arXiv:2301.04492v1 [physics.chem-ph] 11 Jan 2023

大规模分子动力学模拟数据的快速构象聚类*

Simon Hunkler,[1] Kay Diederichs,[1] Oleksandra Kukharenko,[2, †] 和 Christine Peter[1, ‡]

[1] *Department of Chemistry, University of Konstanz*

[2] *Theory Department, Max Planck Institute for Polymer Research*（日期：2023年1月12日）

我们提出了一种无监督数据处理工作流程，专为快速对长分子动力学模拟轨迹进行构象聚类而设计。在这一方法中，我们将两种降维算法（cc 分析和 encodermap）与一种基于密度的空间聚类算法（HDBSCAN）相结合。该方案充分发挥了这三种算法的优势，同时避免了各自的大部分缺点。在此，我们首次将 cc 分析算法应用于分子模拟数据。encodermap 通过提供一种高效处理大量数据并将其分配到各簇的方法，补充了 cc 分析。该流程的主要目标是在保持所发现簇具有清晰构象特征的同时，尽可能多地为给定轨迹中的帧分配簇。在实践中，我们通过采用迭代聚类方法，并在最终簇分配中使用可调的、基于均方根偏差的判据来实现这一目标。这样既能找到密度不同的簇，也能找到结构相似性程度不同的簇。我们借助四个测试系统展示了这一聚类工作流程的能力和性能：Trp-cage蛋白的野生型和耐热突变体（TC5b 和 TC10b）、NTL9 以及 Protein B。这些系统各自给该方案带来了不同的挑战，综合来看，它们很好地展现了所提方法的优势，以及使用该方法时可能遇到的困难。

I. 引言

在过去几十年里，随着计算机能力不断增强，分子动力学（MD）领域的研究人员得以获取越来越长的轨迹，从而获得越来越大量的数据。专门用于生成分子动力学轨迹的超级计算机（Anton [1]和 Anton 2 [2]）问世，是这一发展历程中最新的里程碑。此外，新的采样方法 [3, 4]以及 Fold- ing@home [5], 等分布式计算项目，也促使生成的模拟轨迹数量大幅增加。随着数据量不断增长，强大的分析工具对于处理数据并理解底层系统及过程至关重要。

使用无监督机器学习方法分析分子模拟数据的应用迅速增加。最常用的两类分析技术是聚类和降维（DR）算法。它们有助于找到能够保留原始数据重要特征的低维子空间，并根据给定的相似性度量/指标对数据进行分组，从而更好地概览和理解数据。实际应用中，聚类和降维方法大多结合使用。降维算法大致可分为：线性方法（最著名的是主成分分析（PCA） [6, 7] 和时间滞后独立成分分析（TICA） [6, 7] ）、非线性方法（核方法和非线性主成分分析、多维尺度分析（MDS）以及基于MDS的方法，如sketch-map等距映射扩散映射或 UMAP等），以及基于自编码器的方法，例如（encodermap时间自编码器变分自编码器和高斯混合变分自编码器）。

在聚类算法方面，同样有许多不同的方法：K均值 [22, 23], 谱聚类 [24], DBSCAN [25],密度峰值聚类 [26], CNN聚类 [27], 基于均方根偏差（RMSD）的聚类 [28], 基于神经网络的 VAMPnets [29], 等。关于常用于分析分子动力学模拟数据的无监督机器学习方法的全面综述，请参阅文献 30。

即使仅从这份并不完整的现有方法清单中，也显而易见，聚类方法和降维方法种类繁多。这些方法各有优缺点。然而，如何成功运用所列方法处理具有高空间和时间分辨率的模拟数据，仍面临诸多挑战。这些挑战可能与超参数的恰当选择有关（例如降维方法的维数、某些聚类算法所预期状态的数量、神经网络架构、不同的截断阈值、相关时间等），也可能与昂贵的优化步骤或可同时处理的数据量有关。本研究提出了一种数据处理方案，将三种不同算法整合到一个工作流程中，以构建强大的聚类机制。该方案应对了上述若干挑战：它能够确定适当的数据较低维度，无需预先提供预期状态数量的信息，

* 版权所有 2023年 Hunkler，Peter。本文依据知识共享署名（CC BY）许可协议发布。

† kukharenko@mpip-mainz.mpg.de

‡ christine.peter@uni-konstanz.de

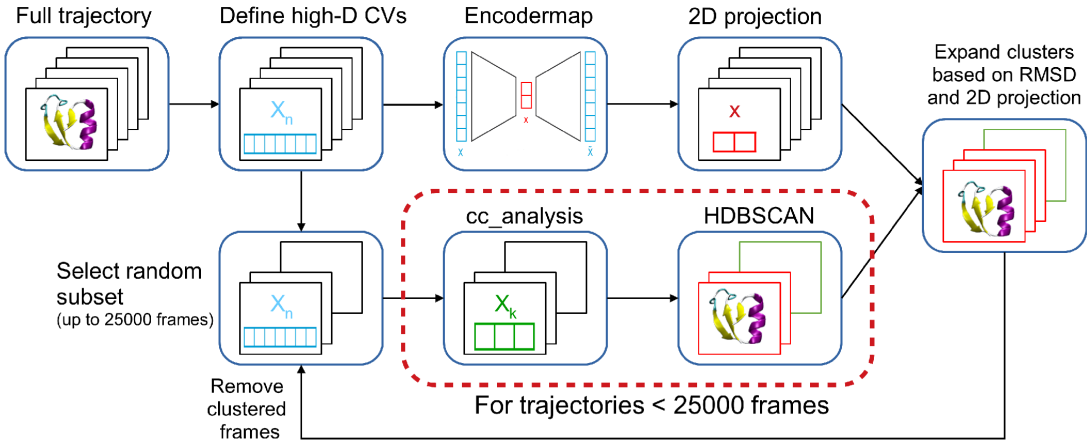


图1. 本文介绍的数据处理流程。

而且能够快速处理维数极高、观测值数量庞大的分子动力学轨迹。该方案专为从长时间的分子模拟数据中识别构象簇而设计（图1）。

我们使用两种不同的降维算法，即一种称为“cc 分析”的算法和 encodermap 算法。cc 分析方法属于基于多维尺度分析的技术系列，最初用于晶体学数据分析 [31, 32]。本文首次将其用于投影蛋白质构象数据。cc 分析空间通常需要超过两个维度（对于本文所示的系统，维度为 10 至 40），而可同时高效投影的数据量受可用内存限制（在配备 72 GB 内存的工作站上约为 50000 帧）。为处理更长的轨迹并获得二维表示，我们使用第二种降维算法——encodermap [33]。不过，它的损失函数由两部分组成：自动编码器损失和类 MDS 距离损失，后者使生成的二维表示具有可解释性。此外，训练好 encodermap 网络后，可以利用编码器函数，以极高的效率将数据投影到二维映射图中。我们使用 encodermap 将数据投影到二维空间，并快速将额外的数据点分配到 cc 分析空间中定义的簇。最后，我们使用 HDBSCAN 算法 [34] 对 cc 分析空间中的数据进行聚类，然后在二维 encodermap 空间中可视化所得簇。HDBSCAN 是一种结合了密度聚类和层次聚类的方法，能够高效处理密度各异的簇，忽略稀疏区域，并且所需的超参数极少。我们以一种非经典的迭代方式应用该方法，通过改变 RMSD 截断值，提取相似程度不同的蛋白质构象。

这三种算法结合使用，使我们能够发挥各自的优势，同时避免单一方法的缺点。接下来，我们将展示该方案在野生型和突变型 Trp-cage 的长时间分子动力学轨迹上的表现——这些轨迹包含天然态和错误折叠的亚稳态（208 μs 和 3.2 μs 的长时间模拟）；此外，还有 NTL9 的超大规模模拟（1877 μs）；以及 Protein B 的模拟，其中只有少量模拟数据（5%）处于折叠态（104 μs）。

II. 方法

A. cc 分析

为实现降维，我们采用了参考文献 31、32 中介绍的 cc 分析。该算法最初是为分析晶体学数据而开发的；在某些实验条件下，噪声的存在以及观测值缺失会给数据处理带来挑战。该方法将数据集间由随机误差引起的影响与由系统差异引起的影响区分开来，并通过将高维输入特征表示为低维空间中的向量，揭示这些特征之间的关系。鉴于这一特性，我们预计它非常适用于蛋白质模拟数据；在这类数据中，人们希望忽略随机波动造成的差异，并根据系统差异对构象进行区分。本文将展示，这一假设是正确的。

cc 分析算法属于 MDS方法家族 [10]。其主要区别在于，它最小化高维描述符对的皮尔逊相关系数与表示这些描述符的低维向量点积之间的平方差之和（见公式 (1)）。该过程将向量置于低维空间中的单位球面上。高维特征之间的系统差异会导致表示它们的向量在角度方向上有所不同，而数据点之间纯粹随机的差异则会导致这些向量在角度方向相同的情况下具有不同的长度。该算法通过迭代方式（例如使用 L-BFGS [35], ）最小化

表达式

$$\Phi(\mathbf{x}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (r_{ij} - x_i \cdot x_j)^2 \tag{1}$$

关于$\mathbf{x}$的函数，其中x是由 N 个低维向量组成的列向量 $\{_k\}$。其中， r_{ij} 是高维空间中描述符 i 和 j 之间的相关系数， $x_i \cdot x_j$ 表示低维空间中代表数据的单位向量 x_i 和 x_j 的点积； N 是观测值的数量，例如蛋白质构象。cc 分析的输出是 N 个低维向量x $\{_k\}$以及$\mathbf{x}\mathbf{x}^T$ 矩阵的特征值。

