Temperature and crystallization history and a magma reinjection event in the Breidallur volcanic complex

- Connor Griffin -

Temperature and crystallization history and a magma reinjection event in the Breidallur volcanic complex.

Abstract
This investigation used geochemical samples to study magmatic processes in the Breidallur volcanic complex in eastern Iceland. Analysis of thin sections, electron microprobe data and x-ray fluorescence data from 12 rock samples reveal the existence of fractional crystallization, the temperature histories of the samples, the crystallization history of the magma chamber, and the existence of a magma reinjection event.

1. Introduction
Unlike many geological settings, a magma chamber and its processes cannot be studied directly. We must instead study them through proxies, usually geochemical data. I use geochemical data to answer questions about the Breidallur volcanic complex in Iceland, such as: Did fractional crystallization occur? What is the qualitative temperature history of the chamber? Which minerals were crystallised and in what order? Was magma reinjected into the chamber from the mantle?

I analyzed data gathered from 12 rock samples of lava flows in the volcanic complex. The data include electron microprobe transects across phenocrysts and element maps of phenocrysts, x-ray fluorescence bulk composition data, and thin sections of each sample.

2. Methods
I surveyed thin sections of each sample for the presence and mineralogy of large phenocrysts.

X-ray fluorescence (XRF) geochemical methods measure the bulk composition of a sample by measuring the wavelength of fluorescent electrons, and determine the relative abundance of each element by comparing the intensity of each wavelength detected.

I corrected XRF data for components lost on ignition, then took an average of three measurements for each sample. I then plotted weight percent MgO against weight percent Al2O3 to examine trends in fractional crystallization. I added linear regression trendlines to subsets of the data to show possible liquid lines of descent.

Electron probe micro-analysis (EPMA) directs a concentrated beam of electrons at a specific point within a sample in order to measure the composition of that point. It does so by measuring the wavelengths and relative intensities of electrons reflected off the sample.

Students in previous years studied samples 1516_3, 1516_5, 1516_6 and 1516_8 by measuring composition along a transect between the core and rim of several phenocrysts in each sample. I assigned each data point in these transects a decimal value based on its distance between the core and the rim of the phenocryst, assuming that the measured points are equally spaced within each transect. I then plotted decimal distance values against weight percent Na2O. I used linear regression to make trendlines for each transect, and for all the transect data in each sample.

For samples 1516_1 and 1516_11, students analysed the cores and rims of selected phenocrysts, but not transects across the phenocryst. Students also analyzed select points within the groundmass of all six samples.

I plotted the weight percent MgO values against weight percent Na2O for plagioclase phenocrysts and groundmass for all six samples.

Students also used the electron microprobe to create element maps of a selection of phenocrysts in each sample, which I refer to in the discussion.

3. Results

Videos of thin sections revealed several key observations for this study. All samples mostly consist of fine-grained groundmass that includes needle-like plagioclase crystals. A majority of samples also include large phenocrysts of plagioclase, clinopyroxene, and olivine. The olivine is always altered entirely to iddingsite.

I also looked at thin sections slides to roughly compare phenocryst sizes (Fig. 1).

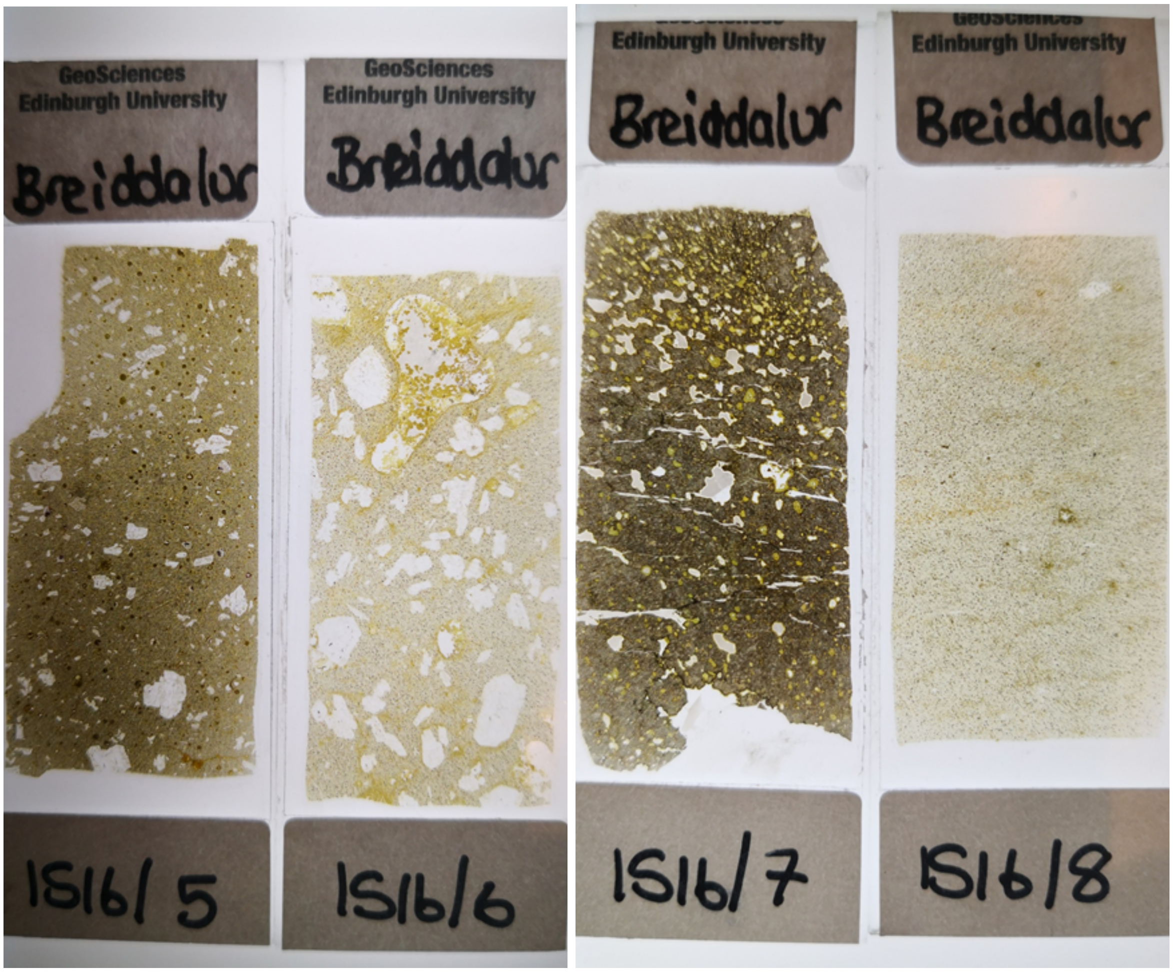


Fig. 1. Thin section slides for samples 1516_5, 1516_6, 1516_7 and 1516_8.

EPMA transects of plagioclase phenocrysts show the weight percent of Na2O in each crystal and its change from core to rim (Fig. 2).

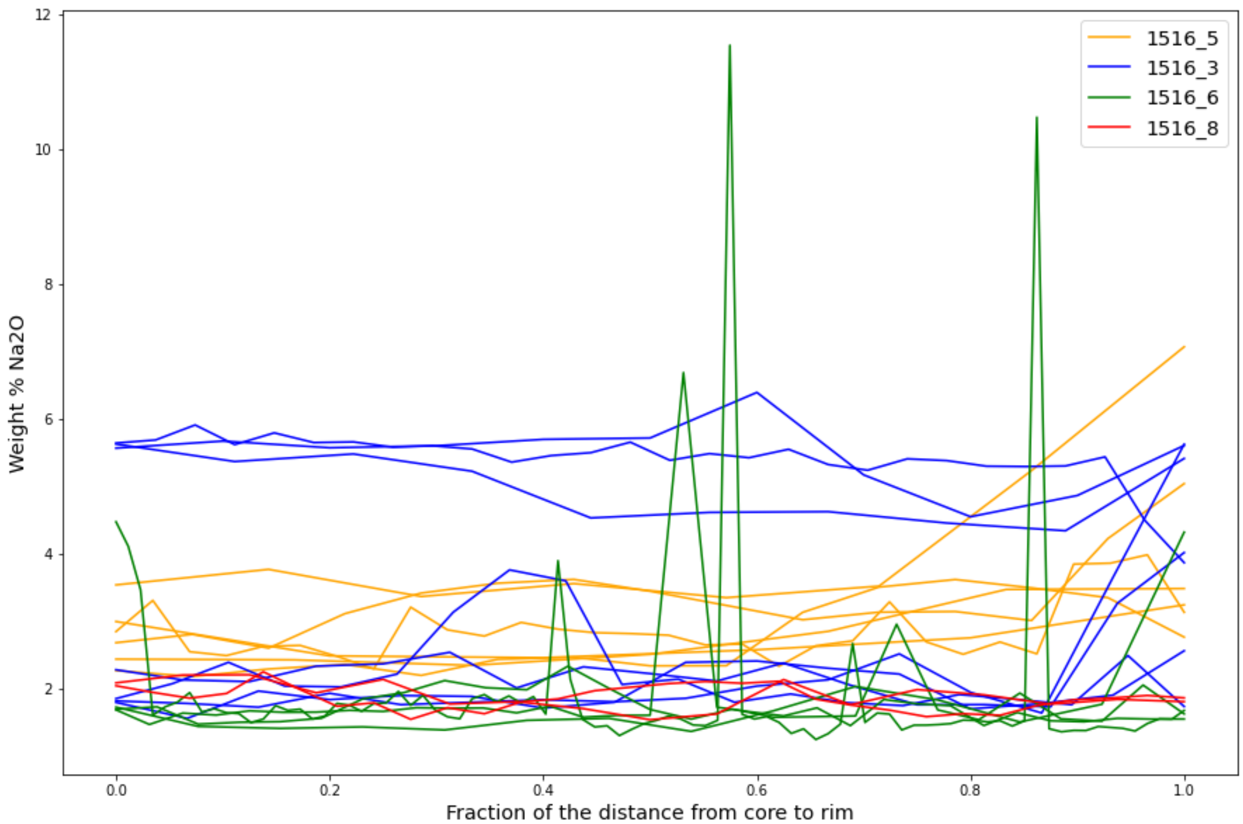


Fig. 2. Raw weight percent Na2O data for each plagioclase phenocryst transect.

A linear regression trendline for each transect shows two distinct clusters for sample 1516_3, generally increasing trends for 1516_5, and flat trends at a relatively low weight percent Na2O for 1516_6 and 1516_8 (Fig. 3).

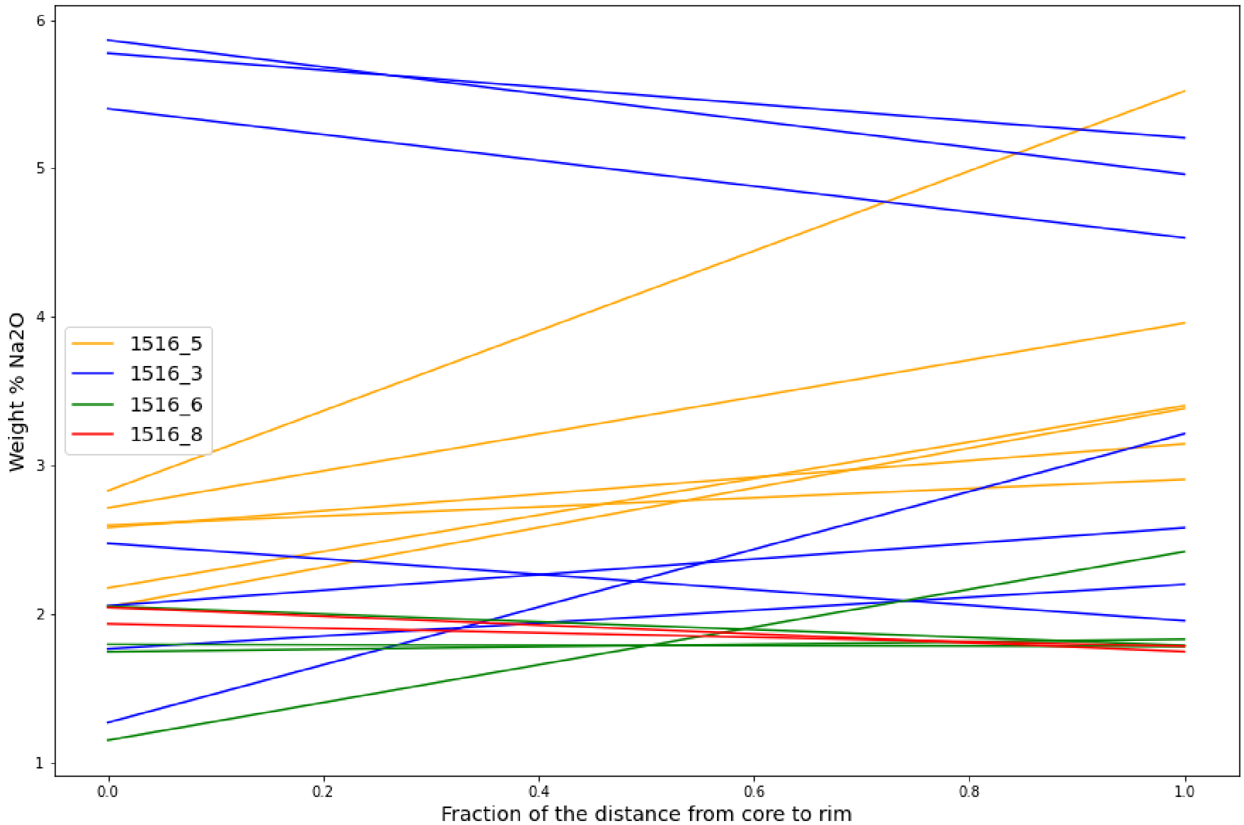


Fig. 3. A linear regression trendline for each transect.

A single trendline for each sample shows the same observations, but obscures the distinct clusters in 1516_3 (Fig. 4).

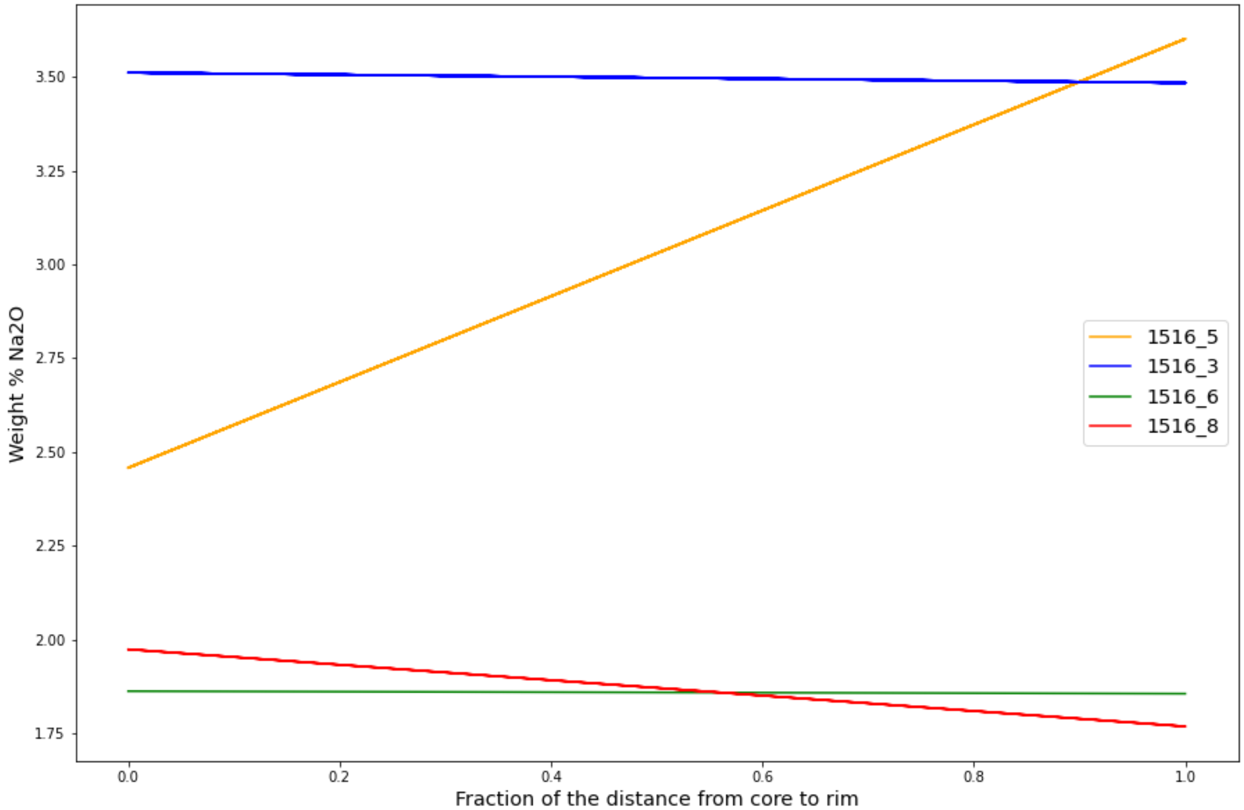


Fig. 4. A trendline for all the transect data from each sample.

Plotting MgO concentrations against Na2O for phenocrysts from all samples analysed with the electron microprobe shows the same low Na2O concentrations for samples 1516_6 and 1516_8, higher concentrations for 1516_5, and two separate clusters for 1516_3 (Fig. 5). There are too few data points to pick out noticeable trends for samples 1516_1 and 1516_11.

