SCIFIO: An Extensible Framework for Scientific Image Interoperability

Mark Hiner[1,2], Curtis Rueden[1,2], Melissa Linkert[2,3], Jason Swedlow[2,3] and Kevin Eliceiri[1,2]

[1]*LOCI, UW-Madison,* [2]*Open Microscopy Environment,* [3]*Glencoe Software, Inc.*

SCIFIO generalizes beyond the life sciences

Bio-Formats is a large collection of translators from PFFs to OME-TIFF. It solves the PFF problem for many life sciences, but is restricted to formats describable by the OME-XML schema (fixed 5-D).

With SCIFIO we have defined a foundation for creating plug-ins like Bio-Formats, facilitating its continued development, but also providing a template for new plug-ins in all image-based disciplines.

SCIFIO provides an extensible foundation

For SCIFIO to succeed as an extensible framework, it must be completely transparent to external developers. Formats are divided into atomic pieces (Figure 1) to make their use and extension as clear as possible.

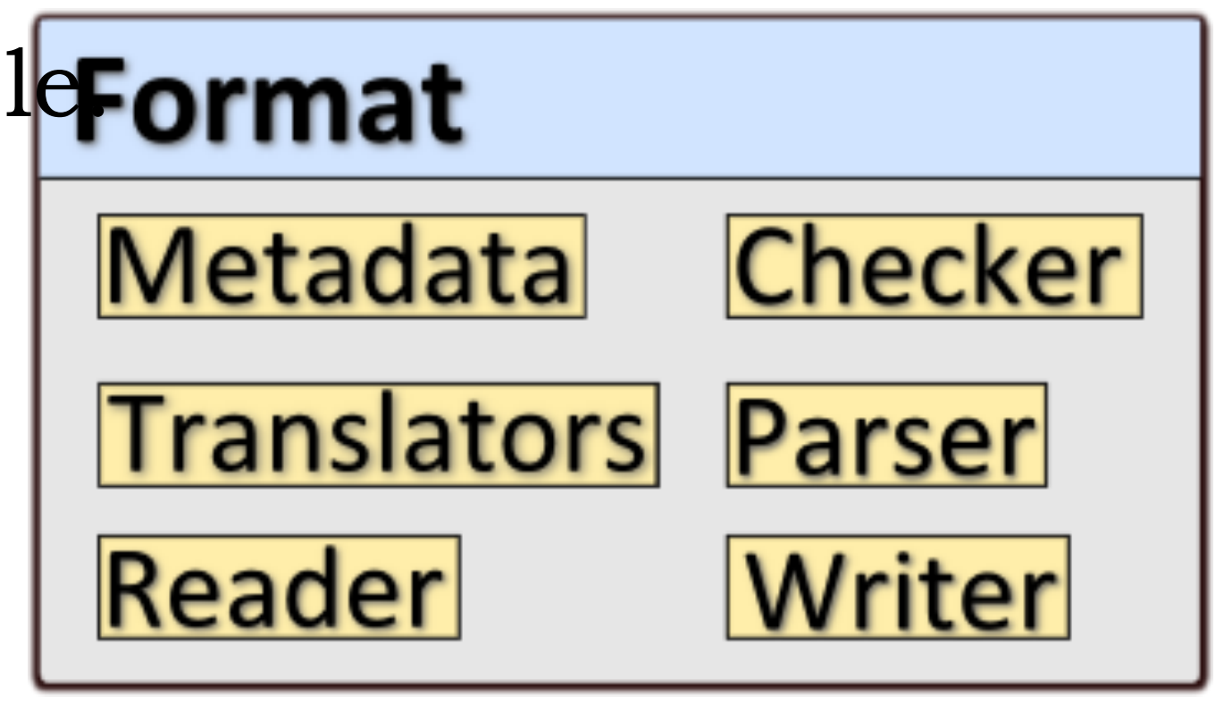

Figure 1: Components of a SCIFIO Format

Checkers indicate if an image is supported. Parsers create Metadata. Readers and Writers use Metadata to read and write the actual pixel data. Translators convert Metadata from one format to another. This modularity keeps code organized and serves as a road map for development. Programmers can focus on one piece at a time, and only the pieces they need.

SCIFIO uses the SciJava plugin framework to maximize extensibility. Each format is a plugin, allowing new formats to be dynamically discovered based on the user's or application's needs.

