pyebsd: an open-source tool for processing EBSD data and determining accurate orientation relationship

Arthur Nishikawa

University of São Paulo

July 6, 2018

Outline

Introduction

EBSD analysis tools

- Electron Backscattered Diffraction
- EBSD analysis is mostly limited to commercial tools (EDAX OIM, Oxford Chanel 5)
- Although quite powerful, licenses are expensive and limited
- MTEX (mtex-toolbox.github.io): Package for EBSD analysis implemented in Matlab
- MTEX is very complete, but Matlab isn't free
- pyebsd: python + ebsd!

Introduction

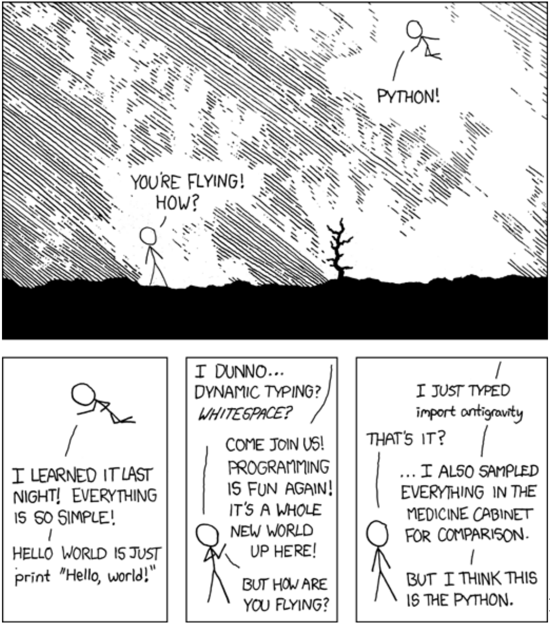
Why Python?

- Python is an interpreted high level programming language
- Fast growing user base
- Supports many programming features (imperative, object-oriented, functional programming)
- dynamic-typing (no need to declare variable types)
- Performance not as good as C, C++, Fortran, etc, but coding is very fast

```python
print("Hello world!")

# defining functions is easy!
def fun(x):
    # return x squared
    return x**2

import numpy as np
# creates array x = [0, 0.1, ..., 9.9],
x = np.arange(0, 10, .1)
# then calculates sin(x)
y = np.sin(x)
```



[1]

[1] xkcd.com/353

Outline

pyebsd

- pyebsd does not perform phases indexing
- pyebsd loads a file containing the Euler angles of the indexed phases (text file, normally .ang extension); any EBSD scanning software can generate such file
- Rotation and orientation matrices are calculated from Euler angles using ZXZ convention

- Implementation takes advantage of python modularity: `numpy` and `scipy` $\longrightarrow$ linear algebra operations; `matplotlib` $\longrightarrow$ plotting

Plot Inverse Pole Figure (IPF)

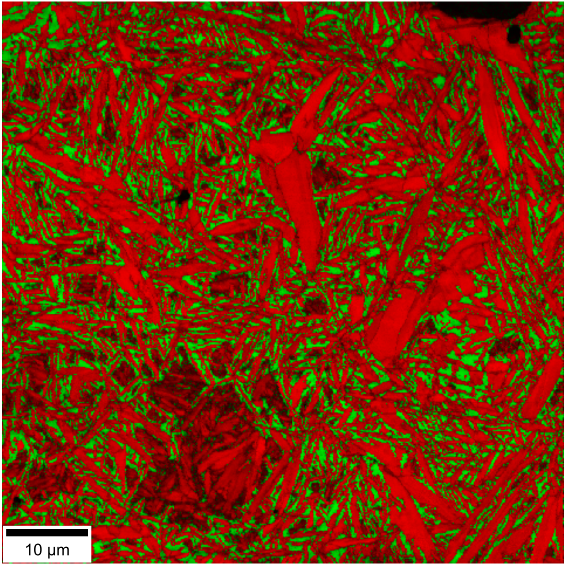
```python
import pyebsd
import matplotlib.pyplot as plt

# enables interactive mode
plt.ion()

# loading scan datafile
scan = pyebsd.load_scandata("path/to/.ang/file")
# plotting phase map; IQ as gray scale
ph = scan.plot_phase(gray=scan.IQ)
# plotting ipf
ipf = scan.plot_IPF(gray=scan.IQ)
```

