GSoC 2025: cBioPortal

AI/LLM Generated gene alteration and expression based subtyping for each tumor type #114

Proposal: Simulated Expertise & Confidence-Optimized LLM Pipeline for Precision Oncology

- **Organization**: cBioPortal
- **Mentors** : Ino de Bruijn, Chris Fong, Angelica
- **Applicant**: Ayush Shaurya Jha
- **Task name**: AI/LLM Generated gene alteration and expression-based subtyping for each tumor type #114
- **Project length**: 350 hours

PERSONAL INFORMATION

CONTACT INFORMATION

- **Name**: Ayush Shaurya Jha
- **Primary Email** : shauryasphinx@gmail.com
- **Secondary Email**: ayush.2022ug3013@iiitranchi.ac.in
- **Github Profile**: https://github.com/jhaayush2004

- **LinkedIn**: Click to visit
- **My Resume**: Resume
- **Location** : Patna, Bihar, India
- **Timezone** : IST (UTC +5:30)

Relevant Experiences

I have extensive experience in integrating AI and harnessing LLMs prowess to increase accuracy of research works and daily tasks in biological and healthcare domain which has given me significant **domain knowledge of bioinformatics workflow, Oncological biology and medicinal chemistry**. And this makes me suitable for this project as I think along with technical knowledge one needs to have domain knowledge also which I have got from numerous works in the field of integrating AI in healthcare. And below mentioned **Personal project, experience and research works are the testaments** to my domain knowledge of bioinformatics and oncological biology and **technical knowledge** of AI, LLMs, Deep Learning, Machine Learning, and Optimizations:

- **NeuroVisionAI:TumorMapperNet** is the project having two parts. The first parts deals with classifying and segmenting brain tumor out of MRI scans of brain with the help of self-designed deep learning architecture. The second part classifies the type of tumor and suggest that which molecule and compounds should be most beneficial for the treatment and thus helps finding out the medicines by embedding similarity calculation with the help of LLM powered RAG pipeline. First part is public but second part is part of ongoing research. **This project gave me a lot of exposure about field of Oncology, bioinformatics and medicinal chemistry.**

• **Generative AI intern (LLM Engineer intern)**, Tapfinity Technologies Pvt. Ltd.

- It's **an AI healthcare startup** dealing in **bioinformatics** as well as personalized AI health assistance and working here gave me extensive experience of bioinformatics and medicinal chemistry and also embeddings generation, visualization, and management.
- I also worked on a Graph RAG which consisted of scraped data of around 1200+ diseases and has around 9+ millions entity nodes and 14+ million entity relationships and deployed it on a neo4j instance, this solution is used to perform diagnosis of users.
- I designed a personalized **AI health coach** which involved exploring, analyzing and molding a whole bunch of data with about 1M+ instances out of which most patients were under supervision of **oncologists**.

- I worked on designing LLM powered RAG pipeline which involved embedding generations, its storage into database and then its retrieval based on similarity check.

• **Applied AI and Research intern, ISRO** (Indian Space Research Organization, NARL)

- I worked on designing a robust and well optimized deep-learning based model from scratch which is getting trained on marsian ionoshpheric data composed of frequency, time-delay and altitude numeric values.
- The pipeline I designed till now is capable of precisely recognizing and segmenting out ionospheric echo (time-delay and frequency) out of numeric data and get the real time plot with accuracy of about 98.2 % as of now and continuous optimizations for better convergence and training is going on.
- This automation has reduced the analysis time of the marsian ionosphere by about 35%. Apart from this, I'm also finetuning a LLM on the marsian data available till now along with doing changes in its architecture and quantization to make it memory efficient keeping its precision maintained.
- I also implemented reranking and filteration techniques which reduced LLM hallucinations by about 40%.

• **AI ML Lead** at House of Geeks, IIIT Ranchi

- I am appointed as Lead of AI ML club at House of Geeks which is technical society of my college where I work as project maintainer and guide passionate developers how to strive in domain of AI and Machine Learning, guiding them in their project and Open-Source contributions.
- Integrated the dynamic LLM and RAG powered bot in the official website of our college, fostering student collaboration and providing valuable resources for academic, technology and extracurricular activities.

My AI Research Publication:

- Deepfake Detection with Multi-Fusion and Residual Learning in Deep CNNs

  The paper, with me as the first author, has been accepted at 5th International CVR conference 2025, NIT Goa (25-26 April,2025). It presents a novel Multi-Fusion Residual Deep CNN (MFuRe-DCNN) approach that enhances deepfake detection by optimizing feature retention efficiency and generalization.

STUDENT AFFILIATION

- **UNIVERSITY**: Indian Institute of Information Technology Ranchi
- **DEGREE** : Bachelors of Technology (B.Tech.)

