Los Alamos NATIONAL LABORATORY

Graph Based Twin Analysis

Daniel J. Savage[a], Rodney J. McCabe[a], Marko Knezevic[b]

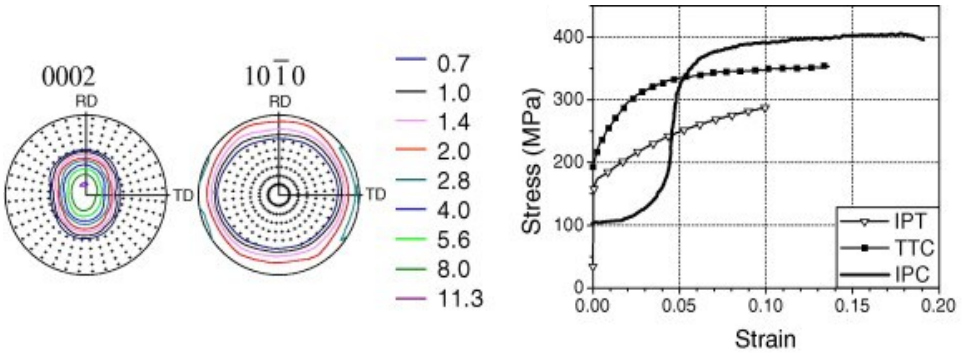
[a] Materials Science and Technology Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[b] Department of Mechanical Engineering, University of New Hampshire, Durham, NH 03824, USA

Introduction

What is deformation twinning?

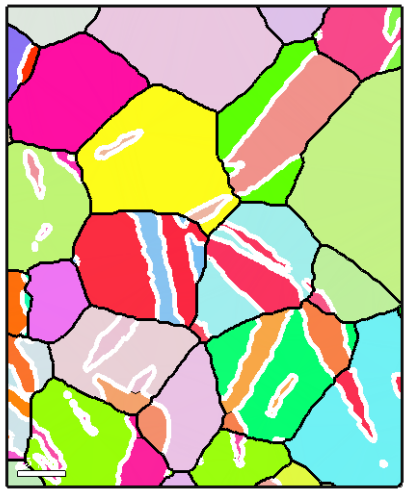
- A crystallographic process that accommodates shear through dislocation glide
- Depends on crystal symmetry, chemistry, stress state, work hardening, etc…

Important for mechanical properties

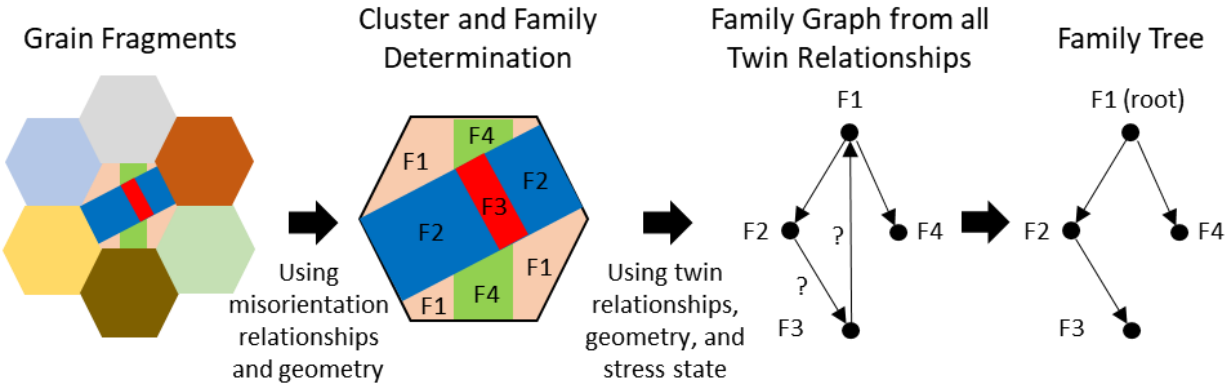


AZ31 example from Proust et al, IJP (2003)

Matlab: mtexdata twins



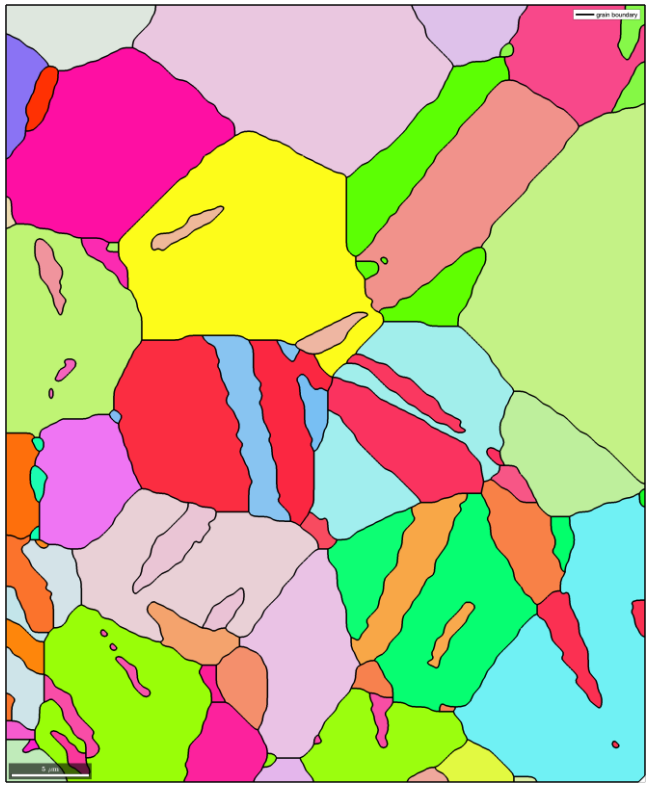
Analysis of twinning in orientation map dataset



Savage et al, MC (2020)

Introduction

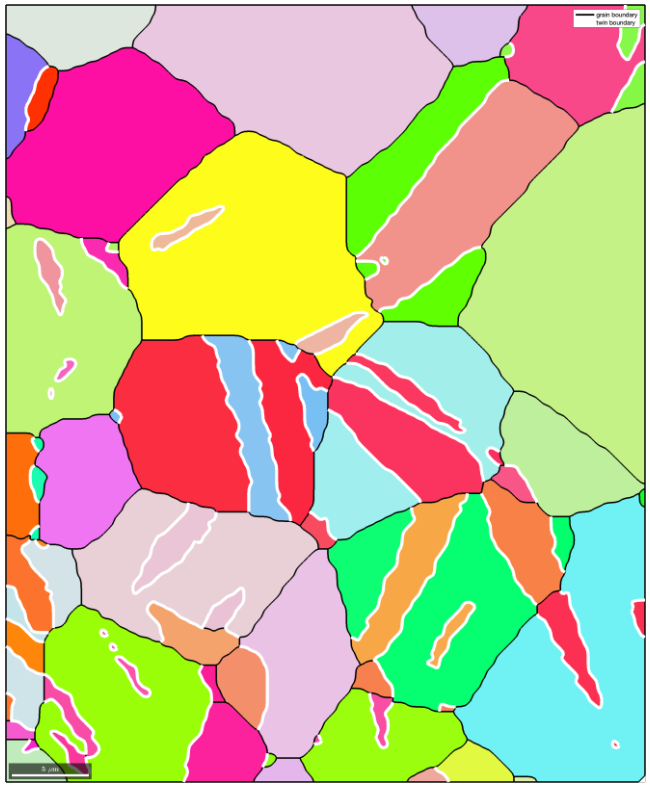
Fragment map



```
% load some example data
mtexdata twins silent

% segment grains
[grains,ebsd.grainId,ebsd.mis2mean] =
calcGrains(ebsd('indexed'),...
    'angle',5*degree);

% remove two pixel grains
ebsd(grains(grains.grainSize<=2)) = [];
[grains,ebsd.grainId,ebsd.mis2mean] =
calcGrains(ebsd('indexed'),...
    'angle',5*degree,'removeQuadruplePoints');

%smooth grains
grains = grains.smooth(5);

%plot map
figure; plot(grains,grains.meanOrientation);hold on
plot(grains.boundary,'linecolor','k','linewidth',2.5,'linestyle','-',...
  'displayName','grain boundary')
```

Introduction

Fragment + twin boundary map

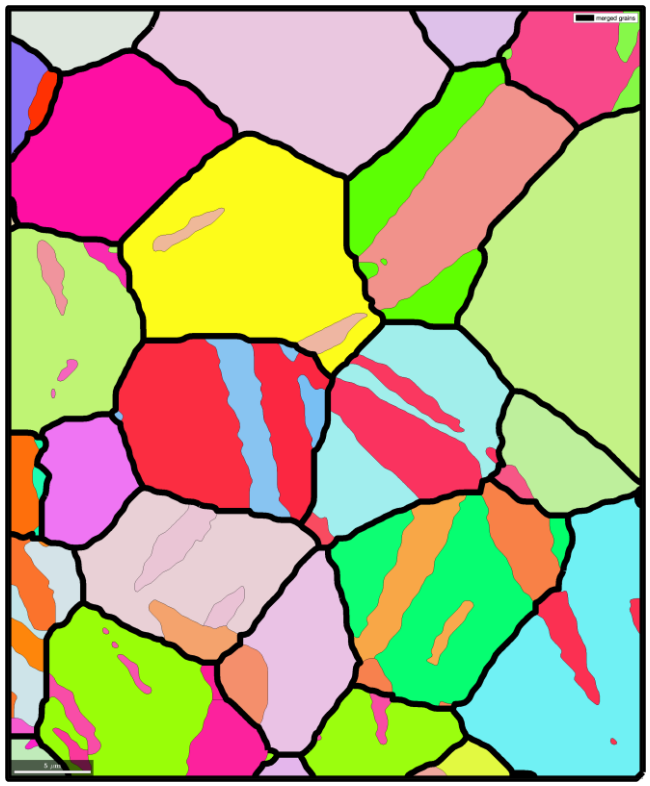


```
CS=grains.CS
twinning = orientation.map(Miller(0,1,-1,-2,CS),Miller(0,-1,1,-2,CS),...
  Miller(2,-1,-1,0,CS),Miller(2,-1,-1,0,CS));

% extract all Magnesium Magnesium grain boundaries
gB = grains.boundary('Magnesium','Magnesium');

% and check which of them are twinning boundaries with threshold 5
degree
isTwinning = angle(gB.misorientation,twinning) < 5*degree;
twinBoundary = gB(isTwinning)

plot(twinBoundary,'linecolor','w','linewidth',4,'displayName',...
    'twin boundary')
```

```
round(twinning.axis)
ans = Miller (show methods, plot)
 size: 1 x 1
 mineral: Magnesium (622, X||a*, Y||b, Z||c)
  h -1
  k  2
  i -1
  l  0

twinning.angle/degree
ans = 86.3471
```

Introduction

Twin boundaries merged

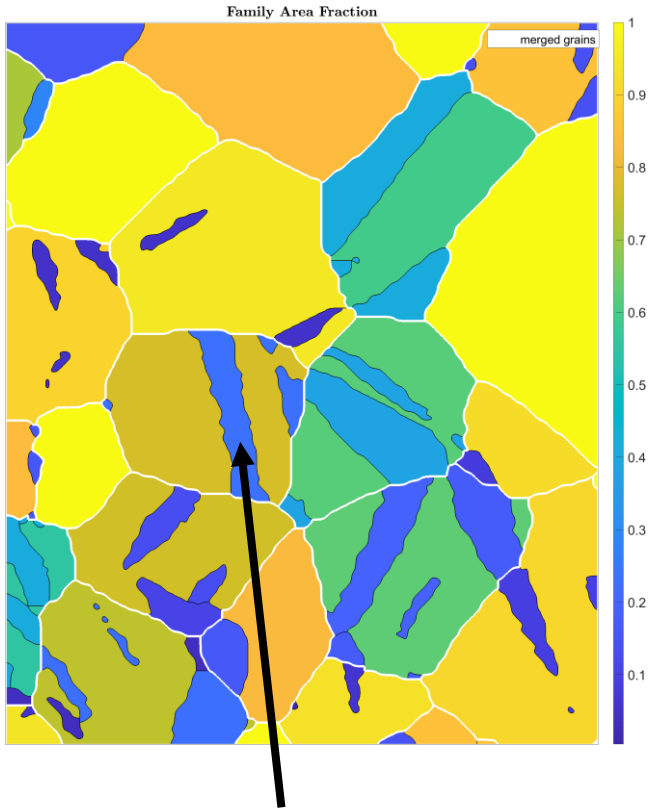


```
%Merge twin boundaries
[mergedGrains,parentId] = merge(grains,twinBoundary);

% plot merged map
figure; plot(grains,grains.meanOrientation)
hold on
plot(mergedGrains.boundary,'linecolor','k','linewidth',10,'linestyle','-',...
  'displayName','merged grains')
```

We can call these merged grains “clusters” in the rest of the talk

Introduction

Family Area Fraction

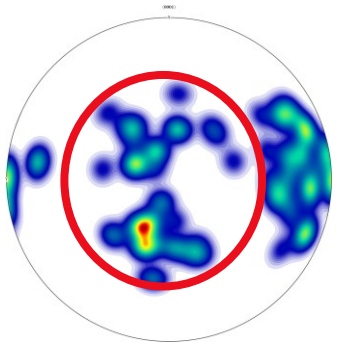


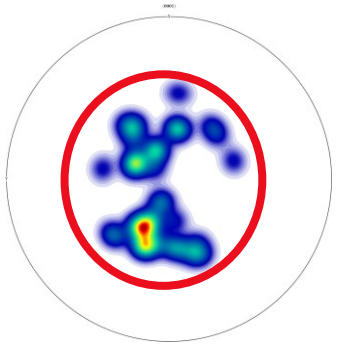
Twin or not?

```
%extract data so not looping over grains and
mGrains structure
ori=grains.meanOrientation;
area=grains.area
mArea=mGrains.area
FArea=zeros(length(grains),1)

%loop over mGrains
for i=1:length(mGrains)
    %Get the logical index of the merged grain
    lid=parentId==i;

    %Group similar orientations into families
    [FId,FCenters] = calcCluster(ori(lid),...
    'maxAngle',10*degree,'method','hierarchical');

    %Compute family areas and store
    area_tmp=area(lid);
    FArea_tmp=zeros(length(FId),1);
    for j=1:length(FCenters)
        lid2=FId==j;
        FArea_tmp(lid2)=sum(area_tmp(lid2));
    end
    FArea(lid)=FArea_tmp/mArea(i);
end
```

What other kind of information do we know about our clusters and fragments?

Texture





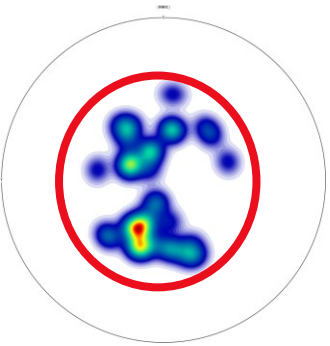
```
%Plot c-axis distribution
figure
plotPDF(grains.meanOrientation,{Miller(0,0,0,1,CS)},'smooth')
```

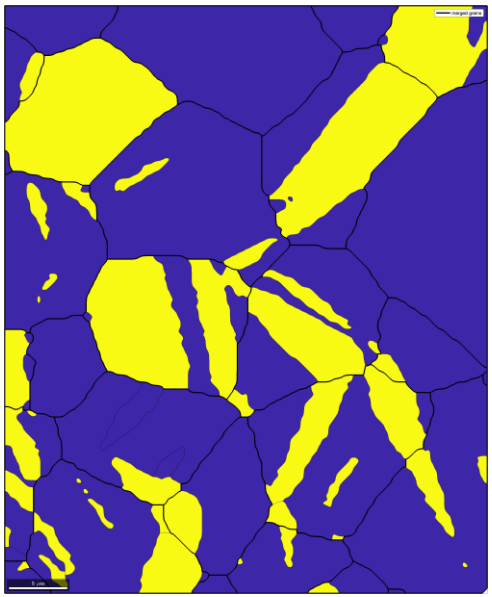
```
%Select c-axis that center around the zvector and plot
cdir=Miller(0,0,0,1,CS)
grain_cdir=grains.meanOrientation.*cdir
lid = angle(grain_cdir, zvector)/degree < 45 | ...
    angle(grain_cdir,-zvector)/degree < 45

figure
plotPDF(grains(lid).meanOrientation,{Miller(0,0,0,1,CS)},'smooth')
```

What other kind of information do we know about our clusters and fragments?

