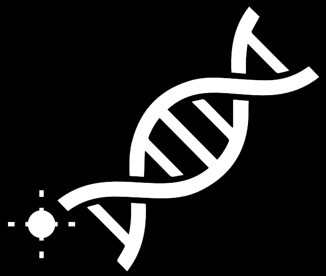
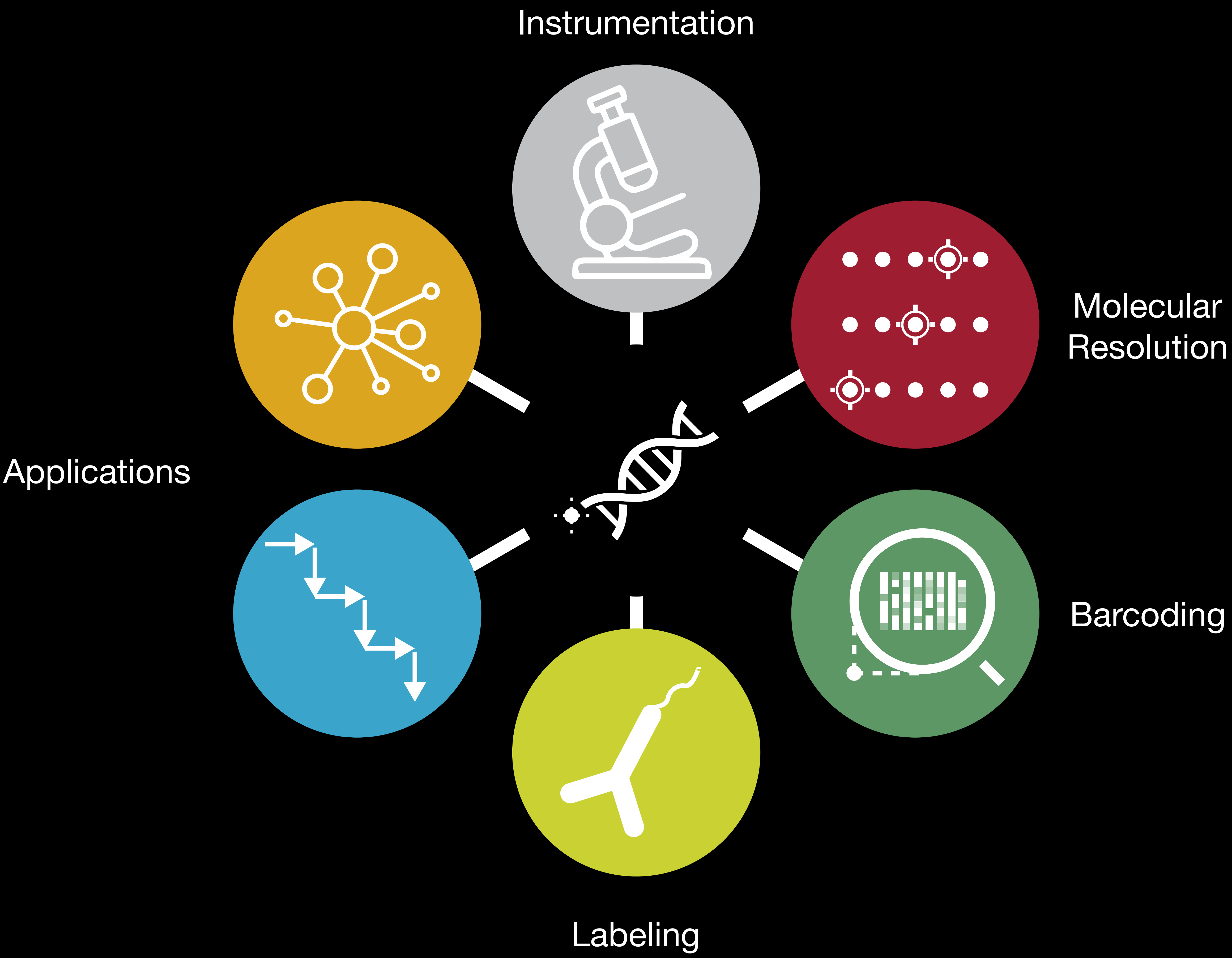
Localizomics

Super-resolution microscopy with DNA molecules

Localizomics using DNA probes

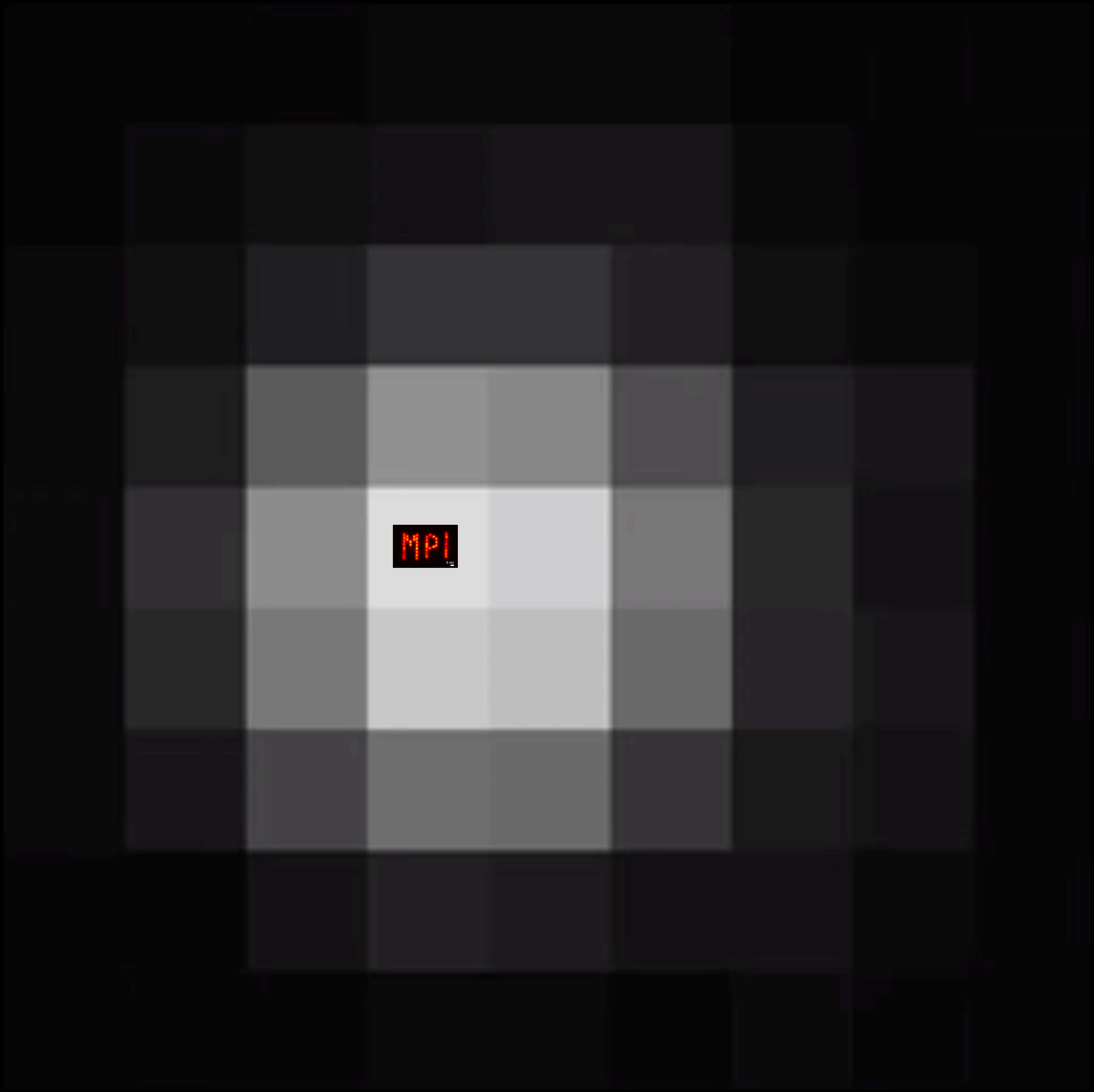


Localizomics using DNA probes



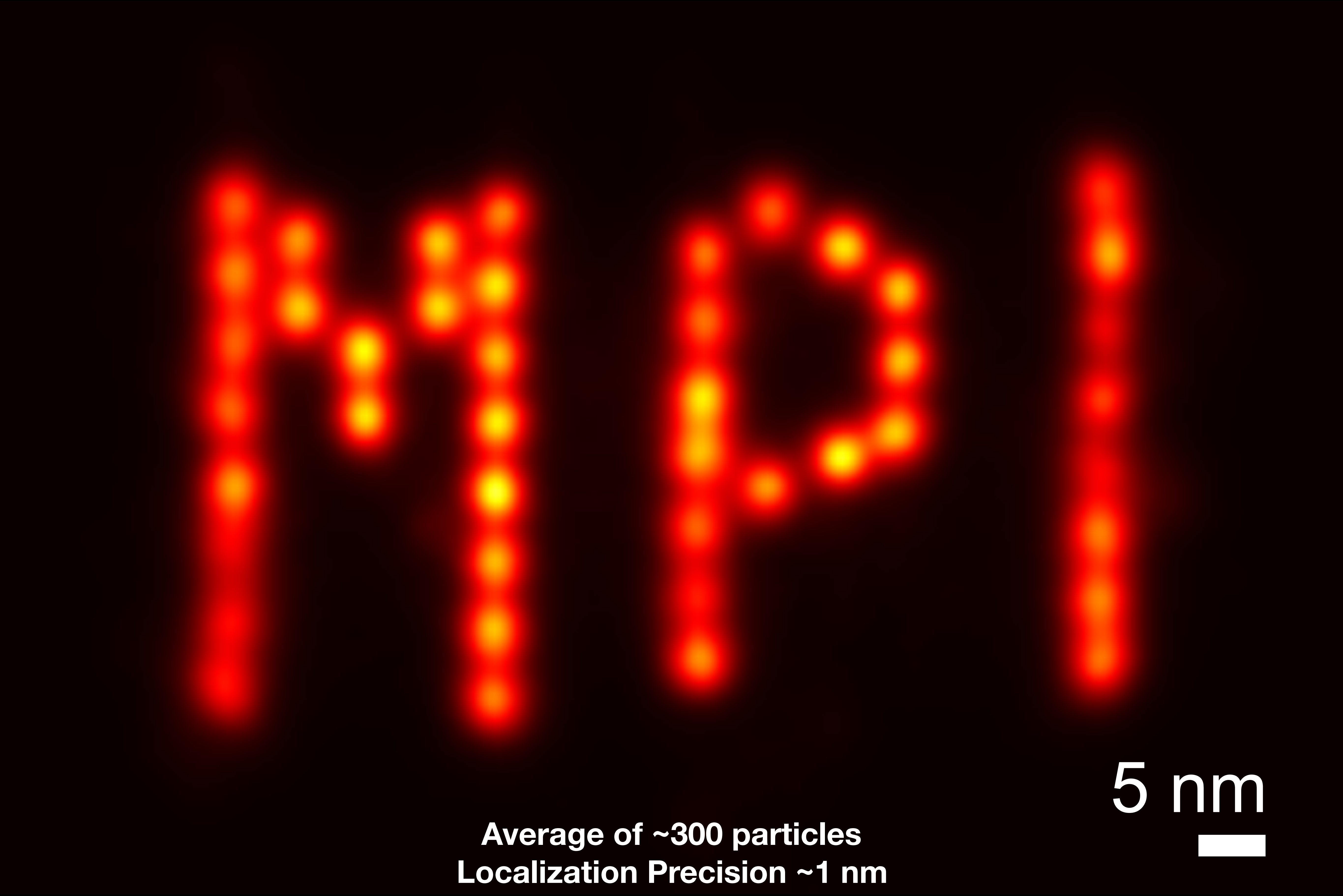
This is how it’s done

Molecular Resolution



Schnitzbauer*, Strauss*, Schlichthaerle, Schueder, Jungmann, ***Nature Protocols***, 2017

