iMKT Pipeline

The McDonald and Kreitman Test

Theoretic explanation
Since adaptive mutations tend to be fixed quickly, they will rarely be detected as polymorphic variants but only as a divergent site (that is once fixed). Thus, the adaptive substitution rate in the genome can be inferred when there is an excess of non-synonymous divergence relative to its non-synonymous polymorphism. Accordingly, in the McDonald-Kreitman test (MKT) (McDonald 1and Kreitman, 1991) the divergence ratio (d N /d S) is normalized by the poly- morphism ratio (n / s), which allows taking into account the constraint on non- d N /d S synonymous sites and, thus, better detect adaptive substitutions (> 1). n / s From that one can also estimate the proportion of fixed variants that are adap- tive ( ). However, the estimation of can be biased due to the segregation of slightly deleterious non-synonymous mutations (Eyre-Walker, 2002). Given a stable population size, slightly deleterious mutations can produce an understimation of because they tend to contribute more to polymorphism than to divergence. Because slightly deleterious substitutions tend to segregate at low frequency, its effect can be partially controlled by removing low-frequency polymorphisms from the analysis, known as Fay-Wycoff-Wu method (Fay et al., 2001). However, Charlesworth and Eyre-Walker (2008) showed that even removing low-frequency variants, the estimate of is always down- wardly biased and only these estimates are reasonable accurate when the rate of adaptive evolution is high and the distribution of fitness effects of slightly deleterious mutations is leptokurtic (because leptokurtic distributions have a smaller proportion of polymorphisms that are slightly deleterious). The modification of the MKT introduced at the DGRP project (Mackay et al., 2012). was proposed as a better method for correcting the effect of slightly deleterious substitutions. Instead of simply removing low-frequency polymorphisms, the count of segregating sites in non-synonymous sites is separated into the num- ber of neutral variants and the number of weakly deleterious variants, allowing evaluating independently adaptive and weakly deleterious selection.

Pipeline

The package is deposited in the official repository CRAN (ojalá), and in the BGD group github. It could be downloaded by the following methods:

```
# install.packages("iMKT") ##If CRAN
# library(devtools) ##If github
# install_github("sergihervas/iMKT")
```

```
library(iMKT)
```

The data used in this tutorial is incorporated inside the package in order to use it as tutorial, or replicate this vignettes to understand better all the package functionalities. You could access the example data and save it in your own variable, but the are loaded in your enviroment by default loading the package!

```
mydafdata
#>      daf    Pi    P0
#> 1  0.025 22490 17189
#> 2  0.075  3217  4780
#> 3  0.125  1616  2874
#> 4  0.175   999  2088
#> 5  0.225   754  1685
#> 6  0.275   679  1443
#> 7  0.325   575  1264
#> 8  0.375   484  1232
#> 9  0.425   427  1148
```

```
#> 10 0.475   437  1068
#> 11 0.525   378   986
#> 12 0.575   341   928
#> 13 0.625   310   893
#> 14 0.675   335   928
#> 15 0.725   315   945
#> 16 0.775   297   822
#> 17 0.825   326   885
#> 18 0.875   369   953
#> 19 0.925   448  1086
#> 20 0.975  1019  1904
mydivergencedata
#>         mi    Di     m0    D0
#> 1 2598805 54641 620019 52537
```

```
exampleDaf<-mydafdata
exampleDiverge<-mydivergencedata
```

The package present the following functions - standard()
- FWW()
- DGRP()
- asymptoticMK()
- iMK()
- completeMKT()
- loadPopFly()
- loadPopHuman()
- subsetPopData()
- multipleDatasets() - PopFlyAnalisys()
- theme_Publication()
- check_input()

Each one execute a test, perform the calculation or load presets to obtain the pipeline resutls. Rembember you always can access to the help, to check more examples or the passing arguments writting `{r}??` and the function in your console!

