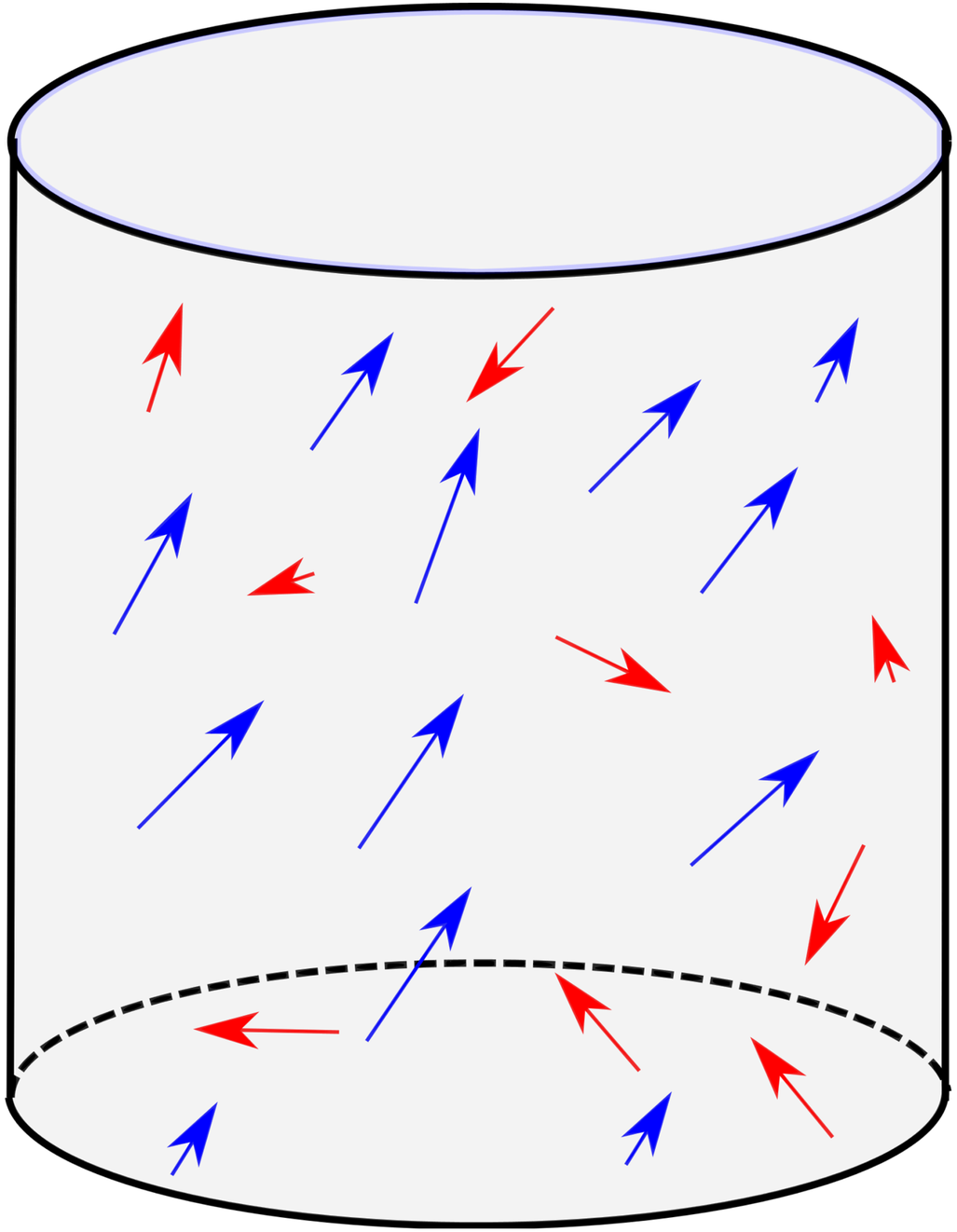
Magali: open software for inversion and analysis of magnetic microscopy data 🧲🔬

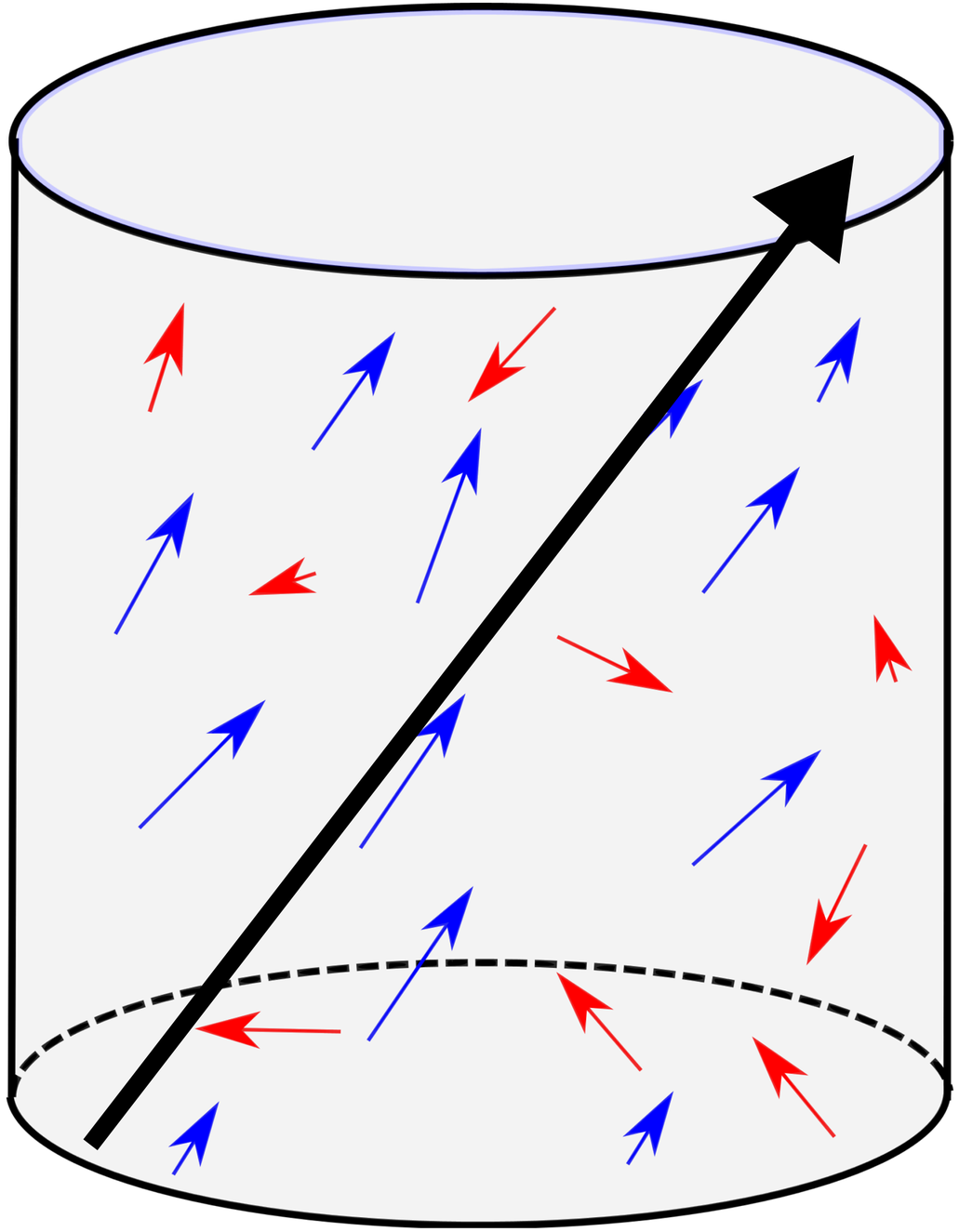
Yago Moreira Castro, Leonardo Uieda, Gelson Ferreira de Souza-Junior