Molecular Resolution

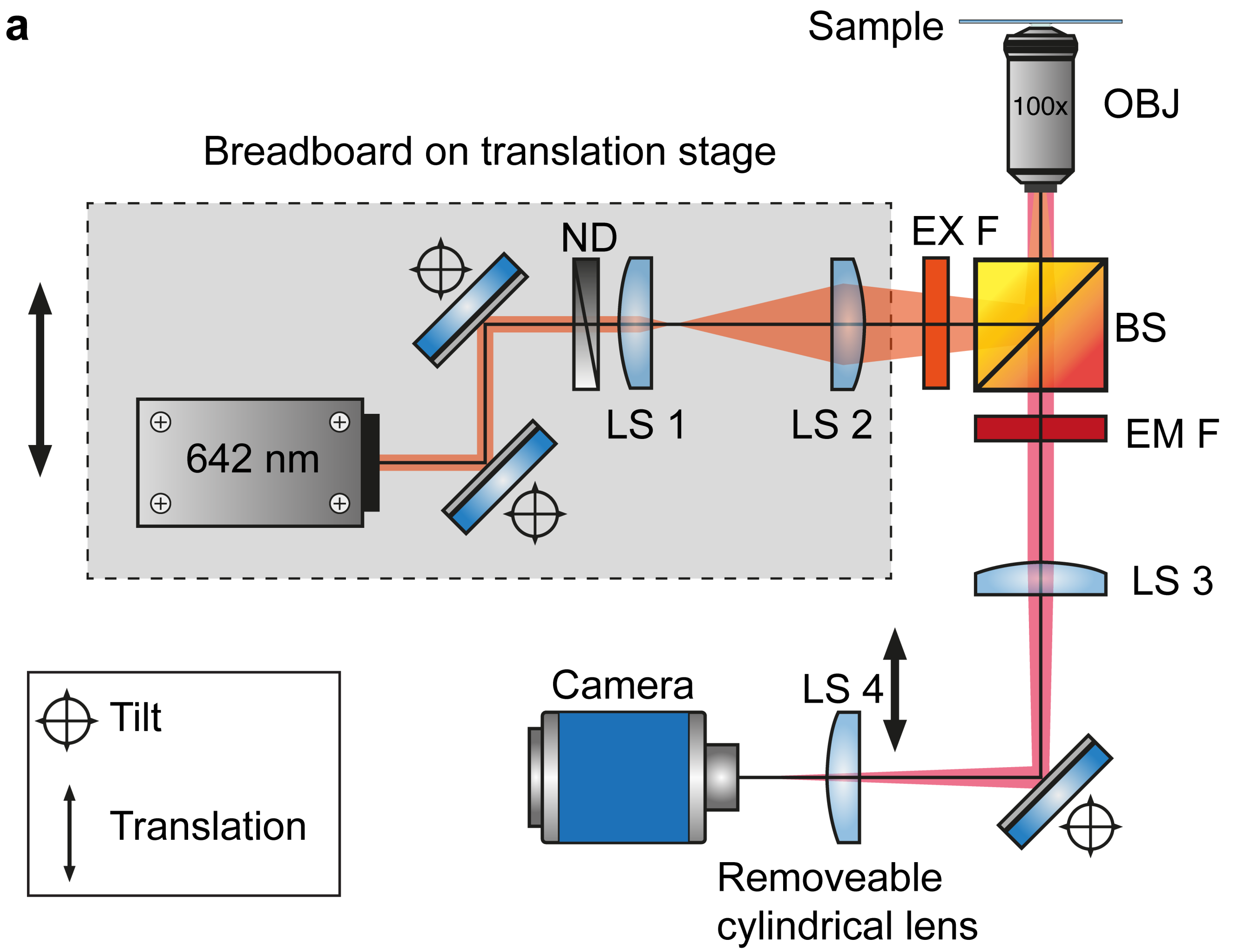


Average of ~300 particles
Localization Precision ~1 nm

Schnitzbauer*, Strauss*, Schlichthaerle, Schueder, Jungmann, ***Nature Protocols***, 2017

What’s “Open Source”?

liteTIRF.jungmannlab.org

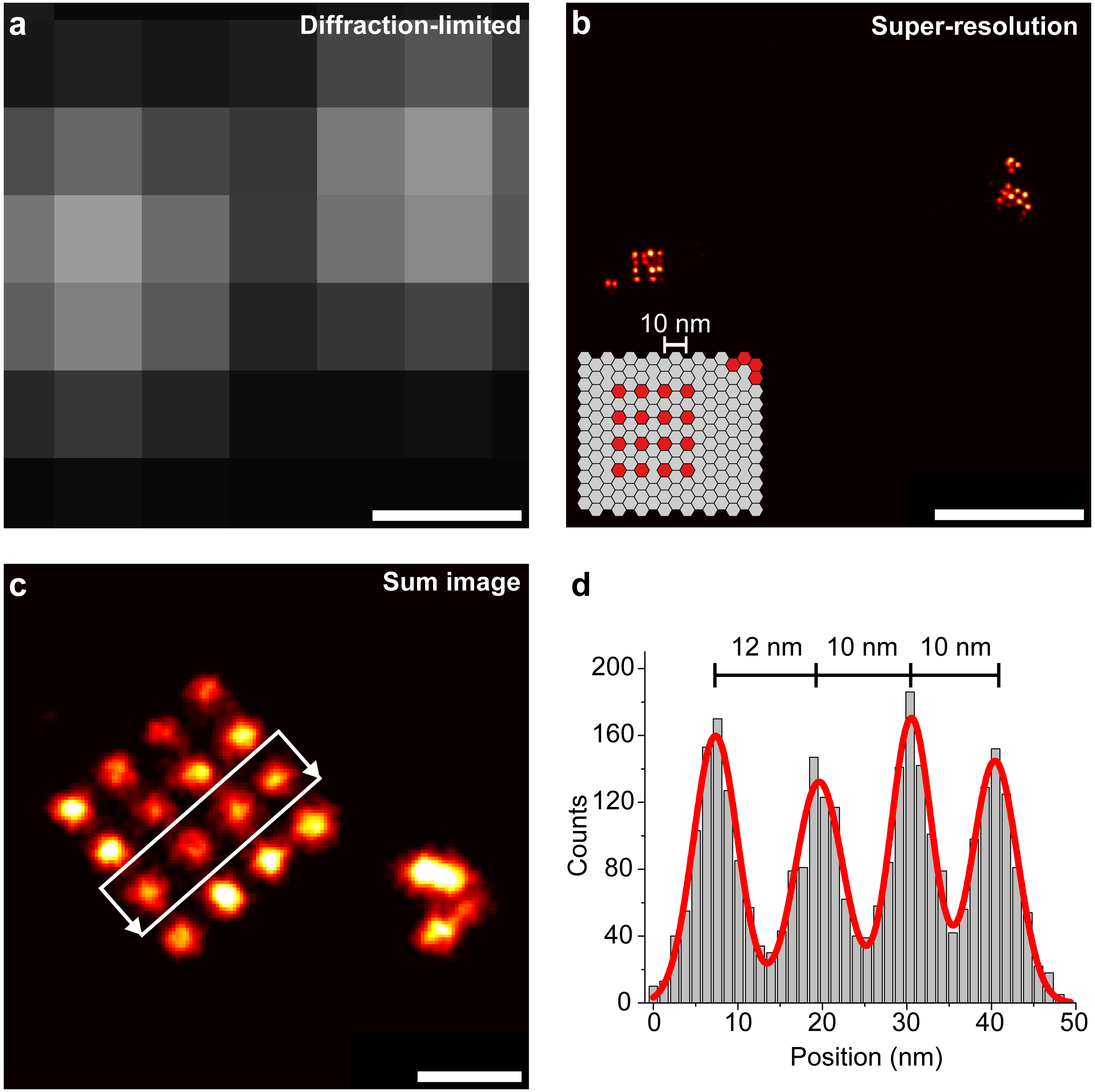


High performance super-resolution microscope

- 10 nm cellular resolution
- Spectrally-unlimited multiplexing
- ~20000 Euro component cost

Auer, Schlichthaerle, Woehrstein, Schueder, Strauss, Grabmayr, Jungmann, ***ChemPhysChem,*** 2018

liteTIRF.jungmannlab.org

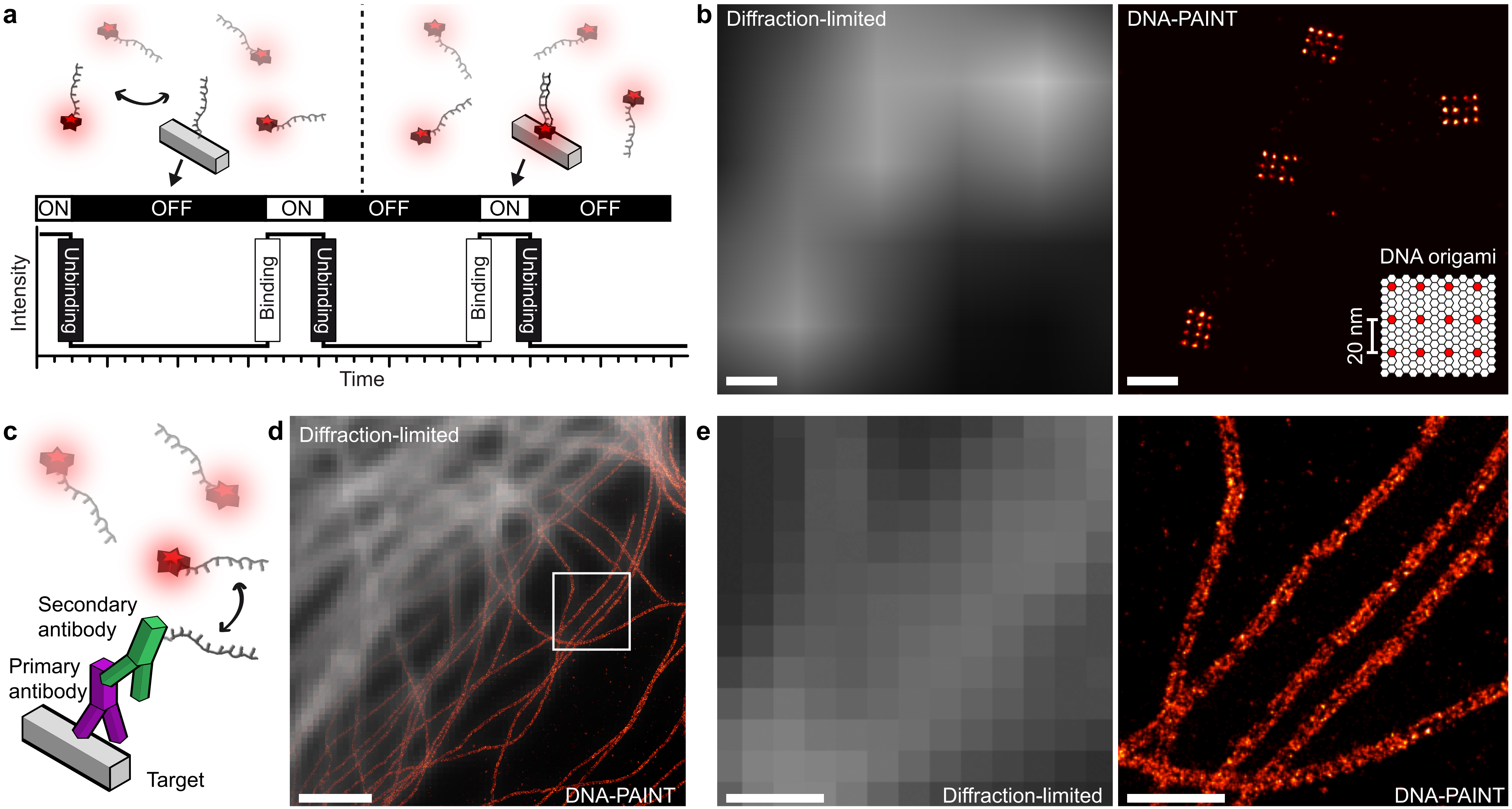


Auer, Schlichthaerle, Woehrstein, Schueder, Strauss, Grabmayr, Jungmann, ***ChemPhysChem,*** 2018

