The Manual for AncestryPainter 2.0

Shuanghui Chen

Contents

AncestryPainter

1 Introduction

AncestryPainter is a graphing tool developed by Qidi Feng and Dongsheng Lu in 2018 (https://doi.org/10.1016/j.gpb.2018.05.002). It can visualize the ancestry composition and genetic difference, and merge ancestry proportion matrix output by ancestry inference tools like ADMIXTURE.

Since Aug, 2021, a new version of AncestryPainter has been developed by Shuanghui Chen with her collaborators. Now this project is still ongoing.

See DESCRIPTION for more details about our team!

2 Dependencies and installation

2.1 Dependencies

- Software

```
R 3.3.3 "Another Canoe".
```

- R packages

The graphing and statistical functions of AncestryPainterV2 are achieved by invoking these attached base packages of R.

```
graphics
grDevices
stats
utils
```

Generally, if you install `R` on your device, it is no need to install or load these packages additionally.

2.2 Installation

You can install our software by an R tool "devtools" given good network connectivity.

```
> if (!require("devtools", quietly = TRUE))
    install.packages("devtools")

> devtools::install_github("Shuhua-Group/AncestryPainterV2")
```

Alternatively, you can obtain the compressed package manually, and install it by R command like:

```
install.packages("/path/to/the/package/AncestryPainterV2.x.y.tar.gz")
```

Substitute "x" or "y" by the version ID of the `AncestryPainterV2` binary package you downloaded from GitHub or other source.

To check whether it is successfully installed, try loading this package like

```
library("AncestryPainterV2")
```

3 Functions and main parameters

This part includes functions and parameters of AncestryPainter. Main parameters that can be used most frequently, are further explained, including those regarding input data, layout of plot, major plotting elements, etc.

We strongly suggest that users save information used as arguments, like ancestry proportion/genetic distance/individual and group/color code, in tab/space-delimited text files and read these information into R environments.

For more details, please refer to the R documentation (in ".Rd" files in the `man` folder, or search it by "Help" on RStudio) of AncestrypainterV2.

3.1 sectorplot

```
sectorplot(Q, ind, target = NULL, poporder = NULL, ancescols = NULL,
  sorting = FALSE, rmin = 2, rmax = 3.7, tar.r = 0.6,
  tarangs = NULL, cendis = 1, amin = -265, amax = 85,
  tarang1 = 0, tarang2 = 180, arrow = FALSE, legend_mode = FALSE,
  ancesnames = NULL, prgap = 0.2, noline = FALSE,
  pop.lab.cex = NULL, pop.lab.col = "black", pop.lab.font = 1,
  tar.lab.cex = 6, tar.lab.col = "navy", arrow.col = "red",
  arrow.lwd = 2, legend.pos = "topright")
```

Q
A numeric data frame of ancestry proportion (columns: ancestry component; rows: individual). e.g., an output ".Q" file of the software ADMIXTURE.

ind
A two-column data frame (1: population; 2: individual).

target
Character. The target populations to be plotted as a pie chart in the center of the circle figure. The population must be included in input "ind" and "Q" data frame.

poporder
Character. The populations to be included in the figure, also the display order of the populations in the figure.

ancescols
The color code of each ancestry component in the figure.

sorting
A logical value to define whether to sort the order of the populations, which will be masked if "poporder" is specified.

rmin
The radium of the inner ring. Default is 2.

rmax
The radium of the outer ring. Default is 3.7.

tarangs
A numeric vector. The angles of the target pie charts.

cendis
A numeric vector. The distance from the center of a target pie chart to the center of the sectorplot. Default: 1.

amin
The angle at which the ring is initiated. Default is -265.

amax
The angle at which the ring is ended. Default is 85.

tarang1
The start angle of the target layout. Default is 0.

tarang2
The end angle of the target layout. Default is 360.

arrow
Logical. Whether to draw the arrows to the target pies.

legend_mode
Logical. Whether to draw the legend of ancestry components.

ancesnames
Character. To specify the names of ancestry components. If not specified, would be shown as "Ancestry_1" "Ancestry_2" and so on.

noline
Logical. Whether to remove the black lines between populations. Default is FALSE.

3.2 radiationplot

```
radiationplot(data, target = "target", sorting = FALSE,
  layers = NULL, num = 4, digits = 2, cenvals = c(0.5, 0.5),
  border = 0.3, amax = -250, amin = 70, rstart = 0.02,
  flat = 1.2, label_mode = TRUE, pop.lab.cex = 0.7,
  pop.lab.font = 1, pop.lab.col = "navy", legend_mode = FALSE,
  legend.pos = "topright", legend.lwd = 5, legend.text.cex = 1,
  ring.text.col = "black", ring.text.cex = 1, ring.text.font = 1,
  ring.line.col = "gray", tar.lab.col = "black", tar.lab.cex = 1.1,
  tar.lab.font = 1, core.line.col = "black")
```

data
A four-column data frame. 1: population label (character); 2: region label (character); 3: genetic difference (numeric); 4: color code.

target
A character value to pass name of the target population. Default is "target".

sorting
Logical. Whether to sort population order according to their genetic difference. Default is FALSE.

layers
Numeric values of layers. This parameter will mask "num" if specified.

num
An integer. Default is 4. Layer number.

digits
A float number. The decimal space that the genetic distance should be rounded to. Default is 2.

amax
Numeric. Maximal angle of radiation bars. Default is -250.

amin
Numeric. Minimal angle of radiation bars. Default is 70.

rstart
A numeric value of radius. Default is 0.02.

label_mode
Logical. Whether to print the population labels. Default is TRUE.

legend_mode
Logical. Whether to draw the legend of region information. Default is FALSE.

3.3 Parameters for aesthetics

The naming style of this group of parameters imitates what is used in the R package "ggplot2". The name of an parameter includes two or three parts. The prefix indicates a certain subject in the plot (e.g., "tar" "legend"), and the suffix, the feature that can be modified (e.g., "col""pos""font""cex"). In some cases, there is a middle part between the two parts, which means a graphing element of the subject.

For example, the parameter `ring.text.col` of `radiationplot` indicates the color of the text annotating outer rings

Again, see the R documentation (in ".Rd" files in the `man` folder, or search it by "Help" on RStudio) of AncestryPainterV2 for more details!

3.4 ancmerge

```
ancmerge(tar_anc_filelist, ref, K, poporder = NULL)
```

tar_anc_filelist
Character. Required. ancestry file names (recommended name format: prefix.K.ancestry). The ancestry matrix file should be(2 + K) columns without header. The columns: 1st-Individual ID; 2nd-Group ID. From the 3rd column, it indicates the ancestry proportion.

ref
Character. Required. The reference ancestry matrix to be matched.

K
Integer. Required. The number of the ancestry components.

poporder
Character. (optional, input files) Population order list.

4 Step-by-step guidance for usage

4.1 Sectorplot

AncestryPainter implements `sectorplot` to visualize the ancestry composition of multiple populations.

4.1.1 Required input The users of our software have to provide 1) an ancestry matrix with rows as individuals and columns as ancestry proportion and 2) the annotation including individual ID and group ID. Information for 1) and 2) can be saved in a tab/space delimited text files ".Q" and ".ind", respectively.

```
> # input
> exp_q <- read.table("./inst/extdata/exp_ances.K8.1.Q", header = F)
> exp_ind <- read.table('./inst/extdata/exp_ances.K8.1.ind', stringsAsFactors = F, header = F)
> exp_q[201:205, 1:5]
          V1       V2       V3       V4    V5
201 0.113863 0.071362 0.018369 0.012654 1e-05
202 0.149520 0.068620 0.017565 0.017855 1e-05
203 0.129546 0.076118 0.054619 0.015374 1e-05
204 0.229236 0.069375 0.030631 0.009024 1e-05
205 0.763557 0.115103 0.001776 0.011903 1e-05
>
> head(exp_ind)
          V1       V2
1 Australian  IHW9195
2 Australian  IHW9193
3 Australian  IHW9118
4    Chukchi ADR00060
5    Chukchi ADR00065
6    Chukchi    MC_06
```

4.1.2 Basic functions / Let's start Users can specify the color code of ancestry components if they like. If not, the colors will be randomly generated.

