Team Project Proposal: LLM-Powered Cell Type Annotation Application

1. Project Title

Enhancing GPTCelltype for Customizable LLM-Based Cell Type Annotation

2. Team Member(s)

Nakial Cross

3. Problem Statement and Motivation

Cell type annotation is a pivotal step in single-cell RNA sequencing (scRNA-seq), where domain experts compare differentially expressed genes (DEGs) to canonical markers. This manual process is labor-intensive, subjective, and time-consuming. Automated methods exist (SingleR, ScType, CellMarker2.0), yet many still rely on reference datasets or produce inconsistent results.

Recent advances with GPT-based methods (e.g., GPTCelltype, an R package using GPT-4) show promising accuracy and reduced manual effort. However, GPTCelltype is currently limited to one model and lacks a graphical interface for interactive verification of annotations. This project seeks to expand GPTCelltype's capabilities by integrating LM Studio for multi-model support and implementing a Shiny-based UI to streamline annotation review.

4. Application Concept and Features

a. Customizable LLM Selection and Integration
   - Modify GPTCelltype to allow use of different language models (OpenAI GPT-4, Mistral, or Llama) via LM Studio.
   - Enable offline usage for cost savings and data privacy.
   - Provide user-adjustable prompt engineering (e.g., controlling chain-of-thought prompts, specifying confidence thresholds).

b. Interactive Shiny Application
   - Offer a front-end interface for uploading and preprocessing scRNA-seq datasets (e.g., Seurat or Scanpy objects).
   - Visualize UMAP clustering with color-coded cell type annotations from GPT-based models and other tools like SingleR.
   - Provide on-the-fly comparison of model-generated annotations versus manual (gold standard) labels.

c. Scalable Batch Processing

- Implement parallel or batched queries for large datasets (>100,000 cells).
- Maintain performance logs, including query speeds and model token usage, to address cost or runtime constraints.

5. Approach and Development Plan

Phase 1: Research and Dataset Selection

- Familiarize with GPTCelltype internals (prompt structure, API usage).
- Identify publicly available benchmark datasets (e.g., GTEx, HCL, Azimuth) with well-established cell type labels.

Phase 2: Model Integration

- Implement an interface within GPTCelltype to query LM Studio models.
- Test open-source LLMs locally, ensuring fine-tuning capabilities and domain-specific adaptation.
- Confirm tokenization and prompt formatting for each model to maintain consistency.

Phase 3: Shiny Application Development

- Build a Shiny app that can load scRNA-seq data, perform basic quality control, and generate a UMAP embedding.
- Integrate GPT-based annotation calls into the backend. Display results in a user-friendly dashboard.
- Create interactive features (dropdowns, hover text, annotation editing, confidence scores).

Phase 4: Testing and Validation

- Compare GPT-based annotations to manual ones using Cohen's kappa, precision, and recall.
- Benchmark accuracy and runtime against SingleR, ScType, and CellMarker2.0 across multiple datasets.

Phase 5: Refinements and Ethical Considerations

- Document all model biases (e.g., hallucinated cell types), recommended disclaimers, and best practices for human oversight.
- Add a confidence interval or probability score to each annotation to help users validate uncertain predictions.
- Ensure data security when handling user data, especially for local vs. cloud-based model inference.

6. **Data Source and Preprocessing**

- Potential Datasets: GTEx (multiple human tissues), HCL (multi-tissue, large sample), Azimuth (reference-based cell annotations), previously analyzed scRNA-seq samples.
- Preprocessing Steps: Normalize gene counts, filter poor-quality cells, identify top DEGs (e.g., Wilcoxon rank-sum) for each cluster, and create structured prompts for LLM queries.

7. **Testing and Validation Plan**

Quantitative Measures

- Cohen's kappa (κ) for inter-annotation reliability.
- Per-class precision, recall, and F1 scores compared with manual benchmarks.
- Scalability metrics (runtime, memory usage) in single-threaded vs. parallel modes.

Qualitative Measures

- Case studies on ambiguous or novel cell types.

Possible Pitfalls

- LLM hallucination leading to invalid cell type assignments.
- Model reliance on training data that might not include rare or novel cell types.
- Variable performance depending on the user's selected language model (GPT-4 vs. an open-source alternative).

8. **Ethical Considerations and Bias Mitigation**

- Hallucination and Overconfidence: Encourage user review before publication or further analysis. Provide confidence scores and gene-level rationale where possible.

9. **Expected Challenges**

- Technical: Efficient integration of multiple LLM backends with GPTCelltype. Handling large single-cell datasets requiring thousands of queries.
- Data-Related: Ensuring consistent performance across tissues and species with different gene names or incomplete references.

10. **Milestones and Timeline**

Week 1-2: Review GPTCelltype code, choose target datasets.

Week 3: Integrate LM Studio-based model selection into GPTCelltype.

Week 4: Create prototype Shiny app (basic UI, data input, UMAP generation).

Week 5: Implement GPT-based annotation calls into Shiny app. Begin basic performance testing.

Week 6: Validation experiments (quantitative and qualitative). Collect user feedback.

Week 7: Refinements, error handling, bias considerations.

Week 8: Final testing, complete project deliverables (presentation, documentation).

11. Expected Project Deliverables

a. Customizable GPTCelltype package with LM Studio integration (source code, documentation).
b. Interactive Shiny application supporting scRNA-seq data upload, UMAP visualization, GPT-based annotation, and comparative benchmarking.
c. Final presentation covering approach, results, scalability, and lessons learned.
d. Written report detailing the system architecture, experiments, validation results, limitations, ethical considerations, and potential directions for future work.