**Using the R version of Horizon Annealing with Biostratigraphic Data**

**Introduction**

Horizon Annealing (HA) is a method for carrying out quantitative stratigraphy, using simulated annealing methods to order collections of data. The simulated annealing procedure minimizes a penalty function (aka an objective function) based on the ordering of the data. Different types of penalty functions may be used in HA, this discussion focuses only on the use of biostratigraphic data, where the penalty function is the sum of the local range extensions implied by the ordering of the collection horizons. A zero range extension would imply an ideal level of agreement of local observations with the global patterns of fads and lads in the data.

Please see the following papers for a more detailed discussion of HA:

Sheets, H. David., Charles E. Mitchell, Zachary T. Izard, John M. Willis, Michael J. Melchin, and Christopher Holmden. "Horizon annealing: a collection‐based approach to automated sequencing of the fossil record." *Lethaia* 45, no. 4 (2012): 532-547.

Melchin, M.J., Sheets, H.D., Mitchell, C.E. and Fan, J., 2017. A new approach to quantifying stratigraphical resolution: application to global stratotypes. *Lethaia*, *50*(3), pp.407-423.

**Input Data Format**

HA works with a presence/absence matrix, referred to as a grand capture matrix, summarizing which taxa (or which measurements) where seen in each horizon of each section of the input data.

Each row of the data matrix represents a particular horizon, the first five columns are information about the horizon:

the *section name*- included in the input file but not actually used in the calculations

the *horizon score* used to order the data, the researcher must assign the initial horizon scores,

the HA program will attempt to find the optimum set of horizon scores for the system, thus producing an optimal ordering of horizons, and events

the *section number*- the section number is assigned by you, the researcher, so that each section

has a number (starting at 1 and going up to N, if there are N sections). The algorithm uses this number in internal record keeping not the name.

the *horizon number*- within each section, the lowest horizon should be horizon 1, with each

successive horizon within the section increasing by 1

the *horizon height*- this is a measure of the height of the horizon, measuring upward

from the lowest horizon. If you don’t have height information, simply re-use the horizon number.

Subsequent columns in the data matrix (or grand capture history) represent present (1), absent(0) or not in section (-1) information for each species, or for other each measured variable of interest. In this discussion, we focus on biostratigraphic data only, but HA can deal with many other types of geological data which might prove to be stratigraphically informative.

Each taxa is coded as a 0 (absent on a particular horizon, but seen in the section), 1 (present on the horizon) or -1 (never seen on any horizon in the section). Other geological information can be coded as presence/absence data or as a continuous floating point value.

Most people will probably assemble their data in Excel, although other approaches are possible. Look at the example file- **Riley\_62\_for\_R.csv.** which is a version of the Riley formation data used by Shaw in the development of graphic correlation, and which has been used by Sadler with CONOP and by use with HA as a test set. Here is the upper left corner of the data matrix in this file.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Section\_Name | Score | Section | Horizon | Height | Holcacephalus tenerus | Modocia cf.oweni |
| Morgan\_Creek | 0.13721328 | 1 | 1 | 1 | 0 | -1 |
| Morgan\_Creek | 0.26197153 | 1 | 2 | 2 | 0 | -1 |
| Morgan\_Creek | 0.28228564 | 1 | 3 | 3 | 0 | -1 |
| Morgan\_Creek | 0.30867342 | 1 | 4 | 4 | 0 | -1 |
| Morgan\_Creek | 0.31921832 | 1 | 5 | 5 | 0 | -1 |
| Morgan\_Creek | 0.32963043 | 1 | 6 | 6 | 1 | -1 |
| Morgan\_Creek | 0.35206387 | 1 | 7 | 7 | 0 | -1 |
| Morgan\_Creek | 0.36084814 | 1 | 8 | 8 | 0 | -1 |
| Morgan\_Creek | 0.41948627 | 1 | 9 | 9 | 0 | -1 |

You will need to assemble this input data file of your own data before running HA, this is least pleasant part of the process. We have a system of storing the data from single publications or collections as single files and then aggregating them into a single large input file. Pete Sadler does something similar in working with CONOP.

Once you have all the horizons for all sections in place, you need to create a starting set of scores to specify what the starting solution looks like. The scores within a given section must increase from the bottom to the top, since HA sorts the data by scores and horizons within a section need to be ordered consistently with their known stratigraphic relationship.