Picasso

A collection of tools for painting super-resolution images.
Learning how to write software as a researcher.
OpenScience Days 2019
Maximilian Thomas Strauß

DNA-PAINT

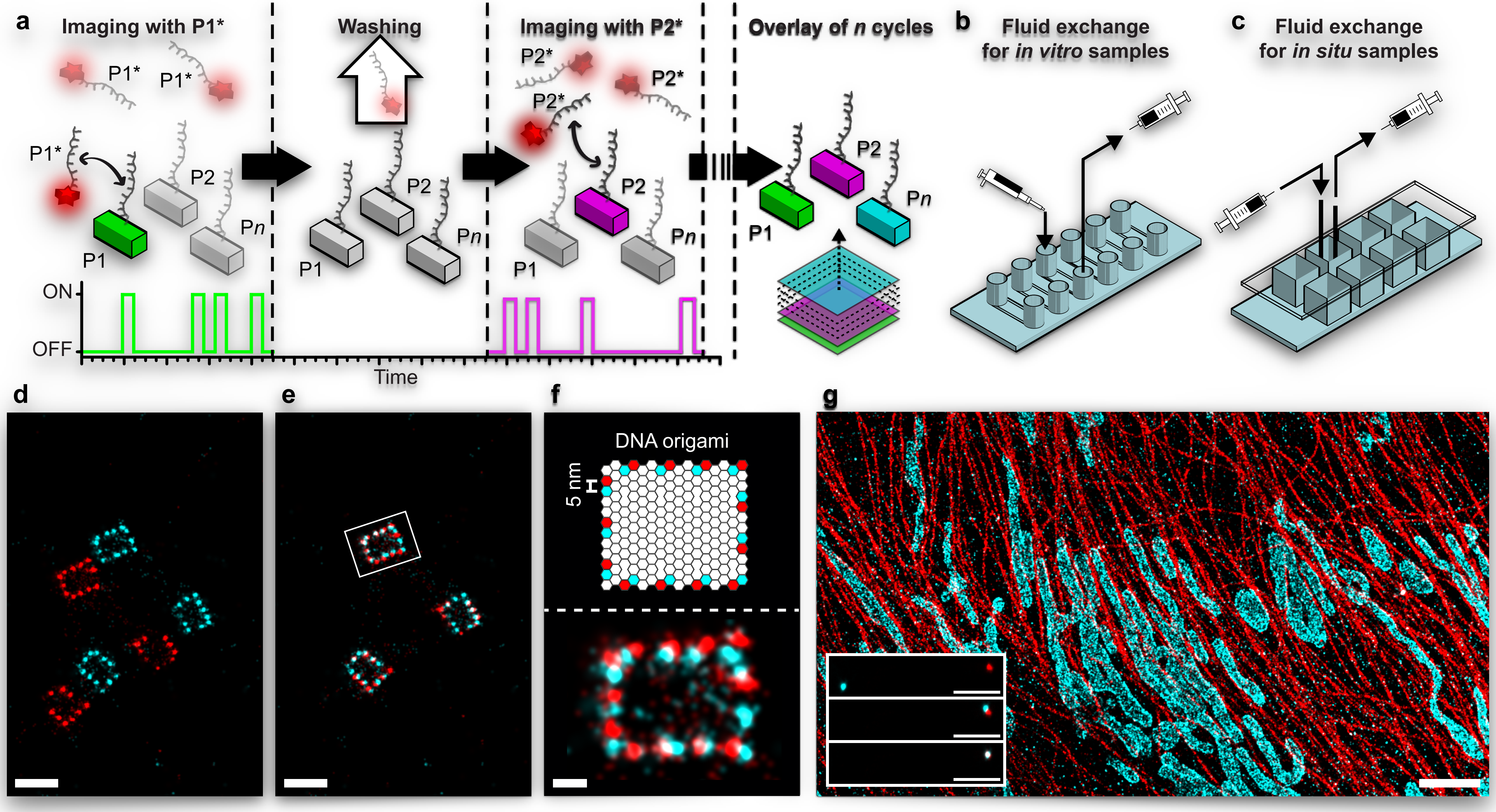


Super-Resolution Microscopy with DNA-PAINT

Joerg Schnitzbauer*, Maximilian T. Strauss*, Thomas Schlichthaerle, Florian Schueder & Ralf Jungmann,

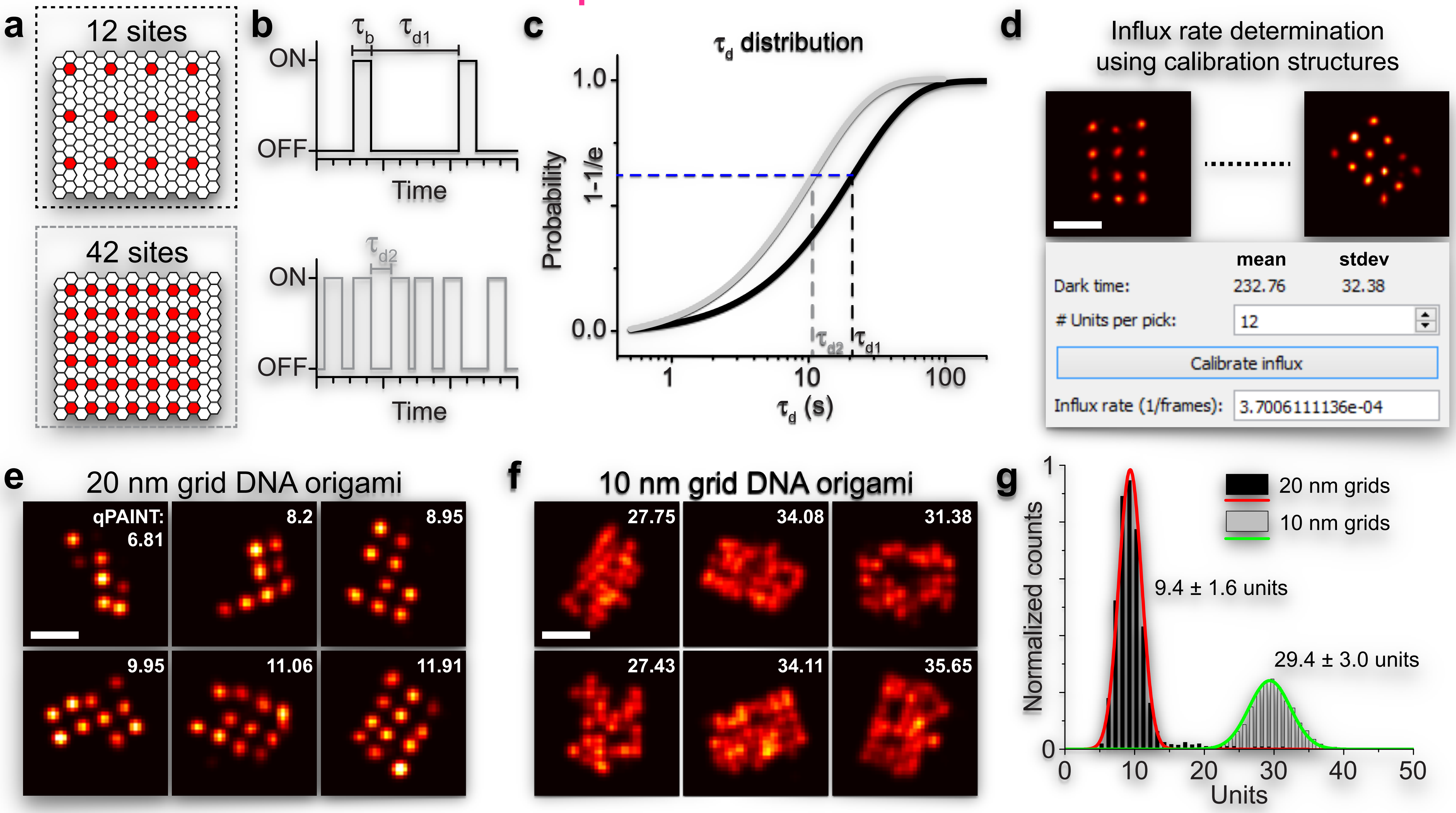
Nature Protocols, (2017). 12: 1198-1228

EXCHANGE-PAINT



Super-Resolution Microscopy with DNA-PAINT
Joerg Schnitzbauer*, Maximilian T. Strauss*, Thomas Schlichthaerle, Florian Schueder & Ralf Jungmann,
Nature Protocols, (2017). 12: 1198-1228

qPAINT



	mean	stdev
Dark time:	232.76	32.38

Units per pick: 12

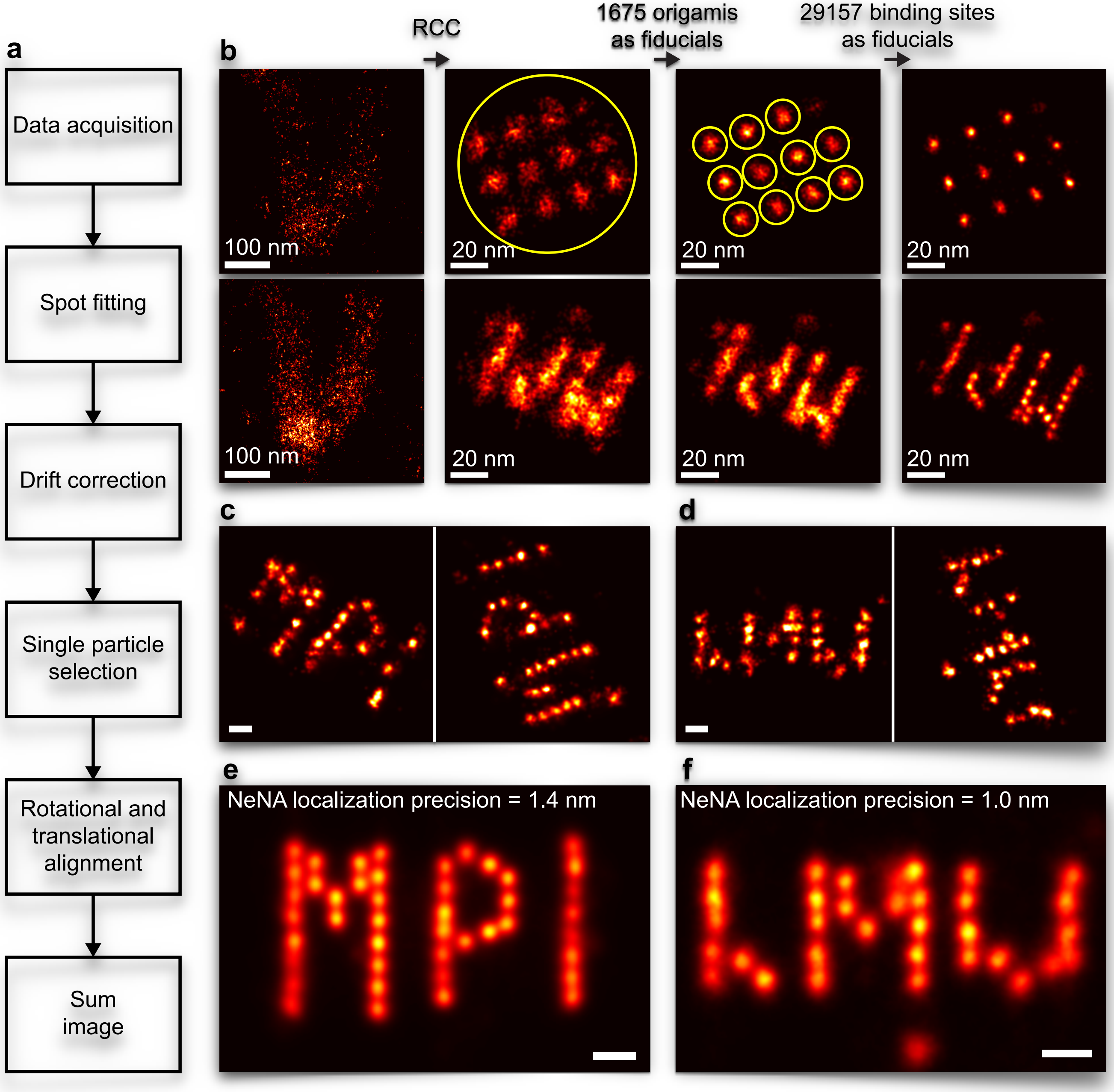
Influx rate (1/frames): 3.7006111136e-04

