CerroNegro_isobar_comparison

August 20, 2020

[14]:
```
import sys
sys.path.append('../../..')

import VESIcal as v
import numpy as np
import scipy
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
#Import the data
basalts = v.ExcelFile("../Datasets/cerro_negro.xlsx")

#Calculate the average composition of the entire dataset
columns = list(basalts.data)
avg_vals = []
for col in columns:
    try:
        avg_vals.append(basalts.data[col].mean())
    except:
        avg_vals.append("AVG")

avg_dict = dict(zip(columns, avg_vals))
avg_dict = v.get_oxides(avg_dict)
```

[3]:
```
#Calculate isobars for all samples at 3,000 bars
isobar_list = []
for index, row in basalts.data.iterrows():
    isobar_list.append(v.calculate_isobars_and_isopleths(sample=basalts.
 ↪get_sample_oxide_comp(sample=row.name), temperature=1200,␣
 ↪pressure_list=[3000], isopleth_list=[0.5], print_status=True).result[0])
```

```
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
```

```
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
```

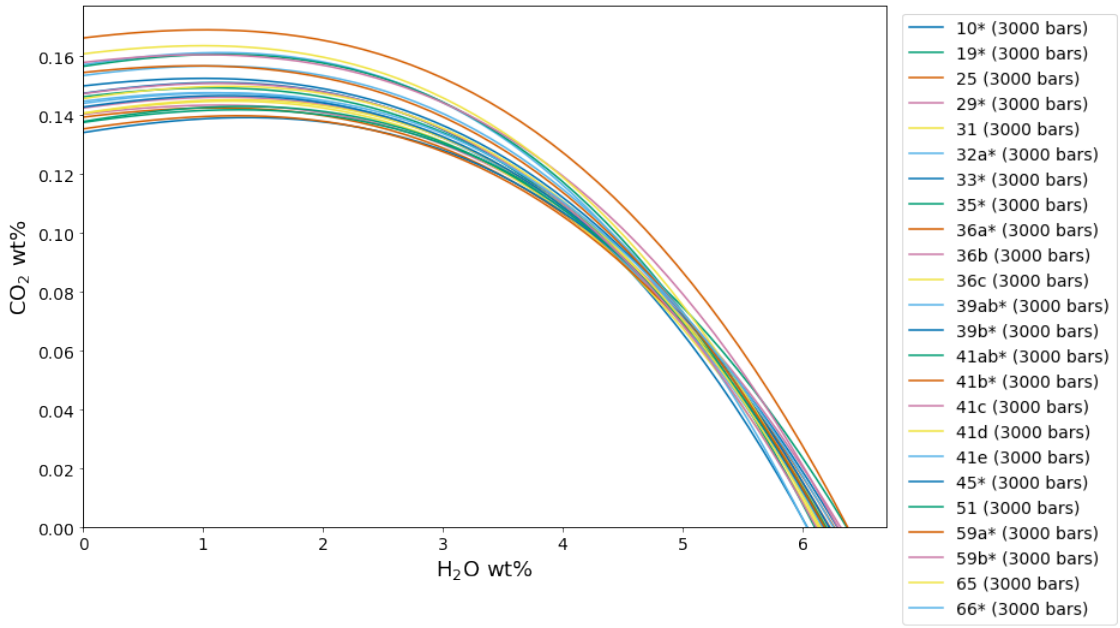
```
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
```

```
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
```

[4]:
```
#Calculate isobar at 3,000 bars for "Average Sample"
avg_isobar = v.calculate_isobars_and_isopleths(sample=avg_dict,␣
 ↪temperature=1200, pressure_list=[3000], isopleth_list=[0.5],␣
 ↪print_status=True).result[0]
```

```
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
```

[5]:
```
#Plot all isobars from dataset
v.plot(isobars=[isobar for isobar in isobar_list], isobar_labels=[row.name for␣
 ↪index, row in basalts.data.iterrows()])
```