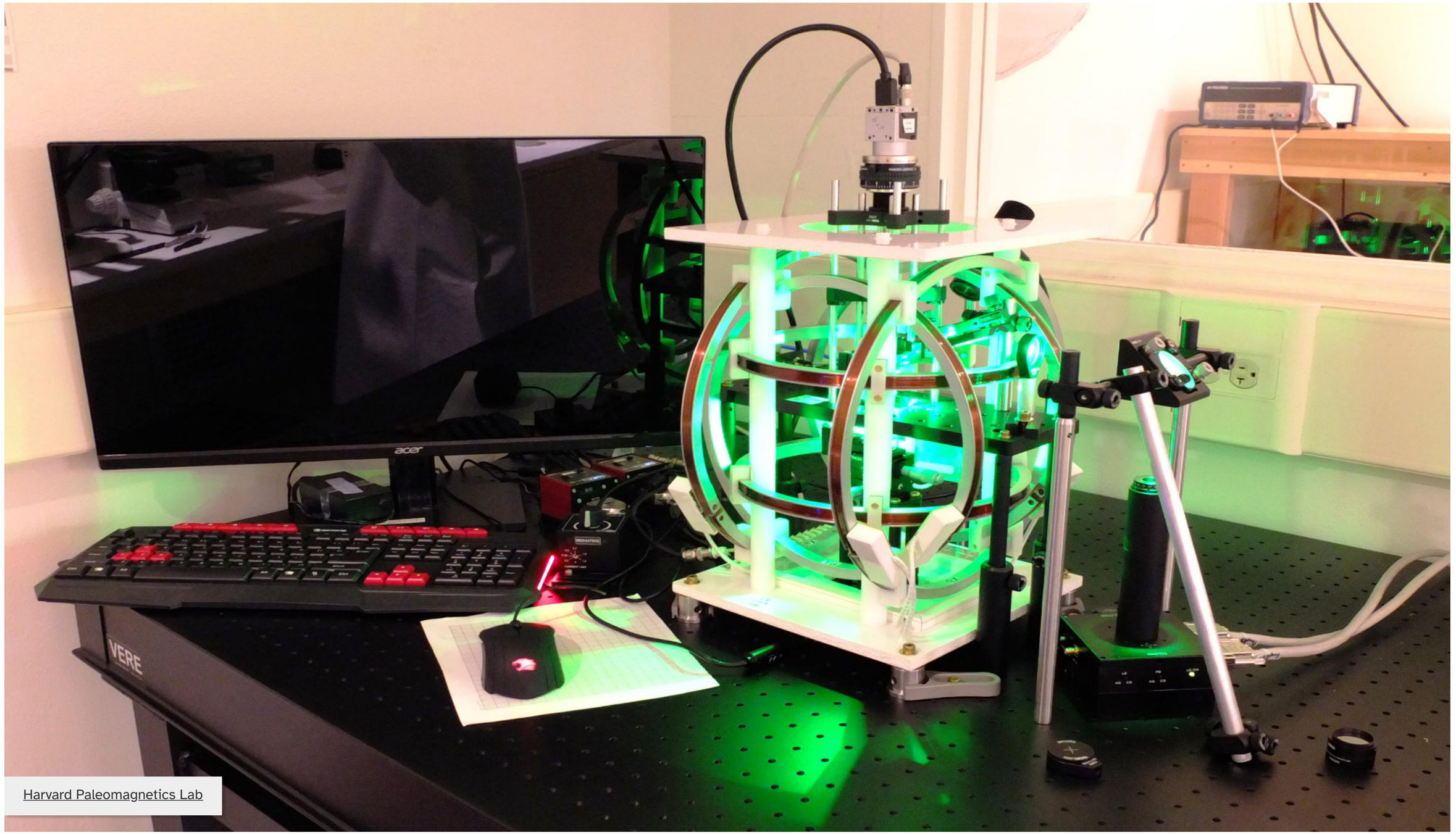
20 of October 2025 8^{th} Biennial LATINMAG Meeting | Morelia, Mexico

COMPUTER-ORIENTED GEOSCIENCE LAB

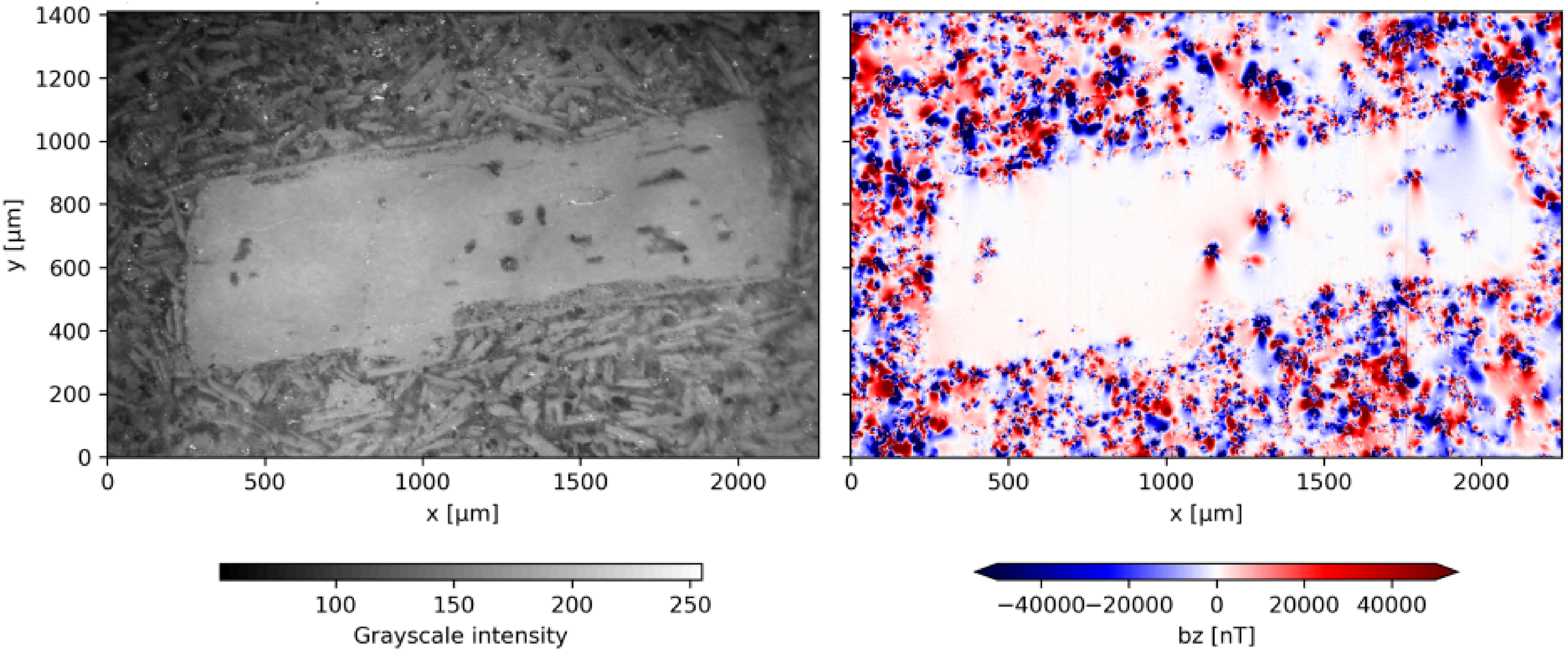
INSTITUTO DE ASTRONOMIA, GEOFÍSICA E CIÊNCIAS ATMOSFÉRICAS







Harvard Paleomagnetics Lab



Souza-Junior et al 2024

Research Article | Open Access

Does size matter? Statistical limits of paleomagnetic field reconstruction from small rock specimens

Thomas Berndt, Adrian R. Muxworthy, Karl Fabian

First published: 14 December 2015 | https://doi.org/10.1002/2015JB012441 | Citations: 46

The copyright line for this article was changed on 11 FEB 2016 after original online publication.

Geophysical Research Letters®

Research Letter | Open Access

Efficiency of Thermoremanent Magnetization Acquisition in Vortex-State Particle Assemblies

U. D. Bellon, W. Williams, A. R. Muxworthy, G. F. Souza-Junior, L. Nagy, L. Uieda, R. I. F. Trindade

First published: 23 April 2025 | https://doi.org/10.1029/2025GL114771 | Citations: 2

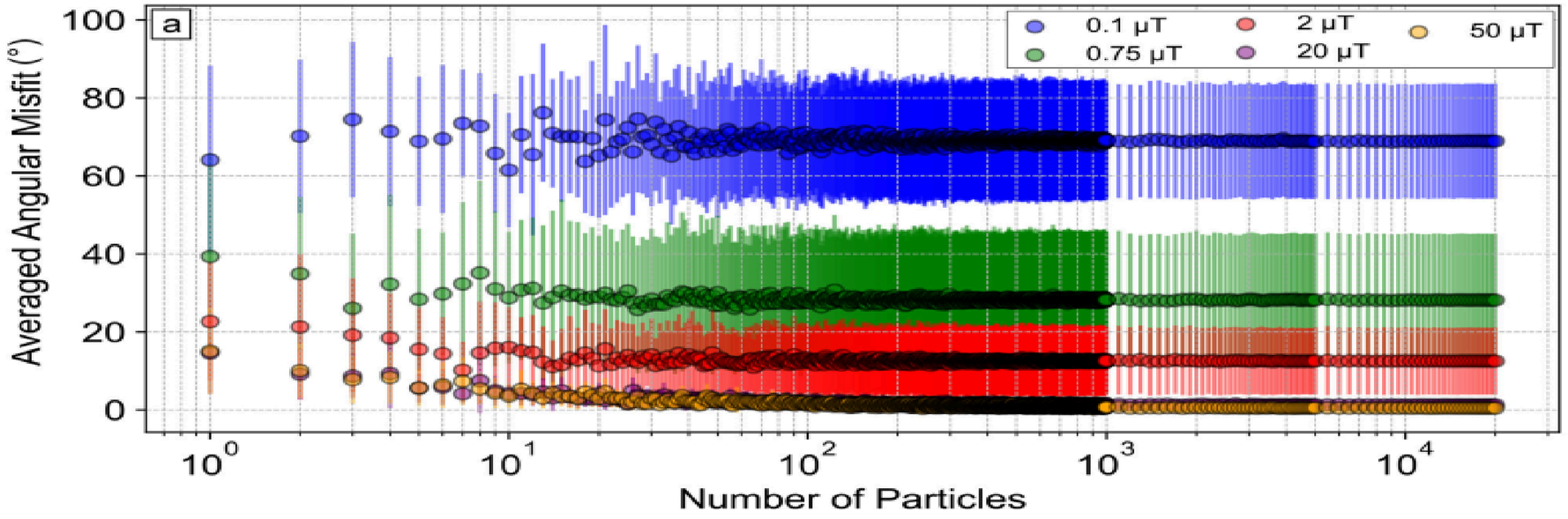
Bellon et al. (2025)

