DichotomousKey for R

Hao Li

2019-02-20

Feel free to share this open source package. For future support or report bugs or writing feedbacks, please contact me by email or on GitHub.

Thank you for your support!

https://github.com/HaoLi111/dichotomousKey

Fitness for purpose

Dichotomous key can be helpful for classification and is important for keeping track of phenotypical characteristics of branches of species.

Manipulatig classification can be hard for amatours, and if the scale is large it will take lots of people to maintain the resource. The need for a software package to organize, edit, load, reform and utilize classificatin tables may be underestimated.

Installation

If you have not install devtools from R, type

```
install.packages('devtools')
```

From R (devtools), type

```
devtools:install_github('HaoLi111/dichotomousKey')
```

to install the current development version.

Utilization

First, load the package with require().

```
require(dichotomousKey)

## Loading required package: dichotomousKey
```

The default dk table

There is a very simplified dk table that we can start with

```
dk_eg

##   id                 P           G ref
## 1  1       plants like      plants   2
## 2  1       animal like      animal   3
## 3  2       woody trunk        tree   0
## 4  2         soft stem      flower   0
## 5  3 live under water        fish   0
## 6  3      live on land land_animal   4
## 7  4              hair      mammal   0
## 8  4          feathers        bird   0
```

Type ?dk_eg to know more, you can also import your own data as data frames (please specify stringsAsFactors = FALSE).

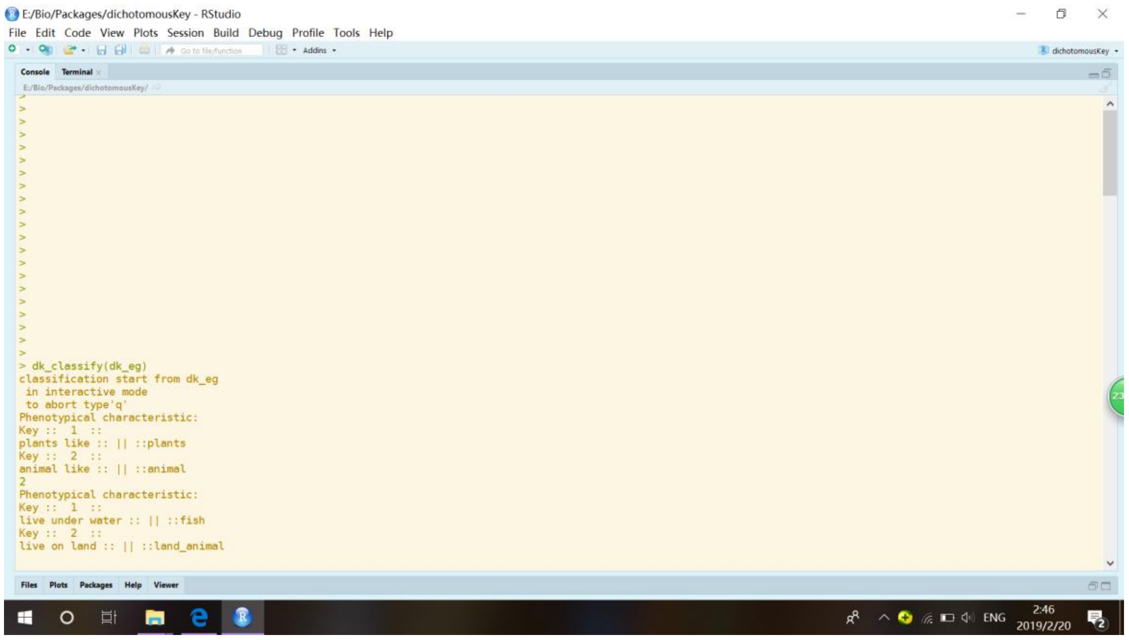
- P stands for Phenotypical data
- G stands for Classification data(Genotype related)
- id is the index , ref is the reference
- You can attach more columns to the right, e.g. img , links, wav …

