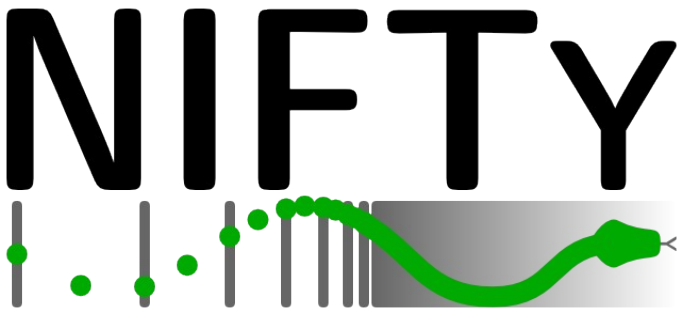
Statistical inference in Python the NIFTy way

Michael R. Bell
Frontier Technology, Inc.

Marco Selig, Henrik Junklewitz, Niels Oppermann,
Martin Reinecke, Maksim Greiner, Carlos Pachajoa, Torsten A. Enßlin

Max Planck Institute for Astrophysics,
Munich, Germany

Selig et al. (2013)
arXiv: 1301.4499



is a Python framework for developing statistical inference algorithms independent of the underlying geometry or resolution.

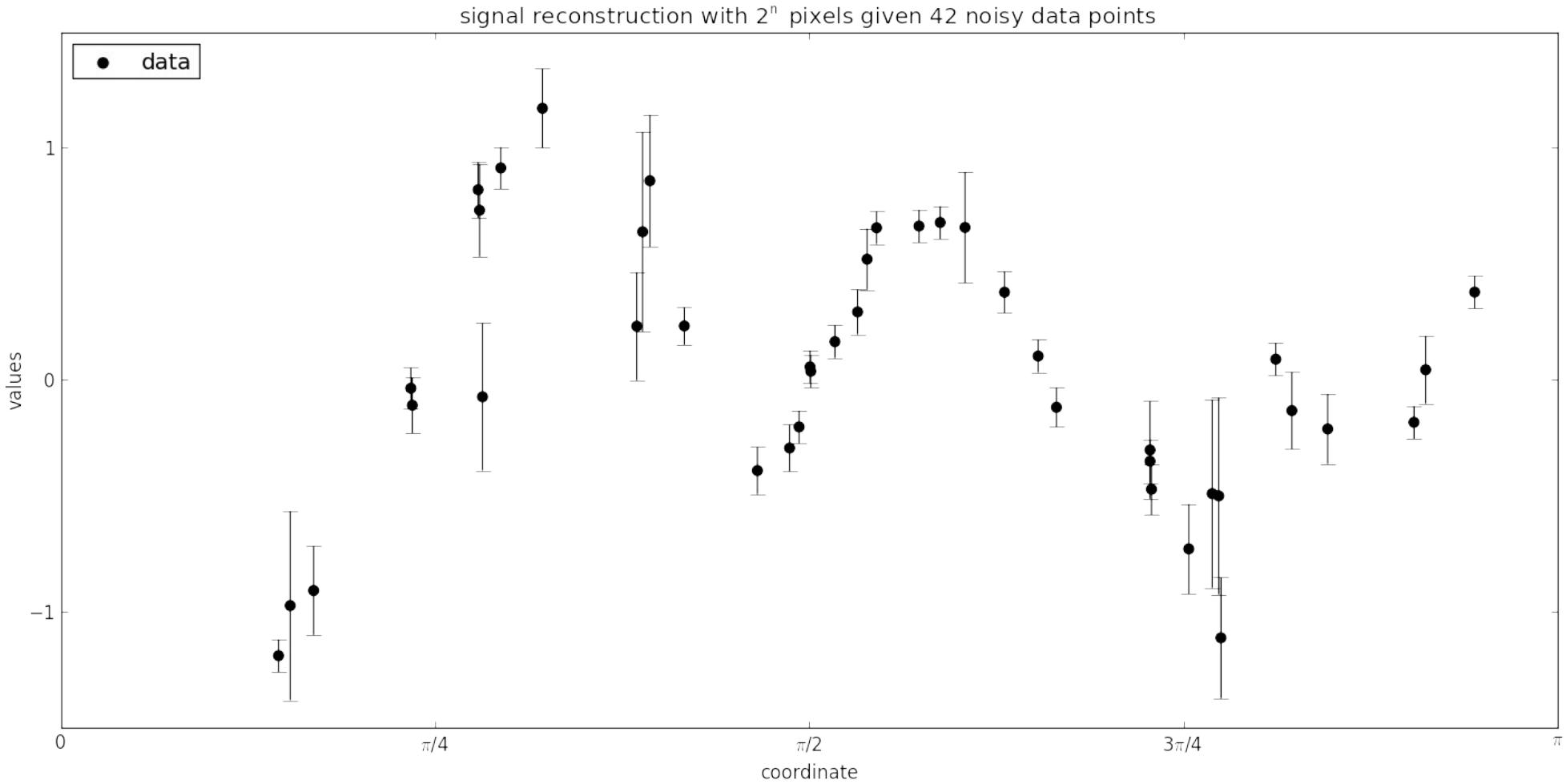
Follow along at: http://tinyurl.com/nifty-demo-slides

