**1. Parallel Efforts for teaching microscopy and image analysis.**

As already mentioned, Andrew, Jon, Mary Teruel, and Rich Lewis are planning to teach a microscopy class in the Spring. It will be a 5-week course in the first half of the spring, and they mentioned the possibility of each course listing the other as a suggested class. Andrew has shared course materials with us, although these are for a full-quarter course that Rich Lewis used to teach with Steven Smith (MCP 222), but Prof. Smith is emeritus now, and the course is being compressed into a 5-week offering as a result.

The course has covered several topics in the past, including –

* Understand how to determine resolution from the scope, what resolution you need for your question
* Size of confocal images, number of pixels with CCD imaging
* Optics in different types of microscopy
* Point spread functions on different instruments
* Optical bench to illustrate the light path of a compound microscope
* iBiology lectures -- it's been done twice in the past and it's worked really well (with Steven Smith, Rich Lewis)
* A one day symposium (attracted 50 people) on microscopy a couple years ago - Forrest Collman talked about image analysis and solicited problems from people in advance and worked through a couple of them during the symposium.
* Pinhole, airy units, how much light to use for image acquisition.
* Image formats and metadata, using the right pixel size, sampling rate.
* Perhaps some deconvolution
* Lightly touch the topic of how to streamline image acquisition for analysis.
* PSF lab: People collect images in different scopes of sub-resolution beads, and also add beads in agarose so you can see the effect of depth on spherical aberration, but nothing really about correcting for these problems.

One of the key ideas seems to be that it’s very important to test out analysis routines before the entire dataset has been collected – both for the sake of convenience, and because people often tend to not realize that if image acquisition has not been thought through carefully, the entire dataset might turn out to be useless. Jon also mentioned the possibility of generating data for our class during the microscopy course, but we would have to specifically request this. We are excited about these topics and would look forward to taking some of the students from this class, especially since they will be introduced to the basic physics of microscopy.

In addition to this course, the Sriram center imaging core is planning for a 2-3 day image analysis workshop that sounds extremely similar to our DataLucence class. They haven’t started planning it out yet, and would anticipate that such a workshop would be offered starting sometime in 2018, but might not do so if our course ends up fulfilling that niche. Our contact for this is Cedric Espenel.

Finally, Cedric mentioned a workshop and hackathon in Berkeley hosted by Stefan Van Der Walt at the Data Center. They have a useful website (imagexd.org) with a blog, and might be useful contacts should the need arise.

**2. Resources at microscopy facilities, and typical use**

We spoke with people at 3 imaging facilities, at Beckman, Lokey, and Sriram.

* Beckman (Jon Mulholland) – The facility has workstations with Imaris and Volocity software packages, but often a lot of analysis is done offsite by users. The vast majority (~80%) of imaging here is done on fixed samples, most often by confocal microscopy. Live cell microscopy is mostly for tissue culture samples, which are usually imaged with wide-field or spinning-disk-confocal microscopy. There is also a small amount of *Drosophila* work on a 2-photon system. Most users tend to have a strong medical background, with minimal to no prior experience with imaging and programming. They are often thinking about finding the “right”/“pretty” image, rather than thinking about drawing quantitative inferences from the images. They also often request help with the statistics of their experiments, which is quite difficult for the facility to provide.
* Lokey (Andrew Olson) – The facility has powerful workstations with Imaris, Neurolucida, Microvolution, ImageJ, and MATLAB. The typical demands vary quite a bit, but one problem that often comes up is identification of cells or nuclei (fairly straightforward segmentation with DAPI). People often do imaging with multiple channels, and sometimes more difficult procedures like identifying and tracking objects (eg. mitochondria in neurons). However, the majority of imaging still centers around confocal microscopy and fixed samples. The facility provides walkthroughs of FIJI but people tend to do the analysis on their own. With the commercial software, or with especially large datasets, the analysis is restricted to the facility. This is becoming quite common, especially because the sizes of datasets are growing, and analysis is only possible on the workstations even if the software is open source.
* Sriram (Cedric Espenel) – Users do a fair amount of live cell imaging, and cell tracking, but there is a lot of imaging of fixed samples as well. They have plans for introducing Jupiter notebooks for image analysis in Python at some point in the future.