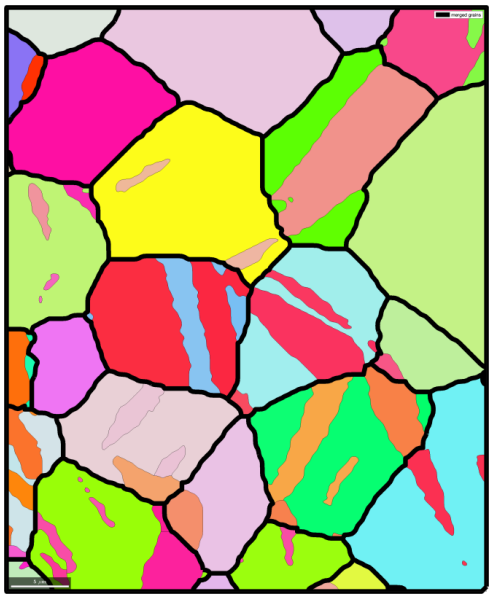
```
%Plot twin (1) and non-twin (0)
Figure
plot(grains,int8(lid))
hold on
plot(mergedGrains.boundary,'linecolor','k','linewidth',2.5,'linestyle','-')
```

Components aligned with loading



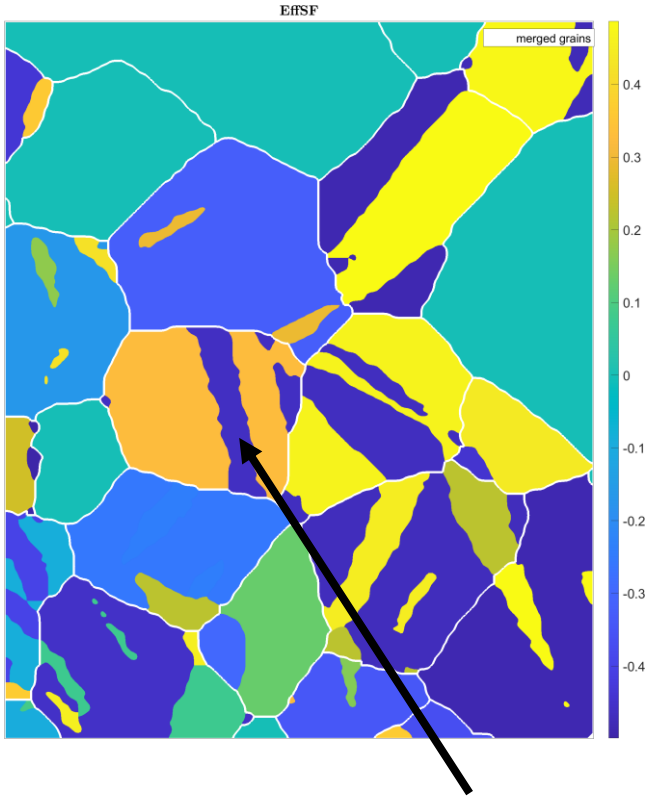


Twin boundaries merged



What other kind of information do we know about our clusters and fragments?

Effective Schmid Factor



Negative means not Twin

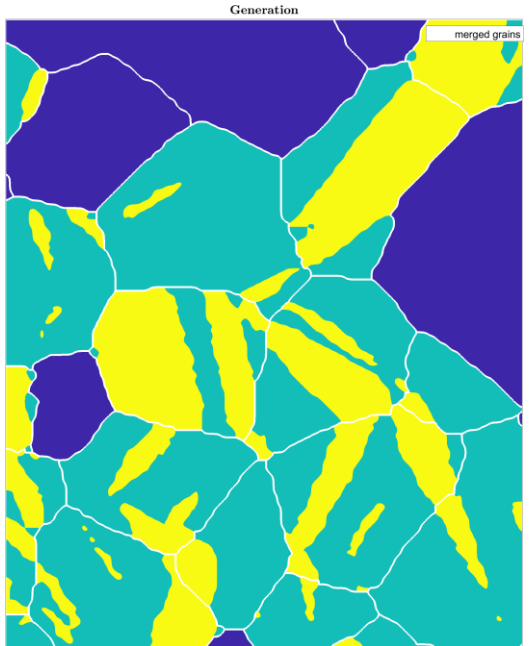
```
%More complicated… pseudo code


%define stress state
sigma=stressTensor([0 0 0; 0 0 0; 0 0 -1])

%define slip systems on the k1 plane in the eta1
direction making sure variants are equivalent to
slip systems
sS=slipSystem(eta1,k1).symmetrise('antipodal')

%determine the active twin variant by finding the
min misorientation among the variants
EffSF = sS.SchmidFactor(grain \ sigma)
```

Main tasks

- Have good quality data
- Clean and reconstruct grain fragments
  - Could be 2d or 3d datasets
- Merge grains that come from a single, root grain
- Family based properties could be useful for identifying relationships in clusters
  - Texture
  - Area
  - Schmid factor
  - Relative boundary ratios
- Need a framework for
  - Editing clusters
  - Interpreting the family data
  - Labeling twin and generation

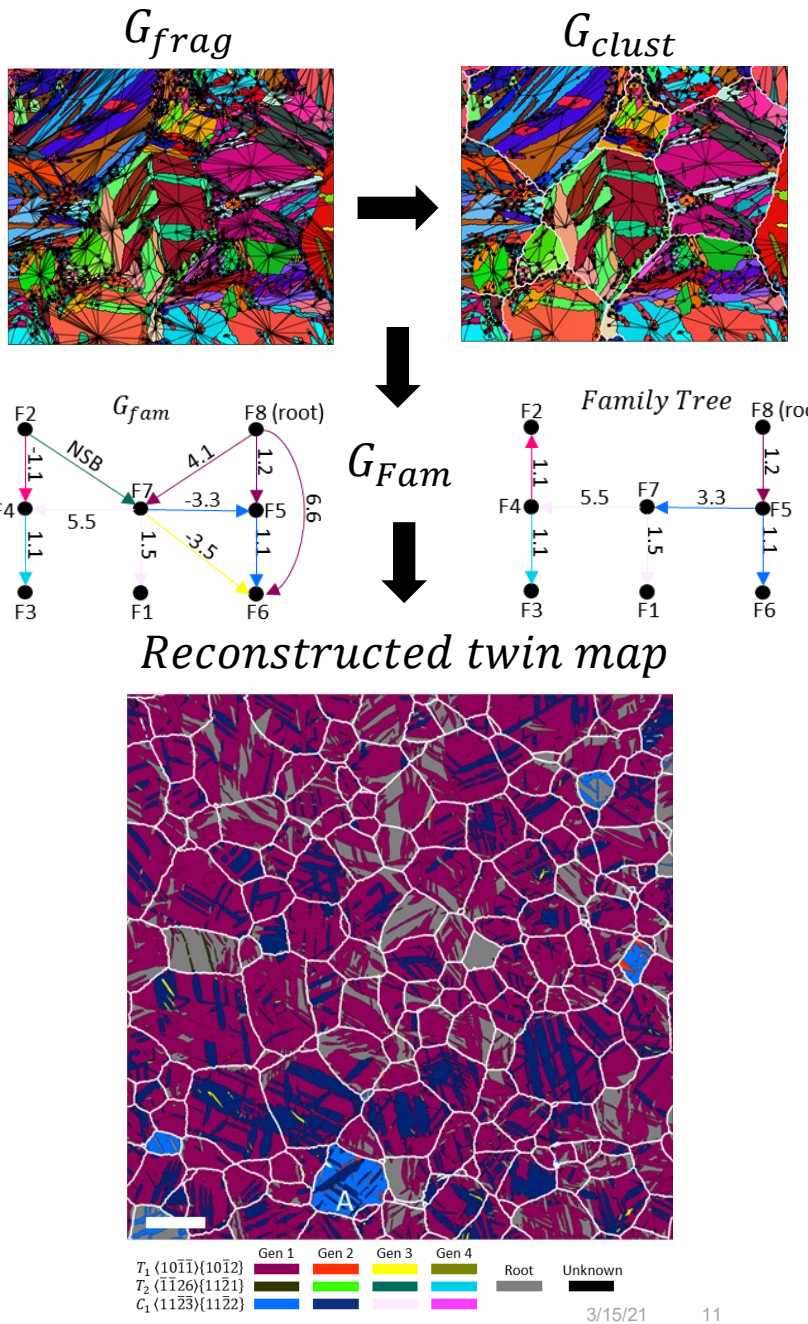
Twin type

Twin Generation



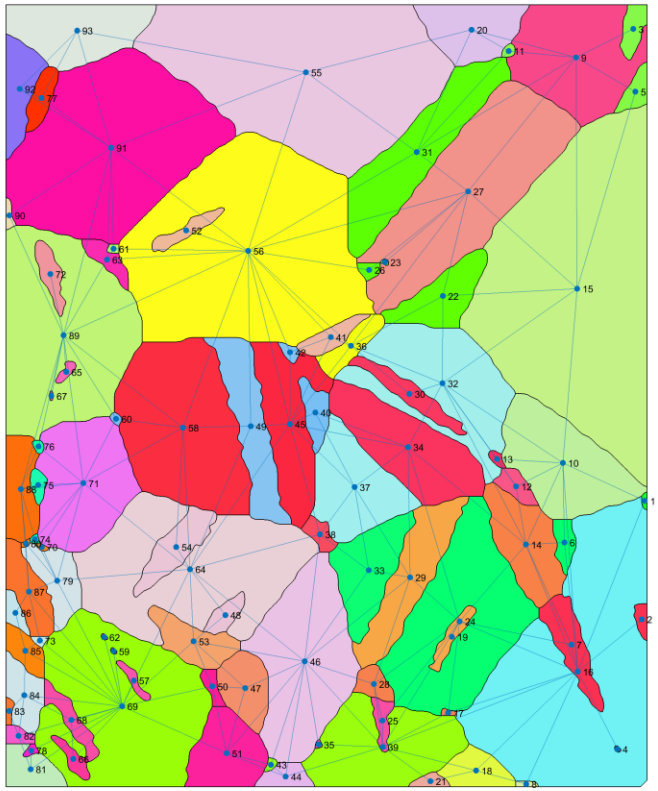
Overview

- Briefly cover what to consider during twin reconstruction
- The graph methodology will be introduced through an example with α-Ti and Mtex Mg dataset
  - Graph theory and algorithms explained
  - Example on specimen with third order twinning
  - Will talk about how crystal plasticity and twin analysis in highly textured samples can be combined
- In case you aren't interested in twinning… the methodology for twin reconstruction can be applied to other crystallographic processes such as phase transformations



The fragment graph

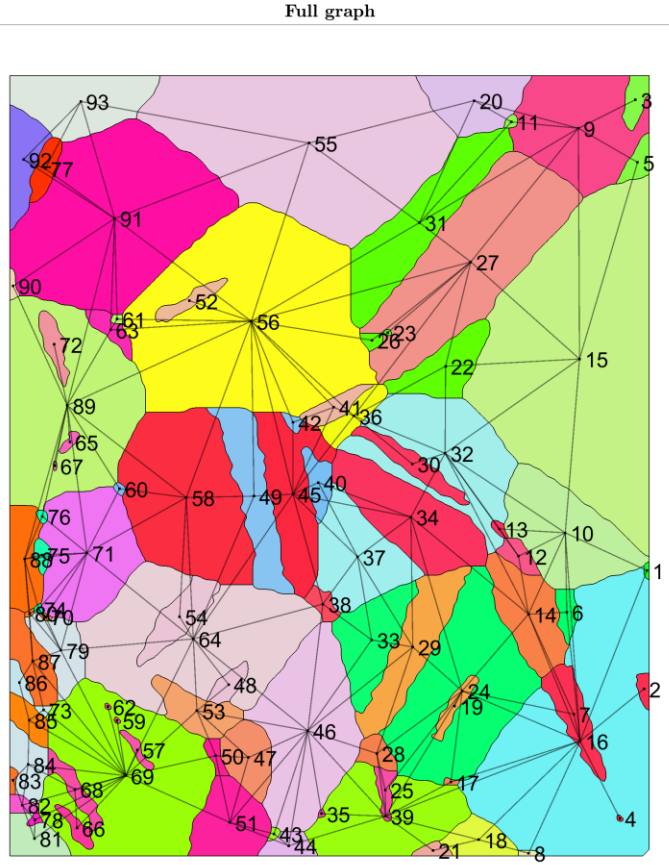
Graph overlay



```
%Compute grain neighbors
[pairs] = neighbors(grains);

%Initialize graph
s=pairs(:,1);
t=pairs(:,2);
G=graph(s,t);
G.Nodes.Id=[1:G.numnodes]'
G.Nodes.centroids=grains.centroid

%plot graph overlay
figure;
plot(grains,...
    G.Nodes.Id,'Micronbar','off','silent');
hold on
p=plot(G,'XData',G.Nodes.centroids(:,1),...
        'YData',G.Nodes.centroids(:,2));
```

The fragment graph with some formatting

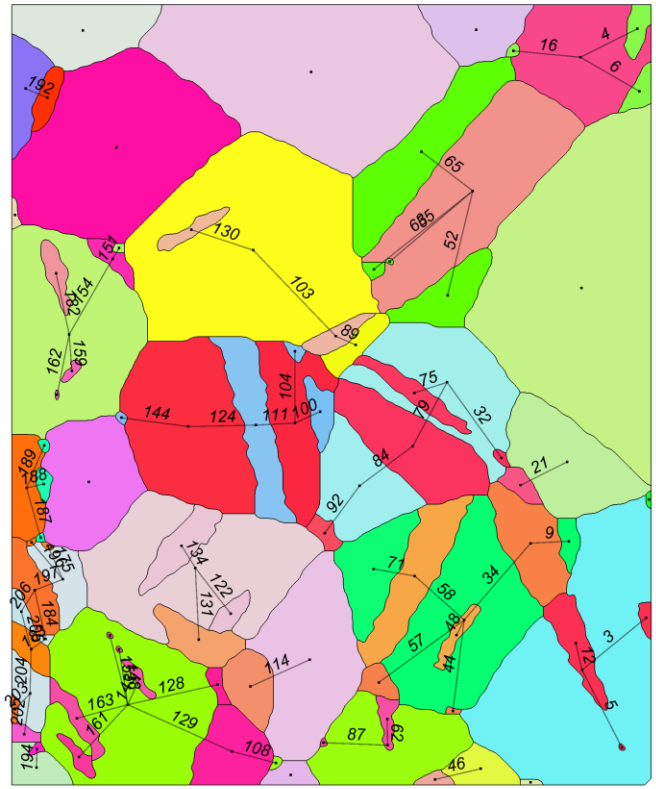
Graph overlay




```
%Compute grain neighbors
[pairs] = neighbors(grains);

%Initialize graph
s=pairs(:,1);
t=pairs(:,2);
G=graph(s,t);
G.Nodes.Id=[1:G.numnodes]'
G.Nodes.centroids=grains.centroid

%plot graph overlay
figure;
plot(grains,...
    G.Nodes.Id,'Micronbar','off','silent');
hold on
p=plot(G,'XData',G.Nodes.centroids(:,1),...
        'YData',G.Nodes.centroids(:,2))
```

The cluster graph

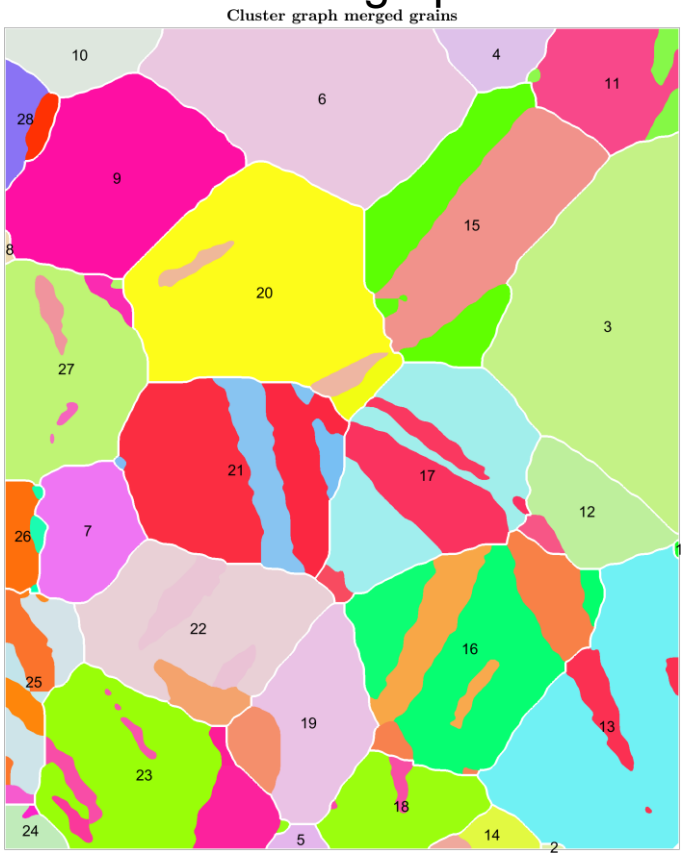
Cluster graph




```
%define the twin misorientation
twinning = orientation.map(Miller(0,1,-1,-
2,CS),Miller(0,-1,1,-2,CS),...
  Miller(2,-1,-1,0,CS),Miller(2,-1,-1,0,CS));

% extract all Magnesium Magnesium grain boundaries
gB = grains.boundary('Magnesium','Magnesium');

% and check which of them are twinning boundaries
with threshold 5 degree
isTwinning = angle(gB.misorientation,twinning) ...
    < 5*degree;
twinBoundary = gB(isTwinning)
combine=zeros(G.numedges,1,'logical');

grainBId=twinBoundary.grainId;
for i=1:length(grainBId)
    combine(all(G.Edges.EndNodes==grainBId(i,:),2)
|...
all(G.Edges.EndNodes==fliplr(grainBId(i,:)),2))=tru
e;
end
G_clust=rmedge(G,G.Edges.EndNodes(~combine,1),...
    G.Edges.EndNodes(~combine,2));
```