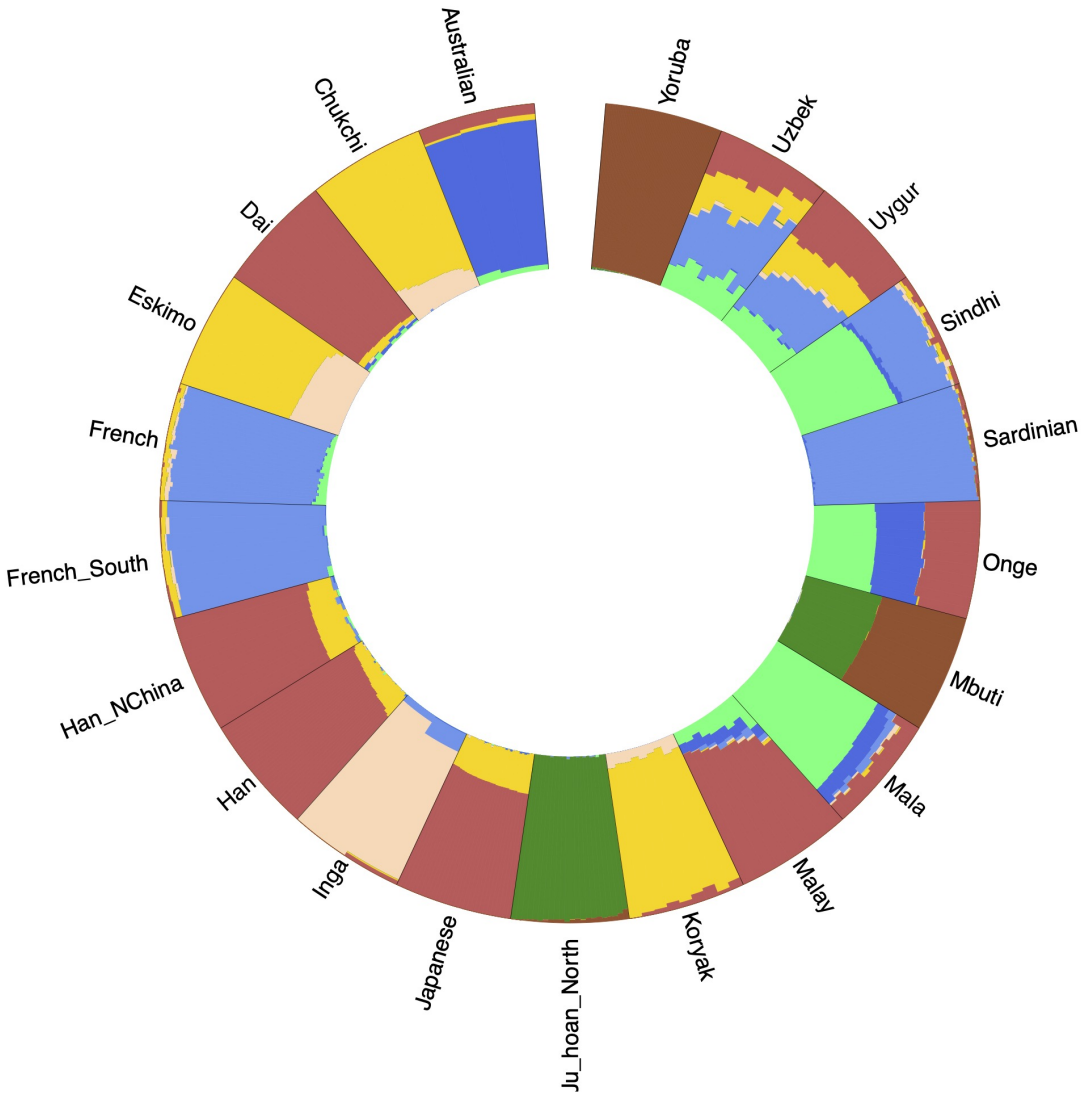
You can input the data manually. However, it would be convenient if the data are saved in files and we strongly recommend that you do so.

Now let's start with a toy sample. Let's specify the color code of ancestry.

```
> exp_cols <- read.table('./inst/extdata/exp_ances.K8.1.color',
stringsAsFactors = F, header = F)$V1
```

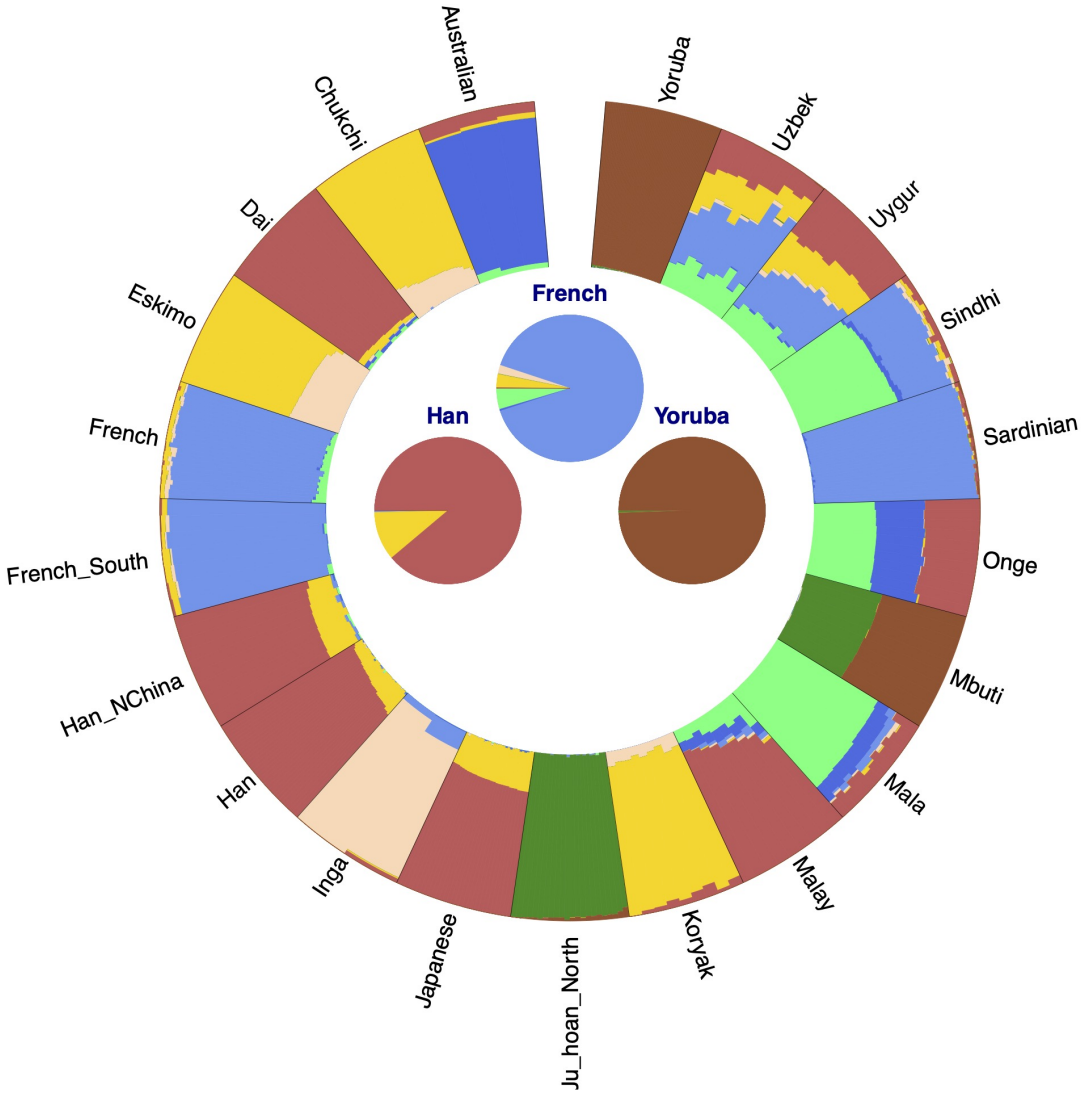
Note that the figure may be so large that it cannot be shown in the "Plots" Session of RStudio on some devices. The error "Figures margin too large" may appear. To get over this, you'd better print your plot in graph files (".pdf", ".png", "jpg", etc), like

```
> # Graphing
> pdf("exp_ances.8.basic.pdf", width = 45, height = 45)
> sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols)
No population order specified.
> dev.off()
```

