Visualize OncoKB annotation and patient report generation

Google Summer of Code 2024

Proposal for cBioPortal for Cancer Genomics

by

Aishika Nandi

Table of Contents

About Me

Name:	Aishika Nandi
Timezone:	Asia/Kolkata(UTC+05:30)
Email:	aishikan@iitbhilai.ac.in
Course:	Bachelor of Technology in Computer Science and Engineering
University:	Indian Institute of Technology, Bhilai, India
Country:	India
GSoC Full Time:	Yes
Links:	Github, Linkedin
Resume:	Resume

Motivation for choosing cBioPortal

The mission and values of your organization truly resonate with my career goals and personal beliefs. As someone who is deeply passionate about using technology to create a positive social impact, I find the community's commitment to advance cancer research and improve patient outcomes in precision medicine very inspiring.

Moreover, cBioPortal community's emphasis on open-source development and collaboration reflects my own values of knowledge sharing and community-driven innovation. By actively contributing to cBioPortal, I aim to harness my technical expertise not only to make meaningful contributions during Google Summer of Code but also to actively engage in the organization's positive endeavors.

Proposed Project

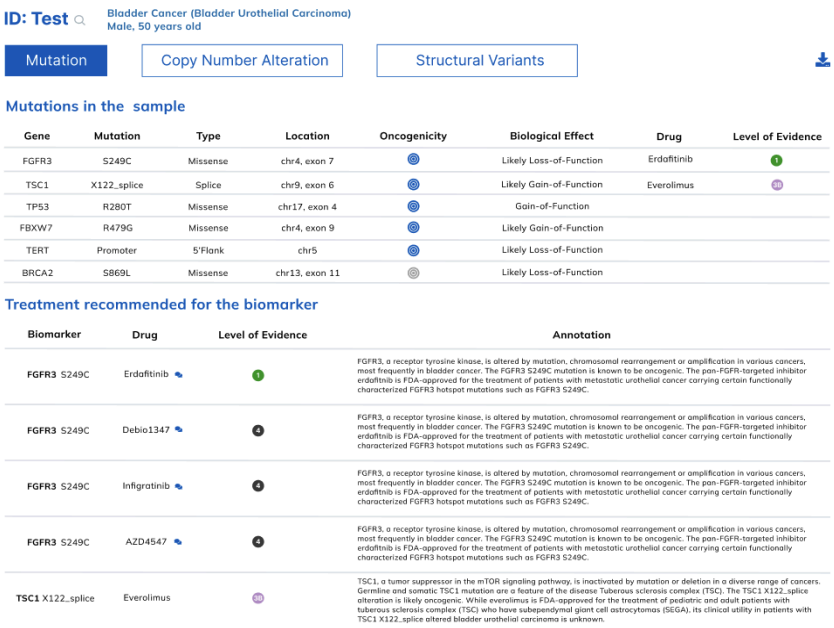
Title: Visualize OncoKB annotation and patient report generation

Abstract:

Precision oncology has revolutionized cancer treatment by tailoring therapies to individual patients based on the molecular characteristics of their tumors. OncoKB™ is a crucial resource providing comprehensive information about genomic alterations in cancer, including both biological and clinical insights. However, despite the availability of APIs and annotation tools, some users, especially those without computational backgrounds, face challenges in effectively annotating variants and generating reports.

