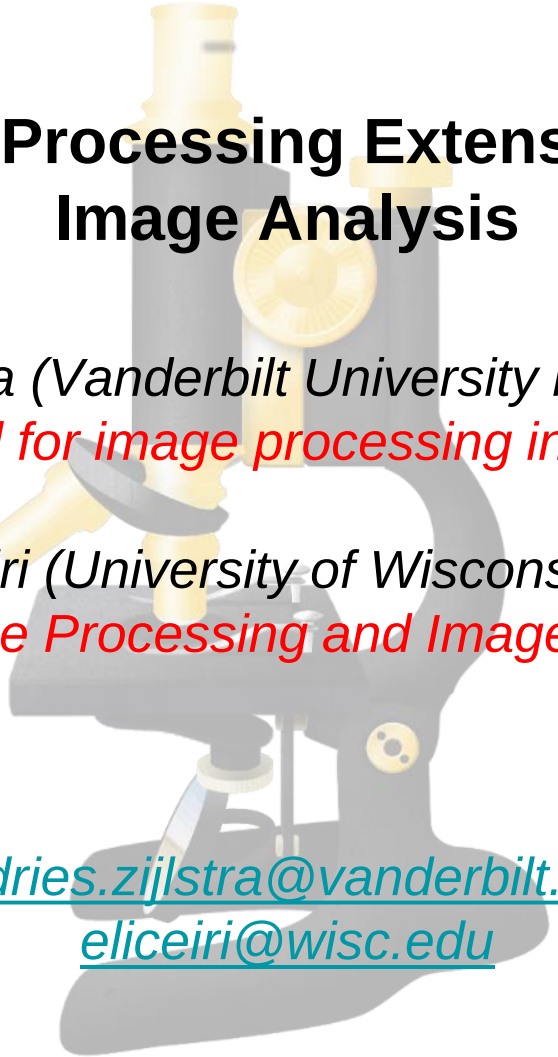
The KNIME Image Processing Extension for Biomedical Image Analysis

Andries Zijlstra (Vanderbilt University Medical Center

The need for image processing in medicine

Kevin Eliceiri (University of Wisconsin-Madison)

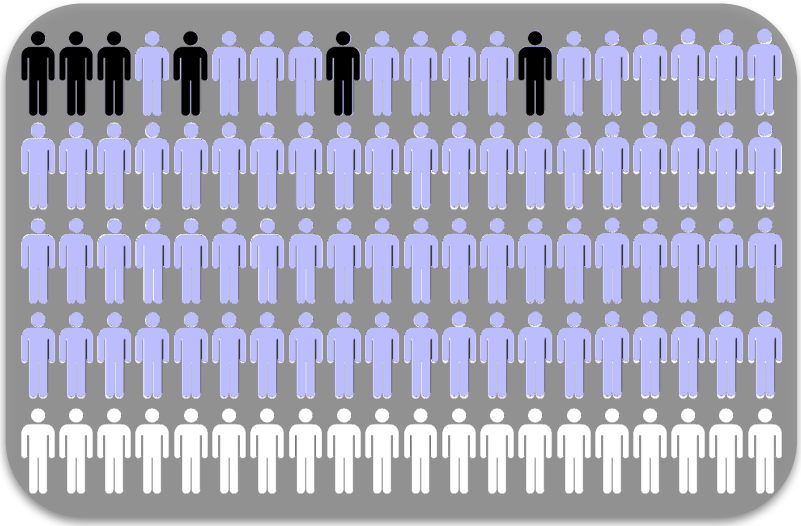
KNIME Image Processing and ImageJ Ecosystem



andries.zijlstra@vanderbilt.edu
eliceiri@wisc.edu

The need for precision oncology

36% of newly diagnosed cancers, and 10% of all cancer deaths in men



The goal: Diagnose patients that have aggressive disease through Precision Medicine

Out of every 100 men...

16 will be diagnosed with prostate cancer in their lifetime

In reality, up to 80 will have prostate cancer by age 70

And 3 will die from it.

But which 3 ?

In the meantime, we over-treat many patients

Objectives of Modern Medicine

- Improved outcome through personalized/precision medicine
- Reduced expense/resource allocation through improved diagnosis, prognosis, treatment
- Maximize quality of life by “targeted” therapy

Approach to Precision Medicine

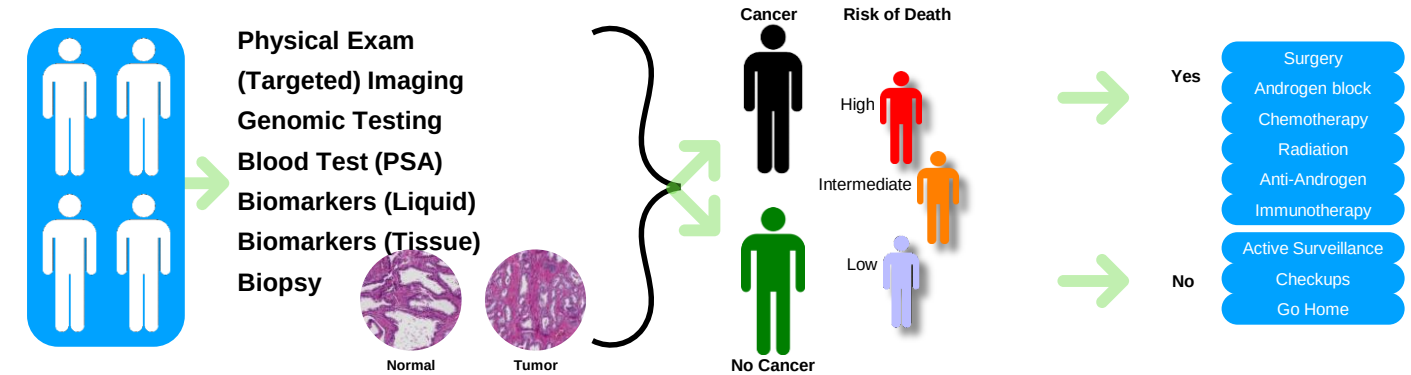
- Measure many things (data density)
- Make very accurate measurements (fidelity)
- Consider multiple perspectives (differential)
- Achieve confidence in the diagnosis
- Match patients with a treatment they are most likely to respond to.

Objectives of Modern Medicine

- Improved outcome through personalized/precision medicine
- Reduced expense/resource allocation through improved diagnosis, prognosis, treatment
- Maximize quality of life by “targeted” therapy

Approach to Precision Medicine

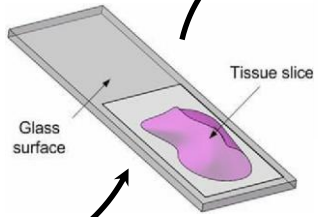
- Measure many things (data density)
- Make very accurate measurements (fidelity)
- Consider multiple perspectives (differential)
- Achieve confidence in the diagnosis
- Match patients with a treatment they are most likely to respond to.



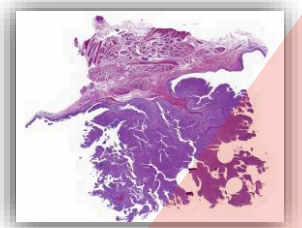
Histopathology in Diagnosis and Prognosis

