name{CPWFABC}

packageTitle{Changing selection in time-sampled data }

packageDescription{detects allele trajectories of changing selection from those of constant selection using ABC model choice, and jointly estimates the position of a change point as well as the strength of both corresponding selection coefficients (and dominance for diploid cases) using Wright-Fisher ABC methods}

input{

```
# variable inputs
N               population size (number of chromosomes: N
                individuals for haploids and N/2 individuals for
                diploids) (1000 by default)
t               total number of generations (100 by default)
t0              time where the mutation appears in generations (1
                for from the beginning by default)
s_start         time in generation when selection starts (s=0
                before s_start) (1 for from the beginning by
                default)
sample_times    vector of exact sampling times in generations
N_sample        vector of sample sizes in number of chromosomes
min_freq        data ascertainment of reaching a given minimum
                frequency at one of the sampling time points (0
                for no condition and 1 to condition on fixation)
N_allele        data frame of observed SNP numbers (row=observed
                allele, column=sampling)
max_sims        maximum number of simulations to do before giving
                up (1 by default)
no_sim          number of simulated datasets to be createdn (1e6
                by default)
best_sim        number of best simulations to be used for
                estimation and model choice (1e3 by default)
set_seed        reproducible numbers (TRUE by default)
post_graph      Posterior densities of M0 and M1 (FALSE by
                default)
post_2D_M1      2D posteriors of M1 estimates (s1&s2, s1&CP,
                s2&CP) (FALSE by default)
h_fixed         h to be fixed in diploid populations (TRUE by
                default)
```

```
 h_given        h to be used if fixed (0.5 by default)

 # fixed inputs
 ploidy         1 for haploids, 2 for diploids
 j              number of A alleles appearing at time t0 in the
                population (j=1 if initial frequency < min_freq,
                j=initial frequency otherwise)
}

output{
 Text files of simulated parameters
 Text files of simulated summary statistics
 PDFs of prior graphs for simulated parameters
 Text files of summary of results (Allele   M1 posteriorBayes
 Factor M0 estimate of s1   M0 estimate of h    M1 estimate of s1   M1 estima
 PDFs of posterior graphs of parameters of interest from observed
 data
}

functions{
WF_2s_simulator{
 usuage
{simulates a Wright-Fisher trajectory with changing selection or
 constant selection}
 arguments
 {N,t,fluc_t,j,t0,s1,s2,h,s_start,ploidy,N_sample,sample_times,ma
 x_sims}
    }

CP_WFABC_diploid_estimator{
 usuage
{jointly estimates the position of a change point as well as the
 strength of both corresponding selection coefficients and
 dominance for a diploid population using Wright-Fisher ABC
 methods}
 arguments
 {N,t,t0,h_fixed,h_given,s_start,sample_times,N_sample,N_allele,m
 in_freq,max_sims,no_sim,best_sim,set_seed=TRUE,post_graph,post_2
 D_M1}
    }

CP_WFABC_diploid_modelchoice{
 usuage
```

{detects allele trajectories of changing selection from those of constant selection using ABC model choice, and jointly estimates the position of a change point as well as the strength of both corresponding selection coefficients and dominance for a diploid population using Wright-Fisher ABC methods}
arguments
{N,t,t0,h_fixed,h_given,s_start,sample_times,N_sample,N_allele,min_freq,max_sims,no_sim,best_sim,set_seed=TRUE,post_graph,post_2D_M1}
}

CP_WFABC_haploid_estimator{
usuage
{jointly estimates the position of a change point as well as the strength of both corresponding selection coefficients for a haploid population using Wright-Fisher ABC methods}
arguments
{N,t,t0,s_start,sample_times,N_sample,N_allele,min_freq,max_sims,no_sim,best_sim,set_seed=TRUE,post_graph,post_2D_M1}
}

