Guide to inferring Probability Model Ensembles (PMEs) for detrital zircon data and calculating Bayesian Population Correlation (BPC)

Table of Contents

1. Included files

First, download all the files by clicking 'Clone or Download' > 'Download ZIP' on Github. Then unzip the resulting folder. Included in this package are scripts necessary in order to infer PMEs, estimate BPC uncertainties, display calculated BPC values, display PME plots, and infer the shared proportions of two samples based on BPC values. In addition to being presented in their original form as MATLAB scripts, these files have been compiled into a standalone application. Installers for the application for Windows and Mac are also included. The main folder, which contains this document, contains a main menu .m file and accompanying .fig file:

'BPCmainmenu.m' – MATLAB script of the main menu for PME inference and BPC calculations, through which all included functionality can be accessed.

'BPCmainmenu.fig' – MATLAB file that contains information about the graphical user interface (GUI) for the main menu script.

The following subdirectories are also contained in this package:

'backend/' – contains scripts the user does not need to access directly. A short readme file in the folder discusses the function of each script. This folder also contains two third-party scripts, which are noted in the Dependencies section, below.

'sample_data/' – includes a set of sample data with which the scripts can be tested (see section Workflow, below).

The installers for the standalone applications (useful if the user lacks MATLAB or one of the required toolboxes—see Dependencies section below) are:

'BPCinstallerWin_web.exe' – this is the installer for Windows

'BPCinstallerMac_web.zip' – this is the installer for Mac, and must be manually unzipped before use

Note that these installers require an Internet connection and were compiled using the MATLAB and operating system versions listed in the Tested Hardware section below.

The package also contains a text readme file, a text license file for the set of scripts (excluding those that we did not author, which fall under pre-existing licenses included with the scripts), and this introductory document.

2. Installation versus execution from MATLAB

If the user lacks the required versions of MATLAB and the toolboxes used by the BPC scripts (see Dependencies section), then the BPC scripts can be installed as a standalone application. This requires running one of the two installers (*'BPCinstallerWin_web.exe'* for Windows machines, *'BPCinstallerMac_web.zip'* for Mac, which must be unzipped first) and clicking through the on-screen instructions. Note that *'BPCinstallerMac_web.zip'* must be unzipped manually before installation. Unzipping the file should produce a file called *'BPCinstallerMac_web.app'*, which will install the application if you double click on it. The installers were created using a built in function in

MATLAB, and were compiled using the MATLAB and OS versions listed under the Tested Hardware section. An Internet connection is required for this installation because the installer downloads and installs the MATLAB runtime environment. Note that these standalone applications operate independently of the .m files contained in this package, and the directory into which you install the application shouldn't affect its functionality. The sample data that we use in the example Workflow section below will not be installed using the installer. In order to work with this data, the user must make sure to download it in addition to the installer. The BPC application will prompt the user for the location of the data they wish to model, at which time, the user can refer the application to the sample data. Finally, we have observed the Mac error sound (a short "bonk") while the script is running, and do not understand why this is happening, but it appears not to affect the result.

Note that if you run the standalone application instead of the MATLAB files, there will likely be times when you start the application or issue a command and the application appears not to do anything for several seconds. After a short delay, you should see that your command has been carried out. Please only click buttons once on the GUI and then wait for the function to execute.

On windows machines, the parallel computing performed by the scripts may cause the objection of Windows firewall. On Mac machines, note that the actual application file will be located at '*installed_directory*/application/BPC', where '*installed_directory*' is specified by the user during installation.

3. Dependencies

Note: Dependencies were assessed using the MATLAB function matlab.codetools.requiredFilesAndProducts. The user can check which toolboxes they have installed in MATLAB by typing 'ver' into the command window.

The BPC scripts require the following MATLAB toolboxes (versions shown in parentheses):
> MATLAB (9.3)
> Optimization Toolbox (8.0)
> Statistics and Machine Learning Toolbox (11.2)
> Curve Fitting Toolbox (3.5.6)
> Parallel Computing Toolbox (6.11)
> Global Optimization Toolbox(3.4.3)

Two additional third-party scripts are required for our collection of scripts to function:

nearestSPD.m, which is copyright (c) 2013, John D'Errico, and is provided according to the license text contained in "nearestSPD_license.txt", distributed with our collection of scripts. *parfor_progressbar.m*, which is copyright (c) 2016, Daniel Terry, and is provided according to the license text contained in "parfor_progressbar_license.txt".