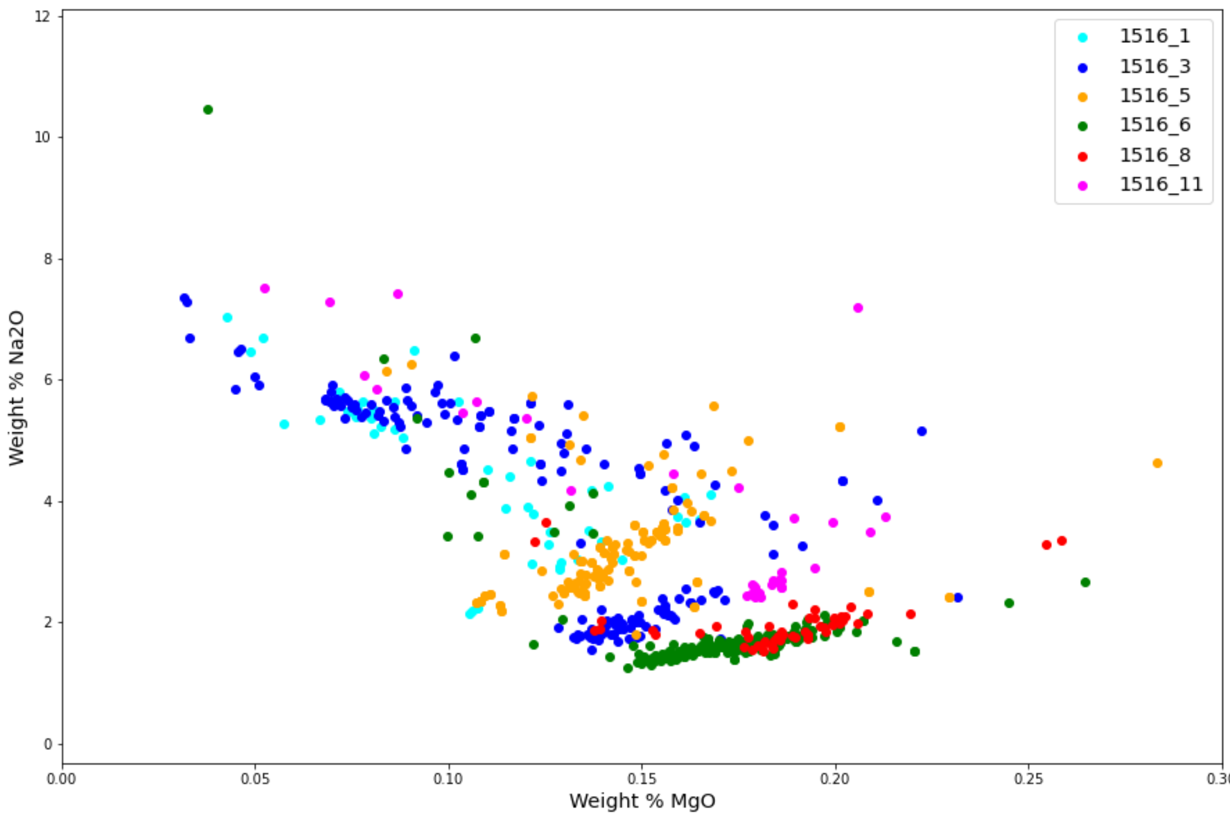


Fig. 5. A scatter plot of weight % MgO vs weight % Na2O for each plagioclase phenocryst in all six samples analysed with the electron microprobe.

Fig. 6 shows the same plot for plagioclase in the groundmass.

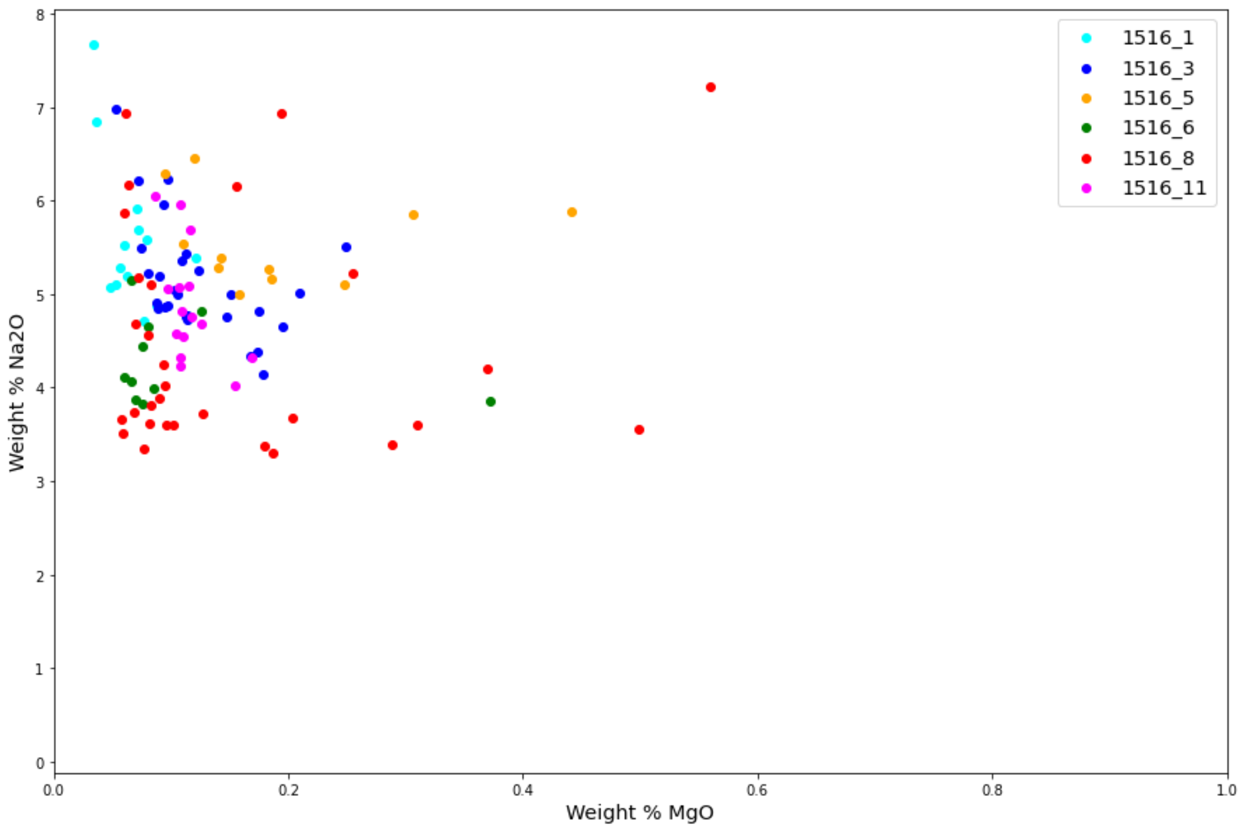


Fig. 6. A scatter plot of weight % MgO vs weight % Na2O for plagioclase in the groundmass, for all six samples analysed with the electron microprobe.

Fig. 7 shows average bulk MgO and Al2O3 concentrations as obtained by XRF for all 12 samples.

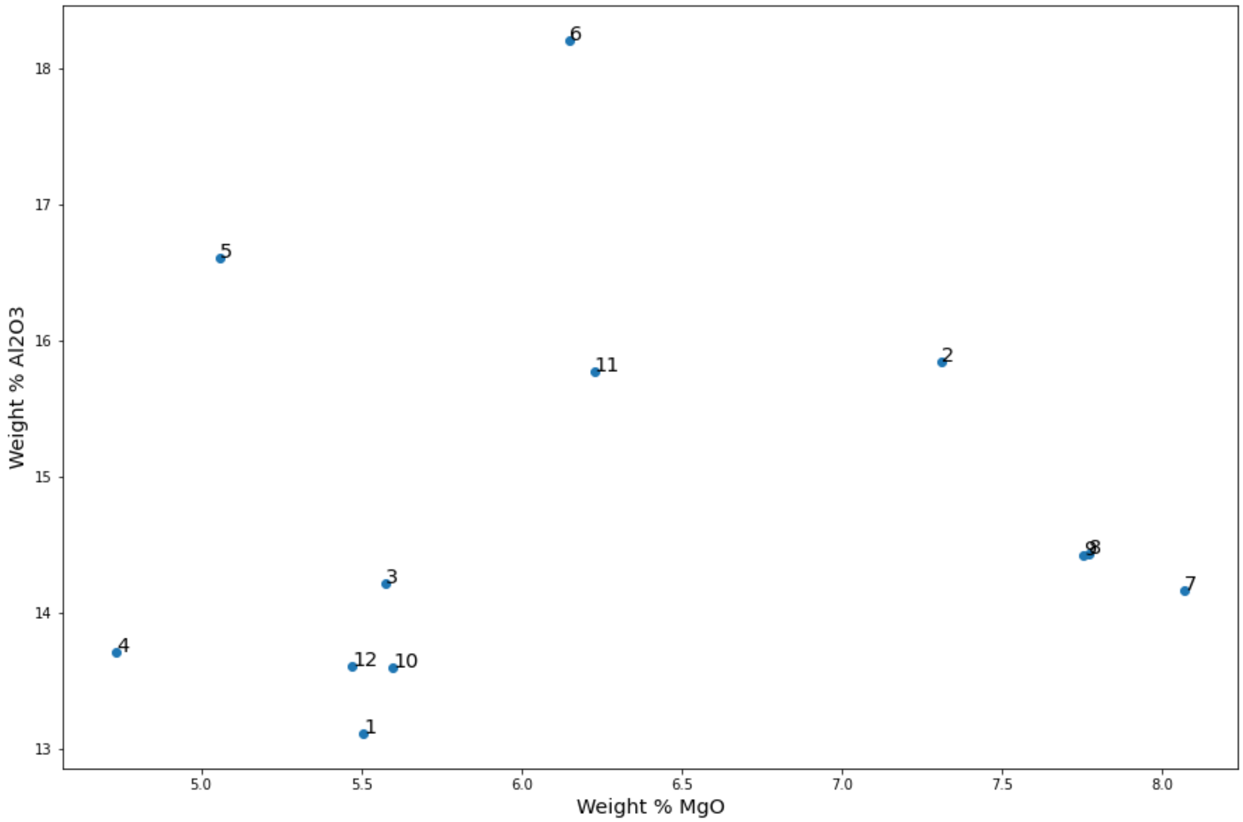
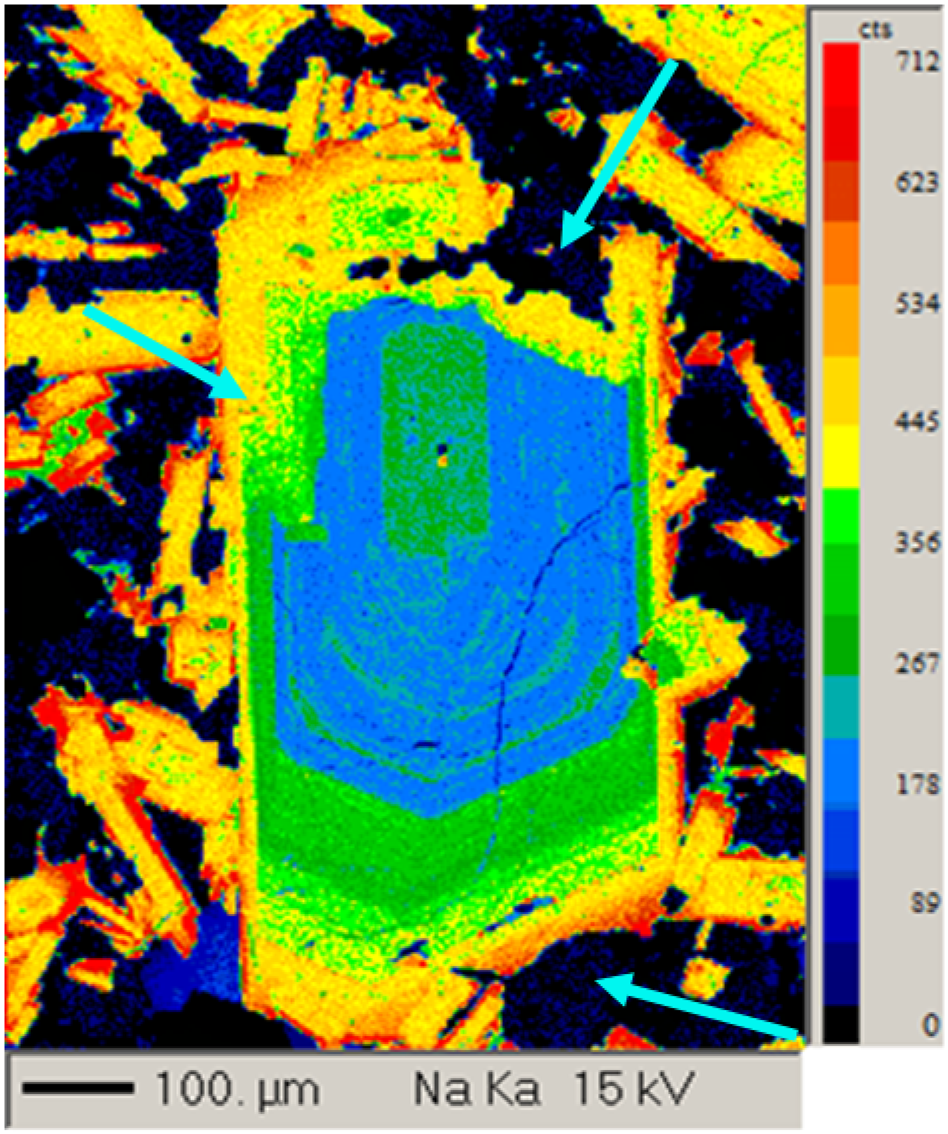


Fig. 7. A scatter plot of bulk weight % MgO and weight % Al2O3 for all 12 samples.

Element maps of samples 1516_1 (Fig. 8), 1516_3 (Fig. 9), and 1516_11 (Fig. 10) show possible evidence of resorption, indicated by arrows.


cts
712
623
534
445
356
267
178
89
0
100. µm
Na Ka 15 kV

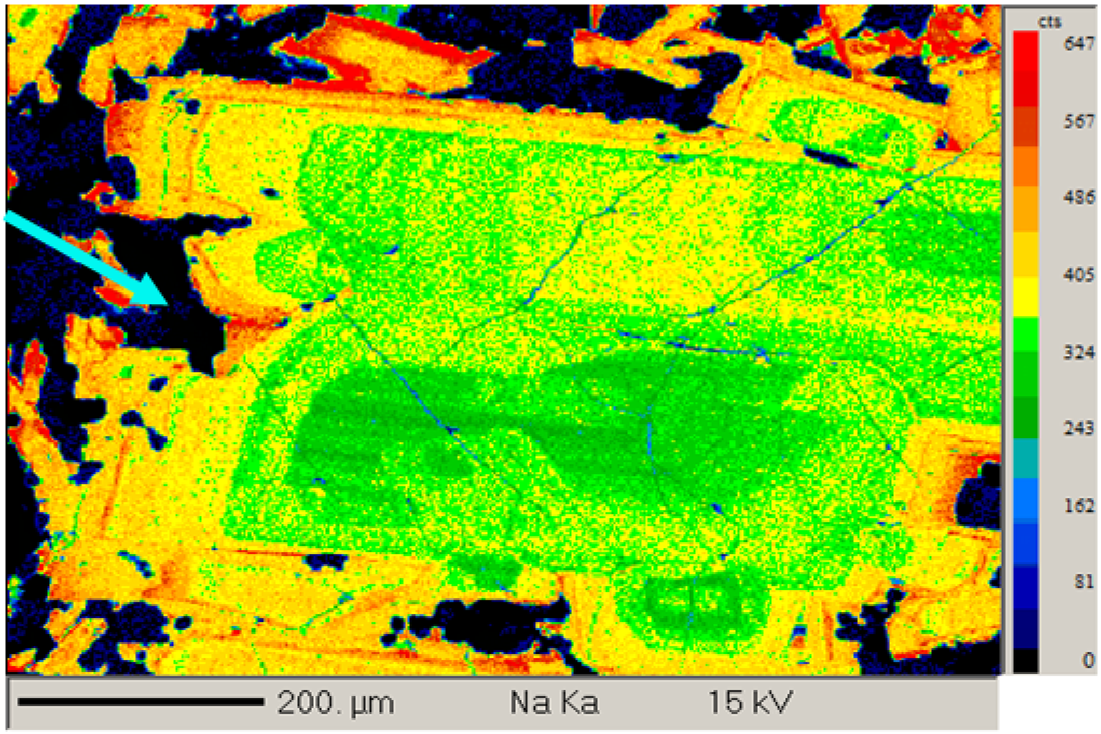
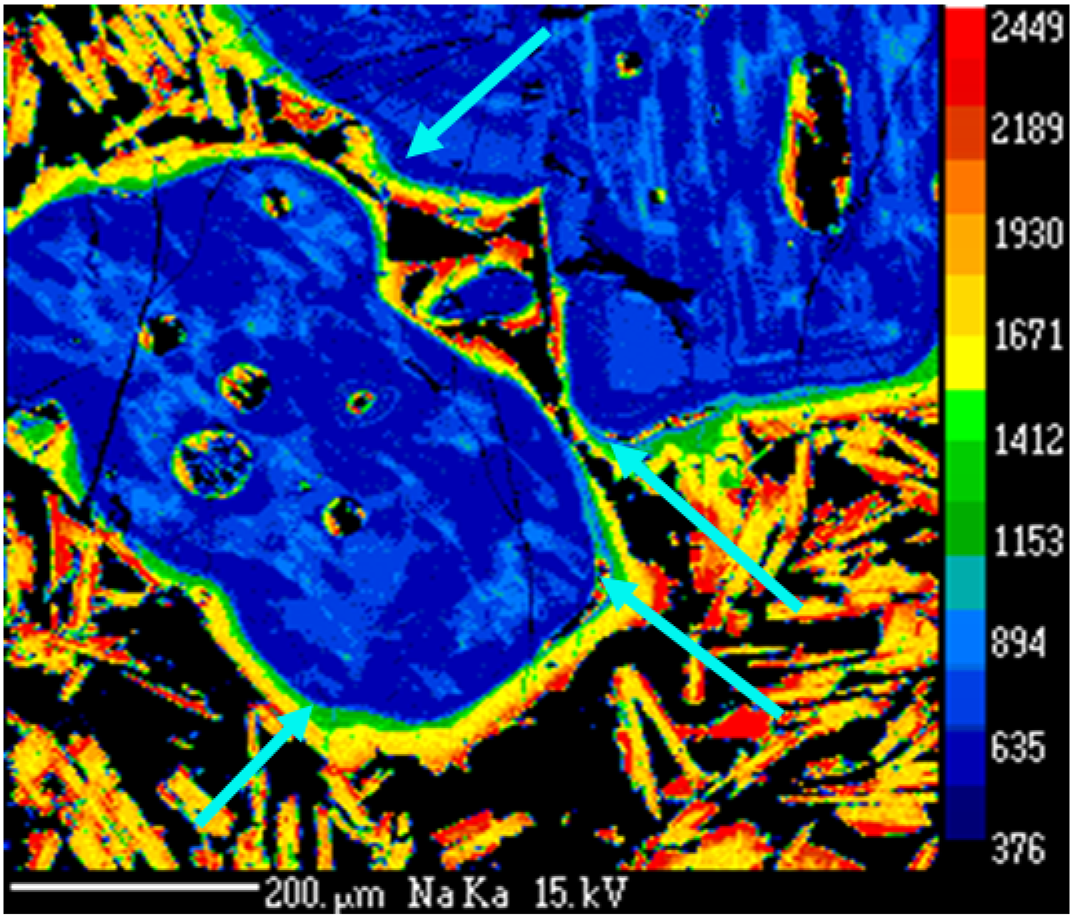


Fig. 8. Element maps of Na concentration for two different phenocrysts in sample 1516_1. Arrows indicate possible sites of resorption.



