

# EMsoft



The "\$" sign used below means that what comes *after it*, in this font, should be entered in a command prompt (Windows) or terminal (macOS/Linux).

Landing page for all things EMsoft: <http://vbf.materials.cmu.edu/emsoft/>

## Why index patterns with EMsoft instead of with EDAX TSL OIM Data Collection?

When an EBSD pattern has weak Kikuchi bands, commercial software using the Hough transform, like TSL, has difficulties identifying the bands and hence cannot index the pattern. Instead of detecting the bands, EMsoft with its dictionary indexing simulates patterns and compares these to the experimental patterns. Dictionary indexing is much more robust towards noise than Hough indexing, and in most cases where there is some band information in the pattern is able to index the pattern. An example of this: <https://www.nature.com/articles/s41598-018-29315-8>

## Installation and setup on Windows

1. Download and unzip
  - a. Navigate on the landing page above to the executable package links, e.g. `EMsoft-4.2.0-Win64.zip`, or get the latest executables via the "BlueQuartz Nightly Builds" badge
  - b. Download and extract the contents to a convenient directory (folder), like `C:\Users\YOUR_USERNAME\Documents\EMsoft`
2. Tell Windows where the executables are located by adding the path to the executables, `C:\Users\YOUR_USERNAME\Documents\EMsoft\bin`, to environment variables (<https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/>)
3. A command line and text editor is needed to interact with EMsoft. On Windows, using Visual Studio Code (<https://code.visualstudio.com/>) is recommended by Prof. Marc DeGraef, the main developer of EMsoft.
4. Make sure EMsoft is setup correctly by opening a terminal in Visual Studio Code and executing the program `EMOpenCLInfo` (type `EMOpenCLInfo` and press enter)
5. Setup the configuration by going to EMsoft's wiki pages on GitHub (<https://github.com/EMsoft-org/EMsoft/wiki/Package-Configuration>) and following the instructions there. (If you are unfamiliar with the Windows Command Prompt, here is a basic tutorial: <https://www.makeuseof.com/tag/a-beginners-guide-to-the-windows-command-line/>).
6. All files produced by EMsoft are in the HDF5 format (Hierarchical Data Format version 5). To open these files, install an HDF file viewer like HDFView from the HDF Group: <https://www.hdfgroup.org/downloads/hdfview>. (They request that you register (for free) to download the viewer.)

## Perform dictionary indexing

See the highly pedagogical hands-on tutorial on how to do this: <https://link.springer.com/article/10.1007%2Fs40192-019-00137-4>.



Note that with EMsoft 5.0, the pattern centre x-coordinate (`xpc`) have to be negated (due to a change of reference frames made in this release). This means that, when converting  $x^*$  from the EDAX TSL to EMsoft's `xpc`, the equation is  $xpc = -Nx(x^* - 0.5)$ , instead of  $xpc = Nx(x^* - 0.5)$  as in the tutorial paper. The conversions from  $y^*$  and  $z^*$  to `ypc` and `L`, respectively, are still as in the tutorial paper.

To be able to index patterns stored in NORDIF's binary format, we need to address two issues:

1. EMsoft (before version 5.0, in which a NORDIF reader was added) cannot read the patterns stored in NORDIF's format, so, read the Pattern.dat file into KikuchiPy and write them to an .h5 file (HDF5), which can be read by EMsoft using the EMEBSD reader.
2. EMsoft has no static background correction, so, before writing the patterns to the .h5 file, use the `remove_static_background()` method in `kikuchipy` (<https://kikuchipy.org>).

## Get .ang file readable by EDAX TSL OIM Analysis from EMsoft dot product (HDF5) file after indexing

Use the program `EMgetANG`. (1) Create the template (text) file with the template flag `-t`, like `$ EMgetANG -t`, (2) change the file extension from .template to .nml and/or the file name if desirable, (3) update the relevant fields in a text editor, and (4) run the program by passing the new file name instead of `-t`, like `$ EMgetANG my_getang.nml`.

## Comparing simulated patterns from an indexing run to the experimental patterns

After a successful indexing run, the dynamically simulated patterns that best matched the experimental patterns can be simulated for a visual comparison by following these steps using both EMsoft and KikuchiPy:

1. Simulate the patterns
  - a. Create a text file with the best matching patterns' orientations using the EMsoft program `EMgetEulers`.
  - b. Simulate the patterns using `EMEBS`D with the file output from `EMgetEulers` as input. This will create an HDF5 file (.h5) with the simulated patterns with the file size equal to the file size of experimental pattern file if patterns were simulated to 8-bit depth (0-255).
2. Visualise the simulated and experimental patterns. The patterns can be read into KikuchiPy by for now (a reader for the simulated patterns will be created so the file can be loaded with the `kikuchipy.load()` function) using the following, followed by plotting the two data sets together, and if desired saving patterns of interest to file, using:

```
import kikuchipy as kp

import h5py

import hyperspy.api as hs

import matplotlib.pyplot as plt

s = kp.load('/path/to/experimental/patterns/Pattern.dat')

with h5py.File('/path/to/emebds/file/simulated_patterns.h5', mode='r') as f:
```

```
simulated_patterns = f['EMData/EBSD/EBSDPatterns'][(  
s_sim = kp.signals.EBSD(simulated_patterns.reshape(s.data.shape))  
hs.plot.plot_signals([s, s_sim])  
x, y = (100, 100) # Navigation indices for pattern of interest (the red letters when plotting)  
plt.imsave(fname='/path/to/place/single/patterns/into/pattern_exp.png', arr=s.inav[x, y].data)  
plt.imsave(fname='/path/to/place/single/patterns/into/pattern_sim.png', arr=s_sim.inav[x, y].data)
```

## Resources

- 30-minutes introductory walkthrough of dynamic EBSD pattern simulation and examples of its usage by Stuart Wright from EDAX TSL: <https://www.youtube.com/watch?v=Jri181evpiA>