结构。

4.3.3 二级结构分析

- 显示二级结构标签: `label cartoon, ss` (在卡通模型上显示二级结构类型, H 为 α -螺旋, E 为 β -折叠, L 为无规则卷曲) ;
- 二级结构统计: 通过 `plugin > PymolWiki > DSSP` 插件 (需先安装 DSSP 软件) , 分析蛋白质二级结构的比例和分布。

第 5 章: PyMOL 脚本编程入门

5.1 为什么需要脚本编程?

PyMOL 的图形化界面适合快速操作, 但在处理以下场景时, 脚本编程更具优势:

- 批量处理: 同时分析多个分子文件 (如批量加载 PDB 文件、统一设置显示样式) ;
- 重复操作: 需要多次执行相同的步骤 (如固定的图像导出流程、特定的分析方法) ;
- 高级定制: 实现图形化界面无法完成的功能 (如自定义分析算法、复杂动画制作) ;

- 可重复性：脚本可保存和分享，确保实验结果的可重复验证。

PyMOL 脚本基于 Python 语言，同时支持 PyMOL 专属命令，无需熟练掌握 Python，只需了解基础语法和 PyMOL 命令即可编写实用脚本。

5.2 脚本文件的创建与执行

5.2.1 脚本文件格式

PyMOL 脚本文件的后缀为 `.pml`，本质是包含一系列 PyMOL 命令和 Python 代码的文本文件，可使用记事本、Notepad++、VS Code 等文本编辑器创建。

5.2.2 脚本执行方式

- 方法 1：在 PyMOL 命令行中输入 `@脚本路径/脚本名.pml`（如 `@D:/pymol_scripts/load_and_display.pml`）；
- 方法 2：在 PyMOL 主界面中，通过 `File > Run Script`，选择脚本文件；
- 方法 3：将脚本文件拖拽到 PyMOL 视图窗口中，自动执行。

5.3 基础脚本示例

示例 1：批量加载分子并设置显示样式

```
# 批量加载 PDB 文件（假设文件位于同一文件夹）
load D:/structures/1AKE.pdb, prot1
load D:/structures/2ABZ.pdb, prot2
load D:/structures/3CDE.pdb, prot3
# 设置所有蛋白质为卡通模型，按链着色
show cartoon, all
color chain, all
# 调整视图和背景
reset view
set background, white
set cartoon_transparency, 0.2
# 导出图像
```

png D:/output/three_proteins.png, width=1000, height=800, dpi=300

示例 2：自动检测氢键并生成分析报告

```
# 加载蛋白质-配体复合物
fetch 1MOL, complex
split_chains complex # 拆分链（假设 A 为蛋白质，B 为配体）
# 选择蛋白质和配体
select protein, chain A
select ligand, chain B
# 检测氢键（供体：蛋白质 N/O 原子，受体：配体 O/N 原子）
distance hbonds, protein and (elem N+O), ligand and (elem O+N), mode=2, cutoff=3.5
show dashes, hbonds
set dash_color, green, hbonds
# 输出氢键信息到文本文件
save_hbonds = open("D:/output/hbond_report.txt", "w")
hbond_list = cmd.get_distance("hbonds", mode="dist")
save_hbonds.write("氢键数量: %d\n" % len(hbond_list))
save_hbonds.write("氢键详情（供体原子 - 受体原子 - 距离Å）: \n")
for i, dist in enumerate(hbond_list):
    donor = cmd.get_model("hbonds and index %d" % (2*i)).atom[0]
    acceptor = cmd.get_model("hbonds and index %d" % (2*i+1)).atom[0]
    save_hbonds.write("%s_%s_%d - %s_%s_%d - %.2f\n" % (
        donor.chain, donor.resn, donor.resi,
        acceptor.chain, acceptor.resn, acceptor.resi,
        dist
    ))
save_hbonds.close()
# 导出带氢键的图像
png D:/output/complex_hbonds.png, width=800, height=600
```

