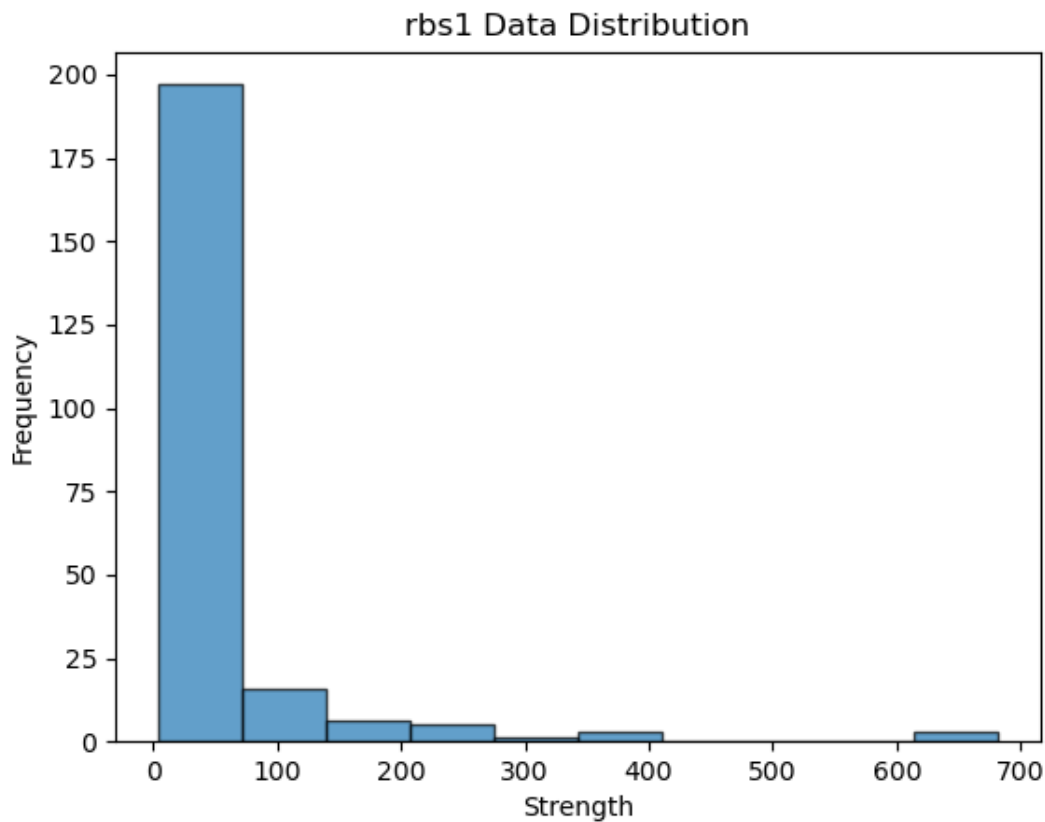


数据集

rbs 317 条长度为 30的序列



编码方式

- one-hot 编码

```
char_map = {  
    'A': [1, 0, 0, 0],  
    'T': [0, 1, 0, 0],  
    'C': [0, 0, 1, 0],  
    'G': [0, 0, 0, 1]  
}
```

- dimer 编码(理化特性)

