测试集：

ITRAP_1：是基于 ITRAP 论文产生的数据集，其中的正例来自 ITRAP 论文提供的数据（GitHub - mnielLab/ITRAP_benchmark: This repo contains the 10x Genomic datasets filtered using ATRAP and ICON denoising frameworks. 中的 ***ITRAP.csv***）
因为其产生的负例不包含 HLA type，所以我通过他描述的方式产生了新的负例：
1） 通过错配 TCR、pMHC 产生 1：1 的负例；
2） 通过背景 TCR 库（GitHub - viragbioinfo/IMMREP_2022_TCRSpecificity 与我们使用的 TCR 库不同）中的 TCR 与正例中的 pMHC 组合，产生 1：1 的负例
最后去除其中 TCR 长度小于 10、大于 20 的数据。得到 12,807 条数据，其中正例 4,261 条，负例 8,546 条。

ITRAP_3：使用论文提供的参数（(1 ， 1，5）分别对应（umi_tra, umi_trb, umi_mhc）只有 umi 大于对应参数的细胞才会被选中以进行后续的筛选）进行筛选，获得正例。同时每个细胞中检测出的其他 pMHC 成为负例（这些负例中与正例相同的数据被去除），最后生成了 6047 个正例和 50292 个负例（去除 cdr3β 长度大于 20 小于 10 的序列）。

Small_ITRAP_3：将 background TCR 中出现的 TCR 从测试集 ITRAP_3 中去掉，获得 Small_ITRAP_3，共包含 3244 个正例和 19991 个负例。

测试方法：

OnlyPep：
1） 训练一个 peptide embedding model，代替原来的 pMHC embedding model，其中训练集由两部分组成：原来训练 pMHC embedding model 的训练集中的 peptide 和原来训练 pMHC embedding model 时用于生成负例的 natural peptide。
只使用 MLM 任务训练该模型（和 TCR embedding model 相同），且 peptide 的最大长度为 20，训练 peptide embedding model 的超参数均与训练 TCR embedding model 相同；

2）之后重新训练 TCR-pMHC prediction model，得到预测结果。

Few-shot：
因为 ITRAP_1 和 ITRAP_3 构建负例的方式不同，且模型训练时需要正负例一一对应，所以这里重新训练模型的方式也略有不同。

ITRAP_1：将数据集等分为 5 份，每次训练时，选一份作为测试集，另外 4 份作为额外的训练集。额外的训练集中负例被移除（这里是因为负例是人为创造的，后续再创造比较简单，而从原来的数据集中重新建立正负例一一对应的关系比较繁琐），重新根据规则创造 1:1 的负例，即错配和背景 TCR 库，若新创建的负例与原测试集中的正例相同，则重新创建负例（和原来的训练相同，训练时，负例每 5 个 epoch 重新生成一次）。随后将额外的训练集混入原训练集中，重新训练模型并测试结果。循环 5 次。

训练集：
正例（70423+4261*0.8）*0.9= **66449**
*其中 4261×0.8 是因为五折交叉验证，取 4 份做额外的训练集，整体×0.9 是因为训练时有 10%的数据用做验证集；

负例 （70423+4261*0.8*2）*0.9*epoch 数/5
*其中 4261*0.8*2 是因为生成负例时有两种方式，*epoch 数/5 是因为·每 5 个 epoch 重新生成一次负例，每次训练大概 50-70 个 epoch，所以正负例的比例大概为 **1:10**；

测试集：12807*0.2 = 2561
取一份做测试集

ITRAP_3 and small_ITRAP_3：在 **ITRAP_3** 中，每个正例会对应 0-n 个负例（对应到实验中则为每种 T cell 中会检测出多种 pMHC）。所以我们通过 TCR 进行分组，即组间不会共享相同的 TCR 序列。同 ITRAP_1 相同，ITRAP_3 也是将数据集为 5 份，每次训练时，选一份作为测试集，另外 4 份作为额外的训练集。额外的训练集中，没有对应负例的 TCR 将被移除（由于我们的训练方式，没有对应的负例无法进行训练，但这个步骤中去掉的数据极少），有对应负例的 TCR 会在所有的负例中随机选择一个作为本次训练的负例（训练时，每 5 个 epoch 会重新选择一次负例），将额外的训练集混入原训练集中，重新训练模型并测试结果。循环 5 次。

训练集：
正例（70423+6047*0.8）*0.9= 67735
*其中 6047×0.8 是因为五折交叉验证，取 4 份做额外的训练集，整体×0.9 是因为训练时有 10%的数据用做验证集；

负例 （70423+6047*0.8）*0.9*epoch 数/5
* *epoch 数/5 是因为·每 5 个 epoch 重新生成一次负例，每次训练大概 50-70 个 epoch，所以正负例的比例大概为 **1:10**；

测试集：56339*0.2 = 11268
取一份做测试集

Total-Learn：
Total-learn 是在 **Few-shot** 的设定下，在训练时只使用额外的训练集。

训练集和测试集均为 **Few-shot** 的设定下去除原始的训练集（70423），但 Total-learn 设定下模型训练的 epoch 更少（因为有设定 early stop，6 个 epoch 内验证集损失不改善就停止训练），平均训练 epoch 在 15-25 之间，所以正负例比例大概为 **1：3**。