Super-Resolution Microscopy with DNA-PAINT

Joerg Schnitzbauer*, Maximilian T. Strauss*, Thomas Schlichthaerle, Florian Schueder & Ralf Jungmann,

Nature Protocols, (2017). 12: 1198-1228

RESOLUTION

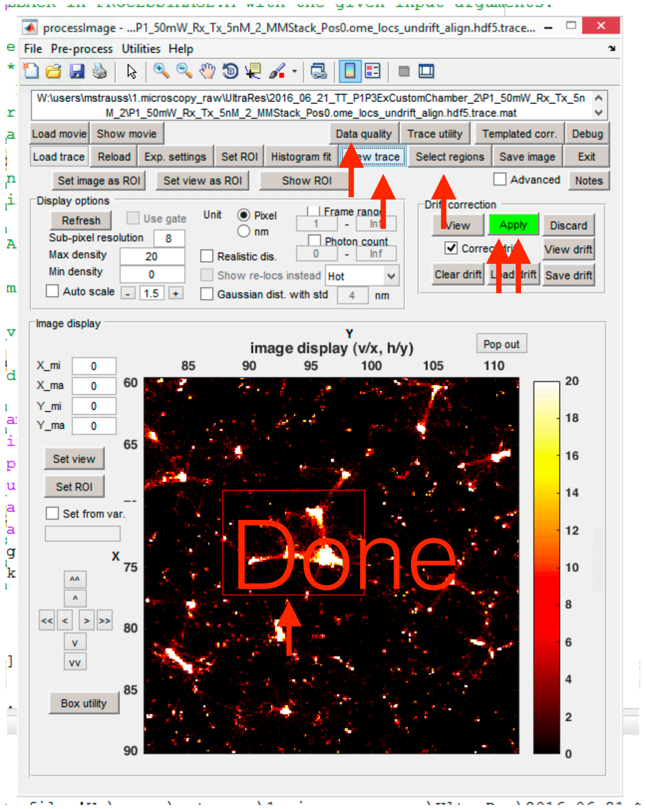

a
Data acquisition
Spot fitting
Drift correction
Single particle selection
Rotational and translational alignment
Sum image
b
RCC
1675 origamis as fiducials
29157 binding sites as fiducials
100 nm
20 nm
20 nm
20 nm
100 nm
20 nm
20 nm
20 nm
c
d
e
NeNA localization precision = 1.4 nm
f
NeNA localization precision = 1.0 nm

Super-Resolution Microscopy with DNA-PAINT
Joerg Schnitzbauer*, Maximilian T. Strauss*, Thomas Schlichthaerle, Florian Schueder & Ralf Jungmann,
Nature Protocols, (2017). 12: 1198-1228

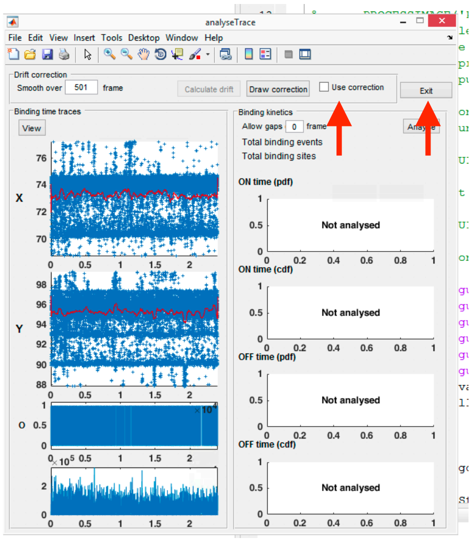
A NEW POSTDOC

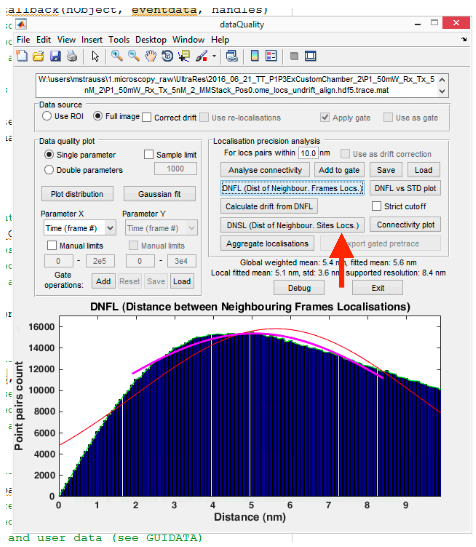
Joerg Schnitzbauer

The old SOFTWARE


processImage - ...P1_50mW_Rx_Tx_5nM_2_MMStack_Pos0.ome_locs_undrift_align.hdf5.trace...

W:\users\mstrauss\1.microscopy_raw\UltraRes\2016_06_21_TT_P1P3ExCustomChamber_2\P1_50mW_Rx_Tx_5nM_2\P1_50mW_Rx_Tx_5nM_2_MMStack_Pos0.ome_locs_undrift_align.hdf5.trace.mat

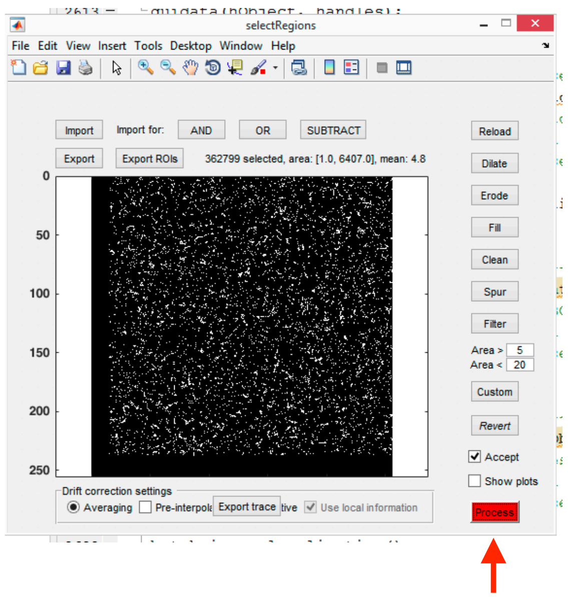



dataQuality

W:\users\mstrauss\1.microscopy_raw\UltraRes\2016_06_21_TT_P1P3ExCustomChamber_2\P1_50mW_Rx_Tx_5nM_2\P1_50mW_Rx_Tx_5nM_2_MMStack_Pos0.ome_locs_undrift_align.hdf5.trace.mat

Global weighted mean: 5.4 nm, fitted mean: 5.6 nm

Local fitted mean: 5.1 nm, std: 3.6 nm supported resolution: 8.4 nm


362799 selected, area: [1.0, 6407.0], mean: 4.8

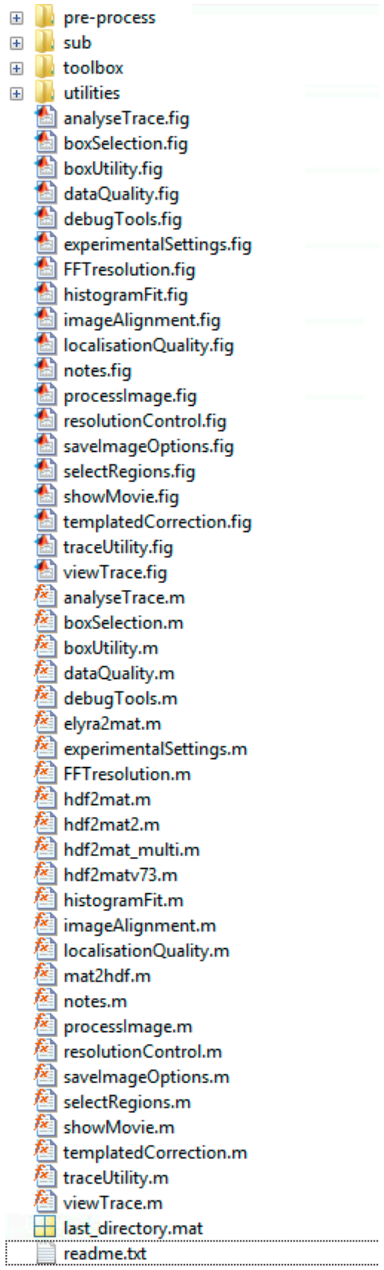
A NEW POSTDOC

Joerg Schnitzbauer

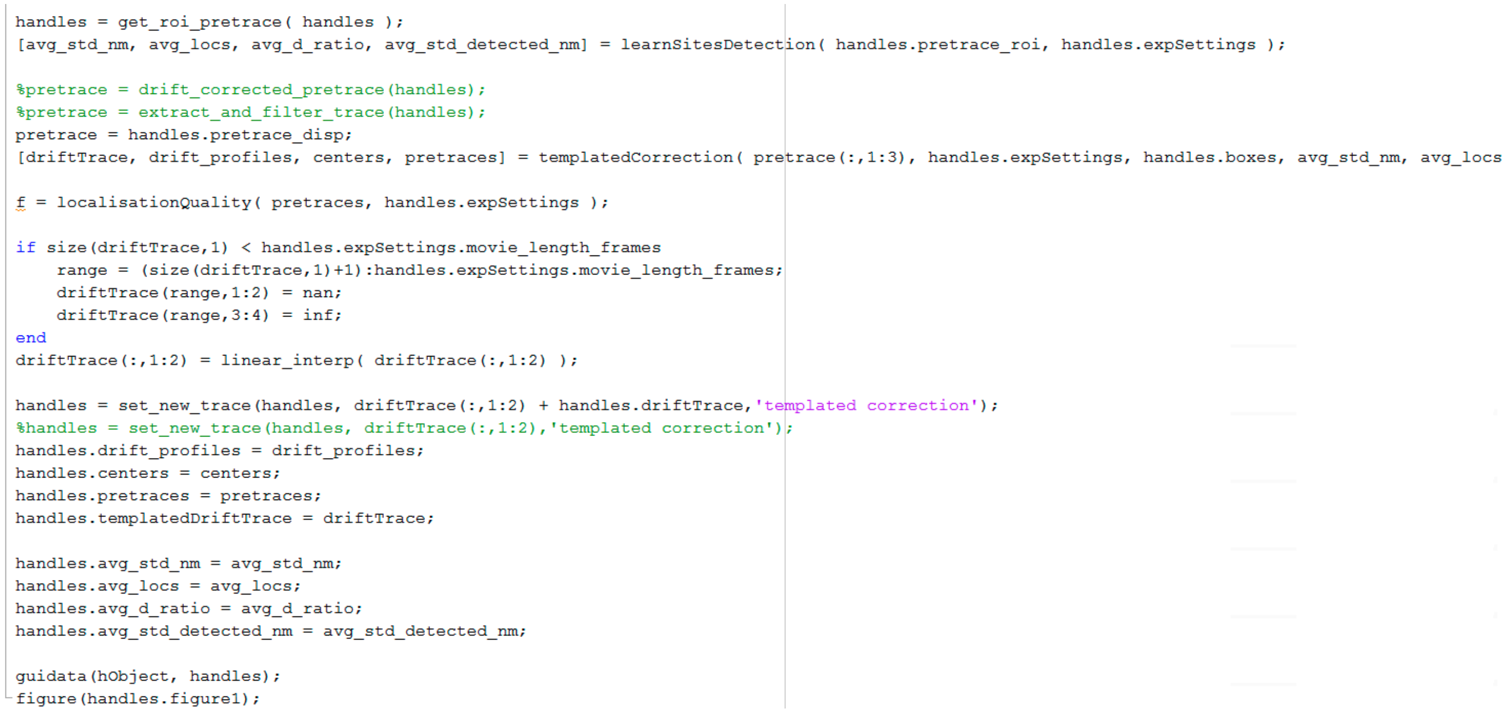
By the time I have mastered these tools
I could just write my own.

