Author: Nadja Stalder

Running your own data with glide

Besides of preparing your data, you will need to create your own DEM file, grid file and control file by following the instructions below.
You need to install **GMT** – The generic mapping tool (http://gmt.soest.hawaii.edu) to do the step 1. Steps 2+3 are executed in the **Terminal** where you run glide

1. Create a new DEM with GMT

a) Download a global digital elevation model (DEM), for example the Gebco data set (https://www.gebco.net/data_and_products/gridded_bathymetry_data/)

b) Clip the DEM to your area of interest with gmt:
In the terminal, go to the folder containing the DEM (***data***?) and `type`:

```
grdcut name_gebco.nc -RlongW/longE/latS/latN -GDEM_myArea.grd
```

- longW, longE, latS, latN are the longitudinal and latitudinal extents of your area
- DEM_myArea.grd is the clipped DEM file of your region.

example:

```
grdcut gebco_08.nc -R-72/-70/-26/-22 -GDEM_Andes.grd
```

c) `grdinfo DEM_myArea.grd`
n_columns, n_rows are the nx, ny values for input 3 in glide.in

d) Convert grd file to xyz file:

```
grd2xyz DEM_myArea.grd > DEM_myArea.xyz
```

update input 2 in *glide.in* with DEM_myArea.xyz

```
bash-3.2$ grdcut gebco_08.nc -R-72/-70/-26/-22 -GDEM_Andes.grd
bash-3.2$ grdinfo DEM_Andes.grd
DEM_Andes.grd: Title: Produced by grdcut
DEM_Andes.grd: Command: grdcut gebco_08.nc -R-72/-70/-26/-22 -GDEM_Andes.grd
DEM_Andes.grd: Remark:
DEM_Andes.grd: Pixel node registration used [Cartesian grid]
DEM_Andes.grd: Grid file format: nf = GMT netCDF format (32-bit float), COARDS, CF-1.5
DEM_Andes.grd: x_min: -72 x_max: -70 x_inc: 0.00833333333333 name: user_x_unit n_columns: 240
DEM_Andes.grd: y_min: -26 y_max: -22 y_inc: 0.00833333333333 name: user_y_unit n_rows: 480
DEM_Andes.grd: z_min: -8128 z_max: 3005 name: user_z_unit
DEM_Andes.grd: scale_factor: 1 add_offset: 0
DEM_Andes.grd: format: netCDF-4 chunk_size: 240,160 shuffle: on deflation_level: 3
bash-3.2$ grd2xyz DEM_Andes.grd > DEM_Andes.xyz
```

2. Make a new grid

In glide, erosion rates are calculated for every data point (red) and on grid points (blue) that surround your data (Fig.1). Therefore, you will need to create a new file that contains the longitude, latitude and number of fault block of every grid point.

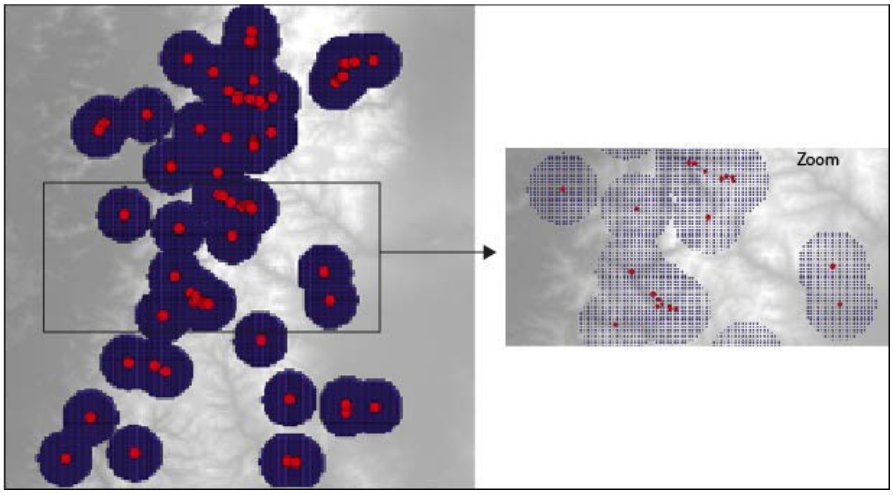


Figure *1***:** Erosion rates are calculated for data points (red) and for grid points (blue) surrounding the data.