The cluster graph

Cluster graph




```
%merge grains using node pairs from G_clust
mGrains=merge(grains,G_clust.Edges.EndNodes)

%plot
figure;plot(grains,grains.meanOrientation,'noBoundary');hold on;
plot(mGrains.boundary,'lineWidth',2,'lineColor','w');
text(mGrains,int2str(mGrains.id));hold off
```

Should be easy to visually interact with these clusters…

Editor - D:\Dropbox\To copy\MATLAB\Twin-Analysis\Analysis Database\Examples\Mg-MTEX standard twin dataset\Segment_Grains.m*

```
103 -    text(mGrains,int2str(mGrains.id));hold off;
```

Segment_Grains.m*

Command Window

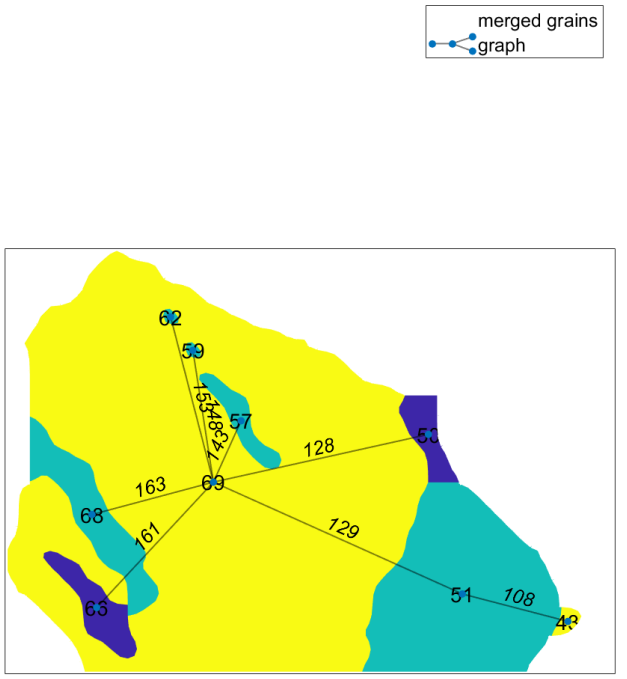
New to MATLAB? See resources for Getting Started.

```
  ====================================
  Group 20
  Node List
  Family 1, Node Id 50 66
  Family 2, Node Id 51 57 59 62 68
  Family 3, Node Id 43 69
  Edge List
  Id:   108, type:   1 Node, Pair/Parent:    43    51,  0 0
  Id:   128, type:   1 Node, Pair/Parent:    50    69,  0 0
  Id:   129, type:   1 Node, Pair/Parent:    51    69,  0 0
  Id:   143, type:   1 Node, Pair/Parent:    57    69,  0 0
  Id:   148, type:   1 Node, Pair/Parent:    59    69,  0 0
  Id:   153, type:   1 Node, Pair/Parent:    62    69,  0 0
  Id:   161, type:   1 Node, Pair/Parent:    66    69,  0 0
  Id:   163, type:   1 Node, Pair/Parent:    68    69,  0 0
  ====================================
  processing options
  Press..
  enter to proceed to next grain cluster
  0 to exit editor and return exit flag
  1 to remove an edge
  2 to add an edge
  3 to remove all edges connected to a node
  4 to try adding all edges connected to a node
  5 merge merged grains
  6 to plot the grain a different way
  7 to change labeling
  8 to get misorientation of grains
  9 to get all edge relationships with node
  10 go back to previous grain
  11 cycle
  12 switch to Family editor
fx Enter Number:
```

Workspace

Figures - Figure 4

Figure 4




```
%Start the graph editor with a list of merge grain ids
groups=unique([23])
value=G.Nodes.FamilyID;
GraphEditor(groups,1,[],G_clust,G,grains,mGrains,value,0,1,0,1,2);

%To remove edge 129 enter 1, enter 129, enter 0 and reconstruct G_clust
```

Modifications stored in a database of edge and node edits

```
text(mGrains,int2str(mGrains.id));hold off;
```

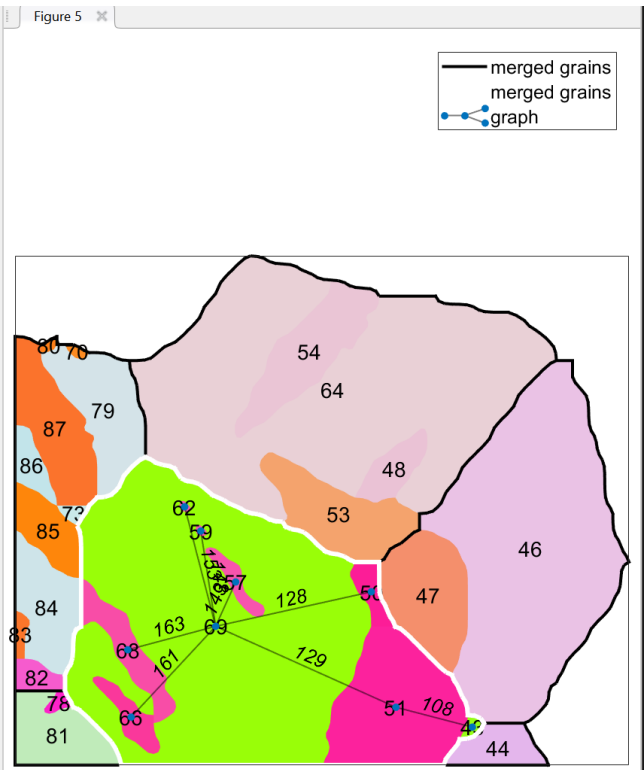
Segment_Grains.m

Command Window

New to MATLAB? See resources for Getting Started.

```
processing options
Press..
enter to proceed to next grain cluster
0 to exit editor and return exit flag
1 to remove an edge
2 to add an edge
3 to remove all edges connected to a node
4 to try adding all edges connected to a node
5 merge merged grains
6 to plot the grain a different way
7 to change labeling
8 to get misorientation of grains
9 to get all edge relationships with node
10 go back to previous grain
11 cycle
12 switch to Family editor
Enter Number: 6
1 to plot mean orientation
2 to plot FamilyId
3 to plot EffSchmid
Enter plot option number: 1
Enter if neighbor should be plotted (0 or 1): 1
```

```
Group 20
Node List
Family 1, Node Id 50 66
Family 2, Node Id 51 57 59 62 68
Family 3, Node Id 43 69
Edge List
Id:   108, type:   1 Node, Pair/Parent:    43    51,  0 0
Id:   128, type:   1 Node, Pair/Parent:    50    69,  0 0
```



Gives a simple, fast way of interacting with grain datasets

Editor - D:\Dropbox\To copy\MATLAB\Twin-Analysis\Analysis Database\Examples\Mg-MTEX standard twin dataset\Segment_Grains.m

Segment_Grains.m | getTwinProperties.m | Inputs.m | InitialGraph.m | Segment_Grains.m | MergeByBoundary.m | subSet.m | subSet.m | subsref.m | merge.m | Grain_reconstruction.m | ClusterGraph.m | eRemoveList.txt | AssignFamilyIDs.m | Mtex_example.m

```
199   %exflagGroup=6 entered fix circular relationship fnc too many times
200   groups=cleanGroups(exflagGroup>0)
201   % groups=unique(G_clust.Nodes.Group(find(G_clust.Nodes.computeFamily))
202   groups=[19]
203   value=G.Nodes.FamilyID;
204   [G_Family] = GraphEditor(groups,1,G_Family,G_clust,G,grains,mGrains,va
205
206
207   %% Compute the twin fraction
208
209   [mGrains,twinVF] = getTwinFractions(G,grains,mGrains,opt);
210
211
212   %% Twin thickness
213
214   [G] = TwinThickness(G,grains,opt);
```

Command Window

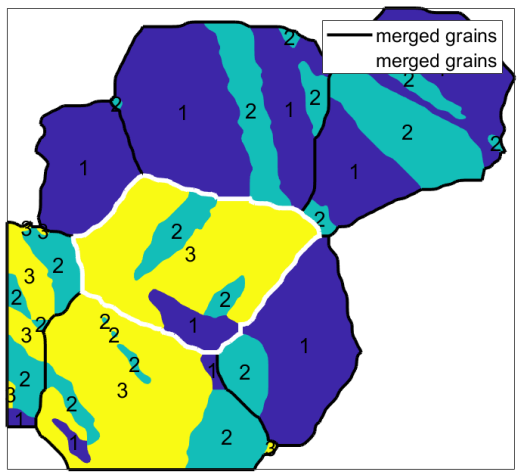
New to MATLAB? See resources for Getting Started.

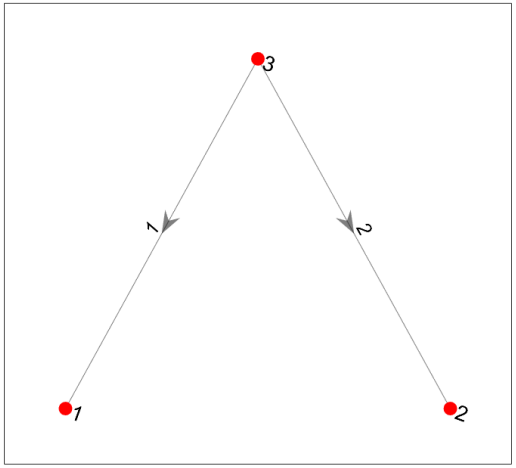
```
Current root: 3
Edge List
Id:     1, type:   1, Pair/Parent:    27    25,  1 0
Id:     2, type:   1, Pair/Parent:    27    26,  1 0
==================================
processing options
Press..
enter to proceed to next grain cluster
0 to exit editor and return exit flag
1 to remove an edge
2 to remove edges connected to Family
3 to flip parent relationship
4 to plot the grain a different way
5 to specify root
6 to switch editor mode to cluster
7 go back to previous grain
8 cycle
Enter Number:
```

Workspace

Figures - Figure 2

Figure 2





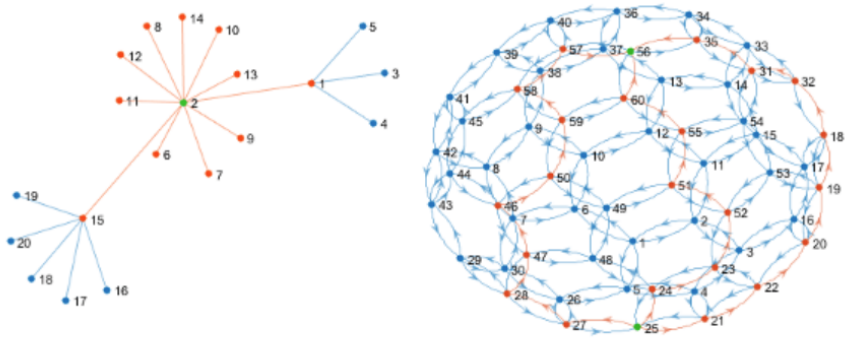
This dataset is not that interesting for understanding the family tree and family graph.

Graph and Network Algorithms

R2021a

Directed and undirected graphs, network analysis

Graphs model the connections in a network and are widely applicable to a variety of physical, biological, and information systems. You can use graphs to model the neurons in a brain, the flight patterns of an airline, and much more. The structure of a graph is comprised of “nodes” and “edges”. Each node represents an entity, and each edge represents a connection between two nodes. For more information, see Directed and Undirected Graphs.



Functions

expand all

- Construction
- Modify Nodes and Edges
- Analyze Structure
- Traversals, Shortest Paths, and Cycles
- Matrix Representation
- Node Information
- Visualization

A more challenging dataset

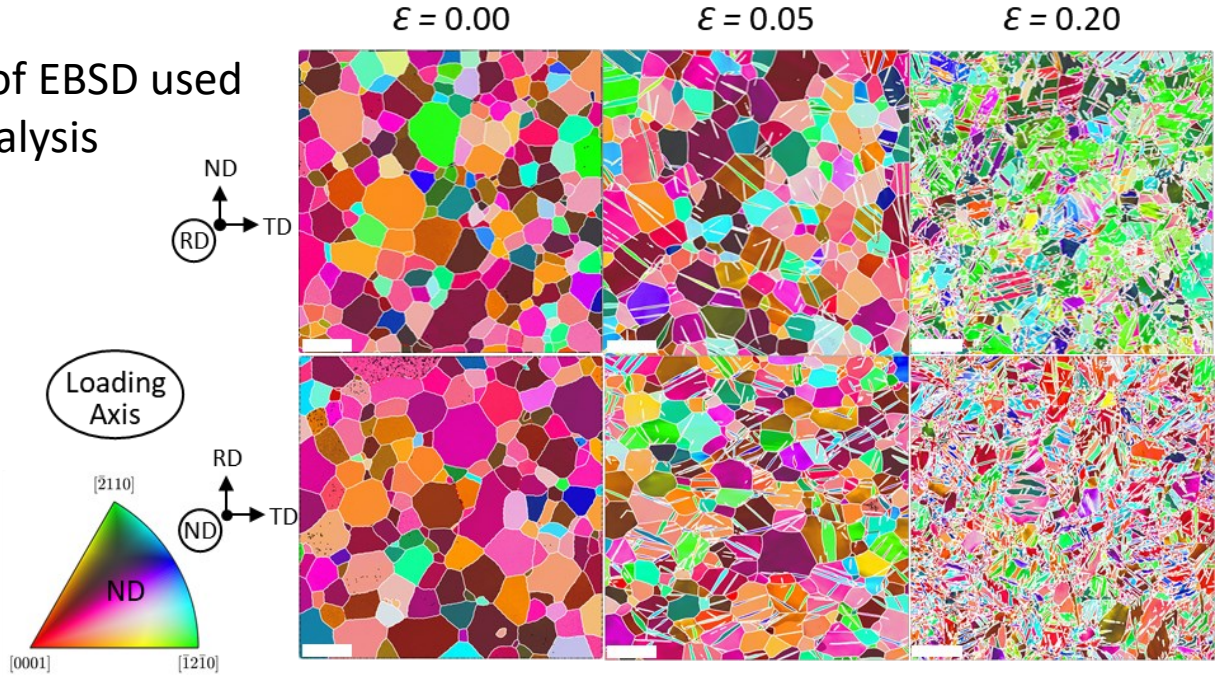
Material: Rolled plate, High purity (99.999%) α-Ti with grain size of 150μm

