APIS: An Auto-Adaptive Parentage Inference Software tolerant to missing parents

Ronan GRIOT [1,2], François ALLAL [3], Marc VANDEPUTTE [2,3]

[1]*SYSAAF, Station LPGP/INRA, Campus de Beaulieu, Rennes, France*
[2]*GABI, INRA, AgroParisTech, Université Paris-Saclay, France*
[3]*MARBEC, Univ. Montpellier, Ifremer, CNRS, IRD, Palavas-les-Flots, France*

2019-02-25

Description

This package include all the functions to assign with APIS (Griot & al., 2019). Parentage assignment is widely used for farmed and natural populations. As most of the likelihood software are based on simulation, the estimation of the simulation parameters is a key point for assignment reliability. Among those parameters, the proportion of missing parent is one of the most important. To avoid estimation of missing parents, we developed APIS (Auto-Adaptive Parentage Inference Software), based on observed average Mendelian transmission probabilities.

Install and load the package

```
devtools::install_github("rgriot/APIS")
library(APIS)
```

If an error message occurs during the installation, use the following command :

```
devtools::install_github("rgriot/APIS", args = "--no-multiarch")
library(APIS)
```

Formate your data

APIS requires matrices of characters as inputs. Each matrix has individuals as rows, markers as columns. The individual labels are set as rownames. Each cell is the genotype of one marker, coded "All1/All2". For example "A/A", "A/B", "B/B" for bi-allelic markers and "NA/NA" for missing value. For multi-allelic markers, use the generic coding "All1/All2".

```r
data("genotype_APIS")

head(off_full[,1:10])
#>         V2      V3      V4      V5      V6      V7      V8        V9      V10     V11
#> off_1 "A/B"   "B/B"   "A/A"   "A/A"   "B/B"   "B/B"   "NA/NA"   "A/A"   "A/A"   "A/B"
#> off_2 "A/B"   "B/B"   "A/B"   "A/A"   "B/B"   "B/B"   "B/B"     "A/A"   "A/A"   "A/B"
#> off_3 "A/B"   "A/B"   "A/B"   "A/A"   "B/B"   "B/B"   "B/B"     "A/A"   "A/A"   "A/A"
#> off_4 "B/B"   "B/B"   "A/A"   "A/A"   "B/B"   "B/B"   "B/B"     "A/A"   "A/B"   "A/A"
#> off_5 "B/B"   "A/B"   "A/B"   "A/A"   "B/B"   "B/B"   "B/B"     "A/A"   "A/B"   "A/B"
#> off_6 "A/B"   "B/B"   "A/A"   "A/A"   "B/B"   "B/B"   "B/B"     "A/A"   "B/B"   "A/B"
rownames(off_full[1:6,])
#> [1] "off_1" "off_2" "off_3" "off_4" "off_5" "off_6"
```

Prepare the inputs

APIS main function requires 4 inputs :

