LOCI, Fiji & ImageJ2
Philosophy of optical research

- *In vivo* developmental biology
- Capture everything possible about a sample
- Multiple dimensions
  - Emission spectra
  - Lifetime
  - Cell polarization
- Greater resolution and range
  - Adaptive optics (peer deeper into the soup)
  - Longer time series
Philosophy of software development

- Interoperability
  - Leverage existing tools
  - “Glue code”
  - Implement missing features

- Open source (federally funded—ethical obligation)

- Of benefit to LOCI's research

- Of benefit to the broader scientific community
Computational tools

- **Acquisition software**
  - WiscScan
  - Micro-Manager

- **Visualization & analysis**
  - ImageJ, Fiji
  - VisBio
  - Slim Plotter, TRI2
  - FARSIGHT, ITK
  - CellProfiler

- **Data management**
  - Open Microscopy Environment

A major goal of ImageJ2 is to integrate these tools into a common framework.
LOCI's vision of scientific workflow

- **Acquire** data however you like (e.g., μManager)
  - If possible, record data in a standard format (OME-TIFF)
  - If not, convert from proprietary format (Bio-Formats)

- **Store and organize** data centrally (OMERO server)

- **Analyze** data through various client & desktop software
  - Smart clients can access data from the server (Fiji via plugins, OMERO.webclient, OMERO.insight)
  - Other software can work with data on disk (Fiji, etc.)

- **Record changes** back in the database, with versioning

- **Share** data over the Internet with colleagues
  - Both informally and as part of publication
What is Fiji?

- An easy way to develop novel algorithms
  - Plugins, macros or scripts
- A distribution platform for those algorithms
- A suite of tools to facilitate these goals
  - Script Editor (currently supports 7 languages)
  - Tutorial Maker
  - Fiji Updater
    - Access latest updates easily
    - Submit code directly to Fiji repository
Fiji vs. ImageJ

- Fiji is a distribution of ImageJ bundled with useful scientific image processing routines.

<table>
<thead>
<tr>
<th></th>
<th>ImageJ</th>
<th>Fiji</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>~3-5 MB</td>
<td>~45 MB</td>
</tr>
<tr>
<td>Focus</td>
<td>Various image processing</td>
<td>Segmentation &amp; registration</td>
</tr>
<tr>
<td>License</td>
<td>Public domain</td>
<td>GPL (various plugin licenses)</td>
</tr>
<tr>
<td>Tools</td>
<td>Minimal</td>
<td>Rich</td>
</tr>
</tbody>
</table>

- Fiji is the GNU/Linux distribution to ImageJ's Linux kernel.
- “Fiji Is Just ImageJ—batteries included”
- “Some assembly required”
Origins of Fiji

- **Albert Cardona ("Father" of Fiji)**
  - Institute of Neuroinformatics, Uni/ETH Zurich
  - Albert is a biologist who wrote TrakEM2
  - Began as a way to easily distribute TrakEM2

- **Johannes Schindelin ("Mother" of Fiji)**
  - Max Planck Institute, Cell Biology Division (MPI-CBG) in Dresden
  - Johannes is the one who did the hard work
  - Maintains the repository, build system, etc.

- **Pavel Tomancak**
  - Principal Investigator at MPI-CBG
  - Working to secure additional funding for Fiji
Other hackathon participants

- **Stephan Saalfeld & Stephan Preibisch**
  - PhD students at MPI-CBG
  - Created excellent imglib library
  - Saalfeld developed much of TrakEM2's registration code
  - Preibisch developed Fiji's 2D & 3D Stitcher plugins

- **Mark Longair**
  - Just finished his PhD in neurobiology
  - Studies drosophila, specifically neuropils
  - Developed Fiji's Simple Neurite Tracer plugin
  - Moving from U. Edinburgh to Albert's group in Zurich
Close the sigma palette window to continue

Instructions:

- Complete Path
- Cancel Path

View paths: projected through all slices (up to 2 slices to each side)

Click to change Path colours:

- Show only selected path
- Hessian-based analysis
- Use preprocessed image

\( \sigma = 0.2884, \text{ multiplier } = \)

Pick Sigma and Maximum

- Load Lab
- Import SWC
- Export as

Save Traces File

Adjust maximum value: 11
Other hackathon participants

- **Bene Schmid**
  - Developed Fiji's 3D Viewer (uses Java3D)