Tissue Block

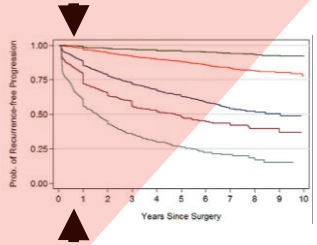
Tissue Section



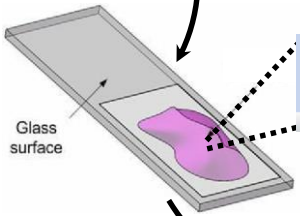
H&E Stain



Pathology Review



IF Stain

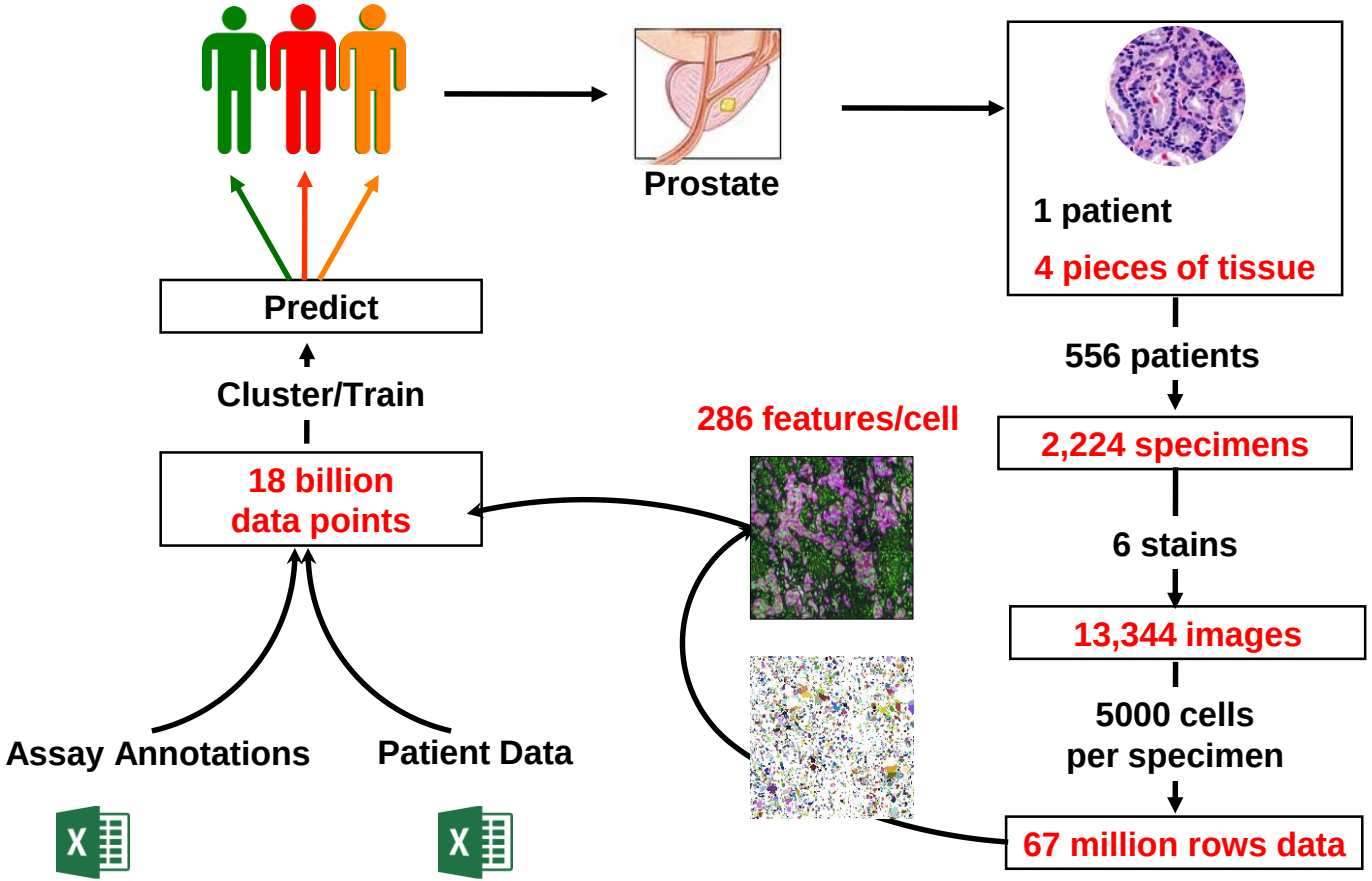


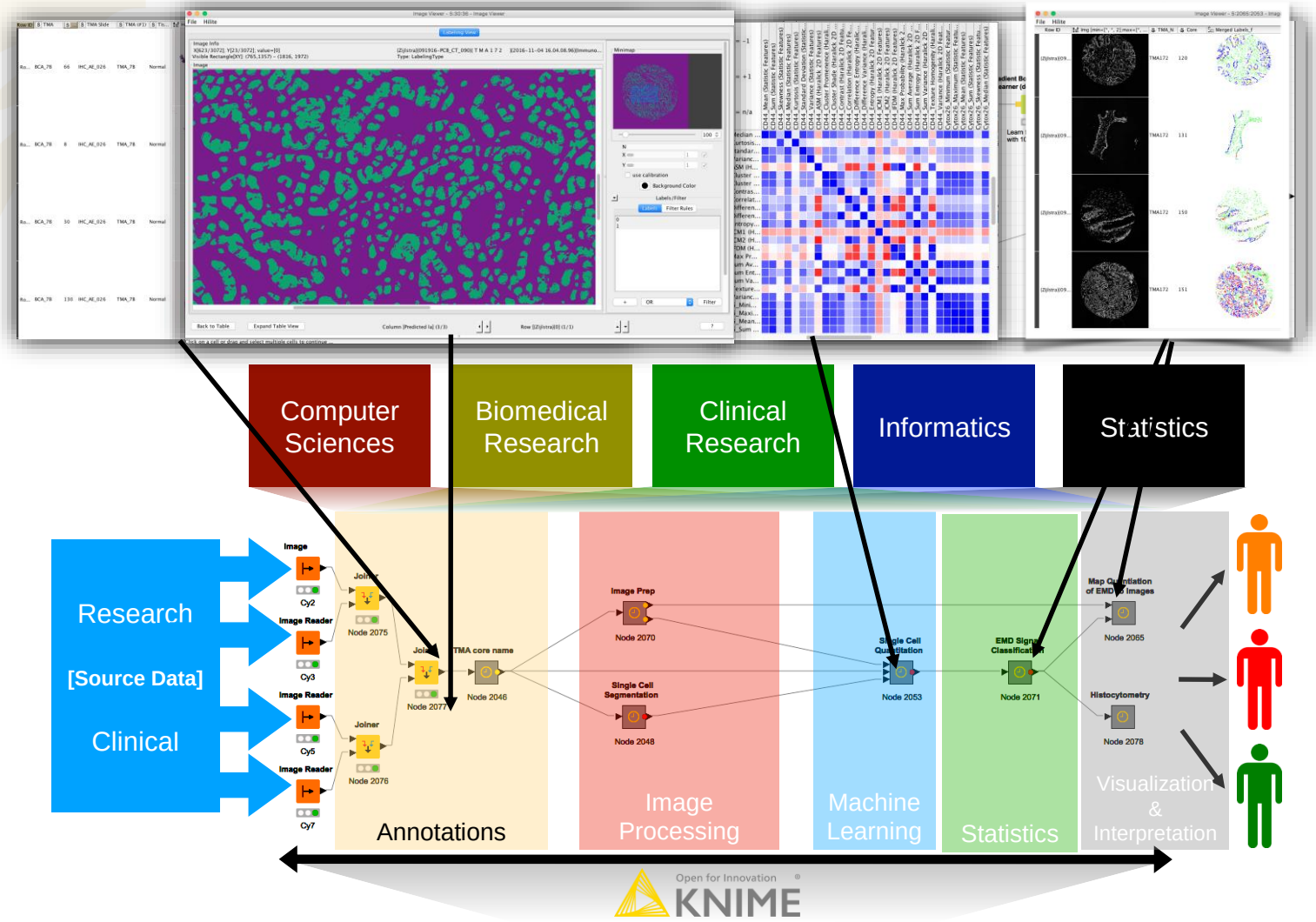
Computer-assisted segmentation, feature extraction and classification

Immunofluorescent Stain

Segmentation/Classification/Clustering

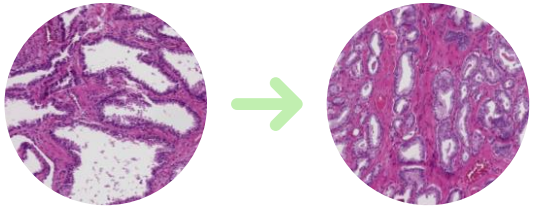
The challenges and opportunities of single-cell analysis

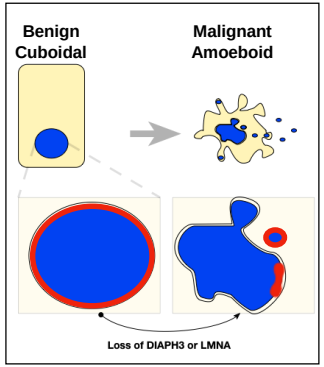



Computer Sciences
Biomedical Research
Clinical Research
Informatics
Statistics
Research
[Source Data]
Clinical
Image
Cy2
Image Reader
Cy3
Image Reader
Cy5
Image Reader
Cy7
Joiner
Node 2075
Joiner
Node 2076
Node 2077
TMA core name
Node 2046
Annotations
Image Prep
Node 2070
Single Cell Segmentation
Node 2048
Image Processing
Single Cell Quantitation
Node 2053
Machine Learning
EMD Signature Classification
Node 2071
Statistics
Map Quantitation of EMD Images
Node 2065
Histocytometry
Node 2078
Visualization & Interpretation
Open for Innovation
KNIME

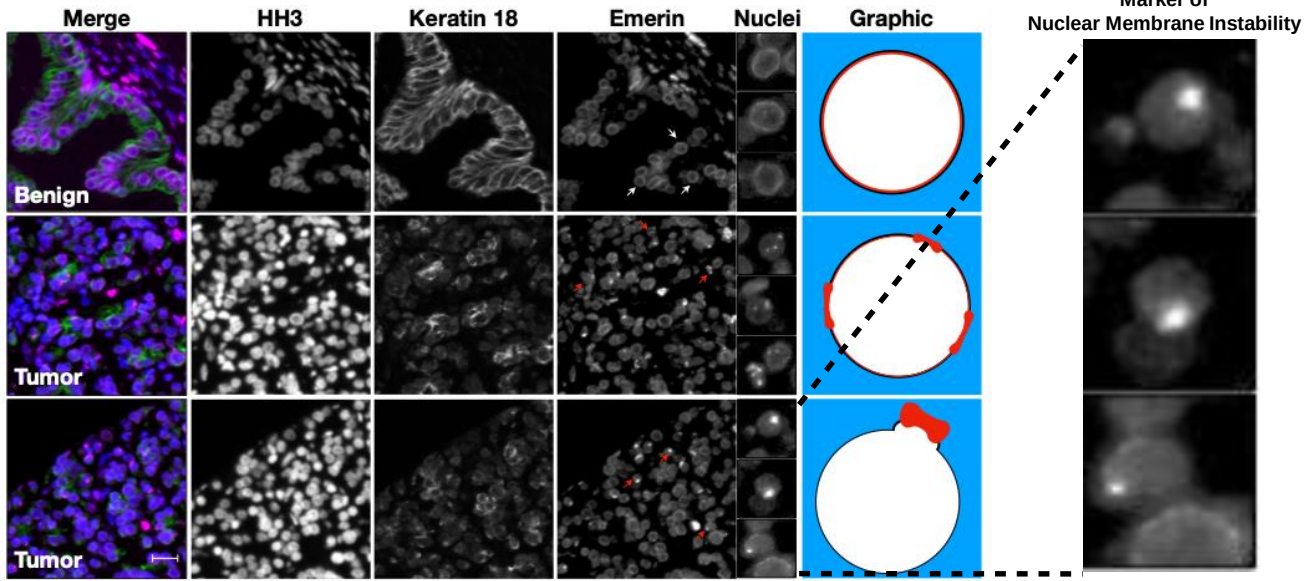
Quantitive detection of aggressive prostate cancer.

