UThwigl - an R package for closed- and open-system uranium-thorium dating

Anthony Dosseto[a], Ben Marwick[b]

[a] *Wollongong Isotope Geochronology Laboratory, School of Earth, Atmosphere & Life Sciences, University of Wollongong, Wollongong NSW Australia*
[b] *Department of Anthropology, University of Washington, Seattle, WA, USA*

Abstract

For several decades, uranium-thorium (U-Th) dating has allowed geochronologists to precisely date geological materials, providing unvaluable geochronological constraints on Quaternary processes. Open-system dating of bones and teeth has also provided ages of human and faunal remains of archaeological significance.

To facilitate access of open-system U-Th dating to the broad scientific community, here we provide an R package, named *UThwigl*, that implements the Diffusion-Adsorption-Decay model of Sambridge et al. (2012). Description of input and output parameters is given, as well as a guide for running the model. The package can be used three different ways: (i) as a web application, (ii) through a web browser with an internet connection, or (iii) in R (most efficiently with RStudio). Examples of application of the model are also provided, showing that it yields ages within error of previously published values.

Email addresses: `tonyd@uow.edu.au` (Anthony Dosseto), `bmarwick@uw.edu` (Ben Marwick)

Introduction

Uranium-thorium (U-Th) dating has revolutionised Quaternary science and archaeology. Dating uses the decay of ^{238}U into ^{230}Th, with ^{234}U and a few short-lived nuclides as intermediary products. It is based on the principle that the age of formation of a material can be dated as it incorporates U and no or little Th at the time of formation, so all the ^{230}Th in the sample comes from decay of ^{238}U. If detrital Th is included to the sample, a correction must be included to account for the fraction of ^{230}Th which is detrital and not derived from ^{238}U decay. Another requirement is that there is no gain or loss of ^{230}Th, ^{234}U or ^{238}U after formation of the material (*closed system*).

Closed-system U-Th dating has been successfully applied to a range of carbonates, from corals (Edwards, Gallup, and Cheng 2003) to speleothems (Richards and Dorale 2003). In corals and most speleothems, detrital correction is minimal; however, it can be significant when dating pedogenic carbonates, for instance (Ludwig and Paces 2002). In this case, detrital correction can be performed using the measured or assumed composition of the detrital fraction (e.g. K. Ludwig 2003). Alternatively, isochron techniques can be applied (Ludwig and Titterington 1994); the latter are beyond the scope of this article but IsoPlot is a commonly used software for isochron calculations and other geochronological applications (K. R. Ludwig 2003), now also available as a R package (Vermeesch 2018).

Closed-system conditions are seldom met in teeth and bones (although enamel can sometimes be quite impervious to isotope gain or loss). Thus, for teeth and bone, U-Th dating requires to take into account open system behaviour. The diffusion-adsorption-decay (DAD) model developed by Sambridge et al. (2012) was instrumental to implement successfully open-system U-Th dating. It allows for advective and diffusive transport of uranium and thorium isotopes, while include synchronous radioactive decay. The software implementation was written in Fortran and is available as a Java GUI (http://www.iearth.org.au/codes/iDaD/).

Open-system U-Th dating of teeth and bones, while challenging, has provided quantitative ages for human and faunal remains (Eggins et al. 2005; Grün et al. 2014; Sambridge, Grün, and Eggins 2012). Thus, this approach has significantly improved our understanding of human evolution (e.g. Dirks et al. 2017; Sutikna et al. 2016).

In this article, we propose a R package, *UThwigl*, which offers functions to perform closed-system, `csUTh()`, and open-system, `osUTh()`, U-Th age calculations. The former implements formulations given in Ludwig (2003) while the latter applies the Diffusion-Adsorption-Decay (DAD) model of Sambridge et al. (2012). The R package *IsoPlotR* provides a more extensive tool for closed-system U-Th dating (Vermeesch 2018), and *UThwigl* only includes closed-system U-Th age calculations for the sake of offering both closed- and open-system calculations.

The motivation for providing an R package is to increase the transparency, reproducibility, and flexibility of the analytical workflow for computing U-Th ages. For instance, with open-system dating, it is difficult to include the Java GUI in a fully scripted data analysis so the method for computing the DAD model is not highly transparent. This can obscure steps where key decisions are made that are important for others to see to verify the reliability of the analysis. Enabling a scripted workflow for computational analysis of geoscience data is important for improving the reproducibiltiy of results. Reproducibility refers the ability to recreate the results or retest the hypotheses leading to a scientific claim, either by rerunning the same code used by the original authors, or by writing new code. High rates of irreproducibility of research results have been estimated in several fields and disciplines (Medical Sciences 2015; Freedman, Cockburn, and Simcoe 2015; Institute 2013; Ioannidis 2005; Collaboration and others 2015; Camerer et al. 2018, 2016). Consequently, the transparency, openness, and reproducibility of results and methods are receiving increased attention, and the norms of research in many fields are changing (Nosek et al. 2015; Miguel et al. 2014; Marwick 2016).

There is strong interest in open, transparent, and reusable research in the geoscience community (Gil et al. 2016) and substantial progress toward open data has been made in the geosciences with the widespread use of data services of NASA, USGS, NOAA and community-built data portals such as OneGeology, EarthChem, RRUFF, PANGAEA, PaleoBioDB, and others (Kattge, Díaz, and Wirth 2014; Ma 2018). However, the use of open source software such as R (Pebesma, Nüst, and Bivand 2012), and sharing of scripted data analysis workflows with research publications is not yet widespread (Hutton et al. 2016). With this R package our goal is to make scripted and reproducible data analysis easy for open-system uranium-thorium dating. This will improve the transparency of geochronology research, and provide a more credible and robust foundation for scientific advancement (Hutton et al. 2016).

To enable re-use of our materials and improve reproducibility and transparency, all the results and visualisations in this paper can be reproduced using the RMarkdown vignette document included with the UThwigl package. We have archived these files at http://doi.org/10.17605/OSF.IO/D5P7S to ensure long-term accessibility. Our code is released under the MIT licence, our data as CC-0, and our figures as CC-BY, to enable maximum re-use (for more details, see Marwick 2016).

Methods

For U-Th dating, two types of analysis are possible: bulk or in-situ. For bulk analysis, a fraction of the samples is dissolved and the solution processed through ion exchange chromatography to separate U and Th (e.g. Luo et al. 1997). Each element is then analysed separately for their isotope ratios by mass spectrometry. For in-situ analysis, laser ablation is commonly used (Eggins et al. 2005). In this case, a laser with a spot size ranging from a few μm to several hundreds of μm produces an aerosol which is carried using a gas (helium or preferably a mixture of helium and nitrogen; Eggins, Kinsley, and Shelley (1998)). While laser ablation offers a better spatial resolution and is less time consuming than bulk analysis, the precision of the data is inferior because of the much smaller amount of material sampled.

Uranium and thorium isotope ratios are then analysed by multi-collector inductively-coupled plasma mass spectrometry (e.g. Luo et al. (1997); although bulk analysis can also be performed by thermal ionisation mass spectrometry). A plasma ionise U and Th atoms, their isotopes are separated through a magnetic field and each collected in a different collector. If using laser ablation, it is best to have two ion counters so ^{230}Th and ^{234}U can be collected simultaneously.

Closed-system dating

Pending closed-system behaviour can be assessed, it is possible to derive an age for each U-Th analysis. The closed-system function `csUTh()` requires that for each analysis to yield an age, (^{234}U/^{238}U), (^{230}Th/^{238}U) and (^{232}Th/^{238}U) activity ratios are measured (parentheses denote activity ratios throughout this article). The (^{232}Th/^{238}U) activity ratio is required for detrital correction (note it is needed to use `csUTh()` whether the detrital correction is performed or not).

Open-system dating

Data required for the DAD model are (^{230}Th/^{238}U) and (^{234}U/^{238}U) activity ratios collected along a transect perpendicular to the surface of the tooth or bone. Sampling for analysis can be done by micro-drilling or laser ablation.

The distance of each analysis location from the inner and outer surfaces of the bones, for instance, needs to be recorded. One surface is given a coordinate of 1 and the other one -1, thus coordinates of analyses take values in between (Figure 1).