4. Workflow

The first step of the workflow is to start the standalone application, or run the *'BPCmainmenu_GUI.m'* script from MATLAB (ensuring that your MATLAB installation meets the version and toolbox requirements shown in the Dependencies section below). Upon doing so, you should see a simple window with buttons:
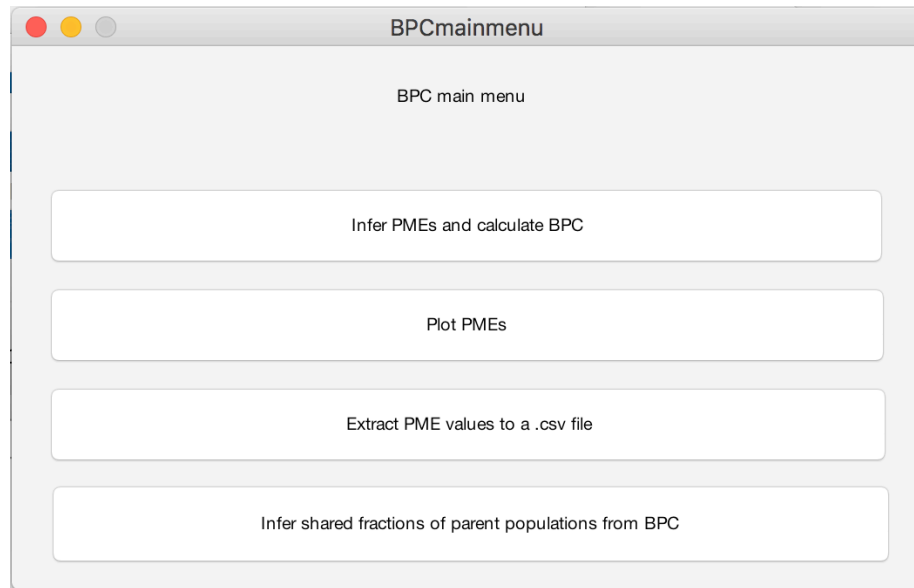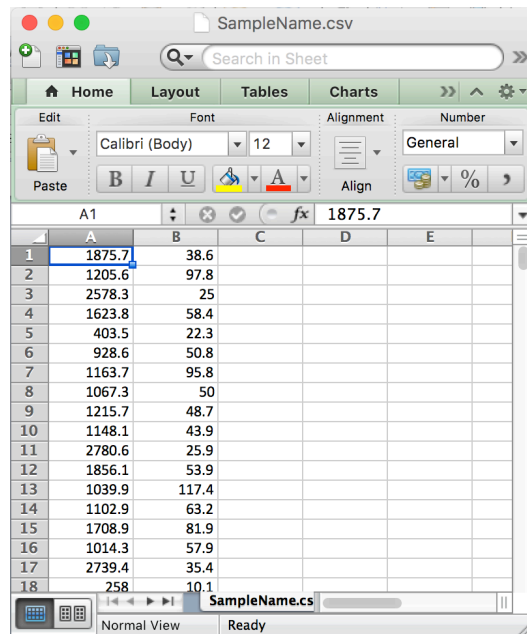

These four buttons allow access to the major functions included in this package by clicking the respective button. The remainder of this workflow includes 4 parts: A, B, C, and D, which correspond with the four buttons. Part A shows how to infer PMEs and calculate BPC. Part B shows how to plot PMEs. Part C shows how to extract probability values from the PMEs to a .csv file. Part D shows how to infer the shared proportions of two zircon age populations from their BPC value.

Note: For this demonstration, we run our collection of scripts on the 4 random subsamples (from data of Pullen et al., 2014, and Thomson et al., 2017), included with the scripts in the folder 'sample_data/'. Of these four

random subsamples, two each are selected from the datasets of Pullen et al. (2014) and Thomson et al. (2017), and two of the samples consist of 60 grain ages each whereas the other two samples consist of 300 grain ages. The filenames indicate which subsamples are which. If the scripts are run on these data, the results should resemble those found in this document.

A. Infer Probability Model Ensembles (PMEs), estimate BPC uncertainties, and display BPC values using BPConeclick_GUI.m.

For the samples you want to model, save the best ages of these samples, along with analytical uncertainties (1σ), in two-column .csv files, as follows:

The first column is the preferred measured age for each analysis in millions of years and the second column is the 1σ analytical uncertainty calculated for the preferred measured age, also in millions of years.

Ensure that the .csv files corresponding to all desired samples are located in a single folder, with no other .csv files. I suggest creating a new folder. Click the *"Infer PMEs and display BPC"* button from the main menu. You should see the following window appear:

Data folder

Select Data Folder

Age lower bound (Ma) `1`

Age upper bound (Ma) `4000`

Select the folder which contains the .csv files corresponding to sample ages and uncertainties. The correct format for these files is discussed in the introductory PDF. This "one-click" script will infer PMEs, estimate BPC uncertainties, and display BPC values for a set of samples. Input the upper and lower age bounds for modeling using the text boxes to the left. These should be set to 4000 and 1 respectively unless there is a compelling reason to do otherwise. Check the boxes for whether or not to account for analytical uncertainties on zircon ages in modeling (doing so increases processing time), whether to calculate BPC uncertainties, and whether to use quick mode, a non-Bayesian quick estimate of BPC with significant drawbacks (see Instruction document). The overwrite tickbox determines whether existing PMEs and BPC uncertainty files will be overwritten. Input the maximum number of models to retain for each PME (we recommend at least 10000). Input the number of cores to use for processing.

