GraspJ - GPU-Run Analysis for STORM and PALM in ImageJ

AFIB - ICFO

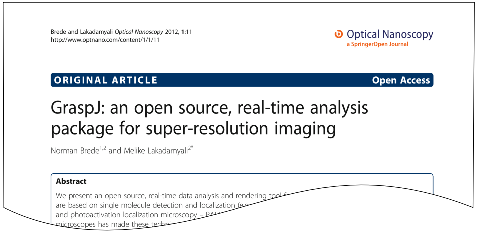
Ismael Benito Altamirano

July 17, 2015

Index

Index

Introducing GraspJ



Brede and Lakadamyali *Optical Nanoscopy* 2012, **1**:11
http://www.optnano.com/content/1/1/11

Optical Nanoscopy
a SpringerOpen Journal

ORIGINAL ARTICLE **Open Access**

GraspJ: an open source, real-time analysis package for super-resolution imaging

Norman Brede[1,2] and Melike Lakadamyali[2*]

Abstract

We present an open source, real-time data analysis and rendering tool
are based on single molecule detection and localization (e
and photoactivation localization microscopy – PAL
microscopes has made these techni

GraspJ: an open source,real-time analysis package for super-resolution imaging
Norman Brede and Melike Lakadamyali

Was developed by Norman Brede at AFIB as his master thesis. It is thougth:

- As a plugin of ImageJ or FIJI.
- To be able to run in a GPU based system.
- To be able to run in realtime with Labview.
- To be open source.

Dependencies & Modes

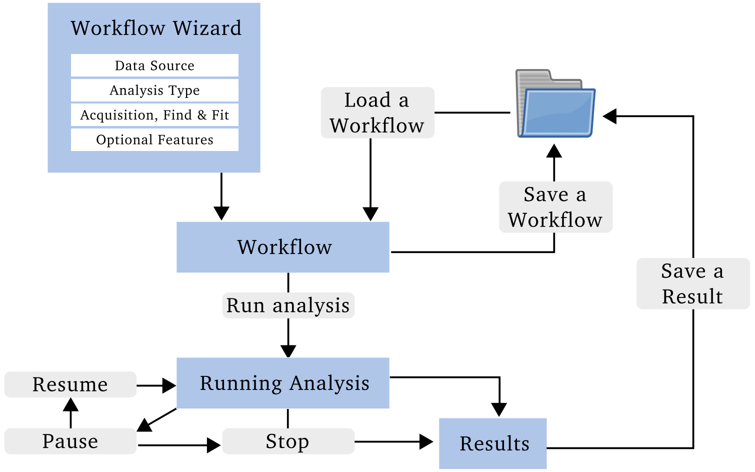
Dependencies

- *Java Virtual Machine (JVM):* needs at least Version 7.
- *OpenCL:* Open Computing Language, allows the possibility to compute in GPUs. Needs compiler and Drivers.

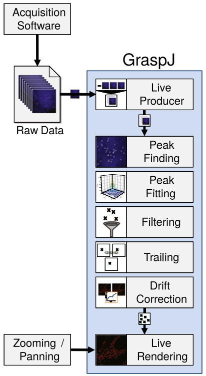
Modes

- *Standalone Mode:* Runs indepently from ImageJ/FIJI. Less realtime features. Easier to run.
- *Plugin Mode:* Runs inside ImageJ/FIJI. More realtime features. JVM versions issues.

Common use


Workflow Wizard
Data Source
Analysis Type
Acquisition, Find & Fit
Optional Features
Load a Workflow
Save a Workflow
Workflow
Save a Result
Run analysis
Resume
Running Analysis
Pause
Stop
Results

Workflow details

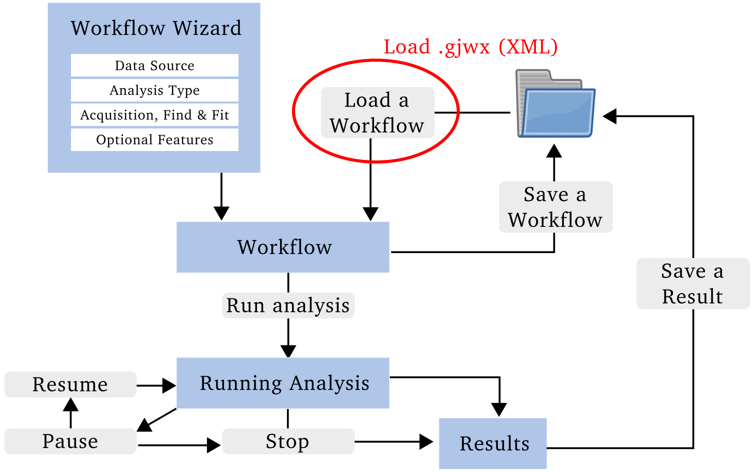


Norman Brede and Melike Lakadamyali.

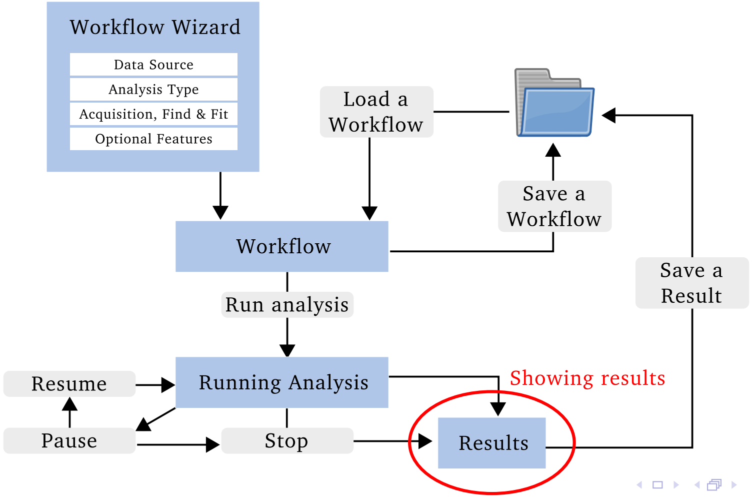
- *Live Producer:* creates the packages of frames.
- *Peak Finding:* Peaks are identified based on a threshold.
- *Peak Filtering:* Mean positions are determined
- *Filtering:* Localizations resulting from peak fitting can be filtered.
- *Trailing:* Peaks that appear in close proximity in consecutive frames are trailed together.
- *Drift correction:* Drift is determined by phase correlating consecutive, temporally split high resolution images.
- *Live Rendering:* Final localizations are rendered as Gaussians.