[6]:
```
#calculate area under each curve for dataset and "Average Sample"
areas = []
samples = [row.name for index, row in basalts.data.iterrows()]
for isobar in isobar_list:
    x_vals = np.array([row["H2O_liq"] for index, row in isobar.iterrows()])
    y_vals = np.array([row["CO2_liq"] for index, row in isobar.iterrows()])
    area_under_the_curve = scipy.integrate.simps(y_vals, x_vals)
    areas.append(area_under_the_curve)

average_area = scipy.integrate.simps(avg_isobar['CO2_liq'],␣
↪avg_isobar['H2O_liq'])
```

[7]:
```
#Get maximum and minimum areas from dataset, with corresponding sample names
area_dict = dict(zip(samples, areas))
max_sample = max(area_dict, key=area_dict.get)
min_sample = min(area_dict, key=area_dict.get)
print("ISM values for entire dataset: \n" + str(area_dict) + "\n")
print("'Average Sample' ISM = " + str(average_area))
```

```
ISM values for entire dataset:
{'10*': 0.6909012678340266, '19*': 0.6983090849381166, 25: 0.6745901941880883,
'29*': 0.6845648808020042, 31: 0.6747876361972984, '32a*': 0.6866730837091439,
'33*': 0.6686724403302219, '35*': 0.6871347773647076, '36a*':
0.6630503234496689, '36b': 0.688993347959344, '36c': 0.6855952161196897,
'39ab*': 0.7376362081664589, '39b*': 0.7134546407478359, '41ab*':
0.7429259382387134, '41b*': 0.8097464284301792, '41c': 0.7593637639098455,
'41d': 0.7580364872981887, '41e': 0.7292831376608033, '45*': 0.6876563252571991,
```

```
51: 0.6763410845036275, '59a*': 0.7270460927699788, '59b*': 0.6976077833589145,
65: 0.6994065845180091, '66*': 0.6990246053885986}

'Average Sample' ISM = 0.7043729373794232
```

[8]:
```
#Now, calculate isobars for the max and min samples at multiple pressures
max_isobars, max_isopleths = v.calculate_isobars_and_isopleths(sample=basalts.
 ↪get_sample_oxide_comp(max_sample), temperature=1200, pressure_list=[500],␣
 ↪isopleth_list=[], print_status=True).result
min_isobars, min_isopleths = v.calculate_isobars_and_isopleths(sample=basalts.
 ↪get_sample_oxide_comp(min_sample), temperature=1200, pressure_list=[500],␣
 ↪isopleth_list=[], print_status=True).result

#Calculate isobars for the average composition
avg_isobars, avg_isopleths = v.calculate_isobars_and_isopleths(sample=avg_dict,␣
 ↪temperature=1200, pressure_list=[500], isopleth_list=[], print_status=True).
 ↪result
```

```
Calculating isobar at 500 bars
Calculating isopleth at 0
Calculating isopleth at 1
Done!
Calculating isobar at 500 bars
Calculating isopleth at 0
Calculating isopleth at 1
Done!
Calculating isobar at 500 bars
Calculating isopleth at 0
Calculating isopleth at 1
Done!
```

[9]:
```
#Now, calculate isobars for the max and min samples at multiple pressures
max_isobars, max_isopleths = v.calculate_isobars_and_isopleths(sample=basalts.
 ↪get_sample_oxide_comp(max_sample), temperature=1200, pressure_list=[500,␣
 ↪1000, 2000, 3000, 4000], isopleth_list=[0.5], print_status=True).result
min_isobars, min_isopleths = v.calculate_isobars_and_isopleths(sample=basalts.
 ↪get_sample_oxide_comp(min_sample), temperature=1200, pressure_list=[500,␣
 ↪1000, 2000, 3000, 4000], isopleth_list=[0.5], print_status=True).result

#Calculate isobars for the average composition
avg_isobars, avg_isopleths = v.calculate_isobars_and_isopleths(sample=avg_dict,␣
 ↪temperature=1200, pressure_list=[500, 1000, 2000, 3000, 4000],␣
 ↪isopleth_list=[0.5], print_status=True).result
```

```
Calculating isobar at 500 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
```

