

# Thermobar: An open-source Python3 tool for thermobarometry and hygrometry

 Penny E. Wieser<sup>\*α,β</sup>,  Maurizio Petrelli<sup>γ</sup>,  Jordan Lubbers<sup>δ</sup>,  Eric Wieser<sup>ε</sup>,  Sinan Özaydın<sup>ζ</sup>,  Adam J. R. Kent<sup>α</sup>, and  Christy B. Till<sup>η</sup>

<sup>α</sup> College of Earth, Ocean, and Atmospheric Sciences, Oregon State University, Oregon, U.S.A.

<sup>β</sup> Department of Earth and Planetary Sciences, UC Berkeley, California, U.S.A.

<sup>γ</sup> Department of Physics and Geology, University of Perugia, Perugia, Italy.

<sup>δ</sup> U.S. Geological Survey Alaska Volcano Observatory, Alaska, U.S.A.

<sup>ε</sup> Department of Engineering, Cambridge University, Cambridge, U.K.

<sup>ζ</sup> School of Natural Sciences, Macquarie University, Sydney, Australia.

<sup>η</sup> School of Earth and Space Exploration, Arizona State University, Arizona, U.S.A.

## ABSTRACT

We present Thermobar, a new open-source Python3 package for calculating pressures, temperatures, and melt compositions from mineral and mineral-melt equilibria. Thermobar allows users to perform calculations with >100 popular parameterizations involving liquid, olivine-liquid, olivine-spinel, pyroxene only, pyroxene-liquid, two pyroxene, feldspar-liquid, two feldspar, amphibole only, amphibole-liquid, amphibole-plagioclase and garnet equilibria. Thermobar is the first open-source tool which can match up all possible pairs of phases from a given region, and apply various equilibrium tests to identify pairs from which to calculate pressures and temperatures (e.g. pyroxene-liquid, two pyroxene, feldspar-liquid, two feldspar, amphibole-liquid). Thermobar also contains functions allowing users to propagate analytical errors using Monte Carlo methods, convert pressures to depths using different crustal density profiles, plot mineral classification and mineral-melt equilibrium diagrams, calculate liquid viscosities, and convert between oxygen fugacity values, buffer positions and Fe speciation in a silicate melt. Thermobar can be downloaded using pip, and extensive documentation is available at <https://bit.ly/ThermobarRTD>.

**KEYWORDS:** Open-source; Thermobarometry; Python; Clinopyroxene; Monte-carlo; Plagioclase; Hygrometry.

## 1 INTRODUCTION

Determining the pressures and temperatures of formation or equilibration of igneous phases in the Earth's crust and mantle (thermobarometry), and the melt compositions from which these phases grew (hygrometry and chemometry), is critical for understanding the behavior of magmatic systems, and for placing them in their geodynamic and tectonic contexts. Estimates of temperature have been used by a wide range of petrologic studies to investigate many important questions in igneous petrology, including the long-term temperature evolution of magmas [e.g. Bachmann and Dungan 2002; Szymanski et al. 2017; Rout et al. 2021], distinguishing between primary and recycled magmatic crystals [Walker et al. 2013], interpreting magma reservoir dynamics [e.g. Evans et al. 2016; Caricchi et al. 2020], and constraining timescales of magmatic processes [e.g. Cooper 2019; Shamloo and Till 2019; Mutch et al. 2021]. Estimating the pressures (and therefore depths) at which various magmatic processes occur is also fundamental to our understanding of igneous processes. For example, evaluating magma storage depths in arcs plays a vital role in determining the growth, chemical, and structural evolution of the Earth's crust [e.g. Rudnick 1995; Ducea et al. 2015; Lee and Anderson 2015]. Precisely constraining magma storage depths beneath active volcanic centers helps to inform risk evaluation during periods of volcanic unrest [e.g. Stock et al. 2018; Andrews et al. 2019; Pritchard et al. 2019]. Hygrometry,

which calculates the H<sub>2</sub>O content of melts, can be used to help understand the processes triggering eruptions, differences in eruptive behavior [Waters and Lange 2015; Stock et al. 2016], and to help constrain H<sub>2</sub>O-sensitive melt properties such as viscosity and temperature. Finally, chemometry, which uses the composition of mineral phases to estimate melt major element contents, is often used to provide insights into the range of magma compositions fractionating within a given volcanic system [Zhang et al. 2017].

Mineral and mineral-melt barometers, thermometers, hygrometers, and chemometers are based on the thermodynamics of reactions that occur in igneous systems. For example, equilibria with significant volume differences between products and reactants are sensitive to pressure, whereas those with entropy differences are sensitive to temperature. Specific phase equilibria are also sensitive to melt H<sub>2</sub>O content, acting as hygrometers [e.g. Waters and Lange 2015; Gavrilenko et al. 2016], and silicate melt composition (acting as chemometers). In reality, while thermodynamics is often used to determine which components are expected to correlate with pressure, temperature or water content, the coefficients attached to these components are calibrated empirically using experiments.

While a number of alternative methods exist to estimate magma storage pressures (e.g. geophysical studies, melt inclusion saturation depths), mineral-only and mineral-melt barometry remains one of the most versatile. Unlike geophysical methods, mineral barometry can be applied to volcanoes with

\*✉ penny\_wieser@berkeley.edu



Table 1: List of abbreviations

Geological Abbreviations	
<i>P</i>	Pressure
<i>T</i>	Temperature
Ol	Olivine
Liq	Liquid
Cpx	Clinopyroxene
Opx	Orthopyroxene
Fspar	Feldspar
Plag	Plagioclase Feldspar
Kspar	Potassium Feldspar
Amp	Amphibole
Sp	Spinel
Gt	Garnet
An, Ab, Or	Anorthite, Albite, and Orthoclase component
$K_D$	Distribution coefficient of Fe-Mg between Phase 1 & Phase 2
DiHd	Diopside-Hedenbergite component
EnFs	Enstatite-Ferrosilite component
CaTs	Ca-Tschermak's component
Jd	Jadeite component
Mg#	Mg/(Mg+Fe) atomic
Python Jargon	
pandas (pd.)	A Python library allowing handling of spreadsheet-like data structures
NumPy (np.)	A Python library that handles the underlying math of most calculations (e.g. log, exp)
Matplotlib (plt.)	A Python library used for plotting
String (str)	A piece of text
Float (float)	A single number that is not an integer
Integer (int)	A single number that is an integer
pandas Series	A 1D column of data
pandas DataFrame	A 2D data structure (labelled column headings, rows). Can visualize as a collection of pandas series (like a single sheet in an Excel spreadsheet)
Dictionary (dict)	Look-up tables from one value to another. In Thermobar, they are frequently used to store multiple pandas dataframes, each associated with a specific "key". These dataframes can be thought of as separate sheets in a single Excel spreadsheet (i.e. the dictionary) with the key corresponding to the sheet name

no ground-based monitoring equipment, to quiescent, dormant, extinct, and heavily eroded volcanic systems, and to ancient volcanic deposits. Additionally, unlike melt inclusion studies which rely on the collection of rapidly cooled tephra samples to minimise diffusive H<sub>2</sub>O-loss and inclusion crystallization, mineral barometers can be applied to tephra, slowly cooled lava flows, and igneous intrusions. Similarly, although

mineral-melt hygrometry provides a less direct measure of H<sub>2</sub>O contents than measurements of melt inclusions or H<sup>+</sup> measurements in minerals, it is an invaluable tool in extrusive rocks which have undergone sufficiently slow cooling that melt inclusions and minerals have likely lost their H<sup>+</sup> by diffusion [Gaetani et al. 2012]. Finally, a near absence of alternative methods to determine temperatures of magmatic storage means that mineral-melt thermometry is a very widely used technique.

The wide utility of barometry, thermometry, and hygrometry is reflected in the hundreds of different expressions relating the composition of igneous phases to intensive parameters such as *T*, *P*, H<sub>2</sub>O, and melt composition. There have also been a number of papers assessing their relative strengths and pitfalls, and updating older models when new experimental data emerges. In particular, the review of Putirka [2008] summarized the most popular thermobarometers, and provided a number of new equations calibrated on experimental data available in LEPR (library of experimental phase relations, [Hirschmann et al. 2008]). Alongside this review, K. Putirka released a series of Excel workbooks\*. These spreadsheets are widely used by the community to perform thermobarometry calculations. New thermobarometers published since this review are available as Excel spreadsheets [e.g. Masotta et al. 2013; Pu et al. 2017], Excel spreadsheets and Python scripts [e.g. Brugman and Till 2019], or Excel spreadsheets and Matlab scripts [e.g. Waters and Lange 2015]. However, a number of other models have no publicly available tool [e.g. Sugawara 2000; Mutch et al. 2016], although resources can sometimes be obtained upon request through the authors. This myriad of different tools, with different input and output structures, means that performing calculations on a variety of different mineral species within a given volcanic system is very time consuming, and requires users to repeatedly reformat their chemical data. The fact that results from different equations can't be easily calculated within a single tool has hindered detailed comparisons between available thermobarometers for a given phase. There is also often little independent quality control or benchmarking, so numerous supplementary spreadsheets contain errors (and there is no version control showing when errors are fixed).

Additionally, a number of methods have been developed in recent years which are very difficult to perform in a spreadsheet. For example, it is common that only a narrow range of melt compositions will be erupted in any given phase of a volcanic system, while the erupted crystal cargo may be very chemically diverse, having grown from a range of melt compositions undergoing chemical differentiation at depth. Thus, it is very challenging to identify an equilibrium melt composition for a given erupted mineral assemblage in order to perform meaningful thermobarometric calculations.

One solution to this problem was developed by Winpenny and MacLennan [2011], who considered all possible pairings of erupted Cpx compositions from a single flow (Borgarfjall, Iceland) with a compilation of 1000 whole-rock and glass analyses from other Icelandic eruptions. They only perform thermobarometry on Cpx-Liq pairs in equilibrium based on Fe-

\*Currently available at: <https://bit.ly/PutirkaSpreadsheets>.



Mg and trace element partitioning laws. This method was adapted by Neave and Putirka [2017], who used filters assessing the degree of equilibrium in terms of the Enstatite-Ferrosilite (EnFs), Calcium-Tschermak (CaTs), and Diopside-Hedenbergite (DiHd) components, as well as Fe-Mg equilibrium (but did not use trace elements). These "melt matching" methods are powerful but are unsuited to spreadsheet calculations; evaluating all possible pairs for 1000 liquids and 200 Cpx would require a spreadsheet with 200,000 rows. In addition, many of these calculations must be performed iteratively, as the equilibrium test values depend on  $P$  and  $T$ . For example, assessing Fe-Mg equilibrium requires knowledge of the temperature, which in turn requires knowledge of the pressure. This makes these calculations very computationally expensive. Although different scripting-based solutions have been developed for calculations of this type, none are publicly available at the time of writing, or particularly computationally efficient (taking tens of minutes to assess several hundred Cpx-Liq pairs).

Finally, most existing tools have no efficient way to propagate uncertainties in input parameters (e.g. using Monte Carlo methods) without having to manually duplicate thousands of inputs. This has meant that there has been very limited assessment of the random errors associated with thermobarometric studies.

## 2 Thermobar: AN OPEN-SOURCE SOLUTION

To address the shortage of user-friendly tools for performing popular and advanced calculations, we present a new software tool: Thermobar, written in the open-source language Python3 (which is growing in popularity within the Earth Sciences; [Petrelli 2021]). Thermobar focuses on thermobarometry, hygrometry and chemometry applicable to the crystallization of igneous phases from silicate melts within the crust and upper mantle, including >100 expressions relating to equilibrium for liquid, olivine-liquid, olivine-spinel, pyroxene, pyroxene-liquid, amphibole, amphibole-liquid, amphibole-plagioclase, garnet, feldspar, and feldspar-liquid equilibrium (Table 1, Figure 1). The full list of thermometers, barometers, and hygrometers available in Thermobar, along with the relevant functions and names used to select these equations are summarized in Figures 1, 3 and A1–A8 at the end of this manuscript.

We do not consider parameterizations calculating the conditions at which primitive liquids last equilibrated with their mantle sources [see Till 2017]. Based on the complexities associated with the local installation of thermodynamic software tools, we also don't provide calculation tools for geothermobarometers developed using rhyoliteMELTS as a framework [e.g. Gualda and Ghiorso 2014; Harmon et al. 2018], thermodynamic models of Fe-Ti oxides relying on ThermoEngine, [Ghiorso and Prissel 2020], or thermobarometers used heavily within the field of metamorphic petrology (e.g. Thermocalc, Perple-X, THERIAK-DOMINO [Powell et al. 1998; Connolly and Petrini 2002; de Capitani and Petrakakis 2010]).

For maximum versatility, Thermobar allows users to easily swap between different barometry, thermometry and hygrometry equations, and to iterate towards a solution when the system is under-constrained (e.g. solving for pressure and

temperature, or H<sub>2</sub>O contents and temperature). Additionally, we provide a number of functions for assessing equilibrium, mineral-liquid and mineral-mineral matching, and Monte Carlo error propagation. Thermobar has been extensively benchmarked to demonstrate that it gives the same results as existing tools\*.

## 3 Thermobar STRUCTURE

### 3.1 Installation

Thermobar can be installed locally on Python versions  $\geq 3.7$  using the command from either the command prompt (Windows) or the terminal (Mac):

```
pip install Thermobar
```

For Python beginners, we recommend using Jupyter environments (e.g. Jupyter Lab and Jupyter Notebook), in which case, Thermobar can be installed in a similar way within a code cell with an additional "!":

```
!pip install Thermobar
```

After installation, the user must load Thermobar into their script (here we load Thermobar as `pt`, but users could choose any letters they wish):

```
import Thermobar as pt
```

Any function from Thermobar is then called by typing the chosen abbreviation, followed by a dot, followed by the function name. For example, to use the function to calculate liquid-only temperatures:

```
pt.calculate_liq_only_temp(args)
```

Input variables for the function are entered inside the brackets (termed "arguments", or `args`).

Documentation for each function, including information on the required arguments, can be accessed using the help feature:

```
help(pt.calculate_liq_only_temp)
```

### 3.2 Python terminology

Thermobar makes extensive use of NumPy [Harris et al. 2020] and pandas [The pandas development team 2020]. For the plots shown in this paper, the plotting library matplotlib is used [Hunter 2007]. We recommend importing all these packages along with Thermobar at the start of the script (see Figure 2):

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Five main types of data are used in Thermobar (Table 1):

\*see <https://bit.ly/ThermobarBenchmarking>





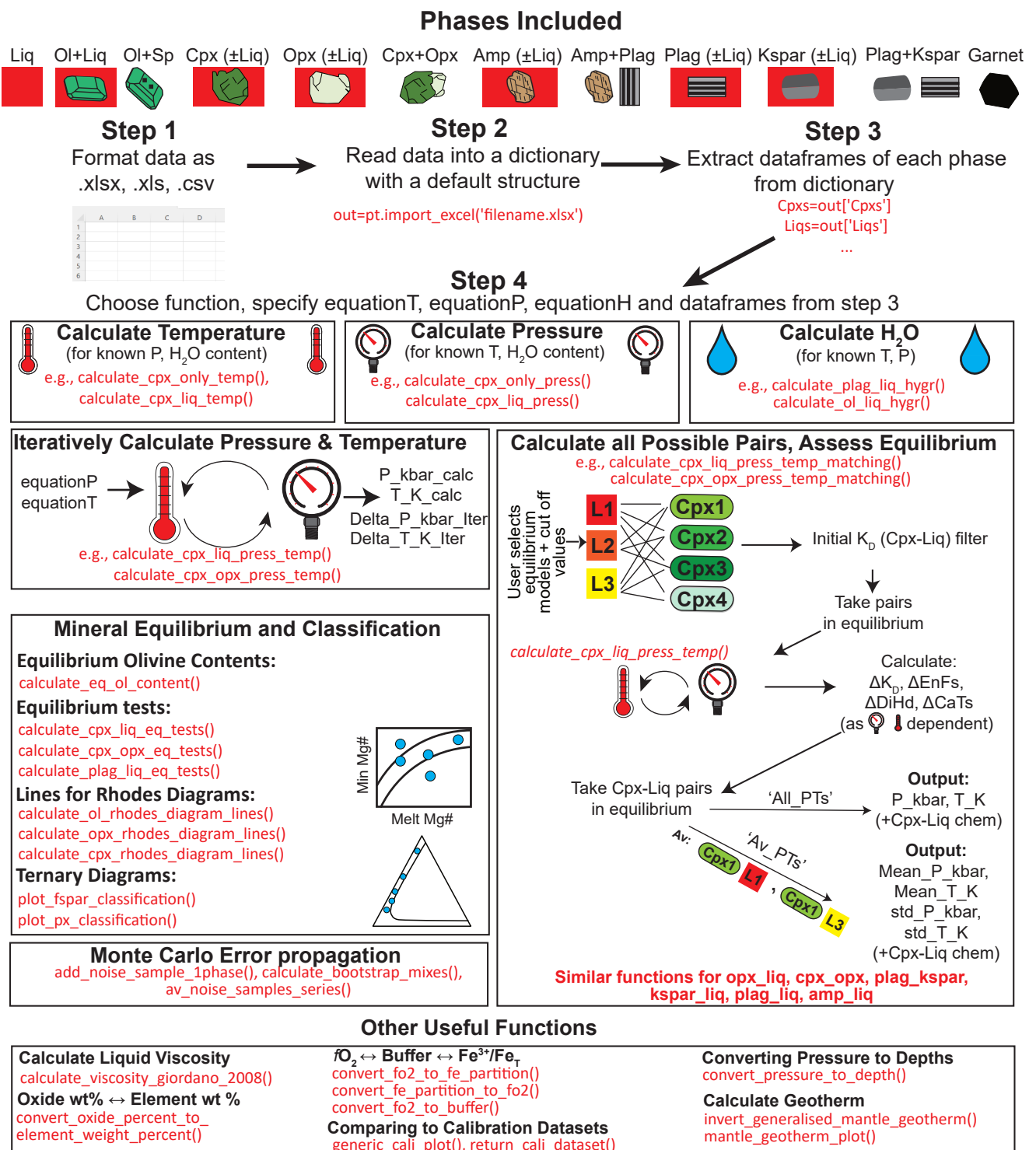


Figure 1: Schematic showing some of the functions available in Thermobar. Thermobar reads in data supplied from a spreadsheet format. The `import_excel` function returns data as separate dataframes for each phase, combined into a single dictionary. Once extracted from this dictionary, these dataframes can be fed into a number of different functions. In addition to simple calculations of  $T$ ,  $P$ , and  $H_2O$  content, Thermobar allows users to iterate different equations for pressure and temperature, assess all possible matches for pairs of phases, and perform a number of other calculations used by petrologists (e.g. calculating liquid viscosity).



1. "strings" are pieces of text (e.g. choosing which equation to use in a function - `equationP="P_Put2008_eq30"`).
2. Floats and integers are numbers, such as specifying `P=5` (integer) or `P=5.5` (float) to perform calculations at 5 kbar and 5.5 kbar respectively.
3. `pandas.Series` can be thought of as a single column of data (like a single column in an Excel spreadsheet).
4. `pandas.DataFrames` are like a single sheet in Excel, comprising columns with clear column headings (and are a collection of `pandas.Series`).
5. Dictionaries are look-up tables from one value to another. In Thermobar, they are frequently used to store multiple `pandas` dataframes, each associated with a specific "key". These dataframes can be thought of as separate sheets in a single Excel spreadsheet (i.e. the dictionary), with the key corresponding to the sheet name.

### 3.3 Data input

Users should format their compositional data as an Excel spreadsheet (.xlsx, .xls) or a comma separated values (.csv) file, with each analysis having its own row, and oxide components in wt.% oxide as column headings (Figure 2). The order of columns does not matter, as columns are identified based on their column heading, rather than position. This spreadsheet can be imported into Thermobar using the `import_excel` function, which recognises different phases based on the presence of an underscore followed by a phase identifier in column headings. For example, the column heading "SiO2\_Liq" tells Thermobar that this is the column containing the SiO<sub>2</sub> content of the liquid/melt phase.

To link a specific bit of text (e.g. lab name of EPMA spot, name of crystal, name of tephra sample, etc.) to each row of oxide contents (i.e. each analysis), this text should be stored in a column with the generic heading "Sample\_ID\_Phase", so for Cpx this column would be "Sample\_ID\_Cpx", and for Opx "Sample\_ID\_Opx". This column can be used to store any text information the user wants, and will be returned from calculations along with other calculated parameters from Thermobar functions. The full list of phase identifiers to use in headings is given below:

- Liquid (\_Liq)
- Olivine (\_Ol)
- Clinopyroxene (\_Cpx)
- Orthopyroxene (\_Opx)
- Plagioclase (\_Plag)
- Alkali feldspar (\_Kspar)
- Spinel (\_Sp)
- Amphibole (\_Amp)
- Garnet (\_Gt)

If only a single phase composition is being loaded at each time (e.g. just Liq compositions), there is no need for users to add "\_Liq" to each column heading. They can simply specify this

suffix in the `import_excel` function itself, which appends the suffix onto every column name:

```
pt.import_excel('FileName.xlsx',
sheet_name='Sheet1', suffix="_Liq")
```

Thermobar also has a function `import_excel_err` which recognises columns of the form "SiO2\_Cpx\_Err". These errors can be absolute values, where `SiO2_Cpx_Err=0.5` would represent an error of  $\pm 0.5$  wt.%. Alternatively, they can be percentage errors, where `SiO2_Cpx_Err=5` represents a  $\pm 5$  % error (e.g. for 60 wt.% SiO<sub>2</sub>, this would be equivalent to an absolute error of  $\pm 3$  wt.%). Users specify which error type is loaded, and which error distribution they wish to use in the function `pt.add_noise_sample_1phase` (e.g. specifying "Abs" or "Perc" for absolute or percent errors, and "uniform" vs. "normal" for the error distribution). Generating synthetic analyses following different error distributions is described in more detail in Section 9.

Both import functions read from the selected Excel spreadsheet, and arrange the columns into a dataframe for each mineral phase. To address the fact that many literature datasets have text values (strings) in certain cells (e.g. bdl, n.d, NA, N/A), Thermobar automatically replaces any string in any oxide column with a zero. If a given column heading Thermobar is expecting is absent, this column is filled with zeros.

The dataframes for all Thermobar-supported phases are collated into a pandas dictionary (named "out" in Figure 2). The dataframes for each phase are accessed from this output using `dictionary_name['Phase_name']` (see Step 2, Figure 2), where phase names are the same as the column identifiers used in the input spreadsheet, with the addition of an "s". For example, `out['Cpxs']` returns the dataframe of Cpx in Figure 2. For simplicity, and to create a uniform output structure, if the input spreadsheet only contains columns with the headings "\_Liq", a dictionary will still be returned containing dataframes for all other phases, but these dataframes will be filled with zeros. We recommend that dataframes are inspected before proceeding using the `.head()` function, which displays the first 5 rows. Column heading for oxides that were not present (or recognized) will be filled with zeros. If users believe they specified a column heading, but it does not appear in this dataframe, they should check for unusual characters in oxide names, decimal points other than full stops (.), and/or spaces before the column name in their spreadsheet. Inspecting outputs at this stage allows these issues to be identified before spurious calculations are performed.

In addition to "recognized" oxide column headings with specified phase identifiers, users may include other column names they wish. For example, for thermometry calculations, pressure derived from other sources, or metadata like latitude, depth within a unit, may be useful. In Figure 2, pressure is entered in a column labelled "P\_kbar\_MIs", which records the average pressure calculated from melt inclusions from the same sample. The exact name does not matter; a dataframe is present in the output dictionary named "my\_input", which contains all columns from the original spreadsheet, and these additional columns can be accessed at any time using `my_input['Column_name']`.