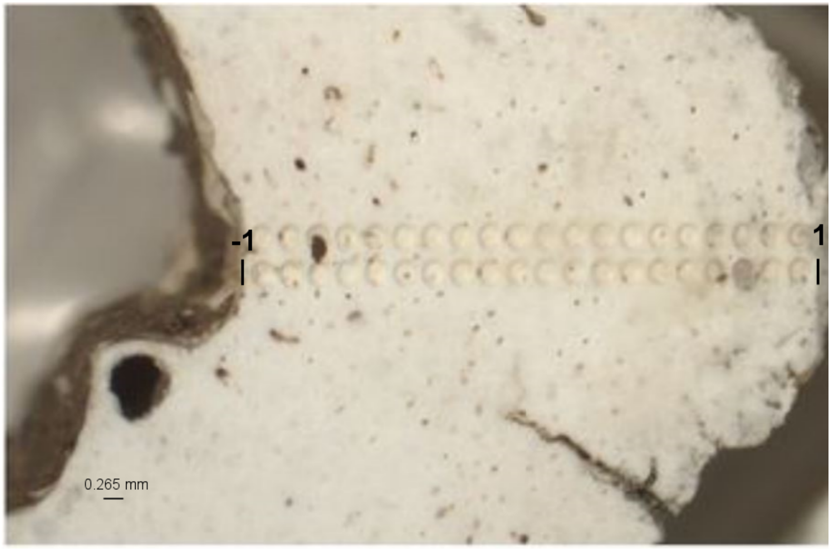

Figure 1: Modern human femur (132A/LB/27D/03) from Liang Bua, Flores, Indonesia. Two analysis transects can be seen. For a given transect, the outer and inner surface of the bone are given 1 and -1 reference coordinates, and the position of each analysis is calculated accordingly. Modified from Sutikna et al. (2016).

Working with the package

We provide three methods for using this package to suit different levels of familiarity with the R programming language. The simplest way to use the package is our web applications, online at https://ben-marwick.shinyapps.io/csUTh/ and https://ben-marwick.shinyapps.io/osUTh/ (Figure 2). Using the web application requires no familiarity with R. To use the web application we upload a CSV file, then click through a series of tabs to inspect the data, adjust the model parameters, run the model, and inspect the output. The interface is mouse-driven and requires no programming. In the web application we upload the data file on the *Load the data* tab, set parameters from the *Set model parameters* tab, run the model by clicking the button *Run Simulation* on the same tab, and observe the results on the *Visualise the model* and *Inspect the model* tabs. We can change the parameters and re-run the model by click the button *Run Simulation*. Once done, close the window.

The second way to use the package is with Binder, a browser-based instance of R and RStudio that includes our package ready to work with. Binder is a server technology that turns computational material, such as an R package, into interactive computational environments in the cloud. Using Binder requires a novice level of familiarity with R, for example to use the code in this paper and adapt it to work with a different CSV file. Because Binder provides a complete R environment, custom R code can be written during a Binder instance to further explore the model's output in the browser. These two methods, the

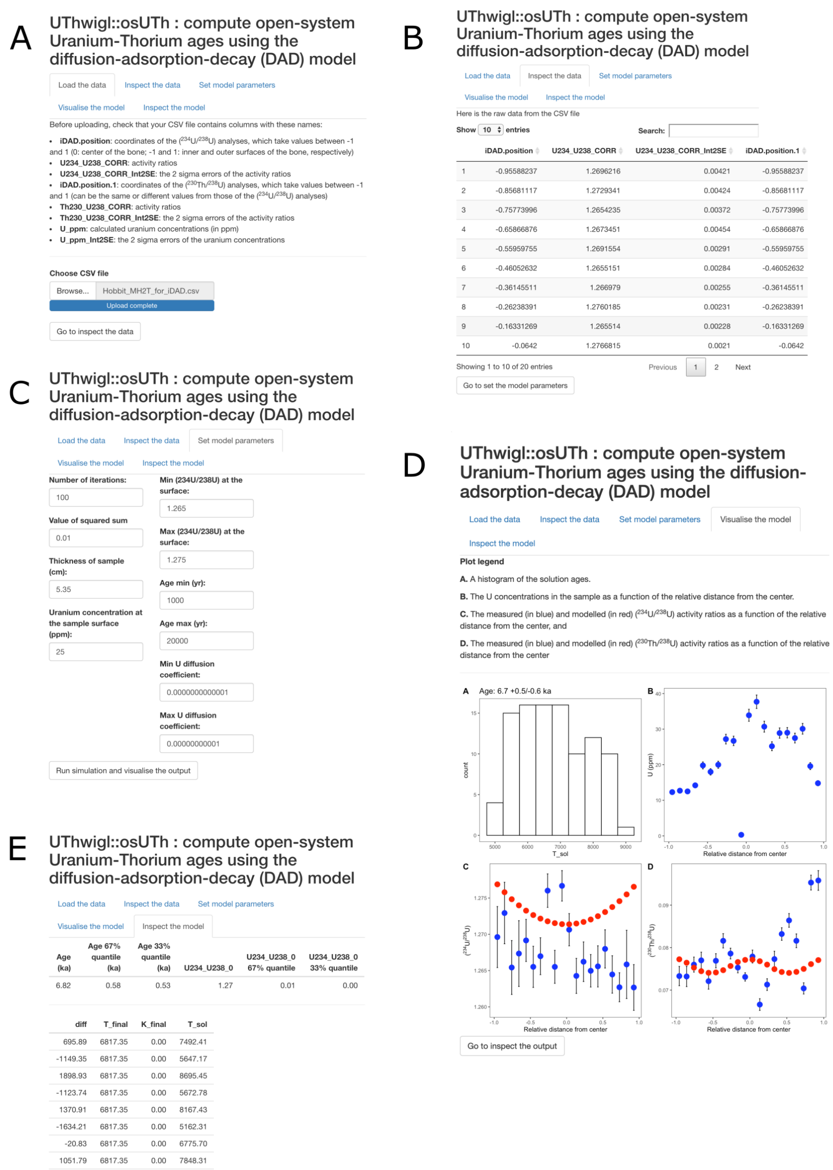

Figure 2: Screenshots of the web application for using the UThwigl package. A: Upload a CSV file of the data to model, B: Inspect a table of the uploaded data. C: Set the model parameters and run the model. D: Inspect visualisations of the model's output. E: Inspect and download the numeric output from the model.

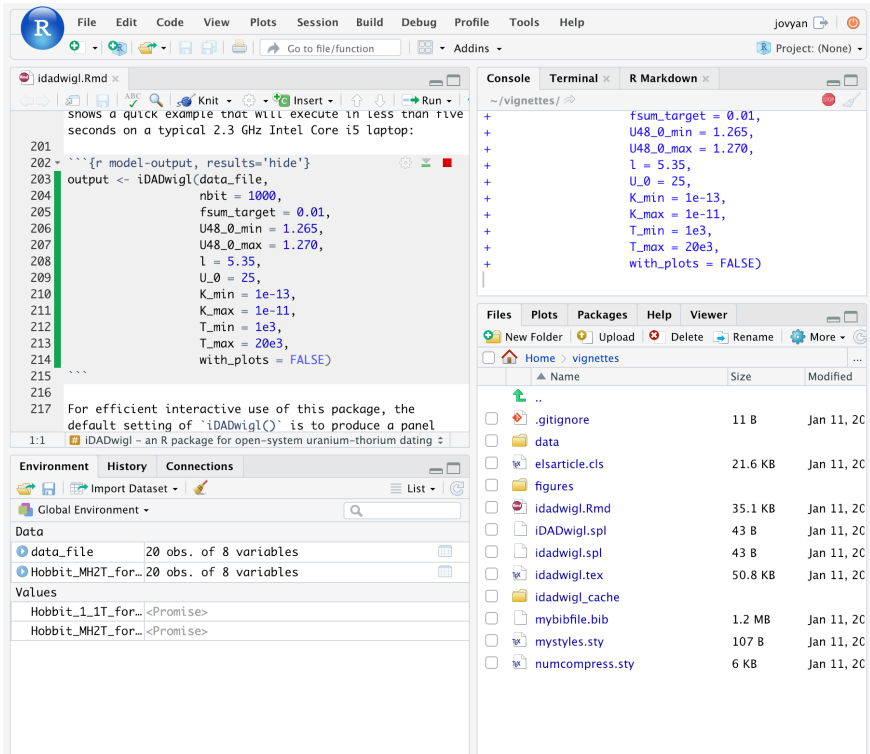

Figure 3: Screenshot of Binder running R and RStudio in a web browser window.

web application and Binder, do not require any software to be downloaded and installed on the user's computer, all computation occurs in the browser. The web application and Binder are suitable for getting a quick start on working with the package, but they require a connection to the internet, and they have limited memory and compute time available per instance.

The third method is to download and install the package locally to the user's computer, and work with it in the user's local installation of R and RStudio. This method requires some familiarity with R, but gives the most flexibility when working with the model because we are not limited by the memory and compute time of the cloud services. Our recommendation is to use Binder or a local installation of UThwigl because then the user can save an R script file that includes the name of the input file, the specific parameters used to generate the model output, and any downstream processing and visualisation. This script file and the CSV file can then be archived in a data repository to ensure long-term accessibility for other researchers. In the following sections we demonstrate the use of UThwigl with a local installation of R and RStudio.

Installing and attaching the package

First the user will need to download and install R, and we also recommend downloading and installing RStudio. To run the model, start `RStudio` and install the package from GitHub. There are many ways to do this, one simple method is shown in the line below. This only needs to be done once per computer.

```r
source("https://install-github.me/tonydoss/UThwigl")
```

For routine data analysis, R scripts need to contain the following line to attach the package to the current working environment. This line needs to be run at the start of each analysis:

```r
# attach the package
library(UThwigl)
```

Closed-system U-Th dating

Input data format

Our package provides the fuction `csUTh()` for closed-system U-Th dating. Data for this function needs to be in a data frame (a form of table in R) with the following column names:

- `Sample_ID`
- `U234_U238_CORR`
- `U234_U238_CORR_Int2SE`
- `Th230_U238_CORR`
- `Th230_U238_CORR_Int2SE`
- `Th232_U238_CORR`
- `Th232_U238_CORR_Int2SE`

To help with preparing data for input into our function, we have included an example of an input file, taken from Pan et al. (2018). Inspecting the included data sets will be helpful for understanding how to prepare new data to prepare for use with this package. After attaching the package, we can access the built-in datasets with the `data()` function, like this:

```
# access the data included in the UThwigl package
data("Pan2018")
```

This will make the built-in data available in the R environment to inspect and explore how to use the `csUTh()` function. To use new data with this package, the user needs to import a CSV or Excel file with the U-Th data into the R environment. This can be done using a generic function such as `read.csv` or `read_excel` from the `readxl` package (Wickham and Bryan 2018). Before reading in the data file, the user needs to set the working directory to the folder containing the data file. This can be done in RStudio using the menu item 'Session' > 'Set Working Directory' > 'To Source File Location'. Alternatively, the working directory can be defined interactively at the R prompt in the Console panel useing `setwd()`. However, we do not recommend including `setwd()` in script files because it is bad for reproducibility, since the path to one user's working directory will not exist on another user's computer.

To download the built-in data to the user's computer so it can be inspected and modified in a spreadsheet program, use `write.csv()`

```
# download the data included in the package
write.csv(Pan2018, "Pan2018.csv")
```

The code chunk below shows how to read one of the CSV files included in the package into the R environment. We assume that the user's working directory contains a directory called `data` and the CSV file is in this `data` directory, and so the data can be imported as follows:

```r
# read in one of the example CSV files included in the package
input_data_cs <-
  read.csv('data/Pan2018.csv')
```

Table 1 shows the data contained in the Pan2018.csv file included in the package.

Sample_ID	U234_U238_CORR	U234_U238_CORR_Int2SE	Th230_U238_CORR	Th230_U238_CORR_Int2SE	Th232_U238_CORR	Th232_U238_CORR_Int2SE
YP002A	1.150	0.005	0.794	0.007	0.010	0.00005
YP002B	1.120	0.004	0.788	0.006	0.004	0.00002
YP003-1_1	1.125	0.004	0.752	0.010	0.000	0.00001
YP003-1_2	1.113	0.007	0.761	0.011	0.000	0.00000
YP003-1_3	1.122	0.005	0.748	0.008	0.001	0.00001
YP003-1_4	1.122	0.005	0.726	0.007	0.001	0.00001
YP003-1_5	1.119	0.006	0.757	0.006	0.002	0.00001
YP002-1_1	1.129	0.006	0.722	0.008	0.001	0.00001
YP002-1_2	1.137	0.005	0.767	0.008	0.001	0.00001
YP002-1_3	1.118	0.008	0.739	0.009	0.002	0.00002
YP002-1_4	1.114	0.006	0.749	0.008	0.003	0.00003
YP002-1_5	1.105	0.007	0.764	0.011	0.003	0.00004

Table 1: Data contained in the example CSV file Pan2018.csv included in the package

The columns `Sample_ID`, `U234_U238_CORR`, `U234_U238_CORR_Int2SE`, `Th230_U238_CORR`, `Th230_U238_CORR_Int2SE`, `Th232_U238_CORR` and `Th232_U238_CORR_Int2SE` must be present in the input data frame with these exact names for the model to function. The `csUTh()` function will check if the input data frame has these columns, and will stop with an error message if it does not find these columns. The `names()` function can be used to update column names of a data frame to ensure they match the names that the model function requires. Alteratively the user can edit the column names in a spreadsheet program such as Microsoft Excel. The order of the columns in the data frame is not important.

Columns `U234_U238_CORR` and `U234_U238_CORR_Int2SE` are the $(^{234}\mathrm{U}/^{238}\mathrm{U})$ activity ratios and their 2σ errors. Columns `Th230_U238_CORR` and `Th230_U238_CORR_Int2SE` are the $(^{230}\mathrm{Th}/^{238}\mathrm{U})$ activity ratios and their 2σ errors. Columns `Th232_U238_CORR` and Th232_U238_CORR_Int2SE are the (^{232}Th/^{238}U) activity ratios and their 2σ errors.

Details of the input parameters of closed-system analysis

sample_name is the name of the sample to calculate closed-system ages for. The function will partially match by sample prefix. For example in Table 1 one sample is indicated by the Sample ID 'YP003'. If the user inputs 'YP003' for the sample_name, then this will match rows where the Sample ID is 'YP003-1', 'YP003-2', 'YP003-3', and so on.

nbitchoice is the number of iterations in the model. Recommended to have at least 100. detcorrectionchoice is a parameter for choosing whether or not to apply a detrital correction to the calculation.

R28det (0.8) and R28det_err (0.08) are the values for the (^{232}Th/^{238}U) activity ratio of the detritus and its standard error (default values in parentheses). Similarly, R08det (1) and R08det_err (0.05) are the values for the (^{230}Th/^{238}U) activity ratio of the detritus and its standard error, and R48det (1) and R48det_err (0.02) are the corresponding values for (^{234}U/^{238}U) activity ratio of the detritus.

How to run the model

Assuming that the package is attached with library(UThwigl), as shown above, and the data have been imported to the working environment as noted above, run csUTh(), specifying the input data frame and the input parameters as described above. The code block below shows a typical example that will execute in less than five seconds on a typical 2.3 GHz Intel Core i5 laptop:

```
# Solve for sample YP003 in-situ analyses
output_cs <-
  csUTh(
    input_data_cs,
    sample_name = 'YP003',
    nbitchoice = 100,
    detcorrectionchoice = TRUE,
    keepfiltereddata = FALSE,
    print_summary = TRUE,
    with_plots = TRUE,
    save_plots = FALSE,
    save_output = FALSE
  )
```

For efficient interactive use of this package, the default setting of csUTh() is to produce a panel plot as seen in Figure 4. The setting with_plots = FALSE prevents plots from being generated which is more useful when the function is part of a longer sequence of code. The function runs faster when not producing pots, which is helpful when replicating many runs. The setting save_output = TRUE will save a csv file to the current working directory so the output data can be used in other contexts. The csv file that is created when `save_output = TRUE` will be given a name that includes a date and time stamp so that the output of each time the function is run can be saved to a unique file.

When run on the R console, this function will print a confirmation that the input data frame has the required columns. If `print_summary` is set to `TRUE`, it will also the resulting mean age value of several analyses on a single sample, with an error reported as 2 Standard Error, for example:

```
All required columns are present in the input data
[1] "Mean age:  117.1 +/- 3.7  ka"
```

`print_summary` should be set to `FALSE` if ages computed are not for analyses of the same sample, since this mean age would be meaningless.

Inspecting and visualizing the models' output

The function returns a data frame with the age, error and summary output for each measurement, as shown in Table 2. This includes calculated ages (with or without detrital correction, depending how `detcorrectionchoice` was set), initial (^{234}U/^{238}U) activity ratios, along with their uncertainties.

