NPLinker updates

Cunliang Geng & Giulia Crocioni

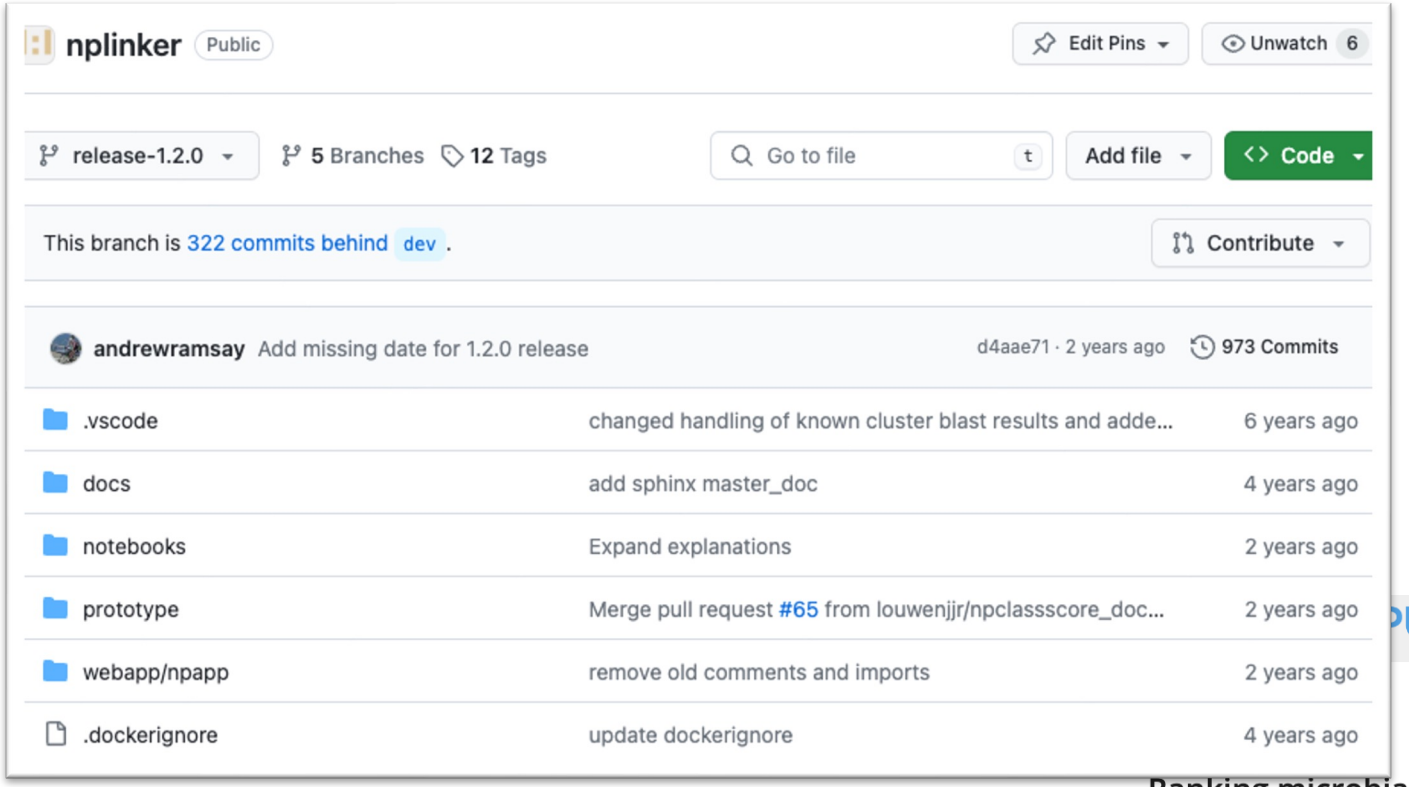
2024-02-09

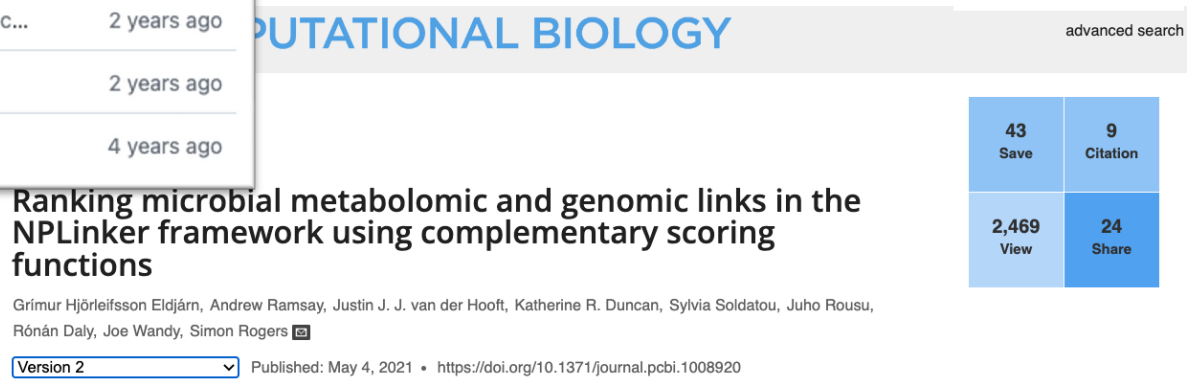
netherlands eScience center

netherlands eScience center

Starting point

A public **software repo** with a published paper





https://doi.org/10.1371/journal.pcbi.1008920

netherlands eScience center

Challenges
we have faced from the beginning

Creating a Scalable NPLinker Framework with Interactive Visualization Module

- Code reusability challenge
- Need a more user-friendly web app

Redesign & Refactor (Rewrite)

netherlands eScience center

The principles of refactoring

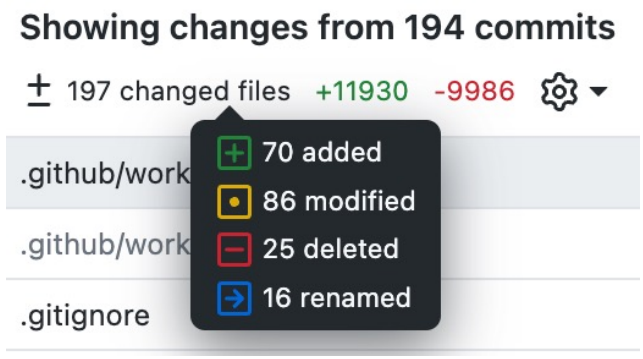
- Easy to install & run 👉 packing code to a package/image, having tutorials and documentations, …
- Easy to read 👉 clear structure of codebase, proper formatting of code, …
- Easy to understand 👉 meaningful names, static typing, good comments, …
- Easy to change 👉 well-designed architecture (abstractions), …
- Modular code 👉 use function and class to organise code, do one thing per function/class,…
- Correct code 👉 unit test, integration test, user test…
- Don’t repeat yourself 👉 eliminating duplicated code
- Don’t reinvent the wheel 👉 taking advantage of existing libraries and packages
- …

netherlands eScience center

What we have done

- Split the software repo into two repos: nplinker and webapp
- Created python package for nplinker (v1.3.2) and simplified the installation
- Redesigned the architecture of nplinker into two main parts: data preparation and scoring
- Ongoing refactoring of the data preparation part
- Redesigned the UI of webapp and explored tech solutions
- Got an internal grant (0.5fte) to improve the sustainability of NPLinker software

