GeoSAMS root: Getting Started

NOTE: Directory delimiter - on UNIX, Lynux, or MAC OS swap ‘\’ for ‘/’

1. Pull Source Code and Data files, from within your working directory. The user may then manually run the commands in the following steps or run a batch script to automatically set up everything

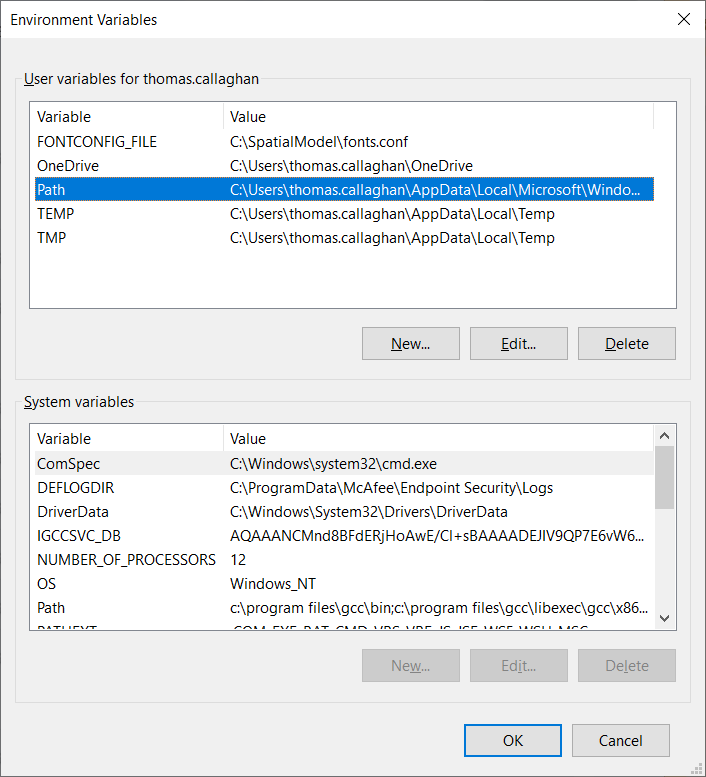
* git clone https://github.com/NEFSC/READ-PDP-GeoSAMS.git <GeoSamsRoot>
* cd <GeoSamsRoot>
* WIN
  + .\GeoSamSetup.bat YearStart YearEnd
* MAC, Unix
  + Edit .octaverc to point where Octave packages are installed
    - addpath ~/.local/share/octave/api-v58/packages/geometry-4.0.0
    - addpath ~/.local/share/octave/api-v58/packages/mapping-1.4.2/
    - addpath ~/.local/share/octave/api-v58/packages/io-2.6.4/
  + chmod 744 GeoSamSetup.sh
  + ./GeoSamSetup.sh YearStart YearEnd

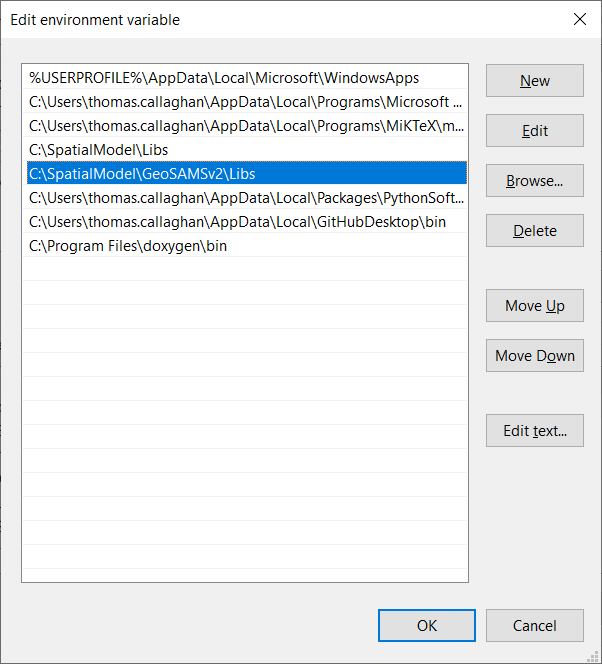
1. Unzip Dredge Data

* cd <GeoSamsRoot>\OriginalData
* WIN: "C:\Program Files\7-Zip\7z" e dredgetowbysize7917.zip  
  Other: unzip dredgetowbysize7917.zip

1. Add Libs to path

* Settings > Environment variables for you account  
  Select Path and click Edit



Select new and add <GeoSamsRoot>\Libs to the list, then click OK  


1. Build Execs

* cd <GeoSamsRoot>\UKsrc
* make setup; creates mod and obj subdirectories
* make
* cd <GeoSamsRoot>\SRC
* make setup; creates mod and obj subdirectories and directories used by ScallopPopDensity
* make