5.4 Python 代码在脚本中的应用

PyMOL 脚本支持直接嵌入 Python 代码，通过 Python 的循环、条件判断、文件操作等功能，实现更复杂的逻辑。

示例 3：使用 Python 循环批量处理分子

```
# 导入 Python 模块
import os

# 定义分子 PDB ID 列表
pdb_ids = ["1AKE", "1MOL", "2ABZ", "3CDE", "4FGH"]

# 循环下载并处理每个分子
for pdb in pdb_ids:
    # 下载分子
    fetch pdb, object_name=pdb

    # 设置显示模式
    show cartoon, pdb

    color chain, pdb

    # 居中显示
    center pdb

    # 导出图像（文件名以 PDB ID 命名）
    png os.path.join("D:/output", f"{pdb}_cartoon.png"), width=800, height=600

    # 隐藏当前分子，处理下一个
    hide all, pdb

# 显示所有分子并导出汇总图像
show cartoon, all

reset view

png D:/output/all_proteins_summary.png, width=1200, height=800
```

示例 4：自定义函数计算原子间平均距离

```

# 定义 Python 函数：计算两个选择集之间的平均距离

def average_distance(sel1, sel2):
    # 获取两个选择集中的所有原子对距离
    distances = cmd.get_distance("(" + sel1 + ")", "(" + sel2 + ")", mode="dist")
    if not distances:
        return 0.0
    # 计算平均值
    avg_dist = sum(distances) / len(distances)
    return avg_dist

# 加载分子并应用函数
fetch 1AKE, prot
select helix, ss H # 选择α-螺旋区域
select sheet, ss E # 选择β-折叠区域

# 计算螺旋与折叠之间的平均距离
avg_dist = average_distance("helix", "sheet")
print(f"α-螺旋与β-折叠的平均距离: {avg_dist:.2f} Å")

# 将结果显示在视图中
label center(helix), f"平均距离: {avg_dist:.2f}Å"

set label_size, 25
set label_color, red

```

5.5 脚本调试与优化技巧

- 逐行执行：在命令行中逐行输入脚本命令，观察每一步的执行结果，定位错误位置；
- 打印信息：使用 `print()` 函数（Python）或 `echo` 命令（PyMOL）输出关键变量值，辅助判断逻辑是否正确；
- 注释代码：对脚本中的关键步骤添加注释（`#`后面的内容为注释），方便后续维护和调试；
- 简化脚本：将复杂脚本拆分为多个功能模块，逐个测试后再整合；
- 参考官方文档：PyMOL 官方网站提供详细的脚本命令手册和示例，遇到问题可查阅 (<https://pymolwiki.org/>)。

第 6 章：高级功能与插件应用

6.1 分子动画制作

PyMOL 支持制作分子结构的动态演示动画，适合学术报告、教学演示等场景，常用动画功能如下：

6.1.1 基础动画：视图旋转

```
# 新建动画轨道
mset 1-100 # 设置动画帧数（1 到 100 帧）
# 定义关键帧（第 1 帧和第 100 帧的视图）
mview store, 1 # 存储第 1 帧视图（默认视图）
turn x, 180, wait=0 # 绕 X 轴旋转 180 度
mview store, 100 # 存储第 100 帧视图
# 插值生成中间帧
mview interpolate
# 播放动画
mpyplay 1-100, loop=1 # loop=1 表示循环播放
# 导出动画（需安装 ffmpeg）
mp4 D:/output/rotation_animation.mp4, fps=10 # fps 为帧率
```

6.1.2 高级动画：分子构象变化

通过加载不同构象的分子文件，制作构象变化动画：

```
# 加载两个不同构象的分子（假设为构象 A 和构象 B）
load conformer_A.pdb, confA
load conformer_B.pdb, confB
hide all
# 新建动画轨道（200 帧）
mset 1-200
# 第 1-50 帧：显示构象 A
```

```
show cartoon, confA
mview store, 1
mview store, 50
# 第 51-150 帧: 构象 A 渐变到构象 B (需要对齐分子)
align confB, confA # 对齐两个构象
show cartoon, confB
set cartoon_transparency, 1, confB # 构象 B 初始完全透明
for frame in range(51, 151):
    transparency = 1 - (frame - 50)/100 # 逐渐降低构象 B 的透明度
    set cartoon_transparency, transparency, confB
    mview store, frame
# 第 151-200 帧: 显示构象 B
set cartoon_transparency, 0, confB
mview store, 151
mview store, 200
# 导出动画
mp4 D:/output/conformer_change.mp4, fps=15
```

