Efficient and extensible single molecule image analysis with PyStachio

Mark Leake

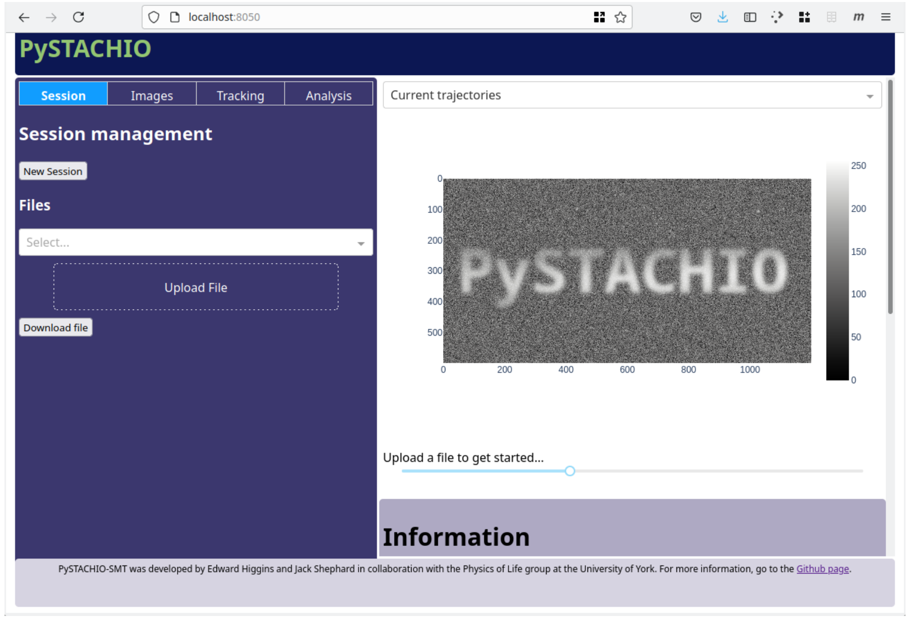
Mark Leake is Anniversary Chair of Biological Physics at the University of York, where he is also Coordinator of the Physics of Life Group at the Departments of Physics and Biology, and one of the Research Champion for Technologies for the Future in the Vice Chancellor's office.

Mark's research team specializes in developing innovative methods of single-molecule biophysics to address a range of complex biological questions. One of these tools involves single-molecule imaging with fluorescence microscopy at millisecond timescales. Post-experiment image analysis is a key part of this, and Mark and his team have devised, implemented, and refined several novel image analysis pipelines to extract key physical parameters such as molecular colocalisation, diffusion constants, and molecular stoichiometries in systems ranging from simple bacteria to complex human cancer cells.

Can you give us an overview of the project?

Thanks to funding from organizations like BBSRC, EPSRC, the EU's Marie Curie scheme, the Royal Academy of Engineers, and the Leverhulme Trust, my group has spent several years updating our fluorescence microscopes to take advantage of new large-chip sCMOS cameras and high-power high-stability excitation lasers and as a result we are now able to produce in the region of 1 TB of data per day, per microscope.

Analysis of this quantity of data is a considerable challenge and our previous analysis software was proving to be a bottleneck, as well as presenting a high barrier to entry for new users. Led by a postdoctoral research assistant we therefore began the daunting process of consolidating our analysis routines into one package written in Python to make use of its natural modularity and highly efficient computational libraries such as NumPy.



PySTACHIO-SMT was developed by Edward Higgins and Jack Shephard in collaboration with the Physics of Life group at the University of York. For more information, go to the Github page.

The launch screen of PySTACHIO

Did you work with a research software engineer (RSE) from the start of the project?

This project was spearheaded by a postdoctoral research assistant (Dr Jack Shepherd) who has taken a leading role in image analysis in my group. However, to ensure the final product was truly extensible and future proof, we realised we needed to draw on the expertise of a professional software developer, and we turned to Ed Higgins, a Research Software Engineer in the University of York IT services department who has expertise across a range of programming languages including MATLAB and Python – the languages we moved between as part of this project.

N8 CIR
COMPUTATIONALLY INTENSIVE RESEARCH

Within the first discussion of project aims it was obvious that Ed's support would prove absolutely invaluable as he had many suggestions for improvement and extension, such as developing a GUI which would allow the software to be web hosted or installed on a dedicated analysis server to reduce pressure on office workstations and make use of server-grade hardware including high-speed data interconnects.

What are the key challenges that RSE collaboration has helped to overcome?

Translating a highly diffuse codebase between different programming languages while simultaneously consolidating it into one software platform is a challenging undertaking which presented many foreseen and unforeseen hurdles.

My research team's previous analysis suite was in reality four or five different utilities which users chained together to produce a bespoke workflow for each experiment. As part of this project we therefore had to solve the problem of creating a program which could be used with almost unrestricted flexibility while remaining one cohesive piece.

Here Ed's expertise was invaluable as he quickly designed an overarching data architecture which we can perform operations on in almost any order. Ed also suggested and implemented a GUI which is suitable for web or server hosting and considerably easier for new users to work with and implemented robust parallelization to improve performance – we have measured an order of magnitude decrease in runtime with identical data and parameters.

Ed also set up our GitHub and introduced workflows for future software development to ensure that our software is maintainable and sustainable – and avoid another total rewrite in the future!

Having worked with RSEs, will it change your approach in the future?

Our approach has certainly been changed by our collaboration with the RSE team. Apart from changing our primary analysis language and relying more on external libraries, we now use proper version control software and archiving, and will now be providing Zenodo-generated DOIs of the exact state of the software used in our research publications to make sure that our research is truly open and reproducible.

Can you tell us more about your current or future research projects?

With the core single-molecule tracking in place, our focus is now turning towards adding functionality which will facilitate analysis of more complex microscopy data which relies on simultaneous imaging of multiple colour channels or separate polarizations of light. While this will require some new code being written, we now have a flexible base to modify which will ensure our ongoing modifications and additions are painless, maintainable, and sustainable.

We are also aiming to work towards more fully open science, and now that our codebase is in Python we have the new opportunity of using Jupyter Notebooks. In the future, we hope that we will be able to include a link in our articles where users can generate the figures in the paper themselves using sample data, the correct version of the code, and the reported analysis parameters.

We hope that the current movement towards transparency in science continues, supported by expert RSEs – in future, we intend to identify specific software needs during project conception and include RSE costs as part of grant proposals.