**КУРСОВОЙ ПРОЕКТ**

**Variant Caller**

по дисциплине «Конструирование программного обеспечения»

Выполнили:

студенты гр. 3530904/80103 Анциферов М.А.

Лещенко К.В.

Назаров Д.А.  
 Цай Г. К.

Руководитель: Иванов А. С.

**Оглавление**

[**1. Задача проекта 3**](#_Toc59084059)

[**2. Требования к проекту 3**](#_Toc59084060)

[**3. Процесс разработки 3**](#_Toc59084061)

[**4. Описание проекта 4**](#_Toc59084062)

[**4.1 Формат входных файлов 4**](#_Toc59084063)

[**4.2 Взаимодействие классов 4**](#_Toc59084064)

[**4.3 Работа программы 5**](#_Toc59084065)

[**4.4 Упрощенная схема алгоритма 6**](#_Toc59084066)

[**4.5 Тестирование 7**](#_Toc59084067)

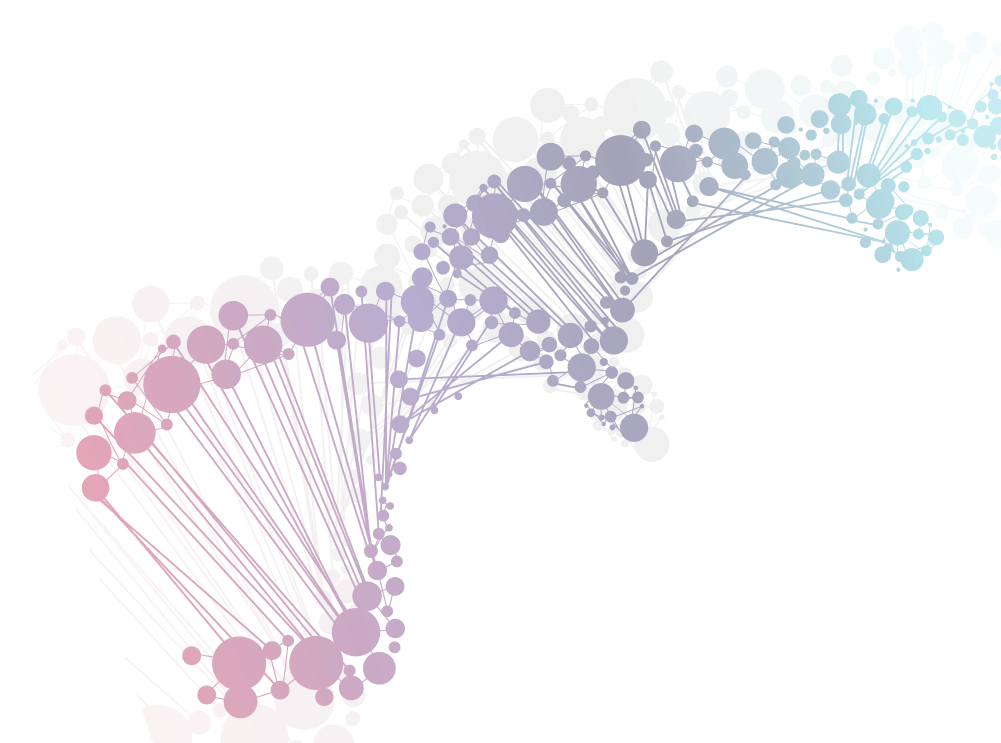
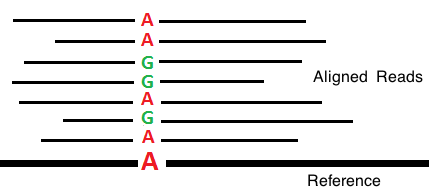
[**4.6 Сборка на GitHub 7**](#_Toc59084068)

[**4.7 Используемые технологии 8**](#_Toc59084069)

[**5. Развитие проекта 8**](#_Toc59084070)

[**6. Итог 8**](#_Toc59084071)