Deformation: Deformed in compression along the RD and ND directions

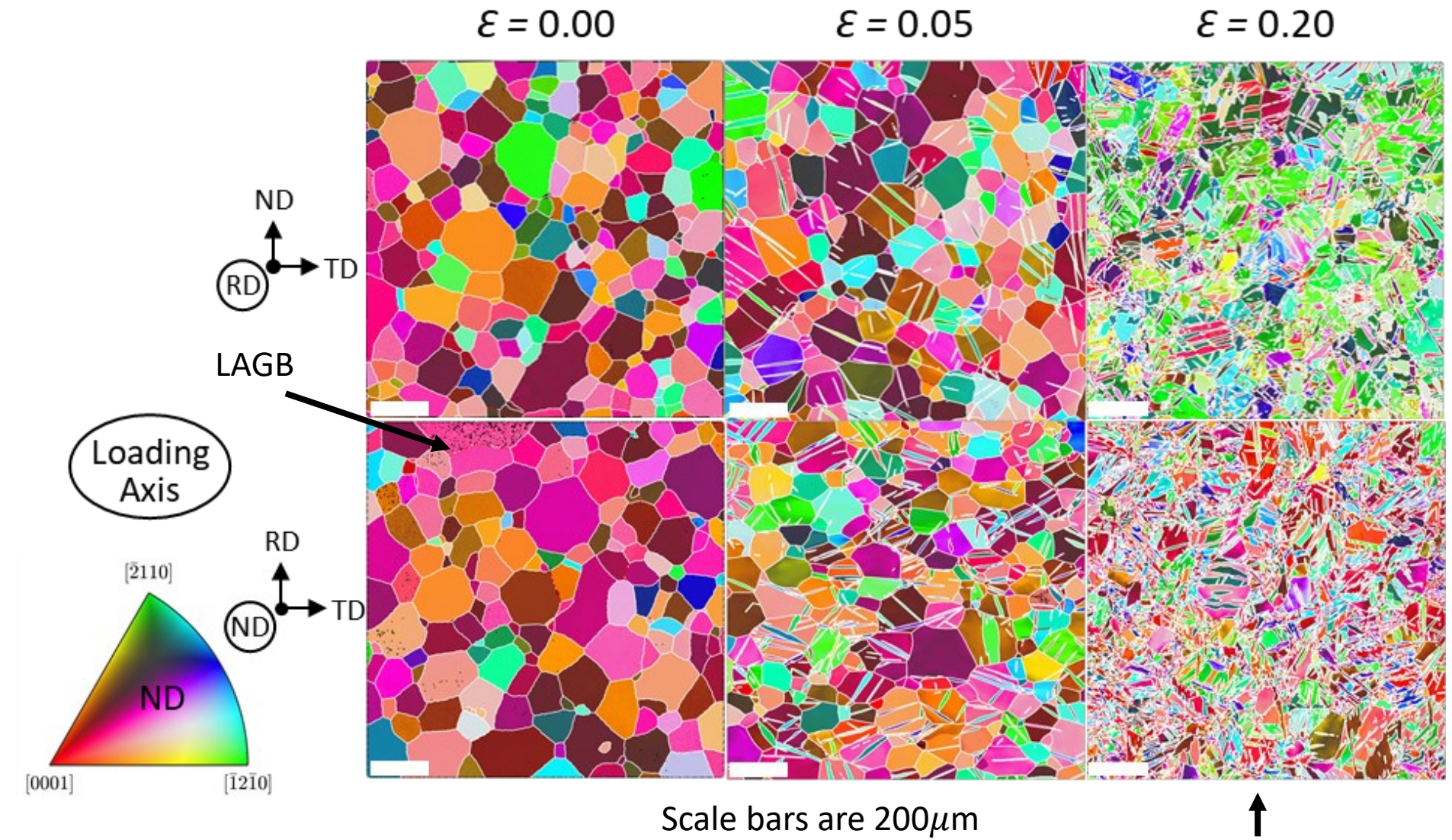
Material characterization:

- Electropolished procedure published in Ferrari et al, MC (2020)
- EBSD collected with 0.2μm step size in the plane of compression
- Analysis published in Savage et al, MC (2020)

IPFs of EBSD used in analysis



Scale bars are 200μm



Scale bars are 200μm

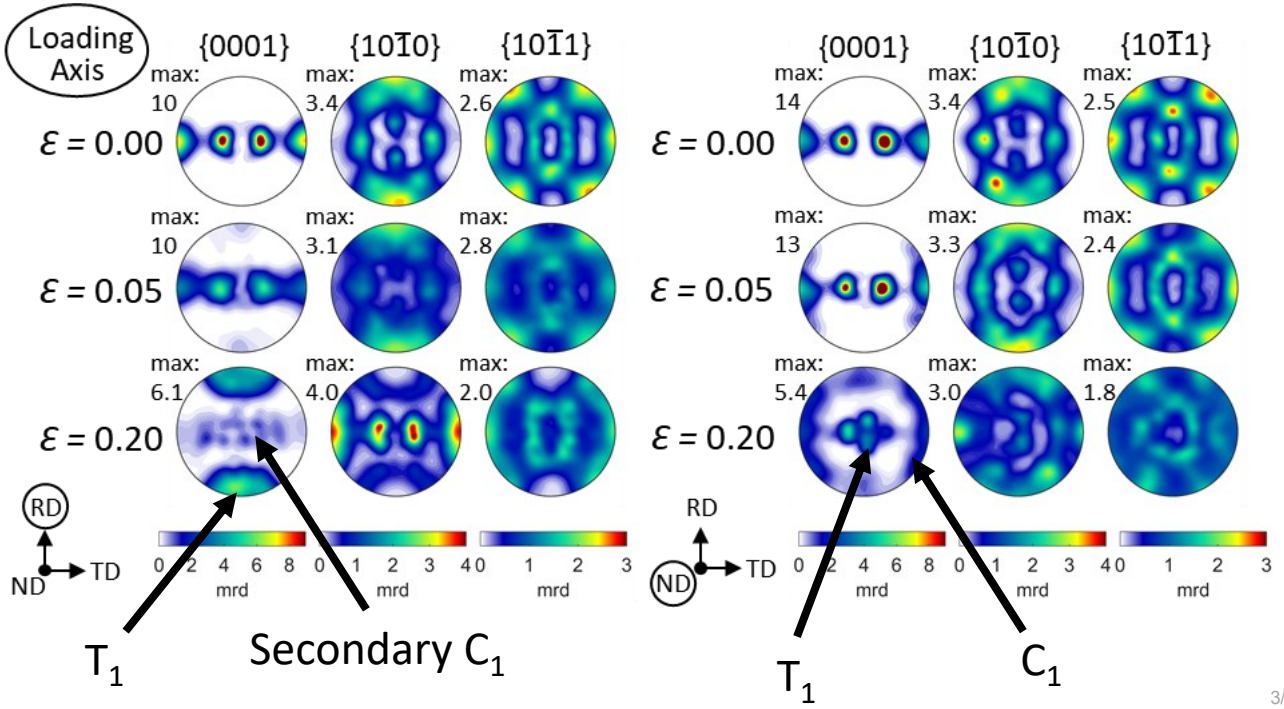
Very twinned and lots of orientation gradients

Texture

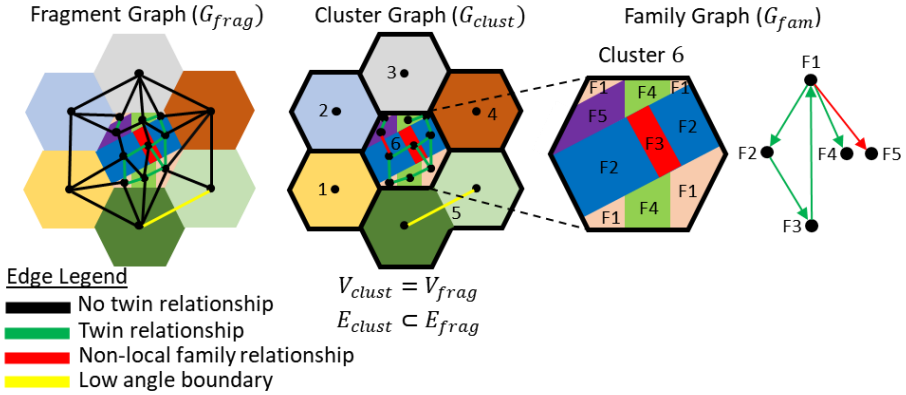
Twins types that are expected.

α	K_1	η_1	Angle [°]	Axis	β[°]	$\beta_{relaxed}$[°]	ϕ[°]
T_1	$\{10\bar{1}2\}$	$\langle\bar{1}011\rangle$	85.04	$\langle\bar{1}2\bar{1}0\rangle$	5	10	10
T_2	$\{11\bar{2}1\}$	$\langle\bar{1}\bar{1}26\rangle$	34.95	$\langle\bar{1}100\rangle$	2	10	10
C_1	$\{11\bar{2}2\}$	$\langle11\bar{2}\bar{3}\rangle$	64.40	$\langle\bar{1}100\rangle$	5	10	10

Pole figures taken from EBSD maps

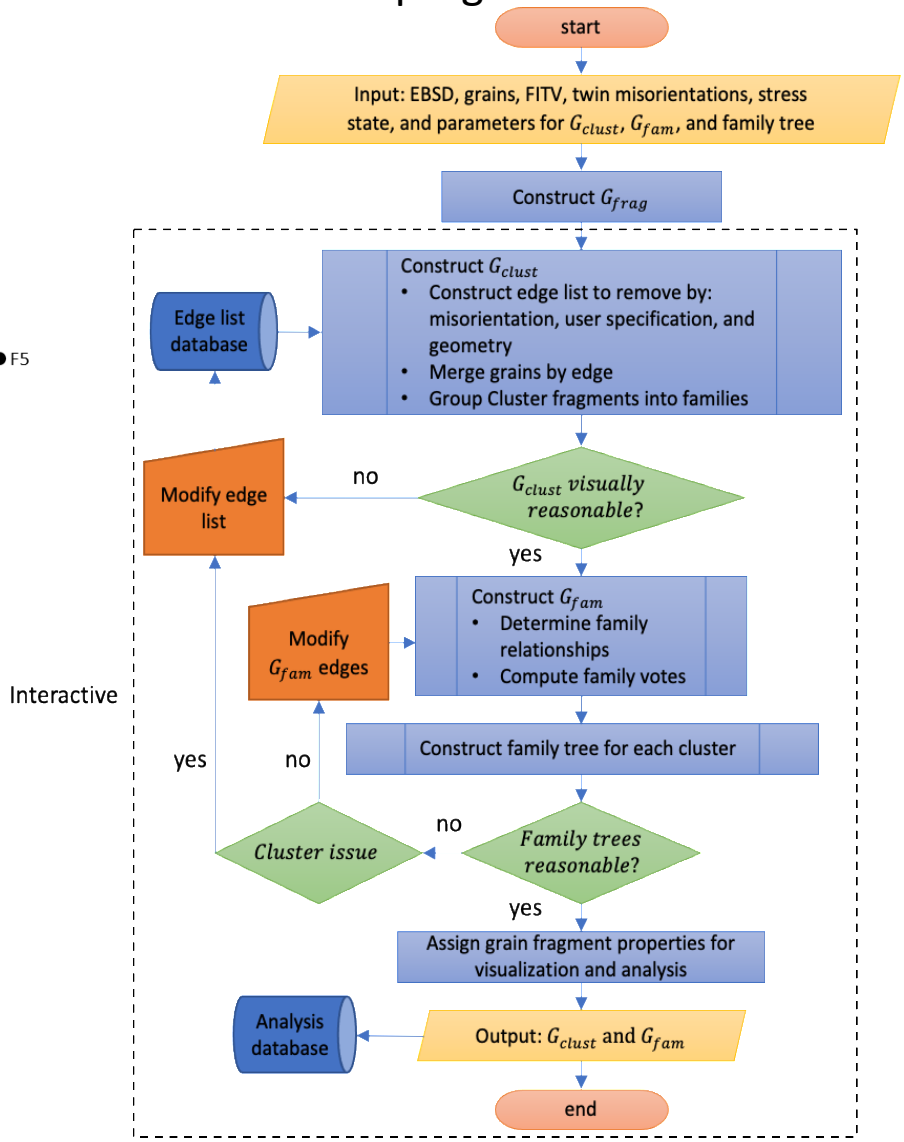


Making the G_{frag}

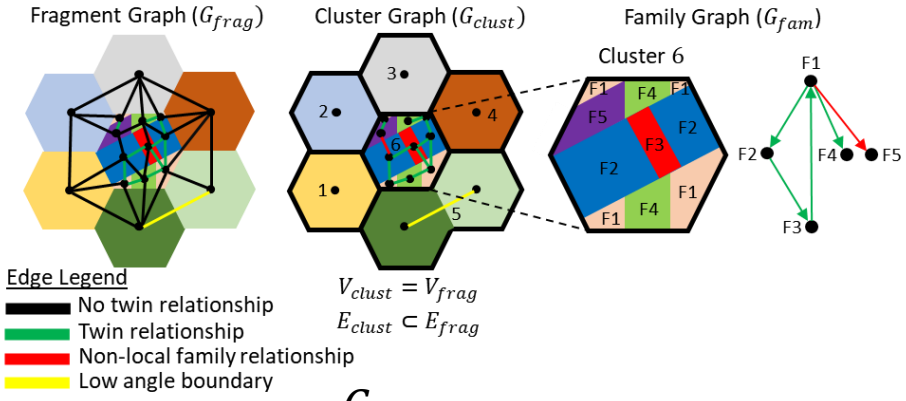


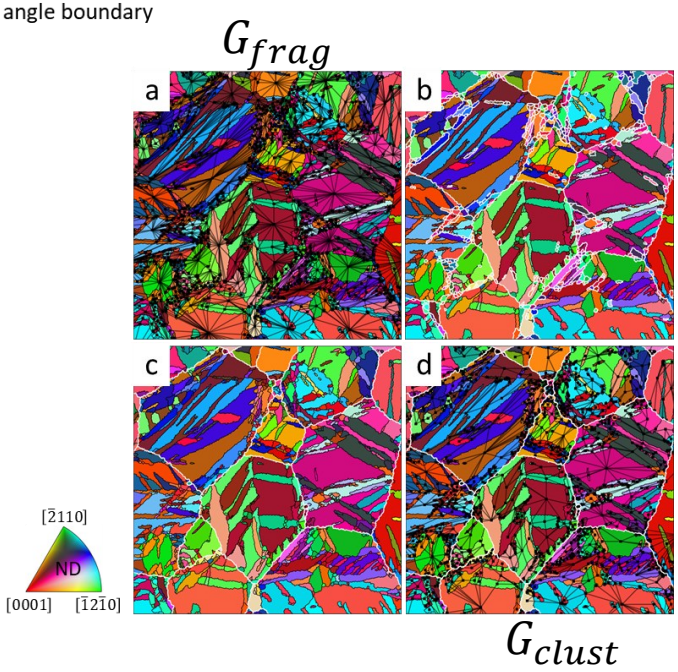

```
%Compute grain neighbors
pairs = neighbors(grains);

%Initialize graph
s=pairs(:,1);
t=pairs(:,2);
G_frag=graph(s,t);
```