Aggressive prostate cancer exhibits nuclear membrane instability (NMI)



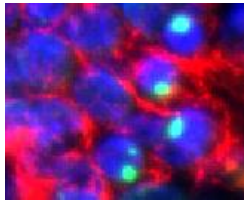


Goal:
300 patients
1200 tissues
6000 images
30,000,000 cells
300 features/cells

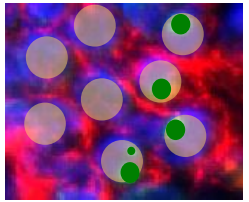


Reis-Sobreiro et. al., Cancer Res. 2018 Aug 28. pii: canres.0608.2018. doi: 10.1158/0008-5472.CAN-18-0608

The NMI image processing pipeline in KNIME



- Label -> Classify ->Quantify



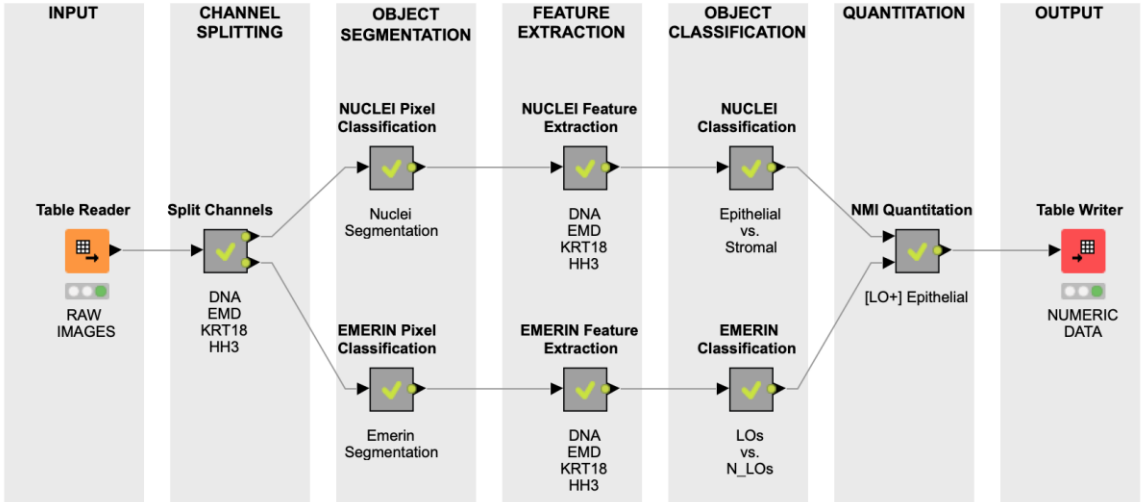
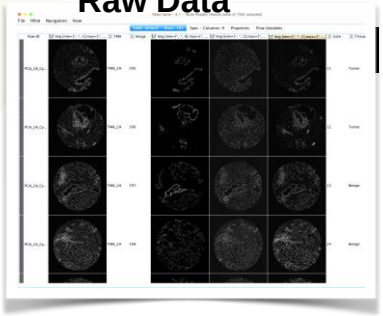
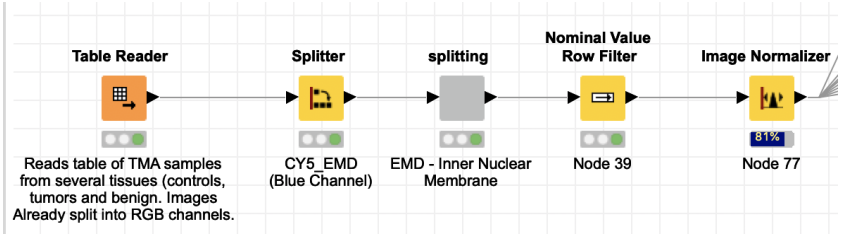
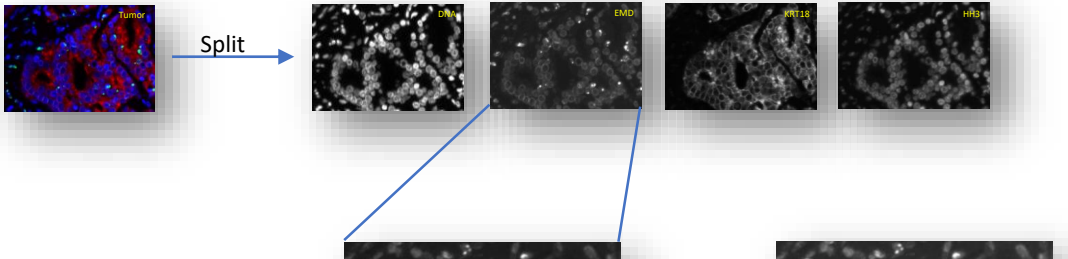


Image Pre-Processing

Raw Data



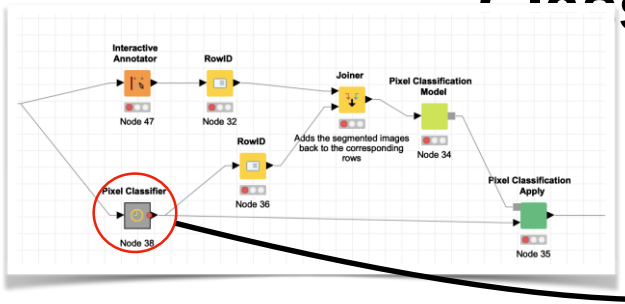


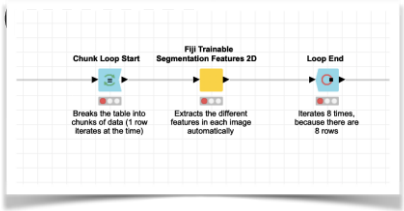


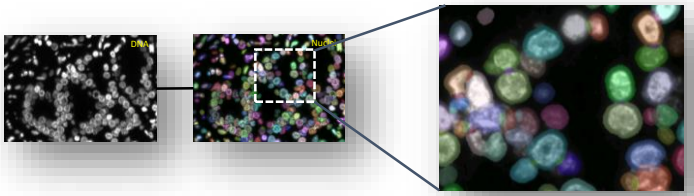
- Process each image with a Normalizer in preparation for Pixel Segmentation
- Intensity range of 0 to 255

INPUT | CHANNEL SPLITTING | OBJECT SEGMENTATION | FEATURE EXTRACTION | OBJECT CLASSIFICATION | QUANTITATION | OUTPUT

Object Segmentation by Pixel Classifi



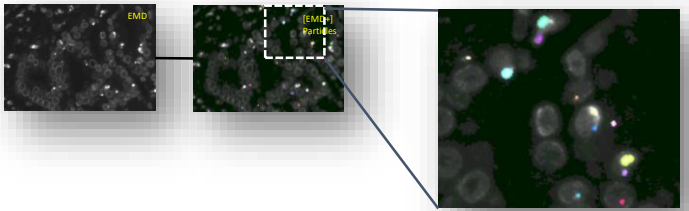




NUCLEI:

Weka Segmentation

Label dilation by 4 pixels

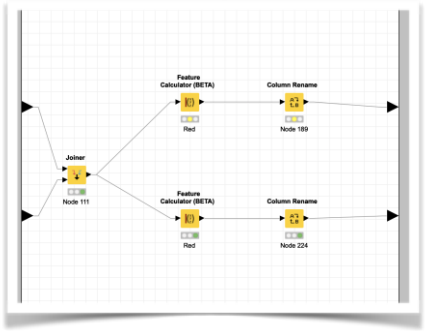


EMERIN:

Ilastik Segmentation

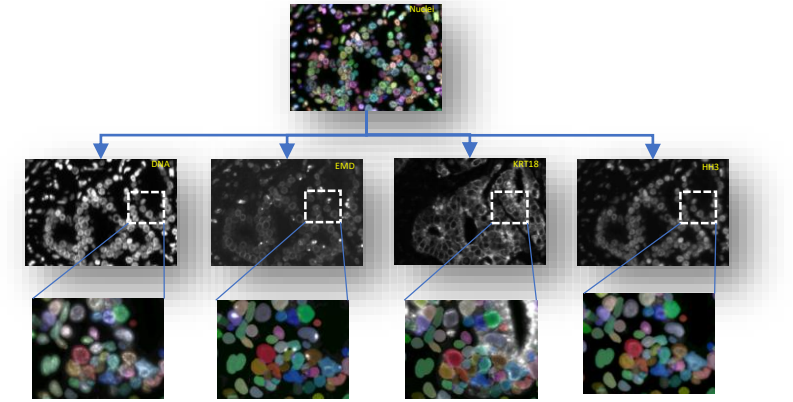
INPUT	CHANNEL SPLITTING	OBJECT SEGMENTATION	FEATURE EXTRACTION	OBJECT CLASSIFICATION	QUANTITATION	OUTPUT