# 1. Задача проекта

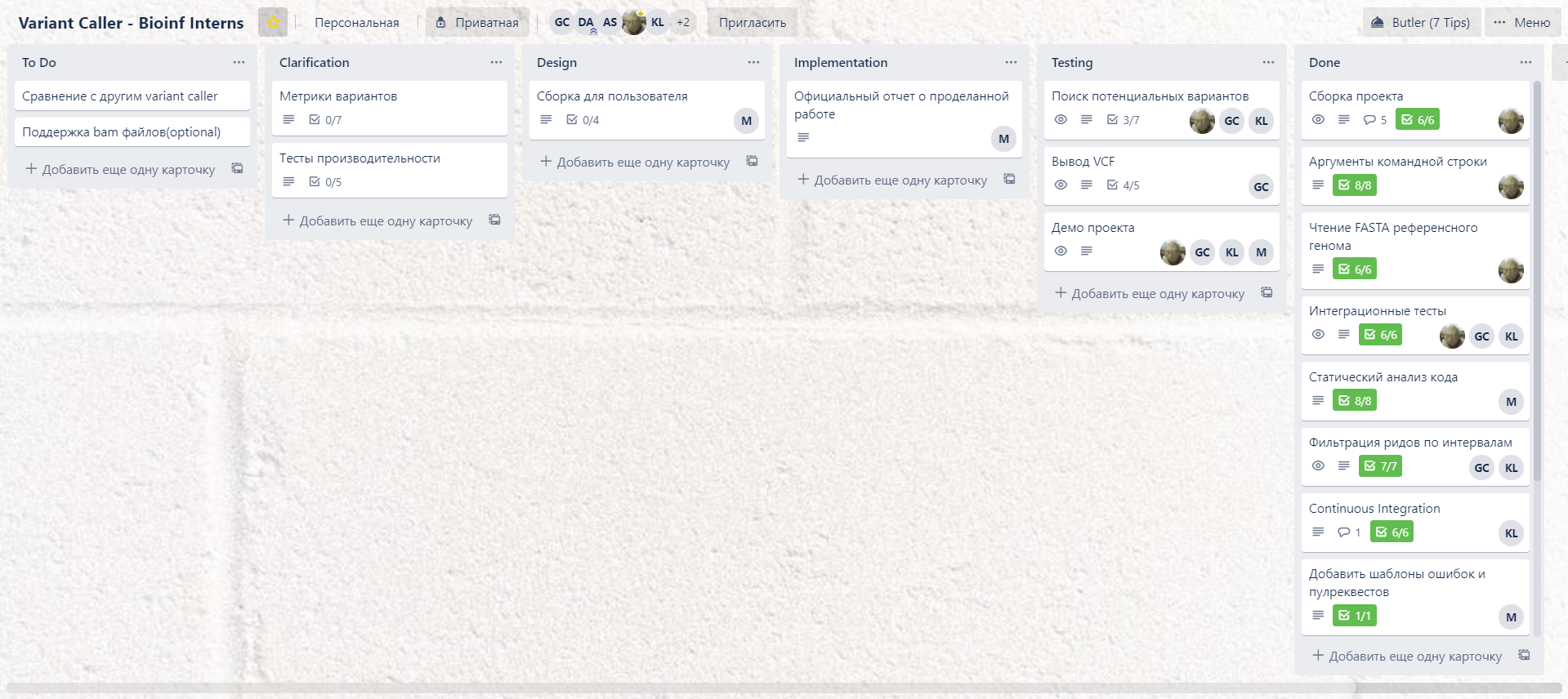
Наш проект называется Variant Caller. Главная цель проекта – создать утилиту, которая позволит выполнять поиск SNP вариаций. 

# 2. Требования к проекту

* Запуск из командной строки с параметрами
* Нахождение single-nucleotide polymorphism (SNP) вариации
* Чтение входных данных:
* SAM (aligned reads)
* BED (интервалы)
* FASTA (reference genome)
* Запись результата в файл формата VCF

# 3. Процесс разработки

Процесс разработки можно охарактеризовать как agile. Если точнее, то мы использовали простой SCRUM-board, в котором создавали задачи и распределяли их между участниками команды. Также у нас были спринты и ревью написанного кода.



