

Technical Report: Biofilter System Status Update

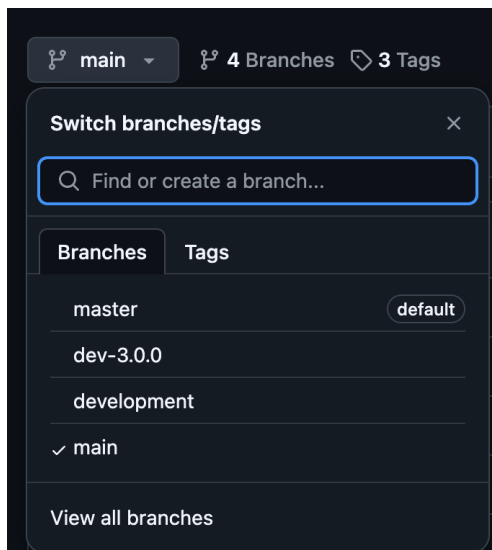
Summary

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

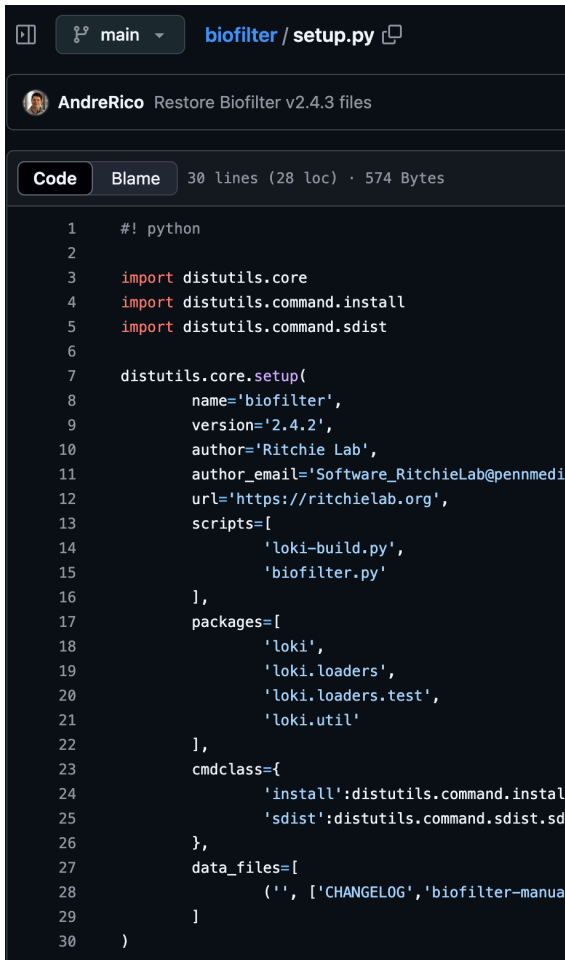
- The team was instructed to identify and document all problems in version 2.4.3 in the GitHub Issues section.
- Recovery of the 2.4.3 codebase was necessary as the repository had a single branch, Master, containing version 3.0.0 code.
- A new branch structure was created to organize and clarify the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