Sample ID	Age (ka)	Age 2se	(234U/238U)i	Ratio 2se
YP003-1_1	116.931	0.3200	1.1740	0.0010
YP003-1_2	121.970	0.3960	1.1580	0.0010
YP003-1_3	116.313	0.2730	1.1700	0.0010
YP003-1_4	111.024	0.2290	1.1660	0.0010
YP003-1_5	119.348	0.2530	1.1680	0.0010

Table 2: Output produced by the csUTh function used with data from Pan et al. 2018

The plots produced by the `csUTh()` function are stored as list objects in the output from the function. We can show the plots by accessing the list like this:

```
output_cs$plots
```

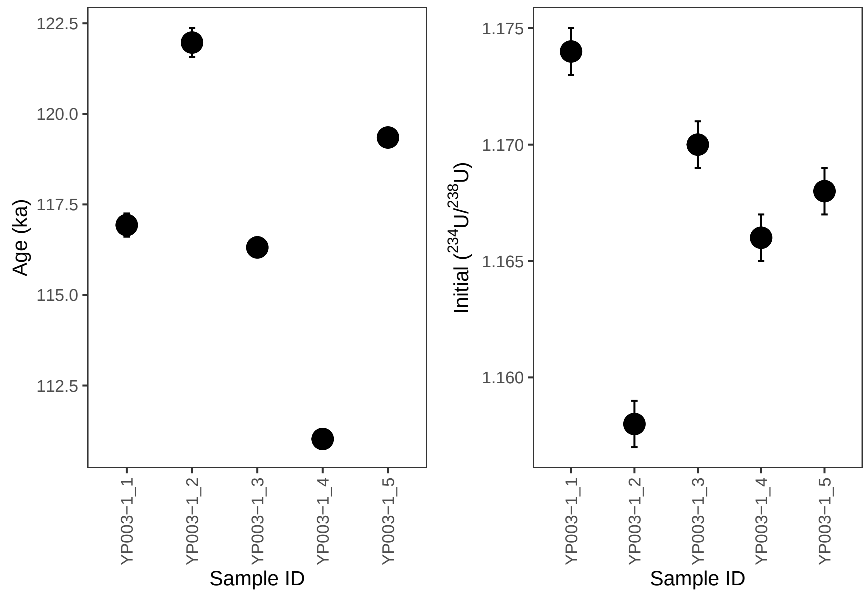

Figure 4: Example of the visualisations produced by the `csUTh()` function, using the demonstration run described above, and five in-situ analyses by laser ablation of coral sample YP003. A: closed-system ages and B: initial (^{234}U/^{238}U) activity ratios for each sample analysis

Open-system U-Th dating

Input data format

For open-system U-Th dating we provide the function `osUTh()`, which requires a data frame with the following column names:

- `iDAD.position`
- `U234_U238_CORR`
- `U234_U238_CORR_Int2SE`
- `iDAD.position.1`
- `Th230_U238_CORR`
- `Th230_U238_CORR_Int2SE`
- `U_ppm`
- `U_ppm_Int2SE`

We have included two examples of input files. After attaching the package, we can access the built-in datasets with the `data()` function, like this:

```
# access the data included in the UThwigl package
data("Hobbit_1_1T_for_iDAD")
data("Hobbit_MH2T_for_iDAD")
```

As above, these datasets can be downloaded from the package with `write.csv()`:

```r
# download the data included in the package
write.csv(Hobbit_1_1T_for_iDAD, "Hobbit_1_1T_for_iDAD.csv")
write.csv(Hobbit_MH2T_for_iDAD, "HHobbit_MH2T_for_iDADcsv")
```

The code chunk below shows how to read one of the CSV files included in the package into the R environment. As above, we assume that the user's working directory contains a directory called `data` and the CSV file is in this `data` directory, and so the data can be imported as follows:

```r
# read in one of the example CSV files included in the package
input_data_os <-
  read.csv('data/Hobbit_MH2T_for_iDAD.csv')
```

Table 3 shows the data contained in the `Hobbit_MH2T_for_iDAD.csv` file included in the package

iDAD.position	U234_U238_CORR	U234_U238_CORR_Int2SE	iDAD.position.1	Th230_U238_CORR	Th230_U238_CORR_Int2SE	U_ppm	U_ppm_Int2SE
-0.956	1.270	0.004	-0.956	0.073	0.002	12.3	0.6
-0.857	1.273	0.004	-0.857	0.073	0.002	12.7	0.6
-0.758	1.265	0.004	-0.758	0.076	0.002	12.5	0.6
-0.659	1.267	0.005	-0.659	0.077	0.002	14.2	0.7
-0.560	1.269	0.003	-0.560	0.072	0.002	19.8	1.0
-0.461	1.266	0.003	-0.461	0.077	0.002	18.0	0.9
-0.361	1.267	0.003	-0.361	0.082	0.002	20.0	1.0
-0.262	1.276	0.002	-0.262	0.079	0.001	27.2	1.4
-0.163	1.266	0.002	-0.163	0.075	0.001	26.7	1.3
-0.064	1.277	0.002	-0.064	0.073	0.001	0.3	0.0
0.035	1.271	0.002	0.035	0.078	0.002	33.9	1.7
0.134	1.264	0.002	0.134	0.067	0.001	37.7	1.9
0.233	1.266	0.003	0.233	0.071	0.001	30.7	1.5
0.332	1.265	0.002	0.332	0.077	0.002	25.2	1.3
0.431	1.266	0.003	0.431	0.083	0.001	28.9	1.4
0.530	1.268	0.003	0.530	0.086	0.002	29.0	1.4
0.629	1.264	0.003	0.629	0.082	0.002	27.5	1.4
0.728	1.263	0.002	0.728	0.070	0.001	30.1	1.5
0.827	1.266	0.005	0.827	0.095	0.002	19.6	1.0
0.926	1.263	0.003	0.926	0.096	0.002	14.8	0.7

Table 3: Data contained in the example CSV file Hobbit_MH2T_for_iDAD.csv included in the package

As for the closed-system function, the columns `iDAD.position`, `U234_U238_CORR`, `U234_U238_CORR_Int2SE`, `Th230_U238_CORR` and `Th230_U238_CORR_Int2SE` must be present in the input data frame with these exact names for the model to function. The `osUTh()` function will check if the input data frame has these columns, and will stop with an error message if it does not find these columns.

The `iDAD.position` column corresponds to the coordinates of the $(^{234}\mathrm{U}/^{238}\mathrm{U})$ analyses, which as indicated above take values between -1 and 1 (Figure 1). The second `iDAD.position.1` column is used if the coordinates of the $(^{230}\mathrm{Th}/^{238}\mathrm{U})$

analyses are different from those of the (^{234}U/^{238}U) analyses.

Columns `U_ppm` and `U_ppm_Int2SE` are the calculated uranium concentrations (in ppm) and their 2σ errors. Uranium concentrations are not necessary for the model but needed to display the U concentration profile in a figure.

Details of the input parameters of open-system analysis

Our key function, `osUTh()` has several arguments that need to be set before meaningful results can be obtained:

`nbit` is the number of iterations. For the first run, set to 1.

`fsum_target` is the sum of the squared differences between the calculated and observed activity ratios. Give it a low value to start with (e.g. 0.01). If script takes too long, try a higher value for `fsum_target`.

`U48_0_min` and `U48_0_max` are the minimum and maximum values allowed for the (^{234}U/^{238}U) activity ratio at the surface of the sample. Since (^{234}U/^{238}U) does not vary greatly over the time period generally studied, the values measured near the surface of the sample can be used as a guide. These values can be adjusted if the model fit to the data is not optimal. For Hobbit_1-1T they are taken to be 1.360 and 1.375, and for Hobbit_MH2T, 1.265 and 1.270, respectively.

`l` is the thickness of the sample in centimeters. For Hobbit_1-1T it is 3.5 cm, for Hobbit_MH2T it is 5.35 cm

`U_0` is the uranium concentration at the surface in ppm. This value does not significantly affect the model results and values from analyses near either surface of the sample can be used as a guide. For Hobbit_1-1T it is taken to be 15 ppm; for Hobbit_MH2T, 25 ppm.

`K_min` and `K_max` are the minimum and maximum values allowed for the uranium diffusion coefficient (in cm^2/s). Values between 10^{-13} and 10^{-11} cm^2/s are generally appropriate.

`T_min` and `T_max` are the minimum and maximum values for the age of the specimen (yr). If there is no estimated knowledge of the sample age, the range of values can be 1,000 to 500,000 yr and adjusted later. For Hobbit_1-1T, in the final model run, they are taken to be 50,000 and 100,000 yr, and for Hobbit_MH2T, 1,000 and 20,000 yr, respectively.

After setting the `U480` maximum and minimum values, run the function and adjust these min and max values by looking at the calculated `U48_0_final`, `K_final`, and `T_final`. Adjust `T_min` and `T_max` using first estimates of the age. As we iterate, increase the `nbit` value to reduce the error.

How to run the model

Attach the package as shown above and then run `osUTh()`, specifying the input data frame and the input parameters as described above. The code block below shows a quick example that will execute in less than five seconds on a typical 2.3 GHz Intel Core i5 laptop:

```r
output_os <- osUTh(input_data_os,
                   nbit = 1000,
                   fsum_target = 0.01,
                   U48_0_min = 1.265,
                   U48_0_max = 1.270,
                   l = 5.35,
                   U_0 = 25,
                   K_min = 1e-13,
                   K_max = 1e-11,
                   T_min = 1e3,
                   T_max = 20e3,
                   with_plots = TRUE,
                   save_plots = FALSE,
                   save_output = FALSE)
```

The default setting of `osUTh()` is to produce a panel plot as seen in Figure 5(Figure 5. The setting `with_plots = FALSE` prevents plots from being generated which is more useful when the function is part of a longer sequence of code. The function runs faster when not producing pots, which is helpful when replicating many runs.

Similar to the `csUTh()` function, when `osUTh()` is run on the R console, it will print a confirmation that the input data frame has the required columns, and print the resulting age value with an error reported as the 67% and 33% quantiles, for example:

```
All required columns are present in the input data
[1] "Age: 7 +0.6/-0.7 ka"
```

The model computes a Monte Carlo simulation where age of the sample, U diffusion coefficient and (^{234}U/^{238}U) ratio at the surface of the sample are taken randomly within the range of values allowed. Results are only kept if the calculated sum of the squared differences between the calculated and observed activity ratios is less than the value set in `fsum_target`. If this is the case, the calculated ratios and the set of solutions for age of the sample, U diffusion coefficient and (^{234}U/^{238}U) ratio at the surface of the sample are saved. The model stops once the number of sets of solutions reaches `nbit`.

The final calculated age `T_final` (in yr), U diffusion coefficient `K_final` (in cm^2/s) and (^{234}U/^{238}U) ratio at the surface of the sample `U48_0_final` are the set of solutions where the solution age is the closest to the median age of the population of solutions. The uncertainty on each output paramter is calculated as the 67% and 33% quantiles of the population of solution sets.

In a typical analysis the user will explore the model fit by first running the model with a single iteration `nbit` and a small value for `fsum_target`, and then changing the range of allowed values for the (^{234}U/^{238}U) ratio at the surface and the age of the sample. Once the user has obtained a satisfying fit (by visual inspection of the produced figures), they would increase `nbit` to a higher value (e.g. 1000) and run the model one last time. We provide an example of a typical analysis analysis below.

Inspecting the models' output

`T_final`, `K_final` and `U48_0_final` are included in the model's output, along with their uncertainties. The function also includes a one-row data frame summarising the age:

Age (ka)	Age 67% quantile (ka)	Age 33% quantile (ka)	U234_U238_0	U234_U238_0 67% quantile	U234_U238_0 33% quantile
7.08	0.58	0.65	1.2689	0.0011	0.0039

Table 4: Summary table of the computed age and error values

The last item in the output is a copy of the input data with two additional columns, the calculated activity ratios, ($^{234}U/^{238}U$) and ($^{230}Th/^{238}U$), for each measurement location on the sample.

Visualising the models' output

`osUTh()` returns several figures useful for visualisation of the model results along with the data:

1. a histogram of the solution ages (Figure 5 A)
2. the U concentrations in the sample as a function of the relative distance from the center (Figure 5 B)
3. the measured (in blue) and modelled (in red) ($^{234}U/^{238}U$) activity ratios as a function of the relative distance from the center (Figure 5 C), and
4. the measured (in blue) and modelled (in red) ($^{230}Th/^{238}U$) activity ratios as a function of the relative distance from the center (Figure 5 D).

We can show the plots produced by `osUTh()` by accessing the list like this:

```
output_os$plots
```

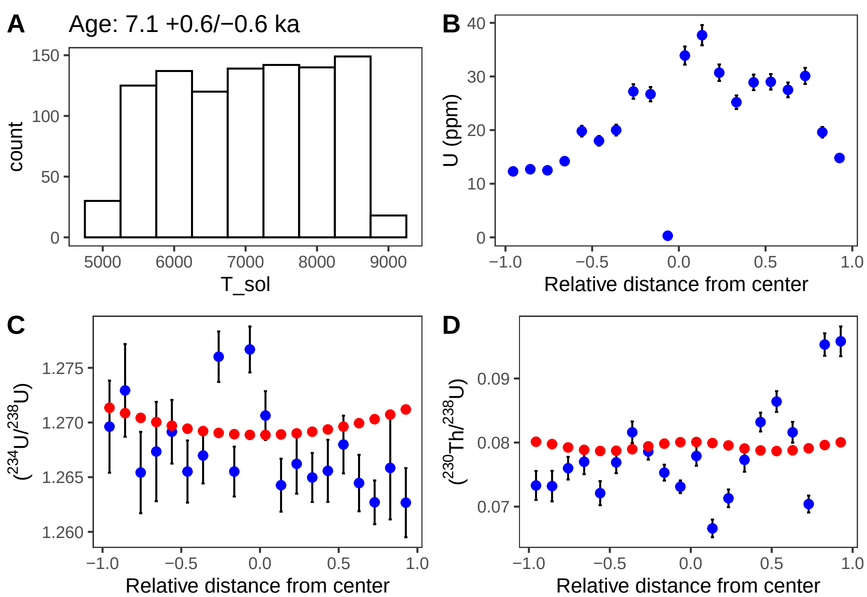

Figure 5: Example of the visualisations produced by the `osUTh()` function, using the demonstration run described above. A: Histogram of the solution ages, B: Uranium concentration profile for transect 2 of modern human femur 132A/LB/27D/03. C: Calculated (red) and observed (blue) ($^{234}U/^{238}U$) activity ratios for transect 2 of modern human femur 132A/LB/27D/03. D: Calculated (red) and observed (blue) ($^{230}Th/^{238}U$) activity ratios for transect 2 of modern human femur 132A/LB/27D/03.

iDAD.position	U234__U238__CORR	U234__U238__CORR__Int2SE	iDAD.position.1	Th230__U238__CORR	Th230__U238__CORR__Int2SE	U__ppm	U__ppm__Int2SE	U234__U238__CALC	Th230__U238__CALC
-0.956	1.270	0.004	-0.956	0.073	0.002	12.3	0.6	1.271	0.080
-0.857	1.273	0.004	-0.857	0.073	0.002	12.7	0.6	1.271	0.080
-0.758	1.265	0.004	-0.758	0.076	0.002	12.5	0.6	1.270	0.079
-0.659	1.267	0.005	-0.659	0.077	0.002	14.2	0.7	1.270	0.079
-0.560	1.269	0.003	-0.560	0.072	0.002	19.8	1.0	1.270	0.079
-0.461	1.266	0.003	-0.461	0.077	0.002	18.0	0.9	1.269	0.079
-0.361	1.267	0.003	-0.361	0.082	0.002	20.0	1.0	1.269	0.079
-0.262	1.276	0.002	-0.262	0.079	0.001	27.2	1.4	1.269	0.079
-0.163	1.266	0.002	-0.163	0.075	0.001	26.7	1.3	1.269	0.080
-0.064	1.277	0.002	-0.064	0.073	0.001	0.3	0.0	1.269	0.080
0.035	1.271	0.002	0.035	0.078	0.002	33.9	1.7	1.269	0.080
0.134	1.264	0.002	0.134	0.067	0.001	37.7	1.9	1.269	0.080
0.233	1.266	0.003	0.233	0.071	0.001	30.7	1.5	1.269	0.080
0.332	1.265	0.002	0.332	0.077	0.002	25.2	1.3	1.269	0.079
0.431	1.266	0.003	0.431	0.083	0.001	28.9	1.4	1.269	0.079
0.530	1.268	0.003	0.530	0.086	0.002	29.0	1.4	1.270	0.079
0.629	1.264	0.003	0.629	0.082	0.002	27.5	1.4	1.270	0.079
0.728	1.263	0.002	0.728	0.070	0.001	30.1	1.5	1.270	0.079
0.827	1.266	0.005	0.827	0.095	0.002	19.6	1.0	1.271	0.080
0.926	1.263	0.003	0.926	0.096	0.002	14.8	0.7	1.271	0.080

Table 5: Example of output table including the input data described above, and two new columns

Case studies

Closed-system dating - Case study from Pan et al. 2018

The package includes sample data from Marine Isotope Stage 5 corals from Pan et al. (2018) (Table 1). Two *Plesiastrea versipora* coral samples were analysed: YP002 and YP003. The first two rows in Table 1 are bulk analyses while the rest are in-situ analyses produced by laser ablation (hence the lower precision compared to the first two row). In Pan et al. (2018), closed-system ages were calculated using IsoPlot 4.15 (K. R. Ludwig 2003). For bulk analyses, Pan et al. (2018) reported detrital-corrected ages of 121.4 ± 2.4 ka and 127.3 ± 2.1 ka for YP002A and YP002B, respectively. For in-situ analyses, Pan et al. (2018) reported mean detrital-corrected ages of five analyses for each sample: 117.5 ± 4.5 ka for YP002 and 115.0 ± 5.4 ka for YP003.