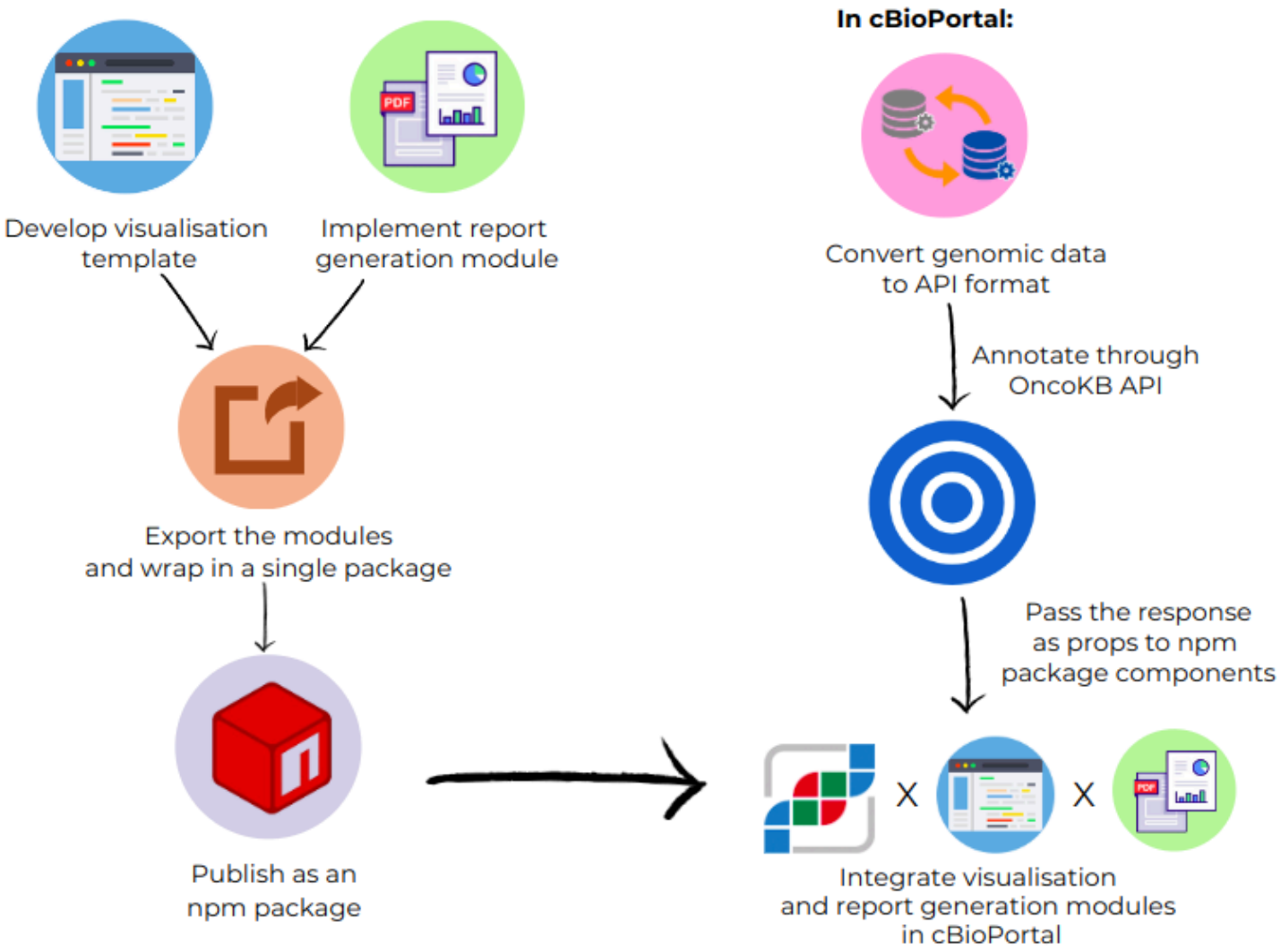
To bridge this gap, my project seeks to:

- **Develop a standalone npm package** for:
  - ➢ Simplifying the visualization of OncoKB annotations through the development of an intuitive interface.
  - ➢ Report generation module that can be downloaded by the users.
- **Seamless integration with cBioPortal** will empower clinicians and researchers ultimately facilitating informed decisions and improving patient outcomes in the fight against cancer. The tentative mockup:



Proposed Features and Implementation Strategy:

The following diagram depicts the high level implementation strategy of my project.



Feature 1: Develop an intuitive visualization template (3 weeks)
While implementing the react component, the following will be considered:
Props Structure: Props for customization, such as annotation content, styling options, column visibility, and pagination settings will be defined.

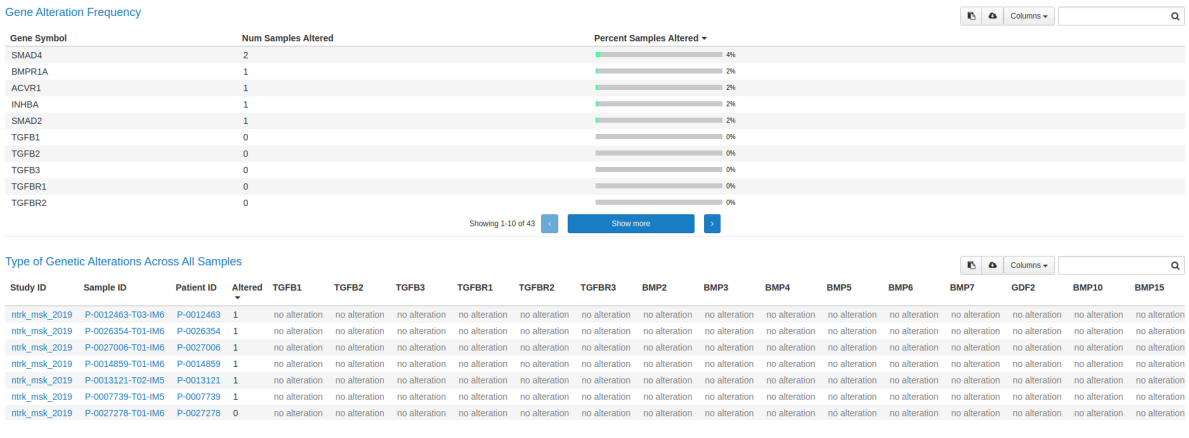
Configurability: The components shall be made configurable to accommodate different use cases and preferences. Default configurations will be provided for ease of use while allowing users to override them as needed.

