

## Grain Tracking Algorithm V2.7.3

**Instructions:** We have included a simple graphical user interface which helps illustrate usage of the Grain Tracking Algorithm (GTA) for two different example simulations. This interface is somewhat specific to our interests regarding nanocrystalline metals, but many of the functions contained within the GTA can be modified for or applied to other materials applications. There are many additional scripts which are not included in the GUI as well but which have also been included in this package. For larger analyses we suggest implementing this script on a high performance cluster to decrease processing time. All of the codes in this package are for your perusal. If any other questions arise, please do not hesitate to email Jason Panzarino: [jpanzari@uci.edu](mailto:jpanzari@uci.edu) or Dr. Timothy Rupert: [trupert@uci.edu](mailto:trupert@uci.edu). More information about this tool can be found in the following article:

Panzarino J F and Rupert T J 2014 Tracking Microstructure of Crystalline Materials: A Post-Processing Algorithm for Atomistic Simulations *JOM* **65** 417-428

### Main Scripts

The following are the main driving functions used by the GTA and their application:

GRAINIDENT1.m	Identification of crystallites
GRAINHISTORYEXECUTABLE.m	Mapping of MD time steps
periodic_boundaries.m	Builds periodic boundaries
OrientBCC	Calculate BCC orientation axis
OrientFCC	Calculate FCC orientation axis
Misorientation9f.m	Calculates disorientation between axis

### Proper input format

In order to use the graphical user interface (GUI) the input file from LAMMPS **must** be in same format as the example files given. To obtain this output from LAMMPS the following columns were calculated and output during the molecular dynamics simulation. X, Y, & Z coordinates are first obtained. Common neighbor analysis (CNA) and Centro-Symmetry Parameter (CSP) are then listed. The GTA only requires CSP to identify grains, however we output CNA from the MD simulation as well since a simple modification to the GRAINIDENT1.m script could be done to use this metric instead. **The easiest way to obtain this input format is to open your LAMMPS output in OVITO and then export the data as a LAMMPS Dump File with the following atom attributes in the proper order:**

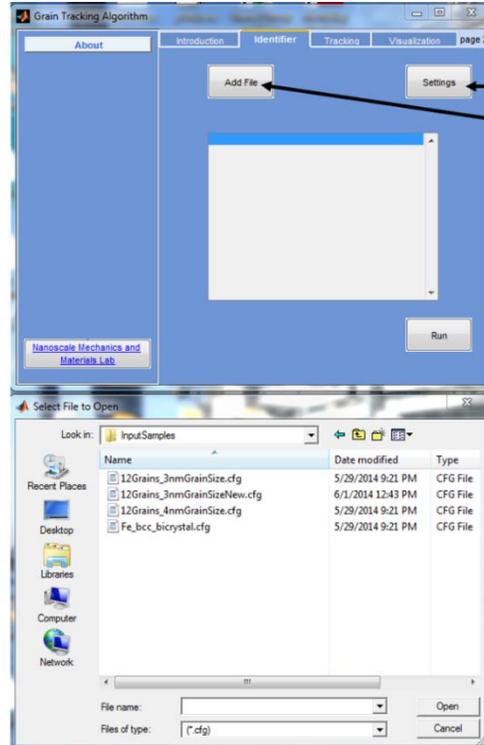
Position.X	Position.Y	Position.Z	CNA	CSP
------------	------------	------------	-----	-----