To do this in excel

1. Sort your data by section and horizon, increasing in section and horizon number.
2. Create a set of random uniform numbers ranges from 0 to 1. Create a list with the same number of random values as your first section. Use Excel to order this list from low to high, then paste this list in as the scores for your first section.
3. Repeat step b for each section in your data, so that when you are done each horizon has a random score, but the scores always increase from the bottom of the section to the top.

You can generate and sort your random values in Excel, or you can do it in R and paste them into Excel, just be sure that the random values respect the known stratigraphic relationships within each section.

Note: the HA program will keep track of the column names in your data set, and will use them to label the species in plots, so use column names that make sense to you particularly if you use abbreviations.

**Section list**

You will also need a separate list of the section names and the corresponding section numbers if you want to use plot functions (see below) that label the sections properly. The table below shows the content of the example file riley\_sections\_names.csv

|  |  |
| --- | --- |
| SectionName | SectionNumber |
| Morgan\_Creek | 1 |
| White\_Creek | 2 |
| James\_River | 3 |
| Little\_Llano\_River | 4 |
| Lion\_Mountain | 5 |
| Pontotol | 6 |
| Streeter | 7 |

Note: the column names need to match these exactly so that the R routines can find the section names. You can easily create this file in Excel.

**The Penalty Specification file**

HA allows for different types of data, so that the program has to be told which columns to use in each penalty calculation. In the case of the Riley data set, all the data columns (62 columns of distinct species) are used in the biostratigraphic penalty. No other penalties are used.

--------------------------------------------------------------------------------------------------------

Here is a copy of the penalty specification file penalty\_spec\_Riley\_62\_set.R

n\_biostrat=62

biostrat=1:62

# for biostrat data, or Taxa FADs,LADs, the biostrat variable is the numbers of the columns with taxa

n\_pmag=0

pmag=63

# pmag is a list of the column(s) with paleomagnetic signals, or really any binary data, NA values are not counted

n\_dates=0

dates=matrix(c(109,2,110,1,100,111,2,112,1,100,113,2,114,1,100),nrow=3,byrow=TRUE)

# each row of the dates matrix is a set of data to be entered into the passing penalty

# the first entry on each row is the column of the lower variale

# second entry on a row is the data type 0- singular date, 1- FAD, 2-LAD

# third and fourth entries on each row are the column and type of the second variable

# fifth value on each row is the weight

n\_ashes=0

ashes=matrix(c(68,100,69,100),nrow=2,byrow=TRUE)

n\_continuous=0

continuous=matrix(c(70,5,71,5),nrow=2,byrow=TRUE)

# this last line assembles all the penalty terms into a list named penalty\_spec\_62 for the riley

# data set

penalty\_spec\_62=list(n\_biostrat=n\_biostrat,biostrat=biostrat,n\_pmag=n\_pmag,pmag=pmag,n\_dates=n\_dates,dates=dates,n\_ashes=n\_ashes,ashes=ashes,n\_continuous=n\_continuous,continuous=continuous)

If you are using only biostratigraphic data, set n\_biostrat equal to the number of species, and set biostrat=1:N, for N species. Note you could use any array of column numbers here, to allow for control of which species are used

**Running the HA program**

Once you have assembled your data files, you are ready to run the program in R.

1. You need to “source” the following files

HorizonAnneal3\_1.R

ColumnRangeExtenstion.R

Penalty\_spec\_Riley\_62\_set.R\*

\*- you would create your own penalty spec file for your data, by modifying the penalty spec file for the Riley 62 set, so this file name would change

Here are the R commands I used, based on where I stored the files

source("/Users/sheets/Dropbox/HA\_Distribution/HorizonAnneal3.R")

source("/Users/sheets/Dropbox/HA\_Distribution/ColumnRangeExtension.R")

source("/Users/sheets/Dropbox/HA\_Distribution/penalty\_spec\_\_Riley\_62\_set.R")

1. Now load your data file, and the file of section names

myCH=read.csv("/Users/sheets/Dropbox/HA\_Distribution/riley\_62\_for\_R.csv", header=TRUE,stringsAsFactors=FALSE)

mySections=read.csv("/Users/sheets/Dropbox/HA\_Distribution/riley\_section\_names.csv",header=TRUE,stringsAsFactors=FALSE)

Note: when reading in files, I often use: mySections=read.csvf(file.choose(),header=TRUE,stringsAsFactors=FALSE)