Data & Signal

Signal $\xrightarrow{\text{observation}}$ Data

$\boldsymbol{s} = s(x) \quad , \quad x \in \Omega$

$\boldsymbol{d} = (d_i)_{i \in \mathbb{N}}$



Signal estimate

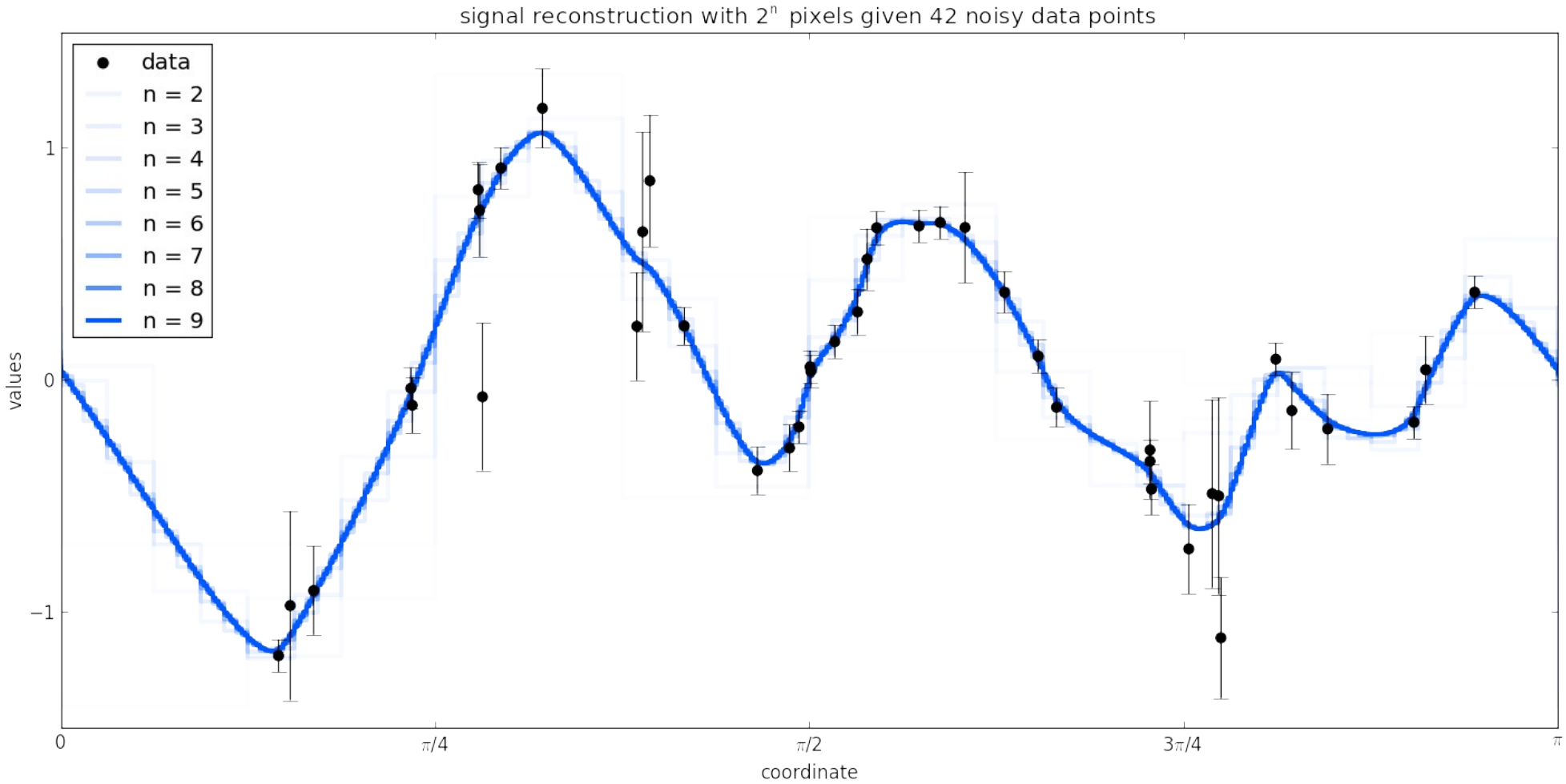
observation

inference

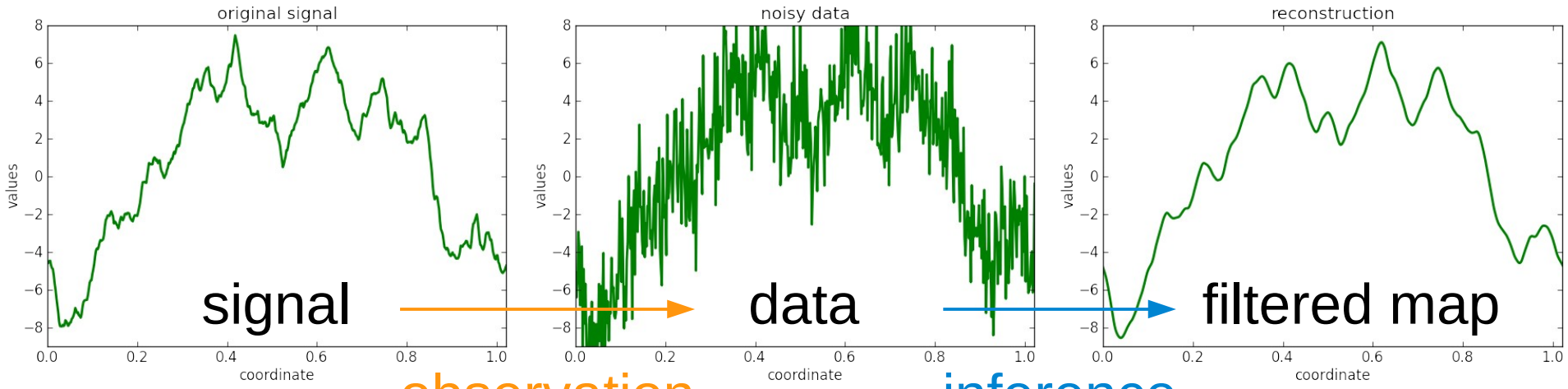
Data

$\boldsymbol{s} = s(x) \quad , \quad x \in \Omega$

$\boldsymbol{d} = (d_i)_{i \in \mathbb{N}}$



Wiener filtering

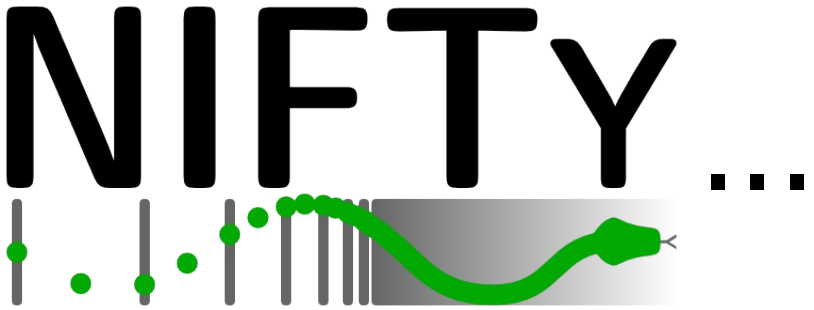


$$s \curvearrowleft \mathcal{G}(s, S)$$

$$n \curvearrowleft \mathcal{G}(n, N) \qquad d = Rs + n$$

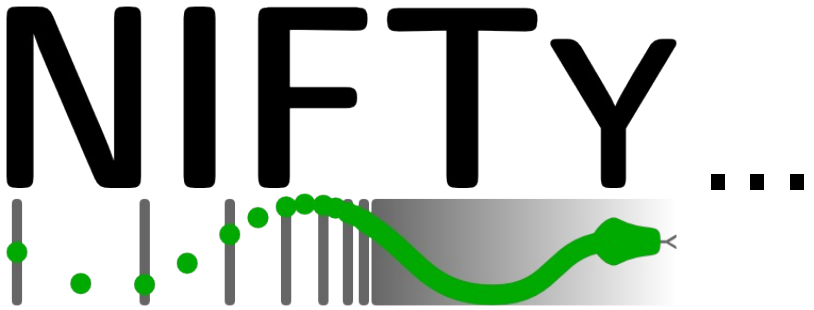
$$m = \underbrace{\left(S^{-1} + R^\dagger N^{-1} R\right)^{-1}}_{D} \underbrace{\left(R^\dagger N^{-1} d\right)}_{j}$$


NIFTy



Selig et al. (2013)
arXiv: 1301.4499

- **is a versatile PYTHON library incorporating CYTHON, C++, and C libraries for performance**



...