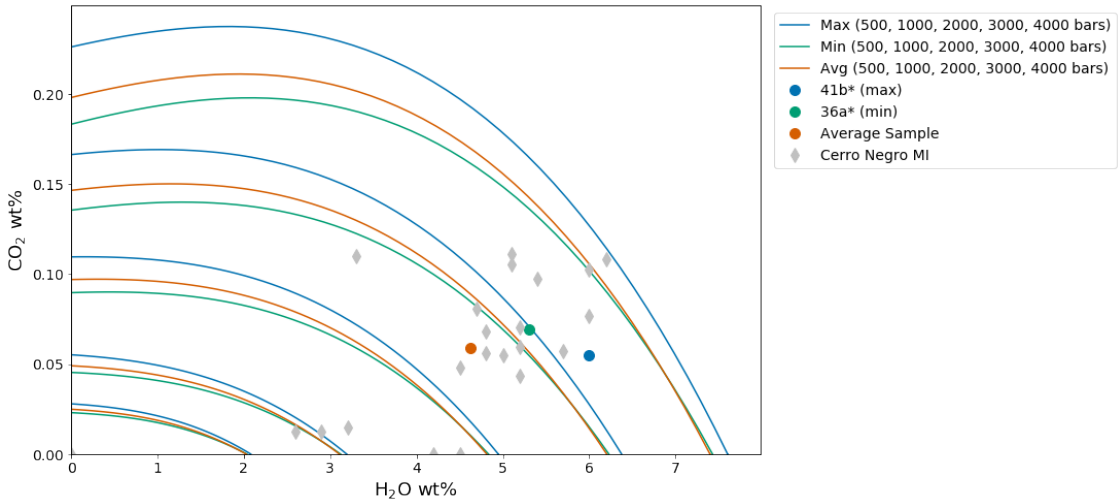
```
Calculating isobar at 1000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 2000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 4000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 500 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 1000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 2000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 4000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
Calculating isobar at 500 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 1000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 2000 bars
Calculating isopleth at 0
```

```
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 3000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Calculating isobar at 4000 bars
Calculating isopleth at 0
Calculating isopleth at 0.5
Calculating isopleth at 1
Done!
```

[10]:
```
#Make dataset with all data except for max and min values
other_data = basalts.data.drop([max_sample, min_sample])
```

[11]:
```
#set up what to pass to v.plot
isobars = [max_isobars,
           min_isobars,
           avg_isobars]

isobar_labels = ["Max",
                 "Min",
                 "Avg"]

custom_H2O=[basalts.get_sample_oxide_comp(max_sample)["H2O"],
            basalts.get_sample_oxide_comp(min_sample)["H2O"],
            avg_dict["H2O"],
            other_data["H2O"]]

custom_CO2=[basalts.get_sample_oxide_comp(max_sample)["CO2"],
            basalts.get_sample_oxide_comp(min_sample)["CO2"],
            avg_dict["CO2"],
            other_data["CO2"]]

custom_labels = [str(max_sample) + " (max)",
                 str(min_sample) + " (min)",
                 "Average Sample",
                 "Cerro Negro MI"]

custom_colors = [v.color_list[0],
                 v.color_list[1],
                 v.color_list[2],
                 'silver']

custom_symbols = ['o',
                  'o',
                  'o',
```

