关于利用对比学习进行酶功能预测论文的复现

颜芬

摘要

本文复现的论文提出了一种名为 CLEAN（对比学习增强的酶功能注释）的机器学习算法，与最先进的工具 BLASTp 相比，其可以更准确、可靠、敏感地分配 EC 号给酶。对比学习框架赋予 CLEAN 可靠地注释少研究的酶，纠正错误标记的酶，鉴定具有两个或多个 EC 号的杂交酶等功能，并通过实验进行了验证。

关键词：对比学习；酶功能注释

1 引言

DNA 测序技术的发展，特别是基因组学和宏基因组学工具的发展，已经导致从所有生命分支的生物体中发现了许多蛋白质序列。例如，UniProt 知识库已经编目了约 1.9 亿个蛋白质序列。然而，这些蛋白质中只有 <0.3%(约 50 万) 经过人类管理员的审查，其中 <19.4% 得到明确的实验证据的支持。因此，蛋白质功能注释高度依赖于计算注释方法。然而，对大规模、基于社区的蛋白质功能注释关键评估 (CAFA) 的研究发现，使用现有计算工具自动注释的酶中约有 40% 被错误地注释。因此，蛋白质的功能注释仍然是蛋白质科学中一个艰巨的挑战。特别是，对于少研究和多功能蛋白质的注释不平衡已经妨碍了生物医学进展和药物发现的进程。

酶委员会 (EC) 编号是最著名的酶的数字分类方案，用四位数字表示酶的催化功能。由于靶酶功能的实验表征通常是费力和昂贵的，因此已经开发了许多用于酶功能注释的计算工具。它们包括但不限于基于序列相似性、基于同源性、基于结构和基于机器学习 (ML) 的方法。其中，基于序列相似性的蛋白质基本局部比对搜索工具 (Basic Local Alignment Search Tools for proteins, BLASTp [1]) 是应用最广泛的工具。然而，BLASTp 和其他序列比对工具仅基于序列相似性进行功能注释，当序列相似性较低时，预测结果的可靠性较低。另一方面，几乎所有现有的 ML 模型，如 DeepEC [3] 和 ProteInfer [4]，都是基于多标签分类框架，并且受到生物学中常见的有限和不平衡的训练数据集的影响。因此，需要一种准确性更高、具有更广泛 EC 覆盖范围的强大工具，以发掘目前尚未表征的蛋白质的潜力，并了解蛋白质功能的范围。

本次课程的论文复现工作 CLEAN 是在 UniProt 的高质量数据上训练的，以氨基酸序列作为输入，并输出一个按可能性排序的酶功能列表（以 EC 号为例）。CLEAN 在多项任务中表现出比其他 EC 号注释工具（包括 BLASTp 和最先进的 ML 模型）更好的性能。

2 CLEAN 模型

CLEAN 使用了对比学习框架。本文的训练目标是学习一个酶的嵌入空间，其中欧几里得距离反映了功能相似性。嵌入指的是蛋白质序列的数字表示 (向量或矩阵)，它可以被机器读取，同时仍然保留酶携带的重要特征和信息。在 CLEAN 的任务中，具有相同 EC 号的氨基酸序列欧几里得距离较小，而具有不同 EC 号的氨基酸序列欧几里得距离较大。利用对比损失对模型进行监督训练。在训练过程中 (如图1A 所示)，对于训练数据集中的每个参考序列 (锚点)，都会随机抽取一个具有相同 EC 号的序列作为正样本，以及一个具有不同 EC 号的序列作为负样本。输入序列被嵌入并通过神经网络处理。一系列颜色温暖的方块表示 ESM-1b 嵌入的输入序列表示。类似地，从监督对比学习神经网络中获得的序列嵌入则用冷色表示。为了提高训练效率，CLEAN 采用了一种策略来选择具有挑战性的负样本，而不是随机选择。该策略是根据嵌入向量之间的欧氏距离来优先选择与锚点具有较小欧氏距离的负样本序列。具有较小欧氏距离的负样本序列在特征空间中更接近于锚定序列，这意味着模型需要更精细的学习才能区分它们。这种方式强迫模型学习到更复杂的特征表示，从而在嵌入空间中更准确地区分不同的 EC 编号。如果负样本序列随机选择，那么很多样本在嵌入空间中距离锚定序列很远，模型很容易区分这些样本。这可能导致模型在遇到难以区分的、相似的样本时性能下降，因为在训练过程中没有学习到足够的鉴别能力。

