Open Source for Science Fund

# Open Source for the Life Sciences (OS4LS)

## Full Application

## Summary

| | |
|---|---|
| **Submission ID** | OS4LS- 2f3c47a2-ce57-40a4-98b2-baa6fdfefc85 |
| **Submission Date** | 2026-07-20T20:47:43.251Z |
| **Applicant Last Name** | Magdinier |
| **Proposal Title** | Auditable and Reproducible LLM Workflows in OpenRefine |
| **Funding Track** | Track 1: Domain-specific Tools |
| **Total Budget** | 249866.66 |

# Section 1: Applicant Information

| Question | Answer |
|---|---|
| First Name | Martin |
| Last Name | Magdinier |
| Email | martin@openrefine.org |
| Applicant Organizational or Institutional Affiliation | OpenRefine |
| Country of Residence | Canada |
| Have you previously received funding as a PI under the CZI EOSS program? | Yes |
| Proposal Team: Full Name | Sunil Natraj |
| Proposal Team:Email | sunil.natraj@gmail.com |
| Proposal Team: Affiliation | OpenRefine |
| Proposal Team: Country of Residence | India |
| Proposal Team: Role | Key Personnel |

# Section 2: Organization Information

| Question | Answer |
|---|---|
| Organization Name | Code For Science and Society |
| Organization Street Address | 3439 SE Hawthorne Blvd, #247 |
| Organization City | Portland |
| Organization State/Province | OR |
| Organization Country | United States |
| Organization Website | https://www.codeforsociety.org/ |
| Organization Tax ID | 81-3791683 |
| Organization Type | Fiscal sponsor |
| If Other: | Fiscal sponsor |
| Administrative Contact First Name | Linda |
| Administrative Contact Last Name | Green |
| Administrative Contact Email | Linda@codeforsociety.org |
| Signing Official First Name | Shannon |
| Signing Official Last Name | Dosemagen |
| Signing Official Email | Shannon@codeforsociety.org |
| Financial Contact First Name | Ron |
| Financial Contact Last Name | Brooks |
| Financial Contact Email | Ron@codeforsociety.org |

# Section 3: Proposal Information

| Question | Answer |
|---|---|
| Proposal Title | Auditable and Reproducible LLM Workflows in OpenRefine |
| Purpose | OpenRefine will improve its large language model (LLM) extension to support reproducible, interoperable, and scalable AI-assisted curation of life-science data. |
| Categories | Knowledge representation and ontologies, Workflows and computational pipelines, Interoperability |
| If Other: | |

| | |
|---|---|
| Short Summary | OpenRefine is a free, open-source, local-first tool for cleaning, transforming, reconciling, enriching, and reviewing messy tabular data. It is used in the life sciences to prepare, organize, and improve the quality of research data. Its built-in operation history lets users document, audit, and rerun data-wrangling steps across similar datasets. Building on OpenRefine's EOSS-5 reproducibility work, this proposal extends that model to AI-assisted curation.<br><br>Many LLM workflows make it difficult to apply the same prompt across a dataset, review outputs row by row, or preserve the information needed to reproduce results. OpenRefine already has a community-contributed LLM extension that integrates LLM prompts to OpenRefine's existing row-by-row tabular curation workflow. Additionally, it provides an interactive environment for users to inspect inputs and outputs, filter errors, and revise prompts. However, the recent popularity of the extension exposed a gap between the opportunities offered by the tool and OpenRefine's life science user base.<br><br>Life-science researchers are increasingly exploring large language models for data extraction, annotation, classification, and validation. These tasks are especially relevant to clinical text curation, biodiversity and specimen records, and other workflows in which researchers need to convert text-rich records into structured fields. Additionally, a large portion of the research process consists of extracting and digesting information from the existing literature, either for publication or for the creation of novel molecules and organic structures. Researchers are familiar with reference software to manage and manually annotate their bibliographic data, such as Zotero and Mendeley. However, literature reviews are tedious and come with a high risk of missing key information buried in a lengthy text or mentioned as a footnote to a more impactful finding. LLMs are an appealing solution to that problem, but come with an additional risk of introducing hallucinated data when used without an appropriate tool to audit and edit results. They are also rarely designed to apply the same prompt across a dataset, review outputs row by row, or preserve the information needed to reproduce results.<br><br>The proposed work would make this LLM-assisted workflow more usable, reliable, and reproducible for life-science research. The result would give researchers a human-in-the-loop interface for AI-assisted data curation within a tool already used in research and life-science workflows. As the use of AI in academia is increasingly discussed and debated, this project would support AI-assisted annotation and information extraction across tabular datasets in ways that are auditable, verifiable, documented, and reproducible, while preserving the transparency expected in research workflows. |