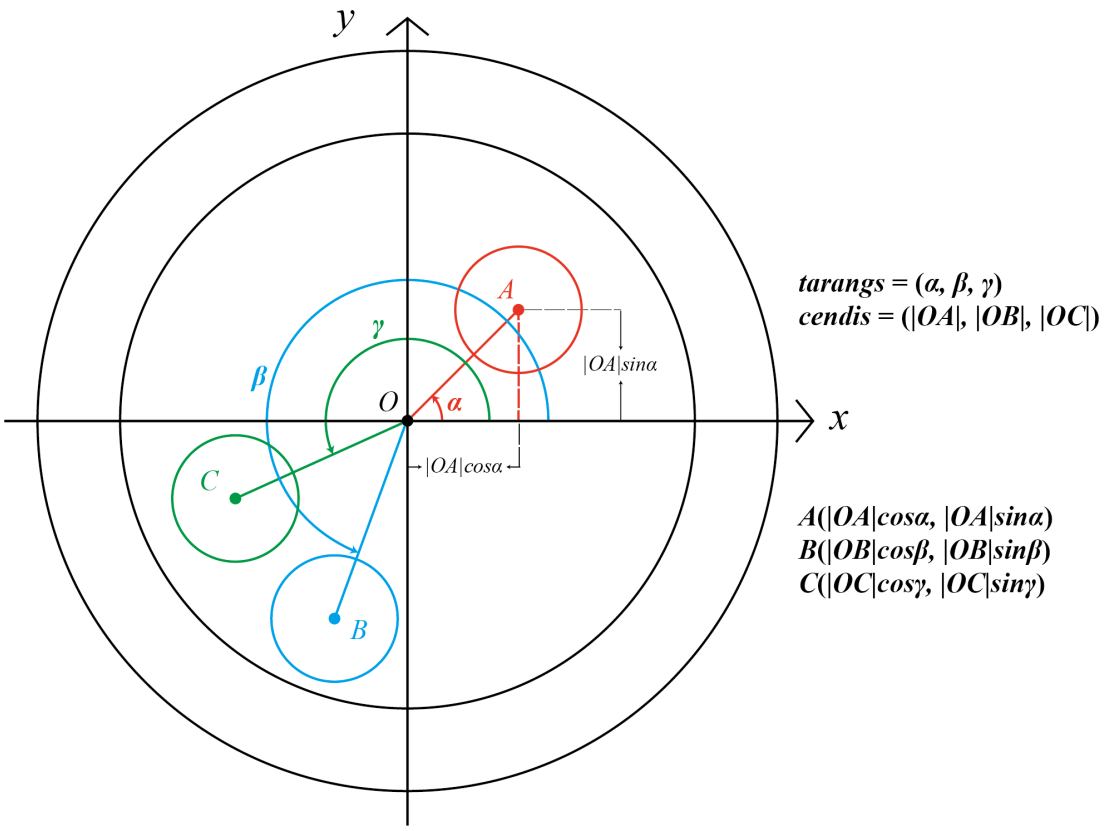


4.1.3 Multiple centric pies with a modified layout An important feature of our software is to display the ancestry composition of the target population(s) by pie charts in the center of the plot. To make use of this feature, you can specify the "target" parameter. In contrast to Version 1, Version 2 supports multiple target pie charts.

```
> exp_tars <- c("Yoruba", "French", "Han")
> sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols, target = exp_tars)
```

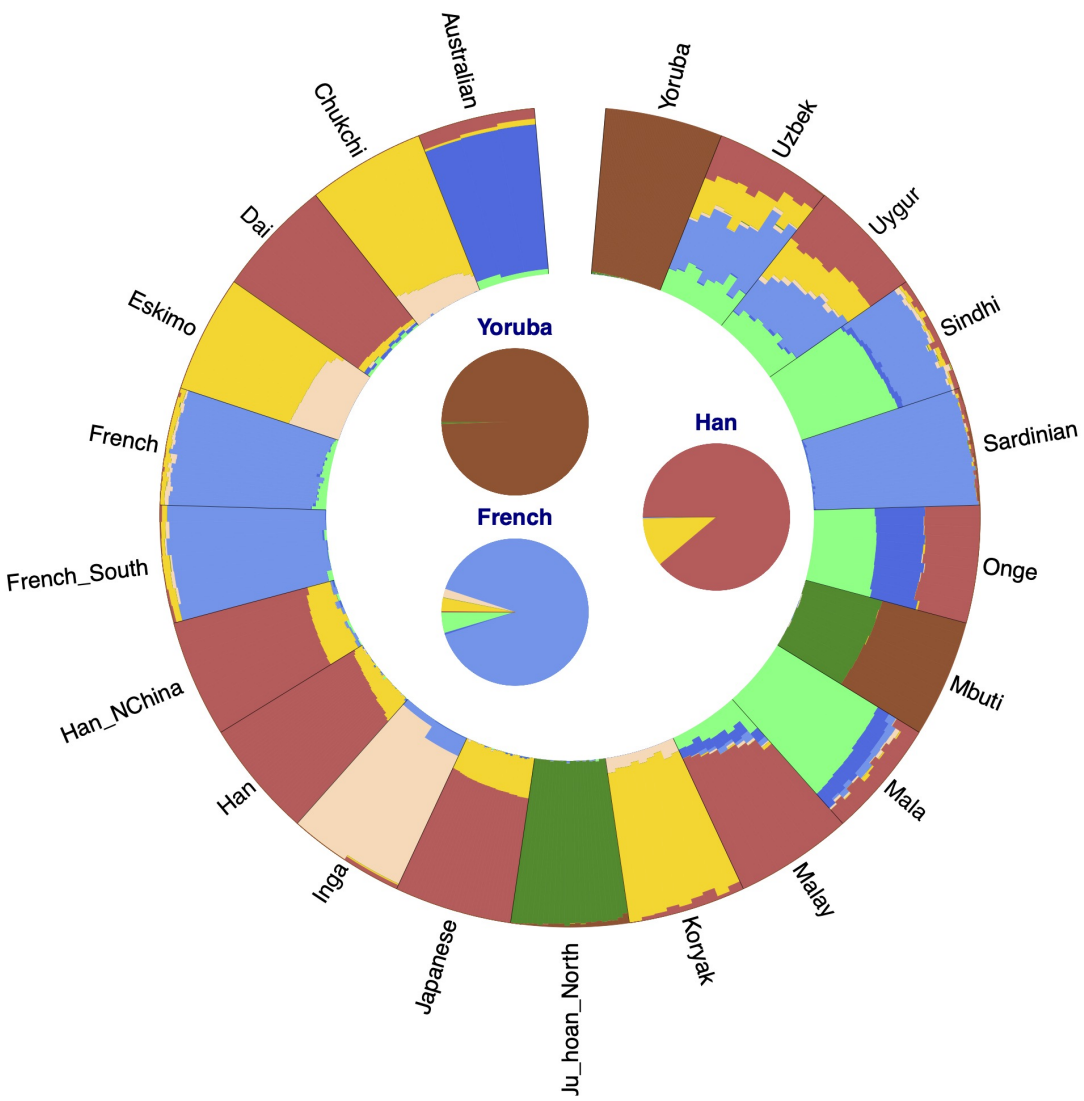


You can modify the layout of the target pies via two parameters "tarangs" (which defines the angle between the line from the center of the sectorplot to the center of a target pie chart and the positive x-axis) and "cendis" (which defines the distance from the center of any target pie chart, like A, B, C in the figure below, to the center of the sectorplot).

