American Mineralogist: May 2020 Deposit AM-20-57007

CONE ET AL.: TEXTURE AND METHODOLOGY AFFECTS CRYSTAL SIZE DISTRIBUTIONS

Supplemental File 1

MTEX SCRIPT

close all

clear all

%% Import Script for EBSD Data

%

% This script was automatically created by the import wizard. You should

% run the whoole script or parts of it in order to import your data. There

% is no problem in making any changes to this script.

% Specify Crystal and Specimen Symmetries

% crystal symmetry

CS = {...

'notIndexed',...

crystalSymmetry('-1', [8.1797 12.8748 14.1721], [93.13,115.89,91.24]\*degree, 'X||a\*', 'Z||c', 'mineral', 'Anorthite', 'color', 'blue')};

% plotting convention

setMTEXpref('xAxisDirection','north');

setMTEXpref('zAxisDirection','outOfPlane');

% Specify File Names

% path to files THIS WILL CHANGE FROM COMPUTER TO COMPUTER

pname = 'C:\Users\Julianskye\Documents\MATLAB\mtex-4.5.1\Samples\Kim A. Cone';

% which files to be imported

fname = [pname '\STITCHED\_LE\_RI\_5570\_NR.cpr'];

% Import the Data

% create an EBSD variable containing the data

ebsd = loadEBSD(fname,CS,'interface','crc',...

'convertEuler2SpatialReferenceFrame');

diary('STITCHED\_LE\_RI\_5570\_NR raw EBSD map properties.txt')

ebsd

diary off

save('STITCHED\_LE\_RI\_5570\_NR\_ebsd','ebsd');

fig\_resolution = [0 0 1200 700];

%% Calculating Plagioclase boundaries (1 degrees)

load 'STITCHED\_LE\_RI\_5570\_NR\_ebsd.mat'

fig\_resolution = [0 0 1200 700];

% perform grain reconstruction, the '1\*degree' terms in this section are

% what determines the threshold for a high-angle grain boundary

[grains\_pl, ebsd.grainId] = calcGrains(ebsd,'angle',1\*degree,'boundary','tight');

%then reassign the unindexed areas to not be filled in

notIndexed = grains\_pl('notIndexed');

% the "not indexed grains" we want to remove

toRemove = notIndexed(notIndexed.grainSize ./ notIndexed.boundarySize<0.8);

% now we remove the corresponding EBSD measurements

ebsd(toRemove) = [];

% remove all grains with less than 3 measurements

large\_grains = grains\_pl(grains\_pl.grainSize >= 3);

%// perform new grain reconstruction

ebsd\_cleaned\_pl = ebsd(large\_grains);

[grains\_cleaned, ebsd\_cleaned\_pl.grainId, ebsd\_cleaned\_pl.mis2mean] = calcGrains(ebsd\_cleaned\_pl,'angle',1\*degree,'boundary','tight');

% And now we can get rid of some of that nasty staircasing effect on the grain boundaries

grains\_smoothed\_pl = smooth(grains\_cleaned);

% Fill the zero solutions using the calculated grains

ebsd\_filled\_pl = fill(ebsd\_cleaned\_pl);

% Some handy analyses that show up in the command window and are saved to a

% text file (if the text file already exists, the new stuff is appended

diary('STITCHED\_LE\_RI\_5570\_NR EBSD map properties\_pl.txt')

'NEW ENTRY NEW ENTRY NEW ENTRY NEW ENTRY NEW ENTRY NEW ENTRY NEW ENTRY NEW ENTRY NEW ENTRY NEW ENTRY'

clock

% Basic info on phases (modal %, symmetries, # of orientations per phase)

ebsd

ebsd\_cleaned\_pl

ebsd\_filled\_pl

% Info on reconstructed grains

grains\_pl

grains\_smoothed\_pl

diary off

% save the variables

save('STITCHED\_LE\_RI\_5570\_NR\_ebsd\_cleaned\_pl','ebsd\_cleaned\_pl');

save('STITCHED\_LE\_RI\_5570\_NR\_ebsd\_filled\_pl','ebsd\_filled\_pl');

save('STITCHED\_LE\_RI\_5570\_NR\_grains\_pl','grains\_pl');

save('STITCHED\_LE\_RI\_5570\_NR\_grains\_smoothed\_pl','grains\_smoothed\_pl');

%}

% General reference map plots

figure

[~,mP] = plot(ebsd);

mP.micronBar.visible = 'off'

legend off;

set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR ebsd.png' -transparent

figure

[~,mP] = plot(ebsd\_filled\_pl);

mP.micronBar.visible = 'off'

legend off;

set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR ebsd\_filled.png' -transparent

legend on

legend('Location','eastoutside')

legend('boxoff')

export\_fig 'STITCHED\_LE\_RI\_5570\_NR ebsd legend.png' -native

figure

[~,mP] = plot(grains\_smoothed\_pl);

mP.micronBar.visible = 'off'

legend('hide')

set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR grains\_smoothed\_pl5.png' -transparent

%% Starting by loading variables instead of re-importing

% if you've already run the above two sections and don't want to spend time re-importing and re-calculating the grains, you can

% comment out those two sections and start here by loading the variables.