| | |
|---|---|
| Expected Value | The project will add auditing and reproducibility to emerging uses of LLMs in life-science research such as LLM-assisted data annotation, classification, and extraction.<br><br>The extension will allow researchers to import article metadata, abstracts, screening notes, or extracted passages into OpenRefine. Then use an LLM to classify papers against inclusion criteria, extract keywords or values, or flag whether a paper supports or contradicts a research claim. Outputs will appear as reviewable columns that researchers can filter, inspect, edit, and compare against the source text and existing metadata.<br><br>This pattern also applies to scientific or clinical text curation for extracting structured attributes from sample labels and notes, including identifiers, locations, and specimen attributes, as well as harmonization workflows that classify variables or prepare values for review against authoritative vocabularies. In each case, the LLM output is not the final assessment; it is an iterative curation step that domain experts need to review, track, and document.<br><br>With the extension, OpenRefine will provide a guided interface for AI-assisted workflows, benefiting domain experts without requiring custom scripts or dedicated software support. Researchers will be able to classify and extract at scale while keeping outputs reviewable and methods documentation reusable. This would extend the reproducibility work developed through EOSS-5 to current AI-enabled life-science data curation. |

| | |
|---|---|
| Landscape Analysis | Researchers who want to apply AI to tabular data currently choose among several tool categories:<br><br>* Spreadsheet software is common in research because it is familiar and convenient, but it offers limited support for tracking, auditing, and reusing data-wrangling steps.<br>* Programming languages such as Python, R, MATLAB, and SAS can leverage LLM-specific packages such as LangChain and LlamaIndex, but they have a steep learning curve and lack an interactive tabular interface for reviewing edits.<br>* Chat-based tools make LLMs accessible to researchers, but they are not designed to support reproducible structured data curation.<br>* Emerging no-code batch LLM tools like BatchGPT or Airtable AI often support row-wise prompting, but lack data curation features and robust support for scientific reproducibility.<br>* Spreadsheet-like AI tools such as Hugging Face AI Sheets or OpenClay show growing demand for applying LLM prompts across tabular data.<br>* Enterprise workflow platforms such as Dataiku and self-hosted automation platforms like n8n can support governed AI workflows, but they come with a complex architecture.<br><br>OpenRefine already supports a wide range of scientific communities, including social, natural, and health sciences. Created more than 10 years ago, it now counts over 20,000 downloads per month and about 1,200 academic citations per year. |

| Software project details | |
|---|---|
| Software Project Names | OpenRefine, OpenRefine LLM Extension |
| Software Project Repository URLs | https://github.com/OpenRefine/OpenRefine, https://github.com/sunilnatraj/llm-extension |
| Software Project Website URLs | https://openrefine.org/ |
| Software Project Short Descriptions | OpenRefine is a powerful free, open source tool for working with messy data: cleaning it; transforming it from one format into another; and extending it with web services and external data. Our goal is to empower everyone to meaningfully engage with data by providing an accessible open source tool and nurturing a diverse, supportive community., The OpenRefine AI Extension bridges the power of modern language models with OpenRefine's robust data transformation capabilities. This extension enables users to leverage any LLM provider that supports a chat completion API endpoint, bringing AI-powered data wrangling, enhancement, and analysis directly into your OpenRefine workflows. |

