Code Review Checklist

Software Engineering for Scientists

Reviewer:

Files reviewed:

Summary of comments:

Modularity

* Adequate separation of concerns
* All functions in a file are related or are supplementary
* Functions don’t do too much
* No modules are too tightly coupled

Comments:

Testability:

* There are unit tests for functions that need testing
* Unit tests that exist test all use cases
* Modules are written in a testable way

Comments:

Performance:

* Excessive loops are avoided
* Extraneous data structures are avoided
* Temporary data structures (if present) are deleted immediately when not needed
* Given the inputs and outputs of the function, you would write the function, in the same way, to do the same thing (you agree that the way the function is written is fast enough and light enough on space)

Comments:

Documentation

* Function documentation clearly states what a function does and if necessary has examples
* Input parameters and output values have types stated in the function definition
* Parameters have a clear description of what they are
* Inline comments are useful, not in the way, and clearly describe a line or block of code

Comments:

Legibility and Style

* Consistent spacing of parenthesis, operators, comma-separated values
* Consistent pythonic (values\_with\_underscores) naming of functions and variables
* CamelCased names of data structures or types
* Lines don’t go off of the screen
* Function name describes what the function does

Comments: