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

Anthony Dosseto[1], Ben Marwick[1]

[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 to closed- and open-system U-Th dating to the broad scientific community, here we provide an R package, named *UThwigl*. 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.

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 incorporated into 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. Ludwig 2003a). 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 (Ludwig 2003b), 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, U-Th dating requires to take into account open system behaviour. The diffusion-adsorption model of Pike and Hedges (2002) and the diffusion-adsorption-decay (DAD) model of Sambridge et al. (2012) were instrumental to implement successfully open-system U-Th dating. They allow for advective and diffusive transport of uranium and thorium isotopes, while include synchronous radioactive decay. Software implementation for the DAD model 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; Pike and Hedges 2002; Hoffmann et al. 2018). Thus, this approach has significantly improved our understanding of human evolution (e.g. Dirks et al. 2017; Sutikna et al. 2016; Hoffmann et al. 2018).

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 (2003a) while the latter applies the 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 are collected in a different collector (Faraday cups or ion counters). 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 sample 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 @ref(fig:femurpic)).

(ref:femurpiccap) 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).

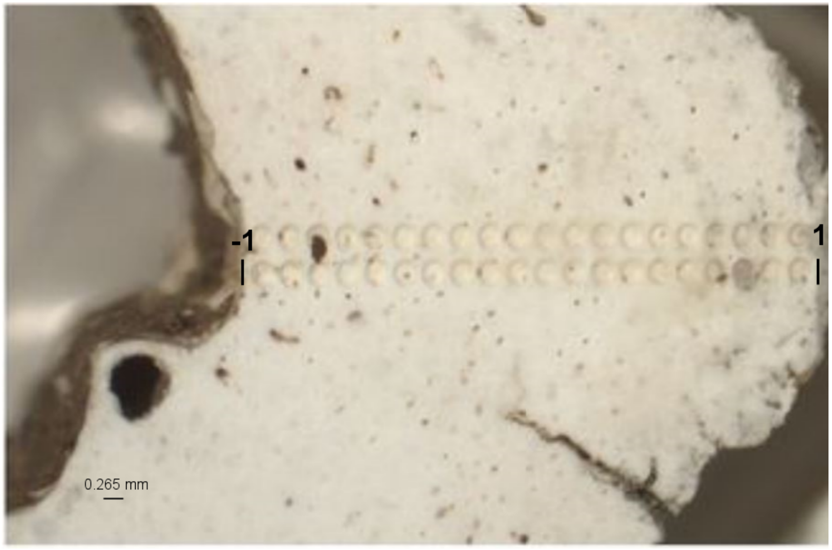

Figure 1: (ref:femurpiccap)

(#fig:femurpic)

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 @ref(fig:shinyfig)). 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.

(ref:shinyfigcap) 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.