% plotting convention

setMTEXpref('xAxisDirection','north');

setMTEXpref('zAxisDirection','outOfPlane');

setMTEXpref('defaultColorMap',hot)

%{

load('STITCHED\_LE\_RI\_5570\_NR\_ebsd\_cleaned\_pl','ebsd\_cleaned\_pl');

load('STITCHED\_LE\_RI\_5570\_NR\_ebsd\_filled\_pl','ebsd\_filled\_pl');

load('STITCHED\_LE\_RI\_5570\_NR\_grains\_smoothed\_pl','grains\_smoothed\_pl');

%}

fig\_resolution = [0 0 1200 700];

%% Grain boundary misorientations

load 'STITCHED\_LE\_RI\_5570\_NR\_grains\_smoothed\_pl.mat'

% Use boundary('Plagioclase','Plagioclase') to chose only Plagioclase-Plagioclase boundaries

gB = grains\_smoothed\_pl.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

% specifying the crystal symmmetry (this is also embedded in the 'ebsd'

% variable during import, but it can be helpful to define it as an individual

% variable

CS = crystalSymmetry('-1', [8.1797 12.8748 14.1721], [93.13,115.89,91.24]\*degree, 'X||a\*', 'Z||c');

% make a quick inverse pole figure to see if there are any super-obvious trends in

% grain boundary misorientations, this is less helpful in triclinic systems

%{

figure

plotAngleDistribution(gB\_PlPl.misorientation)

figure

plotAxisDistribution(gB\_PlPl.misorientation,'contour')

%}

% Twinning definitions

Albite\_twinning = orientation('axis',Miller(0,1,0,CS,'hkl'),'angle',180\*degree,CS,CS);

Carlsbad\_twinning = orientation('axis',Miller(0,0,1,CS,'uvw'),'angle',180\*degree,CS,CS);

AC\_twinning = orientation('axis',Miller(1,0,0,CS,'pole'),'angle',180\*degree,CS,CS);

Pericline\_twinning = orientation('axis',Miller(0,1,0,CS,'uvw'),'angle',180\*degree,CS,CS);

Ala\_twinning = orientation('axis',Miller(1,0,0,CS,'uvw'),'angle',180\*degree,CS,CS);

Manebach\_twinning = orientation('axis',Miller(0,0,1,CS,'pole'),'angle',180\*degree,CS,CS);

Baveno\_twinning = orientation('axis',Miller(0,2,1,CS,'pole'),'angle',180\*degree,CS,CS);

AlbiteAla\_twinning = orientation('axis',Miller(0,0,1,CS,'pole'),'angle',180\*degree,CS,CS);

Prism110\_twinning = orientation('axis',Miller(1,1,0,CS,'pole'),'angle',180\*degree,CS,CS);

Prism1\_10\_twinning = orientation('axis',Miller(1,-1,0,CS,'pole'),'angle',180\*degree,CS,CS);

Prism130\_twinning = orientation('axis',Miller(1,3,0,CS,'pole'),'angle',180\*degree,CS,CS);

Prism1\_30\_twinning = orientation('axis',Miller(1,-3,0,CS,'pole'),'angle',180\*degree,CS,CS);

% restrict to twinnings with threshold 5 degree and have MTEX use the

% definitions to start identifying plag-plag boundaries with those specific

% twinning relationships

isTwinning\_Albite = angle(gB\_PlPl.misorientation,Albite\_twinning) < 5\*degree;

Albite\_twin\_Boundary = gB\_PlPl(isTwinning\_Albite);

isTwinning\_Carlsbad = angle(gB\_PlPl.misorientation,Carlsbad\_twinning) < 5\*degree;

Carlsbad\_twin\_Boundary = gB\_PlPl(isTwinning\_Carlsbad);

isTwinning\_AC = angle(gB\_PlPl.misorientation,AC\_twinning) < 5\*degree;