Here we solve the closed-system model for all samples by simply entering 'YP' against `sample name` since all analyses in the table contain these two characters in their `Sample_ID` column. `print_summary` is set to `FALSE` since we are solving for different samples and a mean age would have no significance.

```
# Solve for all samples
output_cs_all <-
  csUTh(
    input_data_cs,
    sample_name = 'YP',
    nbitchoice = 100,
    detcorrectionchoice = TRUE,
    keepfiltereddata = FALSE,
    print_summary = FALSE,
    with_plots = TRUE,
    save_plots = FALSE
  )
```

We obtain detrital-corrected ages of 123.1 ± 0.3 ka and 128 ± 0.2 ka for bulk analyses of YP002 and YP003, respectively. This is within error of values reported in Pan et al. (2018).

In-situ analyses for YP003-1 were solved above and yielded a mean detrital-corrected age for the five analyses of 117.1 ± 3.7 ka, within error of the value reported in Pan et al. (2018). We can solve also for in-situ analyses of YP002-1 by setting `sample_name` to 'YP002-1' and `print_summary` to `TRUE`:

```
# Solve for YP002 in-situ analyses
output_cs_YP002insitu <-
  csUTh(
    input_data_cs,
    sample_name = 'YP002-1',
    nbitchoice = 100,
    detcorrectionchoice = TRUE,
```

```
    keepfiltereddata = FALSE,
    print_summary = TRUE,
    with_plots = TRUE,
    save_plots = FALSE
  )
```

We obtain a mean detrital-corrected age for the five analyses of 116.9 ± 5.4 ka, also within error of the value reported in Pan et al. (2018).

Open-system dating - Case study of two ages from Sutikna et al. 2016

The package includes two sample data sets derived from Sutikna et al. (2016) : "Hobbit_MH2T_for_iDAD.csv" is data from transect 2 for modern human femur 132A/LB/27D/03 (shown above in Table 3). "Hobbit_1-1T_for_iDAD.csv" is data from transect 1 for *Homo floresiensis* ulna LB1/52 (Table 6). For the latter, six analyses were removed from the set as in Sutikna et al. (2016).

iDAD.position	U234_U238_CORR	U234_U238_CORR_Int2SE	iDAD.position.1	Th230_U238_CORR	Th230_U238_CORR_Int2SE	U_ppm	U_ppm_Int2SE
-0.143	1.369	0.002	-0.143	0.699	0.006	32.0	1.6
0.165	1.370	0.002	0.165	0.733	0.008	41.1	2.1
0.319	1.364	0.002	0.319	0.672	0.006	35.8	1.8
0.472	1.362	0.003	0.472	0.636	0.006	27.6	1.4
0.626	1.365	0.003	0.626	0.641	0.006	31.0	1.6
0.780	1.374	0.003	0.780	0.712	0.005	27.9	1.4

Table 6: Data contained in the example CSV file Hobbit_11T_for_iDAD.csv included in the package

Age of the modern human remains from Sutikna et al. 2016

For transect 2 of 132A/LB/27D/03, Sutikna et al. (2016) reported an age of 7.4 ± 0.5 ka (thousand years before 2014). With UThwigl, we first run the model with `nbit` = 1, `fsum_target` = 0.05, `U48_0_min` and `U48_0_max` = 1.25 and 1.3, respectively, `l` = 5.35 cm, `U_0` = 25 ppm, `K_min` and `K_max` = 10^{-13} and 10^{-11} cm^2/s, respectively, `T_min` and `T_max` = 10^3 and $500x10^3$ yr, respectively. `U48_0_min` and `U48_0_max` are determined by considering the

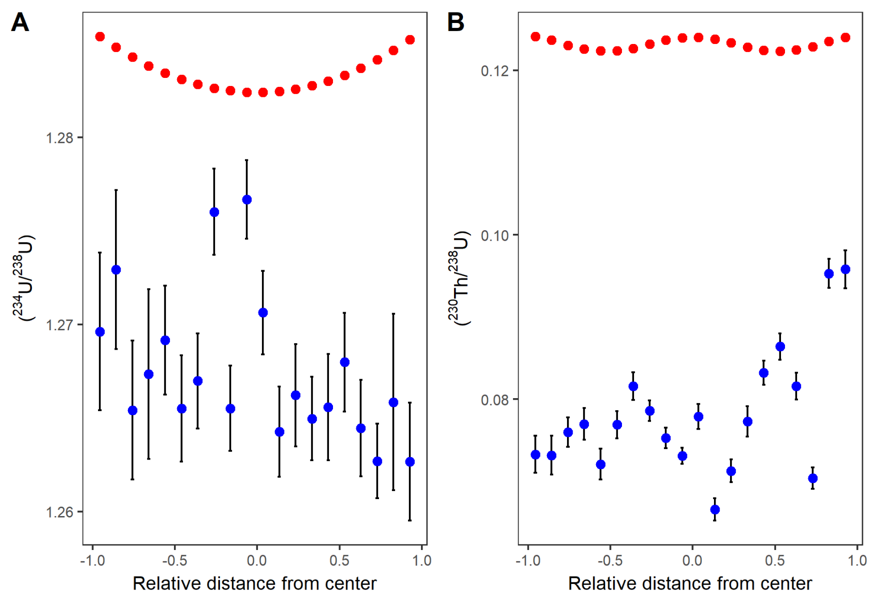

Figure 6: Results from the model's first run with the modern human femur. A: Calculated (red) and observed (blue) (^{234}U/^{238}U) activity ratios for transect 2 of modern human femur 132A/LB/27D/03. B: Calculated (red) and observed (blue) (^{230}Th/^{238}U) activity ratios for transect 2 of modern human femur 132A/LB/27D/03.

measured (^{234}U/^{238}U) values near the surfaces of the sample. `T_min` and `T_max` values were chosen such that no a priori knowledge of the age biases the results.

With this first run, we obtain an age of 11 ka. There is no calculated error on the age since there is only one iteration. In this case, we can see that the calculated (^{234}U/^{238}U) and (^{230}Th/^{238}U) ratios are too high (Figure 7). For the (^{234}U/^{238}U), it makes sense to thus use a lower value for `U48_0_max`. Calculated (^{230}Th/^{238}U) ratios are too high compared to observed values suggest that the calculated age is too old (since this ratio increases with age). Thus, we should adjust `T_max` accordingly.

Since the first run suggests a Holocene age for the sample, the measured (^{234}U/^{238}U) at the surfaces must be similar to the calculated values, thus the chosen values for the range above. Once `U48_0_min`, `U48_0_min`, `T_min` and `T_max` parameters have been adjusted, the model is run again. `fsum_target` can also be decreased to 0.01 in order to get a better fit and error, but it is at the expense of computing time. This operation is repeated until a satisfying fit is obtained (by visual inspection of the figures). Finally, the moden is run once more, increasing the number of iterations to 1000 (or more). Following this method, we obtain an age of 6.2 +1.2/-1.3 ka (Figure 7). Note (^{234}U/^{238}U) and (^{230}Th/^{238}U) are still too high and low, respectively, so `U48_0_min`, `U48_0_min`, `T_min` and `T_max` parameters should be adjusted and the model run again.

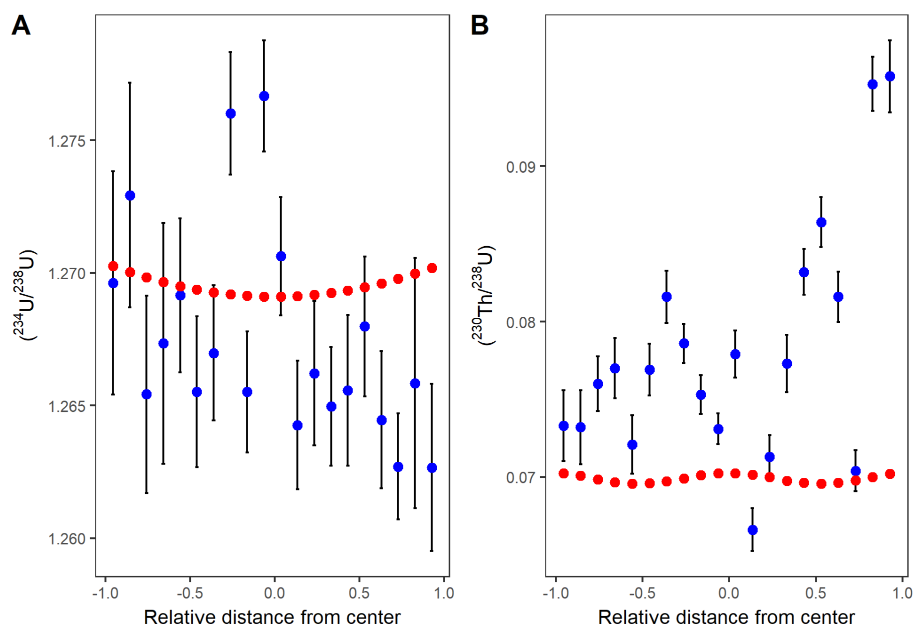

Figure 7: Results from the model's second run with the modern human femur. A: Calculated (red) and observed (blue) ($^{234}U/^{238}U$) activity ratios for transect 2 of modern human femur 132A/LB/27D/03. B: Calculated (red) and observed (blue) ($^{230}Th/^{238}U$) activity ratios for transect 2 of modern human femur 132A/LB/27D/03.