Selig et al. (2013)
arXiv: 1301.4499

- is a versatile PYTHON library incorporating CYTHON, C++, and C libraries for performance
- **abstracts spaces, fields, and operators into an object-orientated framework**

NIFTy classes

Selig et al. (2013)
arXiv: 1301.4499

- para-meters

- domain space
- field values

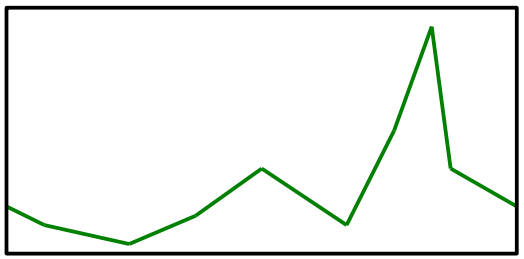
- domain space
- target space
- instance methods applying to fields

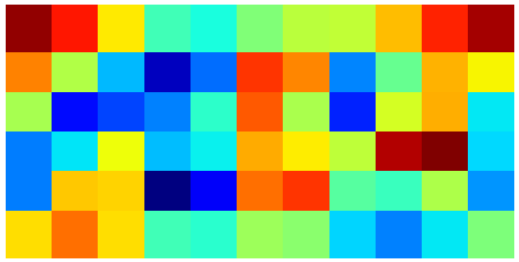
NIFTy classes

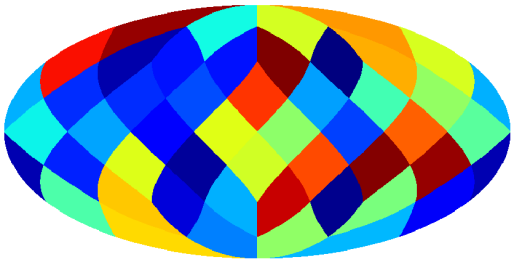
Selig et al. (2013)
arXiv: 1301.4499

- point_space
- rg_space
- lm_space
- hp_space
- gl_space
- nested_space

- unstructured list of points
- n-dimensional regular grid
- spherical harmonics
- HEALPix grid
- Gauss-Legendre grid
- (arbitrary product of grids)

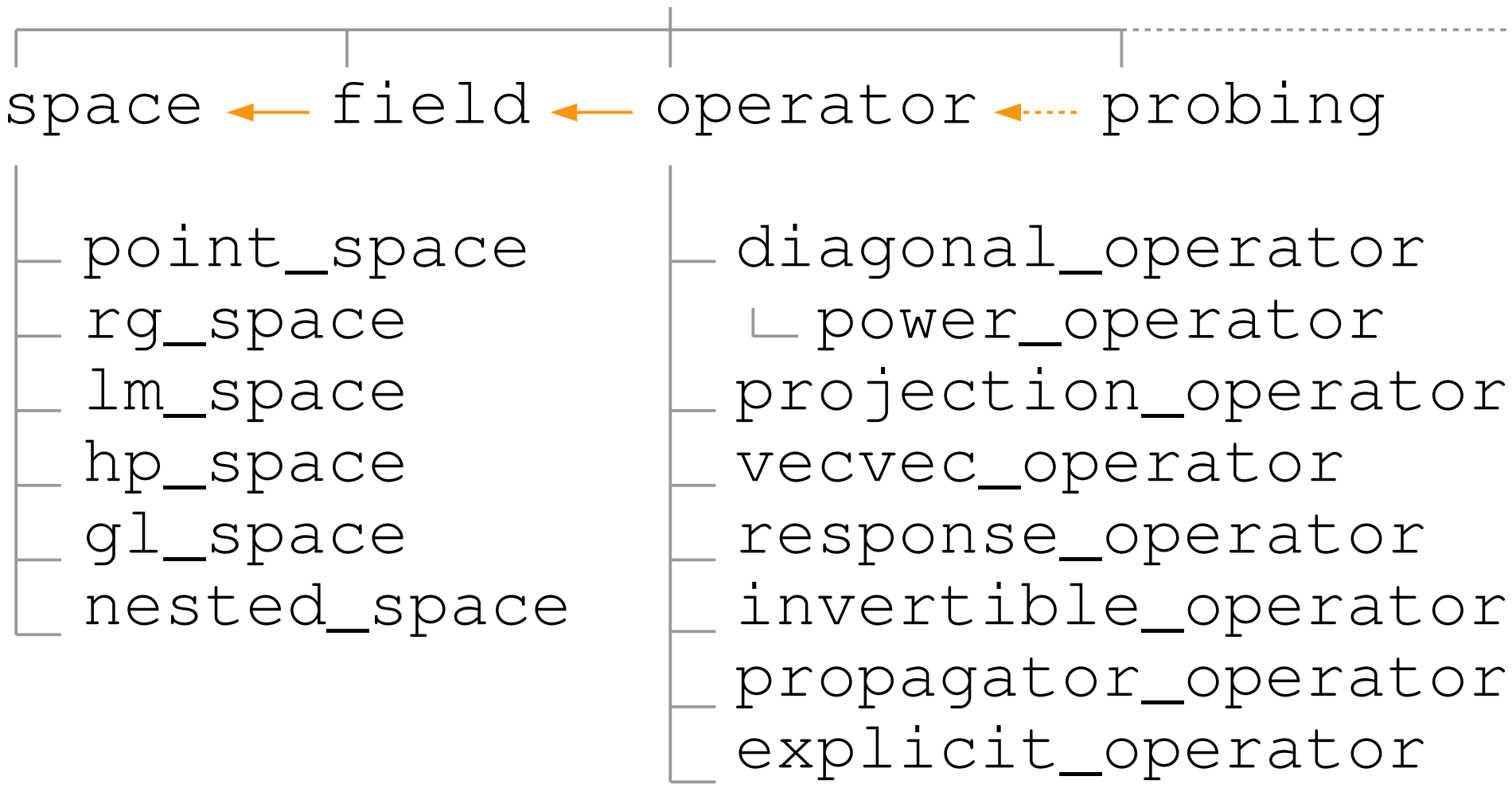


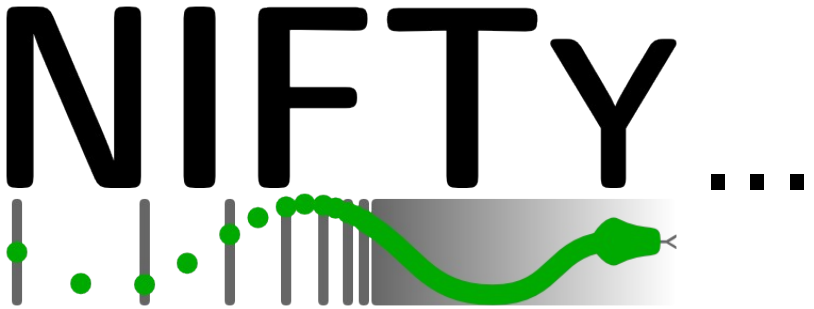




NIFTy classes

Selig et al. (2013)
arXiv: 1301.4499





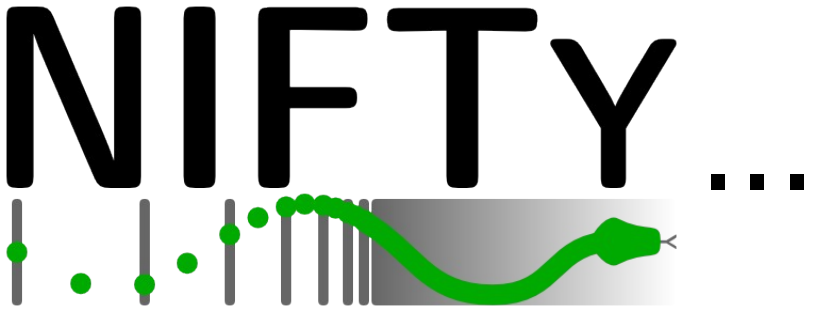
Selig et al. (2013)
arXiv: 1301.4499

- is a versatile PYTHON library incorporating CYTHON, C++, and C libraries for performance
- abstracts spaces, fields, and operators into an object-orientated framework
- **operates regardless of the underlying spatial grid and its resolution**

NIFTy ...