2449
2189
1930
1671
1412
1153
894
635
376
200. µm Na Ka 15. kV

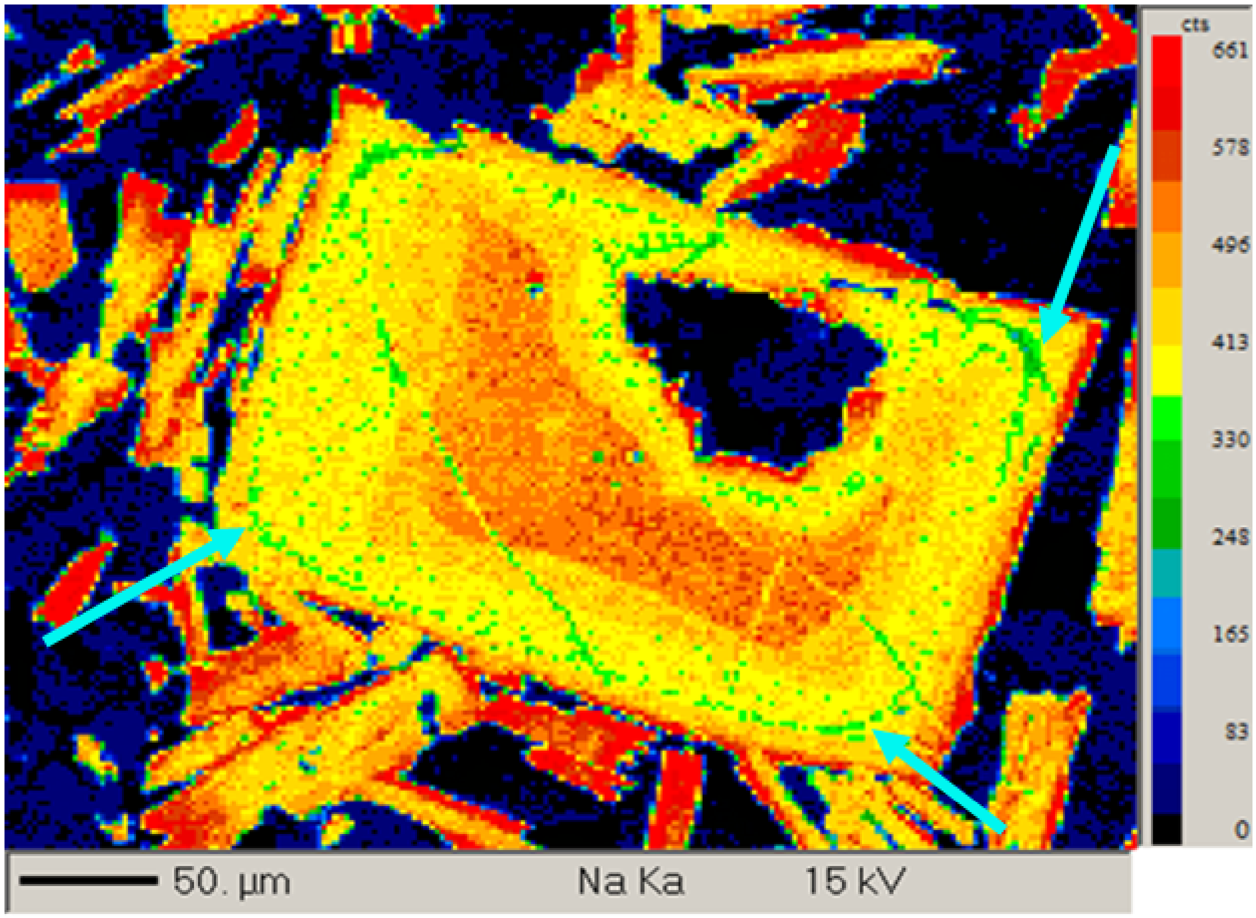


Fig. 9. Element maps of Na concentration for two different plagioclase phenocrysts in sample 1516_3. Arrows indicate rounded corners that are possible evidence of resorption.

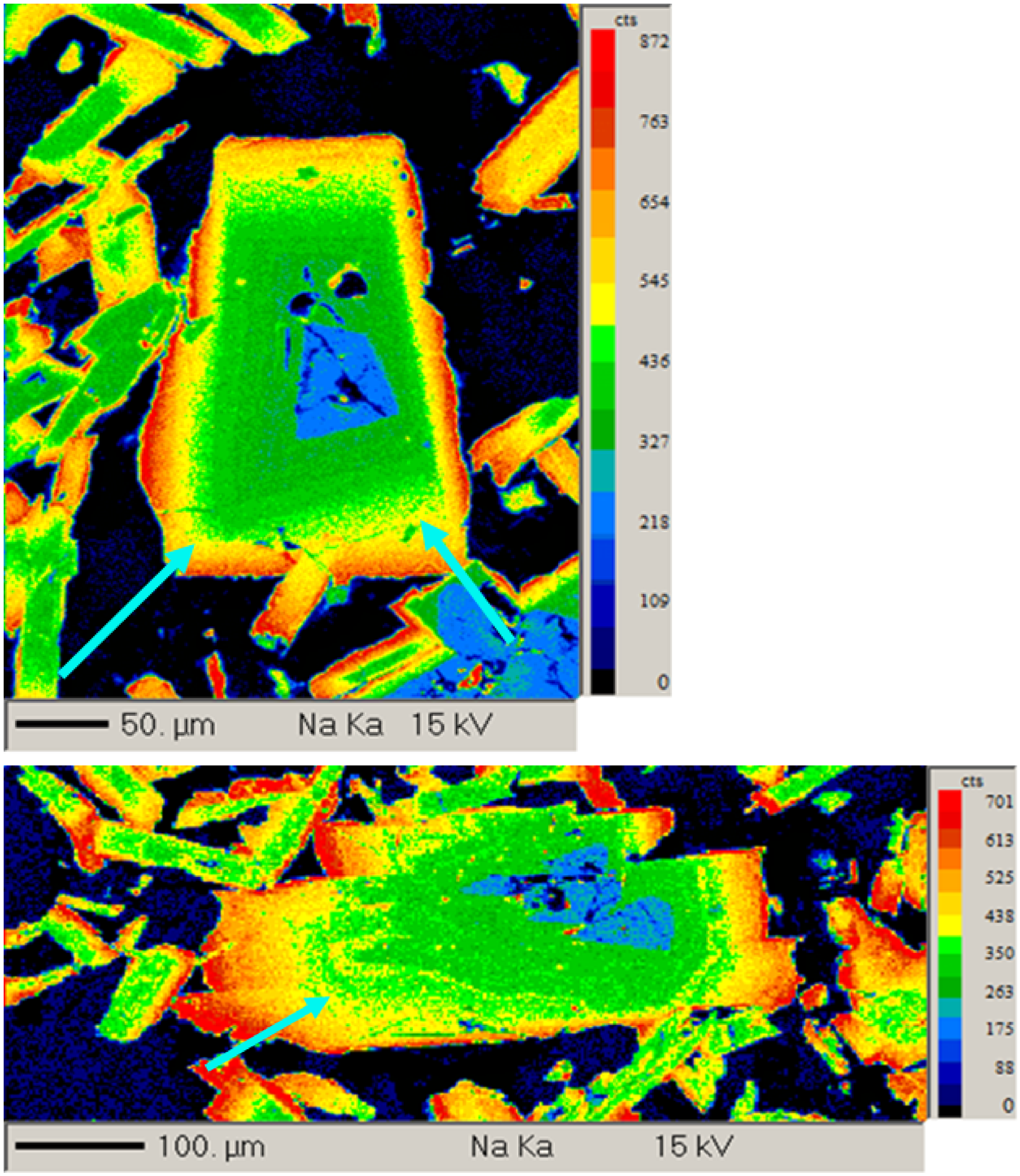


Fig. 10. Element maps of Na concentration for two different phenocrysts in sample 1516_11. Arrows indicate possible sites of resorption.

I calculated the average composition of all plagioclase and clinopyroxene phenocrysts, excluding rims, from the microprobe data (Table 1).

	Plagioclase phenocrysts	Clinopyroxene phenocrysts
MgO	0.4%	15.5%
Al2O3	30.1%	4.2%

Table 1. Selected oxide weight percents in the average plagioclase and clinopyroxene phenocryst.

4. Discussion

4.a EPMA data

I used the EPMA data to chart changes in the magma chamber as preserved by changes in composition between the core and the rim of each phenocryst. Na2O concentration in plagioclase phenocrysts is a proxy for temperature; at lower temperatures, a higher proportion of albite crystallizes. I used the different Na2O concentrations in each sample to construct a temperature history and propose a magma reinjection event. I also used the MgO concentrations in the phenocrysts as a proxy for magma evolution.

4.a.i Samples 1516_6 and 1516_8

The Na2O concentrations for plagioclase phenocrysts in samples 1516_6 and 1516_8 show a simple temperature history. They have a consistent low weight percent of Na2O throughout their phenocrysts (Figs. 2 and 3). Their consistent low Na2O concentrations indicate that the phenocrysts grew at a constant high temperature; higher than for either 1516_3 or 1516_5. Their high weight percent MgO relative to 1516_3 and 1516_5 (Fig. 5) suggests that they also formed in more primitive magma. Together, these two observations suggest that 1516_6 and 1516_8 probably erupted before 1516_3 and 1516_5.

4.a.ii Sample 1516_5

The weight percent of Na2O in the phenocrysts of sample 1516_5 increases between the core and the rim (Fig. 4), suggesting that temperatures in the magma chamber decreased as the phenocrysts formed. The trendline for the whole sample increases strongly, and the trendlines for every single transect have a positive slope (Fig. 3). The trendline has higher average Na2O concentrations than 1516_6 and 1516_8 as well, suggesting that it formed in cooler temperatures overall. Consistent with these observations are its MgO concentrations (Fig. 5), which are lower than those of 1516_6 and 1516_8. These observations suggest that 1516_5 erupted after 1516_6 and 1516_8, and that its phenocrysts formed in cooler, more evolved magma that also experienced a higher rate of cooling than 1516_6 and 1516_8.

There could be other explanations for the increasing concentrations of Na2O, however, which I will consider in turn. It's possible that this trend only reflects increased Na2O concentration in the rims of the phenocrysts that grew after eruption, and that this trend is not reflected throughout the rest of each phenocryst. It's also possible that 1516_5 experienced the same rate of cooling as 1516_6 and 1516_8 and that 1516_5 simply formed over a much longer time period.

The lines for some transects of 1516_5 do indeed tick sharply upwards near the rim (Fig. 2), suggesting that their overall positive trend may only be due to the composition of the rims. I attempt to eliminate the effect of the rims on the overall trend by removing the last three values from each transect and calculating a new trendline (Fig. 11). This is a crude measure given that each transect has a different number of measurements but it still shows a significant change. While the high Na2O concentration in the rims certainly has an effect on the slope of the 1516_5 trendline, it still has a higher slope than the other samples, and higher average concentration than 1516_6 or 1516_8, suggesting that the proposed cooling rate is real, and not only an effect of the rims.

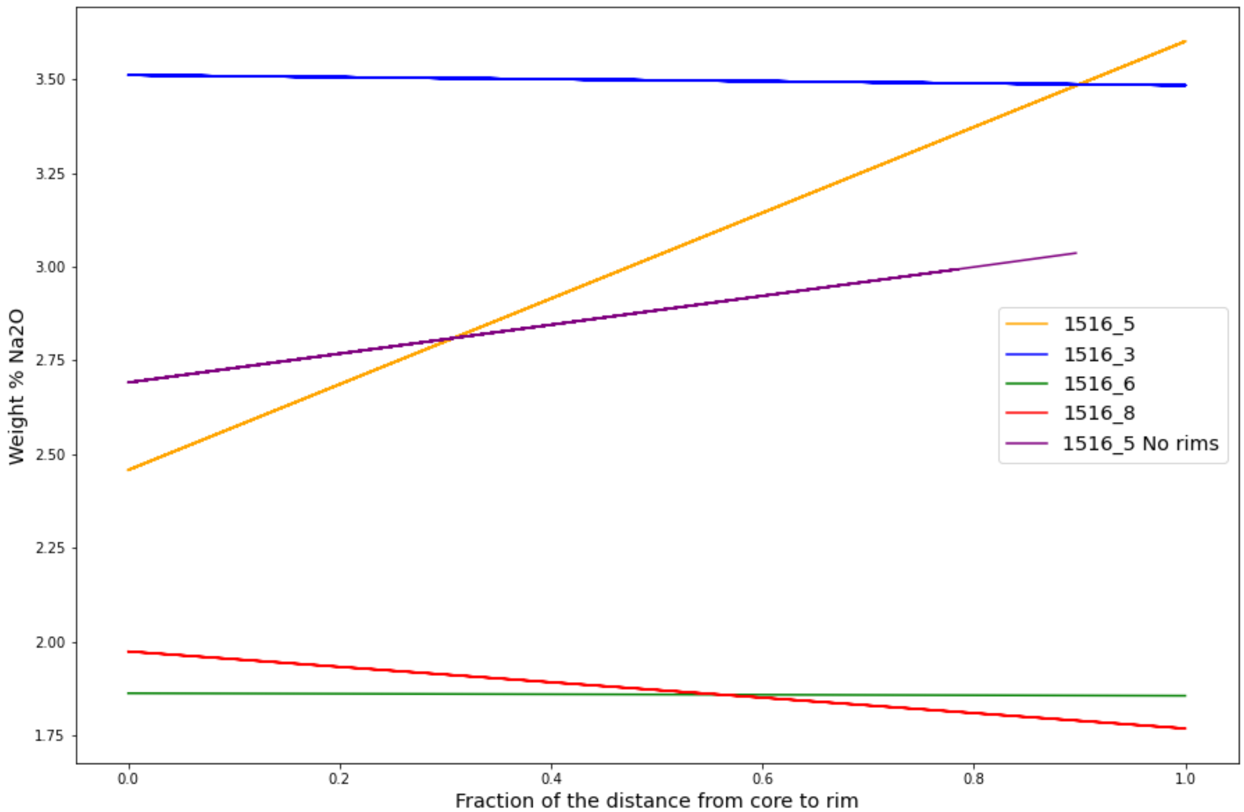


Fig. 11. Trendlines for each sample, and for 1516_5 with the last three measurements of each transect removed (modified from Fig. 4).

Alternatively, the phenocrysts of 1516_5 may have experienced the same rate of temperature decrease as samples 1516_6 and 1516_8 but over a much longer period of time. This is certainly possible, as the EPMA data do not indicate the size of each crystal. Assuming crystals grew at the same rate, the relative time it took for each to grow could be gleaned from naked eye views of the thin section slides. The phenocrysts in 1516_6 are, if anything, larger than those in 1516_5, and 1516_8 lacks large crystals almost entirely (Fig. 1). 1516_6 and 1516_8 have similar Na2O gradients despite having different phenocryst sizes, and 1516_5 and 1516_6 have quite different gradients despite having similar phenocryst sizes (Fig.4), so the gradient of 1516_5 likely has very little to do with the size of its phenocrysts, but instead represents a real magmatic process.

4.a.iii Sample 1516_3

Sample 1516_3 forms two distinct clusters of data (Fig. 3). The weight percent MgO data also show two distinct clusters of data (Fig. 5). Additionally, they show that phenocrysts in 1516_3 with low Na2O concentrations have higher MgO concentrations, suggesting that the magma from which they formed was hotter and more primitive. Lower MgO concentrations in the data with higher Na2O concentrations suggest these phenocrysts formed from cooler, more evolved magma.

I propose that the preservation of phenocrysts from two different magmas may be the result of an injection of new magma from the mantle. In this case, the lower Na2O concentration cluster would have cooled from hot, primitive magma that entered the chamber after the higher Na2O concentration crystals had formed. The more primitive crystals would have formed more recently than the evolved crystals.

Element maps of the phenocrysts should show resorption textures around the edges of some phenocrysts if this is the case. Fig. 9 shows several areas where inner zones have rounded edges that contrast with the sharp corners of the crystal. These could represent areas where hotter magma began to remelt the original crystal. Additionally, one of the high Na2O concentration transects shows a downward turn near its rim; this could also be the result of resorption (Fig. 2).

This would also explain why the composition of the groundmass overlaps with the high Na2O concentration cluster (Fig. 12). Instead of representing a cooler, more evolved version of all the phenocrysts, as it does in other samples, the groundmass in 1516_3 represents only a more evolved version of the newly injected primitive magma, which is similar in composition to the lower temperature phenocrysts.