a) In the **Terminal** (not **GMT**), go to the ***create_grid*** folder and type:

```
make
cd ..
./creategrid
```

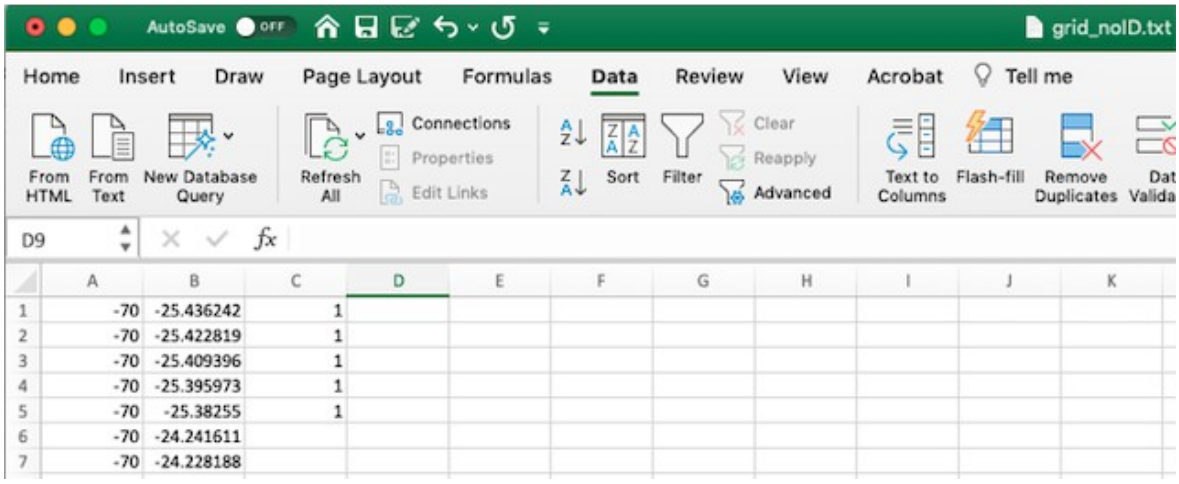
This creates a new file called *grid_noID.txt*

```
[redacted]$ cd Desktop/ExhumationAndes-master/glide_20200723/create_grid/
[redacted]:create_grid [redacted]$ make
gfortran  module_definitions.o initialize_parameters.o   -framework Accelerate -o ../creategrid
[redacted]:create_grid [redacted]$ cd ..
[redacted]:glide_20200723 [redacted]$ ./creategrid
 total ages          33 control points=          0
 creating dummy points
 number of dummy points=        4962
```

b) Assign fault ID's to the grid points.

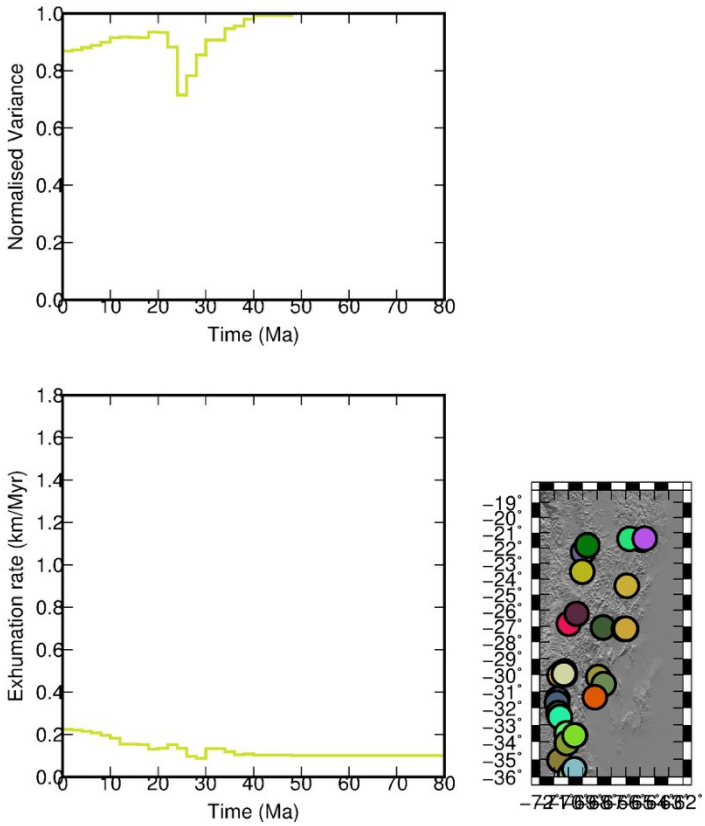
- If you don't use fault blocks:
  Open the *grid_noID.txt* file in excel or matlab, add a third colum with 1 as value respectively the same value that you have in your data file.
  This means that all grid points are in the same fault block.
- If you use fault blocks (see also point 4):
  Assign the individual blockID's by using ArcGIS or a similar program.