Standard MKT

The MK test (McDonald and Kreitman, 1991) was developed to be applied to protein coding sequences, combining both divergence (D) and polymorphism (P) sites, and categorizing mutations as synonymous (P s , D S) and non-synonymous (P n , D N). If all mutations are either strongly deleterious or neutral, then D N /D S is expected to roughly equal P s /P n . In contrast, if positive selection is operating in the region, adaptive mutations rapidly reach fixation and thus contribute relative more to divergence than to polymorphism when compared to neutral mutations, and then D N /D S >P n /P s . Assuming that adap- tive mutations contribute little to polymorphism but substantially to divergence, the proportion of non-synonymous substitutions than have been fixed by posi- tive selection can be inferred as $= 1$ - (P n /P s)(D S /D N) (??). The significance of effect can be easily quantified using a simple 2×2 contingency table (see Table 5), using a Fischer's exact test.

```
standard<-standard(daf = mydafdata,divergence = mydivergencedata)
```

```
standard$alpha.symbol
#> [1] 0.2364499
standard$`Fishers exact test P-value`
#> [1] 1.480943e-183
standard$`MKT table`
```

	Polymorphism	Divergence
Neutral class	45101	52537
Selected class	35816	54641

FWW correction

estimates can be biased by the segregation of slighlty deleterious substitutions. One method to partially controlled its effects is to remove low frequency polymorphisms from the analysis, as proposed by Fay et al. (2001). ***FWW(mydafdata,mydivergencedata)*** generate the output at the console.

```
FWW(daf = mydafdata,divergence = mydivergencedata)
#> $Results
#>                alpha.symbol Fishers exact test P-value
#> Cutoff =  0       0.2364499              1.480943e-183
#> Cutoff =  0.05    0.5409548               0.000000e+00
#> Cutoff =  0.1     0.5798139               0.000000e+00
#>
#> $`Divergence metrics`
#> $`Divergence metrics`$`Global metrics`
#>           Ka        Ks     omega
#> 1 0.02102543 0.0847345 0.2481331
#>
#> $`Divergence metrics`$`Estimates by cutoff`
#>                omegaA.symbol omegaD.symbol
#> Cutoff =  0       0.05867104     0.1894620
#> Cutoff =  0.05    0.13422877     0.1139043
#> Cutoff =  0.1     0.14387102     0.1042621
#>
#>
#> $`MKT tables`
#> $`MKT tables`$`Cutoff =  0`
#>
#>
#> Table: cutoff
#>
#>                   Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class            45101        52537
#> Selected class           35816        54641
#>
#> $`MKT tables`$`Cutoff =  0.05`
#>
#>
#> Table: cutoff
#>
#>                   Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class            27912        52537
#> Selected class           13326        54641
#>
#> $`MKT tables`$`Cutoff =  0.1`
#>
#>
```

