Rust@Wellcome

Group Project
VCF

Why?

Opportunities for practice on a real group project

Exciting use-cases (such as client-side validation)

New VCF spec since January

- Fill in the space as far as supporting recently added functionality

Achievable!

VCF Overview

https://samtools.github.io/hts-specs/VCFv4.4.pdf

1st Milestone

Goal: provide easily accessible methods for working with genetic variation data in the form of VCF files.

Minimum features would be parsing a VCF and validating it

Possible target features are the following operations on VCF files

- Filter out specific variants
- Compare files
- Summarize variants
- Reordering and stripping columns
- Validate and merge files
- API?

Inspiration: https://vcftools.github.io/index.html

2nd Milestone

Web Assembly

- Learning objective majority of us are interested in
- A crate/feature that provides an interface to be used from WebAssembly
- This can then be integrated into any web app that wants to parse/validate VCF files on the user's browser (client-side).

Python bindings

- Create a crate that provides Python bindings for the VCF library
- Enable any Python program to simply import the module and get access to all functions and dispatch to the Rust-compiled executable
- give advantageous memory-safety and performance

Repo Structuring

Library crate: code for parsing VCF files

Binary crate: command-line interface dispatching to the library crate

Top-level cargo workspace

https://github.com/orgs/Rust-Wellcome/repositories

ADRs

How to collab

Github Projects/Issues and platform

- Gitter.im?

Adopting Rust conventions and best practices

- https://rust-lang.github.io/api-guidelines/about.html

Contribution policy

- Collaboration in getting past learning curves
- Code reviews
- Potential goal: is for commits to be from as many different people as possible
- contributing.md

1st task - We need a name