



Icy : A new open source software for bio image analysis

Brought by the Quantitative Image Analysis Unit @ Pasteur Institute



Icy

Icy

Technically
speaking

OMERO

Integration with Icy



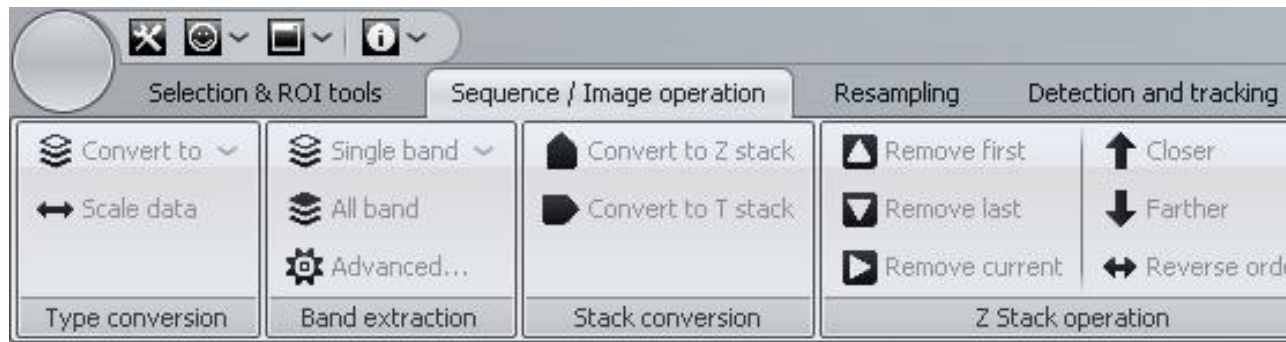
Icy

Bio image analysis software



History of Icy

- 2003
 - The lab needs to deal with 3D+t images
 - Quia is created
 - New needs rise coming from the feedback of our partners
 - As the quality of the software improves, we wanted to make it open source
 - We decided to rewrite a full kernel with last technologies and recents libs, and to make it open source.
- Dec 2009
 - First line of code of Icy.
- Mar 2011
 - Release of Icy. (about 13 months of dev)