CP_WFABC_haploid_modelchoice{
usuage
{detects allele trajectories of changing selection from those of constant selection using ABC model choice, and jointly estimates the position of a change point as well as the strength of both corresponding selection coefficients for a haploid population using Wright-Fisher ABC methods}
arguments
{N,t,t0,s_start,sample_times,N_sample,N_allele,min_freq,max_sims,no_sim,best_sim,set_seed=TRUE,post_graph,post_2D_M1}
}
}

packageAuthor{Hyunjin Shim}

packageMaintainer{Hyunjin Shim <jinenstar@gmail.com>}

references{
Shim, H., Laurent, S., Foll, M.*, Jeffrey D Jensen* (submitted) Detecting and quantifying changing selection intensities from time-sampled polymorphism data.
Foll, M.*, Shim, H.*, & Jeffrey D Jensen (2014b) WFABC: A

Wright-Fisher ABC-Based Approach for Inferring Effective Population Sizes and Selection Coefficients from Time-Sampled Data. Molecular Ecology Resources.
Foll, M., Poh, Y.-P., Renzette, N., Ferrer-Admetlla, A., Bank, C., Shim, H., Malaspinas, A.S., Ewing, G., Liu, P., Wegmann, D., Caffrey, D.R., Zeldovich, K.B., Bolon, D.N., Wang, J.P., Kowalik, T.F., Schiffer, C.A., Finberg, R.W. & Jensen, J.D. (2014a) Influenza Virus Drug Resistance: A Time-Sampled Population Genetics Perspective. PLoS Genetics.
}

keyword{ population genetics; fluctuating selection; change point analysis; time-sampled data; approximate Bayesian computation; Wright-Fisher model; experimental design }