Selig et al. (2013)
arXiv: 1301.4499

- is a versatile PYTHON library incorporating CYTHON, C++, and C libraries for performance
- abstracts spaces, fields, and operators into an object-orientated framework
- operates regardless of the underlying spatial grid and its resolution
- **provides useful tools for the development of statistical inference algorithms (optimization, operator probing, visualization, etc.)**

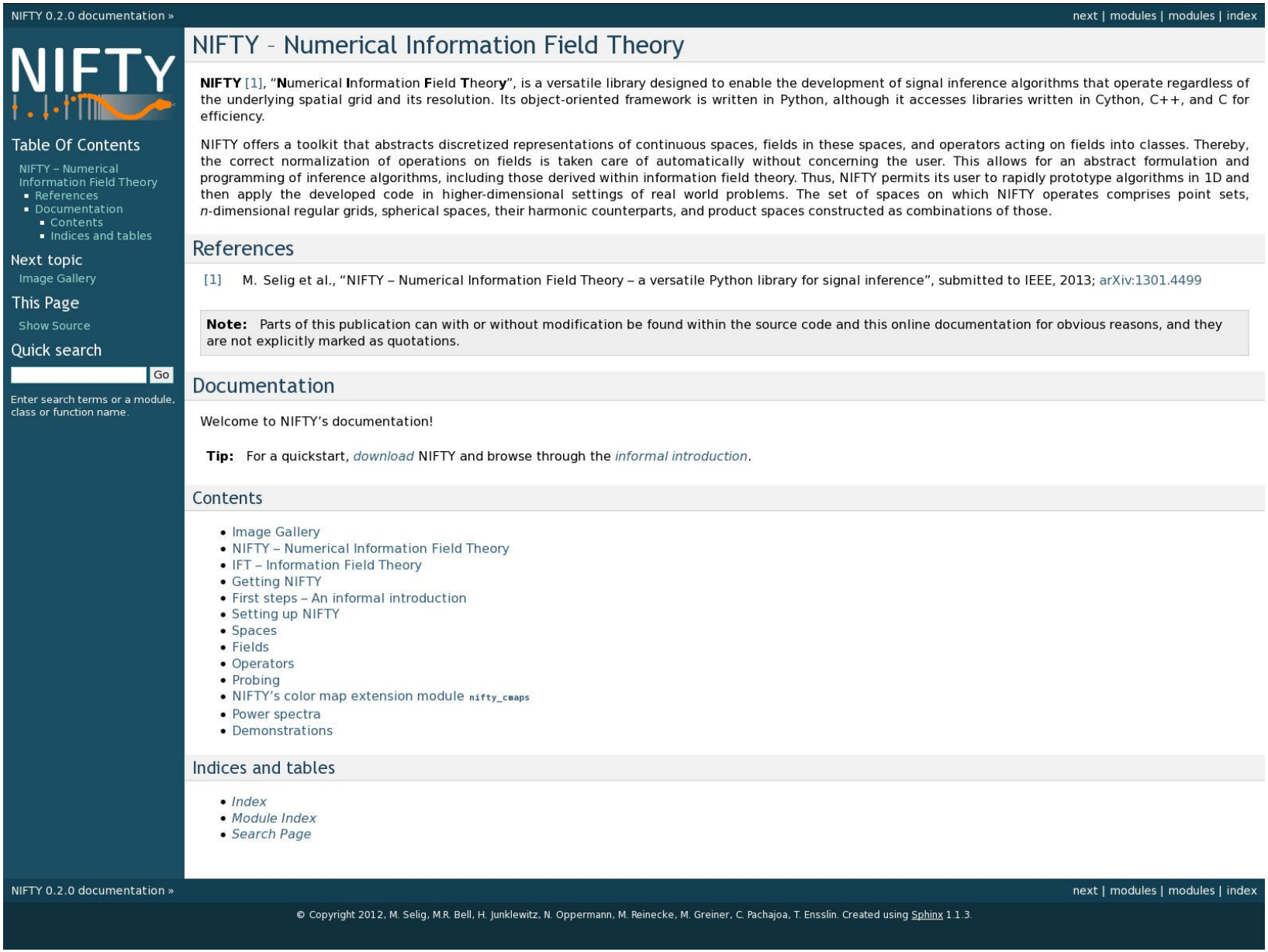


...

Selig et al. (2013)
arXiv: 1301.4499

- is a versatile PYTHON library incorporating CYTHON, C++, and C libraries for performance
- abstracts spaces, fields, and operators into an object-orientated framework
- operates regardless of the underlying spatial grid and its resolution
- provides useful tools for the development of statistical inference algorithms (transformations, operator probing, visualization, etc.)
- **includes extensive on-line documentation**

NIFTy project homepage: http://www.mpa-garching.mpg.de/ift/nifty/



NIFTY 0.2.0 documentation » next | modules | modules | index

NIFTy

Table Of Contents

NIFTY – Numerical Information Field Theory
- References
- Documentation
  - Contents
  - Indices and tables

Next topic
Image Gallery

This Page
Show Source

Quick search
Go
Enter search terms or a module, class or function name.

NIFTY – Numerical Information Field Theory

NIFTY [1], "**N**umerical **I**nformation **F**ield **T**heor**y**", is a versatile library designed to enable the development of signal inference algorithms that operate regardless of the underlying spatial grid and its resolution. Its object-oriented framework is written in Python, although it accesses libraries written in Cython, C++, and C for efficiency.

NIFTY offers a toolkit that abstracts discretized representations of continuous spaces, fields in these spaces, and operators acting on fields into classes. Thereby, the correct normalization of operations on fields is taken care of automatically without concerning the user. This allows for an abstract formulation and programming of inference algorithms, including those derived within information field theory. Thus, NIFTY permits its user to rapidly prototype algorithms in 1D and then apply the developed code in higher-dimensional settings of real world problems. The set of spaces on which NIFTY operates comprises point sets, *n*-dimensional regular grids, spherical spaces, their harmonic counterparts, and product spaces constructed as combinations of those.

References

[1] M. Selig et al., "NIFTY – Numerical Information Field Theory – a versatile Python library for signal inference", submitted to IEEE, 2013; arXiv:1301.4499

Note: Parts of this publication can with or without modification be found within the source code and this online documentation for obvious reasons, and they are not explicitly marked as quotations.

Documentation

Welcome to NIFTY's documentation!

Tip: For a quickstart, *download* NIFTY and browse through the *informal introduction*.