6.2 插件安装与管理

PyMOL 的插件系统极大地扩展了其功能，用户可根据需求安装各类插件（如结构分析、图像美化、数据可视化等）。

6.2.1 插件安装方法

- 方法 1: 通过 Plugin Manager 安装 (推荐)
 - i. 打开 PyMOL, 点击 **Plugin > Plugin Manager**;
 - j. 在弹出的窗口中, 切换到 **Available Plugins** 标签, 浏览可用插件;
 - k. 选择需要安装的插件 (如 SiteFinder、DSSP、PyMOL-2D), 点击 **Install**, 等待安装完成;
 - l. 安装完成后, 插件将出现在 **Plugin** 菜单中。
- 方法 2: 手动安装本地插件
 - m. 从 PyMOL 官网或第三方网站下载插件文件 (通常为.py 格式);

- n. 打开 PyMOL，点击 **Plugin > Install New Plugin**；
- o. 选择下载的插件文件，点击 **Open**，按照提示完成安装。

6.2.2 常用必备插件推荐

- SiteFinder: 蛋白质活性位点（口袋）自动检测，适用于药物设计中的靶点识别；
- DSSP: 蛋白质二级结构分析，可生成详细的二级结构报告（如螺旋、折叠的位置和长度）；
- PyMOL-2D: 将 3D 分子结构转换为 2D 平面结构，方便论文插图和分子结构展示；
- LigPlot+: 生成蛋白质 - 配体相互作用图，清晰展示氢键、疏水作用等相互作用类型；
- APBS Tools: 计算分子的静电势分布，并通过表面颜色直观展示（需先安装 APBS 软件）；
- PyMOL Vaccine: 疫苗设计辅助工具，支持抗原表位预测、疫苗结构分析。

6.3 分子对接结果分析

分子对接是药物研发的核心步骤，PyMOL 可用于对接结果的可视化和分析，判断配体与受体的结合模式是否合理。

6.3.1 加载对接结果

对接软件（如 AutoDock、Vina、Schrödinger）的输出文件通常为 PDB 格式，包含受体和多个配体构象，加载方法：

```
# 加载受体和对接后的配体
load receptor.pdb, receptor
load docking_results.pdb, ligands
# 拆分多个配体构象（若输出文件中多个构象合并）
split_states ligands, prefix=ligand_ # 拆分后命名为 ligand_1, ligand_2, ...
```

6.3.2 结合模式分析

1. 显示受体和配体：

```
show cartoon, receptor
```

```
color gray, receptor
set transparency, 0.5, receptor # 受体半透明, 突出配体
show sticks, ligand_1 # 显示排名第一的配体构象
color yellow, ligand_1
```

1. 检测氢键和疏水作用:

```
# 氢键检测
distance hbonds, receptor and (elem N+O), ligand_1 and (elem O+N), mode=2, cutoff=3.5
show dashes, hbonds
set dash_color, green, hbonds
# 疏水作用检测 (通过距离判断, 非极性原子之间距离<4.0Å)
select receptor_hydrophobic, receptor and resn ALA+VAL+LEU+ILE+PHE+TRP+MET and
elem C
select ligand_hydrophobic, ligand_1 and elem C
distance hydrophobic_interactions, receptor_hydrophobic, ligand_hydrophobic, cutoff=4.0
show dashes, hydrophobic_interactions
set dash_color, orange, hydrophobic_interactions
set dash_radius, 0.03, hydrophobic_interactions
```

1. 配体构象对比:

```
# 显示前 3 个配体构象, 不同颜色区分
show sticks, ligand_1
color red, ligand_1
show sticks, ligand_2
color blue, ligand_2
show sticks, ligand_3
color green, ligand_3
# 对齐配体构象 (以 ligand_1 为参考)
align ligand_2, ligand_1
align ligand_3, ligand_1
```

```
# 显示配体的 RMSD (需安装 rmsd 插件)
plugin load rmsd.py
rmsd ligand_2, ligand_1 # 计算 ligand_2 与 ligand_1 的 RMSD
```