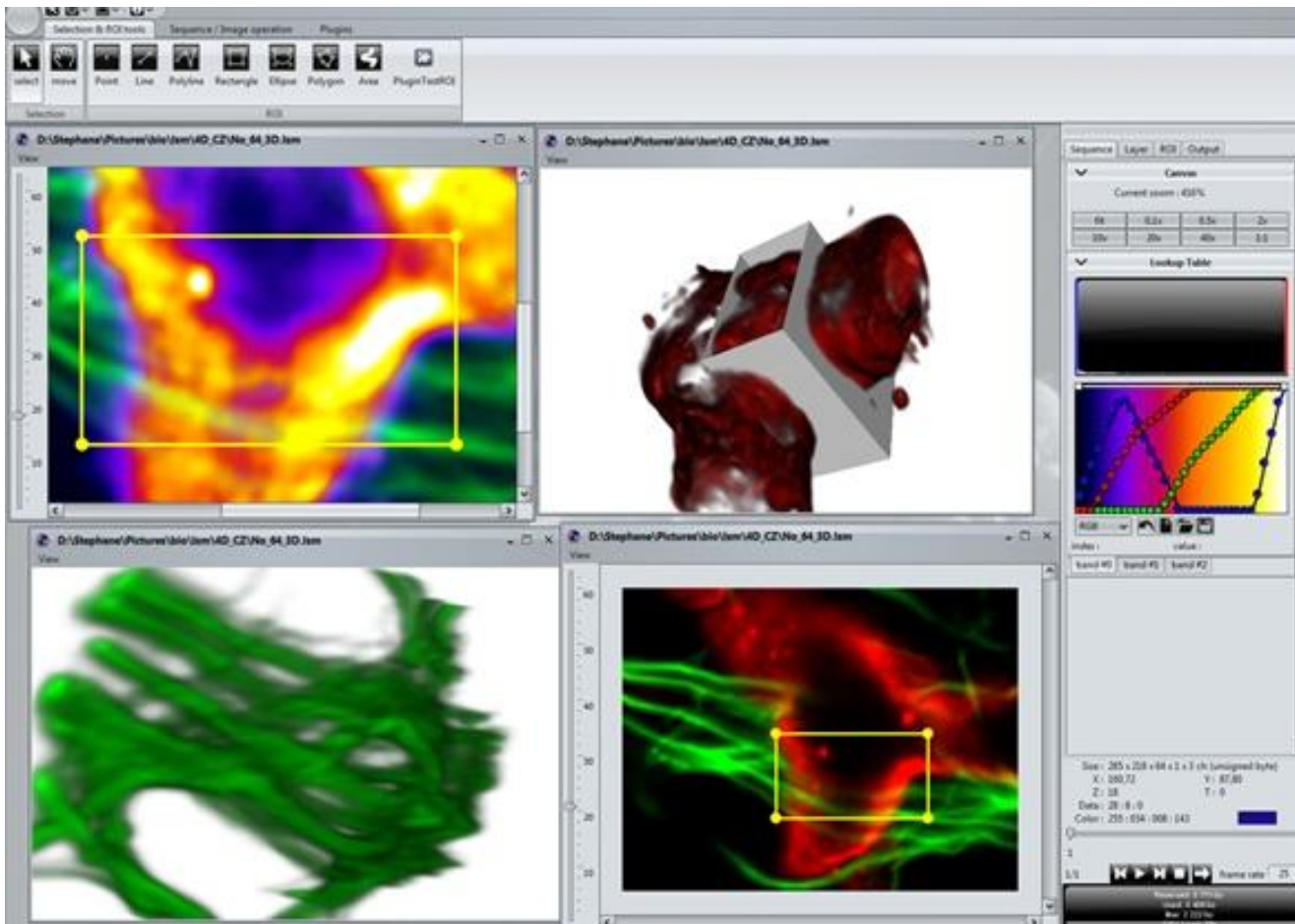
GUI Overview

Plugins first !

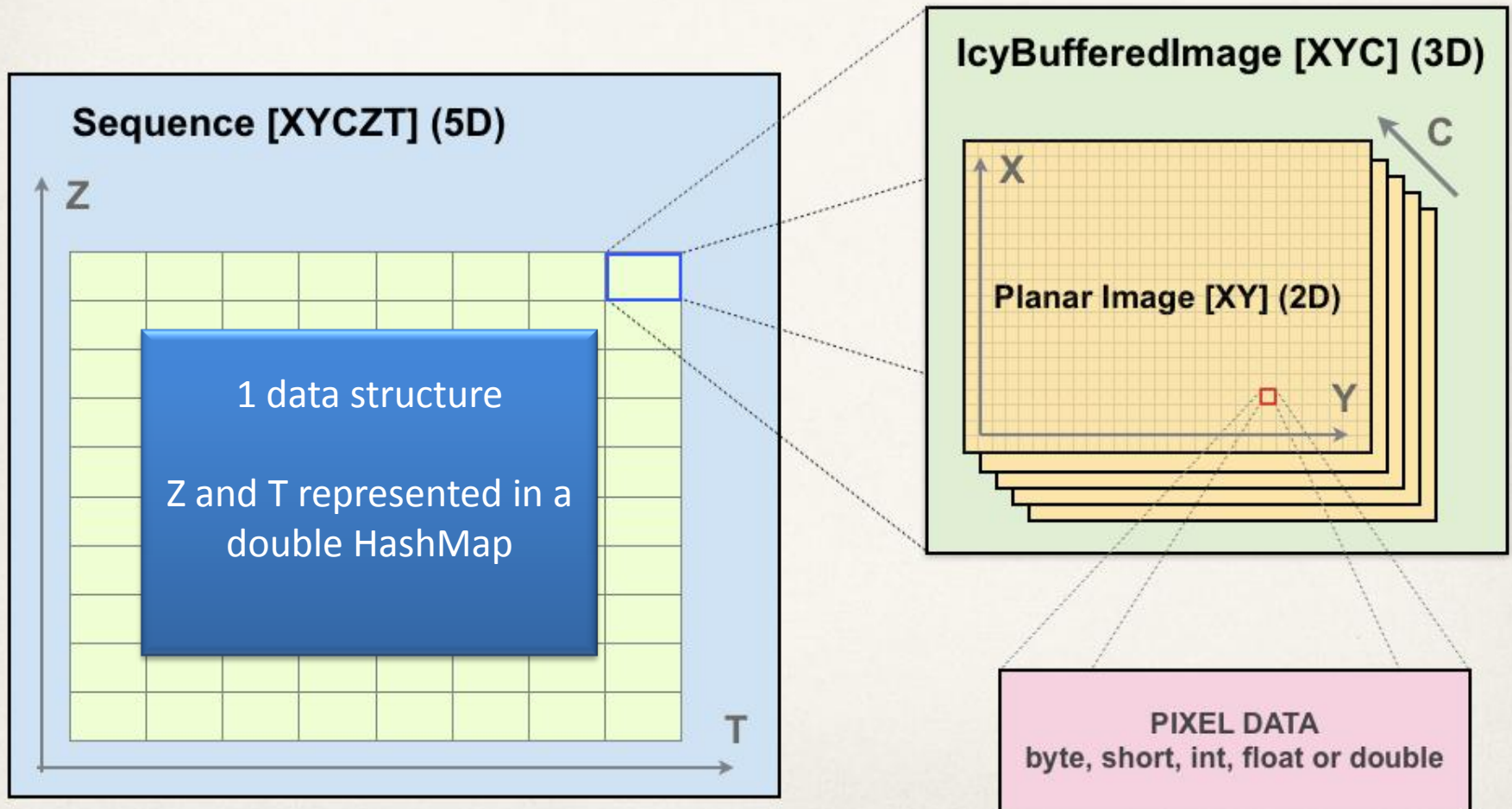
Ribbon without difference between native functionalities and plugins.