Code changes in 2023



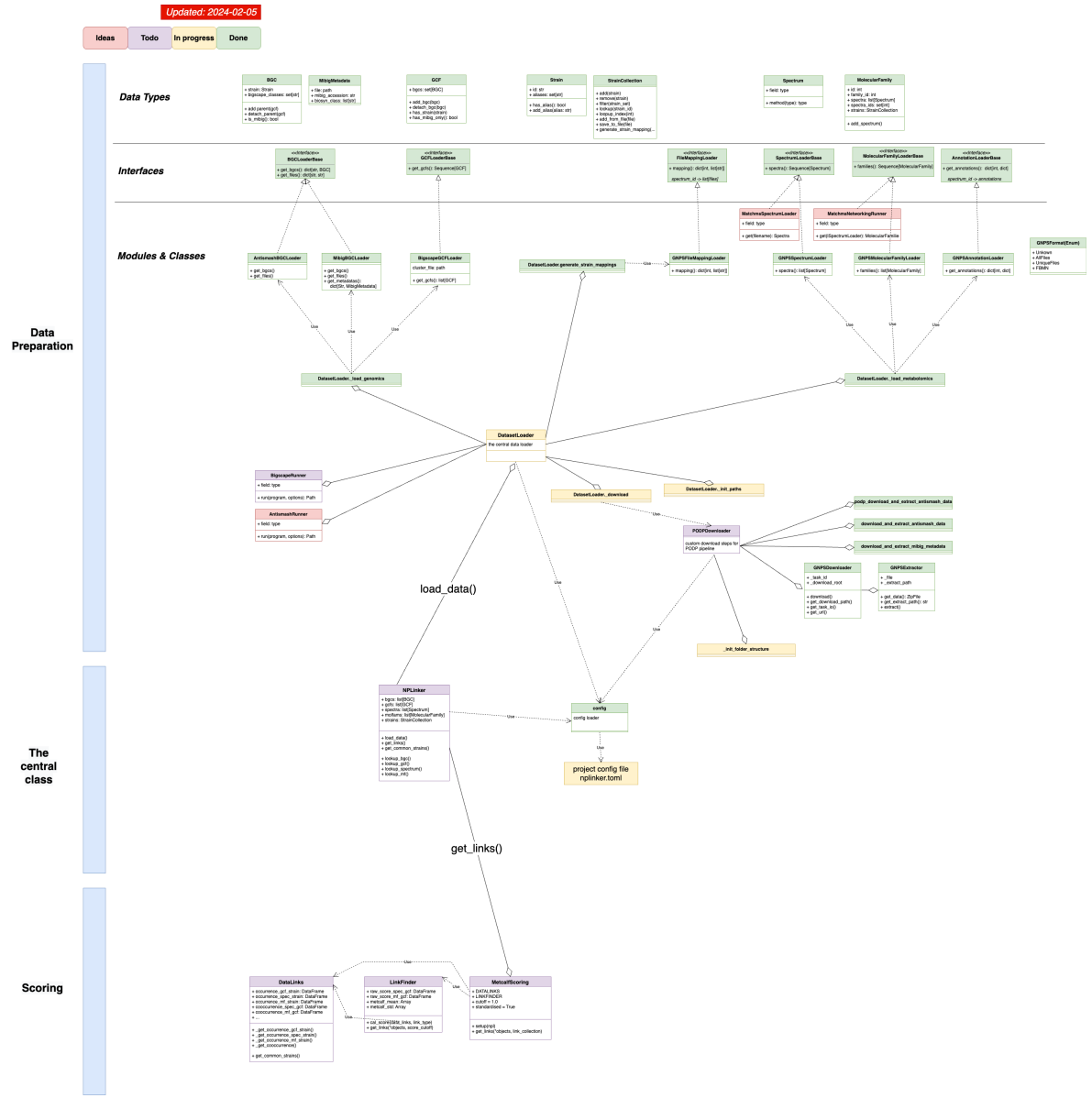
Research Software Engineer
Dr. Cunliang Geng

