Developer Notes for RatBAT

Rat Behavioral Analysis Tool

Table of Contents

[1 Document Information 3](#_gg1pkt7igsma)

[1.1 Revision History 3](#_vglc1l334x09)

[1.2 Project Personnel 3](#_161nszb3a7eb)

[1.3 Document Structure 3](#_caef986prhh4)

[2 Overview 4](#_hwokjp8lx9f)

[2.1 Purpose 4](#_wq95ciu4gxa2)

[2.2 Background 4](#_hg45r717xj4y)

[2.3 Current Status 4](#_ljq68zat28uo)

[3 Developer Notes 5](#_map1qqutejv3)

[3.1 rBat Package 5](#_jn4rwbcumk44)

[3.1.1 Summary Measures 5](#_3hk5wgnsefed)

[3.1.2 Preprocessing 7](#_boujx27qz6tk)

[4 Future Development 8](#_rgnlaz4ld8ak)

[4.1 rBat Package 8](#_nh3ewq1q6jf8)

[4.1.1 Summary Measures 8](#_gy88q8xrdu5f)

[4.1.2 Preprocessing 9](#_3scrjxkfn38d)

[4.2 Web platform 9](#_kupa6bn1zqxq)

[5 Miscellaneous Notes 10](#_wglse1qg3qni)

# 

# 1 Document Information

## 1.1 Revision History

| Date | Version | Notes |
| --- | --- | --- |
| 02/05/2025 | 0 | Original draft of document |

## 1.2 Project Personnel

| Name | Email | Role |
| --- | --- | --- |
| Brandon Carrasco | brandonc.edu@gmail.com | Former Developer |
| Daniel Locke | danielocke03@gmail.com | Former Developer |
| Jamie Wong | jamiewong198@gmail.com | Former Developer |
| Inoday Yadav | yadavinoday@gmail.com | Former Developer |
| Anna Dvorkin-Gheva | [dvorkin@mcmaster.ca](mailto:dvorkin@mcmaster.ca) | Client |
| Henry Szechtman | [szechtma@mcmaster.ca](mailto:szechtma@mcmaster.ca) | Client |

## 1.3 Document Structure

**Overview** contains informational content about RatBAT, including its purpose, status, etc. **Developer Notes** contains information on the architecture of the code and how to extend or alter it. **Future Development** contains information about features that should be implemented, as well as further areas of research that might produce new ideas that could be integrated into RatBAT. **Miscellaneous Notes** contain any information that doesn’t fit above. **Appendix**, should you need to add such a section in the future. includes any in-depth (think raw code) information that belongs to the above categories, but would disrupt the flow of the document if placed within their appropriate sections.

# 2 Overview

## 2.1 Purpose

This project aims to leverage and expand upon the existing dataset generated from the Quinpirole Sensitization Rat Model of OCD experiments, which involved tracking rat movements in an open field and collecting x, y, t time-spatial data. Such experiments had run for several decades in the laboratory of Dr. Szechtman at McMaster University. Recently, the collected data had been annotated and deposited in a public repository, the FRDR, with the expectation that the data will be utilized by others in future research. However, though there exists a substantial amount of raw data, there is currently no easy and straightforward method to access, process, and analyse this stored data. The purpose of the present project is to develop a robust and accessible open platform for researchers to preprocess data, compute summary measures, analyse, and gather rat behaviour data from the FRDR repository. This platform is expected to be valuable for many projects, including creating a talking animal model of OCD by translating the x, y, t coordinate data of rat locomotion into an audio narrative.

## 2.2 Background

This project was originally developed as a capstone project, with Dr. Anna Dvorkin-Gheva and Dr. Henry Szechtman as the project’s supervisors. Initial pieces of documentation were created in October 2024. Development began in November 2024 and continued until the end of the capstone course in April 2025.

## 2.3 Current Status

[Web application](https://ratbat.mcmaster.ca/) is currently hosted and served on RHPCS’ servers at McMaster University, along with the database. RatBAT specific code & documentation is hosted on [Github](https://github.com/brandonc-edu/RatBAT). Test branch can be used for testing. RHPCS is used for the production branch (though you can use Main instead; just notify the RHPCS team).

The summary measures and preprocessing algorithms and other associated code is hosted in its own separate [repo](https://github.com/brandonc-edu/rBat). This code is packaged as *rBat*, available on [PyPI](https://pypi.org/project/rBat/) for download.