- **MAJOR** : Computer Science and Engineering(with specialization in Artificial Intelligence and Data Science)
- **EXPECTED GRADUATION** : 2026

BRIEF BIO

I am Ayush Shaurya Jha, a third-year undergraduate student at the Indian Institute of Information Technology Ranchi (India), with extensive experience in AI, LLMs, Machine Learning, Data Science, RAG and Deep Learning through various projects, researchs, and internships. My interests lie in exploring innovative AI-driven techniques to analyze and interpret complex data, particularly in bioinformatics and computational biology, where machine learning and AI can uncover hidden patterns and enhance scientific discovery. My primary interests lie in applying AI-driven techniques to biomedical research, particularly in cancer genomics, tumor detections and bioinformatics.

My **skill set** includes **AI, Machine Learning, Large Language Models (LLMs), NLP, Algorithm design**, **Retrieval Augmented Generation (RAG), Optimization, Deep Learning, Python, C, C++,** and **JavaScript**.

Due to my involvement in open-source projects, research works and personal projects, I have developed a deep understanding of need for well-documented, maintainable, and efficient code. The gene recommendation proposal with its focus on implementing advanced multi perspective queries, recommended gene filtration, confidence scoring and Multi Model Ensemble (along with pre-trained model, we can also include other biomedical LLMs) to cross check recommended gene list along with Expert-in -the-Loop validation and innovative prompt engineering techniques make it align strongly to my skill set, expertise and passion. I am very excited about leveraging the pretrained LLMs on the literature and improving its precision of response generation to enhance the quality of gene recommendation for cancer subtypes.

PROJECT DESCRIPTION

ABSTRACT

OncoTree provide an in-depth cancer classification and it's a dynamic platform while cBioPortal is an Open-Sourced repository hosting gene alterations, gene alteration and clinical data for cancer genomic data analysis and visualization. But the major challenge is to automate the process of default gene for cancer subtypes from OncoTree in order to make study and research easier. Currently, when researchers explore cancer datasets (Eg: Breast cancer), they need to do it manually which is time consuming and sometimes may be tedious. And so, the proposal aims to solve the issue by incorporating many advanced and innovative ideas.

The project will consist of the following key components:

- **Prompt Designing** – LLM will be compared on various prompting methods like dynamic context injection, Role Specific Prompting, and many more and finally a refined and hybrid prompt will be figured out.

- **Hallucination Mitigation–** We will introduce reasoning, Chain-of-thoughts method and source figuring which will prevent LLM from generating irrelevant response and will also enhance transparency of the system.
- **Confidence Scoring** – This will help us to get relevant results above a particular threshold score which we will set according to the need.
- **Multi-model Ensemble**– Along with the pretrained model, we will consider other biomedical based LLMs particularly trained on Oncological literature like BioBERT, MedBERT and so on and will ensemble the response from all these LLMs and will give most frequently suggested genes as output.
- **Expert-in-Loop-Validation**– All our response will be manually validated by expert ensuring efficiency and accuracy of the system. It will act as a great guide for us to make further improvements in the process to get more relevant response.

This project will enhance cancer classification workflows by providing automatically curated, biologically meaningful gene recommendations, ultimately benefiting oncologists, researchers, and clinical decision support systems.

BACKGROUND

Cancer classification is a crucial aspect of oncology, influencing prognosis, treatment, and research. Platforms like OncoTree and cBioPortal provide large-scale cancer genomic data, helping researchers analyze genetic mutations associated with specific cancer types. However, a major challenge is dynamically identifying the most relevant genes for a given cancer subtype.

Currently, cBioPortal assigns cancer types using OncoTree codes but lacks an automated system to suggest the most relevant genes for classification. This project proposes using LLMs to generate gene lists based on cancer subtypes, ensuring high reliability by incorporating various robust prompting techniques along with filtration of irrelevant responses and advanced hallucination mitigation.

Issue with the current scenario

Whenever a researcher or scientist or general audience tries to explore a dataset, they need to manually explore all the cancer subtypes and then has to determine by themselves that which genes are to be analyzed. This puts a limit on the exploration they could do which ultimately causes hinderance in the research and discoveries.

This proposal thus, proposes a dynamic LLM-driven gene discovery and expert validation that optimizes accuracy, interpretability, and reliability in cancer subtype classification. Instead of relying on complex retrieval systems, our approach leverages pre-trained biomedical LLMs to generate, refine, and validate gene recommendations, ensuring alignment with established oncological knowledge.

Key Innovations:

- **Primary LLM-Driven Gene Discovery** – Directly querying state-of-the-art pretrained on oncological literature (by cBioPortal's team) LLMs to extract gene associations for each cancer subtype using optimized prompt engineering.
- **Iterative Prompt Optimization** – Enhancing LLM responses through structured prompts, context-aware phrasing, and few-shot learning to improve relevance and accuracy.
- **Confidence Scoring & Re-Ranking** – Introducing certainty metrics such as Gene Consensus Score, Pathway Similarity Score, and Mutation Prevalence Ranking to prioritize highly relevant genes.
- **Bias & Hallucination Reduction** – Mitigating incorrect predictions by refining model output with biological plausibility constraints, expert feedback loops, and logical consistency checks.
- **Domain Expert Validation** – Instead of relying solely on automated retrieval, results are reviewed by oncologists and biomedical researchers, ensuring practical applicability.
- **Minimal Infrastructure Overhead** – Unlike traditional RAG pipelines, this approach eliminates the need for FAISS-based similarity search while still ensuring high precision and interpretability.