The program work flow

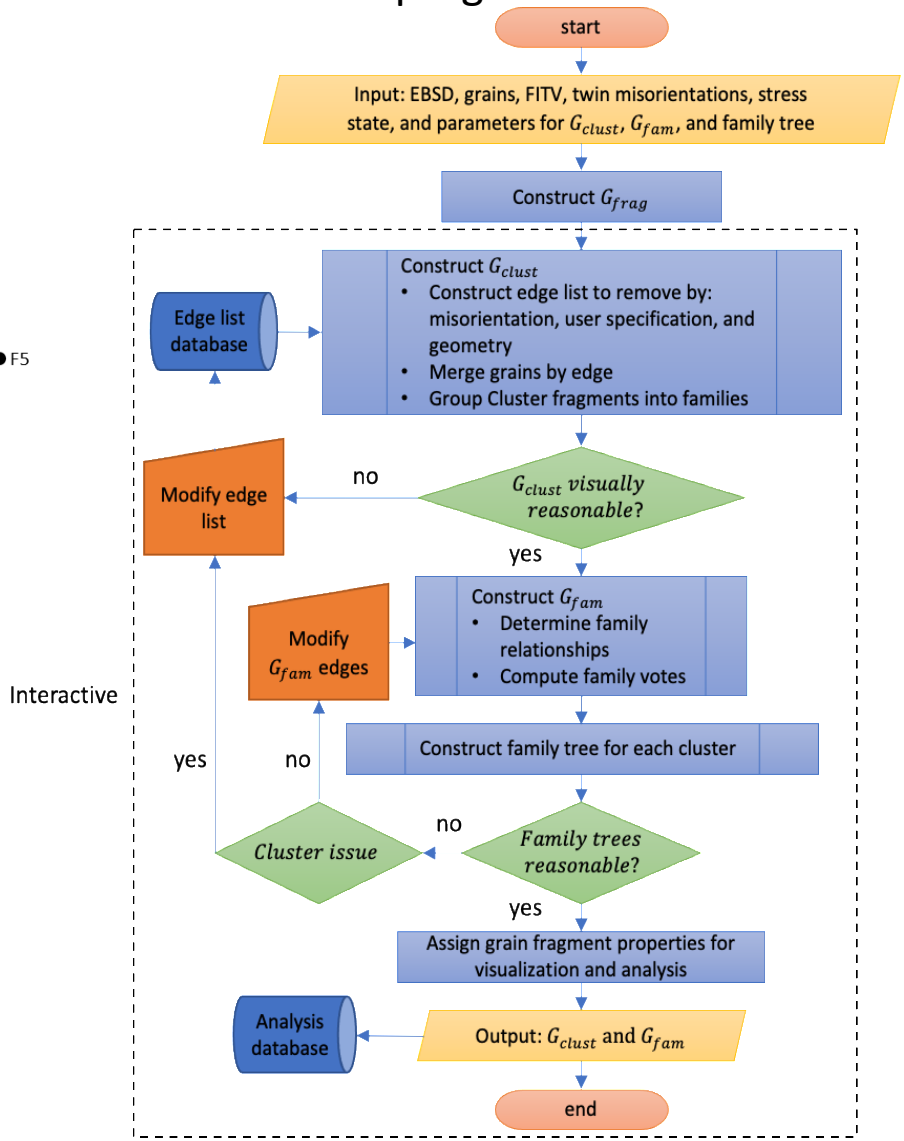


Making the G_{clust}



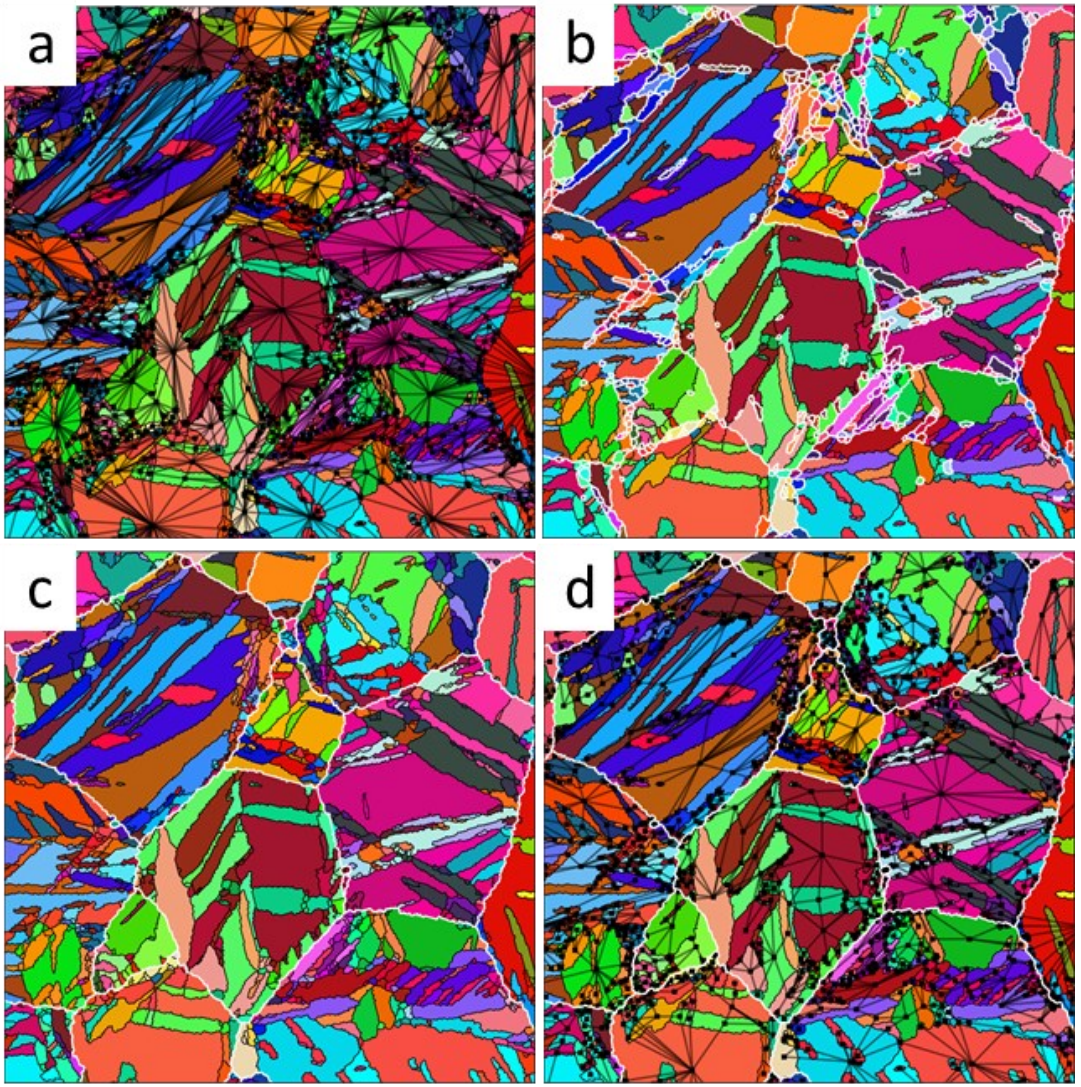


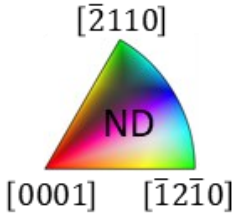
The program work flow



Making the G_{clust}

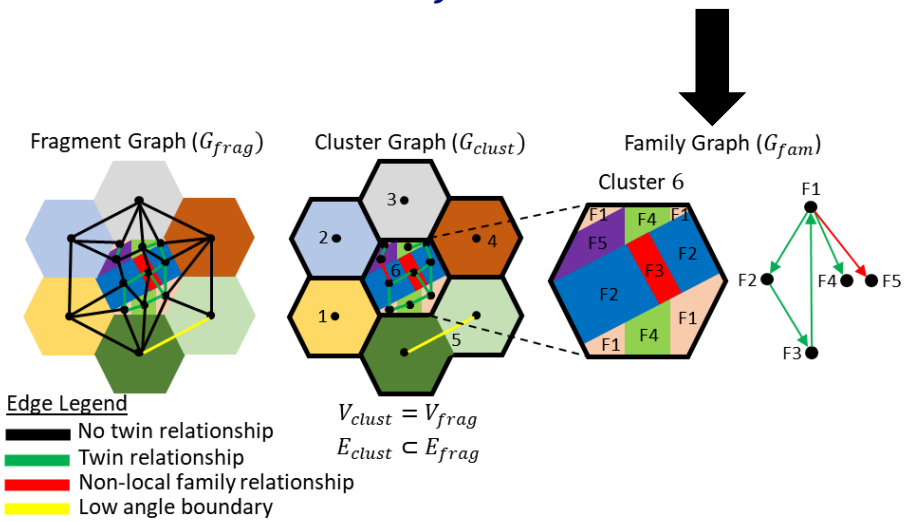
G_{frag}





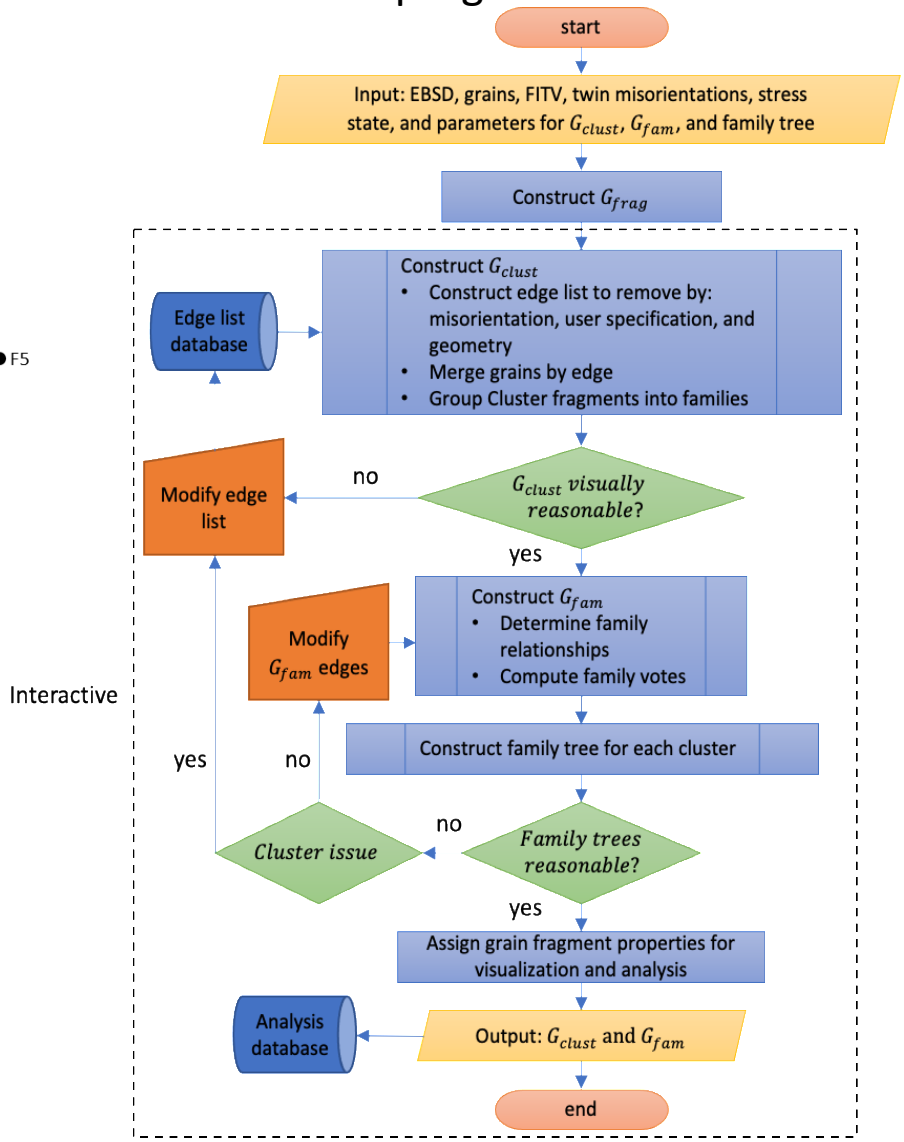
G_{clust}

Making the G_{fam}



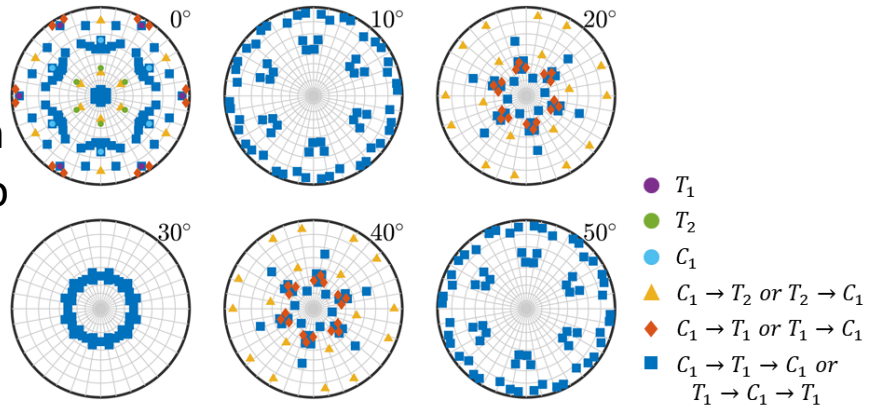
Not so simple…

The program work flow



The number of twin variants with higher order twins could be a problem

(0001) sigma sections with twin orientation applied to the cell definition



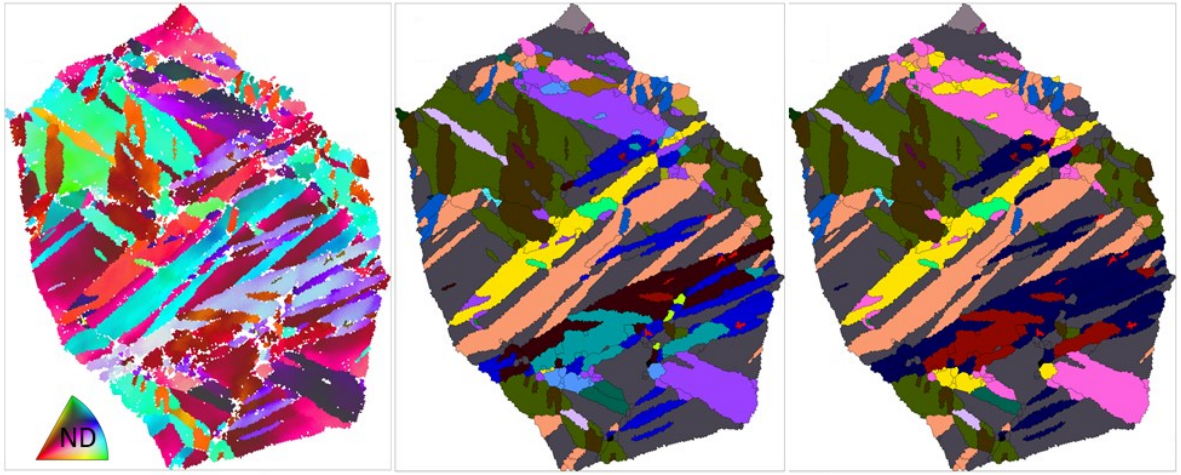
Twinning Sequence	Number of Twin Variants	Number Circular Relationships								
		Mis. Tolerance 5°			Mis. Tolerance 7.5°			Mis. Tolerance 10°		
		T_1	T_2	C_1	T_1	T_2	C_1	T_1	T_2	C_1
$T_1 \rightarrow C_1$ or $C_1 \rightarrow T_1$	36	-	-	-	12	-	-	12	-	-
$T_2 \rightarrow C_1$ or $C_1 \rightarrow T_2$	36	-	-	-	-	6	-	-	6	-
$C_1 \rightarrow T_1 \rightarrow C_1$	216	24	-	-	24	-	-	48	-	-
$T_1 \rightarrow C_1 \rightarrow T_1$	216	-	-	24	-	-	24	-	-	72
$C_1 \rightarrow T_1 \rightarrow C_1 \rightarrow T_1$ or $T_1 \rightarrow C_1 \rightarrow T_1 \rightarrow C_1$	1296	12	-	-	12	-	132	12	12	264

In some grains slip and the number of twin variants are a problem – mitigated with anisotropic clustering