Contents

- Image Gallery
- NIFTY – Numerical Information Field Theory
- IFT – Information Field Theory
- Getting NIFTY
- First steps – An informal introduction
- Setting up NIFTY
- Spaces
- Fields
- Operators
- Probing
- NIFTY's color map extension module `nifty_cmaps`
- Power spectra
- Demonstrations

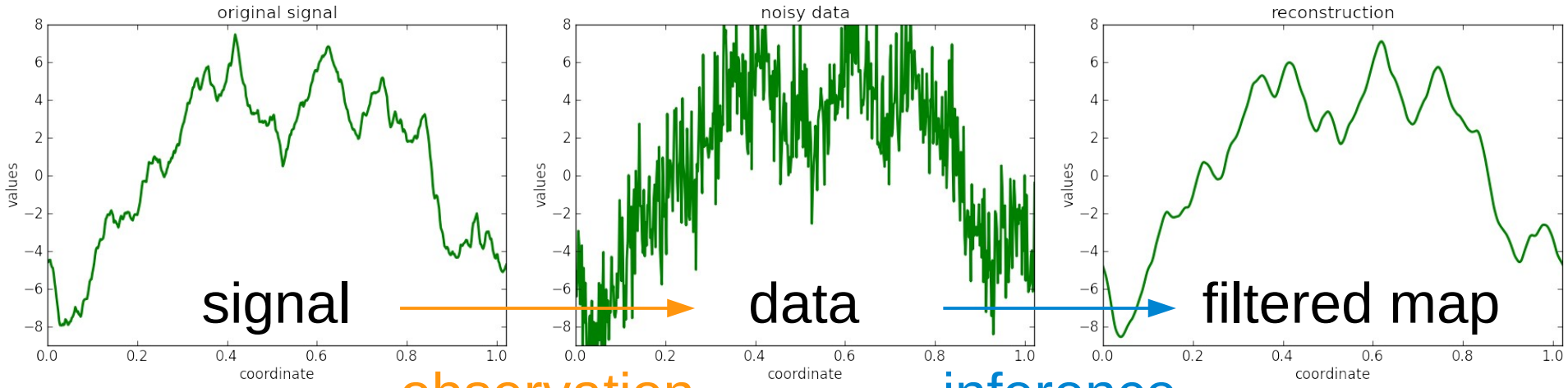
Indices and tables

- *Index*
- *Module Index*
- *Search Page*

NIFTY 0.2.0 documentation » next | modules | modules | index

Demo

Wiener filtering, revisited



$$\boldsymbol{d} = \boldsymbol{R}\boldsymbol{s} + \boldsymbol{n} \qquad \boldsymbol{s} \curvearrowleft \mathcal{G}(\boldsymbol{s}, \boldsymbol{S}) \qquad \boldsymbol{n} \curvearrowleft \mathcal{G}(\boldsymbol{n}, \boldsymbol{N})$$

$$\boldsymbol{m} = \langle \boldsymbol{s} \rangle_{(\boldsymbol{s}|\boldsymbol{d})} = \underbrace{\left(\boldsymbol{S}^{-1} + \boldsymbol{R}^\dagger \boldsymbol{N}^{-1} \boldsymbol{R}\right)^{-1}}_{\boldsymbol{D}} \underbrace{\left(\boldsymbol{R}^\dagger \boldsymbol{N}^{-1} \boldsymbol{d}\right)}_{\boldsymbol{j}}$$

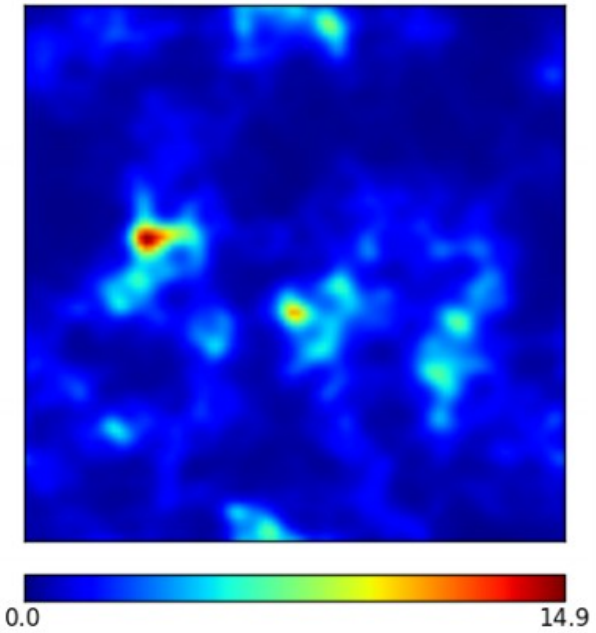
Go here to follow along: http://tinyurl.com/nifty-demo-notebook

Applications

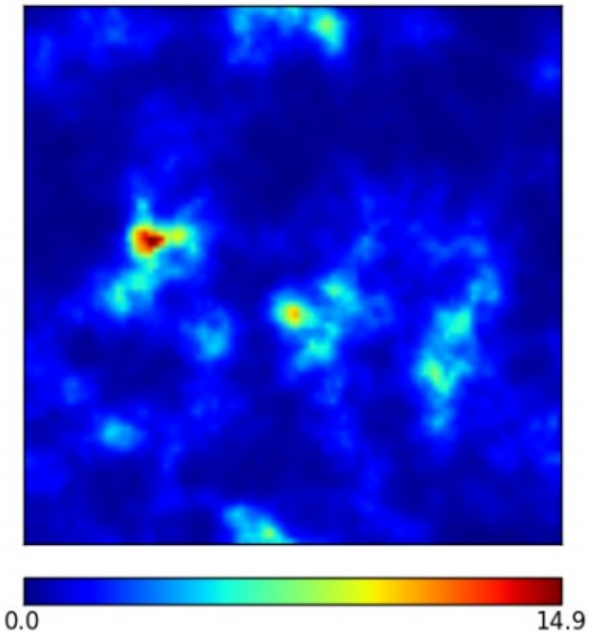
The RESOLVE algorithm:
Improved image reconstruction for radio astronomy.

Narrow bandwidth image reconstruction.

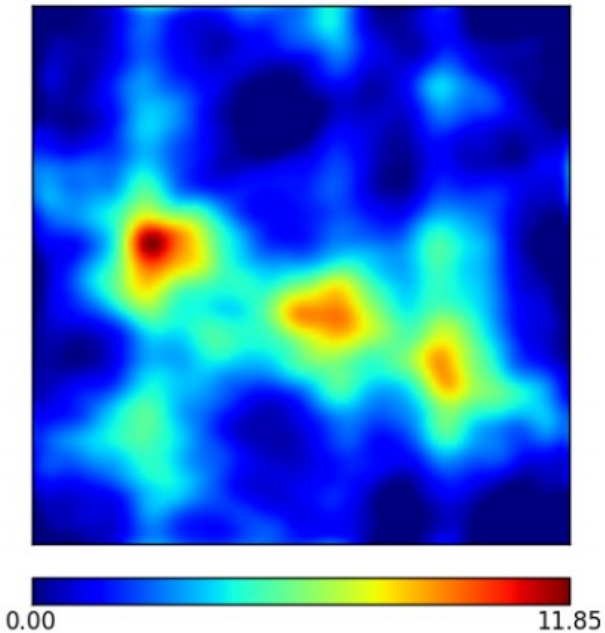
Junklewitz, Bell, Selig, Enßlin (2014)
http://arxiv.org/abs/1401.4711