## Step 1 – Format data as .xlsx, .csv, .xls

Column order doesn't matter

Extra columns, e.g., a P estimate from melt inclusions, and a latitude that might be used for plotting

	A	B	C	D	E	F	L	M	N	O	P	Q	R	S	T
1	Sample_ID_Liq	SiO2_Liq	TiO2_Liq	Al2O3_Liq	FeOt_Liq	Fe3Fet_Liq	...	P2O5_Liq	H2O_Liq	P_kbar_Mls	Latitude	T_input	SiO2_Plug	TiO2_Plug	Al2O3_Plug
2	K33	49.1	3.22	14.4	14.8	0.15	...	bdl	0	3	34.5	1350	57.3	0.09	26.6
3	K34	49.2	3.89	15.3	13.7	0.15	...	bdl	0	3.5	34	1333	56.5	0.12	26.9
4	K44	49.6	3.79	15.8	13	0.15	...	0.02	0	4	35	1440	57.6	0.11	26.3

Phase identifier (tells Thermobar this is a liquid)

Used for calculations of  $K_D$ , enter Fe as FeO<sub>t</sub>, can specify a Fe<sup>3+</sup>/Fe<sup>2+</sup> ratio

Second phase (e.g. touching glass-plag analyses)

## Step 2 – Install Thermobar, import packages

```
!pip install Thermobar
```

← This installs Thermobar. It only needs to be run once on each computer. Put a # in front once you have run it

```
import Thermobar as pt
```

← This imports Thermobar after it is installed

```
import numpy as np
```

← This imports NumPy, used for various math operations

```
import pandas as pd
```

← This imports pandas, used for data storage in spreadsheet-like formats

```
import matplotlib.pyplot as plt
```

← This imports Matplotlib which is used for making figures

## Step 3 – Import data, separate out different phases

```
out=pt.import_excel('Example_Excel_input.xlsx', sheet_name="Sheet1")
```

← Specify file and sheet. Returns a dictionary "out"

```
my_input=out['my_input']
```

← Extracts dataframe from dictionary with all columns from the spreadsheet

```
myLiquids=out['Liqs']
```

← Extracts dataframe from dictionary with liquid compositions (from column headings with \_Liq)

```
myPlags=out['Plags']
```

← Extracts dataframe from dictionary with plag compositions (from column headings with \_Plag)

```
myOls=out['_Ols']
```

← As no columns with \_Ol were entered, this dataframe will be full of zeros.

## Step 4 – Visually inspect the data to ensure it imported correctly

myLiquids.head() ← Returns the first 5 rows of a dataframe

	SiO2_Liq	TiO2_Liq	Al2O3_Liq	FeOt_Liq	MnO_Liq	MgO_Liq	CaO_Liq	Na2O_Liq	K2O_Liq	Cr2O3_Liq	P2O5_Liq	H2O_Liq	Fe3Fet_Liq	NiO_Liq	CoO_Liq	CO2_Liq	Sample_ID_Liq
0	49.1	3.22	14.4	14.8	3.20	3.20	6.72	3.34	1.70	0.0	0.00	0.0	0.0	0.0	0.0	0.0	K33
1	49.2	3.89	15.3	13.7	3.88	3.88	6.76	3.44	1.22	0.0	0.00	0.0	0.0	0.0	0.0	0.0	K34
2	49.6	3.79	15.8	13.0	4.26	4.26	6.59	3.65	1.04	0.0	0.02	0.0	0.0	0.0	0.0	0.0	K44
3	49.6	3.79	15.8	13.0	4.26	4.26	6.59	3.65	1.04	0.0	0.02	0.0	0.0	0.0	0.0	0.0	K46
4	49.7	3.69	15.9	13.1	4.36	4.36	6.49	3.75	1.14	0.0	0.00	0.1	0.0	0.0	0.0	0.0	K49

Use Scroll bar to see all columns

Columns not in the spreadsheet get filled with zeros

myPlags.head()

	SiO2_Plug	TiO2_Plug	Al2O3_Plug	FeOt_Plug	MnO_Plug	MgO_Plug	CaO_Plug	Na2O_Plug	K2O_Plug	Cr2O3_Plug	Sample_ID_Plug
0	57.3	0.09	26.6	0.43	0.0	0.03	8.33	6.11	0.49	0.0	K33_plg1_spot3
1	56.5	0.12	26.9	0.47	0.0	0.05	8.95	5.66	0.47	0.0	K34_plg2
2	57.6	0.11	26.3	0.50	0.0	0.07	8.50	6.27	0.40	0.0	K44_plg1
3	57.6	0.11	26.3	0.50	0.0	0.07	8.50	6.27	0.40	0.0	K46_plg2
4	57.7	0.21	26.2	0.60	0.1	0.00	8.60	6.37	0.30	0.1	K49_plg1

Whatever text the user enters in the Sample\_ID\_{Phase} column is returned. Here we use the EPMA code for each Plag analysis, and the sample name for XRF whole-rock.

Figure 2: Guide to data input. **Step 1:** Format data into a spreadsheet with oxide names followed by \_phase. The order of columns does not matter, and other columns can also be included in the input (e.g. estimates of pressure and temperature, additional metadata, spatial data etc.). **Step 2:** Thermobar is imported, along with NumPy, pandas and matplotlib. **Step 3:** The `import_excel` function extracts data from this spreadsheet into a set of dataframes with a specific column order. The function returns a dictionary (named "out") where all these dataframes are stored with keys corresponding to different phases. For example, the dataframe of liquids is extracted from this dictionary using the key "Liqs". All dictionary keys correspond to the phase identifiers used for inputs with an added "s". If the input does not have specific column headings (e.g. no \_Ol, \_Kspar), the dataframe for this phase will be filled with zeros. **Step 4.** Dataframes for each phase are inspected to check that the spreadsheet has been read in correctly.



### 3.4 Data outputs

Thermobar returns three main types of outputs. For simple calculations, such as calculating temperature for a given melt composition and pressure, it returns a **pandas** series (a single column of data). For more complicated calculations with more than one output (e.g. pressure and temperature for iterative calculations, or when a user specifies they want equilibrium parameters to be evaluated), it returns a **pandas** dataframe (df). Any single column of a dataframe can be accessed by specifying the column name in square brackets after the name of the dataframe: `df['column_name']`. For calculations where multiple dataframes are returned (e.g. for melt matching, one dataframe is returned for all mineral-melt matches, and another with the average for each mineral measurement), these dataframes are stitched together into a dictionary. Each dataframe can be retrieved using `dict['df_name']`.

At any point, the outputs of Thermobar can be written to an Excel spreadsheet using the `to_excel` function of **pandas**. An example of this is provided in the Liquid-only thermometry section below.

### 3.5 Units

Thermobar performs all calculations using temperature in Kelvin, pressure in kbar, and chemistry in wt.% for inputs, and the same units for outputs. The only exception is that for garnet, users can either have a column `Ni_Gt` in ppm or `NiO_Gt` in wt.%.

### 3.6 Fe redox

For liquids, Thermobar allows users to specify how they partition Fe between ferrous and ferric iron, because equilibrium tests involving the partitioning of  $\text{Fe}^{2+}$  and Mg between minerals and melt are sensitive to the proportion of  $\text{Fe}^{3+}$ . To avoid ambiguity, such as in cases where XRF data is reported as  $\text{Fe}_2\text{O}_3$ , but the speciation is unknown compared to situations when the proportions of FeO and  $\text{Fe}_2\text{O}_3$  are known, total FeO contents should be used in input spreadsheets for all phases (labelled "FeOt\_Liq", "FeOt\_Cpx", etc.). To partition melt Fe between redox states, the input spreadsheet may contain a column labelled "Fe3Fet\_Liq" specifying the decimal fraction of  $\text{Fe}^{3+}$  vs.  $\text{Fe}_T$  in the liquid (e.g. `Fe3Fet_Liq=0.2` specifies 20%  $\text{Fe}^{3+}$ , 80 %  $\text{Fe}^{2+}$ ). None of the models considered here require the user to enter Fe redox proportions in phases other than liquid.

By default, functions involving liquid compositions use the value of `Fe3Fet_Liq` in the input spreadsheet, which is 0 if no column heading with this name is provided. `Fe3Fet_Liq` can also be overwritten in each function itself by specifying a fixed value (or referencing a different column in the input spreadsheet, e.g. `Fe3Fet_Liq=0.4`, or `Fe3Fet_Liq=df['column_name']`).

Alternatively, the function `convert_fo2_to_fe_partition` calculates the  $\text{Fe}^{3+}/\text{Fe}_T$  ratio and partitions iron between FeO and  $\text{Fe}_2\text{O}_3$  for a specified oxygen fugacity, as well as a liquid composition, pressure and temperature. Oxygen fugacity can be input as a  $f\text{O}_2$  value, or a buffer position in terms of  $\Delta\text{QFM}$  or  $\Delta\text{NNO}$ . This function allows users to calculate a  $\text{Fe}^{3+}/\text{Fe}_T$

ratio for each row in their input data, which can be then fed into a thermobarometry function, rather than having to use a fixed  $\text{Fe}^{3+}/\text{Fe}_T$  ratio.

### 3.7 Warnings

Thermobar contains a number of warnings which should help to direct users when they are using a model outside its calibration range. These are far from exhaustive, because they rely on the original authors specifying calibration limits beyond which their model should be used with care. For example, if users enter any liquid compositions with  $\text{SiO}_2 > 68$  wt.%, and select the Cpx-Liq barometer of Neave and Putirka [2017], the code will return the warning:

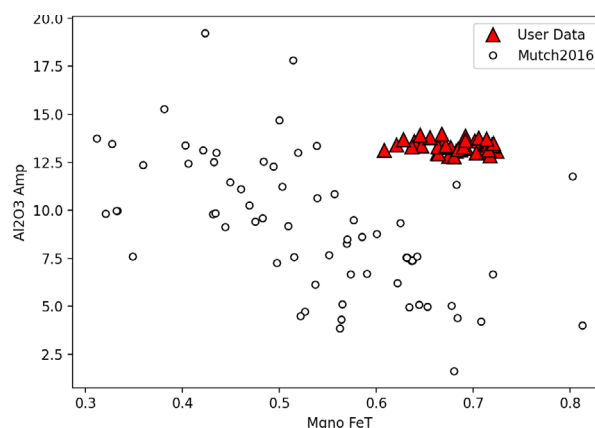
Some inputted liquids have  $\text{SiO}_2 > 68$  wt %, which exceeds the upper calibration range of the Neave and Putirka (2017) model".

### 3.8 Calibration ranges

In addition to pre-programmed warnings, the function `generic_cali_plot` can be used to examine users phase compositions alongside the calibration dataset of different thermobarometry models in  $P - T - X$  space (for models where the dataset was published or obtained by the authors; e.g. Mutch et al. [2016], Putirka [2016], and Ridolfi [2021] for Amp, Putirka [2008], Masotta et al. [2013], Neave and Putirka [2017], Bruggman and Till [2019], Petrelli [2021], Wang et al. [2021], and Jorgenson et al. [2022] for Cpx, Waters and Lange [2015] and Masotta and Mollo [2019] for Plag).

For example, to generate a plot showing  $\text{Al}_2\text{O}_3$  vs.  $\text{Mg\#}$  ( $\text{Mg}/(\text{Mg}+\text{Fe})$  atomic) of the user-entered amphibole compositions stored in the dataframe "Amps1" alongside the calibration data of Mutch et al. [2016]:

```
pt.generic_cali_plot(df=Amps1,
model="Mutch2016", x='Mgno_FeT',
y='Al2O3_Amp')
```



The order of the user data vs. calibration data can be adjusted, along with symbol size, color, transparency, etc. in this custom function. Alternatively, the calibration dataset can be obtained as a **pandas** dataframe allowing users to make their own plots in **matplotlib**:

```
MutchData=pt.return_cali_dataset(model="Mutch2016")
```



### 3.9 Worked examples

In this manuscript, we show a number of examples using snippets of code. Entire workflows can be found on the Read The Docs html webpage\*, with narrated examples on the Thermobar YouTube channel†. The Jupyter Notebooks and associated Excel files for these worked examples can be downloaded directly from the Read The Docs page, or from the Thermobar Github page‡. Available functions for phases are summarized in Figure 3, and worked examples are currently available for the workflows listed below. We will add additional examples in future, and are happy to take user requests of worked examples they would like to see:

#### 3.9.1 Liquid and Olivine-Liquid Equilibria

- Calculating temperature from liquid compositions, and temperatures and H<sub>2</sub>O contents from olivine-liquid pairs.
- Considering all possible olivine-liquid pairs for calculating temperatures and/or H<sub>2</sub>O contents, and applying various equilibrium filters based on  $K_{D, \text{Fe-Mg}}$ .
- Assessing the degree of Fe-Mg equilibrium for olivine-liquid pairs, such as plotting a Rhodes diagram (Ol Fo vs. Liq Mg#) with lines for different equilibrium models. Examples exist for a single sample, and multiple samples (e.g. multiple phases from an eruption, or different eruptive episodes).
- Calculating equilibrium olivine forsterite contents from a specific melt composition using a variety of  $K_{D, \text{Fe-Mg}}$  models.

#### 3.9.2 Cpx and Cpx-Liq Equilibria

- Calculating  $P$  for known  $T$ ,  $T$  for known  $P$ , and iteratively solving  $P$  and  $T$  for Cpx-only and Cpx-Liq pairs, including assessment of various equilibrium tests. These notebooks also show how to plot Cpx-Liq pairs on a Rhodes diagram (mineral Mg# vs. Liq Mg#).
- Calculating  $P$  and  $T$  using Cpx-only and Cpx-Liq machine learning models (showing the additional installation steps required, see Section 7.1.1 for more discussion).
- Plotting Cpx compositions on a ternary classification diagram (En-Fs-Wo), with symbols colored by different parameters.
- Cpx-Liq melt matching recreating the studies of Scruggs and Putirka [2018] and Gleeson et al. [2020].

#### 3.9.3 Opx and Opx-Liquid Equilibria

- Calculating  $P$  for known  $T$ ,  $T$  for known  $P$ , iteratively solving  $P$  and  $T$  for Opx-only and Opx-Liq pairs, including assessment of  $K_{D, \text{Ox-Liq}}$  equilibrium. These notebooks also show how to plot Opx-Liq pairs on a Rhodes diagram.

- Plotting Opx compositions on a ternary diagram (En-Fs-Wo).
- Assessing all possible Opx-Liq pairs filtered by  $K_{D, \text{Fe-Mg}}$ .

#### 3.9.4 Two Pyroxene Equilibria

- Calculating  $P$  for known  $T$ ,  $P$  for known  $T$ , iteratively solving  $P$  and  $T$ , assessment of  $K_{D, \text{Cpx-Opx}}$  equilibrium.
- Assessing all possible Cpx-Opx matches filtered by  $K_{D, \text{Cpx-Opx}}$ .

#### 3.9.5 Amp and Amp-Liq Equilibria

- Calculating  $P$  for known  $T$ ,  $T$  for known  $P$ , iteratively solving  $P$  and  $T$  for Amp-only and Amp-Liquid pairs, including assessment of  $K_{D, \text{Amp-Liq}}$  equilibrium.
- Calculating melt compositions, water contents and redox states from Amp compositions using Putirka [2016] and Zhang et al. [2017].
- Assessing all possible Amp-Liq matches filtered by  $K_{D, \text{Amp-Liq}}$ .
- Plotting Amp compositions on classification diagrams following Leake et al. [1997].

#### 3.9.6 Fspar and Fspar-Liq Equilibria

- Calculating  $T$  for known  $P$  and equilibrium tests for Plag-Liq, Kspar-Liq, and Plag-Kspar equilibria, iteratively solving  $P$  and  $T$  for Plag-Liq.
- Calculating H<sub>2</sub>O using various Plag-Liq hygrometers, including iterating temperature and H<sub>2</sub>O towards a solution.
- Assessing all possible Plag-Liq, Kspar-Liq, and Plag-Kspar matches filtered by various equilibrium tests proposed by Putirka [2008].
- Plotting Plag and Kspar compositions on a ternary diagram (An-Ab-Or).

#### 3.9.7 Garnet and geotherm calculations

- Calculating  $T$ , and  $P$  for known  $T$  using garnet compositions.
- Plotting garnet geotherms and garnet compositional sections.

#### 3.9.8 Error Propagation

- Propagating analytical errors for Liq-only thermometry, Cpx-Liq, and Cpx-only barometry (Errors can be propagated for all phases, we just only show three examples).

\*<https://bit.ly/ThermobarRTD>

†<https://bit.ly/ThermobarYouTube>

‡<https://bit.ly/ThermobarExamples>





### 3.9.9 Melt Inclusion Equilibrium

- Integrating Thermobar with VESICA [Iacovino et al. 2021] to iteratively calculate saturation pressure from melt inclusions with temperatures the melt inclusion composition, or paired analyses of the melt inclusion and host crystal (e.g. Ol-Liq, Plag-Liq, Cpx-Liq, Opx-Liq, Amp-Liq thermometry).
- Assessing Fe-Mg equilibration between melt inclusions and host olivines, and host olivines and co-erupted matrix glass.

### 3.9.10 Other Functions

- Calculating equilibrium mineral compositions (Plag-Cpx-Ol) for a specific liquid line of descent (from Petrolog3 [Danyushevsky and Plechov 2011] in this example).
- Plotting mineral and glass data with the calibration dataset of different models in  $P - T - X$  space.
- Converting from oxide wt.% to element wt.% (El wt.% can be calculated with and without oxygen).
- Converting between  $\text{Fe}^{3+}/\text{Fe}_T$ ,  $f\text{O}_2$  and buffer position.
- Calculating silicate melt viscosity using the model of Giordano et al. [2008].
- Converting pressures to depths using a variety of crustal density models.
- Inverting generalised continental geotherms of Hasterok and Chapman [2011] using thermobarometric data.

## 4 MINERAL-MELT COMPONENT CALCULATIONS

The underlying functions used for a wide range of different thermobarometers, hygrometers, and chemometers calculate mole and cation proportions and fractions for each mineral (functions located within the `core.py` file). For example, the function `calculate_anhydrous_mol_proportions_liquid` calculates the anhydrous mole proportions for user-specified liquid compositions, while `calculate_hydrous_cat_fractions_liquid` calculates cation fractions on a hydrous basis. The functions `calculate_6oxygens_orthopyroxene` and `calculate_6oxygens_clinopyroxene` calculate cations on the basis of 6 oxygens for Opx and Cpx compositions, as well as returning components such as  $\text{Al}^{\text{VI}}$  and  $\text{Al}^{\text{IV}}$  in Opx, and the proportions of Enstatite (En), Ferrosillite (Fs) and Wollastonite (Wo). The function `calculate_23oxygens_amphibole` calculates cations on the basis of 23 oxygens for Amp compositions. More advanced functions such as `calculate_clinopyroxene_liquid_components` calculates mole and cation fractions for Liq and Cpx compositions, as well as various Cpx-Liq components (e.g.  $K_D^{\text{Cpx-Liq}}$ , the `lnK_Jd_DiHd_liq` component used by Equation 33 of Putirka [2008], and other terms used in different thermobarometers). These core functions can be called to investigate natural mineral and melt compositions, as part of workflows when calibrating new thermobarometers, and for other petrological calculations requiring these variables.

## 5 USEFUL PETROLOGIC PLOTS

To aid with the visualization of mineral compositions, and the degree of mineral-melt equilibrium, we also include a number of functions for plotting imported mineral data on common classification diagrams. For example, the function `calculate_ol_rhodes_diagram_lines` calculates the equilibrium lines for an olivine-liquid equilibrium Rhodes Diagram. Together with the functions `calculate_liq_mgno` and `calculate_ol_fo` this allows users to easily plot olivines from different eruptions against the co-erupted glass Mg#, with equilibrium fields of their choosing overlain (Figure 4A). These functions could also be applied to whole-rock data (also loaded with `_Liq` suffixes) to assess olivine-whole rock relationships, such as olivine accumulation.

Thermobar has functions for overlaying mineral compositions data on ternary plots, relying on the python-ternary package from Harper et al. [2015]. The function `tern_points_px` takes imported pyroxene compositions and calculates the coordinates in En-Wo-Fs space, while the function `plot_px_classification` draws the plot and fields on which to overlay these new coordinates (Figure 4B). Similarly, `tern_points_fspar` calculates ternary coordinates in An-Ab-Or space, and `plot_fspar_classification` draws the composition fields from Deer et al. [1992] on the figure (Figure 4C). Example Jupyter notebooks show how to produce these plots in detail can be found on the Read The Docs page\* under the section for each mineral. In the example used to make Figure 4, we show how to color symbols by the FeOt content in the feldspar. As these field boundaries and user data are plotted using Matplotlib, users can easily customize the appearance of the figure, and could easily change the arguments in the `plt.scatter` command to color for a different input variable.

## 6 SINGLE-PHASE THERMOBAROMETERS AND CHEMOMETERS

Thermobar contains a number of thermometers and barometers based on the composition of a single phase:

- Liq-only thermometry
- Cpx-only thermometry and barometry
- Opx-only barometry
- Amp-only thermometry, barometry, and chemometry
- Gt-only thermometry and barometry

We discuss some examples for liquid-only thermometry, but the flexibility of function inputs is the same for other single-phase thermobarometers.

### 6.1 Liquid-only thermometers

Liquid-only thermometers vary widely in complexity. For example, the thermometer of Helz and Thornber [1987] calculates the temperature of a liquid (i.e. melt) based solely on the MgO content, while Equation 15 of Putirka [2008] uses the MgO, FeO,  $\text{Na}_2\text{O}$ ,  $\text{K}_2\text{O}$ ,  $\text{H}_2\text{O}$  content and Mg# of the liquid,

\*<https://bit.ly/ThermobarRTD>





















																					
Equilibrium	Name to substitute for <i>phase(s)</i>	calculate_ <i>phase(s)</i> _temp	calculate_ <i>phase(s)</i> _press	calculate_ <i>phase(s)</i> _press_temp	calculate_ <i>phase(s)</i> _press_temp_matching	calculate_ <i>phase(s)</i> _temp_matching	calculate_ <i>phase(s)</i> _hydr	calculate_ <i>phase(s)</i> _temp_hygr	calculate_ <i>phase(s)</i> _melt_comps	Monte-Carlo simulations	Rhodes Diagrams (Mineral Mg# vs. Liq Mg#)	Equilibrium Tests									
Liquid-only	liq_only	✓	✗	✗	N/A	N/A	✗	✗	✗	✓	N/A	N/A									
Cpx-only	cpx_only	✓	✓	✓	N/A	N/A	✗	✗	✗	✓	N/A	N/A									
Amp-only	amp_only	✓	✓	✓	N/A	N/A	✓*1	✗	✓	✓	N/A	N/A									
Opx-only	opx_only	✗	✓	✗	N/A	N/A	✗	✗	✗	✓	N/A	N/A									
Garnet-only	gt_only	✓	✓	✓	N/A	N/A	✗	✗	✗	✓	N/A	N/A									
Ol & Sp	ol_liq	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗									
Amp & Plag	amp_plag	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗									
Plag & Kspar	plag_kspar	✓	✗	✗	✗	✓	✗	✗	N/A	✓	✗	Activities of An, Ab, Or									
Ol & Liq	ol_liq	✓	✗	✗	✗	✓	✓	✓*2	N/A	✓	✓	K <sub>0</sub>									
Cpx & Liq	cpx_liq	✓	✓	✓	✓	✗	✗	✗	N/A	✓	✓	K <sub>0</sub> , DiHd, EnFs, CaTs									
Opx & Liq	opx_liq	✓	✓	✓	✓	✗	✗	✗	N/A	✓	✓	K <sub>0</sub>									
Cpx & Opx	cpx_opx	✓	✓	✓	✓	✗	✗	✗	N/A	✓	✗	K <sub>0</sub>									
Amp & Liq	amp_liq	✓	✓	✓	✓	✗	✗	✗	N/A	✓	✗	K <sub>0</sub>									
Kspar & Liq	fspar_liq	✓	✗	✗	✗	✓	✗	✗	N/A	✓	✗	✗									
Plag & Liq	fspar_liq	✓	✓	✓	✗	✓	✓	✓	N/A	✓	✗	An-Ab exchange									

Figure 3: Summary table of functions for each phase. \*1: While Amp-only hygrometers exist, in Thermobar, calculations should be performed using the `calculate_amp_only_melt_comps` function instead. \*2: As the Ol-Liq hygrometer is not *T*-sensitive, there is no need to iterate. Thus, users should use the `calculate_ol_liq_hygr` function and specify `equationT` as an input.

as well as an estimate of the pressure. For liquid-only thermometers, most equations calculate the temperature of the liquid, but equations in Thermobar with names ending with "\_sat" calculate the temperature at which a liquid is saturated in a specific phase (Table A1). For example, Equation 34 of Putirka [2008] calculates the temperature at which Cpx would saturate in the liquid (termed the saturation surface).