Age of the Homo floresiensis *remains from Sutikna et al. 2016*

For transect 1 of LB1/52, Sutikna et al. (2016) reported an age of 79.0 ± 3.7 ka. With osUth, using data in the file `Hobbit_1-1T_for_iDAD.csv` provided in the package, and following the same method as above, we obtain an age of 75.4 +1.0/-0.9 ka (Figure 8).

Note that results and errors will vary slightly for each run since populations are solution sets are randomly generated.

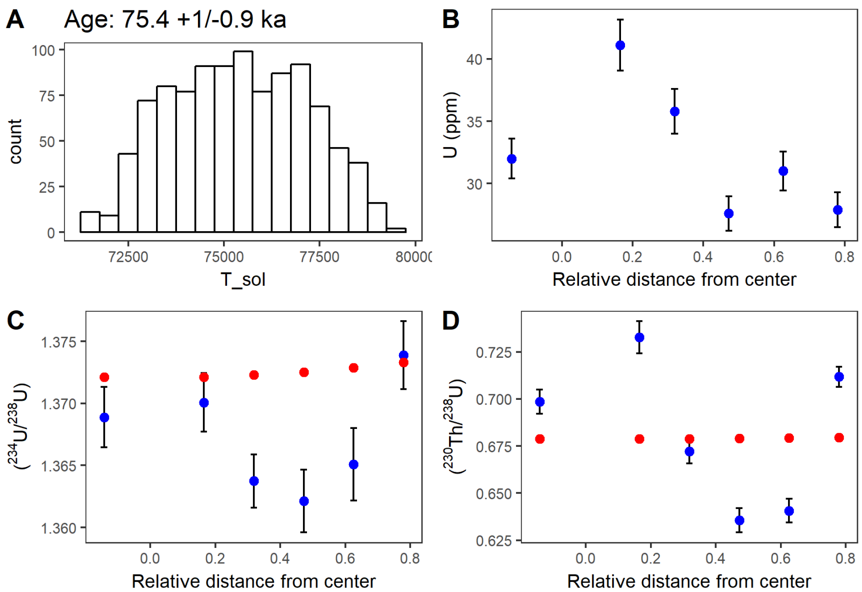

Figure 8: Results from running the model with *Homo floresiensis* ulna LB1/52 data from Sutikna et al. (2016). A: Histogram of the solution ages, B: Uranium concentration profile for transect 1 of *Homo floresiensis* ulna LB1/52. C: Calculated (red) and observed (blue) ($^{234}U/^{238}U$) activity ratios for transect 1 of *Homo floresiensis* ulna LB1/52. D: Calculated (red) and observed (blue) ($^{230}Th/^{238}U$) activity ratios for transect 1 of *Homo floresiensis* ulna LB1/52.

Conclusions

In this paper we have described `UThwigl`, an open source R package for computation of open-system U-Th ages. This helps to enable transparency, reproducibility, and flexibility of the analytical workflow for computing U-Th ages. The examples above show that results from our model are within error of previously published ages.

References

Camerer, Colin F., Anna Dreber, Eskil Forsell, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, et al. 2016. "Evaluating Replicability of Laboratory Experiments in Economics." *Science* 351 (6280): 1433–6. https://doi.org/10.1126/science.aaf0918.

Camerer, Colin F, Anna Dreber, Felix Holzmeister, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, et al. 2018. "Evaluating the Replicability of Social Science Experiments in Nature and Science Between 2010 and 2015." *Nature Human Behaviour* 2 (9): 637.

Collaboration, Open Science, and others. 2015. "Estimating the Reproducibility of Psychological Science." *Science* 349 (6251): aac4716.

Dirks, P. H., E. M. Roberts, H. Hilbert-Wolf, J. D. Kramers, J. Hawks, A. Dosseto, M. Duval, et al. 2017. "The Age of Homo Naledi and Associated Sediments in the Rising Star Cave, South Africa." Journal Article. *Elife* 6. https://doi.org/10.7554/eLife.24231.

Edwards, RL, CD Gallup, and H Cheng. 2003. "Uranium-Series Dating of Marine and Lacustrine Carbonates." Journal Article. *Reviews in Mineralogy and Geochemistry* 52 (1): 363–405.

Eggins, S. M., L. P. J. Kinsley, and J. M. G. Shelley. 1998. "Deposition and Element Fractionation Processes During Atmospheric Pressure Laser Sampling for Analysis by Icp-Ms." Journal Article. *Applied Surface Science* 127–129 (0): 278–86. https://doi.org/http://dx.doi.org/10.1016/S0169-4332(97)00643-0.

Eggins, Stephen M., Rainer Grün, Malcolm T. McCulloch, Alistair W. G. Pike, John Chappell, Les Kinsley, Graham Mortimer, et al. 2005. "In Situ U-Series Dating by Laser-Ablation Multi-Collector Icpms: New Prospects for Quaternary Geochronology." Journal Article. *Quaternary Science Reviews* 24 (23-24): 2523–38. https://doi.org/10.1016/j.quascirev.2005.07.006.

Freedman, Leonard P, Iain M Cockburn, and Timothy S Simcoe. 2015. "The Economics of Reproducibility in Preclinical Research." *PLoS Biology* 13 (6): e1002165.

Gil, Yolanda, Cédric H. David, Ibrahim Demir, Bakinam T. Essawy, Robinson W. Fulweiler, Jonathan L. Goodall, Leif Karlstrom, et al. 2016. "Toward the Geoscience Paper of the Future: Best Practices for Documenting and Sharing Research from Data to Software to Provenance." *Earth and Space Science* 3 (10): 388–415. https://doi.org/10.1002/2015EA000136.

Grün, Rainer, Stephen Eggins, Leslie Kinsley, Hannah Moseley, and Malcolm Sambridge. 2014. "Laser Ablation U-Series Analysis of Fossil Bones and Teeth." Journal Article. *Palaeogeography, Palaeoclimatology, Palaeoecology* 416: 150–67. https://doi.org/http://dx.doi.org/10.1016/j.palaeo.2014.07.023.

Hutton, Christopher, Thorsten Wagener, Jim Freer, Dawei Han, Chris Duffy, and Berit Arheimer. 2016. "Most Computational Hydrology Is Not Reproducible, so Is It Really Science?" *Water Resources Research* 52 (10): 7548–55. https://doi.org/10.1002/2016WR019285.

Institute, Global Biological Standards. 2013. "The Case for Standards in Life Science Research: Seizing Opportunities at a Time of Critical Need."

Ioannidis, John PA. 2005. "Why Most Published Research Findings Are False." *PLoS Medicine* 2 (8): e124.

Kattge, Jens, Sandra Díaz, and Christian Wirth. 2014. "Of Carrots and Sticks." *Nature Geoscience* 7 (October): 778.

Ludwig, KR. 2003. "Mathematical–Statistical Treatment of Data and Errors for 230Th/U Geochronology." Journal Article. *Reviews in Mineralogy and Geochemistry* 52 (1): 631–56.

Ludwig, K. R. 2003. *User's Manual for Isoplot 3.00*. Book. Berkeley, CA, USA: Berkeley Geochronology Center.

Ludwig, K. R., and J. B. Paces. 2002. "Uranium-Series Dating of Pedogenic Silica and Carbonate, Crater Flat, Nevada." Journal Article. *Geochimica et Cosmochimica Acta* 66 (3): 487–506. https://doi.org/Doi:%2010.1016/s0016-7037(01)00786-4.

Ludwig, K. R., and D. M. Titterington. 1994. "Calculation of 230ThU Isochrons, Ages, and Errors." Journal Article. *Geochimica et Cosmochimica Acta* 58 (22): 5031–42. https://doi.org/http://dx.doi.org/10.1016/0016-7037(94)90229-1.

Luo, Xiaozhong, Mark Rehkämper, Der-Chuen Lee, and Alex N. Halliday. 1997. "High Precision 230Th/232Th and 234U/238U Measurements Using Energy-Filtered Icp Magnetic Sector Multiple Collector Mass Spectrometry." Journal Article. *International Journal of Mass Spectrometry and Ion Processes* 171 (1–3): 105–17. https://doi.org/http://dx.doi.org/10.1016/S0168-1176(97)00136-5.