☐ Account for analytical uncertainties on ages

☐ Calculate BPC uncertainty

☐ Quick mode

☐ Overwrite PMEs and BPC uncertainty files

[   ] Number of models to retain per PME (leave blank to retain all).

[   ] Number of cores available for processing (leave blank to use all available).

Samples

Once the data folder is selected, this list box shows the order in which samples will be arranged for the BPC figure. To reorder samples, please type the indices of samples in the desired order below, separated by commas (e.g. '1, 4, 2, 3, 5'). You may omit samples that you do not want to include in the figure (e.g. '1, 4, 3, 5'). If the current order is okay, leave blank or type 'y'. To automatically order samples from lowest to highest mean BPC values when compared with other samples, type 'auto'.

Sample order

Calculate and display BPC

Finally, click calculate and display BPC to begin the process. See introductory document for notes on runtime and data structure.

BPC values:

	1	2
1		
2		
3		
4		

BPC uncertainties (1 sigma):

	1	2
1		
2		
3		
4		

Click the "*Select Data Folder*" button to select the folder where your .csv files are located. The path of this folder will then appear next to the button. Change the age bounds if desired, though bounds of 1 Ma and 4000 Ma should generally be used unless there is a strong reason to use a different set of bounds. *BPC values calculated using a given set of bounds are not directly comparable to BPC values calculated using a different set of bounds.* Several options now need to be specified using the checkboxes on the left. The first option is whether to account for analytical uncertainty in zircon ages when inferring PMEs and calculating BPC. If analytical uncertainties are not being accounted for, the data files must still be in the same format discussed above (two column, preferred ages in the first column, but the second column will not be read so theoretically could be all zeros, etc.). The second option is whether to calculate uncertainty on BPC values. The third option is whether to run the analysis in "quick mode", discussed in Section S2 of the text. Note that if quick mode is used, BPC uncertainties will not be calculated regardless of whether that box is checked because the uncertainties derived from quick mode are unreliable. Running quick mode, ignoring analytical uncertainty on zircon ages, and not calculating BPC uncertainties are all ways to save significant computation time to get a relatively accurate preview of BPC

values, but we strongly suggest that for final analyses, the uncertainties should be modeled and quick mode should not be used. Supplementary Section S2 discusses the mechanics and drawbacks of quick mode in detail. Finally, there is an option to overwrite existing PME and BPC files. If this is not selected, analyses will be skipped for samples for which these files already exist. **Note:** if you have already inferred PMEs using one set of options and wish to redo the analysis with different options, you *must* check the box to overwrite previous files. Enter the number of models you wish to retain for each sample, or leave this field blank to retain all models. We recommend using a value of at least 10,000. Retaining fewer models will not improve processing time, but it will decrease the amount of disk space the outputs from the scripts occupy. Enter the number of cores you desire to be used for the BPC analysis. If this textbox is left blank, all available cores will be used.

If a folder has been selected, the filenames of the sample .csv files will be shown in the listbox on the left side of the window. The BPC values, once calculated, will be shown in an NxN matrix, where N is the number of compared samples. The "*Sample order*" textbox allows you to change the order in which the samples will appear in this matrix. To specify the order, type the indices of the sample filenames (shown in the listbox) in the desired order, separated by commas (e.g. '1, 3, 4, 2, 5'). See the text in the window for further instructions. You can also leave "*Sample order*" blank or enter "y" to use the default order shown in the listbox, or you can type "auto" to automatically order the samples in terms of lowest to highest mean BPC value calculated with all other samples.

Click *"Calculate and display BPC"* in order to begin the process of inferring PMEs, estimating BPC uncertainty, and ultimately displaying BPC values. First, the script will infer PMEs for the samples in the referenced folder. During this time, a progress bar will appear that says "Inferring PMEs…". Next, BPC uncertainties will be estimated during which time a progress bar saying, "Estimating BPC uncertainties…", will appear. Both of these progress bars have cancel buttons that should work to halt the program. Inferring PMEs and calculating BPC uncertainties each take significant time, as discussed in the General Notes section, but *progress is saved* (unless 'Overwrite PMEs and BPC uncertainty files' has been ticked). If the script is interrupted, restart it in the same way as is described here, and the portions of the process that were already completed will not be run again. If a recalculation is desired, check the 'Overwrite PMEs and BPC uncertainty files' box, or manually delete the files created by the program. The directory structure of these saved files is described in Section 5.