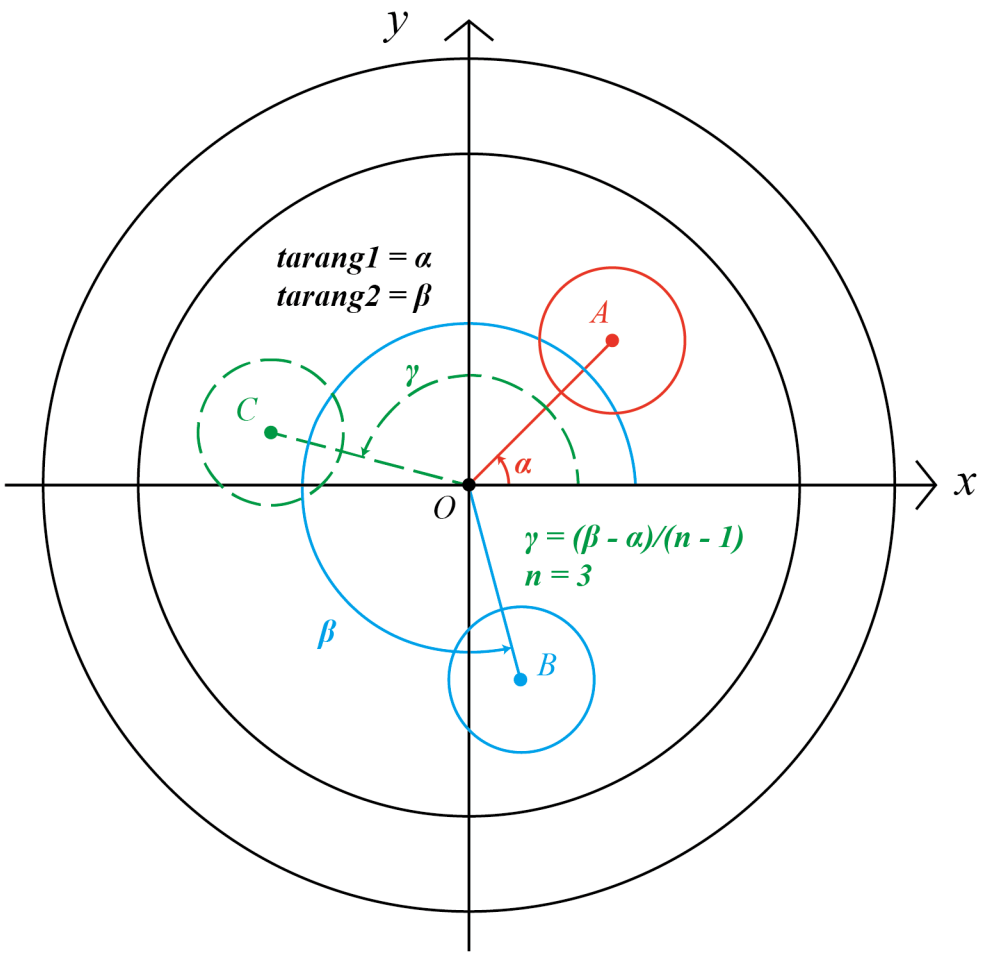


Generally, you can input two numeric vectors with the same length. If they are not of the same length, the shorter one will be coerced to match the longer one by extension and a warning message will be thrown.

```
> sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols, target = exp_tars,
tarangs = c(120, 240, 360), cendis = c(0.9, 0.9, 1.2))
```



To make it more convenient, you can use two parameters “tarang1” and “tarang2”. Given n target pie charts in the plot, the parameters “tarang1” defines the angle between the line from the center of the sectorplot to the center of the first target pie chart, and “tarang2”, that to the center of the last target pie chart. For the nth target pie chart, its center is located at the line with an angle as (tarang2 - tarang2) / (n - 1) from the positive x-axis.

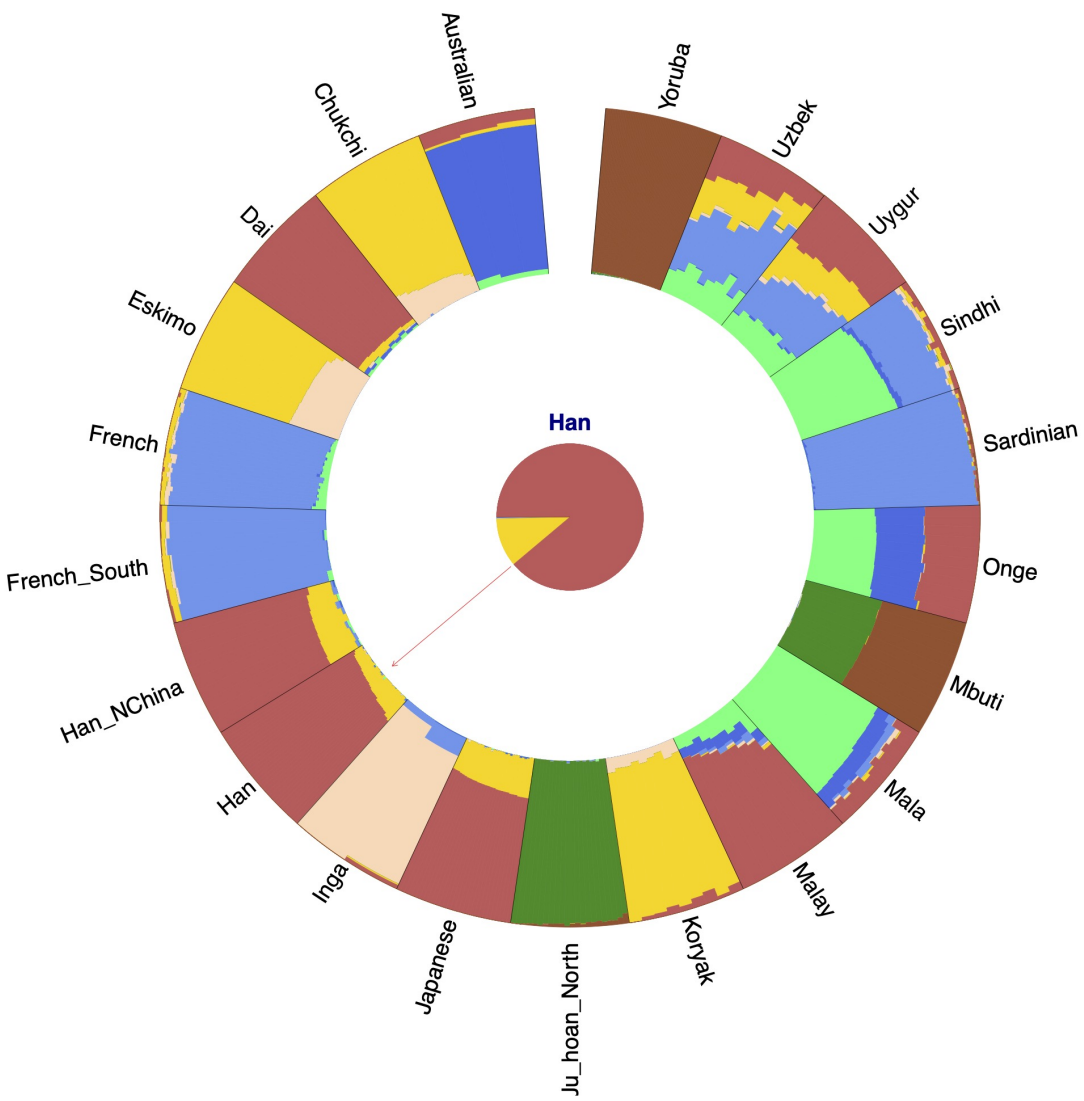


Here shows an example of the usage of the pair of parameters “tarang1” and “tarang2”.

```
> sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols,
target = exp_tars, tarang1 = 90, tarang2 = 330)
```