By optimizing LLM-based gene recommendation, refining prompt strategies, and implementing expert-driven validation, this project enhances the discovery, precision, and usability of gene-cancer associations, accelerating advancements in oncological research and personalized medicine.

PROJECT PROPOSAL

Overview

The proposal aims to develop a robust AI and LLM powered pipeline for generating precise and accurate gene alteration and expression-based subtyping for each tumor type present in OncoTree. The proposal thus leverages the cBioPortal's Pre-Trained LLM on the oncological available dataset and literature as well as other Biomedical based LLM for hallucination-free and accurate response generation. It incorporates innovative multi-perspective and iterative prompting along with dynamic contextual prompting facilitated by re-ranking and confidence score inclusion in the system, thus, guaranteeing the highly precise output. The whole idea has been divided into five phases which are descripted shortly below:

- **Primary LLM-Driven Gene Discovery**: This leverages the pre-trained and already fine tuned LLM alone and along with the other biomedical based oncological LLM such as MedBERT, BioBERT, and so on and compares the ensembled output (which figures out the most frequently suggested genes) with the output by our finetuned LLM alone and thus, guide us which way to take.

- **Iterative and Multi-Perspective Prompt Optimization**: This phase involves refining of response by iteratively refining prompts leveraging few-shot learning,

hierarchical reasoning, multi-perspective prompt injection and self-consistency methods.

- **Simulated Expert Reasoning with Confidence Scoring & Source Attribution**: This ensures the most relevant genes to be output first while making the model to output the confidence score along with simulating it to be oncological expert and explicitly forcing to mention the source or cite the reference topic of its reply.
- **Bias & Hallucination Reduction**: Task is carried out by taking into account Logical Consistency Checks and Cross Validation across LLM (in ensemble model).
- **Domain Expert Validation**: Instead of blindly trusting LLM prediction, we will introduce an expert-in-loop validation mechanism which will help in further improving the LLM's knowledge. This would be the last phase which will let us know that how precise our pipeline is.
- **Integration of Pipeline with cBioPortal**: This phase deals with integrating the pipeline built till now with the cBioPortal page using the FastAPI-powered API layer.

Technologies

The project will employ deep learning, graph-based techniques, and statistical methods, using the following tools and technologies:

- **Programming Languages**: Python (model development), JavaScript (web integration)
- **Generative AI Frameworks**: Langchain, Google-Generative AI, Groq, Huggingface
- **LLMs**: Gemini, LLaMA 3, GPT-4, BioGPT, PubMedBERT, BioBERT, GPT-4
- **Data analysis and manipulation**: Numpy, Pandas, SciPy, matplotlib, Tensorflow
- **API Development**: FastAPI
- **Front-end Integration**: Vanilla JS, HTML, CSS

PLAN OF ACTION

Phase 1: Primary LLM-Driven Gene Discovery

Objective

In this phase, we leverage the prowess of our pre-trained and fine-tuned Large Language Model (LLM) on the oncological literature and dataset. Along with it, we will also use other open-source biomedical based LLM such as MedBERT, BioBERT, PubMedBERT, and BioGPT

to compare that whether the ensemble of result from these models are more precise than that generated by finetuned LLM alone.

The core idea is to evaluate:

- How well does our finetuned LLM model performs alone?
- Does the performance of ensemble approach better than results from our single LLM?
- Is multi-model approach what should we consider next or just our own LLM is sufficient?

By comparing the response of our fine-tuned model, we get to know that:

- Whether our fine-tuned model's response is more accurate than the frequently generated genes by multiple other biomedical models.
- If ensemble prediction approach helps in reducing bias and hallucinations.

TASKS

- **Query our Fine-Tuned LLM**
    - Directly we will query our oncological data Fine-Tuned LLM for the given cancer subtype gene recommendation.
    - We will extract the top N recommendations.

```python
from collections import Counter
from langchain.llms import Groq
from langchain.prompts import PromptTemplate

# Initializing LLM
llm_finetuned = Groq(model="llama3-70b")  # Fine-tuned oncological LLM

# structured prompt(Let's assume to be in json format)
prompt_template = PromptTemplate(
    input_variables=["cancer_subtype"],
    template="""
    Given the cancer subtype "{cancer_subtype}", list the most relevant genes associated with it
    based on biomedical research. Format the output as JSON with "gene_name" and "relevance_score".
    """
)
# Designing query function
def query_llm(llm, cancer_subtype):
    query = prompt_template.format(cancer_subtype=cancer_subtype)
    response = llm(query)
    return response  # Assuming response is a JSON string

genes_finetuned = query_llm(llm_finetuned, cancer_subtype)

# Querying our finetuned LLM
cancer_subtype = "Glioblastoma Multiforme"
```

