1. **Input file preparation**: 4DB convert phenotype spreadsheet with transect\_posi(position), allele frequency, number of recordings.

###Code in Hzar cline analysis####

fwdata <- hzar.doMolecularData1DPops(transect\_data$position,

transect\_data$ratio\_fw,

transect\_data$total\_fw)

1. **Transect files 1st make**: Transects were selected and extracted based on highest and lowest allele frequency between blocks on maps.

**##Code for defining transects in script “Allele frequency maps and Transects file made for initial cline analysis”.**

transects <- as.data.frame(rbind(

c(lon\_start=-14, lat\_start=16, lon\_end=38, lat\_end=-8), #Central Africa

c(lon\_start=26, lat\_start=-32, lon\_end=42, lat\_end=8), #South East Africa

c(lon\_start=10, lat\_start=12, lon\_end=18, lat\_end=-28),#South West Africa

c(lon\_start=-18, lat\_start=12, lon\_end=2, lat\_end=44), #North West Africa

c(lon\_start=102 , lat\_start=0, lon\_end=122, lat\_end=-12)))#South East Asia

**##Code for define transect width**

## 750 is default, in the paper we submit is 450 which we think is more proper and could get rid of strange blocks( like a block with low frequency in the area that should be of high frequency).

max\_dist = 750

**##Code for define block colour**

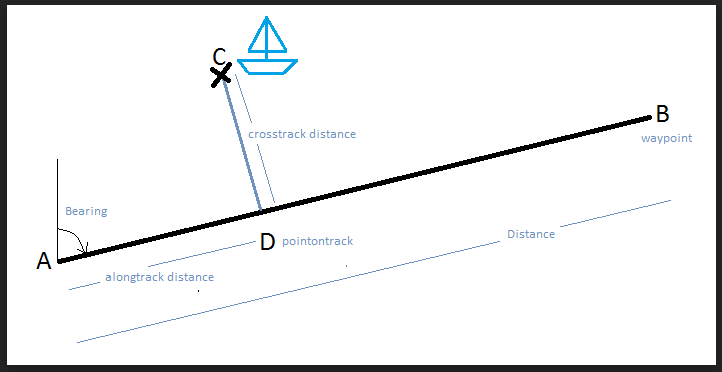
cols = c("BLUE", "GREEN", "YELLOW","ORANGE","RED")