为了理解为什么这是一种合理的方法，可以考虑通过对矩阵 $\mathbf{r}$ =$\{r_{ij}\}$进行特征分析，以代数方式求解公式 (1) 的最小二乘解。在这种情况下，需要求解

$$\mathbf{x}\mathbf{x}^T = \mathbf{r}$$

其中，$\mathbf{r}$ 是相关系数矩阵 r_{ij}。随后可以利用 n 个最大的特征值/特征向量对（对应于最大特征值的特征向量）重构 N 个向量 x_i，这些向量位于一个 n 维单位球面上。输入数据之间的系统差异由此通过这个低维球面中不同的角方向体现出来。这种近似是有意义的，因为通常皮尔逊相关系数可以表示为两个向量的点积（先减去均值，再除以标准差，将向量缩放为单位长度），并且等于两者夹角的余弦。因此，在理想情况下，$\sum_{i,j}^{N} x_i \cdot x_j$ 可以精确重构高维相关系数矩阵，并且公式 (1) 中的 $\Phi(\mathbf{x})$ 将等于零。

向量的长度不如它们之间的夹角重要。后者具有内在含义：两个相关系数为零的高维特征向量会被投影为相对于原点呈 90° 角的单位向量；而两个相关系数为一的特征向量所对应的夹角则为零度。

尽管 cc 分析方法具有普适性，但迄今为止，它仅应用于晶体学数据 [36, 37]) 和蛋白质序列分组 [38]。本文首次将 cc 分析应用于蛋白质模拟数据分析。

B. encodermap

为了加速大型数据集（例如来自大规模模拟的数据集）的处理，除了 cc 分析之外，我们还采用另一种降维技术——编码器映射。该方法由 Lemke 和 Peter [33]开发，在此用于快速将数据点分配到各个簇中，具体内容将在第 II D 节中详细说明。该方法结合了神经网络自动编码器 [17] 与 MDS 的优势，并采用 sketch-map 算法 [12] 的损失函数作为其损失函数（图 2）。这种结合特别适合将大型模拟数据快速投影到二维表示中：sketch-map 损失函数能够只聚焦于构象之间相关的差异（忽略热涨落，并处理由数据高维性导致的较大差异值）。此外，自动编码器方法还可避免 sketch-map 投影中复杂的最小化步骤，并能在极短时间内处理海量数据。

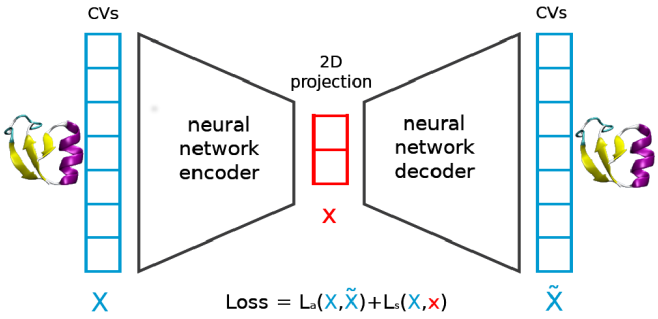


图 2. encodermap 示意图。其架构采用经典的自动编码器，由两个神经网络（编码器和解码器）组成，每个网络的层数相同，且各层的神经元数量相同；两者通过包含两个神经元的瓶颈层相连。除自动编码器损失 $L_a(X, X)$ 外，encodermap 损失还包含一项 sketch-map 损失函数 $L_s(X, x)$，它能提升瓶颈层中所得二维投影的质量（见式 (2)）。

encodermap 损失函数 $L_{encodermap}$ （式中为 (2)），自动编码器损失 L_{auto} （式中为 sketch-map 损失函数 (3)） L_{sketch} （式 (4)），它通过 S 形函数（式 (5)）变换所有距离，从而强调中程距离。

$$L_{encodermap} = k_a L_{auto} + k_s L_{sketch} + Reg, \tag{2}$$

$$L_{auto} = \frac{1}{N} \sum_{i=1}^{N} D(X_i, \tilde{X}_i), \tag{3}$$

$$L_{sketch} = \frac{1}{N} \sum_{i \neq j}^{N} [SIG_h(D(X_i, X_j)) - SIG_l(D(x_i, x_j))]^2, \tag{4}$$

其中， k_a、 k_s 为可调权重， Reg 为用于防止过拟合的正则化项； N 是待投影的数据点数量； $D(\cdot,\cdot)$是点之间的距离， X 是高维输入， x 是低维投影（瓶颈层）； SIG_h 和SIG_l 是形式如式（5）所示的 sigmoid 函数。

$$\mathrm{SIG}_{\sigma,a,b}(D) = 1 - (1 + (2^{\frac{a}{b}} - 1)(\frac{D}{\sigma})^a)^{-\frac{b}{a}}, \tag{5}$$

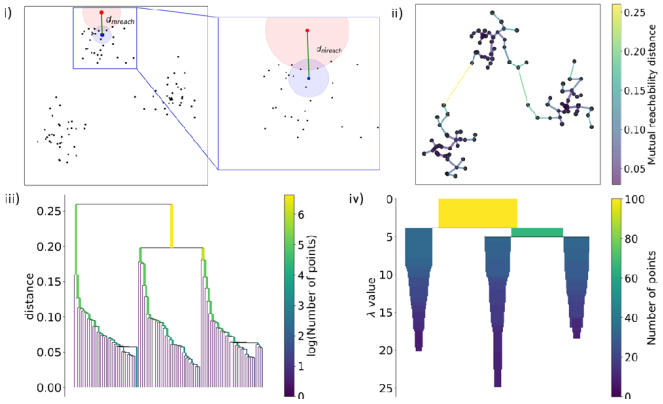


图 3. HDBSCAN 在包含三个簇的玩具数据集上的应用。i）两个点（红色和蓝色）的互可达距离计算示例。红色和蓝色圆圈分别表示两个点到其 5 个最近邻的最远距离。可以看到，红蓝两点之间的距离（绿色线段）大于蓝色圆圈和红色圆圈的半径。因此在这种情况下，绿色线段的距离被选作互可达距离。ii）基于互可达距离的最小生成树。iii）簇层次结构。iv）凝缩聚类。

其中， a、 b 和 σ 是用于定义要保留哪些距离的参数。

C. 基于层次密度的带噪声应用空间聚类（HDBSCAN）

HDBSCAN [34, 39] 可以从两个不同角度来理解：它可以被描述为 J. G. B. Campello 等人提出的一种称为 DBSCAN* 的原始 DBSCAN [25] 算法的新形式的层次化实现 [34]；也可以被表述为单链接聚类的一种稳健版本，并采用一种复杂的方法来获得扁平聚类结果，这种做法由 McInnes 等人提出 [39]。这里我们通过第二种方式进行介绍。

第一步中，算法引入了所谓的互可达距离（MRD）（式（6）），它会转换空间，使稀疏点变得更加稀疏，同时不会显著改变已经密集的点之间的距离。

$$D_{mreach-k}(x_i, x_j) = max\{core_k(x_i), core_k(x_j), D(x_i, x_j)\}, \quad (6)$$

其中， x 是待聚类的点， k 是一个常数，用于确定最近邻点的数量；$core_k(x)$是一个函数，用于求出点 x 与其 k 个最近邻之间的最大距离； $D(\cdot, \cdot)$是两点之间的距离。取三个距离中的最大值作为互可达距离（图 3 i））。

接下来，基于互可达距离构建最小生成树，使用 Prim 算法 [40]（见图 3 ii））。具体做法是从数据集中互可达距离最小的点开始，用一条直线连接这两个点。在后续步骤中，每次都将尚未连接且距离现有树最近的数据点加入树中。

生成最小生成树后，即可构建簇层次结构。首先按距离对树的边进行排序，然后算法遍历这些边，始终合并互可达距离最小的簇。此过程的结果见图 3 iii）。

要从该层次结构中提取扁平聚类，还需要最后一步。在这一步中，通过定义最小簇大小并在每个分裂点检查新形成的簇是否至少包含与最小簇大小相同数量的成员，将簇层次结构凝缩。如果满足条件，则接受新簇；否则，分裂出去的数据点被视为噪声。玩具系统的凝缩树见图 3 iv）。

D. 新聚类工作流程的介绍

本文介绍了一种数据处理流程，我们发现它对于处理大型分子动力学模拟轨迹极为高效。该流程依赖于上文介绍的三种算法。图1给出了示意说明。在这一工作流程中，对给定的数据集进行迭代聚类，直至达到指定数量的数据点被分配到各个簇，或达到最大迭代次数。

图1展示了聚类工作流程中的数据处理步骤顺序。第一步是选择一个高维集体变量（CV）。对于本文展示的所有系统，均选择了 C_α 原子之间的所有两两距离。选定 CV 后，对于包含超过 25,000 帧的轨迹，使用所有数据训练 encodermap。由此得到一个函数，可用于将数据高效地投影到新生成的二维空间中。与此同时，从整个数据集中生成一个随机子集。使用此类子集的原因是 cc 分析降维存在局限性。正如第II A节所述，cc 分析算法使用相关矩阵。这意味着要计算所选 CV（此处为两两 Cα 距离）所有唯一帧对的皮尔逊相关系数，并将其作为 cc 分析的输入。然而，数据集越大，相关系数矩阵也越大，直到由于计算时间过长以及内存问题，处理该矩阵变得不再高效。因此，我们从整个数据集中随机选取最多 25,000 个数据点，构成一个子集。随后在 cc 分析降维中使用该子集，将高维集体变量（本文所述系统为 190 至 1081 维）投影到

低维子空间中（本文所述系统为 20 至 30 维）。通过在 cc 分析特征值中寻找谱隙，确定合适的降维维数。确定 cc 分析空间后，将 HDBSCAN 算法应用于该低维数据，以生成聚类。关于如何选择 cc 分析的维数以及 HDBSCAN 的参数，详见补充信息（SI）S-I 节。