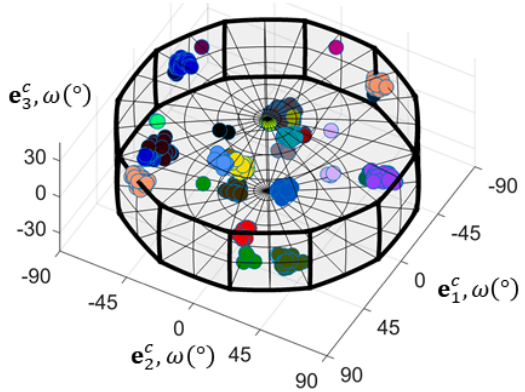
EBSD IPF

Unique Family

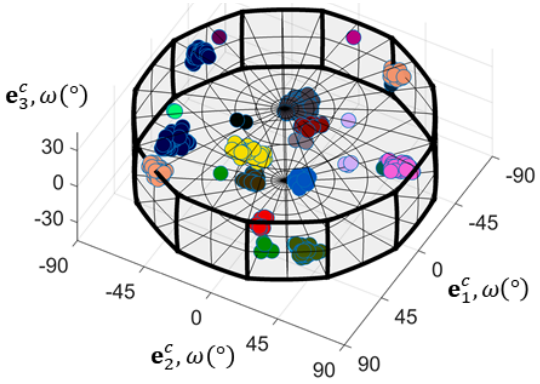
Unique Family +



Unique Family
angle axis plot

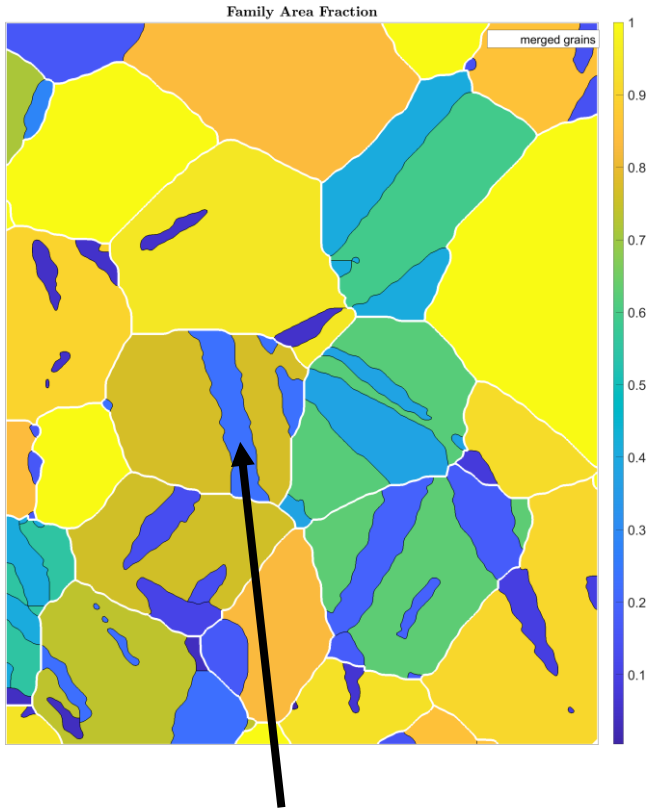


Unique Family +
angle axis plot



Introduction

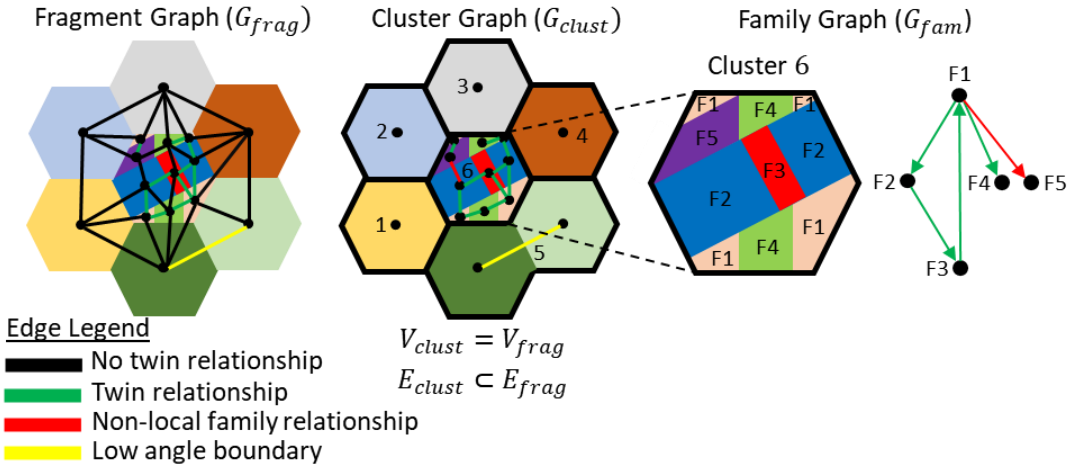
Family Area Fraction



Twin or not?

```
%extract data so not looping over grains and
mGrains structure
ori=grains.meanOrientation;
area=grains.area
mArea=mGrains.area
FArea=zeros(length(grains),1)

%loop over mGrains
for i=1:length(mGrains)
    %Get the logical index of the merged grain
    lid=parentId==i;

    %Group similar orientations into families
    [FId,FCenters] = calcCluster(ori(lid),...
    'maxAngle',10*degree,'method','hierarchical');

    %Compute family areas and store
    area_tmp=area(lid);
    FArea_tmp=zeros(length(FId),1);
    for j=1:length(FCenters)
        lid2=FId==j;
        FArea_tmp(lid2)=sum(area_tmp(lid2));
    end
    FArea(lid)=FArea_tmp/mArea(i);
end
```

Making the G_{fam}



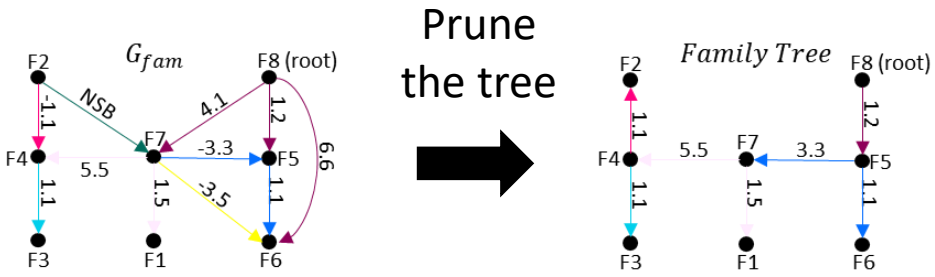
Family with highest relative vote is the parent

$$Vote_1 = w_{FESF}\left(FESF_1 - FESF_2\right) + w_{FA}\left(\frac{FA_1 - FA_2}{FA_1 + FA_2}\right) + w_{FGBL}\left(\frac{FGBL_1 \cap FGBL_2}{FGBL_2} - \frac{FGBL_1 \cap FGBL_2}{FGBL_1}\right)$$

$$Vote_2 = w_{FESF}\left(FESF_2 - FESF_1\right) + w_{FA}\left(\frac{FA_2 - FA_1}{FA_1 + FA_2}\right) + w_{FGBL}\left(\frac{FGBL_1 \cap FGBL_2}{FGBL_1} - \frac{FGBL_1 \cap FGBL_2}{FGBL_2}\right)$$

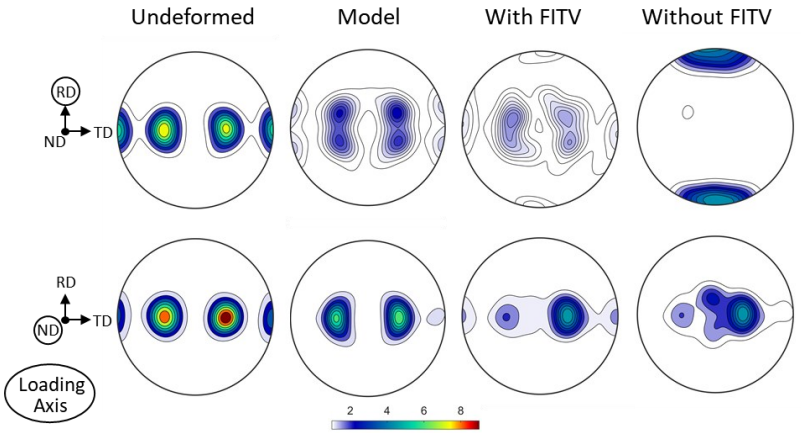
From this we get a directional graph relating all families in a cluster

Making the Family Tree



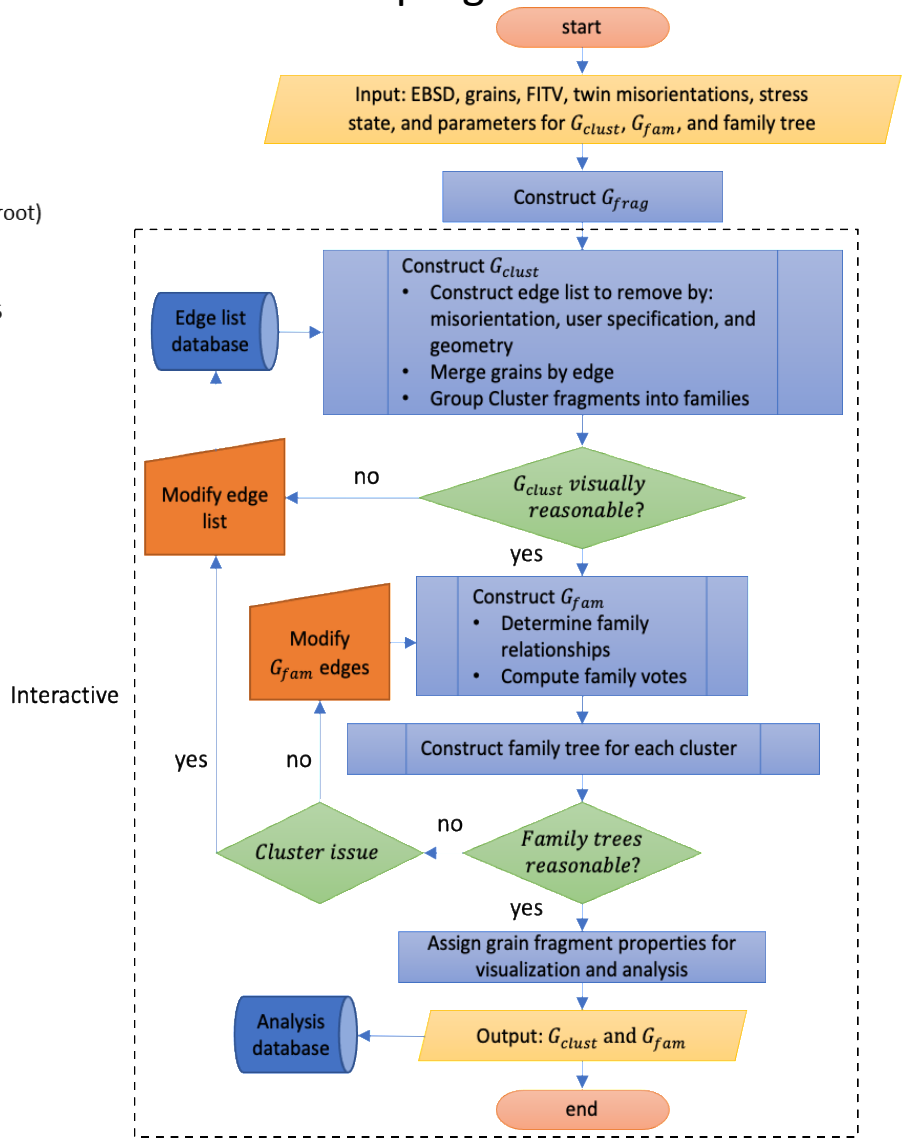
Tasks:

- Find the root family (out-closeness and FITV)
- Determine the children (Prim algorithm)



$$c_i = \left(\frac{A_i}{N-1}\right)^2 \frac{FITV_i}{C_i}, i = 1...N_{fam}$$

The program work flow

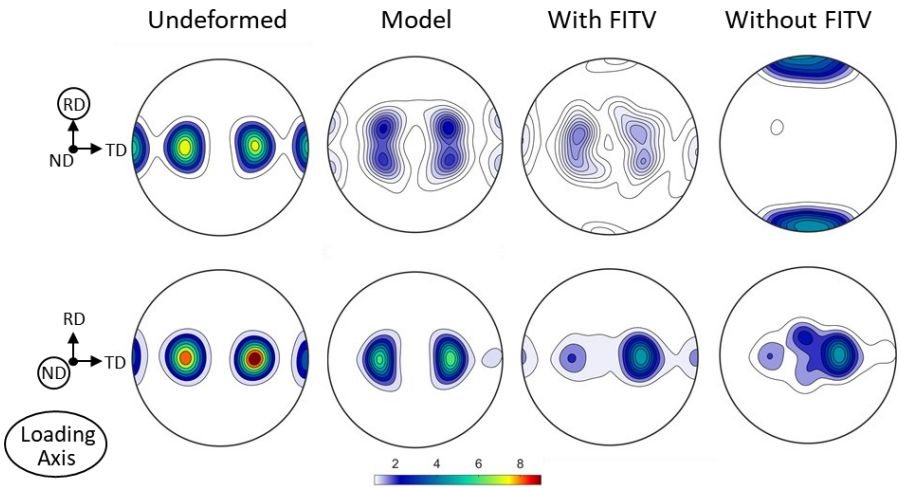


Making the Family Tree – the root

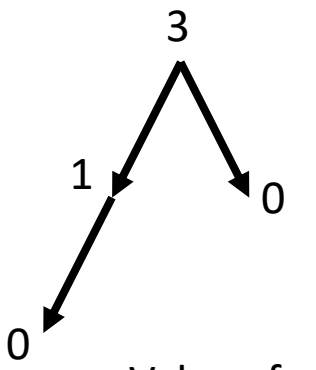
$$c_i = \left(\frac{A_i}{N-1}\right)^2 \frac{FITV_i}{C_i}, i = 1...N_{fam}$$

FITV calculation

```
%radius to compute volume over
radius=10*degree
volumeInitialODF=zeros(length(grains),1);
ori=grains.meanOrientation;

parfor i=1:length(grains)
    FITV(i) = volume(initialODF,ori(i),radius);
end
```

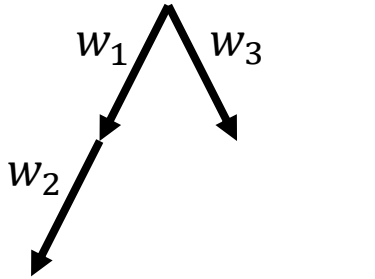


Values for A_i



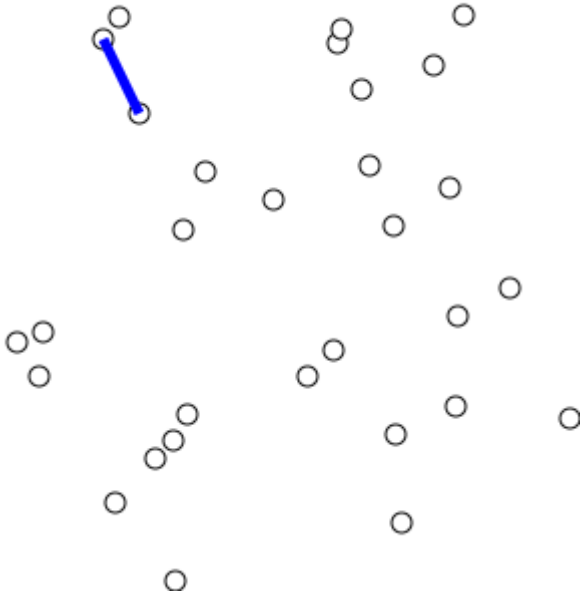
Values for C_i

$$C = w_1 + w_2 + w_3$$



Making the Family Tree

Prim Algorithm



Idea:

- Find a tree that minimizes the total distance traveled and spans every grain

Strengths:

- Highly robust
- Lots of flexibility

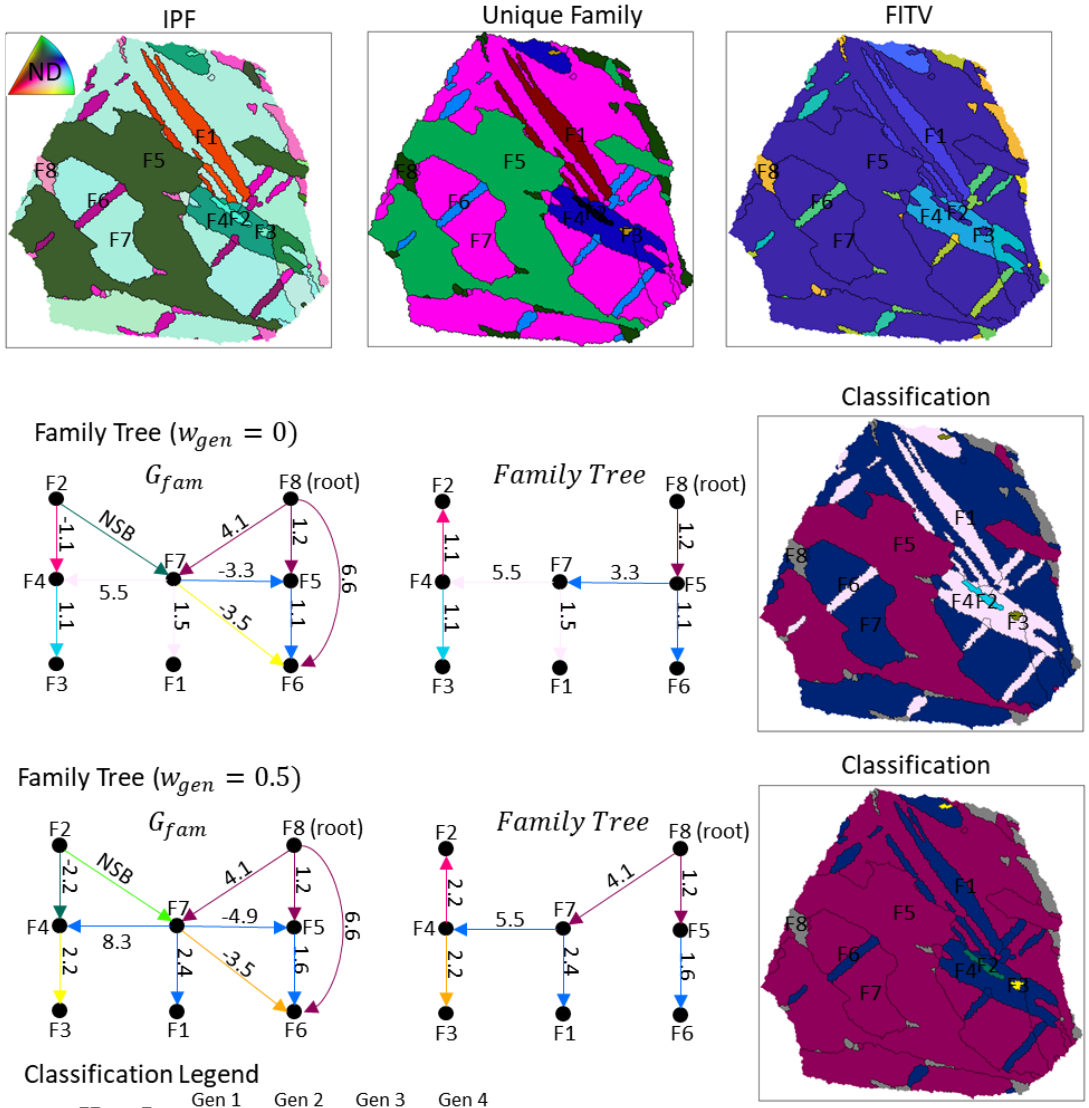
Weaknesses

- Doesn't address whether a grain should be spanned

$$L_E = \left(\frac{1 + w_{gen}\left(gen_{parent}\right)}{3\left(w_{FGBL} + w_{FESF} + w_{FAR}\right)} \left[\begin{array}{l} w_{FGBL} \dfrac{FGBL_{parent} \cap FGBL_{child}}{FGBL_{child}} + \ldots \\ w_{FESF} \dfrac{FESF_{parent} + 0.5}{1} + \ldots \\ w_{FA} \dfrac{FA_{parent}}{FA_{parent} + FA_{child}} \end{array} \right] \right)^{-1}.$$