## Running the GUI

### Grain Identifier

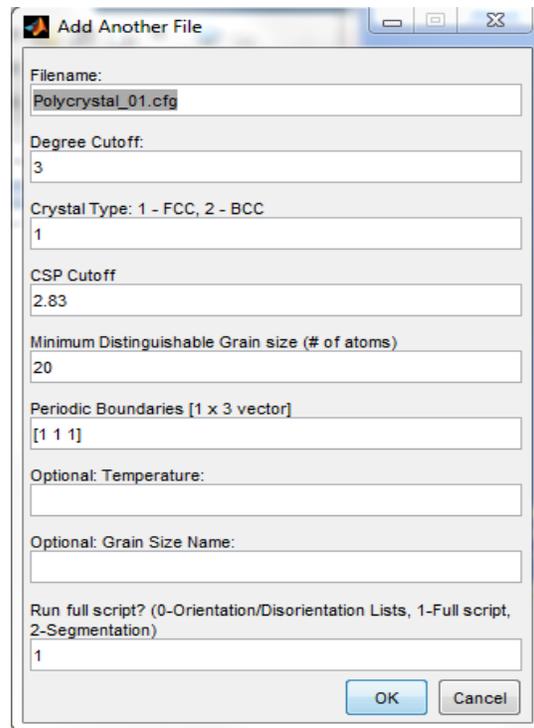
1. After opening MATLAB, run the function `GRAIN_TRACKING_ALGORITHM.m`

2. The "Identifier" tab will need to be utilized first. Use the add file button to select the files you wish to run. NOTE: All files will need to be in the "InputSamples" folder.



3. After selecting an input sample you will need to carefully follow the instructions in the dialogue box. To add another file, repeat this process until all files have been added. Once this is complete, the **settings** button can be used to increase the program speed by allowing the use of multiple cores. Be sure to check how many cores your computer has and how many you are willing to use. The image to the right shows some example inputs for a polycrystalline FCC sample file.

**Note1:** The periodic boundary input is as follows: Use vector format to choose which directions to impose periodic boundaries. 1 – designates true, 0 – False.



Centro-symmetry cut-off for fcc Ni

Periodic in all directions

Optional file - naming designators

For instance, [1 1 0] would mean that periodic boundaries were used only in the  $\pm X$  and  $\pm Y$  faces of the simulation cell during the molecular dynamics simulation.

**Note2:** The final input box allows the user to run either the full grain segmentation portion of the script at once or to run in two parts. Since the orientation & disorientation lists can be built faster using multiple processors, one has the option to build these lists first and then run the grain identification portion of the script later. This may be helpful for running large files more efficiently. If the full script is ran at once, only part I is able to utilize parallel computing and part II will be serial (regardless of how many cores are allocated). Either way the script is run, part II can only be ran in serial.

Once all inputs are defined, you may click Run. Leave any dialogue boxes untouched and you may monitor the command window as well during the code execution. Completed files will be saved in Outputfiles/GrainSegmentedOvitoCFGFiles. For convenience, the files are output in a compatible .cfg format for easy use with the open source visualization tool OVITO. The output files should be checked for accuracy since the degree cutoff and CSP cutoff could be varied for different results.

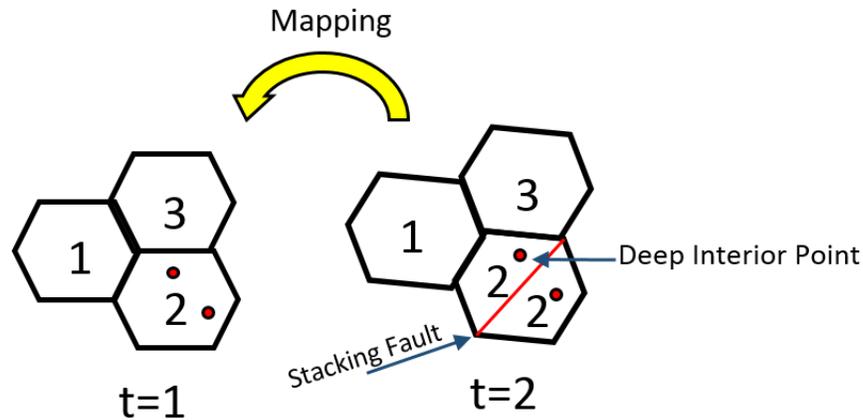
## Grain Tracking

Now that all grains have been specified with orientation information calculated, grain evolution can be tracked between multiple files. The degree of accuracy depends on the amount of evolution occurring during time steps so make sure an adequate number of files are used to ensure proper tracking. As with any experimental tool, all final data should be verified for accuracy.