Several liquid-only thermometers are adapted from olivine-liquid thermometers, where the  $D_{Mg}$  term that would traditionally be calculated from the partitioning of Mg between measured olivine-liquid pairs is replaced with a theoretical value of  $D_{Mg}$ , calculated from the liquid composition using the model of Beattie [1993]. These equations are indicated with `_BeattDMg` in their name, and are particularly useful because many olivine crystals are not in Fe-Mg equilibrium with their co-erupted carrier melts (see Section 7.0.2), so it is difficult to select an olivine and liquid composition in equilibrium.

Liquid-only thermometry calculations are performed using the function `calculate_liq_only_temp`. The required inputs

are a dataframe of liquid compositions, as well as specifying a string for `equationT`. For example, for a pandas dataframe of liquids named "myLiquids" as in Figure 2, temperature using the MgO thermometer of Helz and Thornber [1987] would be calculated as follows:

```
Temp_HT87=pt.calculate_liq_only_temp(
    liq_comps=myLiquids, equationT="T_Helz1987_MgO")
```

If Equation 15 of Putirka [2008] is selected, Thermobar returns an error because this equation is *P*-sensitive:

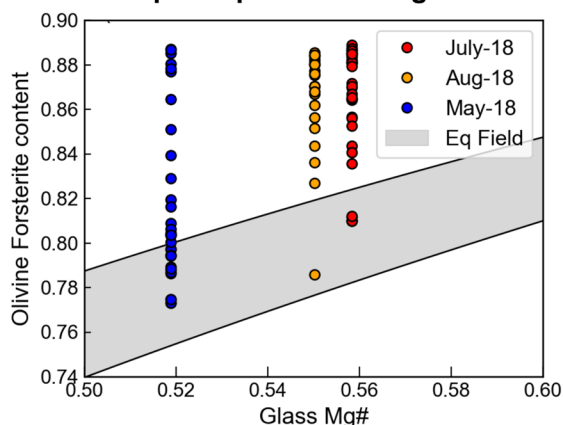
```
Temp_eq15=pt.calculate_liq_only_temp(
    liq_comps=myLiquids, equationT="T_Put2008_eq15")
```

**Exception:** You've selected a P-dependent function, please pass an option for P (see help for more detail)

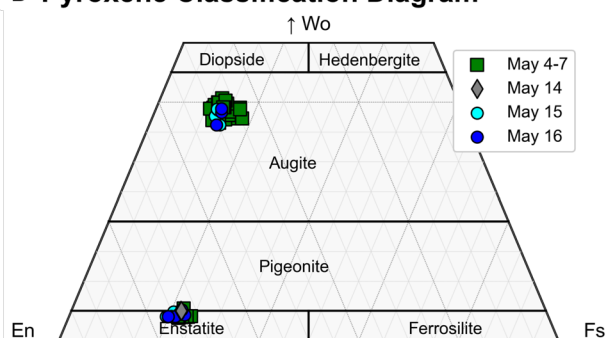
There are a number of ways to specify pressure. Firstly, a constant value of pressure can be specified for all liquids (here,  $P = 5$  kbar):



## A Olivine-Liquid equilibrium diagram



## B Pyroxene Classification Diagram



## C Feldspar Classification Diagram

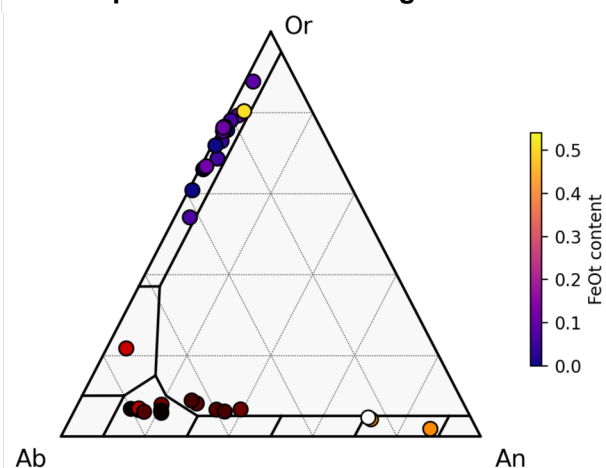


Figure 4: Example plots produced in Thermobar. [A] Olivine-Liquid Equilibrium diagram for samples erupted in May, July and Aug during the 2018 eruption of Kīlauea Volcano from Wieser et al. [2021]. The equilibrium field spans  $K_D^{\text{Ol-Liq}}$  values of 0.27–0.354 (lower bound from Roeder and Emslie [1970], upper bound from Matzen et al. [2011]). [B] Pyroxene classification for more evolved samples from the same eruptive event as in [A]. The symbols and colors representing different phases of the eruption defined by their date. [C] Co-existing Plag and Kspar compositions from the experiments of Elkins and Grove [1990]. Symbols are colored based on the  $\text{FeO}_t$  content of each feldspar.

```
Temp_eq15_Skbar=pt.calculate_liq_only_temp(
    liq_comps=myLiquids, equationT="T_Put2008_eq15",
    P=5)
```

Alternatively, if the input spreadsheet contains a column for  $P$  in kbar (labelled "P\_input") with different values for different liquids,  $P$  can be set to equal the values in this column by referencing the dataframe containing all columns (named my\_input) returned from the `import_excel` function (see Figure 2), and the column name in square brackets:

```
Temp_eq15_Pin=pt.calculate_liq_only_temp(
    liq_comps=myLiquids, equationT="T_Put2008_eq15",
    P=my_input['P_input'])
```

Some liquid-only thermometers are also sensitive to melt  $\text{H}_2\text{O}$  content (see Table A1), which is often poorly constrained in volcanic systems with no rapidly quenched tephra suitable for melt inclusion analyses. By default, Thermobar will read  $\text{H}_2\text{O}$  contents from the `H2O_Liq` column of the input spreadsheet. If the input spreadsheet has no column for  $\text{H}_2\text{O}$ , this column is filled with zeros. Input water contents can be overwritten when calling the function by specifying "`H2O_Liq=...`", allowing an easy way to investigate the effect of uncertain  $\text{H}_2\text{O}$  contents on temperatures. For example, here we evaluate temperatures at 6 wt.%  $\text{H}_2\text{O}$ :

```
Temp_eq15_6H=pt.calculate_liq_only_temp(
    liq_comps=myLiquids, equationT="T_Put2008_eq15",
    P=5, H2O_Liq=6)
```

As for pressure,  $\text{H}_2\text{O}$  can also be set to the value of any column in the input spreadsheet using `H2O_Liq=my_input['column name']`. For example, to use  $\text{H}_2\text{O}$  contents measured by Raman spectroscopy stored in a column labelled "`H2O_Raman`":

```
Temp_eq15_Hin=pt.calculate_liq_only_temp(
    liq_comps=myLiquids, equationT="T_Put2008_eq15",
    P=5, H2O_Liq=my_input['H2O_Raman'])
```

## 6.1.1 Saving to Excel

Once calculations have been performed in Thermobar, there are a number of ways to save calculations to an Excel workbook to interact with them outside of Python. To save the temperatures alongside the liquid compositions, it is easiest to first make a copy of the original dataframe using the `.copy()` function. This means that the original is still preserved in the script for further calculations and previous results are not accidentally overwritten:

```
Liq_T_out=myLiquids.copy()
```

Then, the pandas series generated by each calculation can be added onto this dataframe using the pandas `.insert()` function. Users need to specify a number for which position they want this new column in (`loc=`), the name of the column (`column=`), and the variable they wish to save in that column (`value=`).



```
Liq_T_out.insert(loc=0, column="Temp HT87",
value=Temp_HT87)
Liq_T_out.insert(loc=1, column="Temp eq15 5kbar",
value=Temp_eq15_5kbar)
Liq_T_out.insert(loc=2, column="Temp eq15 Pin",
value=Temp_eq15_Pin)
```

Here, we saved the calculations from [Helz and Thornber \[1987\]](#) to the 1st column of the dataframe (Python numbering starts from zero), and calculations from [Putirka \[2008\]](#) equation 15 at 5 kbar to the second column, and calculations using pressure from the *P* input column to the third column respectively. Finally, this new dataframe can be saved to an Excel spreadsheet (here named "Liquid\_only.xlsx"):

```
Liq_T_out.to_excel('Liquid_only.xlsx')
```

Further examples of saving various data structures to Excel can be found at [Read The Docs\\*](#).

## 6.2 Mineral-only thermometers and barometers

Mineral-only thermometers and barometers are implemented in a very similar way to liquid thermometers. For example, to calculate amphibole-only pressures using the barometer of [Mutch et al. \[2016\]](#):

```
pt.calculate_amp_only_press(
amp_comps=myAmps, equationP="P_Mutch2016")
```

Where myAmps is a dataframe of amphibole compositions from the `import_excel` function.

Similarly, to calculate Cpx-only pressure using the temperature-dependent barometer given by Equation 32b of [Putirka \[2008\]](#):

```
pt.calculate_cpx_only_press(cpx_comps=myCpxs,
equationP="P_Put2008_eq32b", T=1400)
```

Where myCpxs is a dataframe of Cpx compositions from the `import_excel` function, and 1400 is the temperature in Kelvin at which to perform calculations.

## 6.3 Iterative calculations

Unlike for experimental studies, in natural systems it is likely that neither temperature or pressure is known. To address this, Thermobar contains functions to iterate towards a solution using an equation for pressure and an equation for temperature. The names of these function are adapted from those discussed above by adding the ending "press\_temp" (e.g. `calculate_cpx_only_press_temp`).

By default, these functions start with  $T=1300$  K, which is input into the selected barometer to calculate a pressure. This calculated pressure is then entered into the selected thermometer, and this process is repeated for 30 iterations. These iterative functions also return a column labeled "Delta\_P\_kbar\_Iter" and "Delta\_T\_K\_Iter", which shows the difference in calculated pressure and temperature between the penultimate and the final iteration. If this number is not

very small (or 0), users can increase the number of iterations using `iterations=N`. Equally, the number of iterations can be reduced for computational efficiency. In numerous tests,  $N=30$  iterations converged on a solution identical to the Excel iteration used in the spreadsheets of K. Putirka.

For example, the following code calculates both pressure and temperature using only cpx compositions, and the thermometer of [Putirka \[2008\]](#) (Equation 32d for temperature and Equation 32a for pressure):

```
pt.calculate_cpx_only_press_temp(cpx_comps=myCpxs,
equationP="P_Put2008_eq32a",
equationT="T_Put2008_eq32d")
```

This returns a pandas dataframe, with columns for calculated pressure and temperature:

	P_kbar_calc	T_K_calc	Delta_P_kbar_Iter	Delta_T_K_Iter
0	7.286947	1504.081457	0.000000e+00	0.000000e+00
1	5.688497	1482.813216	0.000000e+00	0.000000e+00
2	7.376131	1513.499934	4.547474e-13	3.865352e-12
3	7.093107	1505.881234	0.000000e+00	0.000000e+00
4	8.376068	1512.137825	0.000000e+00	0.000000e+00

## 6.4 Mineral-only chemometers

At present, only Amp-only chemometers are implemented in Thermobar. To calculate co-existing equilibrium liquid compositions using [Zhang et al. \[2017\]](#) for  $\text{SiO}_2$ ,  $\text{TiO}_2$ ,  $\text{FeO}$ ,  $\text{MgO}$ ,  $\text{CaO}$ ,  $\text{K}_2\text{O}$ ,  $\text{Al}_2\text{O}_3$ , and calculated  $\text{H}_2\text{O}$  contents and  $\Delta\text{NNO}$  values from [Ridolfi \[2021\]](#) for a dataframe of amphibole compositions called myAmps:

```
pt.calculate_amp_only_melt_comps(amp_comps=myAmps)
```

	SiO2_Eq1_Zhang17	SiO2_Eq2_Zhang17	SiO2_Eq4_Zhang17	TiO2_Eq6_Zhang17	FeO_Eq7_Zhang17
0	59.333654	60.645146	56.117080	0.685874	5.586412
1	70.902522	70.515408	66.223315	0.273132	2.250664
2	51.363763	52.107144	50.932554	1.245742	11.372546
3	60.687647	62.268993	61.703644	0.690791	4.614921
4	66.494122	68.100781	65.870371	0.454822	2.915419

As well as a dataframe of results, this function also returns a warning telling users that because a temperature wasn't entered, the function has only returned the values for T-independent chemometers, and that a temperature must be entered to get additional columns for other expressions that are T-dependent:

**"You must enter a value for T in Kelvin to get results from equation3 and 5 from Zhang, and SiO2 from Putirka (2016)".**

These additional equations are evaluated when a temperature is specified within the function:

```
pt.calculate_amp_only_melt_comps(
amp_comps=myAmps, T=1300)
```

In many cases, the temperature may not be known. Thus, the user could first calculate Amp-only pressure and temperature

\* <https://bit.ly/ThermobarRTD>





by iterating the barometer of [Ridolfi \[2021\]](#) with the thermometer of [Ridolfi and Renzulli \[2012\]](#):

```
PT_Rid=pt.calculate_amp_only_press_temp(
    amp_comps=myAmps,
    equationP="P_Ridolfi2021",
    equationT="T_Ridolfi2012",
    Ridolfi_Filter=True)
```

By default, Amp compositions failing the various filters of [Ridolfi \[2021\]](#) return NaNs for  $P$  and  $T$ . The user could specify `Ridolfi_Filter=False` such that  $P$  and  $T$  are calculated for all Amps, but in this case we strongly encourage inspection of the column named "Fail Msg" in the dataframe `PT_Rid` to see which ones failed and why (e.g. low totals, Amps not Mg-hornblendes).

The calculated temperature from the output dataframe `PT_Rid` in the column named "T\_K\_Calc" can be input into the chemistry function:

```
pt.calculate_amp_only_melt_comps(
    amp_comps=myAmps, T=PT_Rid['T_K_Calc'])
```

## 7 TWO-PHASE THERMOMETERS AND BAROMETERS

The following thermometers, barometers, and hygrometers are based on equilibrium between two phases. The application of these functions generally requires more thought from the user. In an ideal scenario, calculations are performed on phases which have a clear textural relationship, such as measurements of spinels trapped within a specific olivine crystal [[Matthews et al. 2016](#)], or measurements of touching Cpx-Opx pairs [[Walker et al. 2013](#)]. However, in many natural samples, this is simply not possible. For example, disaggregation of crystals during transport and eruption mean that it is very common that erupted lavas and tephra samples have few, or no touching pairs of crystals. Even if crystals are touching, there is no guarantee that they are in chemical equilibrium, as crystals with different histories can be aggregated into clusters by flow within volcanic conduits and/or crystal settling [[Wieser et al. 2019b](#); [Culha et al. 2020](#)].

Thermobarometers which rely on the equilibrium between a liquid and crystal phase (rather than two crystal phases) are particularly problematic. Generally, only a narrow range of liquid compositions will be erupted in any given phase of an eruption, while the erupted crystal cargo may be chemically diverse, having grown from a range of melt compositions undergoing chemical differentiation at depth. In many volcanic centers, the lack of glassy groundmass means it is difficult to even characterize the composition of this single "carrier liquid" bringing the crystals to the surface, as bulk analysis techniques such as XRF are sensitive to crystal addition. These pitfalls make it very difficult to identify meaningful mineral-melt pairs in many volcanic systems.

In *Thermobar*, we provide a number of functions implementing workflows proposed in the literature for these less-than-optimal (but common) scenarios. We present functions which consider all possible matches between measured phases (e.g. assessing all possible liquid and pyroxene pairs, or all possible pairs of orthopyroxenes and clinopyroxenes), with

user-defined equilibrium filters. Where relevant, the equilibrium tests available for each thermobarometer are discussed below.

### 7.0.1 Olivine-spinel thermometry

*Thermobar* includes the olivine-spinel thermometers of [Wan et al. \[2008\]](#) and [Coogan et al. \[2014\]](#) (Table A2), which are both pressure-independent. The input spreadsheet should be prepared such that each row contains an olivine composition (column headings: "SiO2\_Ol", "MgO\_Ol") and a spinel composition ("SiO2\_Sp", "MgO\_Sp" etc.). After using the `import_excel` function, these thermometers are called using the function `calculate_ol_sp_temp`:

```
pt.calculate_ol_sp_temp(
    ol_comps=myOls, sp_comps=mySps,
    equationT="T_Wan2008")
```

Where `myOls` is a dataframe of olivine compositions, `mySps` is a dataframe of spinel compositions, and the thermometer is from [Wan et al. \[2008\]](#).

To our knowledge, the only proposed Ol-Sp equilibrium test is from [Prissel et al. \[2016\]](#), who propose that  $K_{D,Fe-Mg}^{Sp-Ol}$  can be calculated from a linear regression involving the Cr# of the spinel. However, as Ol-Sp thermometers only use the  $Al_2O_3$  content of the olivine, which is substantially more resistant to diffusive re-equilibration than Fe-Mg, the utility of this equilibrium test for determining Ol-Sp temperatures is unclear (so at the moment, we do not include any Ol-Sp equilibrium tests in *Thermobar*).

### 7.0.2 Olivine-liquid thermometry

As with olivine-spinel thermometry, the default way to calculate olivine-liquid temperatures in *Thermobar* is to prepare an Excel spreadsheet with each row containing an olivine composition paired with a specific liquid composition. For all olivine-liquid thermometers except that of [Pu et al. \[2017\]](#), a pressure needs to be specified (as in Section 7.0.2). For example, temperatures can be calculated using Equation 21 of [Putirka \[2008\]](#) at 5 kbar:

```
pt.calculate_ol_liq_temp(
    liq_comps=myLiquids, ol_comps=myOls,
    equationT="T_Put2008_eq21", P=5)
```

This function returns a pandas dataframe with the temperature in Kelvin as well as the measured  $K_D^{Ol-Liq}$ .

Unlike olivine-spinel thermometry, olivine-liquid thermometry is highly susceptible to issues involving disequilibrium. This is because olivine crystals are commonly "antecrystic", being brought to the surface in chemically unrelated melts [[Balta et al. 2013](#); [Wieser et al. 2019a](#)]. Thus, it is vital to calculate the degree of equilibrium for olivine-liquid pairs to assess the accuracy of thermometric estimates. The most common way to assess olivine-melt equilibrium examines the partition coefficient of Fe-Mg between these two phases ( $K_D^{Ol-Liq}$ , returned by default for Ol-Liq functions). The value of  $K_D^{Ol-Liq}$  is sensitive to the amount of  $Fe^{3+}$  in the melt. By default, all



Thermobar functions use the value of `Fe3Fet_Liq` in the inputted spreadsheet, and  $K_D^{\text{Ol-Liq}}$  is calculated using only  $\text{Fe}^{2+}$  in the liquid phase (and all FeO in the olivine as  $\text{Fe}^{2+}$ ). The proportion of  $\text{Fe}^{3+}$  used in the calculation can be overwritten by specifying a different value for `Fe3Fet_Liq`. Here we perform calculations using 20 %  $\text{Fe}^{3+}$ :

```
pt.calculate_ol_liq_temp(
    liq_comps=myLiquids, ol_comps=myOls,
    equationT="T_Put2008_eq21", P=5,
    Fe3Fet_Liq=0.2)
```

If `eq_tests=True` is specified in the function, equilibrium  $K_D^{\text{Ol-Liq}}$  values are calculated from the liquid composition using the models of [Roeder and Emslie \[1970\]](#), [Toplis \[2005\]](#), and [Matzen et al. \[2011\]](#):

```
pt.calculate_ol_liq_temp(
    liq_comps=myLiquids, ol_comps=myOls,
    equationT="T_Put2008_eq21", P=5,
    eq_tests=True)
```

As well as the calculated temperature, the measured  $K_D^{\text{Ol-Liq}}$ , the calculated  $K_D^{\text{Ol-Liq}}$  for each model, and the input liquid composition, the function returns the difference between measured and predicted  $K_D^{\text{Ol-Liq}}$  values ( $\Delta K_D$ ) for these three models (all as a pandas dataframe):

	T_K_calc	Kd Meas	Kd calc (Toplis)	$\Delta K_D$ , Toplis (M-P)	$\Delta K_D$ , Roeder (M-P)	$\Delta K_D$ , Matzen (M-P)
0	1289.947705	0.314264	0.325040	-0.010776	0.014264	-0.025736
1	1229.813416	0.175383	0.308684	-0.133301	-0.124617	-0.164617
2	1285.857491	0.269582	0.315891	-0.046309	-0.030418	-0.070418
3	1198.240159	0.173799	0.296719	-0.122920	-0.126201	-0.166201

In many cases, none of the pre-matched olivines and liquids will be in equilibrium. To help users determine the composition of olivines that would be in equilibrium with their liquids, the function `calculate_eq_ol_content` calculates the equilibrium olivine forsterite content for a given set of liquid compositions. As for the equilibrium test above, three models for predicting  $K_D^{\text{Ol-Liq}}$  equilibrium are included. Specifying `Kd_model="Roeder1970"` uses  $K_D^{\text{Ol-Liq}} = 0.3 \pm 0.03$  following [Roeder and Emslie \[1970\]](#), `Kd_model="Matzen2011"` uses  $K_D^{\text{Ol-Liq}} = 0.34 \pm 0.012$  following [Matzen et al. \[2011\]](#).

For example, to calculate the equilibrium olivine content using the model of [Roeder and Emslie \[1970\]](#):

```
pt.calculate_eq_ol_content(liq_comps=myLiquids,
    Kd_model="Roeder_1970")
```

The pandas dataframe returned by the function has column headings corresponding to the equilibrium forsterite content for  $K_D^{\text{Ol-Liq}} = 0.3$  (preferred value),  $0.33 (+1\sigma)$ , and  $0.27 (-1\sigma)$ :

	Mg#_Liq_Fe2	Mg#_Liq_Fet	Eq Fo (Roeder, Kd=0.3)	Eq Fo (Roeder, Kd=0.33)	Eq Fo (Roeder, Kd=0.27)
0	0.681666	0.631416	0.877117	0.866470	0.888030
1	0.620707	0.566947	0.845080	0.832188	0.858378
2	0.578196	0.523041	0.820442	0.805970	0.835443

Columns are also returned showing liquid Mg# calculated using  $\text{Fe}^{2+}$  only (used to calculate the Eq Fo contents), and also using  $\text{Fe}_T$ .