AC\_twin\_Boundary = gB\_PlPl(isTwinning\_AC);

isTwinning\_Pericline = angle(gB\_PlPl.misorientation,Pericline\_twinning) < 5\*degree;

Pericline\_twin\_Boundary = gB\_PlPl(isTwinning\_Pericline);

isTwinning\_Ala = angle(gB\_PlPl.misorientation,Ala\_twinning) < 5\*degree;

Ala\_twin\_Boundary = gB\_PlPl(isTwinning\_Ala);

isTwinning\_Manebach = angle(gB\_PlPl.misorientation,Manebach\_twinning) < 5\*degree;

Manebach\_twin\_Boundary = gB\_PlPl(isTwinning\_Manebach);

isTwinning\_Baveno = angle(gB\_PlPl.misorientation,Baveno\_twinning) < 5\*degree;

Baveno\_twin\_Boundary = gB\_PlPl(isTwinning\_Baveno);

isTwinning\_AlbiteAla = angle(gB\_PlPl.misorientation,AlbiteAla\_twinning) < 5\*degree;

AlbiteAla\_twin\_Boundary = gB\_PlPl(isTwinning\_AlbiteAla);

isTwinning\_Prism110 = angle(gB\_PlPl.misorientation,Prism110\_twinning) < 5\*degree;

Prism110\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism110);

isTwinning\_Prism1\_10 = angle(gB\_PlPl.misorientation,Prism1\_10\_twinning) < 5\*degree;

Prism1\_10\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism1\_10);

isTwinning\_Prism130 = angle(gB\_PlPl.misorientation,Prism130\_twinning) < 5\*degree;

Prism130\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism130);

isTwinning\_Prism1\_30 = angle(gB\_PlPl.misorientation,Prism1\_30\_twinning) < 5\*degree;

Prism1\_30\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism1\_30);

% Make a quick command-line summary of the twin proportions in the map,

% helpful so that you can only plot the ones that occur in meaninful

% quantities in the next bit

%{

diary('STITCHED\_LE\_RI\_5570\_NR twin boundary proportions.txt')

Albite = 100\*length(Albite\_twin\_Boundary)/length(gB\_PlPl)

Carlsbad = 100\*length(Carlsbad\_twin\_Boundary)/length(gB\_PlPl)

AC = 100\*length(AC\_twin\_Boundary)/length(gB\_PlPl)

Pericline = 100\*length(Pericline\_twin\_Boundary)/length(gB\_PlPl)

Ala = 100\*length(Ala\_twin\_Boundary)/length(gB\_PlPl)

Manebach = 100\*length(Manebach\_twin\_Boundary)/length(gB\_PlPl)

Baveno = 100\*length(Baveno\_twin\_Boundary)/length(gB\_PlPl)

AlbiteAla = 100\*length(AlbiteAla\_twin\_Boundary)/length(gB\_PlPl)

Prism110 = 100\*length(Prism110\_twin\_Boundary)/length(gB\_PlPl)

Prism1\_10 = 100\*length(Prism1\_10\_twin\_Boundary)/length(gB\_PlPl)

Prism130 = 100\*length(Prism130\_twin\_Boundary)/length(gB\_PlPl)

Prism1\_30 = 100\*length(Prism1\_30\_twin\_Boundary)/length(gB\_PlPl)

diary off

%}

% plot the twinning boundaries as overlays onto a plot of the grains

% colorized by mean average orientation

%{

plot(grains\_smoothed\_pl('Anorthite'),grains\_smoothed\_pl('Anorthite').meanOrientation)

%plot(ebsd('indexed'),ebsd('indexed').orientations)

hold on

plot(Albite\_twin\_Boundary,'linecolor','red','linewidth',2,'displayName','Albite twin boundary')

plot(Carlsbad\_twin\_Boundary,'linecolor','green','linewidth',2,'displayName','Carlsbad twin boundary')

plot(AC\_twin\_Boundary,'linecolor','blue','linewidth',2,'displayName','AC twin boundary')

plot(Pericline\_twin\_Boundary,'linecolor','cyan','linewidth',2,'displayName','Pericline twin boundary')

hold off

set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR all major twin boundaries.png' -transparent

%}

%}

%% Create the new grain set by ignorning the identified twin boundaries

load 'STITCHED\_LE\_RI\_5570\_NR\_ebsd\_filled\_pl.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_grains\_smoothed\_pl.mat'

% In this section, you have to merge the twin boundaries one at a time,

% which means you also have to re-calculate which remaining boundaries are

% twin boundaries. It's a bit redundant to the section above, but seems to

% work and doesn't take TOO long.