1. Run Grain Tracking Executable on the 3<sup>rd</sup> tab of the interface and follow the prompts. The files will be mapped in alphabetical order with the names automatically generated from the folder: Outputfiles/GrainSegmentedOvitoCFGFiles. The files will be mapped in the same alphabetical order that the input files were ordered.

2. In order to map crystallites between time steps we need to find corresponding mapping points within each grain. This leaves us with two options: We can calculate the center of mass of grains, or we can use atomic positions that lie deep within each grain. The GTA uses deep grain interior atoms, but also calculates the grain mass centers as well. Modification of the GRAINHISTORYEXECUTABLE.m script would allow the user to change which method is used. Currently the script uses deep interior points since it is less susceptible to errors due to significant grain sliding or grain coalescence.

That being said, the GUI will prompt you for how many nearest neighbors to use for finding corresponding mapping points for crystallites. 50-100 seems to work well for nanocrystalline grains, but a smaller number can have a significant impact on computational time. If several instances of incorrect mapping occurs, the user can rerun the Grain Tracking Executable with a larger input value.



Example of how the mapping is conducted. Grain numbers ultimately originate from the first MD time step.

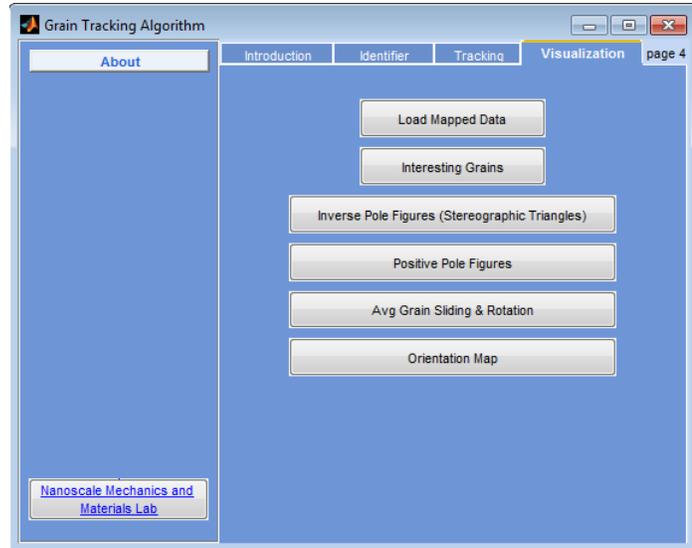
3. An optional file description designator is also allowed for input.
4. Once again, specify which directions periodic boundary conditions were applied during MD.
5. After Running the Tracking algorithm, the mapped files will be stored in the newly created folder: "MappedCFGFiles" and can be easily analyzed using OVITO. This should also be done to verify correct mapping of crystallites since significant microstructural evolution can be difficult to track. If mapping appears to be incorrect, the number of grain interior atoms used for mapping should be increased and the tracking executable should be rerun. Warnings of large variation in the measured orientation across each grain will be output in the MATLAB Command Window during GTA execution. The appearance of these warnings can be indicators of errors or even interesting grain evolution and should be visually verified using OVITO.

## Visualization

Once tracking has been completed, the visualization tab is used to illustrate some of the data which was extracted using the GTA.

1. Load the mapped data by clicking the first button. This button must always be used before selecting any of the visualization buttons.

2. Interesting grains are those grains where either significant rotation ( $> 4$  degrees of misorientation change between time steps) or discrepancies in tracking occurred. This usually corresponds to grain coalescence or grains that may have shrank and disappeared during the simulation. The grains listed may not be the only interesting grains of the sample. In addition, they may indicate erroneous data, and therefore the option to remove them from all data is also given.



3. The remaining visualization tools will develop inverse pole figures, z-direction positive pole figures, qualitative plots regarding average sliding and rotation of grains, and orientation maps. Carefully read the prompts when deciding how you want the data to be displayed.

**Note:** Whenever multiple time steps are plotted on a single plot, changes in color from dark to light will be used to visualize evolution between time steps.