- **Query Other Biomedical LLMs(If possible, otherwise we will move on with only our fine-tuned LLM)**
    - We can query same prompt to BioBERT, MedBERT, PubMedBERT, and BioGPT.
    - Collection of top N recommended gene.

```
# Initializing some biomedical LLMs
llm_biobert = Groq(model="biobert")       # BioBERT
llm_medbert = Groq(model="medbert")       # MedBERT


# same structured prompt
prompt_template = PromptTemplate(
    input_variables=["cancer_subtype"],
    template="""
    Given the cancer subtype "{cancer_subtype}", list the most relevant genes associated with it
    based on biomedical research. Format the output as JSON with "gene_name" and "relevance_score".
    """
)

# Querying all LLMs
cancer_subtype = "Glioblastoma Multiforme"

def query_llm(llm, cancer_subtype):
    query = prompt_template.format(cancer_subtype=cancer_subtype)
    response = llm(query)
    return response  # Assuming response is a JSON string

genes_biobert = query_llm(llm_biobert, cancer_subtype)
genes_medbert = query_llm(llm_medbert, cancer_subtype)
```

- **Ensemble the Predictions**
    - We will aggregate gene predictions from all different models.
    - Computation of gene frequency which tells that how often a gene is recommended by different models.
    - Generation of ranked list of genes.

```python
# Converting responses into lists of gene names
def extract_gene_names(response_json):
    return [gene["gene_name"] for gene in response_json]

genes_all = (
    extract_gene_names(genes_finetuned) +
    extract_gene_names(genes_biobert) +
    extract_gene_names(genes_medbert)
)

# Computing gene frequency scores
gene_frequency = Counter(genes_all)

# Ranking genes by consensus score
ranked_genes = sorted(gene_frequency.items(), key=lambda x: x[1], reverse=True)

# Print ranked gene associations
print("Ensembled Gene List with Consensus Scores:")
for gene, score in ranked_genes:
    print(f"{gene}: {score}")
```

- **Comparison of Fine-Tuned LLM vs. Ensemble Approach**
    - Will compare the most frequently recommended genes by our finetuned LLMs with that generated from the ensemble approach.
    - We will go for expert validation to find out that whether our fine-tuned model alone performs better than ensemble approach.

Why This Works Better?

- Validates that whether our oncological fine tuned LLM is optimal or if considering multiple models gives us better results.
- Ensemble learning reduces the risk of missing important genes.
- Frequent gene suggestions across models indicates high confidence and increased reliability.

Next Steps

- If fine-tuned LLM alone performs better than the Ensemble approach, then we would go without ensemble in later phases.
- If not so, we will incorporate it into our pipeline (if in the scope of project).
- If some discrepancies arise, expert validation will be considered.

This phase thus, lays the foundation for next steps involving bias and hallucination mitigation, advanced prompting, ranking and confidence score, and so on.

Phase 2: Iterative and Multi-Perspective Prompt Optimization

Approach

The accuracy of the LLM generated response depends heavily on the quality and the structure of the prompt provided to the model. This phase thus, focusses on optimizing the prompts using iterative refinements as well as multi-perspective techniques to ensure the relevancy and correctness of the recommended output gene by the model.

Below are the four core techniques that we will employ to refine the prompt:

- **Few-Shot Learning**- This involves providing an example of gene-cancer association in the prompt itself as an example. It helps the models get the prompt easily.
- **Hierarchical Reasoning**- This involves breaking down the reasoning process into multiple biological steps which helps the model to stick to the literature in a structured way and prevents it from giving random responses.
- **Multi-Perspective Prompt Injection-** This involves refining the prompt to incorporate more than one perspective of answering the question by model in order to ensure a more comprehensive and biologically valid response.
- **Self-Consistency for Response validation**- LLMs may recommend different gene even if the same prompt is given, so we can check that whether our LLM is in such condition by prompting it same for multiple times.

Steps for Prompt Optimization

- **<u>Few-Shot Learning for Contextual Precision</u>**

  Why? The LLM provide more structured and relevant response when show with actual example of expected response.

  How? We will not ask the model directly to recommend gene but will ensure to show gene-cancer mapping as an example in the prompt itself.