Link 1 sequence to n synchronized viewers

Multi-Threaded and thread-safe interface



Sequence of image data structure



Built-in look up table
structure coherent with
VTK

Fast conversion to the
VTK layer

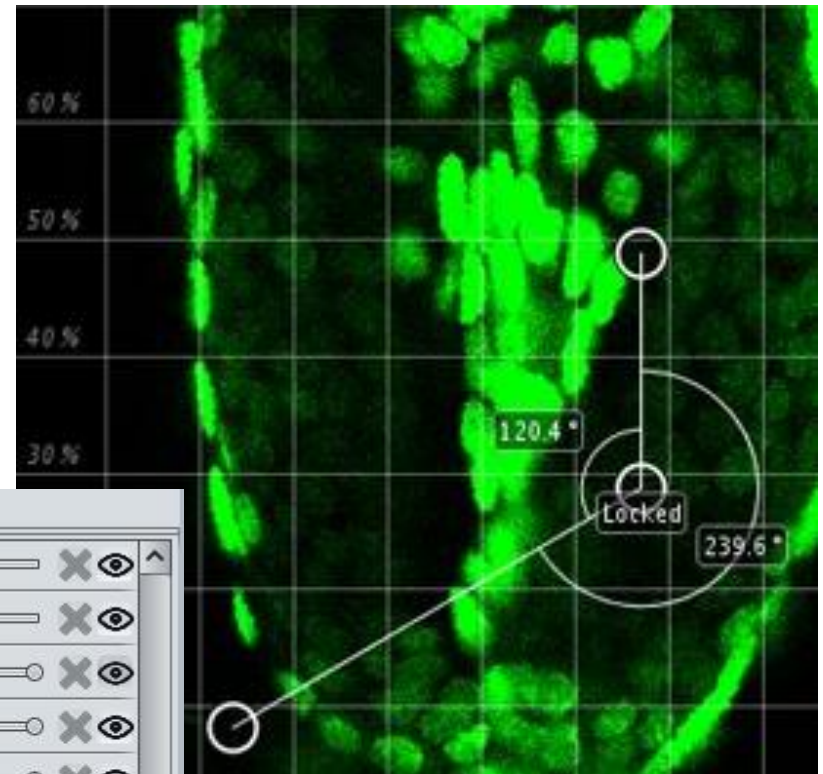
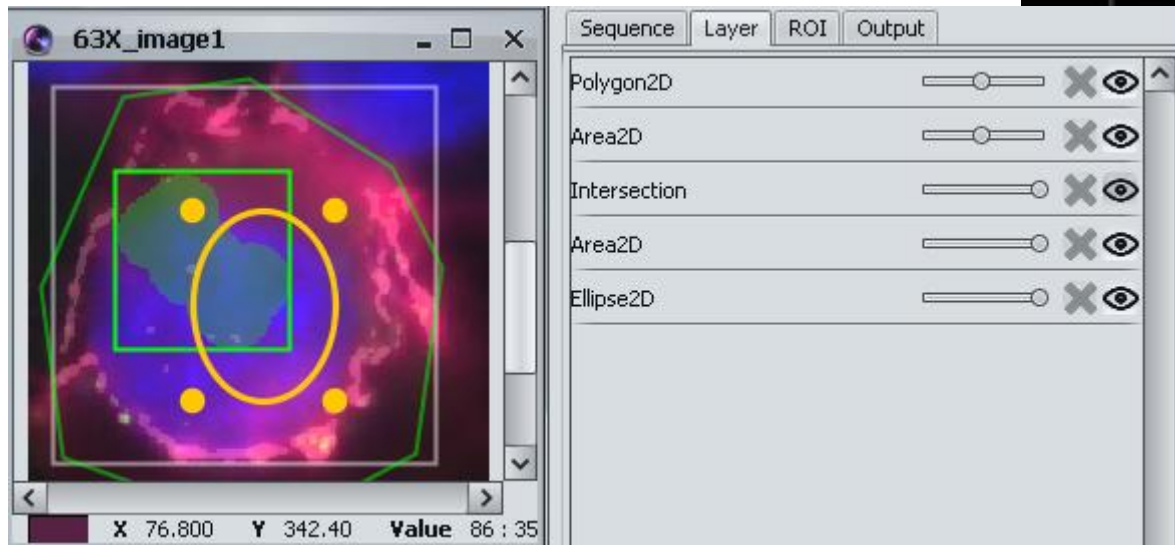
API to convert all data
transparently to specific
pixel format