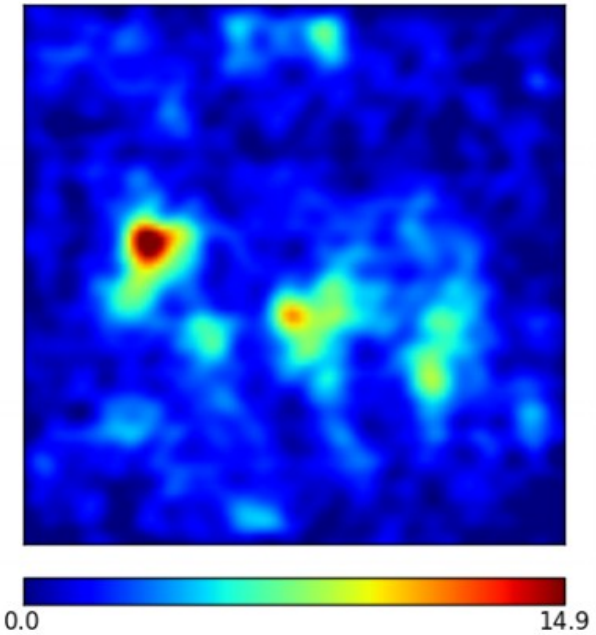
RESOLVE reconstruction



Original signal



Naive reconstruction

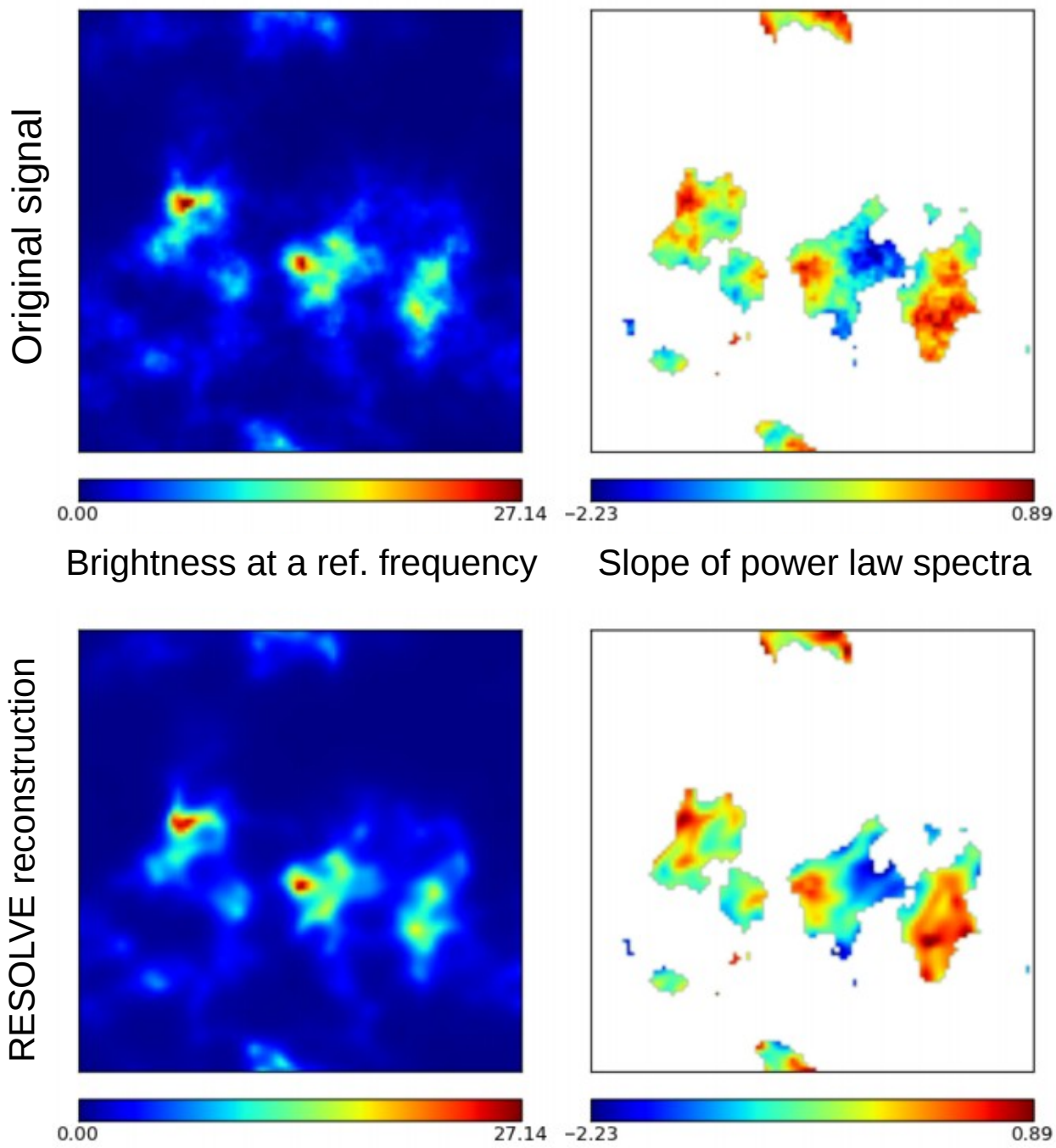


Typical reconstruction

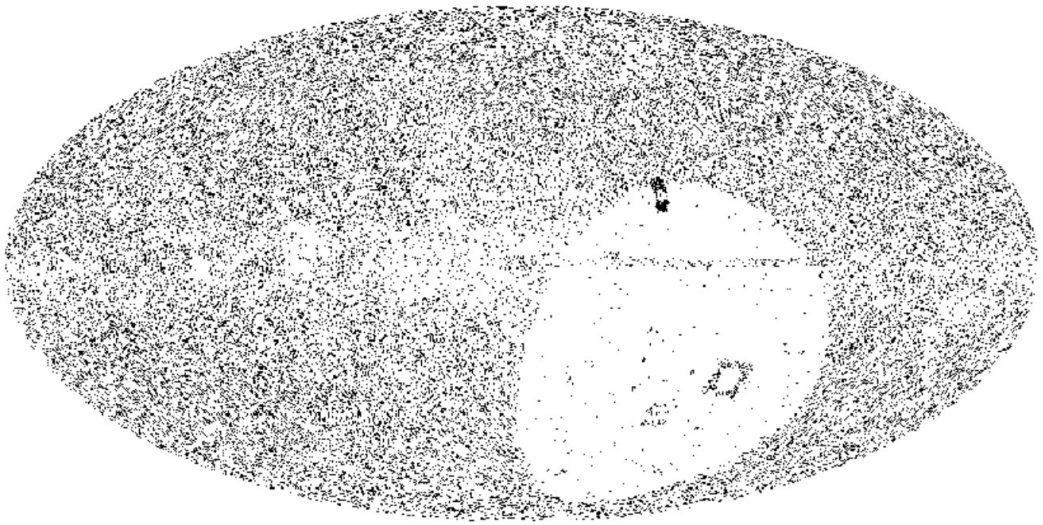
The RESOLVE algorithm:
Improved image reconstruction for radio astronomy.