Research Software Engineer
Giulia Crocioni, MSc

PhD candidate
Helge Hecht, MSc

netherlands eScience center

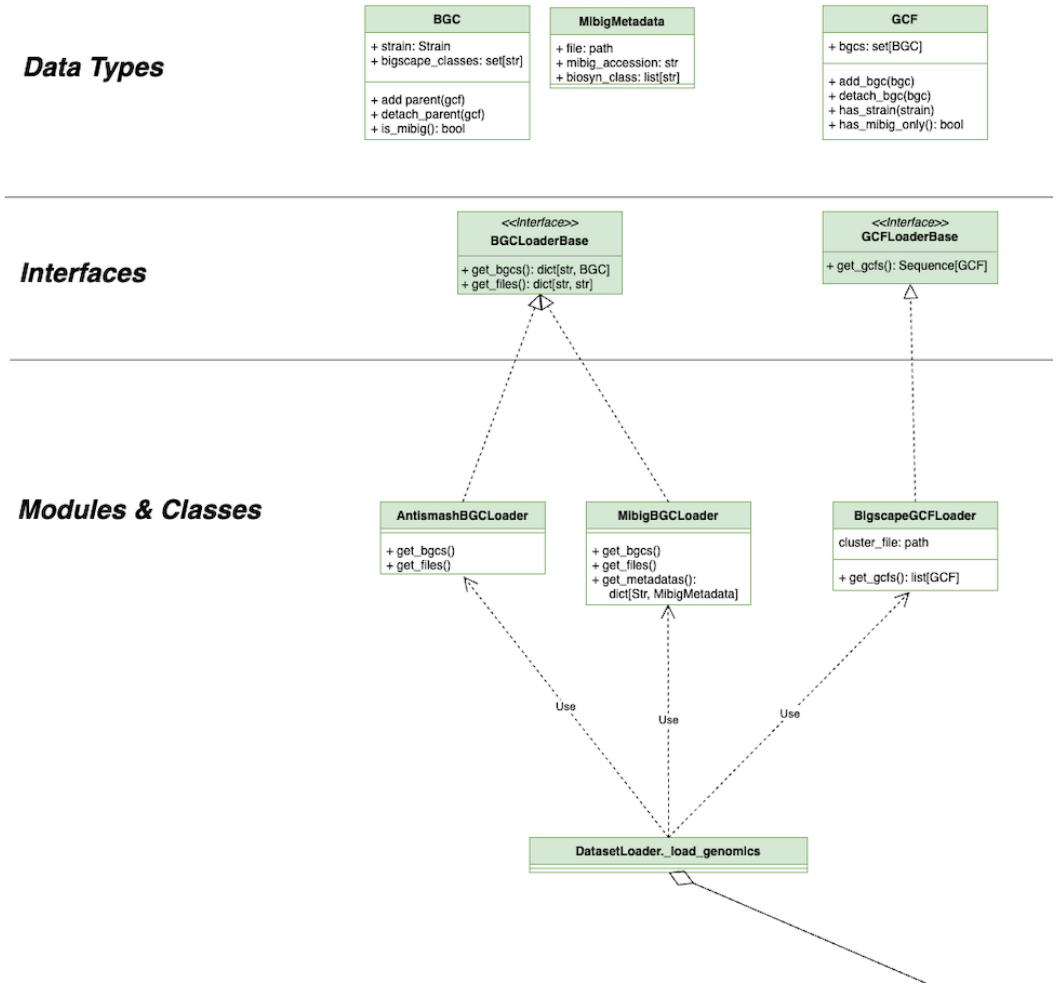
Redesigned NPLinker architecture



Redesigned NPLinker architecture

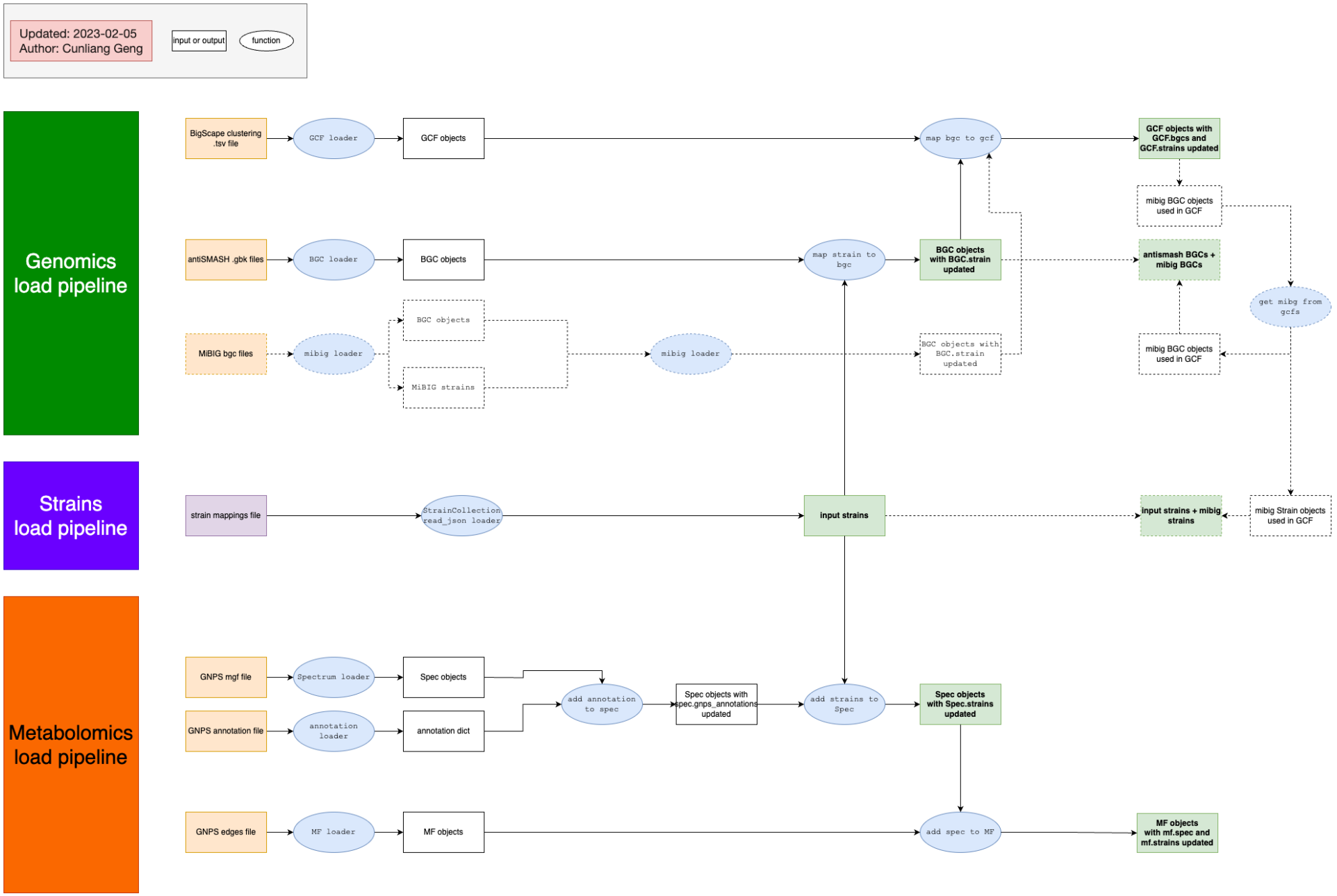
* Not all functions/classes displayed

Data Preparation