Unlike the fixed  $K_D^{\text{Ol-Liq}}$  values of [Roeder and Emslie \[1970\]](#) and [Matzen et al. 2011](#)], the model of [Toplis \[2005\]](#) calculates  $K_D^{\text{Ol-Liq}}$  as a function of liquid composition, pressure, temperature, and olivine forsterite content. Thermobar provides several ways to use this model. First, using paired olivine and liquid compositions:

```
pt.calculate_eq_ol_content(
    liq_comps=myLiquids, ol_comps=myOls,
    Kd_model="Toplis2005", P=2, T=1373.1)
```

Alternatively, just the olivine forsterite content can be input as a single value or a pandas series (instead of the full olivine compositions), along with pressure, temperature, and liquid compositions:

```
pt.calculate_eq_ol_content(
    liq_comps=myLiquids, ol_fo=0.82,
    Kd_model="Toplis2005", P=2, T=1373.1)
```

In both cases, the function returns a pandas dataframe where the first column is the equilibrium  $K_D^{\text{Ol-Liq}}$  calculated using [Toplis \[2005\]](#), and the second column is the equilibrium olivine forsterite content. However, needing to specify an olivine forsterite content to calculate an equilibrium forsterite content is somewhat circular logic. If olivine compositions or a forsterite content are not entered into the function, Thermobar will iterate by first calculating a  $K_D^{\text{Ol-Liq}}$  for  $\text{Fo} = 0.95$ , then use this  $K_D^{\text{Ol-Liq}}$  to calculate an equilibrium Fo content, and then inputting that Fo content into a new calculation for  $K_D^{\text{Ol-Liq}}$  (over 20 iterations):

```
pt.calculate_eq_ol_content(
    liq_comps=myLiquids, Kd_model="Toplis2005",
    P=2, T=1373.1)
```

If `Kd_model="All"`, calculations are performed using all three models (including using the iterative approach for [Toplis \[2005\]](#)):

```
pt.calculate_eq_ol_content(
    liq_comps=myLiquids, ol_comps=myOls,
    Kd_model="All", P=2, T=1373.1)
```

### 7.0.3 Olivine-Liquid melt matching

The function `calculate_ol_liq_temp_matching` considers all possible matches between the input dataframe of Olivine and Liquid compositions (e.g. N = 10 Ol and N = 20 Liq would yield 200 rows). The function returns all possible matches with calculated temperatures. If users specify `eq_tests=True`, rows can be filtered based on the various  $K_D$  filters described above.

## 7.1 Clinopyroxene-liquid thermobarometry

Thermobar contains a number of different thermobarometers applicable to Cpx-Liq pairs ([Table A4](#)). In the simplest scenario where relevant Cpx-Liq pairs have been identified (e.g.



experimental products, groundmass-rim pairs), data should be prepared as an Excel spreadsheet where each row contains a matched pair of Liq and Cpx compositions. The function `calculate_cpx_liq_press` allows users to calculate pressures for a variety of barometers, while the function `calculate_cpx_liq_temp` calculates temperatures. For thermometers which are  $P$ -sensitive, a pressure in kbar must be specified, and a temperature in K must be specified for  $T$ -sensitive barometers (as for the single-phase thermobarometers discussed above). For example, to calculate the temperature using Equation 33 of Putirka [2008] at 5 kbar:

```
pt.calculate_cpx_liq_temp(
    liq_comps=myLiquids, cpx_comps=myCpxs,
    equationT="T_Putirka2008_eq33", P=5)
```

When neither pressure nor temperature is known, the function `calculate_cpx_liq_press_temp` iterates towards a solution using a user-supplied pressure and temperature by specifying an equation for both pressure and temperature (see Section 6.3). For example, here we iterate Equation 33 for  $T$  and equation 30 for  $P$  from Putirka [2008]:

```
pt.calculate_cpx_liq_press(
    liq_comps=myLiquids, cpx_comps=myCpxs,
    equationT="T_Putirka2008_eq33",
    equationP="P_Putirka2008_eq30").
```

To return the values of different equilibrium tests (e.g. DiHd, EnFs, [Neave et al. 2019]), users can specify an additional argument `eq_tests=True` in all Cpx-Liq functions.

### 7.1.1 Machine Learning models

Thermobar also contains implementations of the machine learning (ML) Cpx-only and Cpx-Liq thermometers and barometers of Petrelli et al. [2020] and Jorgenson et al. [2022], which use the extra trees algorithm [Geurts et al. 2006]. Thermobar is distributed using the free service PyPI, so that users can install it using the simple `pip install` command. However, PyPI has a size limit of 100 MB per "release" of the project. Given that pickle (.pkl) or onnx (.onnx) files used to save pre-trained ML models tend to be 10s of MB each, it is not possible to distribute all these preserved models as well as the other Thermobar source code in a single package.

Thus, in addition to `pip install` Thermobar once on their machine, users who wish to use machine learning models will need to run an additional line in their notebook specifying that they wish to download these saved models from the Github repository `Thermobar_onnx`:

```
!pip install "https://github.com/PennyWieser/Thermobar_onnx/
archive/refs/tags/0.02.zip"
```

Once these files have been downloaded, they can be accessed the same way as more conventional empirical thermobarometers:

```
pt.calculate_cpx_liq_press(
    liq_comps=myLiquids, cpx_comps=myCpxs,
    equationP="P_Petrelli2020_Cpx_Liq").
```

Following Jorgenson et al. [2022], Thermobar also returns the median, standard deviation, and interquartile range calculated from all the trees used (as well as calculated pressures or temperatures):

	P_kbar_calc	Median_Trees	Std_Trees	IQR_Trees
0	6.781813	7.000	4.375027	7.300
1	10.679156	9.300	3.759244	5.000
2	8.477356	9.300	4.105941	4.985

This allows users to filter out rows which give very large interquartile ranges or standard deviations.

An ongoing problem with ML-based thermobarometers is that even using the same code, different versions of `scikit-learn` will return different pressures and temperatures (with differences up to ~0.5 kbar). Additionally, ML models saved as pickles in one version of `scikit-learn` will yield a warning message when opened in a different version:

UserWarning: Trying to unpickle estimator StandardScaler from version 1.0.2 when using version 0.24.1. This might lead to breaking code or invalid results. Use at your own risk.

UserWarning: Trying to unpickle estimator ExtraTreeRegressor from version 1.0.2 when using version 0.24.1. This might lead to breaking code or invalid results. Use at your own risk.

These warnings are not concerning in themselves because the answer obtained from one version is not more correct than that from any other version, and differences are well within the stated SEE of the model. However, different results based on the specifics of the local Python installation does represent a problem in terms of ensuring results are reproducible.

One solution is to use ONNX [Open Neural Network Exchange; ONNX-Runtime-developers 2021] to save ML pipelines, which ensures stable results. However, as of yet, there is no way to build the voting of Jorgenson et al. [2022] into these pipelines. Thus, in Thermobar, we include two versions of ML models:

1. `equationP="P_Petrelli2020_Cpx_only"` will calculate pressures using voting for Petrelli et al. [2020] from a model saved as a pickle, but the exact pressure will change with future versions.

2. `equationP="P_Petrelli2020_Cpx_only_onnx"` will use ONNX to give a stable answer, but cannot currently do voting.

The same suffix format applies to the models from Jorgenson et al. [2022] to access these two options.

With time, we anticipate the pickles will eventually stop loading into newer versions of `scikit-learn`. We will re-release new .pkl files (and .onnx files if required) when this happens, so users should check for the latest version number from <https://bit.ly/ThermobarMLTags>, and upgrade their installation:

```
!pip install --upgrade "Paste Latest URL Here"
```

We will also update this repository to add new ML models as they emerge.



### 7.1.2 Cpx-Liq Melt Matching

A number of methods have been developed to perform Cpx-Liq thermometry by comparing all erupted Cpx and Liq compositions from a given volcanic center/region, and identifying liquid-cpx pairs which meet certain equilibrium criteria [e.g. [Winpenny and MacLennan 2011](#); [Neave and Putirka 2017](#); [Scruggs and Putirka 2018](#); [Neave et al. 2019](#)]. In Thermobar, the function `calculate_cpx_liq_press_temp_matching` assesses all possible clinopyroxene-liquid pairs for a user-supplied dataframe of liquid compositions of length M (e.g. all XRF analyses from a given volcanic center), and a user-supplied dataframe of measured Cpx compositions of length N. The function performs the following steps:

1. Liq components and Cpx components (e.g. cation fractions) are calculated for each individual sample (saving computational time vs. calculating them after the duplication steps below).

2. Each Cpx composition (oxides+components) is duplicated M times forming a pandas dataframe with rows for Cpx1-Cpx1-Cpx1, ..., CpxN-CpxN-CpxN. The dataframe of liquid compositions (oxides+components) is duplicated N times forming a dataframe with rows for Liq1-Liq2-Liq3...LiqM, Liq-Liq2-Liq3 ... LiqM. These dataframes are combined, creating a dataframe of length N×M with all possible Cpx-Liq pairings of the format Cpx1-Liq1, Cpx1-Liq2, Cpx1-Liq3, Cpx2-Liq1 ... CpxN-LiqM.

3. Compositional components which require both a Liq and Cpx composition are calculated for this combined dataframe (e.g. the DiHd component,  $K_D^{Cpx-Liq}$ ).

Step 3 is complex. As Cpx-Liq equilibrium tests are sensitive to pressure and/or temperature, equilibrium tests cannot be performed until pressures and temperatures for each pair have been calculated. However, calculating pressures and temperatures iteratively for all possible Cpx-Liq matches can be very time consuming (e.g. 400 Cpx and 2500 possible liquids requires 1 million iterative calculations to be performed). To increase computational efficiency, we apply a preliminary filter in terms of  $K_D^{Cpx-Liq}$  equilibrium (using Equation 35 of [Putirka \[2008\]](#) by default). As  $K_D^{Cpx-Liq}$  parameterizations are sensitive to temperature but not pressure, we use the `calculate_cpx_liq_temp` function to calculate a minimum temperature for each Cpx (for a default value of  $P = -10$  kbar, adjustable using `PMin=...`), and a maximum temperature (for a default value of  $P = 30$  kbar, adjustable using `PMax=...`). This upper pressure limit was set with volcanic systems in mind, but can be easily overridden when calling the function. These maximum and minimum equilibrium  $K_D^{Cpx-Liq}$  values are compared to the measured  $K_{D,Fe-Mg}$  values for each Cpx-Liq pair. If the deviation between measured and calculated  $K_{D,Fe-Mg}$  is greater than the specified value (0.03 by default, changed by specifying `Kd_Err=" "`) for both the minimum and maximum equilibrium  $K_D^{Cpx-Liq}$ , no temperatures in-between will yield a match. Thus, these Cpx-Liq matches can be discarded.

4. The function `calculate_cpx_liq_press_temp` is used to iteratively calculate pressures and temperatures for remaining Cpx-Liq pairs.

5. Using the calculated temperature and pressure for each pair, the equilibrium  $K_D^{Cpx-Liq}$  is calculated using Equation 35 of [Putirka \[2008\]](#), the equilibrium CaTs component using the expression of [Putirka \[1999\]](#), and the updated equilibrium EnFs and DiHd components calculated using the expression of [Mollo et al. \[2013\]](#), following [Neave et al. \[2019\]](#). Other models for these equilibrium tests can also be specified in the function. It is worth noting that the supplementary spreadsheet of [Neave et al. \[2019\]](#) uses the [Putirka et al. \[1996\]](#) anhydrous thermometer to calculate the  $K_D^{Cpx-Liq}$  component, while temperature is calculated using [Putirka \[2008\]](#) (Equation 33). In our code,  $K_D^{Cpx-Liq}$  is calculated using the user-specified thermometer for consistency.

6. By default, Thermobar then selects Cpx-Liq pairs where the measured components calculated using the method of [Putirka et al. \[1996\]](#) and calculated equilibrium components are within  $\pm 0.03$  for  $K_D^{Cpx-Liq}$ ,  $\pm 0.06$  for DiHd,  $\pm 0.05$  for EnFs, and  $\pm 0.03$  for CaTs (following the supporting Excel spreadsheet of [Neave et al. \[2019\]](#)). Users can change these selection criteria using `DiHd_Err=...`, `Kd_Err=...` etc.

7. The function returns a dictionary. Users can extract a pandas dataframe of all Cpx-Liq matches which meet the specified equilibrium criteria using `dictionary_name['ALL_PTs']`. Following the approach of [Neave and Putirka \[2017\]](#), Thermobar also performs calculations to average the pressures and temperatures for each Cpx. For example, if Cpx1 matches with Liq1, Liq3, and Liq9, the values for these three matches will be averaged, and the standard deviation of the pressure and temperature are returned. This information is stored in the second part of the dictionary accessed using `dictionary_name['Av_PTs']`.

The speed at which these calculations are performed are significantly faster than previous tools (seconds vs. tens of minutes for assessing matches between hundreds of possible Cpx and Liqs). This, along with the flexibility provided by the implementation of these tools in Python, offers users more freedom to assess possible Cpx-Liq matches in larger datasets. There is also far more choice of equilibrium filters. For example, users can specify `Kd_Match="Masotta"`, which calculates  $K_D^{Cpx-Liq}$  using Equation 35Alk of [Masotta et al. \[2013\]](#). This equation expresses  $K_D^{Cpx-Liq}$  as a function of temperature, and the cation fractions of Na<sub>2</sub>O and K<sub>2</sub>O in the melt, and was developed for trachyte and phonolitic magmas (extreme care should be taken when applying it to other melt compositions). As with the other functions discussed so far, users can also specify `H2O_Liq` and `Fe3Fet_Liq` ratio in the function itself. This can be a fixed value for all calculations, or could be set as a pandas series with the same length as the input dataframe of liquid compositions.



### 7.1.3 Recreating Scruggs and Putirka (2018)

To demonstrate the versatility of this Cpx-Liq melt matching function, we recreate the analysis of [Scruggs and Putirka \[2018\]](#), who assess Cpx-Liq equilibrium on samples from Chaos Craggs at Lassen Peak. The erupted liquids sampled at Chaos Craggs are strongly bimodal. To capture the compositions of liquids which likely exist at depth between these two erupted end-members, [\[Scruggs and Putirka 2018\]](#) add or subtract the composition of a felsic-whole rock composition from measured mafic liquids, and use the solver functions in Excel to find the mixing proportion that best satisfies equilibrium tests.

We demonstrate how a more automated approach can be implemented in *Thermobar* in [Figure 6](#). A worked example for this is available on the Read The Docs page\*. Step 1 imports an Excel spreadsheet containing possible liquid compositions (whole-rock data in this example), and a separate sheet or spreadsheet containing measured Cpx compositions ([Figure 5](#)). Step 2 defines the silicic and mafic end-member liquid compositions. For the silicic liquid, we apply a filter to only consider measured liquids with >65 wt.% SiO<sub>2</sub>. For the mafic end-member, we filter based on measured samples with <53.8 wt.% SiO<sub>2</sub> and >4 wt.% MgO. We then use the function `add_noise_sample_1phase` to generate five synthetic liquids for each measured liquid of these two end-members. Each oxide for these synthetic liquids was chosen from a normal distribution with the mean set at the measured liquid composition, and  $1\sigma = 1\%$  (percent of measured value, not absolute wt.%). These synthetic liquids help account for the fact there are a number of liquids that exist at depth which were not represented during sampling. The following steps could also be performed using the dataframe of measured liquid compositions without applying filters or adding noise.

Step 3 mixes these end-members to generate synthetic liquids spanning the entire compositional range between measured liquids. The function `calculate_bootstrap_mixes` mixes these two end-members in various proportions with a number of different options (demonstrated in the example notebook). In its simplest form, the function takes two end-members, and mixes a randomly selected composition from one end-member with a randomly selected composition from the other end-member, with the mixing proportion varying randomly between 0 and 1. Additional flexibility is provided by the optional input `self_mixing`. If `self_mixing=True`, the two end-members are combined into a single dataframe, and these compositions are randomly mixed. This means that *Thermobar* mixes between: 1) mafic end-members with silicic end-members (as in the default form), 2) mafic end-members with other mafic end-members, and 3) silicic end-members with other silicic end-member compositions. Self-mixing produces a stronger clustering of synthetic liquids near the end-members, which may be useful in certain circumstances. However, in this specific example, relatively few liquids generated by this function lie within the compositional gap between mafic and silicic compositions for <1000 duplicates. Thus, we use the option `self_mixing="Partial"`,

which creates half the mixes by mixing between silicic and mafic end-members, and the other half from self-mixing.

Step 4 is optional ([Figure 6](#)), and combines this synthetic dataframe of liquids with the original dataframe of liquids using the `pandas concat` function (to include samples which were not selected as end-members).

Because Cpx thermometry is sensitive to the H<sub>2</sub>O content of the liquid, but H<sub>2</sub>O contents at depth cannot be deduced from bulk rock analyses of degassed lava samples, [Scruggs and Putirka \[2018\]](#) calculate the H<sub>2</sub>O of the liquid as a function of the SiO<sub>2</sub> content. Step 5 overwrites the H<sub>2</sub>O in the Liq dataframe (0 as whole-rock data) using their expression.

Step 6 inputs this finalized dataframe of synthetic and measured liquids, and measured Cpx compositions into the function `calculate_cpx_liq_press_temp_matching` ([Figure 6](#)). Step 6 uses `matplotlib` to plot averaged pressures and temperatures from each Cpx as red diamonds with  $1\sigma$  error bars (`plt.errorbar`), and all possible matches as semi-transparent symbols.

### 7.2 Orthopyroxene-liquid thermobarometry

The orthopyroxene-liquid functions in *Thermobar* are very similar to those for Cpx-Liq. If users wish to calculate pressure or temperature for Opx-Liq pairs (e.g. measured rim and matrix glass compositions), they can use the functions `calculate_opx_liq_press` and `calculate_opx_liq_temp`. Similarly,  $P$  and  $T$  can be solved iteratively using `calculate_opx_liq_press_temp`, specifying an `equationP` and `equationT`.

To assesses all possible Liq and Opx pairs, and calculate  $P$  and  $T$  for pairs within user-specified ranges for equilibrium, the function `calculate_opx_liq_press_temp_matching` should be used. Unfortunately, there is only one commonly used equilibrium test for Opx-Liq pairs, which compares measured values of  $K_D^{\text{Opx-Liq}}$  to those predicted from the Liq. [Putirka \[2008\]](#) suggests that the range of  $K_D^{\text{Opx-Liq}}$  values in experiments ranges from  $0.29 \pm 0.06$ , and can be expressed as a function of the cation fraction of Si in the liquid ( $K_D^{\text{Opx-Liq}} = 0.4805 - 0.3773 X_{\text{Si}}^{\text{liq}}$ ). Because this equilibrium test is independent of  $P$  and  $T$ , Opx-Liq pairs can be filtered without any iteration (simplifying the function relative to that for Cpx-Liq). The Opx-Liq melt matching algorithm follows steps 1–3 described in [Section 7.1](#). Then,  $K_D^{\text{Opx-Liq}}$  values are computed for each Opx-Liq pair and compared to equilibrium values. By default, the function calculates equilibrium values using the  $X_{\text{Si}}^{\text{liq}}$  expression of [Putirka \[2008\]](#), and considers all matches within  $\Delta K_D^{\text{Opx-Liq}}$  of  $\pm 0.06$ . Users can override this default option by specifying a value for `Kd_Match`, and `Kd_Err` in the function. To evaluate Opx-Liq pairs with measured  $K_D^{\text{Opx-Liq}} = 0.29 \pm 0.07$ :

```
pt.calculate_opx_liq_press_temp_matching(
    liq_comps=myLiquids, opx_comps=myOpxs,
    equationT="T_Put2008_eq28a",
    equationP="P_Put2008_eq29b",
    Kd_Match=0.29, Kd_Err=0.07)
```

\* <https://bit.ly/ThermobarRTD>





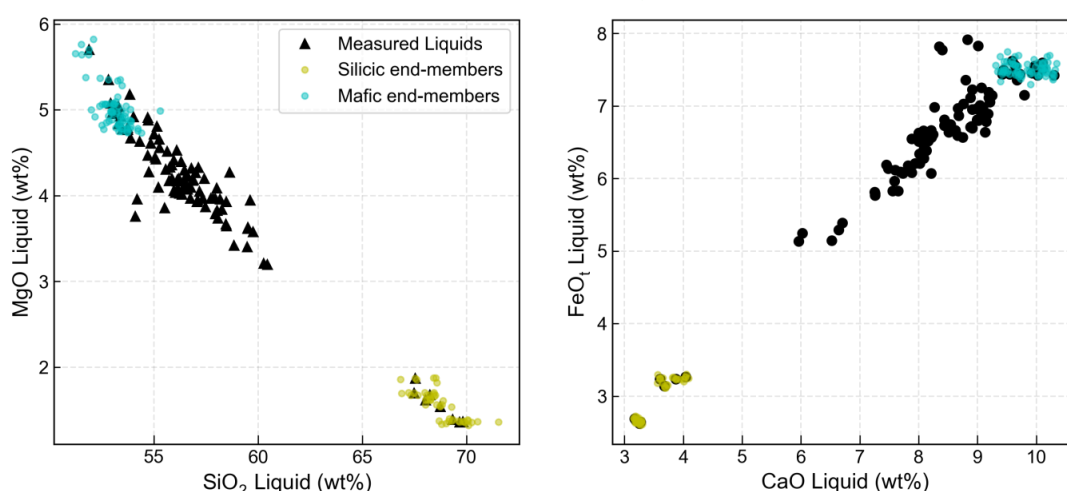
## Step 1 – Import all measured Liqs and Cpxs

```
out=pt.import_excel('Scruggs_Input.xlsx', sheet_name="Liquids")
my_input=out['my_input']
myLiquids1=out['Liqs'] ← Extracts df of liquid compositions

out2=pt.import_excel('Scruggs_Input.xlsx', sheet_name="Cpxs")
my_input2=out2['my_input']
myCpxs1=out2['Cpxs'] ← Extracts df of cpx compositions
```

## Step 2 – Identify silicic and mafic end members, then add noise

```
Sil_endmember_noise1=pt.add_noise_sample_1phase(phase_comp=myliquids1, duplicates=5, filter_q='SiO2_Liq > 65',
phase_err_type="Perc", noise_percent=1, err_dist="normal", append=True)
Maf_endmember_noise1=pt.add_noise_sample_1phase(phase_comp=myliquids1, duplicates=5, filter_q='SiO2_Liq < 53.8 & MgO_Liq>4',
phase_err_type="Perc", noise_percent=1, err_dist="normal", append=True)
Add normally distributed noise (1σ=1% relative)
Compositional filter
Compositional filter
Appends Liqs passing compositional filter to new noisy samples
```



## Step 3 – Generate synthetic liquids by mixing end-members

```
Mixed_noise1_selfbig=pt.calculate_bootstrap_mixes(endmember1=Sil_endmember_noise1,
endmember2=Maf_endmember_noise1, num_samples = 500, self_mixing = "Partial")
Generates 500 synthetic liquids
50% mafic-silicic mixes
50% mixes between all comps
```

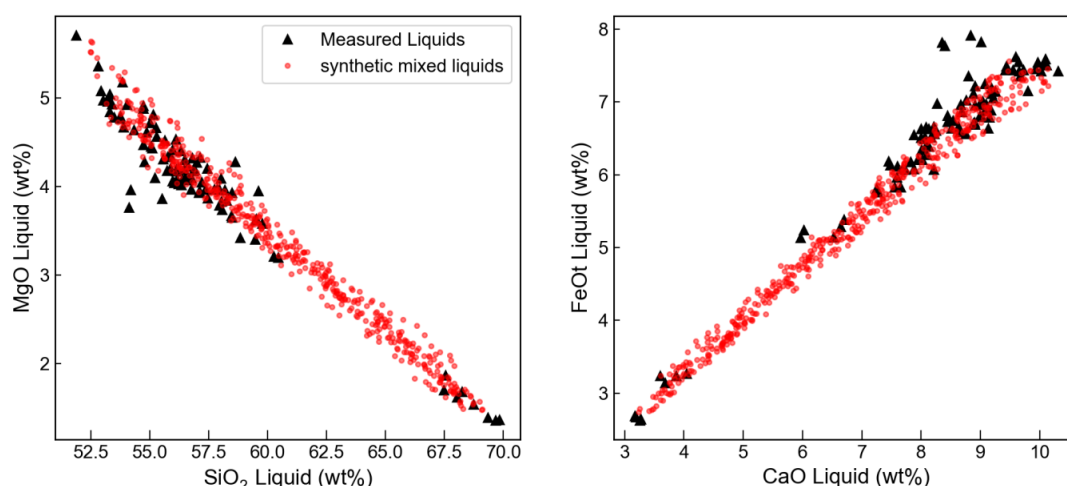


Figure 5: Example of functions allowing users to generate synthetic liquids, adapted from the approach of [Scruggs and Putirka \[2018\]](#). **Step 1:** The user reads in all measured Liq compositions into one pandas dataframe (MyLiquids1), and all Cpx into a second dataframe (myCpxs1). **Step 2:** Using as many compositional filters as required, the user defines 2 end-members. **Step 3:** These end-members are then mixed to generate 500 synthetic liquids which incorporate the variation in the natural data.