For more information

SCIFIO: scif.io
Bio-Formats: openmicroscopy.org/site/products/bio-formats
OME: openmicroscopy.org
ImageJ: imagej.net
SciJava: scijava.org

Proprietary file formats (PFFs) restrict the flow of data

Sharing data is a fundamental part of the scientific process, and for many fields images are the medium of that data. However, instruments for generating images often produce output as PFFs specific to their manufacturers. These PFFs create a barrier between researchers, challenging them to find a common ground.

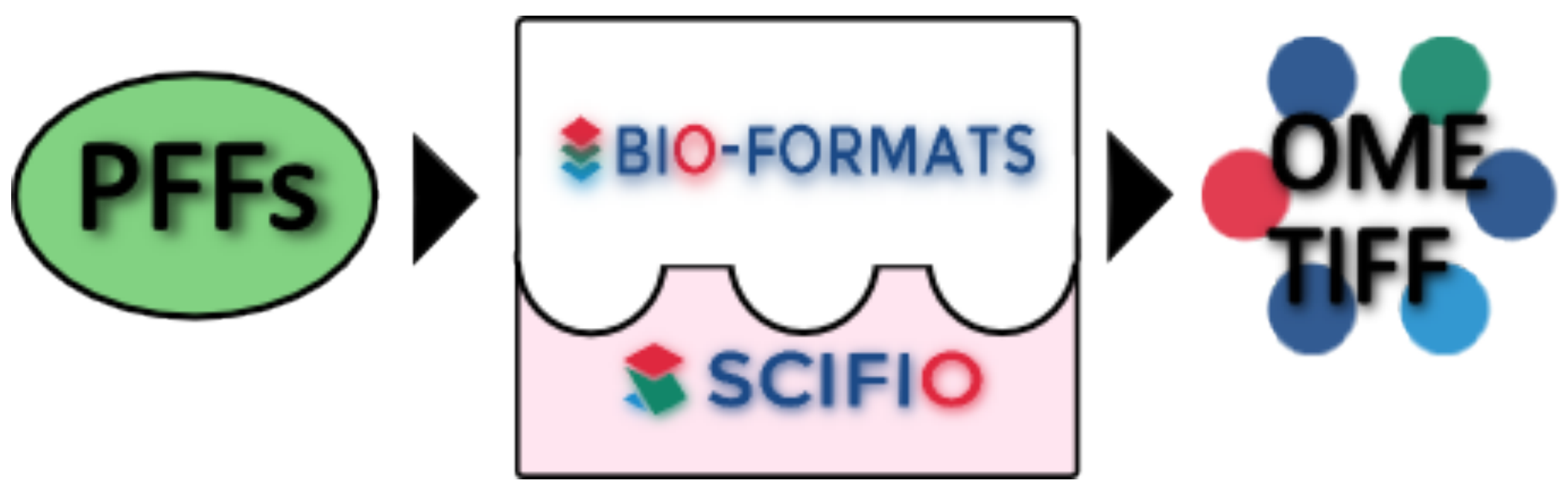

Figure 2: SCIFIO converts images to open formats, such as OME-TIFF

SCIFIO solves the PFF problem

SCIFIO provides a consistent definition for image formats and the tools to convert between these formats. Conversion allows us to take advantage of open exchange formats (Figure 2) which allow data to be shared freely with collaborators. Unfortunately, new PFFs are constantly being invented, but by simplifying and generalizing the process to support them, the SCIFIO framework mitigates a proliferation of incompatibility.

SCIFIO promotes interoperability

Permissive licensing allows any program to incorporate SCIFIO and automatically gain support for all SCIFIO formats. Future addition of new PFFs and plug-ins to the SCIFIO framework will immediately become available to all such programs. The discovery process is designed to be dynamic; therefore the SCIFIO core itself is extremely lightweight and allows users to choose which plug-ins (and thus formats) are relevant to their use. Over 30 applications already take advantage of SCIFIO and Bio-Formats in this way (e.g., ImageJ, Figure 3).