Extracting Features



For Nuclei and Emerin Particles:

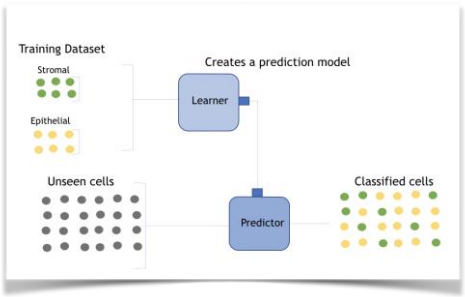
- Overlay labels on each channel (Channels are not normalized, in order to preserve pixel integrity)
- Extract pixel features from labeled areas:
  - Statistic (for intensity)
  - Texture
  - Geometry

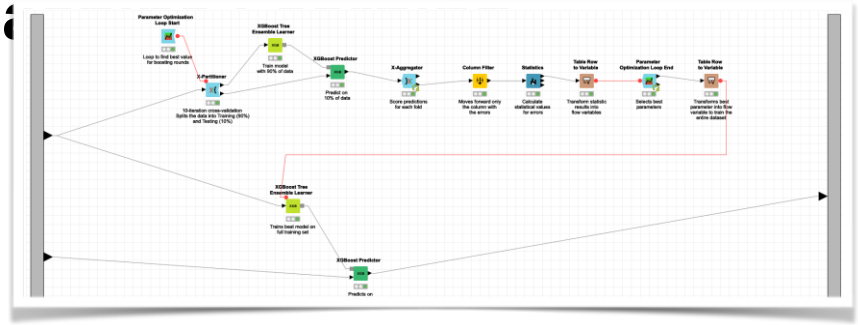


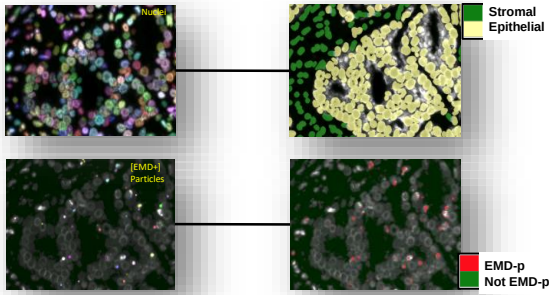
Bitmask	Label	Minimum ...	Maximum ...	Mean ...	Sum ...	Skewness...	Median ...	Kurtosi...	Standard Deviation...
	2049	31	59	44.875	359	0.034	47	1.67	8.855
	2053	15	67	36.125	578	0.438	34	1.935	16.016
	4102	85	127	107.158	2,036	-0.134	107	1.865	12.135
	2056	72	106	85.889	1,546	0.333	88	1.894	10.232
	2060	7	15	11.895	226	-0.401	12	3.275	1.912
	2062	68	110	91.412	1,554	-0.341	92	1.884	12.47
	15	1	2	1.236	68	1.207	1	2.449	0.429
	15	1	2	1.236	68	1.207	1	2.449	0.429
	15	1	2	1.236	68	1.207	1	2.449	0.429
	15	1	2	1.236	68	1.207	1	2.449	0.429

INPUT	CHANNEL SPLITTING	OBJECT SEGMENTATION	FEATURE EXTRACTION	OBJECT CLASSIFICATION	QUANTITATION	OUTPUT

Object Classification through Ma







Features used for classification

Cell type classification (epithelial vs stromal)
Features extracted from
- Statistical and texture features for epithelial and stromal markers
- Geometric measures of the cell

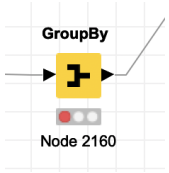
NMI classification
- Statistical and texture features for nuclear markers
- Geometric measures of the NMI

- **Score from 0 to 1**
- **Current cutoff of 0.925**

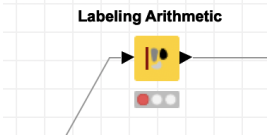
INPUT	CHANNEL SPLITTING	OBJECT SEGMENTATION	FEATURE EXTRACTION	OBJECT CLASSIFICATION	QUANTITATION	OUTPUT

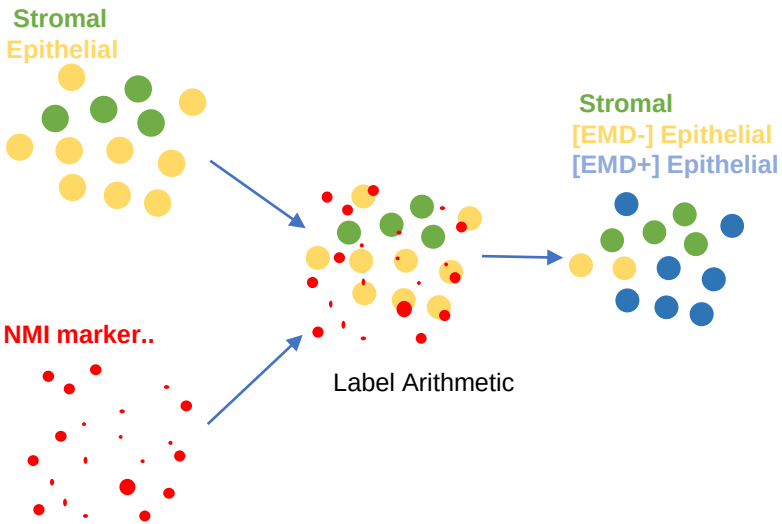
Classifying Cells with NMI

Grouping Labels

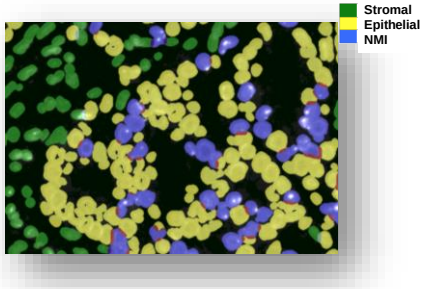


Joining/merging/subtracting Labels

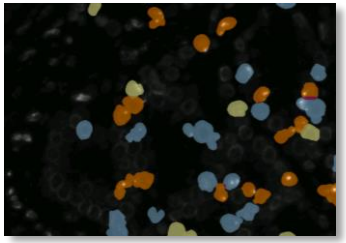




NMI Identification:

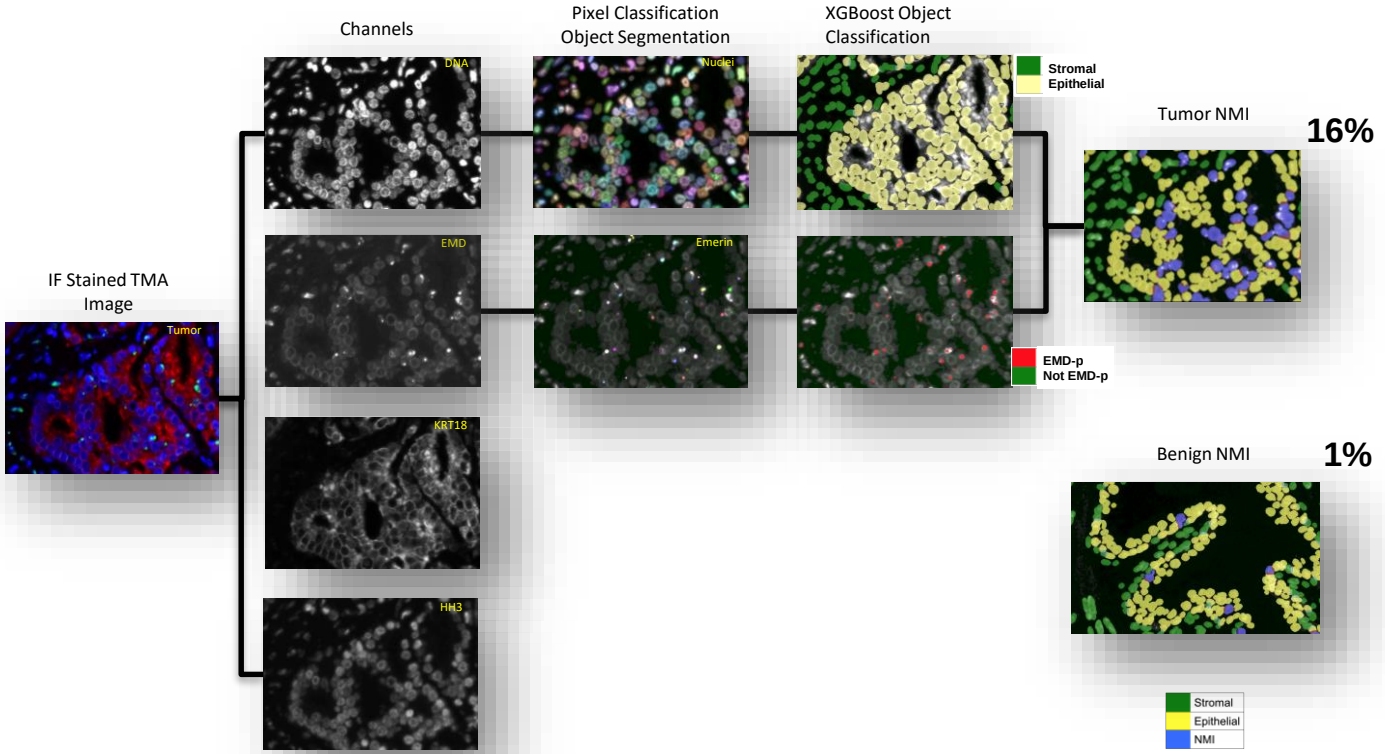


NMI Sub-classification



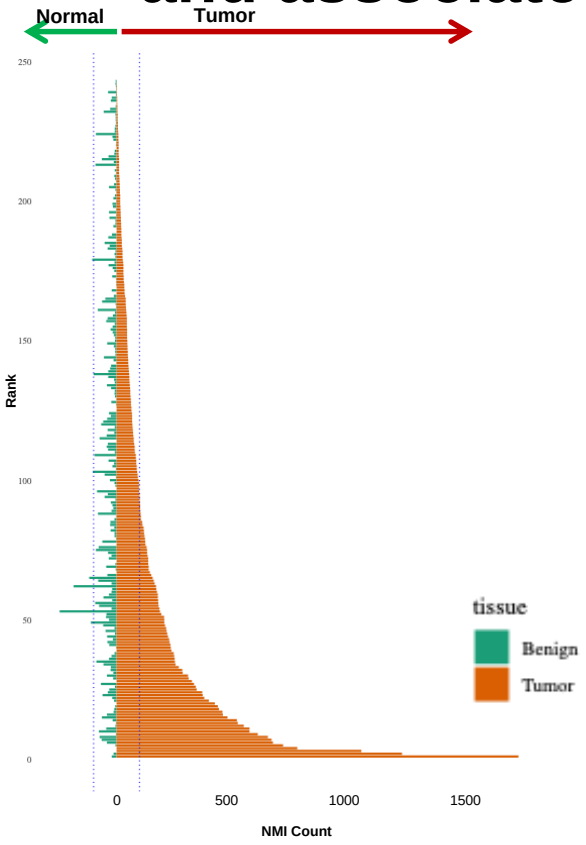
INPUT	CHANNEL SPLITTING	OBJECT SEGMENTATION	FEATURE EXTRACTION	OBJECT CLASSIFICATION	QUANTITATION	OUTPUT

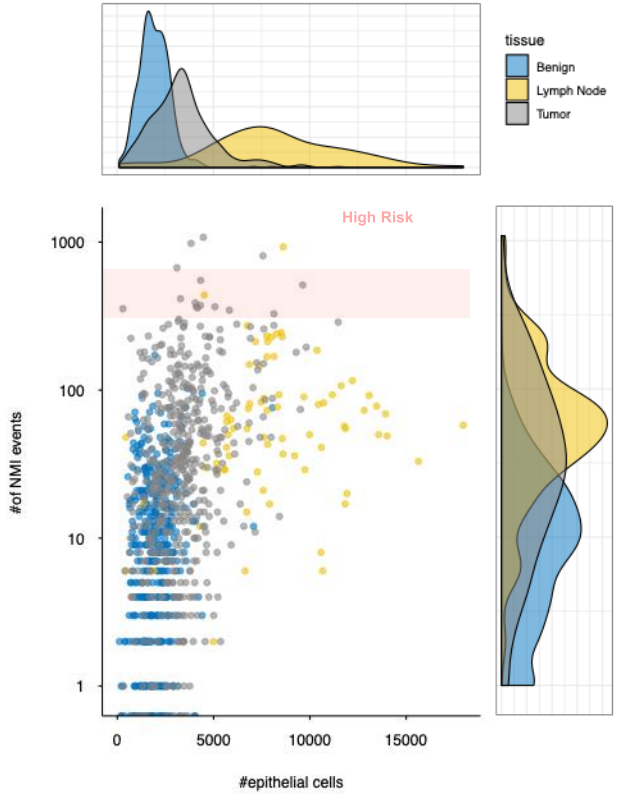
Overview of NMI Classification



Fabiane Fernandes, Tatiana Novitskaya

Nuclear Membrane Instability is overwhelmingly present in tumor tissue and associated with LN metastasis.





KNIME scalability -> Deploy to AWS

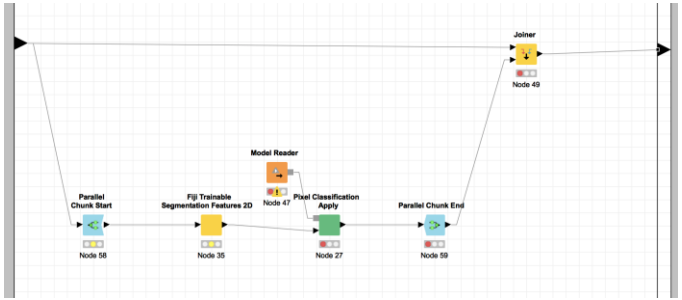
Configuration:
KNIME 3.2
MacPro
12-core
64GB RAM
1TB SSD M2 HD (scratch)
4TB Spindle HD
4-20mb/sec transfer

Configuration:
KNIME 3.2
Ubuntu EC2 r4x4xLarge
16-vCPU
122GB RAM
1TB SSD (scratch)
1001 TB S3
35-400mb/sec transfer

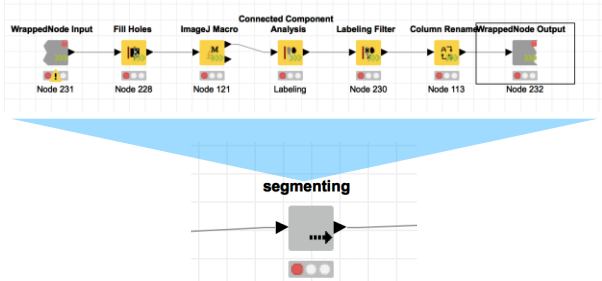
				Local			AWS	
Project	Scenario	Per image	# images	Scratch (GB)	Time (hr)	Problem	Scratch (GB)	Time (hr)
A	1 slide	9.5 megapixel	800	250	12	Slow	20	1
	8 slides	9.5 megapixel	6400	2,000	96	Out of space	200	8
B	1 slide	16 megapixel	800	500	X	No go	100	4
	8 slides	16 megapixel	6400	4,000	X	No go	800	32

Improved performance by a factor of 8-10!!!