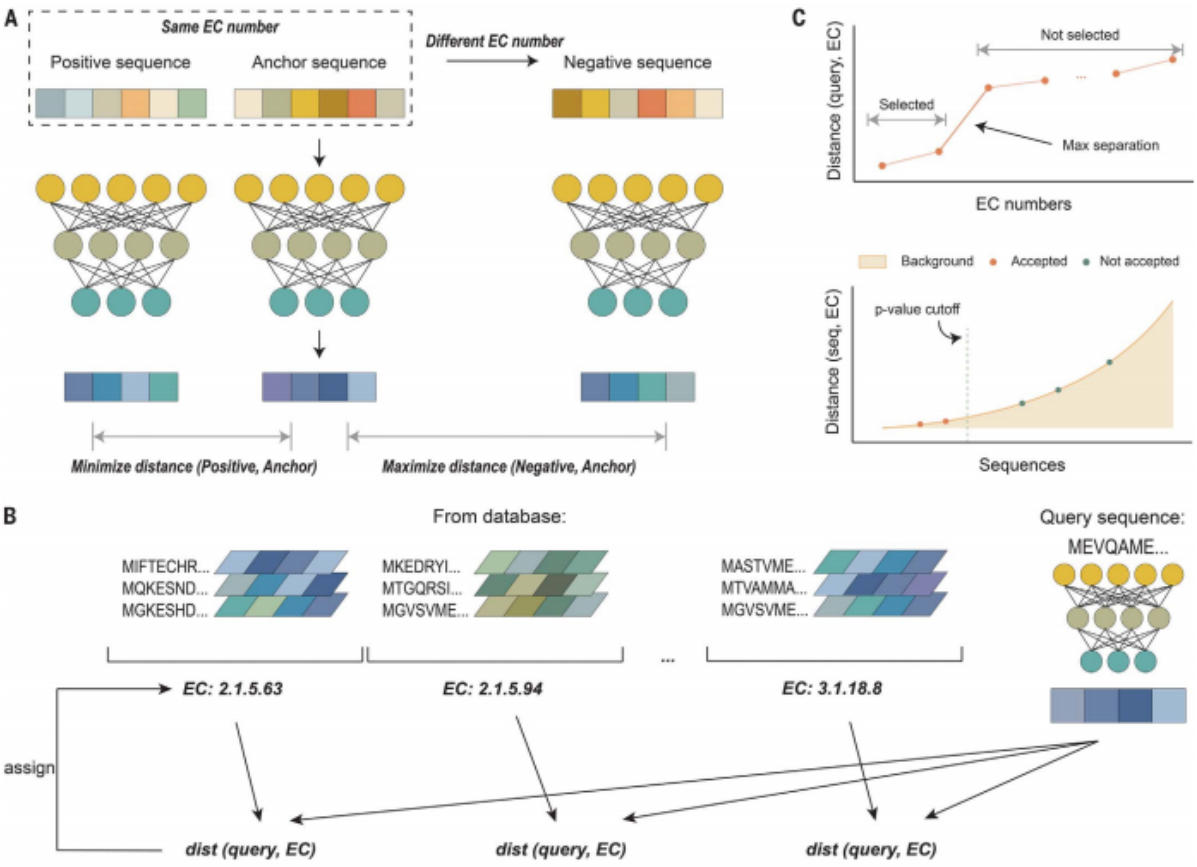

图 1. 基于对比学习的 CLEAN 框架。

在训练阶段，使用从语言模型 ESM1b [2] 得到的蛋白质表示作为前馈神经网络的输入，其输出层生成了输入蛋白质的改进、功能感知的嵌入。学习目标是一个对比损失函数，它通过

最小化锚点和正样本之间的距离，同时最大化锚点和负样本之间的距离。在进行预测时，通过对训练集中属于该 EC 号的所有序列的学习嵌入进行平均，得到了 EC 号簇中心的表示（如图1B 所示）。随后，计算查询序列与所有 EC 号簇中心之间的成对距离。与查询序列显著接近的 EC 号簇被预测为输入蛋白质的 EC 号。

用于模型开发和评估的数据库是通用的蛋白质知识库 UniProt。简单地选择所有查询酶的前 1 名、前 2 名或前 k 名的 EC 号是不合适的，因为选择前 1 名的 EC 号会忽略酶的混杂性，但选择更多的 EC 号，精度会急剧下降，因为大多数酶只有一个 EC 号。本文开发了两种 EC 选择方法来预测从输出排序中获得自信 EC 编号（如图1C 所示）：(i) 最大分离法，即贪婪方法，选择在与查询序列的两两距离方面与其他 EC 编号有最大差异 (突出) 的 EC 编号；(ii) 基于 P 值的方法，通过与背景进行统计显著性比较，识别具有统计意义的 EC 编号。

3 本文方法

3.1 ESM1b 模型

本文使用从 ESM1b 语言模型得到的蛋白质表示作为前馈神经网络的输入，经过超参数优化，训练的一个高模型容量的 Transformer。经过训练后，该模型输出的特征表示中隐含蛋白质的二三级结构、功能、同源性等信息，并且这些信息能够通过线性投影显化。ESM-1b 模型实际上是参数经过调整的 Transformer，采用系统优化法优化 Transformer 中的超参数，再在 UR50 数据库上进行预训练，即可得 ESM-1b 模型。本模型的最终目标是探究 Transformer 能否从蛋白序列中提取结构信息，但该目标不容易直接通过训练实现，因此模型训练使用的是代理任务——masked training，即随机遮盖序列中部分片段，基于序列中其他未被遮盖的残基预测被遮盖部分真实的残基是什么。

3.2 两种 EC 选择方法

3.2.1 基于 P 值的方法

通过与背景进行统计显著性比较，识别具有统计意义的 EC 编号。通过使用 p 值截止值将其与距离的背景分布进行比较来确定的。通过选择不同的 p 值截止值，用户可以调整预测的精度和召回率。较小的 p 值截断使接受阈值更紧，有利于精度，较大的 p 值截断使阈值更灵活，有利于召回。具体步骤如下：

1. 背景集的建立：
    - 从训练集中随机选择 n 个酶嵌入 (如 20000 个)，作为背景集，这些嵌入是从 CLEAN 模型中提取的。
2. 统计显著性的决定：
    - 设定一个 p-value 阈值 (如 0.001)，用于判断 EC 号码是否在统计学上显著。
    - 这些背景嵌入和 p-value 被用来确定一个 EC 号码是否应该被考虑为统计上显著的。

3. 背景选择的权重:

   - 在挑选背景时,不是均匀选择,而是赋予每个酶的选择概率一个权重,权重为 $1/|EC_i|$,即 EC 类别中酶的数量的倒数。

4. 欧几里得距离的记录:

   - 记录选中背景的欧几里得距离与特定 EC 类别中心之间的距离。
   - 通过这种方式，可以获得所有 EC 类别中心在训练集中和所有背景之间的距离矩阵。

5. 最小距离 EC 号码的选择:

   - 当需要为特定查询酶调用一组 EC 号码时，算法首先选择与查询酶距离最小的 EC 号码 EC_0。然后，将这个最小距离 s_0 与背景中 EC_0 类别中心的距离进行比较。

6. p-value 的计算与判断:

   - 假设 s_0 在背景分布中的排名是 r，则如果 r/n 小于设定的 p-value 截断值，则将 EC_0 号码指定为查询酶的 EC 号码。

3.2.2 最大分离法

选择在与查询序列的两两距离方面与其他 EC 编号有最大差异 (突出) 的 EC 编号；它通过寻找查询序列与不正确的 EC 号码集群之间存在的最大分离距离来工作，它可以减少假阳性的预测，同时确保选择的 EC 号码在生物学上是有意义的。这个方法的基本假设是：任何正确的 EC 号码与查询序列之间的欧几里得距离会显著小于与不正确的 EC 号码集群之间的距离。

表 1. Max-Separation selection method algorithm

Step	**Description**
1	Function MAXSEP(S)[a]
2	Let background noise distance $\gamma = \text{mean}(s_1 + s_2 + \ldots + s_{n-1})$
3	Let noise separation distances $D = d_0, \ldots, d_{n-1} = \lvert s_0 - \gamma \rvert, \ldots, \lvert s_{n-1} - \gamma \rvert$
4	Let slope of separation curve $G = g_0, \ldots, g_{n-1} = \lvert d_1 - d_0 \rvert, \ldots, \lvert d_{n-1} - d_{n-2} \rvert$
5	Initialize maximum separation index $i \leftarrow 0$
6	Let mean slope $\bar{g} = \text{mean}(G)$
7	Let maximum separation index $i \leftarrow i'$ be the first i that satisfies $g_i > \bar{g}$
8	Return the correct set of EC numbers for query $\{EC\}_i = \{EC_0, \ldots, EC_i\}$

Note: [a] S is defined as the sequence of distances between the query sequence and each EC number cluster $S_0, S_1, \ldots, S_{n-1}$ in sorted order.