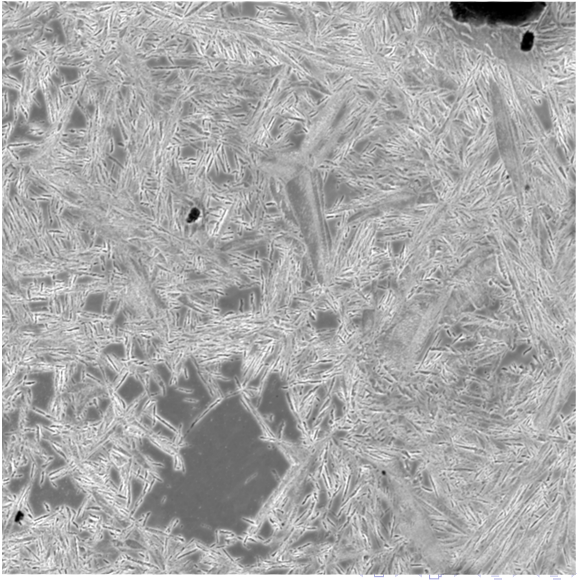
martensite + bainitic ferrite + austenite

(Q&P 170/375 °C, 0.8 C)



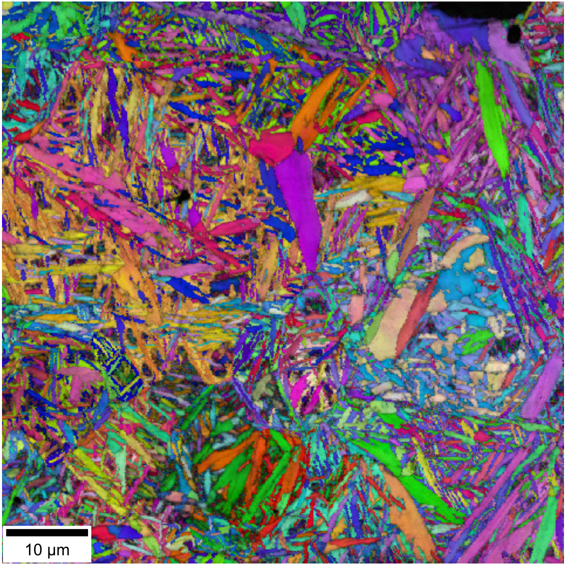
martensite + bainitic ferrite + austenite

(Q&P 170/375 °C, 0.8 C)



martensite + bainitic ferrite + austenite

(Q&P 170/375 °C, 0.8 C)

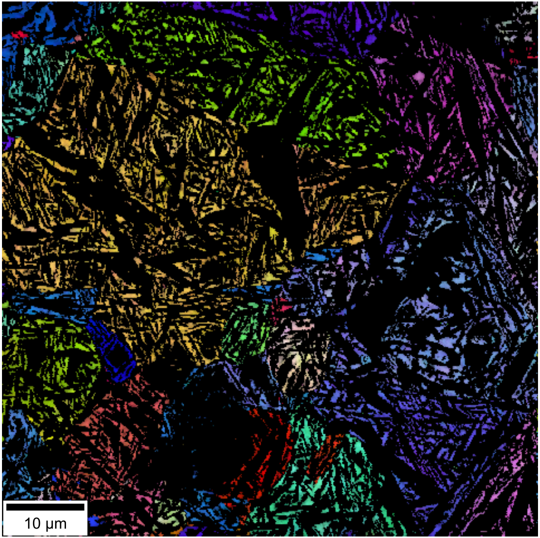


- Boolean logic in python is very powerful.
- In pyebsd it can be used to select phases, regions with specific values of confidence index, etc.