Type safety: Since typescript shall be used, the expected data formats or values shall be pre-defined to avoid invalid inputs. Edge cases like null values, empty lists in data fetched from API response shall be gracefully handled.

Responsiveness: The components shall be made responsive to varying screen sizes.

Error Handling: Error handling components to handle unexpected scenarios shall be developed. Provision of informative error messages or fallbacks to guide users in case of data loading or rendering issues.

Moreover while exploring cBioPortal codebase, I came across visualizations similar to the mockup under the **Download** tab.



This is implemented using the `GeneAlterationTableComponent` component from [cbioportal-frontend](). A part of this component looks like this:

```
<GeneAlterationTableComponent
    data={this.props.geneAlterationData}
    columns={[
        {
            name: 'Gene Symbol',
            render: (data: IGeneAlteration) => (
                <span>{data.gene}</span>
            ),
            download: (data: IGeneAlteration) => data.gene,
            sortBy: (data: IGeneAlteration) => data.gene,
            filter: (
                data: IGeneAlteration,
                filterString: string,
                filterStringUpper: string
            ) => {
                return (
                    data.gene
                        .toUpperCase()
                        .indexOf(filterStringUpper) > -1
                );
            },
        },
```

I shall take reference from the existing components to learn more about the required configurations and the level of customization to be supported.

Feature 2: Develop the report generation module (2 weeks)

The npm packages that support pdf generation are [React-pdf](), [jsPDF](), [html2canvas]() and [ironpdf](). The one that suits our application the most is [ironpdf]() as it helps to generate pdf directly from the react components passed as an argument, thereby retaining the theme in the pdf as well. Here `htmlContent` would be replaced with the React component that wraps the entire visualization.

```
1. import { PdfDocument } from '@ironsoftware/ironpdf';
2. const HTMLToPDFComponent = () => {
3.  const convertHTMLToPDF = async (htmlContent) => {
4.      try {
5.          const pdf = await PdfDocument.fromHtml(htmlContent);
6.          await pdf.saveAs('generated_pdf.pdf');
7.          alert('PDF generated successfully!');
8.      } catch (error) {
9.          console.error('Error generating PDF:', error);
10.     }};
```

Implementing pdf generation module using ironpdf

Feature 3: Publish as a standalone npm package (1 week)

After exploring numerous documentations, the steps involved in publishing a react component as an npm package can be summarized as follows. I shall:

- Make a dedicated directory for the potential npm components, in the repository.
- Initialise npm environment using `npm init` in this directory. This creates the metadata for the package.
- Install dependencies using `npm install <package_name>` Delete the created node_modules directory. Change the dependencies field in the package.json to peerDependencies.
- Create an account on the npm website and login into the account using `npm login`.
- Publish the npm package using `npm publish`.

After publishing the npm package:

- The components shall be tested to verify correctness and reliability of the package.
- Detailed documentation for installation, usage and configurable options will be written.
- Semantic versioning shall be used to publish any newer versions.

Feature 4: Integrate into the cBioPortal (2 weeks)

Steps involved while integrating the above modules in cBioPortal:

- Install and import the components from the above package.
- Now, we need data for components' props which can be fetched from the OncoKB API. Through the API one can annotate the mutations by genomic change, protein change or HGVSg. Each of them has its dedicated APIs and query parameters. Hence for making use of the APIs, the query parameters need to be fetched from the patient's genomic data.