%

[mergedGrains,parentId] = merge(grains\_smoothed\_pl,Albite\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Carlsbad = angle(gB\_PlPl.misorientation,Carlsbad\_twinning) < 5\*degree;

Carlsbad\_twin\_Boundary = gB\_PlPl(isTwinning\_Carlsbad);

[mergedGrains,parentId] = merge(mergedGrains,Carlsbad\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_AC = angle(gB\_PlPl.misorientation,AC\_twinning) < 5\*degree;

AC\_twin\_Boundary = gB\_PlPl(isTwinning\_AC);

[mergedGrains,parentId] = merge(mergedGrains,AC\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Pericline = angle(gB\_PlPl.misorientation,Pericline\_twinning) < 5\*degree;

Pericline\_twin\_Boundary = gB\_PlPl(isTwinning\_Pericline);

[mergedGrains,parentId] = merge(mergedGrains,Pericline\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Ala = angle(gB\_PlPl.misorientation,Ala\_twinning) < 5\*degree;

Ala\_twin\_Boundary = gB\_PlPl(isTwinning\_Ala);

[mergedGrains,parentId] = merge(mergedGrains,Ala\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Manebach = angle(gB\_PlPl.misorientation,Manebach\_twinning) < 5\*degree;

Manebach\_twin\_Boundary = gB\_PlPl(isTwinning\_Manebach);

[mergedGrains,parentId] = merge(mergedGrains,Manebach\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Baveno = angle(gB\_PlPl.misorientation,Baveno\_twinning) < 5\*degree;

Baveno\_twin\_Boundary = gB\_PlPl(isTwinning\_Baveno);

[mergedGrains,parentId] = merge(mergedGrains,Baveno\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_AlbiteAla = angle(gB\_PlPl.misorientation,AlbiteAla\_twinning) < 5\*degree;

AlbiteAla\_twin\_Boundary = gB\_PlPl(isTwinning\_AlbiteAla);

[mergedGrains,parentId] = merge(mergedGrains,AlbiteAla\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Prism110 = angle(gB\_PlPl.misorientation,Prism110\_twinning) < 5\*degree;

Prism110\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism110);

[mergedGrains,parentId] = merge(mergedGrains,Prism110\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Prism1\_10 = angle(gB\_PlPl.misorientation,Prism1\_10\_twinning) < 5\*degree;

Prism1\_10\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism1\_10);

[mergedGrains,parentId] = merge(mergedGrains,Prism1\_10\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Prism130 = angle(gB\_PlPl.misorientation,Prism130\_twinning) < 5\*degree;

Prism130\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism130);

[mergedGrains,parentId] = merge(mergedGrains,Prism130\_twin\_Boundary,'calcMeanOrientation');

gB = mergedGrains.boundary.reorder;

gB\_PlPl = gB('Anorthite','Anorthite');

isTwinning\_Prism1\_30 = angle(gB\_PlPl.misorientation,Prism1\_30\_twinning) < 5\*degree;

Prism1\_30\_twin\_Boundary = gB\_PlPl(isTwinning\_Prism1\_30);

[mergedGrains,parentId] = merge(mergedGrains,Prism1\_30\_twin\_Boundary,'calcMeanOrientation');

%}

%{

figure

plot(ebsd\_filled\_pl('Anorthite'),ebsd\_filled\_pl('Anorthite').orientations)

hold on

%plot(gB\_MgMg,angle(gB\_MgMg.misorientation,twinning),'linewidth',4)

plot(mergedGrains.boundary,'linecolor','k','linewidth',1,'displayName','merged grains')

hold off

set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR all Twins Merged.png' -transparent

%}

save('STITCHED\_LE\_RI\_5570\_NR\_merged\_grains','mergedGrains');

%

%% Comparison between pre-merged (all twins their own grain) and merged

% this section generates an orientation distribution function for groups of

% grains and plots them as contoured pole figures

% calculate the orientation distribution function using a full width at

% half-maximum calculated using a Kullback-Leibler optimization scheme

% (more mathematically sound that just using '10 degrees' like almost

% everyone else who does EBSD

%{

psi\_beforetwins = calcKernel(grains\_smoothed\_pl('Anorthite').meanOrientation)

psi\_merged = calcKernel(mergedGrains('Anorthite').meanOrientation)

odfbeforetwins = calcODF(grains\_smoothed\_pl('Anorthite').meanOrientation,psi\_beforetwins);

odfmerged = calcODF(mergedGrains('Anorthite').meanOrientation,psi\_merged);

save('STITCHED\_LE\_RI\_5570\_NR\_odf','odfbeforetwins','odfmerged');