Identification

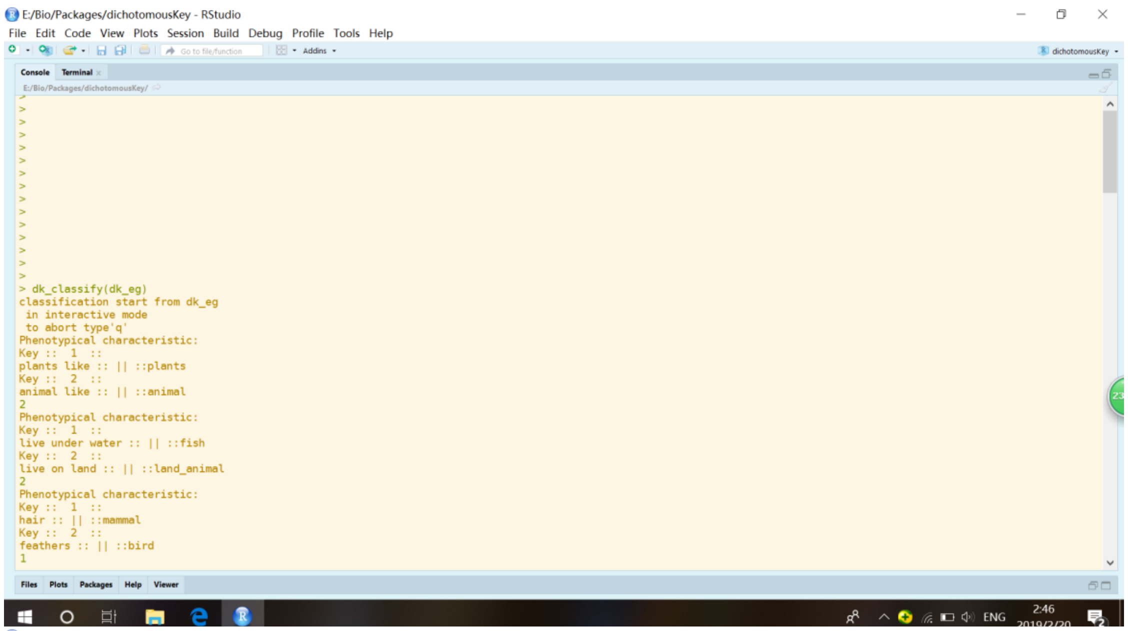
The interactive mode can be activated with the function dk_classify().To classify from your own imported dataset, specify the name of your dataset as the 1st argument of the function.

```
dk_classify(dk_eg)
```

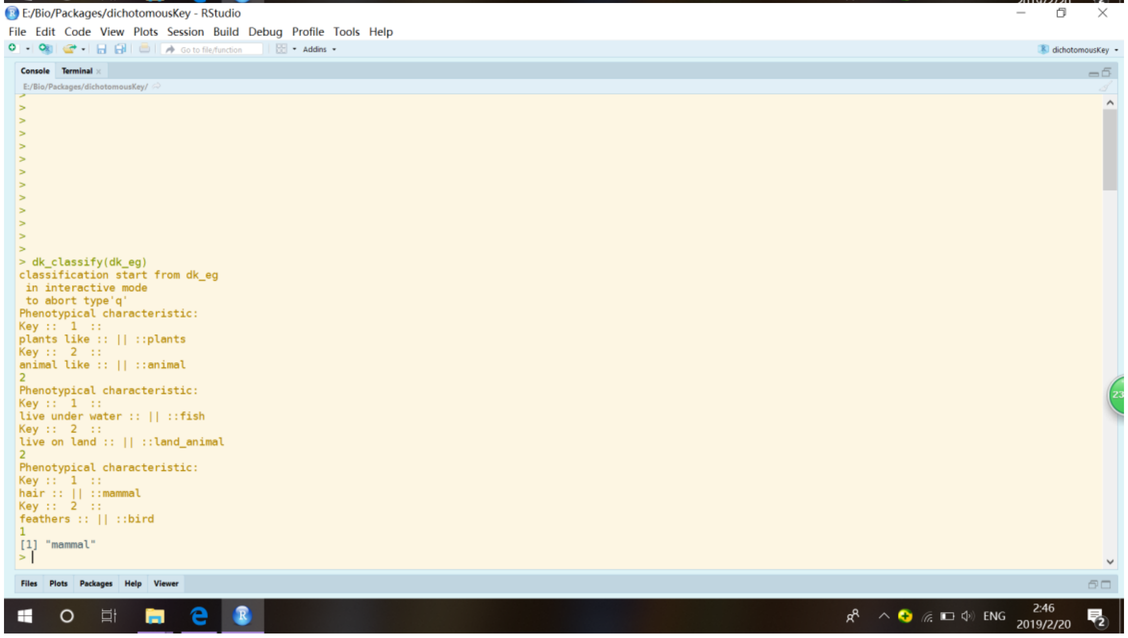
Then use it as a CLI program, but with your own specified data. Extenstions of rich version of data may include showing images or playing audios, which may be enabled in the future.