我们在工作流程中使用两种不同的降维算法，原因如下。首先，cc 分析算法用于将较小的子集投影到维度仍然相对较高的子空间中；与 encodermap 的二维投影相比，该子空间保留了更多信息。因此，这个高维子空间非常适合作为聚类空间。数据子集在 cc 分析空间中完成聚类后，我们使用二维 encodermap 空间，将不属于该子集的点分配到已找到的簇中。二维投影非常适合快速分配数据集中其他点，也可用于可视化。此外，encodermap 能够以极高的时间效率投影大型数据集。因此，对于给定数据集生成的二维投影，可以用来规避我们在此处使用 cc 分析算法时的主要缺点：由于内存问题，必须使用数据子集。为避免这一缺点，我们在二维空间中为 cc 分析空间里找到的每个簇构建一个凸包。如果一个尚未分配的点位于凸包内，就计算该点与该簇中心构象之间的 RMSD。若 RMSD 低于给定截断值，则认为该数据点属于该簇；否则，不将其分配到该簇。这个 RMSD 截断值取自第一次聚类迭代中所有簇内平均 RMSD [1] 的加权平均值。我们发现，由于均方根偏差判据基于通过 cc 分析与 HDBSCAN 结合得到的明确构象状态，这一过程产生的簇在结构上相当明确，且簇内均方根偏差较低。不过，也可以通过在分配时采用较大的 RMSD 截断值，识别出仅共享一般结构基序的模糊簇。III B 节介绍了此类模糊簇识别的一个示例。

通过在最后一步引入均方根偏差判据，我们强制聚类过程仅将结构上非常相似的构象纳入相应的簇中。当然，还有各种其他聚类算法可以根据其 RMSD 值对分子动力学数据集进行分组，例如 GROMACS 软件包中的一种实现 [28]。此类基于 RMSD 的聚类算法至少已有 20 年被分子动力学领域采用，并且它们是对分子动力学轨迹进行构象聚类的一个非常明显的选择。它们直接比较分子不同构象中特定原子的位置，然后使用截断值沿轨迹对各个构象进行分组。然而，这些方法通常依赖于给定数据集的完整 RMSD 矩阵。对于较大的轨迹，由于计算时间极长，以及处理如此庞大的矩阵时会产生内存问题，计算这些矩阵几乎变得不可行。在我们的工作流程中，我们只需计算每个簇的中心构象中 C_α 原子的坐标与二维空间中位于相应簇凸包内的数据点之间的 RMSD，从而可以规避这些问题。

如果给定系统的帧数少于约 50,000 帧，原则上也可以省略 encodermap 部分，因为 cc 分析算法能够处理整个数据集。如果选择这种方法，用户可以完全跳过均方根偏差判据，也可以根据 RMSD 截断值，对在 cc 分析空间中找到的簇成员进行接受或拒绝。将 cc 分析算法与 encodermap 算法结合使用的一个优点是，可以即时检查降维步骤。由于聚类是在一个降维空间中进行的，而可视化则在另一个空间中进行，因此，二维空间中狭窄且定义明确的簇表明二维映射图能够很好地区分不同的构象簇，也表明所选的 encodermap 参数恰当。

我们的聚类方案并不十分依赖 encodermap 投影的质量，因为该投影仅用于将其他结构分配到已识别的簇中。聚类本身是在更高维的 cc 分析空间中完成的，而最终的簇分配则使用 RMSD 截断值。该方案对二维映射的唯一要求是：相似构象在映射图中彼此接近。encodermap 整体损失函数中的类 MDS 距离损失部分实现了这一点。

在一次聚类迭代后，未被分配到任何簇的剩余数据点将作为新的数据池，并从中构建新的随机子集。整个循环会重复进行，直到一定数量的数据点被分配到簇中，或完成一定次数的聚类迭代。为确定迭代过程的停止时机，我们采用两种可能的收敛判据：已分配构象所占的比例，或某次迭代中发现的平均簇大小。如果连续数次迭代中未分配数据点的百分比趋于平稳，就会停止聚类过程。由于工作流程的设计，新加入簇的平均簇大小会随着每次迭代而减小。因此，也可以将新加入簇的平均大小，或该属性在连续迭代中的收敛情况，作为停止判据。示例见补充信息的 S-II 节、图 S2。

[1] 簇内平均均方根偏差指所有构象到簇质心的均方根偏差的平均值。

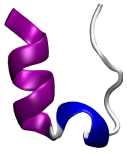
	Trp-cage RE（TC5b）	Trp-cage Anton（TC10b）	NTL9	Protein B
				
轨迹长度（单位： μ秒）	3.2	208	1877	104
帧数	1,577,520	1,044,000	9,389,654	520,250
输入集体变量维数	190	190	703	1081
cc 分析维度数	20	20	20	30
平均迭代时间 在我们的本地工作站上 （见补充信息，第 S-V 节） [分钟]	15	18	55	12
平均迭代时间 在所有使用的 CPU 线程 [分钟]	24×15 $= 360$	24×18 $= 432$	24×55 $= 1320$	24×12 $= 288$
分配给簇的帧 经过 10 次迭代后	60%	33.1%	80.9%	20%
CPU 总时间 所有迭代共计 [分钟]	3600	4320	13200	2880

表 I：本研究分析的蛋白质及聚类方案性能概览。

III. 结果与讨论

A. 用于分析的蛋白质轨迹描述

为了展示所提方案的能力和性能，我们选择了四个测试系统：40 条 Trp-cage 蛋白（TC5b）的温度副本交换（RE）轨迹，这些轨迹在原始 encodermap 论文中进行了分析 [33]；另外三个系统是由 Shaw团队在 Anton 超级计算机上模拟的 Trp-cage（TC10b）、NTL9 和 Protein B 的长轨迹 [42]，并由他们慷慨提供。四个系统列于表 I。对于所有系统，我们都选择 C_α 原子之间的距离作为输入集体变量。

本研究分析的第一种蛋白质是 Trp-cage 系统（TC5b）（Trp-cage RE）。这是一种相对较小的蛋白质（20 个残基），在室温下模拟时具有非常稳定的天然态。由 40 条温度副本交换轨迹组成的组合（温度范围为 300 至 570 K，模拟时间为 3.2 µs，共 1,577,520 帧）产生了种类繁多的结构，其中既包括系统非常稳定、几乎不偏离天然态的轨迹，也包括系统高度无序、会出现高能构象的轨迹。由于预期的簇数量众多，且其规模和密度差异极大，这种构象组合使数据集极为多样，也给分析带来了很大难度。

其次，我们考察了 Lindorff-Larsen 等人对耐热性最高的 Trp-cage 变体 TC10b（Trp-cage Anton）的 K8A 突变体进行的模拟 [42]（208 微秒；1,044,000 帧）。该模拟在 290 K 下进行，产生了与 TC5b 系统的低温副本模拟相比，其轨迹的无序程度要高得多。尽管 TC5b 与 TC10b 的 K8A 突变体的氨基酸序列略有不同，我们仍使用同一个训练好的 encodermap 将两个系统投影到同一张二维映射图上（见图 4 和图 5），因为两个系统的残基数量相同，因此集体变量的维数也相同。这使我们得以展示，将不同系统投影到同一二维空间后，可以非常直观地对它们进行比较。

接下来，我们使用对核糖体蛋白 L9 较大的 N端片段（NTL9）进行的超长模拟来检验聚类方案。该片段由 39 个氨基酸组成，天然态极其稳定；模拟时长为 1877 µs [2]，共 9,389,654 帧 [42]。除了可以展示算法如何处理如此庞大的数据集外，该系统也已由其他几位研究人员进行过研究 [29, 44]。因此，我们可以将结果与他们的发现进行比较。Schwantes 和 Pande [44] 报道了占比极低的状态，其中涉及构成 NTL9 β折叠结构的残基之间发生寄存器移位。这为我们提供了机会，借此展示聚类工作流程能否在同一数据集中识别出规模很大的状态以及占比极低的状态。

最后，我们选择分析 Protein B 的模拟（104 微秒；520,250 帧） [42]。与前述

[2] 我们根据文献 42 中的命名法，使用了轨迹 0、2 和 3。我们没有使用轨迹 1，因为该轨迹对应的拓扑文件在原子的顺序和编号方面与另外三条轨迹略有不同。其他研究人员也报告过这一问题 [43]。