% explictly define the crystallographic planes and axes that you're

% interested in seeing plotted

%Anorthite\_beforetwins

cs\_Anorthite\_beforetwins = ebsd\_filled\_pl('Anorthite').CS;

h\_Anorthite\_beforetwins=[Miller(0,0,1,cs\_Anorthite\_beforetwins), Miller(1,0,0,cs\_Anorthite\_beforetwins),Miller(1,1,1,cs\_Anorthite\_beforetwins),Miller(1,1,0,cs\_Anorthite\_beforetwins),Miller(0,1,0,cs\_Anorthite\_beforetwins)];

o\_Anorthite\_beforetwins = grains\_smoothed\_pl('Anorthite').meanOrientation;

h\_axes\_Anorthite\_beforetwins =[Miller(1,0,0,cs\_Anorthite\_beforetwins,'uvw'), Miller(0,1,0,cs\_Anorthite\_beforetwins,'uvw'),Miller(0,0,1,cs\_Anorthite\_beforetwins,'uvw'),Miller(1,0,1,cs\_Anorthite\_beforetwins,'uvw'),Miller(1,1,0,cs\_Anorthite\_beforetwins,'uvw')];

o\_axes\_Anorthite\_beforetwins = grains\_smoothed\_pl('Anorthite').meanOrientation;

%Anorthite\_merged

cs\_Anorthite\_merged = ebsd\_merged('Anorthite').CS;

h\_Anorthite\_merged=[Miller(0,0,1,cs\_Anorthite\_merged), Miller(1,0,0,cs\_Anorthite\_merged),Miller(1,1,1,cs\_Anorthite\_merged),Miller(1,1,0,cs\_Anorthite\_merged),Miller(0,1,0,cs\_Anorthite\_merged)];

o\_Anorthite\_merged = mergedGrains('Anorthite').meanOrientation;

h\_axes\_Anorthite\_merged =[Miller(1,0,0,cs\_Anorthite\_merged,'uvw'), Miller(0,1,0,cs\_Anorthite\_merged,'uvw'),Miller(0,0,1,cs\_Anorthite\_merged,'uvw'),Miller(1,0,1,cs\_Anorthite\_merged,'uvw'),Miller(1,1,0,cs\_Anorthite\_merged,'uvw')];

o\_axes\_Anorthite\_merged = mergedGrains('Anorthite').meanOrientation;

% Plotting

figure

plotPDF(o\_Anorthite\_beforetwins,h\_Anorthite\_beforetwins,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR Anorthite\_beforetwins planes.png' -m1 -transparent

close(figure)

figure

plotPDF(o\_axes\_Anorthite\_beforetwins,h\_axes\_Anorthite\_beforetwins,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR Anorthite\_beforetwins axes.png' -m1 -transparent

close(figure)

figure

plotIPDF(odfbeforetwins,[xvector,yvector,zvector],'antipodal','resolution',1\*degree,'contourf','colorrange','equal');

mtexColorbar('location','eastOutSide','title','mud');

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR odfAnorthite\_beforetwins IPF.png' -m1 -transparent

close(figure)

% Plotting

figure

plotPDF(o\_Anorthite\_merged,h\_Anorthite\_merged,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig '1STITCHED\_LE\_RI\_5570\_NR Anorthite\_merged planes.png' -m1 -transparent

close(figure)

figure

plotPDF(o\_axes\_Anorthite\_merged,h\_axes\_Anorthite\_merged,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR Anorthite\_merged axes.png' -m1 -transparent

close(figure)

figure

plotIPDF(odfmerged,[xvector,yvector,zvector],'antipodal','resolution',1\*degree,'contourf','colorrange','equal');

mtexColorbar('location','eastOutSide','title','mud');

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR odfAnorthite\_merged IPF.png' -m1 -transparent

close(figure)

%}

%% Groundmass differentiation by size

% this section generates an orientation distribution function for groups of

% grains and plots them as contoured pole figures

load 'STITCHED\_LE\_RI\_5570\_NR\_merged\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_grains\_pl.mat'

% quick log of how many grains are in the 'porphyroclast' and 'groundmass'

