FSFP

一、实现目标

解决蛋白质工程中的数据稀缺问题：传统的蛋白质工程方法（如定向进化和理性设计）依赖于大量的实验数据，而 FSFP 则旨在通过极少的实验数据来优化蛋白质语言模型，从而降低蛋白质工程的门槛。

融合无监督和监督学习的优势：无监督的蛋白质语言模型（如 ESM-1v, ProGen, SaProt, ProtT5）可以独立于实验数据预测突变效应，但准确性有限；而监督的深度学习模型需要大量标注数据才能提高性能。FSFP 试图结合两者的优势，利用少量数据实现更准确的预测。

提高蛋白质适应性预测的准确性：通过元学习、学习排序和参数高效微调等方法，FSFP 可以显著提升各种蛋白质语言模型的性能，仅需数十个标记的单点突变数据即可实现性能提升。

二、实现思路：

元迁移学习（MTL）：FSFP 利用元学习来训练蛋白质语言模型，使其能够在少量数据的情况下快速适应新任务。通过构建辅助任务，FSFP 训练模型以更好地利用目标蛋白的标签数据。

学习排序（LTR）：FSFP 将适应性预测视为排序问题，通过 ListMLE 损失函数来训练模型，使模型能够根据突变体之间的相对有效性进行排序，而非关注具体的数值。

参数高效微调（LoRA）：通过 LoRA 技术，FSFP 在冻结大部分预训练模型参数的情况下，仅微调少量参数，避免了模型在少量数据下过拟合。

三、实现方法

主要包含三个阶段：为元学习构建辅助任务，在辅助任务上元训练 PLM，以及将 PLM 转换为目标任务。

1. 构建元学习辅助任务

检索相似实验数据集： 利用 PLM 对目标蛋白质的野生型序列或结构进行编码，并与 ProteinGym 数据库中的蛋白质进行相似度比较，选择最相似的两个蛋白质的突变数据集作为前两个辅助任务。

基于 MSA 估计突变效应： 使用 GEMME 算法，根据目标蛋白质的 MSA 信息，对候选突变体进行评分，并构建第三个辅助任务的伪标签数据集。

数据集划分： 将每个辅助任务的数据集随机划分为训练集和测试集。

2. 在辅助任务上进行元训练

使用 MAML 算法： 应用模型无关的元学习 (MAML) 算法，在构建的辅助任务上进行元训练，使 PLM 学习如何快速适应新的任务。

应用 LoRA： 为了防止 PLM 在少量训练数据上过拟合，使用低秩自适应 (LoRA) 技术将可训练的低秩分解矩阵注入 PLM，并将模型更新限制在这些少量可训练参数上。