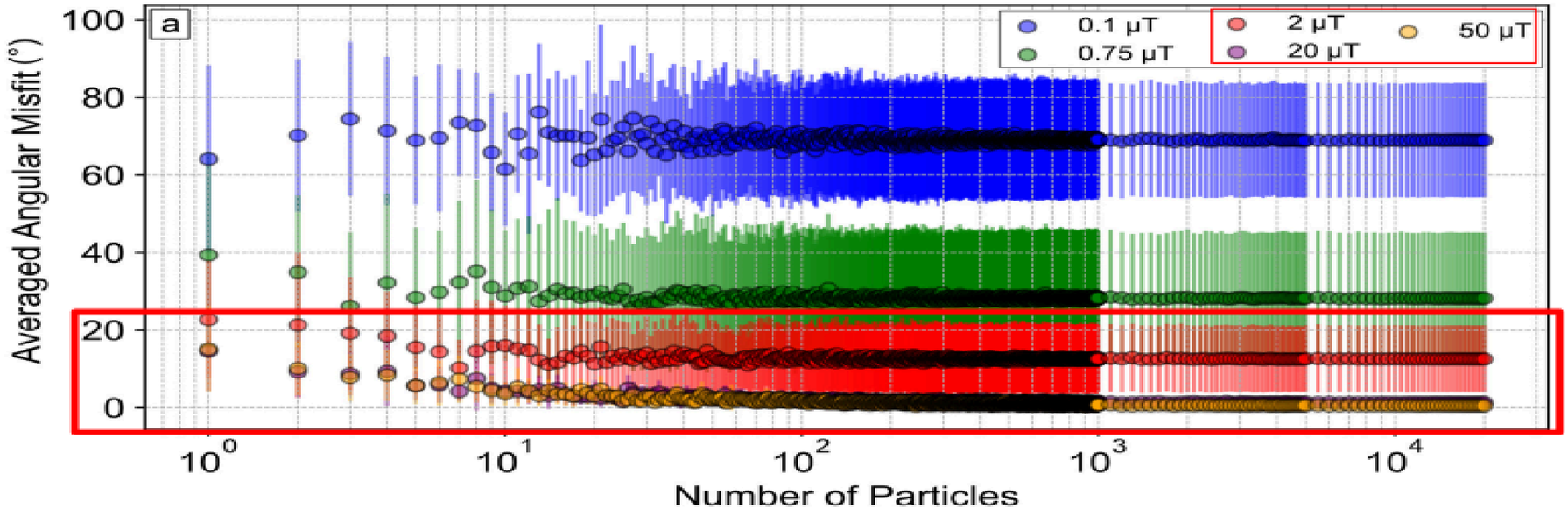
- Bellon et al. (2025) modeled vortex-state grains, a more realistic scenario, using micromagnetic simulations.

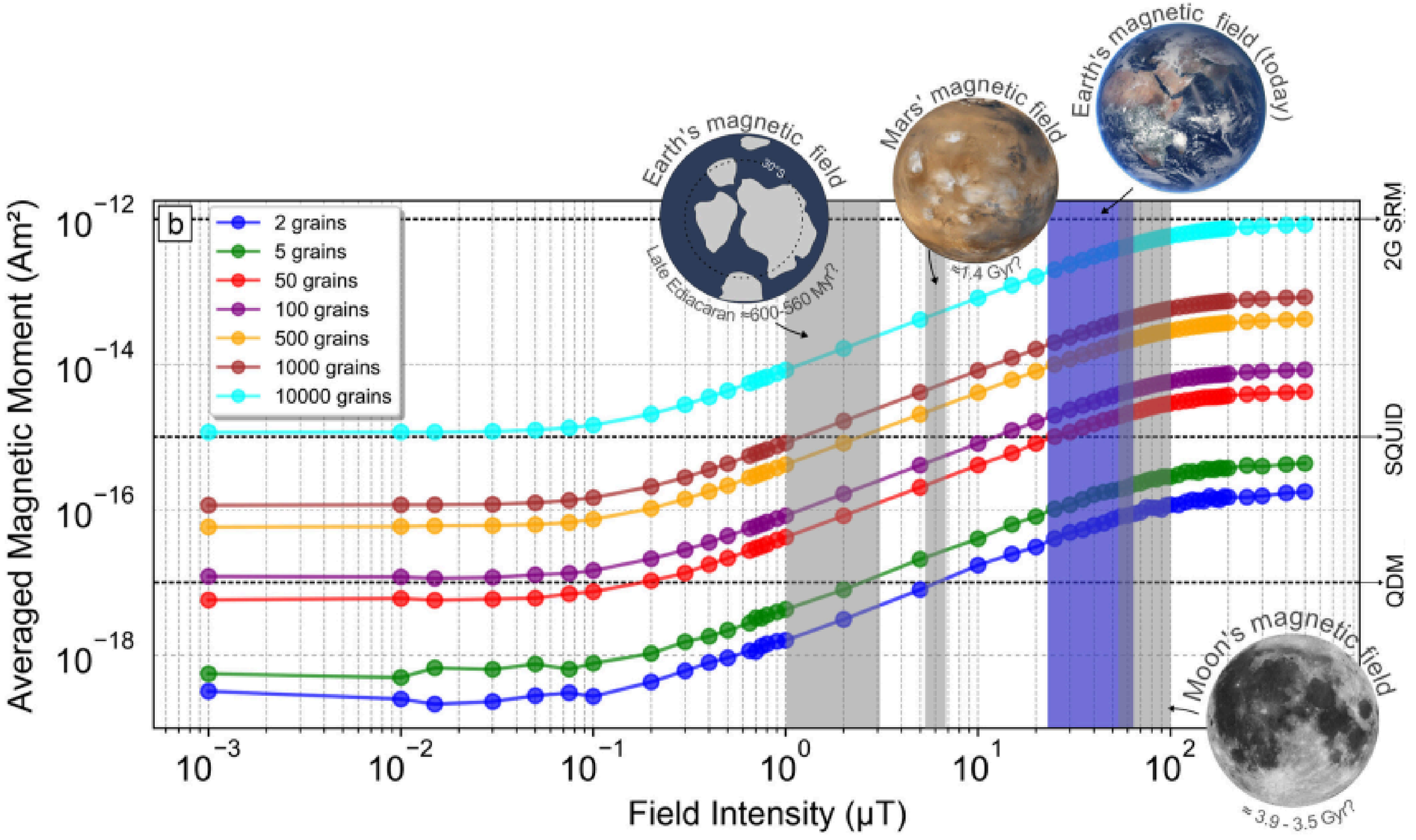
Bellon et al. (2025)

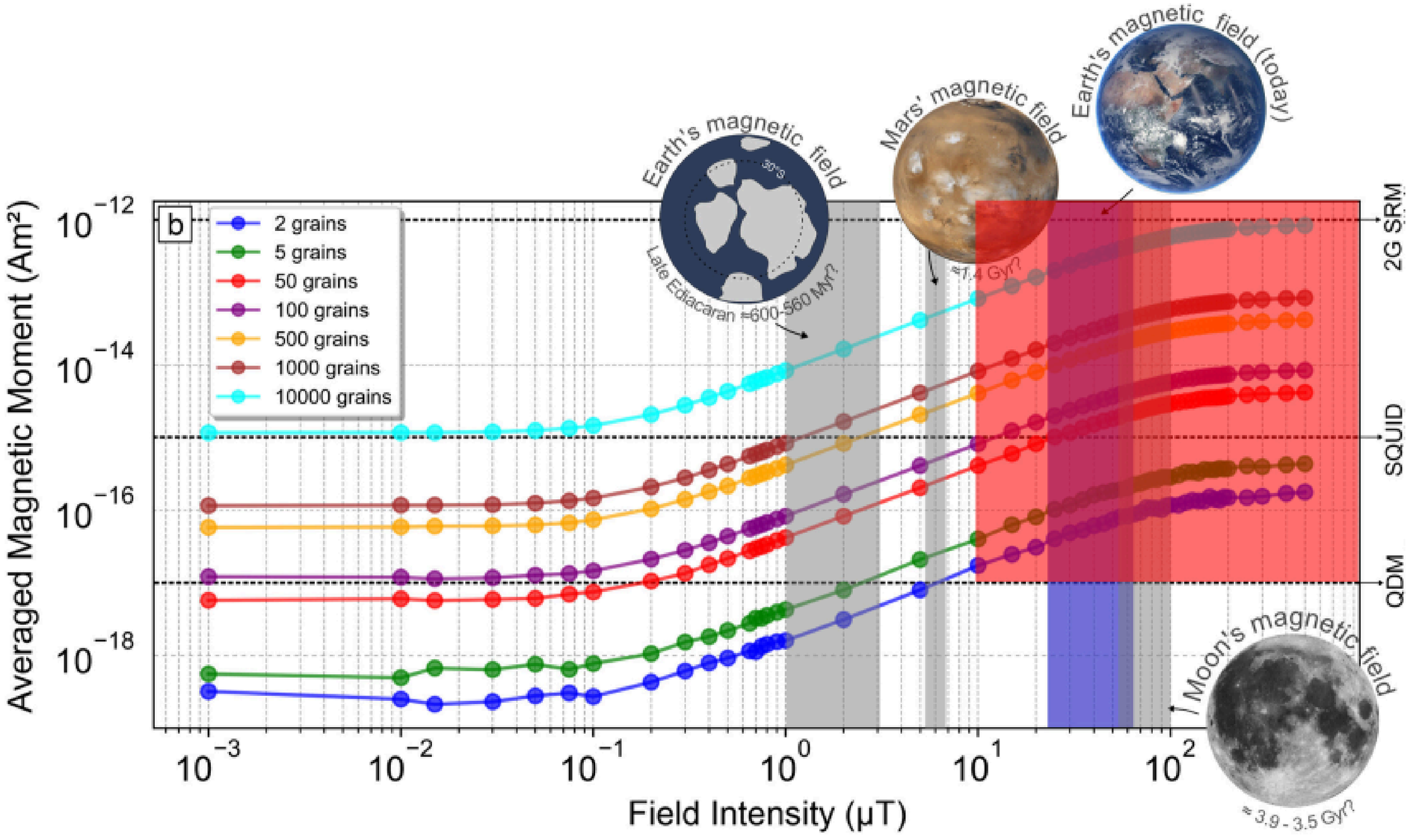
- Bellon et al. (2025) modeled vortex-state grains, a more realistic scenario, using micromagnetic simulations.
- **Hundreds to thousands** of vortex-state grains record a reliable TRM, indicating that the field was not ultra-weak.