6.4 高质量图像导出技巧

PyMOL 生成的图像常用于学术论文、海报、报告等，需保证图像的清晰度、美观度和专业性，以下是关键技巧：

6.4.1 图像分辨率设置

- 论文插图：通常要求 300 dpi 以上，尺寸根据期刊要求调整（如单栏图宽度约 8-10 cm，双栏图约 16-18 cm）；
- 命令行设置：`png 输出文件名.png, width=1200, height=900, dpi=300`（width 和 height 控制像素，dpi 控制分辨率）；
- 矢量图导出：对于需要缩放的图像（如海报、PPT），建议导出为 SVG 格式（`save 文件名.svg`），矢量图可无限缩放而不失真。

6.4.2 图像美化技巧

- 背景选择：论文插图优先使用白色背景（`set background, white`），演示可使用黑色背景增强立体感；
- 去除冗余元素：隐藏坐标轴、网格、对象管理等（`hide axes, all`、`hide grid, all`），保持画面简洁；
- 颜色搭配：避免使用过于鲜艳或相近的颜色，确保分子各部分对比清晰（如蛋白质用灰色，配体用黄色，氢键用绿色）；
- 光照优化：开启阴影（`set shadow, on`）、调整环境光（`set ambient, 0.4`）和漫反射光（`set diffuse, 0.6`），增强分子质感。

6.4.3 批量导出与格式转换

- 批量导出：通过脚本循环导出多个分子或不同视角的图像（参考第 5 章脚本示例）；
- 格式转换：若需将 PNG 转换为 TIFF（期刊常用格式），可使用图像编辑软件（如 Photoshop、GIMP），或通过 PyMOL 命令 `tiff 输出文件名.tiff` 直接导出。

第 7 章：常见问题与实战案例

7.1 常见问题解答 (FAQ)

问题 1: PyMOL 加载 PDB 文件后, 分子显示不完整或出现错误?

- 原因: PDB 文件可能存在格式错误、缺失原子或残基;
- 解决方案:
 - a. 检查 PDB 文件完整性, 可通过 RCSB PDB 网站 (<https://www.rcsb.org/>) 重新下载该分子的 PDB 文件;
 - b. 使用 PyMOL 命令 `remove solvent` 去除溶剂分子, `remove hetero` 去除杂原子 (如金属离子、小分子配体), 只保留蛋白质主链;
 - c. 若仍有错误, 使用 `repair` 命令修复分子结构 (需安装 `repair` 插件)。

问题 2: 如何提高 PyMOL 的运行速度?

- 原因: 分子结构复杂 (如大分子蛋白质、多分子复合物)、显示质量过高或计算机配置较低;
- 解决方案:
 - d. 降低显示质量: `set quality, low` (处理复杂分子时), 完成后再切换为高质量;
 - e. 隐藏不必要的部分: `hide all, solvent` (隐藏溶剂)、`hide lines, all` (隐藏线条模式);
 - f. 关闭阴影和抗锯齿: `set shadow, off`、`set antialias, off`;
 - g. 增加计算机内存 (建议 8GB 以上), 使用 64 位操作系统和 64 位 PyMOL 版本。

问题 3: 如何在 PyMOL 中添加文字注释?

- 方法 1: 图形化操作: `Edit > Label > Add Label`, 点击视图中的目标位置, 输入注释文字;
- 方法 2: 命令行操作: `label 选择对象, "注释文字"` (如 `label ligand, "Inhibitor"`, 在配体上添加 "Inhibitor" 注释);
- 自定义文字样式: `set label_size, 20` (字体大小)、`set label_color, blue` (字体颜色)、`set label_font, Arial` (字体类型)。

问题 4: 如何将 PyMOL 图像插入 Word、PPT 等文档?

- 导出高质量图像: 优先导出 PNG (300 dpi 以上) 或 SVG 格式;
- 插入 Word: 打开 Word 文档, `插入 > 图片 > 此设备`, 选择导出的图像, 可通过 "图片格式" 调整大小和亮度;

- 插入 PPT: 插入 > 图片 > 此设备, 插入后右键点击图像, 选择“设置图片格式”, 勾选“随对象变化而变化”, 确保缩放时不失真。

问题 5: 如何在不同电脑间同步 PyMOL 的设置和插件?