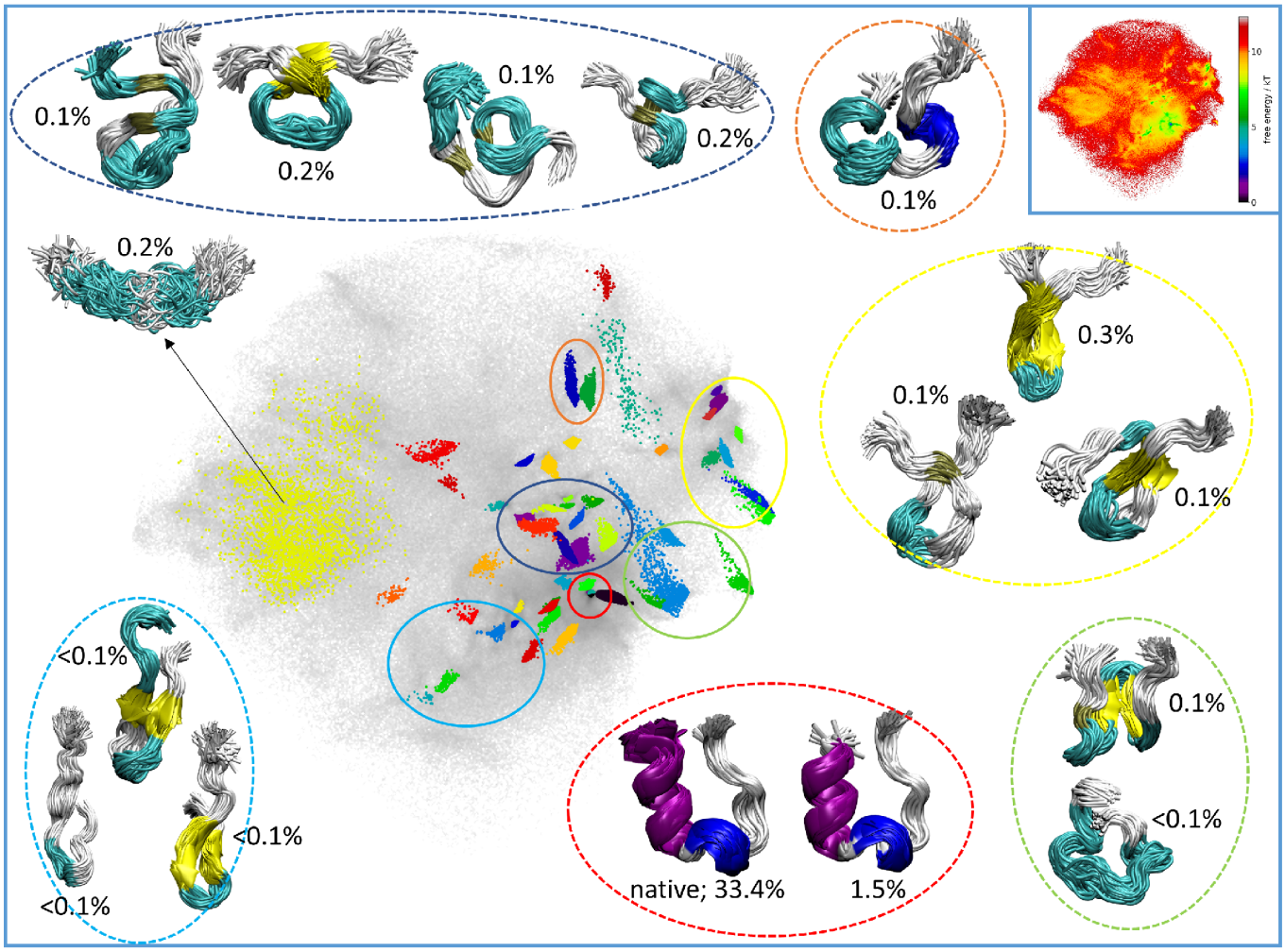


图 4. Trp-cage TC5b（40 条温度 RE 轨迹）：各彩色圆圈所示区域中找到的群体数量最多的簇的示例构象及其所占百分比。簇代表构象展示了整个簇的平均二级结构。簇的颜色随机分配，且颜色会重复。因此，在二维空间中彼此分离但颜色相同的簇包含不同的构象。图中所示的簇包含了所有构象的 36.5%。其余已分配到簇中的构象有 24%，其中大多数是天然结构的细微变体，出于可视化原因未在此处展示。箭头所指的簇是通过提高 RMSD 截断值生成的模糊簇之一。右上：二维 encodermap 空间的直方图。

蛋白质相比，Protein B 并没有单一且非常稳定的状态，反而是三条螺旋可以相当容易地彼此相对移动。这形成了一个宽广的构象空间，各个状态之间的能垒很小。因此，各个构象状态不易清晰区分，而是逐渐彼此融合／转变。考虑到较长的模拟时间，这个系统在构象上的聚类非常困难。

为展示我们的聚类方案如何运作，我们选择将其应用于这四个面临截然不同挑战的系统（例如，超大数据集、同一数据中既有高丰度状态也有极低丰度状态，以及沿轨迹折叠/未折叠构象数量的差异）。对于每个系统，我们最初都进行了相同次数的聚类迭代（10 次），然后评估聚类结果，并判断特定系统是否需要进一步迭代。

B. Trp-cage

a. TC5b. 对于 Trp-cage 的副本交换模拟，聚类方案运行了 10 次迭代，并将所有构象中的 60.5% 分配到各个簇中。图 4 展示了全部 40 个副本的 encodermap 投影，以及经过 10 次迭代后发现的一些规模最大的簇和这些簇的代表性构象。相似构象被归为一组，而罕见结构则分散在映射图各处。例如，Trp-cage 副本交换模拟中的天然构象（占所有构象的 33.4%

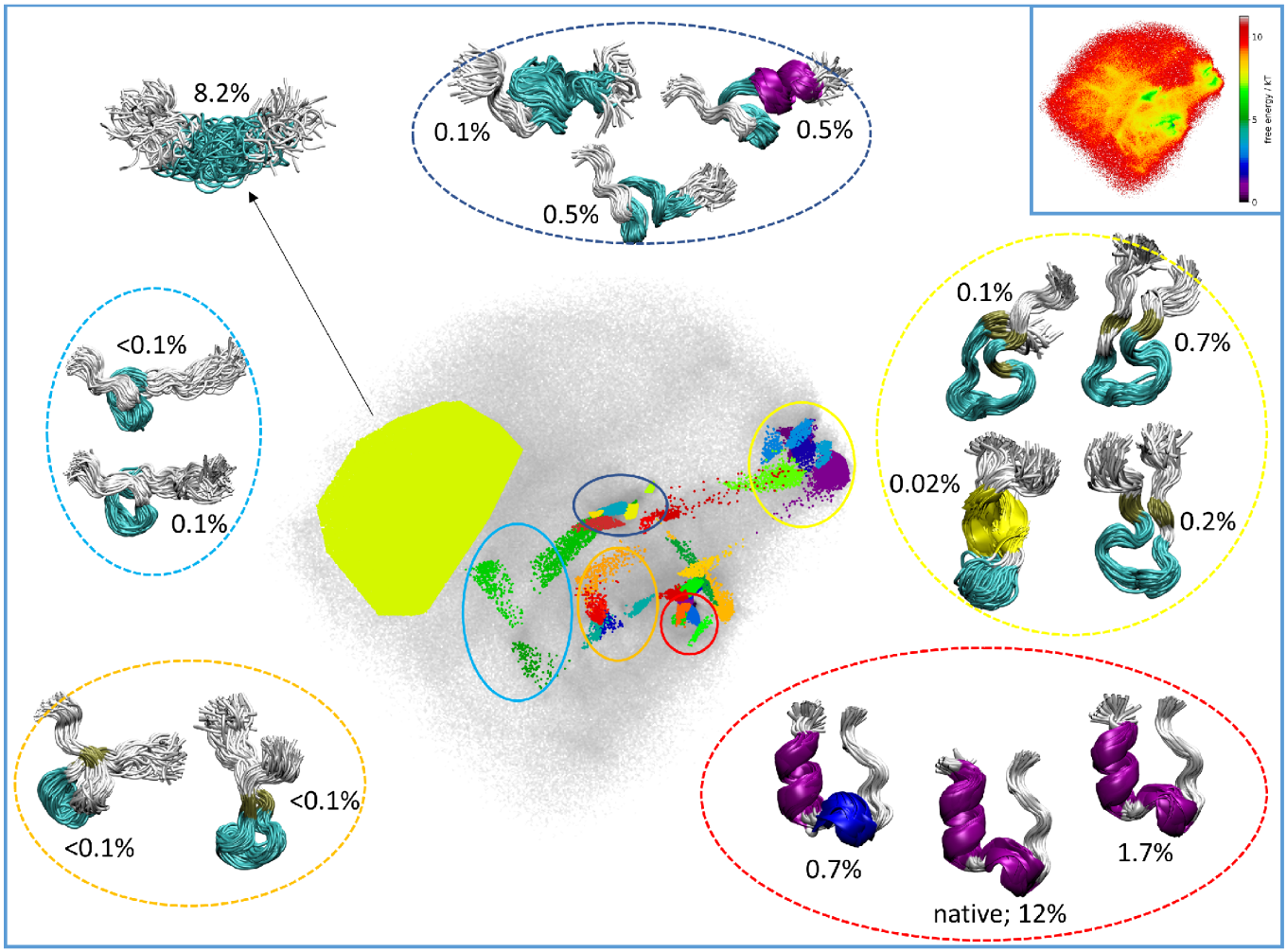


图 5. Trp-Cage TC10b 中数量最多的簇及其相应构象 [42] 投影到与 TC5b 相同的二维 encodermap 空间（图 4）。b右上：二维投影的直方图。

）显示在图 4 二维映射的右下方。左下方是主链中部附近有一个转角的构象。这些构象的主链链段的两部分彼此紧邻，并部分形成 β-折叠结构。

在基于 RMSD 将结构分配到簇时采用更大的截断距离（其他簇是通过对中心构象应用 1.8 ˚A 的 RMSD 截断值生成的），我们得到了更大且相当弥散的伸展构象簇（其中一个簇显示在图 4 投影的左侧，并由箭头标出）。对于每个模糊簇，我们分别计算当前迭代中识别出的簇的质心与簇成员之间 RMSD 值中最大的 20% 的均值，以此确定适当的 RMSD 截断值大小（此处所示簇的截断值为 5.5 ˚A）。在识别模糊簇之前，我们先基于固定 RMSD 截断值（TC5b 为 1.8 ˚A）持续分配结构，直至达到第 II D 节定义的某个停止点（TC5b 的平均簇大小）。达到这一停止点后，便按照上述方法调整 RMSD 截断值，从而得到模糊簇。这样可以确保在识别模糊簇之前，所有能够分配到定义明确的簇中的构象都已被排除。采用这种可变截断值有助于识别弥散簇：这类簇中的成员共享某种结构基序，但并未收敛为定义明确的构象，正如这里所示的簇。