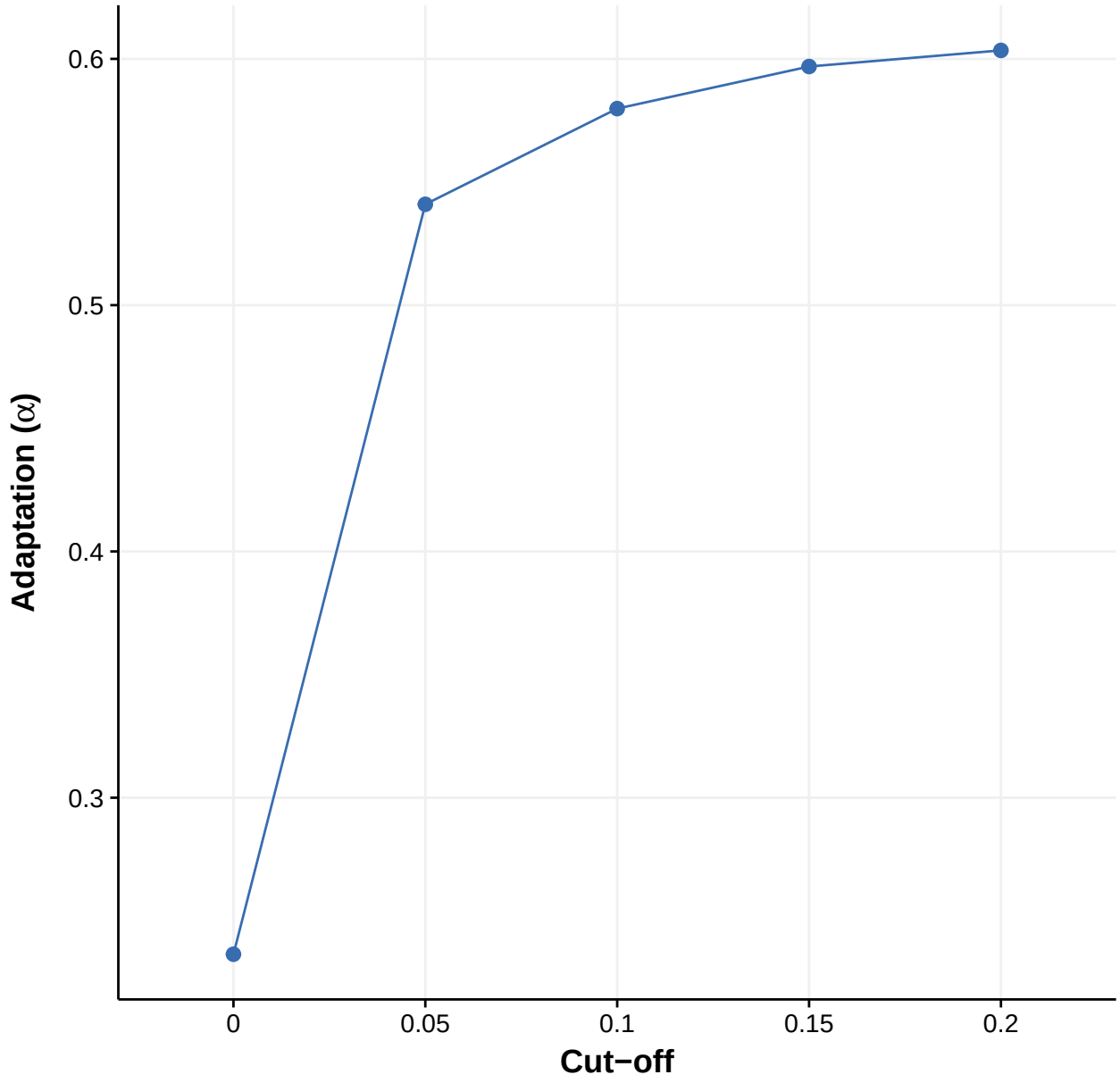
```
#> Table: cutoff
#>
#>                   Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class            23132        52537
#> Selected class           10109        54641
```

You could save it in a variable and the access to different data saved inside. Check it!

```
methodFWW<-FWW(daf = mydafdata,divergence = mydivergencedata)
methodFWW$Results
#>                alpha.symbol Fishers exact test P-value
#> Cutoff =  0       0.2364499              1.480943e-183
#> Cutoff =  0.05    0.5409548               0.000000e+00
#> Cutoff =  0.1     0.5798139               0.000000e+00
methodFWW$`MKT tables`
#> $`Cutoff =  0`
#>
#>
#> Table: cutoff
#>
#>                   Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class            45101        52537
#> Selected class           35816        54641
#>
#> $`Cutoff =  0.05`
#>
#>
#> Table: cutoff
#>
#>                   Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class            27912        52537
#> Selected class           13326        54641
#>
#> $`Cutoff =  0.1`
#>
#>
#> Table: cutoff
#>
#>                   Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class            23132        52537
#> Selected class           10109        54641
methodFWW$`Divergence metrics`
#> $`Global metrics`
#>           Ka        Ks     omega
#> 1 0.02102543 0.0847345 0.2481331
#>
#> $`Estimates by cutoff`
#>                omegaA.symbol omegaD.symbol
#> Cutoff =  0       0.05867104     0.1894620
#> Cutoff =  0.05    0.13422877     0.1139043
#> Cutoff =  0.1     0.14387102     0.1042621
```

By default the arguments ***list_cutoffs***, pass a list of cutoffs with the following values: c(0, 0.05, 0.1). And moreover th argument ***plot*** is setting in ***FALSE***. You can change the cutting values and switch on the plot to visulize your results!

```
methodFWW<-FWW(daf = mydafdata, divergence = mydivergencedata,list_cutoff=c(0, 0.05, 0.1,0.15,0.2),
               plot=TRUE)
savePlotInVariable<-methodFWW$Graph
savePlotInVariable
```