The gen parameter is a depth vs breadth control

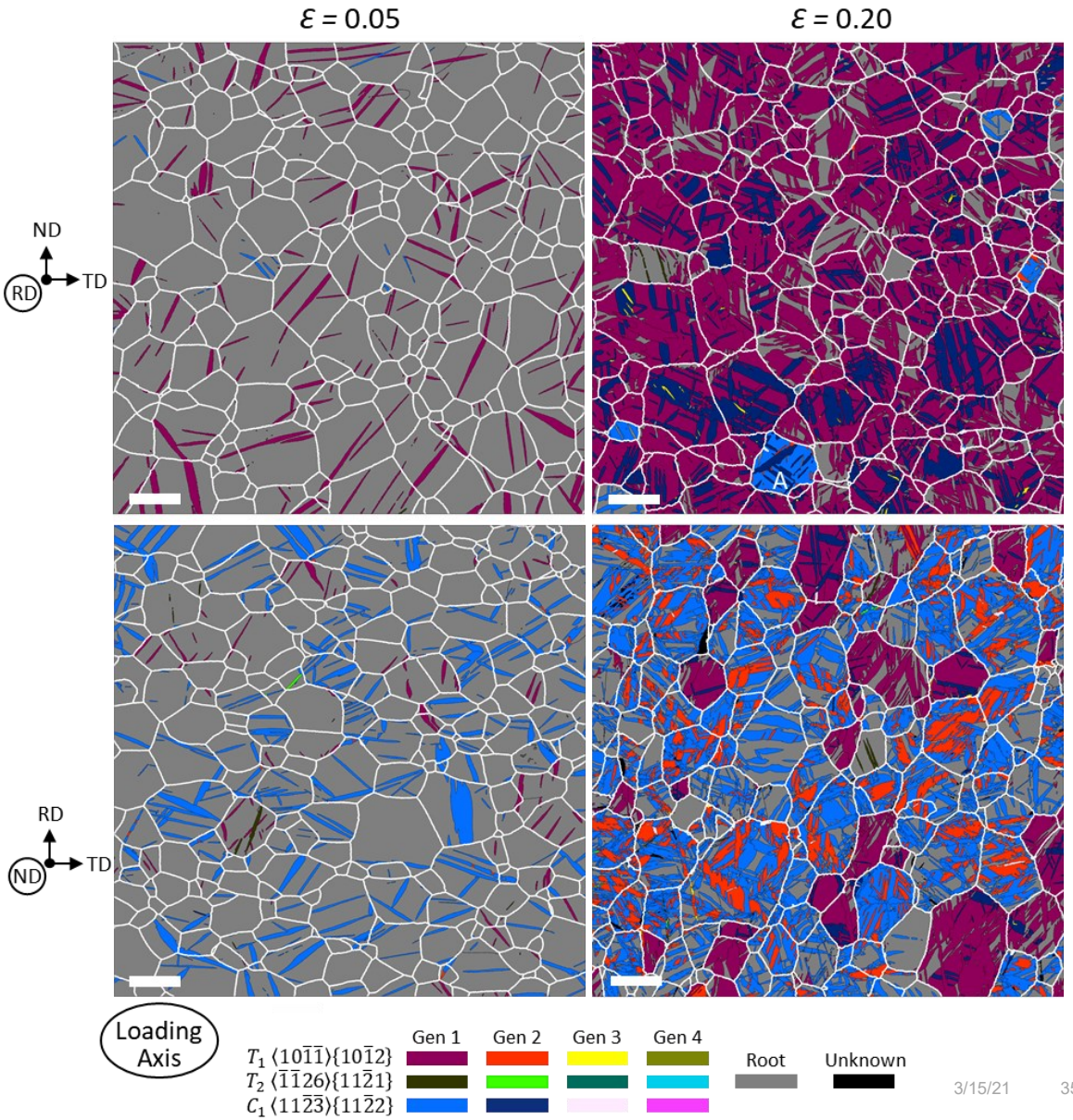
Example family tree and twin classification



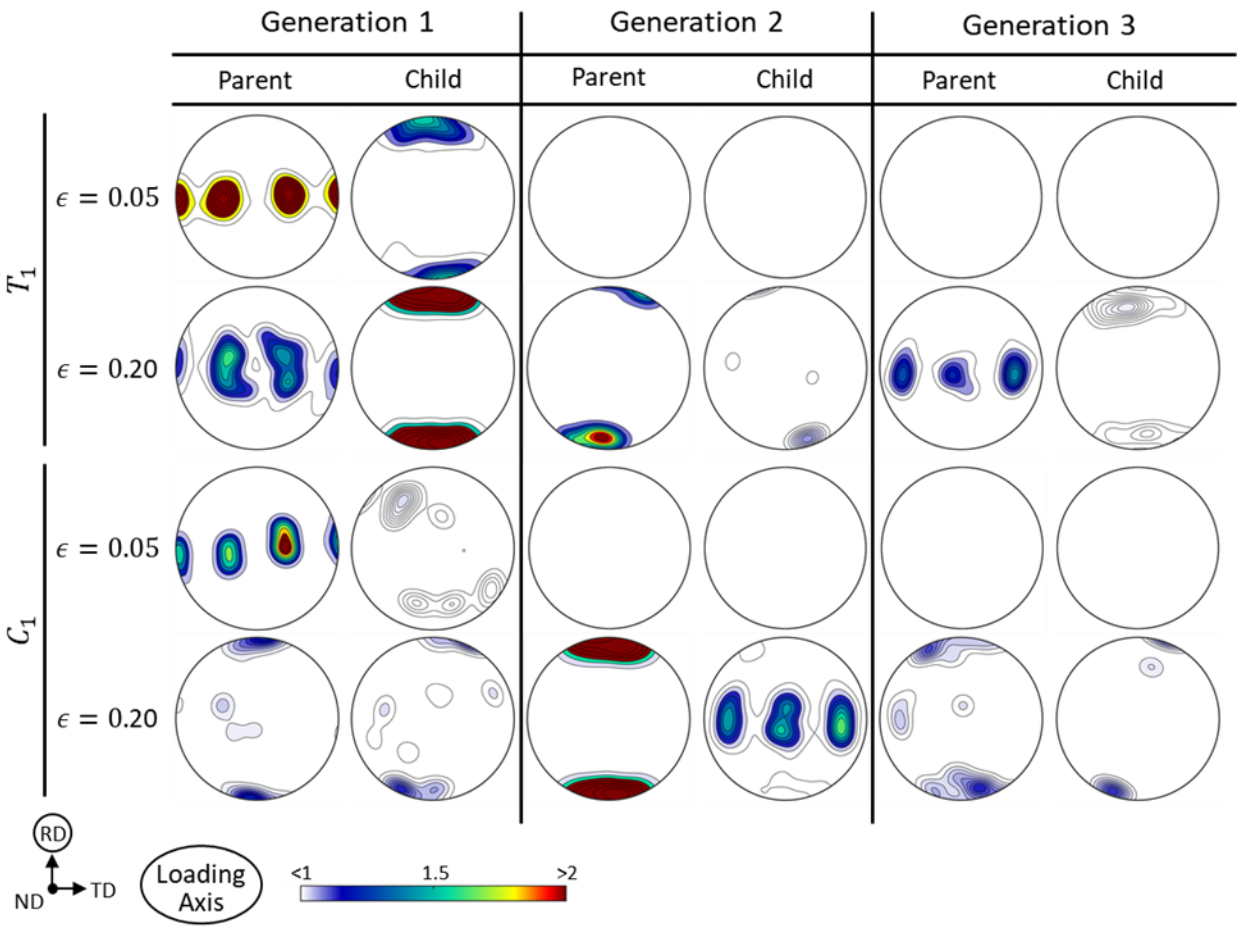
Twin classification applied to the EBSD maps

Insights:

- The twin trends at both strains a similar
- The methodology works for heavily twinned microstructures
- Tertiary twinning is observed in small quantities
- We recover the grain cluster size that is expected

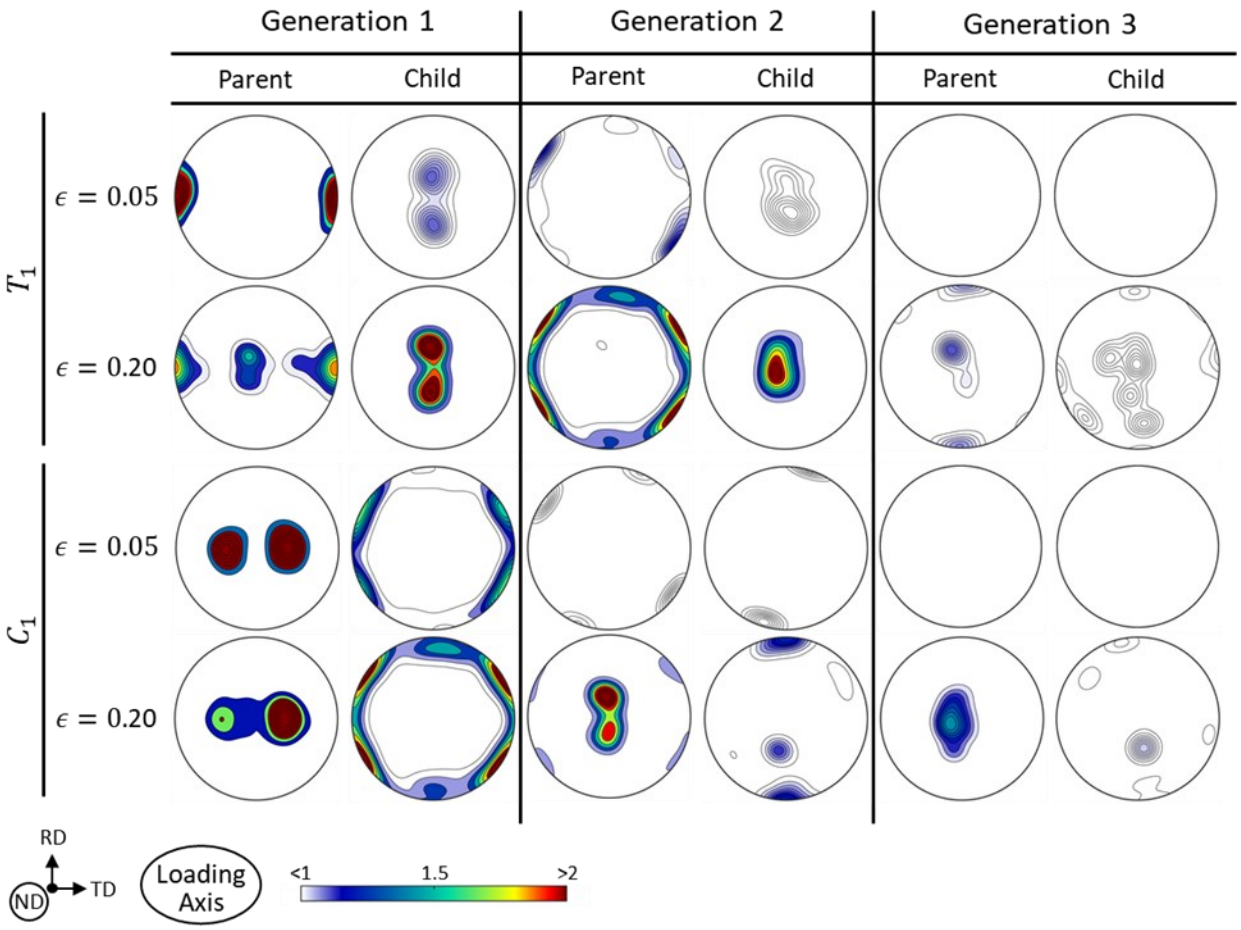


The hierarchical nature of twinning in α-Ti



$$ODF_{gen,type} = \frac{\sum A_{gen,type}}{\sum A} componentODF(g_{gen,type}, A_{gen,type}) + \left(1 - \frac{\sum A_{gen,type}}{\sum A}\right) uniformODF$$

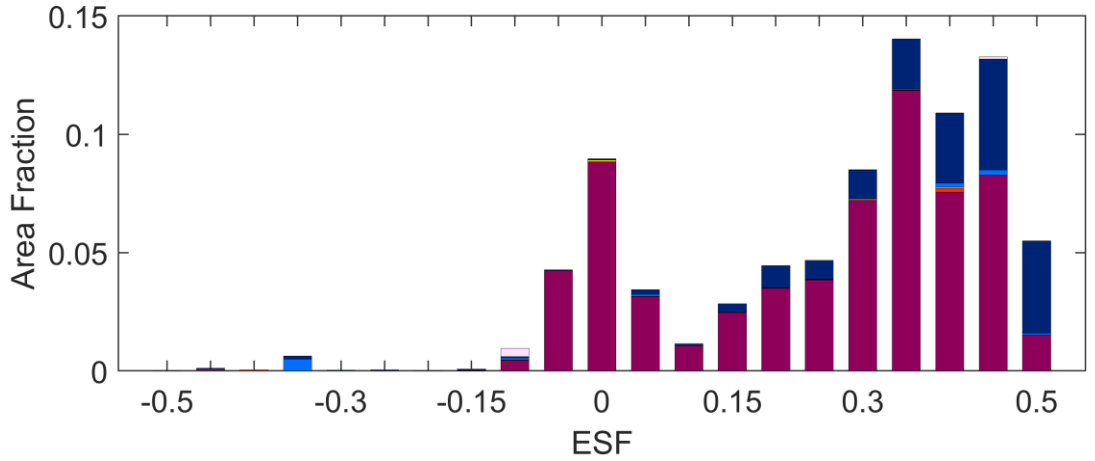
The hierarchical nature of twinning in α-Ti



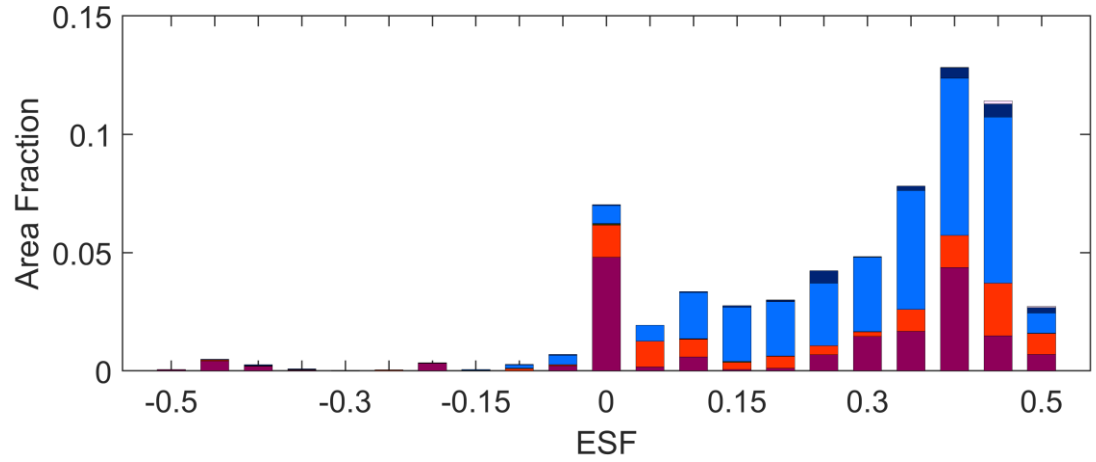
$$ODF_{gen,type} = \frac{\sum A_{gen,type}}{\sum A} componentODF(g_{gen,type}, A_{gen,type}) + \left(1 - \frac{\sum A_{gen,type}}{\sum A}\right) uniformODF$$