The query parameters are as follows:

Annotate Mutations by Protein Change

`GET`
`https://www.oncokb.org/api/v1/annotate/mutations/byProteinChange`

Query Parameters

Name	Type	Description
referenceGenome	string	Either GRCh37 or GRCh38. The default is GRCh37.
hugoSymbol	string	The gene symbol used in Human Genome Organization. Example: BRAF
entrezGeneId	integer	The entrez gene ID. (Higher priority than hugoSymbol). Example: 673
alteration	string	Protein Change. Example: V600E
consequence	string	Consequence. Example: missense_variant
proteinStart	integer	Protein Start. Example: 600
proteinEnd	integer	Protein End. Example: 600
tumorType	string	OncoTree(http://oncotree.mskcc.org) tumor type name. The field supports OncoTree Code, OncoTree Name and OncoTree Main type. Example: Melanoma

Annotate Mutations by HGVSg

`GET` `ttps://www.oncokb.org/api/v1/annotate/mutations/byHGVSg`

Query Parameters

Name	Type	Description
referenceGenome	string	Either GRCh37 or GRCh38. The default is GRCh37.
hgvsg	string	HGVS genomic format. Example: 7:g.140453136A>T
tumorType	string	OncoTree(http://oncotree.mskcc.org) tumor type name. The field supports OncoTree Code, OncoTree Name and OncoTree Main type. Example: Melanoma

Headers

Name	Type	Description
Authentication	string	OncoKB Token

Annotate Mutations by Genomic Change

`GET`

`https://www.oncokb.org/api/v1/annotate/mutations/byGenomicChange`

Query Parameters

Name	Type	Description
referenceGenome	string	Either GRCh37 or GRCh38. The default is GRCh37.
genomicaLocation	string	Genomic location. Example: 7,140453136,140453136,A,T
tumorType	string	OncoTree(http://oncotree.mskcc.org) tumor type name. The field supports OncoTree Code, OncoTree Name and OncoTree Main type. Example: Melanoma

The genomic data fields must be mapped to the above query parameters.

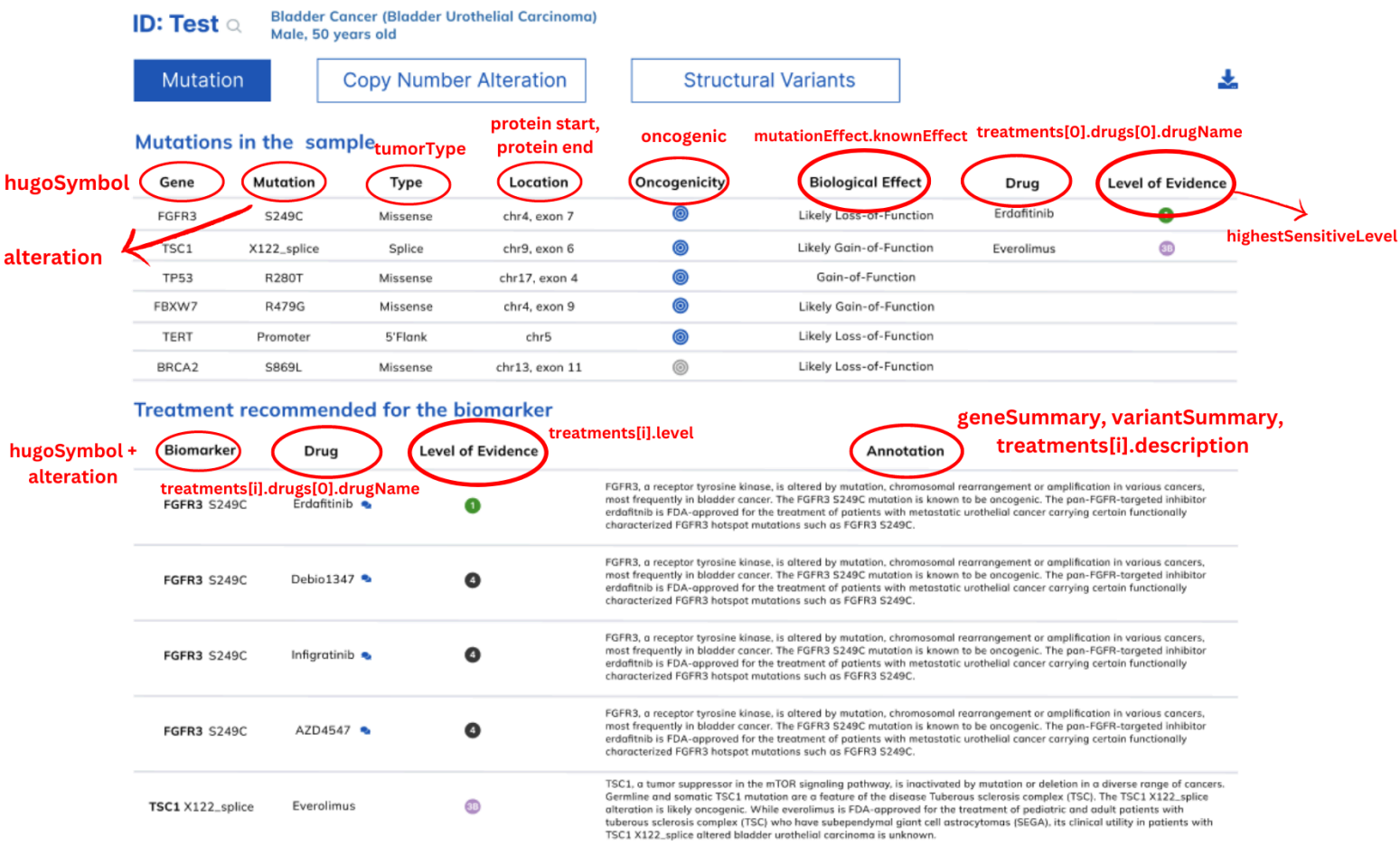
I will need to create an account at https://www.oncokb.org/ and generate my bearer token. This will allow me to annotate the genomic data using OncoKB API.