While the script is running, MATLAB may appear not to be busy, causing confusion (see further discussion in the General Notes section). After the

required calculations are complete, you should see a color-coded table output to a new figure:
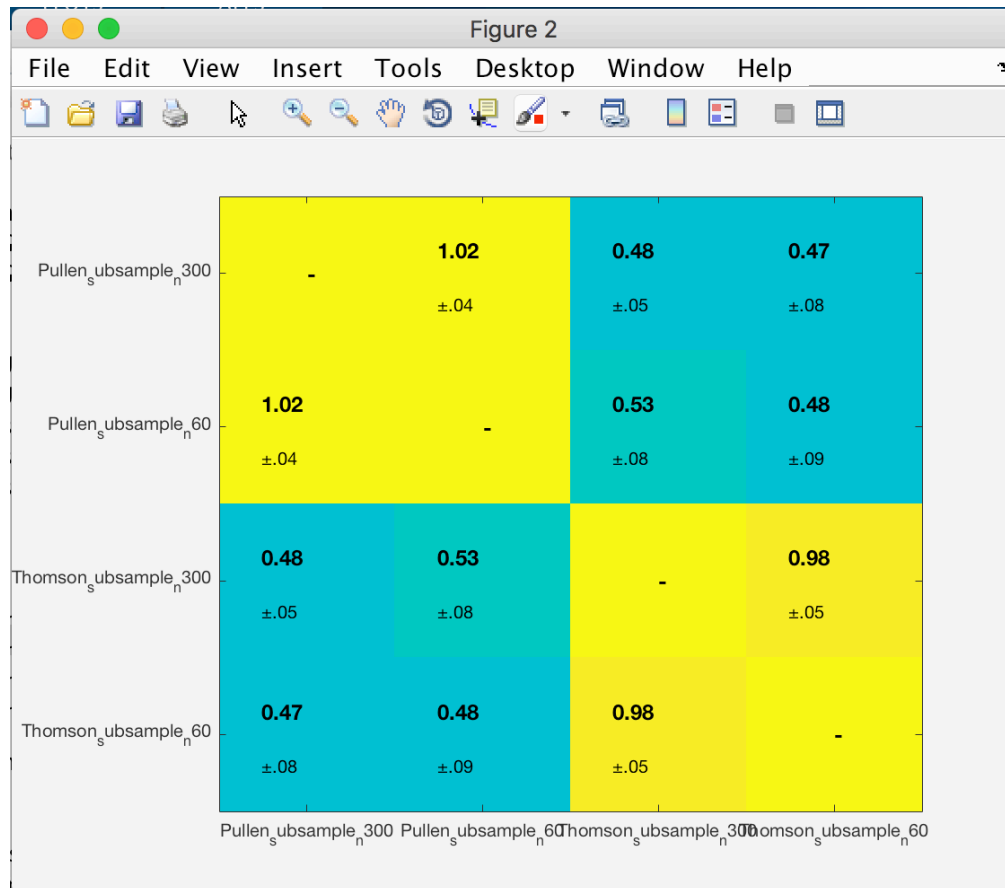
This figure shows the BPC value and uncertainty for each pair of compared samples. The colors illustrate the BPC value on the MATLAB parula colormap stretched from 0 to 1. In addition, the "*BPC value*" and "*BPC uncertainties (1 sigma)*" fields in the window are populated with values that can be copied to the clipboard. If the option to calculate BPC uncertainties was not checked, uncertainties will not appear in the figure shown above, and the *"BPC uncertainties (1 sigma)"* table will be populated with NaN (not a number) values.

To re-display BPC values after the analysis is complete, simply re-run BPConeclick_GUI.m by clicking on the *"Infer PMEs and display BPC"* button from the main menu (without selecting the option to overwrite existing files). Because completed analyses won't be re-done, the function will quickly display BPC values.

B. Displaying Probability Model Ensembles (PMEs) using PMEplot_GUI.m

Plotting a PME can be done only after PMEs have been inferred using
'BPConeclick_GUI.m' above. Click the *"Plot PMEs"* button on the main menu.
You should see the following window:

Again use "*Select Data Folder*" to select the folder where the sample .csv files
are stored. As with evalBPC_GUI.m, the folder path will be shown next to the
button and the samples contained in the folder will be shown in a listbox on
the left. Highlight the desired sample(s) in the listbox and specify options
using the text boxes and check boxes. Options that can be specified in
textboxes include the x domain over which modeling was conducted (this is
not just the desired bounds of the plot; use the same x min and x max values
as for 'makePME_GUI.m' and 'BPCunc_GUI.m'), and x and y resolution for the
plot, and the number of the figure for output. If more than one figure is
plotted, then subsequent figures will be plotted in sequentially numbered

plots. In addition, checkboxes allow you to specify whether to highlight the maximum likelihood probability model in the PME plot, whether to title the plot with the sample name, whether to use a linear or logarithmic probability scale, and whether to include a dotplot of measured ages in the figure. The dotplot appears in a thin band beneath the probability models, similar to plots shown in Pullen et al. (2014). The dotplot and maximum likelihood model can be plotted in their own figure windows, as well, which can be useful if further processing of the figures is intended in vector-based graphics software. The number of cores for the operation can also be specified. Once the desired options are specified, click "*Plot PME*". A progress bar should appear. Once reading and processing the data is complete, the resulting plot(s) will appear:
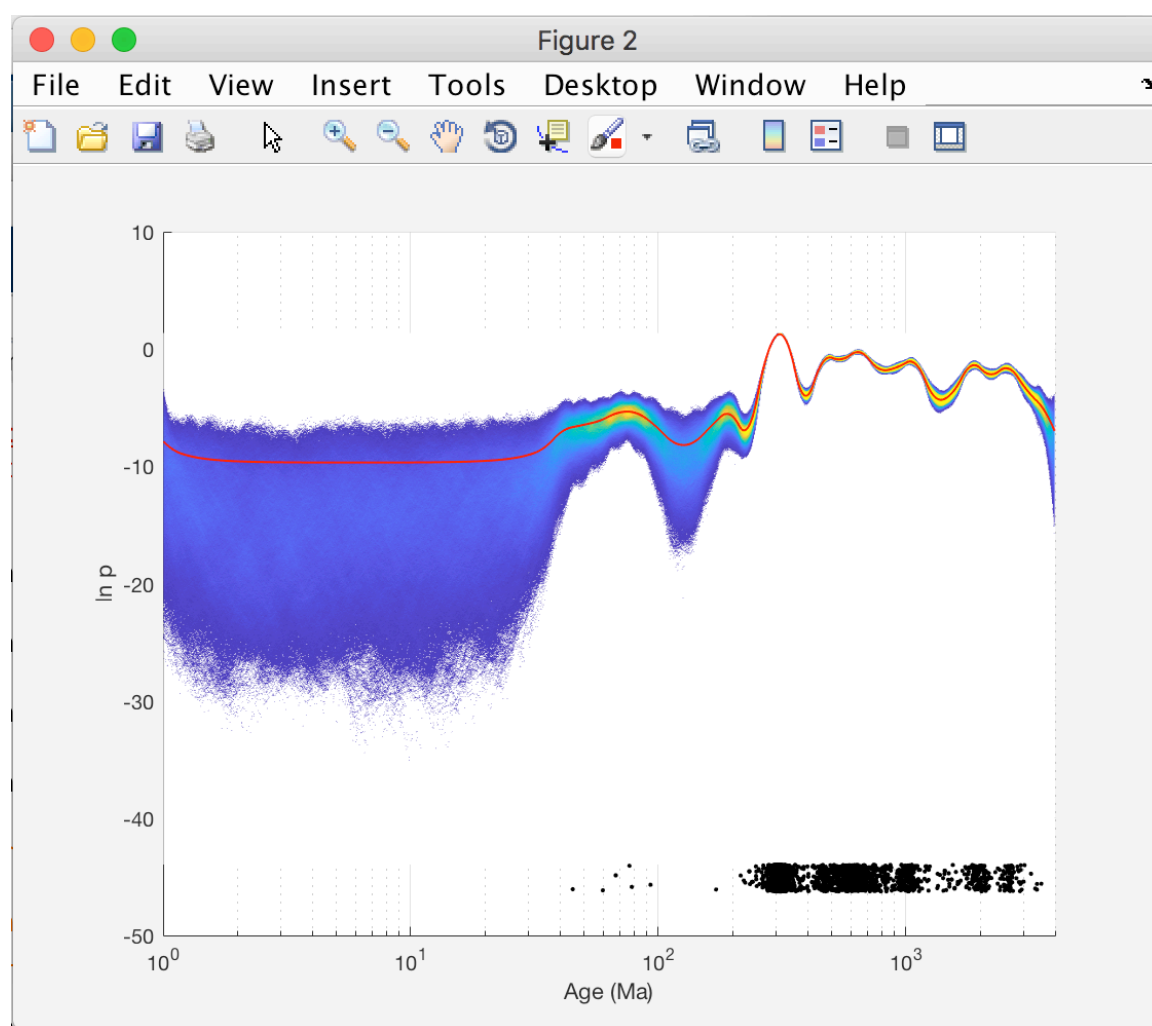
In this example, "Highlight maximum likelihood model" was selected, along with "Show dot plot of measured ages", and all other check boxes were left blank. The x and y resolutions were set to 1000. This figure shows a natural logarithmic age scale and probability scale. In this script, the area covered by

the probability models of the PME is discretized into cells in the x and y directions with the number of cells in one dimension equal to the resolution input into the GUI. Then, the cells that have probability models that pass through them are colored according to the number of models that pass through them, using the MATLAB parula colormap.

If more than one sample was selected in the listbox in the main window, the PME plots for each selected sample will appear sequentially.

C. Extract probability values from PMEs using PME2CSV_GUI.m

A PME consists of many probability density functions (PDFs), and their values can be queried at a given set of x values using PME2CSV_GUI.m. This is the way to obtain the function values of a PME for further processing. First, click *"Extract PME values to a .csv file"* in the main menu. You should see the following window:

First, click *"Select Data Folder"*, which will open a directory selection dialog. The selected directory should contain the .csv files that correspond to each modeled sample—this is the same data folder that BPConeclick_GUI.m uses. The listbox will then show the names of samples with .csv files saved in the data folder. Select the sample(s) for which you want to extract values, then choose options, including the x min and x max values—these must be the

values that you used for modeling the data. Two checkboxes control whether the output uses linear or logarithmic age and probability scales. Some notes on the use of logarithmic versus linear age scales are included in Section S4 of the text. Enter the maximum number of probability models from which you wish to extract values—a default value of 10000 is listed. Remove this default and leave the space blank to use all the probability models in a PME, but this may result in a very large output that takes a long time to write to disk. Optionally enter a number of cores to use for processing (leave blank to use all cores available). In the lower left corner, enter the number of points for which you want probability values returned. These points will be evenly distributed over the range of x values used for modeling on either a linear or logarithmic scale, depending on options chosen. A progress bar should appear which says *"Evaluating PDFs..."*, and a message box will appear when the process is complete.

