The Journal of Open Source Software

GenrAltOR: Generative AI for 'Omics Research

Daniel Claborne [1], **Matthew Jensen**[1], **Javier E. Flores** [1], **Lisa Bramer** [1], **and Samantha Erwin** [1¶]

1 Pacific Northwest National Laboratory ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Abhishek Tiwari

Submitted: 29 July 2025
Published: unpublished

Summary

We present software to fine-tune Llama 3 using Retrieval Augmented Fine-Tuning (RAFT) on proteomics literature. The package includes a command-line interface (CLI) that simplifies the fine-tuning process. The resulting model answers biomolecule-related queries using biological context.

Advances in LLMs enhance human-dependent tasks like interpreting biomolecule sets identified as important for model predictions. Domain experts can query an LLM for biological insights and draw inferences contextualized by the LLM's knowledge. Most LLMs are general-purpose, trained on broad datasets like social media. These lack the domain-specific language needed for 'omics queries. Our work uses RAFT (Zhang et al., 2024) to adapt an open-source LLM into an AI-assistant for domain experts. RAFT fine-tunes models by including irrelevant context in question-answer tasks, making them more robust than traditional retrieval-augmented generation (RAG) systems (Zhang et al., 2024). We provide a CLI to perform RAFT, from data collection to training, enabling researchers to create their own AI-assistants for biological research.

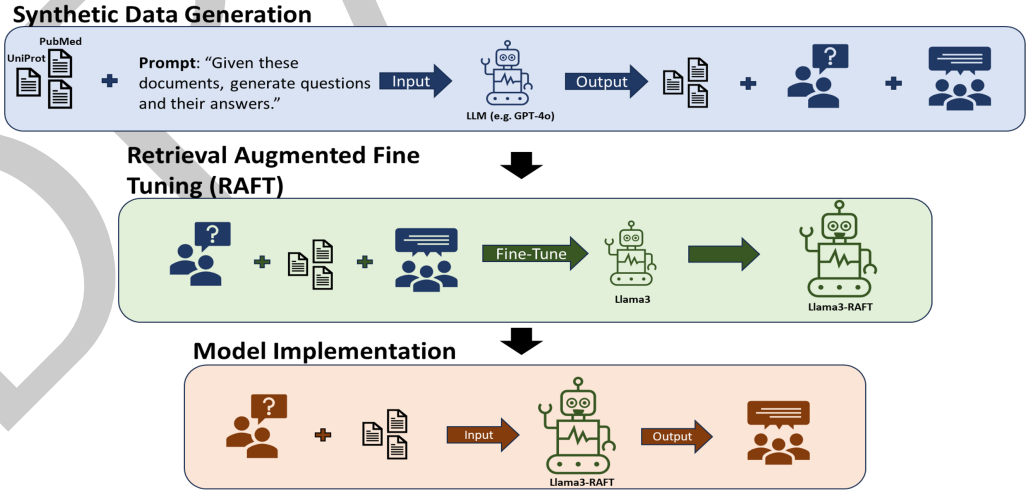

Figure 1: GenrAltOR Process Overview. Synthetic training data are generated using ChatGPT-4o. These question-answer-context triplicates are then used to fine-tune Llama 3 in a RAFT context. The output RAFT model is then implemented/evaluated on a hold-out set of generated triplicates.

Statement of Need

RAFT has shown up to 30% performance gains over standard fine-tuning and RAG (Zhang et al., 2024). However, implementing RAFT presents many technical barriers for researchers.

Our software offers an easy-to-use package for performing RAFT in proteomics and serves as a template for extending RAFT capabilities to other areas of molecular biology.

An overview of our development of a RAFT model is provided by Figure 1. The general steps are:

1. PubMed abstracts and UniProt data were retrieved using public APIs.
2. Data chunks were processed with GPT-4o to generate synthetic question-answer pairs. Context chunks were grouped using text embeddings. An example is shown in Figure 2.
3. Contexts were augmented with random 'distractor' documents to vary relevancy levels.
4. A training split of synthetic data was used to fine-tune Llama 3 for text completion.

We evaluated RAFT-Llama 3 against base Llama 3 using AlignScore (Zha et al., 2023).