具体步骤如下：

1. 背景噪声距离 (γ):

- 假设存在一个背景噪声距离 γ，这是查询序列与任何不正确 EC 号码集群之间的平均距离。

2. 欧几里得距离 (S_i) :

   - 计算查询序列与每个 EC 号码集群之间的欧几里得距离，从最小到最大排序。

3. 选择 EC 号码的条件:

   - 如果查询序列与某个 EC 号码集群之间的距离远小于 γ(即小于 $\gamma - \delta$)，则该 EC 号码被认为是正确的。

4. 选择过程:

   - 从排序的距离列表中，选取前 10 个最小距离，例:[5.6,61,7.5,12.22...]，并且以的值（例如:12.88）作为参考点，通过人类直觉或算法来选择那些距离显著小于 γ 的 EC 号码

5. 最大分离指数 (i):

   - 初始化一个最大分离指数 i，这是一个标记点，用来区分正确和不正确的 EC 号码集群之间的分界线。

6. 计算分界线:

   - 计算分界线，通过比较每个距离与的差的绝对值来形成一个分离曲线的斜率数组 G。
   - 找到一个 i 值，使得从这个点开始，至少有 50% 的距离来自正确的 EC 号码集群。

7. 最大分离指数的优化:

   - 优化最大分离指数 i，使得满足两个条件: 1)$|\varepsilon - \delta|$ 最小化，其中 ε 是距离与 γ 的差的绝对值;2)$|EC_i|$ 最小化。
   - 找到一个 i 值，使得从这个点开始，至少有 50% 的距离来自正确的 EC 号码集群。

8. 返回结果:

   - 返回的正确 EC 号码集合是那些满足上述条件的 EC 号码。

3.3 评估指标

评价指标本研究采用的评价指标为 precision score, recall score, F1-score,(AUC)。所有指标都是由 Python 包 scikit-learn 计算的。为了考虑多标签设置，除了使用样本平均值的组合数据集外，所有研究都使用加权平均值。分数是先用 scikitlearn 对测试数据集的真值标签进行二值化处理，然后用各种模型对预测结果进行二值化处理得到的。根据 scikit-learn 文档，将二值化的基础真值和预测结果作为输入。

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$F1\text{-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

3.4 损失函数

偏向数据集对分类模型构成的挑战在于对研究不充分的 EC 号缺乏正例。因此，分类模型很难从有限的正例中学习。为了进一步分析 CLEAN 能够通过对比学习利用不仅有正例还有负例的假设，本文实现了 Supcon-Hard loss（SupconH）—这是一种与三元组损失相比采样更多负例的损失函数。

3.4.1 Triplet Margin Loss

Triplet Margin Loss 是一种训练方法，用于学习输入数据的有效表示。它使用三元组的概念，每个三元组由一个锚点 z_a，一个正样本 z_p，和一个负样本 z_n 组成。正样本与锚点属于同一类别，而负样本属于不同类别。损失函数的目的是使锚点与正样本之间的距离小于锚点与负样本之间的距离至少一个边际 α。这里使用的是 L_2 范式（即欧几里得距离）来计算距离。

Triplet Margin Loss 的公式为：

$$L_{TM} = \max(\|z_a - z_p\|_2 - \|z_a - z_n\|_2 + \alpha, 0)$$

其中 $\|\cdot\|_2$ 表示 L_2 范数，α 是一个边界值，用来确保正样本和锚点之间的距离比锚点和负样本之间的距离小一个确切的值。

优点: 直观且易于理解；它模仿人类区分不同类别的方式。强制性的边界有助于在嵌入空间中明确分离不同类别。

缺点: 选择合适的正样本和负样本对于模型性能至关重要，但这个过程可能很难优化。在每次迭代中仅使用一个正样本和一个负样本可能不足以捕捉类内和类间的整体分布。对于数据不平衡问题，可能不会表现得很好，因为它不考虑类别中样本的数量。

3.4.2 Supcon-Hard Loss

Triplet Margin Loss 是一种训练方法，用于学习输入数据的有效嵌入。它使用三元组的概念，每个三元组由一个锚点 z，一个正样本 z，和一个负样本 zn 组成。正样本与锚点属于同一类别，而负样本属于不同类别。损失函数的目的是使锚点与正样本之间的距离小于锚点与负样本之间的距离至少一个边际 α。这里使用的是 L_2 范数 (即欧几里得距离）来计算距离。

Supcon-Hard Loss 的公式为：

$$L^{sup} = \sum_{e \in E} \frac{-1}{P(e)} \sum_{z_p \in P(e)} \log \frac{\exp(z_e \cdot z_p / \tau)}{\sum_{z_i \in A(e)} \exp(z_i \cdot z_a / \tau)}$$

在这里，E 表示所有可能的酶类别，$P(e)$ 表示与锚点 z_a 相同类别 e 的正样本集合，$A(e)$ 是锚点和正样本集合的并集，τ 是温度参数，用于控制分布的紧密程度。归一化因子 $\frac{1}{|P(e)|}$ 用于消除正样本数量对损失的影响，使得可以灵活选择正样本的数量。

优点: 能够更有效地处理数据不平衡问题，因为它不直接依赖于类别中样本的数量。通过考虑多个正负样本，它能更好地捕捉类内和类间的整体分布。它为正样本和负样本分配权重，允许模型专注于那些对于决策边界更重要的样本。

缺点: 计算上比 Triplet Margin Loss 更复杂，因为它需要在每个批次中处理更多的样本关系。权重的分配可能需要更细致的调优，以便正确地平衡正负样本的影响。尽管它更适合不平衡数据，但是在实践中找到合适的权重和温度参数可能会增加调优的复杂性。

3.5 量化预测结果置信度

作者拟合了酶序列嵌入和 EC 数嵌入之间欧几里得距离分布的双分量高斯混合模型 (GMM)。置信度量化还可以帮助 CLEAN 在置信度较低时报告 EC 号的第三级，从而避免过度预测。