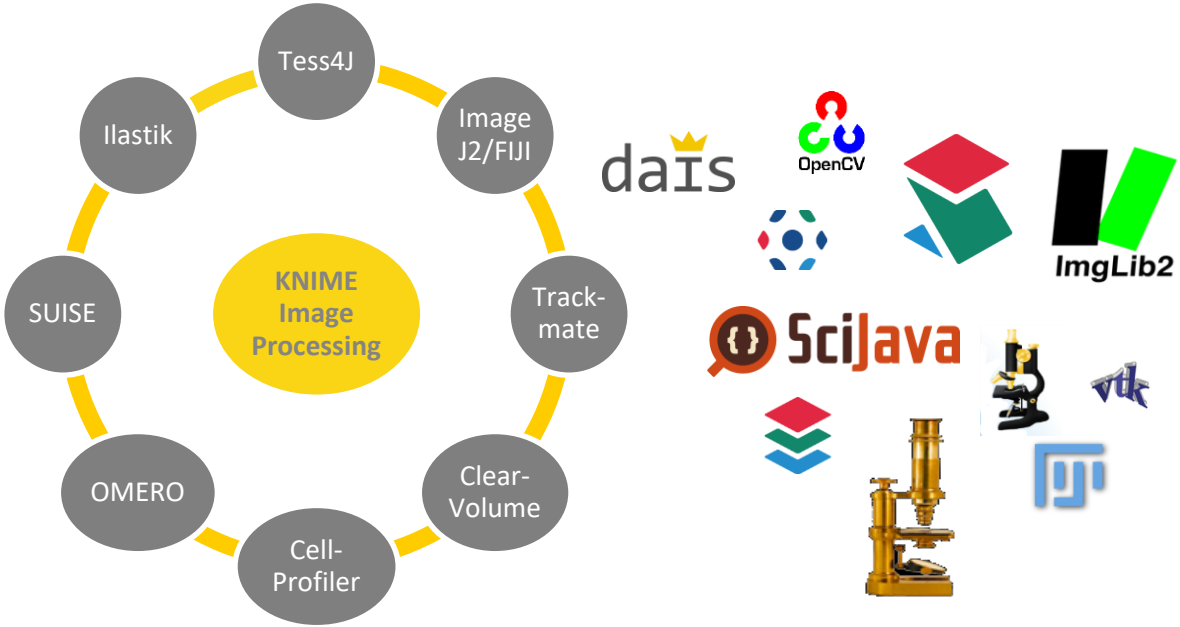
Parallel Chunks



Streaming in wrapped metanodes



Open Source Image Processing- leverage (improve) image community tools

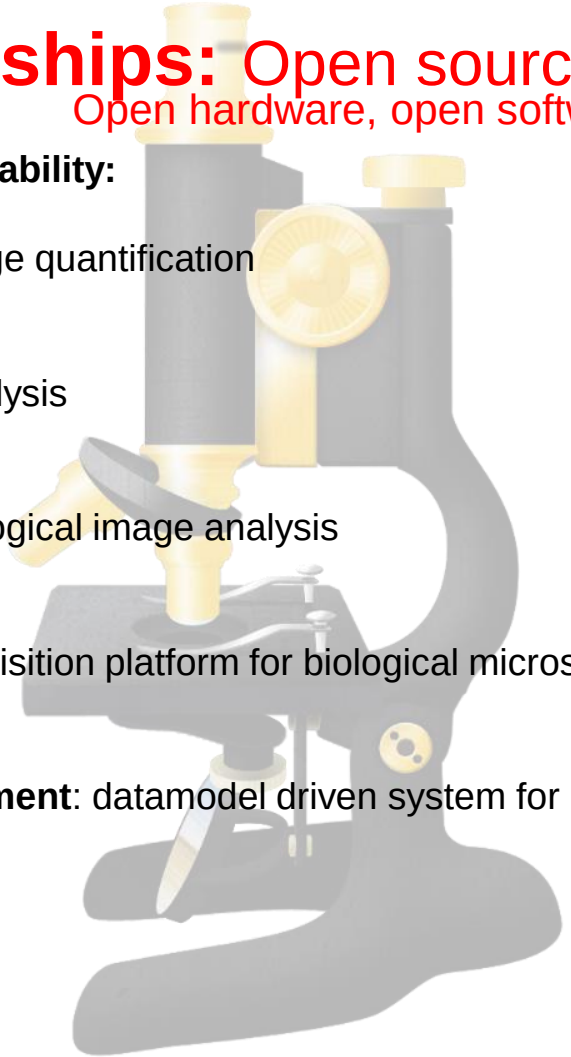


Example Partnerships: Open source Image Informatics

Open hardware, open software

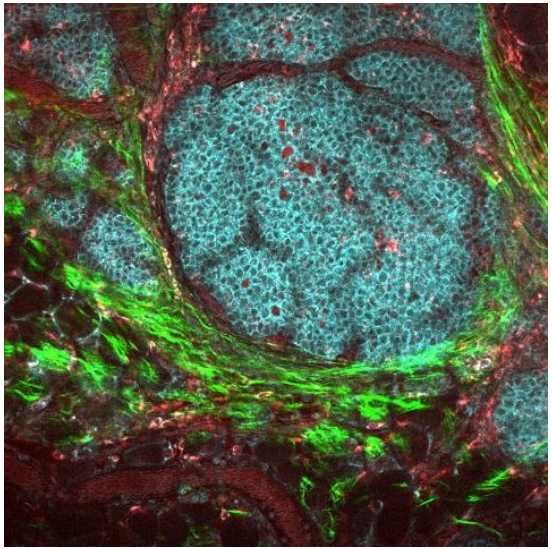
Community Building/Interoperability:

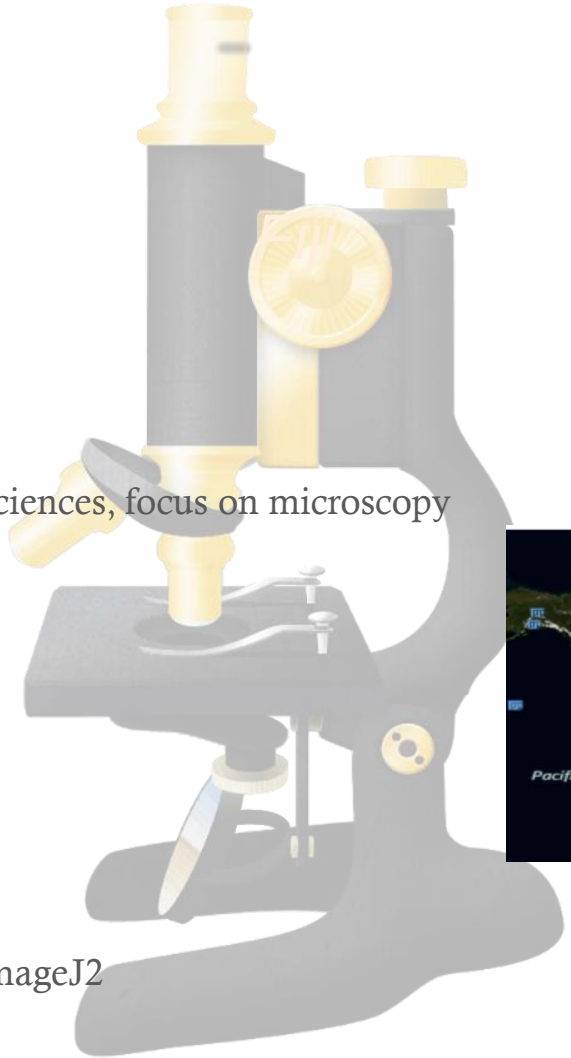
- **Cell Profiler**: automatic image quantification
  www.cellprofiler.org
- **ImageJ and FIJI**: image analysis
  www.imagej.net
- **KNIME:** pipeline tool for biological image analysis
  www.knime.org
- **MicroManage**r: flexible acquisition platform for biological microscopy-*analysis at runtime*
  www.micro-manager.org
- **Open Microscopy Environment**: datamodel driven system for image sharing—*Bio-Formats*
  www.openmicroscopy.org



KNIME Image Processing Plugin (summary)

- reads in more than 120 different kinds of images (thanks to SCIFIO/Bio-Formats API)
- apply well known methods on images, like preprocessing. segmentation, feature extraction, tracking and classification in KNIME.
- In general these nodes operate on multi-dimensional image data which is made possible by the internally used ImgLib2-API.
- Several nodes are available to calculate image features (e.g. zernike-, texture- or histogram features) for segmented images (e.g. a single cell).
- feature vectors can be used to apply machine learning methods in order to train and apply a classifier.
- Currently the Image Processing Plugin for KNIME provides over 100 nodes for (pre)-processing, filtering, segmentation, feature extraction, various views (2D, 3D), etc.
- Active work on bidirectional integration of ImageJ2-allow the users to use directly use/update ImageJ2 Plugins inside KNIME as well as recording and running KNIME Workflows in ImageJ2.





Fiji Is Just ImageJ

A distribution of ImageJ for the life sciences, focus on microscopy

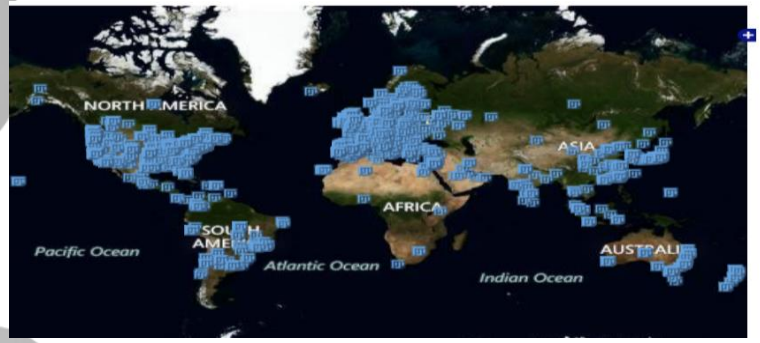
A community of ImageJ developers

Over 15,000 users

Built on the ImageJ2 platform

Includes over 700 new commands

Deployment platform of choice for ImageJ2



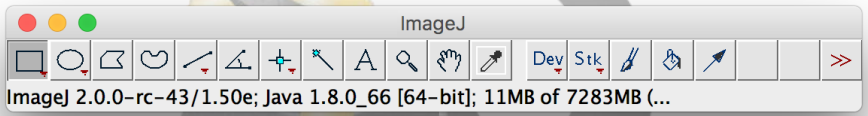
http://fiji.sc/

Mission of ImageJ2

Design	Create the next generation of ImageJ, driven by the needs of the community.
Collaborate	Work together across organizations, fostering open development through sharing + reuse.
Broaden	Make ImageJ useful and relevant across many disciplines of the scientific community.
Maintain	Preserve backwards compatibility with existing ImageJ functionality.
Unify	Provide a central online resource for the ImageJ community.
Lead	Drive ImageJ development forward with a clear vision.