### Final Output Formats

The following table describes the column numbers for both output .cfg files as well as the MATLAB variable, A, which are created by the GTA. Intermediate variables are saved following execution of both the Grain Identification Executable as well as the Grain Tracking Executable. Columns labeled N/A are can be disregarded and may be utilized in a future release.

**Final Grain Segmentation Output .cfg Format / Matlab A-Matrix**

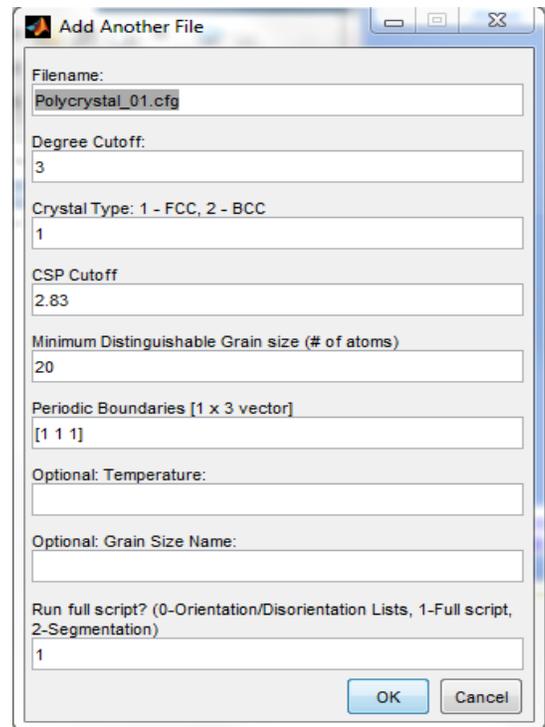
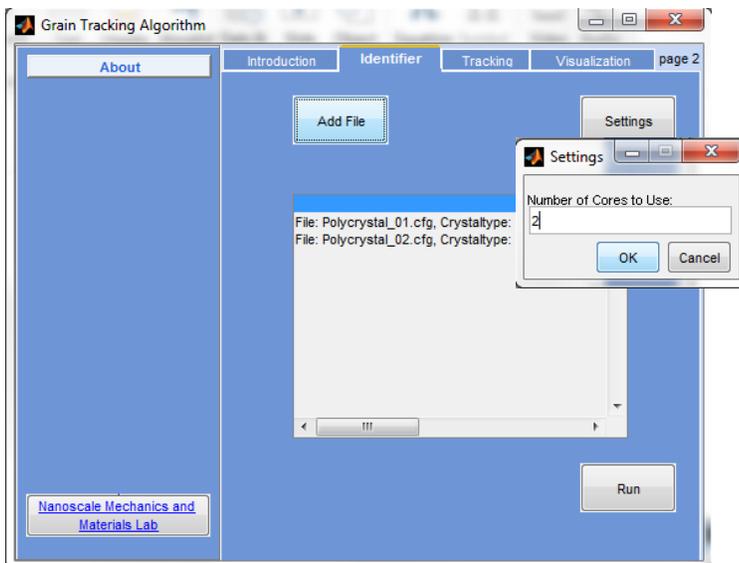
1. X
2. Y
3. Z
4. CNA
5. CSP
6. Crystalline Identifier
7. Atom #
8. Grain #
9. Check Designator
10. Edge Atom Designator
11. N/A – Will Be used in Future Release
12. N/A
13. N/A
14. N/A
15. Orientation Axis 1 (component)
16. Orientation Axis 1 (component)
17. Orientation Axis 1 (component)
18. Orientation Axis 2 (component)
19. Orientation Axis 2 (component)
20. Orientation Axis 2 (component)
21. Orientation Axis 3 (component)
22. Orientation Axis 3 (component)
23. Orientation Axis 3 (component)

## Examples

All examples already include the associated output files. A rerun of the GUI will simply overwrite these pre-existing outputs.

Example 1: Polycrystalline Ni with an average grain size of 3nm.

1. Load both polycrystalline files with the following inputs.
2. After loading both files and their required inputs, use the settings dialogue to change the number of cores to the amount of available processors you are able to commit to the analysis. Then you may click Run. Depending on the file size the script can take a significant amount of time to run.