从图 4 所示的聚类结果可以看出，所提出的聚类工作流程能够高效地识别出 TC5b 系统中结构定义非常明确的簇。在 10 次聚类迭代中，该算法将所有构象的 60.5% 分配到了 260 个簇中。除了数量众多的天然态（33.4%）外，该算法还找到了非常“罕见”的状态，其中仅包含极少量构象（≤0.1%），但结构相似性仍然非常明确。

b. TC10b. 图 5 展示了将相同分析应用于 K8A 突变体 TC10b Trp-cage 轨迹的结果。

我们使用在 TC5b 上训练的 encodermap，将轨迹投影到相同的二维空间。不过，簇的识别当然完全独立，且两种情况下各不相同，因为聚类是在更高维的 cc 分析空间中进行的。

值得注意的是，该突变体天然态的主链构象与 TC5b 系统中的构象极为相似。然而，与 TC5b 系统中的 33.4% 相比，这个最大的簇仅包含轨迹中全部构象的 12%。如果将中心构象与天然构象的均方根偏差在 a 2 ˚A 以内的所有簇合并，得到的天然构象占比为 16.9%。这与 Deng 等人报告的天然簇大小非常吻合。 [45], Ghorbani 等人 [46] 分析了 Lindorff-Larsen 等人提供的相同 Trp-cage 轨迹。[42]此外，我们得到的 33.4% 已归类构象与 Sidky 等人的报告也非常一致。 [47]他们发现，共有 31% 的构象分布在八个亚稳态宏观态中，其余 69% 则构成一个大的“熔融球状体”态。

TC10b 轨迹的无序程度更高，这可以从二维空间中较为均匀的投影（图 5 右上图）以及补充信息 S-III 节图 S3 中与天然构象的均方根偏差值看出。这也是聚类方案经过 10 次迭代后仅将所有构象中的 33.4% 分配到簇中的原因。如果希望将更多帧分配到簇中，可以增加聚类迭代次数、提高 RMSD 截断值，或同时采用这两种方法（本文稍后将展示 Protein B 系统采用这种方法的结果）。

然而，与 Trp-cage RE 的二维投影中相似区域内的簇相比，映射图正中央的簇（深蓝色圆圈）更加紧凑、塌缩。此外，在副本轨迹的映射图左侧最底部发现的一些簇（浅蓝色圆圈），在 TC10b 轨迹中完全没有出现。映射图左侧那个非常大且分散的簇则同时存在于两个系统中。

c. 直接在 *TC5b* 的二维空间中进行聚类。上述聚类是在应用 cc 分析算法后，在 20 维空间中完成的，并仅通过 encodermap 得到的二维投影进行展示。为说明我们方法的优势，我们还使用 HDBSCAN 直接在二维 encodermap 空间中进行聚类。用于此次聚类的 encodermap 空间，与我们在图 4 和图 5 中用于展示 cc 分析聚类的空间相同。此次聚类的结果以及若干选定的簇见图 6。总共有 13.5% 的构象被分配到 362 个簇中。发现的最大簇是天然簇，但它仅包含 0.8% 的构象，而在 cc 分析空间中聚类得到的比例为 33.4%。二维空间中的聚类识别出了一些结构定义非常明确的簇，例如簇 0、1 和 3，但

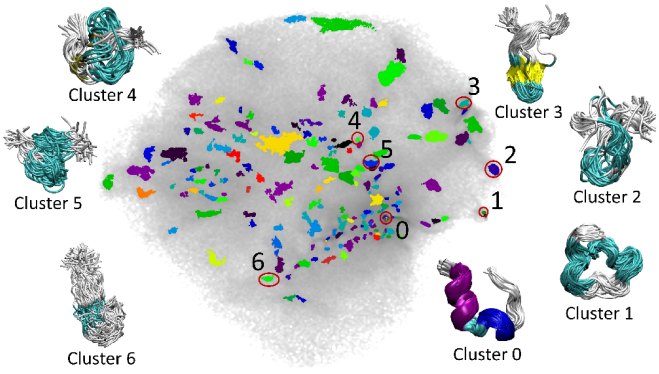


图 6. 使用 HDB-SCAN 对 TC5b 进行聚类得到的二维 encodermap 空间。图中展示了所选簇的表示，这些簇在二维映射中的位置与图 4 所示聚类方案得到的簇相同。

同时也产生了许多非常分散且非均匀的簇。为了量化这种非均匀性，我们计算了簇内均方根偏差的平均值。对于 TC5b 系统，我们的聚类工作流程得到的平均簇 RMSD 为 1.34 ˚A，加权平均 RMSD 为 1.03 ˚A，其中权重定义为各簇占所有已聚类数据的比例。直接在二维空间中聚类时，平均 RMSD 为 2.25 ˚A，加权平均 RMSD 为 2.73 ˚A。这清楚地表明，直接在二维空间中聚类时，簇内均方根偏差方差的平均值要大得多。此外，二维空间中的聚类本身也显然高度依赖于二维映射图的质量。

除了各个簇的构象特征更加清晰（通过簇内均方根偏差体现）之外，我们的聚类方案还成功将 60.5% 的所有构象分配到了不同的簇中。相比之下，二维投影中的聚类根据聚类参数的选择，仅将 9-14% 的所有构象分配到了簇中。

d. 与其他聚类方法的比较。为了进一步评估我们的聚类方案，我们还将一种常用的聚类流程应用于 TC5b 数据。补充信息的 S-IV 节以及图 S4 和 S5 展示了将 K-means 算法应用于相同集体变量（TC5b 的成对 C_α 距离）的 11 维 PCA 投影所得的结果。

总之，该方案在所考察的系统中识别出了结构定义明确的簇和相当弥散的簇。尽管 40 条 RE 轨迹组合后形成了一个多样性极高的数据集，该聚类方案仍成功将大量构象（60%）归入簇中。我们对 TC10b 的聚类结果与其他研究者的发现高度一致 [45–47]。此外，与二维空间中的聚类结果进行比较后，明显可以看出，在 HDBSCAN 中使用 cc 分析算法获得的更多维度，优于仅依赖低维表示。

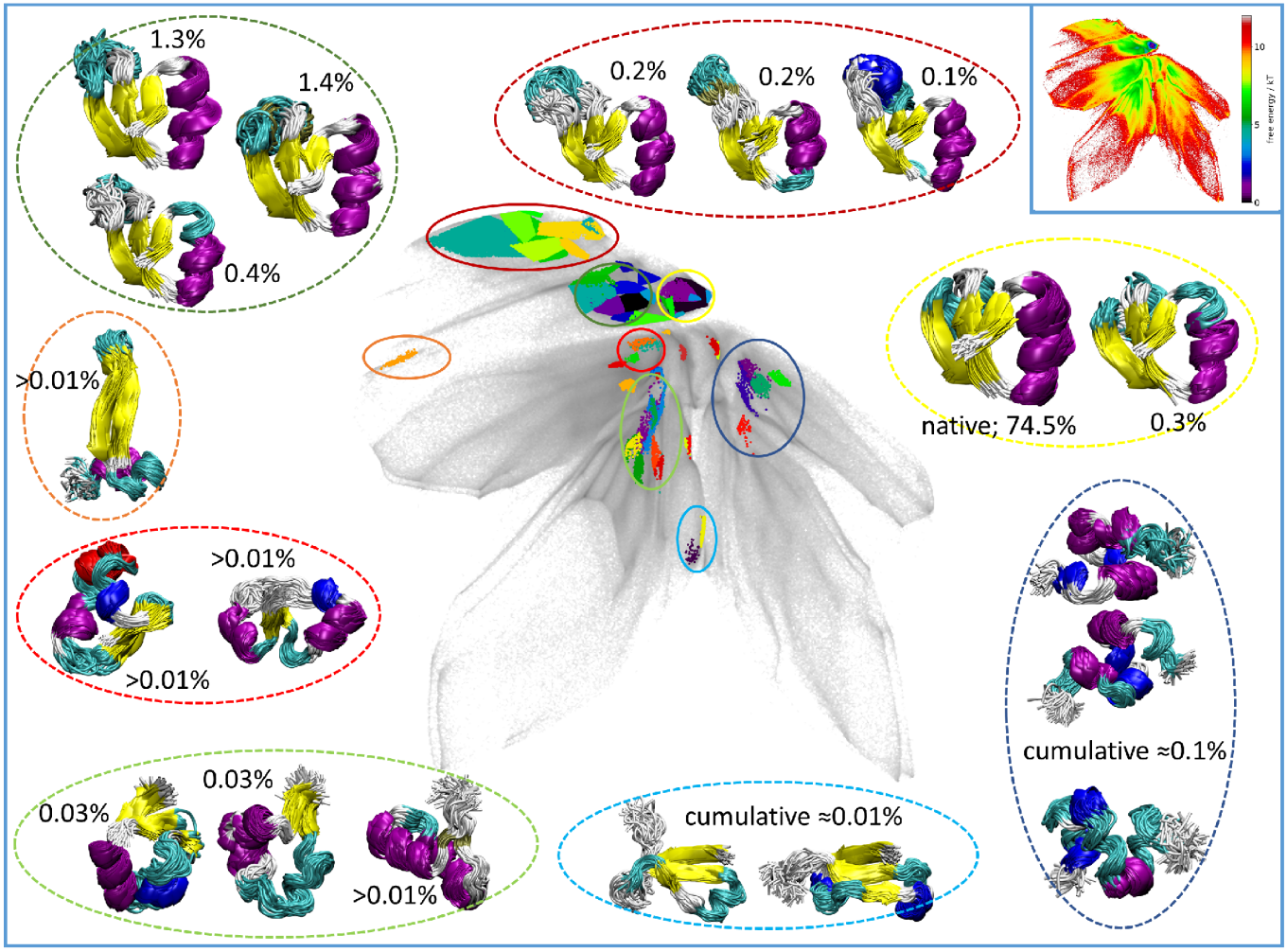


图 7. NTL9 的二维 encodermap 投影。该投影大致可分为三个部分：上部是最密集的区域（类天然态所在之处）；左下和右下区域被一条无群体分布的垂直间隙隔开。左侧包含多种构象，其中单个 β折叠主要由蛋白质的开头和末端部分形成。相比之下，右侧主要是沿主链带有多个螺旋的伸展构象。图中展示了映射图各标记区域内一些群体最多的簇的代表性构象及其群体。黄色圆圈内的所有簇都与天然簇极为相似，合计占全部构象的 76%。此处展示的结构占全部构象的 78.4%。右上：二维 encodermap 空间的直方图。