## Step 4 – Combine synthetic liquids and measured liquids

```
Liq_Comp=pd.concat([Mixed_noise1_selfbig.reset_index(drop=True),
                    myLiquids1.reset_index(drop=True)], axis=0).reset_index(drop=True).fillna(0)
```

← Combines 2 dataframes  
← Gets rid of indexing issues, replaces NaN with zeros

## Step 5 – Set water content (following Scruggs and Putirka, 2018)

```
Liq_Comp['H2O_Liq']=Liq_Comp['SiO2_Liq']*0.06995+0.383
```

← Overwrites water contents with a value dependent on SiO<sub>2</sub>

## Step 6 – Perform melt matching to calculate pressures and temperatures

```
melt_match_out_syn=pt.calculate_cpx_liq_press_temp_matching(liq_comps=Liq_Comp, cpx_comps=myCpxs1,
                    equationP="P_Neave2017", equationT="T_Put2008_eq33",
                    Kd_Match=0.27, Kd_Err=0.03)
Syn_Avs=melt_match_out_syn['Av_PTs']
Syn_All=melt_match_out_syn['All_PTs']
```

← Select equations to use  
← Considers matches with  $K_D=0.27\pm0.03$   
← Returns a dataframe with averages for all Liq matching each Cpx (e.g., P, T, Eq Tests etc.)  
← Returns a dataframe of all Cpx-Liq matches

```
Syn_Avs.head()
```

← Inspect first 5 rows of averaged dataframe

	Sample_ID_Cpx	# of Liqs Averaged	Mean_T_K_calc	Std_T_K_calc	Mean_P_kbar_calc	Std_P_kbar_calc	ID_CPX	Mean_Delta_Kd_Put2008	Mean_Delta_Kd_N
0	12	120	1306.499315	9.238036	-0.538424	0.418270	12	0.045973	C
1	16	56	1278.839001	19.757801	1.302060	0.782959	16	0.057909	C
2	26	5	1304.251537	4.271816	0.623434	0.441238	26	0.024975	C
3	29	210	1308.368271	11.250564	-0.459510	0.529018	29	0.014200	C
4	30	11	1302.787795	7.103075	-0.561921	0.219530	30	0.047838	C

## Step 7 – Visualizing matches

```
fig, (ax1) = plt.subplots(1, 1, figsize=(6, 5))
ax1.plot(Syn_All['T_K_calc'], Syn_All['P_kbar_calc'], 'or',
         ms=2, mfc='red', alpha=0.1, label='All Matches')
ax1.errorbar(Syn_Avs['Mean_T_K_calc'], Syn_Avs['Mean_P_kbar_calc'],
             xerr=Syn_Avs['Std_T_K_calc'].fillna(0),
             yerr=Syn_Avs['Std_P_kbar_calc'].fillna(0),
             fmt='d', ecolor='k', elinewidth=0.8,
             mfc='red', ms=8, mec='k', label='Average Matches')
ax1.invert_yaxis()
ax1.set_xlabel('Temp (K)')
ax1.set_ylabel('Pressure (kbar)')
ax1.legend()
fig.savefig('AllMatches_PT.png', dpi=300)
```

← Plots light red circles for all matches  
← Plots error bar for the average PT for each Cpx  
← Makes pressure increase downwards  
← Adds legend  
← Saves figure to png

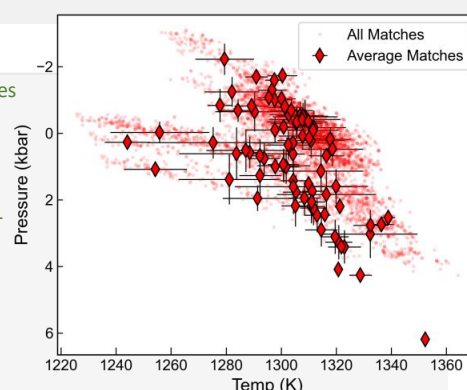


Figure 6: **Step 4:** Once synthetic liquids have been calculated, users may wish to combine them with measured liquid compositions to get the largest number of available comparisons. **Step 5:** Columns in this combined dataframe can be easily overwritten. Here, the liquid H<sub>2</sub>O content is calculated from the SiO<sub>2</sub> content of the liquid (following [Scruggs and Putirka 2018]). **Step 6:** Once the liquid input is set, the function `calculate_cpx_liq_press_temp_matching` is called, specifying the choice of liquid and Cpx compositions, as well as the equation for pressure and temperature. The function returns a dictionary, which can be subdivided into a pandas dataframe containing all matches, and a dataframe where pressures and temperatures have been averaged for all the liquids in equilibrium with a given Cpx composition. **Step 7:** Plotting both outputs gives insight into the amount of scatter associated with each Cpx-Liq pair compared to averaged outputs.

Following this filtering step, the function takes pairs in equilibrium and uses the `calculate_opx_liq_press_temp` function to calculate pressure and temperature for each pair. A dictionary is returned, containing the pressure and temperature for each pair. A second output is also calculated, where all matches for a given orthopyroxene are averaged (e.g. Opx1-Liq1, Opx1-Liq10, Opx1-Liq32). Users also have the option to overwrite the Fe<sup>3</sup>Fet\_Liq value specified in the input, as this function uses only Fe<sup>2+</sup> in the melt to calculate  $K_D^{\text{Opx-Liq}}$ .

### 7.3 Two pyroxene thermobarometry

The function `calculate_cpx_opx_press` allows users to calculate temperatures for matched Cpx-Opx pairs, `calculate_cpx_opx_press` calculates pressures, and `calculate_cpx_opx_press_temp` iterates towards a pressure and temperature. Unlike for Opx-Liq and Cpx-Liq, the function for assessing all possible Cpx-Opx pairs, `calculate_cpx_opx_press_temp_matching`, returns all pairs by default. This is because the partitioning of Fe-Mg between Cpx-Opx is the only available equilibrium test, and this





parameter shows a lot of variation between different volcanic systems. We do not intend for users to consider all pairs, but instead we strongly encourage them to think carefully about a suitable equilibrium cut-off for their system of interest. At the moment, a numerical value can be specified for `Kd_Match` and `Kd_Err`. Alternatively, specifying `Kd_Match="HighTemp"` will calculate pressures and temperatures for all Cpx-Opx pairs with  $K_D^{\text{Cpx-Opx}} = 1.09 \pm 0.14$  (suggested by Putirka [2008] for high temperature systems). Similarly, `Kd_Match="LowTemp"` uses pairs within  $0.7 \pm 0.2$  (for subsolidus systems [Putirka 2008]). As for Cpx- and Opx-Liq, the function returns a dictionary containing pressures and temperatures for all matches, as well as pressures and temperatures averaged for each Cpx, and for each Opx.

#### 7.4 Plagioclase-liquid and alkali feldspar-liquid thermobarometry

For Plag-Liq and Kspar-Liq thermobarometry, Thermobar has generic functions because the mineral component calculations of Putirka [2008] are the same for all feldspar end-members (`calculate_fspar_liq_temp`, `calculate_fspar_liq_press`, `calculate_fspar_liq_press_temp`). When these functions are used for Plag compositions, the dataframe of oxides should be entered as `plag_comps=""`, and for Kspars, as `kspar_comps=""`.

Equilibrium tests are currently only implemented for Plag, comparing the calculated and predicted An, Ab, and Or components between Plag and Liq. In particular, Putirka [2008] suggest that the Ab-An exchange coefficient is a good equilibrium test, as it varies little as a function of pressure, temperature or melt H<sub>2</sub>O content ( $\sim 0.27 \pm 0.18$ ). In their supporting spreadsheet updated since 2008, they use values of  $0.28 \pm 0.11$  for  $T > 1050^\circ\text{C}$ , and  $0.1 \pm 0.05$  for  $T < 1050^\circ\text{C}$ . In the example Jupyter notebook on the Read The Docs page\*, we demonstrate how to filter pairs using this equilibrium criteria.

`calculate_fspar_liq_temp_matching` allows all possible matches between Fspar and Liq compositions to be evaluated. As well as returning a dataframe of all matches, a second dataframe containing averages for each Fspar is returned (as for Cpx-Liq). For plagioclase inputs, if `Ab_An_P2008=True`, pairs will be filtered using the Ab-An equilibrium test of Putirka [2008].

#### 7.5 Plagioclase hygrometers

The function `calculate_fspar_liq_hygr` allows the H<sub>2</sub>O contents of liquids which crystallized Plag to be estimated. These hygrometers require users to specify the composition of the liquid, as well as the anorthite and albite content of each Plag. Analogous to the other two-phase functions, the composition of Liq and Plag dataframes are specified in the function, along with the pressure and temperature, and choice of equation (here, using the hygrometer of Waters and Lange [2015]):

```
pt.calculate_fspar_liq_hygr(
    liq_comps=myLiquids, plag_comps=myPlags,
    equationH="H_Waters2015", T=1300, P=5)
```

This returns a pandas dataframe of the calculated H<sub>2</sub>O content, along with an indicator of whether the pair passed the recommended equilibrium test of Putirka [2008] based on the temperature input by the user.

	Pass An- Ab Eq Test Put2008?	H2O_calc	Delta_An	Delta_Ab	Delta_Or	Pred_An_EqE
0	Low T: Yes	2.183611	0.056252	0.141146	0.029165	0.360876
1	Low T: Yes	2.671574	0.083157	0.227579	0.028164	0.369968

Alternatively, users can just enter the anorthite and albite content of the Plag:

```
pt.calculate_fspar_liq_hygr(
    liq_comps=myLiquids, XAn=0.5, XAb=0.4,
    equationH="H_Waters2015", T=1300, P=5)
```

As with other optional inputs, XAn and XAb can be a single value, or a pandas series with a different value for each row of the calculation.

Plag-Liq hygrometers are very sensitive to temperature. In the Read The Docs example†, we show that an increase in temperature from 1100 to 1200 K corresponds to a drop in calculated H<sub>2</sub>O contents from 5.85 to 3.64 wt.% H<sub>2</sub>O. In many cases, temperatures to use with Plag hygrometers are estimated from other mineral pairs (e.g. Fe-Ti oxides [Waters and Lange 2015]). However, there is no guarantee that different mineral phases are recording the same part of the magmatic history, and in many systems, no independent constraint on temperature exists. Given that Plag-Liq equilibria are sensitive to temperature and H<sub>2</sub>O content, we incorporate a function into Thermobar which iterates temperature and calculated H<sub>2</sub>O content by specifying a thermometer and hygrometer:

```
Dict_HT=pt.calculate_fspar_liq_temp_hygr(
    liq_comps=myLiquids, plag_comps=myPlags,
    equationT="T_Put2008_eq23",
    equationH="H_Waters2015",
    P=5, iterations=30)
```

This function returns a dictionary, comprising two dataframes:

```
Calc_HT=Dict_HT['T_H_calc']
Evol_HT=Dict_HT['T_H_Evolution']
```

The first dataframe, indicated by the key `'T_H_calc'`, contains calculated temperatures and H<sub>2</sub>O contents, as well as an indication of the change in *T* and H<sub>2</sub>O content between the final iterative step and the penultimate iterative step. If these Delta values are small, it indicates sufficient iterations were used. If these numbers are larger (e.g.  $> 0.01$ ), it indicates that the iteration has not converged. At this point, it

\*<https://bit.ly/ThermobarRTD>

†<https://bit.ly/ThermobarRTD>



is worth inspecting the second output, indicated by the key `'T_H_Evolution'`, which shows the evolution of  $T$  and  $H_2O$  for each sample against the number of iterations.

`calculate_fspar_liq_temp_hygr_matching` first considers all possible matches between Plag and Liq comps, and then performing the matching routines described above. All matches, and the average per Plag, are returned as a dictionary. This function is not currently supported for Kspar-Liq, as no Kspar-Liq hygrometers currently exist.

## 7.6 Two feldspar thermobarometry

Temperatures from co-existing Kspar-Plag pairs can be calculated using the function `calculate_plag_kspar_temp`. The function `calculate_plag_kspar_temp_matching` considers all possible pairs between a dataframe of Plag compositions, and a dataframe of Kspar compositions. Putirka [2008] suggests that a comparison of activities for An, Ab, and Or in Plag and Kspar using the models of Elkins and Grove [1990] can be used as an equilibrium test. However, Putirka [2008] notes that while the values should nominally be zero, further examination of experimental data is required to determine reasonable cut-offs. Thermobar returns the difference between these theoretical values and measured values for each pair if `eq_tests=True` (these values are returned automatically for the matching function). We provide a detailed example showing users how they could filter pairs using different values for these equilibrium tests on the Read The Docs page\*.

## 8 CONVERTING PRESSURES TO DEPTHS

It can be very useful to convert pressures from thermobarometry into depths below the surface (e.g. to compare to geophysical signals of unrest). This conversion can be done assuming a constant crustal density and the following equations:

$$P = \rho \times g \times H \quad (1)$$

Where  $P$  is pressure in Pa,  $\rho$  is the density of the crust in  $\text{kg m}^{-3}$ , and  $H$  is the height of the crustal column in m (i.e. depth). This equation can be re-arranged to calculate height (depth):

$$H = \frac{P}{\rho \times g} \quad (2)$$

After calculating pressure using any of the tools in Thermobar, users can easily convert to depth (in km) using a constant crustal density.

```
pt.convert_pressure_to_depth(
    P_kbar=Calc_P['P_kbar_calc'],
    crust_dens_kgm3=2700)
```

For example, `Calc_P` may be the dataframe returned from a Cpx-only pressure-temperature iteration.

Alternatively, a number of parametrizations between pressure and depth that account for varying crustal density are available [e.g. Putirka 2017; Lerner et al. 2021; Rasmussen et al. 2022]. These density models can be selected by specifying `model=""`. For example, to perform calculations using the

average global arc density model derived from seismic data from Rasmussen et al. [2022]:

```
pt.convert_pressure_to_depth(
    P_kbar=Calc_P['P_kbar_calc'],
    model='rasmussen')
```

Regardless of whether a density value or model is used, this function always return a `panda` series of depths in km. This function can be used in a variety of different circumstances to convert depths to pressures, including applications outside of Thermobar (e.g. melt inclusion saturation pressures). Any `panda` series, `NumPy` array or float/integer can be fed into this function using the argument `P_kbar=...`

We also provide the option for a different value of the gravitational constant to be specified in the function, so that constant-density calculations and these terrestrial profiles can be applied to other planets (although differences in crustal lithology should be evaluated).

## 9 MONTE CARLO ERROR PROPAGATION

Estimating uncertainty when performing thermobarometry and hygrometry calculations is important, particularly given that many calibrations are highly sensitive to the concentration of minor components which are difficult to measure with high precision (e.g.  $\text{Na}_2\text{O}$  in Cpx). Additionally, sometimes parameters like melt  $\text{H}_2\text{O}$  contents are poorly constrained, particularly for volcanic systems where melt inclusion analyses are sparse or absent.

The function `add_noise_sample_1phase` can be used to make synthetic distributions of mineral or liquid compositions distributed about each measured value, with options for the types (e.g. percentage or absolute) and distribution (e.g. normal or uniform) of errors. Simply, if this function is given a dataframe of five mineral or liquid compositions, it generates  $N$  duplicates of each of these rows, with a specified amount of noise added. There are a number of ways to use this function, with several worked examples on the Read the Docs page†.

In the example shown in Figure 7, we import absolute  $1\sigma$  errors from repeated analyses of Cpx and Liq in each experiment of Feig et al. [2010] (Step 1–3). We then generate 1000 synthetic Liq and Cpx compositions for each experiment (e.g. e142, e146, e148, e153, Step 4). For each actual EPMA measurement and each oxide, a value is drawn from a normal distribution with a mean of zero, and  $1\sigma$  equal to the inputted value. These values are then added to the measured value (resulting compositions shown in Step 5). These synthetic compositions can be input into any of the Thermobar functions (Step 6). In this example, we use `calculate_cpx_liq_press_temp` to iterate temperatures from Equation 31 of Putirka [2008] with the Neave and Putirka [2017] barometer to calculate the spread of  $P$  and  $T$  from each experiment.

The `av_noise_sample_series` can be used to calculate statistics for any given calculated variable, grouping simulations by the original sample name of the Liq or Cpx used to make the synthetic values (Step 7). For each Cpx-Liq pair, the

\*<https://bit.ly/ThermobarRTD>

†<https://bit.ly/ThermobarRTD>



### Step 1 – Compile measured values and errors into spreadsheet

	A	B	C	D	E	F		S	T	U	V	
1	Sample_ID_Liq	SiO <sub>2</sub> _Liq	TiO <sub>2</sub> _Liq	Al <sub>2</sub> O <sub>3</sub> _Liq	FeO <sub>t</sub> _Liq	MgO_Liq	Measured values	SiO <sub>2</sub> _Liq_Err	TiO <sub>2</sub> _Liq_Err	Al <sub>2</sub> O <sub>3</sub> _Liq_Err	FeO <sub>t</sub> _Liq_Err	1σ error (absolute, in wt%)
2	Feig2010_e142	50.97	0.49	19.35	5.33	4.51		0.33	0.04	0.22	0.43	
3	Feig2010_e146	53.64	0.62	19.32	4.88	3.26		0.33	0.03	0.24	0.21	
4	Feig2010_e148	49.63	0.37	19.10	5.30	6.40		0.48	0.03	0.24	0.30	
5	Feig2010_e153	51.62	0.44	18.45	6.30	6.13		0.44	0.05	0.23	0.22	

### Step 2 – Load measured values

```
out=pt.import_excel('Cpx_Liq_error_prop_Feig2010_example.xlsx', sheet_name="Sheet1")
my_input=out['my_input'] ← Extracts a df of all inputs
myCpxs1=out['Cpxs'] ← Extracts a df for cpx
myLiqs1=out['Liqs'] ← Extracts a df for liq
```

### Step 3 – Load Errors

```
out_err=pt.import_excel_errors('Cpx_Liq_error_prop_Feig2010_example.xlsx', sheet_name="Sheet1")
myLiqs1_err=out_err['Liqs_Err'] ← Extracts dfs of errors for each phase from
myCpxs1_err=out_err['Cpxs_Err'] ← input columns with "_Err"
myinput_out=out_err['my_input_Err']
```

### Step 4 – Make synthetic compositions distributed around each measured composition

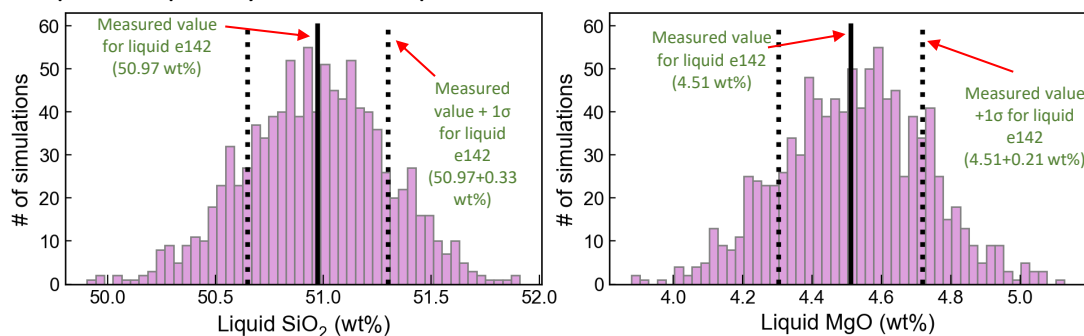
```
Liquids_st_noise=pt.add_noise_sample_1phase(phase_comp=myLiqs1, phase_err=myLiqs1_err,
                                             phase_err_type="Abs", duplicates=1000, err_dist="normal")
Cpxs_st_noise=pt.add_noise_sample_1phase(phase_comp=myCpxs1, phase_err=myCpxs1_err,
                                             phase_err_type="Abs", duplicates=1000, err_dist="normal")
```

df of measured liquid compositions → df of errors on liquid compositions → Errors are 1σ of normal distribution

Specifies df contains absolute errors in wt% → Makes 1000 synthetic analysis per measured Liq

All negative numbers replaced with zeros. If you wish to keep these, set positive=False → By default, neg values set to 0.

### Step 5 – Inspect synthetic compositions



### Step 6 – Input synthetic compositions into whatever function is of interest

```
Cpx_Liq_with_noise=pt.calculate_cpx_liq_press_temp(liq_comps=Liquids_st_noise, cpx_comps=Cpxs_st_noise,
                                                    equationP="P_Put2008_eq31", equationT="T_Put2008_eq33", eq_tests=True)
```

### Step 7 – Calculate statistics for each Cpx-Liq pair

```
Stats_P_kbar=pt.av_noise_samples_series(calc=Out_5_noise_cpx['P_kbar_calc'],
                                         sampleID=Out_5_noise_cpx['Sample_ID_Liq'])
```

Column you want to average across N duplicates →

Column with sample ID (used to group duplicates) →

	Sample	# averaged	Mean_calc	Median_calc	St_dev_calc	Max_calc	Min_calc	
0	Feig2010_e142_0	1000	3.944778	3.954911	0.242792	4.652993	3.172481	
1	Feig2010_e146_1	1000	3.745615	3.749175	0.227286	4.328572	2.909834	Outputted statistics for N duplicates for the variable of interested (P_kbar_calc here)

### Step 8 – Plot results!

Distribution for 1000 synthetic Cpx matched to 1000 synthetic liquids from a single measured Cpx-Liq pair

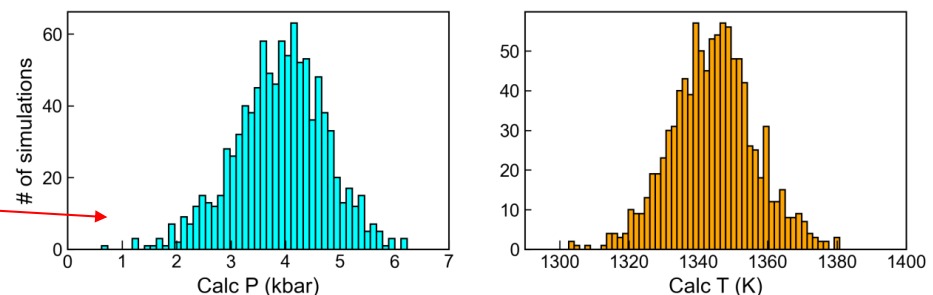


Figure 7: Investigating the range of calculated pressures and temperatures for a given distribution of noise (here, 1σ values from repeated measurements of Cpx and Liq in the experimental study of Feig et al. [2010]).