ImageJ2 is…

An end user application



ImageJ2 is…

A reusable library

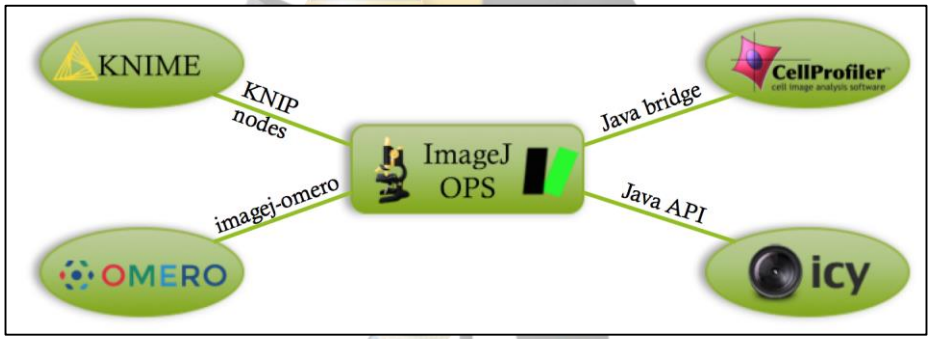
```
import net.imagej.ImageJ
ij = new ImageJ()
image = ij.io().open("image.tif")
ij.ui().show(image)
```

ImageJ2 is…

An extensible collection of services & plugins

ImageJ2 is…

"Write once, run anywhere" image processing



ImageJ2 is… **Modular**

Separation of concerns – each layer has a single focus

SciJava Common – the foundation: plugin framework + application container

ImgLib2 – core N-dimensional data model

SCIFIO/Bio-Formats – Image reading & writing

ImageJ Ops – Image processing algorithms

ImageJ Legacy – Backwards compatibility with ImageJ 1.x

Fiji – A “batteries included” distribution of ImageJ

ImageJ2 ..enhanced power

Built on **ImgLib2** data model

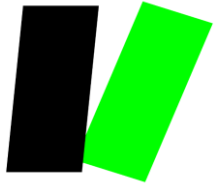
No limit to dimensionality, LUTs, etc…

More data types (see right); extensible

Planes larger than 2 gigapixels

Access data beyond only files on disk

Beyond 1 user, 1 desktop, 1 machine



Name	Bit depth	Signedness	Values	ImageJ1	ImageJ2
bool	1-bit	N/A	boolean	no	yes
bit	1-bit	unsigned	binary	no	yes
uint2	2-bit	unsigned	integer	no	yes
uint4	4-bit	unsigned	integer	no	yes
uint8	8-bit	unsigned	integer	yes	yes
uint12	12-bit	unsigned	integer	no	yes
uint16	16-bit	unsigned	integer	yes	yes
uint32	32-bit	unsigned	integer	no	yes
uint64	64-bit	unsigned	integer	no	yes
uint128	128-bit	unsigned	integer	no	yes
int8	8-bit	signed	integer	no	yes
int16	16-bit	signed	integer	no	yes
int32	32-bit	signed	integer	no	yes
int64	64-bit	signed	integer	no	yes
float32	32-bit	signed	floating point	yes	yes
float64	64-bit	signed	floating point	no	yes
cfloat32	2 × 32-bit	signed	floating point	no	yes
cfloat64	2 × 64-bit	signed	floating point	no	yes
bigint	unlimited	signed	integer	no	yes
bigdec	arbitrary	signed	decimal	no	yes
RGB	3 × 8-bit	unsigned	integer	yes	yes (legacy)
8-bit color	8-bit	indexed	integer	yes	yes (legacy)

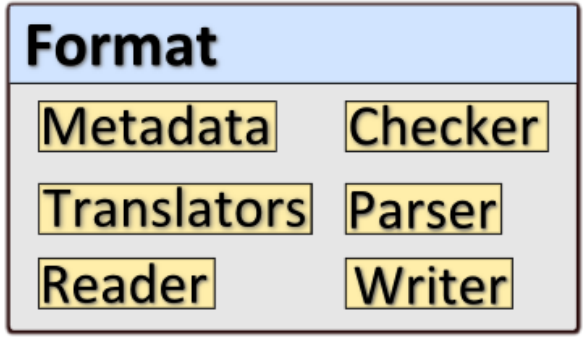
ImageJ2 is… **enhanced format support**

SCIFIO: pluggable data formats + metadata models

Leverages **Bio-Formats (over 150 file formats)**

Writable virtual stacks via caching “cells”

Well-structured image format plugins

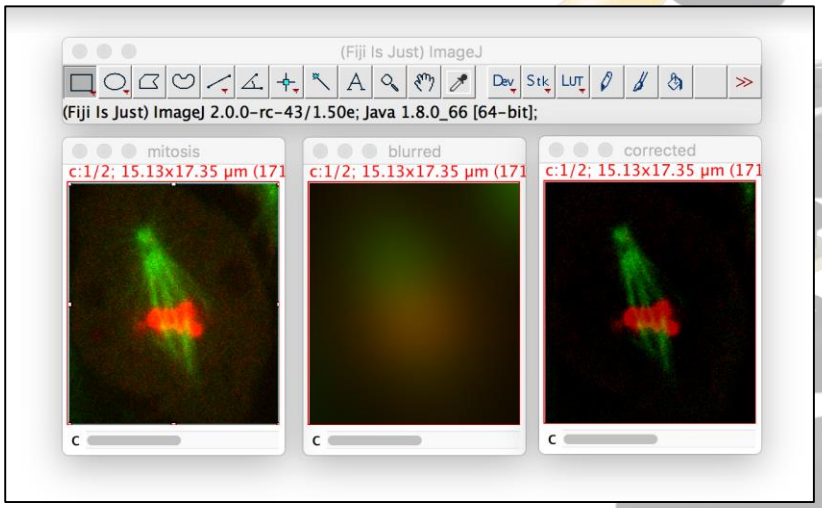


ImageJ2 is… **Powerful**

ImageJ Ops: a unified image processing framework

Powerful and general – built on ImgLib2

Easy to use from scripts (see right)




```
// @OpService op
// @Dataset mitosis
// @Dataset blurred

// subtract numbers
ten = op.math().subtract(70, 60)

// subtract a value from an image
dimmed = op.math().subtract(mitosis, 50)

// subtract two images
corrected = op.math().subtract(mitosis, blurred)

// write your own whiz-bang "math.subtract" ops!
result = op.math().subtract(whiz, bang)
```

ImageJ2 is… **Powerful**

ImageJ Ops: which functions are already available?

- ops
  - (global)
    - eval
    - help
    - identity
    - join
    - loop
    - map
    - op
    - run
    - slicewise
  - convert
    - bit
    - cfloat32
    - cfloat64
    - clip
    - copy
    - float32
    - float64
    - imageType
    - int16
    - int32
    - int64
    - int8
    - normalizeScale
    - scale
    - uint12
    - uint128
    - uint16
    - uint2
    - uint32
    - uint4
    - uint64
    - uint8
- copy
  - img
  - imgLabeling
  - iterableInterval
  - labelingMapping
  - rai
  - type
- create
  - img
  - imgFactory
  - imgLabeling
  - imgPlus
  - integerType
  - kernelGauss
  - kernelLog
  - labelingMapping
  - nativeType
- deconvolve
  - richardsonLucy
  - richardsonLucyTV
- filter
  - addNoise
  - addPoissonNoise
  - convolve
  - correlate
  - dog
  - fft
  - fftSize
  - gauss
  - ifft
  - max
  - mean
  - median
  - min
  - sigma
  - variance
- geom
  - boundarypixelcount
  - boundarypixelcountconvexhull
  - boundarysize
  - boundarysizeconvexhull
  - boundingbox
  - boxivity
  - centerofgravity
  - centroid
  - circularity
  - compactness
  - contour
  - convexhull
  - convexity
  - eccentricity
  - feret
  - feretsangle
  - feretsdiameter
  - mainelongation
  - majoraxis
  - marchingcubes
  - medianelongation
  - minoraxis
  - roundness
  - rugosity
  - secondmultivariate
  - size
  - sizeconvexhull
  - smallestenclosingboundingbox
  - solidity
  - spareness
  - sphericity
  - vertexinterpolator
- haralick
  - asm
  - clusterpromenence
  - clustershade
  - contrast
  - correlation
  - differenceentropy
  - differencevariance
  - entropy
  - icm1
  - icm2
  - ifdm
  - maxprobability
  - sumaverage
  - sumentropy
  - sumvariance
  - texturehomogeneity
  - variance
- image
  - ascii
  - cooccurrencematrix
  - crop
  - equation
  - histogram
  - invert
  - normalize
  - project
  - scale
- imagemoments
  - centralMoment00
  - centralMoment01
  - centralMoment02
  - centralMoment03
  - centralMoment10
  - centralMoment11
  - centralMoment12
  - centralMoment20
  - centralMoment21
  - centralMoment30
  - huMoment1
  - huMoment2
  - huMoment3
  - huMoment4
  - huMoment5
  - huMoment6
  - huMoment7
  - moment00
  - moment01
  - moment10
  - moment11
  - normalizedCentralMoment02
  - normalizedCentralMoment03
  - normalizedCentralMoment11
  - normalizedCentralMoment12
  - normalizedCentralMoment20
  - normalizedCentralMoment21
  - normalizedCentralMoment30
- labeling
  - cca
- lbp
  - lbp2D
- logic
  - and
  - bool
  - equal
  - greaterThan
  - greaterThanOrEqual
  - lessThan
  - lessThanOrEqual
  - not
  - notEqual
  - or
  - xor
- math
  - abs
  - add
  - and
  - arccos
  - arccosh
  - arccot
  - arccoth
  - arccsc
  - arccsch
  - arcsec
  - arcsech
  - arcsin
  - arcsinh
  - arctan
  - arctanh
  - ceil
  - complement
  - cos
  - cosh
  - cot
  - coth
  - csc
  - csch
  - cubeRoot
  - divide
  - exp
  - expMinusOne
  - floor
  - gamma
  - invert
  - leftShift
  - log
  - log10
  - log2
  - max
  - min
  - multiply
  - nearestInt
  - negate
  - or
  - power
  - randomGaussian
  - randomUniform
  - reciprocal
  - remainder
  - rightShift
  - round
  - sec
  - sech
  - signum
  - sin
  - sinc
  - sincPi
  - sinh
  - sqr
  - sqrt
  - step
  - subtract
  - tan
  - tanh
  - ulp
  - unsignedRightShift
  - xor
  - zero
- stats
  - geometricMean
  - harmonicMean
  - kurtosis
  - max
  - mean
  - median
  - min
  - minMax
  - moment1AboutMean
  - moment2AboutMean
  - moment3AboutMean
  - moment4AboutMean
  - percentile
  - quantile
  - size
  - skewness
  - stdDev
  - sum
  - sumOfInverses
  - sumOfLogs
  - sumOfSquares
  - variance
- tamura
  - coarseness
  - contrast
  - directionality
- thread
  - chunker
- threshold
  - apply
  - huang
  - ij1
  - intermodes
  - isoData
  - li
  - localBernsen
  - localMean
  - localMedian
  - localMidGrey
  - localNiblack
  - localPhansalkar
  - localSauvola
  - maxEntropy
  - maxLikelihood
  - mean
  - minError
  - minimum
  - moments
  - otsu
  - percentile
  - renyiEntropy
  - shanbhag
  - triangle
  - yen
- zernike
  - magnitude
  - phase