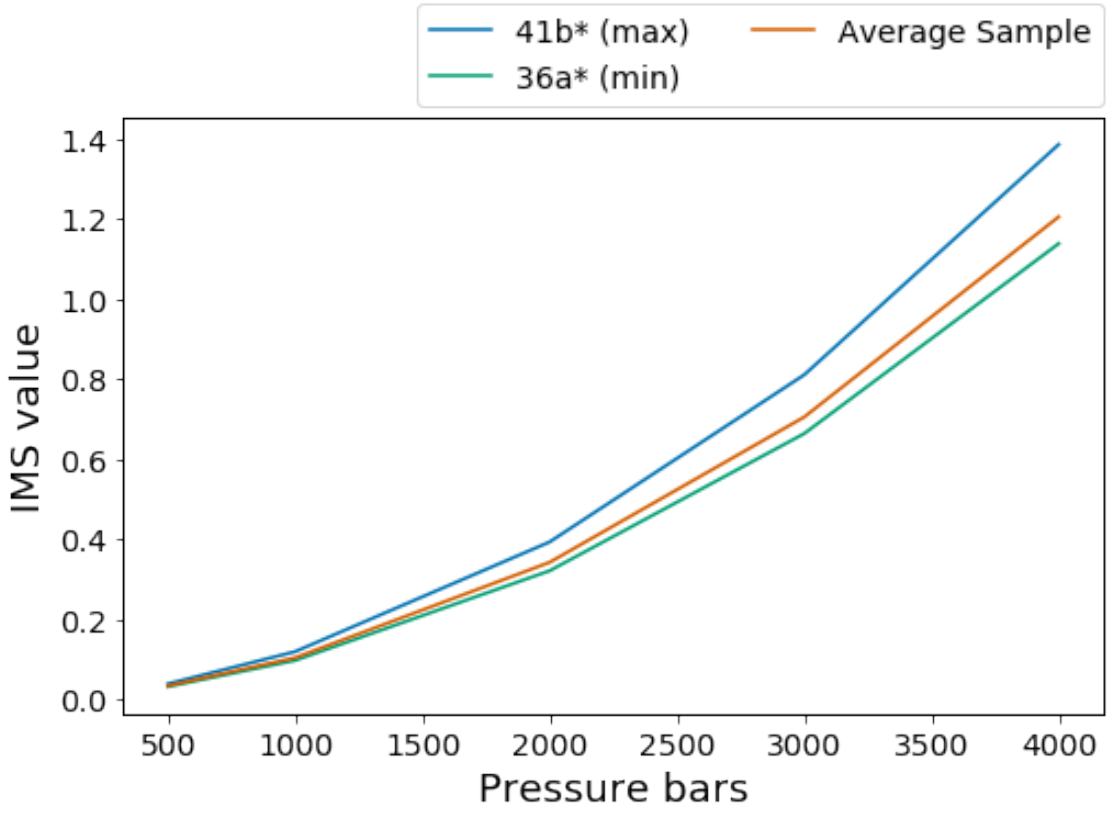
```
                  'd']

v.plot(isobars=isobars, isobar_labels=isobar_labels,
       custom_H2O=custom_H2O,
       custom_CO2=custom_CO2,
       custom_labels=custom_labels,
       custom_colors=custom_colors,
       custom_symbols=custom_symbols)
```



[15]:
```
pressure_vals = [500, 1000, 2000, 3000, 4000]

max_IMS_dict = {}
min_IMS_dict = {}
avg_IMS_dict = {}

IMS_dicts = [max_IMS_dict,
             min_IMS_dict,
             avg_IMS_dict]

for i in range(len(isobars)):
    IMS_dicts[i].update({"Pressure": pressure_vals})
    IMS_list = []
    for pressure in pressure_vals:
        IMS_list.append(scipy.integrate.simps(isobars[i].
 ↪loc[isobars[i]['Pressure']==pressure]["CO2_liq"], isobars[i].
 ↪loc[isobars[i]['Pressure']==pressure]["H2O_liq"]))
        IMS_dicts[i].update({"IMS": IMS_list})


labels = ["Maximum, Minimum, Average"]
```

```
fig, ax = plt.subplots(1, figsize = (8,5))

for i in range(len(IMS_dicts)):
    ax.plot(IMS_dicts[i]["Pressure"], IMS_dicts[i]["IMS"],␣
 ↪label=custom_labels[i])
    ax.set_xlabel("Pressure bars")
    ax.set_ylabel("IMS value")

ax.legend(bbox_to_anchor=(0., 1.02, 1., .102), loc='lower right',
           ncol=2, borderaxespad=0.)
```