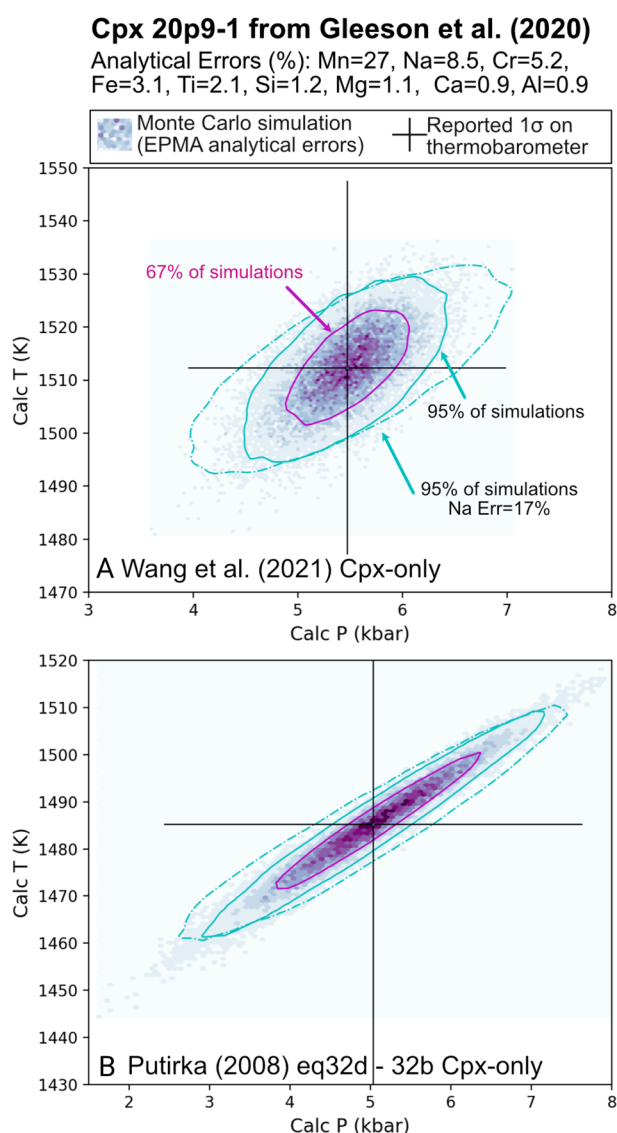


Figure 8: Propagated analytical errors from EPMA analyses into resulting distributions of pressures and temperatures. 1 $\sigma$  errors obtained from EPMA software during the analysis of a Cpx from Gleeson et al. [2020] with 0.38 wt.% Na<sub>2</sub>O was used to make 20,000 synthetic Cpx compositions. Pressures and temperature were then calculated using the Cpx-only thermobarometers of Wang et al. [2021] (their Equations 1 and 2) and Putirka [2008] (their Equations 32b–d). These results are colored using the hexbin function, and contours around 67 % and 95 % of the data are overlain using Pyrolite [Williams et al. 2020]. We also show the 95 % contour calculated for an analytical error on Na twice that reported by Gleeson et al. [2020].

distribution of pressures and temperatures can be visualized with histograms (Step 8).

It can also be informative to display calculated pressures and temperatures with contouring to show the distribution of results. Figure 8 shows a Monte Carlo simulation propagating analytical errors for measurement of a single Cpx of Gleeson et al. [2020] into a resulting error distribution for pressure and

temperature for Cpx-only thermobarometry. The propagated 1 $\sigma$  error on calculated pressure using the [Wang et al. 2021] barometer is  $\pm 0.39$  kbar and  $\pm 7$  K for temperature. Iterative solving of Equations 32b–d from Putirka [2008] yields a 1 $\sigma$  error of  $\pm 0.85$  kbar and  $\pm 10$  K. If Na<sub>2</sub>O was counted for a shorter time (or using a lower current) during electron microprobe analyses such that the analytical error was twice as large (17 %), the 1 $\sigma$  error increases to  $\pm 0.62$  kbar from Wang et al. [2021], and  $\pm 0.96$  kbar from Putirka [2008]. Importantly, these functions allow users to estimate the uncertainty resulting from their specific analytical conditions, and by extension, can be used to decide appropriate EPMA conditions to obtain a certain level of precision. The effect of analytical errors on Cpx-based barometry using these Monte Carlo functions will be discussed in detail in a follow-up publication.

## 10 SINGLE GARNET XENOCRYST THERMOBAROMETRY

Thermobarometric calculations of peridotitic garnet xenocrysts are widely used to determine the thermal structure of the underlying lithospheric mantle. The composition of the peridotitic garnet can be used as a diamond indicator [Grütter et al. 2004] and to depict the style of mantle metasomatism [Griffin et al. 2002]. Garnet thermometers utilize the strong temperature dependence on Ni-partitioning between garnet and olivine [Ryan et al. 1996; Canil 1999; Sudholz et al. 2021]. Geobarometers, on the other hand, are based on Cr-solubility in coexisting garnet and hypothetical peridotitic orthopyroxene [Ryan et al. 1996]. Thermometers and geobarometers in Thermobar can be calculated with the functions `calculate_gt_temp`, `calculate_gt_press` and `calculate_gt_press_temp`, respectively, after a user loads in garnet compositions from a spreadsheet with `_Gt` suffixes.

Constructing a geotherm with garnet thermobarometry is different to conventional curve-fitting methods. First, one must construct generalised continental geotherms [Pollack and Chapman 1977; Hasterok and Chapman 2011] and select a well-fitting one dependent on the locus defined by the maximum pressures. This is because not all garnets would potentially satisfy the Cr-saturation (in equilibrium with Cr-spinel) condition and are likely to underestimate the pressures. For this reason, the best determination can be made with depleted garnets with more numerous Cr-spinel temperatures. To determine the depths of these garnets, they have to be projected vertically down to the constructed continental geotherm. The constructed geotherms can be chosen to be kinked at the temperature at the base of the depleted lithosphere, which can be determined by a sudden population decrease of depleted garnets (Y-in-garnet <10 ppm). The temperatures after this point are not well-constrained and can be assumed to follow a kinked geotherm parallel to the diamond-graphite transition since they seem to follow that trend [Griffin et al. 2003]. This possibly indicates a local and temporal disturbance of the geotherm inflicted by a heat source [Ryan et al. 1996; Griffin et al. 2003]. These calculations can be made via the function `plot_garnet_geotherm`.



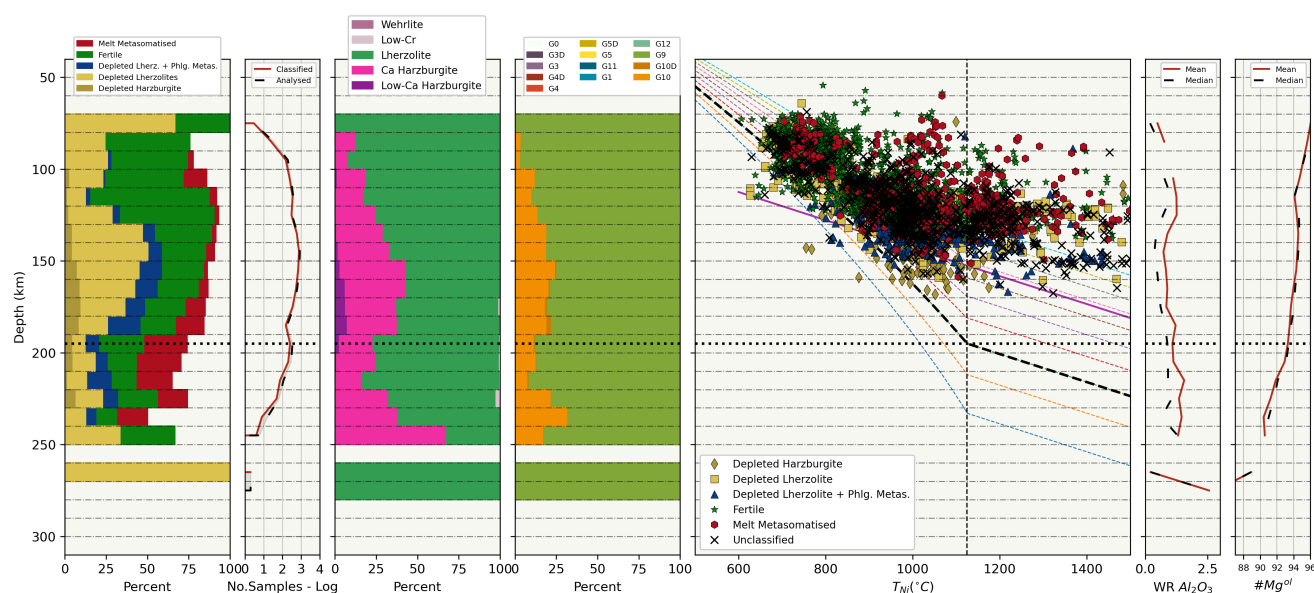


Figure 9: Composition and thermal structure gathered from garnet xenocrysts thermobarometry and chemical classification methods, with all calculations and plotting performed in Thermobar. Data is taken from Özyaydin et al. [2021]. Compositional sections are reported with histograms at each ten-kilometre section. From left to right; histogram of CARP classification scheme [Griffin et al. 2002], number of samples that could be classified with CARP scheme, CaO-Cr<sub>2</sub>O<sub>3</sub> based garnet classification scheme, the garnet classification scheme of Grütter et al. [2004], calculated *P-T* of garnet xenocryst samples against the chosen generalised continental geotherm (37 mW m<sup>-2</sup>), whole-rock Al<sub>2</sub>O<sub>3</sub> content calculated from Y-content of garnets [O'Reilly and Griffin 2006], and Mg# contents of olivine co-existing along garnet calculated with the method of Gaul et al. [2000].

### 10.1 Garnet chemical tomography

Garnet data and constructed garnet-based paleogeotherms can be utilised to depict the compositional structure of the underlying lithospheric mantle with several methods [Griffin et al. 2002]. These classifications can be carried out and plotted with the function `plot_garnet_composition_section` function in the `garnet_plot` module. To use this functionality, one needs to have the additional trace element data in addition to the major element composition.

For example, Figure 9 shows compositional and thermal information obtained from garnet xenocryst thermobarometry and chemical classification methods after Özyaydin et al. [2021], recalculated and plotted using functions in Thermobar.

## 11 INTEGRATION WITH OTHER OPEN-SOURCE Python TOOLS

In the last few years, there has been an increase in the number of petrological tools available in Python (e.g. `Pyrolite` for geochemical plotting [Williams et al. 2020], `MiMIC` for melt inclusion modification [Rasmussen et al. 2020], `VESICA` for volatile solubility [Iacovino et al. 2021]). Having thermobarometry tools available in Python through Thermobar will allow increased integration between various codes. For example, one of the most common uses of volatile solubility models is to calculate the pressure at which a melt inclusion was trapped based on reconstructing its H<sub>2</sub>O, CO<sub>2</sub>, and major element contents at the time of melt inclusion entrapment. To convert these chemical parameters into a pressure, the temperature of the melt inclusion at the time of entrapment must also be esti-

mated. On Read The Docs\* and YouTube†, we show how the functions `convert_to_VESICA` and `convert_from_VESICA` can be used to convert oxide data back and forth from the formats used in Thermobar and `VESICA` so the tools can be used together.

## 12 FUTURE WORK

The open-source nature of Thermobar, with code available on GitHub, means that users can adapt functions, add their own, or incorporate new thermobarometry or hygrometry equations as they are published. Authors publishing new thermobarometry equations can contact the author team of Thermobar, and an effort will be made to continue to update the available equations. To reflect the probable evolving nature of this tool, when citing Thermobar, users should specify which version they used, as well as citing the original equations used for calculations. For example "Cpx-Liq pressures and temperatures were calculated using equation 30 and 31 of Putirka (2008), implemented through the Python3 tool Thermobar (version 1.0.1, Wieser et al. 2022)". The version can be found after importing Thermobar by running the command:

```
pt.__version__
```

Ideally, users should provide the Jupyter notebook used for calculations for maximum reproducibility, and to outline the various options used (particularly for more complicated operation such as melt matching, error propagation).

\*<https://bit.ly/ThermobarRTD>

†<https://bit.ly/ThermobarYouTube>



### 13 CONCLUSIONS

Thermobar is a new tool that provides access to more than 100 popular thermometers, barometers, and hygrometers through easy-to-implement and customizable functions within the open-source programming language, Python3. Users can easily change the equation, pressure, temperature, proportion of  $\text{Fe}^{3+}$  and water content of calculations, iterate towards a solution when neither pressure nor temperature is known, compute equilibrium tests, and assess all possible matches of equilibrium pairs (Cpx-Liq, Opx-Cpx, Opx-Liq, Fspar-Liq) in a single line of code. The functionality of this tool will allow more robust interpretation of the systematic and random errors associated with thermobarometry and hygrometry in igneous systems. For example, the design of the functions means that users can easily switch between equations to investigate systematic differences between published parametrizations. The Monte Carlo error propagation functions allow users to assess the amount of random error introduced by their specific analytical protocol, which complements published uncertainty estimates for each equation. The fact that users can publish their workflows in a single Jupyter Notebook (rather than a myriad of different tools) will help to make thermobarometry calculations more reproducible.

### AUTHOR CONTRIBUTIONS

PW and MP conceived the project, with help from AK and CT. PW wrote the manuscript, documentation, examples and the majority of the Python code, as well as performing the benchmarking of this code to existing tools. MP and JL helped with aspects of code writing (e.g. MP-bootstrapped liquids and JL-amphibole site occupancy, boundaries for Fspar classification diagrams), as well as code testing and debugging. SO wrote the functions involving garnet and geotherms, and PW merged it into Thermobar. EW helped optimize computational speed for various iterative calculations, as well as providing guidance for writing documentation in sphinx, creating a binder file, and making the code available through pip. All authors provided feedback on the manuscript.

### ACKNOWLEDGEMENTS

We are very grateful to Keith Putirka for answering a lot of questions about the implementation of different barometers in his Excel spreadsheets, as well as very helpful discussions regarding  $K_{\text{D,Fe-Mg}}$  in different phases. We thank Euan Mutch for sharing a spreadsheet for his amphibole barometer, Tim Holland for information on his Plag-Amp thermometer, and David Neave for helpful discussions regarding his melt-matching tool. PW thanks Kayla Iacovino and Simon Matthews for introducing her to the wonderful world of developing open-source Python tools. This contribution was supported by funding from National Science Foundation grants 1948862 and 1949173 to AJRK and CBT, and start up funds to PW from UC Berkeley. MP was supported by funding from the PRIN2020 grant id:202037YPCZ\_001: “Dynamics and timescales of volcanic plumbing systems: a multidisciplinary approach to a multifaceted problem”.

### DATA AVAILABILITY

All files are available on GitHub (<https://bit.ly/ThermobarGitHub>), with documentation and examples at Read The Docs (<https://bit.ly/ThermobarRTD>) - latest version of documentation found by clicking on “latest”. The code can be run through binder on Read The Docs. YouTube videos explaining various aspects of the tool are available on the Thermobar channel (<https://bit.ly/ThermobarYouTube>). Appendix A contains summary tables showing available equations and the names they have been allocated in Thermobar.

### COPYRIGHT NOTICE

© The Author(s) 2022. This article is distributed under the terms of the **Creative Commons Attribution 4.0 International License**, which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

### REFERENCES

- Anderson, J. L. and D. R. Smith (1995). “The effects of temperature and fO<sub>2</sub> on the Al-in-hornblende barometer”. *American Mineralogist* 80(5-6), pages 549–559. DOI: [10.2138/am-1995-5-614](https://doi.org/10.2138/am-1995-5-614).
- Andrews, B. J., K. S. Befus, D. L. Blatter, M. L. Coombs, R. deGraffenried, J. E. Hammer, J. E. Gardner, J. F. Larsen, T. Shea, and H. M. N. Wright (2019). “Rapid experimental determination of magmatic phase equilibria: coordinating a volcanic crisis response protocol”. *AGU Fall Meeting Abstracts*. Volume 2019, V33A–03.
- Bachmann, O. and M. A. Dungan (2002). “Temperature-induced Al-zoning in hornblendes of the Fish Canyon magma, Colorado”. *American Mineralogist* 87(8-9), pages 1062–1076. DOI: [10.2138/am-2002-8-903](https://doi.org/10.2138/am-2002-8-903).
- Balta, J. B., M. Sanborn, H. Y. McSween Jr, and M. Wadhwa (2013). “Magmatic history and parental melt composition of olivine-phyric shergottite LAR 06319: Importance of magmatic degassing and olivine antecrysts in Martian magmatism”. *Meteoritics & Planetary Science* 48(8), pages 1359–1382. DOI: [10.1111/maps.12140](https://doi.org/10.1111/maps.12140).
- Beattie, P. (1993). “Olivine-melt and orthopyroxene-melt equilibria”. *Contributions to Mineralogy and Petrology* 115(1), pages 103–111. DOI: [10.1007/BF00712982](https://doi.org/10.1007/BF00712982).
- Blundy, J. D. and T. J. B. Holland (1990). “Calcic amphibole equilibria and a new amphibole-plagioclase geothermometer”. *Contributions to mineralogy and petrology* 104(2), pages 208–224. DOI: [10.1007/BF00306444](https://doi.org/10.1007/BF00306444).
- Brey, G. P. and T. Köhler (1990). “Geothermobarometry in four-phase lherzolites II. New thermobarometers, and practical assessment of existing thermobarometers”. *Journal of Petrology* 31(6), pages 1353–1378. DOI: [10.1093/petrology/31.6.1353](https://doi.org/10.1093/petrology/31.6.1353).
- Brugman, K. K. and C. B. Till (2019). “A low-aluminum clinopyroxene-liquid geothermometer for high-silica mag-



- matic systems". *American Mineralogist: Journal of Earth and Planetary Materials* 104(7), pages 996–1004.
- Canil, D. (1999). "The Ni-in-garnet geothermometer: calibration at natural abundances". *Contributions to Mineralogy and Petrology* 136(3), pages 240–246. DOI: [10.1007/s004100050535](#).
- Caricchi, L., M. Petrelli, E. Bali, T. Sheldrake, L. Pioli, and G. Simpson (2020). "A data driven approach to investigate the chemical variability of clinopyroxenes from the 2014–2015 Holuhraun–Bárdarbunga eruption (Iceland)". *Frontiers in Earth Science* 8, page 18. DOI: [10.3389/feart.2020.00018](#).
- Connolly, J. A. D. and K. Petrini (2002). "An automated strategy for calculation of phase diagram sections and retrieval of rock properties as a function of physical conditions". *Journal of Metamorphic Geology* 20(7), pages 697–708. DOI: [10.1046/j.1525-1314.2002.00398.x](#).
- Coogan, L., A. Saunders, and R. Wilson (2014). "Aluminum-in-olivine thermometry of primitive basalts: Evidence of an anomalously hot mantle source for large igneous provinces". *Chemical Geology* 368, pages 1–10. DOI: [10.1016/j.chemgeo.2014.01.004](#).
- Cooper, K. M. (2019). "Time scales and temperatures of crystal storage in magma reservoirs: Implications for magma reservoir dynamics". *Philosophical Transactions of the Royal Society A* 377(2139), page 20180009. DOI: [10.1098/rsta.2018.0009](#).
- Culha, C., J. Suckale, T. Keller, and Z. Qin (2020). "Crystal Fractionation by Crystal-Driven Convection". *Geophysical Research Letters* 47(4), e2019GL086784. DOI: [10.1029/2019GL086784](#).
- Danyushevsky, L. V. and P. Plechov (2011). "Petrolog3: Integrated software for modeling crystallization processes". *Geochemistry, Geophysics, Geosystems* 12(7). DOI: [10.1029/2011GC003516](#).
- De Capitani, C. and K. Petrakakis (2010). "The computation of equilibrium assemblage diagrams with Theriak/Domino software". *American mineralogist* 95(7), pages 1006–1016. DOI: [10.2138/am.2010.3354](#).
- Deer, W. A., R. A. Howie, and J. Zussman (1992). *An introduction to the Rock-forming minerals*, 3rd edition. Mineralogical Society of Great Britain & Ireland. ISBN: 978-0-90305-627-4.
- Ducea, M. N., J. B. Saleeby, and G. Bergantz (2015). "The architecture, chemistry, and evolution of continental magmatic arcs". *Annual Review of Earth and Planetary Sciences* 43, pages 299–331. DOI: [10.1146/annurev-earth-060614-105049](#).
- Elkins, L. T. and T. L. Grove (1990). "Ternary feldspar experiments and thermodynamic models". *American Mineralogist* 75(5-6), pages 544–559.
- Evans, B. W., W. Hildreth, O. Bachmann, and B. Scaillet (2016). "In defense of magnetite-ilmenite thermometry in the Bishop Tuff and its implication for gradients in silicic magma reservoirs". *American Mineralogist* 101(2), pages 469–482. DOI: [10.2138/am-2016-5367](#).
- Feig, S. T., J. Koepke, and J. E. Snow (2010). "Effect of oxygen fugacity and water on phase equilibria of a hydrous tholeiitic basalt". *Contributions to Mineralogy and Petrology* 160(4), pages 551–568. DOI: [10.1007/s00410-010-0493-3](#).
- Gaetani, G. A., J. A. O'Leary, N. Shimizu, C. E. Bucholz, and M. Newville (2012). "Rapid reequilibration of H<sub>2</sub>O and oxygen fugacity in olivine-hosted melt inclusions". *Geology* 40(10), pages 915–918. DOI: [10.1130/G32992.1](#).
- Gaul, O. F., W. L. Griffin, S. Y. O'Reilly, and N. J. Pearson (2000). "Mapping olivine composition in the lithospheric mantle". *Earth and Planetary Science Letters* 182(3-4), pages 223–235. DOI: [10.1016/S0012-821X\(00\)00243-0](#).
- Gavrilenko, M., C. Herzberg, C. Vidito, M. J. Carr, T. Tenner, and A. Ozerov (2016). "A calcium-in-olivine geohyrometer and its application to subduction zone magmatism". *Journal of Petrology* 57(9), pages 1811–1832. DOI: [10.1093/petrology/egw062](#).
- Geurts, P., D. Ernst, and L. Wehenkel (2006). "Extremely randomized trees". *Machine learning* 63(1), pages 3–42. DOI: [10.1007/s10994-006-6226-1](#).
- Ghiorso, M. S. and K. B. Prissel (2020). "ENKI Cloud App: Implementation of the Fe-Ti Oxide Geothermobarometer of Ghiorso and Evans, 2008". *Zenodo*, page 1033. DOI: [10.5281/zenodo.3866660](#). [Software project].
- Giordano, D., J. K. Russell, and D. B. Dingwell (2008). "Viscosity of magmatic liquids: a model". *Earth and Planetary Science Letters* 271(1-4), pages 123–134. DOI: [10.1016/j.epsl.2008.03.038](#).
- Gleeson, M. L. M., S. A. Gibson, and M. J. Stock (2020). "Upper mantle mush zones beneath low melt flux ocean island volcanoes: insights from Isla Floreana, Galápagos". *Journal of Petrology* 61(11-12), ega094. DOI: [10.1093/petrology/egaa094](#).
- Griffin, W. L., N. I. Fisher, J. H. Friedman, S. Y. O'Reilly, and C. G. Ryan (2002). "Cr-pyroxene garnets in the lithospheric mantle 2. Compositional populations and their distribution in time and space". *Geochemistry, Geophysics, Geosystems* 3(12), pages 1–35. DOI: [10.1029/2002GC000298](#).
- Griffin, W. L., S. Y. O'Reilly, L. M. Natapov, and C. G. Ryan (2003). "The evolution of lithospheric mantle beneath the Kalahari Craton and its margins". *Lithos* 71(2-4), pages 215–241. DOI: [10.1016/j.lithos.2003.07.006](#).
- Grütter, H. S., J. J. Gurney, A. H. Menzies, and F. Winter (2004). "An updated classification scheme for mantle-derived garnet, for use by diamond explorers". *Lithos* 77(1-4), pages 841–857. DOI: [10.1016/j.lithos.2004.04.012](#).
- Gualda, G. A. R. and M. S. Ghiorso (2014). "Phase-equilibrium geobarometers for silicic rocks based on rhyolite-MELTS. Part 1: Principles, procedures, and evaluation of the method". *Contributions to Mineralogy and Petrology* 168(1), page 1033. DOI: [10.1007/s00410-014-1033-3](#).
- Hammarstrom, J. M. and E.-a. Zen (1986). "Aluminum in hornblende: an empirical igneous geobarometer". *American mineralogist* 71(11-12), pages 1297–1313.
- Harmon, L. J., J. Cowlyn, G. A. R. Gualda, and M. S. Ghiorso (2018). "Phase-equilibrium geobarometers for silicic rocks based on rhyolite-MELTS. Part 4: plagioclase, orthopyroxene, clinopyroxene, glass geobarometer, and application to Mt. Ruapehu, New Zealand". *Contributions to Mineralogy*



