**D17O Data Reduction – R Code Instructions**

**Code runs in three programs:**

1. Format the lab data file (ReactorSpreadsheetFormatter)
2. Correct the data (IPL\_17O\_correction)
3. Make lists of all corrected data and their averages (IPL\_17O\_Sorting)

**Folder,** *program*, and (file) organization

1. **Data Reduction Procedure**
   1. **0000\_LabFile Formatting**
      1. **000\_Reactor Spreadsheet Raw**
         1. (raw reactor spreadsheets from lab, excel)
      2. **001\_Reactor Spreadsheet Formatted**
         1. (formatted, raw reactor spreadsheets from lab, csv)
      3. *ReactorSpreadsheetFormatter.R*
   2. **0001\_ReductionCode**
      1. **ReductionCodeFunctions**
         1. *correct.SMOW.SLAP.basic.R*
         2. *correct.SMOW.SLAP.linearSMOW.R*
         3. *ggplot.maker.d33d34.R*
         4. *ggplot.maker.IPLD17O.R*
         5. *ggplot.maker.IPLD17O.std.R*
         6. *multiplot.R*
         7. *segment.finder.R*
      2. (acceptedStds.csv, contains values for lab standards)
      3. *IPL\_17O\_correction.Rmd*
   3. **0002\_LabFileCorrectedOutput**
      1. (postProcessInput.csv, contains post-processing information for the calibration scheme to use)
      2. (all corrected data files, csv)
   4. (D17O\_dataReduction\_Instructions.docx, the instructions you are reading)

**Instructions for use:**

**ReactorSpreadsheetFormatter**

This program formats the lab spreadsheet so that it can be more easily corrected. The code is set up to reduce a single file at a time, but can be easily changed to run all at once.

1. (if code does not find path automatically) Update the path to the main data reduction folder, “Data Reduction Procedure” (line 15)
2. Update the data file name for each reactor (line 17)
3. Add the variable name we want associated with the reactor (line 33)
4. Choose which reactor to run ( line 41)
5. Note the input and output paths (line 50)
   1. path.in – where the unformatted spreadsheets are stored
   2. path.out – where you want the formatted .csv file output to
6. Excel sometimes changes the format of DateTime from “text” to “number” behind the scenes. I’m not sure why this is. Since we have so few reactors, just double check the .csv output format for DateTime, which should look like a date string (e.g., yyyy-mm-dd HH:MM:SS) and not a number. If it is output as a number, go back to the lab spreadsheet and, in a new column, convert the number to text using =text(A1,”yyyy-mm-dd HH:MM:SS”), where A1 is the DateTime cell. Copy and paste the *values* back into the original cells.

**Instructions for use:**

**IPL\_17O\_correction**

This program corrects the formatted data spreadsheet using the SMOW-SLAP correction. By convention, a linear correction is always applied based on SMOW d33 and d34 vs. analysis time. Note that values are indifferent to poor correlations – a linear or basic correction will produce the same results in this case.

1. Update the reactor name list (line 37)
2. (if code does not find path automatically) Update the path to the main data reduction folder, “Data Reduction Procedure” (line 54)
3. Choose the option for the “tertiary” D17O correction (line 60)
   1. ter.cor.option
      1. 1 - Use 'best' available standard set (in order of decreasing priority: 2 pt. correction with NBS-18, NBS-19, or IAEA-603, 1 pt. correction with NBS-18, NBS-19, or IAEA-603, 1 pt. correction with 102-GC-AZ01).
      2. 2 - Always use 102-GC-AZ01 (keeps normalization standard constant between reactors)
4. Choose which reactor to run (line 68)
5. Note the input and output paths (line 74)
   1. path.in – where the formatted spreadsheets are stored
   2. path.out – where you want the corrected data .csv file output to
   3. path.to.accept.std – where the file containing accepted standard values is
   4. accepted.std.file –file containing accepted standard values
   5. path.to.functions – path to functions for correcting and plotting
6. Define parameters (line 104). In general, do not change these values. They control whether or not a drift correction is performed and the d18O-thresholds for priming.
7. The post-processing information file (postProcessInput.csv) must be set up beforehand in the output folder (path.out, it already exists). This auto-assigns the preferred correction scheme to columns with names “prefer.auto…”, but you must verify it by hand before data can be post-processed. This is achieved by the post-processing program calling only from secondary, user-defined values that are reset each time the IPL\_17O\_correction is run (see columns with “prefer.user…”).

**Instructions for use:**

**IPL\_17O\_Sorting**

This program sorts all of the data and outputs a list of all individual analyses as well as the average values (averages are based on sample ID, so if spelling is off it cannot catch all the replicates). It chooses the correction scheme based on what user input in postProcessInput.csv, which must be updated every time IPL\_17O\_correction is run for a reactor.

1. Check that postProcessInput.csv has been updated. Columns are as follows (with -9999 meaning no entry is needed/available):

A - reactor file number

B - the last time this spreadsheet was updated (just so you have a sense when you last altered it)

C - the number of segments in the reactor (up to 5 is allowed)

D-I - information for the automatically selected ("preferred") correction scheme.

for D:

1 - basic correction (no segments)

2 - linear correction (no segments)

3 - segments (which are then subdivided in E-I)

E-I - the automatically selected correction scheme for each segment within the reactor

1 - basic correction

2 - linear correction

J-O – same as for D-I, but this information is the user-selected (“user” correction scheme

1. (if code does not find path automatically) Update the path to the main data reduction folder, “Data Reduction Procedure” (line 12)
2. Note the input and output paths (line 14)
   1. path.in – where the corrected reactor data csv files are stored
   2. path.out – where you want the formatted .csv file output to (0002\_LabFileCorrectedOutput)