  Example Prompt Using Few-Shot Learning:

```
few_shot_prompt = """
Here are known gene associations for different cancer subtypes:

{
    "Breast Cancer": {
        "Mutation-Based": ["BRCA1", "BRCA2", "TP53"],
        "Expression-Based": ["ERBB2", "GATA3"],
        "Pathways": ["PI3K-AKT", "p53"]
    },
    "Lung Cancer": {
        "Mutation-Based": ["EGFR", "KRAS", "ALK"],
        "Expression-Based": ["MET", "NRF2"],
        "Pathways": ["RAS", "MAPK"]
    }
}

Given the cancer subtype "{cancer_subtype}", list the most relevant genes in the following JSON format:
{
    "{cancer_subtype}": {
        "Mutation-Based": ["Gene1", "Gene2"],
        "Expression-Based": ["GeneX", "GeneY"],
        "Pathways": ["PathwayA", "PathwayB"]
    }
}
"""
```

- **<u>Hierarchical Reasoning for Layered Querying</u>**

  Why? Instead of asking the model for all relevant genes at once, we will break down the problem into multiple biological steps (e.g., mutations, signaling pathways, drug response genes). This will help model to reach the final step correctly instead of getting diverted in between and will serve as a reasoning path for the model.

  How? We will structure the prompt to first focus on the broader categories and then hierarchically come down the ladder to reach final answer.

  Example Prompt for Hierarchical Reasoning:

```
hierarchical_prompt = """
Step 1: Identify the primary oncogenic pathways involved in "{cancer_subtype}".
Step 2: Based on these pathways, list key genes that regulate or influence them.
Step 3: Rank genes by their biological significance and mutation prevalence in patient datasets
Step 4: Format the final output in JSON with "gene_name" and "relevance_score".
"""
```

- **Multi-Perspective Prompt refinement for Robustness**

Why? Cancer gene association can be viewed definitely from various different perspective

- Mutation-based associations (e.g., tumor suppressor genes, oncogenes).
- Pathway-based associations (e.g., genes involved in PI3K, RAS, and p53 pathways).
- Therapeutic associations (e.g., genes targeted by existing drugs).

How? Model will be prompted to take into account multiple perspective and finally, aggregate the responses.

Example Multi-Perspective Prompt:

```
multi_perspective_prompt = """
For the cancer subtype "{cancer_subtype}", provide a gene association list considering three perspectives:
1. Mutation-based associations (e.g., oncogenes, tumor suppressor genes).
2. Pathway-based associations (e.g., signaling networks commonly altered).
3. Therapeutic relevance (genes targeted by approved or experimental drugs).
Ensure that the output is structured in JSON format.
"""
```

This will help the model to broaden it reasoning radius, thus, ensuring a more comprehensive response.

- ## Self-Consistency for Response Validation

Why? LLMs sometimes generate different output for same prompt. Self-Consistency involves generating multiple outputs and comparing them to select most stable and repeated prediction.

How? We will query the LLM multiple times and compute the consensus score for each gene on the basis of number of their occurrence.

Sample code for Self-Consistency in Gene Prediction:

```
# we will Define multi-perspective prompt
prompt_template = PromptTemplate(
    input_variables=["cancer_subtype"],
    template="""
    For the cancer subtype "{cancer_subtype}", list the most relevant genes considering:
    1. Mutation-based associations
    2. Pathway-based associations
    3. Therapeutic relevance
    Format the output as JSON with "gene_name" and "relevance_score".
    """
)

# Quering LLM multiple times for self-consistency
cancer_subtype = "Glioblastoma Multiforme"
num_queries = 5  # Number of times we query the LLM

responses = []
for _ in range(num_queries):
    query = prompt_template.format(cancer_subtype=cancer_subtype)
    response = llm(query)
    responses.append(response)

#Now, we will extract and count gene occurrences across responses
gene_counts = Counter()
for response_json in responses:
    for gene in response_json:
        gene_counts[gene["gene_name"]] += 1

# Finally, Ranking genes by self-consistency score
ranked_genes = sorted(gene_counts.items(), key=lambda x: x[1], reverse=True)

print("Final Gene List with Self-Consistency Scores:")
for gene, score in ranked_genes:
    print(f"{gene}: {score}/{num_queries} occurrences")
```

Why this phase is Critical?

- **Few-shot learning** helps improve model performance by using previously known gene-cancer association
- **Hierarchical reasoning** prevents the model from guessing randomly by providing it a well-defined pathway to follow to reach the answer.
- **Multi-perspective prompting** ensures that model considers gene from different points of view.

- **Self-consistency method** helps in reducing hallucination by getting the response validated by the model itself.

Out of these four core techniques, whichever gives more accurate results, we will modify our prompt accordingly.

Phase 3: Simulated Expert Reasoning with Confidence Scoring & Source Attribution

To ensure the reliability of the response by LLM, we integrate a confidence scoring mechanism in the directly in the LLM's response by mentioning it explicitly in the prompt. And apart from this, we simulate the model as an oncological expert and direct it to always cite the reference or the topic's name from which it derived the information giving the answer.

Approach

- **Simulating the LLM as an Expert Oncologist**
    - The LLM will be instructed in the prompt itself to think itself as an experienced oncologist or experienced cancer specialist.
    - This would make the model to reply after proper reasoning as model will try to pretend to be an oncologist.
    - This make the model to be more contextually aware and prioritize its task while forcing model to do deep reasoning as oncologists use complex reasoning to interpret.
- **Generating Confidence Scores Directly**
    - The model will be explicitly asked in the prompt to assign a confidence score (0-1) for each recommended gene.
    - This will help us to know that which gene is model recommending confidently.
- **Forcing Justification & Source Attribution**
    - The model will be instructed to provide the reasoning before recommending genes.
    - It will also be prompted to mention the reference or cite the topic on the basis of which its giving response.