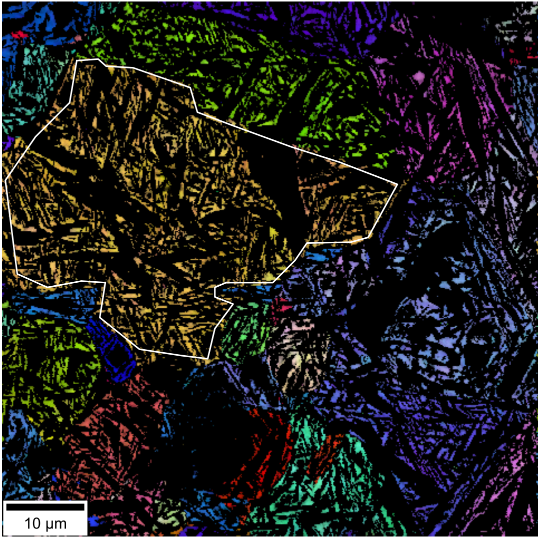
```
import numpy as np
x = np.arange(0, 10, .1)
y = np.sin(x)

selection = y > 0
x2 = x[selection]
y2 = y[selection]
```

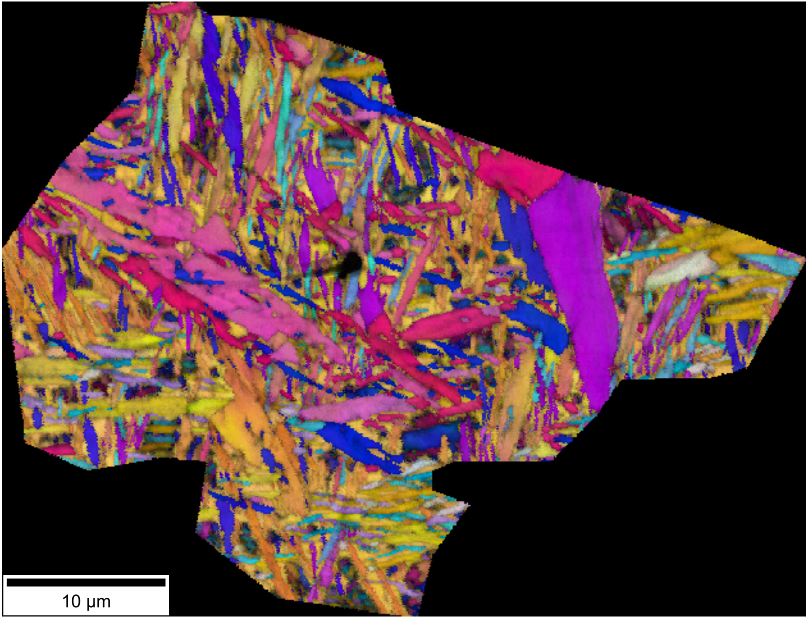
```
# austenite indexed as phase #2
ipf = scan.plot_IPF(sel=(scan.ph == 2), gray=scan.IQ)
```



```
# ipf as defined before
ipf.lasso_selector()
```



```
ipf2 = scan.plot_IPF(sel=ipf.sel, gray=scan.IQ)
```



```
# bcc phase indexed as phase #1
scan.plot_PF(sel=ipf.sel & (scan.ph == 1))
```