为了评估 CLEAN 的不确定性估计能否有效防止低置信度下的过度预测，研究者创建了一个负控制数据集。这个数据集通过从训练数据集中移除某些 EC 号，然后使用这些 EC 号作为测试数据集来设计，以便 CLEAN 在做出预测时具有高度的不确定性。GMM 的实施细节具体如下：

1. 随机选择 EC 号：选择一千个 EC 号，并计算这些 EC 号嵌入与相应酶序列嵌入之间的距离，以拟合 GMM 内部 EC 号的高斯分布。
2. 计算距离：对于每个选定的 EC 号，还需要计算该 EC 号与不同 EC 号的酶序列之间的距离，以拟合不同 EC 号间的高斯分布。
3. 迭代过程：这个过程重复 40 次，拟合 40 个 GMM，以减少随机误差。

4 复现细节

4.1 与已有开源代码对比

本文复现了 CLEAN 模型。在此基础上，我将预训练模型 ESM-1b 升级成为 ESM2 模型，从而允许更长的序列长度。如图2所示，为了在低维上可视化学习嵌入，我使用 t-SNE（可将高维数据投影到低维时可以保留距离信息）将高维嵌入降为二维图，对比了 CLEAN 与 ESM-1b 的二维可视化，图中的每个点代表一种酶，每种颜色代表一个 EC 值。出于可视化目的，突出显示了几个随机选择的 EC 编号（约 14 种类型）。此外，对开源代码中部分代码如损失函数代码做出修改与优化。

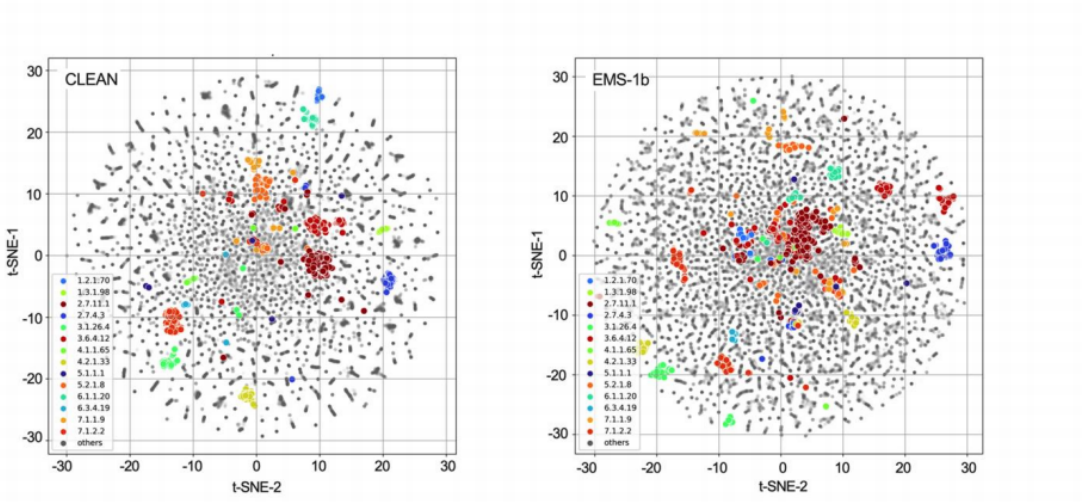

图 2. CLEAN 嵌入与使用 t-SNE 的 ESM-1b 嵌入的二维可视化比较

4.2 训练数据

为了保持高质量的数据，本文只关注 SwissProt，这是 UniProt 经过专家审查的部分。通过额外的筛选，本文只选择具有完整 EC 标签的酶，使得总训练数据约为 220,000 条。

在模型开发过程中，为了防止测试数据过于类似于训练数据，本文使用 MMSeqs2 根据不同的序列相似度阈值（从 10% 到 70%）对数据进行聚类。聚类后的数据集按照 80/20 比例划分，并进行五折交叉验证，其中每个测试集包括与训练集中的任何序列共享相似度不超过 100%、70%、50%、30% 或 10% 的序列。值得注意的是，聚类分割很具挑战性，因为训练集和测试集之间的序列相似性降低了。聚类分割排除了 CLEAN 性能优秀是因为测试数据集与训练数据集相似的可能性。该分割代表了查询酶与当前注释酶具有低相似性的情况。超参数调整也是使用聚类分割进行的。CLEAN 在独立数据集上的评估在训练过程中没有使用聚类分割，因为测试数据集被排除在所有模型的训练过程之外。

4.3 测试数据集

第一个数据集 New-392 由 392 个酶序列组成，涵盖 177 个不同的 EC 编号。

第二个独立数据集 Price-149，由 149 个酶序列组成，涵盖 56 个不同的 EC 编号。是由 ProteInfer 作为一个具有挑战性的数据集进行管理的，因为现有的序列在诸如京都基因和基因组百科全书 (KEGG) 等数据库中被自动注释方法确定为不正确或不一致的标记。

4.4 实验环境搭建

4.4.1 环境配置

CPU: AMD Ryzen 7 7735H with Radeon Graphics

GPU: NVIDIA GeForce RTX 4060

环境	版本
python	3.10.4
pytorch	1.11.0
matplotlib	3.7.0
numpy	1.22.3
pandas	1.4.2
scikit_learn	1.2.0
scipy	1.7.3
tqdm	4.64.0
fair-esm	2.0.0

4.5 使用说明

4.5.1 从 FASTA 文件预先计算 ESM-1b embedding

1. 检索所有 SwissProt 序列的所有嵌入（速度较慢，但训练需要）

   对于选项 1，请在 python 中运行以下命令：

```
>>> from CLEAN.utils import *
>>> ensure_dirs("data/esm_data")
>>> ensure_dirs("data/pretrained")
>>> csv_to_fasta("data/split100.csv", "data/split100.fasta")
```

2. 仅检索要推断的酶的嵌入（快速）

```
>>> from CLEAN.utils import *
>>> ensure_dirs("data/esm_data")
>>> ensure_dirs("data/pretrained")
>>> retrive_esm1b_embedding("test")
```