Bellon et al. (2025)


a
0.1 µT
0.75 µT
2 µT
20 µT
50 µT
Averaged Angular Misfit (°)
100
80
60
40
20
0
10^0
10^1
10^2
10^3
10^4
Number of Particles


a
0.1 µT
0.75 µT
2 µT
20 µT
50 µT
Averaged Angular Misfit (°)
100
80
60
40
20
0
10^0
10^1
10^2
10^3
10^4
Number of Particles


b
2 grains
5 grains
50 grains
100 grains
500 grains
1000 grains
10000 grains
Averaged Magnetic Moment (Am²)
Field Intensity (μT)
2G SRM
SQUID
QDM
Earth's magnetic field
Late Ediacaran ≈600-560 Myr?
Mars' magnetic field
≈1.4 Gyr?
Earth's magnetic field (today)
Moon's magnetic field
≈ 3.9 - 3.5 Gyr?


b
2 grains
5 grains
50 grains
100 grains
500 grains
1000 grains
10000 grains
Averaged Magnetic Moment (Am²)
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Field Intensity (μT)
2G SRM
SQUID
QDM
Earth's magnetic field
Late Ediacaran ≈600-560 Myr?
Mars' magnetic field
≈1.4 Gyr?
Earth's magnetic field (today)
Moon's magnetic field
≈ 3.9 - 3.5 Gyr?

Full Vector Inversion of Magnetic Microscopy Images Using Euler Deconvolution as Prior Information

Gelson F. Souza-Junior[1], Leonardo Uieda[1,2], Ricardo I. F. Trindade[1], Janine Carmo[1], and Roger Fu[3]

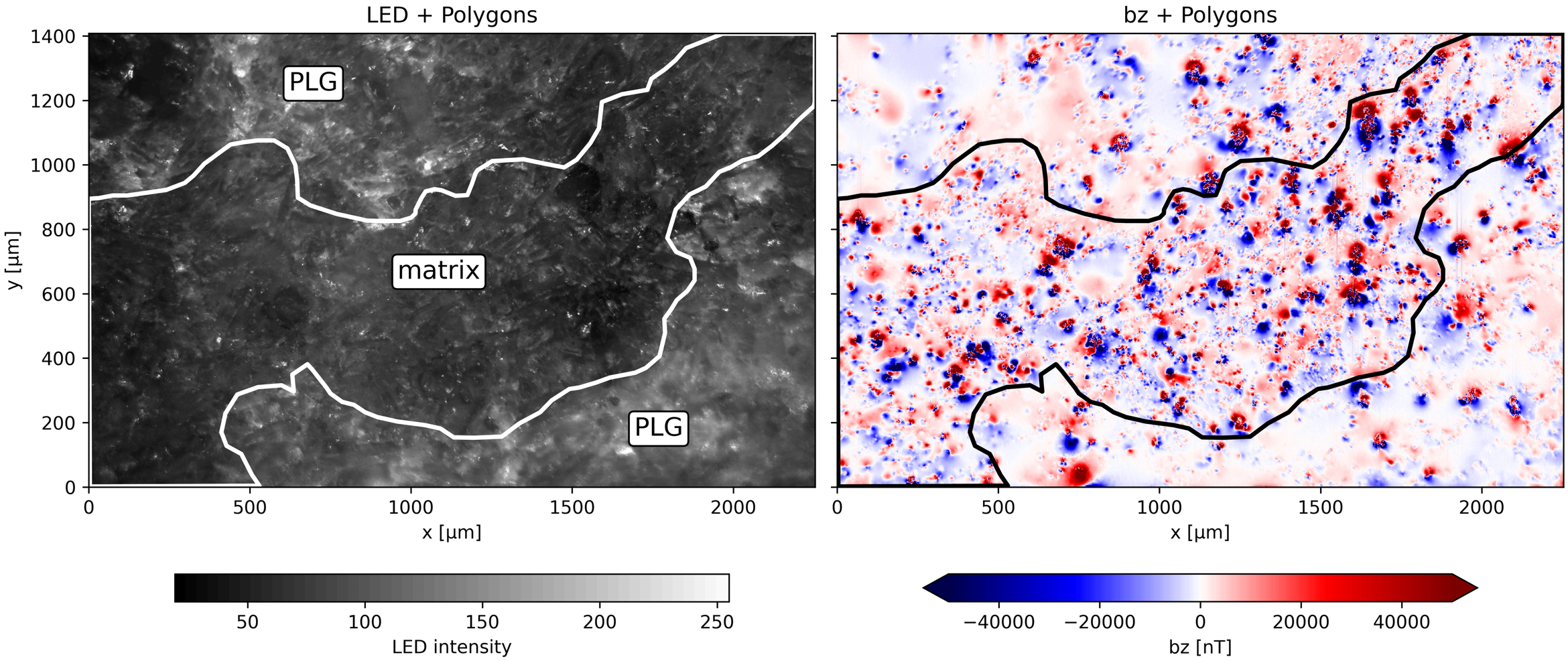
[1]Universidade de São Paulo, São Paulo, Brazil, [2]University of Liverpool, Liverpool, UK, [3]Harvard University, Cambridge, MA, USA

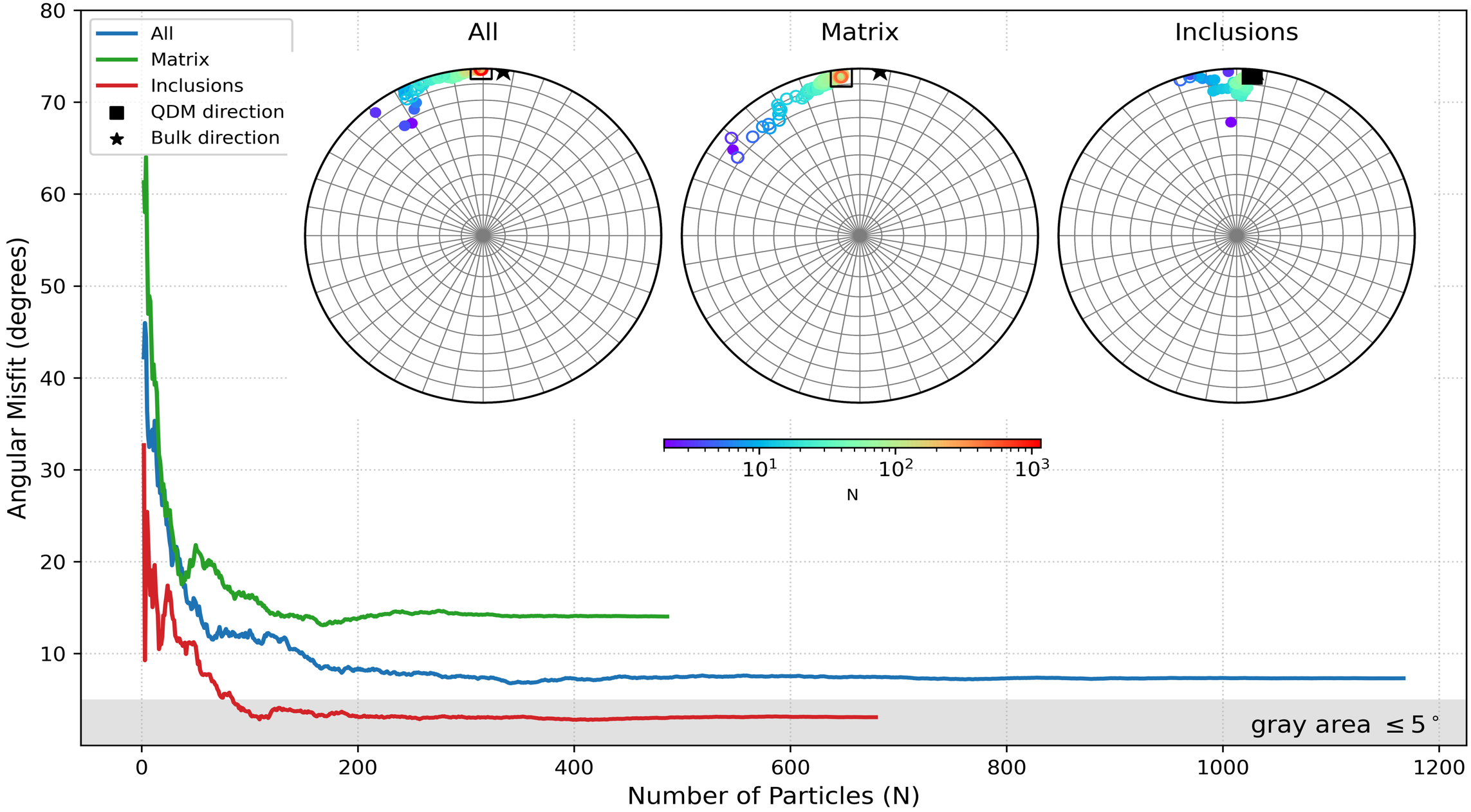
Robust directional analysis of magnetic microscopy images using non-linear inversion and iterative Euler deconvolution

Gelson F. Souza-Junior[1,2], Leonardo Uieda[1], Ricardo I. F. Trindade[1], Roger R. Fu[2], Ualisson D. Bellon[3] and Yago M. Castro[1]

[1] Universidade de São Paulo, Brazil; [2] Harvard University, USA; [3] University of Edinburgh, UK.