用RNA的11个理化性质，用 长度为11 的向量进行编码

```
char_map = {  
    'GG': [-0.01, -1.78, 3.32, 0.3, 12.1, 32.0, -11.1, -12.2, -29.7,  
-3.26, 0.17],  
    'GA': [0.07, -1.7, 3.38, 1.3, 9.4, 32.0, -14.2, -13.3, -35.5, -2.35,  
0.1],  
    'GC': [0.07, -1.39, 3.22, 0.0, 6.1, 35.0, -16.9, -14.2, -34.9, -3.42,  
0.26],  
    'GT': [0.23, -1.43, 3.24, 0.8, 4.8, 32.0, -13.8, -10.2, -26.2, -2.24,  
0.27],  
    'AG': [-0.04, -1.5, 3.3, 0.5, 8.5, 30.0, -14.0, -7.6, -19.2, -2.08,  
0.08],  
}
```

```

        'AA': [-0.08, -1.27, 3.18, -0.8, 7.0, 31.0, -13.7, -6.6, -18.4,
-0.93, 0.04],
        'AC': [0.23, -1.43, 3.24, 0.8, 4.8, 32.0, -13.8, -10.2, -26.2, -2.24,
0.14],
        'AT': [-0.06, -1.36, 3.24, 1.1, 7.1, 33.0, -15.4, -5.7, -15.5, -1.1,
0.14],
        'CG': [0.3, -1.89, 3.3, -0.1, 12.1, 27.0, -15.6, -8.0, -19.4, -2.36,
0.35],
        'CA': [0.11, -1.46, 3.09, 1.0, 9.9, 31.0, -14.4, -10.5, -27.8, -2.11,
0.21],
        'CC': [-0.01, -1.78, 3.32, 0.3, 8.7, 32.0, -11.1, -12.2, -29.7,
-3.26, 0.49],
        'CT': [-0.04, -1.5, 3.3, 0.5, 8.5, 30.0, -14.0, -7.6, -19.2, -2.08,
0.52],
        'TG': [0.11, -1.46, 3.09, 1.0, 9.9, 31.0, -14.4, -7.6, -19.2, -2.11,
0.34],
        'TA': [-0.02, -1.45, 3.26, -0.2, 10.7, 32.0, -16.0, -8.1, -22.6,
-1.33, 0.21],
        'TC': [0.07, -1.7, 3.38, 1.3, 9.4, 32.0, -14.2, -10.2, -26.2, -2.35,
0.48],
        'TT': [-0.08, -1.27, 3.18, -0.8, 7.0, 31.0, -13.7, -6.6, -18.4,
-0.93, 0.44]}

```

- triplet 编码

将三个碱基为一组进行编码，用 长度为64 的向量

表示如'AAA'会被编码为[1,0,...,0]

一个长度为30的序列会被编码成为一个64*28的矩阵

模型

EnsembleNet_M

EnsembleNet_M 模型是一个集成模型，它由两个 MobileNet 模型（即 `self.mobilenet_dimer` 和 `self.mobilenet_triplet`）组成。这两个模型分别以不同的方式编码输入数据，一个使用 "dimer" 编码，另一个使用 "triplet" 编码。

整体模型的原理如下：

1. 对于输入的 "dimer" 数据 (`x_dimer`)，通过 `self.mobilenet_dimer` 进行前向传播。
`self.mobilenet_dimer` 是一个 MobileNet 模型，它使用修改后的 MobileNetV2 架构来处理输入数据。具体地，它首先通过一个自定义的卷积层修改输入通道数，然后使用预训练的 MobileNetV2 模型提取特征，并去掉最后的分类器层。
2. 类似地，对于输入的 "triplet" 数据 (`x_triplet`)，通过 `self.mobilenet_triplet` 进行前向传播。`self.mobilenet_triplet` 也是一个 MobileNet 模型，它与 `self.mobilenet_dimer` 结构相同，但是处理的是 "triplet" 数据。
3. 在每个模型中提取的特征上，使用自适应平均池化层将特征图大小调整为 1x1。
4. 将经过池化后的 "dimer" 特征和 "triplet" 特征在第一个维度上拼接起来，形成一个新的特征张量。
5. 将拼接后的特征张量展平成一维向量，作为输入传递给全连接层 (`self.fc`)。
6. 最终，通过全连接层获得模型的输出。

```
class EnsembleNet_M(nn.Module):
```

```

def __init__(self, num_classes, input_channels):
    super(EnsembleNet_M, self).__init__()
    self.mobilenet_dimer = MobileNet(num_classes, input_channels)
    self.mobilenet_triplet = MobileNet(num_classes, input_channels)
    # Define an adaptive average pooling layer with output size 1x1
    self.pool = nn.AdaptiveAvgPool2d(1)
    self.fc = nn.Linear(1280 * 2, 1)

    def forward(self, x_dimer, x_triplet):
        # Extract the features from both mobilenets and pool them
        features_dimer =
self.pool(self.mobilenet_dimer.mobilenet.features(x_dimer))
        features_triplet =
self.pool(self.mobilenet_triplet.mobilenet.features(x_triplet))
        # Concatenate the features along the channel dimension and flatten them
        features = torch.flatten(torch.cat([features_dimer, features_triplet],
dim=1), 1)
        # Use the fully connected layer to get the final output
        out = self.fc(features)
        return out

```

结果

	MobileNet-dimer	MobileNet-triplet	Model: EnsembleNet-M
MAE	88.9604	61.4571	68.0180
PCC	0.3745	0.4210	0.3999
R ²	-0.4886	0.1190	0.1177