% groups for both the pre-merged and merged grain sets

diary('STITCHED\_LE\_RI\_5570\_NR\_grains\_pl sizes.txt')

groundmass\_condition\_notwins = grains\_pl.diameter>=150;

porphyroclast\_condition\_notwins= grains\_pl.diameter<150;

groundmass\_grains\_notwins = grains\_pl(groundmass\_condition\_notwins)

porphyroclast\_grains\_notwins = grains\_pl(porphyroclast\_condition\_notwins)

diary off

diary('STITCHED\_LE\_RI\_5570\_NR\_merged\_grains sizes.txt')

groundmass\_condition = mergedGrains.diameter<=150;

porphyroclast\_condition= mergedGrains.diameter>150;

groundmass\_grains = mergedGrains(groundmass\_condition)

porphyroclast\_grains = mergedGrains(porphyroclast\_condition)

diary off

%}

% generate the ODF using the same strategy as the above section

psi\_groundmass = calcKernel(groundmass\_grains('Anorthite').meanOrientation)

psi\_porphyroclast = calcKernel(porphyroclast\_grains('Anorthite').meanOrientation)

odfgroundmass = calcODF(groundmass\_grains('Anorthite').meanOrientation,psi\_groundmass);

odfporphyroclast = calcODF(porphyroclast\_grains('Anorthite').meanOrientation,psi\_porphyroclast);

odfgroundmass10 = calcODF(groundmass\_grains('Anorthite').meanOrientation,'halfwidth',10\*degree);

odfporphyroclast10 = calcODF(porphyroclast\_grains('Anorthite').meanOrientation,'halfwidth',10\*degree);

save('STITCHED\_LE\_RI\_5570\_NR\_groundmass\_grains','groundmass\_grains');

save('STITCHED\_LE\_RI\_5570\_NR\_porphyroclast\_grains','porphyroclast\_grains');

save('STITCHED\_LE\_RI\_5570\_NR\_odf','odfgroundmass','odfporphyroclast');

%% Texture indicies and grain info

close all

clear all

load 'STITCHED\_LE\_RI\_5570\_NR\_merged\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_porphyroclast\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_groundmass\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_odf.mat'

%Generate J and M texture indicies for the odfs just created

diary('STITCHED\_LE\_RI\_5570\_NR\_merged\_grains Texture Indicies.txt')

groundmass\_M = calcMIndex(odfgroundmass)

groundmass\_J =textureindex(odfgroundmass)

porphyroclast\_M = calcMIndex(odfporphyroclast)

porphyroclast\_J = textureindex(odfporphyroclast)

diary off

% mean values and standard deviations of physical properties of grains

diary('STITCHED\_LE\_RI\_5570\_NR\_merged\_grains various mean physical measurements.txt')

groundmass\_grains\_mean\_diameter=2\*(mean(equivalentRadius(groundmass\_grains('Anorthite'))))

groundmass\_grains\_mean\_diameter\_std=2\*(std(2\*(equivalentRadius(groundmass\_grains('Anorthite')))))

groundmass\_grains\_mean\_aspectratio=mean(groundmass\_grains('Anorthite').aspectRatio)

groundmass\_grains\_mean\_aspectratio\_std=2\*(std(groundmass\_grains('Anorthite').aspectRatio))

porphyroclast\_grains\_mean\_diameter=2\*(mean(equivalentRadius(porphyroclast\_grains('Anorthite'))))

porphyroclast\_grains\_mean\_diameter\_std=2\*(std(2\*(equivalentRadius(porphyroclast\_grains('Anorthite')))))

porphyroclast\_grains\_mean\_aspectratio=mean(porphyroclast\_grains('Anorthite').aspectRatio)

porphyroclast\_grains\_mean\_aspectratio\_std=2\*(std(porphyroclast\_grains('Anorthite').aspectRatio))

diary off

% Lists of equal area ellipse properties

[omega,a,b]= principalComponents(groundmass\_grains);

%Groundmass grains

diary('STITCHED\_LE\_RI\_5570\_NR groundmass grains ellipse principal components (grain id,long,short).txt')

groundmass\_grains\_equalareaellipse\_principal\_components = [groundmass\_grains.id,a,b]

diary off

diary('STITCHED\_LE\_RI\_5570\_NR groundmass grains aspect ratio.txt')

groundmass\_grains\_AR = aspectRatio(groundmass\_grains)

groundmass\_grains\_aspectratio = [groundmass\_grains.id,groundmass\_grains\_AR]

diary off

diary('STITCHED\_LE\_RI\_5570\_NR groundmass grains longest line inside grain (not ellipse).txt')

groundmass\_grains\_LL = diameter(groundmass\_grains)