[15]: <matplotlib.legend.Legend at 0x1a2aa3ef50>

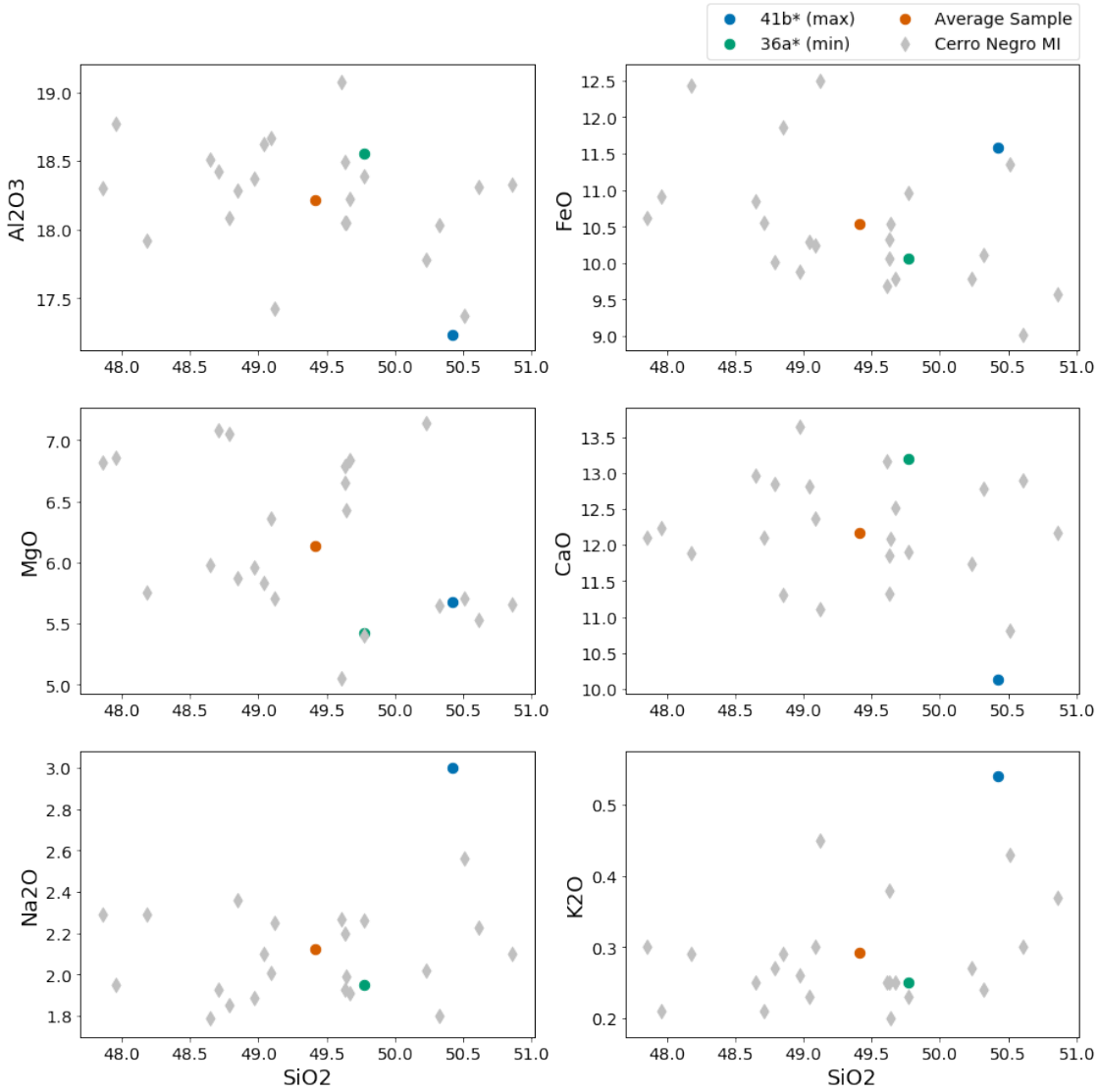


[16]:
```
other_oxides = ["Al2O3", "FeO", "MgO", "CaO", "Na2O", "K2O"]
my_samples = [basalts.get_sample_oxide_comp(max_sample),
              basalts.get_sample_oxide_comp(min_sample),
             avg_dict,
             other_data]

fig, axs = plt.subplots(3,2, figsize = (15,15))
```

```
print(len(axs))

for j in range(len(my_samples)):
    axs[0][0].scatter(my_samples[j]["SiO2"], my_samples[j]["Al2O3"],␣
 ↪marker=custom_symbols[j], s=70, color=custom_colors[j],␣
 ↪label=custom_labels[j])
    axs[0][0].set_ylabel("Al2O3")
    axs[0][1].scatter(my_samples[j]["SiO2"], my_samples[j]["FeO"],␣
 ↪marker=custom_symbols[j], s=70, color=custom_colors[j],␣
 ↪label=custom_labels[j])
    axs[0][1].set_ylabel("FeO")
    axs[1][0].scatter(my_samples[j]["SiO2"], my_samples[j]["MgO"],␣
 ↪marker=custom_symbols[j], s=70, color=custom_colors[j],␣
 ↪label=custom_labels[j])
    axs[1][0].set_ylabel("MgO")
    axs[1][1].scatter(my_samples[j]["SiO2"], my_samples[j]["CaO"],␣
 ↪marker=custom_symbols[j], s=70, color=custom_colors[j],␣
 ↪label=custom_labels[j])
    axs[1][1].set_ylabel("CaO")
    axs[2][0].scatter(my_samples[j]["SiO2"], my_samples[j]["Na2O"],␣
 ↪marker=custom_symbols[j], s=70, color=custom_colors[j],␣
 ↪label=custom_labels[j])
    axs[2][0].set_ylabel("Na2O")
    axs[2][0].set_xlabel("SiO2")
    axs[2][1].scatter(my_samples[j]["SiO2"], my_samples[j]["K2O"],␣
 ↪marker=custom_symbols[j], s=70, color=custom_colors[j],␣
 ↪label=custom_labels[j])
    axs[2][1].set_ylabel("K2O")
    axs[2][1].set_xlabel("SiO2")

axs[0][1].legend(bbox_to_anchor=(0., 1.02, 1., .102), loc='lower right',
           ncol=2, borderaxespad=0.)
```

```
3
```