4.5.2 模型训练

- **准备工作**

Supcon-Hard Loss 对多个正样本和负样本进行采样，并且比 Triplet Margin Loss 小训练数据集表现更好，但训练时间更长。

在训练之前，一个必需的步骤是用 “孤儿” EC 编号（“孤儿”，即这个 EC 编号只有一个序列的方式）来改变序列。由于本文需要对锚定序列以外的正序列进行采样，因此本文对锚定序列进行了突变，并将突变序列用作正序列。**对于每个训练文件，这只需要执行一次！**运行以下命令：

```
>>> from CLEAN.utils import mutate_single_seq_ECs,
    retrive_esm1b_embedding
>>> train_file = "split10"
>>> train_fasta_file = mutate_single_seq_ECs(train_file)
```

```
>>> retrive_esm1b_embedding(train_fasta_file)
```

接下来，为了加快训练速度，需要预先计算成对距离矩阵和嵌入矩阵。**这只需要为每个训练文件执行一次！**运行以下命令：

```
>>> from CLEAN.utils import compute_esm_distance
>>> train_file = "split10"
>>> compute_esm_distance(train_file)
```

这会将两个矩阵（split10.pkl 和 split10_esm.pkl）保存在文件夹位置 /data/distance_map

- **训练一个损失函数为 triplet margin 的 CLEAN 模型**

```
python ./train-triplet.py --training_data split10 --model_name
    split10_triplet --epoch 2500
```

本文建议使用不同的 epoch 来训练不同的 split：

- 10% split: epoch = 2000
- 30% split: epoch = 2500
- 50% split: epoch = 3500
- 70% split: epoch = 5000
- 100% split: epoch = 7000

- **训练一个损失函数为 SupCon-Hard 的 CLEAN 模型**

```
python ./train-supconH.py --training_data split10 --model_name
    split10_supconH --epoch 1500 --n_pos 9 --n_neg 30 -T 0.1
```

与有相同训练数据的 triplet margin 损失相比，该损失函数的模型使用的 epoch 数量减少了 25%。

4.5.3 推断

- **准备工作**

在推断之前，要推断的 AA 序列存储在 CSV 文件中，格式与 CSV 文件中的字段 EC 编号相同，如果不知道，则可以是任何 EC 编号，推断序列的 ESM-1b 嵌入需要使用以下命令（作为示例）进行预计算：

```
>>> from CLEAN.utils import *
>>> csv_to_fasta("data/new.csv", "data/new.fasta")
>>> retrive_esm1b_embedding("new")
```

- **基于 p 值的推断**

```
>>> from CLEAN.infer import infer_pvalue
>>> train_data = "split100"
>>> test_data = "new"
>>> infer_pvalue(train_data, test_data, p_value=1e-5, nk_random=20,
    report_metrics=True, pretrained=True)
```

- **基于最大分离法的推断**

```
>>> from CLEAN.infer import infer_maxsep
>>> train_data = "split100"
>>> test_data = "new"
>>> infer_maxsep(train_data, test_data, report_metrics=True,
    pretrained=True)
```

- **对单个 FASTA 文件进行推理**

```
python CLEAN_infer_fasta.py —fasta_data query
```

4.5.4 量化预测结果置信度

```
python gmm.py
```

4.6 创新点

虽然 CLEAN 模型使用 ESM1b 模型进行预训练将得到的蛋白质表示作为前馈神经网络的输入，但是使用 mutate_singl_seq_ECs 方法时对于所有长度超过 1024 的序列无法处理的问题无法解决。基于此，本文使用 ESM2 模型发布预训练的嵌入和接口，从而允许更长的序列长度。

5 实验结果分析

5.1 数据集

本文只选择具有完整 EC 标签的酶，使得总训练数据约为 220,000 条。使用 MMSeqs2 根据不同的序列相似度阈值（从 10% 到 70%）对数据进行聚类。聚类后的数据集按照 80/20 比例划分，并进行五折交叉验证，其中每个测试集包括与训练集中的任何序列共享相似度不超过 100%、70%、50%、30% 或 10% 的序列。测试集分为 New-392（由 392 个酶序列组成，涵盖 177 个不同的 EC 编号）与 Price-149（由 149 个酶序列组成，涵盖 56 个不同的 EC 编号）。

5.2 实验结果与分析

如图3，是 CLEAN 与使用 ESM-1b 但没有对比学习的基线方法相比，CLEAN 取得了更高的性能

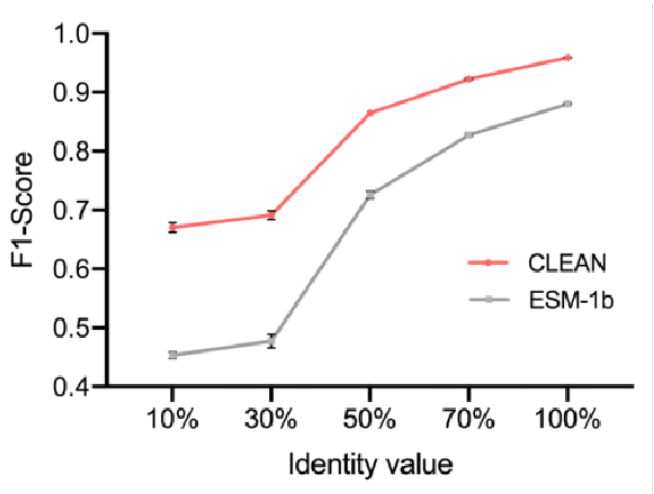

图 3. CLEAN 在五折交叉验证下对不同聚类的评价结果，并与 ESM-1b 进行比较

本文使用聚类分割执行了超参数调优。CLEAN 在独立数据集上的评估没有使用聚类分割进行训练，因为测试数据集被排除在所有模型的训练过程之外。除了 precision、recall 和评价 CLEAN 性能的 F1 分数外，还计算了 AUC(接收者工作特征曲线下的面积)，如图4a 所示。验证集由训练数据集中出现次数不超过 5 次的 EC 号组成。由图可知 SupConH 损失可以进一步改善 CLEAN 在研究不足的酶上的性能。