The driving force on twins by twin type and generation

RD

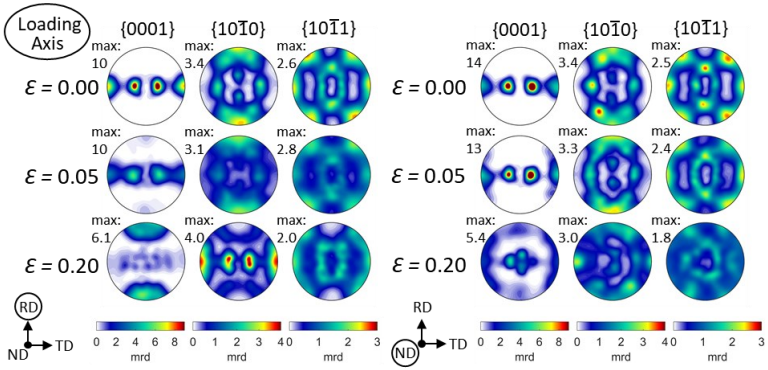


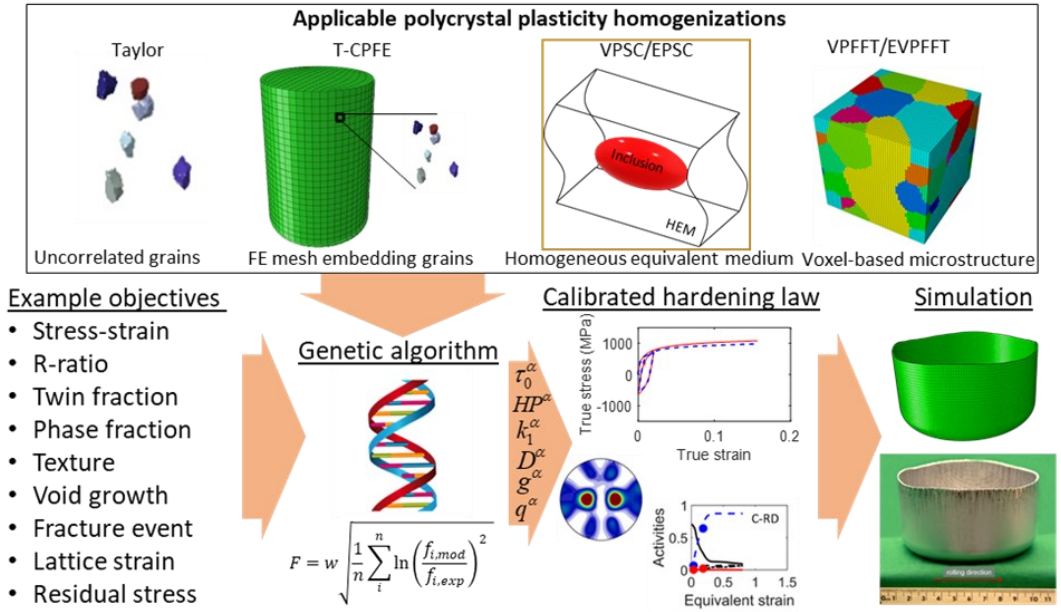
ND



Cross-validation with crystal plasticity modeling

- Textures from EBSD
- Stress-strain



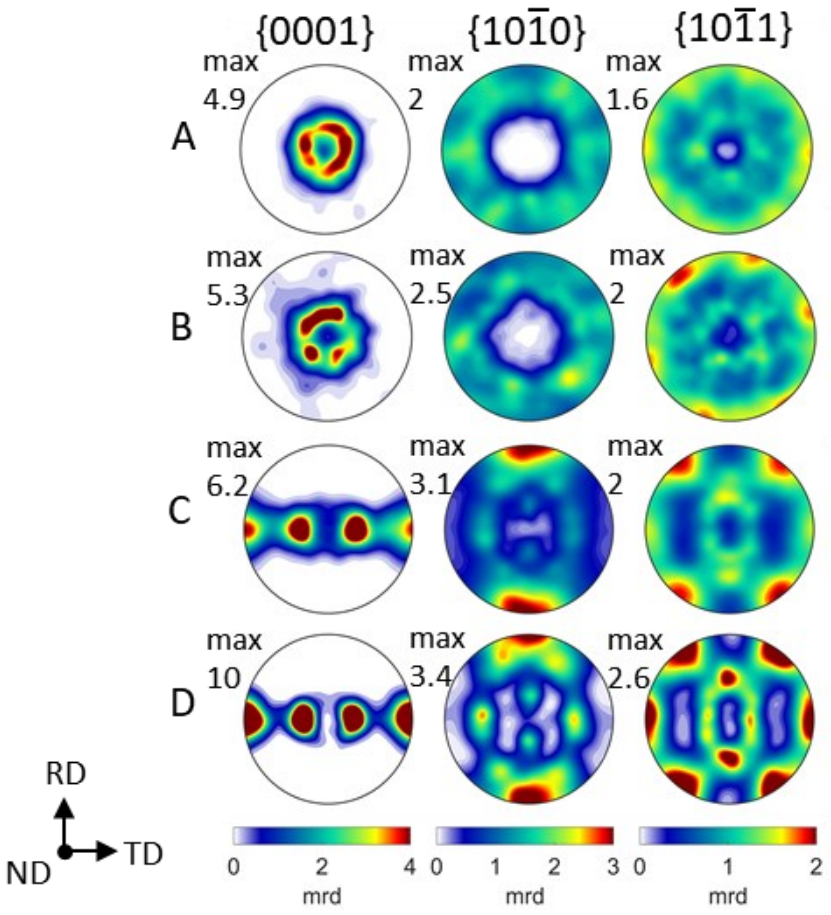


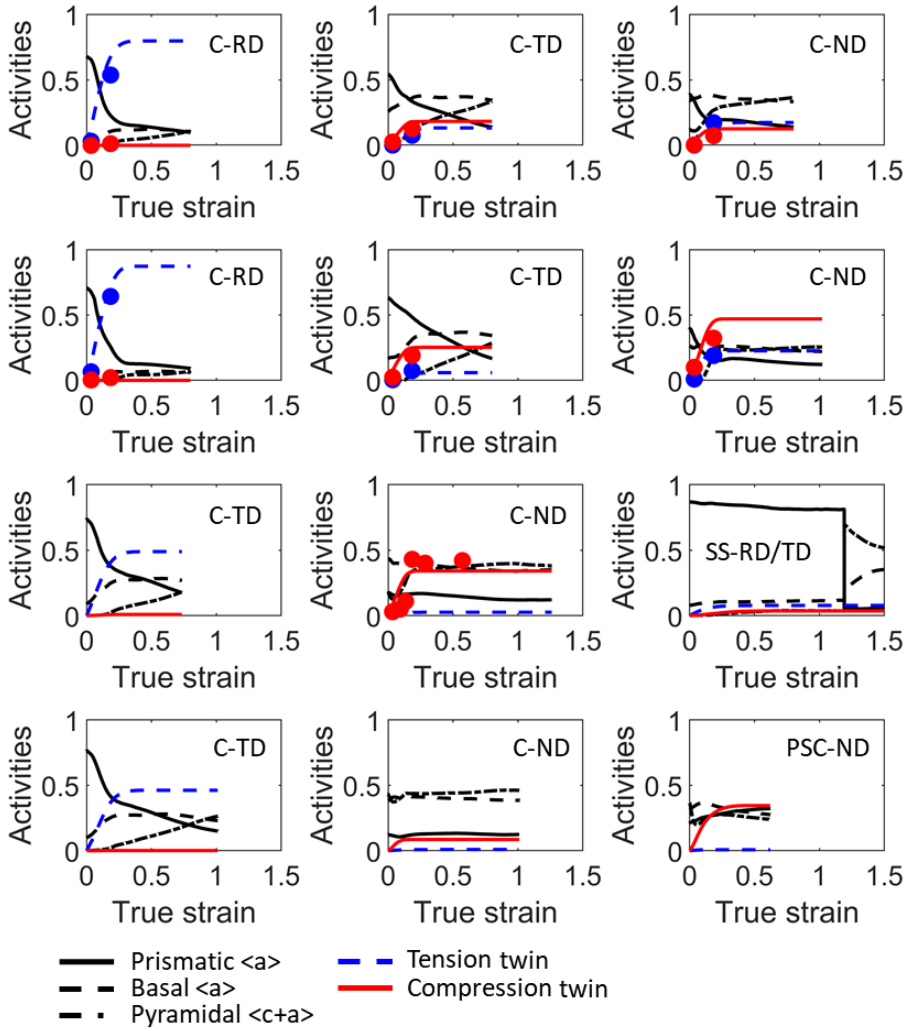
Savage et al. CMAME (in press)

Summary of Experiments

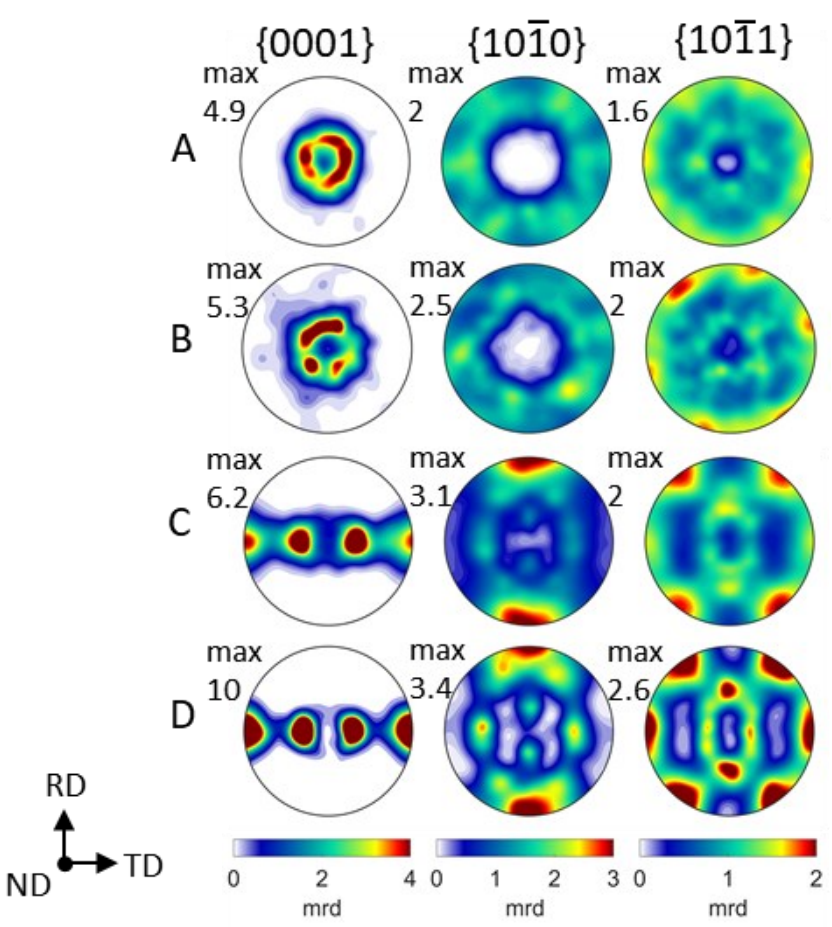
Dataset	Purity	Grain Size [µm]	Initial Texture	Loading	Deformed Texture ε	Twin Fractions ε
A (Fromm, 2009) (Savage, 2020)	99.998% I_{tot}=113 [PPM] H=0[PPM]	17.1 (area weight EBSD)	EBSD	C-TD C-ND PSC-ND	Neutron Diffraction 1.04 1.02 0.63	Not available
B (Salem et al., 2005)	99.998% I_{tot}=113 [PPM] H=0[PPM]	27 (area weighted EBSD)	EBSD	C-TD C-ND SS-RD/TD	Available in (Salem et al., 2005)	- [0 to 0.6] -
C (Savage, 2020)	99.9993% I_{tot}=241.5 [PPM] H=1[PPM]	25.5 (area weight EBSD)	EBSD	C-ND C-RD C-TD	EBSD [0,0.05, 0.2] [0,0.05, 0.2] [0,0.05, 0.2]	Quantitative reconstruction [0,0.05, 0.2] [0,0.05, 0.2] [0,0.05, 0.2]
D (Savage, 2020)	99.9993% I_{tot}=241.5 [PPM] H=1[PPM]	150.4 (area weight EBSD)	EBSD	C-ND C-RD C-TD	EBSD [0,0.05, 0.2] [0,0.05, 0.2] [0,0.05, 0.2]	Quantitative reconstruction [0,0.05, 0.2] [0,0.05, 0.2] [0,0.05, 0.2]

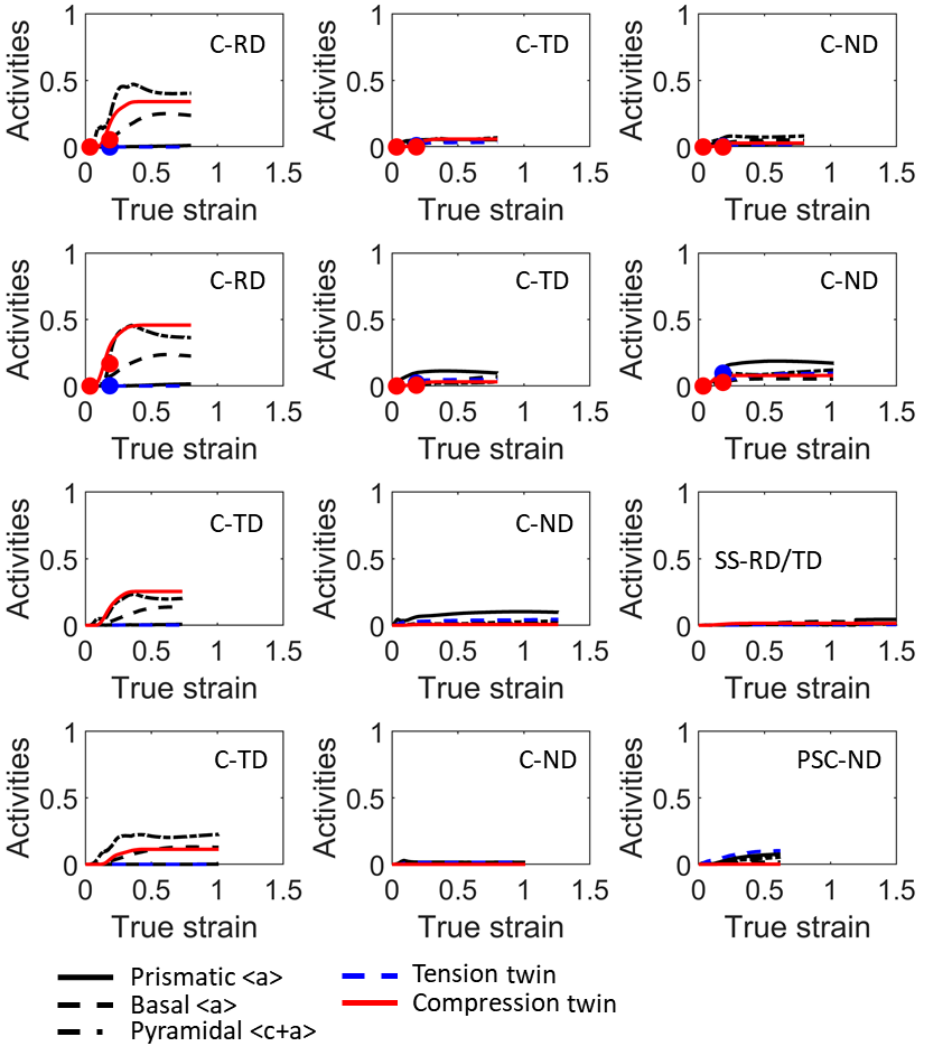
Texture calibrated twin activities compared to experiments for first order twins





Texture calibrated twin activities compared to experiments for second order twins





Conclusions

- A general twin code implementation is presented leveraging the MATLAB graph toolbox and MTEX toolbox.
- Algorithms for grain fragment grouping and segmentation are presented and their parameters discussed.
- A general procedure for twin family tree determination is presented that automatically addresses complex twin relationships that arise in heavily twinned microstructures and its utility is demonstrated on a demanding α-Ti dataset.
- The results are a compelling example of the utility of the code for studying twinning in metals.
- The approach reveals that α-Ti will continue to at least third generation twinning and that higher order twins occur independent of the starting twinning sequence (i.e. $T_1 \rightarrow C_1$ versus $C_1 \rightarrow T_1$).

Acknowledgements

- This research was sponsored by the U.S. National Science Foundation and was accomplished under the CAREER grant no. CMMI-1650641.
- DJS gratefully acknowledges partial support through the University of New Hampshire Dissertation Year Fellowship and the Seaborg Institute under a post-doctoral fellowship.
- RJM acknowledges support through the Office of Basic Energy Sciences, Project FWP 06SCPE401, under US DOE contract no. W-7405-ENG-36.