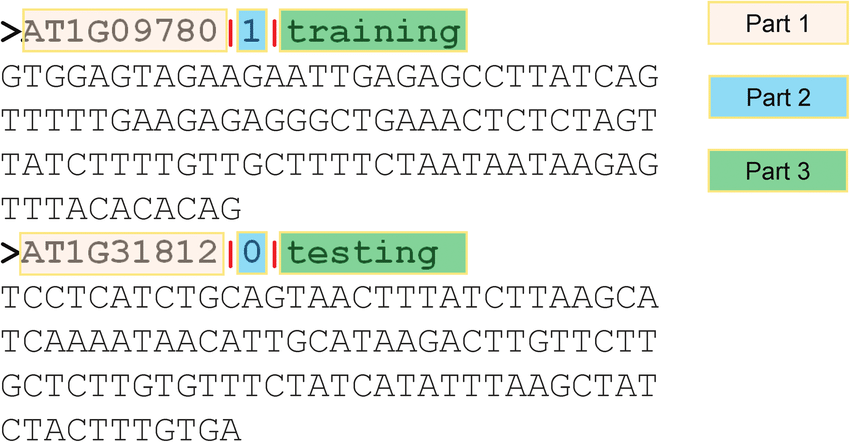
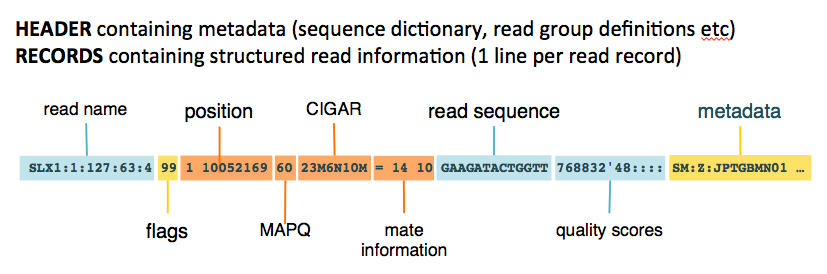
SCRUM-board

# 4. Описание проекта

# 4.1 Формат входных файлов

На вход программа получает референс геном в FASTA файле, а также выровненные риды в SAM файлах и интервалы в BED файлах.

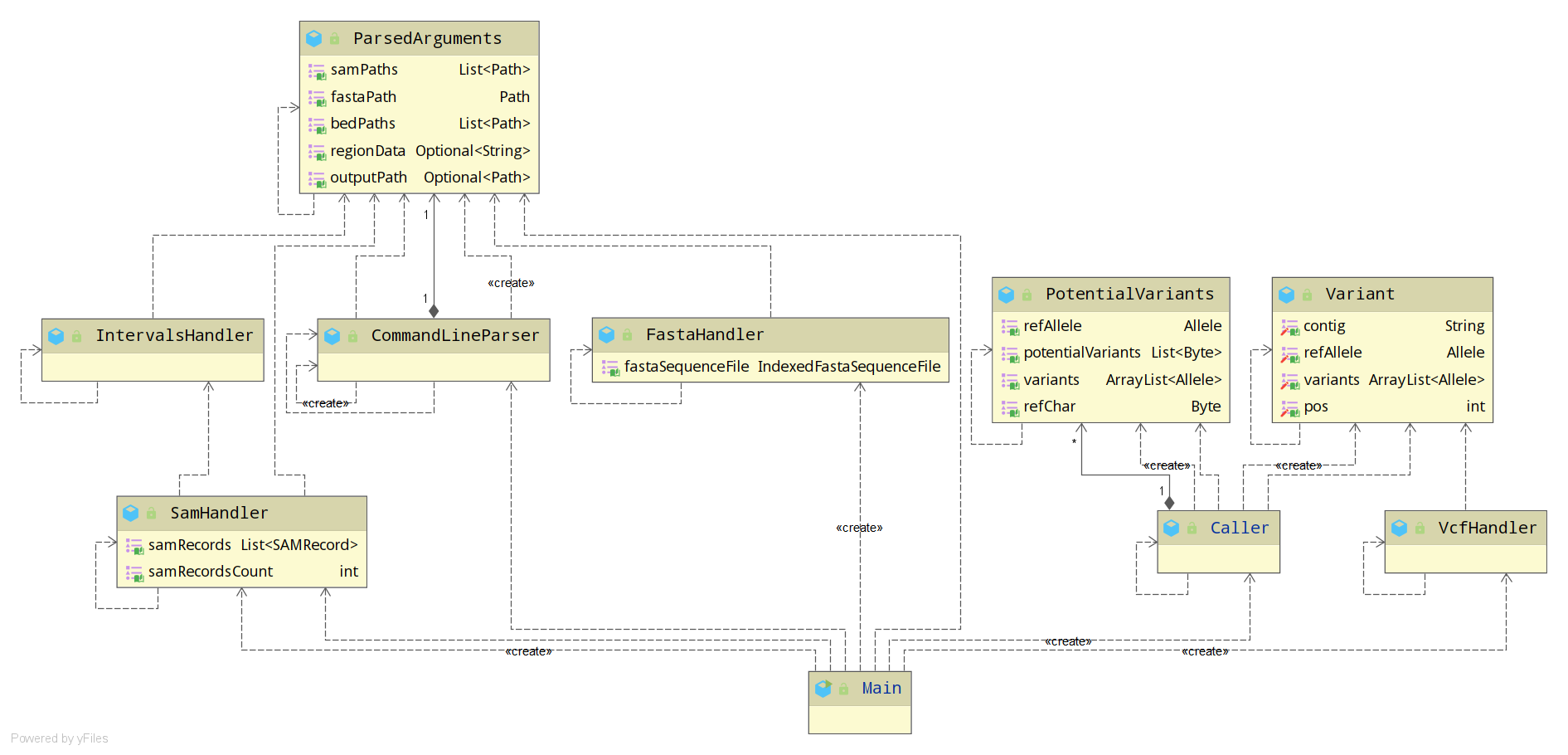
* В FASTA файле в первой строчке содержится имя последовательности и комментарий. Дальше идет сама последовательность.
* В SAM файле содержится имя последовательности, позиция рида на референсе, данные о качестве, и другая информация.
* Еще есть BED файлы в которых содержатся интервалы.