After annotating the data, an exemplar API response is as follows:


```
{
  "query": {
    "id": null,
    "type": "regular",
    "hugoSymbol": "BRAF",
    "entrezGeneId": 673,
    "alteration": "V600E",
    "alterationType": null,
    "svType": null,
    "tumorType": "Melanoma",
    "consequence": "missense_variant",
    "proteinStart": 600,
    "proteinEnd": 600,
    "hgvs": null
  },
  "geneExist": true,
  "variantExist": true,
  "alleleExist": true,
  "oncogenic": "Predicted Oncogenic",
  "mutationEffect": {
    "knownEffect": "Unknown",
    "description": "",
    "citations": {
      "pmids": [],
      "abstracts": []
    }
  },
  "highestSensitiveLevel": null,
  "highestResistanceLevel": null,
  "highestDiagnosticImplicationLevel": null,
  "highestPrognosticImplicationLevel": null,
  "otherSignificantSensitiveLevels": [],
  "otherSignificantResistanceLevels": [],
  "hotspot": true,
  "geneSummary": "BRAF, an intracellular kinase, is frequently mutated in melanoma, tl
  "variantSummary": "The BRAF V600E mutation is known to be oncogenic.",
  "tumorTypeSummary": "",
  "prognosticSummary": "",
  "diagnosticSummary": "",
  "diagnosticImplications": null,
  "prognosticImplications": null,
  "treatments": [],
  "dataVersion": "v2.1",
  "lastUpdate": "06/04/2018",
  "vus": false
}
```

The data from the above response will be passed as props into my npm package.

The mapping between the cBioPortal annotation headers (in black) and OncoKB API response fields (in red) is as follows:



The above mapping will help using the data as props thereby helping visualize the annotations and generate reports for the patients.

Feature 4: Testing (2.5 weeks)

I shall be testing the code changes using the following:

- Unit Tests: I will write unit tests to verify the functionality of individual components, functions, or modules in isolation using testing libraries like Jest and React Testing Library.

- End-to-End (E2E) Tests: End-to-end tests will help simulate real user interactions and test the entire application from the user's perspective. I shall use Cypress or the recommended technology stack for writing E2E tests.

- Integration Tests: I shall validate the interaction between different components, services, and modules within the portal to ensure seamless integration and functionality across the entire system. Integration testing can be performed using Jest.

Feature 5: Deploy, document & miscellaneous tasks (1.5 weeks)

I will follow the appropriate deployment strategies used by your community for deploying the portals. I will ensure comprehensive documentation in the project's README.md file to bridge any knowledge gaps and provide clear instructions for developers, contributors, and users. It will contain essential information about the project like setup instructions, usage guidelines, contribution guidelines, and any other relevant details. This period of time will also be used as a buffer time for suggested changes, additional features or bug fixes for the project. Making the final evaluation report for this program will mark the end of my project for **GSoC, not in general.**

Potential challenges

- Analyzing possible use cases of the NPM Package:
  This is crucial for understanding the range of configurations and props needed to cater to diverse user requirements. Finding an exhaustive list of potential use cases would be challenging.

- Handling edge cases:
  The component should gracefully handle situations such as null values, undefined formats, and empty lists, ensuring appropriate visualization. For large datasets, implementing pagination functionality is required to optimize performance and user experience. Understanding the possible data formats and scenarios encountered when interacting with external APIs needs considerate time.

- Writing error messages for the npm package:
  Providing informative error messages or error codes to assist users in troubleshooting issues is complex in itself as it requires detection of errors beforehand.

All the above challenges can be dealt with by devoting sufficient time for exploration. I shall utilize the bonding period (May 1- May 27) for planning the specifications of the package beforehand.