Why This Will Work

- **Built-in Expert Reasoning** → It ensures responsible as well as logical gene selection and also focusses on the context related to oncology regardless of other topics which prevents it from getting distracted.
- **Confidence Scoring** → It tells us that how confidently the model is responding, thus increasing reliability of its response.
- **Citing Sources** → This makes the process more transparent and easy error detection gets possible.
- **Direct & Simple** → It eliminates the need for external datasets or multiple LLMs

Sample Example of Prompt Structure

```
prompt = """
You are an expert oncologist specializing in cancer genomics.

Given the subtype {cancer_subtype}, identify the most relevant genes involved in its progression.
For each gene, provide the following in JSON format:

{
    "{cancer_subtype}": {
        "Mutation-Based": [
            {"gene": "Gene1", "confidence": 0.92, "reasoning": "Gene1 is frequently mutated in this
            subtype, particularly affecting tumor suppressor pathways.", "source": "p53 Signaling Pathway"},
            {"gene": "Gene2", "confidence": 0.85, "reasoning": "Gene2 mutations are observed in
            multiple patient datasets, impacting cell cycle regulation.", "source": "Cell Cycle Regulation"}
        ],
        "Expression-Based": [
            {"gene": "GeneX", "confidence": 0.89, "reasoning": "GeneX is consistently overexpressed
            in aggressive cases of this subtype.", "source": "Oncogene Activation"}
        ],
        "Pathways": [
            {"pathway": "PathwayA", "genes": ["Gene1", "Gene3"], "relevance": "PathwayA is known to
            drive tumor progression in this cancer type."}
        ]
    }
}

Make sure that:
1. You **assign a confidence score (0-1) for each gene**.
2. You **justify** each selection with expert reasoning.
3. You **mention the biological source/topic** that supports this selection.
"""
```

Example Output


```
{
    "Breast Cancer": {
        "Mutation-Based": [
            {
                "gene": "TP53",
                "confidence": 0.95,
                "reasoning": "TP53 mutations are found in over 50% of breast cancer cases, leading to
                              loss of tumor suppressor function.",
                "source": "p53 Signaling Pathway"
            },
            {
                "gene": "BRCA1",
                "confidence": 0.90,
                "reasoning": "BRCA1 mutations significantly increase the risk of hereditary breast cancer.",
                "source": "DNA Repair Mechanism"
            }
        ],
        "Expression-Based": [
            {
                "gene": "ERBB2",
                "confidence": 0.87,
                "reasoning": "ERBB2 is frequently amplified in HER2-positive breast cancer, promoting aggressive tumor growth.",
                "source": "HER2 Pathway"
            }
        ],
        "Pathways": [
            {
                "pathway": "PI3K-AKT",
                "genes": ["PIK3CA", "AKT1"],
                "relevance": "The PI3K-AKT pathway is a key driver of breast cancer cell proliferation.",
                "source": "Oncogenic Signaling"
            }
        ]
    }
}
```


Phase 4: Bias Mitigation & Hallucination Control via Logical Consistency & Self-Validation

Approach

Our strategy will be to focus on self-consistency mechanism within the fine tuned oncological LLM itself. The goal is to ensure that the generated responses are biologically sound, accurate and free from hallucinations.

To achieve this, we will implement three simple steps:

Logical Consistency Checks

We will simulate an expert-like reasoning process where the model evaluates its own responses against learned oncological principles. If any inconsistency or contradiction is detected, the model must refine its answer before finalizing the output.

- The LLM will be prompted to **self-check** whether the suggested gene is associated with the given cancer subtype.
- It must **justify each recommendation** using oncological reasoning.
- If the explanation contradicts established biological mechanisms, the model is forced to **reevaluate** and refine the response.

Self-Consistency Based Re-Evaluation

We will make our fine-tuned model to generate **multiple responses independently** and then compare them for stability.

- The model will **queried multiple times** with slightly varied phrasings of the same prompt to check whether the core gene recommendations remain consistent.
- If there is high variability in its responses, it triggers a **confidence downgrade**, prompting further reevaluation.

Expert-Like Cross-Questioning

Instead of relying on external validation, we simulate an expert oncologist challenging the LLM's reasoning:

- The model will need to **defend** its predictions in a simulated expert discussion format.
- It must explicitly mention why each gene is relevant and **what evidence (literature, known mechanisms, pathway alignment, etc.) supports it**.
- If the reasoning will be weak or vague, the model revises its response before finalizing the output.

Why This Works Better

- **Eliminates Hallucinated Genes –** The LLM will justify each and every response, reducing unsupported claims.
- **Improves Stability –** If responses fluctuate too much across re-queries, we will flag them for expert review.
- **Simulates an Expert's Thought Process** – The model gets forced to reason critically, just like a human oncologist would.