Corresponding author: Gelson F. Souza-Junior <gelson.ferreira@usp.br>


LED + Polygons
bz + Polygons
PLG
matrix
PLG
y [µm]
x [µm]
x [µm]
LED intensity
bz [nT]


All
Matrix
Inclusions
QDM direction
Bulk direction
All
Matrix
Inclusions
10^1
10^2
10^3
N
Angular Misfit (degrees)
Number of Particles (N)
gray area ≤ 5 °
0
200
400
600
800
1000
1200
10
20
30
40
50
60
70
80

Needs

Needs

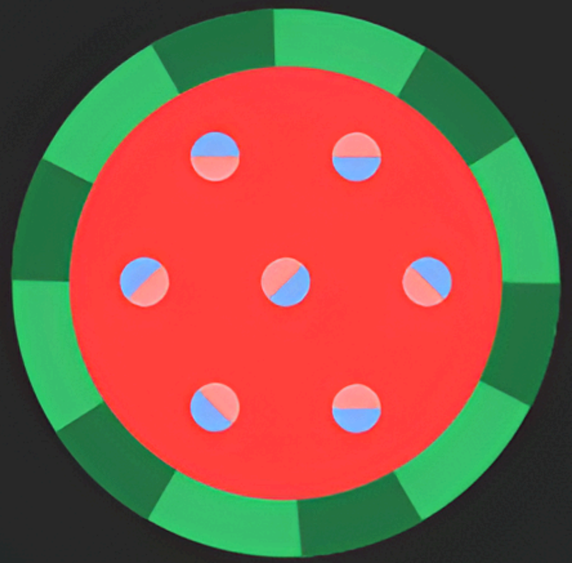
- Algorithms for **automatic detection** of magnetic **grains** and its **magnetic moment** determination

Needs

- Algorithms for **automatic detection** of magnetic **grains** and its **magnetic moment** determination
- **Open software** for forward **modelling** and **inversion** techniques specific to magnetic microscopy

Needs

- Algorithms for **automatic detection** of magnetic **grains** and its **magnetic moment** determination
- **Open software** for forward **modelling** and **inversion** techniques specific to magnetic microscopy
- **Data conventions**



magali

What is Magali?

What is Magali?

Free and open source

What is Magali?

Free and open source

Python library

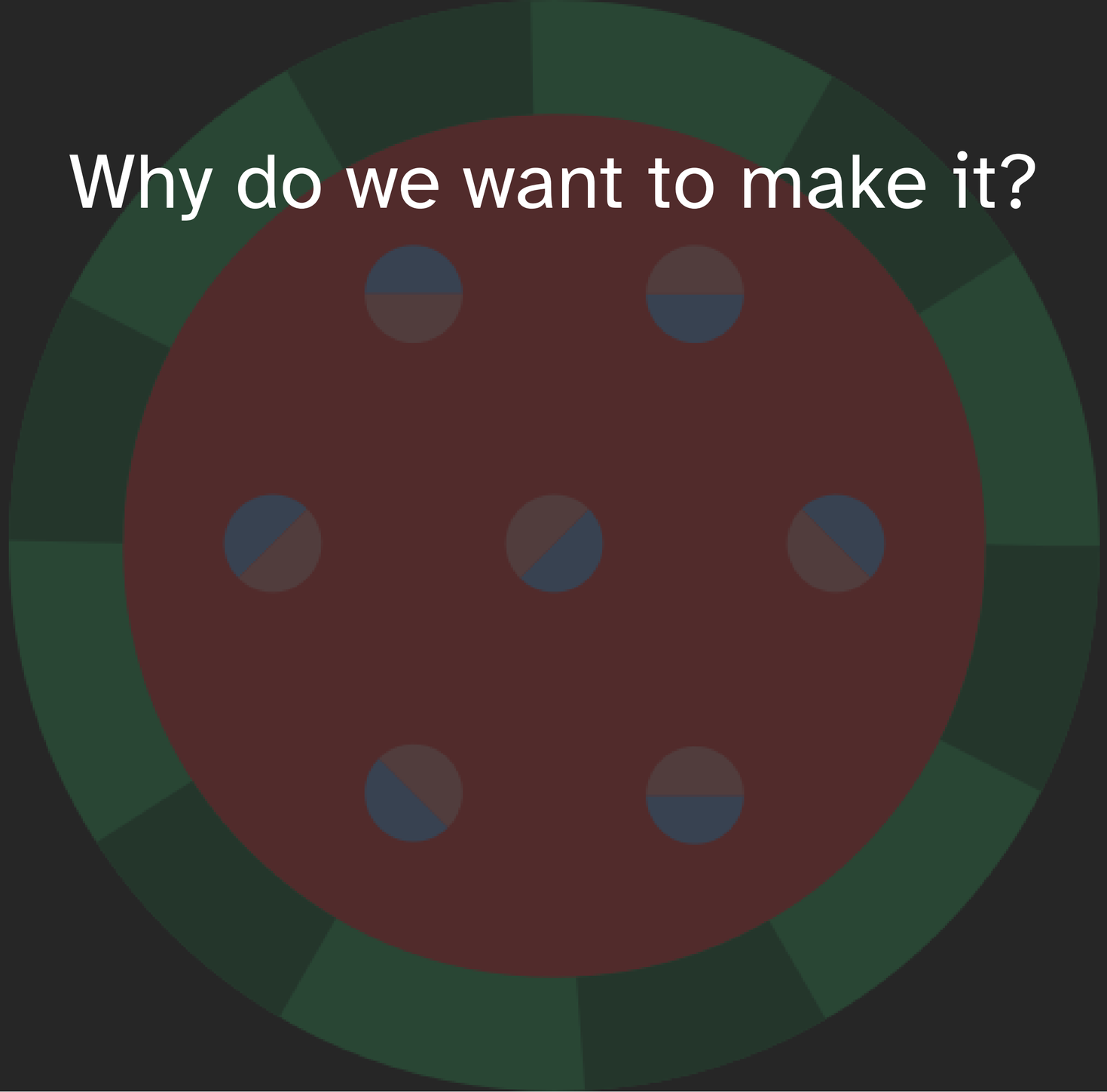
What is Magali?

Free and open source

Python library

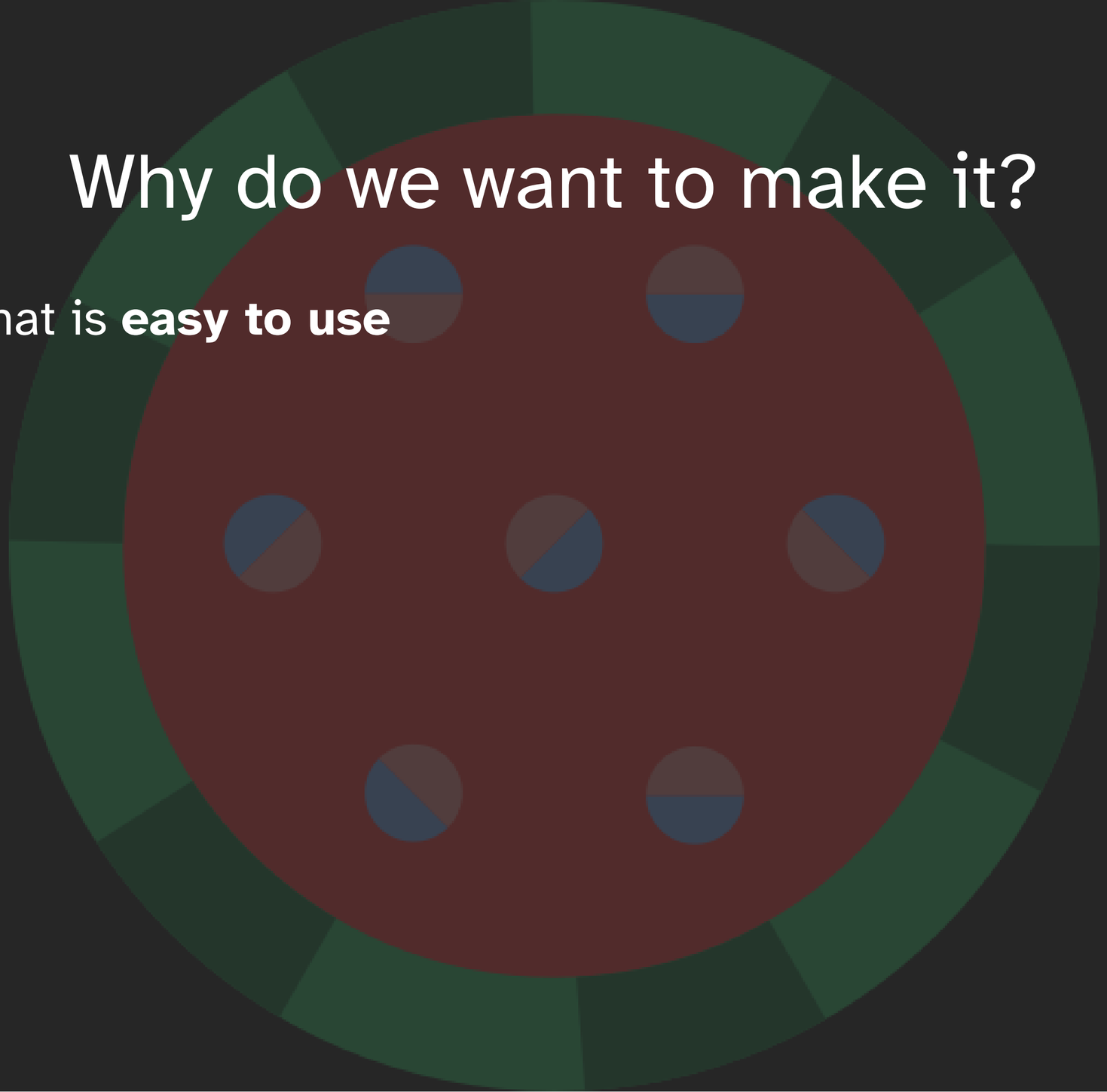
Modelling and processing magnetic microscopy data

Why do we want to make it?