To follow the development of SCIFIO:
github.com/scifio/scifio/issues

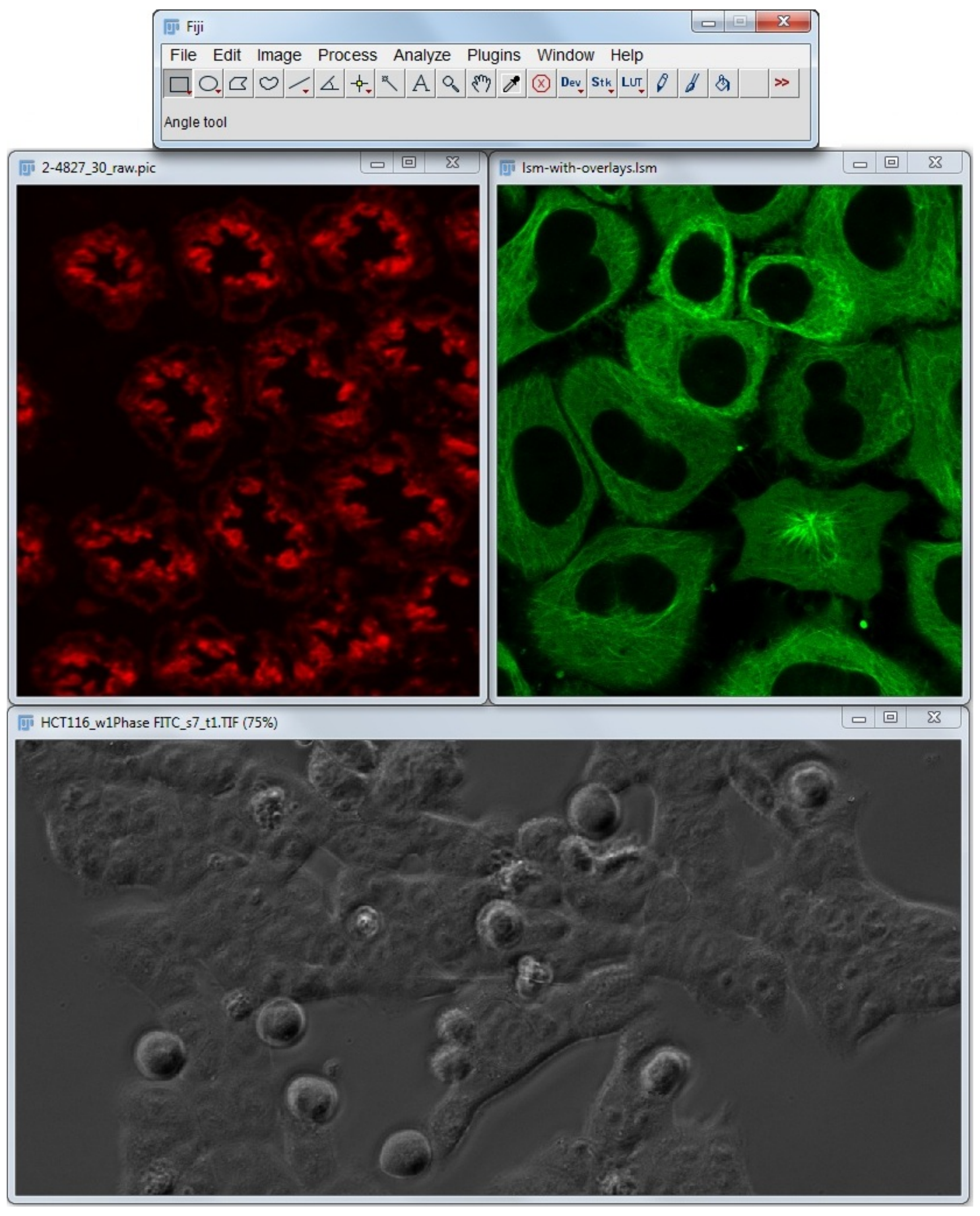

Figure 3: Applications using SCIFIO can support a variety of image formats.

Development Progress

- Core framework completed with 20+ open formats supported.
- Refactored to N-dimensional data model to maximize flexibility.
- Dynamic plug-in incorporation, including: formats, filters (dynamic behavior modifiers) and command line tools. Removes the need for configuration files.
- Bio-Formats backwards compatibility layer (scifio-bf-compat).
- SCIFIO implementation of proprietary formats (ome-formats, scifio-lifesci).
- Plane caching to perform operations on arbitrarily large datasets.
- Provides the I/O core for ImageJ2, KNIME image processing (KNIP)

Goals for SCIFIO

The past year saw significant public exposure for SCIFIO – from the ImageJ2 beta, KNIP collaboration, and the first community-contributed SCIFIO formats. The common feedback is that SCIFIO is intuitive for developers, but lacks performance optimization for users. Thus our immediate goals are iterative refinements to get SCIFIO to the performance levels of Bio-Formats and ImageJ 1.x. For common formats (e.g. TIFF) and use scenarios (repeated reads of a given dataset) we plan to exceed these standards. Further short-term goals include a unification with the ImgLib2 metadata model, automatic pyramiding of large image planes and continued expansion to existing software platforms (e.g. OMERO, ITK) as we work