E:/Bio/Packages/dichotomousKey - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
dichotomousKey
Console Terminal
E:/Bio/Packages/dichotomousKey/
Files Plots Packages Help Viewer
ENG
2:46
2019/2/20

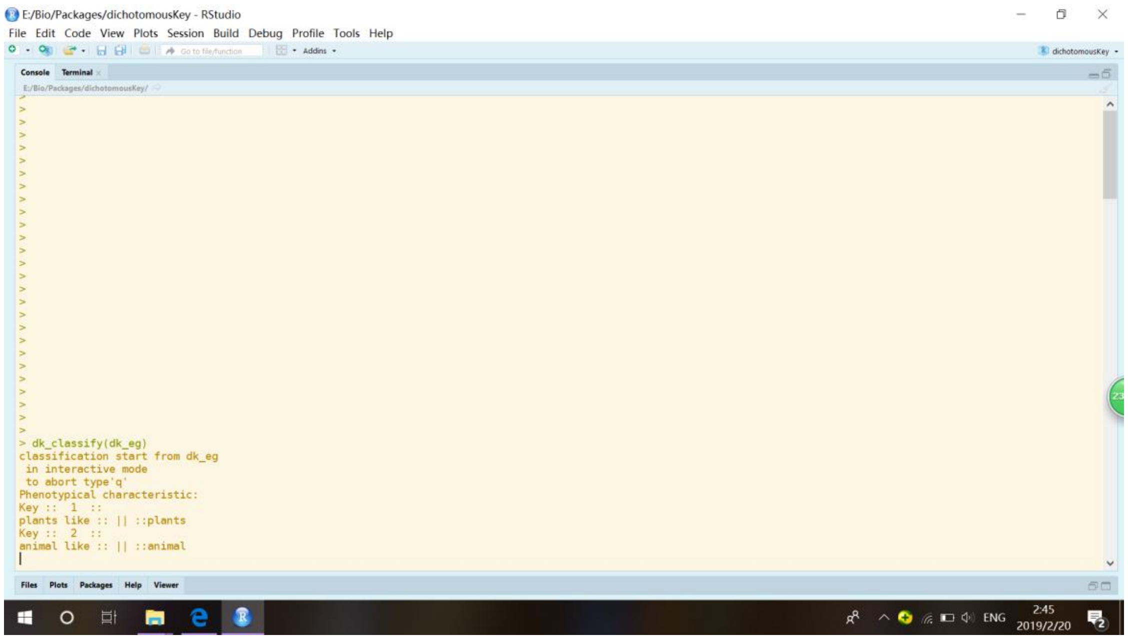
```
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
> dk_classify(dk_eg)
classification start from dk_eg
 in interactive mode
 to abort type'q'
Phenotypical characteristic:
Key ::  1  ::
plants like :: || ::plants
Key ::  2  ::
animal like :: || ::animal
2
Phenotypical characteristic:
Key ::  1  ::
live under water :: || ::fish
Key ::  2  ::
live on land :: || ::land_animal
```


E:/Bio/Packages/dichotomousKey - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal
Files Plots Packages Help Viewer

```
>
>
>
>
>
>
>
>
>
>
>
>
>
>
> dk_classify(dk_eg)
classification start from dk_eg
 in interactive mode
 to abort type'q'
Phenotypical characteristic:
Key ::  1  ::
plants like :: || ::plants
Key ::  2  ::
animal like :: || ::animal
2
Phenotypical characteristic:
Key ::  1  ::
live under water :: || ::fish
Key ::  2  ::
live on land :: || ::land_animal
2
Phenotypical characteristic:
Key ::  1  ::
hair :: || ::mammal
Key ::  2  ::
feathers :: || ::bird
1
```