The old SOFTWARE

367 files for functions



```
pre-process
sub
toolbox
utilities
analyseTrace.fig
boxSelection.fig
boxUtility.fig
dataQuality.fig
debugTools.fig
experimentalSettings.fig
FFTresolution.fig
histogramFit.fig
imageAlignment.fig
localisationQuality.fig
notes.fig
processImage.fig
resolutionControl.fig
saveImageOptions.fig
selectRegions.fig
showMovie.fig
templatedCorrection.fig
traceUtility.fig
viewTrace.fig
analyseTrace.m
boxSelection.m
boxUtility.m
dataQuality.m
debugTools.m
elyra2mat.m
experimentalSettings.m
FFTresolution.m
hdf2mat.m
hdf2mat2.m
hdf2mat_multi.m
hdf2matv73.m
histogramFit.m
imageAlignment.m
localisationQuality.m
mat2hdf.m
notes.m
processImage.m
resolutionControl.m
saveImageOptions.m
selectRegions.m
showMovie.m
templatedCorrection.m
traceUtility.m
viewTrace.m
last_directory.mat
readme.txt
```



```
handles = get_roi_pretrace( handles );
[avg_std_nm, avg_locs, avg_d_ratio, avg_std_detected_nm] = learnSitesDetection( handles.pretrace_roi, handles.expSettings );

%pretrace = drift_corrected_pretrace(handles);
%pretrace = extract_and_filter_trace(handles);
pretrace = handles.pretrace_disp;
[driftTrace, drift_profiles, centers, pretraces] = templatedCorrection( pretrace(:,1:3), handles.expSettings, handles.boxes, avg_std_nm, avg_locs

f = localisationQuality( pretraces, handles.expSettings );

if size(driftTrace,1) < handles.expSettings.movie_length_frames
    range = (size(driftTrace,1)+1):handles.expSettings.movie_length_frames;
    driftTrace(range,1:2) = nan;
    driftTrace(range,3:4) = inf;
end
driftTrace(:,1:2) = linear_interp( driftTrace(:,1:2) );

handles = set_new_trace(handles, driftTrace(:,1:2) + handles.driftTrace,'templated correction');
%handles = set_new_trace(handles, driftTrace(:,1:2),'templated correction');
handles.drift_profiles = drift_profiles;
handles.centers = centers;
handles.pretraces = pretraces;
handles.templatedDriftTrace = driftTrace;

handles.avg_std_nm = avg_std_nm;
handles.avg_locs = avg_locs;
handles.avg_d_ratio = avg_d_ratio;
handles.avg_std_detected_nm = avg_std_detected_nm;

guidata(hObject, handles);
figure(handles.figure1);
```