This method therefore ensures that even without external databases or additional LLMs, we can generate biologically sound and scientifically correct gene associations.

A sample Implementational code

```
prompt_template = PromptTemplate(
    input_variables=["cancer_subtype"],
    template="""
    As an oncological expert, identify key genes associated with "{cancer_subtype}".
    Ensure your response follows these checks:
    1. Logical Consistency: Does the gene have a known oncological basis?
    2. Self-Validation: Can you justify each gene's inclusion based on pathways or mutation relevance?
    3. Consistency Check: If re-asked, would your response remain stable?

    Provide your answer in JSON format:
    {
        "Cancer Subtype": "{cancer_subtype}",
        "Genes": [
            {"gene_name": "TP53", "justification": "TP53 mutations are common in multiple cancers due to its tumor suppressor role."
            {"gene_name": "PIK3CA", "justification": "PIK3CA mutations drive oncogenic signaling in breast and colorectal cancer."}
        ],
        "Confidence_Score": 0.92
    }
    """
)
```

Phase 5: Domain Expert Validation – Closing the Loop for Precision

While reaching to this phase, our prompt will get robust enough with all innovative techniques applied ensuring the correct response, but then also we can go for final adoption without a manual review of doubtful recommendation by model. This phase integrates expert biologist and oncologist review of model's output which are doubtful or flagged in early phases.

How It Works

- Expert will review the flagged gene recommendation of a particular cancer subtype.
- If it's not biologically sound and incorrect, then we will flag the gene as incorrect and its presence in the future predictions is reconsidered.
- Experts feedback are used to iteratively refine prompts, improve the reasoning chain of the model and also ensure more accurate and precise output.
- The model can be finetuned periodically based on the expert reviews of validated gene lists, thus creating a self-improving automated AI pipeline.

A sample implementational code

```python
# Sample model-generated gene predictions for Lung Cancer
predicted_genes = {
    "TP53": True,   # Expert confirms correctness
    "KRAS": True,   # Expert confirms correctness
    "XYZ1": False,  # Expert marks as incorrect
    "ABC2": False   # Expert marks as incorrect
}

# Designing Function to filter genes based on expert feedback
def validate_predictions(predicted_genes):
    validated_genes = {gene: status for gene, status in predicted_genes.items() if status}
    return validated_genes

# Extracting validated gene list
final_gene_list = validate_predictions(predicted_genes)

# Displaying validated genes
print("Final Approved Gene List:", list(final_gene_list.keys()))
```

Why This Matters

- It helps us to **Removes false positives** before adoption.
- It ensures that only **biologically relevant genes** get suggested.
- It leads to the creation of a **self-improving pipeline** by incorporating expert feedback.

So, this phase ensures expert intervention thus making our pipeline biologically relevant, reliable and dynamically refined over time.

Phase 6: Integration of Pipeline with cBioPortal

To make the gene-cancer association easily accessible, we will integrate the pipeline with the cBioPortal using a FastAPI-Powered API Layer.

Approach

- The API will accept the user query containing the cancer subtype and will return the associated gene with confidence score (If required) in JSON Format.
- A dedicated UI section on cBioPortal will make it easy for the users to seamlessly interact.
- This seamless integration will **empower researchers with real-time AI-driven insights**, and thus, enhancing and advancing genomic data exploration.

Thus, by integrating our pipeline with the cBioPortal pages, we will minimize the distance between AI driven discovery and practical research applications.

Final Deliverable

A Highly Optimized and Reliable Biomedical cancer gene recommendation pipeline with key deliverables as mentioned below:

- LLM-Driven Gene Prediction System
- Multi-Step Reasoning with Confidence Scoring Framework
- Hallucination-Free and No Bias Response
- Expert-in-the-Loop Validation Framework
- Integration with cBioPortal for Real-World Usability
- Comprehensive Documentation & Deployment Pipeline

Timeline/Project Plan

After thoroughly reading the GSoC 2025 timelines, I have planned to carry out five phases of the project with a time allocation designed to guarantee the hassle-free completion of each phase within its time limit.

Time Frame	Start Date	End date	Task
Community Bonding	8th May	1st June	Discussion about the new ideas, suggestions, and getting into implementation details with mentors.
Coding begins Officially			
Phase 1	2nd June	4th June	Setting up and loading the fine-tuned LLM trained on Oncological literature. Performing initial basic prompt engineering.
	4th June	7th June	Setting up other BioMedical LLMs like MedBERT, BioGPT, BioBERT, etc. for implementing the ensemble approach.
	8th June	12th June	Comparing ensemble and single LLM approach by prompting all models. Keeping the better approach side for further processes and moving with single LLM approach to next phase, i.e., our fine tune LLM. All processes in coming phases will be applied to both approaches.
Phase 2	12th June	17th June	Refining prompt with multiple few-shot learning, hierarchical reasoning and multi perspective technique. Trying out various refined prompts.

