**Plum**

This document describes the general use of Plum. Plum is an R package under development; any suggestions and bug reporting will be appreciated.

Note:

Because Plum uses library rPython, Plum is only able to run in UNIX based systems (Linux/Mac OS).

#################################

## Installing Plum

#################################

Plum uses Python 2.7 to run the computer-intensive MCMC necessary for most Bayesian methods. For this reason, Python 2.7 should be installed on the system.

To install python 2.7 in Linux:

Arch

# pacman -S python2 python2-numpy python2-scipy python2-matplotlib

Fedora

# dnf install python python2-numpy python2-scipy python2-matplotlib

Ubuntu

# apt install python python-numpy python-scipy python-matplotli

To install in Mac, we recommend the installation of Anaconda which includes all the necessary packages (https://www.anaconda.com/download).

Now that python and the necessary libraries are installed we can install rPython in R by using:

*install.packages("rPython", configure.vars= "RPYTHON\_PYTHON\_VERSION=2")*

To install plum from the github repository we need the library devtools, to install it we can use

*install.packages("devtools")*

Now to install Plum we can simply call the library devtools and the function install\_github:

*library(devtools)*

install\_github("maquinolopez/Plum")

#################################

## Running Plum

#################################

To load Plum we can use:

*library(Plum)*

This loads Plum and the necessary packages.

To test Plum, a simulated core can be created and run (similar to the one shown in Aquino-López et al. (2018)) by using:

*runPlum()*

This will create a folder named Plum in the ~/ directory, and within this folder a new folder named “Simulation-#” (where # is a random number between 1 and 99).

In this folder a csv file with the same name can be found with the simulated data; this file contains the format which Plum uses.

Depth (cm), Density g/cm³, 210Pb (Bq/kg), sd(210Pb), Thickness (cm), 226Ra (Bq/kg), sd(226Ra)

1,0.10,101.95,4.58,1,14.41,0.67

This run will take several minutes (40-60) depending on computer specs.

At the end of the run Plum will show the energy of the MCMC (this helps to observe if the MCMC has converged) and the resulting chronology. The plot with resulting chronology will show the memory of the age-depth model, the accumulation in yr/cm, the supported 210Pb and the supply of 210Pb, as well as the age-depth model with the 95% credible intervals (in red) and a series of blue rectangles showing where samples were taken (in depth).

Plum adds this plot to the Core folder with a new folder named results-Corename; in this folder all the information to re-plot the age-depth model can be found.

#################################

##How to use Plum using your own data

#################################

To use Plum in your own data, you can create a csv file with the same structured as the one from the simulated data;

Depth (cm), Density g/cm³, 210Pb (Bq/kg), sd(210Pb), Thickness (cm), 226Ra (Bq/kg), sd(226Ra)

If data from 226Ra is missing delete this column (including the headers).

It is important to have a header like the one shown above or a similar one. The order of the variables is very important.

This file should be put into a directory with the same name as the file, i.e.:

~/directory/Corename/Corename.csv

with this file in place, you can use the function:

runPlum(Core.name=“Corename”, Folder= “~/directory”)

To run Plum, if you use the default ~/Plum directory (~/Plum/Corename/Corename.csv). It is not necessary to specify the directory and runPlum(Core.name=“Corename”) is sufficient.

If 226Ra data are present, Plum will suggest the use of linear regressions to determine which samples to use to infer the supported 210Pb. Given that the user knows how many samples should be used to infer the supported 210Pb the option “number\_supported = ” can be added with the number of samples from the deepest part of the core should be used only for inferring the supported 210Pb. In the case 226Ra is available, Plum will ask if you want to infer a single supported 210Pb for the whole core or use a supported 210Pb variable per sample.

Note: Using individual supported 210Pb estimates per sample will double the number of parameters in the model which will result in longer computing times.

Plum can also incorporate data from 137Cs into the chronology. If this data is available,

runPlum(…, Cs = spike’s depth, Cs\_date = spike’s assumed date, Sampledate = Core’s sample year)

This will let Plum know that it is highly likely that such depth is to have the given date. Plum will use this data to improve the chronology.

#################################

##Getting ages of specific depths

#################################

Plum will add a file named intervals.csv. This file contains the age estimates and credible intervals for the age-depth model sections’ depths; Plum by default uses 20 sections (note: these depths may be different from the sampled depths). The first column shows the depths, second the lower interval, the third the mean age and the last shows the upper interval.

If you want age estimates for specific depths of interest you can use the function age\_of()

ageof(x=Depth of interest, folder= ~/directory/Corename/, interval= the probability of the desired interval -in decimals 95% =.95-)

This function will provide a graph with the posterior distribution of the age of the given depth and return a list with the lower limit mean and upper limit of the age using the given probability (by default 95%).