# 3 Developer Notes

## 3.1 rBat Package

### 3.1.1 Summary Measures

**Architecture**

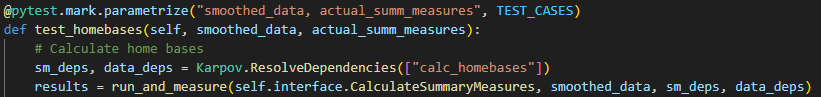
*FunctionalSM.py* contains all summary measure and data calculation algorithms, as well as dependency tables (necessary for the *Commander* class). *FieldSM.py* contains the *Environment* class and several helper functions and constants which are necessary for the calculation of the summary measures. *DependenciesSM.py* contains the *Karpov* class and its methods, which are used to ensure all data and summary measure dependencies are handled. *CommanderSM.py* contains the *Commander* class, used for safe and efficient calculations of summary measures.

The intended use is as follows:

1. A list of strings representing reference ids to summary measures are passed to *Karpov’s* *ResolveDependencies* method. This will return two lists of strings: list of reference ids of summary measures that satisfy all dependent summary measures that were originally passed, and a list of reference ids of data calculations that are necessary to calculate the summary measures.
   1. Reference ids refer to either:
      1. Developer names for summary measures
      2. Developer names for data calculations
   2. Data calculations refers to calculations on the time-spatial data that are commonly shared.
      1. By pre-calculating them prior to calculating summary measures, this allows all summary measures that use those calculations to share them, reducing extra calculations.
2. The *Commander* will be instantiated with the string corresponding to which environment the time-spatial data is associated with. The *Commander*’s *CalculateSummaryMeasures* will receive the time-spatial data and both lists of reference ids calculated by *Karpov*.
   1. This will return a dictionary containing the results of all summary measure calculations, with the summary measure’s reference ids acting as the keys to their associated results.
3. The *Commander* will clip all coordinates to the min and max values of 20 & 180. Then the *Commander* will calculate all data calculations. Finally, the *Commander* will go through the summary measures list and calculate all summary measures.
   1. Because we used *Karpov*, all dependent summary measures will have their prerequisite summary measures calculated before they are meant to be calculated.

**Extending Summary Measures & Data Calculations**

Adding new summary measures involves the following process:

1. Come up with a summary measure reference id
   1. Starts with *calc\_*
   2. Used by *Karpov, Commander*, and the RatBAT backend to more easily call functions.
2. Create function skeleton:
   1. Name, doc string, etc.
   2. Every summary measure algorithm takes in the same arguments. See *FunctionalSM.py* for examples.
   3. Function must start off with *HandleMissingInputs* being called
   4. Function must return one or more values.
3. Create algorithm (flesh out the function)
4. Add reference id to mappings in *FunctionalSM.py*
   1. *SM\_MAPPING*: ref id key to function name string; used to actually call the function in *Commander*
   2. (OPTIONAL) *SM\_DEPENDENCIES*: ref id key to list of other ref ids for summary measures. The list of ref ids will be the ref ids of summary measures that are required for your new summary measure to be calculated.
   3. (OPTIONAL) *DATA\_DEPENDENCIES:* ref id to list of other ref ids for data calculations. The list of ref ids will be the ref ids of data calculations that are required for your new summary measure to be calculated.
5. (OPTIONAL) Create test functions for your new summary measure
   1. Assign test id to ref id for your new summary measure in *conftest.py*
      1. *test\_* in place of *calc\_* for the test id.
      2. Used for recording time results.
   2. Create test function under *test\_sm\_package.py* (testing accuracy of SM results)
      1. Follow the below skeleton.
      2. 
      3. Flesh out the assertion of the results, comparing *actual\_summ\_measures* to *results*.

Adding new data calculations is much the same process:

1. Come up with a data calc reference id
   1. No prefix necessary
2. Create function skeleton:
   1. Name, doc string, etc.
   2. Every data calculation takes in time-spatial data and an *Environment*. See *FunctionalSM.py* for examples
   3. Function must return one or more values
3. Create algorithm
4. Add reference id to *DATA\_MAPPING* in *FunctionalSM.py*, with reference id mapping to the name of the function to be called.

### 3.1.2 Preprocessing

**Architecture**

***Preprocessor.py*** contains the *Preprocessor* class, responsible for running the full preprocessing algorithms. s*egmentation.py* contains the necessary files for the EM algorithm to be run and ***smoothing.py*** contains the algorithms for LOWESS and RRM.