| Software project information | |
|---|---|
| Software License | Permissive (e.g. MIT, BSD, Apache), Permissive (e.g. MIT, BSD, Apache) |
| If Other: | |
| Main Programming Language | Java, java |
| Canonical Citation | https://zenodo.org/records/10689569 |
| Code of Conduct Link | https://github.com/OpenRefine/OpenRefine/blob/master/CODE_OF_CONDUCT.md, https://github.com/OpenRefine/OpenRefine/blob/master/CODE_OF_CONDUCT.md |
| End-User Documentation Link | https://openrefine.org/docs/ |
| Contributor / Developer Guidelines Link | https://openrefine.org/docs/technical-reference/contributing/, https://github.com/sunilnatraj/llm-extension#install-this-extension-in-openrefine |
| Package Entry in a Package Manager Link | |
| Community Engagement / Q&A Forum Link | https://forum.openrefine.org/, https://forum.openrefine.org/ |

| Software project metrics | |
|---|---|
| Monthly Users Estimate | 20000.0, 1078.0 |
| Notes | We average 20,000 downloads per month<br>, The latest release of the extension on May 20, 2025, received 1,078 downloads |
| Number of Dependent Projects and Packages | |
| Notes | |
| Scholarly Citations Estimate | 9400.0, 20.0 |
| Notes | using Google Scholar search term `OpenRefine OR "Google Refine"`<br>, using Google Scholar search term `OpenRefine" + ("AI Extension" OR "LLM Extension")` |

# Section 4: Budget Information

| Question | Answer |
|---|---|
| Amount for Year 1 | 124933.33 |
| Amount for Year 2 | 124933.33 |
| **Total Amount** | 249866.66 |
| Recent Financial Support | Recent financial support for OpenRefine related to this proposal includes:<br><br>1) Chan Zuckerberg Initiative EOSS-5 grant, “Improving Reproducibility in OpenRefine”: USD 310,100 total, from Jan. 2023 to Dec. 2025. This grant supported OpenRefine’s reproducibility roadmap, including operation history, workflow export, project architecture, documentation, and community processes that this proposal builds on.<br><br>2) FLOSS/fund unrestricted contribution: USD 25,000, June 2026. This general operating support helps sustain OpenRefine’s core coordination, community support, and infrastructure while the project pursues longer-term grant and institutional funding.<br><br>3) Community giving and merchandise: approximately USD 1500 per year (based on last 6 months average) through one-time donations, merchandise profit, and recurring GitHub sponsorship.<br><br>In-kind support includes OpenRefine’s existing codebase, documentation, website, governance and release infrastructure, forum, community channels, and volunteer contributor base. Over the previous 12 months, the project counted on an active volunteer community with 243 new forum users, 228 forum topics, 7 active GitHub contributors, and 103 merged pull requests (excluding Dependabot). This volunteer and community infrastructure directly supports the proposed work by providing user feedback, testing, documentation review, issue triage, and maintainer oversight. |

## Section 5: Policy Section

| Question | Answer |
|---|---|
| Statement of PI Involvement | Confirm |
| Policies | Confirm |

Thank you for your submission. Please reach out to info@os4science.org with any questions and reference your submission ID as noted above in the email subject.

Open Source for Science Fund

# Open Source for the Life Sciences (**OS4LS**)

**Proposal Title:** Auditable and Reproducible LLM Workflows in OpenRefine

**Applicant Name:** **Martin Magdinier**

## Work plan (max 750 words)

OpenRefine is a free, open-source, local-first tool for cleaning, transforming, reconciling, enriching, and reviewing messy tabular data. This request supports a two-year effort to improve its community-driven large language model (LLM) extension. We aim to add structure, interoperability, scalability, and reproducibility to existing AI-assisted curation workflows and to expand them to life-sciences use cases.

