

Язык запросов для геномного браузера GemlBee

Выполнил:

Егор Горбунов

Руководитель:

Олег Шпынов

20 мая 2016 г.

Геномный браузер

- ▶ **Геномный браузер** – приложение, отображающее различную информацию о геноме, привязанную к конкретным локациям этого генома
- ▶ USCS, Ensembl, BioViz, ..., **GemlBee**, ...

Текущий участок хромосомы

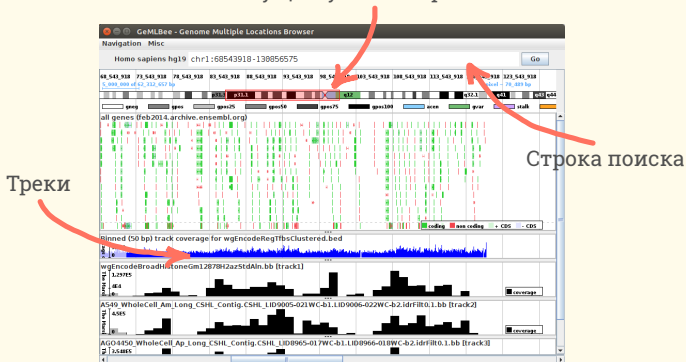


Рис. 1: GemlBee

Треки

- ▶ **Трек** — отдельная дорожка, отображаемая геномным браузером, заключающая в себе какую-то информацию про геном
- ▶ Бывают разные, в зависимости от этой информации
 - ▶ Какая-либо статистическая информация о нуклеотидах в тех или иных частях днк
 - ▶ Карты с конкретными генами с привязкой к участкам днк
 - ▶ ...
- ▶ Нам интересны треки, которые полезны при исследовании эпигенетических изменений (экспрессии генов)

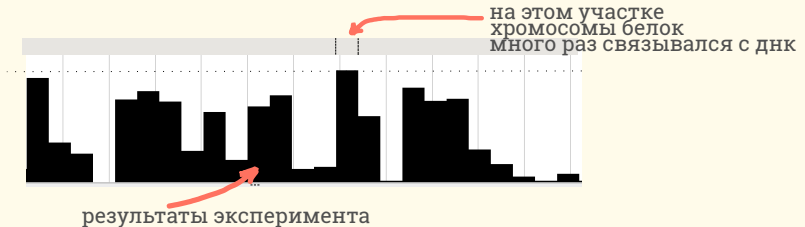


Рис. 2: Пример трека, который нас интересует

Идея: имея результаты нескольких экспериментов, представленных в виде треков, было бы удобно уметь как-то комбинировать эти результаты и совершать над ними операции с целью упрощения визуального анализа результатов в целом.

- ▶ Предложить операции, которые можно совершать над треками
- ▶ Сделать язык, позволяющих описывать эти операции
- ▶ Встроить этот язык в геномный браузер GEM1Bee

- ▶ Каждый трек – это вектор из N чисел
- ▶ Арифметические операции ($\circ \in \{+, -, \cdot, /\}$):
 $(a_1, a_2, \dots, a_N) \circ (b_1, b_2, \dots, b_N) = (a_1 \circ b_1, a_2 \circ b_2, \dots, a_N \circ b_N)$
- ▶ Операции сравнения треков: $<, >, \dots$. Результат – **предикат** – множество отрезков генома, на котором результат сравнения истинный
- ▶ Логические связки \vee, \wedge, \neg
- ▶ Условная генерация трека **предикату**

```
a := 2 + 2
```

```
b := if (a < 1) then 1 else 3
```

```
pred := (a < b) AND (b == c)
```

```
c := (b + (if (pred) then b else a) * b) / a
```

```
show pred
```

```
show c
```

Пример

- ▶ 2 разных эксперимента по нахождению регуляторных участков днк медуллобластомы
- ▶ генерация треков помогает смотреть на различия результатов и пр.

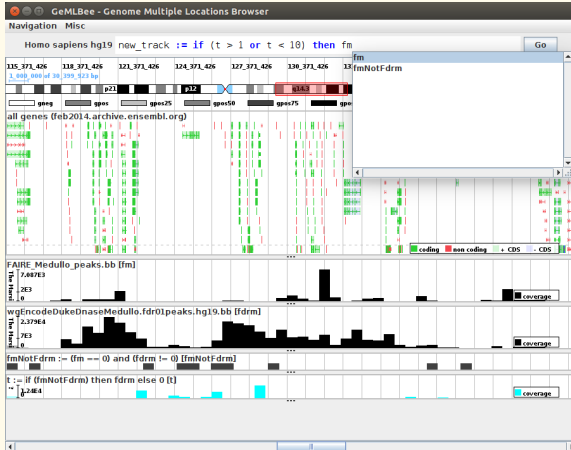


Рис. 3: Скриншот программы

Что сделано:

- ▶ Реализован интерпретатор языка
- ▶ Язык встроен в десктопную версию GemlBee
- ▶ Добавлена подсветка синтаксиса и автодополнения
- ▶ <https://github.com/egorbunov/gemlbee>

Что дальше:

- ▶ Добавить поддержку в web версии GemlBee

Что узнал и что использовал:

- ▶ Kotlin
- ▶ Swing
- ▶ Parsing expression grammars