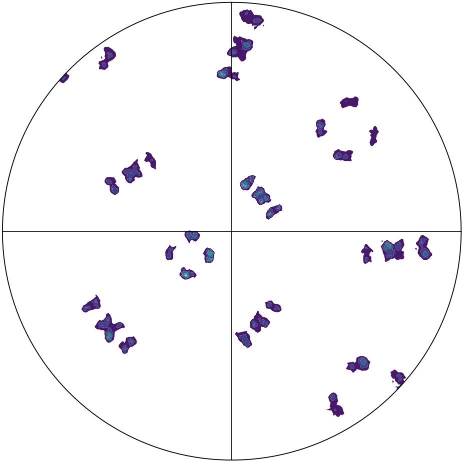


Figure: 001_{bcc} pole figure

Outline

Accurate OR

- Martensite, Widmanstätten ferrite, bainite grow with OR near to KS $\longrightarrow$ normally indexed as bcc
- Miyamoto et al., 2009[2]: method to accurately determine near KS OR using EBSD (accurate OR).
- Austenite orientation matrix M^{fcc} is necessary; it can be obtained from RA
- Miyamoto: RA is not needed $\longrightarrow$ reconstruction from expected near-KS OR

[2] Miyamoto, G., Takayama, N. & Furuhara, T. Accurate measurement of the orientation relationship of lath martensite and bainite by electron backscatter diffraction analysis. Scr. Mater. 60, 1113–1116 (2009).

Accurate OR

- For each pixel indexed as bcc (orientation matrix M^{bcc}), OR matrix $V^{fcc \rightarrow bcc}$ is calculated:

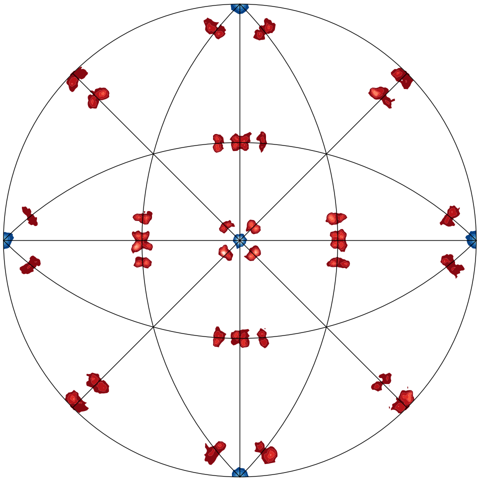
$$V^{fcc \rightarrow bcc} = M^{bcc} \cdot {M^{fcc}}^{-1}$$

- Going further, the 24 variants are determined from $V^{fcc \rightarrow bcc}$
- Finally, $V^{fcc \rightarrow bcc}$ for each variant can be compared to KS OR matrix $V_{KS}^{fcc \rightarrow bcc} \longrightarrow$ checking for deviations

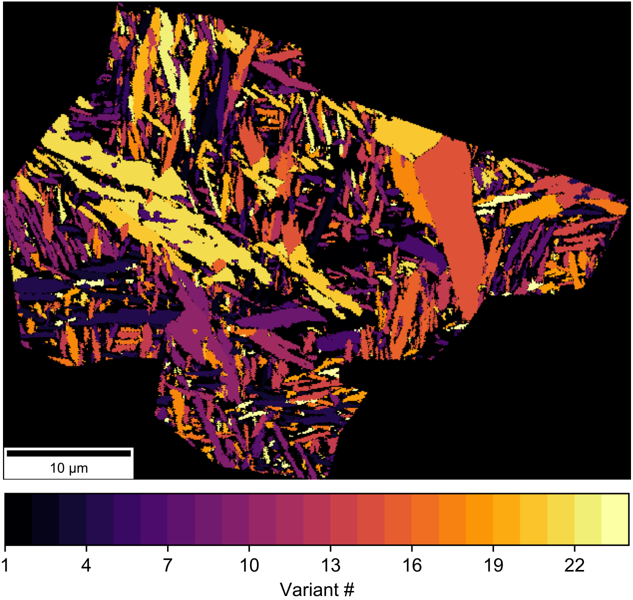
Accurate OR

- Besides QP sample (martensite + bainitic ferrite + austenite), austempered sample also analyzed (only bainite ferrite + austenite)
- High carbon (0.8 wt.%). Partitioning temp. = austempering temp. = 375 °C
- Orientation of parent austenite determined from (RA)
- Austenite reconstruction not implemented yet!