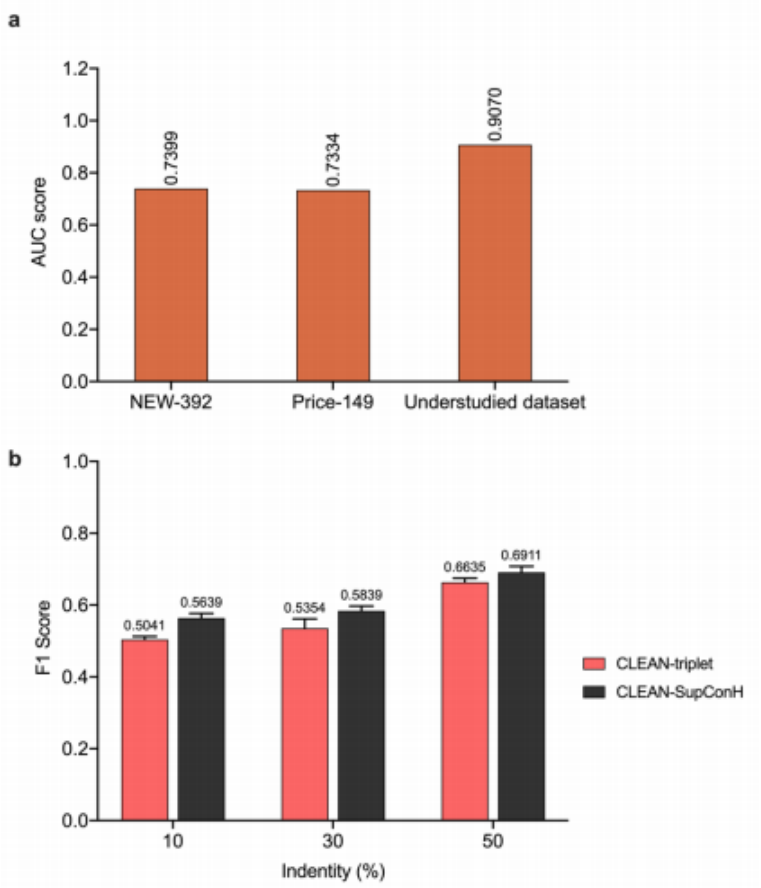

图 4. 在三个数据库对 CLEAN 的 AUC 性能进行评估，并比较了两个损失函数下 CLEAN 的性能

如图5所示，CLEAN 还在从 TrEMBL 数据库获取的大规模数据集上（约有 543k 个酶条目）展示了比 DeepEC 更好的准确性。数据集的条目仅在其注释得分至少为 4（满分为 5）时才被选中，以确保标签质量。箱线图上的每个点代表每个分割数据集的得分。Precision 反映了模型预测为正的样本中实际为正的比例。CLEAN 和 DeepEC 在精确度上的中位数接近，但 CLEAN 的分布似乎稍微紧凑一些，表明其性能较为稳定。Recall 表示模型识别出的正样本占所有实际正样本的比例。在召回率上，CLEAN 的中位数略高于 DeepEC，并且 CLEAN 的数据分布也更紧凑。F1 是精确度和召回率的调和平均值，是评价模型整体性能的一个重要指标。CLEAN 和 DeepEC 的 F1 得分中位数接近，但 CLEAN 的箱体更短，说明其性能变异较小。CLEAN 在所有三个指标上都展示出与 DeepEC 相似甚至略好的性能，特别是在召回率方面似乎有更好的表现。此外，CLEAN 在每个指标上的性能变异似乎都较小，表明其在不同数据子集上的表现更加稳定。

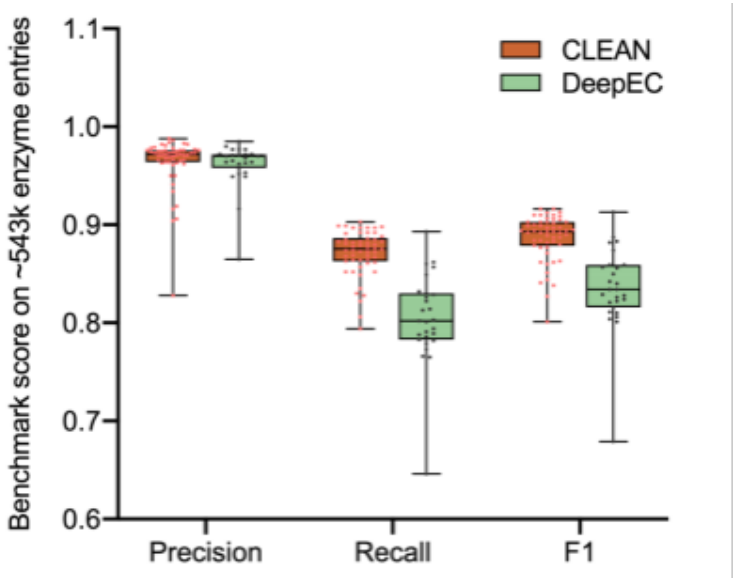

图 5. CLEAN 和 DeepEC 在 TrEMBL 大（543k 酶）数据集上的比较

如表2为本文复现工作后的指标数据，图 6为论文中作者对 CLEAN 的三个多标签准确度指标 (Precision、Recall 和 F1-score) 进行评估所制成的图，通过对比我们可以发现数据非常接近。

表 2. 对 New-392 与 Price-149 数据库上对 CLEAN 的三个多标签准确度指标进行评估

数据集	**precision**	**recall**	**F1**
New-392	0.596	0.479	0.497
Price-149	0.558	0.477	0.482

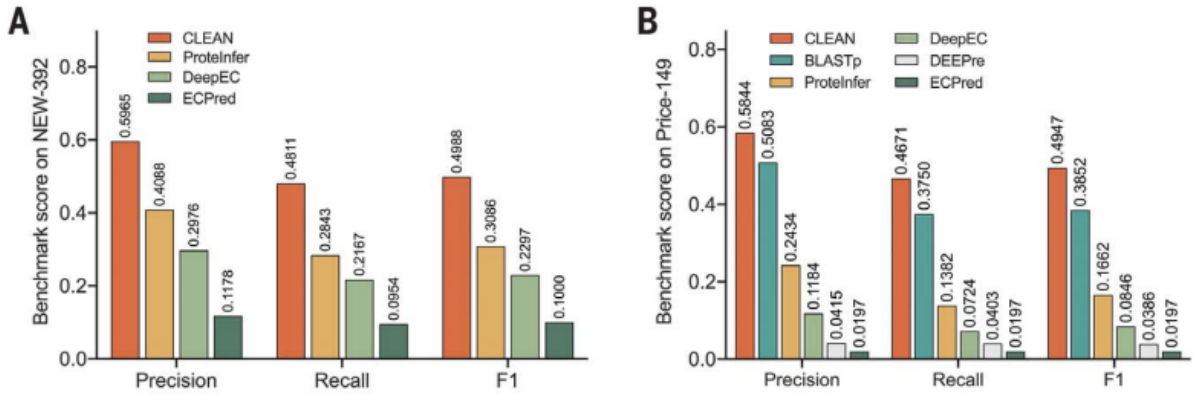

图 6. 对 CLEAN 的三个多标签准确度指标评估

在我复现的代码文件夹/result 下，有对 New-392 与 Price-149 数据集进行不同 EC 选择方法的结果，如图7所示，其中第一列是酶的 id，第二列是预测的 EC 编号与查询序列与所有 EC 号簇中心之间的成对距离。注意，在第 10 行酶 D0UZK2 中预测了三种酶的功能。