0 comments

Available software packages

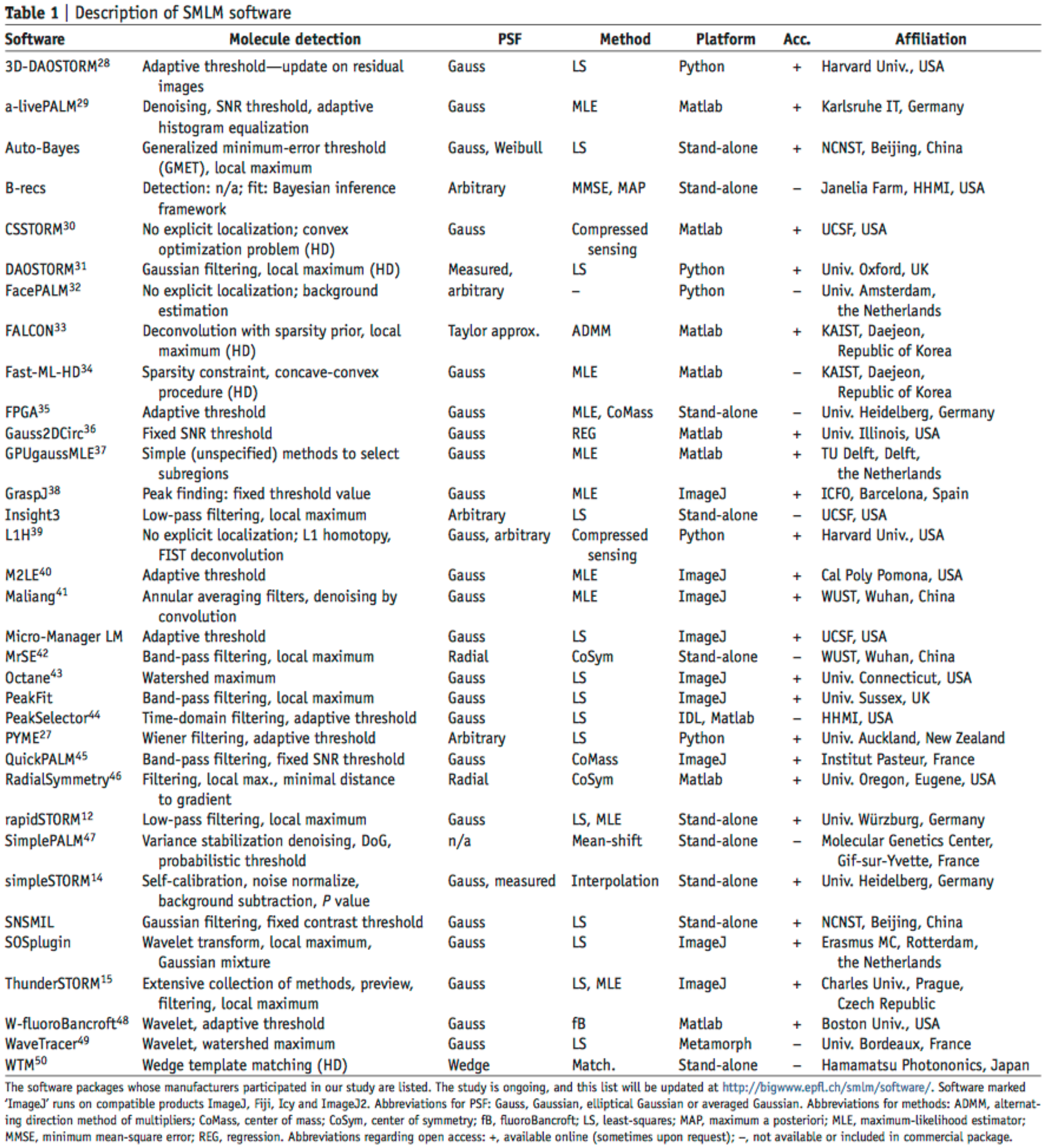


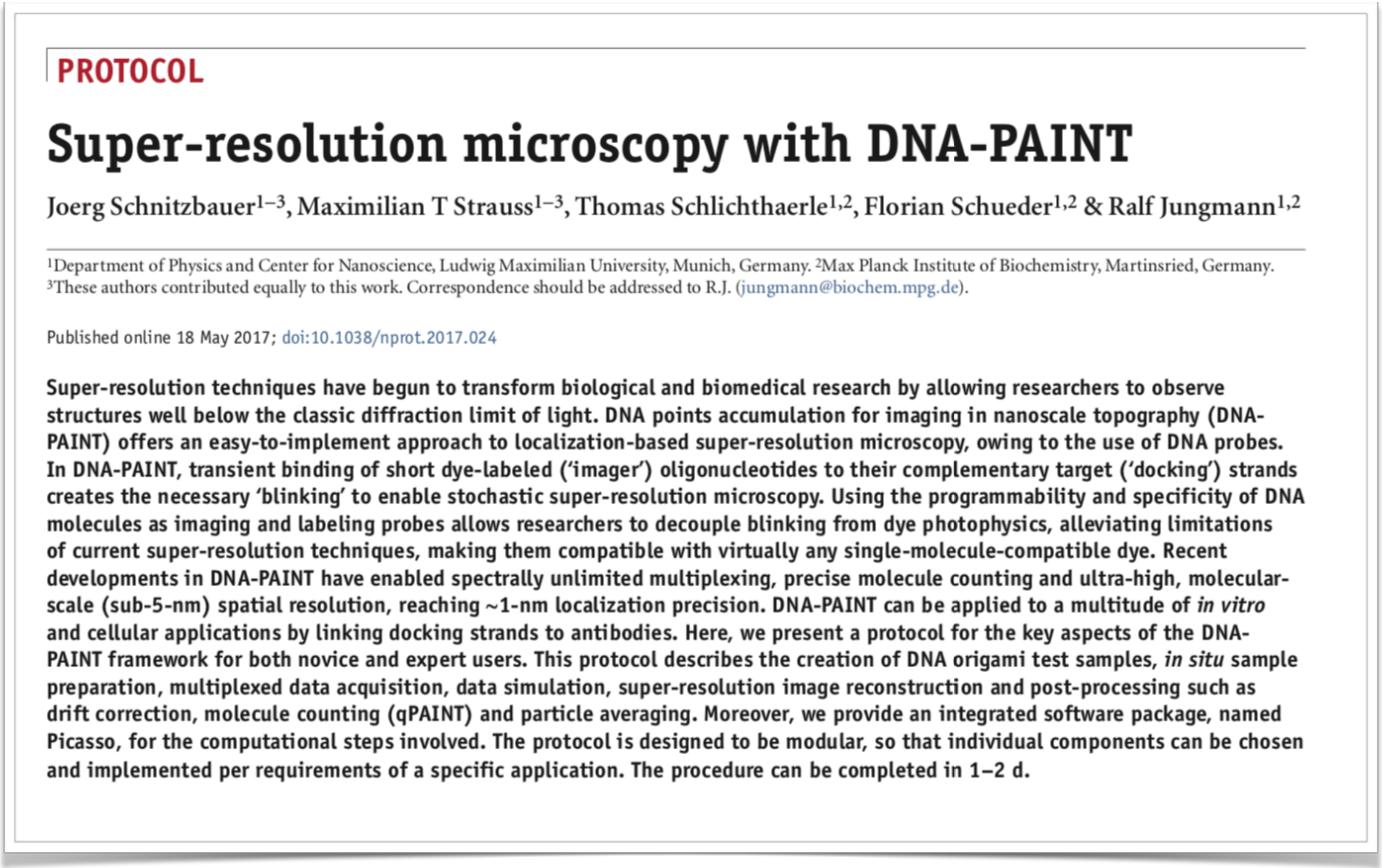
Table 1 | Description of SMLM software

Software	Molecule detection	PSF	Method	Platform	Acc.	Affiliation
3D-DAOSTORM[28]	Adaptive threshold—update on residual images	Gauss	LS	Python	+	Harvard Univ., USA
a-livePALM[29]	Denoising, SNR threshold, adaptive histogram equalization	Gauss	MLE	Matlab	+	Karlsruhe IT, Germany
Auto-Bayes	Generalized minimum-error threshold (GMET), local maximum	Gauss, Weibull	LS	Stand-alone	+	NCNST, Beijing, China
B-recs	Detection: n/a; fit: Bayesian inference framework	Arbitrary	MMSE, MAP	Stand-alone	–	Janelia Farm, HHMI, USA
CSSTORM[30]	No explicit localization; convex optimization problem (HD)	Gauss	Compressed sensing	Matlab	+	UCSF, USA
DAOSTORM[31]	Gaussian filtering, local maximum (HD)	Measured, arbitrary	LS	Python	+	Univ. Oxford, UK
FacePALM[32]	No explicit localization; background estimation		–	Python	–	Univ. Amsterdam, the Netherlands
FALCON[33]	Deconvolution with sparsity prior, local maximum (HD)	Taylor approx.	ADMM	Matlab	+	KAIST, Daejeon, Republic of Korea
Fast-ML-HD[34]	Sparsity constraint, concave-convex procedure (HD)	Gauss	MLE	Matlab	–	KAIST, Daejeon, Republic of Korea
FPGA[35]	Adaptive threshold	Gauss	MLE, CoMass	Stand-alone	–	Univ. Heidelberg, Germany
Gauss2DCirc[36]	Fixed SNR threshold	Gauss	REG	Matlab	+	Univ. Illinois, USA
GPUgaussMLE[37]	Simple (unspecified) methods to select subregions	Gauss	MLE	Matlab	+	TU Delft, Delft, the Netherlands
GraspJ[38]	Peak finding: fixed threshold value	Gauss	MLE	ImageJ	+	ICFO, Barcelona, Spain
Insight3	Low-pass filtering, local maximum	Arbitrary	LS	Stand-alone	–	UCSF, USA
L1H[39]	No explicit localization; L1 homotopy, FIST deconvolution	Gauss, arbitrary	Compressed sensing	Python	+	Harvard Univ., USA
M2LE[40]	Adaptive threshold	Gauss	MLE	ImageJ	+	Cal Poly Pomona, USA
Maliang[41]	Annular averaging filters, denoising by convolution	Gauss	MLE	ImageJ	+	WUST, Wuhan, China
Micro-Manager LM	Adaptive threshold	Gauss	LS	ImageJ	+	UCSF, USA
MrSE[42]	Band-pass filtering, local maximum	Radial	CoSym	Stand-alone	–	WUST, Wuhan, China
Octane[43]	Watershed maximum	Gauss	LS	ImageJ	+	Univ. Connecticut, USA
PeakFit	Band-pass filtering, local maximum	Gauss	LS	ImageJ	+	Univ. Sussex, UK
PeakSelector[44]	Time-domain filtering, adaptive threshold	Gauss	LS	IDL, Matlab	–	HHMI, USA
PYME[27]	Wiener filtering, adaptive threshold	Arbitrary	LS	Python	+	Univ. Auckland, New Zealand
QuickPALM[45]	Band-pass filtering, fixed SNR threshold	Gauss	CoMass	ImageJ	+	Institut Pasteur, France
RadialSymmetry[46]	Filtering, local max., minimal distance to gradient	Radial	CoSym	Matlab	+	Univ. Oregon, Eugene, USA
rapidSTORM[12]	Low-pass filtering, local maximum	Gauss	LS, MLE	Stand-alone	+	Univ. Würzburg, Germany
SimplePALM[47]	Variance stabilization denoising, DoG, probabilistic threshold	n/a	Mean-shift	Stand-alone	–	Molecular Genetics Center, Gif-sur-Yvette, France
simpleSTORM[14]	Self-calibration, noise normalize, background subtraction, *P* value	Gauss, measured	Interpolation	Stand-alone	+	Univ. Heidelberg, Germany
SNSMIL	Gaussian filtering, fixed contrast threshold	Gauss	LS	Stand-alone	+	NCNST, Beijing, China
SOSplugin	Wavelet transform, local maximum, Gaussian mixture	Gauss	LS	ImageJ	+	Erasmus MC, Rotterdam, the Netherlands
ThunderSTORM[15]	Extensive collection of methods, preview, filtering, local maximum	Gauss	LS, MLE	ImageJ	+	Charles Univ., Prague, Czech Republic
W-fluoroBancroft[48]	Wavelet, adaptive threshold	Gauss	fB	Matlab	+	Boston Univ., USA
WaveTracer[49]	Wavelet, watershed maximum	Gauss	LS	Metamorph	–	Univ. Bordeaux, France
WTM[50]	Wedge template matching (HD)	Wedge	Match.	Stand-alone	–	Hamamatsu Photononics, Japan

The software packages whose manufacturers participated in our study are listed. The study is ongoing, and this list will be updated at http://bigwww.epfl.ch/smlm/software/. Software marked 'ImageJ' runs on compatible products ImageJ, Fiji, Icy and ImageJ2. Abbreviations for PSF: Gauss, Gaussian, elliptical Gaussian or averaged Gaussian. Abbreviations for methods: ADMM, alternating direction method of multipliers; CoMass, center of mass; CoSym, center of symmetry; fB, fluoroBancroft; LS, least-squares; MAP, maximum a posteriori; MLE, maximum-likelihood estimator; MMSE, minimum mean-square error; REG, regression. Abbreviations regarding open access: +, available online (sometimes upon request); –, not available or included in commercial package.

Sage, D. *et al.* Quantitative evaluation of software packages for single-molecule localization microscopy. *Nat. Methods* **12,** 717–724 (2015).

How could we sell this?



PROTOCOL

Super-resolution microscopy with DNA-PAINT

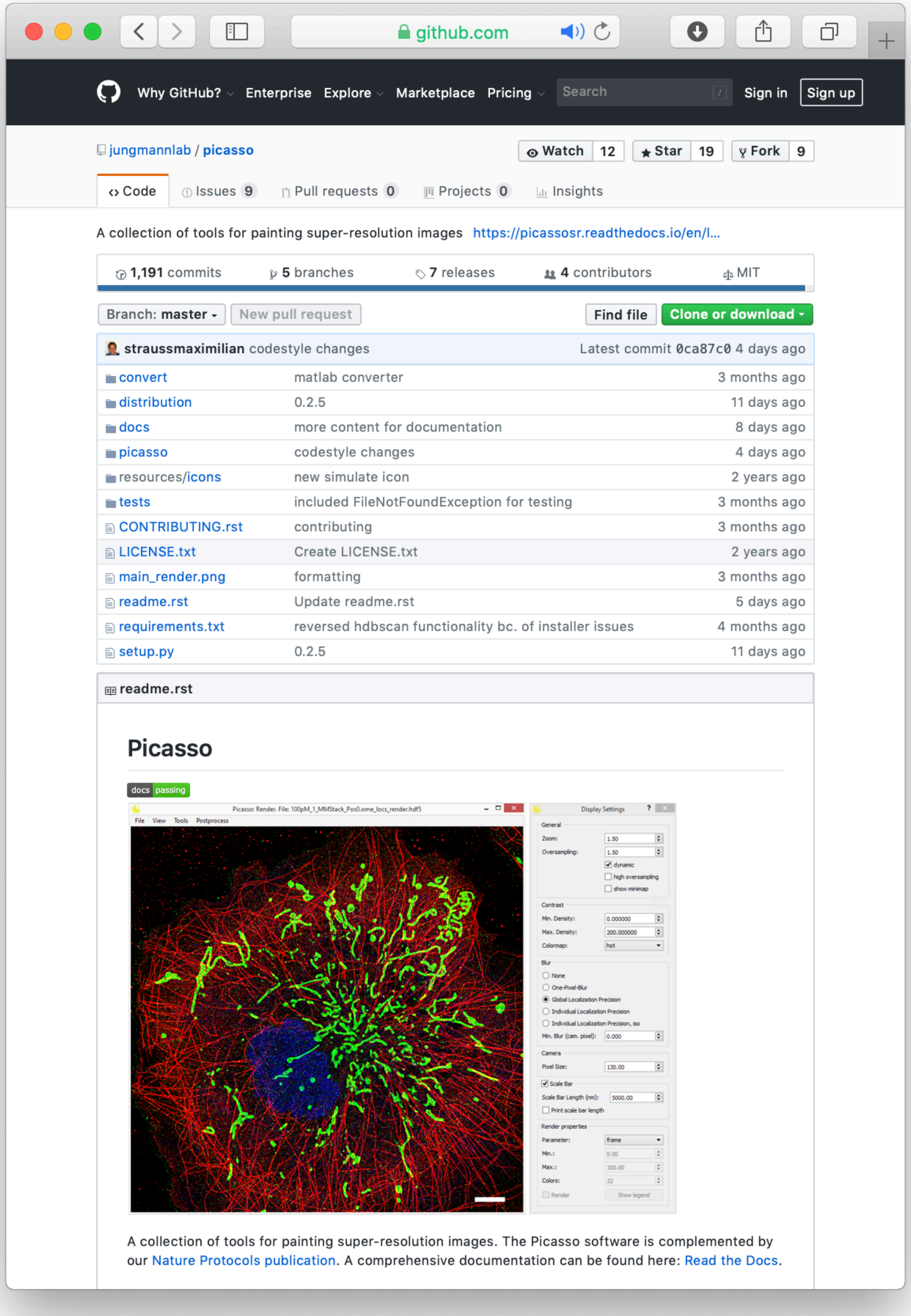
Joerg Schnitzbauer[1–3], Maximilian T Strauss[1–3], Thomas Schlichthaerle[1,2], Florian Schueder[1,2] & Ralf Jungmann[1,2]

[1]Department of Physics and Center for Nanoscience, Ludwig Maximilian University, Munich, Germany. [2]Max Planck Institute of Biochemistry, Martinsried, Germany. [3]These authors contributed equally to this work. Correspondence should be addressed to R.J. (jungmann@biochem.mpg.de).

Published online 18 May 2017; doi:10.1038/nprot.2017.024

Super-resolution techniques have begun to transform biological and biomedical research by allowing researchers to observe structures well below the classic diffraction limit of light. DNA points accumulation for imaging in nanoscale topography (DNA-PAINT) offers an easy-to-implement approach to localization-based super-resolution microscopy, owing to the use of DNA probes. In DNA-PAINT, transient binding of short dye-labeled ('imager') oligonucleotides to their complementary target ('docking') strands creates the necessary 'blinking' to enable stochastic super-resolution microscopy. Using the programmability and specificity of DNA molecules as imaging and labeling probes allows researchers to decouple blinking from dye photophysics, alleviating limitations of current super-resolution techniques, making them compatible with virtually any single-molecule-compatible dye. Recent developments in DNA-PAINT have enabled spectrally unlimited multiplexing, precise molecule counting and ultra-high, molecular-scale (sub-5-nm) spatial resolution, reaching ~1-nm localization precision. DNA-PAINT can be applied to a multitude of *in vitro* and cellular applications by linking docking strands to antibodies. Here, we present a protocol for the key aspects of the DNA-PAINT framework for both novice and expert users. This protocol describes the creation of DNA origami test samples, *in situ* sample preparation, multiplexed data acquisition, data simulation, super-resolution image reconstruction and post-processing such as drift correction, molecule counting (qPAINT) and particle averaging. Moreover, we provide an integrated software package, named Picasso, for the computational steps involved. The protocol is designed to be modular, so that individual components can be chosen and implemented per requirements of a specific application. The procedure can be completed in 1–2 d.

The current state of Picasso



A collection of tools for painting super-resolution images https://picassosr.readthedocs.io/en/l...