Timeline

May 1, 2024 - May 26, 2024 (Bonding Period)

Outcome: Finalizing target features and strategy.

- Get familiar with the existing codebase. Go through the documentation of different tech stacks used.
- Study the similar visualization components from cBioPortal or other sources, their structure, configurations, flexibility provided.
- Planning the npm package configurations and error handling.
- Exchange views with the team and re-consider the features and implementation strategies.

May 27, 2024 - June 16, 2024 (3 weeks)

Outcome: Template for visualizing annotations.

- Develop these modular components, keeping the mockup in mind.

June 17, 2024 - June 30, 2024 (2 weeks)

Outcome: Report generation module.

- Use ironpdf to generate patient reports.
- Retain the theme of the components used for visualization in the reports.

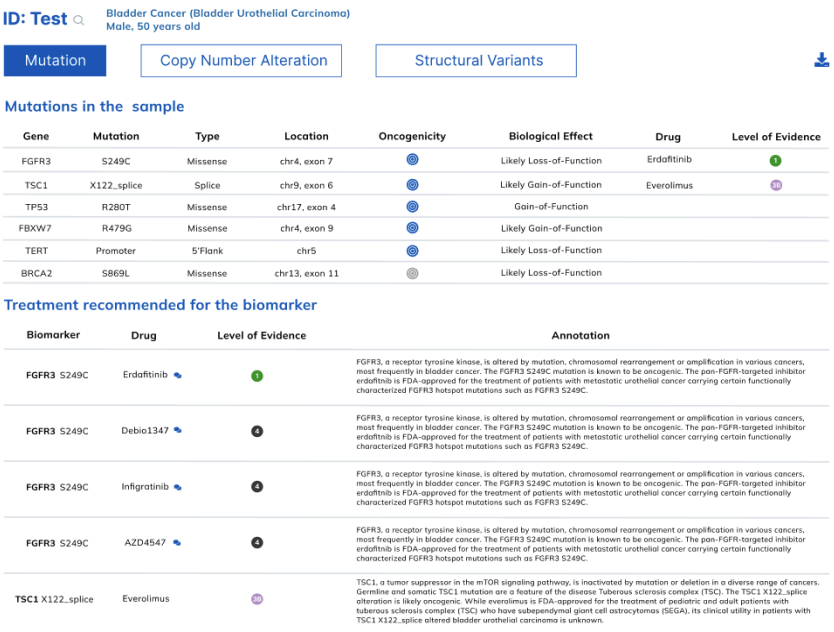
July 01, 2024 - July 7, 2024 (1 week)

Outcome: Publish as a standalone npm package.

- Wrap the above modules and publish as a single npm package.
- Document the package along with installation guidelines, implementation examples and configurations.
- Prepare the mid evaluation report.

July 8, 2024 - July 21, 2024 (2 weeks)

Outcome: Integrate the above in cBioPortal



- Take feedback from the mentors and reform the proposed timeline according to the mid evaluation.

- Set up routes and buttons for navigating to this section.
- Import our npm package and deploy the visuals and report generation module using props.

July 22, 2024 - August 7, 2024 (2.5 weeks)
Outcome: Test the platform. Enhance efficiency and coverage.

- Build/Reform the test components.
- Implement unit, E2E and integration tests. Improve the coverage and efficiency in this time period.

August 8, 2024 - August 18, 2024 (1.5 week)
Outcome: Document the project changes.

- Buffer time for suggested changes, additional features or bug fixes for the project.
- Deploy the portal using the standard technologies.
- Document the project details like setup, requirements, etc. in ReadME.md other documents for clear understanding of the project.
- Write a blog on my journey and experiences throughout the program.
- Prepare the final evaluation report for the program.

August 19, 2024 - August 26, 2024 (Submission period)
Outcome: Final submission of the project.

- Submission of final work product and final evaluation report.

Open Source Contributions and Projects

- I have fixed bugs in iitbh-cgpa browser extension which is used by most of the students in our institute for calculating grade points and the credits of a student. These bug fixes make the extension accessible for the postgraduate students of IIT Bhilai as well. Merged Pull request: Link
- I have given a modern interface to another browser extension called Watch Party, which helps people to stream shows online together. Merged Pull request: Link

