

Backend реализован на FastAPI, frontend – на React. Одностраничное приложение разворачивается на порте 3000 и посылает get-запросы к API (порт 8000).

Функция для поиска консервативных участков. На вход подаём последовательность ДНК, на выходе получаем отформатированную строку со списком подпоследовательностей и индексами позиций, а также сам массив этих позиций. Проходимся циклом по последовательности. Мы храним значения левой l и правой r границ рассматриваемого участка. Изначально $r - l = 9$, но если следующий kmer классифицирован как 1, то мы увеличиваем r. Делаем так до тех пор, пока не встретится 2 участка класса 0 подряд. Добавляем подстроку seq[l:r] и индекс её позиции l к возвращаемым результатам.

```
def find_conservative(seq):
    indexes = []
    output_str, conserv = "", ""
    cnt = 0
    l, r = -1, -1
    for i in range(len(seq) - k + 1):
        label = classifier(**tokenizer(seq[i:i+k],
return_tensors="pt")).logits.argmax().item()
        if label == 1:
            cnt = 0
            if l == -1:
                l = i
                r = i + k - 1
                conserv = seq[l:r+1]
            else:
                r = i + k - 1
                conserv += seq[r]
        else:
            if i <= r:
                continue
            cnt += 1
            if cnt > 1:
                if l != -1:
                    r -= 1
                    conserv = conserv[:-1]
                    indexes.append((l, r))
                    output_str += "{0} {1}\n".format(conserv, l)
                    conserv = ""
                    l, r = -1, -1
                continue
            if l != -1:
                r += 1
```

```

        conserv += seq[r]
    if l != -1:
        indexes.append((l, r))
        output_str += "{0} {1}\n".format(conserv, l)
    return output_str, indexes

```

Функция для генерации. На вход принимает последовательность, режим, длину, строку позиций через пробел. Если задан режим добавления, то проходимся циклом по последовательности и генерируем к каждому 8-меру 9 нуклеотид, пока не сгенерируем количество, заданное длиной. Если задан режим изменения и позиции, то генерируем нуклеотиды на данных позициях. Если же позиции не заданы, то генерируем нуклеотиды в тех местах, где классификатор относит 9-мер к классу 0, т. е. считает участок неконсервативным.

```

def generate_sequence(seq, mode, leng, pos):
    generated_seq = seq
    if mode == 'add':
        for i in range(leng):
            nucl_id = generator(**tokenizer(generated_seq[i:i+k-1],
return_tensors="pt")).logits.argmax().item()
            generated_seq += id2label[nucl_id]
    else:
        if pos:
            for p in pos.split(' '):
                nucl_id = generator(**tokenizer(generated_seq[int(p)-k:int(p)],
return_tensors="pt")).logits.argmax().item()
                generated_seq = generated_seq[:int(p)] + id2label[nucl_id] +
generated_seq[int(p)+1:]
            return generated_seq
        for i in range(k, len(seq)):
            if classifier(**tokenizer(generated_seq[i-k:i],
return_tensors="pt")).logits.argmax().item() == 0:
                nucl_id = generator(**tokenizer(generated_seq[i-k:i],
return_tensors="pt")).logits.argmax().item()
                generated_seq = generated_seq[:i] + id2label[nucl_id] +
generated_seq[i+1:]
    return generated_seq

```

В TypeScript имеем 2 основных компонента: Prediction и Generation

В компоненте Prediction заданы состояния inputState (состояние строки ввода), outputState (состояние выходного текстового поля) и indexes (массив позиций). Функция writeSequence изменяет состояние строки ввода при наборе букв пользователем, findConservative посылает get-запрос к API и выводит полученный результат с подсветкой текста, вызывается при нажатии на кнопку. В компоненте Generation заданы состояния inputSeq (входная

последовательность), generatedSeq (сгенерированная последовательность), leng (заданная длина) и positions (заданные позиции). Функции writeLen, writeSequence, writePositions изменяют состояния входных данных при вводе букв в соответствующих полях, функции addSeq и changeSeq вызываются при нажатии на кнопки “Добавить” и “Изменить” соответственно, они посылают get-запросы к серверу с входными данными и изменяют состояние выходного текстового поля.