Why do we want to make it?

- Provide code that is **easy to use**



Why do we want to make it?

- Provide code that is **easy to use**
- Determine the **spatial positions** of **multiple** grains

Why do we want to make it?

- Provide code that is **easy to use**
- Determine the **spatial positions** of **multiple** grains
- Facilitate the creation of **synthetic data**

Why do we want to make it?

- Provide code that is **easy to use**
- Determine the **spatial positions** of **multiple** grains
- Facilitate the creation of **synthetic data**
- Propose a standard **data format**

Why do we want to make it?

- Provide code that is **easy to use**
- Determine the **spatial positions** of **multiple** grains
- Facilitate the creation of **synthetic data**
- Propose a standard **data format**
- Serve as a **foundation** for new methods

Why do we want to make it?

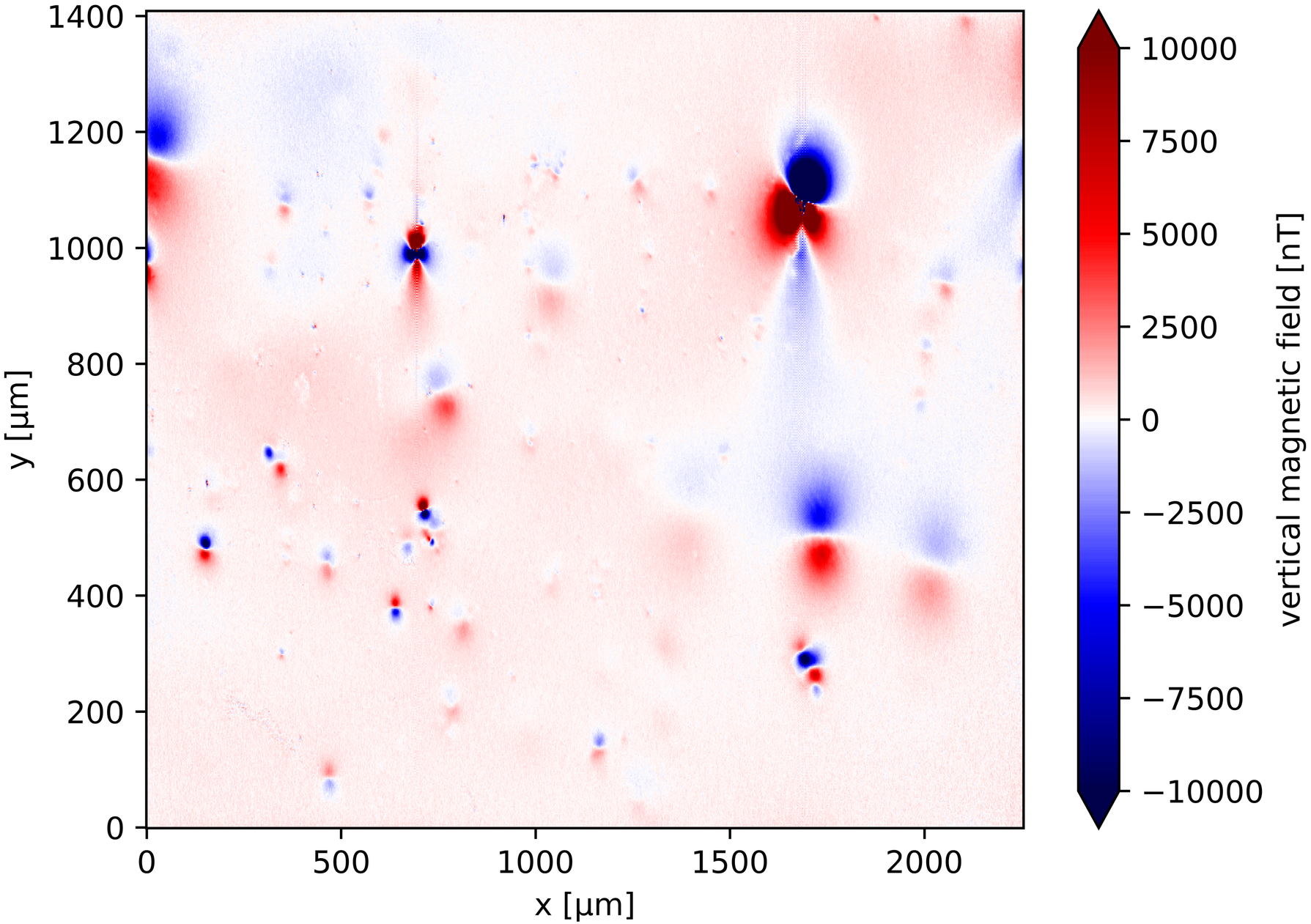
- Provide code that is **easy to use**
- Determine the **spatial positions** of **multiple** grains
- Facilitate the creation of **synthetic data**
- Propose a standard **data format**
- Serve as a **foundation** for new methods
- Leverage the potential of emerging **magnetic microscopy** studies

```
# Import standard Python libraries
import numpy as np
import skimage.exposure
import xarray as xr
import matplotlib.pyplot as plt
```

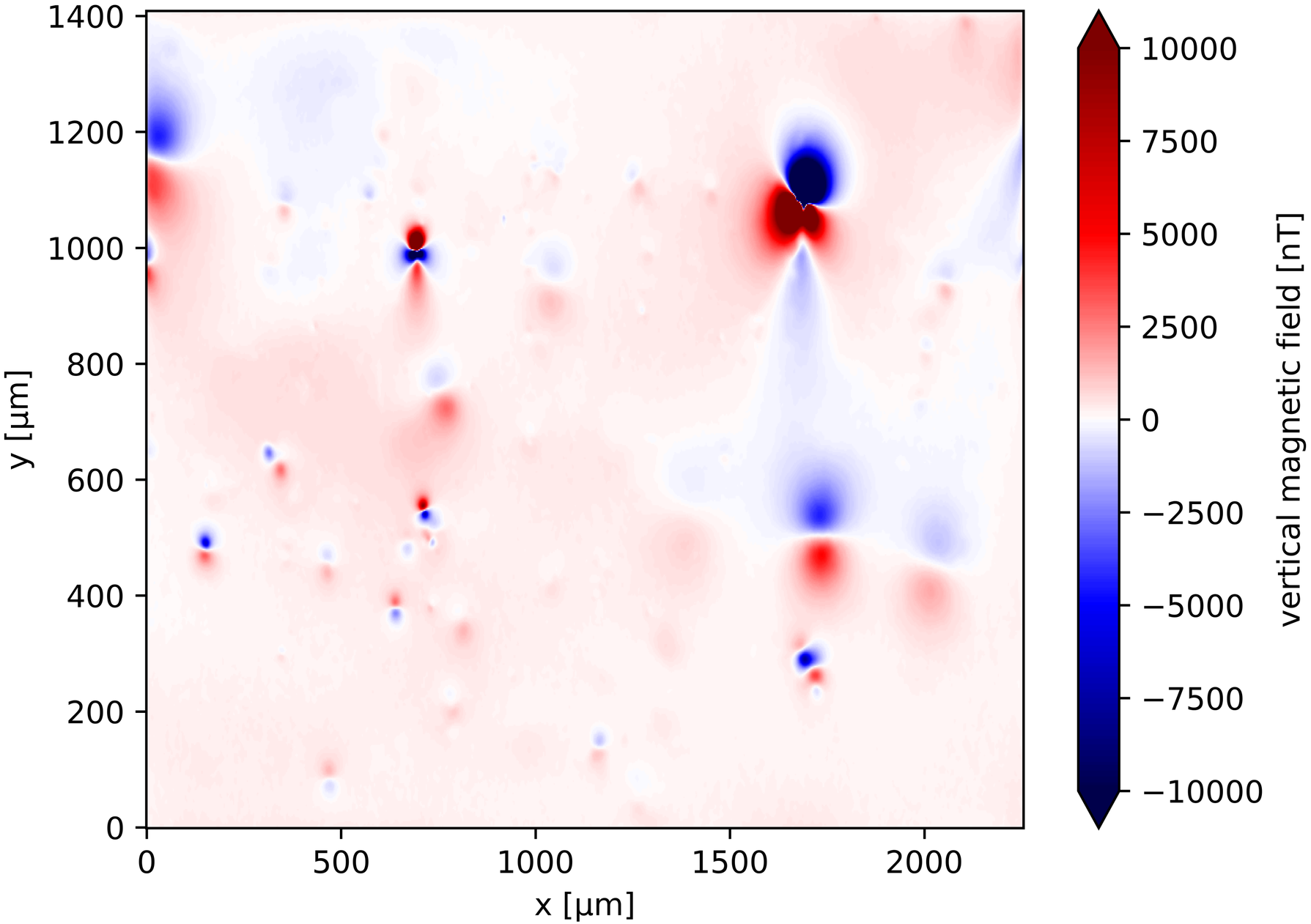
```python
# Import standard Python libraries
import numpy as np
import skimage.exposure
import xarray as xr
import matplotlib.pyplot as plt

# Import Fatiando a Terra libraries
import harmonica as hm
import ensaio
import magali
```

```python
# Import standard Python libraries
import numpy as np
import skimage.exposure
import xarray as xr
import matplotlib.pyplot as plt

# Import Fatiando a Terra libraries
import harmonica as hm
import ensaio
import magali

# Download the data
fname = ensaio.fetch_morroco_speleothem_qdm(version=1, file_format="matlab")
data = magali.read_qdm_harvard(fname)
```

```
xarray.DataArray 'bz' (y: 600, x: 960)> Size: 5MB
array([[ 352.40587477,   94.8913792 ,   41.61924299, ...,  470.18833933,
         129.20055397,   18.50120941],
       [ 525.04809649,  624.84659897,   53.45418   , ...,  450.42515609,
         240.12455308,  -73.61367693],
       [ 105.0939369 ,  638.76559489,  307.60736872, ...,  236.91326522,
         386.8498122 ,  -86.44215589],
       ...,
       [ -83.74367957,   32.98078244, -411.75073652, ...,  745.99373583,
        1036.20033954, -140.64317643],
       [ 171.17113661, -214.47801235,  159.23437984, ...,  124.58138395,
         258.54331931,  -90.3376945 ],
       [  80.60950354,  273.08367487,  118.23499313, ...,   -4.19572521,
         -53.55728012,    2.10335918]])
Coordinates:
  * x        (x) float64 8kB 0.0 2.35 4.7 7.05 ... 2.249e+03 2.251e+03 2.254e+03
  * y        (y) float64 5kB 0.0 2.35 4.7 7.05 ... 1.403e+03 1.405e+03 1.408e+03
    z        (y, x) float64 5MB 5.0 5.0 5.0 5.0 5.0 5.0 ... 5.0 5.0 5.0 5.0 5.0
Attributes:
    long_name:  vertical magnetic field
    units:      nT
```