[16]: <matplotlib.legend.Legend at 0x1a2adb9390>



1 Alternative plots

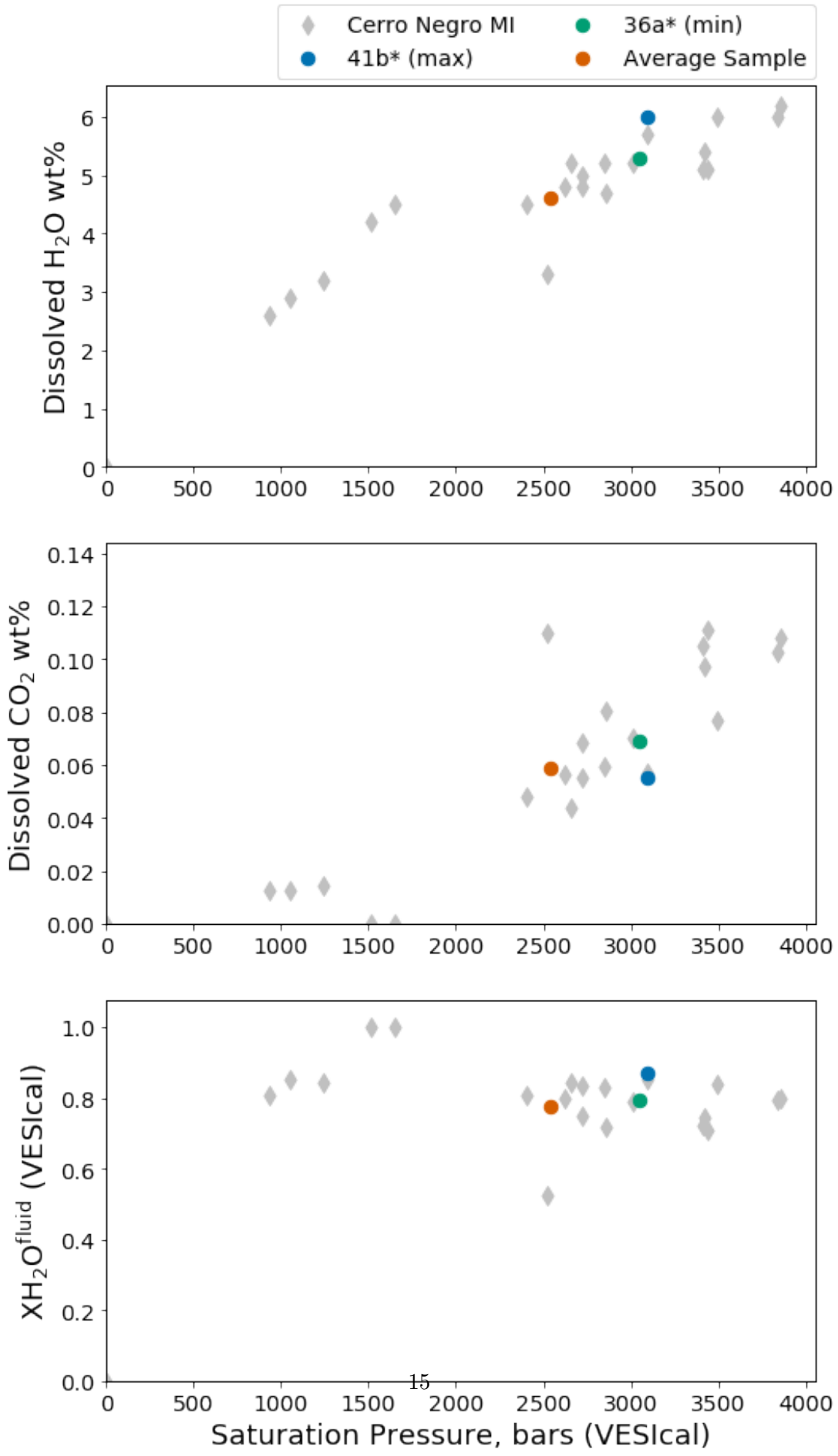
[17]:
```
#Calculate Saturation Pressure for all samples
other_file = v.ExcelFile(filename=None, dataframe=other_data)
satP_other = other_file.calculate_saturation_pressure(temperature=1200)
satP_max = v.calculate_saturation_pressure(sample=basalts.
 ↪get_sample_oxide_comp(max_sample), temperature=1200, verbose=True).result
satP_min = v.calculate_saturation_pressure(sample=basalts.
 ↪get_sample_oxide_comp(min_sample), temperature=1200, verbose=True).result
satP_avg = v.calculate_saturation_pressure(sample=avg_dict, temperature=1200,␣
 ↪verbose=True).result
```

```
Calculating sample 10*
Calculating sample 19*
Calculating sample 25
Calculating sample 29*
Calculating sample 31
Calculating sample 32a*
Calculating sample 33*
Calculating sample 35*
Calculating sample 36b
Calculating sample 36c
Calculating sample 39ab*
Calculating sample 39b*
Calculating sample 41ab*
Calculating sample 41c
Calculating sample 41d
Calculating sample 41e
Calculating sample 45*
Calculating sample 51
Calculating sample 59a*
Calculating sample 59b*
Calculating sample 65
Calculating sample 66*
Done!
```