- off.genotype = matrix of offspring genotypes coded as explained above

- sire.genotype = matrix of sires genotypes coded as explained above

- dam.genotype = matrix of dams genotypes coded as explained above

- error = accepted assignment error rate (What is the maximum error rate I accept in my assignment results ?)

```
head(off_full[,1:10])
#>        V2    V3    V4    V5    V6    V7    V8      V9   V10   V11
#> off_1 "A/B" "B/B" "A/A" "A/A" "B/B" "B/B" "NA/NA" "A/A" "A/A" "A/B"
#> off_2 "A/B" "B/B" "A/B" "A/A" "B/B" "B/B" "B/B"   "A/A" "A/A" "A/B"
#> off_3 "A/B" "A/B" "A/B" "A/A" "B/B" "B/B" "B/B"   "A/A" "A/A" "A/A"
#> off_4 "B/B" "B/B" "A/A" "A/A" "B/B" "B/B" "B/B"   "A/A" "A/B" "A/A"
#> off_5 "B/B" "A/B" "A/B" "A/A" "B/B" "B/B" "B/B"   "A/A" "A/B" "A/B"
#> off_6 "A/B" "B/B" "A/A" "A/A" "B/B" "B/B" "B/B"   "A/A" "B/B" "A/B"
head(sire_full[,1:10])
#>         V2    V3    V4    V5    V6    V7    V8    V9    V10   V11
#> sire_1 "A/A" "A/B" "A/A" "A/A" "B/B" "B/B" "B/B" "A/A" "A/B" "A/B"
#> sire_2 "B/B" "A/B" "B/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/B" "B/B"
#> sire_3 "A/B" "A/B" "A/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/B" "A/B"
#> sire_4 "A/B" "B/B" "A/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/B" "A/B"
#> sire_5 "B/B" "A/B" "A/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/A" "A/B"
#> sire_6 "B/B" "A/A" "A/A" "A/A" "B/B" "B/B" "B/B" "A/A" "B/B" "A/A"
head(dam_full[,1:10])
#>        V2    V3    V4    V5    V6    V7    V8    V9    V10   V11
#> dam_1 "B/B" "A/B" "B/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/A" "B/B"
#> dam_2 "A/A" "B/B" "B/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/A" "B/B"
#> dam_3 "A/B" "A/B" "A/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/B" "B/B"
#> dam_4 "A/B" "B/B" "A/B" "A/A" "B/B" "B/B" "B/B" "A/A" "A/B" "B/B"
#> dam_5 "A/B" "B/B" "A/A" "A/A" "B/B" "B/B" "B/B" "A/A" "A/B" "A/A"
#> dam_6 "A/A" "B/B" "B/B" "A/A" "B/B" "B/B" "B/B" "A/A" "B/B" "B/B"
error <- 0.05 #I accept 5% of errors in the results
```

Running the assignment

The main function to perform parentage assignment with APIS the "APIS" function. Use the function as below, with default parameters for exclusion threshold and preselection of parents for maximizing the reliability.

```
result <- APIS(off.genotype = off_full,
               sire.genotype = sire_full,
               dam.genotype = dam_full,
               error = error)
```

Analyse the results

APIS gives you 3 different outputs :

- pedigree
- log containing Mendelian transmission probabilities, mismatches and deltas for the first 3 parent pairs
- graphs of the distributions of deltas, Mendelian transmission probabilities and mismatches

According to the graphs, you can change the thresholds to improve your assignment.

If you want to set up your threshold on Mendelian probabilities, use :

```
new.result <- personalThreshold(APIS.result = result,
                                method = 'Pmendel',
                                threshold = 0.7)
```

If you want to set up your threshold on mismatches, use :

```
new.result <- personalThreshold(APIS.result = result,
                                method = 'exclusion',
                                threshold = 5)
```

Examples

Full data

This example uses the full data provided by the package (Figure 1).

```
result <- APIS(off.genotype = off_full,
               sire.genotype = sire_full,
               dam.genotype = dam_full,
               error = 0.05)
```