- 找到 PyMOL 配置文件夹:
 - Windows: C:\Users\用户名\.pymol;
 - macOS: /Users/用户名/.pymol;
 - Linux: /home/用户名/.pymol;
- 复制配置文件夹中的 `pymolrc.pml` (自定义命令文件)、`plugins` 文件夹 (插件) 到另一台电脑的对应目录, 重启 PyMOL 即可同步设置。

7.2 实战案例 1: 蛋白质结构可视化与论文插图制作

案例目标

加载一个蛋白质 (PDB ID: 1AKE), 设置卡通模型显示, 突出 α -螺旋和 β -折叠, 添加活性位点标注, 导出符合期刊要求的高质量图像。

操作步骤

1. 加载蛋白质并设置基础显示:

```
fetch 1AKE, protein # 从 PDB 数据库下载 1AKE
show cartoon, protein # 卡通模型显示
color ss, protein # 按二级结构着色 ( $\alpha$ -螺旋红色,  $\beta$ -折叠黄色, 无规则卷曲灰色)
reset view # 重置视图
set background, white # 白色背景
set cartoon_width, 1.5 # 调整丝带宽度
set cartoon_transparency, 0.1 # 轻微透明, 增强立体感
```

1. 识别并标注活性位点 (假设活性位点包含 100-120 号残基):

```
select active_site, resi 100-120 # 选择活性位点残基
```

```
show sticks, active_site # 球棍模型显示活性位点
color magenta, active_site # 品红色突出
label center(active_site), "Active Site" # 添加注释
set label_size, 22
set label_color, darkblue
```

1. 优化光照和图像质量:

```
set shadow, on # 开启阴影
set ambient, 0.3 # 环境光强度
set diffuse, 0.7 # 漫反射光强度
set specular, 0.2 # 镜面反射光强度
```

1. 导出图像:

```
png 1AKE_active_site.png, width=1000, height=800, dpi=300 # 300 dpi, 适合论文
svg 1AKE_active_site.svg # 矢量图, 适合缩放使用
```

最终效果

图像显示蛋白质的整体二级结构（红色 α -螺旋、黄色 β -折叠），活性位点以品红色球棍模型突出，添加清晰的文字注释，白色背景无冗余元素，符合学术论文插图要求。

7.3 实战案例 2：蛋白质 - 配体复合物相互作用分析

案例目标

加载蛋白质 - 配体复合物（PDB ID: 1MOL），分析配体与蛋白质的结合模式，检测氢键和疏水作用，生成相互作用图。

操作步骤

1. 加载复合物并拆分链:

```
fetch 1MOL, complex # 下载复合物 (1MOL 为 HIV 蛋白酶与抑制剂复合物)
split_chains complex # 拆分链, A 和 B 为蛋白酶链, C 为配体
select protein, chain A+B # 选择蛋白质 (两条链)
select ligand, chain C # 选择配体
```

1. 设置显示模式:

```
show cartoon, protein # 蛋白质卡通模型
color gray, protein
set transparency, 0.6, protein # 半透明显示
show sticks, ligand # 配体球棍模型
color yellow, ligand
show spheres, ligand and elem O+N # 配体的氧、氮原子球体显示, 突出极性原子
set sphere_scale, 0.5, ligand
```

1. 检测并显示相互作用:

```
# 氢键检测 (蛋白质为供体/受体, 配体为受体/供体)
distance hbonds, protein and (elem N+O), ligand and (elem O+N), mode=2, cutoff=3.2
show dashes, hbonds
set dash_color, green
set dash_radius, 0.06
set dash_gap, 0.1 # 氢键虚线间隙

# 疏水作用检测 (蛋白质非极性残基与配体非极性原子)
select protein_hydrophobic, protein and resn ALA+VAL+LEU+ILE+PHE+TRP+MET+PRO
and elem C
select ligand_hydrophobic, ligand and elem C
distance hydrophobic, protein_hydrophobic, ligand_hydrophobic, cutoff=4.0
show dashes, hydrophobic
set dash_color, orange
set dash_radius, 0.04
```

```
set dash_style, 2 # 疏水作用用虚线表示
```

