1 Usage and examples

```
### This tests the result of the first example of the article
print("##############################")
print("This tests the result of the first example of the article \n")
dd2=matrix(c(4,0,0,0,30,0,0,0,23),ncol=3,byrow=TRUE)
callhaplotype(dd2)
callhaplotype(dd2)/(2*57)
### This tests the cubic routine
print("##############################")
print("This tests the cubic routine")
haplotypeit(4,0,0,0,30,0,0,0,23)
### Formated of 4 digits
print("Formated of 4 digits")
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$A)),digit=4)
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$B)),digit=4)
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$C)),digit=4)
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$D)),digit=4)
### The second example
print("##############################")
print("The second example: \n")
dd=matrix(c(1212, 2, 0, 679, 0,0,75,0,0), byrow=TRUE, nrow=3)
colnames(dd)=c("CC","CT","TT")
rownames(dd)=c("CC","CT","TT")
callhaplotype(dd)
### Check the result of the cubic equation of the second example
print("##############################")
print("Check the result of the cubic equation of the second example: \n")
temp2haplo =as.numeric(t(dd));
haplotypeit(temp2haplo[1],temp2haplo[2],temp2haplo[3],temp2haplo[4],temp2haplo[5]
            ,temp2haplo[6],temp2haplo[7],temp2haplo[8],temp2haplo[9]);
rm(temp2haplo)
### Third example
print("##############################")
print("Third example : \n")
dd3=matrix(c(1030,678,123,1,1,0,0,0,0),ncol=3,byrow=TRUE)
colnames(dd3)=c("AA","AG","GG")
rownames(dd3)=c("CC","CT","TT")
```

```
callhaplotype(dd3)
###  Check for alternative solutions
print("##############################")
print("Check for alternative solutions: \n")
temp2haplo =as.numeric(t(dd3));
haplotypeit(temp2haplo[1],temp2haplo[2],temp2haplo[3],temp2haplo[4],temp2haplo[5]
            ,temp2haplo[6],temp2haplo[7],temp2haplo[8],temp2haplo[9]);
rm(temp2haplo)
print("##############################")
```

2 Add ons within the package structure

2.0.1 For application user

Below the library path of the enviroment variable **R_LIBS_USER** with **package name**. In the subdirectory **inst** there are some example of R examples: **testcalles.R**. There is also a programm using the package compiler implemented within base to use a **bit compiler** for a faster version. The advantage of this method is to allow the user to modify and read the routine within R rather tahn using C classes and C libraries.
Load the package into an empty workspace and copy the R comands of CompilerScripts. If no error message is printed you are finished.

2.0.2 For R application programmer

Evenso this method is slower than plan C code, it allowes almost as fast modifications. User who need mode speed might change some loops to apply using the exaple of **?cmpfun**. This might fasten the algorithm, but increases storage usage. However for whole genome analysis on a server it might be worth it. Using a PC it might not be worth it. For those who need the last speed amounts of this program you might look up the parameter **mode** to get control of the jitter routines and also take care about the **garbagetorture** for active storage managemant. If you exeed high speed stoarge limit all processes will decrease in speed. To store results effectively you might use the package SQLite to buffer results before writing the results in a data base. Please take care to create indices of the database after filling it. This avoid double sorting stepts within the data of unknown size.

2.1 Simple example

```
###  This R programm compiles the R source function and stores the
###  compiled function on the same name as the source code.
###  Using package compiler which is inplemented in bases package
###  of R Version > 3.1.3.
library(compiler)
bcubic=cmpfun(cubic)
bhaplotypeit=cmpfun(haplotypeit)
boptimalfrequency=cmpfun(optimalfrequency)
bfindoptimal=cmpfun(findoptimal)
bcallhaplotype=cmpfun(callhaplotype)
#### The optimization level can be choosen as 3.
#### Because all steps are evaluated.
#### However the increase in speed is fine anyway.
cubic=bcubic
haplotypeit=bhaplotypeit
optimalfrequency=boptimalfrequency
findoptimal=bfindoptimal
callhaplotype=bcallhaplotype
####
```

3 Literature

[1], [4], [5]. [3], [2],

Literatur

[1] David Clayton. An r package for analysis of whole-genome association studies. *Human Heredity*, 64(1):45 – 51, 2007. doi: doi:10.1001/archgenpsychiatry.2010.25. URL http://archpsyc.jamanetwork.com/article.aspx?articleid=210679.

[2] Nathan Jacobson. *Basic Algebra I: Second Edition (Dover Books on Mathematics)*. Dover Publication, 2009.

[3] Montgomery Slatkin Laurent Excofie. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population.

Molecular biology and evolution, 12(5):921 –927, 1995. URL http://mbe.oxfordjournals.org/content/12/5/921.full.pdf.

[4] Tianhua Niu. Algorithms for inferring haplotypes. *Genetic Epidemiology*, 27:334–347, 2004. doi: DOI: 10.1002/gepi.20024. URL http://biostat.gru.edu/Journal%20Club/Niu_2004.pdf.

[5] Werner A. Stahel. *Statistische Datenanalyse Eine*. Vieweg Verlag, 2002.