	C1	C2	C3	C4	C5
1	E0VIU9	EC:2.3.2.31/4.2112	<unset>	<unset>	<unset>
2	Q838J7	EC:4.2.1.113/6.3904	<unset>	<unset>	<unset>
3	P47482	EC:2.7.7.3/7.2080	EC:2.7.7.2/7.5292	<unset>	<unset>
4	B1VB82	EC:2.7.1.177/2.3378	<unset>	<unset>	<unset>
5	R9QMR1	EC:4.2.3.17/7.3352	EC:4.2.3.56/7.3756	<unset>	<unset>
6	R9QMW2	EC:4.2.3.17/7.1061	<unset>	<unset>	<unset>
7	E0W1I1	EC:2.7.11.1/6.0764	<unset>	<unset>	<unset>
8	Q66H88	EC:3.6.1.43/7.2053	<unset>	<unset>	<unset>
9	Q8IWZ4	EC:2.3.2.27/6.4954	<unset>	<unset>	<unset>
10	D0UZK2	EC:4.2.3.13/5.0487	EC:4.2.3.133/5.1969	EC:4.2.3.194/5.5152	<unset>
11	G0Y7D1	EC:4.2.3.27/6.8387	<unset>	<unset>	<unset>
12	A0A1V0E492	EC:4.2.3.194/5.8471	<unset>	<unset>	<unset>
13	I7GPX8	EC:3.1.7.11/6.9502	<unset>	<unset>	<unset>
14	A0A142BX70	EC:4.2.3.75/5.0969	<unset>	<unset>	<unset>
15	I7H727	EC:4.2.3.61/6.5063	<unset>	<unset>	<unset>
16	G0Y7D3	EC:4.2.3.108/6.9211	<unset>	<unset>	<unset>
17	A0A2R4QKX7	EC:4.2.3.101/6.0510	EC:4.2.3.81/6.0510	EC:4.2.3.55/6.0510	EC:4.2.3.94/6.0510

图 7. 结果示例截图

为了拟合 GMM 的 EC 数内高斯分布，本文首先随机选择一千个 EC 数，对于每个选择的 EC 数，本文计算 EC 数的嵌入和具有该 EC 的酶序列的嵌入之间的欧几里得距离数字。这些距离生成一种高斯分布，并形成 GMM 中的组成部分之一（图8左峰）。本文类似地拟合跨 EC 数分布：对于所选的每个 EC 数，EC 数和来自不同 EC 数的酶序列之间的距离生成另一个高斯分布并形成 GMM 的另一个组成部分（图8右峰）。此过程已重复 40 次以拟合 40 个 GMM，以减少随机误差。在 CLEAN 中，两个高斯分布分别代表了同一 EC 号内嵌入距离的经验分布和不同 EC 号间嵌入距离的经验分布。本文 EC 数嵌入与单个酶序列之间的欧几里德距离分布以直方图显示，其中 x 轴是距离，y 轴是计数。左峰是由匹配的 EC 编号和酶形成的，右峰是由不匹配的 EC 编号和酶形成的。

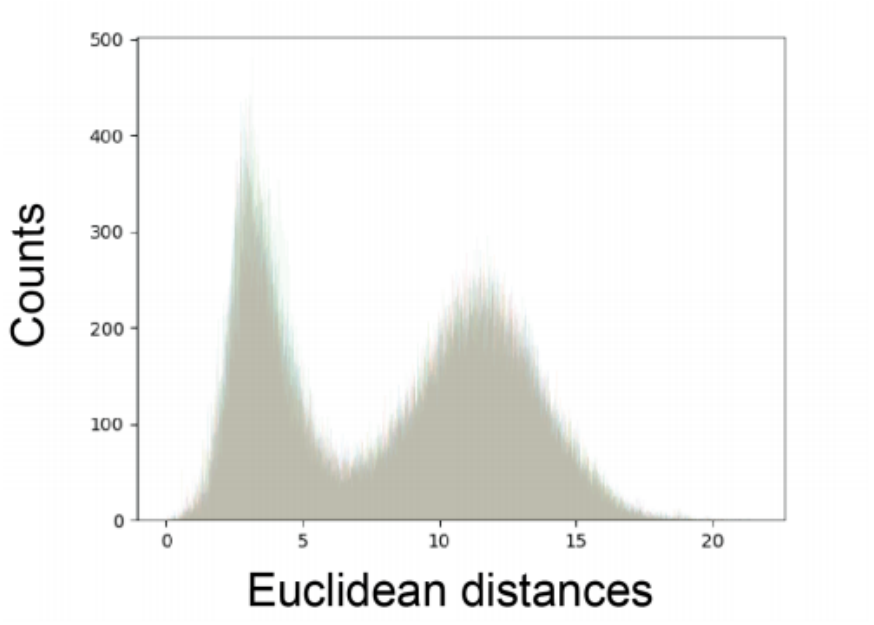

图 8. EC 数嵌入与单个酶序列之间的欧几里德距离分布

使用 GMM 是否可以通过检测低置信度来有效地防止过度预测，本文首先评估了 CLEAN 在不确定预测时是否能够标记低置信度分数。如果 CLEAN 的置信度估计是准确的，那么与 CLEAN 对这些抵抗蛋白的预测相关的置信度分数应该非常低，因为测试蛋白被排除在 CLEAN 的训练数据集中。本文的结果证实了本文的假设:CLEAN 对这些测试蛋白的预测显示出极低的置信分数，绝大多数预测低于 0.2 的置信分数 (图9左)。本文进一步构建了另一个测试数据集，CLEAN 的预测精度超过 0.95 作为阳性对照 (图9右)，并观察到这些蛋白质预测的置信度得分被富集到一个高置信度区域 (>0.9)。这些结果表明，CLEAN 的置信度估计信息量很大，并且与预测精度相关。

