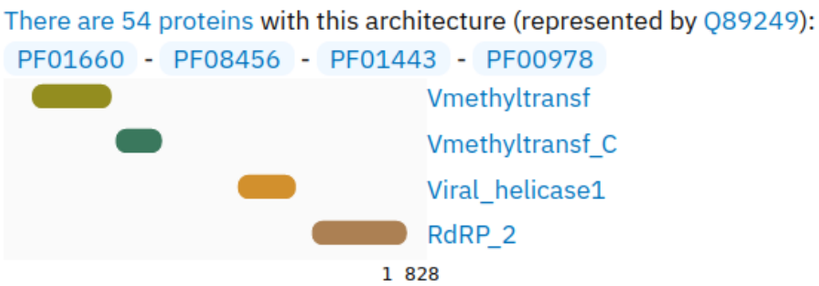
Signals 4

Для работы возьмём семейство `PF00978`, включающее в себя домен `RdRP_2` - то есть мы сегодня смотрим на белки, содержащие в себе фргамент, являющийся РНК-зависимой РНК-полимеразой. Среди названных белков было выбрано подсемейство таких, что обладают следующей архитектурой:

```
N->Vmethyltransf->Vmethyltransf_C->Viral_helicase_2->RdRP2->C
```



То есть белки этого подсемейства являются РНК-зависимыми РНК-полимеразами, способными, к тому же, метилировать РНК. Вероятно, такие химические модификации помогают вирусной РНК избежать распознавания клеточными РНКазами - а для одноцепочечных (+)РНК-вирусов, из которых происходит большая часть из 54 белков подсемейства, это очень важно.

Почему я выбрал четырёхдоменный, а не двудоменный вариант, как того требует задание? Доступные дву- и трёхдоменные варианты варианты мне показались недостаточно надёжными: между доменами были слишком большие расстояния, а поскольку белки подсемейства доступны на InterPro в виде разрозненных последовательностей, а не готовое выравнивание, попытка "в лоб" отровнять эти последовательности могла привести к получению большого количества шума.

Выравнивание

Среди последовательностей белков подсемейства были по половине белков были определены в тренировочную выборку и в набор белков для контроля. Далее для `train`-выборки с помощью `mafft` с умолчательными параметрами было получено множественное выраванивание. Из последнего были вырезаны столбцы: а) не соответствующие каким-либо доменам, б) состоящие целиком или почти целиком из гэпов, в) низкоконсервантивные участки. На этом этапе чрезвычайно важно **не выравнивать все белки подсемейства вместе** и только потом оставлять от этого выравнивания `train`-часть: так у нас будет происходить утечка данных из контрольной выборки в тренировочную, и на этапе тестирования модели получатся завышенные метрики.

Таким образом, как `train` рассматривалась половина последовательностей белков, образующих наше подсемейство, как `control` - другая половина, как `test` - последовательности белков подсемейства, находящиеся в Swiss-Prot (почти все относятся к негативному классу) + `control`.

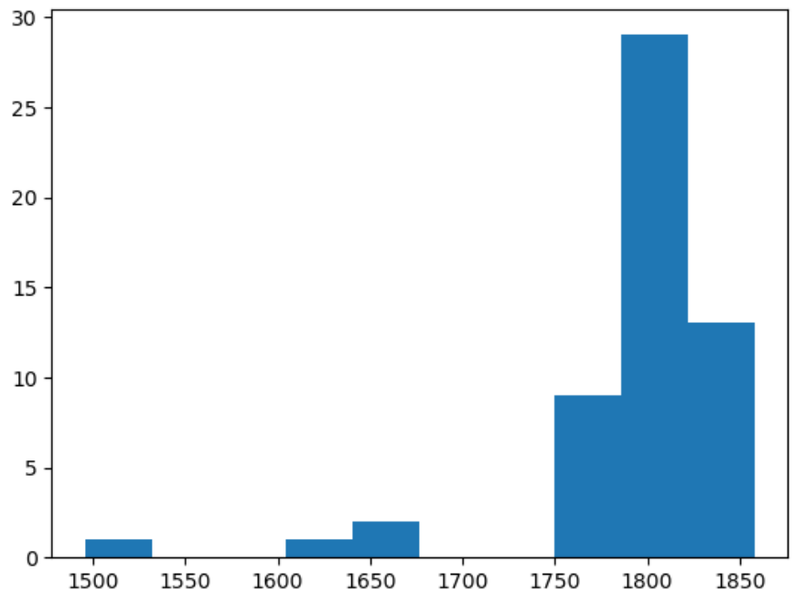
Построение HMM-профиля

Расчёт HMM-профиля, его калибровка и получение предсказаний на контрольной и тестовой выборках производились следующим набором команд:

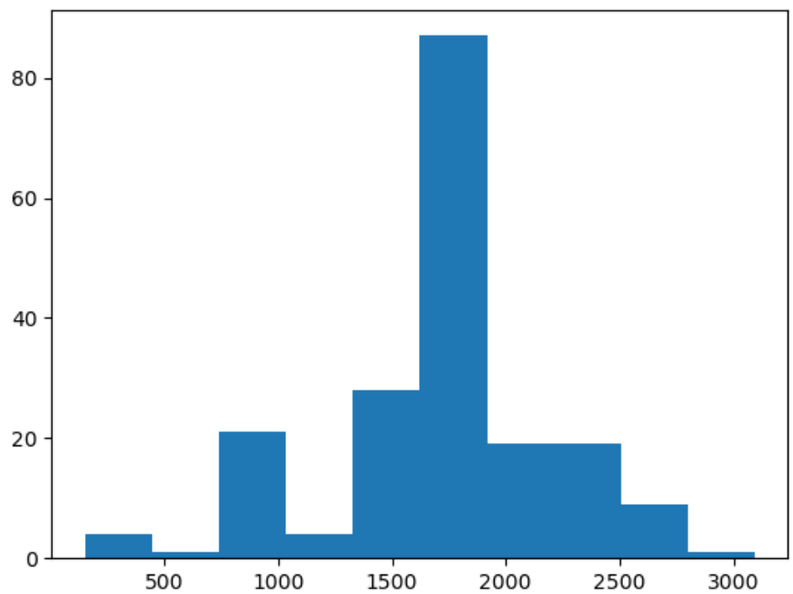
```
hmm2build -g hmm.txt train.fa
hmm2calibrate hmm.txt
hmm2search --cpu 1 hmm.txt control.fa > hmm_control.txt
hmm2search --cpu 1 hmm.txt test.fa > hmm_test.txt
```