Index

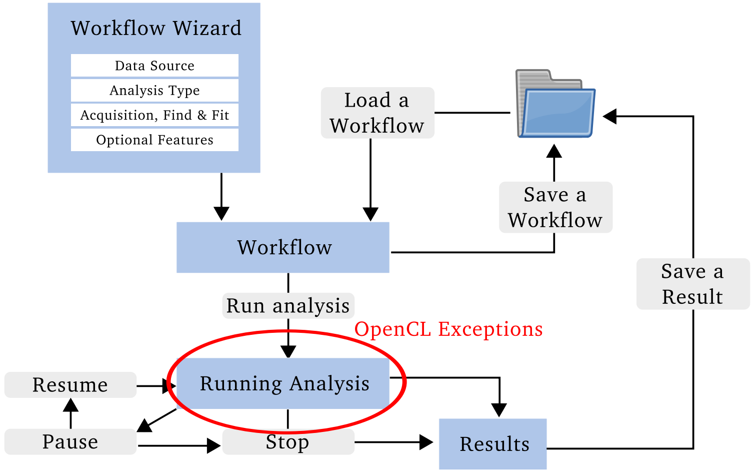
Load Error


Workflow Wizard
Data Source
Analysis Type
Acquisition, Find & Fit
Optional Features
Load .gjwx (XML)
Load a Workflow
Save a Workflow
Workflow
Save a Result
Run analysis
Resume
Running Analysis
Pause
Stop
Results

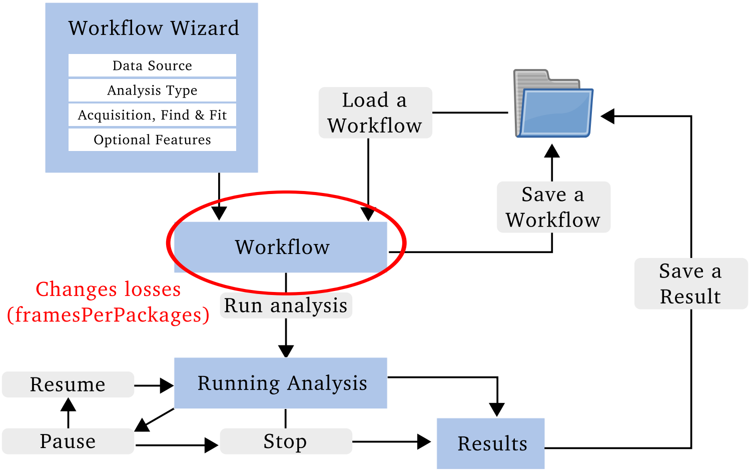
Result Error



OpenCL Exception

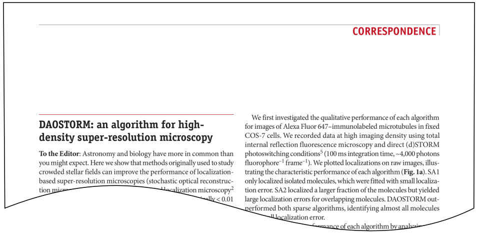


Workflow “Mistake”



Index

DAOSTORM



CORRESPONDENCE

DAOSTORM: an algorithm for high-density super-resolution microscopy

To the Editor: Astronomy and biology have more in common than you might expect. Here we show that methods originally used to study crowded stellar fields can improve the performance of localization-based super-resolution microscopies (stochastic optical reconstruction mic [illegible] localization microscopy[2] [illegible] lly < 0.01

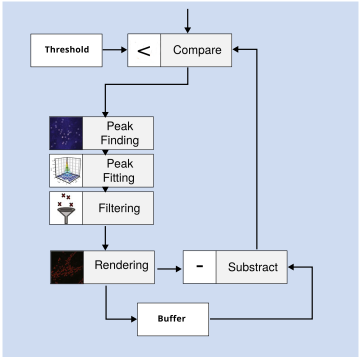
We first investigated the qualitative performance of each algorithm for images of Alexa Fluor 647–immunolabeled microtubules in fixed COS-7 cells. We recorded data at high imaging density using total internal reflection fluorescence microscopy and direct (d)STORM photoswitching conditions[5] (100 ms integration time, ~4,000 photons fluorophore^{-1} frame^{-1}). We plotted localizations on raw images, illustrating the characteristic performance of each algorithm (**Fig. 1a**). SA1 only localized isolated molecules, which were fitted with small localization error. SA2 localized a larger fraction of the molecules but yielded large localization errors for overlapping molecules. DAOSTORM outperformed both sparse algorithms, identifying almost all molecules [illegible] ll localization error.

[illegible] formance of each algorithm by analyzi

DAOSTORM: an algorithm for high-density super-resolution microscopy

Seamus J Holden, Stephan Uphoff & Achillefs N Kapanidis

Was developed forking DAOPHOT II algotrithm from star detections. The basis of the algorithm are:


Threshold
<
Compare
Peak Finding
Peak Fitting
Filtering
Rendering
-
Substract
Buffer