Ma, Xiaogang. 2018. "Data Science for Geoscience: Leveraging Mathematical Geosciences with Semantics and Open Data." In *Handbook of Mathematical Geosciences*, 687–702. Springer.

Marwick, Ben. 2016. "Computational Reproducibility in Archaeological Research: Basic Principles and a Case Study of Their Implementation." *Journal of Archaeological Method and Theory*, 1–27. https://doi.org/10.1007/s10816-015-9272-9.

Medical Sciences, Academy of. 2015. "Reproducibility and Reliability of Biomedical Research: Improving Research Practice." In *Symposium Report*.

Miguel, Edward, Colin Camerer, Katherine Casey, Joshua Cohen, Kevin M Esterling, Alan Gerber, Rachel Glennerster, et al. 2014. "Promoting Transparency in Social Science Research." *Science* 343 (6166): 30–31.

Nosek, Brian A, George Alter, George C Banks, Denny Borsboom, Sara D Bowman, Steven J Breckler, Stuart Buck, et al. 2015. "Promoting an Open Research Culture." *Science* 348 (6242): 1422–5.

Pan, Tsun-You, Colin V. Murray-Wallace, Anthony Dosseto, and Robert P. Bourman. 2018. "The Last Interglacial (Mis 5e) Sea Level Highstand from a Tectonically Stable Far-Field Setting, Yorke Peninsula, Southern Australia." Journal Article. *Marine Geology* 398: 126–36. https://doi.org/10.1016/j.margeo.2018.01.012.

Pebesma, Edzer, Daniel Nüst, and Roger Bivand. 2012. "The R Software Environment in Reproducible Geoscientific Research." *Eos, Transactions American Geophysical Union* 93 (16): 163–63. https://doi.org/10.1029/2012EO160003.

Richards, David A, and Jeffrey A Dorale. 2003. "Uranium-Series Chronology and Environmental Applications of Speleothems." Journal Article. *Reviews in Mineralogy and Geochemistry* 52 (1): 407–60.

Sambridge, M., R. Grün, and S. Eggins. 2012. "U-Series Dating of Bone in an Open System: The Diffusion-Adsorption-Decay Model." Journal Article. *Quaternary Geochronology*. http://www.scopus.com/inward/record.url?eid=2-s2.0-84857958926&partnerID=40&md5=01e7f1b7d3554da26b247125425ff496.

Sutikna, Thomas, Matthew W. Tocheri, Michael J. Morwood, E. Wahyu Saptomo, Jatmiko, Rokus Due Awe, Sri Wasisto, et al. 2016. "Revised Stratigraphy and Chronology for Homo Floresiensis at Liang Bua in Indonesia." Journal Article. *Nature* 532 (7599): 366–69. https://doi.org/10.1038/nature17179.

Vermeesch, Pieter. 2018. "IsoplotR: A Free and Open Toolbox for Geochronology." *Geoscience Frontiers* 9 (5): 1479–93.

Wickham, Hadley, and Jennifer Bryan. 2018. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.

Colophon

This report was generated on 2020-10-29 10:41:37 using the following computational environment and dependencies:

```
# which R packages and versions?
devtools::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.0.2 (2020-06-22)
##  os       Windows 10 x64
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_Australia.1252
##  ctype    English_Australia.1252
##  tz       Australia/Sydney
##  date     2020-10-29
##
## - Packages -------------------------------------------------------------------
##  package     * version date       lib source
##  assertthat    0.2.1   2019-03-21 [1] CRAN (R 4.0.2)
##  backports     1.1.7   2020-05-13 [1] CRAN (R 4.0.0)
##  bookdown      0.21    2020-10-13 [1] CRAN (R 4.0.3)
##  callr         3.4.4   2020-09-07 [1] CRAN (R 4.0.2)
##  cli           2.0.2   2020-02-28 [1] CRAN (R 4.0.2)
##  colorspace    1.4-1   2019-03-18 [1] CRAN (R 4.0.2)
##  cowplot       1.1.0   2020-09-08 [1] CRAN (R 4.0.3)
##  crayon        1.3.4   2017-09-16 [1] CRAN (R 4.0.2)
##  desc          1.2.0   2018-05-01 [1] CRAN (R 4.0.2)
##  deSolve       1.28    2020-03-08 [1] CRAN (R 4.0.3)
##  devtools      2.3.2   2020-09-18 [1] CRAN (R 4.0.3)
##  digest        0.6.25  2020-02-23 [1] CRAN (R 4.0.2)
##  dplyr       * 1.0.2   2020-08-18 [1] CRAN (R 4.0.3)
##  ellipsis      0.3.1   2020-05-15 [1] CRAN (R 4.0.2)
##  evaluate      0.14    2019-05-28 [1] CRAN (R 4.0.2)
##  fansi         0.4.1   2020-01-08 [1] CRAN (R 4.0.2)
##  farver        2.0.3   2020-01-16 [1] CRAN (R 4.0.2)
##  fs            1.5.0   2020-07-31 [1] CRAN (R 4.0.2)
##  generics      0.0.2   2018-11-29 [1] CRAN (R 4.0.2)
##  ggplot2     * 3.3.2   2020-06-19 [1] CRAN (R 4.0.3)
##  glue          1.4.1   2020-05-13 [1] CRAN (R 4.0.2)
##  gtable        0.3.0   2019-03-25 [1] CRAN (R 4.0.2)
##  htmltools     0.5.0   2020-06-16 [1] CRAN (R 4.0.2)
##  knitr       * 1.30    2020-09-22 [1] CRAN (R 4.0.3)
##  labeling      0.3     2014-08-23 [1] CRAN (R 4.0.0)
```

```
## lifecycle      0.2.0   2020-03-06 [1] CRAN (R 4.0.2)
## magrittr       1.5     2014-11-22 [1] CRAN (R 4.0.2)
## memoise        1.1.0   2017-04-21 [1] CRAN (R 4.0.2)
## munsell        0.5.0   2018-06-12 [1] CRAN (R 4.0.2)
## pillar         1.4.6   2020-07-10 [1] CRAN (R 4.0.2)
## pkgbuild       1.1.0   2020-07-13 [1] CRAN (R 4.0.2)
## pkgconfig      2.0.3   2019-09-22 [1] CRAN (R 4.0.2)
## pkgload        1.1.0   2020-05-29 [1] CRAN (R 4.0.2)
## prettyunits    1.1.1   2020-01-24 [1] CRAN (R 4.0.2)
## processx       3.4.4   2020-09-03 [1] CRAN (R 4.0.2)
## ps             1.4.0   2020-10-07 [1] CRAN (R 4.0.2)
## purrr          0.3.4   2020-04-17 [1] CRAN (R 4.0.2)
## R6             2.4.1   2019-11-12 [1] CRAN (R 4.0.2)
## remotes        2.2.0   2020-07-21 [1] CRAN (R 4.0.2)
## rlang          0.4.7   2020-07-09 [1] CRAN (R 4.0.2)
## rmarkdown      2.5     2020-10-21 [1] CRAN (R 4.0.3)
## rprojroot      1.3-2   2018-01-03 [1] CRAN (R 4.0.2)
## rticles        0.16    2020-09-22 [1] CRAN (R 4.0.3)
## scales         1.1.1   2020-05-11 [1] CRAN (R 4.0.2)
## sessioninfo    1.1.1   2018-11-05 [1] CRAN (R 4.0.2)
## stringi        1.4.6   2020-02-17 [1] CRAN (R 4.0.0)
## stringr        1.4.0   2019-02-10 [1] CRAN (R 4.0.2)
## testthat       2.3.2   2020-03-02 [1] CRAN (R 4.0.2)
## tibble         3.0.3   2020-07-10 [1] CRAN (R 4.0.2)
## tidyselect     1.1.0   2020-05-11 [1] CRAN (R 4.0.2)
## usethis        1.6.3   2020-09-17 [1] CRAN (R 4.0.2)
## UThwigl      * 0.1.0   2020-10-28 [1] local
## vctrs          0.3.2   2020-07-15 [1] CRAN (R 4.0.2)
## withr          2.2.0   2020-04-20 [1] CRAN (R 4.0.2)
## xfun           0.16    2020-07-24 [1] CRAN (R 4.0.2)
## xtable       * 1.8-4   2019-04-21 [1] CRAN (R 4.0.3)
## yaml           2.2.1   2020-02-01 [1] CRAN (R 4.0.2)
##
## [1] C:/Users/tonyd/Documents/R/win-library/4.0
## [2] C:/Program Files/R/R-4.0.2/library
```

The current Git commit details are:

```
# git2r::repository(here::here())
```