DGRP correction

The null hypothesis of neutrality is rejected in a MKT when DN/DS > Pn/Ps , inferring adaptation, but also when P n /P s >D N /D S . In this later case, there is an excess of polymorphism relative to divergence for the non-synonymous class n, due to (i) slightly deleterious variants segregating at low frequency in the population subject to weak negative selection, which contribute to polymorphism but not to divergence, or

(ii) relaxation of selection where sites previ- ously under strong or weak purifying selection have become neutral, causing an increased level of polymorphism relative to divergence Adaptive mutations and weakly deleterious selection act in opposite direc- tions on the MKT, so will be underestimated when the two selection regime occur. Because slightly deleterious mutations tend to segregate at lower fre- quencies than do neutral mutations, they can be partially controlled for by removing low frequency polymorphisms from the analysis, generally the 5% (Fay et al., 2001). However, this method is still expected to lead to biased estimates. To take adaptive and slightly deleterious mutation mutually into account, P n , the count of segregating sites in the non-synonymous class, should be sepa- rated into the number of neutral variants and the number of weakly deleterious variants, P n = P n-neutral +P n weakly del. . If both numbers are estimated, adaptive and weakly deleterious selection can be evaluated independently. Consider the following pair of 2×2 contingency tables (Table 5): The table on the left if the standard MKT table with the theoretical counts of segregating sites and divergent sites for each cell. The table on the right contains the count of P n and P s for two-frequency categories. The estimate of the fraction of sites segregating neutrally within the MAF (minor allele frequency) < 5% (f neutral MAF<5%) is f neutral MAF<5% =P s MAF<5% /P s . The expected number of segregating sites in the non-synonymous class which are neutral within the MAF<5% is P neutral MAF<5% =P n ×f neutral MAF<5% . The ex- pected number of neutral segregating sites in the non-synonymous class is P n neutral =P neutral MAF<5% +P n MAF>5%. To estimate from the standard MKT table correcting by the segregationof weakly deleterious variants, we have to substitute the P n by the expected number of neutral segregating sites, P neutral . The correct estimate of is then = 1 - (P n neutral/Ps)(DS/DN)

```
DGRP(daf = mydafdata, divergence = mydivergencedata)
#> $Results
#>                 alpha.symbol Fishers exact test P-value
#> Cutoff =  0       0.2364499              1.480943e-183
#> Cutoff =  0.05    0.4249071               0.000000e+00
#> Cutoff =  0.2     0.3842950               0.000000e+00
#>
#> $`Divergence metrics`
#> $`Divergence metrics`$`Global metrics`
#>           Ka        Ks     omega
#> 1 0.02102543 0.0847345 0.2481331
#>
#> $`Divergence metrics`$`Estimates by cutoff`
#>                 omegaA.symbol omegaD.symbol
#> Cutoff =  0       0.05867104     0.1894620
#> Cutoff =  0.05    0.10543351     0.1426996
#> Cutoff =  0.2     0.09535630     0.1527768
#>
#>
#> $`MKT tables`
#> $`MKT tables`$`Number of segregating sites by DAF category - Cutoff =  0`
#>
#>
#> Table: cutoff
#>
#>                   DAF.below.cutoff   DAF.above.cutoff
#> ----------------  -----------------  -----------------
#> Neutral class                     0              45101
#> Selected class                    0              35816
#>
#> $`MKT tables`$`Number of segregating sites by DAF category - Cutoff =  0.05`
#>
#>
```

```
#> Table: cutoff
#>
#>                   DAF.below.cutoff   DAF.above.cutoff
#> ---------------  -----------------  -----------------
#> Neutral class                17189              27912
#> Selected class               22490              13326
#>
#> $`MKT tables`$`Number of segregating sites by DAF category - Cutoff =  0.2`
#>
#>
#> Table: cutoff
#>
#>                   DAF.below.cutoff   DAF.above.cutoff
#> ---------------  -----------------  -----------------
#> Neutral class                26931              18170
#> Selected class               28322               7494
#>
#> $`MKT tables`$`MKT standard table`
#>
#>
#>                   Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class            45101        52537
#> Selected class           35816        54641
#>
#>
#> $Fractions
#>          0       0.05        0.2
#> d 0.810538 0.81053943 0.81053627
#> f 0.189462 0.14269958 0.15277678
#> b 0.000000 0.04676099 0.03668695
```