C. NTL9

接下来，我们研究了 NTL9 [42] 的超长模拟（1877 微秒）。该系统共有 938 万帧需要聚类，是展示所提出的算法如何处理大量数据的理想案例。经过 10 次迭代，81% 的构象被归入簇中。图 7 展示了使用 encodermap 制作的二维投影，点的颜色对应于该方案迭代 10 次后识别出的簇；右上角为二维空间的直方图。总共，我们识别出 157 个簇，并在 10 次聚类迭代中将 81% 的构象归入这些簇。

将相对于折叠态的均方根偏差值时间序列与 Trp-cage Anton 模拟的数据（补充信息，S-III 节，图 S3）进行比较后发现，两个系统表现出截然不同的动力学特征。在 Trp-cage 的情形中，均方根偏差显示出系统的无序特性；而在 NTL9 轨迹中，均方根偏差值总体相当低，只在较短时间内跃升至较大值。这表明，在大部分模拟时间内，NTL9 系统都处于类似天然态的状态。聚类方案的第一次迭代就证实了这一点。我们发现有两个簇，共包含全部构象的 75.8%。

这个例子也很好地说明了迭代聚类方法如何高效地识别大小和密度迥异的簇（高密度的天然态和低密度簇）。在找到

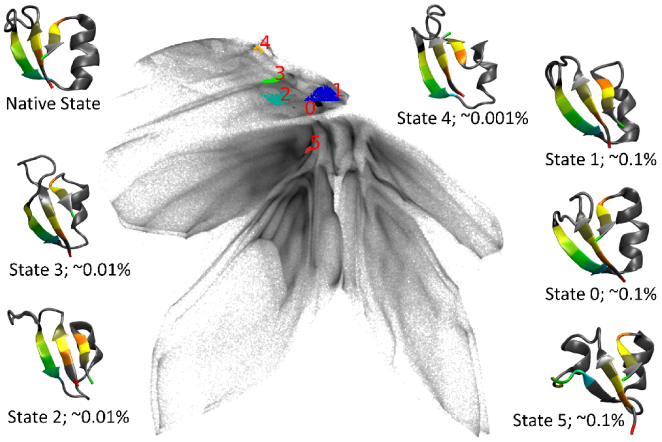


图 8. 在 NTL9 轨迹 0、2 和 3 中发现的移码态。构成天然态中 β 折叠的残基根据其残基 ID 着色。

并移除前两个簇（占数据的 75.8%）后，聚类算法在接下来的聚类迭代中对集体变量空间中密度较低的区域会变得敏感得多。

我们将聚类结果与其他分析参考文献 [42]中 NTL9 轨迹的研究进行了比较。Mardt 等人 [29] 将 VAMPnets 应用于轨迹 0，发现其中折叠态、类天然构象共占 89.1%。如果取我们通过分析轨迹 0、2 和 3 得到的簇，并评估来自轨迹 0 的构象（轨迹 0 在模拟时间中处于类天然态的比例更大；参见补充信息 S-III 节中的均方根偏差图和图 S3），那么我们发现的折叠态、类天然构象数量与 [29] 非常吻合。此外，Schwantes 和 Pande [44] 报告发现了三种“移码”态，这些态的占比很低，因此很难发现。“移码”指的是参与形成类天然态中 β 折叠结构的特定残基（残基 1–6、16–21 和 35–39）。我们使用该方法，在 NTL9 轨迹 0、2 和 3 中识别出六种不同的移码态（见图 8）。

状态 0、1 和 2 也在 [44] 中被发现。据我们所知，状态 3、4 和 5 尚未见报道。在状态 0 中，三条 β 折叠链中间的一条向下移位；而在状态 2 中，最右侧的链向下移位。在状态 1 中，与天然态相比，中间和最右侧的链都发生了移位。状态 3 与状态 1 相似，中间和最右侧的链都发生了移位；但在状态 3 中，最右侧的链向上移位，而非像状态 1 中那样向下移位。在这六种状态中，状态 4 独一无二，因为其中最右侧的链旋转了 180 度。最后，状态 5 与其他状态不同，其链上最左侧链和中间链之间多了一段螺旋。由于这段额外的螺旋，最左侧的链相较于天然态发生了极大的移位。

对这些移码态的识别突显了所提出工作流程的一项优势：它能够在同一数据集中同时找到占比极高的状态（天然态，占 74.5%）和占比极低的簇（<0.001%）。

D. 蛋白质-B

我们分析的最后一个系统是 Protein B。该系统的天然态并不十分稳定，三条螺旋可以相对自由地彼此移动。这一点可从补充信息的 S-III 节图 S3 所示的、相对于最接近的实验同源蛋白（1PRB）的均方根偏差时间序列中看出。系统中没有任何一段较长时期内数值保持稳定，这意味着不同可及构象之间不存在较大的自由能势垒，因此系统会不断转变为不同构象。 [42], 也发现了这一点，其中作者指出，他们无法确定 Protein B 的折叠态与未折叠态之间存在自由能势垒（在许多不同的反应坐标上进行了测试）。

这种高度动态的系统给构象聚类带来了很大挑战。这里我们想展示算法的局限性，以及如何获得令人满意的聚类结果。图 9 概述了该方案经过十次迭代后找到的一些簇。这些簇仅包含 Protein B 轨迹的 20%，因此仍有 80% 的所有构象未被聚类。

为了将更多数据分配到簇中，可以调整两个参数。首先，可以增大均方根偏差截断值，从而将更多构象分配到已找到的簇中。在这个具体案例中，这种调整是合理的，因为不同状态之间的自由能势垒较低，各个簇在构象方面的界定并不十分清晰。图 9 展示的 10 次聚类迭代中，我们使用的均方根偏差截断值为 3.0 ˚A。在第二次运行中，我们将其增大到 3.5 ˚A。这样一来，所有构象中有 31% 被分配到了界定通常较为宽松的簇中。

第二种方法是增加聚类迭代次数。对于此前分析的系统，在前 10 次聚类迭代中，我们手动调整了聚类参数，包括 cc 分析维度的数量，以及 HDBSCAN 的最小样本数和最小簇大小参数。当然，为了自动运行脚本并执行更多聚类迭代，手动调整参数并不可行。由于在首次聚类迭代中确定合适的 cc 分析维度数量后，通常很少需要再作更改，因此脚本的自动化仅需依赖对 HDBSCAN 参数的选择。当单次迭代得到的簇数量低于某个阈值（本例中为 10 个簇）时，数值

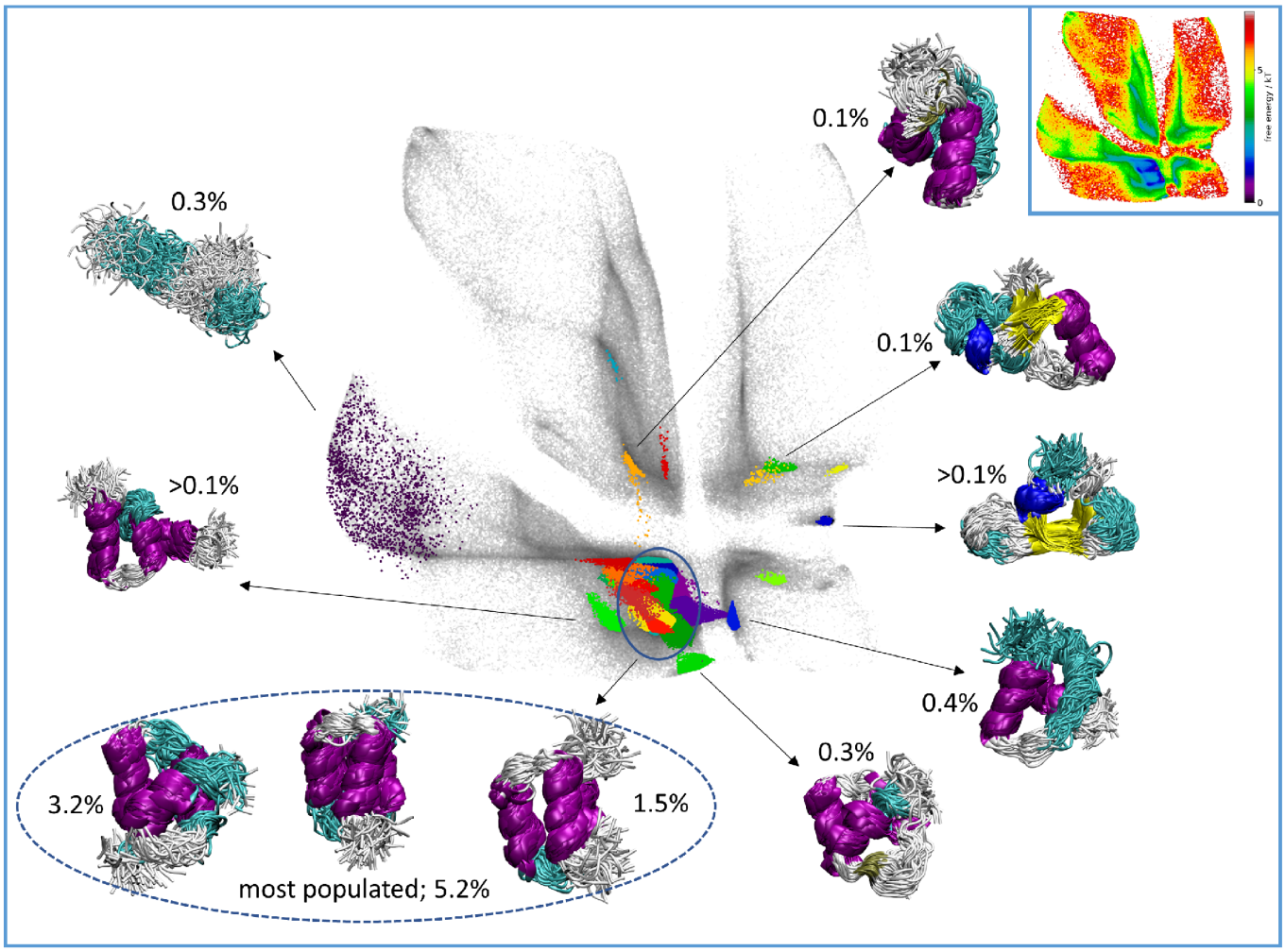


图 9. Protein B：Protein B 系统经过 10 次聚类迭代后发现的部分占比最高的簇的代表性构象及其占比；右上：二维 encodermap 空间的直方图。