Длины

Тут интересно посмотреть, как распределены длины в тренировочной и тестовой выборках.
Распределение длин белков тренировочной выборки:



Распределение длин белков во всём семействе:

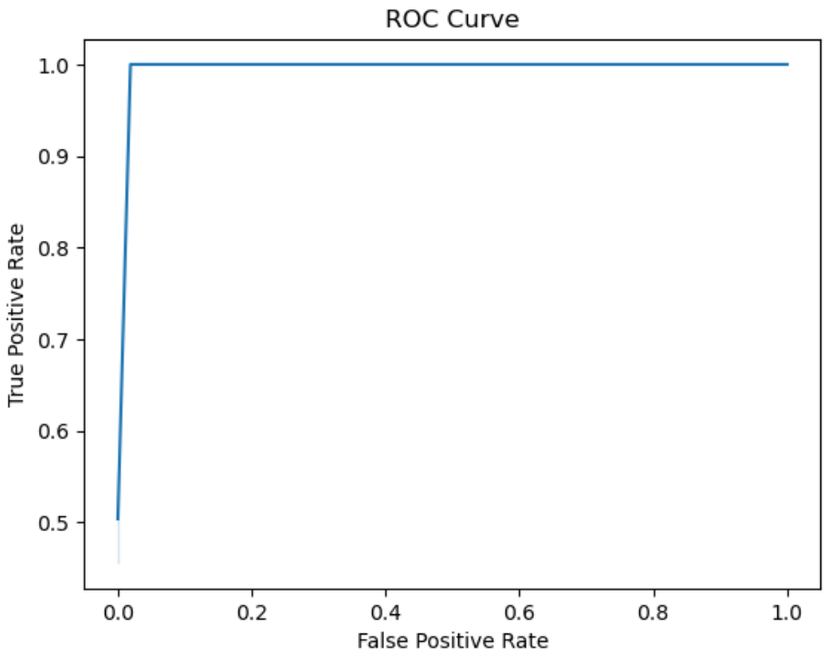


Метрики и подбор порогов

Посмотрим на два принятых способа оценки качества бинарных классификаторов в приложении к нашей HMM-модели.

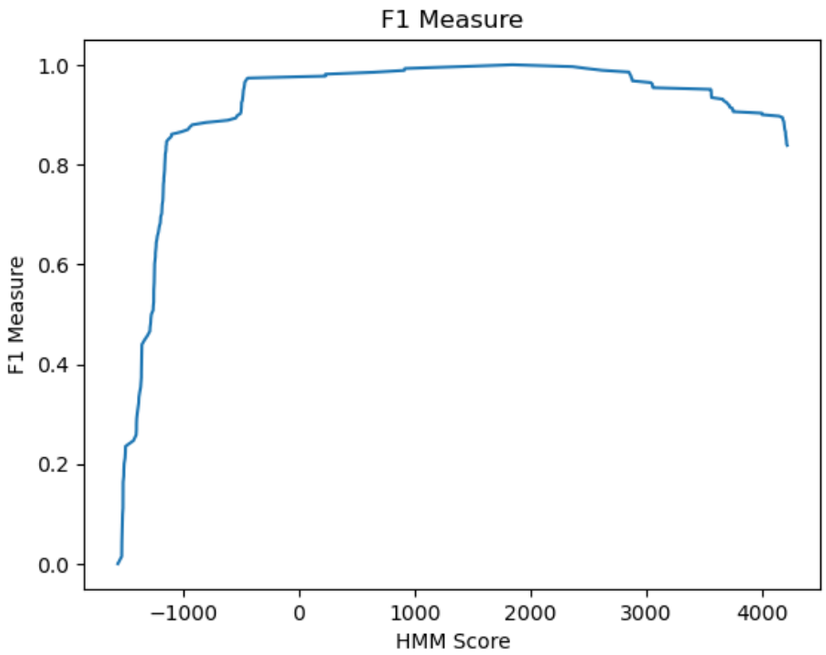
ROC AUC

Минимальным порогом на `score`, при котором достигается оптимум TPR и FPR, является `917.3`.



F1-мера

Для этой характеристики оптимальным порогом оказывается `1845.7`.



Метрики позволяют заключить, что качество классификатора получилось довольно высоким, особенно учитывая простоту его построения. Возможно, именно по этой причине для задач аннотации биоинформатика традиционно использует HMM-, а не нейросетевые подходы.

Файлы

Лежат в директории:

```
tree
.
├── control.fa # последовательности контроля
├── hmm2_control.txt # выдача модели на контрольной выборке
├── hmm_test.txt # выдача модели на тестовой выборке
├── hmm_out.txt # "веса" модели
```

```
├── test.fa # тестовая выборка
└── train.fa # тренировочная выборка
```