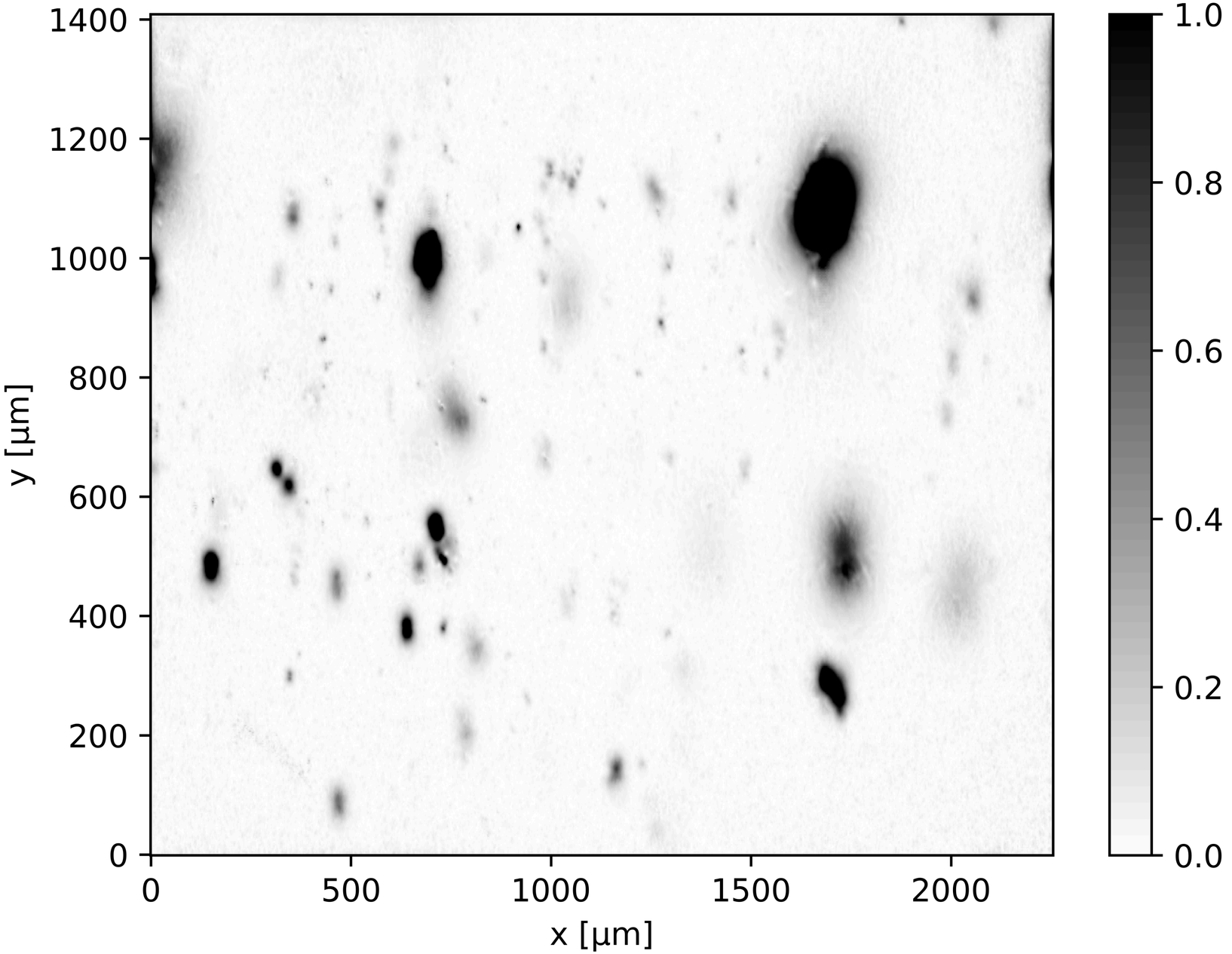

1400
1200
1000
800
600
400
200
0
y [µm]
0
500
1000
1500
2000
x [µm]
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
vertical magnetic field [nT]

```
# Upward continuation
height_difference = 5.0 # µm
data_up = (
    hm.upward_continuation(data, height_difference)
    .assign_attrs(data.attrs)
    .assign_coords(x=data.x, y=data.y)
    .assign_coords(z=data.z + height_difference)
    .rename("bz")
)
```

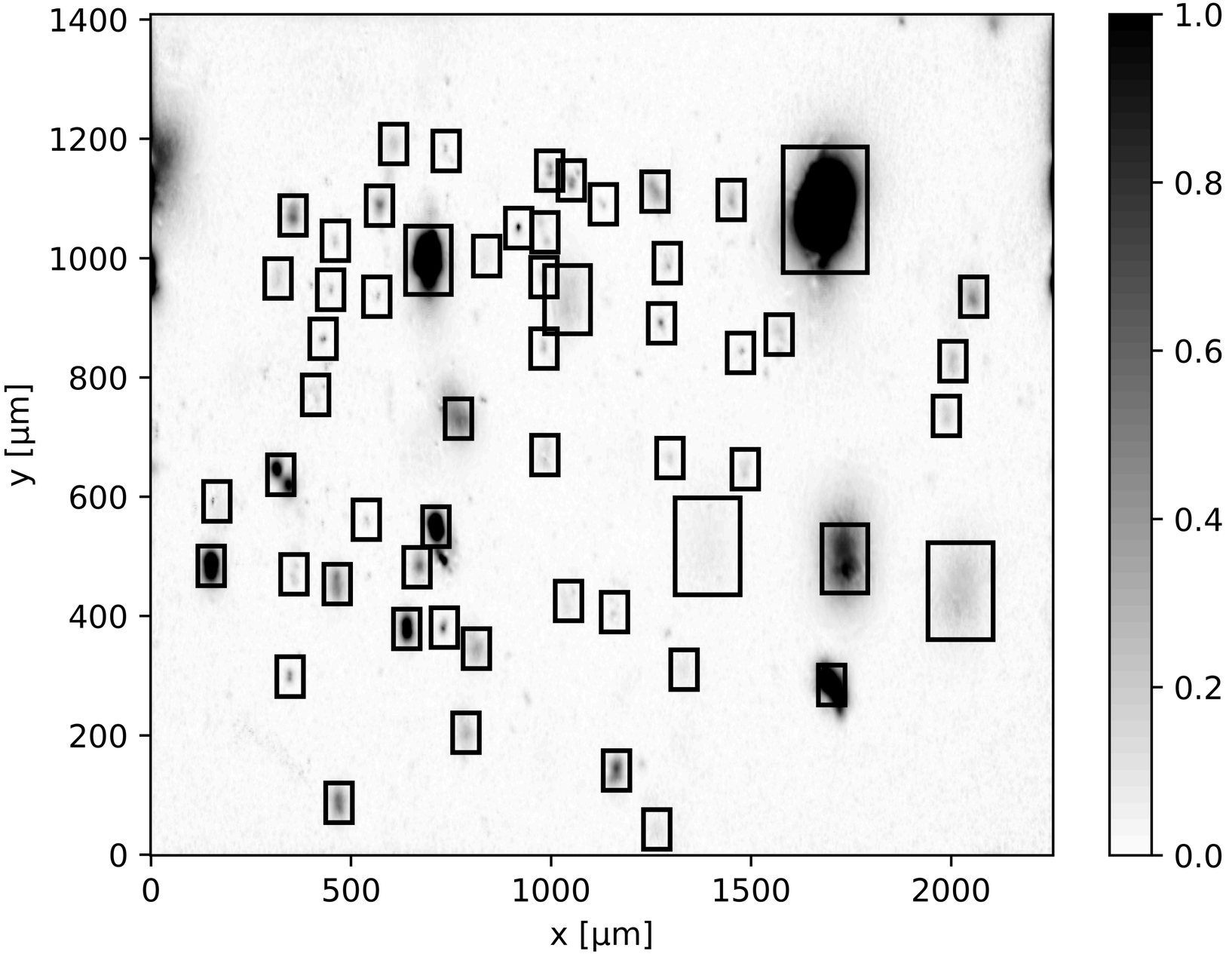

1400
1200
1000
800
600
400
200
0
y [µm]
0
500
1000
1500
2000
x [µm]
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
vertical magnetic field [nT]

```
# Calculate Total Gradient Amplitude (TGA)
dx, dy, dz, tga = magali.gradient(data_up)
data_up["dx"], data_up["dy"], data_up["dz"], data_up["tga"] = dx, dy, dz, tga
```

```
# Calculate Total Gradient Amplitude (TGA)
dx, dy, dz, tga = magali.gradient(data_up)
data_up["dx"], data_up["dy"], data_up["dz"], data_up["tga"] = dx, dy, dz, tga

# Stretch the contrast of TGA image
stretched = skimage.exposure.rescale_intensity(
    tga, in_range=tuple(np.percentile(tga, (1, 99)))
)
data_tga_stretched = xr.DataArray(stretched, coords=data_up.coords)
```