netherlands eScience center

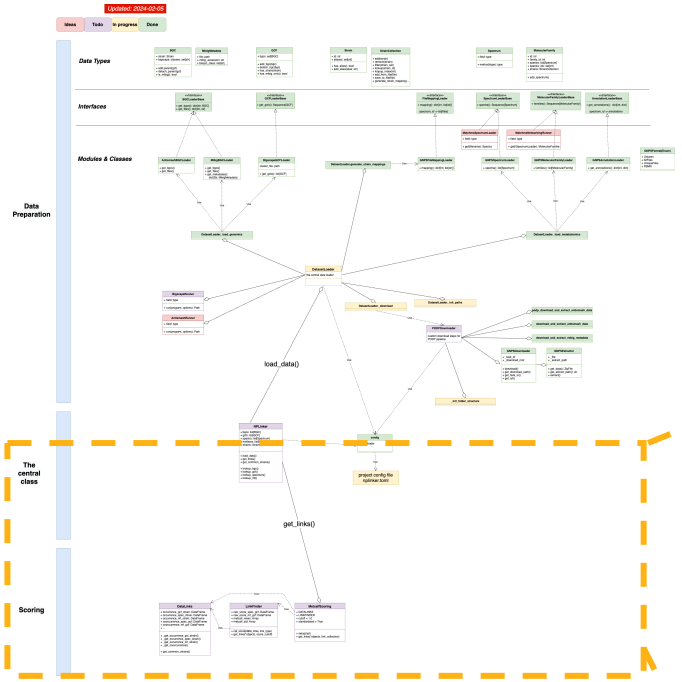
Data loading pipeline



To be refactored

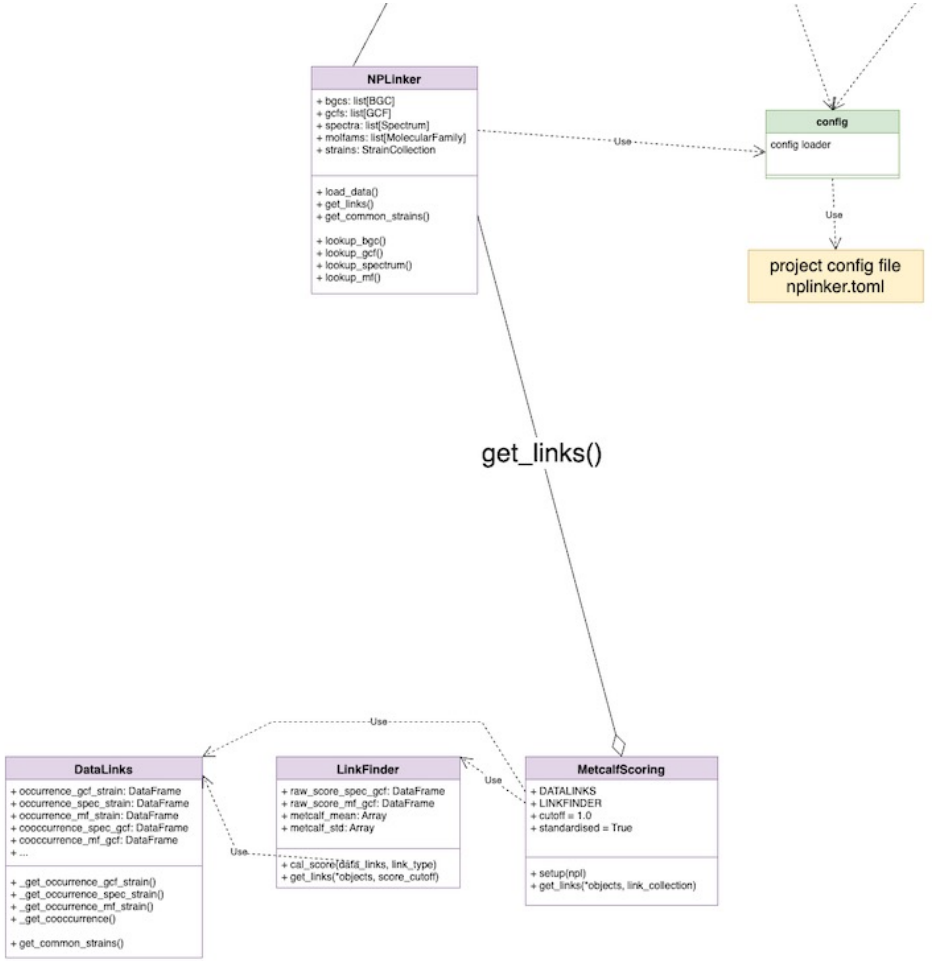
Redesigned NPLinker architecture

* Not all functions/classes displayed



The central class

Scoring



netherlands eScience center

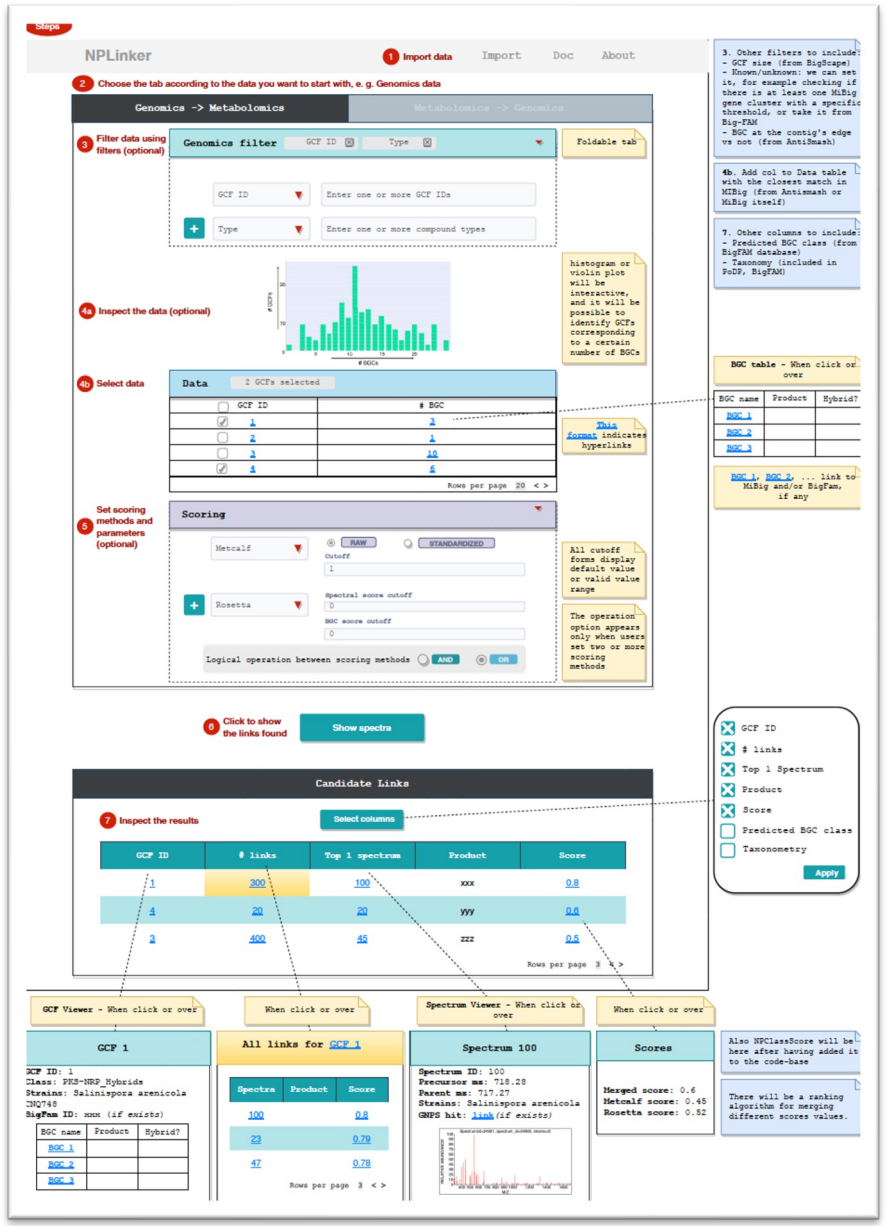
Some highlights of the refactoring

