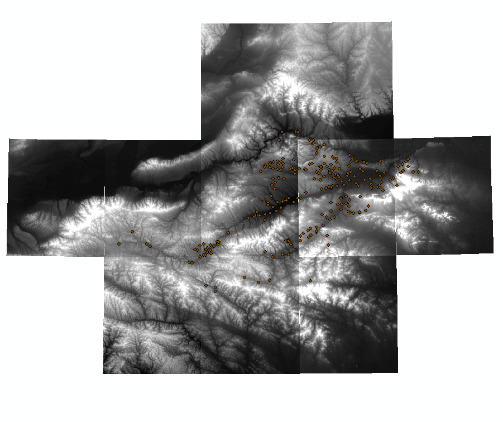
**Technical Process of the Study**

This document list the data and technics used for the study.

1. Download DEM data from the <http://www.gscloud.cn/sources/accessdata/310?pid=302>, the location of the area is around Lat 34 degree and Long 111 degree. This is the 30m DEM opensource data of China. A total of 10 images need to be download to cover the whole area



1. Add all the downloaded images to QGIS and use **merge** tool to combine them to one image
2. Add **studyareaboundary.shp** boundary map to QGIS and use the **Clipper** tool to extract the study area image as **DEMOnefit.tif**
3. Use the **aspect, slope** tools to get the aspect and slope raster: **aspect.tiff, slope.tiff**
4. Digitalize the Yiluo River Basin Archeological Site Atlas into **YangshaoCultures.shp, Xiashang.shp, LongshanCultures.shp**
5. Convert the above shapefile to raster as **Yangshao.tif, Xiashang.tif, Longshan.tif,** reclassify known archeological sites as 1 and others as 0
6. Select the major sites by attribute and save as **MajorSites.shp**
7. Add the **rivers.shp**, and **MajorSites.shp** to QGIS
8. Use the proximity tool to generate **DistancetoRivers.tif and DistancetoMajorsites.tif**
9. Download Rstudio and install R 4.1
10. Create the first R script **YangshaoModel.R** as Yangshao prediction model
    1. The following codes install the needed packages

*####need to install the packages if first time run the model, by uncomment the lines*

*# install.packages("Rcpp")*

*# install.packages("raster")*

*# install.packages("rgdal")*

*# install.packages("sp")*

*##########################*

*library(ggplot2)*

*library(raster)*

*library(rgdal)*

* 1. The following codes load the raster data to R

*##*

*archeologicalsites <- raster("Yangshao.tif")*

*elevation <- raster("DEMOnefit.tif")*

*slope <- raster("Slope.tif")*

*aspect <- raster("Aspect.tif")*

*distoriver <- raster("DistancetoRivers.tif")*

*distomajorsites <- raster("DistancetoMajorsites.tif")*

* 1. The following codes plot out the elevation, slope and aspect data

*Raster\_vars <- stack (elevation, slope, aspect)*

*plot (Raster\_vars)*

* 1. The following codes select 80% of the known sites from the known cells

*##convert to spatial points data frame*

*archeologicalsitespoints <- as(archeologicalsites, "SpatialPointsDataFrame")*

*#separate the dataset as truepoints and false poitns, truepoints means identified archeological sites*

*truepoints = subset(archeologicalsitespoints, archeologicalsitespoints[[1]] == 1)*

*#sample true points*

*selectedtruepoints <- sample(1:length(truepoints[[1]]), 0.8 \* length(truepoints))*

*truepointframe= truepoints[selectedtruepoints,]*

* 1. the following codes select same amount of unknow sites from the unknown cells

*#falsepoints means identified nonarcheological sites*

*falsepoints = subset(archeologicalsitespoints, archeologicalsitespoints[[1]] == 0)*

*#sample false points*

*selectedfalsepoints <- sample(1:length(falsepoints[[1]]), 0.8 \* length(truepoints))*

*falsepointframe= falsepoints[selectedfalsepoints,]*

* 1. the following codes extract the corresponding independent variable values

*##true sites list*

*tarcheologicalsites <- raster::extract(archeologicalsites, truepointframe)*

*televation <- raster::extract(elevation, truepointframe)*

*tslope <- raster::extract(slope, truepointframe)*

*taspect <- raster::extract(aspect, truepointframe)*

*tdistoriver <- raster::extract(distoriver, truepointframe)*

*tdistomajorsites <- raster::extract(distomajorsites, truepointframe)*

*##false sites list*

*farcheologicalsites <- raster::extract(archeologicalsites, falsepointframe)*

*felevation <- raster::extract(elevation, falsepointframe)*

*fslope <- raster::extract(slope, falsepointframe)*

*faspect <- raster::extract(aspect, falsepointframe)*

*fdistoriver <- raster::extract(distoriver, falsepointframe)*

*fdistomajorsites <- raster::extract(distomajorsites, falsepointframe)*

* 1. the following codes combine the known and unknown data together as trainingdata

*##combine the two datasets together*

*trainarcheologicalsites <- append(tarcheologicalsites,farcheologicalsites)*

*trainelevation <- append(televation,felevation)*

*trainslope <- append(tslope,fslope)*

*trainaspect <- append (taspect, faspect)*

*traindistoriver <- append (tdistoriver, fdistoriver)*

*traindistomajorsites <- append (tdistomajorsites, fdistomajorsites)*

*## put data to data frame*

*trainingdata <- data.frame(trainarcheologicalsites,trainelevation,trainslope,trainaspect,traindistoriver,traindistomajorsites)*

* 1. the following codes is the Logistic regression model and summary results

*## logistic regression*

*apredictmodel <- glm(formula = trainarcheologicalsites ~ trainelevation + trainslope + trainaspect + traindistoriver + traindistomajorsites,*

*data=trainingdata, family = binomial)*

*Summary(apredictmodel)*

1. Following the same method ,create the first R script **XiashangModel.R** as Xiashang prediction model
2. Following the same method ,create the first R script **LongshanModel.R** as Longshan prediction model
3. Get the model results as list in the Table 3, table 4 and table 5
4. Use the constant value and the beta values of each model to create the probability raster by using the **Raster Calculator tool**
   1. Constant + beta1 \* elevation + beta2 \* slope + beta3 \* aspect + beta4 \* distorivers + beta5 \* distomajorsites
5. Compare the predict rasters with corresponding culture site rasters to evaluate the accurate rate