
Multi-K ADMIXTURE Plot Script - Sanitized and Fully Documented

Description

This R script generates admixture plots for multiple K values, mapping ancestry components across different K values to maintain consistency. It uses hierarchical clustering to order populations, a partial topological sorting algorithm to handle component swaps with potential cycles, and a combination of direct renaming and pairwise exchanges to ensure ancestry consistency across K.

Features

- Reads ADMIXTURE Q files and sample information.
- Supports multiple K values.
- Maps ancestry components from lower K to the reference K (maximum K).
- Handles cycles in component swaps using partial topological sorting.
- Orders populations using hierarchical clustering.
- Generates aligned bar plots for each K and side plots for target populations.
- Fully customizable color palette.

Input Files

- `.fam` file containing sample identifiers.
- `.info` file containing sample metadata (including population labels).
- ADMIXTURE `.Q` files for each K in the specified range.

Output

- A multi-K ADMIXTURE plot in PNG format.
- Plots include:
 - Left: full sample ancestry proportion bars.
 - Right: average ancestry proportions for specified target populations.

Usage

```
# Set directories and file paths
run_dir <- "YOUR_RUN_DIRECTORY"
fam_file <- file.path(run_dir, "samples.fam")
info_file <- file.path(run_dir, "sample_info.txt")
K_range <- 3:7
target_pops <- c("Pop1", "Pop2", "Pop3")
output_png <- "Admixture_Plot_K3-7.png"

# Read and process data
admixture_data <- read_admixture_data(run_dir, fam_file, info_file, K_range)
```

Perform population clustering

```
admix_data <- cluster_populations(admix_data, K_range)
```

Map ancestry components across K

```
admix_mapped <- map_ancestries(admix_data, K_range)
```

Create plots

```
main_plots <- create_main_plots(admix_mapped, color_dict, K_range)
```

```
pop_axis <- create_population_axis(sample_order_df)
```

```
target_plots <- create_target_population_plots(admix_mapped, target_pops, color_dict, K_range)
```

Save final plot

```
final_plot <- combine_plots(main_plots, pop_axis, target_plots)
```

```
ggsave(output_png, final_plot, width=plot_width, height=plot_height, units="in")
```

Notes

- Hierarchical clustering is performed using `ward.D2` method.
- Color palette can be customized with `RColorBrewer` schemes.
- Partial topological sort ensures safe execution of component swaps even when cycles exist.
- Unmapped components are renamed with prefix `Extra_` to avoid conflicts.
- Compatible with R ≥ 4.0 .

Example Color Palette

```
c("#A6CEE3", "#1F78B4", "#B2DF8A", "#33A02C", "#FB9A99", "#E31A1C", "#FDBF6F")
```