

arplsbaseline Package Test Plan

Jessica Sam

Testing Instructions

This test plan should cover most, if not all different cases that need to be tested. The baseline function is essentially the main function in the package and this requires the most testing, especially for input validation.

Package Structure

In my package I have a **baseline** function, which should create an object of class **Spectrum**. **Spectrum** objects have callable elements and methods. Extractable elements are **baseline**, **x**, **y** and **corrected**, which should all be vectors of the same length. Methods that exist are **print**, **summary**, **plot** and **as.data.frame** which print and summary should print output to the console, plot should create a visualisation and as.data.frame will coerce the spectrum into a dataframe.

Running Unit Tests

To run all the unit tests at the same time with the assumption that the package **testthat** is installed in to the user's R packages, as well as the **arplsbaseline** package, then the following command can be called:

```
testthat::test_package("arplsbaseline", "tap")
```

The results of this command should show 25 comments about all the expectations which have passed, and for what reason. An explanation of these tests are below.

Components Tested

The data used for the baseline testing are embedded into the test-baseline.R file.

The expectations numbered **1-11** are made regarding input validation of the **baseline function**.

Baseline Input Validation

1. Incorrect data format throws an error

This test checks whether the function is properly throwing the error, “**Data must be in the form of a dataframe**”, whenever the function is given data that isn't a dataframe. There should be three expectations for this error, with NA, a vector [c(2,3)] and a single value [2] tested as inputs to the baseline function.

2. NA values have been removed

This test checks that a dataframe that contains a few NA values, but is otherwise reasonable will successfully compute a spectrum. It should be expected that both x and y vectors that are extracted from the spectrum, no longer contain NA values and that the updated x and y are shorter than the original x and y. There should be four expectations to pass.

3. The function throws an error if data is not numeric

This test checks that if the function has a dataframe, but one of the columns is not numeric, that the error “**Both columns of dataframe must be numeric**” is thrown. A dataframe with a character element is used to test this condition.

4. The function throws an error if the data does not have enough elements

This test checks that the error “**There must be at least 10 non-NA elements in each column**” is thrown, when each column of the dataframe doesn’t have enough data to compute an inverse matrix on. This test uses a dataframe which has 9 elements in each column.

5. Lambda incorrectly entered, gives a message is given that the default value is used

The test checks the message “**Lambda must be a single numeric value between 1 and 1e10, default lambda of 1e4 will now be used**” is thrown when lambda is entered incorrectly, where a vector c(1,2), or NA were used to test. This test has two expectations.

Baseline Edge Cases

The expectations numbered **12-14** are made regarding edge cases of the **baseline** function.

6. Empty baseline function throws an error

This test checks that the function throws an error, “**Data argument cannot be empty**”, when **baseline()** is called with no inputs.

7. Function throws an error if signals are all 0

This test checks that the function throws an error, “**Data cannot be all 0**”, when all the y values, also known as signal intensities are all 0.

8. Function throws an error if there are not enough wavenumbers

This test checks that the function throws an error, “**Samples need more unique wavenumbers**”, if the x, or wavenumbers are not very unique. In this, it was tested that all the wavenumbers are all the same value [0].

Spectrum Components

The expectations numbered **15-25** are made regarding correct **Spectrum** functionality and structure, given that the baseline function was successful. A spectrum made from the strawberry dataset within the package with a lambda of 1e7 was used for these tests.

9. The baseline function returns a Spectrum object

This test confirms that the object is indeed of class Spectrum.

10. Each element from spectrum is a vector

This test confirms that x, y, baseline and corrected are all vectors. There should be 4 expectations for this test.

11. Each vector is the same length

This test confirms that x, y, baseline and corrected are all the same length. There should be 3 expectations for this test.

12. Spectrum can convert to dataframe

This test confirms that the as.data.frame method on a Spectrum object, does in fact return a dataframe.

13. Summary methods give outputs

This test confirms that the print and summary method on a Spectrum object, in this order, does in fact, print output to the console. There should be 2 expectations.