RESEARCH METHODS VOLCANICA

DiadFit: An open-source Python3 tool for peak fitting of Raman data from silicate melts and volcanic fluids

Penny E. Wieser* and Charlotte L. DeVitre
Earth and Planetary Science, UC Berkeley, CA, USA.

Submitted: 2023-10-26 | Accepted: 2024-04-26 | Published: 2024-06-14

Editor: M. O. Chevrel | Typesetter: J. I. Farquharson

Abstract

We present DiadFit—an open-source Python3 tool for efficient processing of Raman spectroscopy data collected from fluid inclusions, melt inclusions and silicate melts. DiadFit is optimized to fit the characteristic peaks from CO_2 fluids (Fermi diads, hot bands, ^{13}C), gas species such as SO_2, N_2, solid precipitates (e.g. carbonates), and Ne emission lines with easily tweakable background positions and peak shapes. DiadFit's peak fitting functions are used as part of a number of workflows optimized for quantification of CO_2 in melt inclusion vapour bubbles and fluid inclusions. DiadFit can also convert between temperature, pressure and density using various CO_2 and CO_2-H_2O equations of state (EOS), allowing calculation of fluid inclusion pressures (and depths in the crust), conversion of homogenization temperatures from microthermometry to CO_2 density, and propagation of uncertainties associated with EOS calculations using Monte Carlo methods. There are also functions to quantify the area ratio of the silicate vs. H_2O region of spectra collected on silicate glasses to determine H_2O contents in glasses and melt inclusions.

Keywords: Raman Spectroscopy; Melt Inclusions; Fluid Inclusions; Volcanology; Open-Source; Python.

1 Introduction

In recent years, Raman Spectroscopy has been increasingly used by igneous petrologists to perform non-destructive, in-situ measurements of melt and fluid compositions [Steele-Macinnis et al. 2011; Morizet et al. 2013; Hartley et al. 2014; Moore et al. 2015; Schiavi et al. 2018; Giordano et al. 2020; González-García et al. 2020; Schiavi et al. 2020]. Since 2014, there has been a growing body of literature using Raman Spectroscopy to measure the density of CO_2-rich fluids in melt inclusion vapour bubbles to more accurately obtain the total CO_2 content of the melt inclusion, and thus the magma storage depth [Hartley et al. 2014; Moore et al. 2015; Lamadrid et al. 2017; Allison et al. 2021; Wieser et al. 2021; DeVitre et al. 2023b]. Raman spectroscopy also shows enormous potential to quantify the densities of CO_2-rich fluid inclusions [Wang et al. 2011; Kobayashi et al. 2012], allowing rapid and precise estimates of magma storage depths [Dayton et al. 2023; DeVitre and Wieser 2024; Zanon et al. 2024].

However, while there have been a large number of studies optimizing calibration and analysis protocols for CO_2-rich fluids by Raman Spectroscopy [e.g. Lamadrid et al. 2017; Bakker 2021; DeVitre et al. 2021], there has been comparatively less focus on improving software tools for Raman peak fitting in volcanology [e.g. Yuan and Mayanovic 2017]. A number of highly flexible Python packages for fitting Raman spectroscopy data are available (e.g. Rampy [Le Losq 2018], RamanSPy [Georgiev et al. 2023]). At the other end of the spectrum, van Gerve and Namur [2023] published a Python-based graphic user interface for the specific use case of quantifying H_2O contents of silicate glasses. However, there are no packages providing optimized workflows for the various procedures used to quantify CO_2 in volcanological systems. Instead, many studies use proprietary peak fitting software from Raman instrument companies [e.g. Moore et al. 2015], which affects reproducibility because data is collected (and thus processed) on different Raman instruments (e.g. WITec, HORIBA, Bruker, Renishaw). Another popular approach is to use the open-source peak fitting GUI Fityk [Wojdyr 2010]. While scripting is possible in this tool, most users fit curves by manually clicking to select peak and background positions [e.g. DeVitre et al. 2021]. Other studies use OriginLab software which has an annual subscription fee of ~200 USD per license per year [e.g. Hartley et al. 2014], or have developed their own Matlab/Python codes for specific workflows and instruments [e.g. Allison et al. 2021; Wieser et al. 2021].

DiadFit is a high-level Python package which aims to bridge the gap between generic Raman packages which allow experienced coders to build highly customizable workflows for the specific application of interest, and easier-to-use GUI-based methods which are hard to automate. DiadFit is aimed at volcanologists with a wide range of coding experience. As well as providing functions for peak fitting, this package also addresses the fact that there are no widely-available Python tools for common calculations associated with fluid inclusion data, such as calculating CO_2 densities from microthermometry data, and converting CO_2 densities from fluid inclusions into pressure and depth using the CO_2 or CO_2-H_2O equation of state. At present, EOS calculations are typically performed using computer programs with a graphical or text-based user interface (e.g. FLINCOR [Brown 1989], FLUIDS [Bakker 2003]), which can be difficult to install on modern operating systems. Many available tools require calculations to be performed manually for each fluid inclusion, with no efficient way to propagate uncertainties in input parameters.

By integrating functions for peak fitting and subsequent EOS calculations into a single package, entire workflows can be performed in DiadFit without having to swap between different software tools (Figure 1). In addition to reducing time spent data processing, DiadFit will also result in more consis-

*✉ penny_wieser@berkeley.edu

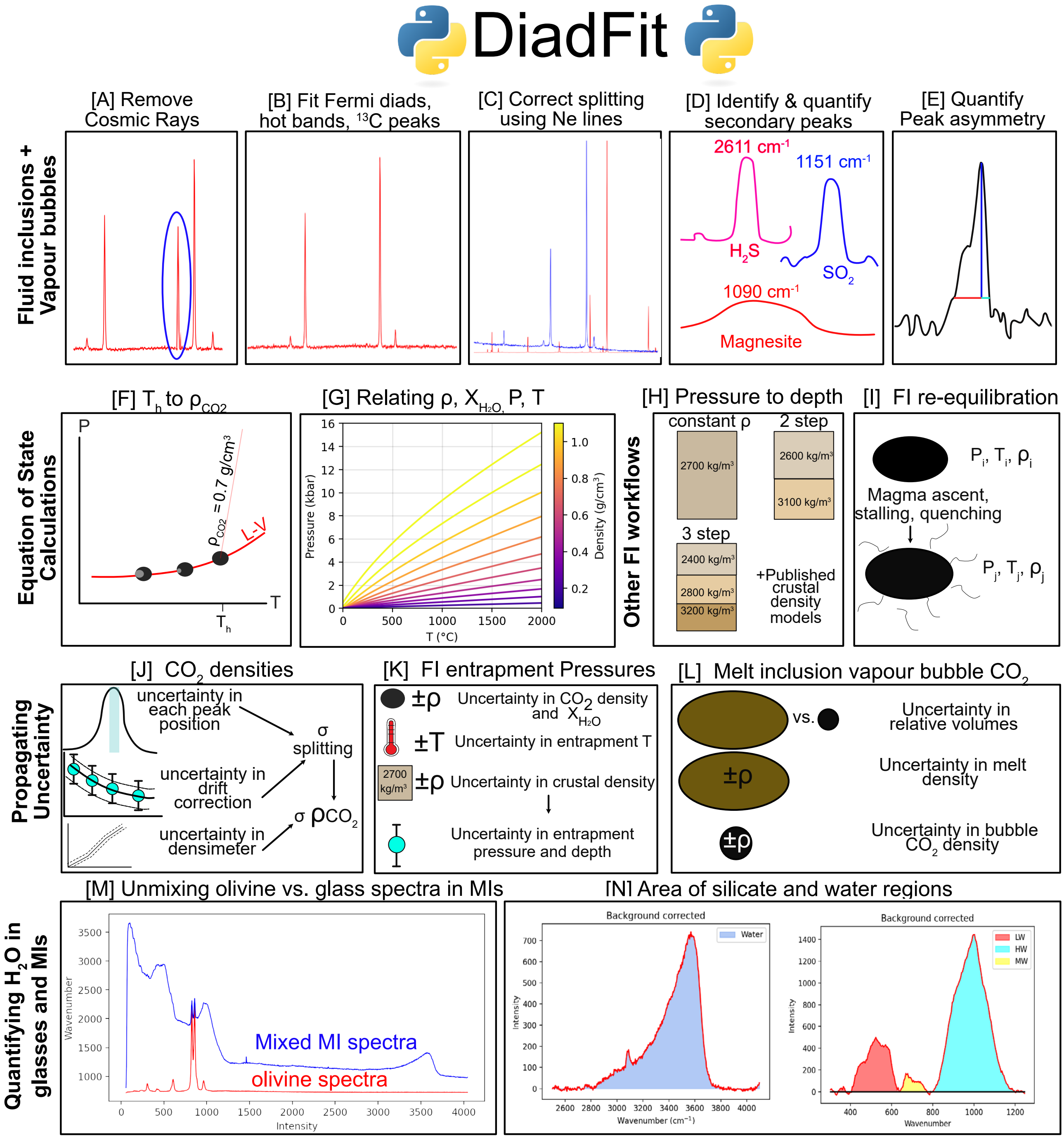

DiadFit
Fluid inclusions + Vapour bubbles
[A] Remove Cosmic Rays
[B] Fit Fermi diads, hot bands, ^{13}C peaks
[C] Correct splitting using Ne lines
[D] Identify & quantify secondary peaks
2611 cm^{-1}
1151 cm^{-1}
H_2S
SO_2
1090 cm^{-1}
Magnesite
[E] Quantify Peak asymmetry
Equation of State Calculations
[F] T_h to ρ_{CO2}
P
T
T_h
ρ_{CO2} =0.7 g/cm^3
L-V
[G] Relating ρ, X_{H_2O}, P, T
Pressure (kbar)
T (°C)
Density (g/cm^3)
Other FI workflows
[H] Pressure to depth
constant ρ
2 step
2700 kg/m^3
2600 kg/m^3
3100 kg/m^3
3 step
2400 kg/m^3
2800 kg/m^3
3200 kg/m^3
+Published crustal density models
[I] FI re-equilibration
P_i, T_i, ρ_i
Magma ascent, stalling, quenching
P_j, T_j, ρ_j
Propagating Uncertainty
[J] CO_2 densities
uncertainty in each peak position
σ splitting
uncertainty in drift correction
uncertainty in densimeter
σ ρCO_2
[K] FI entrapment Pressures
±ρ Uncertainty in CO_2 density and X_{H_2O}
±T Uncertainty in entrapment T
2700 kg/m^3 ±ρ Uncertainty in crustal density
Uncertainty in entrapment pressure and depth
[L] Melt inclusion vapour bubble CO_2
vs.
Uncertainty in relative volumes
±ρ
Uncertainty in melt density
±ρ
Uncertainty in bubble CO_2 density
Quantifying H_2O in glasses and MIs
[M] Unmixing olivine vs. glass spectra in MIs
Wavenumber
Intensity
Mixed MI spectra
olivine spectra
[N] Area of silicate and water regions
Background corrected
Water
Intensity
Wavenumber (cm^{-1})
Background corrected
LW
HW
MW
Wavenumber

Figure 1: Schematic diagram showing the different workflows that can be performed in DiadFit. MI = melt inclusion.

tent fitting between different groups, increasing reproducibility. For example, to align with FAIR principles (Findable, Accessible, Interoperable, Reproducible), users can publish a supporting folder containing their spectra, relevant metadata, and Jupyter Notebooks showing exactly how spectra were fitted and subsequent calculations performed [e.g. Dayton et al. 2023; DeVitre and Wieser 2024].

The shapes and locations of the Fermi diad and Ne lines are well defined, and spectra collected on any given Raman instrument have similar features; DiadFit capitalizes on this similarity to automate many workflows. Each user should tweak generic peak fit parameters for their specific analytical set up until they get good visual fits to representative spectra and acceptable residuals/peak fitting errors. Then, hundreds of spectra can be fitted automatically within minutes using the tweaked parameters, eliminating tedious (and subjective) workflows requiring lots of clicking. DiadFit also outputs the uncertainty associated with each peak fit, which allows for a deeper appreciation of the errors associated with different instrument hardware and acquisition conditions.

To demonstrate the time-saving nature of DiadFit, we compare the time taken to fit Neon lines and CO_2 acquisitions collected over a 24 hour period during instrument calibration by DeVitre et al. [2021] in Fityk vs. DiadFit. Fitting these spectra took ~8 hrs using Fityk, and ~15 minutes using DiadFit on a typical laptop with 16 GB of RAM and an i7 processor. Given the potential for fluid inclusion analyses to provide rapid estimates of magma storage depths during volcanic crises [Dayton et al. 2023; DeVitre and Wieser 2024; Zanon et al. 2024], it is vital to speed up data processing as much as possible to reap the full benefits of this speedy technique. We anticipate that users who are not familiar with Python will simply use the provided Jupyter Notebooks and narrated YouTube videos, changing simple parameters like the path to their files and peak fit parameters to adjust for the different appearance of spectral peaks on different Raman instruments. More experienced coders can adapt the base functions in DiadFit to create their own workflows to address whatever science question they desire.

2 Code installation, underlying dependencies, and architecture

For detailed instructions on code installation, dependencies (i.e. underlying packages required by DiadFit), and supported file types, we refer readers to the ReadTheDocs page*. Briefly, DiadFit is available on PyPI†, so can be installed using pip within a Jupyter Notebook or through the command prompt. DiadFit uses a number of functions from different Python libraries. The SciPy [Virtanen et al. 2020] find_peaks function is used to identify approximate peaks positions and attributes such as height, prominence, and presence/absence of certain peaks (based on pixel values, not fitted spectra). These estimated peak parameters can be used to subdivide spectra into groups with similar characteristics prior to fitting. Grouping spectra by characteristics (e.g. Figure 2) and having good estimates of peak position and prominence greatly increases computational efficiency during iterative fitting of multiple overlapping peaks. Peak-fitting functions utilize the Python package lmfit, which supports iterative fitting of different peak types (e.g. Gaussian, Voigt, Pseudovoigt) and provides errors on peak fitting parameters (e.g. peak center, area, amplitude) from the covariance matrix. Numpy [Harris et al. 2020] is used for all basic math and regression operations, including fitting splines to spectra, and extracting subregions from spectra. Pandas [The pandas development team 2020] is used for importing data from spreadsheet-type datafiles (.xlsx, .csv), and to output fit parameters in a tabular format. Matplotlib [Hunter 2007] is used for all plots produced by functions. EOS calculations using the Span and Wagner [1996] model are implemented through CoolProp [Bell et al. 2014]. This package is not a dependency for all of DiadFit, but if users want to perform calculations using this model, they need to have CoolProp installed (else they will receive an error containing installation instructions). Calculations using this EOS should cite both DiadFit and CoolProp.

3 Python jargon

A user interacts with DiadFit by calling various functions. Each of these functions requires specific input parameters (called arguments). Five main datatypes are used in DiadFit as inputs/arguments and outputs:

1. A "string" is a piece of text. These are used to tell a function something about the data input, or specify a certain thing you want the function to do. For example, users must specify their filetype when using functions that load in spectra (e.g. spectra_filetype='headless_txt').

2. A numpy.array is a column or array of data without headings. DiadFit uses these to store spectral data after it has been extracted from proprietary instrument files. The wavenumber is stored in the 1st column (index 0 in Python), and the intensity in the 2nd column (index 1 in Python). Numpy allows for very fast computation.

3. A pandas.Series is a column of data with a heading.

4. A pandas.DataFrame is a collection of pandas.Series, and can be visualized as a single sheet in an excel workbook with labelled columns. In DiadFit, these are used to store peak fit parameters in columns with column headings (Figure 5). Information in a given column of the dataframe (called df) can be accessed using the column heading: df['column_heading'].

5. dataclasses are used to define default peak finding and fitting configurations. These defaults can be tweaked as much or as little as required for each specific Raman spectrometer. For example, the default parameters for fitting Diad1 are stored in the dataclass diad1_fit_config:

```
diad1_fit_config(model_name='PseudoVoigtModel',
fit_peaks=2, fit_gauss=False,
gauss_amp=1000, diad_sigma=0.2,
diad_sigma_min_allowance=0.2,
diad_sigma_max_allowance=5,
N_poly_bck_diad1=1,
lower_bck_diad1=(1180, 1220),
upper_bck_diad1=(1300, 1350),
diad_prom=100, HB_prom=20,
x_range_baseline=75, y_range_baseline=100,
plot_figure=True, dpi=200,
x_range_residual=20)
```

The input argument model_name='PseudoVoigtModel' specifies that the diad and hot band peaks should be fitted with a PseudoVoigt function, which is a weighted sum of a Gaussian and Lorentzian distribution that share values for amplitude, center, and full width half maximum. A Voigt distribution function can be used instead by editing this argument:

```
pf.diad1_fit_config(model='VoigtModel')
```

Any number of these inputs can be tweaked in this manner. For example, if the user only wants one peak (e.g. for weak spectra), wants to fit a third degree polynomial to the background, and wants saved figures showing spectra fits to have a dpi of 300:

*https://DiadFit.readthedocs.io/en/latest/

†https://pypi.org/project/DiadFit/

```
diad1_fit_config(model='VoigtModel',
fit_peaks=1, N_poly_bck_diad1=3, dpi=300)
```

Once these dataclasses are tweaked for a given instrument/set of samples, they can be used to automatically loop through large numbers of spectral files.

4 Worked examples

We have produced a number of example Jupyter Notebooks demonstrating specific workflows. These notebooks are available on GitHub[*] and the ReadTheDocs page[†]. We include Notebooks showing how to:

1. Fit peaks from CO_2-rich fluid inclusions and melt inclusion vapour bubbles and correct data for instrument drift using Ne line acquisitions. Some of these examples also show how to quantify the area of carbonate and SO_2 peaks, and calculate molar ratios of different gas species (Figure 1A–D).

2. Perform calculations using the CO_2 and CO_2-H_2O equation of state, including converting homogenization temperatures from microthermometry into CO_2 densities, converting CO_2 densities from Raman or microthermometry into pressures, and then converting these pressures to depths using different crustal density profiles (Figure 1F–H).

3. Propagate uncertainty in microthermometry and Raman-based fluid inclusion barometry into pressure and depth distributions (Figure 1J, K).

4. Model fluid inclusion re-equilibration for different magma ascent, stalling and quenching scenarios following DeVitre and Wieser [2024] (Figure 1I).

5. Quantify peak asymmetry using the approach of DeVitre et al. [2023a] to help identify fluid inclusions which contain both liquid and vapour phases at the time of analysis (Figure 1E).

6. Quantify H_2O contents in silicate glasses using an approach adapted from Di Genova et al. [2017] and Schiavi et al. [2018] (Figure 1N).

7. Quantify the ratio of H_2O to silicate glass peaks on Raman acquisitions taken on unexposed olivine-hosted melt inclusions (Figure 1M, N).

8. Propagating uncertainties from Raman measurements, volume determinations, and melt density calculations when determining the contribution of vapour bubble CO_2 to the total CO_2 budget of melt inclusions (Figure 1L).

5 Fitting the CO_2 Fermi diad

The Raman spectrum of CO_2 consists of two relatively strong, well-defined peaks collectively termed the Fermi diad. These peaks result from the interaction of a symmetrical stretching mode and an active bending mode in the CO_2 molecule by the process of Fermi resonance [Fermi 1931; Rosso and Bodnar 1995; Lamadrid et al. 2017]. One of the peaks is centered at ∼1280–1290 cm^{-1} (referred to as Diad1), and the second peak at ∼1386–1390 cm^{-1} (Diad2). It is well accepted that the distance between Diad1 and Diad2 (commonly referred to as splitting, diad splitting, Fermi diad separation, or Δ) correlates with the density of the CO_2 fluid. This is because with increasing CO_2 density, Diad1 moves to lower wavenumbers (shift of 6 cm^{-1} from 0 to 0.8 g cm^{-3}), while the position of Diad2 stays more constant (only varying by ∼1–2 cm^{-1}). The exact relationship between diad splitting and CO_2 density, commonly called a "densimeter", has been shown to vary as a function of instrument hardware and acquisition conditions [Lamadrid et al. 2017; DeVitre et al. 2021].

The Fermi diad is flanked by low intensity hot bands (HB1 at lower wavenumbers than Diad1 and HB2 at higher wavenumbers than Diad2, Figure 2). These arise from molecules populating the first excited vibron due to their thermal energy at room temperature, which are also perturbed by a Fermi resonance effect. The peak height of the HBs relative to the diad peaks increases with increasing temperature [Rosso and Bodnar 1995], and in general, hotbands become more pronounced in spectra collected on denser CO_2 fluids. Diad2 also has a flanking ^{13}C peak at lower wavenumbers which becomes more visible as the CO_2 density increases and with increasing proportions of ^{13}C relative to ^{12}C [Wang and Lu 2022]. Fitting these peaks can be used to quantify the relative abundance of $^{12}C/^{13}C$ [see Lu et al. 2023].

In the literature, density estimates from CO_2-rich fluids such as melt inclusion vapour bubbles or fluid inclusions have been obtained by fitting the position of the two diad peaks using various background models combined with Voigt, Pseudovoigt, Gaussian, or Lorentzian peaks. These peak fitting routines greatly increase the precision at which the diad splitting can be determined relative to the spectral resolution of the instrument (distance between datapoints). While Raman instruments typically acquire an intensity reading every 0.1–1.5 cm^{-1}, the error of the peak position on a fitted peak can be 10–30× smaller if an appropriate peak profile is fitted (e.g. Pseudovoigt/Voigt for diads, [Fukura et al. 2006; Lin et al. 2007; Yuan and Mayanovic 2017], see also Section 9.1). After the peaks are fit, splitting is calculated and corrected for instrument drift; an instrument-specific densimeter is used to convert corrected splitting into CO_2 density [Lamadrid et al. 2017; DeVitre et al. 2021].

When using DiadFit to peak fit CO_2 spectra, the first step is to find all the spectral files you wish to fit (by filtering based on file names, Figure 3). Then, a single file is selected, and the DiadFit function identify_diad_peaks is used to tweak the SciPy find_peaks parameters, displaying the identified peaks with yellow stars. This step is necessary because different Raman instruments have vastly different intensities and spectral resolutions. We find that the prominence, defined as the vertical distance above the neighbouring background, is the most important parameter to tweak between different instruments (some instruments have peaks 100s–1000s of counts above background while others are 100,000–1 million counts above background). Then, the function loop_approx_diad_fits

[*] https://github.com/PennyWieser/DiadFit

[†] https://DiadFit.readthedocs.io/en/latest/

is used to loop through all files to obtain approximate peak positions and prominences for the diad, hotbands and ^{13}C peaks for each spectra, which is stored in a `dataframe`. This function also calculates various additional parameters that are very helpful when grouping spectra in later steps (e.g. signal to noise ratio, elevation of the diad-HB region above background).

These approximate peak parameters are then used alongside a filter to remove any cosmic rays present in the spectra. The filtering process calculates the intensity factor F based on the intensity (I) of any given spectral datapoint relative to the intensities of the datapoints to the left (i-1) and right (i+1):

$$F_{i-1} = \frac{I_i - I_{i-1}}{I_{i-1}} \tag{1}$$

$$F_{i+1} = \frac{I_i - I_{i+1}}{I_{i+1}} \tag{2}$$

These two factors are then multiplied together:

$$M_i = F_{i-1} * F_{i+1}. \tag{3}$$

If the spectra has a lower intensity to the left of datapoint i, F_{i-1} will be >1. If the spectra also has lower intensity on the right hand side of point i, F_{i+1} will be >1, so M_i will be $\gg$1. High values of M_i for individual pixels are indicative of short, sharp peaks (i.e. most cosmic rays). A datapoint is considered a cosmic ray if M_i exceeds a user determined value (e.g. `dynfact=0.001`). The highest pixel of Diads, HB, and ^{13}C peaks can also have high M_i values, so any points with an x coordinate equal to an identified peak position (± the spectral resolution) are ignored. Users can also enter a custom range, within which no cosmic rays are removed (e.g. the area around the narrow SO_2 peak). If a cosmic ray is identified, the filter is run again, in case the cosmic ray was wider than a single pixel. The pixels identified as cosmic rays are then removed from the spectrum and a new file is saved with the same filename with the addition of the suffix `_CRR_DiadFit`.

After removal of cosmic rays, the `identify_diad_peaks` function is used again to obtain approximate peak parameters without inference from cosmic rays. These parameters are then used to divide spectra into groups. Dividing spectra into groups is advantageous because spectra from CO_2 fluids vary greatly in their characteristics as a function of the inclusion size, depth, CO_2 density, laser power, and acquisition time (Figure 2). Spectra collected on CO_2 fluids with low densities, at relatively deep depths, or using low laser power tend to have a weak to non-existent hot bands (Figure 2A–C). Stronger spectra have prominent hot bands and ^{13}C peaks which overlap with the tails of the diad peaks (Figure 2D–F). In the strongest spectra, the entire region around the diad peaks and hotbands is elevated above the near-linear background seen at greater distance from the diad peaks (Figure 2G–I). Different fit parameters are needed to obtain satisfactory fits on these different spectral types.

A full description of the different parameters used for classification is provided on the ReadTheDocs page, and the most useful parameters will depend greatly on the Raman instrument and samples. In general, spectra should be classified as "Weak" when none of the peaks are strong enough to interfere with each other (e.g. Figure 2C, where the hot bands and diad peaks do not have overlapping tails). Spectra should be classified in the "Medium" group once the hotbands and diads begin to overlap substantially. This will likely result in the user wanting to specify background positions which are further from the peaks than for the "Weak" group. Often, a Gaussian background is required in addition to the polynomial background for Diad2. Spectra with very strong hot bands, clear ^{13}C peaks, and greatly elevated troughs between diad peaks and HBs should be placed into the "Strong" category. These spectra will need to be fitted with the addition of a Gaussian background to recreate the elevation of the spectra for both Diad1 and Diad2 (`fit_gauss=True`, Figure 2I). The choice of groups is somewhat arbitrary, and is best viewed as an easy way to be able to set three different peak fit routines. In many studies, all spectra will fit in a single group. Users should not stress too much about whether to add a Gaussian background or not. If a Gaussian background is not really required (e.g. Figure 2F) but is requested, the code will converge on a Gaussian with a very small amplitude. Thus, the only real advantage of putting those not really needing Gaussian backgrounds into a "Medium" Category is that the iterative fitting routine will be a bit faster (favorable when fitting large numbers of spectra).

One file from each group is then used to tweak the fit parameters (Figure 3), which are applied to all spectra in the group (taking 1–5 s per spectra on a 16 GB intel i7 laptop). A figure of each fit is saved in the same folder as the spectra and displayed in the Notebook—fits should be carefully inspected for each spectrum. In particular, attention should be paid to the subplot showing the residual around the peak, and the selected background positions and background fit. The returned `dataframe` of fitted parameters also contains the uncertainties in peak positions, which can also be very helpful for identifying poor fits (Figure 10). In addition to plots for visual inspection of residuals, this function also calculates a measure of the residual for each peak:

$$\text{Residual} = \frac{\sum \sqrt{(y_{\text{meas}} - y_{\text{fitted}})^2}}{N_{\text{points}}} \tag{4}$$

where y_{meas} is the y-coordinate of the background-subtracted data, y_{fitted} is the y coordinate of the best composite model fit, and N_{points} is the number of discrete points along the x axis used in the fit.

The next step is to combine the `dataframe` of peak fit parameters from each group, and combine this with information extracted from the metadata. In particular, it is necessary to obtain the time stamp of the analysis to apply the Ne correction model. Finally, splittings corrected for instrument drift are converted into CO_2 densities using a instrument-specific densimeter. Uncertainties from each step of this process are propagated (see Section 9.1).

5.1 Fitting peaks to secondary phases

Raman spectra collected from fluid inclusions and melt inclusion vapour bubbles commonly contain peaks arising from other gaseous species (e.g. SO_2, N_2) or solid phases on the

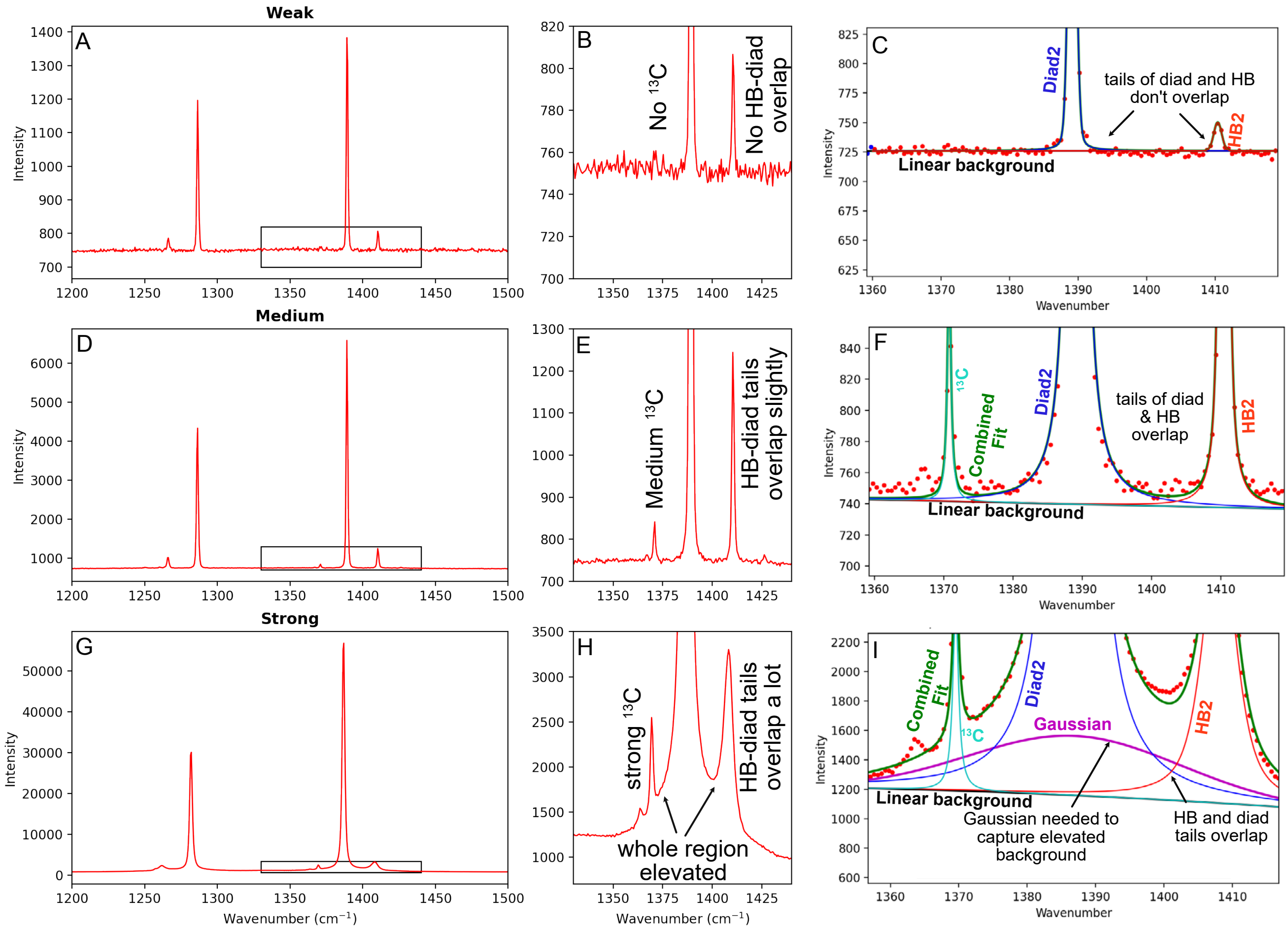


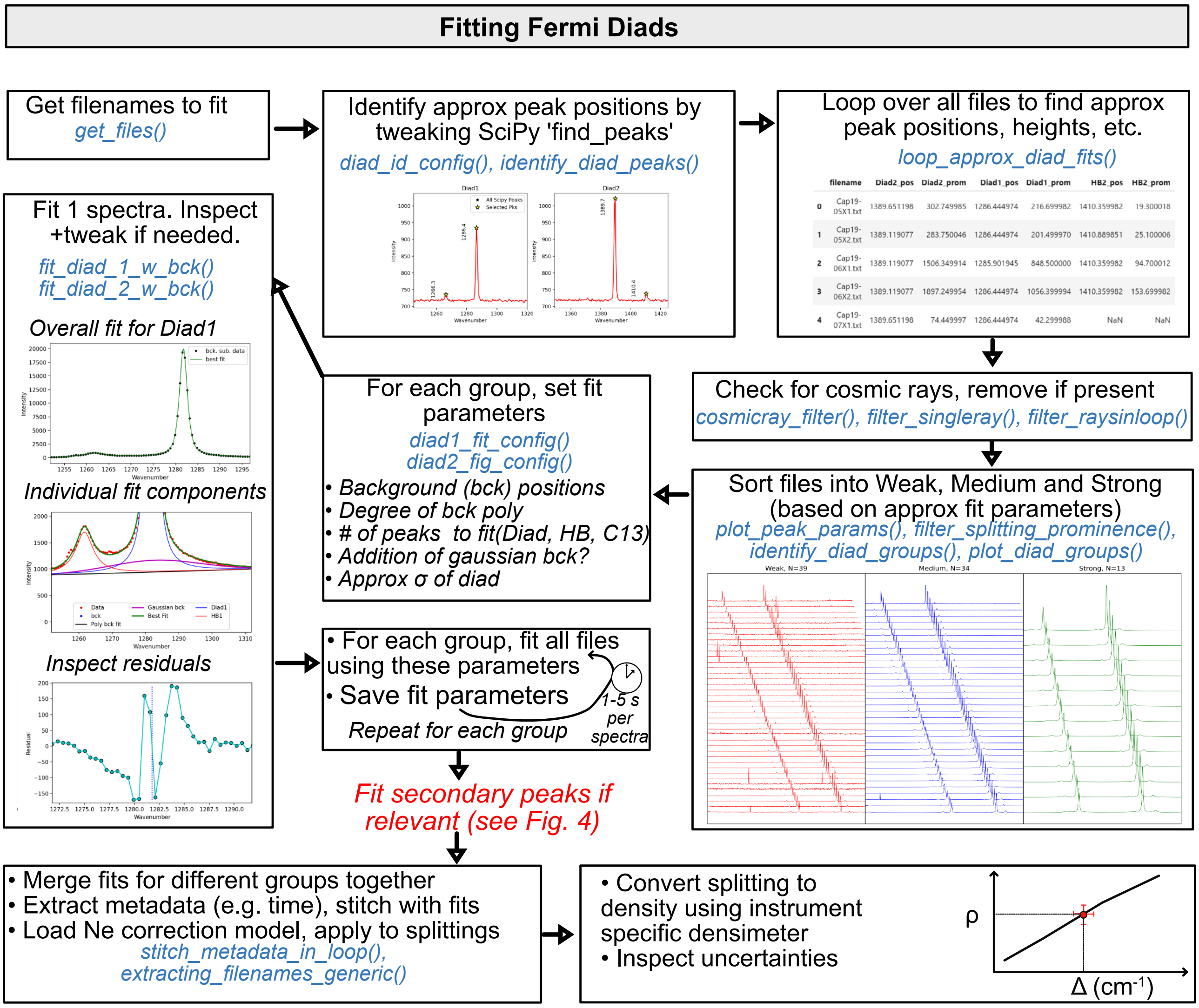
Figure 2: Spectra from DeVitre et al. [2021] showing different characteristics to demonstrate how to group spectra for ease of fitting. Spectra classified as "Weak" [A]–[C] have no ^{13}C peak and a small HB which does not overlap with the diad [C]. Spectra classified as "Medium" [D]–[F] have a ^{13}C peak, and the tails of the ^{13}C, diad and HB peak all overlap [F]. In this case, a Gaussian background was not added, but for any stronger spectra it would be required as the red dots are beginning to diverge from the best fit line (green curve) inbetween Diad2 and HB2. In spectra classified as "Strong" [G]–[I], the troughs between ^{13}C, Diad2 and HB2 are greatly elevated above the background. This requires a Gaussian background (purple curve) in addition to the three other peaks (red, blue, cyan) to minimise residuals.

inclusion wall (e.g. carbonates, sulfates [Frezzotti et al. 2012; Schiavi et al. 2020]). It can be helpful to identify which spectra contain such phases, and quantify characteristics of these peaks (e.g. peak heights, positions, relative peak areas [Burke 2001]). The first step is to obtain all the spectra you wish to fit (Figure 4). Then, the function `plot_secondary_peaks` plots the spectra in a specific wavenumber range from all selected files. Intensities are normalized based on the range of intensities in the selected wavenumber window, allowing multiple spectra to be stacked and inspected. It is clear from this visualization strategy that files #42 and #33 in Figure 4 have prominent carbonate peaks. For any given set of spectra, you can inspect as many regions as you want in this way. For example, the second panel in Figure 4 is centered around the SO_2 peak at ~1151 cm^{-1}.

As well as plotting spectra to visually identify secondary phases, the function `plot_secondary_peaks` can identify peak positions within a user-defined wavenumber range using scipy `find_peaks` or a prominence filter. Identified peaks will be marked on the returned figure with a yellow star (Figure 4). The function also returns a `dataframe` of approximate peak parameters based on the maximum pixel (position, height, prominence)—where no peaks are found, these columns are filled with NaNs ("not a number"). After identifying spectra containing secondary peaks, peak fitting parameters should be tweaked based on one selected spectrum. Using the `model_name` parameter, secondary peaks can be fit with Gaussian, PseudoVoigt or Voigt curves, or a cubic spline. Tweaked fit parameters are then applied to all files. More than one type of secondary peak can be quantified in a given notebook. Finally, all identified secondary peaks are merged together with fitting parameters for diad peaks (see Figure 3).

6 Fitting Ne lines

During any given analytical session, there may be some stretching or contraction of the Raman spectra; these changes are often referred to as "non-linearity of the Raman shift axis". For brevity and to draw parallels with various mass spec-


Fitting Fermi Diads
Get filenames to fit
get_files()
Identify approx peak positions by tweaking SciPy 'find_peaks'
diad_id_config(), identify_diad_peaks()
Diad1
Diad2
All Scipy Peaks
Selected Pks
Wavenumber
Intensity
Loop over all files to find approx peak positions, heights, etc.
loop_approx_diad_fits()
Check for cosmic rays, remove if present
cosmicray_filter(), filter_singleray(), filter_raysinloop()
Sort files into Weak, Medium and Strong (based on approx fit parameters)
plot_peak_params(), filter_splitting_prominence(), identify_diad_groups(), plot_diad_groups()
Weak, N=39
Medium, N=34
Strong, N=13
For each group, set fit parameters
diad1_fit_config()
diad2_fig_config()
• Background (bck) positions
• Degree of bck poly
• # of peaks to fit(Diad, HB, C13)
• Addition of gaussian bck?
• Approx σ of diad
Fit 1 spectra. Inspect +tweak if needed.
fit_diad_1_w_bck()
fit_diad_2_w_bck()
Overall fit for Diad1
bck. sub. data
best fit
Individual fit components
Data
bck
Poly bck fit
Gaussian bck
Best Fit
Diad1
HB1
Inspect residuals
Residual
Wavenumber
• For each group, fit all files using these parameters
• Save fit parameters
Repeat for each group
1-5 s per spectra
Fit secondary peaks if relevant (see Fig. 4)
• Merge fits for different groups together
• Extract metadata (e.g. time), stitch with fits
• Load Ne correction model, apply to splittings
stitch_metadata_in_loop(), extracting_filenames_generic()
• Convert splitting to density using instrument specific densimeter
• Inspect uncertainties
ρ
Δ (cm-1)

	filename	Diad2_pos	Diad2_prom	Diad1_pos	Diad1_prom	HB2_pos	HB2_prom
0	Cap19-05X1.txt	1389.651198	302.749985	1286.444974	216.699982	1410.359982	19.300018
1	Cap19-05X2.txt	1389.119077	283.750046	1286.444974	201.499970	1410.889851	25.100006
2	Cap19-06X1.txt	1389.119077	1506.349914	1285.901945	848.500000	1410.359982	94.700012
3	Cap19-06X2.txt	1389.119077	1897.249954	1286.444974	1056.399994	1410.359982	153.699982
4	Cap19-07X1.txt	1389.651198	74.449997	1286.444974	42.299988	NaN	NaN

Figure 3: Schematic of the workflow used to efficiently fit diad peaks with very different spectral characteristics.

trometry techniques, we term this "instrument drift". Commonly, instrument drift within the spectral region containing the Fermi diad is assessed by repeatedly measuring the emission spectrum of Neon (Ne) produced by a lamp every ~5–10 minutes. As well as correcting drift during a given session, Ne lines are also useful for variations between sessions, allowing continued use of a Raman densimeter months to years after it was calibrated [DeVitre et al. 2021]. An alternative approach to correct for instrument drift uses repeated analyses of a sample/capillary with known CO_2 density [Le et al. 2021; Wieser et al. 2021]. This approach is particularly useful on instruments without a Ne lamp, or with insufficient spectral resolution to quantify the narrow Ne peaks.

The Ne emission spectrum has a number of distinctive "lines," several of which lie close to the position of the CO_2 Fermi diad. The wavelength of the different Ne lines in air (λ_1) are converted into Raman shifts (Δ_v) in cm^{-1} using the excitation wavelength of the laser (e.g. λ_0=532.05 nm; [Lin et al. 2007]):

$$\Delta_v = \frac{10^7}{\lambda_0\,(\mathrm{nm})} - \frac{10^7}{\lambda_1\,(\mathrm{nm})}. \tag{5}$$

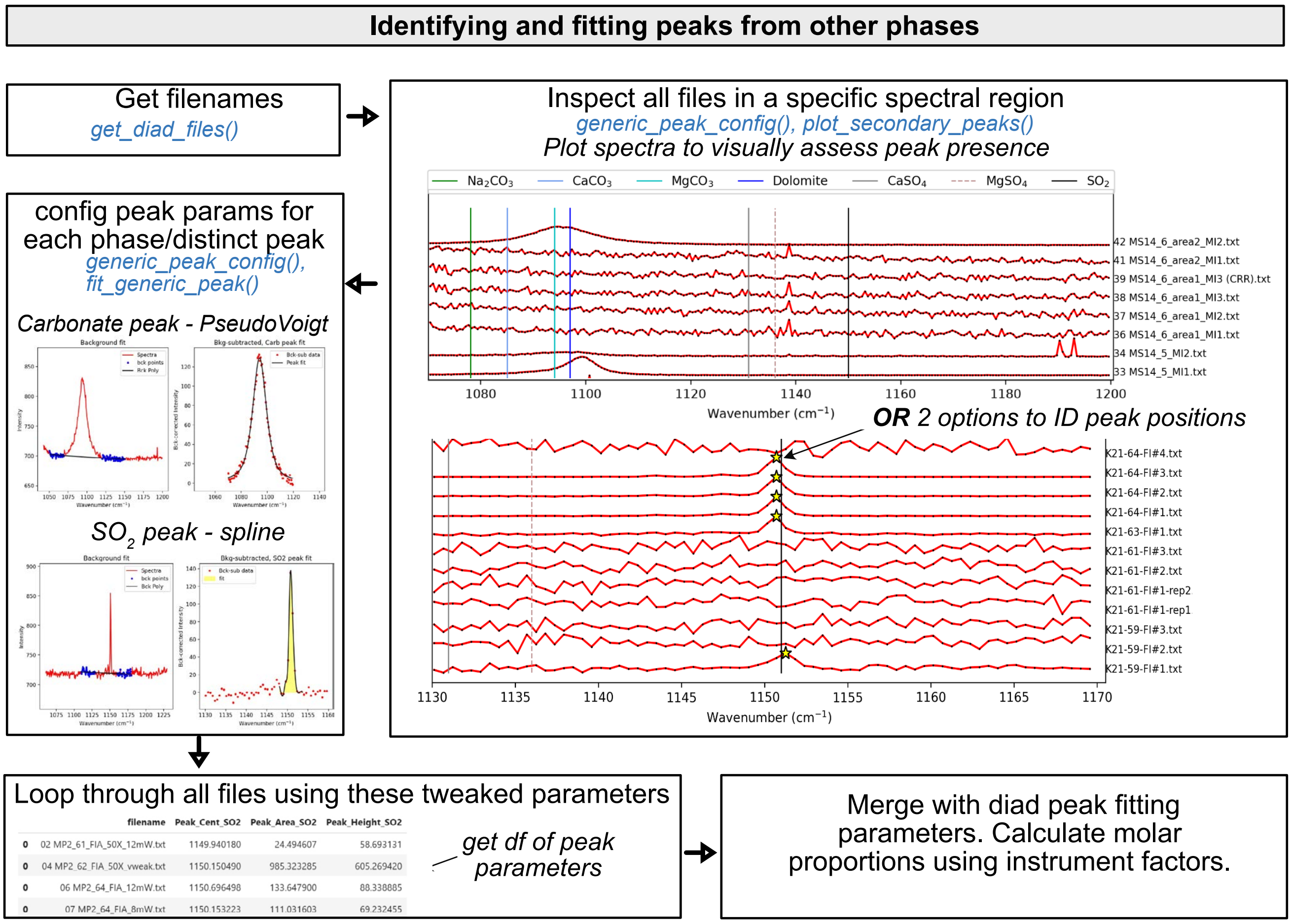
Traditionally, a measured diad splitting has been corrected for instrument drift and non-linearity by comparing the measured distance between two selected Ne lines ($\Delta_{\mathrm{Ne}}^{\mathrm{Measured}}$) and the theoretical distance between those lines using Equation 5 ($\Delta_{\mathrm{Ne}}^{\mathrm{Theoretical}}$):

$$\text{Correction Factor} = \frac{\Delta_{\mathrm{Ne}}^{\mathrm{Theoretical}}}{\Delta_{\mathrm{Ne}}^{\mathrm{Measured}}}. \tag{6}$$

This correction factor is then multiplied by the mesaured diad splitting:

$$\Delta_{\mathrm{CO_2}}^{\mathrm{Corrected}} = \text{Correction Factor} \times \Delta_{\mathrm{CO_2}}^{\mathrm{Measured}}. \tag{7}$$

This approach has been termed the "line segment" technique by Bakker [2021], and a typical workflow of how this method is performed in DiadFit is summarized in Figure 5.


Identifying and fitting peaks from other phases
Get filenames
get_diad_files()
Inspect all files in a specific spectral region
generic_peak_config(), plot_secondary_peaks()
Plot spectra to visually assess peak presence
Na2CO3
CaCO3
MgCO3
Dolomite
CaSO4
MgSO4
SO2
42 MS14_6_area2_MI2.txt
41 MS14_6_area2_MI1.txt
39 MS14_6_area1_MI3 (CRR).txt
38 MS14_6_area1_MI3.txt
37 MS14_6_area1_MI2.txt
36 MS14_6_area1_MI1.txt
34 MS14_5_MI2.txt
33 MS14_5_MI1.txt
Wavenumber (cm−1)
OR 2 options to ID peak positions
K21-64-FI#4.txt
K21-64-FI#3.txt
K21-64-FI#2.txt
K21-64-FI#1.txt
K21-63-FI#1.txt
K21-61-FI#3.txt
K21-61-FI#2.txt
K21-61-FI#1-rep2
K21-61-FI#1-rep1
K21-59-FI#3.txt
K21-59-FI#2.txt
K21-59-FI#1.txt
Wavenumber (cm−1)
config peak params for each phase/distinct peak
generic_peak_config(), fit_generic_peak()
Carbonate peak - PseudoVoigt
Background fit
Bkg-subtracted, Carb peak fit
SO2 peak - spline
Background fit
Bkg-subtracted, SO2 peak fit
Loop through all files using these tweaked parameters
get df of peak parameters
Merge with diad peak fitting parameters. Calculate molar proportions using instrument factors.

	filename	Peak_Cent_SO2	Peak_Area_SO2	Peak_Height_SO2
0	02 MP2_61_FIA_50X_12mW.txt	1149.940180	24.494607	58.693131
0	04 MP2_62_FIA_50X_vweak.txt	1150.150490	985.323285	605.269420
0	06 MP2_64_FIA_12mW.txt	1150.696498	133.647900	88.338885
0	07 MP2_64_FIA_8mW.txt	1150.153223	111.031603	69.232455

Figure 4: Schematic of workflow used to fit secondary peaks (e.g. carbonate, SO_2). Secondary peak fitting results are then merged with diad parameters (allowing calculation of molar proportions of gas species).

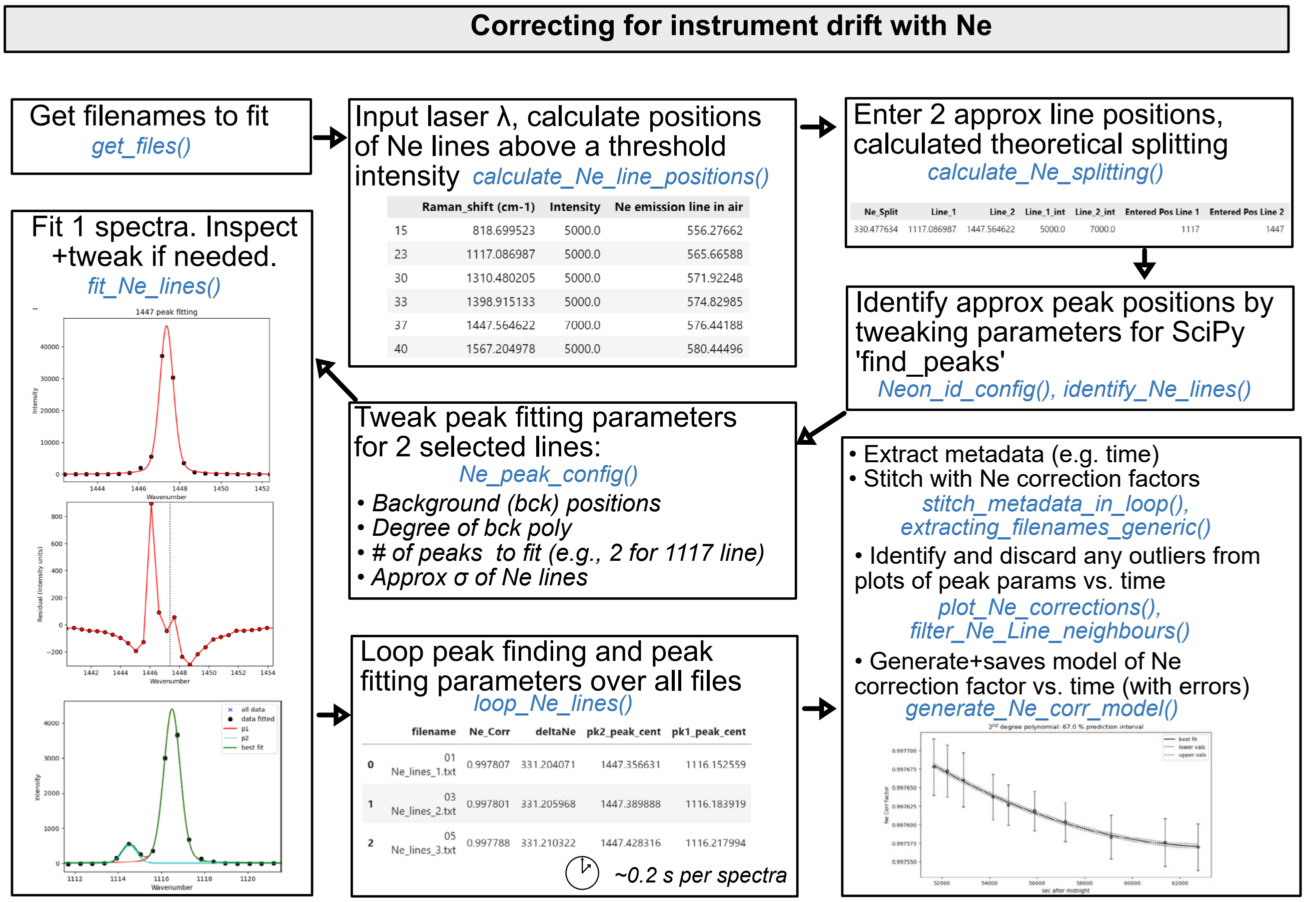
First, identify all the files which contain Ne lines based on a unique string in their name (e.g. "Ne"). Next, the function `calculate_Ne_line_positions` is used to calculate the wavenumber of each Ne line using the specific wavelength of the excitation laser, and a threshold intensity to filter out weaker lines:

```
df_Ne=pf.calculate_Ne_line_positions(
wavelength=531.885, cut_off_intensity=2000)
```

This returns a `dataframe`, with the Raman shift (wavenumber) expected for each Ne line position for the specific wavelength (see Figure 5):

Two Ne lines are selected (1117 cm^{-1} and 1447 cm^{-1} in this example), and the theoretical splitting is calculated for the closest lines to those specified wavenumbers. Next, one representative Ne acquisition file is selected, and used to tweak `SciPy find_peaks` parameters stored in the class `Neon_id_config` to identify the approximate position of each Ne line (and its intensity) by identifying the largest peak within ± 10 datapoints of the Ne line of interest (e.g. for a spectra resolution of 0.4 cm^{-1} and for line 1117, the spectral region 1113–1121 cm^{-1} would be searched). Obtaining approximate peak positions and intensities greatly improves the computational efficiency of the fitting process.

After identifying approximate peak positions for each file, peak fitting parameters are tweaked in the class `Ne_peak_config`. The peak fitting function fits a polynomial background to two regions either side of the Ne line, discarding points within this region which are outside of a certain number of standard deviations of the median background position (default 3σ, can be tweaked in `Ne_peak_config`). This background is then subtracted. For most Ne lines, a single PseudoVoigt/Voigt curve can be fitted to background-subtracted data (see ~1447 cm^{-1} peak, Figure 5). For the line at ~1117 cm^{-1}, two peaks need to be fitted iteratively because of the prominent shoulder to the left of this line (red and cyan curves, combined into the green curve, Figure 5). In general, the default peak fitting values stored in a dataclass `Ne_peak_config` will only need tweaking once for each Ne line on each instrument, because repeated Ne acquisitions are so similar. The most important parameters to change when swapping between lines are the positions of the background. These are expressed in terms of distance from the peak center identified above. For example, specifying `lower_bck_pk1=(-40, -25)` positions a background window 40–25 wavenumbers to the left of the pixel with maximum intensity. Other important parameters include the degree of polynomial to fit between background points



Correcting for instrument drift with Ne

Get filenames to fit
get_files()

Input laser λ, calculate positions of Ne lines above a threshold intensity *calculate_Ne_line_positions()*

	Raman_shift (cm-1)	Intensity	Ne emission line in air
15	818.699523	5000.0	556.27662
23	1117.086987	5000.0	565.66588
30	1310.480205	5000.0	571.92248
33	1398.915133	5000.0	574.82985
37	1447.564622	7000.0	576.44188
40	1567.204978	5000.0	580.44496

Enter 2 approx line positions, calculated theoretical splitting
calculate_Ne_splitting()

Ne_Split	Line_1	Line_2	Line_1_int	Line_2_int	Entered Pos Line 1	Entered Pos Line 2
330.477634	1117.086987	1447.564622	5000.0	7000.0	1117	1447

Identify approx peak positions by tweaking parameters for SciPy 'find_peaks'
Neon_id_config(), identify_Ne_lines()

Tweak peak fitting parameters for 2 selected lines:
Ne_peak_config()
- *Background (bck) positions*
- *Degree of bck poly*
- *# of peaks to fit (e.g., 2 for 1117 line)*
- *Approx σ of Ne lines*

Fit 1 spectra. Inspect +tweak if needed.
fit_Ne_lines()

Loop peak finding and peak fitting parameters over all files
loop_Ne_lines()

	filename	Ne_Corr	deltaNe	pk2_peak_cent	pk1_peak_cent
0	01 Ne_lines_1.txt	0.997807	331.204071	1447.356631	1116.152559
1	03 Ne_lines_2.txt	0.997801	331.205968	1447.389888	1116.183919
2	05 Ne_lines_3.txt	0.997788	331.210322	1447.428316	1116.217994

~0.2 s per spectra

- Extract metadata (e.g. time)
- Stitch with Ne correction factors
  stitch_metadata_in_loop(), extracting_filenames_generic()
- Identify and discard any outliers from plots of peak params vs. time
  plot_Ne_corrections(), filter_Ne_Line_neighbours()
- Generate+saves model of Ne correction factor vs. time (with errors)
  generate_Ne_corr_model()

Figure 5: Schematic showing how DiadFit can be used to fit Ne lines, and build a model of correction factor vs. time with full propagation of uncertainty.

N_poly_pk1_baseline=2, and the approximate sigma of the PseudoVoigt/Voigt curve (pk1_sigma=0.6, pk2_sigma=0.7). These tweaked peak fit parameters are then used to fit one spectra file with the function fit_Ne_lines. This function returns the peak positions and other fit parameters, as well as graphs showing the overall best fit, the residual of the fit, and the background positions.

After inspecting these graphs for one Neon acquisition and tweaking any fit parameters as necessary (e.g. background positions, plotting parameters), the tweaked find peaks and peak fit parameters can then be used to loop through all Ne spectra. On a US$1000 laptop (e.g. 16 GB RAM, Intel i7), it takes approximately 0.2 s to fit each Ne file. There is also an option for the function to save a figure showing the fit for each file in a subfolder it creates within the spectra path. This increases the run time by a factor of five for each file, but allows users to check the fits, or publish fits as part of a data repository accompanying a paper. After all files are looped through, DiadFit returns a pandas.DataFrame with the filename, and all the peak fit parameters (including the all important Ne correction factor).

In most published studies, Fermi diad acquisitions have been corrected using the average correction factor of the Ne line acquired before and after a specific sample (although occasionally concurrent acquisitions are used). However, the relatively narrow shape of the Ne peaks means that the error on the peak center of each line, and thus the Ne correction factor, is relatively large (see error bars on Ne correction factor on Figure 5). Thus, we suggest it may be better to perform a regression of the correction factor as a function of time to average out this random peak fitting noise. On many instruments (e.g. WITec, some HORIBAs), the spectral file doesn't include a useful time stamp (and the file write/edit date may be the time of export, not acquisition). Sometimes (e.g. WITec), the timestamp is stored in the metadata file, meaning that data and metadata files can be stitched together to obtain the time for each Ne correction factor (see documentation for examples). In this instance, DiadFit is designed to work best when spectra and metadata files have the same name—if users wish to use different names, they will have to adapt the automatic stitching functions. If a metadata file is not available, DiadFit contains functions to extract the time based on the read or edit time stamp on the spectra file. Once a dataframe is obtained that contains both the Ne correction factor and the time, it is useful to inspect changes in Neon correction factors and peak positions with time using built-in visualization functions (e.g. plot_Ne_corrections). This allows outliers to be discarded that differ substantially from adjacent acquisitions using the function `filter_Ne_lines_neighbours`. Finally, a polynomial or spline model is used to parameterize the change in Ne correction factor as a function of time (expressed as seconds after midnight, Figure 5), along with the associated confident interval. This model is saved, so that it can be loaded when processing CO_2 spectra from the same session to calculate the correction factor at each point in time (and the associated uncertainty).

It should be noted that Bakker [2021] criticized the "line segment" technique, because it assumes that the correction factor at any given point between two lines is the same, which is not true if there is any non-linearity in the spectrometer. `DiadFit` allows users to fit as many different Ne lines as they wish, so more complex Ne correction routines could be utilized (we address differences between drift correction methods in a later contribution).

7 Equation of state calculations

An equation of state (EOS) describes the relationship between physical properties of a system such as pressure, temperature, and density. EOS calculations are widely used in volcanology—they allow conversion of fluid inclusion densities into pressures, conversion of homogenization temperatures from microthermometry into CO_2 densities, and calculations of densities inside optical cells widely used to calibrate Raman spectrometers. At present, `DiadFit` supports calculations using pure CO_2 and mixed CO_2-H_2O EOS. In future, we could implement additional EOS upon request from users (so check ReadTheDocs).

7.1 Pure CO_2 EOS

`DiadFit` currently supports calculations using the EOS for pure CO_2 using the model of Sterner and Pitzer [1994] (hereafter SP94) or Span and Wagner [1996] (hereafter SW96). While the SP94 EOS is hard-coded directly into `DiadFit`, we perform calculations using the SW96 EOS through the `Python` package `CoolProp` [Bell et al. 2014], which needs to be installed to use the `DiadFit` functions relying on this EOS. There are three core functions used to perform EOS calculations in `DiadFit`:

1. `calculate_rho_for_P_T`: calculates CO_2 density for a known pressure and temperature.

2. `calculate_P_for_rho_T`: calculates pressure for a known CO_2 density and temperature.

3. `calculate_T_for_rho_P`: calculates temperature for a known pressure and CO_2 density.

7.1.1 Calculating CO_2 density for a given P and T

The function `calculate_P_for_rho_T` can be used to calculate CO_2 density for a specified Pressure (P) and Temperature (T). This calculation is very useful when processing data collected from Raman gas cell calibration apparatus, where T is measured by a thermistor and P is measured with a pressure transducer in the fluid within the optical cell [DeVitre et al. 2021]. An entire spreadsheet of pressures and temperatures can be imported and all densities calculated:

```
df=pd.read_excel('Cali_Data.xlsx')
dens_SW96=pf.calculate_rho_for_P_T(
P_kbar=df['P_kbar'], T_K=df['T_K'],
EOS='SW96')
```

7.1.2 Calculating pressures from CO_2 densities and temperatures

The function `calculate_P_for_rho_T` can be used to calculate pressure for a specified CO_2 density (rho) and temperature (T). This calculation is commonly used to calculate fluid inclusion pressures. This pressure may represent the entrapment pressure under the assumption that the volume and mass of the inclusion is fixed from the [Roedder 2018], or a re-equilibration pressure if the inclusion underwent prolonged stalling [Hansteen 1991].

To convert a CO_2 density of 0.5 g cm^{-3} into a pressure at a temperature of 1200 K using the Span and Wagner [1996] EOS (SW96):

```
P_SW96=pf.calculate_P_for_rho_T(
CO2_dens_gcm3=0.5, T_K=1200, EOS='SW96')
```

The Sterner and Pitzer [1994] EOS (SP94) can be used instead simply by changing the `EOS` argument:

```
P_SP94=pf.calculate_P_for_rho_T(
density_gcm3=0.5, T_K=1200, EOS='SP94')
```

In addition to single calculations, this function can also be applied to a `pandas.DataFrame` with as many rows as the user wants. For example, to load an entire spreadsheet with columns for CO_2 densities and an estimate of entrapment temperatures in °C (as well as an optional column with the sample name):

```
df=pd.read_excel('FI_densities.xlsx')
P_SW96=pf.calculate_P_for_rho_T(
T_K=df['Temp in C']+273.15,
CO2_dens_gcm3=df['Density_g_cm3'],
EOS='SW96', Sample_ID=df['Sample'])
```

This returns a `pandas.DataFrame` with calculations done for each input row.

7.1.3 Comparing two EOS

`DiadFit` makes it very simple to compare calculations using the EOS of Sterner and Pitzer [1994] and Span and Wagner [1996]. In Figure 6, we calculate pressure for 11 discrete densities between 0.1 and 1.1 g cm^{-3} and temperatures between 34 and 2000 °C (see ReadTheDocs for a worked example). These calculations would have been tedious in existing tools, but can be performed in several lines of code in `DiadFit` (and in seconds of computational time). Figure 6 demonstrates that the EOS are extremely similar at temperatures relevant to basaltic magmas (<2 % discrepancies).

7.1.4 Converting homogenization temperatures to CO_2 densities

Microthermometry provides an alternative method to Raman spectroscopy to determine the density of CO_2-rich fluids. A

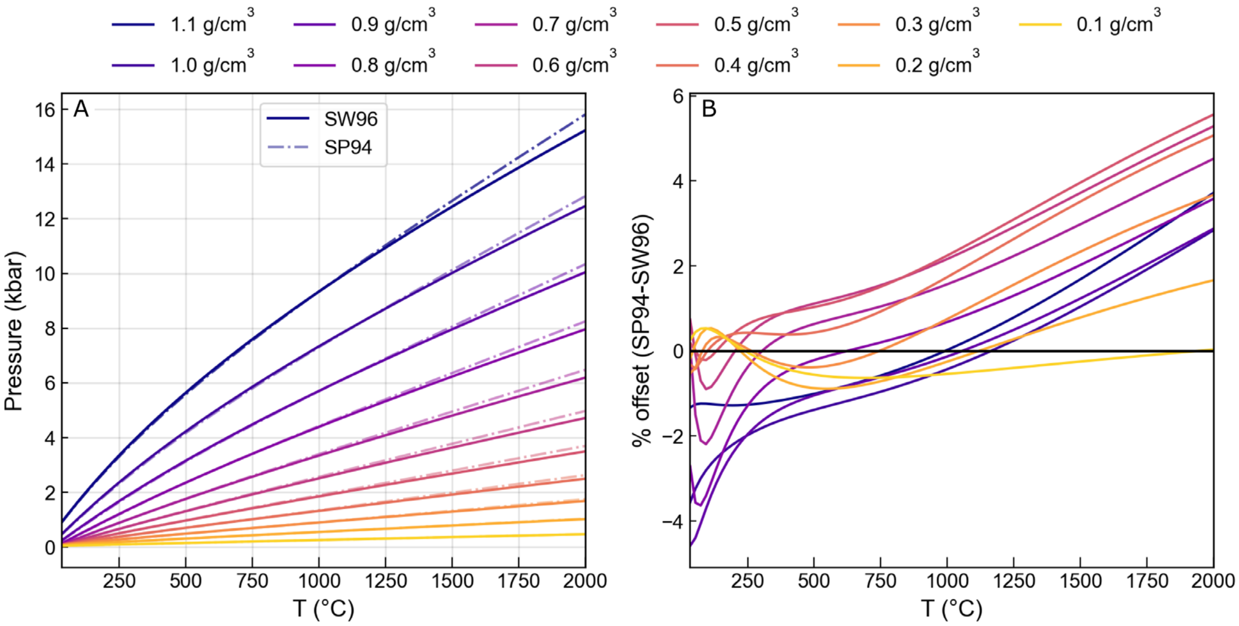


Figure 6: Comparison of the CO_2 EOS of Sterner and Pitzer [1994] and Span and Wagner [1996]. The y axis on [B] shows the pressure calculated by SP94 minus that from SW96 divided by the average of these two pressures.

fluid inclusion is cooled down, and then slowly heated up to determine the temperature at which a two-phase inclusion containing liquid and vapour transitions to a single homogenous phase (the homogenization temperature). CO_2 densities can be calculated from these homogenization temperatures using the function `calculate_CO2_density_homog_T`, which uses the homogenization temperature to calculate the pressure based on the position of the L-V phase boundary from Span and Wagner [1996]. Knowing the temperature and pressure of this transition, the CO_2 density can be calculated using the Span and Wagner [1996] EOS [e.g. Hansteen 1991; Kobayashi et al. 2012]. To calculate the CO_2 density of a fluid inclusion which homogenized at −18 °C to a liquid phase:

```
CalcDens=pf.calculate_CO2_density_homog_T(
T_h_C=-18, homog_to='L')
```

This returns a `dataframe` showing the bulk density, the density of the co-existing liquid and vapour just before homogenization, and the user-inputted parameters.

An entire excel sheet of homogenization temperatures and phases can be loaded as a `dataframe`. There is also an optional input of entering a sample name for each row (so it is returned in the outputted `dataframe`). Uncertainties in homogenization temperature can be propagated to determine errors in density using Monte Carlo techniques (see example on ReadTheDocs).

7.2 CO_2-H_2O EOS

In reality, many igneous systems will have some proportion of H_2O in the exsolved fluid phase, and thus the entrapped fluid inclusion. In this scenario, entering CO_2 densities measured by Raman spectroscopy or microthermometry into a pure CO_2 equation of state to calculate pressure is problematic, and a mixed CO_2-H_2O EOS must be used instead. However, detecting the presence of H_2O in fluid inclusions, let alone quantifying H_2O mole proportions required to perform calculations using a H_2O-CO_2 EOS, is very challenging for a number of reasons [Morgan et al. 1993]. First, H_2O may no longer be present in the inclusion, because of diffusive loss during stalling, ascent and syn-eruptive quenching, or because it reacted with the mineral host to form hydrous phases [Morgan et al. 1993; Mavrogenes and Bodnar 1994; Zanon et al. 2024]. If H_2O is still present, H_2O has a very low solubility in CO_2 fluids at temperatures typical of routine Raman analyses (~0.5 mol%, [Spycher et al. 2003; Frezzotti and Peccerillo 2007]). The small proportion of molecules that do dissolve will be characterized by a weak and relatively narrow (15 cm^{-1}) peak at ~3640 cm^{-1} that may not be detectable [Wopenka and Pasteris 1987; Frezzotti and Peccerillo 2007; Berkesi et al. 2009]). The remaining H_2O that cannot dissolve will be present as a very thin film on the outside of the inclusion; in inclusions 3–5 μm in size, ~20 mol% H_2O can be hosted in a water film only 0.2 μm thick [Frezzotti and Peccerillo 2007]. The extremely thin nature of this film makes is very challenging to detect optically or with Raman spectroscopy. Another issue is that at room temperature, peaks resulting from H_2O (dissolved or in films) are at ~3000–3800 cm^{-1} [Frezzotti et al. 2012], which is not in the typical spectral window used when measuring CO_2 at high spectral resolution, requiring a separate acquisition to detect them. Heating the fluid inclusion to ~120–150 °C, where the solubility of dissolved H_2O is far higher, results in a larger peak, which can be more easily identified/quantified [Berkesi et al. 2009]. However, perhaps because the H_2O has been lost in many cases, and even with heating, quantification is challenging, the majority of fluid inclusion studies simply pick a fixed value of X_{H_2O} (e.g. 0.1 for

Forte et al. [2023], Sandoval-Velasquez et al. [2023], and Zanon et al. [2024]). More recently, DeVitre and Wieser 2024 determine X_{H_2O} as a function of pressure for the volcanic system of interest using melt inclusion measurements (many solubility models used to calculate saturation pressures also calculate X_{H_2O} at the point of vapour saturation [e.g. Ghiorso and Gualda 2015])

To calculate pressures for mixed fluids, an assumption also must be made about whether H_2O was lost or remained in the fluid inclusion at the time of measurement. By default, `DiadFit` calculates the pressure for both scenarios. For the calculation assuming H_2O was lost prior to measurement, `DiadFit` follows the correction method of [Hansteen and Klügel 2008], where the measured CO_2 density is corrected back to a bulk density of the trapped CO_2-H_2O mix (where 44 is the molar mass of CO_2, 18 of H_2O; Figure 7D):

$$\alpha = X_{H_2O}/(1 - X_{H_2O}) \tag{8}$$

$$\rho_{bulk} = \rho_{CO_2,\,meas}\left(1 + \alpha\frac{18}{44}\right). \tag{9}$$

If H_2O is assumed to still be present in the fluid inclusion as a thin film of liquid around the edge, the bulk density at the time of trapping can be calculated as follows (assuming no volume changes; Figure 7E). Following the law of partial pressures:

$$P_{Tot} = P_{CO_2} + P_{H_2O} \tag{10}$$

$$P_{CO_2} = P_{Tot}X^{mol}_{CO_2} \tag{11}$$

$$P_{H_2O} = P_{Tot}X^{mol}_{H_2O}, \tag{12}$$

the pure CO_2 EOS is used to calculate the pressure of the CO_2 fluid (P_{CO_2}) from the measured density and analysis temperature (Figure 7E). Re-arranging Equation 11 allows calculation of P_{Tot}. Then Equation 12 is used to calculate P_{H_2O} for a fixed value of $X^{mol}_{H_2O}$. Unlike the scenario where H_2O is lost, this calculation requires knowledge of the analysis temperature, which is used alongside P_{H_2O} to calculate ρ_{H_2O} using the pure H_2O EOS of Wagner and Pruß [2002] (implemented in [Bell et al. 2014]). By converting $X^{mol}_{H_2O}$ to $X^{mass}_{H_2O}$, the volume of CO_2 and H_2O phases can be calculated from their densities. By re-arranging, ρ_{bulk} can be calculated, which can then be entered in a CO_2-H_2O EOS (Figure 7E).

`DiadFit` performs calculations using the CO_2-H_2O of [Duan and Zhang 2006]. This EOS was implemented in `DiadFit` by translating and adapting the `C` code of Yoshimura [2023]. In `DiadFit`, pressure or molar volume are solved iteratively from Equation 9 of Duan and Zhang [2006] using a Newton-Raphson method. At certain bulk densities and X_{H_2O} values, a negative pressure is converged upon during iteration. To resolve this issue, and other convergence issues, we generate a lookup table of molar volumes for 40 temperatures (T = 300–2000 K), 40 X_{H_2O} values (0–1), and 40 pressures (P = 1–100 kbar), yielding 62,000 lookup values for all possibly combinations of these (40 × 40 × 40). When a user enters a bulk density, this is first converted to molar volume, and then the closest match to the entered conditions in the look up table is identified, and the corresponding pressure is used as the start point for the algorithm. This resolves convergence issues, and speeds up computation time.

Overall, to convert fluid inclusion CO_2 densities into pressures using the CO_2-H_2O EOS, users should use the function `calculate_entrapment_P_XH2O`, specifying measured CO_2 densities, entrapment temperature, analysis temperature, and $X^{mol}_{H_2O}$. The function returns the pressure for the mixed EOS for the H_2O loss and no H_2O loss scenario, and the calculated bulk densities. It also returns pressures for the pure CO_2 EOS of SW96, SP94 and Duan and Zhang [2006] for comparison. Unlike existing tools, this allows pressures to be calculated for hundreds of fluid inclusions very quickly, using different X_{H_2O} for each inclusion if so desired. This function can also be used to visualize the magnitude of the correction factor for different CO_2 densities and molar proportions of H_2O in the fluid (e.g. Figure 8).

We also include the function `calc_prop_knownP_EOS_DZ2006` for calculating molar volumes, compressability factors, fugacity and activities of each species using the Duan and Zhang [2006] EOS where pressure, temperature and X_{H_2O} are known. We anticipate this will be useful for experimentalists and/or those building solubility models or other parameterizations requiring this thermodynamic data.

8 Additional calculations relating to FI barometry

8.1 Converting Pressures into depths

Building on functionality in `Thermobar` [Wieser et al. 2022], `DiadFit` can convert pressures to depths using various options (Figure 1H) implemented in the function `convert_pressure_to_depth`:

- A fixed crustal density (e.g. ρ = 2700 kg m^{-3})
- A 2-step or 3-step crustal density profile (e.g. ρ = 2700 kg m^{-3} at <10 km, ρ = 3300 kg m^{-3} at >10 km)
- Pressure-depth models of:

1. A crustal profile for arc magmas from Rasmussen et al. [2022] (`'rasmussen'`).
2. The combined model of Mavko and Thompson [1983] and DeBari and Greene [2011] as parameterized by Putirka [2017] (`'mavko_debari'`) for use in continental arcs.
3. A crustal profile for Hawaii from Hill and Zuccal [1987] as parameterized by Putirka [2017] (`'hill_zucca'`)
4. A crustal profile for Hawaii from Ryan [1988] parameterized by Lerner et al. [2021] (`'ryan_lerner'`).

We will add more density profiles as they are published, so we encourage users to check the documentation for more details.

8.2 Modeling fluid inclusion re-equilibration

DeVitre and Wieser [2024] implement a `Python3` version of the mechanical re-equilibration model for olivine-hosted fluid inclusions of Wanamaker and Evans [1989] in their package `RelaxiFI`. We incorporate this methodology into `DiadFit` for

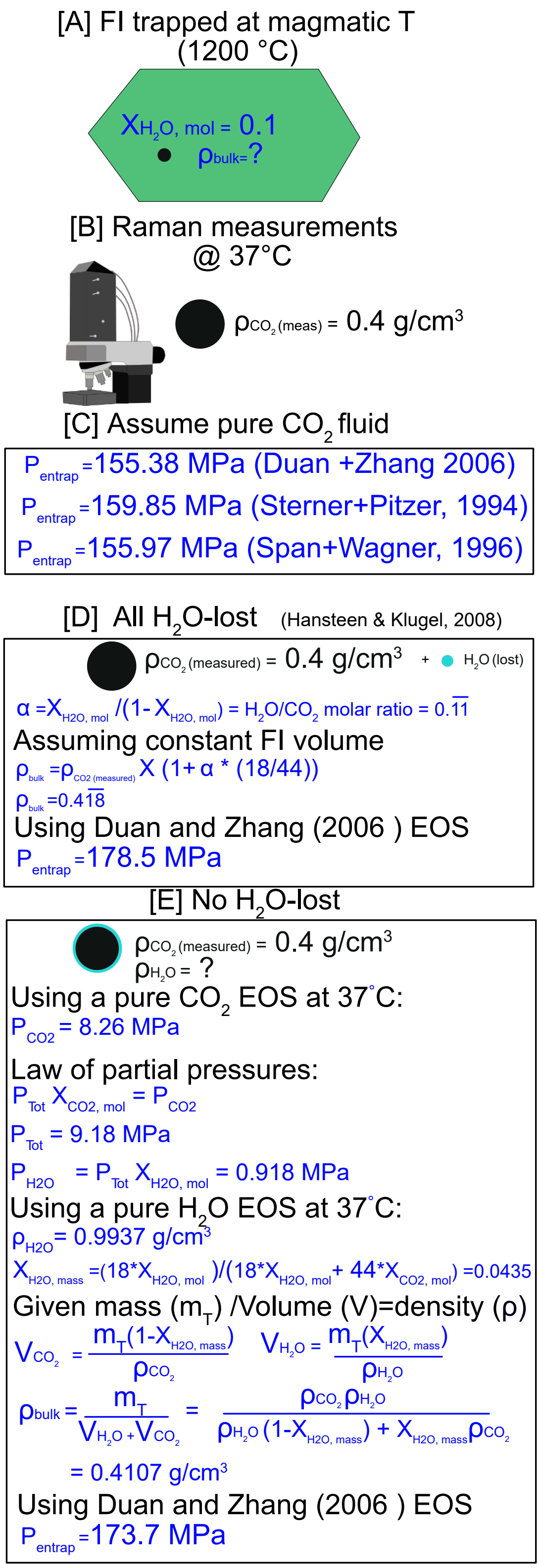

[A] FI trapped at magmatic T (1200 °C)
$X_{H_2O,\ mol}$ = 0.1
ρ_{bulk}=?
[B] Raman measurements @ 37°C
$\rho_{CO_2\,(meas)}$ = 0.4 g/cm^3
[C] Assume pure CO_2 fluid
P_{entrap}=155.38 MPa (Duan +Zhang 2006)
P_{entrap}=159.85 MPa (Sterner+Pitzer, 1994)
P_{entrap}=155.97 MPa (Span+Wagner, 1996)
[D] All H_2O-lost (Hansteen & Klugel, 2008)
$\rho_{CO_2\,(measured)}$ = 0.4 g/cm^3 + H_2O (lost)
$\alpha = X_{H2O,\ mol} / (1 - X_{H2O,\ mol})$ = H_2O/CO_2 molar ratio = $0.\overline{11}$
Assuming constant FI volume
$\rho_{bulk} = \rho_{CO2\ (measured)} \times (1 + \alpha * (18/44))$
$\rho_{bulk} = 0.4\overline{18}$
Using Duan and Zhang (2006) EOS
P_{entrap}=178.5 MPa
[E] No H_2O-lost
$\rho_{CO_2\,(measured)}$ = 0.4 g/cm^3
ρ_{H_2O} = ?
Using a pure CO_2 EOS at 37°C:
P_{CO2} = 8.26 MPa
Law of partial pressures:
$P_{Tot}\, X_{CO2,\ mol} = P_{CO2}$
P_{Tot} = 9.18 MPa
$P_{H2O} = P_{Tot}\, X_{H2O,\ mol}$ = 0.918 MPa
Using a pure H_2O EOS at 37°C:
ρ_{H2O}= 0.9937 g/cm^3
$X_{H2O,\ mass} = (18 * X_{H2O,\ mol}) / (18 * X_{H2O,\ mol} + 44 * X_{CO2,\ mol})$ =0.0435
Given mass (m_T) /Volume (V)=density (ρ)
$V_{CO_2} = \frac{m_T(1-X_{H2O,\ mass})}{\rho_{CO_2}}$ $V_{H_2O} = \frac{m_T(X_{H2O,\ mass})}{\rho_{H_2O}}$
$\rho_{bulk} = \frac{m_T}{V_{H_2O} + V_{CO_2}} = \frac{\rho_{CO_2}\rho_{H_2O}}{\rho_{H_2O}(1-X_{H2O,\ mass}) + X_{H2O,\ mass}\rho_{CO_2}}$
= 0.4107 g/cm^3
Using Duan and Zhang (2006) EOS
P_{entrap}=173.7 MPa

Figure 7: Schematic diagram showing corrections for H_2O for a fluid inclusion with a CO_2 density of 0.4 g cm^{-3}, and $X^{mol}_{H_2O}$=0.1.

ease of installation and maintenance, because it uses the underlying EOS and density-depth conversions from DiadFit to track changes in fluid inclusion volume, density, and pressure for different stalling, ascent, and quenching paths. Worked examples are available on ReadTheDocs. If this model is used, it should be described as follows "fluid inclusion re-equilibration was assessed using RelaxiFI [DeVitre and Wieser 2024], a Python3 adaptation of the mechanical re-equilibration model of Wanamaker and Evans [1989] implemented through DiadFit v.x.x.x".

For example, Figure 9 shows an example tracking the change in fluid inclusion radius, CO_2 density and internal pressure of a FI during isothermal ascent (1200 ° C) towards the surface. Specifically, this model considers FI with an initial radius of 1 µm (R) located 100 µm away from a crystal defect (b) ascending from 10 km to the surface in 20 steps (depth_path_ini_fin_step with an ascent rate of 1 m s^{-1} (ascent_rate_ms), which is 0.03 MPa s^{-1}. Relationships between pressure and density are tracked using the Span and Wagner [1996] EOS (SW96). The function returns a dataframe with columns for time, external pressure change, internal pressure change, stretching rate, FI radius, fractional change in radius, CO_2 density and depth. Depth can be calculated using the different models available in DiadFit and is an optional argument in the function above. As well as a dataframe, the function also returns a figure (Figure 9).

9 Quantifying Uncertainty

DiadFit also allows rigorous propagation of uncertainty through various volcanological workflows.

9.1 Uncertainty when determining CO_2 density

DiadFit provides a framework for propagating the three main sources of analytical uncertainty when quantifying CO_2 density by Raman spectroscopy:

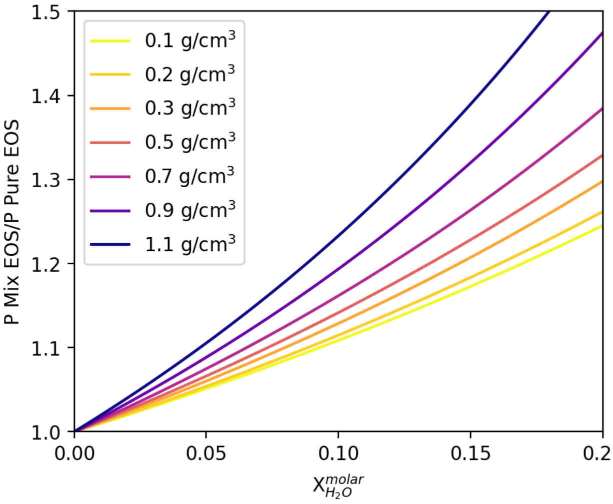
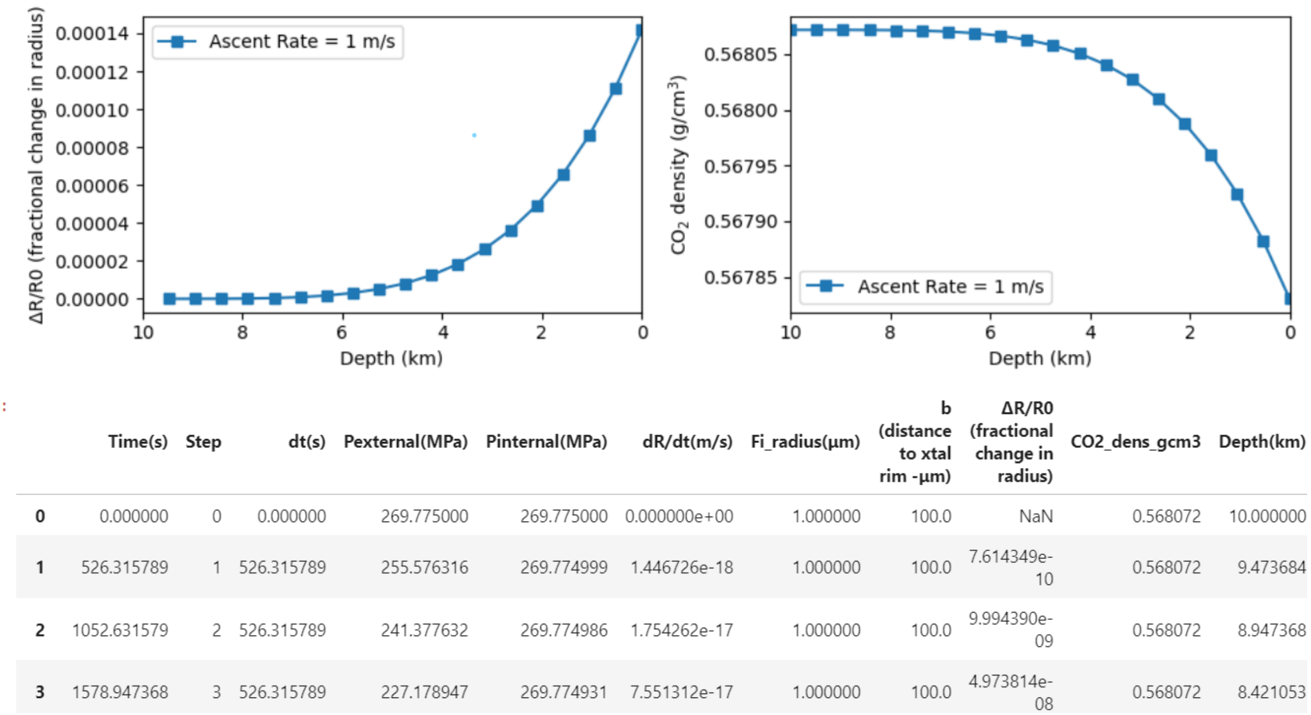


Figure 8: Difference between the mixed CO_2-H_2O EOS and the pure CO_2 EOS of Duan and Zhang [2006] for fluid inclusions with different measured CO_2 densities (under the assumption of H_2O loss).



	Time(s)	Step	dt(s)	Pexternal(MPa)	Pinternal(MPa)	dR/dt(m/s)	Fi_radius(μm)	b (distance to xtal rim -μm)	ΔR/R0 (fractional change in radius)	CO2_dens_gcm3	Depth(km)
0	0.000000	0	0.000000	269.775000	269.775000	0.000000e+00	1.000000	100.0	NaN	0.568072	10.000000
1	526.315789	1	526.315789	255.576316	269.774999	1.446726e-18	1.000000	100.0	7.614349e-10	0.568072	9.473684
2	1052.631579	2	526.315789	241.377632	269.774986	1.754262e-17	1.000000	100.0	9.994390e-09	0.568072	8.947368
3	1578.947368	3	526.315789	227.178947	269.774931	7.551312e-17	1.000000	100.0	4.973814e-08	0.568072	8.421053

Figure 9: Screenshot of the output from `pf.stretch_in_ascent` for isothermal ascent (1200 ° C) towards the surface for a FI with an initial radius of 1 μm for an ascent rate of 1 m s^{-1}. The returned figure shows the fractional change in the fluid inclusion radius as a function of depth (e.g. it gets bigger as it ascents), and the accompanying drop in CO$_2$ density caused by the larger FI volume. The returned `pandas.DataFrame` shows changes in internal and external pressure, fluid inclusion radius, along with other parameters.

1. Peak fitting error when determining the center of each diad peak (Figure 10A–B).

2. Error associated with correcting for instrument drift (Figure 10A–C).

3. Error associated with the densimeter used to convert drift-corrected splitting into CO$_2$ density (Figure 10D).

`DiadFit` obtains the 1σ error associated with each peak center from `lmfit` (which uses the covariance matrix). As each peak is fitted independently because the Diad1 and Diad2 peaks do not overlap, the error associated with the position of each peak center is combined in quadrature to get the error on the peak splitting. The Ne line correction factor vs time regression model (Figure 5) is used to determine the error on the Ne correction factor for each acquisition (Figure 10C). This error is combined with the splitting error in quadrature to get the uncertainty in the corrected splitting (black crosses, Figure 10A). To propagate the uncertainty associated with the densimeter, `DiadFit` uses the regression model through the calibration data. This error is combined with the error on the corrected splitting to yield the overall error in density (σ `Density g/cm3`, Figure 10D). Assessing the relative contribution from each source of uncertainty helps optimization of analytical routines (Figure 10). If the majority of the error is being introduced at the peak fitting stage, it may be that longer acquisitions are required to get stronger signals, or a higher resolution grating is required. Figure 10A, F shows that a few acquisitions at ~0.18–0.2 g m^{-3} have far larger peak fitting errors than other acquisitions from the same analytical session. These anomalously large errors would justify inspection of the spectra and fits in more detail, and perhaps re-analysis.

9.2 Uncertainty in fluid inclusion pressures

`DiadFit` can be used to propagate uncertainty when calculating pressures and depths from fluid inclusions based on Raman and microthermometry measurements. The function `propagate_FI_uncertainty` allows propagation of uncertainties in input parameters using Monte Carlo methods. The function input arguments are the magnitude and distribution of uncertainty for CO$_2$ density, temperature, the mol fraction of H$_2$O (X_{H_2O}), and the crustal density. Input errors for each variable can be absolute (e.g. ±50 K for temperature) or percentage errors (e.g. ±5 %), and can follow a normal or uniform distribution. For example, a user may specify that the uncertainty in their temperature is best approximated with a normal distribution with a standard deviation of 50 K. For each fluid inclusion, the function generates N duplicates of each input parameter randomly drawn from these specified prior distributions. It is assumed that all input parameters are independent of each other. Pressure and depths are calculated for each of these N duplicates. The function outputs a figure showing the simulation for a single specified fluid inclusion (here, FI1, Figure 11), in addition to two `dataframes`. The first `dataframe` shows the mean, median, standard deviation of calculated pressures and depths for each fluid inclusion (Figure 11, Output1). For skewed distributions, the

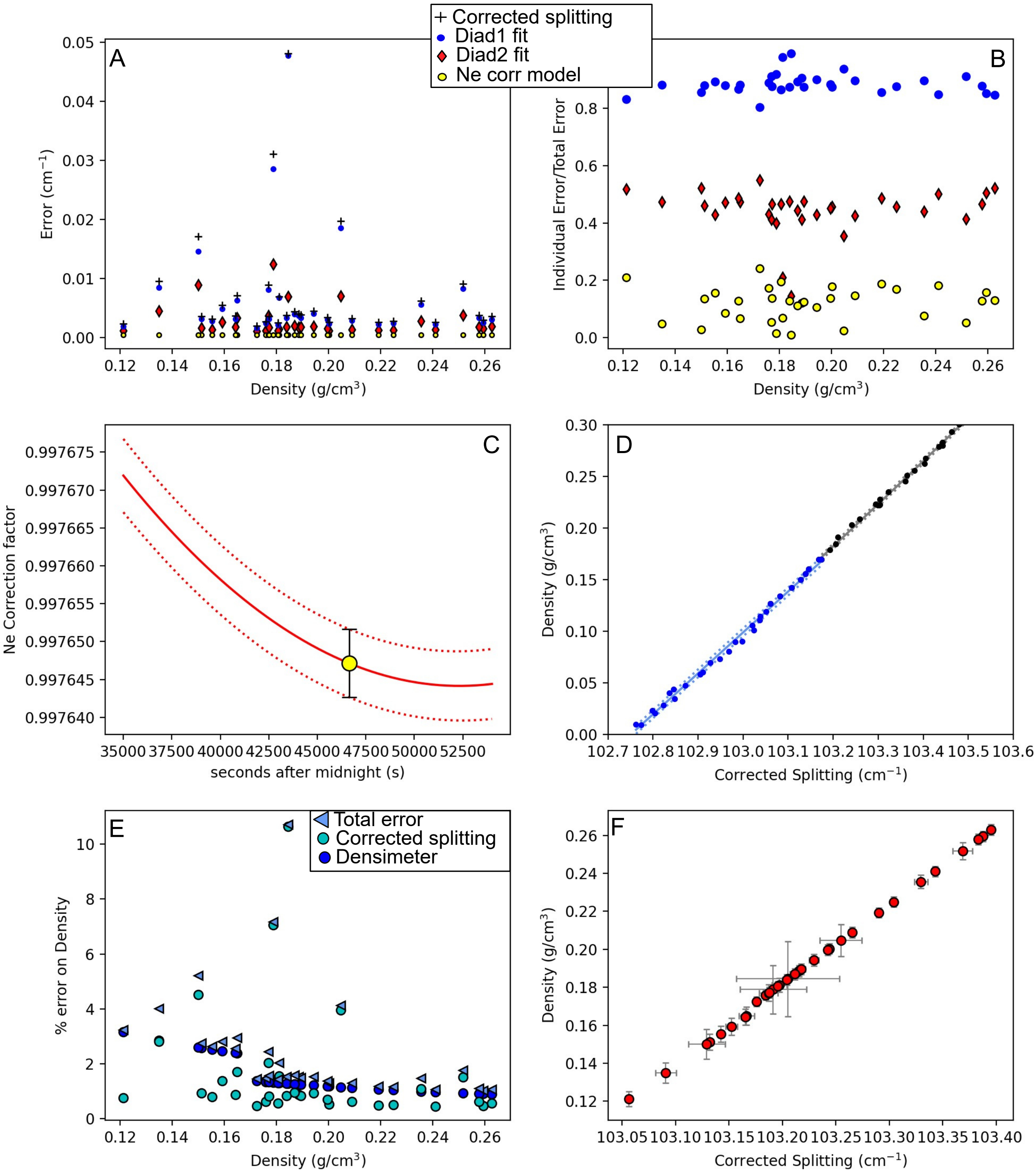


Figure 10: Assessing different sources of analytical error. [A] The black cross shows the total error on the corrected splitting from propagating the error in fitting CO_2 peaks and Ne correction model for an instrument with a spectral resolution of 0.56 cm^{-1}. Individual contributions from each peak and the Ne correction model are shown. Note, the total error is smaller than the sum of each error shown, as a result of propagating these errors in quadrature. [B] To allow easier comparison of the proportional contribution from each source of error, we also take each individual error and divide by the total error on the corrected splitting. This show that the fitting error on Diad1 is the largest source of uncertainty. [C] Ne regression model with uncertainty represented by red-dashed lines, along with the uncertainty for a single diad acquisition. [D] Error associated with the densimeter shown with dashed lines, with the underlying acquisitions shown as dots. The densimeter is regressed in three parts for low, medium, and high densities (see [DeVitre et al. 2021]). [E] Percentage % error on calculated density, showing the contribution from errors on the corrected splitting (see [A]) and from the densimeter. [F] 1 σ errors on density and corrected splitting.

standard deviation may be best approximated by half the difference between the 84th − 16th percentile (also provided). The second dataframe showing the results for all simulations: if $N = 1000$ duplicates and $N = 10$ fluid inclusions, the dataframe has 10,000 rows (Figure 11, Output2). Monte Carlo simulations using the Duan and Zhang [2006] EOS are computationally expensive. To optimize efficiency for the propogate_FI_uncertainty function, calculations using this equation of state utilize the python multiprocessing package [McKerns et al. 2011]. This speeds up calculations ~4× for a 32GB laptop with 8 processors. This option can be turned off and on for all calculations using if 'multiprocess=True' or 'False'.

9.3 Uncertainty in equivalent CO_2 contents of vapour bubbles

The Raman method of reconstructing vapour bubbles present within melt inclusions relies on measuring the CO_2 density, and then converting this into an equivalent amount of CO_2 that would have been dissolved in the glass at the time of melt inclusion entrapment by mass balance:

$$CO_2^{\text{equivalent glass ppm}} = 10^4 \frac{VB\text{vol}\ \% \cdot \rho_{CO_2}}{\rho_{\text{melt}}} \tag{13}$$

where ρ_{CO_2} is the density of CO_2, ρ_{melt} is the density of silicate melt, and $VB_{\text{vol}\ \%}$ is the volume % of the inclusion occupied by the vapour bubble.

Similar to the fluid inclusion method described above, DiadFit contains functionality to propagate the uncertainties in each of these variables using Monte Carlo methods. The function propagate_CO2_in_bubble allows users to specify the magnitude of the error for each parameter, whether the error is a % error or an absolute error, and whether the error is normally or uniformly distributed. This function returns two dataframes—the first with the average and standard deviation for each melt inclusion, and the second showing all simulation outputs. The uncertainty in CO_2 density can be obtained from DiadFit. The uncertainty in melt density can be obtained from Iacovino and Till [2019]. Uncertainty in estimating bubble volumes depend greatly on the method used and the geometry of inclusions [e.g. Tucker et al. 2019; DeVitre et al. 2023b].

10 Other useful functions for volcanological workflows

10.1 Quantifying peak asymmetry/skewness to identify co-existing liquid and gaseous CO_2

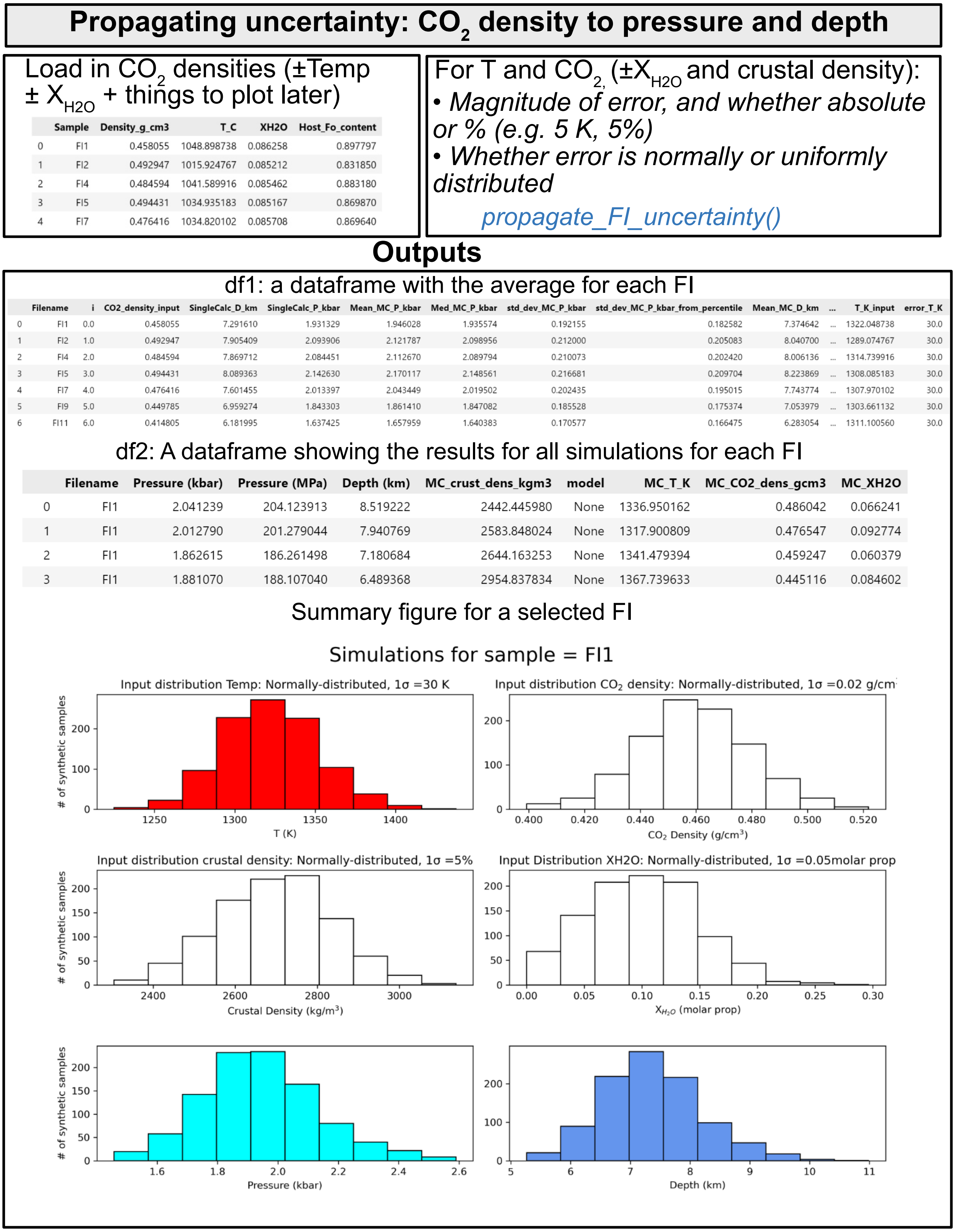
At room temperature (18–22 °C), a CO_2 fluid with a density between ~0.18–0.21 and 0.75–0.79 g cm^{-3} will consist of an inner shell of CO_2 vapour, and an outer shell of CO_2 liquid [Span and Wagner 1996]. Using the calculate_CO2_homog_T function, we can calculate that at 20 °C the gaseous phase will have a density of 0.194 g cm^{-3} and the liquid will have a density of 0.773 g cm^{-3}. DeVitre et al. [2023a] show that Raman measurements performed in inclusions with two co-existing phases at room temperature do not always show two distinct peaks for the gaseous and liquid CO_2 phase, but often show a single skewed peak, with contributions from both phases. Quantifying peak asymmetry can help to identify such spectra, which cannot be reliably quantified for CO_2 density.

To assess asymmetry in each spectra, an N^{th} degree polynomial is fitted between specified baseline positions, and this background is subtracted. A cubic-spline is fitted between the baseline positions. The x and y coordinates of the highest point on this spline fit are identified. The position on each shoulder of the peak with an intensity equal to a certain fraction of the peak height is identified. In Figure 12, this cut-off parameter int_cut_off is set at 0.3. The ratio of the x-distance between the peak center and these shoulder points (green and grey lines in Figure 12) defines the peak asymmetry. After tweaking these parameters for one spectra, the function loop_diad_skewness can be used to loop through all files in a specific folder, and stitch the results into a dataframe. Typical skewness parameters have to be identified for each instrument, after which spectra with high skewness can be identified, and reanalysed about the critical temperature where a single peak will be present [see DeVitre et al. 2023a].

10.2 H_2O fitting

It is well established that Raman spectroscopy can be used to quantify H_2O within silicate glasses. The glass alumino-silicate region consists of multiple bands; a low frequency/low wavenumber (LF or LW) band centered on ~550 cm^{-1}, a medium frequency/wavenumber (MF, MW) band centered at ~550 cm^{-1}, and a high frequency/wavenumber (HF, HW) band at ~1000 cm^{-1} (Figure 13, see [van Gerve and Namur 2023] for detailed description of the relevant vibrations). The H_2O region is characterized by a broad, asymmetrical peak centered at ~3600 cm^{-1}. There are a number of different methods to use these different peaks to quantify H_2O; some use the height or intensity of the water Raman band (~3000-3800 cm^{-1}), and others using the ratio of the area under the water region divided by the area under the alumino-silicate band at ~200–1250 cm^{-1} [Thomas 2000; Zajacz et al. 2005; Behrens et al. 2006; Thomas et al. 2006; Mercier et al. 2010; Le Losq et al. 2012; Di Genova et al. 2017; Schiavi et al. 2018]. The exact relationship between peak areas/ratios/heights and the H_2O content depends on instrument hardware and the glass major element composition, so standards with known H_2O contents should be used for calibration (see Schiavi et al. [2018] and references within).

In terms of quantification strategies, Behrens et al. [2006] provide two empirical equations relating silicate:water areas to absolute water contents. One equation is for albite, haplogranite, and dacite samples, and the other is for intermediate and depolymerized compositions. Le Losq et al. [2012] build on this to produce a global empirical calibration. Importantly, they note that different silicate glass compositions require different background positions prior to quantifying silicate peak areas. Schiavi et al. [2018] refine this method further, noting that as well as being sensitive to glass composition, the baseline is also sensitive to redox state. They suggest fitting a default cubic baseline through predefined baseline positions for four melt compositions (basalt, basanite, andesite, rhyolite) results in the area ratio (LW+HW±MW to H_2O area) being reasonably independent of melt composition. However,



Propagating uncertainty: CO_2 density to pressure and depth

Load in CO_2 densities (±Temp ± X_{H2O} + things to plot later)

	Sample	Density_g_cm3	T_C	XH2O	Host_Fo_content
0	FI1	0.458055	1048.898738	0.086258	0.897797
1	FI2	0.492947	1015.924767	0.085212	0.831850
2	FI4	0.484594	1041.589916	0.085462	0.883180
3	FI5	0.494431	1034.935183	0.085167	0.869870
4	FI7	0.476416	1034.820102	0.085708	0.869640

For T and CO_2, (± X_{H2O} and crustal density):

- *Magnitude of error, and whether absolute or % (e.g. 5 K, 5%)*
- *Whether error is normally or uniformly distributed*

propagate_FI_uncertainty()

Outputs

df1: a dataframe with the average for each FI

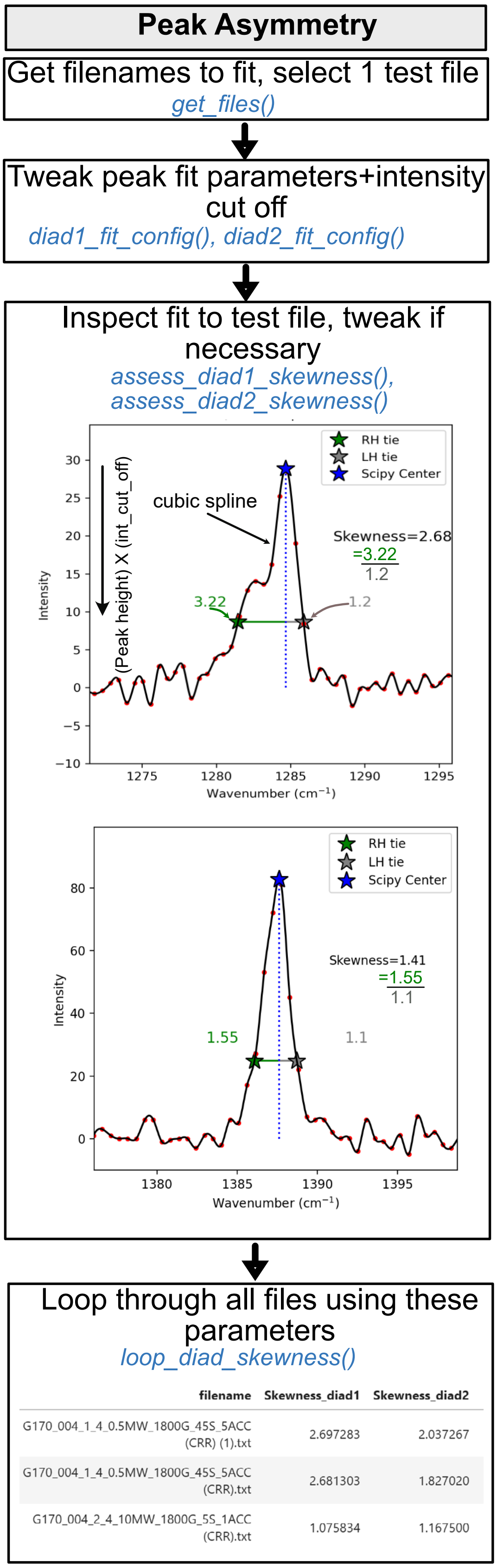
	Filename	i	CO2_density_input	SingleCalc_D_km	SingleCalc_P_kbar	Mean_MC_P_kbar	Med_MC_P_kbar	std_dev_MC_P_kbar	std_dev_MC_P_kbar_from_percentile	Mean_MC_D_km	...	T_K_input	error_T_K
0	FI1	0.0	0.458055	7.291610	1.931329	1.946028	1.935574	0.192155	0.182582	7.374642	...	1322.048738	30.0
1	FI2	1.0	0.492947	7.905409	2.093906	2.121787	2.098956	0.212000	0.205083	8.040700	...	1289.074767	30.0
2	FI4	2.0	0.484594	7.869712	2.084451	2.112670	2.089794	0.210073	0.202420	8.006136	...	1314.739916	30.0
3	FI5	3.0	0.494431	8.089363	2.142630	2.170117	2.148561	0.216681	0.209704	8.223869	...	1308.085183	30.0
4	FI7	4.0	0.476416	7.601455	2.013397	2.043449	2.019502	0.202435	0.195015	7.743774	...	1307.970102	30.0
5	FI9	5.0	0.449785	6.959274	1.843303	1.861410	1.847082	0.185528	0.175374	7.053979	...	1303.661132	30.0
6	FI11	6.0	0.414805	6.181995	1.637425	1.657959	1.640383	0.170577	0.166475	6.283054	...	1311.100560	30.0

df2: A dataframe showing the results for all simulations for each FI

	Filename	Pressure (kbar)	Pressure (MPa)	Depth (km)	MC_crust_dens_kgm3	model	MC_T_K	MC_CO2_dens_gcm3	MC_XH2O
0	FI1	2.041239	204.123913	8.519222	2442.445980	None	1336.950162	0.486042	0.066241
1	FI1	2.012790	201.279044	7.940769	2583.848024	None	1317.900809	0.476547	0.092774
2	FI1	1.862615	186.261498	7.180684	2644.163253	None	1341.479394	0.459247	0.060379
3	FI1	1.881070	188.107040	6.489368	2954.837834	None	1367.739633	0.445116	0.084602

Summary figure for a selected FI

Figure 11: Schematic showing the workflow used to propagate uncertainty in CO_2 density, Temperature, X_{H_2O}, and crustal density into error distributions in pressure and depth for each fluid inclusion.



filename	Skewness_diad1	Skewness_diad2
G170_004_1_4_0.5MW_1800G_45S_5ACC (CRR) (1).txt	2.697283	2.037267
G170_004_1_4_0.5MW_1800G_45S_5ACC (CRR).txt	2.681303	1.827020
G170_004_2_4_10MW_1800G_5S_1ACC (CRR).txt	1.075834	1.167500

Figure 12: Schematic showing how DiadFit can be used to assess diad assymetry/skewness following the method of DeVitre et al. [2023a].

González-García et al. [2021] note that the presence of nanolites with a peak at 670–690 cm^{-1} complicate total silicate area methods, because this nanolite region overlaps with the LW silicate region used in the “total area method” of Schiavi et al. [2018]. They propose a new protocol using only the HW area to characterize the silicate portion, rather than the overall silicate region as in Schiavi et al. [2020].

To allow maximum flexibility when using these different methods, DiadFit quantifies three different silicate band areas, and the overall silicate and water area (Figure 13), with lots of options for users to tweak background positions, under the assumption protocols will continue to change. When fitting spectra acquired on hydrous glasses in DiadFit, the first step is to select the files of interest. As for diad peaks and Ne lines, default peak fit parameters are stored in dataclasses. There are dataclasses for the four suggested background positions of Schiavi et al. [2018], e.g. sil_bck_pos_Schiavi_basalt, sil_bck_pos_Schiavi_andesite. Users can tweak these background positions for their specific samples if they wish (e.g. for a basaltic andesite). For example, the lower background position can be easily changed for basalt from the default of 300–340 to 320–350 cm^{-1}:

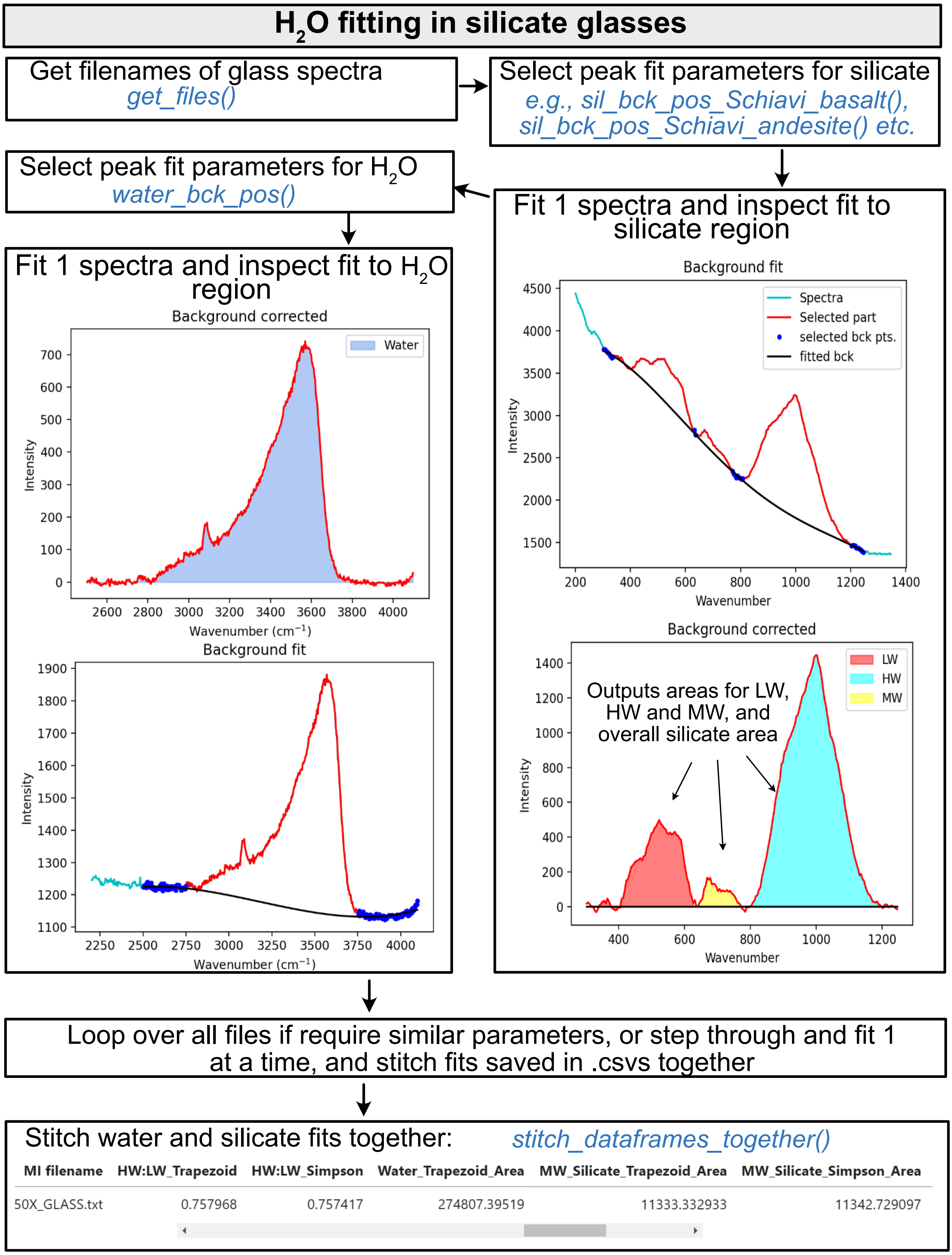
```
pf.sil_bck_pos_Schiavi_basalt(
lower_range_sil=[320, 350])
```

These dataclasses also store options for baseline fitting. After removing baseline points outside of a specified sigma - either a polynomial of degree N or a cubic spline can be fit to all the datapoints within the baseline region. After subtracting away this polynomial, the area under the background-corrected spectra is calculated using the Simpson and the trapezoid method [Tallarida and Murray 1987] implemented in SciPy and NumPy respectively. Overall, the function function returns these areas, along with the ratio of the silicate to water areas for each area method (as well as other useful parameters such as the positions of choosen baselines, degrees of polynomials etc. for future reference).

10.2.1 H_2O fitting within unexposed melt inclusions

A common criticism of the Raman method when quantifying the CO_2 content of vapour bubbles is that it does not account for the sequestration of carbon as solid carbonate phases on the inclusion walls [e.g. Moore et al. 2015]. DeVitre et al. [2023b] developed a technique where melt inclusions with carbonate-bearing vapour bubbles are heated to their liquidus temperature using a Linkam TS1400XY stage, causing carbonate to redissolve as CO_2 which can then be measured by Raman spectroscopy. A concern with any heating method when working with melt inclusions is the possibility of diffusive loss of H_2O [Chen et al. 2011; Gaetani et al. 2012]. DeVitre et al. [2023b] develop a method to assess H_2O loss by acquiring Raman spectra on unexposed melt inclusion glasses before and after homogenization.

Quantifying H_2O in spectra acquired from unexposed melt inclusions is more complicated than in exposed glasses, because the Raman signal contains a contribution from the glass and the overlying olivine host. The contribution from the host



H_2O fitting in silicate glasses

Get filenames of glass spectra *get_files()*

Select peak fit parameters for silicate *e.g., sil_bck_pos_Schiavi_basalt(), sil_bck_pos_Schiavi_andesite() etc.*

Select peak fit parameters for H_2O *water_bck_pos()*

Fit 1 spectra and inspect fit to silicate region

Fit 1 spectra and inspect fit to H_2O region

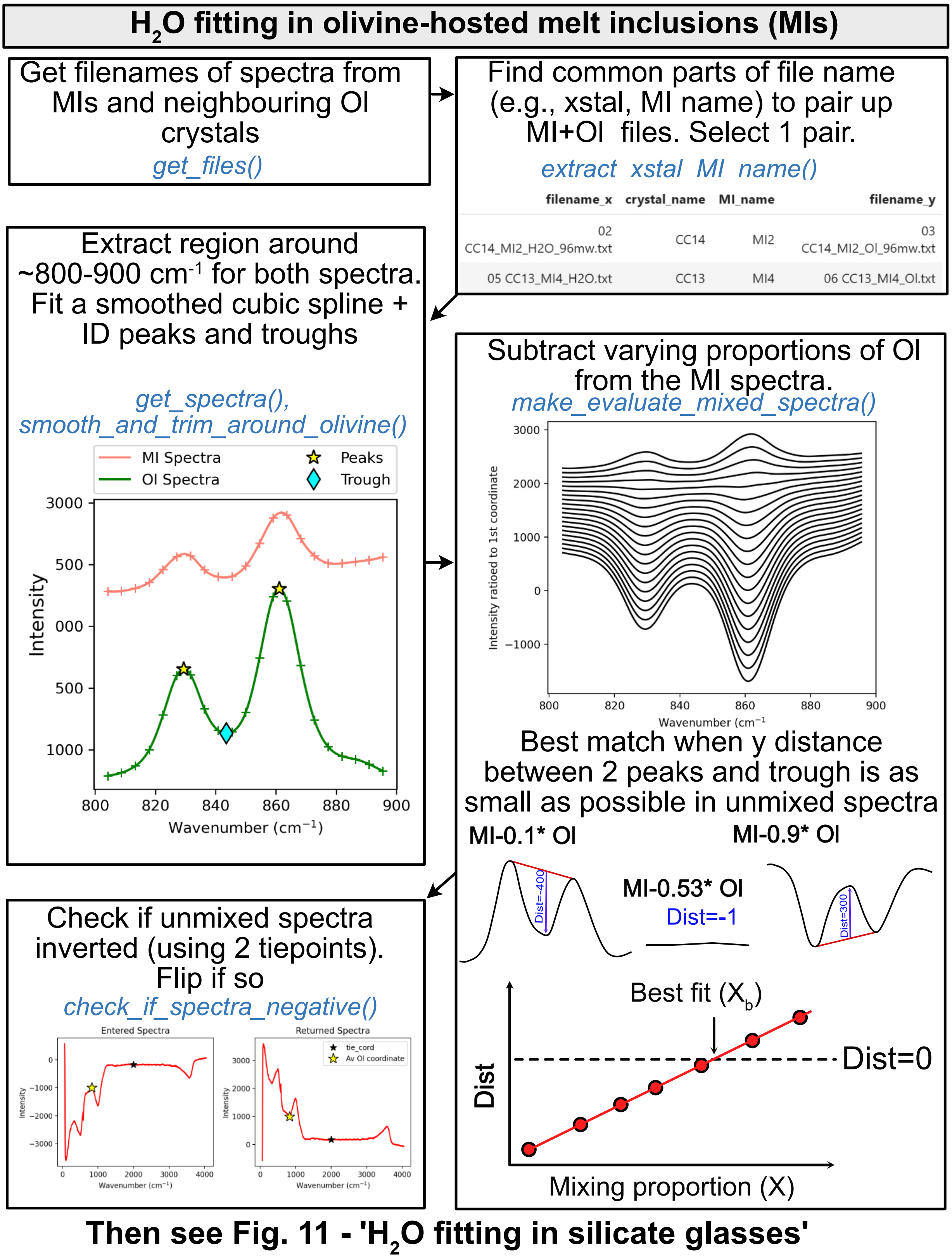
Outputs areas for LW, HW and MW, and overall silicate area

Loop over all files if require similar parameters, or step through and fit 1 at a time, and stitch fits saved in .csvs together

Stitch water and silicate fits together: *stitch_dataframes_together()*

MI filename	HW:LW_Trapezoid	HW:LW_Simpson	Water_Trapezoid_Area	MW_Silicate_Trapezoid_Area	MW_Silicate_Simpson_Area
50X_GLASS.txt	0.757968	0.757417	274807.39519	11333.332933	11342.729097

Figure 13: Schematic showing how DiadFit can be used quantify the relative areas of silicate and H_2O peaks in Raman spectra collected from silicate glasses.


H_2O fitting in olivine-hosted melt inclusions (MIs)
Get filenames of spectra from MIs and neighbouring Ol crystals
get_files()
Find common parts of file name (e.g., xstal, MI name) to pair up MI+Ol files. Select 1 pair.
extract_xstal_MI_name()
Extract region around ~800-900 cm^{-1} for both spectra. Fit a smoothed cubic spline + ID peaks and troughs
get_spectra(), smooth_and_trim_around_olivine()
MI Spectra
Ol Spectra
Peaks
Trough
3000
500
000
500
1000
Intensity
800
820
840
860
880
900
Wavenumber (cm^{-1})
Subtract varying proportions of Ol from the MI spectra.
make_evaluate_mixed_spectra()
3000
2000
1000
0
−1000
Intensity ratioed to 1st coordinate
800
820
840
860
880
900
Wavenumber (cm^{-1})
Best match when y distance between 2 peaks and trough is as small as possible in unmixed spectra
MI-0.1* Ol
Dist=-400
MI-0.53* Ol
Dist=-1
MI-0.9* Ol
Dist=300
Best fit (X_b)
Dist
Dist=0
Mixing proportion (X)
Check if unmixed spectra inverted (using 2 tiepoints). Flip if so
check_if_spectra_negative()
Entered Spectra
0
−1000
−2000
−3000
Intensity
0
1000
2000
3000
4000
Wavenumber (cm^{-1})
Returned Spectra
tie_cord
Av Ol coordinate
3000
2000
1000
0
Intensity
0
1000
2000
3000
4000
Wavenumber (cm^{-1})
Then see Fig. 11 - 'H_2O fitting in silicate glasses'

filename_x	crystal_name	MI_name	filename_y
02 CC14_MI2_H2O_96mw.txt	CC14	MI2	03 CC14_MI2_Ol_96mw.txt
05 CC13_MI4_H2O.txt	CC13	MI4	06 CC13_MI4_Ol.txt

Figure 14: Schematic showing how DiadFit can be used to unmix the contribution of olivine and glass from spectra taken on unexposed melt inclusions. The code could be easily adapted for other silicate phases. There is an option to loop if glass compositions are sufficiently similar. Alternatively, each file can be stepped through manually to tweak positions, and the code saves the output for each file to a .csv, and these are then stitched together after all files are fitted.

mineral must be subtracted to obtain an "unmixed spectra", allowing reliable estimation of the silicate area. Fortunately, most magmatic olivine crystals have very low H_2O contents; typically a few 10s of ppm, [Newcombe et al. 2020; Towbin et al. 2023], and occasionally up to 100s of ppm [Wang et al. 2022]. Thus, peaks in the H_2O region of the spectra are extremely weak/completely absent, even at laser powers far higher than is typical for glass analyses (~70–140 mW, [Martinek and Bolfan-Casanova 2021]), which simplifies demixing of the signals. Here, we describe the demixing method used for olivine-hosted melt inclusions, although it could be adapted for other phases which do not produce significant peaks in the H_2O region. We note that demixing and deconvolution of glass from other signals is also available in the program SilicH2O [van Gerve and Namur 2023].

To reliably unmix the spectra, it is best to acquire a spectra in the melt inclusion at the depth where the H_2O peak is the strongest, and a spectra of the olivine next to the melt inclusion. DiadFit extracts the region of these two spectra between 800–900 cm^{-1} containing the strong olivine doublet. A cubic spline is fitted to both spectra to smooth out noise. SciPy is used to find the positions of the two strong peaks in the olivine spectra, and the trough position between these peaks. DiadFit then creates N spectra, where the olivine spectra is subtracted from the mixed spectra. For each of these mixed spectra, a linear regression is fit between the two points with the wavenumbers of the identified olivine peaks. The vertical (y) distance is then calculated between this linear regression and the y value at the x-cordinate of the trough (Figure 14). When too little signal from the olivine has been subtracted from the mixed spectra, the trough position will be lower than the linear regression (e.g. Dist = −400 in sketch 1 on Figure 14, and when too much olivine has been subtracted, the peaks will invert, and the distance will be a positive number. When exactly the right amount of olivine has been subtracted, there will be no clear peaks or troughs, so the distance will be zero. The function fits a curve to this calculated distance vs. the mixing proportion, to determine the best-fit mixing proportion where the distance is zero. The resulting, "unmixed" spectra is taken as the spectra from the melt inclusion itself. Then, the workflow discussed above can be applied to quantify the relative silicate and H_2O area.

11 Future Work

The open-source nature of DiadFit means that users can customize functions and build their own (either on a local fork, or using a pull request on GitHub). Additionally, we anticipate that we will continue to add workflows as new applications of Raman spectroscopy appear in volcanology. To reflect the evolving nature of this tool, when citing DiadFit, users should make sure they specify the version they used, obtained using:

```
pf.__version__
```

Care should also be taken to cite the root packages used by DiadFit. e.g. if converting CO_2 densities to pressures using the functions here, we recommend wording similar to this: "CO_2 densities were converted into pressures using DiadFit (Wieser and DeVitre, 2024, v.x.x.x), using the Equation of State of Span and Wagner [1996] implemented in CoolProp [Bell et al. 2014].

12 Conclusions

DiadFit is a high-level Python package that provides easy-to-use functions for common workflows involving Raman spectroscopy and microthermometry (e.g. peak fitting, EOS calculations etc.), with particular applicability to the analysis of melt inclusions and fluid inclusions. DiadFit not only greatly reduces the time spent on data processing/reduction, it also includes numerous functions to propagate uncertainty. In turn, this will be vital to identify the biggest sources of uncertainty, and identify ways to minimise these errors in many difficult volcanological applications.

Author Contributions

PW conceived the project and lead the code writing (apart from the functions discussed below). CD performed extensive testing and tweaking of existing functions. She also wrote the cosmic ray removal code, the fluid inclusion relaxation code, and the code to calculate molar proportions from gas peak areas. She conceptualized the peak asymmetry method, which PW implemented in Python.

Acknowledgements

We are grateful to Felix Boschetty, Lowell Moore, Chelsea Allison, Sarah Hickernell, Allan Lerner, Bob Bodnar, and Kenneth Befus for providing example Raman file formats from different instruments. We thank Lyndsey Fisher, and Kyle Dayton for providing feedback on the code.This work was supported by UC Berkeley start up funds, the Rose Hills innovator award, a Sloan Research Fellowship, and NSF grants 2217371 and 2342156. We thank Oryaëlle Chevrel for editorial handling, Charles Le Losq and one anonymous reviewer for greatly improving the quality of the package and manuscript.

Data Availability

All files are available on GitHub https://github.com/PennyWieser/DiadFit. Documentation is available at ReadTheDocs (https://DiadFit.readthedocs.io/en/latest/), and YouTube videos explaining various aspects of the tool are available on the DiadFit YouTube channel bit.ly/DiadFitYouTube.

Copyright Notice

References

Allison, C. M., K. Roggensack, and A. B. Clarke (2021). "Highly explosive basaltic eruptions driven by CO2 exsolution". *Nature communications* 12(1), pages 1–10. DOI: https://doi.org/10.1038/s41467-020-20354-2.

Bakker, R. J. (2003). "Package FLUIDS 1. Computer programs for analysis of fluid inclusion data and for modelling bulk fluid properties". *Chemical Geology* 194(1-3), pages 3–23.

– (2021). "The perfection of Raman spectroscopic gas densimeters". *Journal of Raman Spectroscopy* 52(11), pages 1923–1948. DOI: https://doi.org/10.1002/jrs.6245.

Behrens, H., J. Roux, D. R. Neuville, and M. Siemann (2006). "Quantification of dissolved H2O in silicate glasses using confocal microRaman spectroscopy". *Chemical Geology* 229(1-3), pages 96–112. DOI: 10.1016/j.chemgeo.2006.01.014.

Bell, I. H., J. Wronski, S. Quoilin, and V. Lemort (2014). "Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp". *Industrial & engineering chemistry research* 53(6), pages 2498–2508. DOI: https://doi.org/10.1021/ie4033999.

Berkesi, M., K. Hidas, T. Guzmics, J. Dubessy, R. J. Bodnar, C. Szabo, B. Vajna, and T. Tsunogae (2009). "Detection of small amounts of H2O in CO2-rich fluid inclusions using Raman spectroscopy". *Journal of Raman Spectroscopy: An International Journal for Original Work in all Aspects of Raman Spectroscopy, Including Higher Order Processes, and also Brillouin and Rayleigh Scattering* 40(11), pages 1461–1463. DOI: 10.1002/jrs.2440.

Brown, P. E. (1989). "FLINCOR; a microcomputer program for the reduction and investigation of fluid-inclusion data". *American Mineralogist* 74(11-12), pages 1390–1393.

Burke, E. A. (2001). "Raman microspectrometry of fluid inclusions". *Lithos* 55(1-4), pages 139–158. DOI: https://doi.org/10.1016/S0024-4937(00)00043-8.

Chen, Y., A. Provost, P. Schiano, and N. Cluzel (2011). "The rate of water loss from olivine-hosted melt inclusions". *Contributions to Mineralogy and Petrology* 162, pages 625–636. DOI: https://doi.org/10.1007/s00410-011-0616-5.

Dayton, K., E. Gazel, P. Wieser, V. R. Troll, J. C. Carracedo, H. La Madrid, D. C. Roman, J. Ward, M. Aulinas, H. Geiger, et al. (2023). "Deep magma storage during the 2021 La Palma eruption". *Science Advances* 9(6), eade7641. DOI: 10.1126/sciadv.ade7641.

DeBari, S. M. and A. R. Greene (2011). "Vertical stratification of composition, density, and inferred magmatic processes in exposed arc crustal sections". *Arc-continent collision*. Edited by D. Brown and P. D. Ryan. Springer, pages 121–144. DOI: https://doi.org/10.1007/978-3-540-88558-0_5.

DeVitre, C. L., C. M. Allison, and E. Gazel (2021). "A high-precision CO2 densimeter for Raman spectroscopy using a Fluid Density Calibration Apparatus". *Chemical Geology* 584, page 120522. DOI: https://doi.org/10.1016/j.chemgeo.2021.120522.

DeVitre, C. L., K. Dayton, E. Gazel, A. Pamukçu, G. Gaetani, and P. E. Wieser (2023a). "Laser heating effect on Raman analysis of CO2 co-existing as liquid and vapor in olivine-hosted melt inclusion bubbles". *Volcanica* 6(2), pages 201–219. DOI: https://doi.org/10.30909/vol.06.02.201219.

DeVitre, C. L., E. Gazel, R. S. Ramalho, S. Venugopal, M. Steele-MacInnis, J. Hua, C. M. Allison, L. R. Moore, J. C. Carracedo, and B. Monteleone (2023b). "Oceanic intraplate explosive eruptions fed directly from the mantle". *Proceedings of the National Academy of Sciences* 120(33), e2302093120. DOI: https://doi.org/10.1073/pnas.2302093120.

DeVitre, C. L. and P. E. Wieser (2024). "Reliability of Raman analyses of CO2-rich fluid inclusions as a geobarometer at Kīlauea". *Geochemical Perspective Letters* 29, pages 1–8. DOI: 10.7185/geochemlet.2404.

Di Genova, D., S. Sicola, C. Romano, A. Vona, S. Fanara, and L. Spina (2017). "Effect of iron and nanolites on Raman spectra of volcanic glasses: A reassessment of existing strategies to estimate the water content". *Chemical Geology* 475, pages 76–86. DOI: https://doi.org/10.1016/j.chemgeo.2017.10.035.

Duan, Z. and Z. Zhang (2006). "Equation of state of the H2O, CO2, and H2O–CO2 systems up to 10 GPa and 2573.15 K: Molecular dynamics simulations with ab initio potential surface". *Geochimica et cosmochimica acta* 70(9), pages 2311–2324. DOI: 10.1016/j.gca.2006.02.009.

Fermi, E. (1931). "Über den Ramaneffekt des Kohlendioxyds". *Zeitschrift für Physik* 71, pages 250–259.

Forte, F. M. L., A. Aiuppa, S. G. Rotolo, and V. Zanon (2023). "Temporal evolution of the Fogo Volcano magma storage system (Cape Verde Archipelago): a fluid inclusions perspective". *Journal of Volcanology and Geothermal Research* 433, page 107730. DOI: 10.1016/j.jvolgeores.2022.107730.

Frezzotti, M. L., F. Tecce, and A. Casagli (2012). "Raman spectroscopy for fluid inclusion analysis". *Journal of Geochemical Exploration* 112, pages 1–20. DOI: https://doi.org/10.1016/j.gexplo.2011.09.009.

Frezzotti, M.-L. and A. Peccerillo (2007). "Diamond-bearing COHS fluids in the mantle beneath Hawaii". *Earth and Planetary Science Letters* 262(1-2), pages 273–283. DOI: 10.1016/j.epsl.2007.08.001.

Fukura, S., T. Mizukami, S. Odake, and H. Kagi (2006). "Factors determining the stability, resolution, and precision of a conventional Raman spectrometer". *Applied Spectroscopy* 60(8), pages 946–950. DOI: https://doi.org/10.1366/000370206778062.

Gaetani, G. A., J. A. O'Leary, N. Shimizu, C. E. Bucholz, and M. Newville (2012). "Rapid reequilibration of H_2O and oxygen fugacity in olivine-hosted melt inclusions". *Geology* 40(10), pages 915–918. DOI: https://doi.org/10.1130/G32992.1.

Georgiev, D., S. V. Pedersen, R. Xie, A. Fernández-Galiana, M. M. Stevens, and M. Barahona (2023). "RamanSPy: An open-source Python package for integrative Raman spectroscopy data analysis". *arXiv preprint arXiv:2307.13650*.

Ghiorso, M. S. and G. A. Gualda (2015). "An H 2 O–CO 2 mixed fluid saturation model compatible with rhyolite-MELTS". *Contributions to Mineralogy and Petrology* 169, pages 1–30. DOI: 10.1007/s00410-015-1141-8.

Giordano, D., J. K. Russell, D. González-García, D. Bersani, D. B. Dingwell, and C. Del Negro (2020). "Raman spectroscopy from laboratory and proximal to remote sensing: a tool for the volcanological sciences". *Remote Sensing* 12(5), page 805. DOI: https://doi.org/10.3390/rs12050805.

González-García, D., D. Giordano, A. Allabar, F. Andrade, L. Polo, V. d. A. Janasi, A. Lucchetti, K.-U. Hess, C. De Campos, and D. Dingwell (2021). "Retrieving dissolved H2O content from micro-Raman spectroscopy on nanolitized silicic glasses: application to volcanic products of the Paraná Magmatic Province, Brazil". *Chemical Geology* 567, page 120058. DOI: https://doi.org/10.1016/j.chemgeo.2021.120058.

González-García, D., D. Giordano, J. K. Russell, and D. B. Dingwell (2020). "A Raman spectroscopic tool to estimate chemical composition of natural volcanic glasses". *Chemical Geology* 556, page 119819. DOI: https://doi.org/10.1016/j.chemgeo.2020.119819.

Hansteen, T. H. (1991). "Multi-stage evolution of the picritic Maelifell rocks, SW Iceland: constraints from mineralogy and inclusions of glass and fluid in olivine". *Contributions to Mineralogy and Petrology* 109(2), pages 225–239. DOI: https://doi.org/10.1007/BF00306481.

Hansteen, T. H. and A. Klügel (2008). "Fluid inclusion thermobarometry as a tracer for magmatic processes". *Reviews in Mineralogy and Geochemistry* 69(1), pages 143–177. DOI: https://doi.org/10.2138/rmg.2008.69.5.

Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al. (2020). "Array programming with NumPy". *Nature* 585(7825), pages 357–362. DOI: https://doi.org/10.1038/s41586-020-2649-2.

Hartley, M. E., J. Maclennan, M. Edmonds, and T. Thordarson (2014). "Reconstructing the deep CO2 degassing behaviour of large basaltic fissure eruptions". *Earth and Planetary Science Letters* 393, pages 120–131. DOI: https://doi.org/10.1016/j.epsl.2014.02.031.

Hill, D. P. and J. J. Zuccal (1987). "Geophysical constraints on the structure of Kilauea and Mauna Loa Volcanoes and some implications for seismomagmatic processe". *US Geological Survey Professional Paper 1350* 2(1350), page 903. DOI: http://pubsdata.usgs.gov/pubs/pp/1987/1350/chapters.html.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". *Computing in Science & Engineering* 9(3), pages 90–95. DOI: 10.1109/MCSE.2007.55.

Iacovino, K. and C. B. Till (2019). "DensityX: A program for calculating the densities of magmatic liquids up to 1,627 C and 30 kbar". *Volcanica* 2(1), pages 1–10. DOI: https://doi.org/10.30909/vol.02.01.0110.

Kobayashi, T., J. Yamamoto, T. Hirajima, H. Ishibashi, N. Hirano, Y. Lai, V. S. Prikhod'Ko, and S. Arai (2012). "Conformity and precision of CO2 densimetry in CO2 inclusions: Microthermometry versus Raman microspectroscopic densimetry". *Journal of Raman Spectroscopy* 43(8), pages 1126–1133. DOI: https://doi.org/10.1002/jrs.3134.

Lamadrid, H., L. Moore, D. Moncada, J. Rimstidt, R. Burruss, and R. Bodnar (2017). "Reassessment of the Raman CO2 densimeter". *Chemical Geology* 450, pages 210–222. DOI: https://doi.org/10.1016/j.chemgeo.2016.12.034.

Le, V.-H., M.-C. Caumon, and A. Tarantola (2021). "FRAnCIs calculation program with universal Raman calibration data for the determination of PVX properties of CO2–CH4–N2 and CH4–H2O–NaCl systems and their uncertainties". *Computers & Geosciences* 156, page 104896.

Le Losq, C. (2018). "Rampy: a Python library for processing spectroscopic (IR, Raman, XAS...) data." DOI: 10.5281/zenodo.1168729.

Le Losq, C., D. R. Neuville, R. Moretti, and J. Roux (2012). "Determination of water content in silicate glasses using Raman spectrometry: Implications for the study of explosive volcanism". *American Mineralogist* 97(5-6), pages 779–790. DOI: 10.2138/am.2012.3831.

Lerner, A. H., P. J. Wallace, T. Shea, A. J. Mourey, P. J. Kelly, P. A. Nadeau, T. Elias, C. Kern, L. E. Clor, C. Gansecki, et al. (2021). "The petrologic and degassing behavior of sulfur and other magmatic volatiles from the 2018 eruption of Kīlauea, Hawaiʻi: melt concentrations, magma storage depths, and magma recycling". *Bulletin of Volcanology* 83(6), pages 1–32. DOI: https://doi.org/10.1007/s00445-021-01459-y.

Lin, F., R. Bodnar, and S. Becker (2007). "Experimental determination of the Raman CH4 symmetric stretching (v1) band position from 1–650 bar and 0.3–22 C: Application to fluid inclusion studies". *Geochimica et Cosmochimica Acta* 71(15), pages 3746–3756. DOI: https://doi.org/10.1016/j.gca.2007.05.016.

Lu, W., X. Wang, Q. Wan, W. Hu, I.-M. Chou, and Y. Wan (2023). "In situ Raman spectroscopic measurement of the 13C/12C ratio in CO2: Experimental calibrations on the effects of fluid pressure, temperature and composition". *Chemical Geology* 615, page 121201. DOI: 10.1016/j.chemgeo.2022.121201.

Martinek, L. and N. Bolfan-Casanova (2021). "Water quantification in olivine and wadsleyite by Raman spectroscopy and study of errors and uncertainties". *American Mineralogist* 106(4), pages 570–580. DOI: 10.2138/am-2021-7264.

Mavko, B. B. and G. A. Thompson (1983). "Crustal and upper mantle structure of the northern and central Sierra Nevada". *Journal of Geophysical Research: Solid Earth* 88(B7), pages 5874–5892. DOI: https://doi.org/10.1029/JB088iB07p05874.

Mavrogenes, J. and R. Bodnar (1994). "Hydrogen movement into and out of fluid inclusions in quartz: Experimental evidence and geologic implications". *Geochimica et Cosmochimica Acta* 58(1), pages 141–148. DOI: 10.1016/0016-7037(94)90452-9.

McKerns, M. M., L. Strand, T. Sullivan, A. Fang, and M. A. G. Aivazis (2011). "Building a framework for predictive science". *Proceedings of the 10th Python in Science Conference*. DOI: 10.48550/arXiv.1202.1056.

Mercier, M., A. Di Muro, N. Métrich, D. Giordano, O. Belhadj, and C. W. Mandeville (2010). "Spectroscopic analysis (FTIR, Raman) of water in mafic and intermediate glasses and glass inclusions". *Geochimica et Cosmochimica Acta* 74(19), pages 5641–5656. DOI: https://doi.org/10.1016/j.gca.2010.06.020.

Moore, L. R., E. Gazel, R. Tuohy, A. S. Lloyd, R. Esposito, M. Steele-MacInnis, E. H. Hauri, P. J. Wallace, T. Plank, and R. J. Bodnar (2015). "Bubbles matter: An assessment of the contribution of vapor bubbles to melt inclusion volatile budgets". *American Mineralogist* 100(4), pages 806–823. DOI: https://doi.org/10.2138/am-2015-5036.

Morgan, G., I.-M. Chou, J. Pasteris, and S. Olsen (1993). "Reequilibration of CO2 fluid inclusions at controlled hydrogen fugacities". *Journal of Metamorphic Geology* 11(1), pages 155–164.

Morizet, Y., R. A. Brooker, G. Iacono-Marziano, and B. A. Kjarsgaard (2013). "Quantification of dissolved CO2 in silicate glasses using micro-Raman spectroscopy". *American Mineralogist* 98(10), pages 1788–1802. DOI: https://doi.org/10.2138/am.2013.4516.

Newcombe, M. E., T. Plank, A. Barth, P. D. Asimow, and E. Hauri (2020). "Water-in-olivine magma ascent chronometry: Every crystal is a clock". *Journal of Volcanology and Geothermal Research* 398, page 106872. DOI: 10.1016/j.jvolgeores.2020.106872.

Putirka, K. D. (2017). "Down the crater: where magmas are stored and why they erupt". *Elements* 13(1), pages 11–16. DOI: https://doi.org/10.2113/gselements.13.1.11.

Rasmussen, D. J., T. A. Plank, D. C. Roman, and M. M. Zimmer (2022). "Magmatic water content controls the pre-eruptive depth of arc magmas". *Science* 375(6585), pages 1169–1172. DOI: 10.1126/science.abm5174.

Roedder, E. (2018). "Reviews in mineralogy". *Fluid inclusions*. Edited by P. H. Ribbe. De Gruyter, pages II–II. DOI: 10.1515/9781501508271.

Rosso, K. and R. Bodnar (1995). "Microthermometric and Raman spectroscopic detection limits of CO_2 in fluid inclusions and the Raman spectroscopic characterization of C02". *Geochimica et Cosmochimica Acta* 59(19), pages 3961–3975. DOI: https://doi.org/10.1016/0016-7037(95)94441-H.

Ryan, M. P. (1988). "The mechanics and three-dimensional internal structure of active magmatic systems: Kilauea Volcano, Hawaii". *Journal of Geophysical Research: Solid Earth* 93(B5), pages 4213–4248. DOI: https://doi.org/10.1029/JB093iB05p04213.

Sandoval-Velasquez, A., A. Rizzo, F. Casetta, T. Ntaflos, A. Aiuppa, M. Alonso, E. Padrón, M. Pankhurst, A. Mundl-Petermeier, V. Zanon, et al. (2023). "The noble gas signature of the 2021 Tajogaite eruption (La Palma, Canary Islands)". *Journal of Volcanology and Geothermal Research* 443, page 107928. DOI: https://doi.org/10.1016/j.jvolgeores.2023.107928.

Schiavi, F., N. Bolfan-Casanova, R. Buso, M. Laumonier, D. Laporte, K. Medjoubi, S. Venugopal, A. Gómez-Ulla, N. Cluzel, and M. Hardiagon (2020). "Quantifying magmatic volatiles by Raman microtomography of glass inclusion-hosted bubbles". *Geochemical Perspectives Letters* 16, pages 17–24. DOI: 10.7185/geochemlet.2038.

Schiavi, F., N. Bolfan-Casanova, A. C. Withers, E. Médard, M. Laumonier, D. Laporte, T. Flaherty, and A. Gómez-Ulla (2018). "Water quantification in silicate glasses by Raman spectroscopy: Correcting for the effects of confocality, density and ferric iron". *Chemical Geology* 483, pages 312–331. DOI: https://doi.org/10.1016/j.chemgeo.2018.02.03.

Span, R. and W. Wagner (1996). "A new equation of state for carbon dioxide covering the fluid region from the triple-point temperature to 1100 K at pressures up to 800 MPa". *Journal of physical and chemical reference data* 25(6), pages 1509–1596. DOI: https://doi.org/10.1063/1.555991.

Spycher, N., K. Pruess, and J. Ennis-King (2003). "CO2-H2O mixtures in the geological sequestration of CO2. I. Assessment and calculation of mutual solubilities from 12 to 100 C and up to 600 bar". *Geochimica et cosmochimica acta* 67(16), pages 3015–3031. DOI: https://doi.org/10.1016/S0016-7037(03)00273-4.

Steele-Macinnis, M., R. Esposito, and R. J. Bodnar (2011). "Thermodynamic model for the effect of post-entrapment crystallization on the H2O–CO2 systematics of vapor-saturated, silicate melt inclusions". *Journal of Petrology* 52(12), pages 2461–2482. DOI: https://doi.org/10.1093/petrology/egr052.

Sterner, S. M. and K. S. Pitzer (1994). "An equation of state for carbon dioxide valid from zero to extreme pressures". *Contributions to Mineralogy and Petrology* 117(4), pages 362–374. DOI: https://doi.org/10.1007/BF00307271.

Tallarida, R. J. and R. B. Murray (1987). "Area under a curve: trapezoidal and Simpson's rules". *Manual of Pharmacologic Calculations*. Springer, pages 77–81.

The pandas development team (2020). "pandas-dev/pandas: Pandas". Version latest. DOI: 10.5281/zenodo.3509134.

Thomas, R. (2000). "Determination of water contents of granite melt inclusions by confocal laser Raman microprobe spectroscopy". *American Mineralogist* 85(5-6), pages 868–872. DOI: 10.2138/am-2000-5-631.

Thomas, R., V. S. Kamenetsky, and P. Davidson (2006). "Laser Raman spectroscopic measurements of water in unexposed glass inclusions". *American Mineralogist* 91(2-3), pages 467–470. DOI: 10.2138/am.2006.2107.

Towbin, H. W., T. Plank, E. Klein, and E. Hauri (2023). "Measuring H2O concentrations in olivine by secondary ion mass spectrometry: Challenges and paths forward". *American Mineralogist* 108(5), pages 928–940. DOI: 10.2138/am-2022-8247.

Tucker, J. M., E. H. Hauri, A. J. Pietruszka, M. O. Garcia, J. P. Marske, and F. A. Trusdell (2019). "A high carbon content of the Hawaiian mantle from olivine-hosted melt inclusions". *Geochimica et Cosmochimica Acta* 254, pages 156–172. DOI: https://doi.org/10.1016/j.gca.2019.04.001.

Van Gerve, T. D. and O. Namur (2023). "SilicH2O: a graphical user interface for processing silicate glass Raman spectra and quantifying H2O". DOI: https://doi.org/10.30909/vol.06.02.405413.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods* 17, pages 261–272. DOI: 10.1038/s41592-019-0686-2.

Wagner, W. and A. Pruß (2002). "The IAPWS formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use". *Journal of physical and chemical reference data* 31(2), pages 387–535. DOI: 10.1063/1.1461829.

Wanamaker, B. J. and B. Evans (1989). "Mechanical re-equilibration of fluid inclusions in San Carlos olivine by power-law creep". *Contributions to Mineralogy and Petrology* 102(1), pages 102–111. DOI: https://doi.org/10.1007/BF01160194.

Wang, C., Z. Zhang, A. Giuliani, S. Demouchy, C. Thoraval, L. Krmíček, H. Bo, W. Zhang, and X. Xia (2022). "Hydrogen concentrations and He isotopes in olivine from ultramafic lamprophyres provide new constraints on a wet Tarim plume and Earth's deep water cycle". *Journal of Geophysical Research: Solid Earth* 127(12), e2022JB024961. DOI: 10.1029/2022JB024961.

Wang, W. and W. Lu (2022). "High-precision carbon isotopic composition estimation for individual CO2 fluid inclusions through Raman spectroscopy". *Geoscience Frontiers*, page 101528. DOI: 10.1016/j.gsf.2022.101528.

Wang, X., I.-M. Chou, W. Hu, R. C. Burruss, Q. Sun, and Y. Song (2011). "Raman spectroscopic measurements of CO2 density: Experimental calibration with high-pressure optical cell (HPOC) and fused silica capillary capsule (FSCC) with application to fluid inclusion observations". *Geochimica et Cosmochimica Acta* 75(14), pages 4080–4093. DOI: https://doi.org/10.1016/j.gca.2011.04.028.

Wieser, P., M. Petrelli, J. Lubbers, E. Wieser, S. Ozaydin, A. Kent, and C. Till (2022). "Thermobar: an open-source Python3 tool for thermobarometry and hygrometry". *Volcanica* 5(2), pages 349–384. DOI: https://doi.org/10.30909/vol.05.02.349384.

Wieser, P. E., H. Lamadrid, J. Maclennan, M. Edmonds, S. Matthews, K. Iacovino, F. E. Jenner, C. Gansecki, F. Trusdell, R. L. Lee, et al. (2021). "Reconstructing magma storage depths for the 2018 Kīlauean eruption from melt inclusion CO2 contents: the importance of vapor bubbles". *Geochemistry, Geophysics, Geosystems* 22(2), e2020GC009364. DOI: 10.1029/2020GC009364.

Wojdyr, M. (2010). "Fityk: a general-purpose peak fitting program". *Journal of Applied Crystallography* 43(5-1), pages 1126–1128. DOI: https://doi.org/10.1107/S0021889810030499.

Wopenka, B. and J. D. Pasteris (1987). "Raman intensities and detection limits of geochemically relevant gas mixtures for a laser Raman microprobe". *Analytical Chemistry* 59(17), pages 2165–2170. DOI: 10.1021/ac00144a034.

Yoshimura, S. (2023). "Carbon dioxide and water in the crust. Part 1: Equation of state for the fluid". *Journal of Mineralogical and Petrological Sciences* 118(1), 221224a. DOI: 10.2465/jmps.221224a.

Yuan, X. and R. A. Mayanovic (2017). "An empirical study on Raman peak fitting and its application to Raman quantitative research". *Applied spectroscopy* 71(10), pages 2325–2338. DOI: https://doi.org/10.1177/0003702817721527.

Zajacz, Z., W. Halter, W. J. Malfait, O. Bachmann, R. J. Bodnar, M. M. Hirschmann, C. W. Mandeville, Y. Morizet, O. Müntener, P. Ulmer, et al. (2005). "A composition-independent quantitative determination of the water content in silicate glasses and silicate melt inclusions by confocal Raman spectroscopy". *Contributions to Mineralogy and Petrology* 150, pages 631–642. DOI: 10.1007/s00410-005-0040-9.

Zanon, V., L. D'Auria, F. Schiavi, K. Cyrzan, and M. J. Pankhurst (2024). "Toward a near real-time magma ascent monitoring by combined fluid inclusion barometry and ongoing seismicity". *Science Advances* 10(6), eadi4300. DOI: 10.1126/sciadv.adi4300.