groundmass\_grains\_longest\_line = [groundmass\_grains.id,groundmass\_grains\_LL]

diary off

%Porphyroclast grains

[omega,a,b]= principalComponents(porphyroclast\_grains);

diary('STITCHED\_LE\_RI\_5570\_NR porphyroclast grains ellipse principal components (grain id,long,short).txt')

porphyroclast\_grains\_equalareaellipse\_principal\_components = [porphyroclast\_grains.id,a,b]

diary off

diary('STITCHED\_LE\_RI\_5570\_NR porphyroclast grains aspect ratio.txt')

porphyroclast\_grains\_AR = aspectRatio(porphyroclast\_grains)

porphyroclast\_grains\_aspectratio = [porphyroclast\_grains.id,porphyroclast\_grains\_AR]

diary off

diary('STITCHED\_LE\_RI\_5570\_NR porphyroclast grains longest line inside grain (not ellipse).txt')

porphyroclast\_grains\_LL = diameter(porphyroclast\_grains)

porphyroclast\_grains\_longest\_line = [porphyroclast\_grains.id,porphyroclast\_grains\_LL]

diary off

%% Grain ID plotting

load 'STITCHED\_LE\_RI\_5570\_NR\_merged\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_porphyroclast\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_groundmass\_grains.mat'

fig\_resolution = [0 0 1200 650];

figure

plot(groundmass\_grains('Anorthite'),groundmass\_grains('Anorthite').meanOrientation,'micronbar','off')

text(groundmass\_grains('Anorthite'),int2str(groundmass\_grains('Anorthite').id))

%set(gcf,'Position',fig\_resolution)

%export\_fig 'STITCHED\_LE\_RI\_5570\_NR Groundmass Grains Grain IDs.png'

figure

plot(porphyroclast\_grains,porphyroclast\_grains.meanOrientation,'micronbar','off')

text(porphyroclast\_grains,int2str(porphyroclast\_grains.id))

%set(gcf,'Position',fig\_resolution)

%export\_fig 'STITCHED\_LE\_RI\_5570\_NR Porphyroclast Grains Grain IDs.png'

%{

% explictly define the crystallographic planes and axes that you're

% interested in seeing plotted

%

%Anorthite\_groundmass

cs\_Anorthite\_groundmass = ebsd\_filled\_pl('Anorthite').CS;

h\_Anorthite\_groundmass=[Miller(0,0,1,cs\_Anorthite\_groundmass), Miller(1,0,0,cs\_Anorthite\_groundmass),Miller(1,1,1,cs\_Anorthite\_groundmass),Miller(1,1,0,cs\_Anorthite\_groundmass),Miller(0,1,0,cs\_Anorthite\_groundmass)];

o\_Anorthite\_groundmass = groundmass\_grains('Anorthite').meanOrientation;

h\_axes\_Anorthite\_groundmass =[Miller(1,0,0,cs\_Anorthite\_groundmass,'uvw'), Miller(0,1,0,cs\_Anorthite\_groundmass,'uvw'),Miller(0,0,1,cs\_Anorthite\_groundmass,'uvw'),Miller(1,0,1,cs\_Anorthite\_groundmass,'uvw'),Miller(1,1,0,cs\_Anorthite\_groundmass,'uvw')];

o\_axes\_Anorthite\_groundmass = groundmass\_grains('Anorthite').meanOrientation;

%Anorthite\_porphyroclast

cs\_Anorthite\_porphyroclast = ebsd\_filled\_pl('Anorthite').CS;

h\_Anorthite\_porphyroclast=[Miller(0,0,1,cs\_Anorthite\_porphyroclast), Miller(1,0,0,cs\_Anorthite\_porphyroclast),Miller(1,1,1,cs\_Anorthite\_porphyroclast),Miller(1,1,0,cs\_Anorthite\_porphyroclast),Miller(0,1,0,cs\_Anorthite\_porphyroclast)];

o\_Anorthite\_porphyroclast = porphyroclast\_grains('Anorthite').meanOrientation;

h\_axes\_Anorthite\_porphyroclast =[Miller(1,0,0,cs\_Anorthite\_porphyroclast,'uvw'), Miller(0,1,0,cs\_Anorthite\_porphyroclast,'uvw'),Miller(0,0,1,cs\_Anorthite\_porphyroclast,'uvw'),Miller(1,0,1,cs\_Anorthite\_porphyroclast,'uvw'),Miller(1,1,0,cs\_Anorthite\_porphyroclast,'uvw')];

o\_axes\_Anorthite\_porphyroclast = porphyroclast\_grains('Anorthite').meanOrientation;