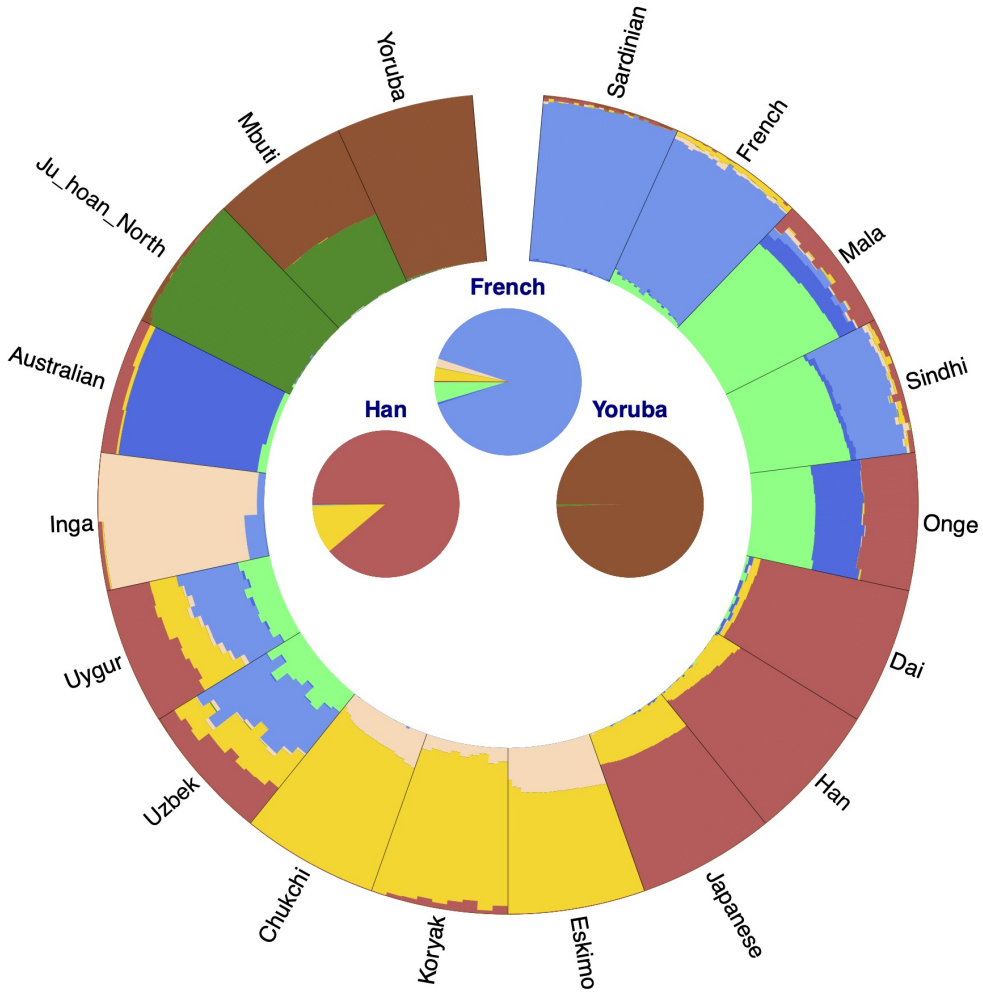
Specially, if you want a target pie to be located right at the center of the plot, you can specify “cendis” as zero. To show the corresponding position of target populations on the outer ring, you can add arrows to the targets. But remember to make your populations well sorted if you specify multiple targets.

```
> sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols,
target = exp_tars[3], cendis = 0, arrow = T)
```



4.1.4 Specify or automatically sort the population You can specify the population order, like

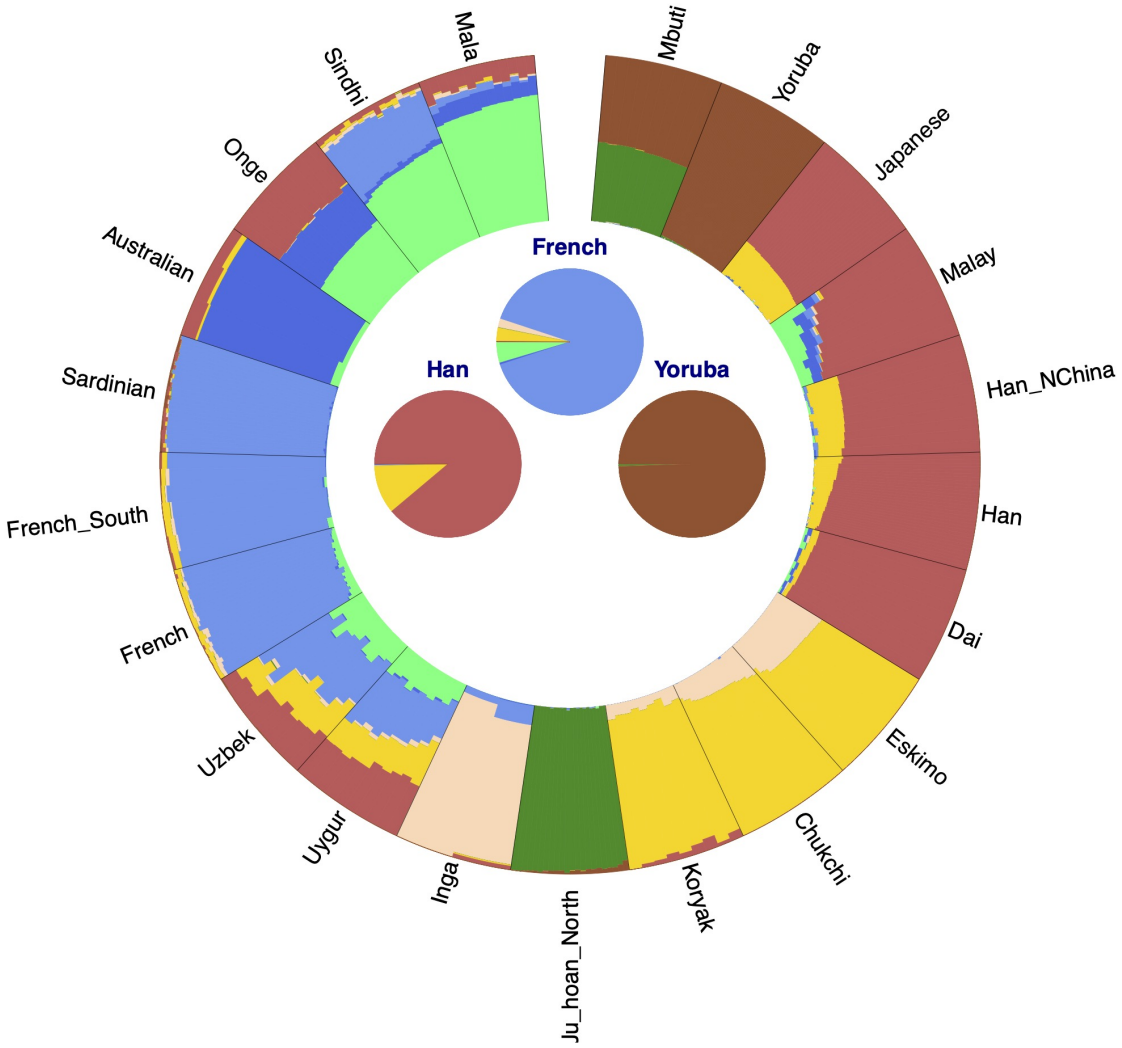
```
> exp_order <- read.table('./inst/extdata/exp_ances.K8.1.order',
stringsAsFactors = F, header = F)$V1
> exp_order
 [1] "Australian"    "Chukchi"       "Dai"           "Eskimo"
 [5] "French"        "Han"           "Inga"          "Japanese"
 [9] "Ju_hoan_North" "Kazak"         "Kirgiz"        "Koryak"
[13] "Mala"          "Mbuti"         "Onge"          "Sardinian"
[17] "Sindhi"        "Yoruba"
> sectorplot(Q = exp_q, ind = exp_ind,
target = exp_tars, ancescols = exp_cols, poporder = exp_order)
Use the specified population order...
```



Note that when loading the population order from a text file, the "stringsAsFactors" should be set as FALSE to make it possible to sort the population order as specified, Otherwise, it will cause errors.