1. in Matlab/Octave

* cd <GeoSamsRoot>
* addpath mfiles
* addpath mfiles/latlonutm
* addpath PreProcess

**In Octave** may also need path to installed packages

* pkg list
* addpath ~/packages/geometry-x.y.z, where x,y,z is installed version
* addpath ~/packages/io-x.y.z
* addpath ~/packages/mapping-x.y.z

**In Matlab** one can simply enter

* cd <GeoSamsRoot>
* addpath (genpath(<GeoSamsRoot>))

Run:

>>TrawlData5mmbin

**TrawlData5mmbin.m** takes the trawl data from “*OriginalData/dredgetowbysize7917.csv*” and makes a csv, e.g."*Data/bin5mmYYYY[MA|GB].csv*". The output has the dredge observations in 5mm bins for each year and domain. 1979 to 2017

>>PullOutRecruitData

**PullOutRecruitData.m** for each location it sums together the scallop density from shell length 3cm to 6 cm, inclusive. It then adds this value as a new column along with the current data for size grp 4 as a single row for the location and writes this out to "OriginalData/NewRecruits.csv".

>>ProcessRec**r**uitData

**ProcessRecruitData.m** take the recruit estimates from " *OriginalData/NewRecruits.csv* ":

(A) Convert to scallops per m2 🡪"Data/RecruitsUnadjusted.csv"

(B) Convert to scallops per m2, adjusted for dredge under-sampling 🡪 "*Data/RecruitsRockStrataAdjustment.csv*". This assumes 40% detection generally and 27% detection in rock chain strata.

(C) Split data into GB and MA segments based on a dividing line at *longitude* = -70:5. Add UTM19 coordinates for GB and UTM18 coordinates for MA. 🡪 "*Data/RecruitsGB.csv","Data/RecruitsMA.csv*"

(D) Split data into individual years 1979-2019. 🡪 "*Data/RecruitsYYYY[MA|GB].csv*"

This script produces

* RecruitsYYYY[MA|GB].csv
* Recruits[MA|GB].csv
* Recruits[MA|GB]Lump1NM.csv
* RecruitsRockStrataAdjustment.csv
* RecruitsUnadjusted.csv

>>NearestNeighborRecInterp(yrStart, yrEnd)

**NearestNeighborRecInterp.m:** Loops through both ‘MA’ and ‘GB’ domain names, DN. Uses the year in yrStart=YYYY to establish the size of locations from “*Data/RecruitsYYYYDN.csv*”. It then writes the recruit estimates to “*KriginingEstimates/SimDNYYYY/KrigingEstimate.txt*”. For each subsequent year from yrStart+1 to yrEnd it will read data from “*Data/RecruitsYYYYDN.csv*” and using a nearest neighbor approach align this data to the size of locations initially established and writes the resulst to “*KriginingEstimates/SimMAYYYY/KrigingEstimate.txt*”.

**TODO:** There exists many apparently unused files in SimDN subdirectories.Determine if they are necessary and perhaps reduce the output to single file for each year and domain name, i.e. *“KriginingEstimates/KrigingEstimateYYYYDN.txt”.*

NOTE: SetUpICS() is used to create InitialCondition.csv. This is no longer used for forecasting scallop growth.

>> SetUpICS(2000,’MA’)

>> SetUpICS(2000,’GB’)

1. $ cd <GeoSamsRoot>

If necessary edit UK.inp for interpolation / sampling parameters.

1. $ Python3|python PythonScripts/EstimateRecruitFields.py StartYear EndYear

Reads in Data/RecruitsYYYY[MA | GB].csv

This will generate data for directory KrigingEstimates/Sim[MA | GB]YYYY with files for each year given on the command line: YYYY = StartYear to EndYear

**NOTE: Don’t run if using NearestNeighborRecInterp**

**TODO:** There exists many apparently unused files in SimDN subdirectories.Determine if they are necessary and perhaps reduce the output to single file for each year and domain name, i.e. *“KriginingEstimates/KrigingEstimateYYYYDN.txt”.*

* beta.txt
* CovBeta.csv
* DIntV.txt
* epsilon.txt
* GammaIntV.txt
* KrigingEstimate.txt
* KRIGpar.txt
* KrigSTD.txt
* NLSFpar.csv
* OLSresidual.txt
* RandomField1.txt
* RandomField2.txt
* RandomField3.txt
* RandomField4.txt
* RandomField5.txt
* RandomField6.txt
* RandomField7.txt
* RandomField8.txt
* RandomField9.txt
* RandomField10.txt
* residuals.csv
* SIntV.txt
* SpatialFunctions.csv
* SpatialTrend.txt