- and Petrology 173(1), page 7. DOI: [10.1007/s00410-017-1428-z](https://doi.org/10.1007/s00410-017-1428-z).
- Harper, M., B. Weinstein, C. Simon, N. Swanson-Hysell, M. Greco, G. Zuidhof, et al. (2015). “python-ternary: Ternary Plots in Python”. *Zenodo*. DOI: [10.5281/zenodo.594435](https://doi.org/10.5281/zenodo.594435). [Software project].
- 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, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant (2020). “Array programming with NumPy”. *Nature* 585(7825), pages 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- Hasterok, D. and D. S. Chapman (2011). “Heat production and geotherms for the continental lithosphere”. *Earth and Planetary Science Letters* 307(1-2), pages 59–70. DOI: [10.1016/j.epsl.2011.04.034](https://doi.org/10.1016/j.epsl.2011.04.034).
- Helz, R. T. and C. R. Thornber (1987). “Geothermometry of Kilauea Iki lava lake, Hawaii”. *Bulletin of volcanology* 49(5), pages 651–668. DOI: [10.1007/BF01080357](https://doi.org/10.1007/BF01080357).
- Herzberg, C. and M. J. O’hara (2002). “Plume-associated ultramafic magmas of Phanerozoic age”. *Journal of Petrology* 43(10), pages 1857–1883.
- Hirschmann, M. M., M. S. Ghiorso, F. A. Davis, S. M. Gordon, S. Mukherjee, T. L. Grove, M. Krawczynski, E. Medard, and C. B. Till (2008). “Library of Experimental Phase Relations (LEPR): A database and Web portal for experimental magmatic phase equilibria data”. *Geochemistry, Geophysics, Geosystems* 9(3). DOI: [10.1029/2007GC001894](https://doi.org/10.1029/2007GC001894).
- Holland, T. and J. Blundy (1994). “Non-ideal interactions in calcic amphiboles and their bearing on amphibole-plagioclase thermometry”. *Contributions to mineralogy and petrology* 116(4), pages 433–447. DOI: [10.1007/BF00310910](https://doi.org/10.1007/BF00310910).
- Hollister, L. S., G. Grissom, E. Peters, H. Stowell, and V. Sisson (1987). “Confirmation of the empirical correlation of Al in hornblende with pressure of solidification of calc-alkaline plutons”. *American Mineralogist* 72(3-4), pages 231–239.
- Hunter, J. D. (2007). “Matplotlib: A 2D graphics environment”. *Computing in Science & Engineering* 9(3), pages 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- Iacovino, K., S. Matthews, P. E. Wieser, G. M. Moore, and F. Bégué (2021). “VESIcal Part I: An open-source thermodynamic model engine for mixed volatile (H<sub>2</sub>O-CO<sub>2</sub>) solubility in silicate melts”. *Earth and Space Science* 8(11), e2020EA001584. DOI: [10.1029/2020EA001584](https://doi.org/10.1029/2020EA001584).
- Johnson, M. C. (1988). “Experimental calibration of an aluminum-in-hornblende geobarometer applicable to calc-alkaline rocks”. *Eos* 69, page 1511. DOI: [10.1130/0091-7613\(1989\)017<0837:ECOTAI>2.3.CO;2](https://doi.org/10.1130/0091-7613(1989)017<0837:ECOTAI>2.3.CO;2).
- Jorgenson, C., O. Higgins, M. Petrelli, F. Bégué, and L. Caricchi (2022). “A machine learning based approach to clinopyroxene thermobarometry: model optimisation and distribution for use in Earth Sciences”. *Journal of Geophysical Research: Solid Earth*, e2021JB022904. DOI: [10.1029/2021jb022904](https://doi.org/10.1029/2021jb022904).
- Krawczynski, M. J., T. L. Grove, and H. Behrens (2012). “Amphibole stability in primitive arc magmas: effects of temperature, H<sub>2</sub>O content, and oxygen fugacity”. *Contributions to Mineralogy and Petrology* 164(2), pages 317–339. DOI: [10.1007/s00410-012-0740-x](https://doi.org/10.1007/s00410-012-0740-x).
- Leake, B. E., A. R. Woolley, C. E. Arps, W. D. Birch, M. C. Gilbert, J. D. Grice, F. C. Hawthorne, A. Kato, H. J. Kisch, V. G. Krivovichev, K. Linthout, J. Laird, J. A. Mandarino, W. V. Maresch, E. H. Nickel, N. M. S. Rock, J. C. Schumacher, D. C. Smith, N. C. N. Stephenson, L. Ungaretti, E. J. W. Whittaker, and G. Youzhi (1997). “Nomenclature of amphiboles; report of the Subcommittee on Amphiboles of the International Mineralogical Association Commission on new minerals and mineral names”. *Mineralogical magazine* 61(405), pages 295–310.
- Lee, C.-T. A. and D. L. Anderson (2015). “Continental crust formation at arcs, the arclogite “delamination” cycle, and one origin for fertile melting anomalies in the mantle”. *Science Bulletin* 60(13), pages 1141–1156. DOI: [10.1007/s11434-015-0828-6](https://doi.org/10.1007/s11434-015-0828-6).
- 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, R. L. Lee, L. R. Moore, and C. A. Werner (2021). “The petrologic and degassing behavior of sulfur and other magmatic volatiles from the 2018 eruption of Kilauea, Hawaii: melt concentrations, magma storage depths, and magma recycling”. *Bulletin of Volcanology* 83(6). DOI: [10.1007/s00445-021-01459-y](https://doi.org/10.1007/s00445-021-01459-y).
- Masotta, M. and S. Mollo (2019). “A new plagioclase-liquid hygrometer specific to trachytic systems”. *Minerals* 9(6), page 375. DOI: [10.3390/min9060375](https://doi.org/10.3390/min9060375).
- Masotta, M., S. Mollo, C. Freda, M. Gaeta, and G. Moore (2013). “Clinopyroxene-liquid thermometers and barometers specific to alkaline differentiated magmas”. *Contributions to Mineralogy and Petrology* 166(6), pages 1545–1561. DOI: [10.1007/s00410-013-0927-9](https://doi.org/10.1007/s00410-013-0927-9).
- Matthews, S., O. Shorttle, and J. MacLennan (2016). “The temperature of the Icelandic mantle from olivine-spinel aluminum exchange thermometry”. *Geochemistry, Geophysics, Geosystems* 17(11), pages 4725–4752. DOI: [10.1002/2016GC006497](https://doi.org/10.1002/2016GC006497).
- Matzen, A. K., M. B. Baker, J. R. Beckett, and E. M. Stolper (2011). “Fe–Mg partitioning between olivine and high-magnesian melts and the nature of Hawaiian parental liquids”. *Journal of Petrology* 52(7-8), pages 1243–1263. DOI: [10.1093/petrology/egq089](https://doi.org/10.1093/petrology/egq089).
- Médard, E. and J.-L. Le Pennec (2022). “Petrologic imaging of the magma reservoirs that feed large silicic eruptions”. *Lithos* 428, page 106812. DOI: [10.1016/j.lithos.2022.106812](https://doi.org/10.1016/j.lithos.2022.106812).
- Molina, J. F., J. A. Moreno, A. Castro, C. Rodríguez, and G. B. Fershtater (2015). “Calcic amphibole thermobarometry in metamorphic and igneous rocks: New calibrations based on plagioclase/amphibole Al–Si partitioning and amphibole/liquid Mg partitioning”. *Lithos* 232, pages 286–305. DOI: [10.1016/j.lithos.2015.06.027](https://doi.org/10.1016/j.lithos.2015.06.027).
- Mollo, S., K. D. Putirka, V. Misiti, M. Soligo, and P. Scarlato (2013). “A new test for equilibrium based on clinopyroxene–



- melt pairs: clues on the solidification temperatures of Etnean alkaline melts at post-eruptive conditions". *Chemical Geology* 352, pages 92–100. DOI: [10.1016/j.chemgeo.2013.05.026](https://doi.org/10.1016/j.chemgeo.2013.05.026).
- Montierth, C., A. D. Johnston, and K. V. Cashman (1995). "An empirical glass-composition-based geothermometer for Mauna Loa lavas". *Washington DC American Geophysical Union Geophysical Monograph Series* 92, pages 207–217. DOI: [10.1029/GM092p0207](https://doi.org/10.1029/GM092p0207).
- Mutch, E. J. F., J. D. Blundy, B. C. Tattitch, F. J. Cooper, and R. A. Brooker (2016). "An experimental study of amphibole stability in low-pressure granitic magmas and a revised Al-in-hornblende geobarometer". *Contributions to Mineralogy and Petrology* 171(10), pages 1–27. DOI: [10.1007/s00410-016-1298-9](https://doi.org/10.1007/s00410-016-1298-9).
- Mutch, E. J. F., J. MacLennan, O. Shorttle, J. F. Rudge, and D. A. Neave (2021). "DFENS: Diffusion Chronometry Using Finite Elements and Nested Sampling". *Geochemistry, Geophysics, Geosystems* 22(4). DOI: [10.1029/2020gc009303](https://doi.org/10.1029/2020gc009303).
- Neave, D. A., E. Bali, G. H. Guðfinnsson, S. A. Halldórsson, M. Kahl, A.-S. Schmidt, and F. Holtz (2019). "Clinopyroxene-liquid equilibria and geothermobarometry in natural and experimental tholeiites: the 2014–2015 Holuhraun eruption, Iceland". *Journal of Petrology* 60(8), pages 1653–1680. DOI: [10.1093/petrology/egz042](https://doi.org/10.1093/petrology/egz042).
- Neave, D. A. and K. D. Putirka (2017). "A new clinopyroxene-liquid barometer, and implications for magma storage pressures under Icelandic rift zones". *American Mineralogist* 102(4), pages 777–794. DOI: [10.2138/am-2017-5968](https://doi.org/10.2138/am-2017-5968).
- O'Reilly, S. Y. and W. L. Griffin (2006). "Imaging global chemical and thermal heterogeneity in the subcontinental lithospheric mantle with garnets and xenoliths: Geophysical implications". *Tectonophysics* 416(1–4), pages 289–309. DOI: [10.1016/j.tecto.2005.11.014](https://doi.org/10.1016/j.tecto.2005.11.014).
- ONNX-Runtime-developers (2021). *ONNX Runtime*. <https://www.onnxruntime.ai>. Version: x.y.z.
- Özaydin, S., K. Selway, and W. L. Griffin (2021). "Are xenoliths from southwestern Kaapvaal Craton representative of the broader mantle? Constraints from magnetotelluric modeling". *Geophysical Research Letters* 48(11), e2021GL092570. DOI: [10.1029/2021GL092570](https://doi.org/10.1029/2021GL092570).
- Petrelli, M. (2021). *Introduction to Python in Earth Science Data Analysis*. Springer Textbooks in Earth Sciences, Geography and Environment. Springer International Publishing, pages 1–229. ISBN: 978-3-030-78054-8. DOI: [10.1007/978-3-030-78055-5](https://doi.org/10.1007/978-3-030-78055-5).
- Petrelli, M., L. Caricchi, and D. Perugini (2020). "Machine learning thermo-barometry: Application to clinopyroxene-bearing magmas". *Journal of Geophysical Research: Solid Earth* 125(9), e2020JB020130. DOI: [10.1029/2020JB020130](https://doi.org/10.1029/2020JB020130).
- Pollack, H. N. and D. S. Chapman (1977). "On the regional variation of heat flow, geotherms, and lithospheric thickness". *Tectonophysics* 38(3–4), pages 279–296. DOI: [10.1016/0040-1951\(77\)90215-3](https://doi.org/10.1016/0040-1951(77)90215-3).
- Powell, R., T. Holland, and B. Worley (1998). "Calculating phase diagrams involving solid solutions via non-linear equations, with examples using THERMOCALC". *Journal of metamorphic Geology* 16(4), pages 577–588. DOI: [10.1111/j.1525-1314.1998.00157.x](https://doi.org/10.1111/j.1525-1314.1998.00157.x).
- Prissel, T. C., S. W. Parman, and J. W. Head (2016). "Formation of the lunar highlands Mg-suite as told by spinel". *American Mineralogist* 101(7), pages 1624–1635. DOI: [10.2138/am-2016-5581](https://doi.org/10.2138/am-2016-5581).
- Pritchard, M. E., T. A. Mather, S. R. McNutt, F. J. Delgado, and K. Reath (2019). "Thoughts on the criteria to determine the origin of volcanic unrest as magmatic or non-magmatic". *Philosophical Transactions of the Royal Society A* 377(2139), page 20180008. DOI: [10.1098/rsta.2018.0008](https://doi.org/10.1098/rsta.2018.0008).
- Pu, X., R. A. Lange, and G. Moore (2017). "A comparison of olivine-melt thermometers based on D Mg and D Ni: The effects of melt composition, temperature, and pressure with applications to MORBs and hydrous arc basalts". *American Mineralogist* 102(4), pages 750–765. DOI: [10.2138/am-2017-5879](https://doi.org/10.2138/am-2017-5879).
- Pu, X., G. M. Moore, R. A. Lange, J. P. Touran, and J. E. Gagnon (2021). "Experimental evaluation of a new H<sub>2</sub>O-independent thermometer based on olivine-melt Ni partitioning at crustal pressure". *American Mineralogist: Journal of Earth and Planetary Materials* 106(2), pages 235–250. DOI: [10.2138/am-2020-7014](https://doi.org/10.2138/am-2020-7014).
- Putirka, K. D. (1999). "Clinopyroxene+ liquid equilibria to 100 kbar and 2450 K". *Contributions to Mineralogy and Petrology* 135(2–3), pages 151–163. DOI: [10.1007/S004100050503](https://doi.org/10.1007/S004100050503).
- (2005). "Igneous thermometers and barometers based on plagioclase+ liquid equilibria: Tests of some existing models and new calibrations". *American Mineralogist* 90(2–3), pages 336–346. DOI: [10.2138/am.2005.1449](https://doi.org/10.2138/am.2005.1449).
- (2008). "Thermometers and barometers for volcanic systems". *Reviews in mineralogy and geochemistry* 69(1), pages 61–120. DOI: [10.2138/rmg.2008.69.3](https://doi.org/10.2138/rmg.2008.69.3).
- (2016). "Amphibole thermometers and barometers for igneous systems and some implications for eruption mechanisms of felsic magmas at arc volcanoes". *American Mineralogist* 101(4), pages 841–858. DOI: [10.2138/am-2016-5506](https://doi.org/10.2138/am-2016-5506).
- (2017). "Down the crater: where magmas are stored and why they erupt". *Elements* 13(1), pages 11–16. DOI: [10.2113/gselements.13.1.11](https://doi.org/10.2113/gselements.13.1.11).
- Putirka, K. D., M. Johnson, R. Kinzler, J. Longhi, and D. Walker (1996). "Thermobarometry of mafic igneous rocks based on clinopyroxene-liquid equilibria, 0–30 kbar". *Contributions to Mineralogy and Petrology* 123(1), pages 92–108. DOI: [10.1007/s004100050145](https://doi.org/10.1007/s004100050145).
- Putirka, K. D., F. J. Ryerson, and H. Mikaelian (2003). "New igneous thermobarometers for mafic and evolved lava compositions, based on clinopyroxene+ liquid equilibria". *American Mineralogist* 88, pages 1542–1554. DOI: <https://doi.org/10.2138/am-2003-1017>.
- 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](https://doi.org/10.1126/science.abm5174).
- Rasmussen, D. J., T. A. Plank, P. J. Wallace, M. E. Newcombe, and J. B. Lowenstern (2020). "Vapor-bubble growth



- in olivine-hosted melt inclusions". *American Mineralogist: Journal of Earth and Planetary Materials* 105(12), pages 1898–1919. DOI: [10.2138/am-2020-7377](https://doi.org/10.2138/am-2020-7377).
- Ridolfi, F. (2021). "Amp-TB2: An Updated Model for Calcic Amphibole Thermobarometry". *Minerals* 11(3), page 324. DOI: [10.3390/min11030324](https://doi.org/10.3390/min11030324).
- Ridolfi, F. and A. Renzulli (2012). "Calcic amphiboles in calc-alkaline and alkaline magmas: thermobarometric and chemometric empirical equations valid up to 1,130° C and 2.2 GPa". *Contributions to Mineralogy and Petrology* 163(5), pages 877–895. DOI: [10.1007/s00410-011-0704-6](https://doi.org/10.1007/s00410-011-0704-6).
- Ridolfi, F., A. Renzulli, and M. Puerini (2010). "Stability and chemical equilibrium of amphibole in calc-alkaline magmas: an overview, new thermobarometric formulations and application to subduction-related volcanoes". *Contributions to Mineralogy and Petrology* 160(1), pages 45–66. DOI: [10.1093/petrology/egaa094](https://doi.org/10.1093/petrology/egaa094).
- Roeder, P. L. and R. F. I. Emslie (1970). "Olivine-liquid equilibrium". *Contributions to Mineralogy and Petrology* 29(4), pages 275–289. DOI: [10.1007/BF00371276](https://doi.org/10.1007/BF00371276).
- Rout, S. S., M. Blum-Oeste, and G. Wörner (2021). "Long-Term Temperature Cycling in a Shallow Magma Reservoir: Insights from Sanidine Megacrysts at Taápaca Volcano, Central Andes". *Journal of Petrology* 62(9). DOI: [10.1093/petrology/egab010](https://doi.org/10.1093/petrology/egab010).
- Rudnick, R. L. (1995). "Making continental crust". *Nature* 378(6557), pages 571–578. DOI: [10.1038/378571a0](https://doi.org/10.1038/378571a0).
- Ryan, C. G., W. L. Griffin, and N. J. Pearson (1996). "Garnet geotherms: Pressure-temperature data from Cr-pyroxene garnet xenocrysts in volcanic rocks". *Journal of Geophysical Research: Solid Earth* 101(B3), pages 5611–5625. DOI: [10.1029/95JB03207](https://doi.org/10.1029/95JB03207).
- Schmidt, M. W. (1992). "Amphibole composition in tonalite as a function of pressure: an experimental calibration of the Al-in-hornblende barometer". *Contributions to mineralogy and petrology* 110(2-3), pages 304–310. DOI: [10.1007/BF00310745](https://doi.org/10.1007/BF00310745).
- Scruggs, M. A. and K. D. Putirka (2018). "Eruption triggering by partial crystallization of mafic enclaves at Chaos Crags, Lassen Volcanic Center, California". *American Mineralogist: Journal of Earth and Planetary Materials* 103(10), pages 1575–1590. DOI: [10.2138/am-2018-6058](https://doi.org/10.2138/am-2018-6058).
- Shamloo, H. I. and C. B. Till (2019). "Decadal transition from quiescence to supereruption: petrologic investigation of the Lava Creek Tuff, Yellowstone Caldera, WY". *Contributions to Mineralogy and Petrology* 174(4), pages 1–18. DOI: [10.1007/s00410-019-1570-x](https://doi.org/10.1007/s00410-019-1570-x).
- Sisson, T. W. and T. L. Grove (1993). "Temperatures and H<sub>2</sub>O contents of low-MgO high-alumina basalts". *Contributions to Mineralogy and Petrology* 113(2), pages 167–184. DOI: [10.1007/BF00283226](https://doi.org/10.1007/BF00283226).
- Stock, M. J., M. Bagnardi, D. A. Neave, J. MacLennan, B. Bernard, I. Buisman, M. L. M. Gleeson, and D. Geist (2018). "Integrated petrological and geophysical constraints on magma system architecture in the western Galápagos Archipelago: insights from Wolf volcano". *Geochemistry, Geophysics, Geosystems* 19(12), pages 4722–4743. DOI: [10.1029/2018GC007936](https://doi.org/10.1029/2018GC007936).
- Stock, M. J., M. C. Humphreys, V. C. Smith, R. Isaia, and D. M. Pyle (2016). "Late-stage volatile saturation as a potential trigger for explosive volcanic eruptions". *Nature Geoscience* 9(3), pages 249–254. DOI: [10.1038/ngeo2639](https://doi.org/10.1038/ngeo2639).
- Sudholz, Z., G. Yaxley, A. Jaques, and J. Chen (2021). "Ni-in-garnet geothermometry in mantle rocks: a high pressure experimental recalibration between 1100 and 1325° C". *Contributions to Mineralogy and Petrology* 176(5), pages 1–16. DOI: [10.1007/s00410-021-01791-8](https://doi.org/10.1007/s00410-021-01791-8).
- Sugawara, T. (2000). "Empirical relationships between temperature, pressure, and MgO content in olivine and pyroxene saturated liquid". *Journal of Geophysical Research: Solid Earth* 105(B4), pages 8457–8472. DOI: [10.1029/2000JB900010](https://doi.org/10.1029/2000JB900010).
- Szymanowski, D., J.-F. Wotzlaw, B. S. Ellis, O. Bachmann, M. Guillong, and A. von Quadt (2017). "Protracted near-solidus storage and pre-eruptive rejuvenation of large magma reservoirs". *Nature Geoscience* 10(10), pages 777–782. DOI: [10.1038/ngeo3020](https://doi.org/10.1038/ngeo3020).
- The pandas development team (2020). "pandas-dev/pandas: Pandas". Version latest. *Zenodo*. DOI: [10.5281/zenodo.3509134](https://doi.org/10.5281/zenodo.3509134). [Software].
- Till, C. B. (2017). "A review and update of mantle thermobarometry for primitive arc magmas". *American Mineralogist* 102(5), pages 931–947. DOI: [10.2138/am-2017-5783](https://doi.org/10.2138/am-2017-5783).
- Toplis, M. J. (2005). "The thermodynamics of iron and magnesium partitioning between olivine and liquid: criteria for assessing and predicting equilibrium in natural and experimental systems". *Contributions to Mineralogy and Petrology* 149(1), pages 22–39. DOI: [10.1007/s00410-004-0629-4](https://doi.org/10.1007/s00410-004-0629-4).
- Walker, B. A., E. W. Klemetti, A. L. Gruner, J. H. Dilles, F. J. Tepley, and D. Giles (2013). "Crystal rearing during the assembly, maturation, and waning of an eleven-million-year crustal magma cycle: thermobarometry of the Aucanquilcha Volcanic Cluster". *Contributions to Mineralogy and Petrology* 165(4), pages 663–682. DOI: [10.1007/s00410-012-0829-2](https://doi.org/10.1007/s00410-012-0829-2).
- Wan, Z., L. A. Coogan, and D. Canil (2008). "Experimental calibration of aluminum partitioning between olivine and spinel as a geothermometer". *American Mineralogist* 93(7), pages 1142–1147. DOI: [10.2138/am.2008.2758](https://doi.org/10.2138/am.2008.2758).
- Wang, X., T. Hou, M. Wang, C. Zhang, Z. Zhang, R. Pan, F. Marxer, and H. Zhang (2021). "A new clinopyroxene thermobarometer for mafic to intermediate magmatic systems". *European Journal of Mineralogy* 33(5), pages 621–637. DOI: [10.5194/ejm-33-621-2021](https://doi.org/10.5194/ejm-33-621-2021).
- Waters, L. E. and R. A. Lange (2015). "An updated calibration of the plagioclase-liquid hygrometer-thermometer applicable to basalts through rhyolites". *American Mineralogist* 100(10), pages 2172–2184. DOI: [10.2138/am-2015-5232](https://doi.org/10.2138/am-2015-5232).
- Wells, P. R. (1977). "Pyroxene thermometry in simple and complex systems". *Contributions to mineralogy and Petrology* 62(2), pages 129–139. DOI: [10.1007/BF00372872](https://doi.org/10.1007/BF00372872).
- Wieser, P. E., M. Edmonds, J. MacLennan, F. E. Jenner, and B. E. Kunz (2019a). "Crystal scavenging from mush piles recorded by melt inclusions". *Nature communications* 10(1), pages 1–11. DOI: [10.1038/s41467-019-13518-2](https://doi.org/10.1038/s41467-019-13518-2).