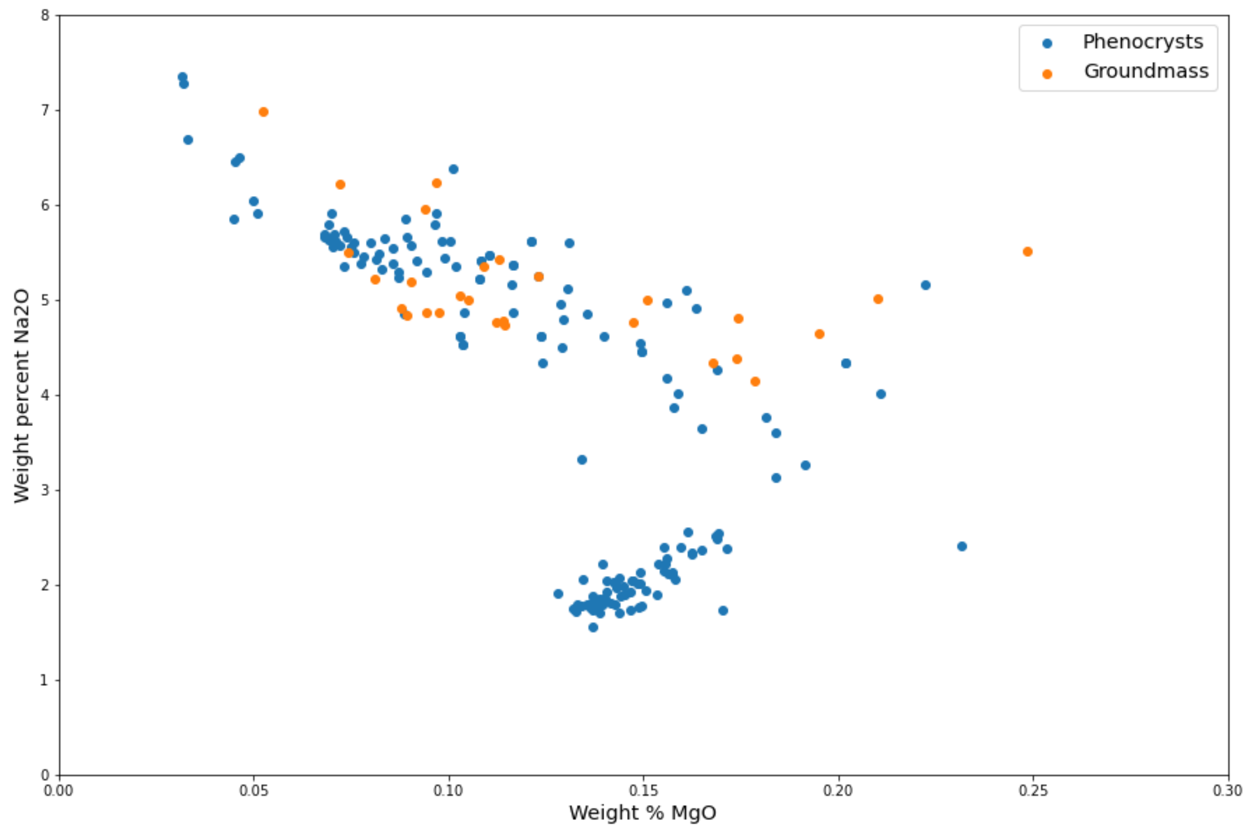


Fig. 12. Weight % MgO and Na2O for phenocrysts and groundmass in sample 1516_3 (modified from Figs. 5 and 6).

4.a.iv Summary

From the EPMA data, I propose the following temperature history for these four samples. Samples 1516_6 and 1516_8 formed in hot, primitive magma with a stable temperature. 1516_5 formed in cooler evolved magma whose rate of cooling had also increased. The higher Na2O cluster of 1516_3 formed at stable temperatures in even cooler, more evolved magma. Finally, hot primitive magma was injected into the chamber, forming the lower Na2O cluster of 1516_3 phenocrysts.

4.b XRF data

XRF data show the bulk composition of each sample. The thin sections show that the majority of each sample is groundmass, not phenocrysts, so the bulk rock composition can be used as an approximation for the liquid lava from which the groundmass formed. Bulk rock samples from different eruptions represent snapshots in the evolution of the liquid in the magma chamber as it changes due to fractional crystallization.

The samples should form liquid lines of descent that retreat from the composition of the crystallizing phenocrysts. I test this by plotting the average composition of the plagioclase and clinopyroxene phenocrysts (Table 1) along with the XRF data (Fig. 13).

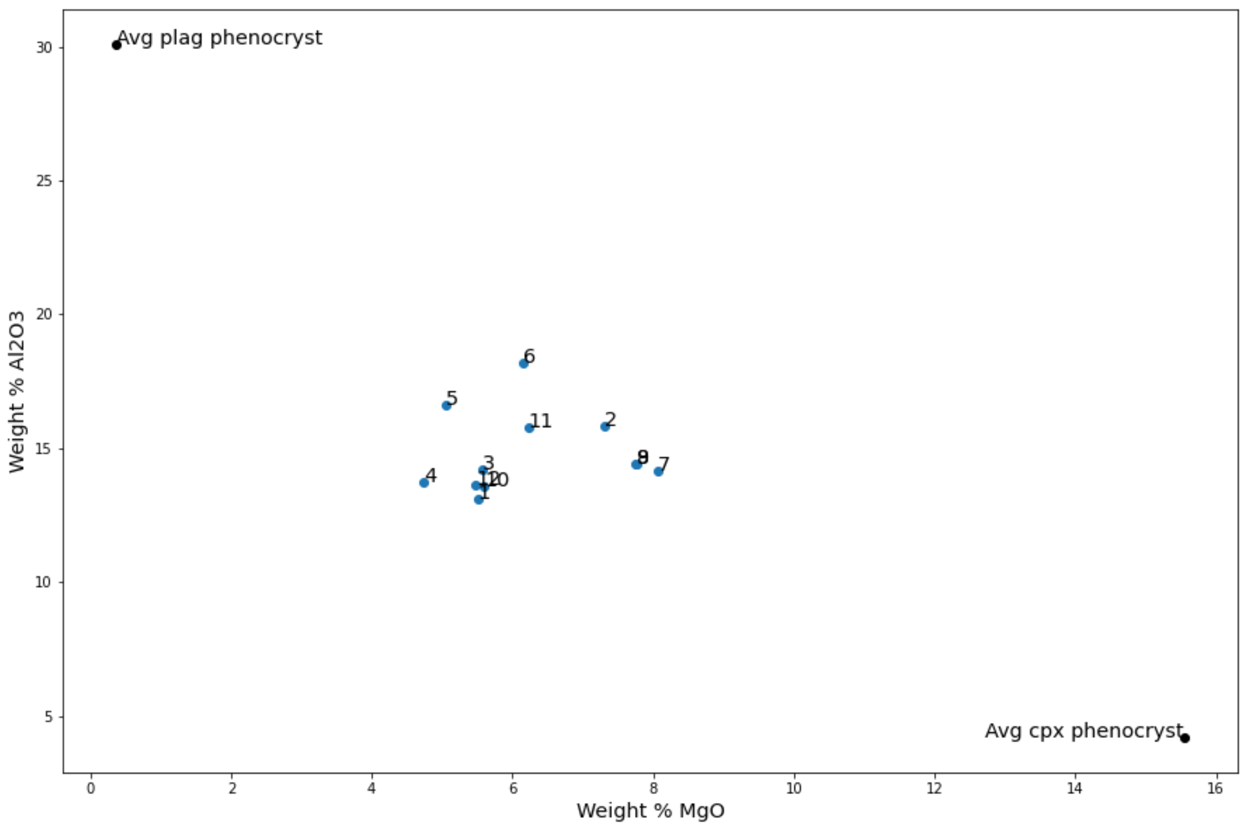


Fig. 13. Bulk rock compositions and average phenocryst compositions (modified from Fig. 7 and Table 1).

Fig. 13 shows samples 1516_2, 1516_7, 1516_8 and 1516_9 retreating from the average clinopyroxene composition, while the other samples retreat from a mixture of clinopyroxene and plagioclase. Proposed liquid lines of descent are added in Fig. 14, based on linear regression.

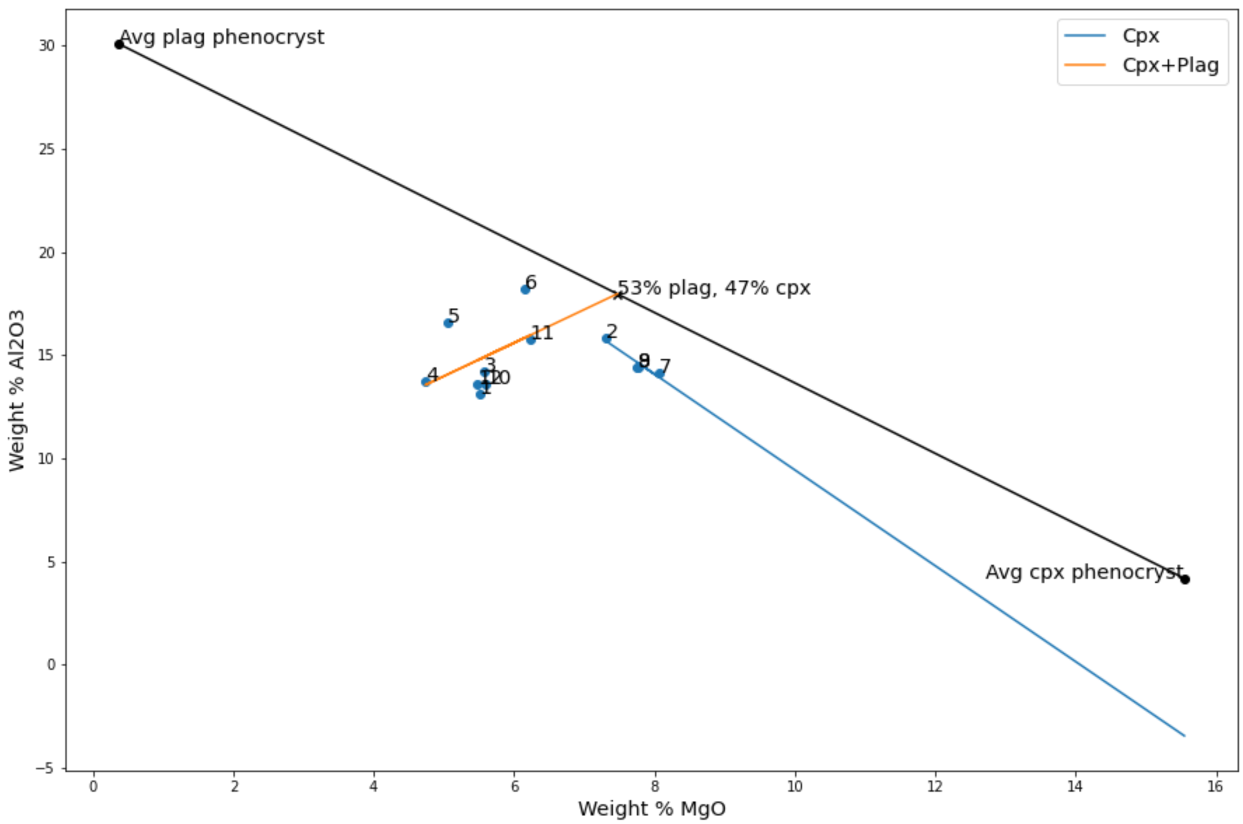


Fig. 14. Liquid lines of descent, from clinopyroxene and a mixture of clinopyroxene and plagioclase.

This proposed history, in which the magma chamber initially crystallizes clinopyroxene, then a mix of clinopyroxene and plagioclase, agrees with the basalt phase diagram (Fig. 15). In my proposed history, magma evolution follows the red arrow. The roughly equal proportions of plagioclase and clinopyroxene in the second line of descent is in agreement with the phase diagram as well.

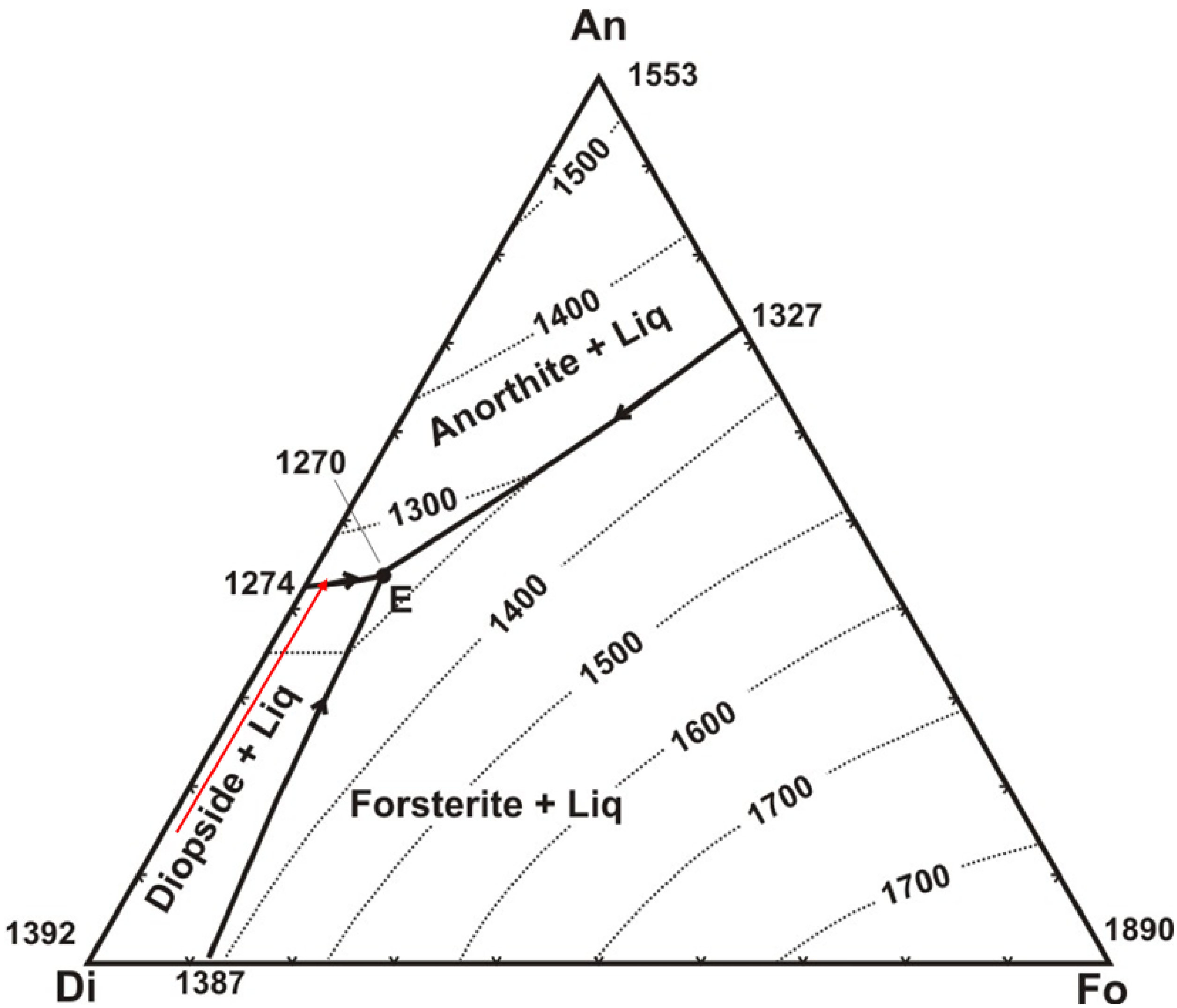


Fig. 15. The forsterite-anorthite-diopside basalt phase diagram. Proposed crystallization history follows the red arrow.

Though the clinopyroxene line does not perfectly fit the average clinopyroxene phenocryst, it is within natural variation given the limited number of samples. The mixed clinopyroxene-plagioclase line, however, does not fit the data very well, and by putting 1516_3 on the same line of descent as 1516_5 and 1516_6, this model ignores the reinjection event that I proposed.

Instead, I propose that the cluster of samples 1516_1, 1516_3, 1516_10, 1516_11 and 1516_12 represents a second liquid line of descent occurring after the reinjection event (Fig. 16). The reinjected magma will be more primitive, explaining this line's higher MgO concentration. This model also agrees with the chronology proposed by the electron microprobe data, where 1516_8 is the oldest, followed by 1516_6, 1516_5 and finally 1516_3 after the reinjection event.

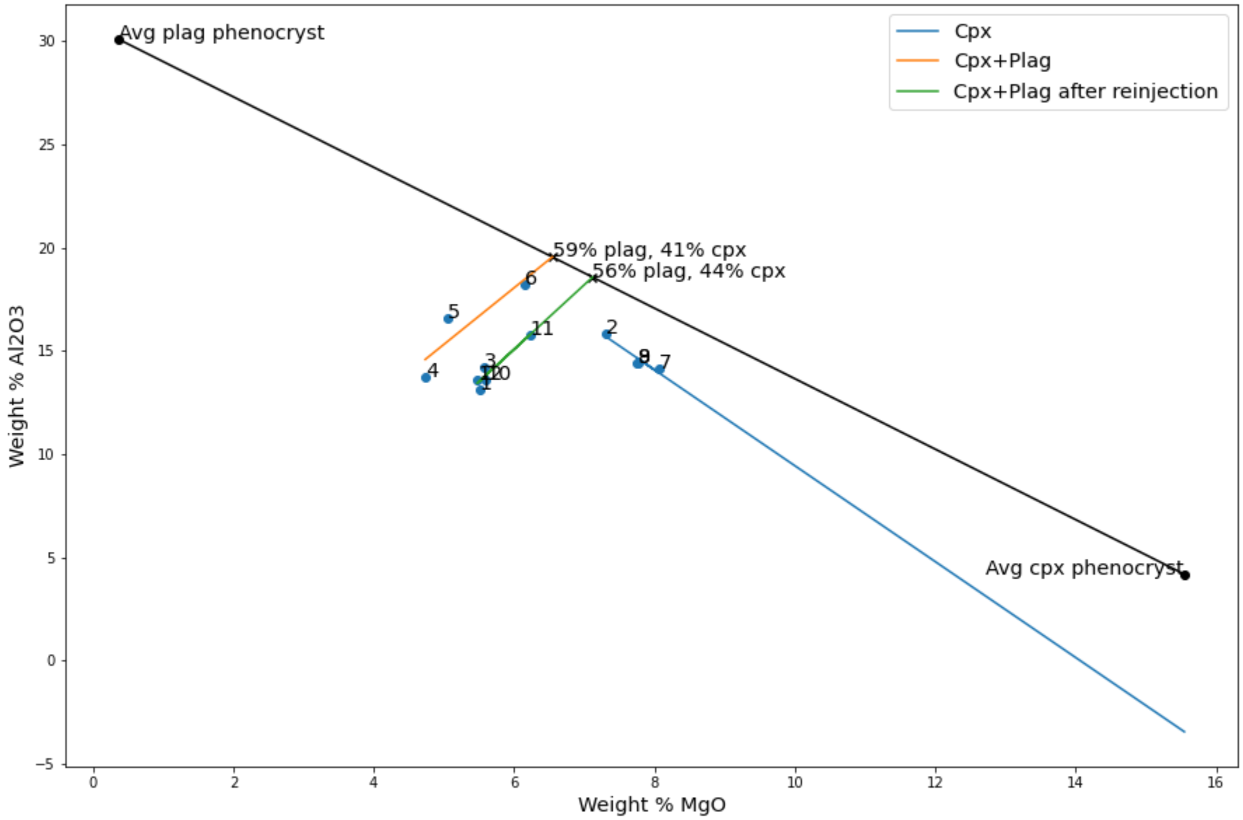


Fig. 16. Liquid lines of descent before and after reinjection.

