Project Breakdown – Otogenetics Variant Caller

We’re developing a variant calling pipeline for Otogenetics which is to be wrapped in a Graphical User Interface that runs on a web browser. For this RAD run, we want to develop the GUI wrapper using RADzen or Amper.

Conceptually, the deliverables can be broken down into three components:

**Component 1:** Variant calling pipeline

**Component 2:** Report generator

**Component 3:** Graphical user interface (GUI)

Component 1 and 2 are developed as a Python pipeline/script. Component 3 is to be developed using RAD.

Component 3, the GUI wrapper, will receive the input parameters for components 1 and 2. These parameters are expected in a JSON format. Component 1 takes this JSON format and generates an output VCF format (a bioinformatics tab-delimited text format). Component 2 generates a PDF report and requires input from the GUI about the header, database, filtering options and the footer. Optionally, the user may opt to only execute component 2 starting with a VCF input to this component.

For Component 3, the web-based GUI, we need:

1. A user interface (UI) that can run on a web-browser, with a good user experience (UX)
2. The interface should be able to read in:
   1. A reference archive that contains
      1. Reference Sequence (FASTA format)
      2. BWA Index for the provided reference
   2. Sequencing reads
   3. Output format specification (VCF or GVCF)
3. Able to parallelize the runs. *I.e.*, run a batch of samples concurrently producing one report per sample.
4. A backend database that keeps track of the input parameters of previously submitted jobs.

Component 3 can be broken down into the following tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Features** | **Timeline** | **Roadblocks** |
| **Early UI Planning** | 1. Identify all possible states of web interface | 1 Week | 1. Understanding client requirements |
| **Planning Data Storage and Database Model** | 1. Describe data storage plan and database model to be used | 1 Week | 1. Listing and understanding all use-cases |
| **Designing Web Application** | 1. User-friendly  2. Provides functionality to upload files  3. Provides functionality to specify run-parameters for Component 1 (bioinformatics pipeline) | 2 Weeks | 1. Choosing framework which enables:   1. User-friendly design 2. File-handling capabilities 3. Ease of binding to existing scripts |
| **Designing Logging Database** | 1. Integrates into the web-interface | 1 Week | 1. Deciding possible backup options |
| **Linking Web Interface, Database, Report Generator, and Bioinformatics Process** | 1. Ensures data integrity when combining components | 2 Weeks | 1. Setting up a web service stack  2. Describing interaction schema |
| **Testing** | 1. Unit testing  2. End-to-end testing | 1 Week | 1. Identifying boundary conditions for all use-cases |
| **Usability Testing** | 1. Ensuring that the web-interface is intuitive and easy to use | 1 Week | 1. User studies (?) |