Broad bandwidth image reconstruction.

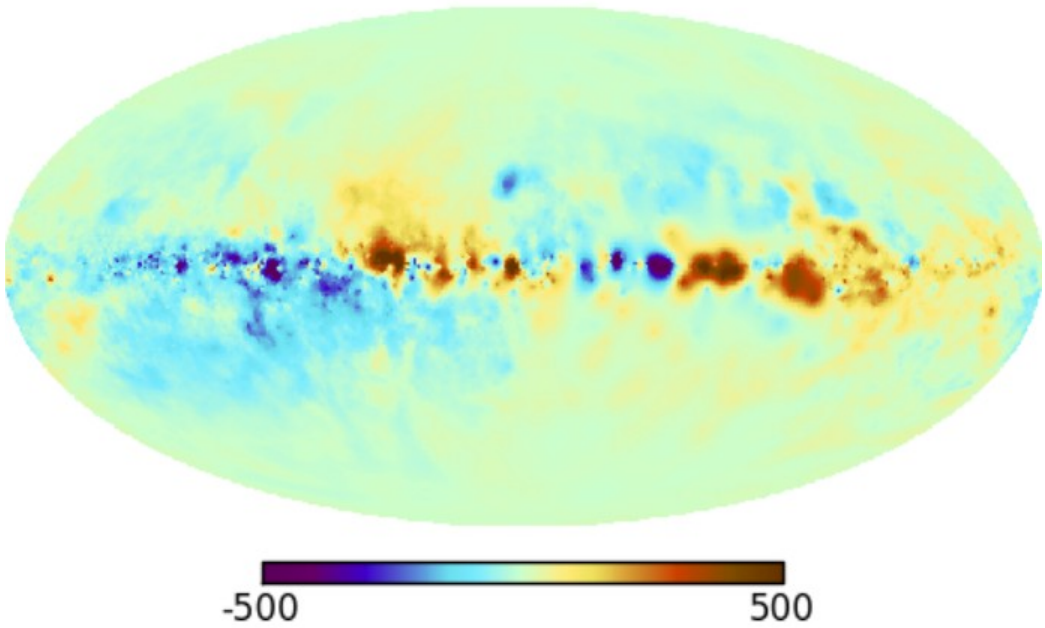
Junklewitz, Bell, Enßlin (2014)
http://arxiv.org/abs/1311.5282



Locations of 41,330 measurements



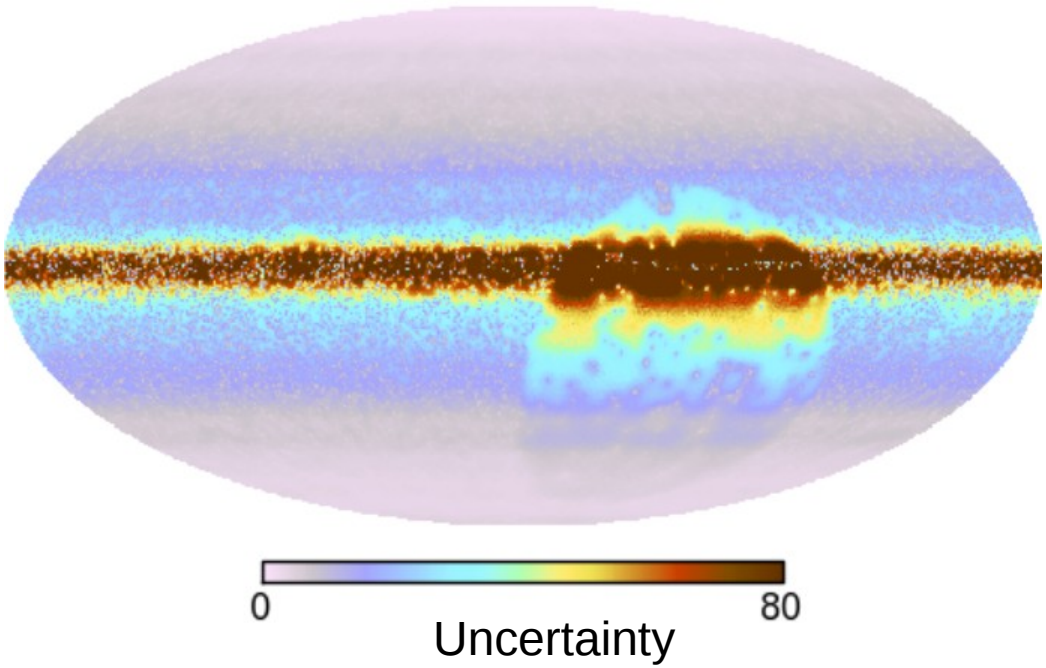
Inferred Galactic Faraday depth



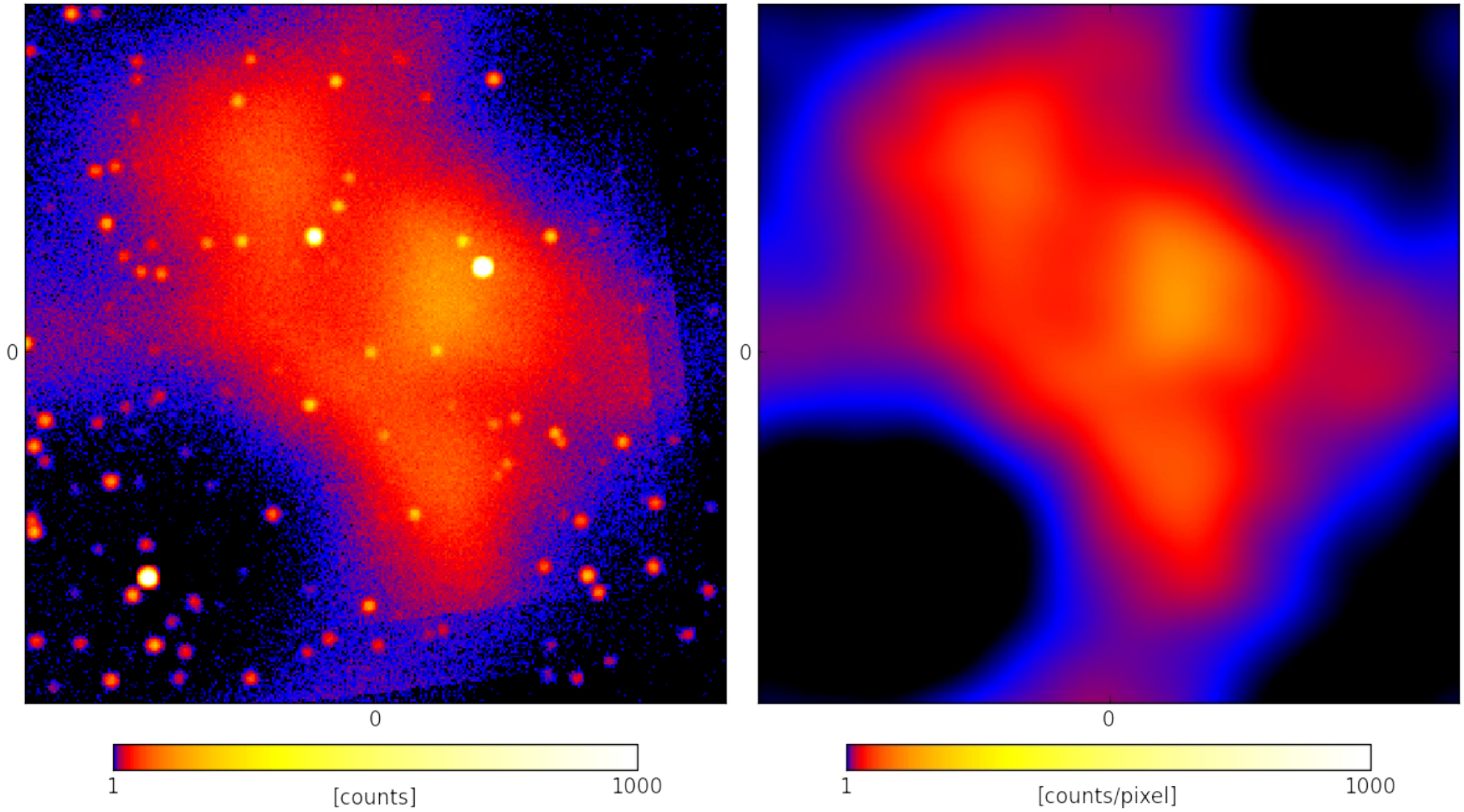
Studying magnetic fields in the Milky Way through inference of “Faraday depth.”