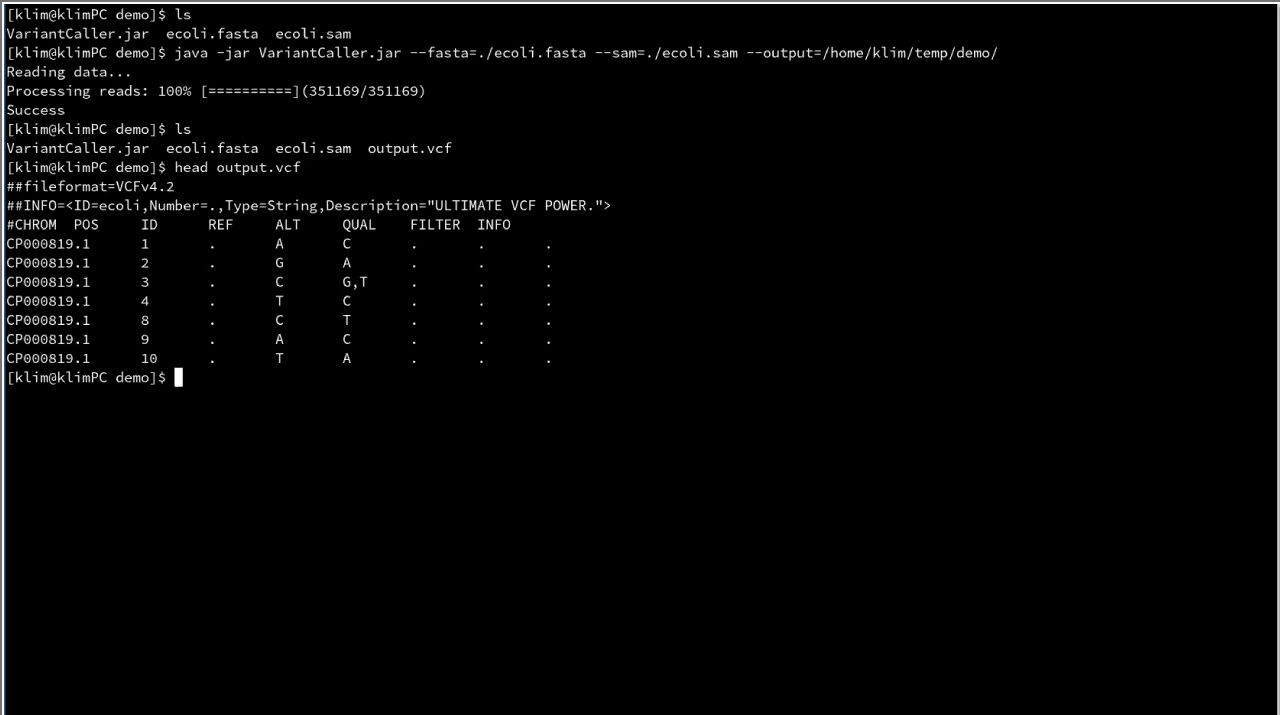
FASTA SAM

# 4.2 Взаимодействие классов

Наша программа принимает аргументы командной строки-пути к файлам, эти аргументы обрабатываются в классе команд line parser. После чего передаются в класс ParsedArguments, где они валидируются и хранятся. Далее мы обрабатываем полученные файлы в соответствующих классах-обработчиках. Например, FASTA файл в классе FastaHandler. Затем, обработанные данные передаются классу Caller, который выполняет поиск и возвращает список вариантов. Которые выводятся в VCF файл.



# 4.3 Работа программы

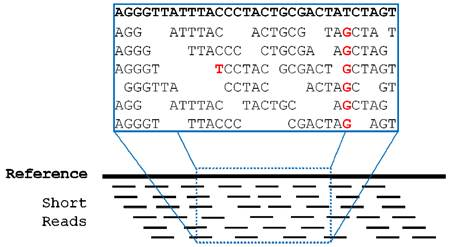
 Здесь программа запускается, обрабатывает файл и результат записывает в VCF файл. Который можно посмотреть в существующем геномном браузере.



Здесь можно видеть найденные варианты. Мутация скорее всего происходит там, где таких вариантов много. В нашем VCF файле отмечаются такие участки.

# 4.4 Упрощенная схема алгоритма

Мы итерируем по каждому риду, получаем его позицию на референсе. Если один из нуклеотидов в риде не совпадает с нуклеотидом на референсе, то мы записываем его как потенциальный вариант. Затем мы обрабатываем список потенциальных вариантов и выбираем те, позиции, в которых имеется хотя бы два варианта.



# 4.5 Тестирование

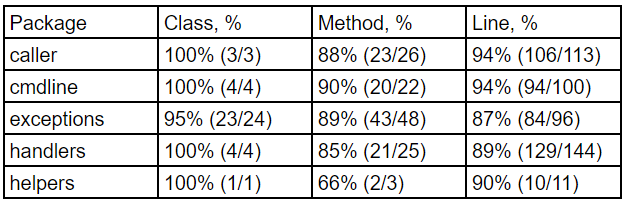
Методы тестирования проекта:

* Integration tests
* Unit-tests

Статический анализ:

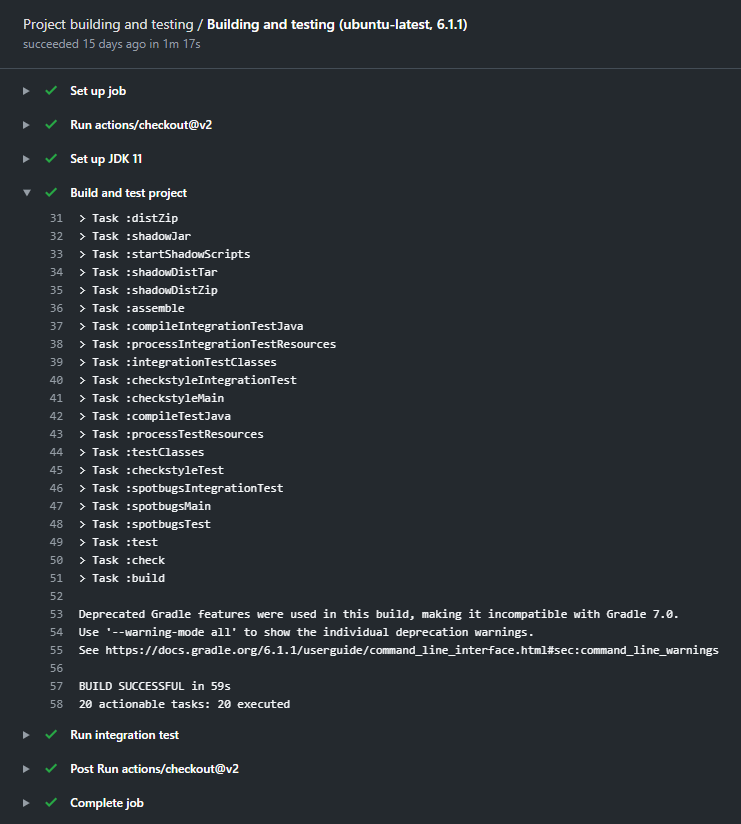
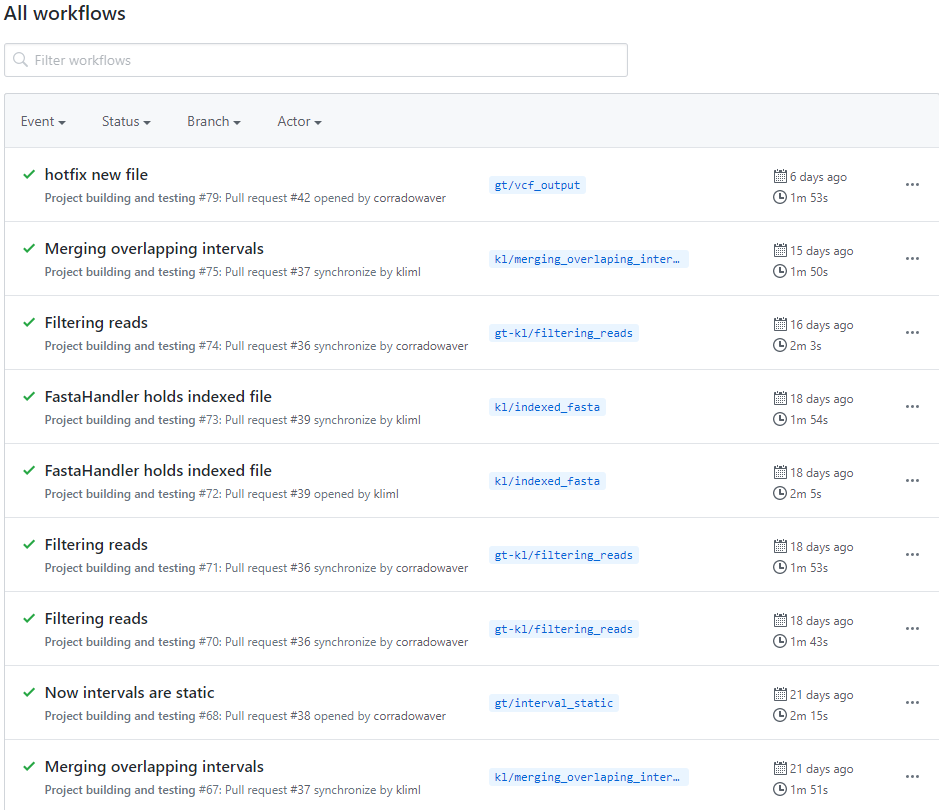
* Checkstyle
* SpotBugs

Code coverage:



# 4.6 Сборка на GitHub

На GitHub’е у нас есть CI, который мы использовали для выполнения частых автоматизированных сборок и для выявления потенциально возможных багов.



# 4.7 Используемые технологии

Набор технологий:

* Java 11
* Gradle 6.1.1
* JUnit 5

Сторонние зависимости:

* jopt-simple 6.0
* htsjdk
* shadowJar
* checkstyle
* spotbugs

# 5. Развитие проекта

* Усовершенствование алгоритма путем параллелизации вычислений
* Более сложные алгоритмы поиска и фильтрации вариантов
* Метрики вариантов

# 6. Итог

По итогу проекта мы наладили работу в команде, а также познакомились с современными методами разработки и получили ценный опыт в области биоинформатики.