Compatible with the
native java
BufferedImage object

Overlay structure

API for layer management

- 2D / 3D
- Animated
- Cumulative
- Key/Mouse Events
- Natively integrated in the viewers





Home Downloads Plugin Resources

Plugin List

Current number of plugins: 44

a

- [Active Contours](#) Fast region-b
- [Animated painter](#) This tutorial e
- [Anisotropic Filter](#) Anisotropic fil

c

- [Color Picker Threshold](#) Image segme
- [Connected Components](#) Connected co
- [Create Simple Area ROI](#) This tutorial e

d

- [Diffusion Filters Tool \(Sorin\)](#) Some function

e

- [EzPlug](#) Utility library
- [EzPlug Tutorial](#) Tutorial show

f

- [Filter Toolbox](#) Filtering toolb
- [FlickrImageRetrieve](#) Grab random

g

- [Generate a bug](#) This plugin generates a bug in a thread, to show the capabilities of the automatic bug report.
- [Generating Bar Chart](#) Tutorial explaining how to display a JFreeChart graph in ICY.
- [Generating Deviation Chart](#) This tutorial explain how to use a deviation renderer JFreeChart chart in ICY.
- [Generating images of differents types](#) This tutorial details how to generate images of all differents types. (byte, int, short, ushort, float, double...
- [GUI design tutorial](#) Details how to create a custom GUI

h

- [Hello world tutorial](#) The first hello world tutorial

i

- [ImageJ Lab](#) Plugin for the ij4icy wrapper. It finds and loads ImageJ plugins from a user-selected folder, and displays a t...

ICY Preferences

General
Repository
Plugin
Local Plugin
Online Plugin
Workspace
Local Workspace
Online Workspace

Repository : default - <http://www.bioimageanalysis.org/icy/repository/>

	Name	Version	
	Active Contours	1.1.0.2	installed
	Animated painter	1.0.0.0	installed
	Anisotropic Filter	1.0.0.2	
	Color Picker Threshold	1.0.0.7	
	Connected Components	1.3.0.0	installed
	Create Simple Area ROI	1.0.0.0	installed
	Diffusion Filters Tool (Sorin)	1.0.0.0	
	EzPlug	1.2.4.0	installed
	EzPlug Tutorial	1.0.0.0	
	Filter Toolbox	1.0.1.0	installed
	FlickrImageRetrieve	1.0.1.0	installed

Reset to default Apply Ok

As a user

Browse plugin

One click install

As a developer

Register online

Submit the jar

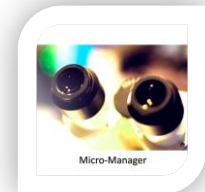
Fill optional info
(dependencies/icon
update/description
tips/tags)

You are ready to be
distributed !