1,191 commits · 5 branches · 7 releases · 4 contributors · MIT

strausmaximilian codestyle changes — Latest commit 0ca87c0 4 days ago

convert	matlab converter	3 months ago
distribution	0.2.5	11 days ago
docs	more content for documentation	8 days ago
picasso	codestyle changes	4 days ago
resources/icons	new simulate icon	2 years ago
tests	included FileNotFoundException for testing	3 months ago
CONTRIBUTING.rst	contributing	3 months ago
LICENSE.txt	Create LICENSE.txt	2 years ago
main_render.png	formatting	3 months ago
readme.rst	Update readme.rst	5 days ago
requirements.txt	reversed hdbscan functionality bc. of installer issues	4 months ago
setup.py	0.2.5	11 days ago

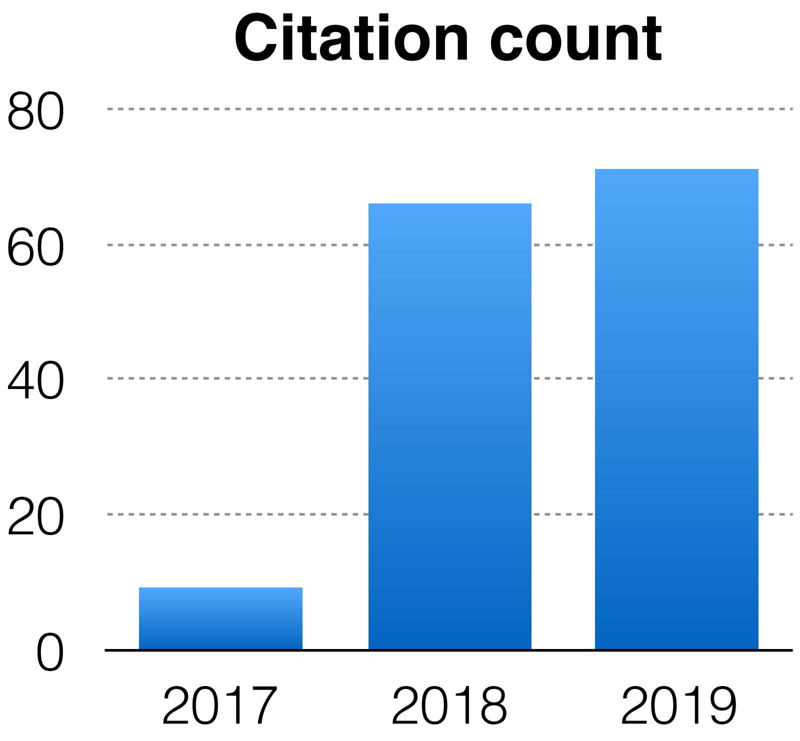
readme.rst

Picasso

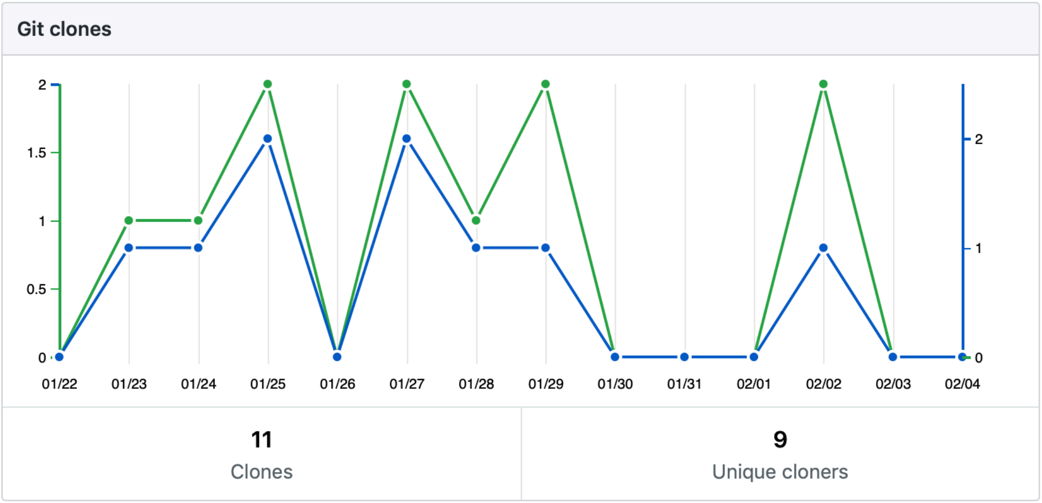
A collection of tools for painting super-resolution images. The Picasso software is complemented by our Nature Protocols publication. A comprehensive documentation can be found here: Read the Docs.

Python based
available on GitHub
one-click installer for windows
~ 20.000 lines of code

Some Statistics



Currently 71 citations


Git clones

11
Clones

9
Unique cloners

~ 9 unique clones in the last two weeks

Support requests

Wyss Institute - Harvard University
University of Wuerzburg
University of New South Wales
University of Glasgow
University of Cambridge
University of Baltimore
University Medical Center Göttingen
Technical University of Munich
TU Vienna
Osaka University
Max Planck Institute of Biochemistry
Max Planck Institute for Brain Research
Ludwig-Maximilians-Universität München
Karolinska Institutet
Johann Wolfgang Goethe-University
Jawaharlal Nehru Centre for Advanced Scientific Research
Institute of Chemical and Biological Technology
Imperial College London
Freie Universität Berlin
Emory University
European Molecular Biology Laboratory
Duke University
Montpellier Cell Biology Research Center
Centro de Investigaciones en Bionanociencias
Arizona State University
Aix-Marseille University

How did we get there?
or: What did we learn?

Why open-source?

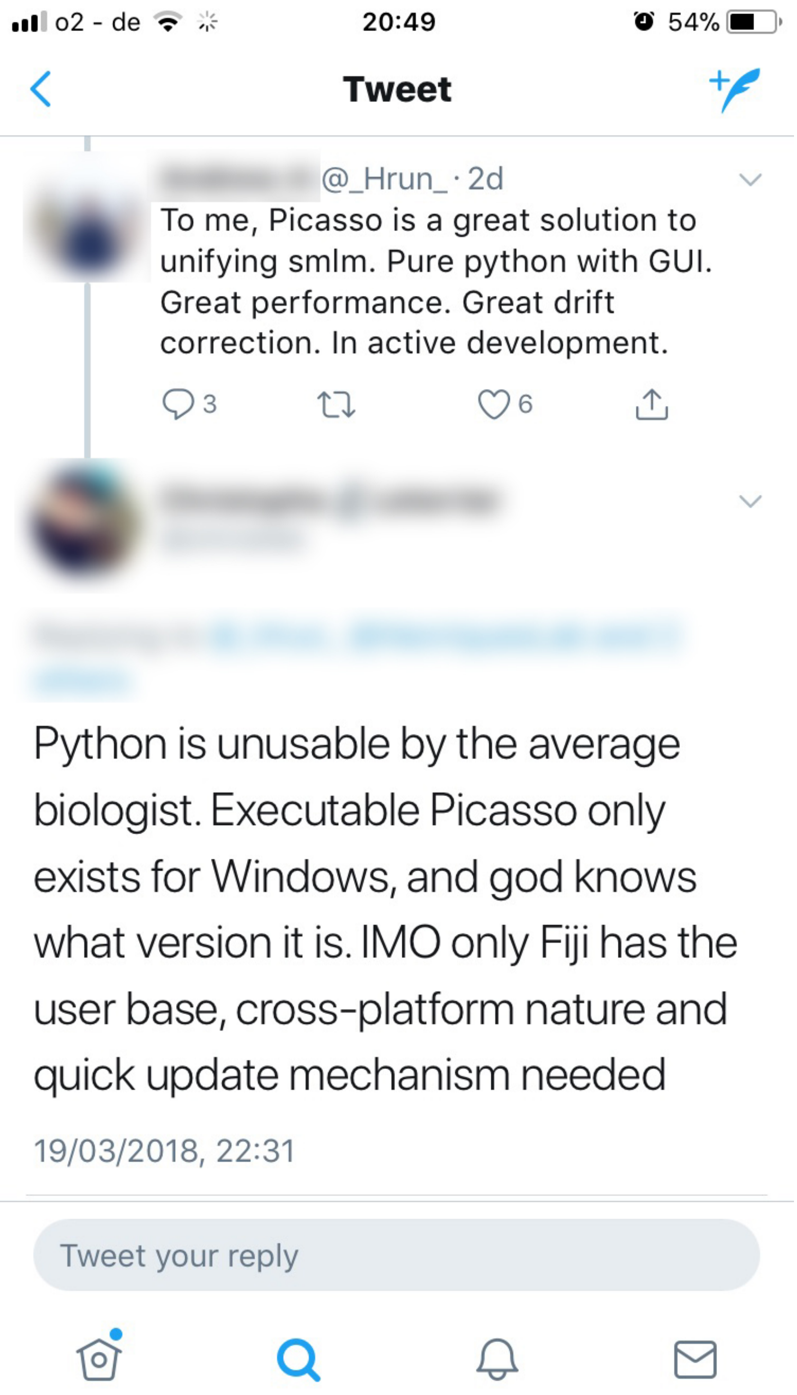
vulnerable to get scooped?

If someone is so good that
by reading the source-code they
can figure out what one is working on and still be faster
they deserve credit.

GitLab

Main Motivation: version control

Communication



[redacted] @_Hrun_ · 2d

To me, Picasso is a great solution to unifying smlm. Pure python with GUI. Great performance. Great drift correction. In active development.

[redacted]

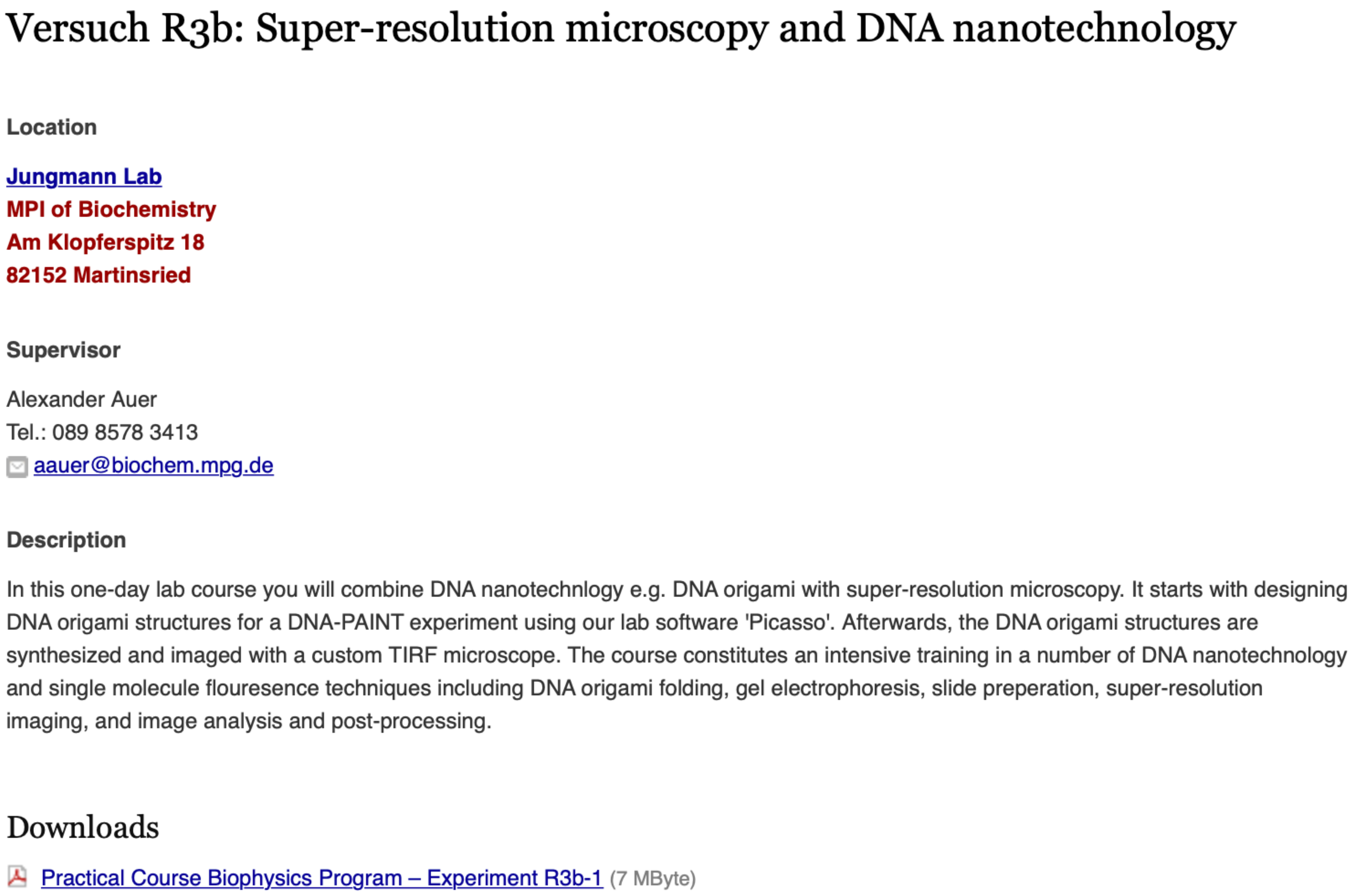
Python is unusable by the average biologist. Executable Picasso only exists for Windows, and god knows what version it is. IMO only Fiji has the user base, cross-platform nature and quick update mechanism needed