Oppermann, et al. (2012)
http://arxiv.org/abs/1111.6186

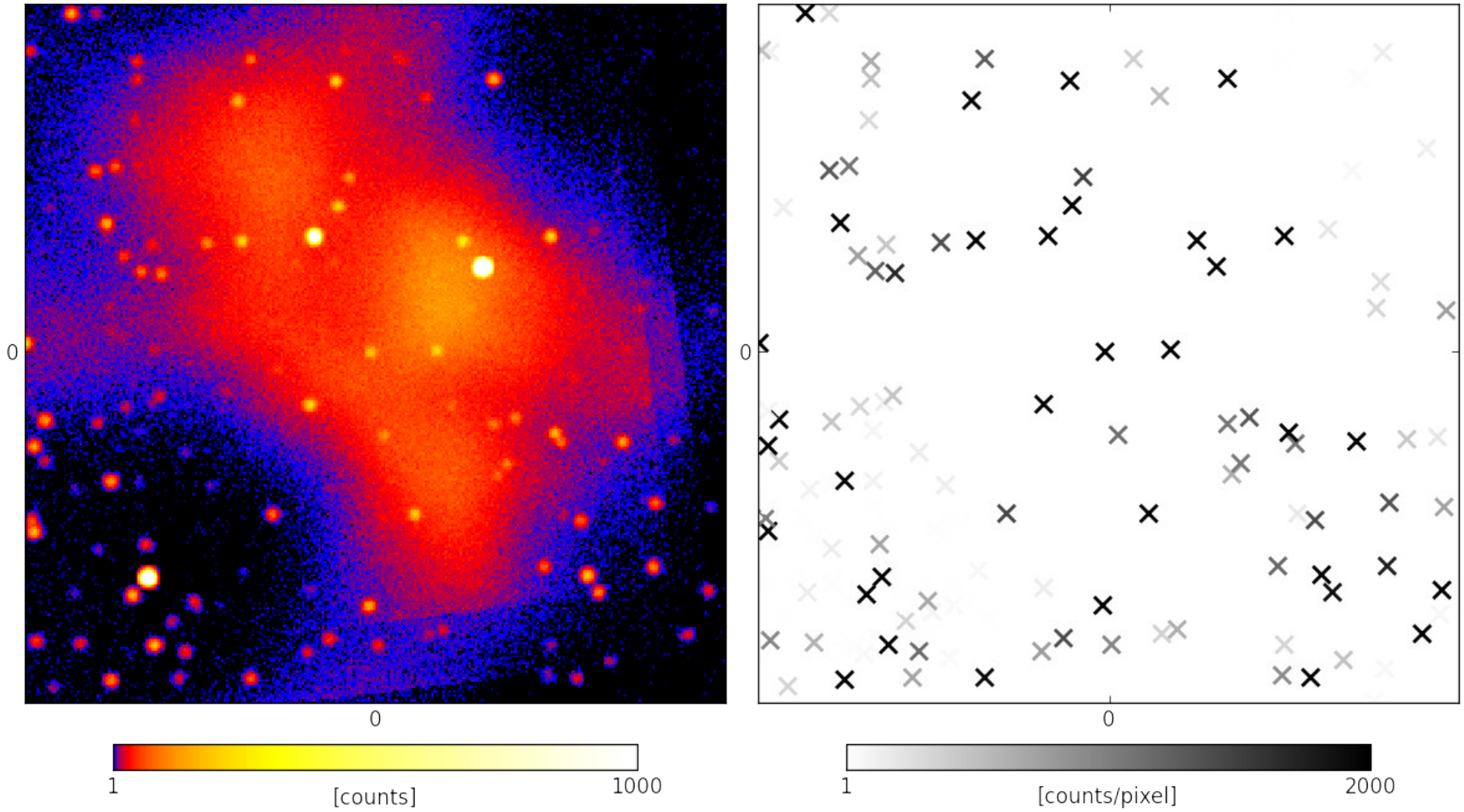
Oppermann, et al. (2014)
http://arxiv.org/abs/1404.3701

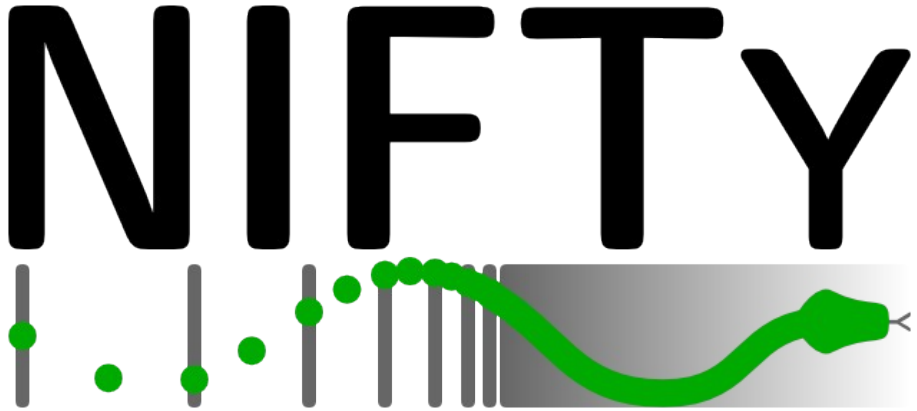


Denoising, Deconvolving, and Decomposing Photon Observations – **D^{3}PO**
(Selig, et al., in prep.)



Denoising, Deconvolving, and Decomposing Photon Observations – **D^{3}PO**
(Selig, et al., in prep.)





NIFTy project homepage: http://www.mpa-garching.mpg.de/ift/nifty/

and on GitHub: https://github.com/mselig/nifty

bellmr@gmail.com
@mryanbell

Thank you!