The script creates a new subdirectory within the data folder you specified called 'PMEvalues/' and writes .csv files containing PDF values at the queried ages in that subdirectory. Output .csv files are named according to the sample name, followed by 'PMEvals.csv'. In these output .csv files, the first column lists the queried ages. Then, each successive column displays the probability density values of one PDF at each of the queried age values. Each column (except the first) contains values from a single PDF. **Note that if a log age scale was requested in the GUI**, the first column of the output file will contain ln age values (e.g. if modeling had been conducted in the age range 1-4000 Ma, the first column will contain values from 0, which equals ln 1, to ~8.3, which equals ln 4000).

D. Inferring the shared fractions of two populations from BPC values using BPC2frac_GUI.m

BPC values have a functional relationship to the shared fraction of two detrital zircon populations (the fraction of age peaks of each population that is shared with the other population), which can be derived analytically (see paper text). Thus, a BPC value can non-uniquely constrain the shared fractions of both populations. This calculation is facilitated by the BPC2frac_GUI.m script. First, click the *"Infer shared fractions of parent populations from BPC"* button in the main menu. The following window should appear:

Fill out the text box fields, including BPC value and uncertainty, along with the sample sizes of the compared samples. The BPC age and uncertainty come from the results of part A. In specific circumstances, the shared proportion of one sample may be assumed. For instance, if two samples are taken from two positions on the same river network such that the water and sediment that flow past the first sample location subsequently flow past the second sample location, then all the age peaks in the first sample can be assumed to be shared with the second sample, resulting in an f1 value of 1 (see text for discussion). If f1 can be assumed, then enter it as well. If no f1 value is entered, then the output will appear like this:
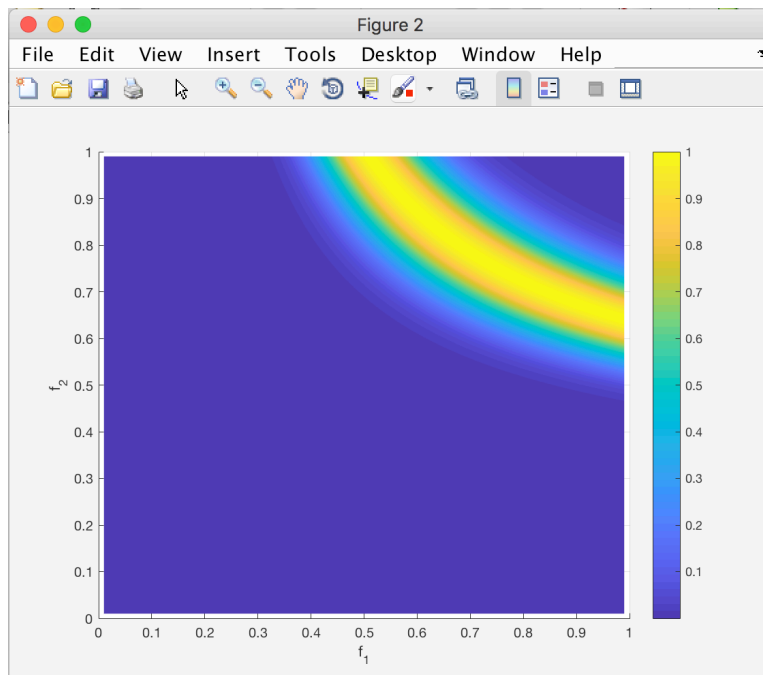
Here, colors indicate the relative likelihood of coordinate pairs of (f1, f2) values. These values are obtained by solving Eqn. C.7 in Appendix C in the text numerically for the given BPC value and uncertainty. This result was

generated for a BPC value of 0.75, uncertainty of 0.05, and sample sizes of 100 and 300.