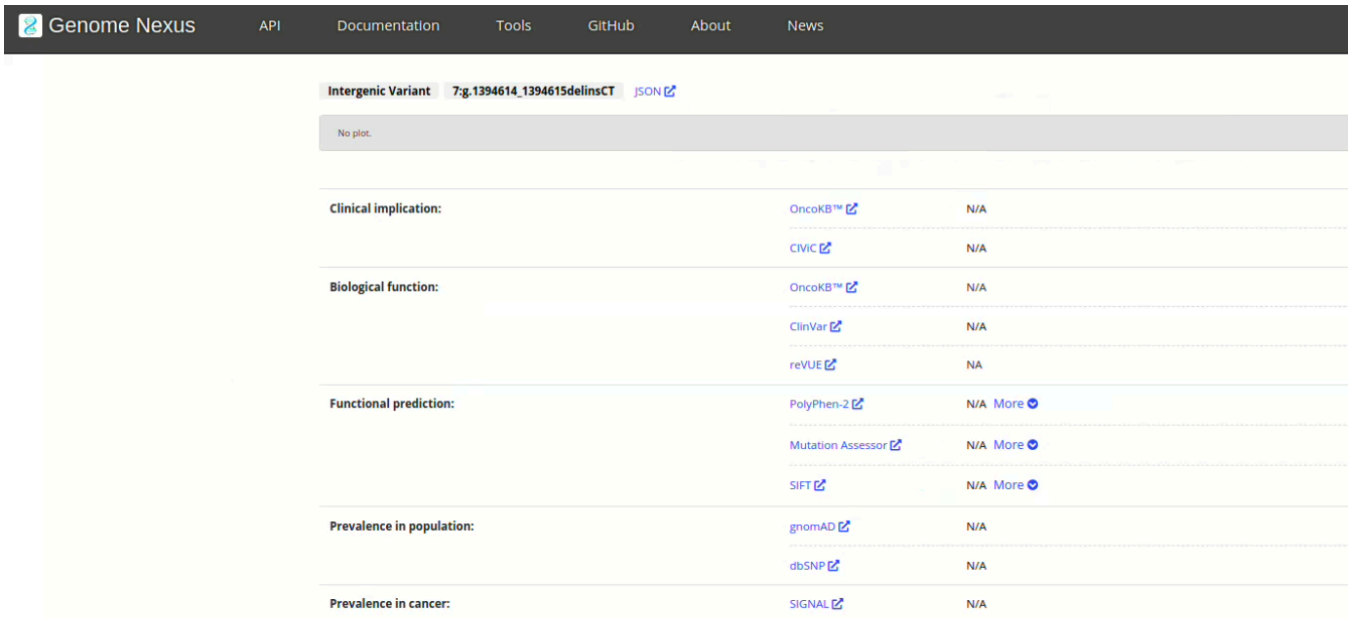
- I had started with open source contributions at FOSS Overflow 2022 which is the official Winter of Code program at IIT Bhilai. I have mentored 2 projects in the same program in 2023, so I understand the **importance of delivering quality code** and **maintaining professionalism** in the community.

My contributions for cBioPortal community:

Issue	Pull Request
#674 The search bar does not consider variants starting with “chr” prefix	https://github.com/genome-nexus/genome-nexus-frontend/pull/155
#734, #721 Handle intergenic variants	https://github.com/genome-nexus/genome-nexus-frontend/pull/157

Output for #674
Demo link: Link

Output for #734
Demo link : Link



What are my other commitments?

I have no other commitments during the summer other than this program. Moreover I will be on a summer break, which means I have no academic commitments during the same period. I will be able to dedicate ~30 hours in a week and I am ready to get it extended as and when required. I shall be available over mails and online platforms.

What are my expectations from mentors?

I have been part of the community over slack, and I found them to be very active in discussing issues and resolving queries. After selection, I would like to discuss proposed features in detail with the mentors beforehand so that my efforts are always aligned with the team's perspective. I also hope they help me make decisions when there are multiple ways of carrying out the same job. Having said that, I will ensure that I maintain professionalism and remain in the said boundaries.

Who am I?

I am Aishika Nandi, a pre-final year Computer Science undergraduate at Indian Institute of Technology, Bhilai. I am a tech enthusiast and I love to learn new technologies for the betterment of existing automation. I was selected for the TCS Research and Innovation program last summer under the project "Synthesizing Transformation for Data Transformation". I had to carry out the feature extraction and analyze the code transformations in C++. I have also created a full stack blogging application for my client with edit and comment privileges. Speaking of hackathons, my team has secured 3rd prize at WebWave, a website development hackathon for the open source community of our institute. Moreover, my team was one of the finalists at Smart India Hackathon 2023, one of the most prestigious countrywide hackathons of India. In addition to this, I have represented my institute IIT Bhilai at Inter IIT Tech Meet 2023. Our team had to develop a university ecosystem platform prototype. **Regarding my skills, I am proficient in c++, javascript, python, React, Django, Flask, NodeJs, java, ShadCN UI, Tailwind, MongoDB, SQL, Docker, Git, OpenAPI and many more.** Currently, I am developing a campus delivery portal using

React, Typescript, flask, OpenAPI and MongoDB that connects the services and the users in any given campus. I strongly believe my expertise in software development and my motivation to work in projects aligns with the principles of cBioPortal's community. The fact that I have had an internship experience in the past, makes me aware of the industrial standards and expectations. Hence I will be able to deliver my best in this program as well.