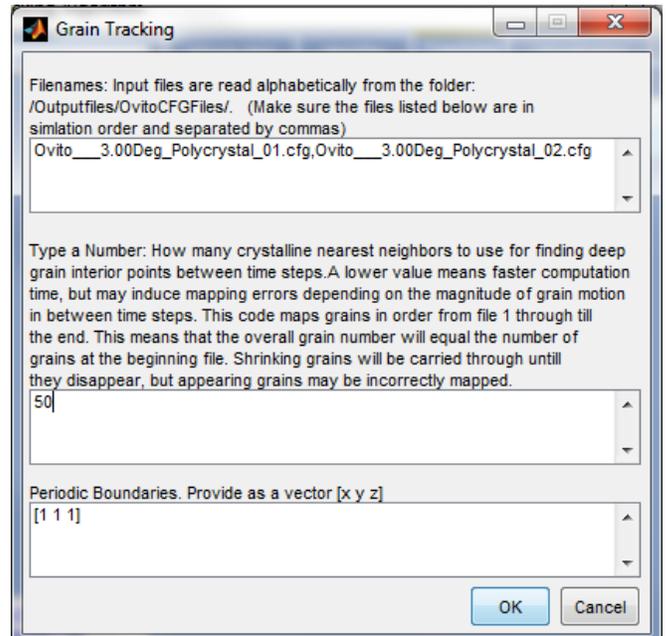
- When this is finished, click Run Grain Tracking Executable and follow input instructions.



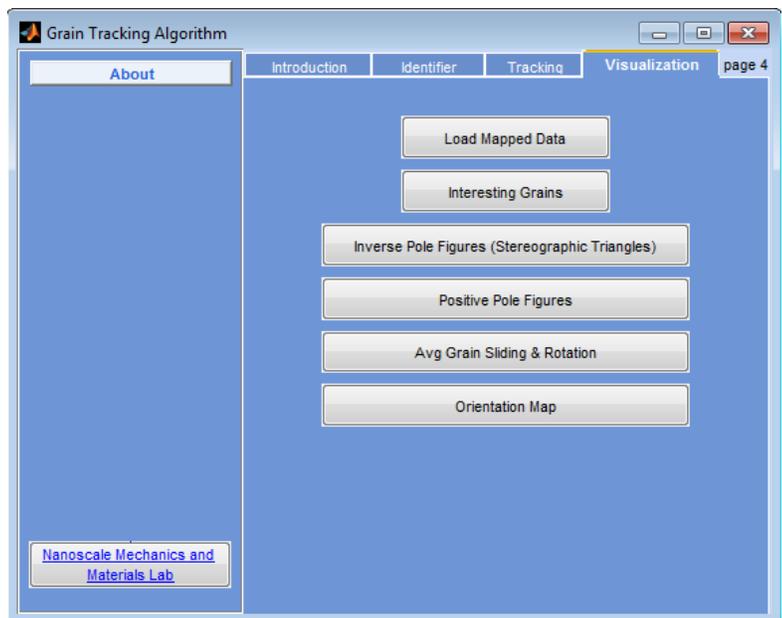
- The filenames (ordered alphabetically) will be populated from the folder:  
/Outputfiles/GrainSegmentedOvitoCFGFiles

Check to ensure proper order

- Because of the extremely small grain size 50 crystalline nearest neighbors is sufficient for successful grain mapping.
- Once grain tracking has completed, the data will be automatically stored for use with the data visualization tools.