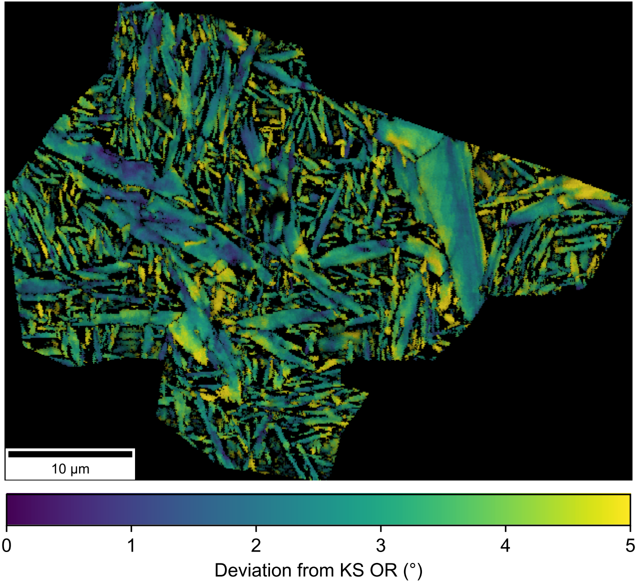
martensite + bainitic ferrite + austenite



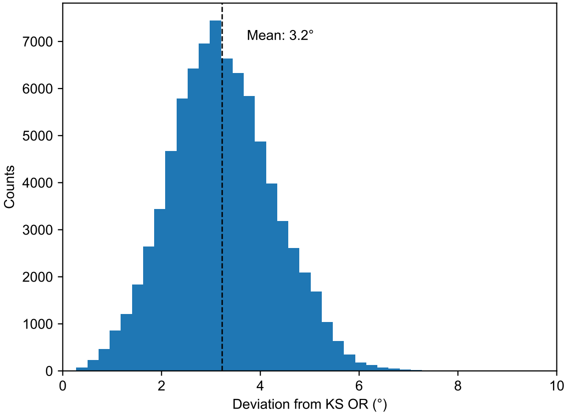
martensite + bainitic ferrite + austenite