1400
1200
1000
800
600
400
200
0
y [µm]
0
500
1000
1500
2000
x [µm]
1.0
0.8
0.6
0.4
0.2
0.0

```
# Detect anomalies
bounding_boxes = magali.detect_anomalies(
    data_tga_stretched,
    size_range=[20, 150], # µm
    detection_threshold=0.02,
    border_exclusion=2,
)
```


1400
1200
1000
800
600
400
200
0
y [µm]
0
500
1000
1500
2000
x [µm]
1.0
0.8
0.6
0.4
0.2
0.0

```
# Iterative nonlinear inversion
data_updated, locations_, dipole_moments_, r2_values = magali.iterative_nonlinear_
    data_up,
    bounding_boxes,
    height_difference=height_difference,
    copy_data=True,
)
```

```
# Plot the data
```

```
# Plot the data
locations_arr = np.array(locations_)
```

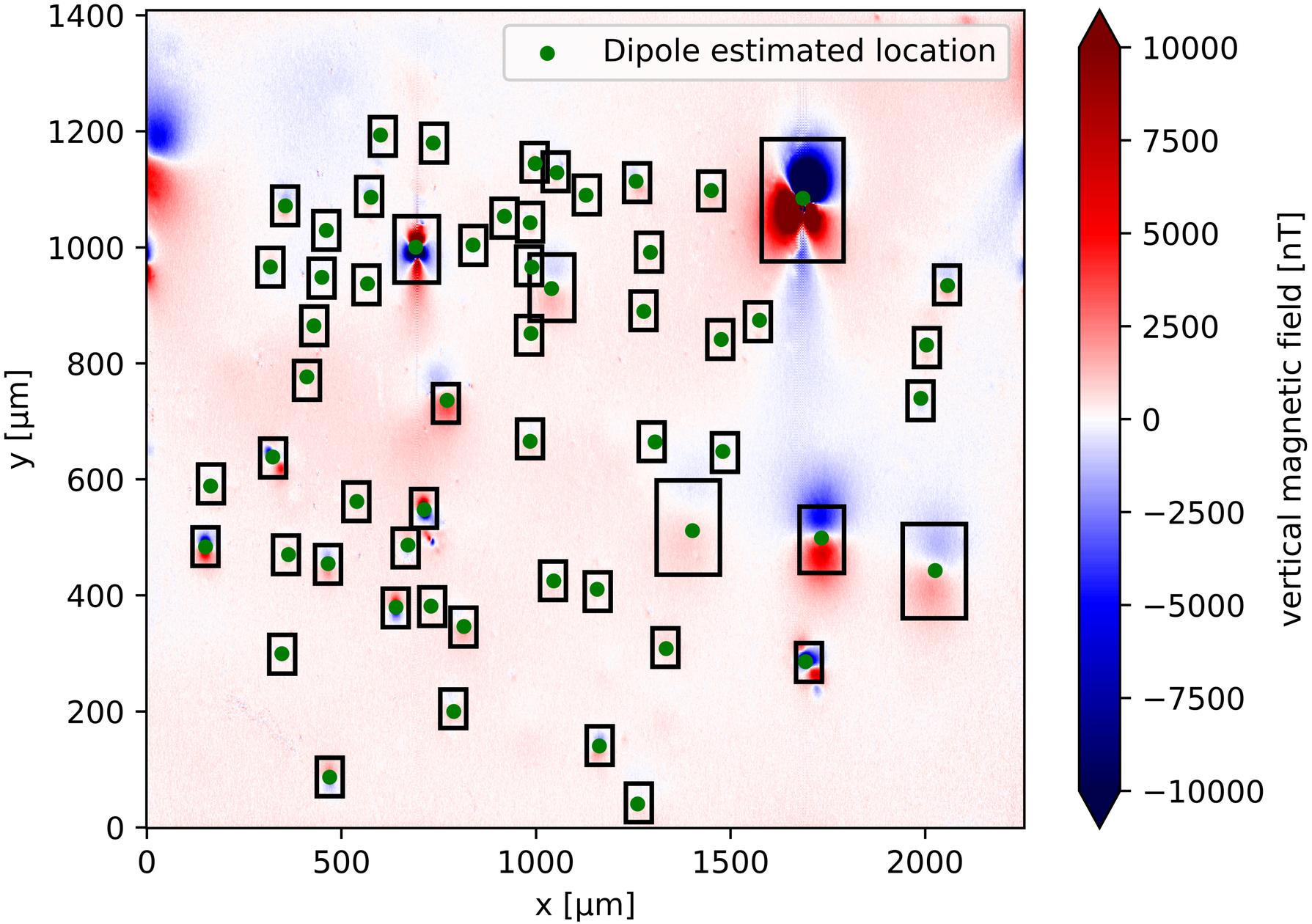
```
# Plot the data
locations_arr = np.array(locations_)
fig, ax = plt.subplots()
```

```
# Plot the data
locations_arr = np.array(locations_)
fig, ax = plt.subplots()
data.plot.pcolormesh(ax=ax, cmap="seismic", vmin=-10000, vmax=10000)
```

```
# Plot the data
locations_arr = np.array(locations_)
fig, ax = plt.subplots()
data.plot.pcolormesh(ax=ax, cmap="seismic", vmin=-10000, vmax=10000)
magali.plot_bounding_boxes(bounding_boxes, ax=ax, color="black", linewidth=1.5)
```

```
# Plot the data

locations_arr = np.array(locations_)

fig, ax = plt.subplots()

data.plot.pcolormesh(ax=ax, cmap="seismic", vmin=-10000, vmax=10000)

magali.plot_bounding_boxes(bounding_boxes, ax=ax, color="black", linewidth=1.5)

ax.scatter(
    locations_arr[:, 0],  # x
    locations_arr[:, 1],  # y
    c="green",
    marker=".",
    s=60,
    label="Dipole estimated location"
)
plt.legend()
```

```
# Plot the data

locations_arr = np.array(locations_)

fig, ax = plt.subplots()

data.plot.pcolormesh(ax=ax, cmap="seismic", vmin=-10000, vmax=10000)

magali.plot_bounding_boxes(bounding_boxes, ax=ax, color="black", linewidth=1.5)

ax.scatter(
    locations_arr[:, 0],  # x
    locations_arr[:, 1],  # y
    c="green",
    marker=".",
    s=60,
    label="Dipole estimated location"
)
plt.legend()

plt.show()
```


Dipole estimated location
1400
1200
1000
800
600
400
200
0
y [µm]
0
500
1000
1500
2000
x [µm]
10000
7500
5000
2500
0
−2500
−5000
−7500
−10000
vertical magnetic field [nT]

Conclusions

Conclusions

- Magnetic microscopy lets us investigate magnetism at the grain scale

Conclusions

- Magnetic microscopy lets us investigate magnetism at the grain scale
- **Magali** brings **automation, reproducibility**, and **speed** to these analyses

Conclusions

- Magnetic microscopy lets us investigate magnetism at the grain scale
- **Magali** brings **automation, reproducibility**, and **speed** to these analyses
- It integrates open tools,

Conclusions

- Magnetic microscopy lets us investigate magnetism at the grain scale
- **Magali** brings automation, reproducibility, and speed to these analyses
- It integrates open tools, FAIR data,

FAIR: **F**indable, **A**ccessible, **I**nteroperable and **R**eusable

Conclusions

- Magnetic microscopy lets us investigate magnetism at the grain scale
- **Magali** brings automation, reproducibility, and speed to these analyses
- It integrates open tools, FAIR data, and transparent workflows for magnetic research

Future work

Future work

- Provide, discuss, and establish **data conventions** for magnetic microscopy

Future work

- Provide, discuss, and establish **data conventions** for magnetic microscopy
- Write functions to read data from **different microscope systems**

Future work

- Provide, discuss, and establish **data conventions** for magnetic microscopy
- Write functions to read data from **different microscope systems**
- Add more datasets to **Ensaio** for testing and **community use**

Future work

- Provide, discuss, and establish **data conventions** for magnetic microscopy
- Write functions to read data from **different microscope systems**
- Add more datasets to **Ensaio** for testing and **community use**
- Release **Magali 1.0** with improved docs and structure

Acknowledgements

Obrigado! ¡Gracias! Thank you!

Contact: yagomcastro1@gmail.com

Source code for this presentation:
github.com/YagoMCastro/latinmag-magali-presentation

github.com/fatiando/magali