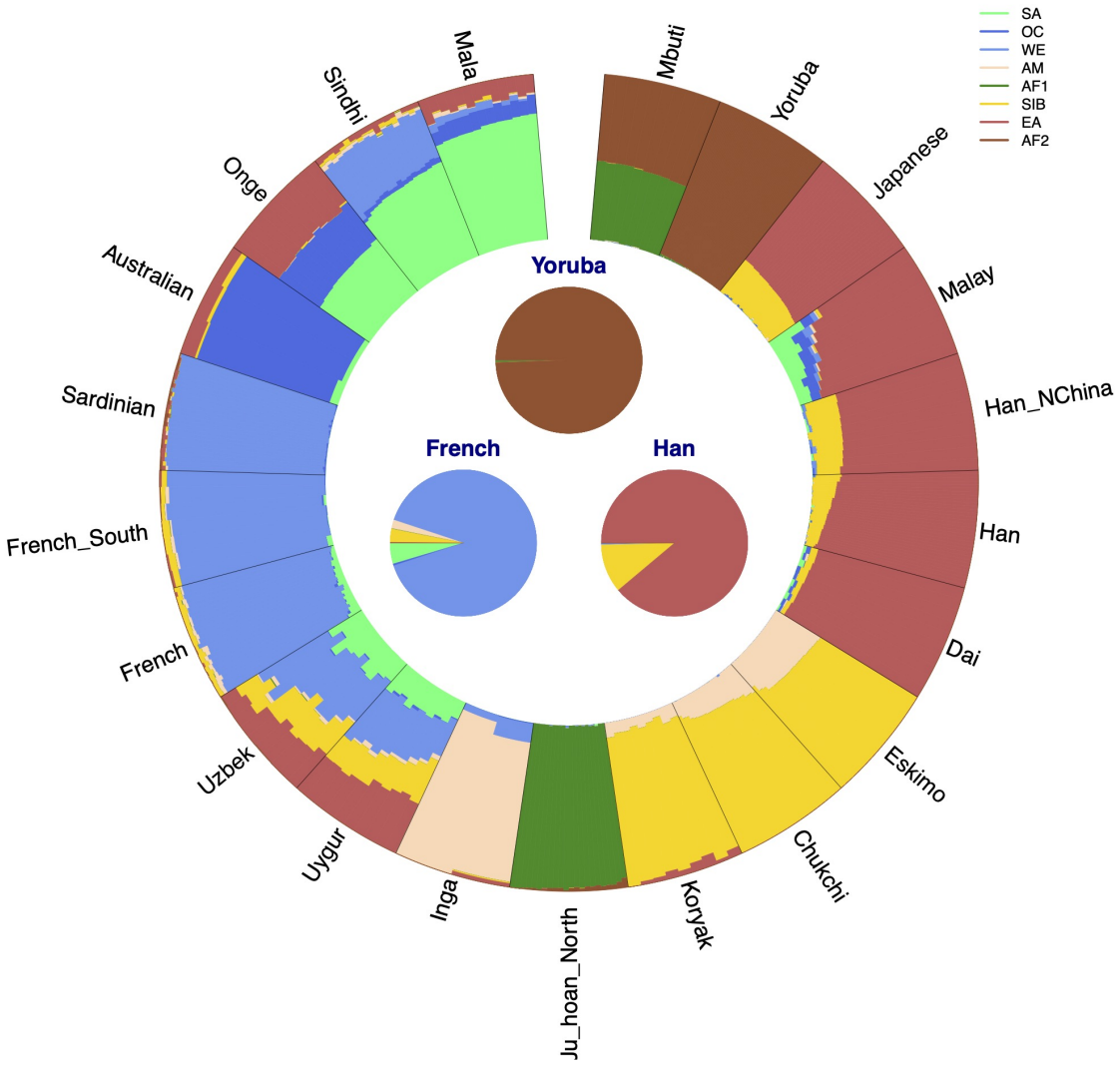
Alternatively, you can just let the software sort the populations.

```
> sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols,
target = exp_tars, sorting = T)
No population order specified.
Sort the population order...
```



4.1.5 Legend To show the type of ancestry components, you can use a legend. You can define the ancestries by yourself or just let the software name them "Ancestry_1" "Ancestry_2" and so on.

```
> exp_anc <- c("SA", "OC", "WE", "AM", "AF1", "SIB", "EA", "AF2")
> sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols,
target = exp_tars, tarang1 = 90, tarang2 = 330,
sorting = T, legend_mode = T, ancesnames = exp_anc)
```

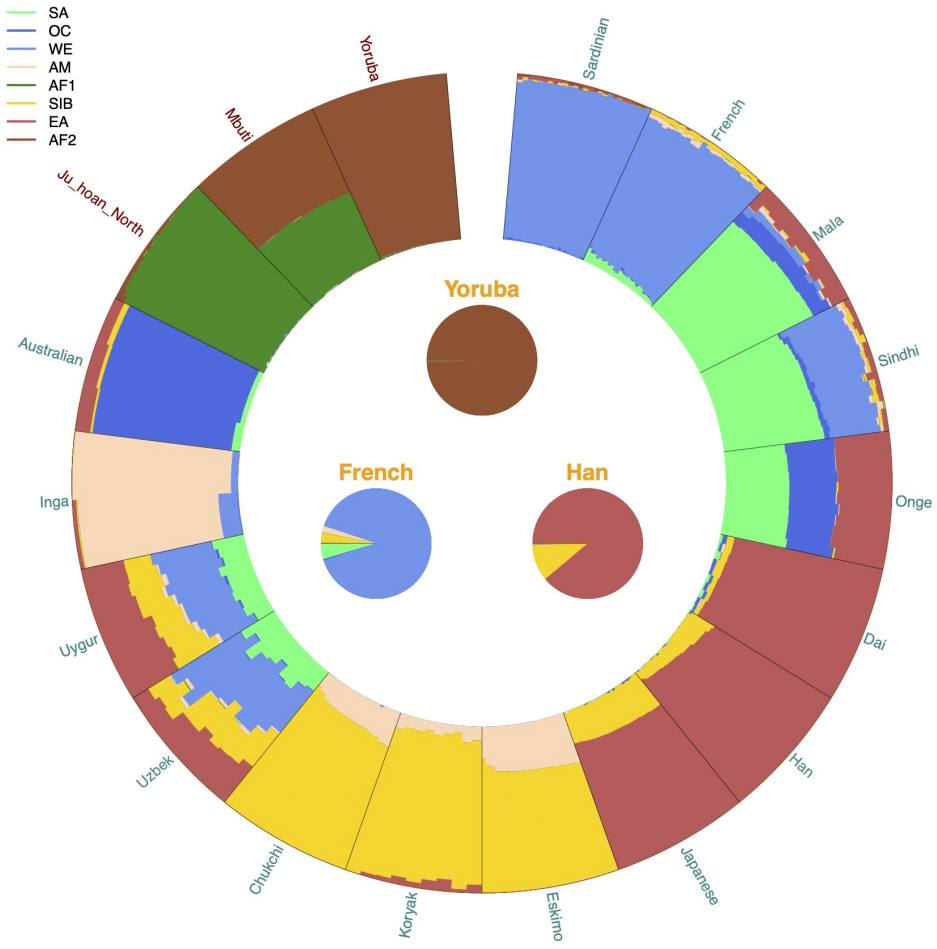


4.1.6 Aesthetics of your graph We provide some parameters to help you make your graph more beautiful.

You can modify the font, size, and color of the target/population labels, the position of the legend, and so on. Note that the color of the population labels can be specified as a vector with a length the same as the number of the populations so that the populations from different groups can be indicated by different colors.

```
> exp_info <- read.table('./inst/extdata/exp_ances.K8.1.info',
header = T, row.names = 1)
> head(exp_info)
                      Region Latitude Longitude
Australian           Oceania    -13.0     143.0
Chukchi      CentralAsiaSiberia     69.5     168.8
```

```
Dai                    EastAsia     21.0     100.0
Eskimo       CentralAsiaSiberia     64.5     172.9
French              WestEurasia     46.0       2.0
French_South        WestEurasia     43.4      -0.6
> exp_info$popcol <- sapply(exp_info$Region,
FUN = function(x) ifelse(x == "Africa", "darkred", "cyan4"))
sectorplot(Q = exp_q, ind = exp_ind, ancescols = exp_cols,
target = exp_tars, tarangs = c(90, 210, 330), poporder = exp_order,
legend_mode = T, ancesnames = exp_anc, legend.pos = "topleft",
tar.r = 0.45, tar.lab.col = "orange",
pop.lab.col = exp_info$popcol,  pop.lab.cex = 4)
```



4.2 Radiationplot

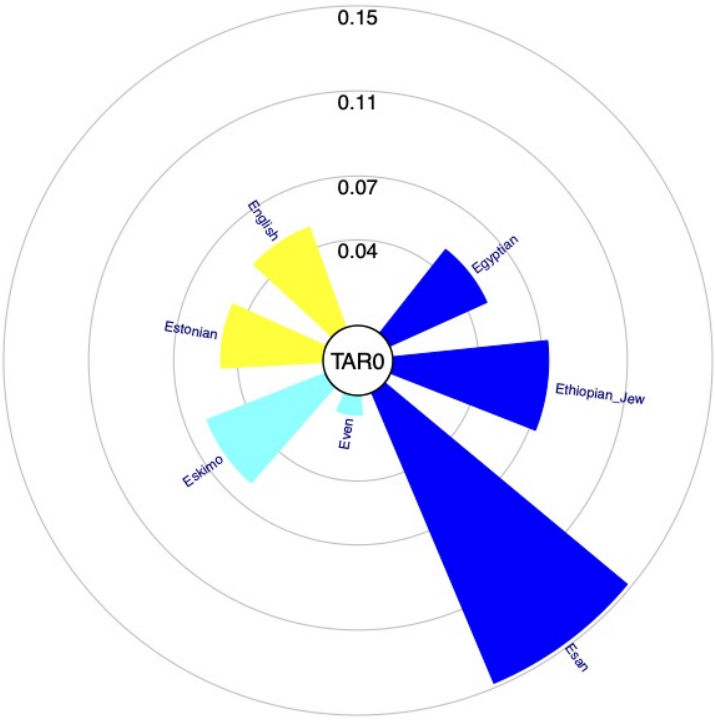
`radiationplot` can be used to visualize the genetic distance from one target population to other populations. The code was adapted from what was used in a study of the genetic history of Tibetan highlanders (Lu, et al, 2016).