The intended use is as follows:

1. The preprocessor class is instantiated with (optionally) a dictionary of dictionaries, each corresponding to and containing the parameters for each algorithm (LOWESS, RRM, EM).
   1. If this dictionary is passed, the preprocessor function will prepare the algorithms with the appropriate parameters. Otherwise, the *Preprocessor* will set the algorithms to their default parameters (as specified in *DEFAULT\_PARAMS*).
2. The *Preprocessor*’s *preprocess\_data* method takes in raw time-spatial data ([frame, x-coord, y-coord]) and preprocesses the data, running it through all three algorithms.
   1. This returns a smoothed and segmented data of the form: [frame, x-coord, y-coord, velocity, segment\_type]
      1. Where segment\_type is the type of movement episode the specimen is a part of during that array. By default (and to work with summary measures), the only possible values are 0 or 1 — lingering or progression.

**NOTE:** If the number of movement types do not equal 2, then the summary measures will not work properly. If only one, it’ll return very strange and useless results. If more than two, it’ll cause a runtime error!

# 4 Future Development

## 4.1 rBat Package

### 4.1.1 Summary Measures

| **To be Implemented** | |
| --- | --- |
| Summary Measures | * Path Stereotypy, Locomotion, and 2SDE Summary Measures   + Plus associated testing functions * *HandleMissingInputs* function should automatically calculate missing required summary measures instead of just raising an error. * Create test functions for all bout summary measures. * Better, more complete documentation. |
| Interface | * Proper error messages should be propagated up from *Karpov* and *Commander*. * Better, more complete documentation. |
| **Areas of Research & Suggestions** | |
| Summary Measures | * Record improvements in testing accuracy (compared to the original summary measure results) to ensure any alterations to the algorithms don’t result in regression of performance. * Optimize algorithms, in particular the calculation of bouts, if possible. * Reduce error in summary measure results. In other words, try to piece together the original algorithm for the summary measures (in the case that the clients wish to use the original algorithms). * Use better test metrics for testing the accuracy of the summary measure algorithms. * P2 & P3 requirements in the SRS document on the Github Repo. |
| Interface | * Try and see if there’s any way to prevent circular dependency loops (i.e., SM 1 is dependent on SM 2, SM 2 is dependent on SM 3, SM 3 is dependent on SM 1) from forming in the ordering dependency functionality in *Karpov*. * Optimize the way that dependencies are added and reordered in *Karpov*. Not critical; total number of possible summary measures are very little. |

### 4.1.2 Preprocessing

| **To be Implemented** | |
| --- | --- |
| * Better, more complete documentation. * Add manual interface for segmentation *(*[*see here for reference*](https://www.tau.ac.il/~ilan99/see/help/)*)* to give users control over segmentation thresholds so that they can finetune movement classification for more accurate results. | |
| **Areas of Research & Suggestions** | |
| * Finetune default preprocessing parameters for all algorithms such that the time it takes to preprocess a raw data file is as low as possible while maintaining the smallest chance of failure (in the EM algorithm). | |

## 4.2 Web platform

| **Extending current capabilities** | |
| --- | --- |
| Summary Measures | * For incorporating new summary measures to the frontend navigate to *frontend/src/pages/ComputeSummaryMeasures.jsx* and update the summary measures display names, tool tips, and definitions objects (dictionaries) accordingly. * If a new summary measures has multiple return values that need to be formatted update the *const handleDownloadSelected* block of code for updating the download and the *<table className="result-table">* block of code for updating the table accordingly. Make these changes to *frontend/src/pages/CompileDataPage.jsx* as well. Update the info sheet in *frontend/public/info\_sheet.xlsx* for the new summary measures. * Make it so that trials that are a part of Q21, Q22, and Q23 use the Q20s environment specified in rBat package. * Prevent the user from running summary measures on the circular environment associated with trials a part of Q17. |
| FRDR Query | * FRDRQuery consists of the main FRDRQuery.jsx and 3 other jsx files in the components folder. It pulls files from the config folder located in the src. |
| **Areas of Research & Suggestions** | |
| Frontend | * Add tooltips for users to understand how our web app is supposed to be used. * Add functionality to completely lock out all next step pages until the user fulfills a set of requirements on their current page that are needed for the next page to be usable. Add more guidelines/tips to help the user meet the requirements to proceed to the next steps. * UI/UX design improvements (theme, navigation tabs, result tables, background etc.) * FRDRQuery page data viewer is lacking intractability (sorting by meta variable etc). |
| Other | * P2 and P3 functional requirements in documentation/Software Requirements Specification - Group 8.pdf |

# 5 Miscellaneous Notes

Documentation, especially for the summary measures and how they are meant to be calculated, can be provided and explained by Dr. Anna Dvorkin-Gheva and Dr. Henry Szechtman. Please contact them for scientific papers and explanations.