Quick Start Guide (v1.0)

Download Software:

This software is open sourced and available for download at https://github.com/eilion/DPGP-Stack. The algorithm runs on MATLAB. Extract the folder from the zip file and save it to the desired location. To confirm the algorithm is running correctly, open MATLAB and type 'results = Core_Aligner('Default') into the MATLAB command line. When the run is finished, type 'results = AgeVsDepth('Default') to view the results.

Note: Most of the software's adjustable parameters are not described here because they generally do not need to be changed from the default setting. For a complete list of adjustable parameters, see the full documentation.

To Build A Stack:

Step 1. Create Input Folder

Create a folder with a user specified name in Inputs/Stack_Construction/. This folder will contain three text files. The 'Default' folder serves as an example.

Step 2. Populate Input Folder Files

Copy and paste the three text files from the 'Default' folder into the new input folder.

- 'records.txt' contains a list of cores used in the run. Delete the current list and enter a list of core names that match the core folder names you will create in step 3.
- The file 'setting_stack.txt' contains a list of settings for the run. The variable 'data_type' indicates if $\delta^{18}O$ or/and ^{14}C data will be used in the run. Setting 'data_type' to 'd18O', '14C', or 'both', indicates that the software will use only $\delta^{18}O$ data, only ^{14}C data, or both types of data respectively. Beginning users should leave the other entries in 'setting_stack.txt' equal to the default values.
- Copy one of the pre-loaded stacks from the folder XXXX to your Input folder, and then rename it 'initial_stack.txt'. The file 'initial_stack.txt' provides a "guess" for the first iteration of the stack; it must be at least as long as the new stack you want to create. If 14C data are used, the new stack will not be very sensitive to this choice. If only d18O is used, the age model of the final stack is likely to be influenced by the initial stack. The default stack is the global stack presented in Lisiecki & Stern (2016), which covers the age range 0-145 kyr. For longer stacks, we have also included HMM-Stack and LR04 which each cover approximately 0-5 Myr.

Step 3. Create Core Folders

In addition to creating an input folder, you must create a folder for each core in 'Cores\'. The default core folder serves as an example. The folder names must match the names in 'records.txt'. If the names do not match, or if there is a name in 'records.txt' and no core folder, you will encounter an error message. If you have other additional, non-14C age information, this can be entered in additional_ages.txt (see the full documentation).

- d18O_data.txt contains the $\delta^{18}O$ data for the core and has two-columns: depth (meters) and $\delta^{18}O$ observations (per-mil). If 'data_type' is set to '14C', you may omit this file.
- C14_data.txt contains the ^{14}C ages for the core and has six columns: depth (meters), ^{14}C ages (uncalibrated ^{14}C years), ^{14}C measurement error (years), reservoir age (years), the reservoir age standard deviation

(years), and type of calibration curve (1: IntCal13, 2: Marine13, 3: SHCal13). If 'data_type' is set to 'd18O', you may omit this file.

Step 4. Run the Software

In Matlab, enter the commands:

- addpath('Codes/');
- results = Stack_Learner(inputFile);

Here, inputFile indicates the name (not path) of the folder made in step 1. For example: results = Stack_Learner('Default');

All results are stored in the folder 'Outputs/Stack_Construction/':

- results.mat contains all results. The summary of inferred ages is stored as a set of text files in /ages.
- stack.txt contains the constructed stack from the stack construction algorithm. The first column is age (kyr), the second column is $\delta^{18}O$ (per-mil), and the third is standard deviation (per-mil).
- CORENAME_age.txt for each core in the stack where CORENAME is the name indicated in the records.txt input file. This text file has four columns: depth, median age, and lower and upper 95% confidence bands.

Step 5. View the Results

Run any of the following commands to visualize the results:

- AgeVsDepth(results_path); returns depth vs. age plots for each record in results_path. For example, AgeVsDepth('Outputs/Stack_Construction/Default/');
- AgeVsD18O(results_path); returns a visualization of the alignment results for each record in results_path. For example, AgeVsD18O('Outputs/Stack_Construction/Default/');

To Align Cores to an Existing Stack (or To Create ^{14}C-only Age Models):

Step 1. Create Input Folder

To align $\delta^{18}O$ records to an existing stack or to construct a radiocarbon age model, create a folder with a user specified name in Inputs/Core_Alignments/. These folders will contain three text files. The 'Default' folder serves as an example.

Step 2. Populate Input Folder Files

Copy and paste the three text files from the 'Default' folder into the new input folder.

- 'records.txt' contains a list of cores used in the run. Delete the current list and enter a list of core names that match the core folder names you will create in step 4.
- 'setting_align.txt' contains a list of settings for the run. The variable 'data_type' indicates if $\delta^{18}O$ or/and ^{14}C data will be used in the run. Setting 'data_type' to 'd18O', '14C', or 'both', indicates that the software will use only $\delta^{18}O$ data, only ^{14}C data, or both types of data respectively.

Beginning users should leave the other settings in 'setting_stack.txt' equal to the default values.

To set your alignment target, copy one the pre-loaded stacks from the folder XXXX to your Input folder, and then rename it 'stack.txt'. If constructing a radiocarbon age model, you may omit this file. The default stack is the global stack presented in Lisiecki & Stern (2016), which covers 0-145 kyr. If using a global population of cores, we

recommend using this stack. However, if using a regional population of cores, or only aligning a single core, we provide the eight regional stacks from Lisiecki & Stern (2016). For aligning longer cores, we have also included HMM-Stack and LR04, which each cover approximately 0-5 Myr. See full documentation for creating your own files for alignment targets.

Step 3. Create Core Folders

In addition to creating an input folder, you must create a folder for each core in 'Cores\'. The default core folder serves as an example. The folder names must match the names in 'records.txt'. If the names do not match, or if there is a name in 'records.txt' and no core folder, you will encounter an error message. If you have other additional, non-14C age information, this can be entered in additional_ages.txt (see the full documentation).

- d18O_data.txt contains the $\delta^{18}O$ data for the core and has two-columns: depth (meters) and $\delta^{18}O$ observations (per-mil). If 'data_type' is set to '14C', you may omit this file.
- C14_data.txt contains the ^{14}C ages for the core and has six columns: depth (meters), ^{14}C ages (uncalibrated ^{14}C years), ^{14}C measurement error (years), reservoir age (years), the reservoir age standard deviation (years), and type of calibration curve (1: IntCal13, 2: Marine13, 3: SHCal13). If 'data_type' is set to 'd18O', you may omit this file.

Step 4. Run the Software

Enter the commands:

- addpath('Codes/');
- results = Core_Aligner(inputFile);

Here, inputFile indicates the name (not path) of the folder made in step 1. For example: results = Core_Aligner('Default'); Results are stored in the folder 'Outputs/Core_Alignments/':

- results.mat contains all results. The summary of inferred ages is stored as a set of text files in /ages.
- CORENAME_age.txt for each core in the stack where CORENAME is the name indicated in the records.txt input file. This text file has four columns: depth, median age, and lower and upper 95% confidence bands.

Step 5. View the Results

Run any of the following commands to visualize the results:

- AgeVsDepth(results_path); returns depth vs. age plots for each record in results_path. For example, AgeVsDepth('Outputs/Core_Alignments/Default/');
- AgeVsD18O(results_path); returns a visualization of the alignment results for each record in results_path. For example, AgeVsD18O('Outputs/Core_Alignments/Default/');