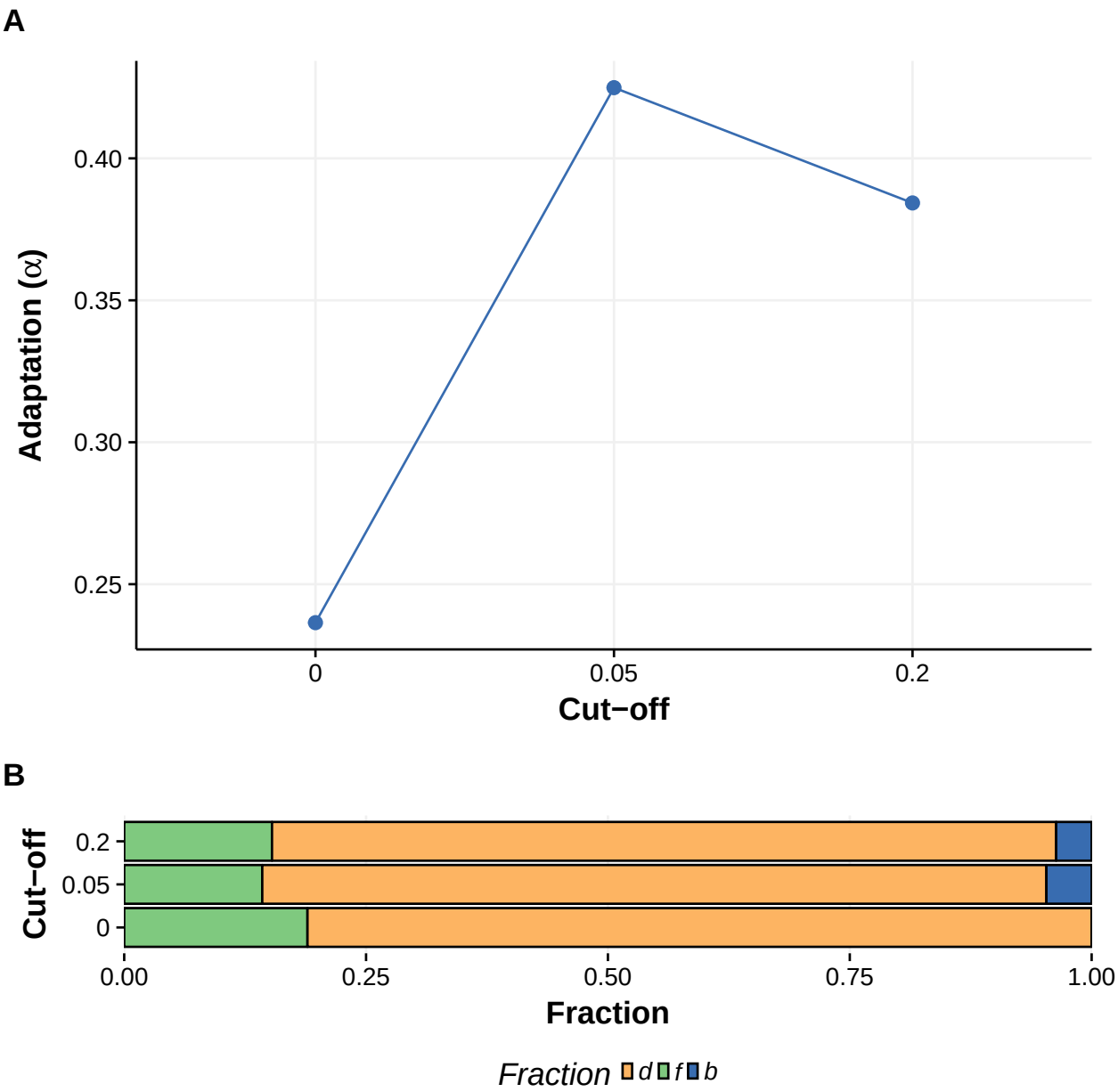
You could save it in a variable and the access to different data saved inside. Check it!

```
methodDGRP<-DGRP(daf = mydafdata, divergence = mydivergencedata)
methodDGRP$Results
#>                alpha.symbol Fishers exact test P-value
#> Cutoff =  0       0.2364499              1.480943e-183
#> Cutoff =  0.05    0.4249071               0.000000e+00
#> Cutoff =  0.2     0.3842950               0.000000e+00
methodDGRP$Fractions
#>          0       0.05        0.2
#> d 0.810538 0.81053943 0.81053627
#> f 0.189462 0.14269958 0.15277678
#> b 0.000000 0.04676099 0.03668695
methodDGRP$`MKT tables`
#> $`Number of segregating sites by DAF category - Cutoff =  0`
#>
#>
#> Table: cutoff
#>
#>                   DAF.below.cutoff   DAF.above.cutoff
#> ---------------  -----------------  -----------------
#> Neutral class                    0              45101
#> Selected class                   0              35816
```

```
#>
#> $`Number of segregating sites by DAF category - Cutoff =  0.05`
#>
#>
#> Table: cutoff
#>
#>                  DAF.below.cutoff   DAF.above.cutoff
#> ---------------  -----------------  -----------------
#> Neutral class                17189              27912
#> Selected class               22490              13326
#>
#> $`Number of segregating sites by DAF category - Cutoff =  0.2`
#>
#>
#> Table: cutoff
#>
#>                  DAF.below.cutoff   DAF.above.cutoff
#> ---------------  -----------------  -----------------
#> Neutral class                26931              18170
#> Selected class               28322               7494
#>
#> $`MKT standard table`
#>
#>
#>                  Polymorphism   Divergence
#> ---------------  -------------  -----------
#> Neutral class           45101        52537
#> Selected class          35816        54641
```