If samples 1516_1 and 1516_11 were also formed after the reinjection event, they should show the same resorption textures in element maps and separate clusters of phenocryst composition in EPMA data as 1516_3. Element maps of 1516_1 and 1516_11 do not obviously show the same rounded inner zones characteristic of resorption as 1516_3, however (Figs. 10 and 12), but there are some possible sites where they may have been resorbed.

As previous students only analysed specific points within phenocrysts with the electron microprobe, not whole transects, 1516_1 and 1516_11 lack sufficient data to make separate clusters obvious, but it is arguable that they exist (Fig. 17). Their phenocryst compositions certainly have very similar distributions and values to 1516_3, and their phenocryst compositions also overlap with the groundmass compositions. Their similarity in phenocryst composition to 1516_3 is good evidence that these samples also formed in a separate liquid line of descent after a magma reinjection.

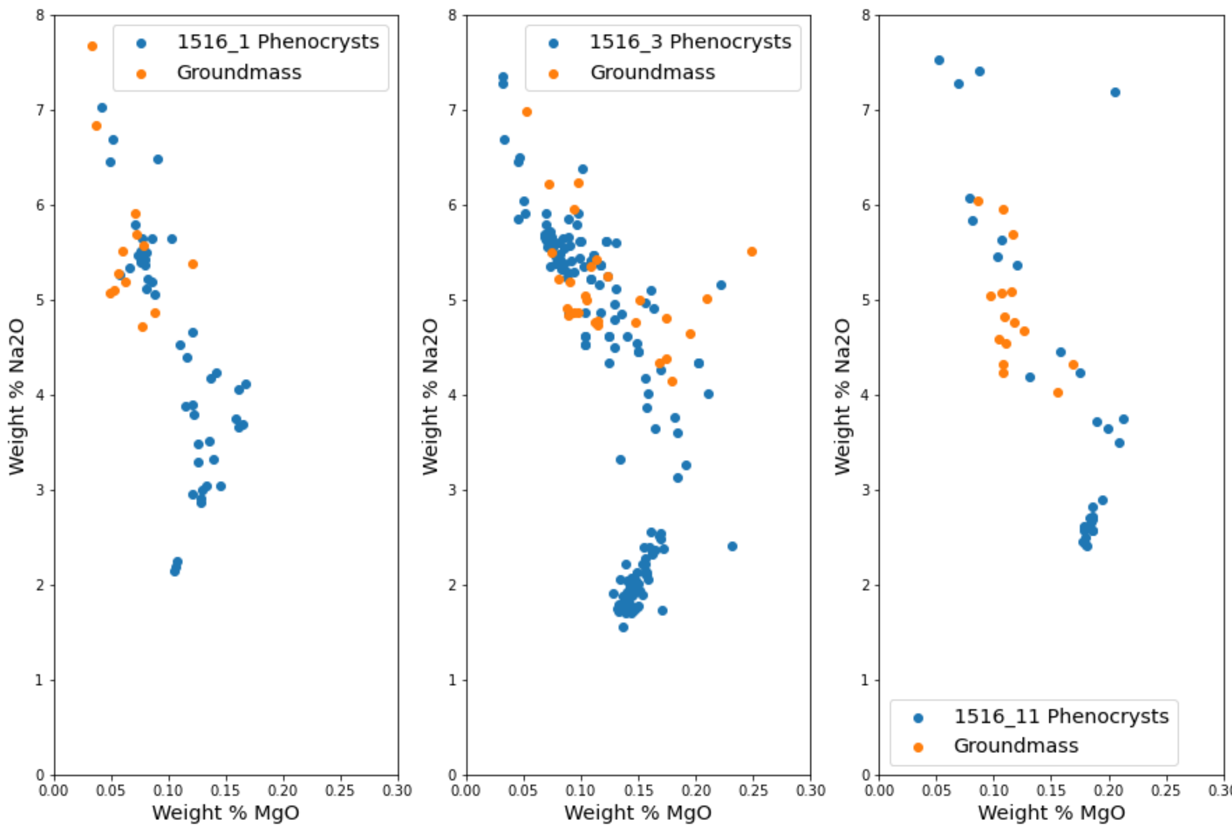


Fig. 17. Plagioclase phenocryst and groundmass compositions for samples 1516_1, 1516_3 and 1516_11 (modified from Figs. 5 and 6).

Though well supported by the microprobe data, this model of magma evolution has flaws. Most glaringly, it does not account for olivine crystallisation at all, even though most of the thin sections contain numerous olivine crystals altered to iddingsite. It may be that the alteration of olivine has changed the bulk composition of the samples so that they no longer show liquid lines of descent from olivine.

The XRF data suggest a model in which clinopyroxene is crystallized, then a mix of plagioclase and clinopyroxene. After new magma is injected, a slightly more primitive mix of plagioclase and clinopyroxene are crystallized. Though this model is well-supported by the data, it does not account for the presence of olivine, so it can, at best, be considered preliminary. Additional samples with unaltered olivine would help further constrain the model.

4.c Sources of error

Inconsistent labelling of the EPMA data and mistakes in EPMA mineral identification led to difficulties in processing the data efficiently and may have led to errors in this report.

5. Conclusion

From the available data, I have constructed a tentative history of the magma chamber and the samples, and answered questions posed at the outset. The abundance of large phenocrysts in the thin sections suggests that fractional crystallization occurred in the magma chamber. EPMA transects suggest that samples 1516_6 and 1516_8 formed in stable high temperatures in primitive magma. Transects across phenocrysts in 1516_5 suggest that it formed in cooler, evolved magma that also experienced a higher rate of cooling. Finally, the data suggest that 1516_3 contains two distinct groups of phenocrysts, one of which formed in cooler evolved magma, and the other in hot primitive magma, and that the presence of these two distinct groups may be the result of the injection of new hot magma from the mantle into the magma chamber. The cooler cluster would have formed after 1516_5, and the hotter cluster only after the reinjection event.

Bulk rock composition data support the chronology suggested by the microprobe data and the theory of magma reinjection. They also show that the magma first crystallized clinopyroxene, then an even mix of clinopyroxene and plagioclase.

This analysis gives a detailed picture of the processes in the magma chamber below the Breidallur complex, and helps shed further light on Icelandic volcanology.

References

Thordarson, T. and Höskuldsson, Á., 2008. Postglacial volcanism in Iceland. *Jökull*, *58*(198), p.e228.

Appendix 1: Python code

Import relevant packages

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

Load and clean data

In [2]:

```
details=pd.read_excel('Sample details.xlsx',header=4,usecols='C:H')
M={'/':'_'}
details['Sample number']=["".join([M.get(c,c) for c in STR]) for STR in details['Sample number']]
```

In [3]:

```
#Load n Clean xrf
xrf=pd.read_excel('whole rock data(1).xlsx',header=5)
xrf.dropna(axis=0,how='all',inplace=True)
xrf.dropna(axis=1,how='all',inplace=True)
xrf.drop(0,inplace=True)
xrf.drop('Unnamed: 16',axis=1,inplace=True)
xrf.fillna(method='ffill',inplace=True)
xrf.rename(columns={'Sample ':'Sample'},inplace=True)
M={'/':'_'}
xrf['Sample']=["".join([M.get(c,c) for c in STR]) for STR in xrf['Sample']]
```

In [4]:

```
xrfte=pd.read_excel('whole rock data(1).xlsx',sheet_name='trace elements',header=5)
xrfte.drop('Unnamed: 21',axis=1,inplace=True)
xrfte.dropna(how='all',axis=0,inplace=True)
xrfte.dropna(how='all',axis=1,inplace=True)
xrfte.drop(0,inplace=True)
xrfte[xrfte=='n.d.']=np.nan
xrfte.rename(columns={'Unnamed: 0':'Sample'},inplace=True)
xrfte['Sample'].fillna(method='ffill',inplace=True)
xrfte['Sample']=["".join([M.get(c,c) for c in STR]) for STR in xrfte['Sample']]
xrfte.fillna(0,inplace=True)
```

In [5]:

```
#Normalise to actual total
```

```
tots=xrf[xrf.columns[1:11]].sum(axis=1).to_numpy()
xrfc=pd.DataFrame().reindex_like(xrf)
for i in range(len(xrf)):
    xrfc.iloc[i,1:11]=(xrf.iloc[i,1:11]/tots[i])*100
xrfc['Sample']=xrf['Sample']
xrfc.dropna(how='all',axis=1,inplace=True)
```

In [6]:

```
xrfc=xrfc.join(xrfte.set_index('Sample'),on='Sample')
xrfc['Eruption']=[details[details['Sample number']==i]['Source'].tolist()[0] for i in xrfc['Sample']]
```

In [7]:

```
xrf_means=xrfc.groupby('Sample').mean()
```

In [8]:

```
def clean_epma(sample,number):
    sample.dropna(thresh=2,axis=1,inplace=True)
    sample.dropna(thresh=2,axis=0,inplace=True)
    sample.rename(columns={sample.columns[-2]:'Comment',sample.columns[-1]:'Geoff
notes'},inplace=True)
    sample.index=np.arange(len(sample))
    sample['Sample']=[number]*len(sample)
```

In [9]:

```
epma_18_5=pd.read_excel('EPMA 2018 Data.xlsx',sheet_name='1516_5',header=4,index_col=0)
clean_epma(epma_18_5,'1516_5')
```

In [10]:

```
epma_18_3=pd.read_excel('EPMA 2018 Data.xlsx',sheet_name='1516_3',header=4,index_col=0)
clean_epma(epma_18_3,'1516_3')
```

In [11]:

```
epma_18_6=pd.read_excel('EPMA 2018 Data.xlsx',sheet_name='1516_6',header=4,index_col=0)
clean_epma(epma_18_6,'1516_6')
```

In [12]:

```
epma_18_8=pd.read_excel('EPMA 2018 Data.xlsx',sheet_name='1516_8',header=4,index_col=0)
clean_epma(epma_18_8,'1516_8')
```

In [13]:

```
epma_20_3=pd.read_excel('EPMA 2020 Data.xlsx',sheet_name='1516_3',header=5,index_col=0)
clean_epma(epma_20_3,'1516_3')
```

In [14]:
```
epma_20_8=pd.read_excel('EPMA 2020 Data.xlsx',sheet_name='1516_8',header=5,index_col=0)
clean_epma(epma_20_8,'1516_8')
```

In [15]:
```
epma_20_5=pd.read_excel('EPMA 2020 Data.xlsx',sheet_name='1516_5',header=5,index_col=0)
clean_epma(epma_20_5,'1516_5')
```

In [16]:
```
epma_20_6=pd.read_excel('EPMA 2020 Data.xlsx',sheet_name='1516_6',header=5,index_col=0)
clean_epma(epma_20_6,'1516_6')
```

In [17]:
```
epma_19_6=pd.read_excel('EPMA 2019 Data.xlsx',sheet_name='1516_6',header=5,index_col=0)
clean_epma(epma_19_6,'1516_6')
```

In [18]:
```
epma_19_5=pd.read_excel('EPMA 2019 Data.xlsx',sheet_name='1516_5',header=5,index_col=0)
clean_epma(epma_19_5,'1516_5')
```

In [19]:
```
epma_19_1=pd.read_excel('EPMA 2019 Data.xlsx',sheet_name='1516_1',header=5,index_col=0)
clean_epma(epma_19_1,'1516_1')
```

In [20]:
```
epma_19_3=pd.read_excel('EPMA 2019 Data.xlsx',sheet_name='1516_3',header=6,index_col=0)
clean_epma(epma_19_3,'1516_3')
```

In [21]:
```
epma_19_8=pd.read_excel('EPMA 2019 Data.xlsx',sheet_name='1516_8',header=5,index_col=0)
clean_epma(epma_19_8,'1516_8')
```

In [22]:
```
epma_19_11=pd.read_excel('EPMA 2019 Data.xlsx',sheet_name='1516_11',header=5,index_col=0)
clean_epma(epma_19_11,'1516_11')
```

Assemble DataFrame

In [23]:

```
dfs=[epma_18_3,epma_18_6,epma_18_8,epma_20_3,epma_20_5,epma_20_6,epma_20_8,epma_
19_1,epma_19_3,epma_19_5,epma_19_6,epma_19_8,epma_19_11]
for i in dfs:
    epma_18_5=epma_18_5.append(i)
all_data=epma_18_5
#all_data['Comment'].fillna(all_data['Comment from probe session'],inplace=True)
#del all_data['Comment from probe session']
all_data.fillna(0,inplace=True)
all_data.index=np.arange(len(all_data))
```

In [24]:

```
#Add column: 'Mineral'
all_data['Mineral']=np.zeros_like(all_data['SiO2'])

mask=(all_data['Geoff notes'].str.lower().str.contains('x',na=False) |
all_data['Comment'].str.lower().str.contains('x',na=False))
all_data['Mineral']=np.where(mask,'Clinopyroxene',all_data['Mineral'])

mask=(all_data['Geoff notes'].str.lower().str.contains('plag',na=False) |
all_data['Comment'].str.lower().str.contains('plag',na=False))
all_data['Mineral']=np.where(mask,'Plagioclase',all_data['Mineral'])

mask=(all_data['Geoff notes'].str.lower().str.contains('sp',na=False) |
all_data['Comment'].str.lower().str.contains('sp',na=False))
all_data['Mineral']=np.where(mask,'Spinel',all_data['Mineral'])
```

Create useful sub-dataframes

In [25]:

```
color_dict={'1516_3': 'blue', '1516_5': 'orange', '1516_6': 'green', '1516_8':
'red','1516_1':'cyan','1516_11':'magenta'}
mineral_dict={'Plagioclase':'blue','Clinopyroxene':'orange','Spinel':'green'}
eruption_dict={'Fissure':'blue','Volcano':'orange'}
```

In [26]:

```
details['Sample
number']=['1516_1','1516_2','1516_3','1516_4','1516_5','1516_6','1516_7','1516_8','1516_9','1516_1
0','1516_11','1516_12']
eruptions_nested_list=[details[details['Sample number']==i]['Source'].tolist() for i in
all_data['Sample']]
all_data['Eruption']=[i for j in eruptions_nested_list for i in j]
```

In [27]:

```
plag=all_data[all_data['Mineral']=='Plagioclase']
cpx=all_data[all_data['Mineral']=='Clinopyroxene']

#remove misidentified ones
not_cpx=all_data[all_data['Geoff notes'].str.contains('not',na=False)]['Comment'].unique()
for i in not_cpx:
    cpx=cpx[cpx['Comment']!=i]
```

In [28]:

```
#transects
transect_comments=all_data['Comment'].value_counts()>=6
transect_counts=all_data['Comment'].value_counts()[transect_comments]
uniques=all_data['Comment'].unique()
transects=pd.DataFrame()
for i in uniques:
    if i in transect_counts:
        transects=transects.append(all_data[all_data['Comment']==i])
```

In [29]:

```
#remove mistakes
not_transects=['Section 11 Groundmass plagioclase','1. Plag rim to rim','Phenocryst 2
core','Phenocryst 3 core',
              'Phenocryst 4 core','Phenocryst 5 core','Phenocryst 7 core','Phenocryst 8 core',
              'IS/16-5 Plag phenocryst 4 core','sample 6 groundmass core','Groundmass Core Sample
8','Phenocryst 7 rim']
for i in not_transects:
    transects=transects[transects['Comment']!=i]
```

In [30]:

```
#add distance columns to transects

#distance as a fraction
frac_dist_arrays=[np.linspace(0,1,len(transects[transects['Comment']==i])) for i in
transects['Comment'].unique()]
transects['frac_dist']=[i for j in frac_dist_arrays for i in j]

#distance as an integer
ab_dist_arrays=[np.arange(len(transects[transects['Comment']==i])) for i in
transects['Comment'].unique()]
transects['ab_dist']=[i for j in ab_dist_arrays for i in j]
```

In [31]:

```
#Plagioclase transects
plag_transects=transects[transects['Mineral']=='Plagioclase']
```

In [32]:

```
mask=all_data['Comment'].str.lower().str.contains('ground',na=False)
gmass=all_data[mask]
gmass.index=np.arange(len(gmass))
```

In [33]:

```
g_plag=gmass[gmass['Mineral']=='Plagioclase']
```

In [34]:

```
samples=['1516_3','1516_5','1516_6','1516_8']
```

In [35]:

```
all_pcrysts=plag_transects.append(plag[plag['Comment'].str.contains('cryst',na=False)])
```

In [36]:

```
psect_cores=plag_transects[plag_transects['frac_dist']==0]
```

In [37]:

```
mask=(plag['Comment'].str.lower().str.contains('ground',na=False) |
      plag['Comment'].str.lower().str.contains('rim',na=False) |
      plag['Geoff notes'].str.lower().str.contains('rim',na=False))
pcrysts=plag[~mask]
pcrysts.index=np.arange(len(pcrysts))

mask=(cpx['Comment'].str.lower().str.contains('ground',na=False) |
      cpx['Comment'].str.lower().str.contains('rim',na=False) |
      cpx['Geoff notes'].str.lower().str.contains('rim',na=False))
xcrysts=cpx[~mask]
xcrysts.index=np.arange(len(xcrysts))
```

Define useful functions

In [38]:

```
def trendline(x,y,z,a):
    m,c,_,_,_=stats.linregress(x,y)
    return plt.plot(x,m*np.array(x)+c,c=z,label=a)
```