1. $ Python3|python PythonScripts/EstimateRecruitFieldsClim.py DEPRECATED

Reads in Data/ Recruits[MA | GB]Lump1NM.csv

This will generate data for directory KrigingEstimates/Sim[MA|GB]Clim with files

* beta.txt
* CovBeta.csv
* DIntV.txt
* epsilon.txt
* GammaIntV.txt
* KrigingEstimate.txt
* KRIGpar.txt
* NLSFpar.csv
* OLSresidual.txt
* residuals.csv
* SIntV.txt
* SpatialFunctions.csv
* SpatialTrend.txt

1. To run scallop population / dynamics software, should still be in <GeoSamsRoot>
2. Edit Configuration/Scallop.cfg for desired output

* # set largest expected number of grids
* Max Number of Grids = 500
* Domain Name = MA
* Beginning Year = 2005
* Ending Year = 2008
* Time steps per Year = 13
* # Fishing can be USD, BMS, or, CAS
* Fishing = USD

1. $ .\SRC\ScallopPopDensity.exe Scallop.cfg
2. PlotLatLonData(file name)

**PlotLatLonData.m:** Used to create a geoscatter plot of the data in file name. Assumes that data is arranged as

grid location, <--- grid parameter --->

**lat, lon, p1, p2, .... , pN**

1. <GeoSamsRoot>\Results

The first set presents a sum of the data at each grid location, for each time step. The second set presents the data by size for each time step in a separate file for each grid.

|  |  |  |
| --- | --- | --- |
| **FileName** | **Organized as** | **Equation** |
| **First Set** | | | |
| Abundance\_MA|GB.csv | #time steps by #grids | Abundance\_psqm \* grid\_area\_sqm |
| Abundance\_psqm\_MA|GB | #time steps by #grids | sum(state(grid,1:num\_size\_classes)) |
| BMS\_mt\_MA|GB.csv | #time steps by #grids | BMS\_mtpsqm \* grid\_area\_sqm |
| BMS\_mtpsqm\_MA|GB.csv | #time steps by #grids | dot\_product( state( grid, : ), weight\_grams(grid, :) / grams\_per\_metric\_ton) |
| ExplBMS\_mtpsqm\_MA|GB | #time steps by #grids | dot\_product(selectivity(grid, :) \* state(grid, :),  weight\_grams(grid,:)) |
| ExplBMS\_mt\_MA|GB | #time steps by #grids | ExplBMS\_mtpsqm \* grid\_area\_sqm |
| FishingEffortMA|GB | #time steps by #grids | Dependent on USD, BMS, CAS |
| FishingMortMA|GB | #time steps by #grids | C \* dot\_product(selectivity(grid, :) \* state(grid, :),  weight\_grams(grid,:)) |
| NumLandingsMA|GB | #time steps by #grids | sum[ (1.\_dp - exp(-F\_mort\_by\_loc(grid) \* delta\_time))  \* selectivity(grid, :) \* state(grid, :) \* grid\_area\_sqm ] |
| WgtLandings\_mtMA|GB | #time steps by #grids | dot\_product(sum[ (1.\_dp - exp(-F\_mort\_by\_loc(grid) \* delta\_time))  \* selectivity(grid, :) \* state(grid, :) \* grid\_area\_sqm ],  weight\_grams(grid,:))/grams\_per\_metric\_ton |
| **Second Set** | | | |
| StateMA|GB\_AtTime\_YYYY\_NNN | At time step NNN, #grids by # size classes | Current State in scallops per square meter |
| StateMA|GB\_AtGrid-NNNNN\_BySizeOverTime | At grid NNNNN, #timesteps by # size classes | Current State in scallops per square meter |
| TotalMortMA\_AtGrid-NNNNN\_BySizeOverTime | At grid NNNNN, #timesteps by #size classes | Natural\_mort(grid, :) + FishingEffort(grid) \* Selectivity(grid, :) + Incidental(grid) + Discard(grid, :) ) |
| **Third Set** | | | |
| X\_Y\_EBMS\_DN.csv | Each row is grid information starting with lat lon coordinates, followed by values as each time step | Used by Matlab to plot exploitable biomass in a Geoscatter plot |
| X\_Y\_Feffort\_DN.csv | Each row is grid information starting with lat lon coordinates, followed by values as each time step | Used by Matlab to plot fishing effort biomass in a Geoscatter plot |

mt: metric tons

psqm: per square meter

DN: Domain Name, ‘MA’ or ‘GB’