When I look at the mismatch distributions, I prefer to use exclusion and allow for 5 mismatches (Figure 2).

```
new.result <- personalThreshold(APIS.result = result,
                                method = 'exclusion',
                                threshold = 5)
```
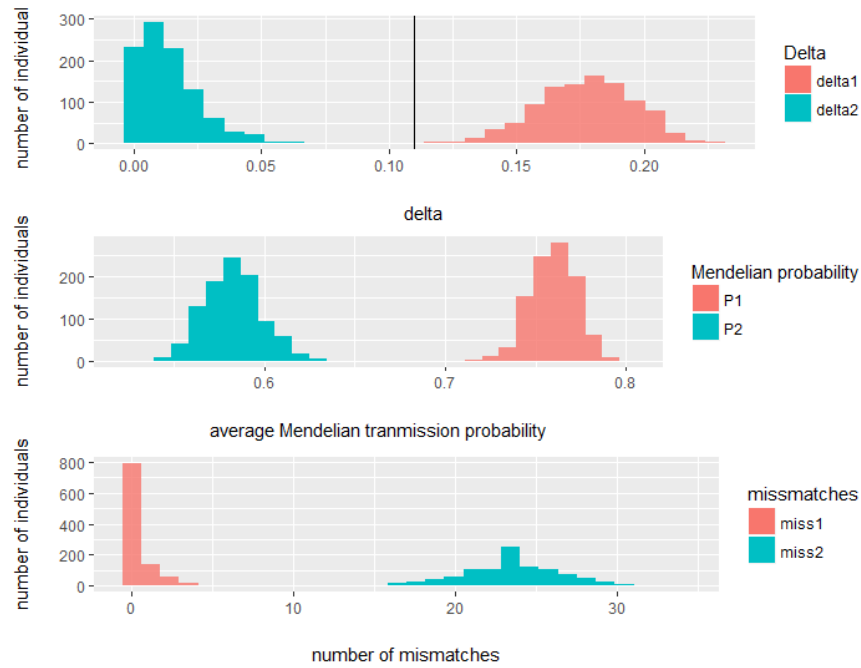
Figure 1: APIS outputs calculated on 1000 offspring from 10 sires and 10 dams, genotyped on 400 markers
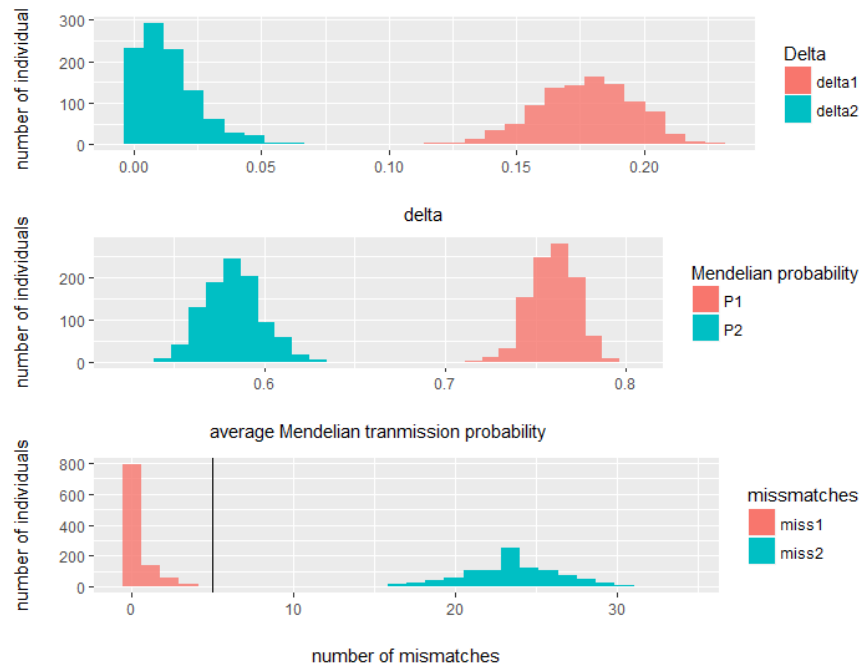

Figure 2: APIS outputs based on mismatches calculated on 1000 offspring from 10 sires and 10 dams, genotyped on 400 markers

Degraded data

This example uses the degraded data provided by the package (Figure 3).

```
result <- APIS(off.genotype = off_degraded,
               sire.genotype = sire_degraded,
               dam.genotype = dam_degraded,
               error = 0.05)
```

In this situation, the theoretical assignment power is low and there are missing parents. The distribution graphs do not give you more information about a new threshold value.