4.2.1 Required input To use this plot, a four-column matrix containing the information of populations, regions, genetic difference, and color code is required. These information can be saved in a tab/space-delimited text file.

```
> # input
> exp_fst.local <- read.table("./inst/extdata/exp_fst.local.txt",
stringsAsFactors = F, header = F)
> exp_fst.local
             V1                 V2        V3  V4
1      Egyptian             Africa 0.0497950 blue
2 Ethiopian_Jew             Africa 0.0729730 blue
3          Esan             Africa 0.1479100 blue
4          Even CentralAsiaSiberia 0.0090195 cyan
5        Eskimo CentralAsiaSiberia 0.0596530 cyan
6      Estonian        WestEurasia 0.0478740 yellow
7       English        WestEurasia 0.0499280 yellow
```

4.2.2 Basic functions

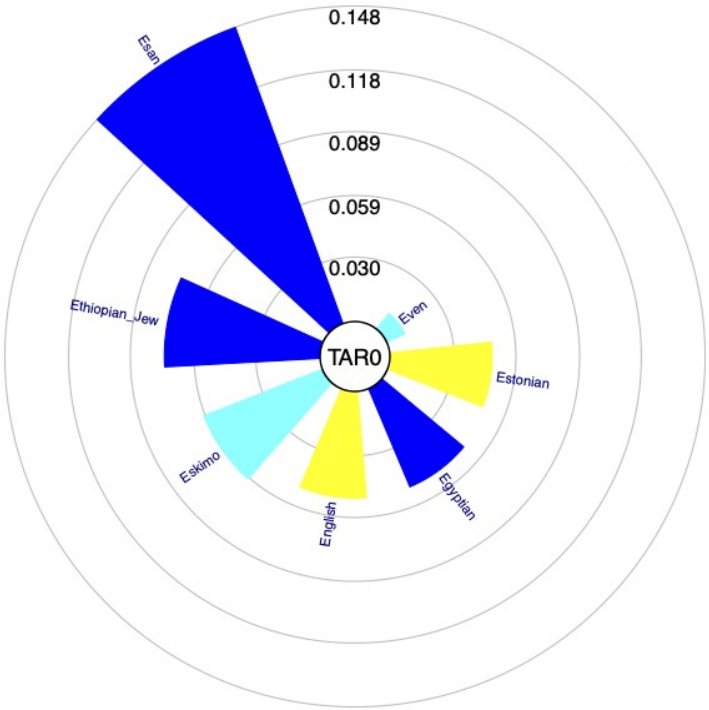
```
> # Graphing
> pdf("exp_fst.local.8.basic.pdf", width = 10, height = 10)
>     radiationplot(data = exp_fst.local, target = "TAR0")
> dev.off()
```



4.2.3 Automatically sorting To make the populations sorted in an order according to their genetic distance to the target, you can use the flag "sorting".

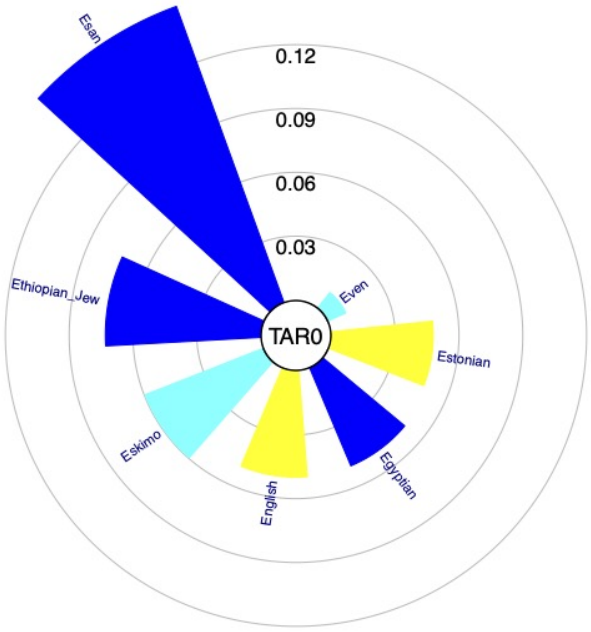
4.2.4 Layout of the rings You can change the number of the outer layers indicating the genetic distance via the parameter "num", and make the values on the ring rounded to a certain decimal space via "digits".

```
> radiationplot(data = exp_fst.local, target = "TAR0", sorting = T,
num = 5, digits = 3)
```



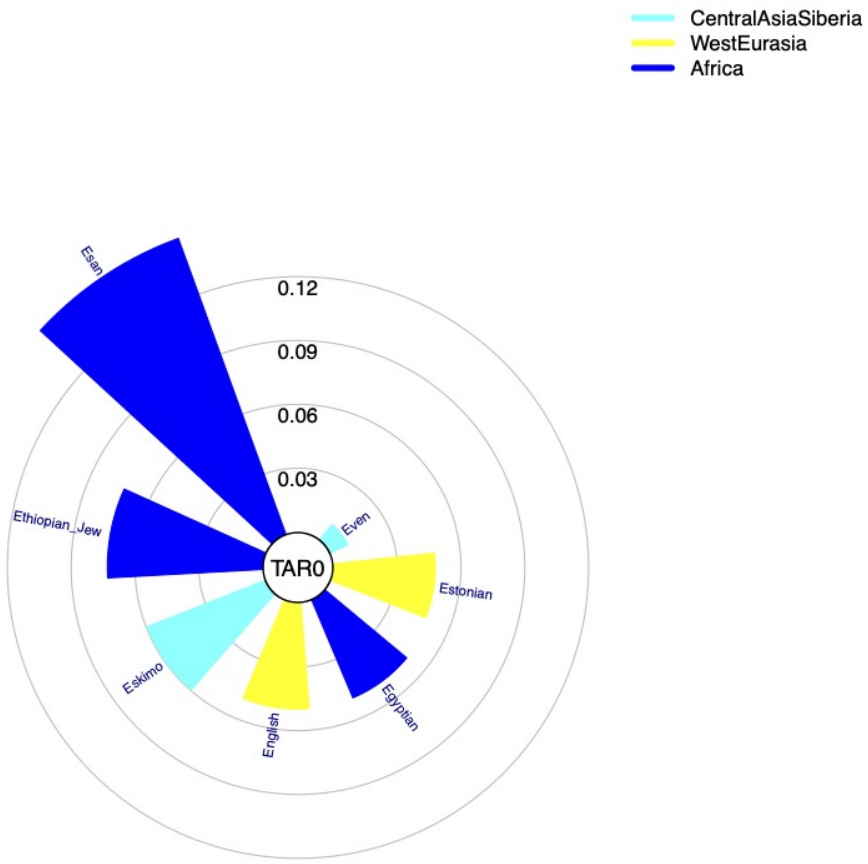
You can also specify the values of the outer layers directly by the parameter “layers”.

```
> radiationplot(data = exp_fst.local, target = "TAR0", sorting = T,
layers = c(0.03, 0.06, 0.09, 0.12))
```



4.2.5 Legend If populations are from several regions and you would like to show the information, you can switch the "legend_mode" flag to `TRUE`, and color the radiation bars according to their region info.

```
> radiationplot(data = exp_fst.local, target = "TAR0",  sorting = T,
layers = c(0.03, 0.06, 0.09, 0.12), legend_mode = T)
```



`radiationplot` uses similar parameters for aesthetics as `sectorplot`. For more details, please refer to the R documentation.