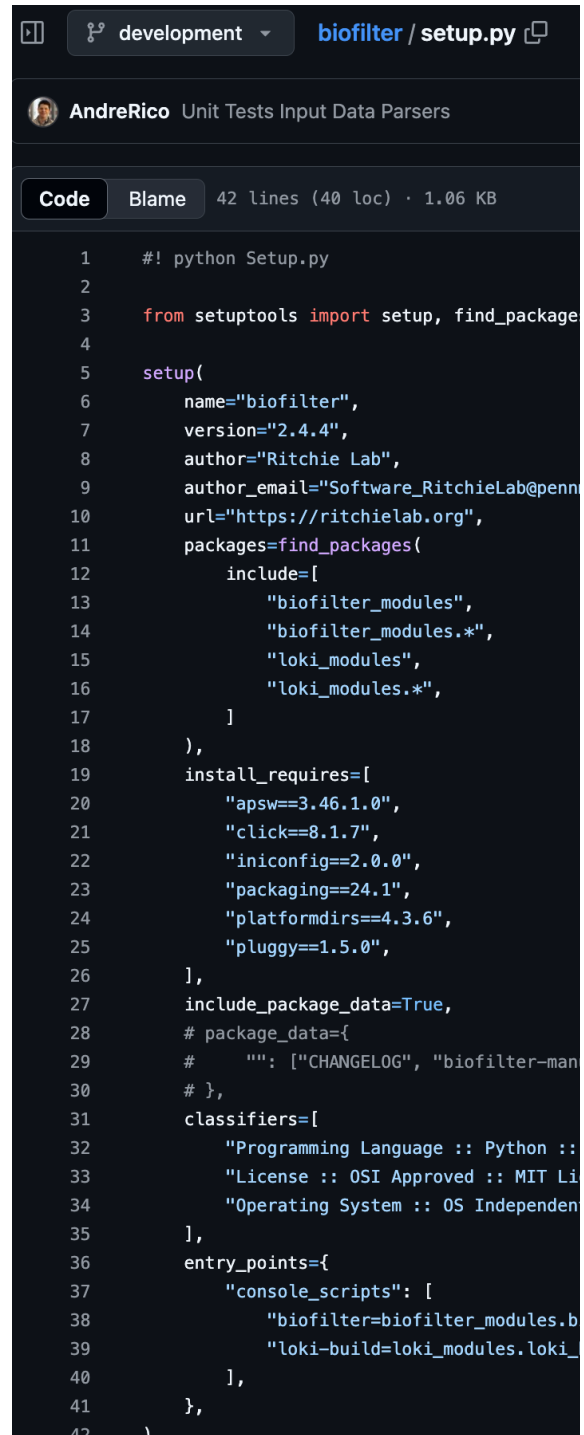
- **Main:** Contains the restored 2.4.3 code and is designated as the future default branch.
- **Development** (tagged 2.4.4): Branch for active work on version 2.4.3 improvements.
- **dev-3.0.0:** Copy of the Master branch to isolate version 3.0.0 development.
- Upon team approval, the current default 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.



```
1  #! python
2
3  import distutils.core
4  import distutils.command.install
5  import distutils.command.sdist
6
7  distutils.core.setup(
8      name='biofilter',
9      version='2.4.2',
10     author='Ritchie Lab',
11     author_email='Software_RitchieLab@pennmedicine.edu',
12     url='https://ritchielab.org',
13     scripts=[
14         'loki-build.py',
15         'biofilter.py'
16     ],
17     packages=[
18         'loki',
19         'loki.loaders',
20         'loki.loaders.test',
21         'loki.util'
22     ],
23     cmdclass={
24         'install':distutils.command.install.install,
25         'sdist':distutils.command.sdist.sdist
26     },
27     data_files=[
28         ('', ['CHANGELOG', 'biofilter-manual'])
29     ]
30 )
```



```
1  #! python Setup.py
2
3  from setuptools import setup, find_packages
4
5  setup(
6      name="biofilter",
7      version="2.4.4",
8      author="Ritchie Lab",
9      author_email="Software_RitchieLab@pennmedicine.edu",
10     url="https://ritchielab.org",
11     packages=find_packages(
12         include=[
13             "biofilter_modules",
14             "biofilter_modules.*",
15             "loki_modules",
16             "loki_modules.*",
17         ]
18     ),
19     install_requires=[
20         "apsw==3.46.1.0",
21         "click==8.1.7",
22         "iniconfig==2.0.0",
23         "packaging==24.1",
24         "platformdirs==4.3.6",
25         "pluggy==1.5.0",
26     ],
27     include_package_data=True,
28     # package_data={
29     #     "": ["CHANGELOG", "biofilter-manual"]
30     # },
31     classifiers=[
32         "Programming Language :: Python :: 3",
33         "License :: OSI Approved :: MIT License",
34         "Operating System :: OS Independent",
35     ],
36     entry_points={
37         "console_scripts": [
38             "biofilter=biofilter_modules.biofilter",
39             "loki-build=loki_modules.loki_build",
40         ],
41     },
42 )
```

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:
 - o Entry point file
 - o Argument configuration file
 - o 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.

Version 2.4.3

biofilter.py

Version 2.4.4

biofilter/biofilter_modules/

biofilter.py

argparse_config.py

arg_utils.py

biofilter_class.py

biofilter/biofilter_modules/mixins/

database_management_mixin.py

filter_annot_model_mixin.py

input_data_parsers_mixin.py

input_gene_mixin.py

input_group_mixin.py

input_locus_position_mixin.py

input_region_mixin.py

input_snp_mixin.py

input_source_mixin.py

internal_query_build_mixin.py

paris_mixin.py

logger_mixin.py

schema.py

user_knowledge_input_mixin.py

loki_data_retrieval_mixin.py

user_knowledge_retrieval_mixin.py

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.

