# The testing framework and its justification

The success of a SIM system relies on the accuracy of the computational imaging process. In general, the computational imaging process in microscopy includes different stages: data collection from a microscope, data pre-processing, image reconstruction, and image post-processing. Each stage can have independent codes in different programming languages. For example, the image collection is usually coded in LabVIEW/MATLAB. The data pre-processing and image post-processing are usually done with scripting languages like MATLAB, Python, to name a few, for convenience.



Fig. 1. Flowchart of testing framework for computational imaging in microscopy.

The image reconstruction step needs more computing resources and can be first written in MATLAB, Python for prototyping, then re-coded in Julia or C/C++, taking advantage of parallel processing to run on a graphical processing unit (GPU) on a single computer or on a high performance computing (HPC) cluster, for improved computational performance. For the development of a robust commercial microscope system, one needs a testing framework to make sure that the entire computational optical sensing and imaging (COSI) system is working correctly.

Our testing framework contains the following steps (Fig. 1):

1. Simulating the microscope image formation process and unit-testing of the simulation code that implements the image formation model.
2. Using the simulated data with ground truth in step S1 to test the image reconstruction method with unit testing.
3. Unit-testing the data collection and data pre-processing codes.
4. Running the reconstruction methods on experimental (collected) data to obtain the restored results.
5. Simulating image formation using the restored results from step S4 and comparing the simulated results with the collected data.

This testing framework will guarantee that:

1. Simulation works correctly and simulated data predicts the experimental data.
2. Reconstruction methods work correctly with the simulated data since one knows the ground truth in simulation.
3. The reconstruction methods work correctly with the experimental data because of the consistency and guarantee from steps S1 and S2.

This testing framework is independent of the programming language used. For convenience, we will build this testing framework in MATLAB and make sure the MATLAB codes work correctly for the flowchart in Fig. 1. Then we will translate the MATLAB part of the reconstruction into high-performance languages, such as, Python or Julia on HPC clusters for much better performance. The correctness of the Python or Julia codes on HPC clusters can be guaranteed by the unit-tests using the MATLAB codes’ results since the MATLAB codes would have already been tested and verified.

# 3. Methodology

The correctness of the framework in Section 2 relies on the unit tests of steps S1, S2, and S3 and the metrics used to make decisions at these steps. Since the microscopy imaging process deals with 2D and 3D images, we use the mean square error (MSE) [4] and the structural similarity index measure (SSIM) [5] for the decision metrics.

For the unit tests in step S1, one collects experimental data from a small and simple test object, specifically polystyrene beads labelled with fluorescence, and thus a numerical object replicating these beads is used in our SIM simulation. Then one establishes a mathematical model with its parameters, which captures the physics of the microscope’s image formation process and compares the simulated data from this model and numerical object with the collected experimental data from the test object. This guarantees a correct model to be used for the reconstruction methods. Besides testing for the correct model, one also needs unit tests for the simulation codes.

Once the model in step S1 is tested, one can use this tested model on some numerical objects (ground truth) to obtain simulated data. One then runs the reconstruction methods on the simulated data to obtain the restored results and compares the restored results with the ground truth using the MSE and SSIM similarity metrics. This guarantees the correctness of the reconstruction algorithms. Variant numerical objects should be used for this testing step S1 and one should construct both small-scale and large-scale objects. One also needs the unit tests for the reconstruction algorithm codes themselves.

The step S3 often has codes to collect digital data from the CCD camera. The tests here are simply unit tests of these codes.

In step S5, once one has the restored results of the experimental data, one would like to apply the model to the restored results to obtain model-generated data and compare it to the experimental data. Note that in step S1, one only uses simple objects to test the model and does not include the reconstruction method, so step S5 is necessary to test the model and the reconstruction methods at the same time.