Booting Up the RPD SQL Database

Bart’s GCD analysis and composite curve generating scripts live here:

<https://pjbartlein.github.io/GCDv3Analysis/index.html>

—-- Instructions for installing MySQL and Workbench and setting up the database —---

Hi Nick,

I've just gotten the database connection working on my laptop and it took some tinkering. Here are general instructions you can try, followed by explanations of Bart's R scripts. The first few lines of the first R script are the hard part, and there is a bunch of stuff to do before you get to R.

First install MySQL, which runs in the background on startup on your laptop. Then you also have to install MySQLWorkbench which is the interface to the database. Links to these two items are inside the first R script (Read\_db.R) in the files attached.

After you have those two pieces installed, open your System Preferences and look for MySQL Server - there should be a new tile there for MySQL.

Click on MySQL Server in System Preferences, click on Initialize Database (on the Instances tab), and enter a password (for the database).

Click on the Start MySQL Server button if the server isn’t running.

Start MySQL Workbench.

You should see a tile that says something like "Local instance 3306/root/local host3306…”. Click on that to open the “Local Instance” management window.

Click on the Schemas tab. There should be an existing Scheme called “sys”

Right-click in the left-hand window (that lists the schemas) and select Create Schema…. Name it RPDv1b, click Apply

Then, on the Administration tab, click on Data Import/Restore

Select the Import from Self-Contained File, fill in the path to the database (i.e. wherever you unzipped rpdv1b.sql), and enter “RPDv1b” into the “Default Target Scheme” box (or use the dropdown box arrows), then click Start Import

Back to the Schemas tab, if you click on “RPDv1b” > “Tables”, you should see a list of tables in the database.

You can then write your first query in what should be the blank page showing.

Double-click on "RPDv1b so it knows which database you are querying.

Then type in the whitespace:

**SELECT \* FROM entity;**

And then click the Execute Query tool (lightning bolt). If all was successful, you should get a listing of all the records in the database below.

Now onto the R scripts. I put the database inside my working directory and started a new Project in R in that same directory. I'm not sure if that is necessary but it is probably good practice in any case - to have the sql database file and the scripts all in the same folder.

Open the first one -- read\_db.R and install the packages, then load the libraries

install.packages("RMariaDB")

install.packages("DBI")

library(RMariaD)

library(DBI)

continue and try to connect. If you don't get an error message on the first line, keep going! (use your own password).

The strategy for the subsequent R scripts is to make a bunch of .csv files, one per entity now, as opposed to one per site, and then use an entity list (like the previous site list) to determine which sites go into a particular curve, etc. I still like being able to pop open a .csv file to see what’s what. So, here’s how things work:

read\_db.R — opens and reads tables the MySQL data base. I hadn’t thought of this but the Actual ODBC Pack should be able to do this, but I followed the idea of using MySQL (the Community edition, see the URL in the file). After installing, it puts a control icon in System Preferences that lets you turn on or turn off the server. Then MySQL Workbench lets you import an exported database as a “new schema” or something. Takes a couple of tries.

sample2\_01.R basically rewrites the sample table, adding info on the measurement type (influx, concentration, etc.) from the entity table, the original chronology and the new chronology from the appropriate tables. If there’s a new chronology, that gets used.  
  
entity2\_01.R rewrites the entity table, adding info from the samples table, and generates an entity list.  
  
fixups\_01.R does a few fixups, and also sets a flag as to whether individual sites should be skipped, ones with, e.g. no charcoal data, or just one or two samples. The .xlsx file is an entity list sorted to show the skipped and fixed-up entities.  
  
loop1\_01.R loops over the entities, calculating sed rate, influx (if the data was concentrations), concentration (if the data was influx), etc., and writes out a .csv file for each entity wasn’t skipped. There’s an issue with sed rate we need to address (see below).  
  
trans-and-zcore\_11\_code.R, trans-and-norman\_11\_code.R, prebin\_zscore\_code\_11.R and prebin\_norman\_code\_11.R are pretty much the same as before.  
  
smooth\_curve\_11\_code.R and bin-boot\_11.R, ditto, but I cleaned a bunch of stuff up.  
  
(normans (normalized anomalies) and bin-booting are ideas from Sandy and Colin’s ESD feedback paper.)

The sed rate issue: The databases don’t contain sample thickness values for all sites, and estimating it by interpolation leads to “spikey” sed rate values (look at Cygnet, for example), and in turn, influx values. I think the solution would be to adopt the thickness values if they are there, and make sense, otherwise, get the slope (i.e. numerical derivative) of the age vs. depth curve (or maybe just do that for everything). Needs some more thinking about.

Let me know how it goes - if it's not working we can troubleshoot when we meet.

J