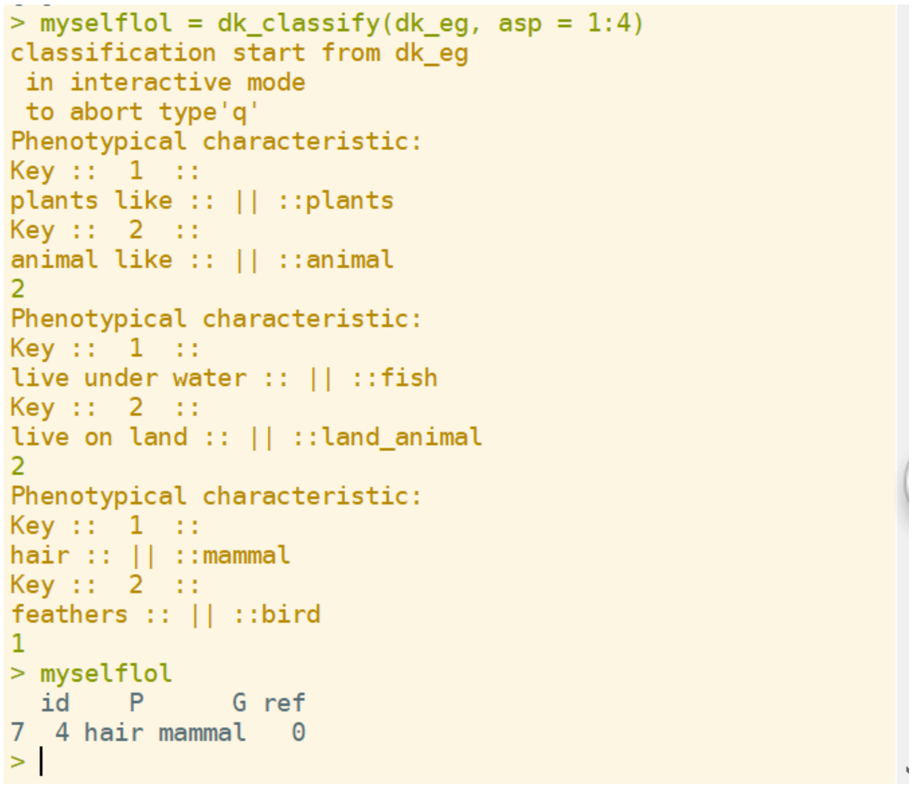

E:/Bio/Packages/dichotomousKey - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal
Files Plots Packages Help Viewer

```
>
>
>
>
>
>
>
>
>
>
>
>
> dk_classify(dk_eg)
classification start from dk_eg
 in interactive mode
 to abort type'q'
Phenotypical characteristic:
Key ::  1  ::
plants like :: || ::plants
Key ::  2  ::
animal like :: || ::animal
2
Phenotypical characteristic:
Key ::  1  ::
live under water :: || ::fish
Key ::  2  ::
live on land :: || ::land_animal
2
Phenotypical characteristic:
Key ::  1  ::
hair :: || ::mammal
Key ::  2  ::
feathers :: || ::bird
1
[1] "mammal"
>
```


E:/Bio/Packages/dichotomousKey - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal
Files Plots Packages Help Viewer

```
>
> dk_classify(dk_eg)
classification start from dk_eg
 in interactive mode
 to abort type'q'
Phenotypical characteristic:
Key ::  1  ::
plants like :: || ::plants
Key ::  2  ::
animal like :: || ::animal
```

The program has output as character 'mammal', you can let it return the entire row by typing asp = 1:4 and then assign it to a variable, in this case, I name it : myselflol.



```
> myselflol = dk_classify(dk_eg, asp = 1:4)
classification start from dk_eg
 in interactive mode
 to abort type'q'
Phenotypical characteristic:
Key ::  1  ::
plants like :: || ::plants
Key ::  2  ::
animal like :: || ::animal
2
Phenotypical characteristic:
Key ::  1  ::
live under water :: || ::fish
Key ::  2  ::
live on land :: || ::land_animal
2
Phenotypical characteristic:
Key ::  1  ::
hair :: || ::mammal
Key ::  2  ::
feathers :: || ::bird
1
> myselflol
  id    P      G ref
7  4 hair mammal   0
> 
```

Phenotype extraction

Given a species,extract its phenotypical characteristrics from a table(reverse reference).

In this case find 'mammal'

```
dk_extract(dk_eg,Gtarget = 'mammal')
```

```
##                P           G
## 2  animal like      animal
## 6 live on land land_animal
## 7         hair      mammal
```

So that a mammal is animal like, lives on land, and has hair according to the example data frame.

Conversion

You can convert data.frame like dk table to lists and vice versa. You can extend/conjunct tables with this tactic. First convert dk_eg to list called list_eg, print the structure of it.