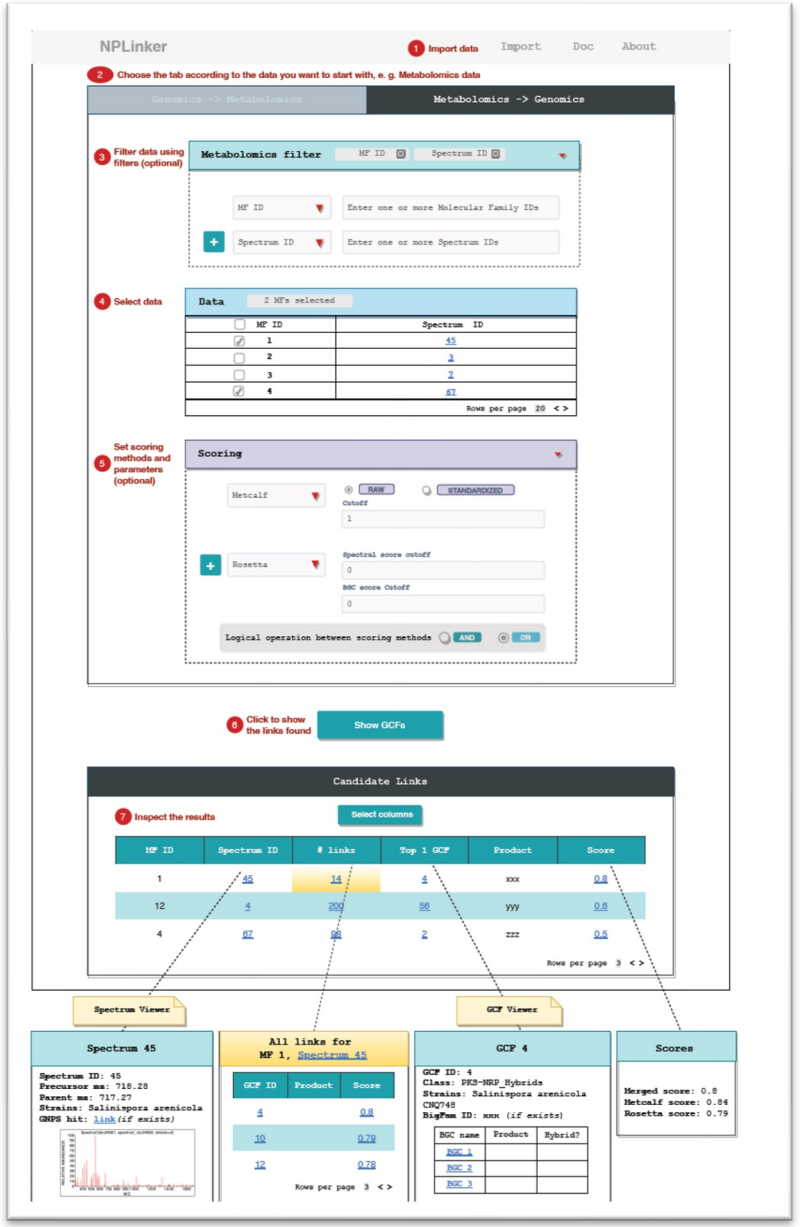
- Restructured the codebase and uniformed the formatting of code with ruff
- Added docstrings
- Added python static typing
- Added unit tests and parallel testing
- Added/updated code for data objects, downloaders, extractors and loaders
- Updated the logics of loading GEN and MET data, making them consistent with each other
- Updated the logics of loading strain mappings, making it decoupled from loading of GEN/MET data
- Changed the output file format (csv/tsv) to JSON
- Added schema for input/output JSON files and added schema validators
- Updated the loading of config with config manager Dynaconf
- …

