ImageXD 2017

Breakout Session:

Describe your image processing workflow. What pain points do you encounter? What directions you see for improvement? Identify one data/algorithm you need. Describe a data/algorithm you can share.

Pick a notetaker/s.
Come back 11:30am.

Group 1

Justin Kitzes: bats, birds - acoustic , spectrograms, clustering - matching to known library - 3-4GBs per day, process files in parallel what computer to use?Shareable at least 3 years, not that very real time, kaggle competitions - template matching + random forest, the phase is missing in the spectrogram, scale??? Python OpenCV

Yue Photoreceptor cells - Segmentation Gaussian mixture models, hierarchical models/machine learning What methods to pick??? Not spheres

The languages/packages matter! Standardize algorithms - input-output. Is it easy to run it in reproducible ways.

Kevin Koy - commercial software, Google Earth Engine, images are available, JS, microscopes with software, data is registered, but raw data is more useful,

Talita Perciano - search for data, metadata, people interested in results are correct, but data scientist care about what is not correct to implement

Confocal microscope imaging, multi channel Imaris, Amaris, manually labeling, some semi-automatic, How to make the biologist using these algorithms??? Inconsistent processing

Doctor-doctor dice coefficient is same as doctor-ML dice coefficient

Stella - workflow - input - output, how well are we doing, consensus, error estimate which can be propagated through the workflow.

General Ideas:

- Hard to decide between traditional methods and deep learning
- How do we share algorithms and data to benchmark (like computer vision scientists)
- Create a questionnaire to guide researchers to what algorithms to use
- Documentation - human readable, machine readable
- Google for algorithms - syntax
- Google for data -
- Inconsistent processing (through different packages)
- Propagating uncertainties through workflows
- Google for patterns
- Scale

FAIR Principles - findable, accessible,interooperable, reusable
Generating ground truth dataset
Aggregative frameworks for software (small modules)
Maintenance Overhead
Extending Algorithms to 3D <<<<

Group 2

Claire McQuin (Broad): fluorescence data (many channels) - custom built software which is a ”glorified wrapper around skimage”. Experiments generate a lot of data - variation in intensity across images. Automating a pipeline that can deal with high dynamic range is challenging. Computationally expensive operations. Computer vision knowledge - biologists don’t have it. fluorescence cells; 5D; segmentation is a pain point. Creating a library for segmentation using deep learning models
Pain points:

Aaron Marburg (APL, UW): Ocean observatories data. CamHD - camera pointed at a volcanic mound underwater. TB of video data that is not currently being analyzed. Quick prototyping. pain point: quickly prototyping

Bernease: Data Scientist at eScience. A variety of questions around Computer Vision and NLP. The generation of synthetic images. What properties of the image to retain.

Maryana (UCSF): Computer scientist. Postdoc at UCSF. Registration of different modalities of imaging of human brains. Cross-platform distribution of software written in Python with Qt. Biologists and physicians don’t know how to use programming.

Tanvi (UW): biologist. High-speed videos of insects moving around in space, or fixed on a rod. Identification of regions of interest. Physical objects that change shape, location, color, etc. Direct linear transformation (DLT) used to take this into account. Depends on legacy software - slow, hard to tweak, etc. Another data-set is x-ray tomography data. This is a standard data-format. Because structures are not stained, low contrast, manual processing needed. STL files - need to manipulate these structures, so that you can do “actual analysis” -- a feature extraction problem.

Ariel: brain data (MRI) image analysis; pain point: scalability, tried to use spark, dask, but deploying software is a lot of work, somewhat unreliable, finicky; but the domain user would like to have a button to do the processing instead.

Mike: work with Ariel, work with computational neuroscience and understand how neural implants; use scikit image; use of simulation data to train models; pain points: scalability

Dani (BIDS): challenges are what are general-purpose tools that are useful to people in many fields? Image search for example. Other is geometric descriptions that depend more closely on the domain; catalog of patterns, e.g., blobs and knobs to represent the statistical mechanics

Andre (Physicist - Zurich): computer scientists and domain scientists. One of the domains is solar physics. How solar flares work. When they happen there are implications for us over here, but we don't have models to predict when they will happen. One of the projects is based on solar dynamics observatory - 1 hz, 4000 x 4000 px pictures of the sun. Most of this data never gets analyzed, because scale. Making it possible to use the full PB of data is hard. Identifying interesting events and classifying them. X-ray space-craft (a telescope on a space-craft).

Nick (UW Pathology): working with engineers to create a custom light-sheet microscope to image large chunks of tissue. The instrument is there, but the data is ginormous. It takes time to analyze, and uses proprietary software. In the midst of working on a pipeline on the cloud that gets the data from the instrument. Speed is important, interactive visualization would be good.

Kyle (University of Idaho): image-based modeling for computational biology. Multi-scale models. Light sheet imaging of zebra-fish. Clojure expert - Lisp person who hates Java, but somehow works on ImageJ, so hates his life on a daily basis (?)