4.3 Ancestry merging

This function is translated from a part of the python code authored by Yuwen Pan (https://github.com/Shuhua-Group/ADMIXTURE.merge). This function merges the ancestry proportion matrices estimated by software like ADMIXTURE with the same dataset and the same ancestry component number (K).

4.3.1 Required input Here, the input ancestry matrices are loaded into the R environment by the R base function `list.files`.

The input file is tab/space-delimited, and contains (2 + K) columns, in which the first two columns are IDs

of individuals and groups and the rest indicates the proportion of K ancestry components. The input file can be constructed by merging the information in ".Q" and ".ind" files.

```
> ancfiles <- list.files("./inst/extdata/",
pattern = "[0-9]\\.ancestry", full.names = T)
> print(ancfiles)
 [1] "./inst/extdata//ances.8.10.ancestry" "./inst/extdata//ances.8.1.ancestry"
 [3] "./inst/extdata//ances.8.2.ancestry"  "./inst/extdata//ances.8.3.ancestry"
 [5] "./inst/extdata//ances.8.4.ancestry"  "./inst/extdata//ances.8.5.ancestry"
 [7] "./inst/extdata//ances.8.6.ancestry"  "./inst/extdata//ances.8.7.ancestry"
 [9] "./inst/extdata//ances.8.8.ancestry"  "./inst/extdata//ances.8.9.ancestry"
> ancmat <- read.table(ancfiles[1], header = F)
> head(ancmat)
```

4.3.2 Processing Then you can use the function `ancmerge` to analyze. Remember to specify a reference ancestry matrix file as the target to compare. Also, you need to tell AncestryPainter how many the ancestry components in the matrices are.

```
> results <- ancmerge(tar_anc_filelist = ancfiles,
ref = ancfiles[1], K = 8)
Time:
2023-01-22 20:00:05
Path:
/home/users/yourpath/AncestryPainterV2
Done.
```

4.3.3 Output `ancmerge` returns a list of four elements, containing the merged ancestry proportion matrix, the supporting ratio of each ancestry component, and the names of files consensus/conflicted with the reference, respectively.

```
> # Output
> str(results)
List of 4
 $ merged_ancestry   :'data.frame':      2538 obs. of  10 variables:
  ..$ 1 : Factor w/ 2538 levels "20100320-1","20100320-2",..: 425 427 433 442 436 434 432 429 441 439 .
  ..$ 2 : Factor w/ 215 levels "Egyptian","Moroccan_Jew",..: 1 1 1 1 1 1 1 1 1 1 ...
  ..$ 3 : num [1:2538] 0.1166 0.1146 0.1005 0.1238 0.0822 ...
  ..$ 4 : num [1:2538] 0.1253 0.1139 0.0933 0.097 0.0975 ...
  ..$ 5 : num [1:2538] 1e-05 1e-05 1e-05 1e-05 1e-05 1e-05 1e-05 1e-05 1e-05 1e-05 ...
  ..$ 6 : num [1:2538] 0.00846 0.00001 0.01142 0.00701 0.00413 ...
  ..$ 7 : num [1:2538] 1.00e-05 4.88e-03 1.00e-05 1.05e-05 1.00e-05 ...
  ..$ 8 : num [1:2538] 0.00001 0.00543 0.00001 0.00001 0.00619 ...
  ..$ 9 : num [1:2538] 0.0867 0.0722 0.1053 0.0826 0.1153 ...
  ..$ 10: num [1:2538] 0.663 0.689 0.689 0.69 0.695 ...
 $ supporting_ratio  :'data.frame':  8 obs. of  4 variables:
  ..$ component     : chr [1:8] "comp1" "comp2" "comp3" "comp4" ...
  ..$ represent_pop : chr [1:8] "Yoruba" "Hadza" "Ju_hoan_North" "Papuan" ...
  ..$ support_counts: int [1:8] 10 2 10 10 10 2 10 10
  ..$ support_ratio : num [1:8] 1 0.2 1 1 1 0.2 1 1
 $ consensus_filelist: chr [1:2] "./inst/extdata//ances.8.10.ancestry" "./inst/extdata//ances.8.4.ancest
 $ conflict_filelist : chr [1:8] "./inst/extdata//ances.8.1.ancestry" "./inst/extdata//ances.8.2.ancest
```

Ancestry proportion matrix In the merged ancestry proportion data frame, the first two columns are individual IDs and group (population) IDs. And the rest of the columns in the data frame are the ancestry proportion matrix, which can be used as the input argument "Q" of "sectorplot".

Supporting ratio of ancestry component `results$supporting_ratio` shows the ancestry components, their representative groups (with the largest ancestry proportion of the corresponding component), the number of files supporting the reference for each component, and the supporting ratio.

```
> results$supporting_ratio
  component represent_pop support_counts support_ratio
1     comp1        Yoruba             10           1.0
2     comp2         Hadza              2           0.2
3     comp3 Ju_hoan_North             10           1.0
4     comp4        Papuan             10           1.0
5     comp5         Chane             10           1.0
6     comp6        Korean              2           0.2
7     comp7          Mala             10           1.0
8     comp8     Sardinian             10           1.0
```

Saving output

```
> write.table(results$merged_ancestry, "inst/extdata/merged_ancestry.txt",
sep = "\t", col.names = F, row.names = F, quote = F)
```

4.4 Admixture history graph (Planned)

Pugach, et al have introduced a method to infer the admixture sequence of multiple ancestry populations by calculating correlation coefficients between ancestry components, which has been applied to our previous study of the Uyghurs in Xinjiang (Feng, et al, 2017). We are going to implement this method in the AncestryPainterV2 package while optimizing the calculation of the correlation coefficients.

5 Citation and contact

If you use AncestryPainterV2 in your project, please cite

https://github.com/Shuhua-Group/AncestryPainterV2

If you have any questions or suggestions, welcome to contact us: chenshh@shanghaitech.edu.cn

6 Reference

Feng Q, Lu Y, Ni X, Yuan K, Yang Y, Yang X, Liu C, Lou H, Ning Z, Wang Y, Lu D, Zhang C, Zhou Y, Shi M, Tian L, Wang X, Zhang X, Li J, Khan A, Guan Y, Tang K, Wang S, Xu S. Genetic History of Xinjiang's Uyghurs Suggests Bronze Age Multiple-Way Contacts in Eurasia. Mol Biol Evol. 2017 Oct 1;34(10):2572-2582. doi: 10.1093/molbev/msx177. PMID: 28595347.

Feng Q, Lu D, Xu S. AncestryPainter: A Graphic Program for Displaying Ancestry Composition of Populations and Individuals. Genomics Proteomics Bioinformatics. 2018 Oct;16(5):382-385. doi: 10.1016/j.gpb.2018.05.002. Epub 2018 Nov 22. PMID: 30472416; PMCID: PMC6364040.

Lu D, Lou H, Yuan K, Wang X, Wang Y, Zhang C, Lu Y, Yang X, Deng L, Zhou Y, Feng Q, Hu Y, Ding Q, Yang Y, Li S, Jin L, Guan Y, Su B, Kang L, Xu S. Ancestral Origins and Genetic History of Tibetan

Highlanders. Am J Hum Genet. 2016 Sep 1;99(3):580-594. doi: 10.1016/j.ajhg.2016.07.002. Epub 2016 Aug 25. PMID: 27569548; PMCID: PMC5011065.

Pan Y, Zhang C, Lu Y, Ning Z, Lu D, Gao Y, Zhao X, Yang Y, Guan Y, Mamatyusupu D, Xu S. Genomic diversity and post-admixture adaptation in the Uyghurs. Natl Sci Rev. 2021 Sep 11;9(3):nwab124. doi: 10.1093/nsr/nwab124. PMID: 35350227; PMCID: PMC8953455.

Pugach I, Matveev R, Spitsyn V, Makarov S, Novgorodov I, Osakovsky V, Stoneking M, Pakendorf B. The Complex Admixture History and Recent Southern Origins of Siberian Populations. Mol Biol Evol. 2016 Jul;33(7):1777-95. doi: 10.1093/molbev/msw055. Epub 2016 Mar 18. PMID: 26993256; PMCID: PMC4915357.