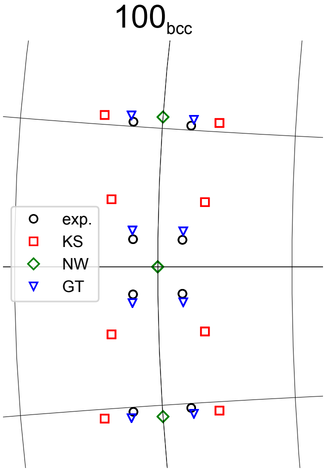


martensite + bainitic ferrite + austenite

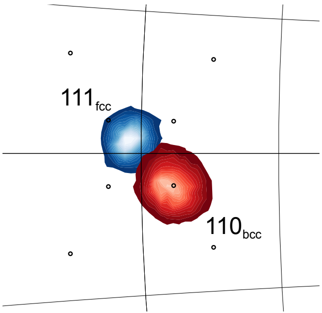


martensite + bainitic ferrite + austenite

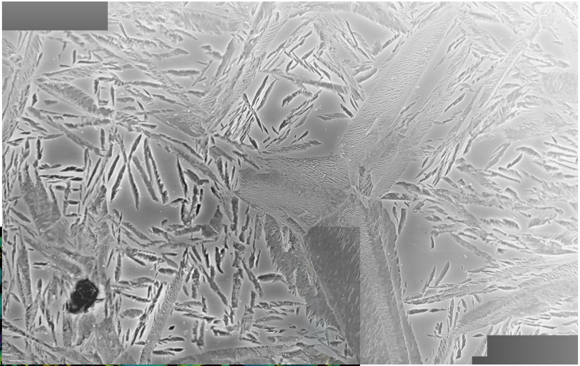


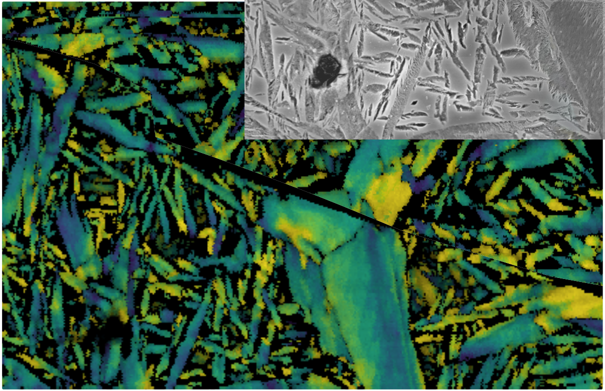


Closed packed planes (CPP)



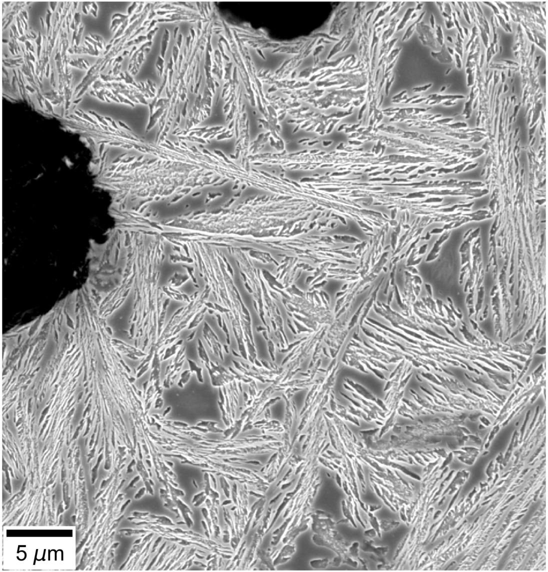
martensite + bainitic ferrite + austenite





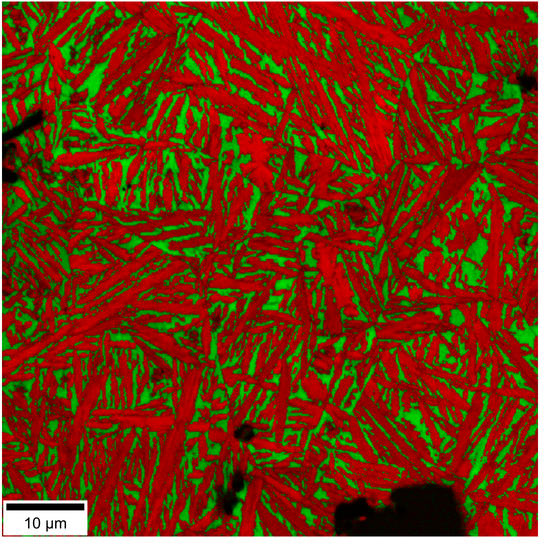
bainitic ferrite + austenite

(austempering at 375 °C, 0.8 C)

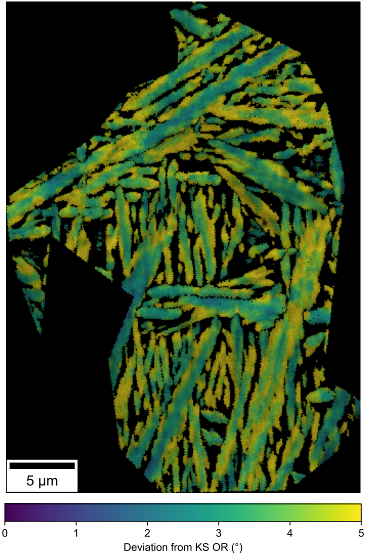


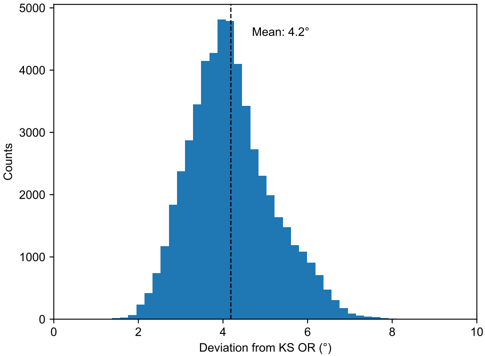
bainitic ferrite + austenite

(austempering at 375 °C, 0.8 C)