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

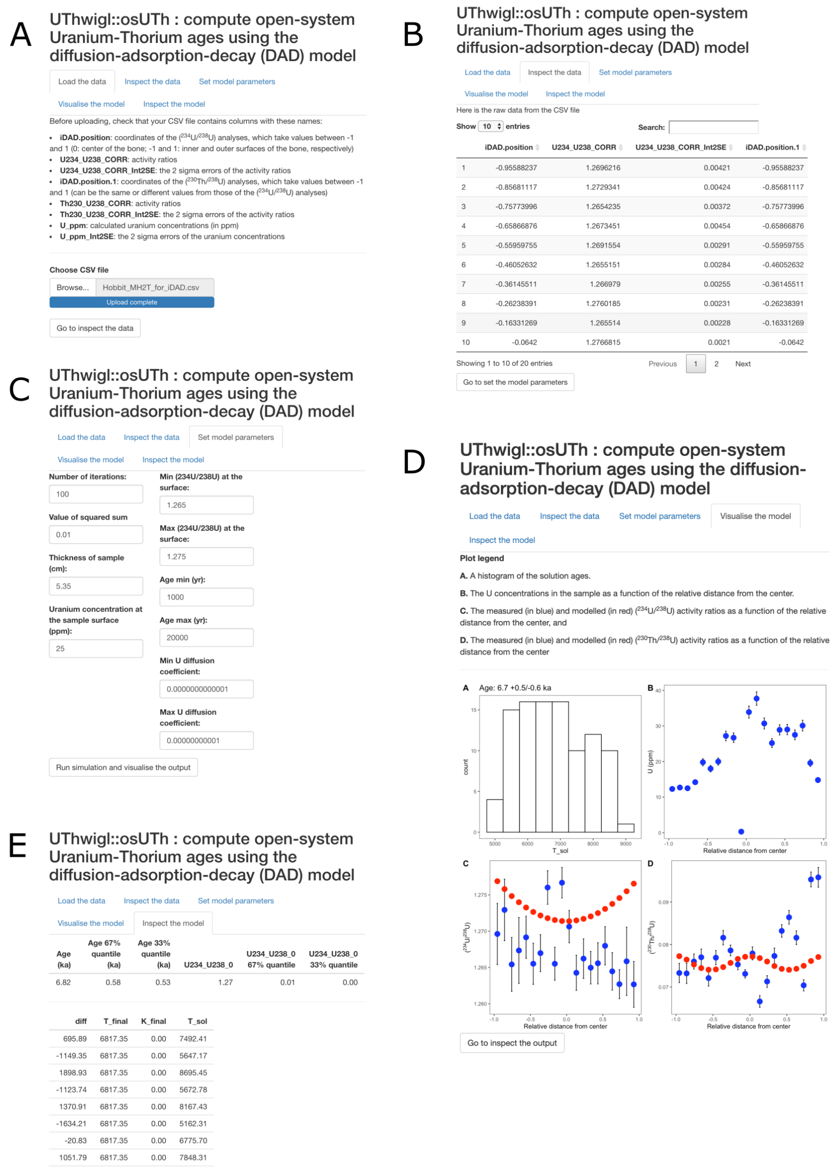

Figure 2: (ref:shinyfigcap)

(#fig:shinyfig)

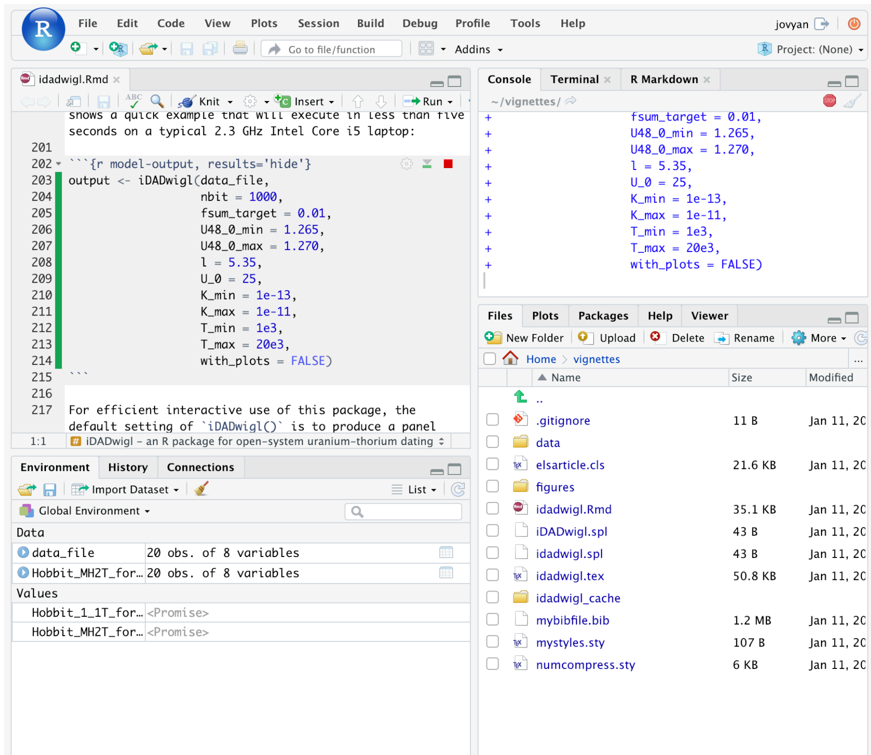

Figure 3: (ref:binderfigcap)

(#fig:binderfig)

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 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.

(ref:binderfigcap) Screenshot of Binder running R and RStudio in a web browser window.

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.

```
if(!require("remotes")) install.packages("remotes")
remotes::install_github("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:

```
# 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:

```r
# 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()`

```r
# 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 @ref(tab:pan) 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 @ref(tab:pan) 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 (it is 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
output_cs <-
  csUTh(
    input_data_cs,
    sample_name = 'YP003',
    nbitchoice = 100,
    detcorrectionchoice = TRUE,
    R28det = 0.8,
    R28det_err = 0.08,
    R08det = 1,
    R08det_err = 0.05,
    R48det = 1,
    R48det_err = 0.02,
    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 @ref(fig:csuthvizfig). 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 plots, 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 @ref(tab:panoutput). 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	117.038	0.3330	1.1720	0.0010
YP003-1_2	121.885	0.4080	1.1590	0.0010
YP003-1_3	116.316	0.2920	1.1710	0.0010
YP003-1_4	110.580	0.2330	1.1670	0.0010
YP003-1_5	119.471	0.2860	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:

(ref:csuthvizfigcap) 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