In 2024, a forum discussion about using the same LLM prompt across thousands of rows developed into a broader exchange among users and contributors. It became a community priority in early 2025, and Sunil Natraj turned the initial code into a plug-and-play extension. Since then, the community has tested and documented the workflow through forum threads, tutorials, training, and community events. In July 2026, OpenRefine ran a community call focused on workflows using the extension with local or external model providers.

This exposed a gap between the opportunities offered by the tool and OpenRefine's life science user base. Life-science curation often involves repeated extraction, classification, enrichment, and validation across large datasets. Researchers also need to calculate costs for current and future projects, which is essential to justify institutional and external funding for research projects. To address this gap, this proposal builds on the existing extension and on open software development practices we used to identify needs, test workflows, and disseminate practices. The funded work is organized around three goals:

1. Structured and reproducible AI curation workflows;
2. Provider and model interoperability;
3. Scalable, token-efficient execution for larger tabular AI workflows.

Development will take place primarily in the LLM extension repository, with related work in the OpenRefine core repository where needed to maintain compatibility with OpenRefine's roadmap and the transition toward OpenRefine 4.x. Core changes will follow OpenRefine's standard contribution and review process. All code will be released under the BSD-3 license. The extension repository will be migrated to the OpenRefine GitHub organization.

The first part of our work will focus on technical improvements to AI-assisted curation workflows. We will develop templates and support multiple source columns as prompt context. To validate LLM answers, we will ensure outputs conform to predefined schemas and field types. We will add auditing and improve review workflows that address malformed or uncertain responses.

Then, we will make setup straightforward across hosted, local, institutional, and self-hosted model environments. We will design an LLM connection configuration schema, support import and export of the configuration file, and release preconfigured templates for major LLM providers. Where provider APIs support it, we will add model discovery and model-change detection for available models and include alerts related to deprecated models, changed

parameters, safer defaults, and fallback model settings. This will be complemented by an onboarding wizard with embedded setup guidance.

Finally, we will improve reliability and efficiency for larger tabular AI workflows. We will add progress and metrics reporting, including token usage, error rates, runtime, and cost estimates where providers expose this metadata. We will improve retry, timeout, fallback, and partial-failure handling so successful rows are preserved and failed rows can be filtered, inspected, and rerun. We will also investigate and implement prompt caching, prompt compression, streaming, or batch execution where supported by model providers.

The AI Extension Developer, Sunil Natraj, will lead extension development and technical design. The Managing Director, Martin Magdinier, will lead product and community work, including user research, life sciences use-case gathering, workflow documentation, community training, release communication, and community validation. This split reflects OpenRefine's prior grant experience: technical work is more effective, and resulting tools are more impactful, when paired with community engagement.

Each workstream will be delivered through iterative development cycles, public releases at meaningful checkpoints, documentation, and community validation. OpenRefine has already used community calls, forum discussions, tutorials, and conferences to document use cases, test emerging workflows, and disseminate practices, including community-wide BarCamps organized in 2024 and 2025. We will continue using community calls, targeted testing sessions, and future BarCamp sessions to validate structured curation templates, provider/model setup, auditability, and scalable execution patterns. Feedback will be converted into tracked issues, documentation updates, and release decisions.

Resources contributed outside the requested funding include the existing OpenRefine codebase, governance and release infrastructure, forum and community channels, documentation infrastructure, prior EOSS-funded reproducibility work, and the current community-contributed LLM extension. The OpenRefine Core Development team (2 members and 7 active contributors in the last 12 months)  and Advisory Committee will also contribute. Julie Faure-Lacroix, an Advisory Committee member at Laval University with life-science research experience, will help test the extension and develop use cases.

## Goals, Outcomes, Milestones and Deliverables (up to 5 goals, not included in the 750-word limit)

### Goal 1: Structured and reproducible AI curation workflows

**Outcome:** Researchers can use the OpenRefine LLM extension to produce structured AI-assisted curation results that are easier to review, correct, reuse, and document. The auditable workflow will be easy to extract and share, along with the proof of data integrity required by some scientific journals. Instead of treating LLM outputs as one-off chatbot responses, they will be able to work with AI-generated classifications, extracted fields, annotations, and candidate terms as part of an auditable OpenRefine workflow. This will make AI-assisted curation more practical for life-science datasets where results need to be checked by domain experts, applied consistently across many records, and described in reusable workflow documentation.