- Wieser, P. E., H. Lamadrid, J. Maclennan, M. Edmonds, S. Matthews, K. Iacovino, F. E. Jenner, C. Gansecki, F. Trussdell, R. L. Lee, et al. (2021). “Reconstructing magma storage depths for the 2018 Kilauean eruption from melt inclusion CO<sub>2</sub> contents: the importance of vapor bubbles”. *Geochemistry, Geophysics, Geosystems* 22(2), e2020GC009364. DOI: [10.1029/2020GC009364](https://doi.org/10.1029/2020GC009364).
- Wieser, P. E., Z. Vukmanovic, R. Kilian, E. Ringe, M. B. Holness, J. Maclennan, and M. Edmonds (2019b). “To sink, swim, twin, or nucleate: A critical appraisal of crystal aggregation processes”. *Geology* 47(10), pages 948–952. DOI: [10.1130/G46660.1](https://doi.org/10.1130/G46660.1).
- Williams, M. J., L. Schoneveld, Y. Mao, J. Klump, J. Gosses, H. Dalton, A. Bath, and S. Barnes (2020). “pyrolite: Python for geochemistry”. *Journal of Open Source Software* 5(50), page 2314. DOI: [10.21105/joss.02314](https://doi.org/10.21105/joss.02314).
- Winpenny, B. and J. Maclennan (2011). “A partial record of mixing of mantle melts preserved in Icelandic phenocrysts”. *Journal of Petrology* 52(9), pages 1791–1812. DOI: [10.1093/petrology/egr031](https://doi.org/10.1093/petrology/egr031).
- Wood, B. J. and S. Banno (1973). “Garnet-orthopyroxene and orthopyroxene-clinopyroxene relationships in simple and complex systems”. *Contributions to Mineralogy and Petrology* 42(2), pages 109–124. DOI: [10.1007/BF00371501](https://doi.org/10.1007/BF00371501).
- Zhang, J., M. C. S. Humphreys, G. F. Cooper, J. P. Davidson, and C. G. Macpherson (2017). “Magma mush chemistry at subduction zones, revealed by new melt major element inversion from calcic amphiboles”. *American Mineralogist: Journal of Earth and Planetary Materials* 102(6), pages 1353–1367. DOI: [10.2138/am-2017-5928](https://doi.org/10.2138/am-2017-5928).



## APPENDIX A

Summary tables showing available equations and the names they have been allocated in Thermobar.

Table A1: Summary of equations for liquid-only thermometry. \*Note, **Putirka 2016** equation 3 does not contain a H<sub>2</sub>O term, but is H<sub>2</sub>O-sensitive because liquid cation fractions are calculated on a hydrous basis. Equations from: **Putirka 2008**, **Sugawara 2000**, **Montieth et al. 1995**, **Helz and Thornber 1987**, **Beattie 1993**, **Herzberg and O'hara 2002**, **Putirka 1999**, **Molina et al. 2015**, **Putirka 2016**.

### Liquid-only thermometry

Function: "calculate\_liq\_only\_temp"

Reference	Name in Thermobar	P-dependent?	H <sub>2</sub> O-dependent?
<b>Olivine-Sat Liquids</b>			
Putirka (2008)	T_Put2008_eq13	X	X
	T_Put2008_eq14	X	✓
	T_Put2008_eq15	✓	✓
Helz & Thornber, (1987)	T_Helz1987_MgO	X	X
Montieth (1995)	T_Montieth1995_MgO	X	X
Sugawara (2000)	T_Sug2000_eq1	X	X
	T_Sug2000_eq3_ol	✓	X
	T_Sug2000_eq6a	✓	X
	T_Sug2000_eq6a_H7a	✓	✓
Beattie (1993)	T_Beatt93_BeattDMg	✓	X
	T_Beatt93_BeattDMg_HerzCorr	✓	X
Putirka (2008)	T_Put2008_eq19_BeattDMg	✓	X
	T_Put2008_eq21_BeattDMg	✓	✓
	T_Put2008_eq22_BeattDMg	✓	✓
<b>Cpx-Sat Liquids</b>			
Putirka (2008)	T_Put2008_eq34_cpx_sat	✓	✓
Putirka (1999)	T_Put1999_cpx_sat	✓	X
Sugawara (2000)	T_Sug2000_eq3_cpx	✓	X
	T_Sug2000_eq3_pig	✓	X
	T_Sug2000_eq6b	✓	X
	T_Sug2000_eq6b_H7b	✓	✓
<b>Opx-Sat Liquids</b>			
Putirka (2008)	T_Put2008_eq28b_opx_sat	✓	✓
Sugawara (2000)	T_Sug2000_eq3_opx	✓	X
Beattie (1993)	T_Beatt1993_opx	✓	X
<b>Amp-Sat Liquids</b>			
Putirka (2008)	T_Put2016_eq3_amp_sat	X	✓*
Molina (2015)	T_Molina2015_amp_sat	X	X
<b>Fspar-Sat Liquids</b>			
Putirka (2005)	T_Put2005_eqD_plag_sat	✓	✓
Putirka (2008)	T_Put2008_eq26_plag_sat	✓	✓
	T_Put2008_eq24c_kspar_sat	✓	✓
<b>OI-Cpx-Plag Sat Liquids</b>			
Putirka (2008)	T_Put2008_eq16	✓	X
Helz & Thornber (1987)	T_Helz1987_CaO	X	X



Table A2: Summary of equations for olivine-liquid and olivine-spinel thermometry. From: Putirka 2008, Beattie 1993, Herzberg and O'hara 2002, Sisson and Grove 1993, Pu et al. 2021, Pu et al. 2017, Wan et al. 2008, Coogan et al. 2014, Gavrilenko et al. 2016.

Olivine Thermometers and Hygrometers				
Reference	Name in Thermobar	T-dependent?	P-dependent?	H <sub>2</sub> O-dependent?
Olivine-Liquid thermometry. Function “calculate_ol_liq_temp”				
Putirka (2008)	T_Put2008_eq19		✓	✗
	T_Put2008_eq21		✓	✓
	T_Put2008_eq22		✓	✓
Beattie (1993)	T_Beatt93_ol		✓	✗
	T_Beatt93_ol_HerzCorr		✓	✗
Sisson and Grove (1992)	T_Sisson1992		✓	✗
Pu et al. (2017)	T_Pu2017		✗	✗
Pu et al. (2021)	T_Pu2021		✓	✗
Olivine-Liquid hygrometers. Function “calculate_ol_liq_hygr”				
Gavrilenko et al. (2016)	H_Gavr2016	✗	✗	
Olivine-Spinel thermometry. Function “calculate_ol_sp_temp”				
Coogan et al. (2014)	T_Coogan2014		✗	✗
Wan et al. (2008)	T_Wan2008		✗	✗

Table A3: Summary of equations for feldspar thermobarometry and hygrometry. From: Putirka 2008, Putirka 2005, Waters and Lange 2015, Masotta and Mollo 2019.

Feldspar Thermometers, Barometers and Hygrometers					
Phase	Reference	Name in Thermobar	T-dependent?	P-dependent?	H <sub>2</sub> O-dependent?
Feldspar-Liquid thermometry. Function “calculate_fspar_liq_temp”					
Plag-Liq	Putirka (2008)	T_Put2008_eq23		✓	✓
		T_Put2008_eq24a		✓	✓
Kspar-Liq	Putirka (2008)	T_Put2008_eq24b		✓	✗
Feldspar-Liquid barometry. Function “calculate_fspar_liq_press”					
Plag-Liq	Putirka (2008)	P_Put2008_eq25	✓		✗
Feldspar-Liquid hygrometry. Function “calculate_fspar_liq_hygr”					
Plag-Liq	Putirka (2008)	H_Put2008_eq25b	✓	✓	
	Putirka (2005)	H_Put2005_eqH	✓	✗	
	Waters & Lange (2015)	H_Waters2015	✓	✓	
	Masotta et al. (2019)	H_Masotta2019	✓	✗	
Plagioclase-Alkali Feldspar thermometry. Function “calculate_plag_kspar_temp”					
Plag-Kspar	Putirka (2008)	T_Put2008_eq27a		✓	✗
		T_Put2008_eq27b		✓	✗
		T_Put_Global_2Fspar		✓	✗



Table A4: Summary of equations for Cpx thermobarometry. Equations marked with \*1 have two forms: in addition to that shown, users can also add \_onnx (e.g. P\_Petrelli2020\_Cpx\_only\_onnx). From: Putirka et al. 1996, Putirka et al. 2003, Putirka 2008, Masotta et al. 2013, Neave and Putirka 2017, Brugman and Till 2019, Petrelli 2021, Wang et al. 2021, Jorgenson et al. 2022.

### Clinopyroxene-Liquid Thermobarometers

Reference	Name in Thermobar	T-dependent?	P-dependent?	H <sub>2</sub> O-dependent?
Clinopyroxene-Liquid Barometry. Function “calculate_cpx_liq_press”				
Putirka (1996)	P_Put1996_eqP1	✓		X
	P_Put1996_eqP2	✓		X
Putirka (2003)	P_Put2003	✓		X
Putirka (2008)	P_Put2008_eq30	✓		✓
	P_Put2008_eq31	✓		✓
	P_Put2008_eq32c	✓		✓
Masotta et al. (2013) <i>recalibration of Putirka eqs. for alkali systems</i>	P_Mas2013_eqPalk1tex	✓		X
	P_Mas2013_eqPalk2	✓		X
	P_Mas2013_eqalk32c	✓		✓
Masotta et al. (2013)	P_Mas2013_Palk2012	X		✓
Neave & Putirka (2017)	P_Neave2017	✓		X
Petrelli et al. (2020)	P_Petrelli2020_Cpx_Liq* <sup>1</sup>	X		✓
Jorgenson et al. (2022)	P_Jorgenson2022_Cpx_Liq* <sup>1</sup>	X		X
Clinopyroxene-Liquid Thermometry. Function “calculate_cpx_liq_temp”				
Putirka (1996)	T_Put1996_eqT1		X	X
	T_Put1996_eqT2		✓	X
Putirka (1999)	T_Put1999		✓	X
Putirka (2003)	T_Put2003		✓	X
Putirka (2008)	T_Put2008_eq33		✓	✓
Masotta et al. (2013) <i>Recalibration of Putirka eqs. for alkali systems</i>	T_Mas2013_eqTalk1		X	X
	T_Mas2013_eqTalk2		✓	X
	T_Mas2013_eqalk33		✓	✓
Masotta et al. (2013)	T_Mas2013_Talk2012		X	✓
Brugman & Till (2019)	T_Brug2019		X	X
Petrelli et al. (2020)	T_Petrelli2020_Cpx_Liq* <sup>1</sup>		X	✓
Jorgenson et al. (2022)	T_Jorgenson2022_Cpx_Liq* <sup>1</sup>		X	X

### Clinopyroxene-only Thermobarometers

Reference	Name in Thermobar	T-dependent?	P-dependent?	H <sub>2</sub> O-dependent?
Clinopyroxene-only Barometry. Function “calculate_cpx_only_press”				
Putirka (2008)	P_Put2008_eq32a	✓		✗
	P_Put2008_eq32b	✓		✓
Petrelli et al. (2020)	P_Petrelli2020_Cpx_only* <sup>1</sup>	✗		✗
*our adaptations	P_Petrelli2020_Cpx_only_withH2O*	✗		✓
Wang et al. (2021)	P_Wang2021_eq1	✗		✗
Jorgenson et al. (2022)	P_Jorgenson2022_Cpx_only* <sup>1</sup>	✗		✗
Clinopyroxene-only Thermometry. Function “calculate_cpx_only_temp”				
Putirka (2008)	T_Put2008_eq32d		✓	✗
	T_Put2008_eq32d_subsol		✓	✗
Wang et al. (2021)	T_Wang2021_eq2		✗	✓
Jorgenson et al. (2022)	T_Jorgenson2022_Cpx_only* <sup>1</sup>		✗	✗



Table A5: Summary of equations for Opx and Cpx-Opx thermobarometry. From: Putirka 2008, Brey and Köhler 1990, Wells 1977, Wood and Banno 1973. The "Global" and "Felsic" orthopyroxene barometers are from the spreadsheets currently available at <https://bit.ly/PutirkaSpreadsheets>. These equations are particularly suited to low pressure, low-Al orthopyroxenes where other equations return a numerical error.

### Orthopyroxene Thermobarometers

Reference	Name in Thermobar	T-dependent?	P-dependent?	H <sub>2</sub> O-dependent?
Orthopyroxene-Liquid Barometry. Function "calculate_opx_liq_press"				
Putirka (2008)	P_Put2008_eq29a	✓		✓
	P_Put2008_eq29b	✓		✓
Putirka Supplement New "Global" calibrations	P_Put_Global_Opx	✗		✗
	P_Put_Felsic_Opx	✗		✗
Orthopyroxene-Liquid Thermometry. Function "calculate_opx_liq_temp"				
Putirka (2008)	T_Put2008_eq28a		✓	✓
	T_Put2008_eq28b_opx_sat		✓	✓
Orthopyroxene-only Barometry. Function "calculate_opx_only_press"				
Putirka (2008)	P_Put2008_eq29c	✓		✗

### Orthopyroxene-Clinopyroxene Thermobarometers

Reference	Name in Thermobar	T-dependent?	P-dependent?	H <sub>2</sub> O-dependent?
Orthopyroxene-Clinopyroxene Barometry. Function "calculate_cpx_opx_press"				
Putirka (2008)	P_Put2008_eq38	✗		✗
	P_Put2008_eq39	✓		✗
Orthopyroxene-Clinopyroxene Thermometry. Function "calculate_cpx_opx_press"				
Putirka (2008)	T_Put2008_eq36		✓	✗
	T_Put2008_eq37		✓	✗
Brey and Kohler (1990)	T_Brey1990		✓	✗
Wells (1977)	T_Wells1977		✗	✗
Wood and Banno (1973)	T_Wood1973		✗	✗



Table A6: Summary of equations for amphibole thermobarometry. From: Ridolfi 2021, Putirka 2016, Mutch et al. 2016, Krawczynski et al. 2012, Ridolfi and Renzulli 2012, Hollister et al. 1987, Ridolfi et al. 2010, Hammarstrom and Zen 1986, Johnson 1988, Blundy and Holland 1990, Schmidt 1992, Anderson and Smith 1995, Holland and Blundy 1994, Médard and Le Pennec 2022.

### Amphibole Thermobarometers

Reference	Name in Thermobar	T-dependent?	P-dependent?	H <sub>2</sub> O-dependent?
Amphibole-Liquid Barometry. Function “calculate_amp_liq_press”				
Putirka (2016)	P_Put2016_eq7a	X		✓
	P_Put2016_eq7b	X		✓ <sup>1*</sup>
	P_Put2016_eq7c	X		X
Amphibole-Liquid Thermometry. Function “calculate_amp_liq_temp”				
Putirka (2016)	T_Put2016_eq4b		X	✓
	T_Put2016_eq4a_amp_sat		X	✓ <sup>1*</sup>
	T_Put2016_eq9		X	✓ <sup>1*</sup>
Amphibole-only Barometry. Function “calculate_amp_only_press”				
Medard & Pennec (2022) <sup>*2</sup>	P_Medard2022_RidolfiSites P_Medard2022_LeakeSites P_Medard2022_MutchSites	X		X
Ridolfi and Renzulli (2012) & Ridolfi (2021)	P_Ridolfi2012_1a	X		X
	P_Ridolfi2012_1b	X		X
	P_Ridolfi2012_1c	X		X
	P_Ridolfi2012_1d	X		X
	P_Ridolfi2012_1e	X		X
	P_Ridolfi2021 <sup>*3</sup>	X		X
Mutch et al. (2016)	P_Mutch2016	X		X
Ridolfi et al. (2010)	P_Ridolfi2010	X		X
Hammarstrom & Zen (1986)	P_Hammarstrom1986_eq1	X		X
	P_Hammarstrom1986_eq2	X		X
	P_Hammarstrom1986_eq3	X		X
Hollister et al. (1987)	P_Hollister1987	X		X
Johnson & Rutherford (1989)	P_Johnson1989	X		X
Blundy et al. (1990)	P_Blundy1990	X		X
Schmidt (1992)	P_Schmidt1992	X		X
Anderson & Smith, 1995	P_Anderson1995	✓		X
Krawczynski et al. (2012)	P_Kraw2012	X		X
Amphibole-only Thermometry. Function “calculate_amp_only_temp”				
Putirka (2016)	T_Put2016_eq5		X	X
	T_Put2016_eq6		X	X
	T_Put2016_SiHbl		X	X
	T_Put2016_eq8		✓	X
Ridolfi and Renzuli, 2012	T_Ridolfi2012		✓	X
Amphibole-Plagioclase Thermometry. Function “calculate_amp_plag_temp”				
Holland and Blundy, 1994	T_HB1994_A		✓	X
	T_HB1994_B			

✓<sup>1\*</sup> H<sub>2</sub>O-dependence because of parameterization in terms of hydrous fractions, not a specific H<sub>2</sub>O-term

<sup>\*2</sup> We provide 3 options for how to calculate Al<sup>VI</sup>

<sup>\*3</sup> Equation P=“P\_Ridolfi2021” uses an algorithm to combine results of eq1a-1e



Table A7: Summary of equations for amphibole chemometers. From: Putirka 2016, Zhang et al. 2017, and Ridolfi 2021.

Amphibole Chemometers			
Reference	Melt parameter	Output name	T-dependent?
<b>Amphibole-only Chemometry. Function "calculate_amp_only_melt_comps"</b> Returns all equations by default (need to specify T to get T-dependent equations)			
Ridolfi (2021)	$\Delta\text{NNO}$	deltaNNO_Ridolfi21	X
	H <sub>2</sub> O	H2O_Ridolfi21	X
Zhang et al. (2017)	SiO <sub>2</sub> (Eq 1)	SiO2_Eq1_Zhang17	X
	SiO <sub>2</sub> (Eq 2)	SiO2_Eq2_Zhang17	X
	SiO <sub>2</sub> (Eq 3)	SiO2_Eq3_Zhang17	✓
	SiO <sub>2</sub> (Eq 4)	SiO2_Eq4_Zhang17	X
	TiO <sub>2</sub> (Eq 5)	TiO2_Eq5_Zhang17	✓
	TiO <sub>2</sub> (Eq 6)	TiO2_Eq6_Zhang17	X
	FeO (Eq 7)	FeO_Eq7_Zhang17	X
	FeO (Eq 8)	FeO_Eq8_Zhang17	X
	MgO (Eq 9)	MgO_Eq9_Zhang17	X
	CaO (Eq 10)	CaO_Eq10_Zhang17	X
	CaO (Eq 11)	CaO_Eq11_Zhang17	X
	K <sub>2</sub> O (Eq 12)	K2O_Eq12_Zhang17	X
	K <sub>2</sub> O (Eq 13)	K2O_Eq13_Zhang17	X
	Al <sub>2</sub> O <sub>3</sub> (Eq 14)	Al2O3_Eq14_Zhang17	X
Putirka (2016)	SiO <sub>2</sub> (Eq 10)	SiO2_Eq10_Put2016	✓

Table A8: Summary of equations for Garnet calculations. From: Ryan et al. 1996, Canil 1999, Sudholz et al. 2021, Griffin et al. 2002, Grütter et al. 2004, Gaul et al. 2000 and O'Reilly and Griffin 2006.

Garnet Thermometers and Barometers				
Reference	Name in Thermobar	T-dependent?	P-dependent?	H2O-dependent?
Garnet-only thermometry. Function "calculate_gt_only_temp"				
Ryan et al. (1996)	T_Ryan1996		X	X
Canil et al. (1999)	T_Canil1999		X	X
Sudholz et al. (2021)	T_Sudholz2021		X	X
Garnet-only barometry Function "calculate_gt_only_press"				
Ryan et al. (1996)	P_Ryan1996	✓		X

## Other Garnet Functions

Garnet classification of Griffin et al. (2002)	"garnet_CARP_class_Griffin2002"
Cr-pyrope classification of Grütter et al. (2004)	"garnet_class_Grutter2004"
Ca-Cr classification of Cr-pyrope of Griffin et al. (2002)	"garnet_ca_cr_class_Griffin2002"
Y-Zr Classification of Cr-pyrope of Griffin et al. (2002)	"y_zr_classification_Griffin2002"
Ol Mg# from Cr-pyrope (Gaul et al. 2000)	"calculate_ol_mg"
Calculate Al <sub>2</sub> O <sub>3</sub> of whole-rock from Cr-pyrope (after O'Reilly et al. 2006)	"calculate_al2O3_whole_rock"