It may look messy. Don't worry at this stage.

```
list_eg = dk_as_list(dk_eg)
str(list_eg)
```

```
## List of 3
##  $      :List of 4
##   ..$      :List of 2
##   .. ..$ pause: logi TRUE
##   .. ..$ prim :'data.frame': 1 obs. of  2 variables:
##   .. .. ..$ P: chr "woody trunk"
##   .. .. ..$ G: chr "tree"
##   ..$      :List of 2
##   .. ..$ pause: logi TRUE
##   .. ..$ prim :'data.frame': 1 obs. of  2 variables:
##   .. .. ..$ P: chr "soft stem"
##   .. .. ..$ G: chr "flower"
##   ..$ pause: logi FALSE
##   ..$ prim :'data.frame':    1 obs. of  2 variables:
##   .. ..$ P: chr "plants like"
##   .. ..$ G: chr "plants"
##  $      :List of 4
##   ..$      :List of 2
##   .. ..$ pause: logi TRUE
##   .. ..$ prim :'data.frame': 1 obs. of  2 variables:
##   .. .. ..$ P: chr "live under water"
##   .. .. ..$ G: chr "fish"
##   ..$      :List of 4
##   .. ..$      :List of 2
##   .. .. ..$ pause: logi TRUE
##   .. .. ..$ prim :'data.frame':  1 obs. of  2 variables:
##   .. .. .. ..$ P: chr "hair"
##   .. .. .. ..$ G: chr "mammal"
##   .. ..$      :List of 2
##   .. .. ..$ pause: logi TRUE
##   .. .. ..$ prim :'data.frame':  1 obs. of  2 variables:
##   .. .. .. ..$ P: chr "feathers"
##   .. .. .. ..$ G: chr "bird"
##   .. ..$ pause: logi FALSE
##   .. ..$ prim :'data.frame': 1 obs. of  2 variables:
##   .. .. ..$ P: chr "live on land"
##   .. .. ..$ G: chr "land_animal"
##   ..$ pause: logi FALSE
##   ..$ prim :'data.frame':    1 obs. of  2 variables:
##   .. ..$ P: chr "animal like"
##   .. ..$ G: chr "animal"
##  $ pause: logi FALSE
```

You can then edit the list.

- you can directly edit with

```
     list_eg<-edit(list_eg)
```

- or with R, append another frame.

For, example, I got another frame classifying bird,(of course I recommend that you edit with excel and the import to R)

```
dk_bird = data.frame(id = c(1,1),P = c('Big','Small'),G=  c('Big Bird',
'Small Bird'),ref = c(0,0),stringsAsFactors = F)
list_bird = dk_as_list(dk_bird)
str(list_bird)

## List of 3
##  $      :List of 2
##   ..$ pause: logi TRUE
##   ..$ prim :'data.frame':    1 obs. of  2 variables:
##   .. ..$ P: chr "Big"
##   .. ..$ G: chr "Big Bird"
##  $      :List of 2
##   ..$ pause: logi TRUE
##   ..$ prim :'data.frame':    1 obs. of  2 variables:
##   .. ..$ P: chr "Small"
##   .. ..$ G: chr "Small Bird"
##  $ pause: logi FALSE
```

Append it, and ...Bang!

```r
list_new = list_eg
list_bird$prim = list_new[[2]][[2]][[2]]$prim
list_new[[2]][[2]][[2]] = list_bird
list_new[[2]][[2]][[2]] = as.list(list_new[[2]][[2]][[2]])
```

Finnally, convert it back, and start our new round of classification.

```r
dk_new = list_as_dk(list_new)
dk_new

##       id  P                  G              ref
##  [1,] "1" "plants like"      "plants"       "2"
##  [2,] "1" "animal like"      "animal"       "3"
##  [3,] "2" "woody trunk"      "tree"         "0"
##  [4,] "2" "soft stem"        "flower"       "0"
##  [5,] "3" "live under water" "fish"         "0"
##  [6,] "3" "live on land"     "land_animal"  "4"
##  [7,] "4" "hair"             "mammal"       "0"
##  [8,] "4" "feathers"         "bird"         "5"
##  [9,] "5" "Big"              "Big Bird"     "0"
## [10,] "5" "Small"            "Small Bird"   "0"
```

Extension

Feel free to share this open source package. For future support or report bugs or writing feedbacks, please contact me by email or on GitHub.

Thank you for your support!

https://github.com/HaoLi111/dichotomousKey