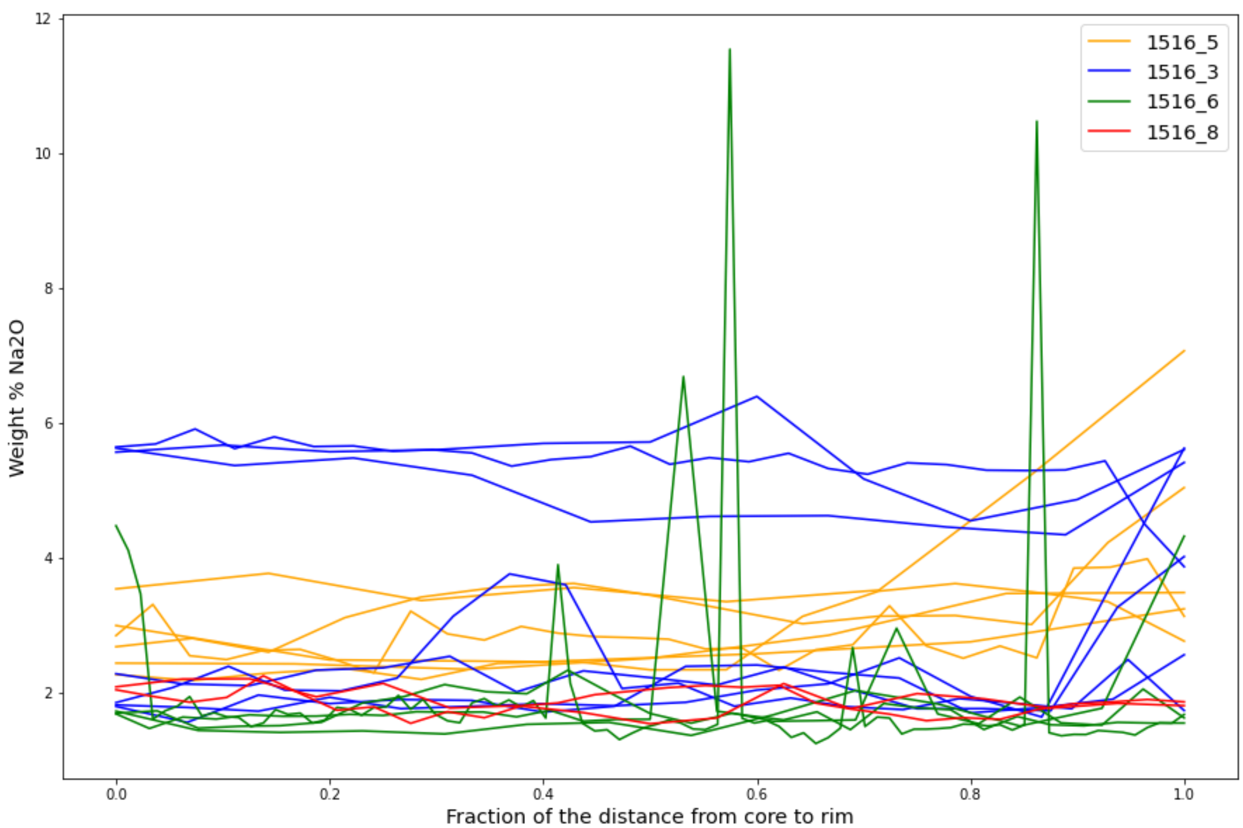
Results

In [39]:

```
#Raw transect data
plt.figure(figsize=(15,10))
for i in plag_transects['Sample'].unique():
    sample=plag_transects[plag_transects['Sample']==i]
    for j in sample['Comment'].unique():
        sect=sample[sample['Comment']==j]
        x=sect['frac_dist']
        y=sect['Na2O']
        if j==sample['Comment'].unique()[0]:
            plt.plot(x,y,c=color_dict[i],label=i)
        else:
            plt.plot(x,y,c=color_dict[i])
plt.xlabel('Fraction of the distance from core to rim',fontsize='x-large')
plt.ylabel('Weight % Na2O',fontsize='x-large')
plt.legend(fontsize='x-large')
```

Out[39]:

<matplotlib.legend.Legend at 0x23b57966f70>



In [40]:

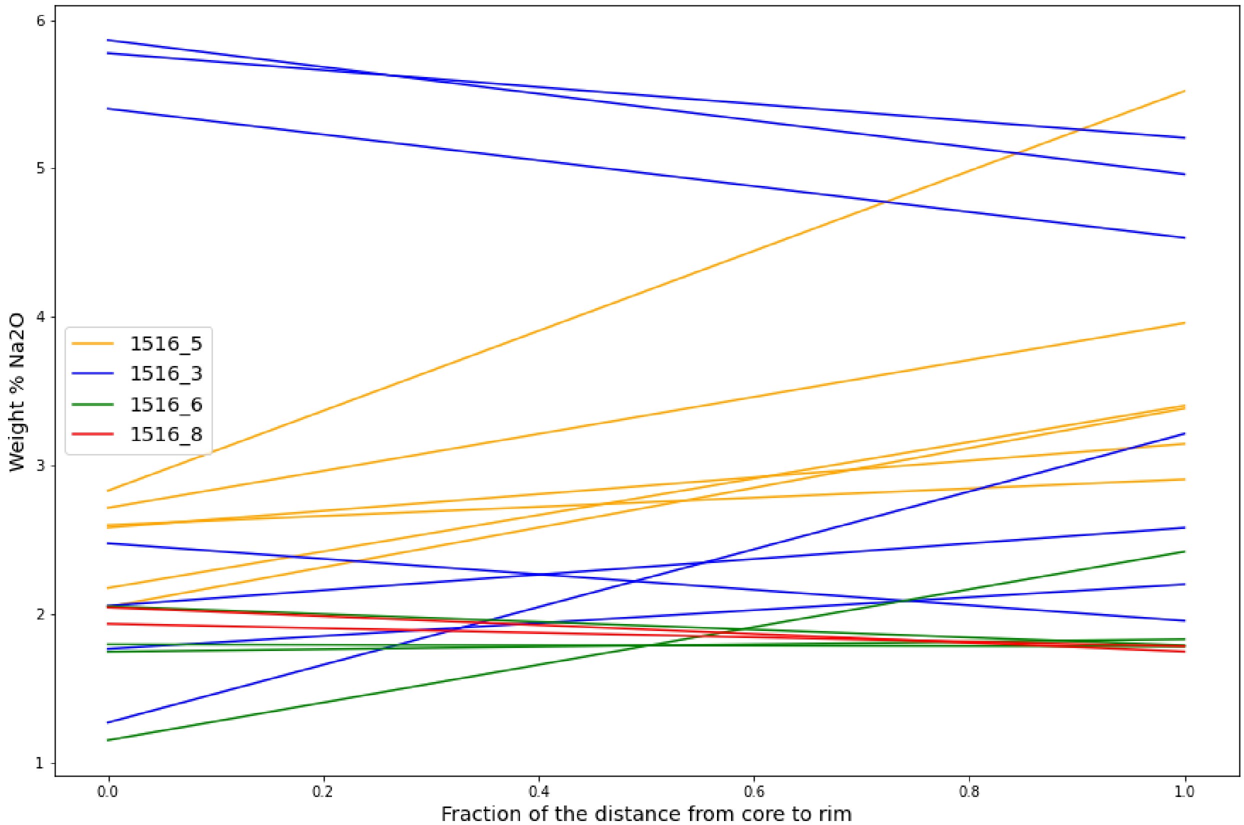
```
plt.figure(figsize=(15,10))
for i in plag_transects['Sample'].unique():
```

```
    sample=plag_transects[plag_transects['Sample']==i]
    for j in sample['Comment'].unique():
        sect=sample[sample['Comment']==j]
        x=sect['frac_dist'].tolist()
        y=sect['Na2O'].tolist()
        if j==sample['Comment'].unique()[0]:
            trendline(x,y,z=color_dict[i],a=i)
        else:
            trendline(x,y,z=color_dict[i],a=None)
plt.xlabel('Fraction of the distance from core to rim',fontsize='x-large')
plt.ylabel('Weight % Na2O',fontsize='x-large')
plt.legend(fontsize='x-large')
```

```
C:\Users\Public\Anaconda3\lib\site-packages\scipy\stats\_stats_mstats_common.py:130:
RuntimeWarning: invalid value encountered in double_scalars
  slope = r_num / ssxm
C:\Users\Public\Anaconda3\lib\site-packages\scipy\stats\_stats_mstats_common.py:140:
RuntimeWarning: invalid value encountered in sqrt
  t = r * np.sqrt(df / ((1.0 - r + TINY)*(1.0 + r + TINY)))
C:\Users\Public\Anaconda3\lib\site-packages\scipy\stats\_stats_mstats_common.py:142:
RuntimeWarning: invalid value encountered in double_scalars
  sterrest = np.sqrt((1 - r**2) * ssym / ssxm / df)
C:\Users\Public\Anaconda3\lib\site-packages\scipy\stats\_stats_mstats_common.py:130:
RuntimeWarning: invalid value encountered in double_scalars
  slope = r_num / ssxm
C:\Users\Public\Anaconda3\lib\site-packages\scipy\stats\_stats_mstats_common.py:140:
RuntimeWarning: invalid value encountered in sqrt
  t = r * np.sqrt(df / ((1.0 - r + TINY)*(1.0 + r + TINY)))
C:\Users\Public\Anaconda3\lib\site-packages\scipy\stats\_stats_mstats_common.py:142:
RuntimeWarning: invalid value encountered in double_scalars
  sterrest = np.sqrt((1 - r**2) * ssym / ssxm / df)
```

Out[40]:

```
<matplotlib.legend.Legend at 0x23b57abe2e0>
```

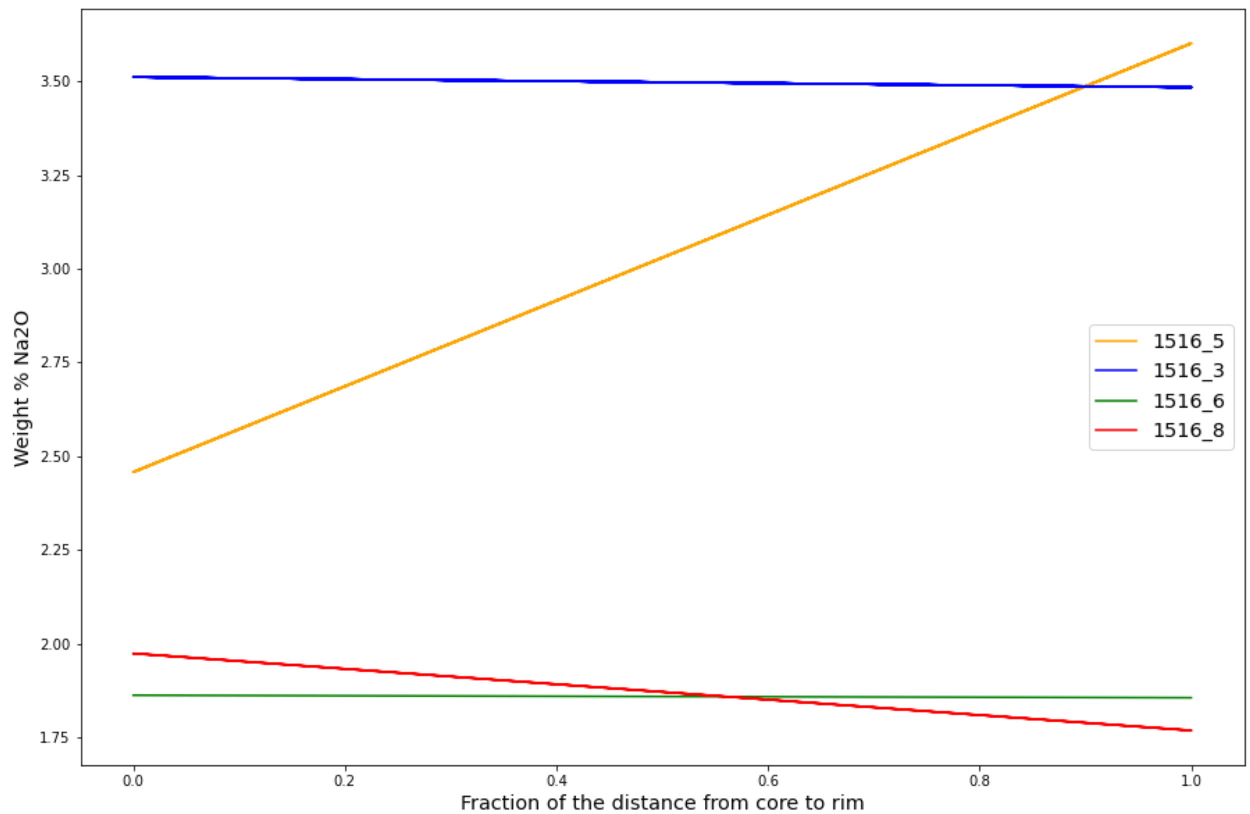


In [41]:

```
plt.figure(figsize=(15,10))
for i in plag_transects['Sample'].unique():
    sample=plag_transects[plag_transects['Sample']==i]
    x=sample['frac_dist'].tolist()
    y=sample['Na2O'].tolist()
    trendline(x,y,z=color_dict[i],a=i)
plt.xlabel('Fraction of the distance from core to rim',fontsize='x-large')
plt.ylabel('Weight % Na2O',fontsize='x-large')
plt.legend(fontsize='x-large')
```

Out[41]:

```
<matplotlib.legend.Legend at 0x23b57dc8a90>
```

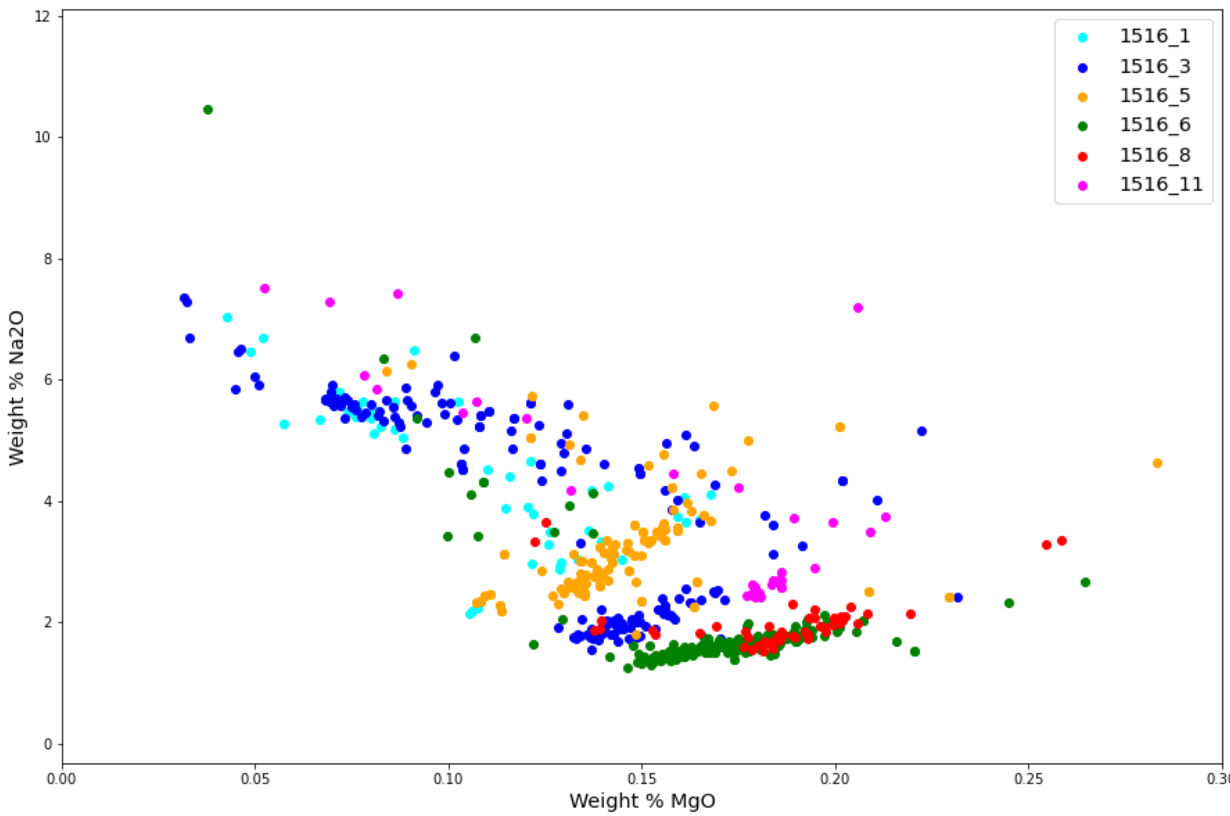


In [42]:

```
plt.figure(figsize=(15,10))
for i in ['1516_1','1516_3','1516_5','1516_6','1516_8','1516_11']:
    sample=all_pcrysts[all_pcrysts['Sample']==i]
    plt.scatter(sample['MgO'],sample['Na2O'],c=color_dict[i],label=i)
plt.xlabel('Weight % MgO',fontsize='x-large')
plt.ylabel('Weight % Na2O',fontsize='x-large')
plt.xlim(0,.3)
plt.legend(fontsize='x-large')
```

Out[42]:

<matplotlib.legend.Legend at 0x23b5977c8b0>

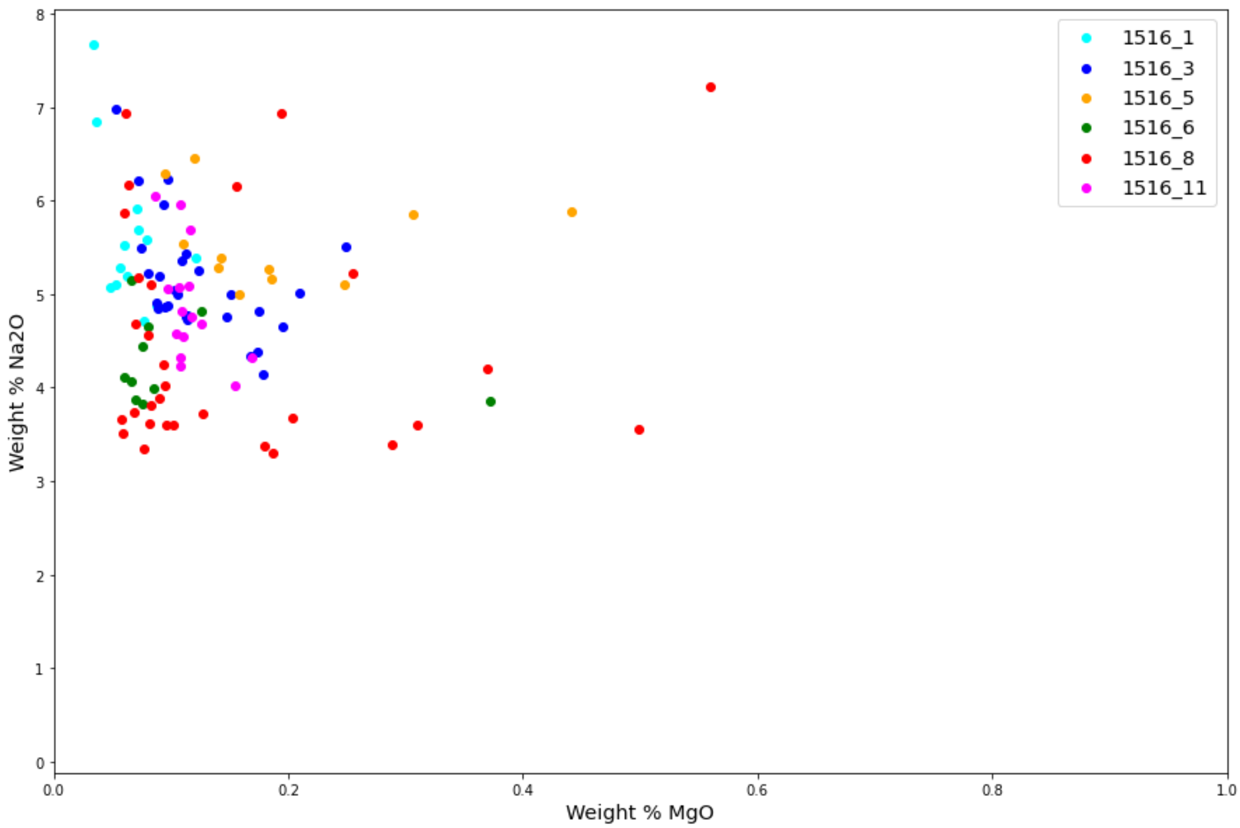


In [43]:

```
plt.figure(figsize=(15,10))
for i in ['1516_1','1516_3','1516_5','1516_6','1516_8','1516_11']:
    sample=g_plag[g_plag['Sample']==i]
    plt.scatter(sample['MgO'],sample['Na2O'],c=color_dict[i],label=i)
plt.xlabel('Weight % MgO',fontsize='x-large')
plt.ylabel('Weight % Na2O',fontsize='x-large')
plt.xlim(0,1)
plt.legend(fontsize='x-large')
```