[18]:
```
#Create alternative plots using Matplotlib
single_data = [satP_max,
               satP_min,
               satP_avg]

single_samples = [basalts.get_sample_oxide_comp(max_sample),
                  basalts.get_sample_oxide_comp(min_sample),
                  avg_dict]

fig, axs = plt.subplots(3, figsize = (8,15))
axs[0].scatter(satP_other["SaturationP_bars_VESIcal"], satP_other["H2O"],␣
 ↪marker=custom_symbols[3], s=70, color=custom_colors[3],␣
 ↪label=custom_labels[3])
axs[1].scatter(satP_other["SaturationP_bars_VESIcal"], satP_other["CO2"],␣
 ↪marker=custom_symbols[3], s=70, color=custom_colors[3],␣
 ↪label=custom_labels[3])
axs[2].scatter(satP_other["SaturationP_bars_VESIcal"],␣
 ↪satP_other["XH2O_fl_VESIcal"], marker=custom_symbols[3], s=70,␣
 ↪color=custom_colors[3], label=custom_labels[3])

for j in range(len(single_data)):
```

```
    axs[0].scatter(single_data[j]["SaturationP_bars"],␣
↪single_samples[j]["H2O"], marker=custom_symbols[j], s=70,␣
↪color=custom_colors[j], label=custom_labels[j])
    axs[0].set_ylabel("Dissolved H$_2$O wt%")
    axs[0].set_ylim(0)
    axs[0].set_xlim(0)
    axs[1].scatter(single_data[j]["SaturationP_bars"],␣
↪single_samples[j]["CO2"], marker=custom_symbols[j], s=70,␣
↪color=custom_colors[j], label=custom_labels[j])
    axs[1].set_ylabel("Dissolved CO$_2$ wt%")
    axs[1].set_ylim(0)
    axs[1].set_xlim(0)
    axs[2].scatter(single_data[j]["SaturationP_bars"],␣
↪single_data[j]["XH2O_fl"], marker=custom_symbols[j], s=70,␣
↪color=custom_colors[j], label=custom_labels[j])
    axs[2].set_ylabel("XH$_2$O$^{fluid}$ (VESIcal)")
    axs[2].set_ylim(0)
    axs[2].set_xlabel("Saturation Pressure, bars (VESIcal)")
    axs[2].set_xlim(0)

axs[0].legend(bbox_to_anchor=(0., 1.02, 1., .102), loc='lower right',
           ncol=2, borderaxespad=0.)
```

[18]: <matplotlib.legend.Legend at 0x1a29ae7b10>


Cerro Negro MI
36a* (min)
41b* (max)
Average Sample
Dissolved H_2O wt%
Dissolved CO_2 wt%
XH_2O^{fluid} (VESIcal)
Saturation Pressure, bars (VESIcal)

[28]:
```
#Create alternative plots using VESIcal's scatterplot() function
single_samples = [basalts.get_sample_oxide_comp(max_sample),
                  basalts.get_sample_oxide_comp(min_sample),
                  avg_dict]

v.scatterplot(custom_x=[satP_max['SaturationP_bars'],
                        satP_min['SaturationP_bars'],
                        satP_avg['SaturationP_bars'],
                       satP_other['SaturationP_bars_VESIcal']],
              custom_y=[single_samples[0]['H2O'],
                        single_samples[1]['H2O'],
                        single_samples[2]['H2O'],
                       satP_other['H2O']],
             custom_symbols=custom_symbols,
             custom_colors=custom_colors,
             custom_labels=custom_labels,
             xlabel="Saturation Pressure, bars (VESIcal)",
             ylabel="Dissolved H2O wt%")
```