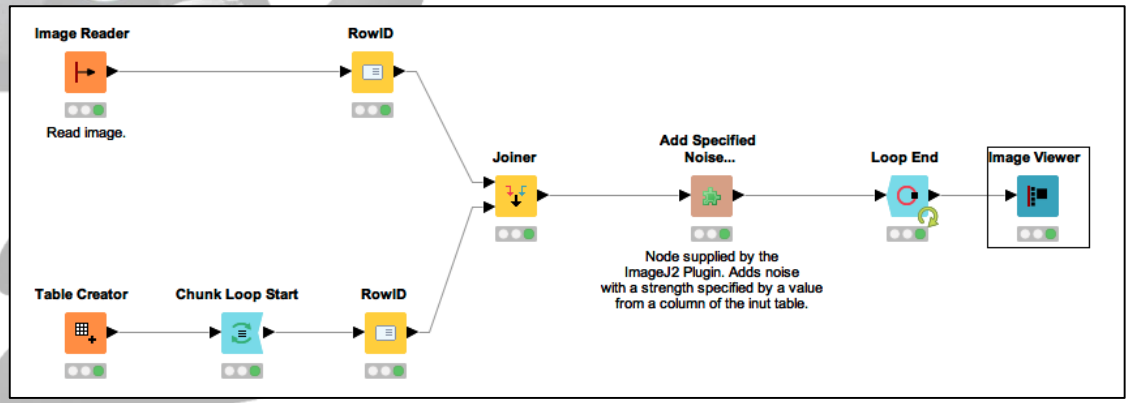
ImageJ2 is… **Interoperable**

ImageJ2 works throughout SciJava!

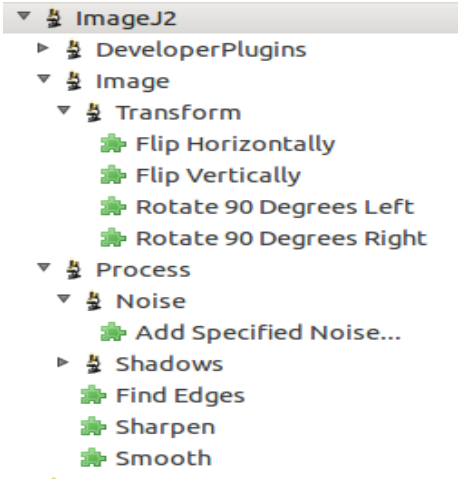
KNIME Image Processing

SCIFIO is used for image reading/writing

KNIME Image Processing is built on ImageJ Ops

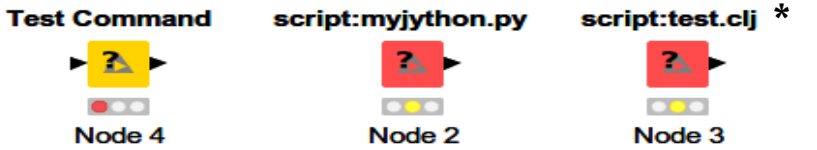


ImageJ2 Integration



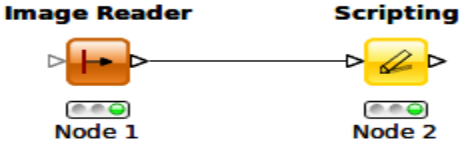
ImageJ2 in KNIME

- “Write once run everywhere”
- Automatic node generation from ImageJ2 plugins
- Add own ImageJ2 Plugins via installer
- Shared algorithm repository (ImageJ-Ops)
- From scripts to nodes*



*Work in Progress

ImageJ2 Plugin Scripting*



Dialog - 0:16 - Scripting(Apply Net)

File

Script Editor | <Autogenerated Tab> | Flow Variables | Memory Policy

Script Editor — Language: Java

Column: Image

```
import net.imagej.ImgPlus;
import net.imglib2.Cursor;
import net.imglib2.img.Img;
import net.imglib2.type.numeric.RealType;

import org.scijava.ItemIO;
import org.scijava.command.Command;
import org.scijava.plugin.Parameter;

public class MyClass<T extends RealType<T>> implements Command {

    @Parameter(type = ItemIO.INPUT)
    private ImgPlus<T> in_img;

    @Parameter(type = ItemIO.OUTPUT)
    private ImgPlus<T> out_img;

    public void run() {
        Img<T> in = in_img.getImg();
        T t = in.cursor().next();
        .
        // create output image with same dimensions
        Img<T> img = in.factory().create(in, t);

        Cursor<T> cin = in.cursor();
        Cursor<T> cout = img.cursor();

        int count = 0; // cur index

        // set every other pixel to black
        while (cout.hasNext() && cin.hasNext()) {
            // set pixel of cout to pixel in
            cout.next().setReal(cin.next().getRealDouble() * (count & 1));

            count++;
        }

        out_img = new ImgPlus<T>(img);
        return;
    }
}
```

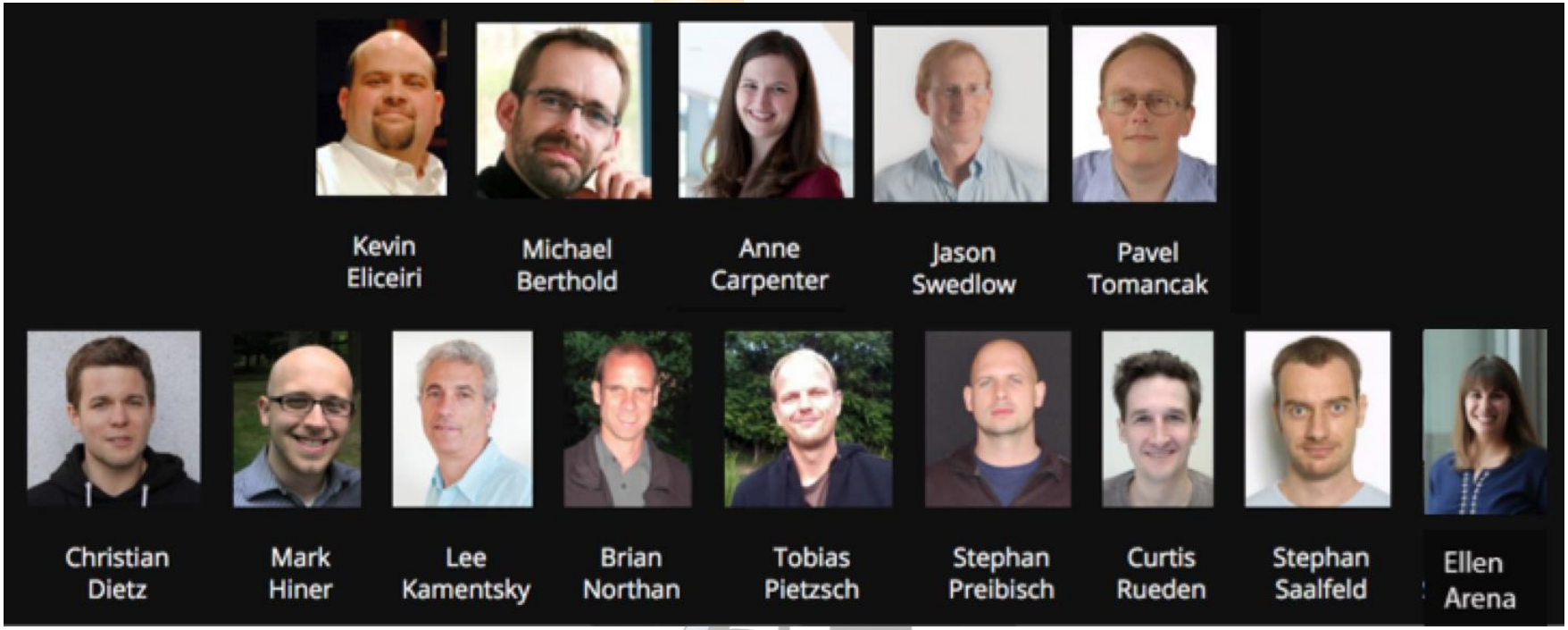
Column/Input Matchings: + -

Column	Active	Input
Image	☑	in_img

OK | Apply | Cancel

*Work in Progress

The SciJava consortium



And everyone supporting open science and open software!

The forum: http://forum.image.sc
Active and vibrant image analysis community discussion