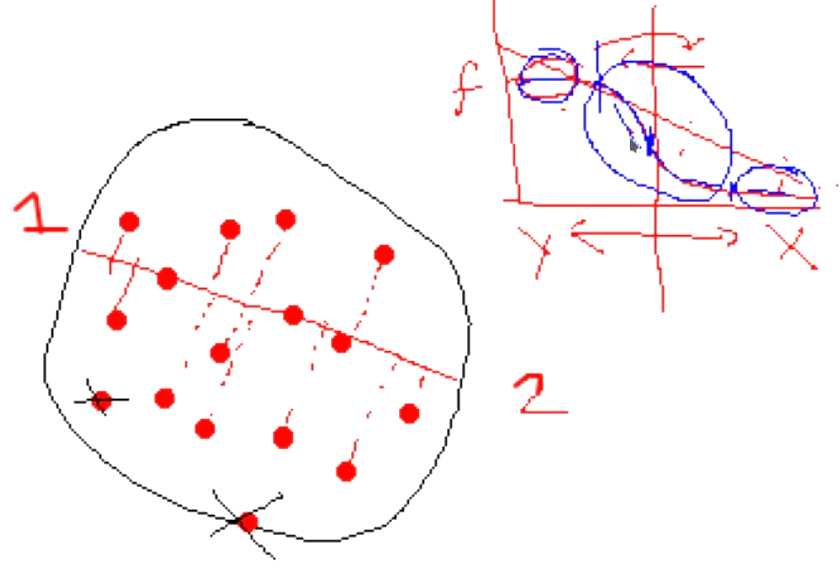
**### Transect meaning####**

# For each transect, we select which blocks are close enough to the transect.

# To do this, we calculate the perpendicular distance of each block to the transect.

# This is called the cross-track distance or distance to grand circle





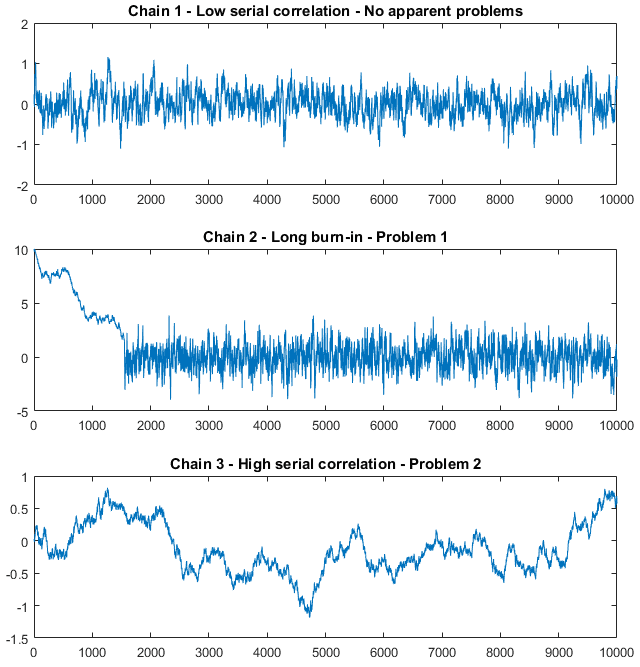
1. **Cline plots 1st make**: To make a cline plot to see if the selected transect fits for cline analysis or whether points(allele frequency of each 4DB block) are close to the cline line or scatter randomly. Input files come from **step 2**.

### **Parameters “scaling” and “tails”** will change the shade of the Cline line, it will be mentioned when selecting the best cline model.####

### For detailed information, you could read Hzar packages manual. (<https://cran.r-project.org/web/packages/hzar/hzar.pdf>) #####

gwmodel <- hzar.makeCline1DFreq(hwdata, scaling="fixed",tails="none");

### **Trace result** explanation, best cline model should be of Trace like Chain 1



####**Cline width and Cline center**#######

### code to show cline center range

abline(v=hzar.getLLCutParam(gwmodelData,"center"))

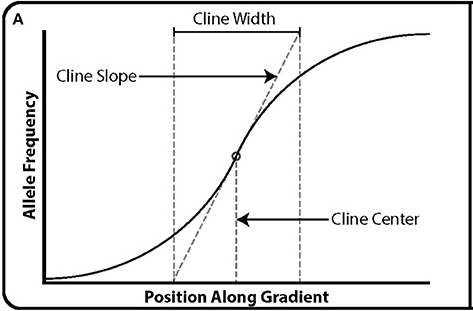
### Specific number of Cline width and cline center

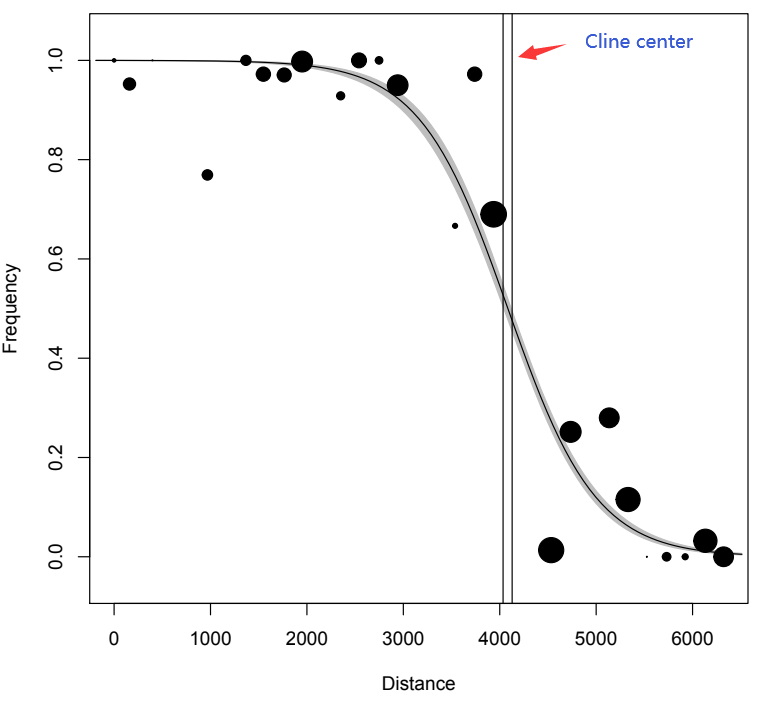
In log file, it showed a range of cline width and center like this,