Out[43]:

```
<matplotlib.legend.Legend at 0x23b59802a60>
```

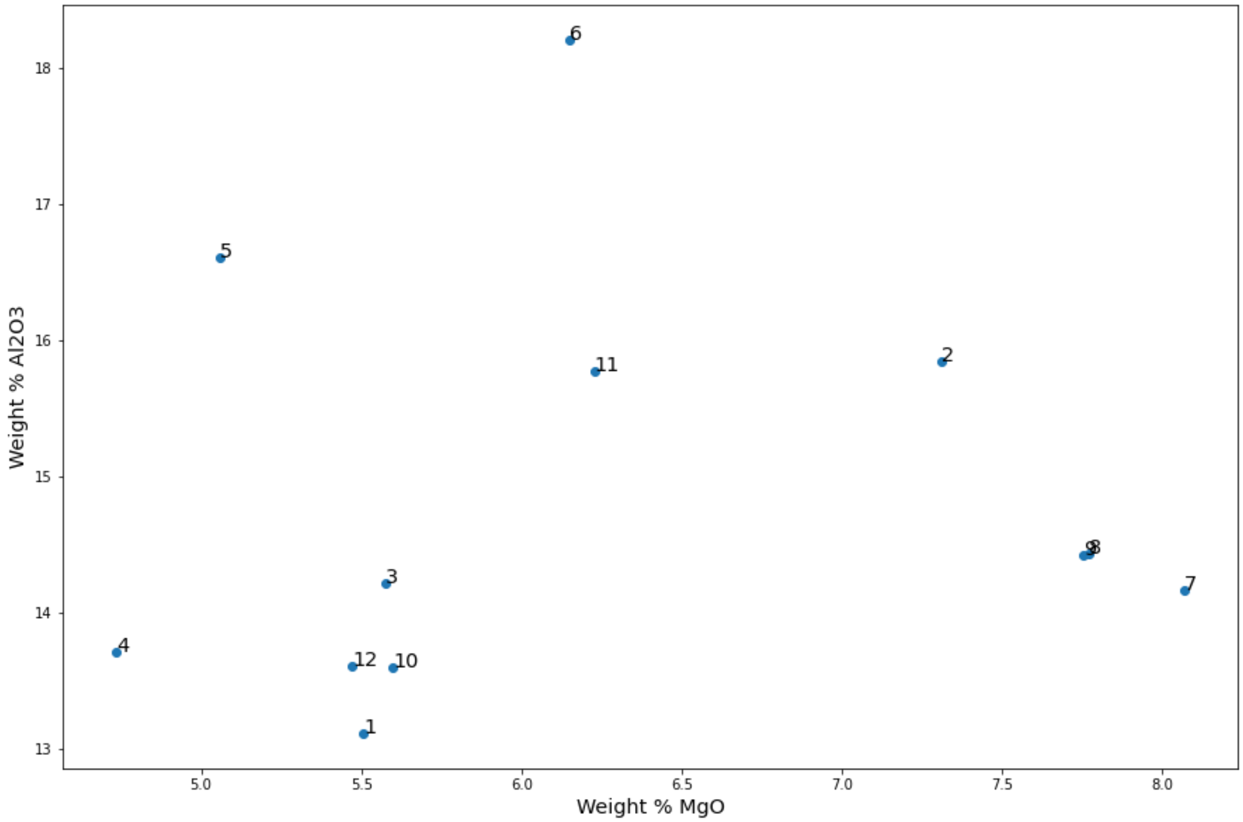


In [44]:

```
all_samples=['1516_1', '1516_2', '1516_3', '1516_4','1516_5', '1516_6', '1516_7', '1516_8',
         '1516_9','1516_10', '1516_11', '1516_12']
plt.figure(figsize=(15,10))
plt.scatter(xrf_means['MgO'],xrf_means['Al2O3'])
for i in range(12):

plt.text(xrf_means.loc[all_samples[i],'MgO'],xrf_means.loc[all_samples[i],'Al2O3'],str(i+1),fontsize='x-large')
plt.xlabel('Weight % MgO',fontsize='x-large')
plt.ylabel('Weight % Al2O3',fontsize='x-large')
```

Out[44]:

```
Text(0, 0.5, 'Weight % Al2O3')
```



In [55]:

```
print('Avg plag MgO = '+str(pcrysts['MgO'].mean()))
print('Avg plag AL2O3 = '+str(pcrysts['Al2O3'].mean()))
print('Avg cpx MgO = '+str(xcrysts['MgO'].mean()))
print('Avg cpx Al2O3 = '+str(xcrysts['Al2O3'].mean()))
```

```
Avg plag MgO = 0.3639004484304933
Avg plag AL2O3 = 30.095021076233184
Avg cpx MgO = 15.548603703703705
Avg cpx Al2O3 = 4.1842
```

Discussion

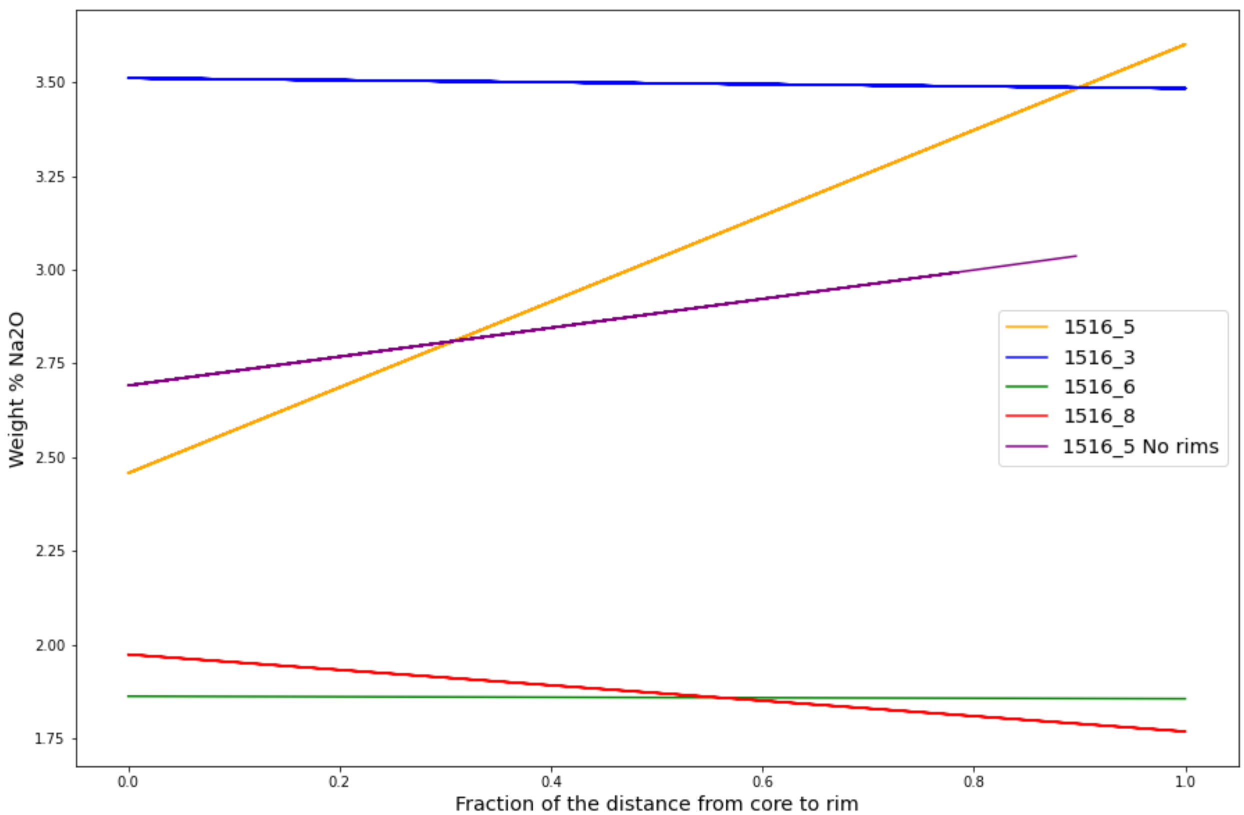
In [45]:

```
trans5=plag_transects[plag_transects['Sample']=='1516_5']
trans5_comments=trans5['Comment'].unique()
trans5_exrims=pd.DataFrame()
for i in trans5_comments:
    trans5_exrims=trans5_exrims.append(trans5[trans5['Comment']==i][:-3])

plt.figure(figsize=(15,10))
```

```
for i in plag_transects['Sample'].unique():
    sample=plag_transects[plag_transects['Sample']==i]
    x=sample['frac_dist'].tolist()
    y=sample['Na2O'].tolist()
    trendline(x,y,z=color_dict[i],a=i)
x=trans5_exrims['frac_dist'].tolist()
y=trans5_exrims['Na2O'].tolist()
trendline(x,y,z='purple',a='1516_5 No rims')
plt.xlabel('Fraction of the distance from core to rim',fontsize='x-large')
plt.ylabel('Weight % Na2O',fontsize='x-large')
plt.legend(fontsize='x-large')
```

Out[45]:

<matplotlib.legend.Legend at 0x23b57dbd5e0>



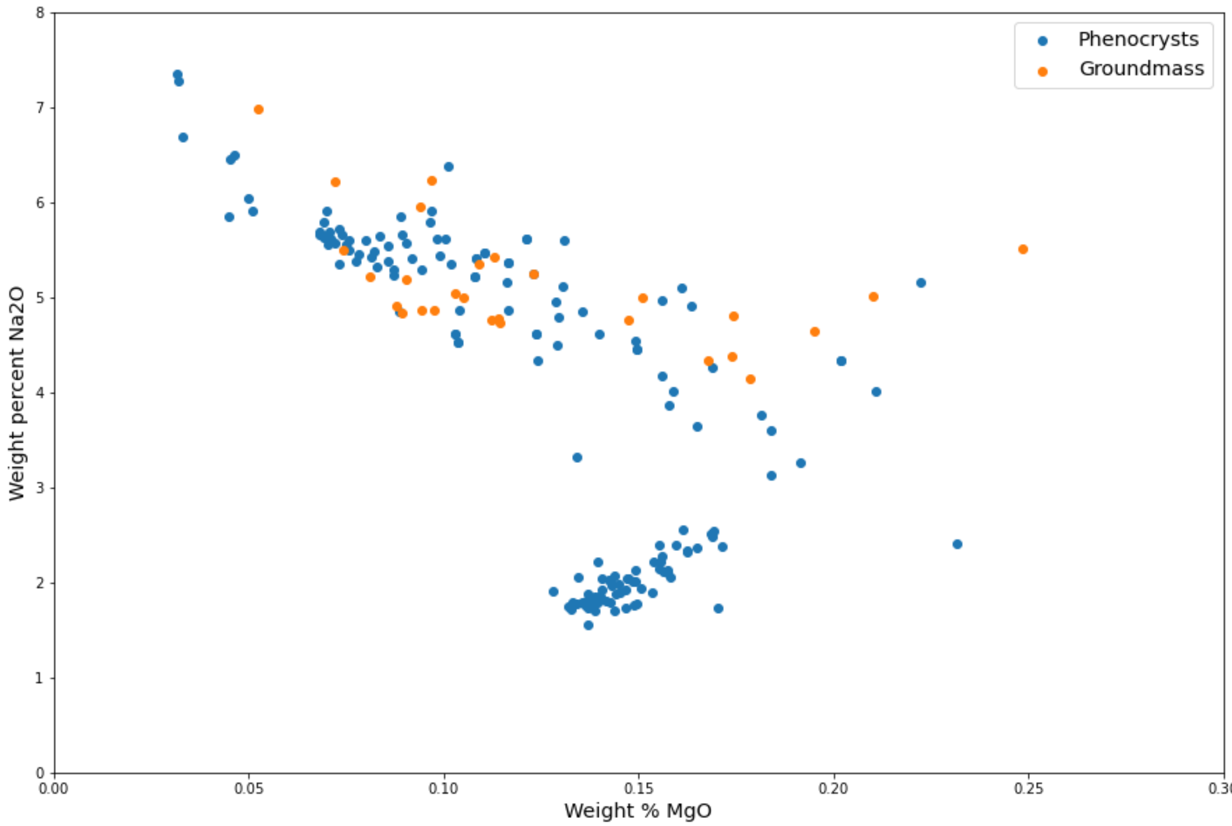
In [47]:

```
plt.figure(figsize=(15,10))
i='1516_3'
sample=all_pcrysts[all_pcrysts['Sample']==i]
plt.scatter(sample['MgO'],sample['Na2O'],label='Phenocrysts')

g_sample=g_plag[g_plag['Sample']==i]
plt.scatter(g_sample['MgO'],g_sample['Na2O'],label='Groundmass')
```

```
plt.xlim(0,.3)
plt.ylim(0,8)
plt.xlabel('Weight % MgO',fontsize='x-large')
plt.ylabel('Weight percent Na2O',size='x-large')
plt.legend(fontsize='x-large')
```

Out[47]:

<matplotlib.legend.Legend at 0x23b57a90cd0>



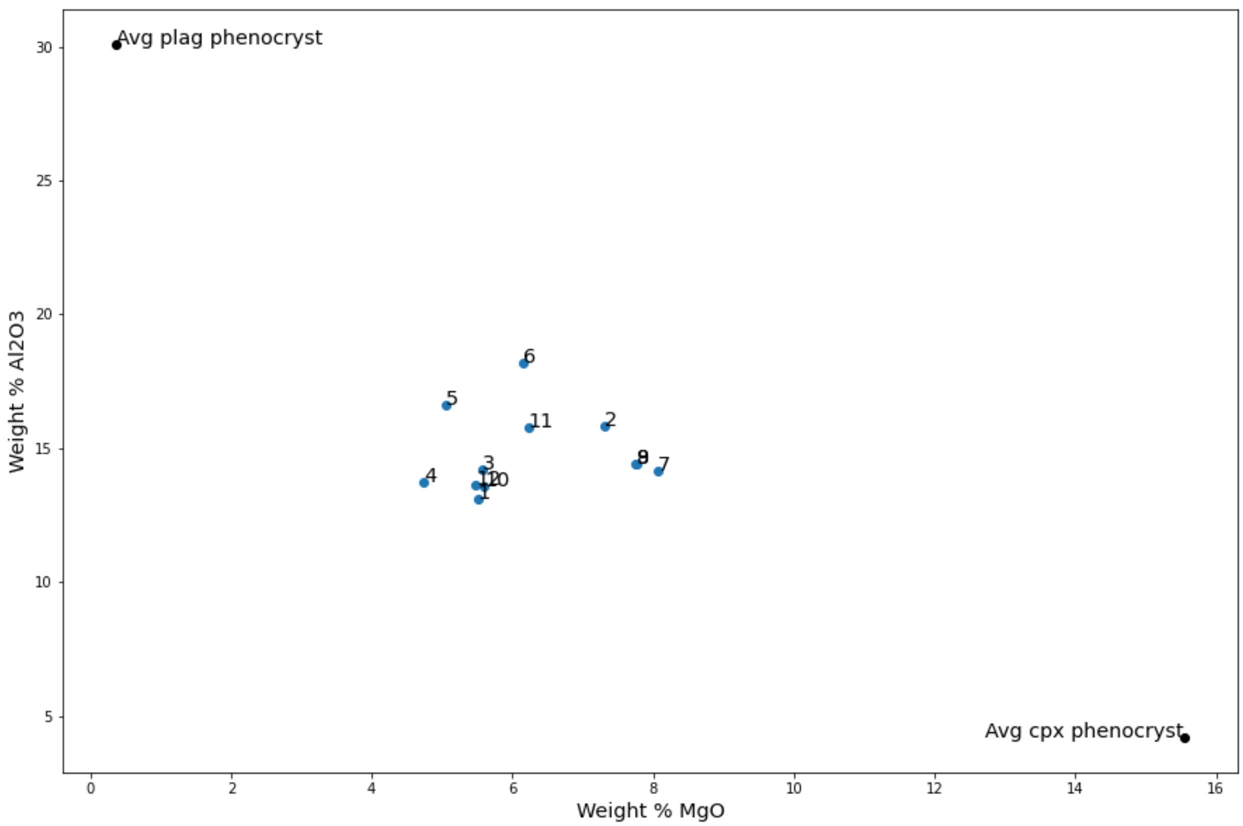
In [48]:

```
plt.figure(figsize=(15,10))
plt.scatter(xrf_means['MgO'],xrf_means['Al2O3'])
for i in range(12):

plt.text(xrf_means.loc[all_samples[i],'MgO'],xrf_means.loc[all_samples[i],'Al2O3'],str(i+1),fontsize='x-l
arge')

plt.scatter(pcrysts['MgO'].mean(),pcrysts['Al2O3'].mean(),c='black')
plt.text(pcrysts['MgO'].mean(),pcrysts['Al2O3'].mean(),'Avg plag phenocryst',fontsize='x-large')

plt.scatter(xcrysts['MgO'].mean(),xcrysts['Al2O3'].mean(),c='black')
```

```
plt.text(xcrysts['MgO'].mean(),xcrysts['Al2O3'].mean(),'Avg cpx
phenocryst',fontsize='x-large',ha='right')



plt.xlabel('Weight % MgO',fontsize='x-large')
plt.ylabel('Weight % Al2O3',fontsize='x-large')
```