If f1 can be assumed, the output will appear like this:
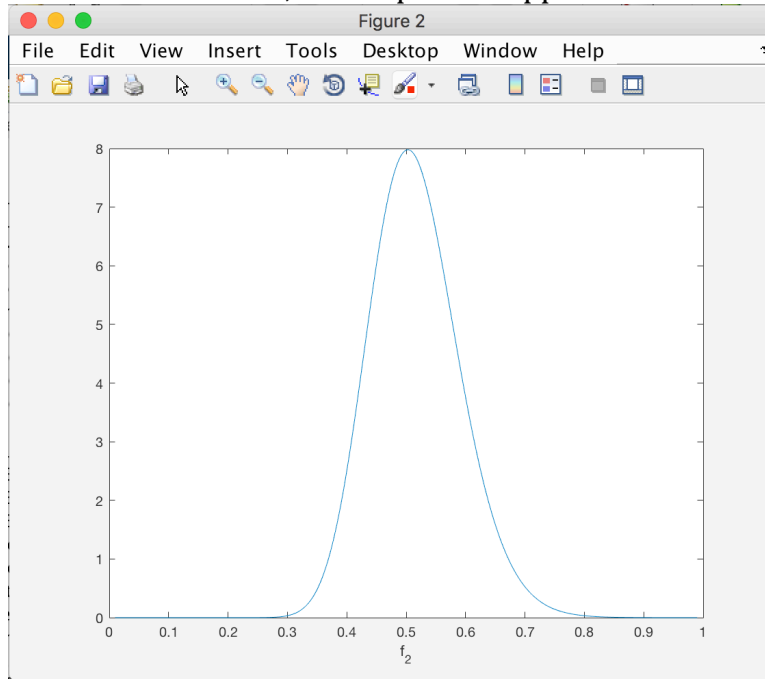
Here, the plot shows the likelihoods of different values of f2 for the given value of f1. This example plot was generated using the same parameters as above, plus an f1 value of 1. In addition, the mean and standard deviation of this distribution are output to the MATLAB Command Window.

5. File architecture

File architecture of scripts for PME inference: Upon running, the script for inferring PMEs generates a new directory within the selected data folder, called 'chains/'. Within this directory, two .csv files per sample are generated, plus two additional .csv files for each possible pairing of samples in the data folder. For a given sample or sample pair, one of the two files contains a PME (a Markov chain), and is named with the sample name followed by 'chain.csv'. Each row of the 'chain.csv' file corresponds to a different probability model that has been accepted into the PME. The columns store the values of the 50 model parameters of each of these probability models. The second of the two files per sample or sample pair contains a list of the log likelihood values of the PME inferred for that sample or sample pair. This second file is named with the sample name followed by 'logLk.csv'. The 'logLk.csv' file is a single column, where each row contains a

log likelihood value that corresponds with the model in the same row of the 'chain.csv' file.

The files resulting from the comparison of two samples are named as above except the names of the two samples are concatenated such that the filenames read "SampleName1_SampleName2chain.csv", where "SampleName1" and "SampleName2" are the filenames of the sample .csv files and "chain.csv" could also be "logLk.csv" or "log.txt".

File architecture of BPCunc.m and associated scripts for estimating BPC uncertainties: BPCunc.m creates an additional subdirectory in the folder you selected, called 'unc/'. In this folder, one additional .csv file is created for each sample along with one .csv file for each pair of samples. The files created for individual samples contain simulated age samples obtained by resampling the PME for each sample, along with the likelihood value of the maximum likelihood model for that simulated sample (see text for discussion). These files end with 'resample.csv' and each column corresponds to a single simulated sample. Column headers are the maximum likelihood values of each simulated sample. The files created for pairs of samples contain the likelihood values of the maximum likelihood model for pairs of simulated samples, and end with 'jointML.csv'. These files are a single row of maximum likelihood values.

Log files. The script that infers PMEs and calculates BPC create log files. This log can be found in the folder where the sample .csv files are located. The log records the settings for the run, and how long the run takes, and should be easily readable. Each line of the log file corresponds to a specific setting chosen on the BPConeclick interface, and successive runs will add onto the log document rather than replacing it.

There are additional log files created for each sample and each pair of samples while PMEs are being inferred and BPC uncertainties calculated, but these are unlikely to be easily readable by users. Log files are generated during each run of the script for PME inference and are stored in a 'chains/log/' directory within the data folder. The filenames of the log files match the filenames of the corresponding sample .csv files with "log.txt" appended. During BPC uncertainty calculation, an additional subdirectory of the data folder, 'unc/log/', is created, where a log file is stored for each sample or pair of samples for which BPC uncertainty is calculated.

6. General Notes, Common Problems, Clarifications

MATLAB may appear not to be busy even though the script is running: Once you start any of these scripts from the GUI (by clicking *"Generate PMEs"* or equivalent), the progress bar may appear under other MATLAB windows.