会略微降低 HDBSCAN 的最小样本数和最小簇大小参数的取值。这样便能检测到此前未识别出的较小簇。对 Protein B 在前 10 次迭代后采用这种自动化方法，并使用 3.5 Å 的均方根偏差截断值，我们在 100 次迭代中将所有构象的 44.3% 归入簇中；这一过程在我们的工作站上耗时约 15 小时。

IV. 讨论

Trp-cage 系统（TC5b）是一种相对较小的蛋白质，其天然构象相当稳定。然而，40 条温度 RE 轨迹的组合产生了极为多样的数据集，其中包括（在标准条件下）极不可能出现的高能构象。经过十次迭代，该算法成功将所有构象中的 60.5% 归入各个簇；所有 CPU 线程平均每次迭代耗时 360 分钟（在配备 24 个 CPU 线程）。表 I 展示了本文讨论的四个体系的聚类性能。将通常固定的均方根偏差截断值改为可变截断值后，我们证明该算法既能生成构象定义非常明确的簇，也能生成相当松散的簇。归入这类松散簇的构象具有共同的整体结构基序。能够识别这两种簇类型，是所提出算法的优势之一。此外，我们还证明，借助 encodermap 算法将不同体系投影到同一张二维映射图上，聚类工作流程便可直接比较不同体系（即使它们的结构略有差异）。这使我们能够直接、直观地比较不同轨迹采样到的相空间及其各自识别出的状态。我们比较了两种聚类结果：一种是在 20 维 cc 分析空间中进行聚类，再投影到二维空间；另一种则完全在二维 encodermap 空间中寻找簇。由此，我们证明了使用更多维度并结合 cc 分析与 encodermap 的优势。该方案生成的簇

具有更加清晰的结构相似性（均方根偏差方差更低），同时对二维映射图质量的依赖也小得多。

我们分析了 NTL9 的长轨迹（938 万帧），以展示所提出的方案如何处理海量数据。所有 CPU 线程每次迭代平均需要 1320 分钟的计算时间（在我们的办公电脑上，每次迭代需要 55 分钟）。由于该系统还存在一个数量极其庞大的天然态，因此也是展示迭代聚类优势的绝佳示例。移除包含天然态的簇后，算法在后续迭代中会对数量较少的区域变得更加敏感。采用这种方法，我们识别出三个数量极少、此前已有报道的移码态 [44], ，以及三个此前未曾发现的移码态。

最后，我们考察了 Protein B，这是一个高度动态的系统。分析这条包含 104 万帧的轨迹，平均每次迭代需要 288 分钟的计算时间（在我们的办公电脑上，每次迭代需要 12 分钟）。该系统中没有将不同构象分隔开来的大型自由能势垒，因此很难进行聚类。事实也证实了这一点：经过十次聚类迭代后，只有 20% 的构象能够被分配到簇中。不过，将均方根偏差截断值从 3.0 $^{\circ}$A 提高到 3.5 $^{\circ}$A 后，可分配到簇中的构象比例便已增至 31%，当然，这也导致簇的结构定义略微不那么明确。还可以将聚类过程自动化，并持续运行，直到一定比例的构象被分配到簇中，或达到指定的迭代次数。在这个具体案例中，我们让该方案自动运行了 100 次迭代（$\approx$15 小时），期间有 44.3% 的构象被分配到簇中。

对于所有考察的系统，所提出的工作流程都能够识别出明确的簇，但代价是有一部分轨迹未被分配。正如我们在此所展示的，其余结构不属于任何特定的簇，可视为未折叠态或过渡态。我们有意不提出任何额外步骤来分配或分类这些构象，因为具体做法高度取决于数据的预期用途。例如，若使用这些数据构建后续的动力学模型，可以采用 PCCA+ analysis [48], 等方法，将其余数据点分配给最近的簇（例如，按模拟时间衡量）；也可以像参考文献 47 中那样，将其定义为亚稳过渡态。此外，还可以将其定义为噪声，并按照参考文献 49 中所述的方法加以使用。

所有性能数据均列于表 I 中，这些数据是通过在补充信息，第 S-V 节中描述的办公室工作站上运行聚类方案脚本获得的。不过，所提出的工作流程具有很强的并行化能力，因为计算开销最大的步骤，是根据凸包和均方根偏差判据，将额外的数据点分配到小子集中的初始识别簇中。如果有大量 CPU 核心可用，二维 encodermap 投影数组可以按核心数量拆分，从而并行执行分配，这将显著提升速度。

利用小型子集所识别出的簇周围的凸包，可以减少在每次聚类迭代中分配额外构象时所需进行的均方根偏差计算量。但原则上，这可能导致一些原本可以分配给某些簇的数据点被排除。为了解这种潜在“簇成员损失”的程度，我们计算了所有被标记为噪声的数据（623,000 个构象；39.5%）与 TC5b 的每个簇中心（260 个簇）之间的均方根偏差。这项计算量极大的任务在我们的工作站上额外耗时 5 小时。我们发现，由于凸包判据，有 42,000 个构象（2.7%）未被分配到已识别的簇。考虑到完整的 10 轮聚类过程耗时 2.5 小时，损失 2.7% 未聚类的数据可视为值得的权衡。

另一个需要考虑的问题是，由于凸包判据，簇可能会被拆分。若某些数据点因满足均方根偏差判据而应归入某个簇，却位于凸包之外，那么在后续某次聚类迭代中，它们可能会被识别为另一个簇。在这种情况下，鉴于这些簇在结构上非常相似，事后将它们合并可能是合理的。为展示这种合并，我们再次分析了 TC5b。我们计算了全部 260 个中心簇构象之间的均方根偏差，并合并了所有均方根偏差为 ≤ 1 $^{\circ}$A 的簇。这样一来，簇的数量减少至 201 个，而对簇内均方根偏差平均值的影响微乎其微。

encodermap 算法的代码可在以下 GitHub 页面获取：https://github. com/AG-Peter/encodermap。ccanalysis 代码可在以下页面找到：https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/Cc_analysis。

五、结论

我们开发了一种聚类方案，将两种不同的降维算法（cc 分析和 encodermap）与 HDBSCAN 结合，通过迭代方法对分子动力学模拟的轨迹进行快速且准确的聚类。cc分析降维方法首先应用于蛋白质模拟数据。该方法将集体变量投影到通常维度相对较高的（$\sim$10-40 维）单位球面上，从而将噪声和波动与重要的结构信息分离开来。随后，可以使用基于密度的聚类方法（如 HDBSCAN）对数据进行高效聚类。HDBSCAN 的迭代应用能够处理投影点在群体和密度方面的非均匀性，这对于蛋白质模拟而言非常典型

数据。由于 cc 分析依赖于计算各帧之间的相关矩阵，因此这极大地限制了可以同时进行投影的数据量。为了能够处理较长的模拟轨迹，我们将 encodermap 纳入该方案。除了二维可视化这一显而易见的优势之外，它还与基于均方根偏差的接受标准结合使用，用于快速地将其他点基于结构分配到最初通过 cc 分析完成的高维投影所识别出的簇中。为了展示所提出方案的准确性和性能，我们将该聚类方案应用于四个测试系统：Trp-cage 的副本交换模拟，以及在 Anton 超级计算机上生成的 Trp-cage 突变体、NTL9 和 Protein B 的三条长轨迹。通过将该方案应用于这四个测试系统，我们证明：该算法能够高效处理极大量的数据；可以在同一张二维映射中比较结构不同系统的簇；还可以对没有非常稳定的天然态、因而在构象上本质上极难聚类的系统进行聚类。此外，该算法能够找到与簇大小无关的簇。通过改变均方根偏差截断值，既可以得到构象定义非常明确的簇，也可以得到模糊簇，后者的成员仅共享一个可以识别整体结构模式。

VI. 补充信息

补充信息（PDF）包括：(S-I)：cc 分析和 HDBSCAN 参数的选择方法。(S-II)：聚类工作流程的停止准则。(S-III)：Trp-cage、Protein B 和 NTL9 轨迹的均方根偏差图。(S-IV)：将所提出的聚类工作流程与 PCA 和 k 均值聚类在 Trp-cage (TC5b) 上进行比较。(S-V)：工作站规格。

VII. 致谢

本研究得到德国研究联合会通过 CRC 969 的资助。我们也非常感谢 bwHPC 集群提供的计算时间，这些计算用于生成 Trp-cage TC5b 轨迹。此外，我们还要感谢 D.E. Shaw 研究组提供 Trp-cage、NTL9 和 Protein B 的轨迹。