Coverage report: 55%

Files Functions Classes

coverage.py v7.6.4, created at 2024-11-19 15:32 -0500

File ▲	statements	missing	excluded	coverage
biofilter_modules/__init__.py	8	0	0	100%
biofilter_modules/arg_utils.py	52	1	0	98%
biofilter_modules/argparse_config.py	148	0	0	100%
biofilter_modules/biofilter_class.py	57	2	0	96%
biofilter_modules/mixins/__init__.py	17	0	0	100%
biofilter_modules/mixins/database_management_mixin.py	38	3	0	92%
biofilter_modules/mixins/filter_annot_model_mixin.py	304	45	0	85%
biofilter_modules/mixins/input_data_parsers_mixin.py	246	0	0	100%
biofilter_modules/mixins/input_gene_mixin.py	67	0	0	100%
biofilter_modules/mixins/input_group_mixin.py	65	0	0	100%
biofilter_modules/mixins/input_locus_position_mixin.py	33	0	0	100%
biofilter_modules/mixins/input_region_mixin.py	33	0	0	100%
biofilter_modules/mixins/input_snp_mixin.py	27	0	0	100%
biofilter_modules/mixins/input_source_mixin.py	37	0	0	100%
biofilter_modules/mixins/internal_query_builder_mixin.py	322	270	0	16%
biofilter_modules/mixins/logger_mixin.py	49	0	0	100%
biofilter_modules/mixins/loki_data_retrieval_mixin.py	43	0	0	100%
biofilter_modules/mixins/paris_mixin.py	215	209	0	3%
biofilter_modules/mixins/schema.py	2	0	0	100%
biofilter_modules/mixins/user_knowledge_input_mixin.py	48	0	0	100%
biofilter_modules/mixins/user_knowledge_retrieval_mixin.py	15	0	0	100%
loki_modules/__init__.py	1	0	0	100%
loki_modules/loki_db.py	724	607	1	16%
Total	2551	1137	1	55%

coverage.py v7.6.4, created at 2024-11-19 15:32 -0500

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

Technical Report: Biofilter System Status Update

testing frameworks, we ensure that new team members can quickly understand the system's structure and functionality, fostering smoother project development and maintenance.

File	2.4.3					2.4.4				
	rows	class	functions	tests	rows_test	rows	class	functions	tests	rows_test
biofilter.py	4,180	4	89	0	0	1,020	0	5		
biofilter_class.py						195	2	4	5	116
argparse_config.py						1,066	0	6	27	464
arg_utils.py						126	2	4	5	117
database_management_mixin.py						241	1	6	8	135
filter_annot_model_mixin.py						953	1	7	25	749
input_data_parsers_mixin.py						1,014	1	15	64	930
input_gene_mixin.py						326	1	4	8	287
input_group_mixin.py						315	1	4	8	291
input_locus_position_mixin.py						159	1	2	4	114
input_region_mixin.py						155	1	2	4	116
input_snp_mixin.py						154	1	2	4	170
input_source_mixin.py						148	1	2	4	109
internal_query_builder_mixin.py						1,901	1	5	13	213
logger_mixin.py						101	1	7	16	152
loki_data_retrieval_mixin.py						274	1	7	13	113
paris_mixin.py						624	1	4		
schema.py						241	1	0	3	71
user_knowledge_input_mixin.py						291	1	4	6	290
user_knowledge_retrieval_mixin.py						87	1	2	3	73
Total	4,180	4	89	0	0	9,391	20	92	220	4,510

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

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.

```
▼ tests
  > __pycache__
  > data
  ▼ funcional
  ▼ issues
    ▼ b15_biofilter_group_annotation
      > __pycache__
      > data-in
      > data-out
      ⚡ doc.md
      ⚡ test_rasika.py
      ☑ todo.md
    ▼ l16_build_37_loki
      > __pycache__
      ≡ input_filename
      ⚡ test_function.py
  ▼ units
    > __pycache__
    ⚡ est_internal_query_builder_mixin.py
    ⚡ test_arg_utils.py
    ⚡ test_argparse_config.py
    ⚡ test_biofilter_class.py
    ⚡ test_database_management_mixin.py
    ⚡ test_filter_annot_model_mixin_1.py
    ⚡ test_filter_annot_model_mixin_2.py
    ⚡ test_input_data_parsers_mixin.py
    ⚡ test_input_gene_mixin.py
    ⚡ test_input_group_mixin.py
    ⚡ test_input_locus_position_mixin.py
    ⚡ test_input_region_mixin.py
    ⚡ test_input_snp_mixin.py
    ⚡ test_input_source_mixin.py
    ⚡ test_internal_query_builder_mixin.py
    ⚡ test_logger_mixin.py
    ⚡ test_loki_data_retrieval_mixin.py
    ⚡ test_schema.py
    ⚡ test_user_knowledge_retrieval_mixin.py
    ⚡ test_user_knowledge_input_mixin.py
```

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.