**Milestones & Deliverables:**

**1.1 Define and document life-science-relevant AI curation task templates** for extraction, classification, annotation, enrichment, tagging, and candidate term review, based on community use cases and existing OpenRefine practice. [Year 1]

**1.2 Add structured output configuration** so users can define expected response formats, including JSON/schema-based outputs, allowed values, required fields, and field types. This will include support for using multiple source columns as prompt context. [Year 1]

**1.3 Add validation and review workflows for structured outputs,** including ways to identify, facet, inspect, correct, and rerun malformed, uncertain, or invalid LLM responses. [Year 1]

**1.4 Add prompt and task template portability,** allowing users to save, reuse, import, export, and share structured task templates within an institution, project, or community of practice. [Year 1]

**1.5 Add a human-readable audit view for AI-assisted operations,** including clearer operation descriptions, prompt and model metadata where available, structured output settings, and methods-ready information that can support workflow documentation and publication methods sections. [Year 2]

**1.6 Publish documented example workflows** for selected life-science curation scenarios, including at least three tutorials or workflow notes showing how structured AI-assisted curation can be reviewed, corrected, exported, and reused. [Year 2]

**Success indicators:**

- At least 6 life-science AI curation templates are documented and released.
- Structured output configuration and validation are included in a public AI extension release.
- Users can identify, filter, inspect, and rerun malformed or invalid structured outputs.
- Prompt/task templates can be saved, reused, imported, and exported.
- At least 3 life-science workflow examples are published as tutorials, blog posts, or documentation pages.
- At least 2 community validation sessions are held.
- Feedback from community testing is converted into tracked issues, documentation updates, or follow-up release work in the LLM extension repository.

### Goal 2: Provider and model interoperability

**Outcome:** Researchers and institutions can use the OpenRefine LLM extension with a wider range of hosted, local, institutional, and self-hosted model services without needing to manually assemble endpoint URLs, model identifiers, authentication settings, and provider-specific parameters. This will lower the barrier to setting up AI-assisted curation workflows, make configurations easier to share and reproduce within an institution or project, and reduce fragility as model providers, APIs, and supported parameters change over time.

**Milestones & Deliverables:**

**2.1 Design a provider/model configuration schema** for the extension, including support for provider settings, model identifiers, authentication patterns, response formats, supported parameters (reasoning effort for example), and relevant execution options such as streaming or batch support where available. The schema will support import/export so configurations can be shared across projects, institutions, or communities. **[Year 1]**

**2.2 Build an onboarding wizard with embedded setup guidance** to help users connect model services without manually assembling provider configuration details. The wizard will include clear setup paths for common hosted, local, and institutionally managed model environments. **[Year 1]**

**2.3 Release reusable provider templates and contributor documentation** for adding and maintaining provider templates. Initial templates will cover common hosted and local model services, including Ollama and LM Studio, as well as commercial or institutionally managed endpoints. **[Year 1]**

**2.4 Add model discovery and model-change detection where provider APIs support it,** including checks for available models, deprecated models, changed parameters, safer defaults, and fallback model settings where feasible. **[Year 2]**

**Success indicators:**

- A provider/model configuration schema is released and documented.
- Users can import and export provider/model configurations, including institutional providers
- The onboarding wizard is included in a public release and validated through at least one community testing session.
- At least **5 reusable provider templates** are released, with additional community-contributed templates tracked where available.
- Contributor documentation for adding or updating provider templates is published.
- Model discovery or model-change detection is implemented for at least **2 providers** where APIs support it.
- Setup guidance is tested with researchers, curators, librarians, research analysts, or OpenRefine trainers who are not LLM API specialists.