- **Erwin Frise**
  - Developed level sets plugin for image segmentation

- **Larry Lindsey**
  - U. Austin
  - Charged with maintaining a package called “Reconstruct”
  - Saw TrakEM2/Fiji could already do most of the same things
  - Spent his time learning about imglib
Other hackathon participants

- **Verena Kaynig**
  - Computer scientist interested in machine learning
  - Wrote a “trainable segmentation” plugin

- **Ignacio Arganda**
  - Developed Fiji's bUnwarpJ plugin for image registration
  - Improved Verena's Trainable Segmentation plugin after she left

- **Rubén Muñoz**
  - EMBL programmer working with Olympus ScanR systems
  - Discussed Bio-Formats and OME
  - Developed his own database with web interface
  - Uses OME-TIFF
Other Fiji developers

- Jean-Yves Tinevez
  - Institut Pasteur
- Gabriel Landini
  - Prof. Of Analytical Pathology
  - School of Dentistry, U. Birmingham, England
- Us!
- Probably others I'm forgetting
What I worked on

- Learned Fiji development workflow
- Algorithm launcher for imglib
  - Execute any imglib OutputAlgorithm in Fiji
- Autogenerated an ImageJ delegation layer
  - Attempt failed; now using a simpler approach
- Fiji/ITK integration
  - Proof of concept; call ITK image filters from Fiji
- Some work on Bio-Formats
  - Same dependency injection pains as us
run("ImgLib Algorithm Launcher", "algorithm=Downsample scale_factor=0.7");
Reminder: Vision of Fiji

- An easy way to develop novel algorithms
  - Plugins, macros or scripts
- A distribution platform for those algorithms
- A suite of tools to facilitate these goals
  - Script Editor (currently supports 7 languages)
  - Tutorial Maker
  - Fiji Updater
    - Access latest updates easily
    - Submit code directly to Fiji repository
Vision of ImageJ2

- Core image processing library for Fiji
- Major goals:
  - Improve the technical design
  - Integration and interoperability
  - Grow the ImageJ community
Vision of ImageJ2

• Improve the technical design
  – A solid base on which to build scientific tools
  – Flexible enough to interface with a variety of existing software
  – Able to represent scientific image data in a variety of contexts (e.g., N-dimensional)
  – Buzzwords like “modular” and “extensible”
Vision of ImageJ2

- Integration and interoperability
  - Adapt Slim Plotter & VisBio features into ImageJ2
  - Glue code between ImageJ2 and CellProfiler
  - Glue code between ImageJ2, ITK and FARSIGHT
Vision of ImageJ2

- Grow the ImageJ community
  - Backwards compatible with existing plugins as much as possible (at least initially)
  - The more users of ImageJ, the easier it will be to share analysis routines and results
  - Fiji certainly shares this goal
Vision: Fiji vs. ImageJ2

<table>
<thead>
<tr>
<th>ImageJ2</th>
<th>Fiji</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software development infrastructure</td>
<td>Image processing algorithm development and deployment</td>
</tr>
<tr>
<td>General image processing</td>
<td>Scientific image processing, particularly life sciences (microscopy, neuroscience, but also computer vision)</td>
</tr>
<tr>
<td>Boring baseline stuff</td>
<td>Fun domain-specific applications</td>
</tr>
</tbody>
</table>

- Where ImageJ2 ends and Fiji begins is fluid
- We may develop analysis routines useful to LOCI that end up in Fiji rather than ImageJ2
Components of ImageJ2

- Major components of ImageJ2
  1) **Data model** – imglib library
  2) **Display** – Java AWT, JAI, Swing
  3) **Input/output** – Bio-Formats architecture
  4) **Regions of interest (ROIs)** – Java AWT, JHotDraw
  5) **Scripting & plugins** – Java 6 Scripting Framework

- All of these areas have significant limitations and will benefit from enhancements and refactoring
Components of ImageJ2

- To start, each of us focus on one?
  1) **Data model** – Aivar?
  2) **Display** – Rick?
  3) **Input/output** – Barry?
  4) **Regions of interest (ROIs)** – Brian?
  5) **Scripting & plugins** – Grant?
An old MIDIAS diagram, just for fun... :-)

Diagram showing the interconnections between scripting tools (Python, MATLAB scripts) and databases (WormBase, OME databases, other databases), with client software (VisBio, WiscScan, ImageJ) and file formats (OME-XML, TIFF files) as key components.