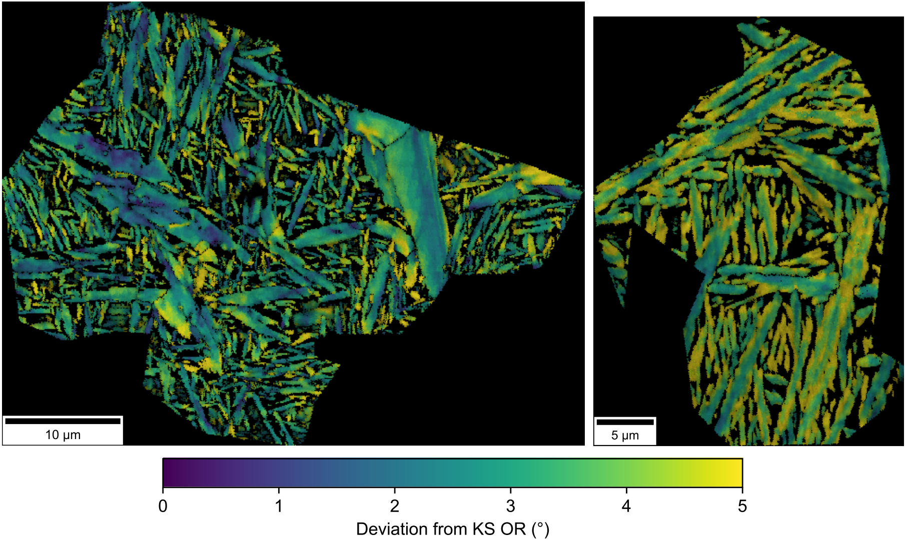


bainitic ferrite + austenite

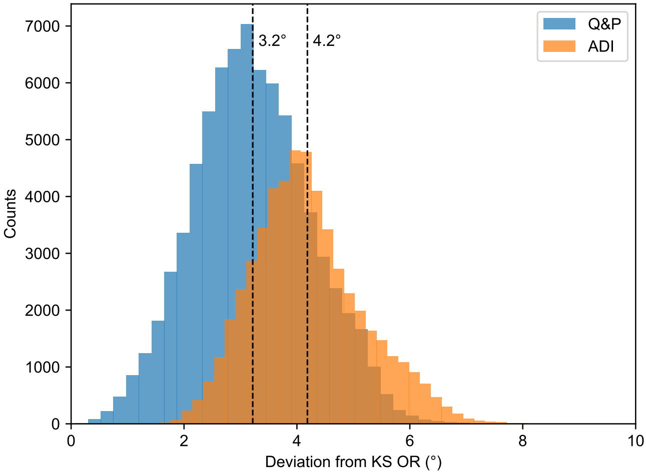




martensite + bainitic ferrite + austenite



martensite + bainitic ferrite + austenite



- Small deviations from KS expected from deformation of austenite during growth of M and BF
- Does BF inherently deviates more from KS than M? BF grows much slower than martensite, defects are accumulated at the interface
- Some kind of effect to nucleation at martensite/austenite interfaces?
- High carbon alloy: large strain associated to *fcc* $\rightarrow$ *bcc* transformation

To test these hypothesis:

- EBSD data from alloy with lower carbon
- Smaller step size
- M: accurate OR from fully martensitic sample (implement austenite reconstruction!)
- BF + M, where bainitic ferrite is formed first
- M + BF, where martensite is formed first
- BF: sample fully bainitic

Differentiate tempered martensite from bainite in low carbon alloys?

nishikawa.poli@gmail.com

github.com/arthursn/pyebsd

Thanks for your attention!