	18th June	22nd June	Merging semantically all the better performed tried out techniques in above step and re-refine the prompt with self-consistency technique for validation.
	23rd June	27th June	Finally, designing the refined prompt on the basis of findings from last two steps. Checking out the performance of final prompt as till now and doing required changes.
Phase 3	28th June	30th June	Carrying out Simulated Expert Reasoning by trying out different prompts. Refining the prompt with the simulated expert reasoning integrated.
	31st June	2nd July	Modifying the prompt to give confidence score and reference behind each response. Merging it semantically into the lastly refined prompt.
Phase 4	3rd July	8th July	Bias mitigation and hallucination control. Advancing the prompt to make the model check and validate its response before responding.
	8th July	10th July	Enabling logical-consistency and self-validation and finally building up the refined prompt.
Phase 5	10th July	18th July	Now, Domain expert validation of all the flagged or doubtful responses.
Midterm Evaluation	14th July	18th July	
Phase 5 Continues	19th July	21st July	Designing function to generate a desired format of these validated responses. Adding these genes explicitly into the prompt, so the model does not repeat the same error.
	22nd July	24th July	Designing the function that take expert verified gene list and finetune the model on it. And passing the list of these genes to the team too.
			If unsatisfactory response comes up, we will again refine the prompt.
	25th July	26th July	Final validation by the Experts and deciding that which approach to take up, multi model ensemble one or our fine-

			tuned single LLM approach.
	27th July	1st Aug	Now finally setting up the fully optimized and refined pipeline of gene recommendation according to the selected approach.
	2nd Aug	3rd Aug	Final validation of pipeline's responses by Experts and mentors.
	4th Aug	8th Aug	Completing left over task and changes suggested by mentors.
Phase 6	9th Aug	12th Aug	API Development and CBioPortal Integration with this pipeline.
	13th Aug	14th Aug	Final Testing, Optimization and deployment.
	15th Aug	22nd Aug	Building up the robust and comprehensive documentation for the process.
	23rd Aug	25th Aug	Discussing Post GSoC plans with mentors and getting their suggestions.
Final Mentor Evaluation	26th Aug	1st Sep	

Commitments

In weekdays (Monday through Friday), I can easily put **5-6 hours** to project work, ensuring alignment with NY (or any other, if required) working hours which will allow for efficient communication and collaboration. In order to compensate for the reduced working hours during weekdays, I'm happy to put extra hours on weekends (Saturday and Sunday) – around **7-8 hours** per day. And this extended weekend commitment will make sure that I meet the proposed timeline while keeping in mind the time zone difference.

Additional Information about the Timeline

- The timeline mentioned above is subject to change and is only an approximate outline of my project work. I will stick to or exceed this schedule and create a more detailed schedule during the Pre GSoC and community bonding phase.
- **I've no other commitments during the summer** and can dedicate **40 to 45 hours a week**. During the last month of the project, my college will begin, and I'll be able to

commit a max of **25 hours a week**. Due to the same, I will do a significant portion of the work before this period.

- Time will be divided (according to workload) each week amongst planning, learning, coding, documenting and testing. All documentation will go hand in hand with the development.

POST GSoC PLANS

I am committed to the long-term sustainability and success of the AI/LLM gene Recommender project. With the initial scope of this proposal primarily aims at developing a robust gene Recommender pipeline with advanced Bias Mitigation, Hallucination Reduction and highly robust and innovative prompting techniques. I recognize the potential for further enhancements and advancements.

- Instead of manually validating the response with the help of experts, we can use Multi-Tier RAG pipeline powered by FAISS to get extra fast retrieval for real time validation of response.
- Two-Tier finetuned Biomedical LLMs can be established to enable the more efficient functioning and accuracy of above pipeline. One LLM will be used to extract and mold the contextual data according to the user query and other finetuned LLM would be provided with this information of first LLM to validate response. And along with process, wrongly detected genes would be getting flagged continuously without human intervention and on weekly basis finetuning of primary LLM (finetuned on oncological data) will take place.
- Thus, the more the query will be asked, the more accurate the pipeline will become.

  Link to the above Post GSoC Idea Proposal: Click

By aiming for these enhancements, I plan and hope for the reliability and high impact value of AI Powered Gene Recommender System beyond GSoC timeline.

Final Notes

I assure you that if I get selected to work with cBioPortal this summer, I will surely try my level best to make this project a great success and will love to continue working with cBioPortal in other project even after the summer. I would really love to work under such great mentors of the organization and it would be a golden opportunity for me to contribute to Open- Source community under the guidance of my mentors. This will

provide me a platform to collaborate, brainstorm and work together with the world level developers who will be there guiding me as my mentor throughout GSoC 2025.

Also, for some reason, If I am not selected this year even then I will try to contribute to this and other projects as much as possible and retry again next year.

Looking forward to working with you

Thanks and Regards

Ayush Shaurya Jha