Stefan (BIDS): visual insight from image processing. Classification of rock types, super-resolution, images of insects, identifying (individual!) sharks on the South Africa coast. 3D modeling of biomedical images (2D ⇔ 3D). Warping images of astro data. Measurements of particles; scikit image

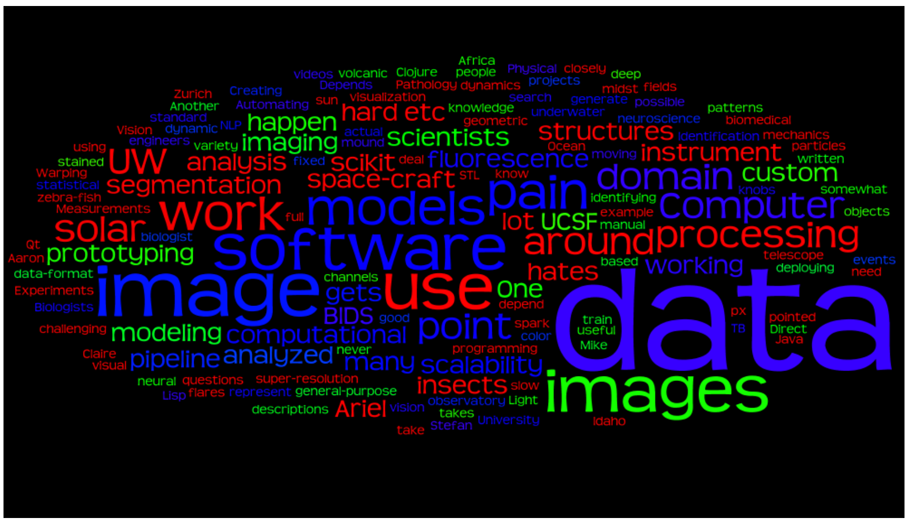
General ideas:

- 3D features: in pathology this has never been explored, but there's potential for knowledge transmission between different fields. This is also true in cell biology.
- "Blob-finding" is a common thing somehow. Does this differ across fields?
- "Deconvolution" - using a known PSF to improve data quality and enhance structural info
- Working with large datasets and need for stitching: tera-stitcher
- Registration of large 3D samples: ITK

* Claire (Broad Institute) - Works on CellProfiler project. Variation in intensity across areas in image. Computational power. Knowledge of computer vision techniques for biologists
* Aaron (UW Applied Physics Lab) - Works on CamHD, images from ocean. Transferring from OpenCV/C++ style image processing to newer paradigm (Python, etc). Limited processing power on autonomous vehicles.
* Bernease - Interested in generating synthetic images for science and working with diagrams. No pipeline now, but still in the problem finding stage.
* Maryana (UCSF) - Has mostly registration on brain images with florescence. Challenge: how to ship her software, especially cross-platform. Memory.
* Tanvi (UW Biology) - Works on 3D high speed or resolution videos of insects. Dynamic linear transformation (DLT). Challenge: segmentation in video with changing structure shape, low contrast and similar intensity/color, lastly one Matlab package used for DLT.
* Ariel (UW eScience) - Challenge: deploying and customizing parallel libraries (e.g., Spark, Dask) on rapidly changing APIs.
* Michael (UW) - Modeling vision with patients with retinal implant. Challenges: scaling, parallelizing pixel-by-pixel computation.

* Dani (Berkeley) - Challenges: creating libraries that categorize images,
* Andre (Zurich) - Analyzing images (~1/sec) of sun for solar flairs. Also working with spectrograms. Challenges: very dynamic changes, various spectral channels,
* Nick (UW Pathology) - Working on custom light sheet microscope. Challenges: large data size.
* Kyle (UIdaho) - Image-based modeling for computational biology. Challenges: Image processing tools in functional, LISP languages. Tools used: Ilastik.
* Stefan (BIDS) - Worked across image processing, with focus on scikit-image. Challenge: what tools are important to scientists, how to prioritize and find what's needed.

Tools: Big Stitcher (ImageJ), KitWare through SlideAtlas (uses ITK-based registration)



Summary

- Hard to choose methods (“too many algorithms”) => typology, decision trees
- “Google for algorithms”, shared language
- “Google for data” => finding and extracting data is time consuming
- Sharing processing pipelines
- Error propagation in multi-step pipelines

- Standards, benchmarking, etc.
    - https://www.nist.gov/publications/do-we-trust-image-measurements-variability-accuracy-and-traceability-image-features
- Software distribution: sometimes hard to get users to use software because it might be hard to compile/build across platforms.
- Education / training
- **SCALE**
- Ground-truth datasets
- Modernizing existing algorithms & applications (do we port Matlab to Python, or what is coming on the horizon?)
- Legacy code, software sustainability
- "Frameworks" vs custom algorithms : do we want a node- or go-like eco-system of image processing tools? On the other hand, maintenance and distribution of software requires work, and it's not clear how this will happen with fully distributed development.
- Extending to 3D, video, hyperspectral, etc.

Other snippets, not sure where they fit:

- Levels of abstraction
- "Blobs" - everyth

FAIR principle == Findable, accessible, interoperable, reusable

SUMMARY WHITE PAPER:
https://docs.google.com/document/d/1G_OMzPcuPuqmrk_PAsocouwQ4iVFZhnVyVLzK-a5ock/edit#

IDEAS ABOUT SCALING
https://docs.google.com/document/d/12AW_Dd-b1y623EXw4BOXTvHq-SmbRvQuvUEhA1iR__E/edit?usp=sharing