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

center2LLLow center2LLHigh width2LLLow width2LLHigh

1 4034.527 4128.599 1706.467 1967.09





1. **Repeat** step 2 and step 3 until you find the best transects.

#### Examples file “” on transect selection. #####

1. **Best cline model selection**: this script is edited based on an example script “men12209-sup-0004-hzarExampleScript.R” of Hzar paper “hzar: hybrid zone analysis using an R software package” (<https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2F1755-0998.12209&file=men12209-sup-0004-hzarExampleScript.R>).

#### **Six cline models** were separated by parameters **“scaling” and “tails”**.

mkn.loadAdaAmodel("none","none","model01");

mkn.loadAdaAmodel("none" ,"both","model02");

mkn.loadAdaAmodel("free","none","model03");

mkn.loadAdaAmodel("free" ,"both","model04");

mkn.loadAdaAmodel("fixed","none","model05");

mkn.loadAdaAmodel("fixed" ,"both","model06");

##

## Just to be thorough, randomize the initial value for each fit

## runif(45,0,4530) center for 15 models

###runif(n, a, b) generates n uniform random numbers between a and b.

# > runif(18,0,4530)

# [1] 510.6717 3967.7692 2429.3038 256.3456 773.1928 1349.7096 3415.9907

# [8] 3022.3552 1504.6555 170.9729 1424.3317 3996.5568 4058.4039 4077.5656

# [15] 1392.7833 3898.4173 4503.2579 230.6215

mkn$AdaA$fitRs$chains[[1]]$modelParam$init["center"]= 510.6717

mkn$AdaA$fitRs$chains[[2]]$modelParam$init["center"]= 3967.7692

mkn$AdaA$fitRs$chains[[3]]$modelParam$init["center"]= 2429.3038

# mkn$AdaA$fitRs$chains[[4]]$modelParam$init["center"]= 256.3456

# mkn$AdaA$fitRs$chains[[5]]$modelParam$init["center"]= 773.1928

# mkn$AdaA$fitRs$chains[[6]]$modelParam$init["center"]= 1349.7096

# mkn$AdaA$fitRs$chains[[7]]$modelParam$init["center"]= 3415.9907

# mkn$AdaA$fitRs$chains[[8]]$modelParam$init["center"]= 3022.3552

# mkn$AdaA$fitRs$chains[[9]]$modelParam$init["center"]= 1504.6555

# mkn$AdaA$fitRs$chains[[10]]$modelParam$init["center"]= 170.9729

# mkn$AdaA$fitRs$chains[[11]]$modelParam$init["center"]= 1424.3317

# mkn$AdaA$fitRs$chains[[12]]$modelParam$init["center"]= 3996.5568

# mkn$AdaA$fitRs$chains[[13]]$modelParam$init["center"]= 4058.4039

# mkn$AdaA$fitRs$chains[[14]]$modelParam$init["center"]= 4077.5656

# mkn$AdaA$fitRs$chains[[15]]$modelParam$init["center"]= 1392.7833

# mkn$AdaA$fitRs$chains[[16]]$modelParam$init["center"]= 3898.4173

# mkn$AdaA$fitRs$chains[[17]]$modelParam$init["center"]= 4503.2579

# mkn$AdaA$fitRs$chains[[18]]$modelParam$init["center"]= 230.6215

# > runif(18,0,4530)

# [1] 3289.7368 2548.5611 2767.6771 831.7151 1691.7406 2970.8066 978.3676

# [8] 823.7172 2345.6927 1878.7275 2827.3355 3608.6201 3913.1552 3574.2020

# [15] 795.0458 2398.6763 145.7594 3288.5337

mkn$AdaA$fitRs$chains[[1]]$modelParam$init["width"]= 3289.7368