All new users are trained at the facility before use of their scopes. The training revolves around optics and operation of the microscopes. Help is usually available for experiment planning, and image analysis, and users are encouraged to talk to the managers about their experiments so they can make the best use of the available resources to answer their scientific questions. However, our impression was that users tend not to ask for a lot of image analysis guidance at the facility, either because they don’t know it’s available, or because their needs are too specific.

The prevailing theme in our conversations on this topic was that people often do not think through the experiments before they’ve finished data collection, and then it turns out that they’ve taken undersampled or saturated images which are mostly useless. They really would like to exhort users to do a pilot of their analysis pipeline as soon as they have any data, to try to avoid such mishaps.

**3. Imaging topics that need attention in the Stanford community**

Image analysis topics that might be useful to cover are –

* Segmentation, object identification, tracking
* Background subtraction
* Flatfield correction
* Measuring properties
* Alignment, dealing with drift, stitching (most software does this for you)
* Adjusting for chromatic aberration
* Different filters that you can use for background and noise (Werner filter is used a lot)
* Deconvolution?
* Counting objects, measuring distances, tracking in 3D?
* Andrew pointed us to a well-worked example of a FIJI tutorial of nuclear watershed separation. Gaussian blur, thresholding, watershed separation. It also touches on the auto-threshold function which tests the 16 different thresholding methods in imageJ.

Mistakes people tend to make/community pain points–

* A lot of people don't understand optics well enough.
* Saturation, insufficient bit-depth
* Naming filenames with spaces
* Not enough contrast in images
* Undersampling – this is a big one, especially with lots of deconvolution
* No appropriate controls, standards during image acquisition
* Spectrum bleed-through, de-mixing?
* Imaging for several hours with all lasers at 100%
* Sometimes images get saved as jpegs, but usually loss of metadata is not a huge issue
* With analysis, the facilities don't look over your shoulder, it's often hard to know what mistakes users tend to make.
* Attempting to measure co-localization just by comparing intensity of the same pixels in two different channels (this was mentioned several times).
* People tend to lose metadata at the very first processing steps when they make their own tiffs. They have to be shown how to go back and pull their own metadata back out. Mostly however, this is not a problem since people generally stick to the proprietary file formats that come with the scopes.
* People quickly get frustrated when they run into errors and don’t know what they mean, don’t know how to ask for help, and the documentation doesn't provide clear answers.
* ImageJ is an accretion of things that have come together over the decades, so automation can be really frustrating. Macros only “sort of” work.

What they most wish people knew –

* The folks at the imaging facilities usually like to schedule a short chat to talk about the experiment before the users begin. This is to get them thinking along the lines of what they want to measure, and getting to a quantitative measurement instead of just taking a picture.
* What's missing at Stanford is a place where people can just help you with image analysis/processing (like with stats?)
* Understanding why you might want to use something other than ImageJ or CellProfiler
* Many people don't have any sense of proper experimental design, and proper controls (MCP 222 plans to cover this)
* FIJI is a good thing to learn, plugins, Python will be great
* People’s datasets often don't really have any naming convention.
* Need to get students to a point where they can look up stuff, program on their own
* There’s wide variation in people's existing knowledge. Plenty of people who still struggle with a spreadsheet, and others who have a code base in matlab. Many users have often never done any programming so it's hard to point them in the direction of Python, but “if you have a problem that you can't solve with an existing package, you need enough of a push to get a headstart and make some progress. Then your motivation to solve the problem pulls you through the rest of the way, but you need to somehow make some initial headway…”
* “If you're trying to plug people into this open source idea, you’ve got to be able to teach more than just instructing the machine how to do this. *Why did you select this particular algorithm instead of another?*”
* “Just as you need to put your work in the context of science that's been done before -- try to do the same with your code.” Reproducibility/Portability.
* “When developing your own analysis, one thing that is difficult as you think about programming, is that they lose sight of communicating to other people how the algorithm works”.