In addition, a blinking cursor will appear in the MATLAB Command Window, and MATLAB will not display the word "Busy", as it usually does for a script run from the command line, even though the script is running. The screenshot below shows the MATLAB screen while the script is running. **This behavior will occur with all scripts included in our package when run from GUIs.**
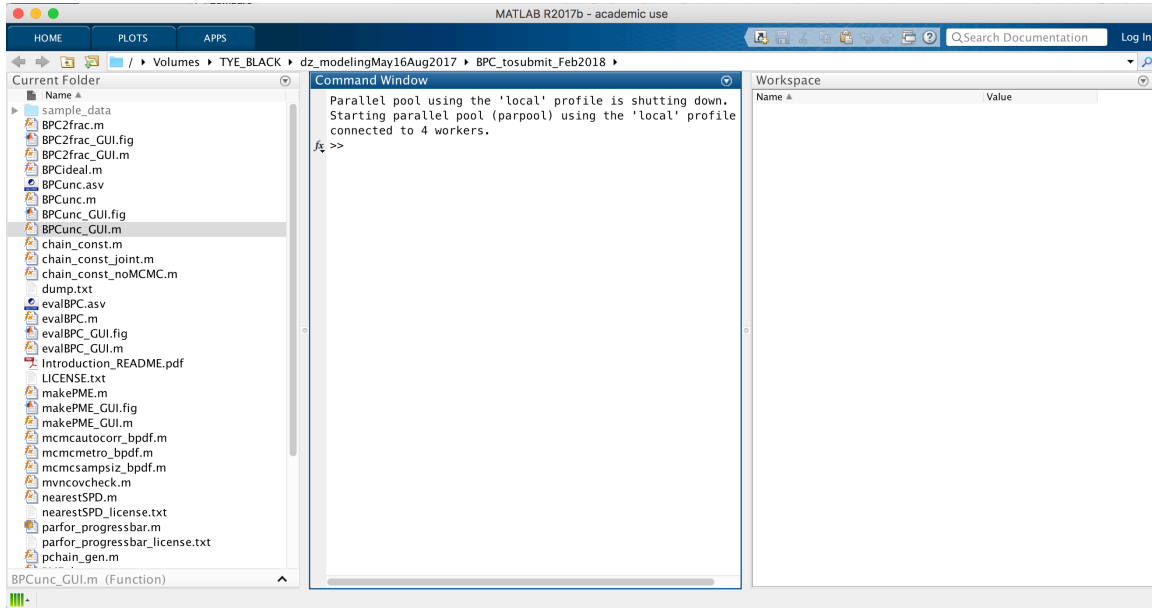
The standalone applications can be slow to respond: As noted above, when the standalone applications are started or a button is clicked, the response of the application may be delayed. The standalone application will often be far more delayed than the MATLAB script. When this occurs, please wait for the application to respond rather than reissuing a command, which will cause operations to be run more than once.

Expected time for script running: The scripts that infer PMEs and calculate BPC uncertainty will take a significant amount of time to run. These scripts compare every possible pair of samples in an input dataset. Mathematically, the number of possible pairs in a given dataset is given by (N choose 2), where N is the number of samples in a given dataset. makePME.m and BPCunc.m both take minutes to tens of minutes for each sample plus an equivalent time for each possible sample pair. This time is distributed over as many cores as are available for processing, such that the total processing time (in hours) can be estimated as

$$T = (0.2 - 0.5 \; hours) * \frac{N + (N \; choose \; 2)}{\# \; cores}$$

These times were calculated on a 2014 Macbook Pro with 2.2 GHz, 4-core processor and 16 GB RAM.

As noted above, changing the options on the scripts can have a significant effect on runtime. For instance, ignoring analytical uncertainties in grain ages should reduce processing time by a factor of ~2, and not calculating BPC uncertainty values should also reduce processing time by a similar factor. Using quick mode should reduce processing time by a factor of 10 or greater, but for final analyses of a particular dataset, we strongly recommend not using quickmode, calculating BPC uncertainties, and modeling analytical uncertainty.

Progress is saved when running the scripts. The scripts that infer PMEs and calculate BPC uncertainties run a Monte Carlo-based analysis on each sample and each possible pair of samples in the input dataset, which takes significant time. These scripts immediately save the results of their analysis on each sample or pair of samples as soon as the analysis is complete. The scripts look for these files when run and do not run duplicate analyses for samples or sample pairs that have already been completed. Thus, if one of these scripts is interrupted in the middle of execution, it can be restarted without losing the analyses that have already been completed. Effectively, your progress is saved as the script runs. Previous analyses can be overwritten by checking the overwrite option on the BPConeclick interface.

BPC values and estimated uncertainties may differ slightly from run to run. Inferring PMEs and estimating BPC uncertainties are Monte Carlo-based approaches, meaning that they rely upon random sampling and are liable to be slightly different from one run to another. In general, the differences observed in calculated BPC values should be significantly less than the 1σ estimated uncertainty on that value, indicating that this variability is within the limits of the resolution provided by the data. Estimated BPC uncertainties are likely to differ by ~10% (occasionally as high as 20%) of their value between runs. For typical BPC uncertainties for sample sizes n > 100, this results in possible variation of 0.01-0.02 from one run to another.

ĺĺExiting the scripts: The progress bars for the longer running scripts have cancel buttons that should halt execution if clicked. However, if operation does not stop, closing windows associated with the scripts/applications may not terminate these processes either. To quit the standalone applications prematurely, use your operating system's force quit/task manager option. To quit the MATLAB scripts prematurely, click on the command window to highlight it and then press Ctrl+C (Mac or Windows) to terminate the script.

7. Tested hardware
This software has been successfully used on:
2014 Macbook Pro, 2.2 GHz, 4-core processor and 16 GB RAM,
    MATLAB_R2017b, macOS 10.12.6

HP Windows 10 machine, 12-core processor and 64 GB RAM, MATLAB_R2016b.