code{http://jensenlab.epfl.ch/page-86730-en.html}

examples{

```
# Example: WF_2s_simulator - simulates WF trajectories with a change in selection intensity from s1 to s2 at fluc_t
source("WF_2s_simulator.R")
N=1000
t=100
fluc_t=50
j=1
t0=1
s1=0.3
s2=-0.3
h=0.5
s_start=1
ploidy=1
N_sample=rep(100,10)
sample_times=c(1,12,23,34,45,56,67,78,89,100)
max_sims=1
WF_trajectory(N,t,fluc_t,j,t0,s1,s2,h,s_start,ploidy,N_sample,sample_times,max_sims)


# Example: CP_WFABC_diploid_estimator - estimates parameters of interest (s1, s2, CP and h) for diploid populations -> Panaxia dominula as an example with h estimated
source("CP_WFABC_diploid_estimator.R")
N=500
```

```
t=60
t0=1
h_fixed=FALSE
h_given=1
s_start=1
sample_times=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
 ,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,49,50,
 51,52,53,54,55,56,57,58,59,60)
N_sample=c(234,922,410,538,992,744,1972,2682,1932,1016,2388,1162,
 3042,2178,2328,630,2616,3224,2766,960,378,344,46,118,62,162,74,1
 00,262,1092,950,1292,680,462,1694,114,340,24,80,1244,1634,1050,4
 132,4580,966,452,284,664,910,828,838)
N_a1=c(26,63,22,30,45,48,84,100,69,29,88,29,108,56,67,7,78,148,10
 2,21,7,7,1,1,0,2,0,0,3,38,31,9,5,1,11,2,3,0,1,6,19,7,34,45,23,8,
 8,10,17,5,11)
N_allele=rbind.data.frame(N_a1)
min_freq=0.02
max_sims=1
no_sim=100000
best_sim=1000
post_2D_M1=TRUE
post_graph=TRUE
CP_WFABC_diploid_estimator(N,t,t0,h_fixed,h_given,s_start,sample_
 times,N_sample,N_allele,min_freq,max_sims,no_sim,best_sim,set_se
 ed=TRUE,post_graph,post_2D_M1)


# Example: CP_WFABC_diploid_modelchoice - detects changing
 selection trajectories using model choice and estimates
 parameters of interest (s1, s2, CP and h) for diploid
 populations -> Panaxia dominula as an example with h fixed
source("CP_WFABC_diploid_modelchoice.R")
N=50
t=60
t0=1
h_fixed=TRUE
h_given=0.5
s_start=1
sample_times=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
 ,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,49,50,
 51,52,53,54,55,56,57,58,59,60)
N_sample=c(234,922,410,538,992,744,1972,2682,1932,1016,2388,1162,
 3042,2178,2328,630,2616,3224,2766,960,378,344,46,118,62,162,74,1
 00,262,1092,950,1292,680,462,1694,114,340,24,80,1244,1634,1050,4
 132,4580,966,452,284,664,910,828,838)
N_a1=c(26,63,22,30,45,48,84,100,69,29,88,29,108,56,67,7,78,148,10
 2,21,7,7,1,1,0,2,0,0,3,38,31,9,5,1,11,2,3,0,1,6,19,7,34,45,23,8,
 8,10,17,5,11)
N_allele=rbind.data.frame(N_a1)
min_freq=0.02
max_sims=1
no_sim=100000
best_sim=1000
post_2D_M1=TRUE
post_graph=TRUE
CP_WFABC_diploid_modelchoice(N,t,t0,h_fixed,h_given,s_start,sampl
 e_times,N_sample,N_allele,min_freq,max_sims,no_sim,best_sim,set_
 seed=TRUE,post_graph,post_2D_M1)


# Example: CP_WFABC_haploid_estimator - estimates parameters of
 interest (s1, s2, CP) for haploid populations -> Influenza as an
 example
source("CP_WFABC_haploid_estimator.R")
N=176
t=105
t0=1
s_start=1
sample_times=c(1, 14, 27, 40, 53, 66, 79, 92, 105)
N_sample=rep(1000,9)
N_a1=c(0, 0, 1, 0, 1, 3, 96, 362, 772, 974, 990, 998, 995)
N_allele=rbind.data.frame(N_a1)
min_freq=0.02
max_sims=1
no_sim=10000
best_sim=100
post_2D_M1=TRUE
post_graph=TRUE
CP_WFABC_haploid_estimator(N,t,t0,s_start,sample_times,N_sample,N
 _allele,min_freq,max_sims,no_sim,best_sim,set_seed=TRUE,post_gra
 ph,post_2D_M1)


# Example: CP_WFABC_haploid_modelchoice - detects changing
 selection trajectories using model choice and estimates
 parameters of interest (s1, s2, CP) for haploid populations ->
 Influenza as an example with h fixed
source("CP_WFABC_haploid_modelchoice.R")
```

```
N=176
t=105
t0=1
s_start=1
sample_times=c(1, 14, 27, 40, 53, 66, 79, 92, 105)
N_sample=rep(1000,9)
a1=c(1, 1, 5, 72, 380, 938, 947, 943, 923)
a2=c(0, 2, 16, 128, 520, 988, 999, 999, 999)
a3=c(0, 0, 0, 20, 199, 971, 988, 990, 983)
a4=c(45, 3, 96, 362, 772, 974, 990, 998, 995)
a5=c(0, 0, 46, 140, 136, 421, 771, 809, 842)
a6=c(0, 0, 0, 0, 0, 1, 134, 215, 517)
N_allele <- rbind.data.frame(a1,a2,a3,a4,a5,a6)
rownames(N_allele) <- c("a1","a2","a3","a4","a5","a6")
colnames(N_allele) <- c("1", "14", "27", "40", "53", "66", "79",
 "92", "105")
N_allele <- N_allele[order(N_allele[,1]),]
min_freq=0.02
max_sims=1
no_sim=10000
best_sim=100
post_2D_M1=FALSE
post_graph=FALSE
#de novo
CP_WFABC_haploid_modelchoice(N,t,t0,s_start,sample_times,N_sample
 ,N_allele[which(N_allele[,1]<(min_freq*N_sample[1])),],min_freq,
 max_sims,no_sim,best_sim,set_seed=TRUE,post_graph,post_2D_M1)
#standing
if(max(N_allele[,1]) > (min_freq*N_sample[1])){
  for (n_s in (min_freq*N_sample[1]):max(N_allele[,1])){
    if(length(N_allele[which(N_allele[,1]==n_s),1]) != 0){

 CP_WFABC_haploid_modelchoice(N,t,t0,s_start,sample_times,N_sampl
 e,N_allele[which(N_allele[,1]==n_s),],min_freq,max_sims,no_sim,b
 est_sim,set_seed=TRUE,post_graph,post_2D_M1)
}
```