# Technical Report: Biofilter System Status Update

## Summary

We initially began development on **Biofilter/LOKI version 3.0.0,** but work was paused upon identifying critical issues with **version 2.4.3** that required resolution before progressing with the new version. Our focus shifted to addressing these issues in version 2.4.3, prioritizing stability and user needs. The following report outlines the actions taken, current progress, challenges, and next steps.

### Key Actions Taken

**1. Refocusing Development on Version 2.4.3**

* The team was instructed to identify and document all problems in version 2.4.3 in the GitHub Issues section.
* Recovery of the 2.4.3 codebase was necessary as the repository had a single branch, Master, containing version 3.0.0 code.
* A new branch structure was created to organize and clarify the development:

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated | * **Main**: Contains the restored 2.4.3 code and is designate d as the future default branch. * **Development** (tagged 2.4.4): Branch for active work on version 2.4.3 improvements. * **dev-3.0.0**: Copy of the Master branch to isolate version 3.0.0 development. * Upon team approval, the current default branch (Master) will be removed, and Main will be set as the default to ensure users access the stable version (2.4.3). |

**2. Setup Improvements**

* Adjusted *`setup.py`* to include dependencies, allowing installation via `*pip install`.* This simplified execution by enabling global use of `*biofilter*` and `*loki`* commands.
* Plan to transition from `*setup.py`* to Poetry (https://python-poetry.org/) for better dependency management and a more transparent development process.

|  |  |
| --- | --- |
|  |  |

**3. Documentation Enhancements**

* Created a dedicated technical documentation area for developers, providing a quick onboarding guide.
* Added a *README.md* to the **Development** branch to explain its purpose and updates.
* Installed and configured tools like **Black**, **Coverage**, and **Pytest**. Development dependencies were excluded from *setup*.*py* but listed in a *requirements-dev.txt.*
* Plans to incorporate **Sphinx** (https://www.sphinx-doc.org/en/master/) for structured documentation in the coming weeks.

**4. Code Restructuring**

* Separated functionalities into distinct files:
  + Entry point file
  + Argument configuration file
  + Class and methods file with **mixins** for better organization
* Added detailed comments to explain methods, constants, and schemas used for generating temporary databases.
* Maintained current functionality while improving code clarity and maintainability.

A screenshot of a computer

Description automatically generated

**5. Integration of LOKI**

* Currently, the Biofilter system requires LOKI files to be embedded due to its packaging model.
* Kept this structure to avoid disrupting Biofilter’s functionality but recommend reviewing this approach in future updates to separate LOKI as an independent, installable package.

### Current Development Focus

**1. Unit Tests**

* Developing tests to validate Biofilter methods' functionality (not LOKI-specific data).
* Using **Coverage** to track statement-level testing, currently at **55% coverage with 220 tests**, including LOKI's embedded code (loki\_db.py only).
* Implementing a test-driven development (TDD) approach to ensure integrity before modifying any code.

A screenshot of a computer

Description automatically generated

In the past few weeks, we have added nearly 10,000 lines of code to the Biofilter project, primarily in the form of detailed comments on methods and unit tests. These additions aim to enhance the project’s manageability and readability, significantly reducing the onboarding time for future collaborators. By providing clear documentation and robust testing frameworks, we ensure that new team members can quickly understand the system’s structure and functionality, fostering smoother project development and maintenance.



The table highlights the progress in modularizing the code, increasing test coverage, and reorganizing functionalities, facilitating the project's maintenance and scalability.

A screenshot of a computer program

Description automatically generated**2. Functional and Issues Tests**

* Addressed two GitHub issues as functional test cases:
* Issue 1: Biofilter Group Annotation <https://github.com/RitchieLab/biofilter/issues/15>
* Issue 2: build 37 LOKI

<https://github.com/RitchieLab/LOKI/issues/16>

* These tests are located under a separate directory (`tests/ISSUE`) for isolation and can be discarded once the problems are fully resolved.
* Collaborating with the team to identify essential functional tests for key features like annotations, filters, models, installation, and LOKI integration.

### Challenges

**1. Codebase Complexity**

* The original Biofilter code was a single file with no comments, making it difficult to debug and enhance.
* While restructuring improved maintainability, further work is needed to ensure long-term project integrity.

**2. Embedded LOKI Files**

* LOKI files are hardcoded into the Biofilter system, limiting modularity and scalability. Separating LOKI into an independent package would resolve this issue.

### Next Steps

**1. Finalize Version 2.4.4**

* Complete unit and functional tests to validate stability.
* Resolve remaining GitHub issues reported by the team.
* Prepare for release using the LOKI database on the LPC environment.

**2. Improve Documentation**

* Implement Sphinx for developer and user documentation.
* Link functional tests to documentation as examples for users.

**3. Plan for Version 3.0.0**

* Review identified improvements for the LOKI database and Biofilter structure.
* Transition to modern development practices, including the use of ORM and modular package management.

This comprehensive approach ensures immediate issues with version 2.4.3 are resolved while laying the foundation for a robust version 3.0.0.