Past Experience

- **Work experience:**
  TCS Research Intern, India (May 2023-July 2023)
  - ➢ *Under the TCS Research & Innovation program, the project "Synthesizing Transformation for Data Transformation" is based on a fuzzing tool "VeriFuzz" that generates failed test cases for a c, java programs .*
  - ➢ *Developed an equivalent to the existing feature extraction tool for improving its accuracy.*
  - ➢ *Had to replace the dependency on PRISM framework with Clang-LLVM frontend using Clang Abstract Syntax Trees and C++ data structures.*
  - ➢ *Certificate*

  Thinkin, Web Developer (Aug 2023-Dec 2023)
  - ➢ *A real-time static blogging website with edit privileges for a content writer and comments section for the audience.*
  - ➢ *Implemented using **CSS, React.js, MongoDB**. Images were hosted using Cloudinary and deployed on Netlify.*
  - ➢ *Website, Github*

- **Web development Experience:**
  CampusHub (Jan 2024-Present)
  - ➢ *A portal to connect the students with the services, stores and products within the campus.*
  - ➢ *One can step into entrepreneurial ventures as well as*

re-sell second hand items in the campus community.

- ➢ *Built using **TypeScript, React, ShadCN, Tailwind, Connexion, Flask, OpenAPI, Pydantic, PyTest**.*
- ➢ *Note: This is a System Design course project which will end by April 2024.*
- ➢ *Website, Github*

Openlake Website ***(July 2023)***

- ➢ *A full-stack responsive website for the Open-Source club of IIT Bhilai.*
- ➢ *It showcases the projects, contributors, open source achievements, programs and events conducted by the club.*
- ➢ *Secured 3rd prize in WebWave hackathon by implementing technologies like **React.js, CSS and Strapi API.***
- ➢ *Website, Github*

Ally ***(Nov 2023-Dec 2023)***

- ➢ *Built using **React.js and Django**, the platform helps clients to search talent for their projects and acts as a private social media for university peeps.*
- ➢ *It includes AI-based features like team generation for hackathons, course and project recommendations to users.*
- ➢ *This was a university ecosystem project for a company named Trumio in the Inter IIT Tech Meet 2023.*
- ➢ *Presentation, Certificate (Github repository is Private)*

NetCompress ***(Dec 2023)***

- ➢ *Our team was one of the **finalists in Smart India Hackathon** 2023 [SIH1412].*
- ➢ *Revolved around data compression for data transmission over networks.*

➢ *It had dynamic implementation of compression algorithms for a multitude of data types like images, zips, audio, video, stream of data, etc. using **Python, React and FastAPI.***
➢ *Github, Certificate*

Hostel.IO ***(October 2022)***

➢ *A Web App built on Google Apps Script, that automates the process of applying for campus in and out using Google forms and Google Sheets Database.*
➢ *An auto-generated mailing system is implemented to notify the applicants of the availability of tokens for campus in and out.*
➢ *Website, Github*

- *Knowledge in bioinformatics:*

➢ With regard to bioinformatics, I have studied the course IC251 Basics of BioInformatics as part of my course curriculum. As the name suggests, I have the basic understanding of the concepts used in this field.

IC251 Basics of Bioinformatics (4 Credits)
Introduction to biomolecules (amino acids, proteins, DNA, RNA, Genes) and different tools to visualize and represent biomolecules on computer (Visual molecular dynamics (VMD) software)
General introduction to bioinformatics; Definition, Scope and applications, brief history of sequence analysis - Protein, DNA and RNA sequences; introduction to different bioinformatics related databases (EMBL, DDBJ, GenBank, PIR, PDB etc.) and their uses.
Sequence analysis - Comparing two sequences, Similarity searches on sequence databases, building multiple sequence alignment, local and global alignment, BLAST, FASTA.
Working with 30 protein structures - introduction to protein data bank (PDB) file, predicting the secondary structure of a protein sequence, primary structure to 30 structure of protein, finding proteins with similar shapes, folding a protein in a computer, predicting interactions; working with RNA.

Apart from cBioPortal, Genome Nexus and OncoKB, I have some practical knowledge of using portals like uniprot, BLAST, etc in the past.