Lecture 5: Debugging

Dr Benjamin J. Morgan[1] and Dr Andrew R. McCluskey[1,2]

[1]*Department of Chemistry, University of Bath, email: b.j.morgan@bath.ac.uk*
[2]*Diamond Light Source, email: andrew.mccluskey@diamond.ac.uk*

August 13, 2019

Aim

This lecture will introduce *debugging* skills and suggest methods to improve your ability to overcome errors in your code.

1   Debugging

Human beings are fallible, therefore code will have bugs. However, there are some tips that we can use to deal with the bugs in our code and make them less of a nuesence.

One of the most powerful tools that is available to us is the quality of the code and libraries that are written in Python. For example, if you `import` the square-root (`sqrt`) function from the basic Python library `math` and give it a list as an argument,

```python
from math import sqrt

a = [0, 1, 2, 3]

print(sqrt(a))
```

This will lead to an error. However, this error will come with a *traceback* showing where the error came from and will give some information, at the bottom, about why the error occurred. You should get an error that looks like this,

```
TypeError                    Traceback (most recent
    call last)
<ipython-input-14-29539b5bdc2d> in <module>
----> 1 print(sqrt(a))

TypeError: must be real number, not list
```

As we can see, the error that has been thrown is a `TypeError`, and we are given the additional information about the error that: `must be real number, not list`. We can use this to parse the root cause of the problem, a `TypeError`

indicates that *something* is of the wrong *type*, and we are told that this *something* should be a `real number` and not a `list`. It is clear that the arugment being passed to the `sqrt()` function is of *type* `list`, and therefore this is the problem.[1]

The majority of Python libraries will have helpful error messages such as the one given above (especially commonly used packages such as NumPy, pandas and matplotlib). However, sometimes it is not immediately clear what the error message means, for this it is often necessary to leverage the most important tool in a programmers arsenal – the internet.

---

[1]Note that the `sqrt` function from the `math` library can only take single values and return the square-root. This NumPy `np.sqrt()` function can however handle objects of type `list` and NumPy arrays.