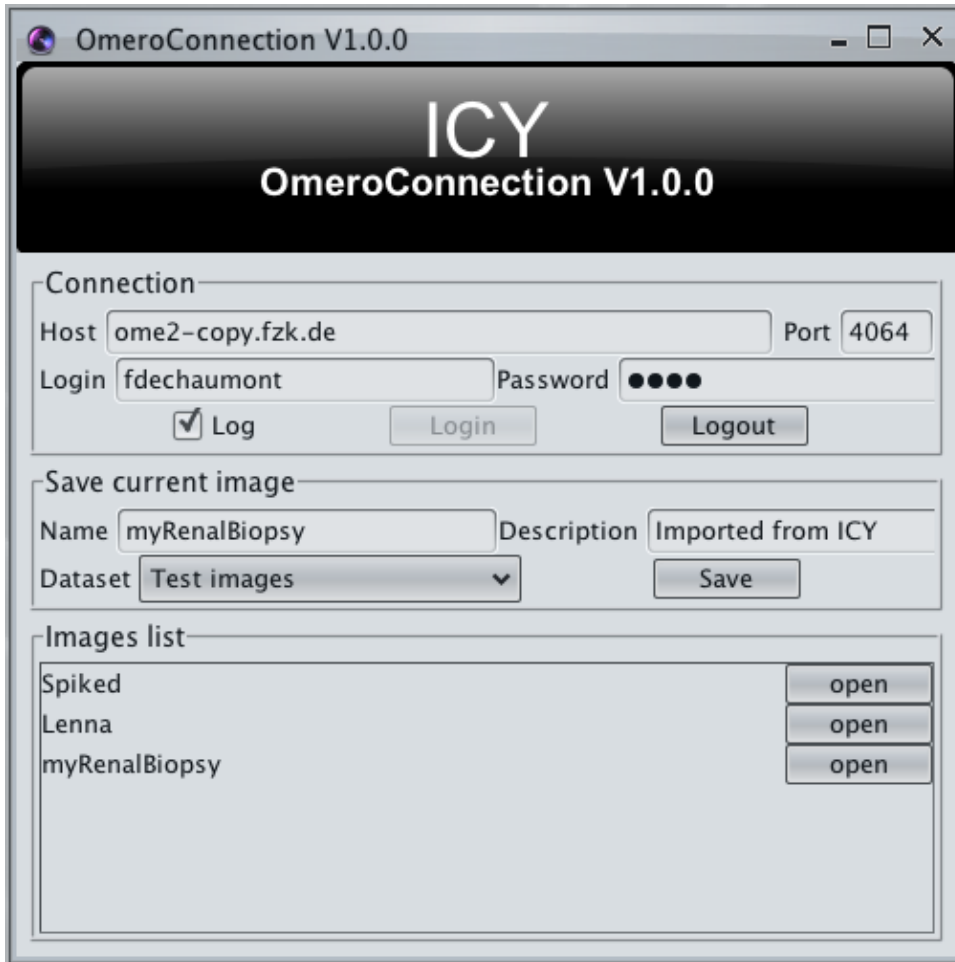
Video

Highly based on libs

- LOCI-Bioformats
- VTK
- Substance / Flamingo / Flanagan
- Jama / JFreeChart / Java Media Framework
JTransforms / JXL / Phys2D / Icy4Ij / JEval
- MicroManager
- !!! OMERO !!!



OMERO & Icy



- We have a load/save plugin working (inspired with the one created for imageJ).
- We wish to integrate transparently the open/save operations with OMERO.
- Plugin in Icy can access meta data of the image, and store their own.
 - Currently in an XML file
 - Could be sent to OMERO database