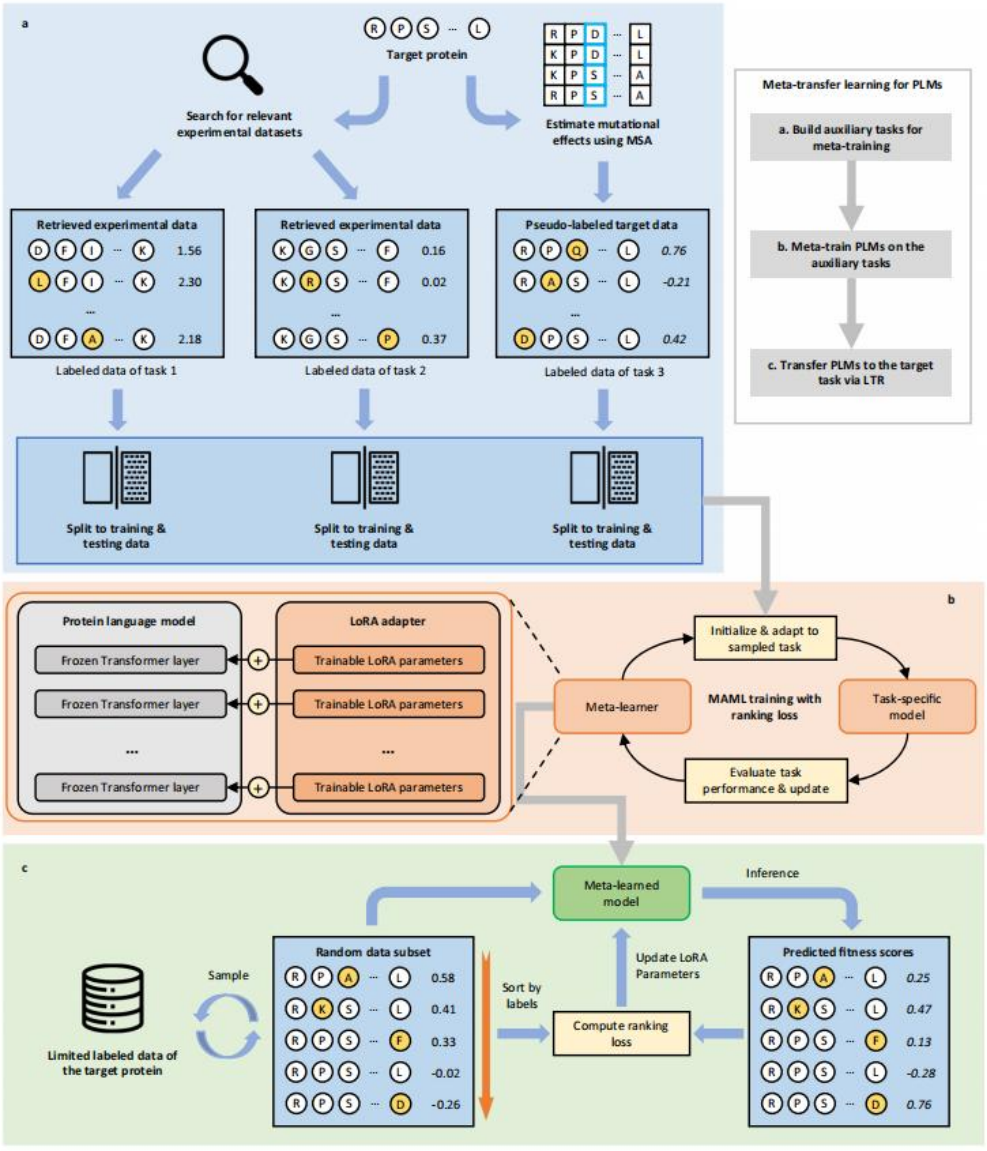
使用排序损失函数： 将适应性预测视为排序问题，并利用排序技术 (LTR) 计算列表排序损失 (ListMLE)，以学习对蛋白质适应性进行排序。

3. 将 PLM 转移到目标任务

使用目标训练数据： 将元训练后的 PLM 转移到目标少样本学习任务，即利用目标蛋白质的少量标注数据学习预测突变效应。

继续使用 LTR： 在目标任务中，继续使用 LTR 技术和排序损失函数进行训练。

模型框架如下：



a 根据目标蛋白的野生型序列或结构，检索两个相似蛋白的标记突变体数据集作为前两个任务。此外，基于 MSA 的方法用于估计候选突变体作为第三任务的伪标记的变体效应。

b MAML 算法用于在构建的任务上对 PLM 进行元训练，并最终将其优化为元学习者，为目标任务提供良好的参数初始化（右）。为了防止 PLM 在小的训练数据上过度拟合，LoRA 被应用于将模型更新约束为有限数量的参数（左）。

c 然后将元训练模型转移到目标少发射学习任务。

本策略将适应度预测视为一个排序问题，并利用 LTR 技术进行迁移学习和元训练。它训练 PLM 通过计算它们的预测和地面事实排列之间的列表式排序损失来排序适应度。

四、方法优势：

数据效率高：仅需目标蛋白质的少量标签数据（仅几十个单点突变体）即可显著提升模型性能。

泛化能力强：FSFP 训练的模型能够很好地泛化到训练数据中未出现的突变位点和突变组合。

适用性广：FSFP 可以应用于任何基于深度学习的蛋白质适应性预测模型，并与其他蛋白质语言模型兼容。

五、环境配置

cudatoolkit 11.8.0
learn2learn 0.2.0
pandas 1.5.3
peft 0.4.0
python 3.10
pytorch 2.0.1
scipy 1.10.1
scikit-learn 1.3.0
tqdm 4.65.0
transformers 4.29.2

六、数据集

包含 87 个数据集，完整 proteingym 数据集可在 https://drive.google.com/file/d/1Sbtlm0JnkSzNVMZiSn6OVw5PEg251LGu/view?usp=sharing 中找到。