Tutorial

FIJI (FIJI is ImageJ)

Michael Uchic
AFRL/RXCM

Integrity ★ Service ★ Excellence
Outline

• Introduction

• Overview of FIJI/ImageJ Functionality
  - Emphasis on filters and techniques useful for serial sectioning data post-processing

• Examples

• Q & A
Introduction

- FIJI (and ImageJ) = Image processing package
  - GUI enables quick prototyping
  - FIJI has lots of plug-ins to assist with 3D data analysis

- **Open source** (GNU) – built off of ImageJ2 (BSD)
  - Java
  - Actively developed

- Close collaboration with other SciJava/open microscopy tools

http://fiji.sc/
Reasons to consider using FIJI/ImageJ

• Core code is robust; comparatively fast execution

• FIJI has very effective filter suite for data post-processing, esp. for common 3D tasks
  - Extensive suite of scripts/plug-ins developed by scientific community

• Macro language enables rapid construction of automation workflows

• Wide proliferation enables sharing/verification of workflow

• Open source enables insight into algorithm construction
Common Operations for 3D Data

- Importing Image Stacks
- Crop/Resizing
- Image Restoration
  - 3D Kernel Operations
- Histogram Adjustment
- Applying Threshold
- Interactive Stack and 3D Viewing
Importing Image Stacks

Helpful to have numeric ordering of image stack
Crop/Resizing

Works on stacks as well as individual images
Histogram Adjustment

Works on stacks as well as individual images
3D kernel operations can be readily performed
Z-stack projection

Single image from sectioning experiment
MIN Z-projection thru 300 sections

Effective for feature enhancement, removing some artifacts
Image Stitching
http://fiji.sc/Grid/Collection_stitching

- Reconstruct big image stacks—arbitrary spacing
- Pair-wise registration to sub-pixel resolution (FFT)
- Global optimization scheme to minimize propagating alignment errors

\[
\arg\min_{T_{VF}} \sum_{A \in V \setminus \{F\}} \left( \sum_{B \in V \setminus \{A\}} \| \gamma_{BF} - \gamma_{AF} - p_{AB} \|^2 \right)
\]

- Fusing options (blending); new tile coordinates also output

Stack Registration

http://fiji.sc/Register_Virtual_Stack_Slices

Use of SIFT (keypoint detection): Scale & Rotation Invariant Matching
Why Macros?

Macros can be used to

- automate repetitive tasks
- document what you did
- share common procedures
- add tools to the toolbar
- add keyboard shortcuts

Please be aware that there are several other available scripting languages that are more powerful than macros, too! See the sidebar on the right.

- Record/Edit/Run options simplify macro construction
Macro

http://fiji.sc/Introduction_into_Macro_Programming

- Example of Macro for Stitching Serial Sectioning Data

```java
1 data_path="C:\Users\"
2 for (i=0;i<101;z++) {
3     open(data_path+"Original_Data\"+i+".tif");
4     run("Canvas Size...", "width=1000 height=1000 position=Center zero");
5     saveAs("Tiff", data_path+"Cropped_Data\"+i+".tif");
6     close();
7 }
8 ```
# Citing SciJava Projects

http://imagej.net/Citing

<table>
<thead>
<tr>
<th>Project</th>
<th>Citation</th>
<th>Google Scholar</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImgLib2</td>
<td>Pietzsch, T.; Preibisch, S. &amp; Tomancak, P. et al. (2012), &quot;ImgLib2—generic image processing in Java&quot;, Bioinformatics <strong>28</strong>(<em>22</em>): 3009-3011, PMID 22962343 [🔗].</td>
<td>on Google Scholar</td>
</tr>
</tbody>
</table>

**Other**

Browse the list of citable software and publications using Fiji for additional citation information.
Helpful Web Resources

• FIJI
  http://fiji.sc/Category:Tutorials

• ImageJ Documentation Page

• University of Chicago Integrated Microscopy Core ImageJ Site (C. Labno)
  https://digital.bsd.uchicago.edu/ImagejandFijiHelp.html
Summary

• Open-source, robust software for image processing

• Plug-ins very good for key 3D tasks (stitching, alignment)

• Macro language simplifies task automation