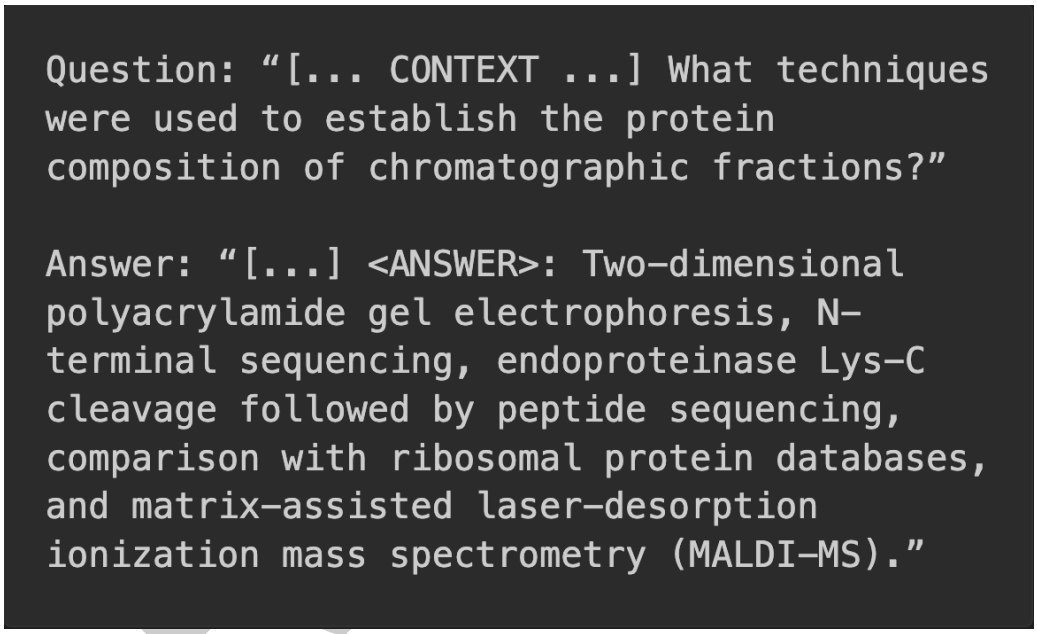

Figure 2: Example of a QA-pair generated using sampled context. The [... CONTEXT ...] chunk is a collection of documents that may or may not contain the text used to generate the QA-pair

Example Implementation

The process is encapsulated in a Python package with a command-line interface. First, we retrieve context in the form of article abstracts and information about protein-protein interactions and associated pathways, starting with a list of protein identifiers in `data/examples/uniprots.txt`:

```
python -m genraitor data:context \
--uniprot_ids=./data/examples/uniprots.txt \
--output_dir=./data
```

This will produce two files in `./data`, one (`uniprot_context_results...`) with the raw results of querying UniProt for pathway information and abstracts, and the other (`uniprot_context_postprocessed...`) with context derived from those results and usable by the `RAFTDatasetPack` class from the `llama-index` Python package (Liu, 2022).

To create synthetic question-answer pairs from this context, use the CLI with an OpenAI API key and optionally a Hugging Face API key:

```
# set keys
export HF_TOKEN=<your-hf-token>
export OPENAI_API_KEY=<your-oai-key>

python -m genraitor raft:data \
--embed local \
--context_path /path/to/context.txt \
--output_path ./data/training/hf_dataset
```

This produces the training dataset at `./data/training/hf_dataset`, loadable via:

```
from datasets import load_from_disk

dataset = load_from_disk('/path/to/save_data_folder')
```

To fine-tune Llama 3 with RAFT, use the CLI target `train:raft`:

```
python -m genraitor train:raft \
-t /path/to/raft_data \
-m meta-llama/Meta-Llama-3.1-8B \
-n data/finetuned
```

The fine-tuned model is saved in `data/finetuned` and can be loaded via:

```
from transformers import (
    AutoModelForCausalLM,
    AutoTokenizer
)

tokenizer = AutoTokenizer.from_pretrained('./data/finetuned', padding_side="left")
model = AutoModelForCausalLM.from_pretrained("./data/finetuned")
```

Commands are configurable via flags or environment variables. Use `--help` for options:

```
python3 -m genraitor train:raft --help
```

In addition to unifying data processing, model training, and result evaluation, this effort extended the Python package `llama-index` by:

- Allowing configurable system prompts for generating questions.
- Modifying the `get_chunks` method to respect the `chunk_size` argument, optimizing text length for training.