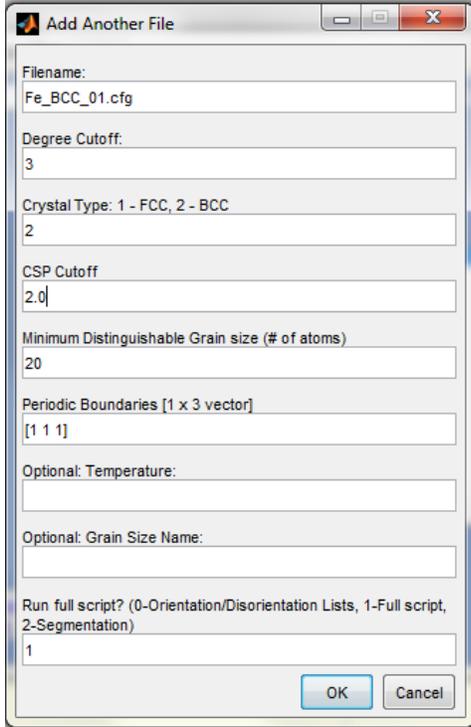


- Now you may run the tools listed on the Visualization tab. After loading the mapped data, Interesting Grains can be utilized to get an initial idea of rotational evolution of crystallites as well as the accuracy of the mapping.
- Follow the on screen prompts to output all of these plots in various different formats.



Example 2:  $\Sigma 5$  (310) Fe BCC Bicrystal

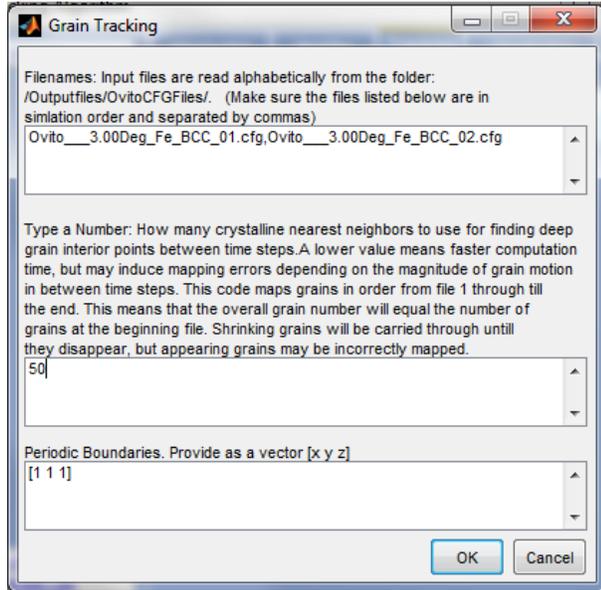
The instructions for the BCC Bicrystal are identical to Example 1 with the following changes in user input due to different periodic boundary directions and simulation conditions:



The 'Add Another File' dialog box contains the following fields and values:

- Filename: Fe\_BCC\_01.cfg
- Degree Cutoff: 3
- Crystal Type: 1 - FCC, 2 - BCC: 2
- CSP Cutoff: 2.0
- Minimum Distinguishable Grain size (# of atoms): 20
- Periodic Boundaries [1 x 3 vector]: [1 1 1]
- Optional: Temperature: (empty)
- Optional: Grain Size Name: (empty)
- Run full script? (0-Orientation/Disorientation Lists, 1-Full script, 2-Segmentation): 1

Buttons: OK, Cancel



The 'Grain Tracking' dialog box contains the following fields and values:

- Files list: Ovito\_\_3.00Deg\_Fe\_BCC\_01.cfg,Ovito\_\_3.00Deg\_Fe\_BCC\_02.cfg
- Type a Number: 50
- Periodic Boundaries. Provide as a vector [x y z]: [1 1 1]

Buttons: OK, Cancel

## Gran Boundary Network Analysis (Beta)

Keep in mind this portion of the script is still undergoing testing. Running the GB Network Analysis will output CFG files which provide grain boundary character data, including grain boundary normal orientation. If the sample is polycrystalline, triple junction types will also be analyzed. It is important to note that the GB Network is analyzed using the original segmented grain numbers (does not include mapping of grain numbers between time steps). The Topology CFG Files add triple junction types based on the number of attached special boundaries as well as Mapped Grain number data (for reference). The data is formatted into OVITO CFG files as follows with the A-matrix columns representing the following data:

<b>GB Mapped Ovito Output File Format</b>	
<b>CFG File Column Names</b>	<b>Explanation</b>
X	X-coordinate
Y	Y-coordinate
Z	Z-coordinate
cna	CNA
centro	Centrosymmetry
ident	Crystallinity yes/no
anum	Atom number
gnum	Grain number
check	Checked during segmentation?
edge	Grain edge?
tj	Number of grain neighbors
secnum	GB section number
a1	Orientation Axis1 component
a2	Orientation Axis1 component
a3	Orientation Axis1 component
b1	Orientation Axis2 component
b2	Orientation Axis2 component
b3	Orientation Axis2 component
c1	Orientation Axis3 component
c2	Orientation Axis3 component
c3	Orientation Axis3 component
gbnormal1	GB Plane Normal component (sample coordinates)
gbnormal2	GB Plane Normal component (sample coordinates)
gbnormal3	GB Plane Normal component (sample coordinates)
ng1a	GB Plane Normal component (grain1 coordinates)
ng1b	GB Plane Normal component (grain1 coordinates)
ng1c	GB Plane Normal component (grain1 coordinates)
ng2a	GB Plane Normal component (grain2 coordinates)
ng2b	GB Plane Normal component (grain2 coordinates)
ng2c	GB Plane Normal component (grain2 coordinates)
disaxisn1	Disorientation Axis component

disaxisn2	Disorientation Axis component
disaxisn3	Disorientation Axis component
disangle	Disorientation Angle
mismisorientation	Mismisorientation Angle
sigma	Sigma Value $\leq 35b$ <b>NOTE:</b> Sigma values with “a” and “b” designators are denoted with numbers. Ex: 25a = 251 and 25b=252. Please take note when calculating sigma fractions, etc.
neighbg1	Neighboring Grain # 1
neighbg2	Neighboring Grain # 2
neighbg3	Neighboring Grain # 3
neighbg4	Neighboring Grain # 4
neighbg5	Neighboring Grain # 5
niehbg6	Neighboring Grain # 6

<b>GB Topology CFG Ovito Output File Format</b>	
<b>CFG File Column Names</b>	<b>Explanation</b>
X	X-coordinate
Y	Y-coordinate
Z	Z-coordinate
cna	CNA
centro	Centrosymmetry
ident	Crystallinity yes-no
anum	Atom number
gnum	Grain number
check	Checked during segmentation?
edge	Grain edge?
tj	Number of grain neighbors
secnum	GB section number
a1	Orientation Axis1 component
a2	Orientation Axis1 component
a3	Orientation Axis1 component
b1	Orientation Axis2 component
b2	Orientation Axis2 component
b3	Orientation Axis2 component
c1	Orientation Axis3 component
c2	Orientation Axis3 component
c3	Orientation Axis3 component
gbnormal1	GB Plane Normal component (sample coordinates)
gbnormal2	GB Plane Normal component (sample coordinates)
gbnormal3	GB Plane Normal component (sample coordinates)
ng1a	GB Plane Normal component (grain1 coordinates)
ng1b	GB Plane Normal component (grain1 coordinates)

ng1c	GB Plane Normal component (grain1 coordinates)
ng2a	GB Plane Normal component (grain2 coordinates)
ng2b	GB Plane Normal component (grain2 coordinates)
ng2c	GB Plane Normal component (grain2 coordinates)
disaxisn1	Disorientation Axis component
disaxisn2	Disorientation Axis component
disaxisn3	Disorientation Axis component
disangle	Disorientation Angle
mismisorientation	Mismisorientation
sigma	Sigma Value
neighbg1	Neighboring Grain # 1
neighbg2	Neighboring Grain # 2
neighbg3	Neighboring Grain # 3
neighbg4	Neighboring Grain # 4
neighbg5	Neighboring Grain # 5
neighbg6	Neighboring Grain # 6
jnum	Triple Junction #
g1	Grain Neighbor1
g2	Grain Neighbor2
g3	Grain Neighbor3
s1	Sigma type of attached boundary 1
s2	Sigma type of attached boundary 2
s3	Sigma type of attached boundary 3
tjtype	Number of attached special boundaries
gnumhist	Time/simulation Mapped Grain Numbers (for reference only)