Save the file as *grid_dummy.txt* in the data folder of your main glide folder.



3. Make a new control file, *control_id.xy*

The *control_id.xy* file defines the locations where you will extract individual time-erosion rate paths that can be plotted with the script *indiv_ttpath.sh* in GMT:



a) Open the file *control_id.xy* in any text editor. Update longitude, latitude and blockID values according to your needs. Choose locations that are well resolved, i.e., where there are a lot of data otherwise you will just get the prior erosion rate.

b) If you change the number of locations (37): go to the ***src*** folder in the **Terminal**.

- Open file *initialize_parameters.f90* (eg, `vi initialize_parameters.f90`) and change the number at " params %control = 37 " (line number 108) to the number of locations you have defined in your *control_id.xy* file. If you use `vi` to open your file in the terminal, hit "`i`" on your keyboard to insert, "ESC" to close Insert – mode and ":q" to close and save the file.

```
 93 !first go through, see which data fall in space (and time) range
 94 k=0
 95 do i=1,20000
 96  k=k+1
 97  read(33,*,end=500) lo,la,tmp1,tmp2,tmp3,tmp4,ib
 98  !print*,lo,la,tmp1,tmp2,tmp3,tmp4
 99  if (lo.gt.180.) lo = lo - 360.
100  if (tmp2.gt.params%t_total.or. &
101      lo.lt.params%lon1.or.lo.gt.params%lon2.or. &
102      la.lt.params%lat1.or.la.gt.params%lat2) k=k-1
103 enddo
104 500 continue
105 params%n=k-1
106 rewind(33)
107
108 params%contr=37
109 !params%dummy=params%dummy+params%contr
110 params%dummy=params%contr
```

- type `make` to compile

- If you use more than 56 control points, then the file *colorsNEW.txt* must be updated with more colors.

4. The usage of faults blocks

a) To use fault blocks, you need to specify the fault number for every data point in your thermochronological data input file, in *grid_dummy.txt* and in *control_id.xy* in the last column. If you have many faults, it is useful to have them mapped in ArcGIS. You can than assign the blockID's automatically to the data points in the text files. If you don't use fault blocks, fill the colums of the blockID with "1".

b) To automatically plot the fault blocks when generating the erosion rate maps with the *script.sh*, you need to generate a text file *"block_boundaries.txt"* that contains the starting and ending coordinates of every fault line:

-70.64372772	-35.75646171	-69.57873468	-35.86252896
-69.64699176	-35.30967174	-69.57873468	-35.86252896
-70.56799992	-35.22300003	-70.64372772	-35.75646171

To do this, you can use the “Split Line at vertices tools” of ArcGIS to extract the coordinates of the mapped faults (that were polylines):

a) Create fields in attribute Table: Open attribute table of Polylines. Add fields: X_Start, Y_Start, X_End, Y_End.
b) Select Column in attribute table and Right click on field header > Calculate Geometry > Select Attribute; Add units and coordinate system!

Alternatively, you can use the “add geometry” tool from the data management:

a) Open Add Geometry attributes (Data Management), select Polylines and Line_START_MID_END. Export attribute table as textfile

Finally, to be able to plot the fault lines automatically when executing *script.sh,* you need to reformat “block_boundaries.txt”.

a) Make a copy of “block_boundaries.txt”
b) In the **Terminal**, type:

```
gfortran reformatfault.f90 -o format
./format
```

block_boundaries.txt should now look like this:

```
-69.34939988   -35.99812067   2
-69.32798473   -36.1515959    3
>              4
-68.27507324   -35.69176506   5
-68.50418797   -36.15644652   6
>              7
-69.55641299   -35.66618586   8
```