Visualize individual's assignment results

2016-05-31

Introduction

- visualize the noise in the assignment
- impact of: marker number, imputation, thl set & subsample set (i.e. the individual composition)

Under dev.

Details of the approach

More iterations will provide better estimate and reduce the bias introduced by the mixture of samples used inside the training and holdhout set. This is more important with admixed populations or populations characterized by lower Fst.

To reduce assignment bias introduced by uneven sample size between your groups, **assigner** provides 2 addtional arguments: `subsample` and `iteration.subsample`. Using more iterations will make sure that all potential combinations of individuals are used and will provide better null distribution for the statistics.

Run an assignment analysis with these arguments. Try without imputations first.

1. Clean your desk and remove the clutter in the global environment

```
rm(list=ls())
```

2. Load the required libraries

```
if (!require("devtools")) install.packages("devtools")
if (!require("reshape2")) install.packages("reshape2")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("stringr")) install.packages("stringr")
if (!require("stringi")) install.packages("stringi")
if (!require("plyr")) install.packages("plyr")
if (!require("dplyr")) install.packages("dplyr")
if (!require("tidyr")) install.packages("tidyr")
if (!require("readr")) install.packages("readr")
if (!require("purrr")) install.packages("purrr")
if (!require("data.table")) install.packages("data.table")
if (!require("adegenet")) install.packages("adegenet")
if (!require("stackr")) install_github("thierrygosselin/stackr")
if (!require("assigner")) install_github("thierrygosselin/assigner", build_vignettes = TRUE)
#install_gsi_sim(fromSource = TRUE) # if assigner was re-installed, uncomment and run
```

3. The first tool we'll use is `assigner::import_subsamples`. This function imports the assignment results for each individuals in each of the subsample folder found in the directory you provide in the function:

```
# Get the folder containing the data:
https://github.com/thierrygosselin/package_data/raw/master/top_markers_assignment.tar.gz
# change the path below to reflect the directory
assignment.data.subsample <- import_subsamples(dir.path = "~/Downloads/top_markers_assignment", imp
```

4. use this pop.levels to order the data

```
pop.levels <- c("north", "south")
```

5. work on the data frame

```
ind.levels <- assignment.data.subsample %>%
  select(INDIVIDUALS, CURRENT) %>%
  distinct(INDIVIDUALS, CURRENT) %>%
  mutate(CURRENT = factor(CURRENT, levels = pop.levels, ordered = TRUE)) %>%
  arrange(desc(CURRENT), INDIVIDUALS) %>%
  select(INDIVIDUALS)
```

6. last step

```
data.prep <- assignment.data.subsample %>%
  select(INDIVIDUALS, CURRENT, INFERRED, MARKER_NUMBER, MISSING_DATA, ITERATIONS, SUBSAMPLE) %>%
  group_by(INDIVIDUALS, CURRENT, MISSING_DATA, ITERATIONS, SUBSAMPLE) %>%
  tidyr::spread(data = ., key = MARKER_NUMBER, value = INFERRED) %>%
  tidyr::gather(data = ., key = GROUPING, value = ASSIGNMENT, -c(INDIVIDUALS, CURRENT, MISSING_DATA
  ungroup() %>%
  mutate(SUBSAMPLE = stri_pad_left(str = SUBSAMPLE, pad = "0", width = 3)) %>%
  tidyr::unite(data = ., col = INDIVIDUALS_SUB, INDIVIDUALS, SUBSAMPLE, sep = "_", remove = FALSE)
  mutate(
GROUPING = factor(GROUPING, levels = c(50, 100, 200,300, 400, 500, 1000, 15454)),
GROUPING = droplevels(GROUPING),
ASSIGNMENT = factor(ASSIGNMENT),
CURRENT = factor(CURRENT),
ITERATIONS = factor(ITERATIONS)
) %>%
arrange(CURRENT, GROUPING, INDIVIDUALS, ITERATIONS)
```

7. Heatmap figure

```
heatmap.fig <- ggplot(data.prep,(aes(x = ITERATIONS, y = as.character(INDIVIDUALS)))) +
  geom_tile(aes(fill = ASSIGNMENT)) +
  #scale_x_discrete(breaks = axis.breaks)+
  labs(x = "Marker resampling (iterations)")+
  labs(y = "Individuals")+
  theme_bw()+
  theme(
panel.grid.minor.x = element_blank(),
panel.grid.major.y = element_blank(),
axis.title.x = element_text(size = 10, family = "Helvetica", face = "bold"),
axis.text.x = element_text(size = 6, family = "Helvetica", angle = 90, hjust = 1, vjust = 0.5),
axis.title.y = element_text(size = 10, family = "Helvetica", face = "bold"),
axis.text.y = element_text(size = 1, family = "Helvetica")
  ) +
  facet_grid(CURRENT~GROUPING + MISSING_DATA, scales = "free", drop = TRUE)
heatmap.fig
# save, inspect and zoom...
ggsave("assignment.heatmap.pdf", height = 30, width = 20, dpi = 600, units = "cm", useDingbats = F)
```

Conclusion

Under construction

References Under construction