netherlands eScience center

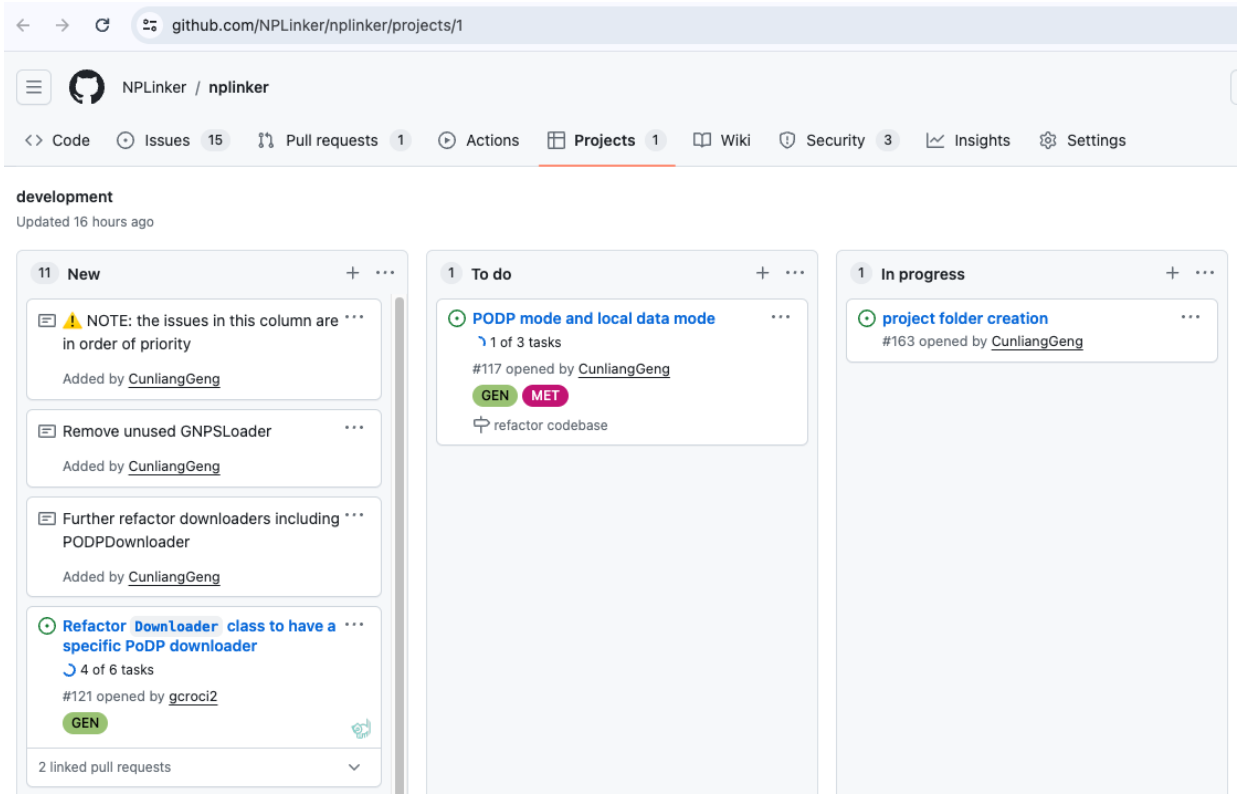
To be developed

Redesigned Webapp UI





Future work



Kanban
https://github.com/NPLinker/nplinker/projects/1

Future work

- Enable the pipeline of loading user's local data (Local mode)
- Update template of config file and the creation of project directory structure
- Write tutorials for user testing on loading data
- Develop webapp
- Refactor scoring part
- Further user testing

netherlands eScience center

Let’s stay in touch

www.eScienceCenter.nl

c.geng@esciencecenter.nl

@CunliangGeng