[1]R. O. Dror、C. Young 和 D. E. Shaw，Anton：一种专用分子模拟计算机，载于《并行计算百科全书》，D. Padua 编（Springer US，马萨诸塞州波士顿，2011年），第 60–71 页。[2] D. E. Shaw、J. P. Grossman、J. A. Bank、B. Batson、J. A. Butts、J. C. Chao、M. M. Deneroff 等，Anton 2：提升专用分子动力学超级计算机的性能与可编程性标准，载于高性能计算、网络、存储与分析国际会议 SC '14（IEEE 出版社，2014年），第 41–53 页。[3] S. Kmiecik、D. Gront、M. Kolinski、L. Wieteska、A. E. Dawid 和 A. Kolinski，粗粒化蛋白质模型及其应用，《化学评论》**116**，7898（2016年）。[4] Y. I. Yang、Q. Shao、J. Zhang、L. Yang 和 Y. Q. Gao，分子动力学中的增强采样，《化学物理学杂志》**151**，070902（2019年）。[5] M. Shirts 和 V. S. Pande，世界各地的屏幕保护程序，团结起来！《科学》**290**，1903（2000年）。[6] K. Pearson，LIII. 关于空间中点系的最佳拟合直线与平面，《哲学杂志》（1798-1977）**2**，559（1901年）。[7] H. Hotelling，将一组复杂统计变量分解为主成分，《教育心理学杂志》**24**，417（1933年）。[8] L. Molgedey 和 H. G. Schuster，利用延迟时间相关分离独立信号的混合，《物理评论快报》**72**，3634（1994年）。[9] A. Hyv¨arinen、J. Karhunen 和 E. Oja，利用时间结构的方法，载于《独立成分分析》（John Wiley & Sons, Inc.，2002年），第 18 章，第 344 页。[10]G. Young 和 A. Householder，《以点之间的相互距离描述点集》，Psychometrika **3**，19（1938）。[11] W. Torgerson，《多维尺度分析：I. 理论与方法》，Psychometrika **17**，401（1952）。[12] M. Ceriotti、G. A. Tribello 和 M. Parrinello，《使用 sketch-map 简化复杂自由能景观的表示》，Proc. Natl. Acad. Sci. **108**，13023（2011）。[13] J. Tenenbaum、V. D. Silva 和 J. Langford，《非线性降维的全局几何框架》，Science **290 5500**，2319（2000）。[14] R. R. Coifman、S. Lafon、A. B. Lee、M. Maggioni、B. Nadler、F. Warner 和 S. W. Zucker，《几何扩散作为调和分析和数据结构定义的工具：扩散映射》，Proc. Natl. Acad. Sci. **102**，7426（2005）。[15] R. R. Coifman 和 S. Lafon，《扩散映射》，Appl. Com- put. Harmon. Anal. **21**，5（2006）。[16] L. McInnes、J. Healy 和 J. Melville，《UMAP：用于降维的均匀流形近似与投影》（2020），arXiv:1802.03426 [stat.ML]。[17] G. E. Hinton 和 R. R. Salakhutdinov，《利用神经网络降低数据维度》，Science **313**，504 2006( )。[18] W. Chen、A. R. Tan 和 A. L. Ferguson，《利用自动编码器发现集体变量并进行增强采样：网络架构与误差函数设计的创新》，J. Chem. Phys. **149**，072312（2018）。[19] C. Wehmeyer 和 F. No´e，《时间滞后自编码器：用于分子动力学中慢集体变量深度学习的

《分子动力学》，J. Chem. Phys. **148**，241703（2018）。[20] C. X. Hern´andez、H. K. Wayment-Steele、M. M. Sultan、B. E. Husic 和 V. S. Pande，《复杂动力学的变分编码》，Phys. Rev. E **97**，062412（2018）。[21] Y. B. Varolg¨une¸s、T. Bereau 和 J. F. Rudzinski，《利用高斯混合变分自编码器从分子模拟中获得可解释的嵌入》，Mach. Learn.: Sci. Technol. **1**，015012（2020）。[22] J. B. MacQueen，《多变量观测值分类与分析的若干方法》，载于《第五届伯克利数学统计与概率研讨会论文集》第1卷，L. M. L. Cam 和 J. Neyman 编（加州大学出版社，1967年），第281–297页。[23] D. Arthur 和 S. Vassilvitskii，《K均值++：精心选择初始种子的优势》，载于《第八届ACM-SIAM 离散算法研讨会论文集》，SODA '07（工业与应用数学学会，费城，宾夕法尼亚州，美国，2007年），第1027–1035页。[24] J. Shi 和 J. Malik，《归一化割与图像分割》，IEEE Transactions on Pattern Analysis and Machine Intelligence **22**，888（2000）。[25] M. Ester、H.-P. Kriegel、J. Sander 和 X. Xu，《一种用于在含噪大型空间数据库中发现簇的基于密度的算法》，载于《第二届国际知识发现与数据挖掘会议论文集》，KDD'96（美国人工智能促进会出版社，1996年），第226–231页。[26] A. Rodriguez 和 A. Laio，《通过快速搜索和查找密度峰进行聚类》，Science **344**，1492（2014）。[27] O. Lemke 和 B. G. Keller，《共有最近邻聚类——一项基准测试》，Algorithms **11**（2018）。[28] X. Daura、K. Gademann、B. Jaun、D. Seebach、W. F. van Gunsteren 和 A. E. Mark，《肽折叠：当模拟遇上实验》，Angew. Chem. Int. Ed. **38**，236 1999( )。[29] A. Mardt、L. Pasquali、H. Wu 和 F. No´e，《用于分子动力学深度学习的VAMPnets》，Nat. Commun. **9**，5（2018）。[30] A. Glielmo、B. E. Husic、A. Rodriguez、C. Clementi、F. No´e 和 A. Laio，《用于分子模拟数据的无监督学习方法》，Chem. Rev. **121**，9722（2021），pMID: 33945269。[31] W. Brehm 和 K. Diederichs，《消除串行晶体学中的索引歧义》，Acta Cryst. Section D **70**，101（2014）。[32] K. Diederichs，《剖析含噪复合数据集中的随机差异和系统差异》，Acta Cryst. Section D **73**，286（2017）。[33] T. Lemke 和 C. Peter，《encodermap：分子构象的降维与生成》，J. Chem. Theory Comput. **15**，1209（2019）。[34] R. J. G. B. Campello、D. Moulavi、A. Zimek 和 J. Sander，《用于数据聚类、可视化和异常值检测的层次密度估计》，ACM Transactions on Knowledge Discovery from Data **10**，1（2015）。[35] D. C. Liu 和 J. Nocedal，论大规模优化中的有限内存 BFGS 方法，Mathematical Pro- gramming **45**，503（1989）。[36] R. J. Gildea 和 G. Winter，在存在索引歧义的情况下从稀疏多晶体数据集确定 Patterson 群对称性，Acta Cryst. Sec- tion D **74**，405（2018）。[37] G. M. Assmann、M. Wang 和 K. Diederichs，多数据集晶体学的突破：简单且确定性的数据缩放/选择方法，Acta Cryst. Section D **76**，636（2020）。[38] K. Su、O. Mayans、K. Diederichs 和 J. R. Fleming，使用 pasimap 进行成对序列相似性映射：以肌联蛋白免疫球蛋白结构域的重新分类为案例研究，bioRxiv 10.1101/2022.05.13.491469（2022）。[39] L. McInnes、J. Healy 和 S. Astels，hdbscan：基于层次密度的聚类，Journal of Open Source Software **2**，205（2017）。[40] S. I. Gass 和 M. C. Fu 编，Prim 算法，载于《运筹学与管理科学百科全书》（Springer US，马萨诸塞州波士顿，2013），第1160–1160页。[41] D. Van Der Spoel、E. Lindahl、B. Hess、G. Groenhof、A. E. Mark 和 H. J. C. Berendsen，GROMACS：快速、灵活且免费，J. Comput. Chem. **26**，1701（2005）。[42] K. Lindorff-Larsen、S. Piana、R. O. Dror 和 D. E. Shaw，快速折叠蛋白质如何折叠，《科学》**334**，517（2011）。[43] E. Su´arez、R. P. Wiewiora、C. Wehmeyer、F. No´e、J. D. Chodera 和 D. M. Zuckerman，马尔可夫状态模型能做什么、不能做什么：蛋白质折叠模型中相关性与路径型观测量的比较，bioRxiv 10.1101/2020.11.09.374496（2020）。[44] C. R. Schwantes 和 V. S. Pande，改进马尔可夫状态模型构建，揭示 NTL9 折叠中的许多非天然相互作用，J. Chem. Theory Co m- put. **9**，2000（2013），PMID：23750122。[45] N.-j. Deng、W. Dai 和 R. M. Levy，未折叠态内部的动力学如何影响蛋白质折叠：基于马尔可夫状态模型和超长分子动力学轨迹的分析，J. Phys. Chem. B **117**，12787（2013），PMID：23705683。[46] M. Ghorbani、S. Prasad、J. B. Klauda 和 B. R. Brooks，使用高斯混合变分自编码器对蛋白质折叠模拟进行变分嵌入，J. Chem. Phys. **155**，194108（2021）。[47] H. Sidky、W. Chen 和 A. L. Ferguson，基于无状态可逆 VAMPnets 识别的慢折叠模态，构建 TRP-cage 微型蛋白质动力学的高分辨率马尔可夫状态模型，J. Phys. Chem. B **123**，7999（2019）。[48] P. Deuflhard 和 M. Weber，构象动力学中的稳健 Perron 簇分析，Linear Algebra and its Applications **398**，161（2005），《矩阵与数学生物学》专题。[49] O. Lemke 和 B. G. Keller，用于识别核心集的基于密度的聚类算法，J. Chem. Phys. **145**，164104（2016）。