1. 标注关键相互作用:

```
# 显示参与氢键的蛋白质残基名称  
label hbonds and protein, resn + " " + resi  
set label_size, 18  
set label_color, darkred  
# 导出相互作用图  
png 1MOL_interactions.png, width=900, height=700, dpi=300
```

分析结果

图像清晰展示了配体在蛋白质活性口袋中的结合位置，绿色虚线表示氢键（配体与蛋白质残基形成 3-4 个氢键），橙色虚线表示疏水作用，配体的极性原子与蛋白质的极性残基形成氢键，非极性部分与蛋白质的疏水残基形成疏水作用，说明该结合模式稳定，符合抑制剂的作用机制。

第 8 章：PyMOL 资源与学习进阶

8.1 官方资源

- PyMOL 官网: <https://pymol.org/2/>, 提供软件下载、官方文档、教程和最新资讯;
- PyMOL Wiki: <https://pymolwiki.org/>, 包含详细的命令手册、脚本示例、插件介绍和常见问题解答, 是最权威的学习资源;
- PyMOL 邮件列表: <https://sourceforge.net/projects/pymol/lists/pymol-users>, 可与全球 PyMOL 用户和开发者交流问题。

8.2 第三方学习资源

- 视频教程:
 - YouTube: 搜索 “PyMOL Tutorial”, 有大量免费视频 (如 “PyMOL for Beginners” 系列);
 - B 站: 国内用户可搜索 “PyMOL 教程”, 多个 UP 主分享了从入门到高级的实战视频;
- 书籍:

- 《PyMOL Essentials》（作者：Thomas Holder）：官方推荐入门书籍，详细介绍 PyMOL 核心功能；
- 《Molecular Visualization with PyMOL》（作者：Jesper Sørensen）：侧重实战应用，包含大量科研案例；
- 博客与论坛：
 - 科研者之家：<https://www.home-for-researchers.com/>，有 PyMOL 专题文章和用户交流区；
 - Stack Overflow：<https://stackoverflow.com/>，搜索 PyMOL 相关问题，获取专业解答。

8.3 进阶学习方向

- 脚本编程进阶：学习 Python 高级语法，结合 PyMOL API 开发自定义工具（如自动化分析流程、批量图像处理）；
- 分子动力学模拟结果分析：使用 PyMOL 加载分子动力学模拟轨迹文件（如 GROMACS、AMBER 的输出文件），可视化分子构象变化、氢键动力学等；
- 机器学习与 PyMOL 结合：利用 Python 机器学习库（如 Scikit-learn、TensorFlow）处理 PyMOL 分析数据，实现结构 - 活性关系预测、活性位点筛选等；
- 插件开发：学习 PyMOL 插件开发规范，编写自定义插件，满足特定科研需求，并分享给社区。

8.4 社区交流与贡献

- 加入 PyMOL 用户社区：通过邮件列表、论坛、社交媒体（如 Twitter、知乎）与其他用户交流经验，分享脚本和插件；
- 参与开源项目：PyMOL 是开源软件，可在 GitHub (<https://github.com/schrodinger/pymol-open-source>) 上贡献代码、修复 bug；
- 分享学习成果：撰写博客文章、录制教程视频，或在学术会议上分享 PyMOL 应用经验，帮助更多学习者。

后记

PyMOL 作为一款强大的分子可视化工具，其学习曲线虽有一定坡度，但只要掌握基础操作、熟悉核心功能，并通过实战案例不断练习，就能逐步灵活运用。本手册从入门到进阶，覆盖了 PyMOL 的主要功能和应用场景，希望能为读者提供清晰的学习路径。

分子可视化是科研工作的重要组成部分，优秀的可视化结果不仅能帮助我们更好地理解分子结构与功能，还能提升学术成果的表现力和影响力。建议读者在学习过程中，结合自身的科研需求，多动手实践、多探索技巧，将 PyMOL 真正融入到科研工作中。

如果在使用过程中遇到问题，不妨查阅官方文档、参与社区交流，或回顾本手册的相关章节。随着 PyMOL 版本的更新和功能的扩展，也请保持学习的热情，持续关注最新的技术动态。祝愿每位读者都能成为 PyMOL 使用高手，在科研道路上取得更多成果！

（注：文档部分内容可能由 AI 生成）