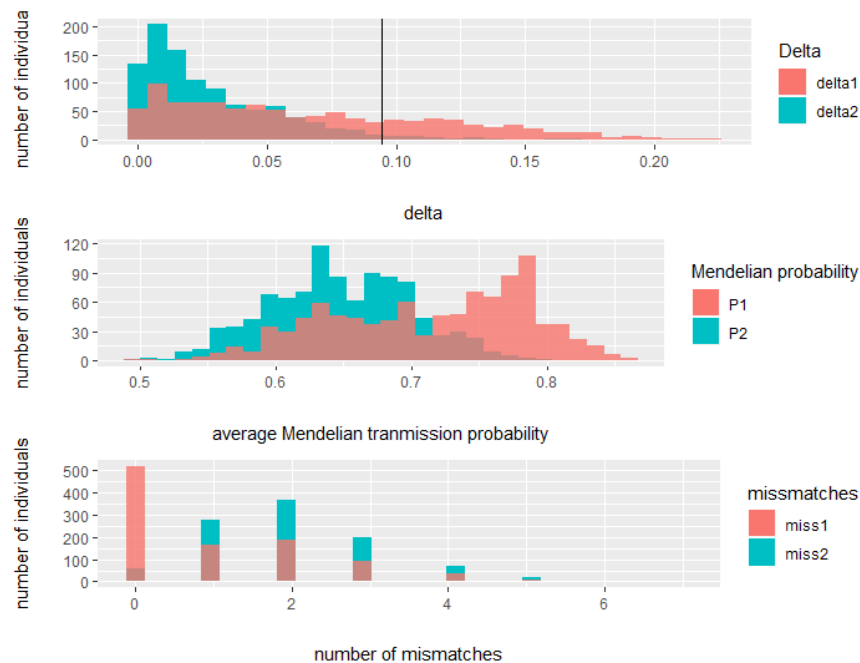Thus, the better answer is to keep APIS results.

Figure 3: APIS outputs calculated on 1000 offspring from 5 sires (5 missing sires) and 10 dams, genotyped on 50 markers

Other parameters of APIS function

APIS function can handle 2 other parameters :

- exclusion.threshold : For exclusion procedure, the number of mismatches allowed by the user. Use default value for Mendelian transmission probability procedure.

- preselect.Parent : Use of a parent preselection function based on Mendelian incompatibilities. Use default value to get the most accurate results. If a value is specified, this will increase computation time but can decrease assignment reliability.

Acknowlegments

Annexe

```
print(sessionInfo(), locale=FALSE)
#> R version 3.4.1 (2017-06-30)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 7 x64 (build 7601) Service Pack 1
#>
#> Matrix products: default
#>
#> attached base packages:
#> [1] parallel  stats     graphics  grDevices utils     datasets  methods
#> [8] base
#>
#> other attached packages:
#> [1] APIS_0.1.0       gridExtra_2.3    ggplot2_3.1.0    doParallel_1.0.14
#> [5] doSNOW_1.0.16    snow_0.4-3       iterators_1.0.10 foreach_1.4.4
#>
#> loaded via a namespace (and not attached):
#>  [1] tidyselect_0.2.5  xfun_0.5          remotes_2.0.2
#>  [4] purrr_0.3.0       colorspace_1.4-0  usethis_1.4.0
#>  [7] htmltools_0.3.6   yaml_2.2.0        rlang_0.3.1
#> [10] pkgbuild_1.0.2    pillar_1.3.1      glue_1.3.0
#> [13] withr_2.1.2       sessioninfo_1.1.1 plyr_1.8.4
#> [16] stringr_1.4.0     munsell_0.5.0     gtable_0.2.0
#> [19] devtools_2.0.1    codetools_0.2-16  memoise_1.1.0
#> [22] evaluate_0.13     knitr_1.21        callr_3.1.1
#> [25] ps_1.3.0          Rcpp_1.0.0        backports_1.1.3
#> [28] scales_1.0.0      desc_1.2.0        pkgload_1.0.2
#> [31] fs_1.2.6          digest_0.6.18     stringi_1.3.1
#> [34] processx_3.2.1    dplyr_0.8.0.1     rprojroot_1.3-2
#> [37] grid_3.4.1        cli_1.0.1         tools_3.4.1
#> [40] magrittr_1.5      lazyeval_0.2.1    tibble_2.0.1
#> [43] crayon_1.3.4      pkgconfig_2.0.2   prettyunits_1.0.2
#> [46] assertthat_0.2.0  rmarkdown_1.11    R6_2.4.0
#> [49] compiler_3.4.1
```