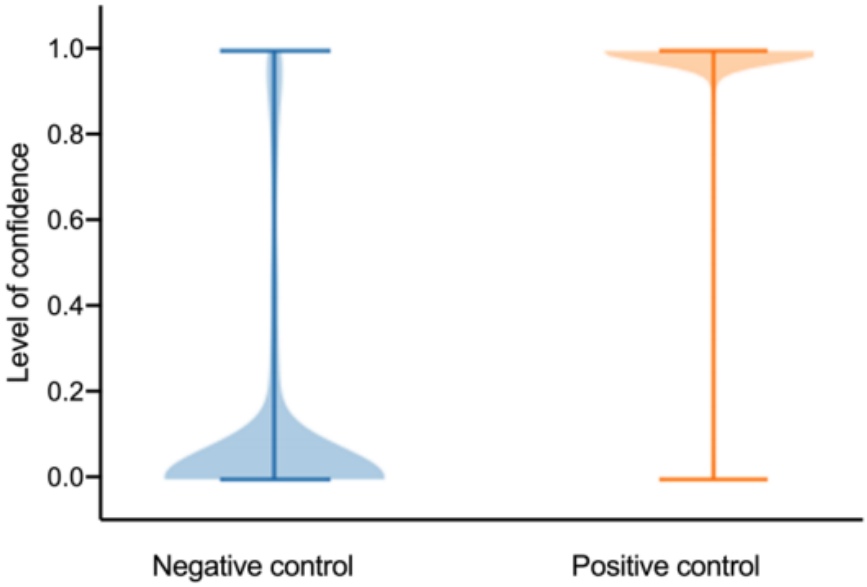

图 9. 极低精度数据集 (左) 与高精度数据集 (右) 置信度分布对比

为了避免过度预测，CLEAN 使用估计的置信度分数来指导其预测: 当与预测相关的置信度分数足够高时，它只将特定的 EC 数预测到第四位数水平，否则它只报告其对输入蛋白质的第三级 EC 数的预测。从表面上看，本文在组合测试数据集 (Price-149 和 New-392) 上重新运行 CLEAN，使得只有当其相关置信度得分 >0.5 时，它才能在第四个级别预测 EC 数，否则它将在第三个级别输出 EC 数。从图10本文观察到，在置信度估计的指导下，与 ProteInfer 和 DeepEC 相比，CLEAN 成功预测了更多的真阳性。这些结果表明，本文的不确定性量化算法使 CLEAN 能够实现自适应预测方案，在很大程度上避免了过度预测。

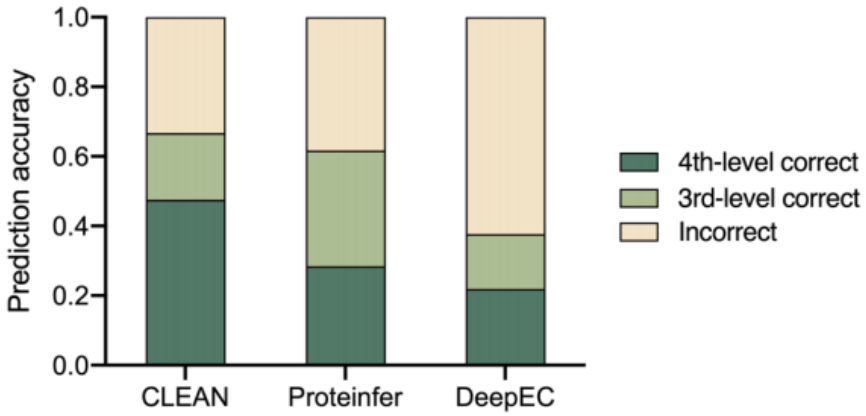

图 10. Price-149 和 New-392 组合数据集的四级 EC 号精度和三级 EC 号精度的分数

此外，在 Price-149 和 New-392 组合测试数据集上，作者进一步分析了预测精度与置信度的相关性。如图11所示，置信度越高，CLEAN 的预测精度越高。结果表明，当置信度较高时，CLEAN 可能是可靠的，这进一步证明了不确定度量化确实具有信息量，并且与预测精度呈正相关。

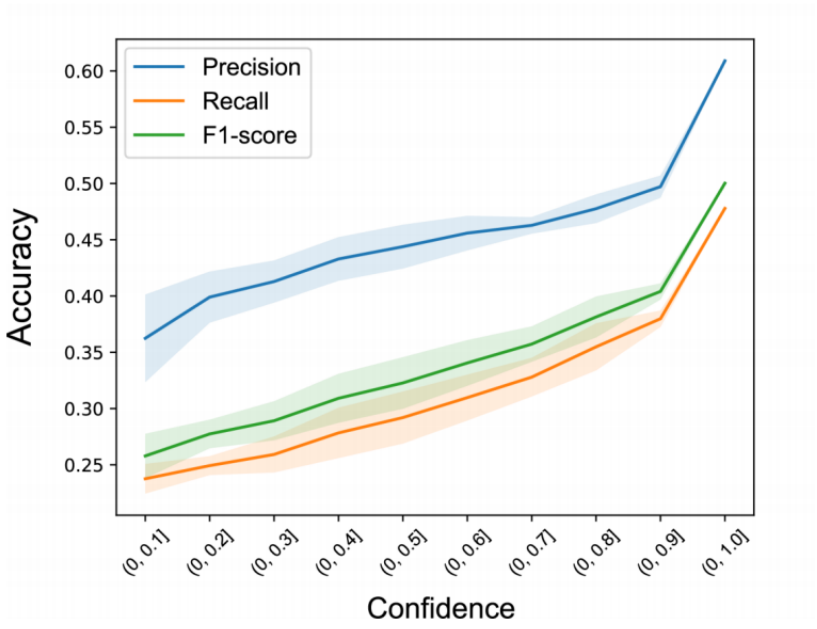

图 11. 置信度 vs 累积预测精度

6 总结与展望

CLEAN 将成为预测查询酶的催化功能的强大工具，这将极大地促进功能基因组学、酶学、酶工程、合成生物学、代谢工程和逆向生物合成等领域的研究。此外，CLEAN 所采用的通用语言模型表示结合对比学习工作流程可轻松适应其他不限于酶活性的预测任务，如功能目录（FunCat）和基因本体（GO）。

参考文献

[1] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[2] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[3] Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. *Proceedings of the National Academy of Sciences*, 116(28):13996–14001, 2019.

[4] Theo Sanderson, Maxwell L Bileschi, David Belanger, and Lucy J Colwell. Proteinfer, deep neural networks for protein functional inference. *Elife*, 12:e80942, 2023.