mkn$AdaA$fitRs$chains[[2]]$modelParam$init["width"]= 2548.5611

mkn$AdaA$fitRs$chains[[3]]$modelParam$init["width"]= 2767.6771

# mkn$AdaA$fitRs$chains[[4]]$modelParam$init["width"]= 831.7151

# mkn$AdaA$fitRs$chains[[5]]$modelParam$init["width"]= 1691.7406

# mkn$AdaA$fitRs$chains[[6]]$modelParam$init["width"]= 2970.8066

# mkn$AdaA$fitRs$chains[[7]]$modelParam$init["width"]= 978.3676

# mkn$AdaA$fitRs$chains[[8]]$modelParam$init["width"]= 823.7172

# mkn$AdaA$fitRs$chains[[9]]$modelParam$init["width"]= 2345.6927

# mkn$AdaA$fitRs$chains[[10]]$modelParam$init["width"]= 1878.7275

# mkn$AdaA$fitRs$chains[[11]]$modelParam$init["width"]= 2827.3355

# mkn$AdaA$fitRs$chains[[12]]$modelParam$init["width"]= 3608.6201

# mkn$AdaA$fitRs$chains[[13]]$modelParam$init["width"]= 3913.1552

# mkn$AdaA$fitRs$chains[[14]]$modelParam$init["width"]= 3574.2020

# mkn$AdaA$fitRs$chains[[15]]$modelParam$init["width"]= 795.0458

# mkn$AdaA$fitRs$chains[[16]]$modelParam$init["width"]= 2398.6763

# mkn$AdaA$fitRs$chains[[17]]$modelParam$init["width"]= 145.7594

# mkn$AdaA$fitRs$chains[[18]]$modelParam$init["width"]= 3288.5337

#### log result file of 03\_Best\_cline\_model\_selection.r script would show **best cline model and range of cline center and width**, like the following content.

[1] "model02"

center2LLLow center2LLHigh width2LLLow width2LLHigh deltaL2LLLow

1 2465.566 2760.842 723.8151 1650.174 255.6105

deltaL2LLHigh tauL2LLLow tauL2LLHigh deltaR2LLLow deltaR2LLHigh

1 624.0435 8.200318e-07 0.04419374 91.73974 1745.575

tauR2LLLow tauR2LLHigh

1 8.094048e-05 0.9850081

They based on codes:

## Do model selection based on the AICc scores

print(mkn$AdaA$analysis$AICcTable <-

hzar.AICc.hzar.obsDataGroup(mkn$AdaA$analysis$oDG));

## Print out the model with the minimum AICc score

print(mkn$AdaA$analysis$model.name <-

rownames(mkn$AdaA$analysis$AICcTable

)[[ which.min(mkn$AdaA$analysis$AICcTable$AICc )]])

1. **Cline center calculation and plot**

To convert Cline center information into coordinate information based on transect coordinates and Cline center distance. **Distance, initial bearing between transects and destination points of low and high cline centers** were calculated at https://www.movable-type.co.uk/scripts/latlong.html, or to apply the package of this website mentioned above.

<https://drive.google.com/drive/folders/1KuI0-KQBde0vySZlmKKdZ_Qu7bIA7EyZ?usp=sharing>