By default the arguments ***list_cutoffs***, pass a list of cutoffs with the following values: c(0, 0.05, 0.1). And moreover th argument ***plot*** is setting in ***FALSE***. You can change the cutting values and switch on the plot to visulize your results!

```
methodDGRP<-DGRP(daf = mydafdata, divergence = mydivergencedata,list_cutoff=c(0, 0.05,0.2),plot=TRUE)
savePlotInVariable<-methodDGRP$Graph
savePlotInVariable
```



Asymptotic MKT

Petrov reference + explanation

```
asymptoticMK(daf = mydafdata, divergence = mydivergencedata, xlow = 0,xhigh = 0.9)
#>          model         a         b         c alpha_asymptotic    CI_low
#> 1 exponential 0.6258904 -1.395108 18.96187        0.6258904 0.6039621
#>    CI_high alpha_original
#> 1 0.647417      0.2157308
```

iMK

Asymptotic explanation + Sergi slightly deleterious approach

```
iMK(daf = mydafdata, divergence = mydivergencedata, xlow = 0, xhigh = 0.9)
#> $`Asymptotic MK table`
```

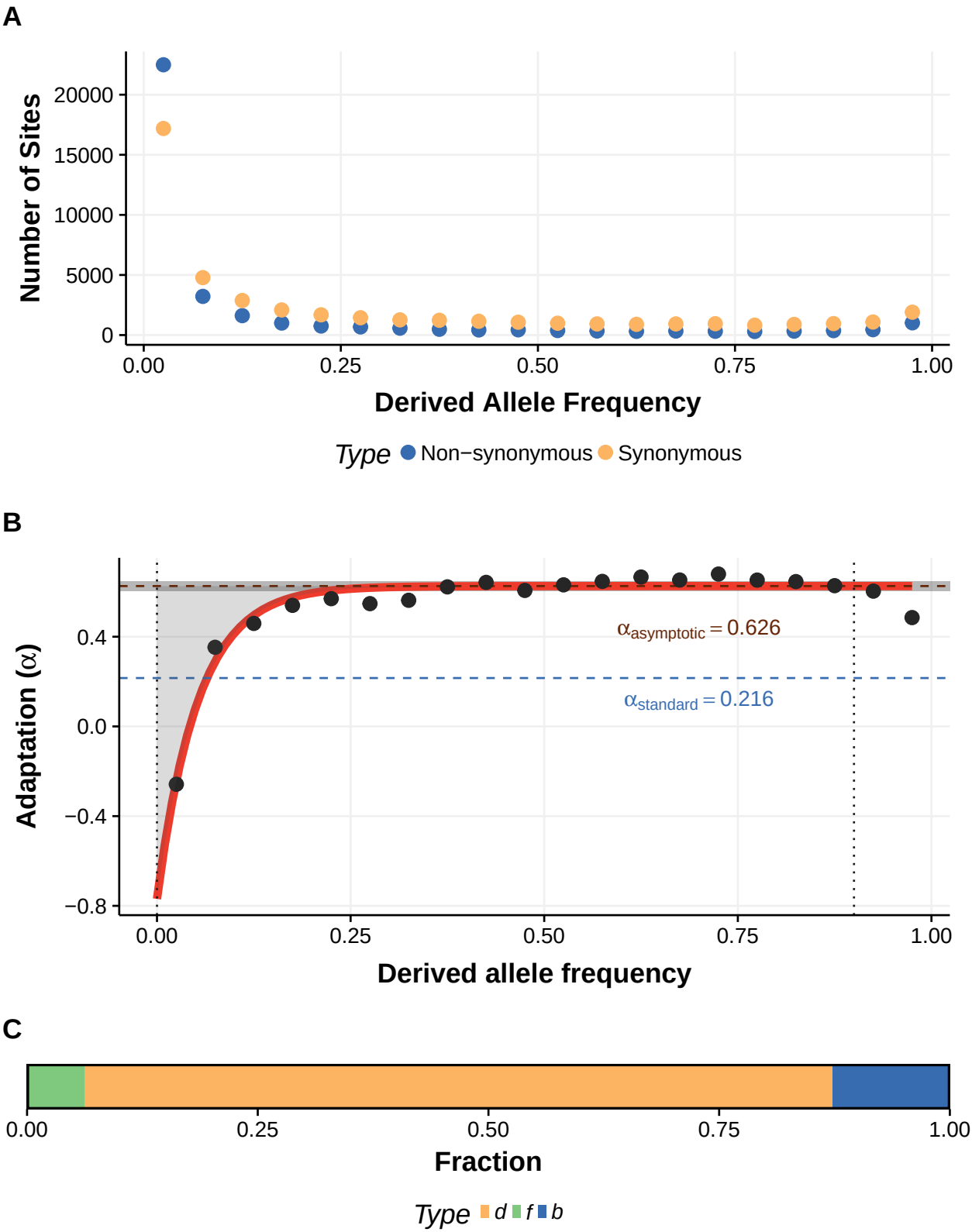
```
#>         model      a       b       c alpha_asymptotic CI_low CI_high
#> 1 exponential 0.6259 -1.3951 18.9619           0.6259 0.6042  0.6467
#>   alpha_original
#> 1         0.2157
#>
#> $`Fractions of sites`
#>   Type   Fraction
#> 1    d 0.81053796
#> 2    f 0.06232362
#> 3    b 0.12713842
```

You could save it in a variable and the access to different data saved inside. Check it!

```
methodiMK<-iMK(daf = mydafdata,divergence = mydivergencedata, xlow = 0, xhigh = 0.9)
methodiMK$Results
#> NULL
methodiMK$`Divergence metrics`
#> NULL
methodiMK$`MKT tables`
#> NULL
```

By default the argument ***plot*** is setting in ***FALSE***. You can change the cutting values and switch on the plot to visulize your results!

```
methodiMK<-iMK(daf = mydafdata, divergence = mydivergencedata, xlow = 0, xhigh = 0.9 ,plot=TRUE)
savePlotInVariable<-methodiMK$Graph
savePlotInVariable
```



If you have a bunch of data like the following, or simply have several genes datasets: Maybe you want to perform some test or compare the test results between your datasets. You could execute the funtion multipleDatasets, putting your datasets in a directory a name them with the extensions **ID.daf.txt/ID.divergence.txt**. Then execute the following commands to perform the tests:

The idList argument allow to the user pass a plain text file with the IDs, in the case you want to subset the analysis to just a few datasets. It is used when fullAnalysis = FALSE, list of IDs to analyze