19/03/2018, 22:31

Twitter-Discussions

- can give helpful input

Teaching



Versuch R3b: Super-resolution microscopy and DNA nanotechnology

Location

Jungmann Lab
MPI of Biochemistry
Am Klopferspitz 18
82152 Martinsried

Supervisor

Alexander Auer
Tel.: 089 8578 3413
✉ aauer@biochem.mpg.de

Description

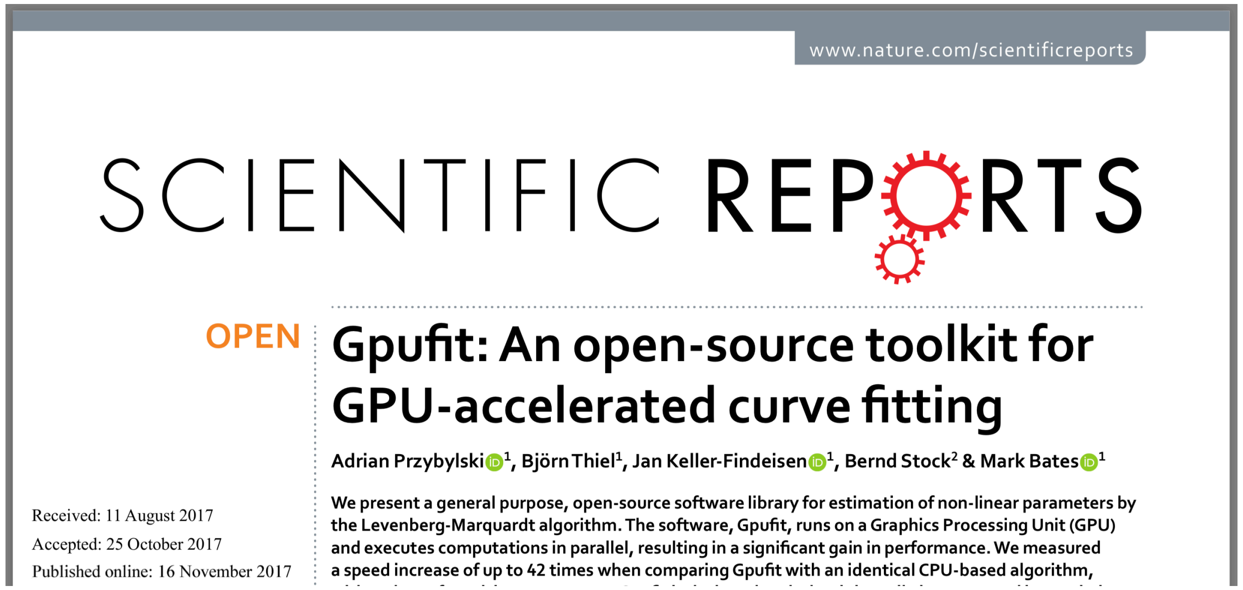
In this one-day lab course you will combine DNA nanotechnlogy e.g. DNA origami with super-resolution microscopy. It starts with designing DNA origami structures for a DNA-PAINT experiment using our lab software 'Picasso'. Afterwards, the DNA origami structures are synthesized and imaged with a custom TIRF microscope. The course constitutes an intensive training in a number of DNA nanotechnology and single molecule flouresence techniques including DNA origami folding, gel electrophoresis, slide preperation, super-resolution imaging, and image analysis and post-processing.

Downloads

Practical Course Biophysics Program – Experiment R3b-1 (7 MByte)

What did we get from making it open-source?

- Quality control
- Bug Reports
- Documentation
- Contributions



www.nature.com/scientificreports

SCIENTIFIC REPORTS

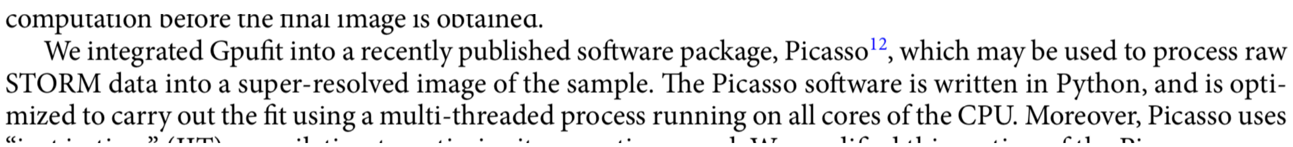
OPEN

Gpufit: An open-source toolkit for GPU-accelerated curve fitting

Adrian Przybylski[1], Björn Thiel[1], Jan Keller-Findeisen[1], Bernd Stock[2] & Mark Bates[1]

Received: 11 August 2017
Accepted: 25 October 2017
Published online: 16 November 2017

We present a general purpose, open-source software library for estimation of non-linear parameters by the Levenberg-Marquardt algorithm. The software, Gpufit, runs on a Graphics Processing Unit (GPU) and executes computations in parallel, resulting in a significant gain in performance. We measured a speed increase of up to 42 times when comparing Gpufit with an identical CPU-based algorithm,



computation before the final image is obtained.

We integrated Gpufit into a recently published software package, Picasso[12], which may be used to process raw STORM data into a super-resolved image of the sample. The Picasso software is written in Python, and is optimized to carry out the fit using a multi-threaded process running on all cores of the CPU. Moreover, Picasso uses

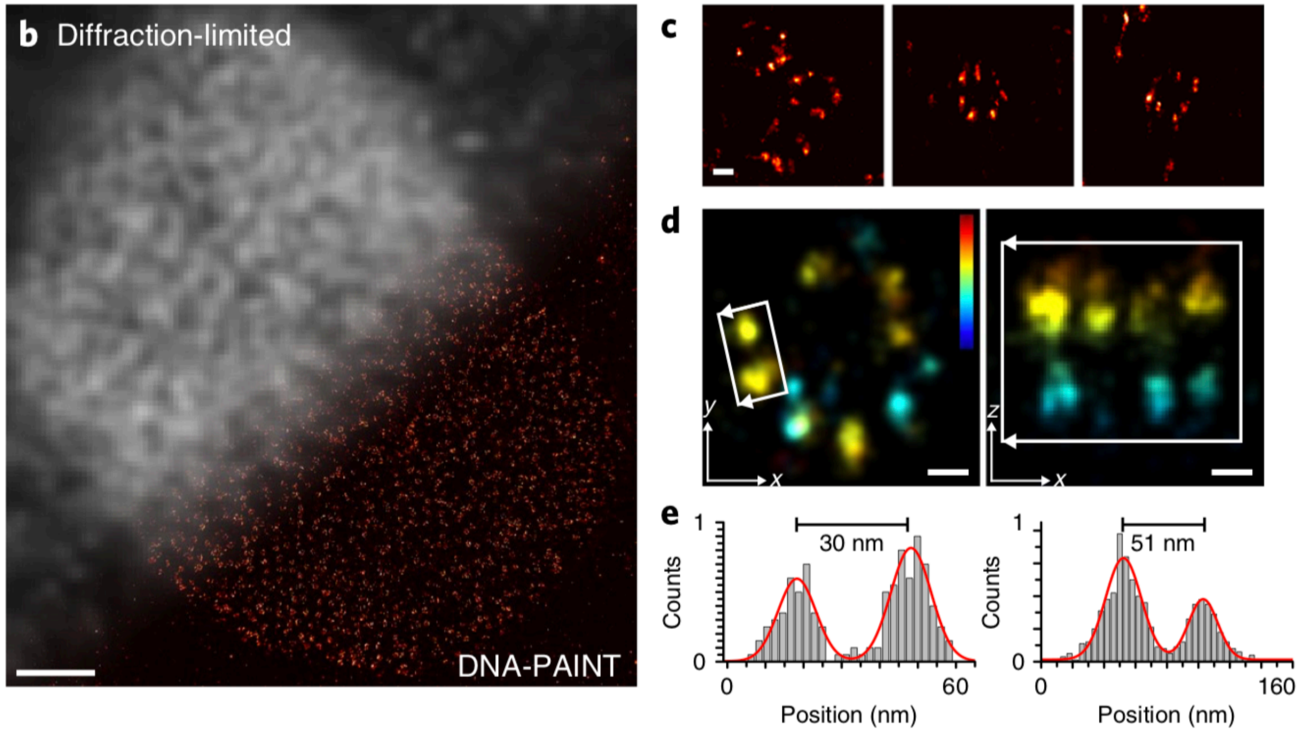
Where is this heading?

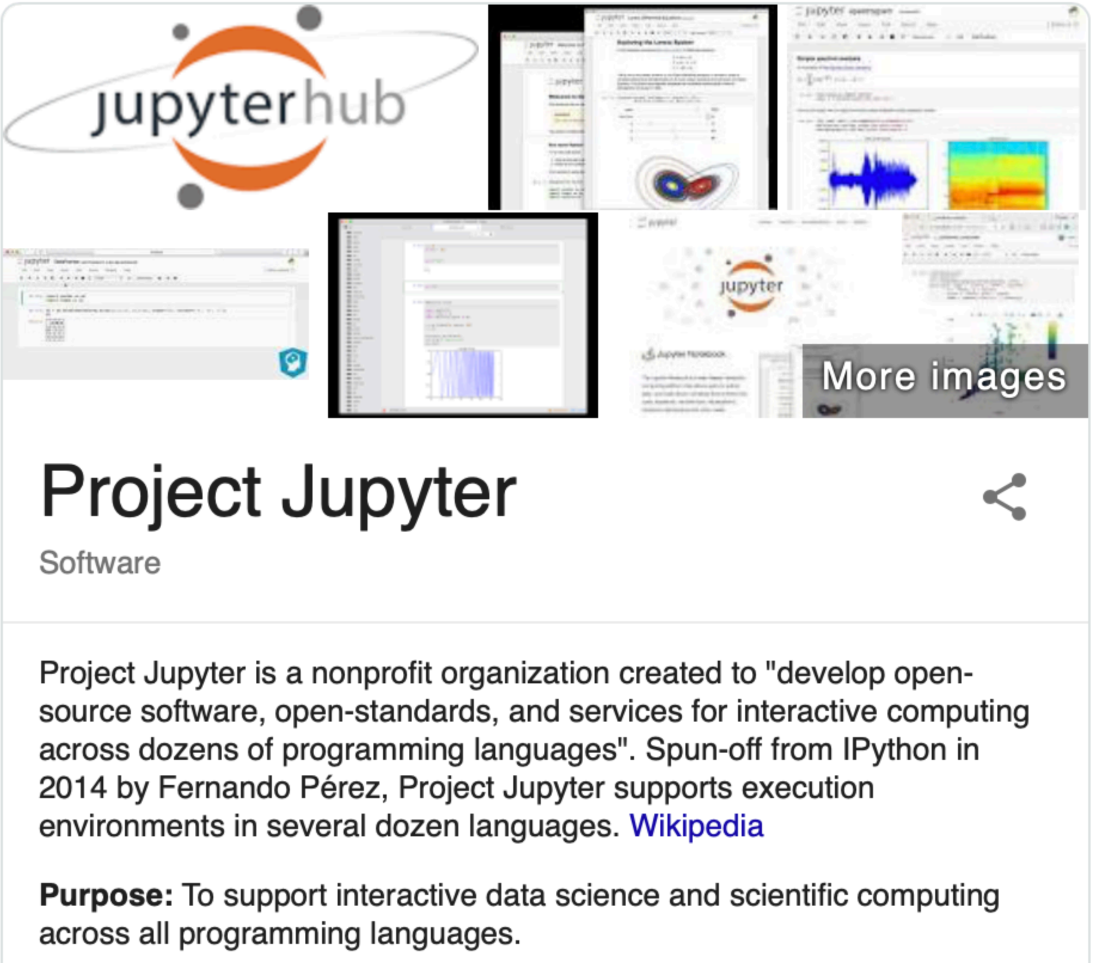
- Better documentation
- Encourage contributions
- Implement automated tests

> Research **Software Engineer**

What could be next?

How could it be?





Project Jupyter

Software

Project Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". Spun-off from IPython in 2014 by Fernando Pérez, Project Jupyter supports execution environments in several dozen languages. Wikipedia

Purpose: To support interactive data science and scientific computing across all programming languages.

- Make raw data available
- Include scripts on how the each figure was created
- Automated reproducibility of the evaluation

Thank you for your attention.