OMERO & Icy



```
private void load(DatasetData dd) throws DSAccessException, DSOutOfServiceException {
    add(dd);

    long id = dd.getId();

    Collection col = ds.loadImages(id);
    Iterator it = col.iterator();
    DataObject object;
    Class klass = dd.getClass();
    while (it.hasNext()) {
        object = (DataObject) it.next();
        if (object.getClass().equals(klass) && object.getId() == id) {
            if (object instanceof DatasetData) {
                for (Object o : ((DatasetData) object).getImages()) {
                    if (o instanceof ImageData) {
                        add((ImageData) o);
                    } else {
                        logError("Found in load(DatasetData dd) : "
                                + o);
                    }
                }
            }
        }
    }
}
```

```
public void refreshUserData() throws OmeroException {
    ds = ServicesFactory.getInstance().getDataService();

    allImages = new ArrayList<ImageData>();
    allDatasets = new ArrayList<DatasetData>();

    try {
        Collection projects = ds.loadProjects();

        Iterator itp = projects.iterator();
        while (itp.hasNext()) {
            Object o = itp.next();
            if (o instanceof DatasetData) {
                load((DatasetData) o);
            } else if (o instanceof ProjectData) {
                load((ProjectData) o);
            } else {
                logError("Found in refreshImageList() : " + o);
            }
        }

        log("Successful retrieve of user data");
    } catch (DSAccessException e) {
        throw new OmeroException(e);
    } catch (DSOutOfServiceException e) {
        throw new OmeroException(e);
    }
}
```

```

public void importFile(File file, DatasetData dd, String name, String description) throws OmeroException {
    try {
        ImportConfig importConfig = new ImportConfig();

        importConfig.email.set("");
        importConfig.sendFiles.set(false);
        importConfig.sendLogFile.set(false);
        importConfig.sendReport.set(false);
        importConfig.contOnError.set(false);
        importConfig.debug.set(false);
        importConfig.encryptedConnection.set(true);

        importConfig.hostname.set(host);
        importConfig.username.set(login);
        importConfig.password.set(password);
        importConfig.port.set(Integer.parseInt(port));

        importConfig.targetClass.set(Dataset.class.getName());
        importConfig.targetId.set(dd.getId());
        importConfig.imageName.set(name);
        importConfig.imageDescription.set(description);

        OMEROWrapper reader = new OMEROWrapper(importConfig);
        ErrorHandler handler = new ErrorHandler(importConfig);
        String[] paths = new String[]{file.getAbsolutePath()};
        ImportCandidates candidate = new ImportCandidates(reader, paths, handler);

        if (!importConfig.canLogin()) {
            throw new OmeroException("Unable to login");
        }

        OMEROMetadataStoreClient store = importConfig.createStore();
        reader.setMetadataOptions(new DefaultMetadataOptions(MetadataLevel.ALL));
        ImportLibrary importLibrary = new ImportLibrary(store, reader);

        importLibrary.importCandidates(importConfig, candidate);

        if (store != null) {
            store.logout();
        }
    } catch (Exception e) {
        throw new OmeroException(e);
    }
}

```

OMERO & Icy



▼ TestOmero [publication/nherve/ICY/TestOmero]

▼ src

- ▶ ome.formats
- ▶ ome.formats.enums
- ▶ ome.formats.enums.handler
- ▶ ome.formats.importer
- ▶ ome.formats.importer.cli
- ▶ ome.formats.importer.gui
- ▶ ome.formats.importer.gui.gfx
- ▶ ome.formats.importer.reactor
- ▶ ome.formats.importer.util
- ▶ ome.formats.model
- ▶ ome.formats.model.handler
- ▶ ome.parameters
- ▶ plugins.nherve.omero
- ▶ plugins.nherve.omero.data
- ▼ plugins.nherve.omero.test

▼ lib

- commons-httpclient-3.1.jar 4635 07/03/11 12:57 nherve
- commons-io-1.4.jar 4635 07/03/11 12:57 nherve
- commons-logging.jar 4635 07/03/11 12:57 nherve
- ini4j-0.3.2.jar 4635 07/03/11 12:57 nherve
- java-getopt-1.0.13.jar 4635 07/03/11 12:57 nherve
- ome_tools-r7418.jar 4635 07/03/11 12:57 nherve
- ome-io-r7418.jar 4635 07/03/11 12:57 nherve
- ome-java-2007-Aug-07-r3052.jar 4635 07/03/11 12:57 nherve
- omero-client.jar 4635 07/03/11 12:57 nherve
- omero-clients-util-r7223-b1483.jar 4635 07/03/11 12:57 nherve
- omerolibs.jar 4697 18/03/11 17:21 nherve
- org.springframework.core-3.0.1.RELEASE-A.jar 4635 07/03/11 12:57 nherve
- perf4j-0.9.10.jar 4635 07/03/11 12:57 nherve
- swingx-0.9.4.jar 4635 07/03/11 12:57 nherve
- TableLayout-bin-jdk1.5-2009-08-26.jar 4635 07/03/11 12:57 nherve

Thank you !

<http://icy.bioimageanalysis.org>