%{

% Plotting

figure

plotPDF(o\_Anorthite\_groundmass,h\_Anorthite\_groundmass,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR Anorthite\_groundmass planes.png' -m1 -transparent

close(figure)

figure

plotPDF(o\_axes\_Anorthite\_groundmass,h\_axes\_Anorthite\_groundmass,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR Anorthite\_groundmass axes.png' -m1 -transparent

close(figure)

figure

plotIPDF(odfgroundmass,[xvector,yvector,zvector],'antipodal','resolution',1\*degree,'contourf','colorrange','equal');

mtexColorbar('location','eastOutSide','title','mud');

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR odfAnorthite\_groundmass IPF.png' -m1 -transparent

close(figure)

% Plotting

figure

plotPDF(o\_Anorthite\_porphyroclast,h\_Anorthite\_porphyroclast,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig '1STITCHED\_LE\_RI\_5570\_NR Anorthite\_porphyroclast planes.png' -m1 -transparent

close(figure)

figure

plotPDF(o\_axes\_Anorthite\_porphyroclast,h\_axes\_Anorthite\_porphyroclast,'uer','projection','eangle','contourf');

CLim(gcm,'equal');

mtexColorbar

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR Anorthite\_porphyroclast axes.png' -m1 -transparent

close(figure)

figure

plotIPDF(odfporphyroclast,[xvector,yvector,zvector],'antipodal','resolution',1\*degree,'contourf','colorrange','equal');

mtexColorbar('location','eastOutSide','title','mud');

%set(gcf,'Position',fig\_resolution)

export\_fig 'STITCHED\_LE\_RI\_5570\_NR odfAnorthite\_porphyroclast IPF.png' -m1 -transparent

close(figure)

%}

%% Grain C-axis orientations

% this section just produces .txt files reporting the orientations of

% c-axes for each grain in the groundmass and porphyroclast grain sets

% produced earlier, in both a cartesian coordinate system locked to the

% directions of the map XYZ, and in polar coordinates in case one is more

% convenient to use than another. Essentially no new information here -

% all of this is what is visible in the pole figure for the C-axis.

%{

load 'STITCHED\_LE\_RI\_5570\_NR\_merged\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_porphyroclast\_grains.mat'

load 'STITCHED\_LE\_RI\_5570\_NR\_groundmass\_grains.mat'

CS = crystalSymmetry('-1', [8.1797 12.8748 14.1721], [93.13,115.89,91.24]\*degree, 'X||a\*', 'Z||c');

% groundmass

diary('STITCHED\_LE\_RI\_5570\_NR groundmass\_grains.txt')

groundmass\_grains('Anorthite')

diary off

groundmass\_Caxes= groundmass\_grains('Anorthite').meanOrientation\*Miller(0,0,1,CS);

diary('STITCHED\_LE\_RI\_5570\_NR groundmass\_grains [001] Cartesian.txt')

groundmass\_Caxes.xyz

diary off

diary('STITCHED\_LE\_RI\_5570\_NR groundmass\_grains [001] Polar.txt')

[theta,rho,r] = polar(groundmass\_Caxes)

diary off

figure

plot(groundmass\_Caxes,'antipodal','markersize',2)

annotate([vector3d.X,vector3d.Y,vector3d.Z],'label',{'X','Y','Z'},'backgroundcolor','w')

export\_fig 'STITCHED\_LE\_RI\_5570\_NR groundmass\_Caxes.png' -m1 -transparent

% porphyroclasts

diary('STITCHED\_LE\_RI\_5570\_NR porphyroclast\_grains.txt')

porphyroclast\_grains('Anorthite')

diary off

porphyroclast\_Caxes= porphyroclast\_grains('Anorthite').meanOrientation\*Miller(0,0,1,CS);

diary('STITCHED\_LE\_RI\_5570\_NR porphyroclast\_grains [001] Cartesian.txt')

porphyroclast\_Caxes.xyz

diary off

diary('STITCHED\_LE\_RI\_5570\_NR porphyroclast\_grains [001] Polar.txt')

[theta,rho,r] = polar(porphyroclast\_Caxes)

diary off

figure

plot(porphyroclast\_Caxes,'antipodal','markersize',2)

annotate([vector3d.X,vector3d.Y,vector3d.Z],'label',{'X','Y','Z'},'backgroundcolor','w')

export\_fig 'STITCHED\_LE\_RI\_5570\_NR porphyroclast\_Caxes.png' -m1 -transparent

%}

%}