1. You are now ready to run the annealing process

The first column of the input data structure myCH will have the section names, which the program does not want as an input, the command below calls HA with this first column removed

j=HorizonAnneal3\_1(myCH[,-1],pen\_str=penalty\_spec\_62)

when you run this, a lot of information will scroll by on the screen showing you the solution update, the Best Pen information is the best penalty value seen to date. When the program stops, you will see something like:

N outer: 399 T: 2.76523e-18 Best pen: 148 0.8524562 6 Recent prop pen: 148

swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop swop

N outer: 400 T: 2.488707e-18 Best pen: 148 0.9053813 3 Recent prop pen: 148

user system elapsed

52.841 2.267 54.521

[1] "Best Total Penalty: 148.000000 "

[1] "Net taxa range extension 148.000000"

[1] "Pmag Penalty: 0.000000"

[1] "Pmag Reversals: 0.000000"

[1] "Ash term: 0"

[1] "Ash levels 0"

[1] "Continuous variable term: 0.000000 "

[1] "Passing Error: 0"

[1] "Move track values 2.523341 5.351496 4.223078 4.724208 8.784038"

0.008403361

The Best Total Penalty value is the ending penalty found by the program

The variable j is a data structure containing the output, j$d is the grand capture history representing the solution.

You need to run simulated annealing programs repeatedly to see if you have really found a minimum. Deciding when to stop is a bit of an art.

Here is the 2nd call to HorizonAnneal3, starting with the output solution of the prior trial

j=HorizonAnneal3\_1(j$d,pen\_str=penalty\_spec\_62)

In this case, there was no improvement in the solution, still at 148

The “swap” statement means the “swap” move, changing just two horizons at a time was used, meaning that the program is finding multiple equivalent solutions if the penalty is not decreasing. There are often many “minor” re-orderings of the data, since HA does not allow horizons to “tie” in position. In many cases, there is not enough information available to conclusively order all the horizons.

1. Controlling Annealing conditions

You can control the initial temperature of the anneal, the decrease rate in temperature and the number of trials at each temperature (the ninner value) and the number of reductions in temperature (nouter value). The total trials used is nouter\*ninner.

Higher temperatures allow more “exploration” of the solution space by allowing more changes in the solution to lower quality solutions, low T does not allow changes that increase the error much.

Here are the default values:

j=HorizonAnneal3\_1(j$d,param=list(nouter=400,ninner=100,temperature=5,cooling=0.9),pen\_str=penalty\_spec\_62)

I will alter the starting temperature, since the solution is already fairly good at this point and run it again

j=HorizonAnneal3\_1(j$d,param=list(nouter=400,ninner=100,temperature=1,cooling=0.9),pen\_str=penalty\_spec\_62)

With this lower temperature, I saw no more decreases in the penalty, but I’ll try one more time at a lower starting temperature, just to be sure

j=HorizonAnneal3\_1(j$d,param=list(nouter=400,ninner=400,temperature=0.2,cooling=0.9),pen\_str=penalty\_spec\_62)

I also did 400 trials at each temperature- see the ninner value, just to be on the safe side. There were no further decreases.

------------------------------------------------------------

**Plotting Results**

The matrix j$d now contains the solution, an ordering of all the horizons that produces the best (lowest) penalty one can locate. This matrix tells you the order of horizons, and also the ordering of events (FADs and LADs of species), plus whatever other information is contained on each horizon.

Plotting species ranges

The plot function is called as

fadladlist=PlotRangeChartPS2(j$d,1:62,tlabels=TRUE,MarkSections=FALSE,barOrder="Fad")



The variable fadladlist will be a listing of all taxa by taxa number, with the position of the FAD and LAD of the species listed, along with the species name

The option

fadladlist=PlotRangeChartPS2(j$d,1:62,tlabels=TRUE,MarkSections=TRUE,barOrder="Fad")

adds the section numbers and heights at which each species was found to the range chart plot, this is helpful in diagnosing problematic species, sections or data



**Plotting a Section Range Chart**

PlotSectionRange(j,sectNames=mySections)

-this plot shows the beginning and ending locations of each section plus all horizons within each section



This variation

PlotSectionRange2(j,sectNames=mySections)

Puts a red label beside each horizon indicating how many species were seen on it, as a way of understanding data density



**Further ideas and Code**

This is a very basic set of examples.

Not shown here at all are the approaches to jacknifing solutions to determine the degree of uncertainty or instability within the solution- this is a rather more complex task.

Also not included here are the approaches to other types of stratographic information which may be included in the solution.

I also have a number of R functions for comparing solutions, which are not discussed here.

Contact me for further information.