### Goal 3: Scalable and token-efficient execution for AI workflows

**Outcome:** Researchers can use the OpenRefine LLM extension on larger tabular datasets with better visibility, recoverability, and control over long-running AI-assisted curation tasks. Instead of manually monitoring prompt execution row by row, users will be able to track progress and understand their token usage. Instead of manually restarting an entire task after a cryptic failure, they will be able to capture and monitor errors disclosed by the providers, save successful results, and rerun failed rows. They will be able to use provider-supported execution options such as batching, streaming, or prompt caching where feasible. This will make AI-assisted curation more practical for repeated classification, extraction, enrichment, and validation tasks across larger life-science datasets.

**Milestones & Deliverables:**

**3.1 Add metrics and progress reporting for LLM operations,** including token usage, success and error counts, runtime, and cost estimates where providers expose the relevant metadata. **[Year 1]**

**3.2 Improve error handling and recovery for long-running AI operations,** including configurable retry policies for rate limits, timeouts, malformed outputs, and transient provider errors. Successful rows will be preserved, and failed rows will be easier to filter, inspect, and rerun. **[Year 1]**

**3.3 Add token-efficiency improvements where feasible,** including prompt caching, prompt compression, structured outputs with tighter schemas, or other provider-supported approaches that reduce repeated token use for common curation tasks. **[Year 2]**

**3.4 Investigate and implement provider-supported batch or streaming execution paths where feasible,** including support for selecting batch or streaming behavior when the configured provider and model support it. **[Year 2]**

**3.5 Publish guidance on scaling AI-assisted curation workflows in OpenRefine,** including practical limits, provider-specific behavior, error recovery, cost/token considerations, and recommended execution patterns for larger tabular datasets. **[Year 2]**

**Success indicators:**

- Metrics and progress reporting are included in a public release and tested with at least **2 model providers**.
- Failed rows can be filtered, inspected, and rerun while preserving successful outputs.
- Retry and timeout handling are documented and validated through community testing.
- At least **one token-efficiency pattern** is implemented or documented for common curation workflows.
- At least **one provider-supported batch or streaming path** is evaluated or implemented.
- User guidance is published for running larger AI-assisted tabular workflows, including cost, token, and recovery considerations.

# Open Source for the Life Sciences (OS4LS)

Proposal Title: Auditable and Reproducible LLM Workflows in OpenRefine

Applicant Name: Martin Magdinier

| Items | Year 1 | Year 2 | Notes |
|---|---|---|---|
| Extension Development | $47,970.00 | $47,970.00 | [1] |
| Product, user research, documentation, training, and community adoption | $47,970.00 | $47,970.00 | [2] |
| Short-term contractor support | $11,375.00 | $11,375.00 | [3] |
| Travel, convenings, equipment, and dissemination | $5,125.00 | $5,125.00 | |
| Code for Science & Society fiscal sponsorship fee (10%) | $12,492.94 | $12,493.33 | |
| **TOTAL** | **$124,933.33** | **$124,933.33** | |
| | **Grant Total** | **$249,866.67** | |

[1] This budget line covers 0.25 FTE for Sunil Natraj the core AI Extension Developer

[2] This budget line covers 0.4 FTE for Martin Magdinier, product/community lead for user research, documentation, training, testing, release communication, and adoption support

[3] Short-term contractor support for design, testing, or specialized implementation needs.

Martin Magdinier and Sunil Natraj will contribute to the project as part-time contractors. Contractor rates are based on role, responsibility, seniority, and project-critical expertise.

The budget includes two primary labor lines. Sunil Natraj will serve as the AI extension developer, leading implementation and technical design for the LLM extension and related OpenRefine integration work. Martin Magdinier will lead product/community work, including user research, documentation, training, testing, release communication, and adoption support.

The budget also includes short-term contractor support for design, testing, or specialized skills that may be needed during implementation, as well as travel, convenings, equipment, and dissemination costs to support community validation and outreach. Code for Science & Society will provide fiscal sponsorship and project administration support through the 10% fiscal sponsorship fee.