Results

We evaluated AlignScores on held-out question-context-answer triplets, benchmarking against base Llama-3. Figure 3 compares RAFT-Llama 3 (blue) and base Llama 3:

- RAFT shows a slightly higher mean AlignScore.
- RAFT's distribution is more left-skewed (towards lower scores).

These results suggest RAFT marginally improves response quality. Additionally, our software streamlines RAFT implementation, enabling researchers to develop RAFT-Llama 3 models.

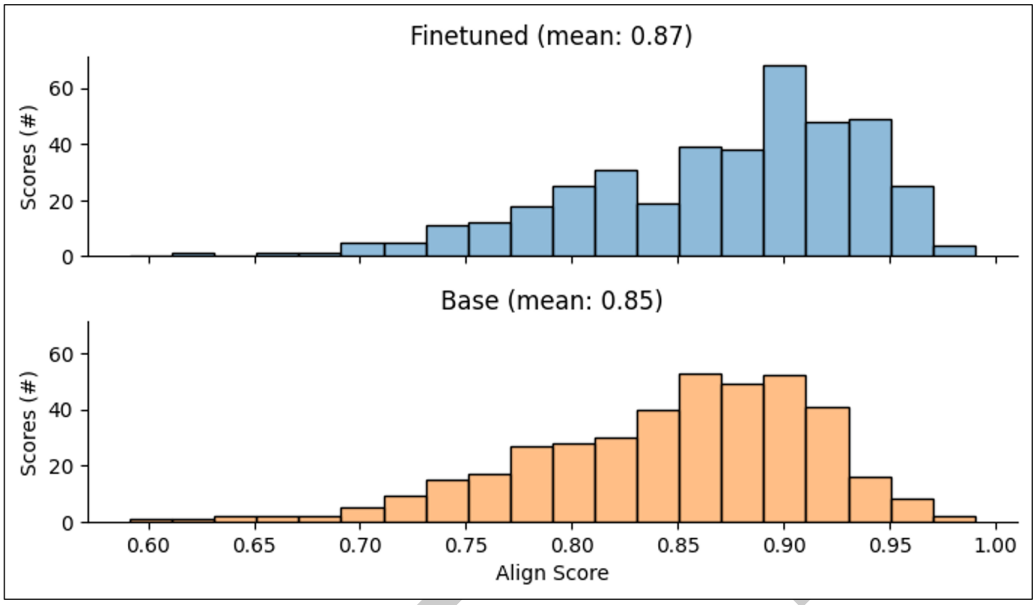

Figure 3: Distribution of AlignScores for the RAFT-Llama 3 ("Finetuned") and the base Llama3 model.

Discussion/Limitations

Some challenges remain:

1. **Resources** Large context chunks strained GPU memory, requiring limitation of context size.
2. **Proteomics focus.** The package focuses on proteomics; adapting to other biomolecules requires custom code, though training is domain-agnostic once data are ready.
3. **Dependencies.** Fine-tuning involves many dependencies, risking version mismatches (e.g., CUDA, PyTorch).

RAFT showed marginal AlignScore improvement over base Llama-3 on our QA task. This likely reflects our limited evaluation: a single metric (AlignScore) on synthetic data that could benefit from expert curation.

Our software package allows users to create context for performing RAFT in the proteomics domain given a list of proteins of interest. We hope this provides an easy-to-use base for researchers to explore the relationships between proteins identified in their experiments and expand the use of RAFT to different domain areas.

Acknowledgements

The research described herein was funded by the Generative AI for Science, Energy, and Security Science & Technology Investment under the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory (PNNL), a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy. This work was also supported by the Center for AI and Center for Cloud Computing at PNNL.

References

Liu, J. (2022). *LlamaIndex*. https://github.com/jerryjliu/llama_index

Zha, Y., Yang, Y., Li, R., & Hu, Z. (2023). *AlignScore: Evaluating Factual Consistency with a Unified Alignment Function*. arXiv. https://doi.org/10.48550/ARXIV.2305.16739

Zhang, T., Patil, S. G., Jain, N., Shen, S., Zaharia, M., Stoica, I., & Gonzalez, J. E. (2024). *RAFT: Adapting Language Model to Domain Specific RAG*. arXiv. https://doi.org/10.48550/ARXIV.2403.10131