Out[48]:

```
Text(0, 0.5, 'Weight % Al2O3')
```



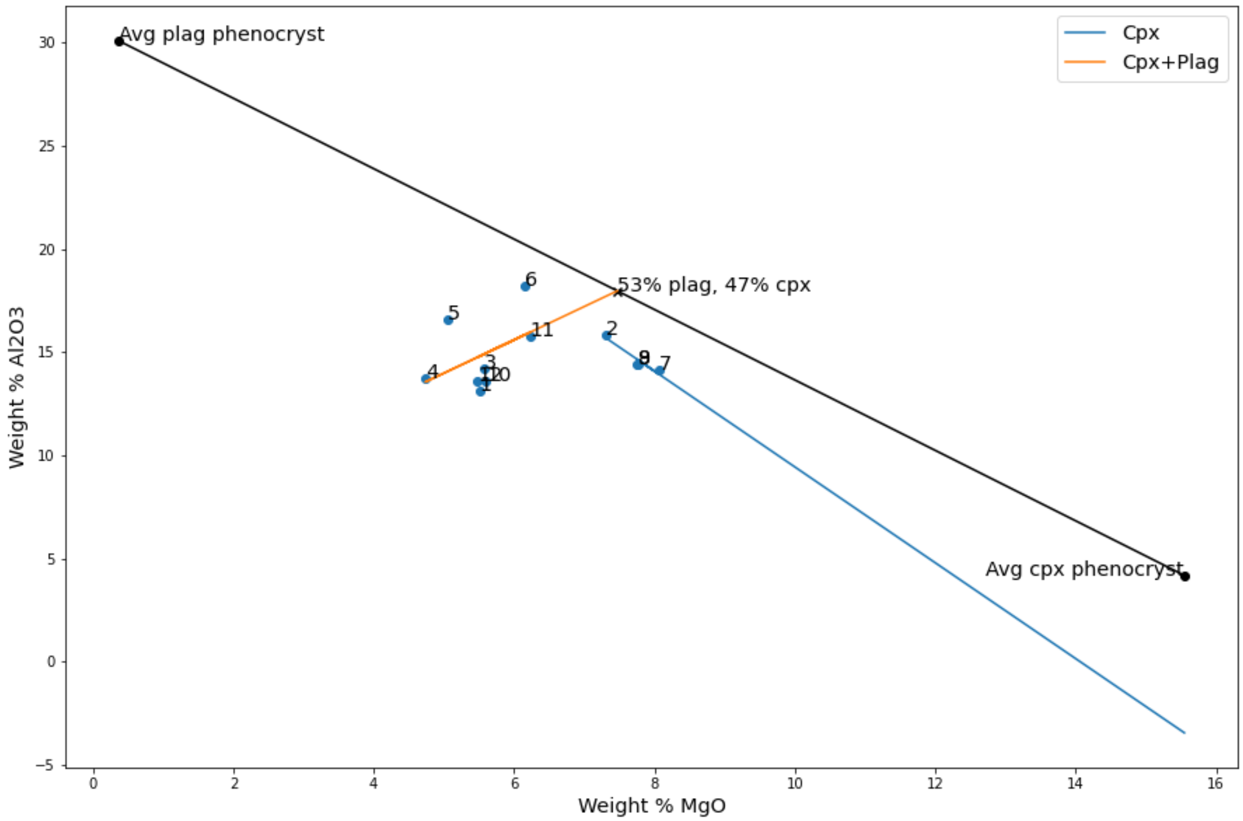
In [49]:

```
plt.figure(figsize=(15,10))
plt.scatter(xrf_means['MgO'],xrf_means['Al2O3'])
for i in range(12):

plt.text(xrf_means.loc[all_samples[i],'MgO'],xrf_means.loc[all_samples[i],'Al2O3'],str(i+1),fontsize='x-l
arge')

plt.scatter(pcrysts['MgO'].mean(),pcrysts['Al2O3'].mean(),c='black')
plt.text(pcrysts['MgO'].mean(),pcrysts['Al2O3'].mean(),'Avg plag phenocryst',fontsize='x-large')

plt.scatter(xcrysts['MgO'].mean(),xcrysts['Al2O3'].mean(),c='black')
```

```
plt.text(xcrysts['MgO'].mean(),xcrysts['Al2O3'].mean(),'Avg cpx
phenocryst',ha='right',fontsize='x-large')

#Cpx
x=xrf_means.loc[['1516_2','1516_7','1516_8','1516_9'],'MgO']
y=xrf_means.loc[['1516_2','1516_7','1516_8','1516_9'],'Al2O3']
#trendline(x,y,z='blue',a='456')
result=stats.linregress(x,y)
x=np.array(x.tolist()+[xcrysts['MgO'].mean()])
plt.plot(x,(result.slope*x)+result.intercept,label='Cpx')

#plt.scatter(xcrysts['MgO'].mean(),(result.slope*xcrysts['MgO'].mean())+result.intercept,marker='x',c='
black')
#plt.text(xcrysts['MgO'].mean(),(result.slope*xcrysts['MgO'].mean())+result.intercept,'Cpx',fontsize='x-
large')

plt.plot([pcrysts['MgO'].mean(),xcrysts['MgO'].mean()],[pcrysts['Al2O3'].mean(),xcrysts['Al2O3'].mean
()],c='black')


#Mix
x=xrf_means.loc[['1516_1','1516_3','1516_6','1516_5','1516_4','1516_10','1516_11','1516_12'],'MgO']
y=xrf_means.loc[['1516_1','1516_3','1516_6','1516_5','1516_4','1516_10','1516_11','1516_12'],'Al2O
3']
#trendline(x,y,z='blue',a='456')
result1=stats.linregress(x,y)
#x=np.array(x.tolist()+[])


#Intersection
result2=stats.linregress([pcrysts['MgO'].mean(),xcrysts['MgO'].mean()],[pcrysts['Al2O3'].mean(),xcrys
ts['Al2O3'].mean()])
intersect=(result1.intercept-result2.intercept)/(result2.slope-result1.slope)
plt.scatter(intersect,(result1.slope*intersect)+result1.intercept,marker='x',c='black')

pct=round(((xcrysts['MgO'].mean()-intersect)/(xcrysts['MgO'].mean()-pcrysts['MgO'].mean()))*100)
plt.text(intersect,(result1.slope*intersect)+result1.intercept,str(pct)+'% plag, '+str(100-pct)+'%
cpx',fontsize='x-large')

x=np.array(x.tolist()+[intersect])
plt.plot(x,(result1.slope*x)+result1.intercept,label='Cpx+Plag')

plt.xlabel('Weight % MgO',fontsize='x-large')
plt.ylabel('Weight % Al2O3',fontsize='x-large')

plt.legend(fontsize='x-large')
```

Out[49]:

<matplotlib.legend.Legend at 0x23b5a50eeb0>



In [50]:

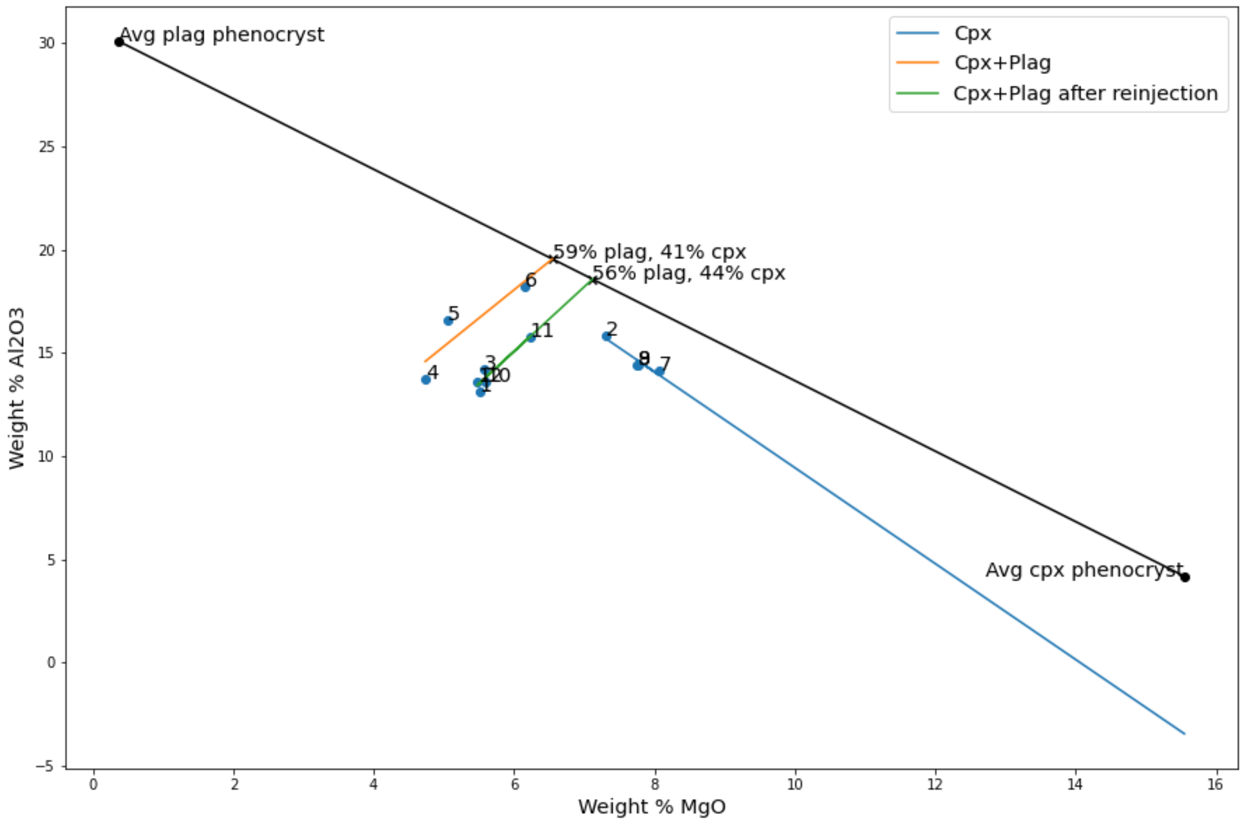
```
plt.figure(figsize=(15,10))
plt.scatter(xrf_means['MgO'],xrf_means['Al2O3'])
for i in range(12):

plt.text(xrf_means.loc[all_samples[i],'MgO'],xrf_means.loc[all_samples[i],'Al2O3'],str(i+1),fontsize='x-large')

plt.scatter(pcrysts['MgO'].mean(),pcrysts['Al2O3'].mean(),c='black')
plt.text(pcrysts['MgO'].mean(),pcrysts['Al2O3'].mean(),'Avg plag phenocryst',fontsize='x-large')

plt.scatter(xcrysts['MgO'].mean(),xcrysts['Al2O3'].mean(),c='black')
plt.text(xcrysts['MgO'].mean(),xcrysts['Al2O3'].mean(),'Avg cpx
phenocryst',fontsize='x-large',ha='right')

#cpx line
x=xrf_means.loc[['1516_2','1516_7','1516_8','1516_9'],'MgO']
y=xrf_means.loc[['1516_2','1516_7','1516_8','1516_9'],'Al2O3']
result=stats.linregress(x,y)
x=np.array(x.tolist()+[xcrysts['MgO'].mean()])
```

```
plt.plot(x,(result.slope*x)+result.intercept,label='Cpx')
#plt.scatter(xcrysts['MgO'].mean(),(result.slope*xcrysts['MgO'].mean())+result.intercept,marker='x',c='
black')
#plt.text(xcrysts['MgO'].mean(),(result.slope*xcrysts['MgO'].mean())+result.intercept,'Cpx',fontsize='x-
large')


plt.plot([pcrysts['MgO'].mean(),xcrysts['MgO'].mean()],[pcrysts['Al2O3'].mean(),xcrysts['Al2O3'].mean
()],c='black')

#First line
x=xrf_means.loc[['1516_4','1516_5','1516_6'],'MgO']
y=xrf_means.loc[['1516_4','1516_5','1516_6'],'Al2O3']
result1=stats.linregress(x,y)

#Calculate intersection point
result2=stats.linregress([pcrysts['MgO'].mean(),xcrysts['MgO'].mean()],[pcrysts['Al2O3'].mean(),xcrys
ts['Al2O3'].mean()])
intersect=(result1.intercept-result2.intercept)/(result2.slope-result1.slope)
plt.scatter(intersect,(result1.slope*intersect)+result1.intercept,marker='x',c='black')

pct=round(((xcrysts['MgO'].mean()-intersect)/(xcrysts['MgO'].mean()-pcrysts['MgO'].mean()))*100)
plt.text(intersect,(result1.slope*intersect)+result1.intercept,str(pct)+'% plag, '+str(100-pct)+'%
cpx',fontsize='x-large')

x=np.array(x.tolist()+[intersect])
plt.plot(x,(result1.slope*x)+result1.intercept,label='Cpx+Plag')

#Second line
x=xrf_means.loc[['1516_1','1516_3','1516_10','1516_11','1516_12'],'MgO']
y=xrf_means.loc[['1516_1','1516_3','1516_10','1516_11','1516_12'],'Al2O3']
result1=stats.linregress(x,y)

#Calculate intersection point
result2=stats.linregress([pcrysts['MgO'].mean(),xcrysts['MgO'].mean()],[pcrysts['Al2O3'].mean(),xcrys
ts['Al2O3'].mean()])
intersect=(result1.intercept-result2.intercept)/(result2.slope-result1.slope)
plt.scatter(intersect,(result1.slope*intersect)+result1.intercept,marker='x',c='black')

pct=round(((xcrysts['MgO'].mean()-intersect)/(xcrysts['MgO'].mean()-pcrysts['MgO'].mean()))*100)
plt.text(intersect,(result1.slope*intersect)+result1.intercept,str(pct)+'% plag, '+str(100-pct)+'%
cpx',fontsize='x-large')

x=np.array(x.tolist()+[intersect])
plt.plot(x,(result1.slope*x)+result1.intercept,label='Cpx+Plag after reinjection')
```

```
plt.xlabel('Weight % MgO',fontsize='x-large')
plt.ylabel('Weight % Al2O3',fontsize='x-large')
plt.legend(fontsize='x-large')
```

Out[50]:

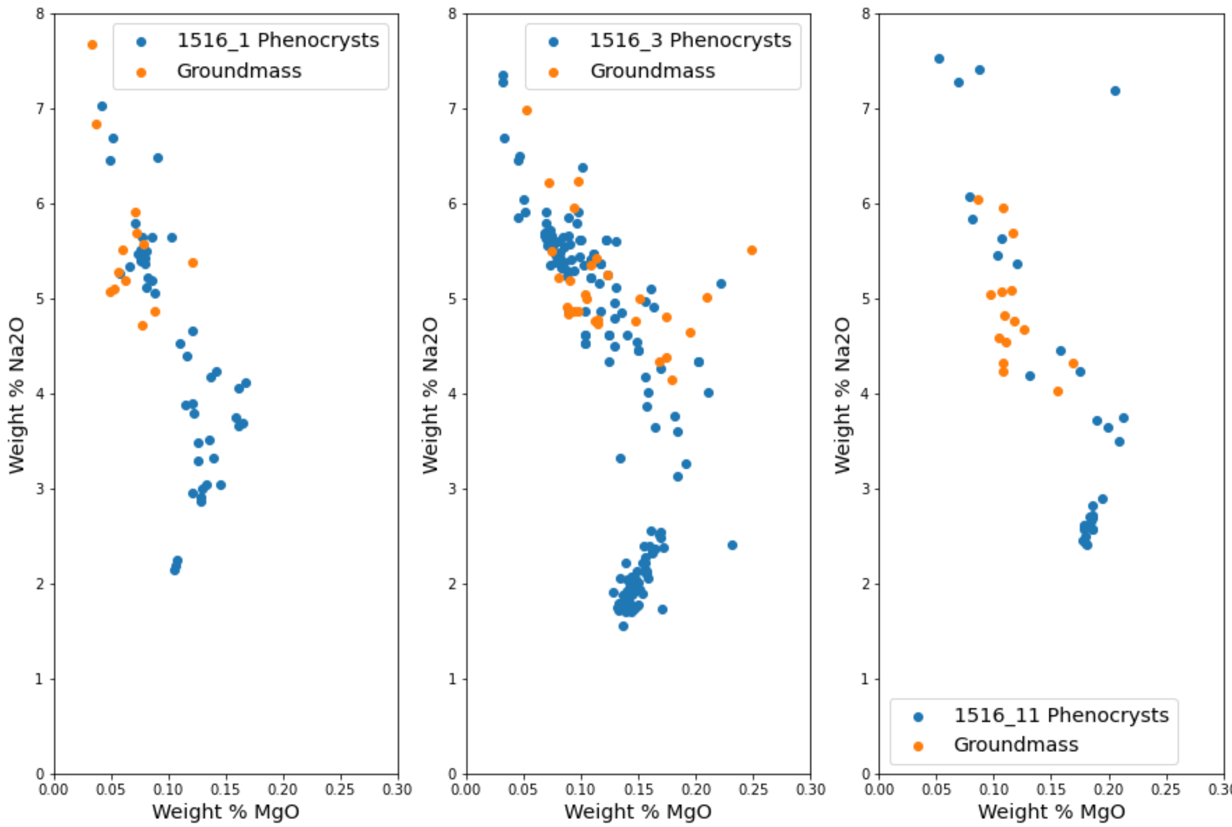
```
<matplotlib.legend.Legend at 0x23b5ac13c70>
```



In [51]:

```
plt.figure(figsize=(15,10))
index=1
for i in ['1516_1','1516_3','1516_11']:
    plt.subplot(1,3,index)
    sample=all_pcrysts[all_pcrysts['Sample']==i]
    plt.scatter(sample['MgO'],sample['Na2O'],label=i+' Phenocrysts')

    g_sample=g_plag[g_plag['Sample']==i]
    plt.scatter(g_sample['MgO'],g_sample['Na2O'],label='Groundmass')

    plt.xlim(0,.3)
    plt.ylim(0,8)
    plt.xlabel('Weight % MgO',fontsize='x-large')
    plt.ylabel('Weight % Na2O',fontsize='x-large')
    plt.legend(fontsize='x-large')
```

```
index+=1
```



1516_1 Phenocrysts
Groundmass
Weight % Na2O
Weight % MgO
1516_3 Phenocrysts
Groundmass
Weight % Na2O
Weight % MgO
1516_11 Phenocrysts
Groundmass
Weight % Na2O
Weight % MgO


```
In [ ]:
```