

# Package ‘Hapi’

July 28, 2018

**Type** Package

**Title** Inference of Chromosome-Length Haplotypes Using Genomic Data of Single Gamete Cells

**Version** 0.0.3

**Author** Ruidong Li,  
Han Qu,  
Jinfeng Chen,  
Shibo Wang,  
Le Zhang,  
Julong Wei,  
Sergio Pietro Ferrante,  
Mikeal L. Roose,  
Zhenyu Jia

**Maintainer** Ruidong Li <rli012@ucr.edu>

**Description** Inference of chromosome-length haplotypes using a few haploid gametes of an individual. The gamete genotype data may be generated from various platforms including genotyping arrays and sequencing even with low-coverage. Hapi simply takes genotype data of known hetSNPs in single gamete cells as input and report the high-resolution haplotypes as well as confidence of each phased hetSNPs. The package also includes a module allowing downstream analyses and visualization of identified crossovers in the gametes.

**Depends** R (>= 3.4.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** false

**Imports** HMM, ggplot2

**Suggests** knitr, testthat

**VignetteBuilder** knitr

**biocViews** SNP, GenomicVariation, Genetics, HiddenMarkovModel,  
SingleCell, Sequencing, Microarray

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-07-28 15:10:07 UTC

## R topics documented:

Hapi-package . . . . .	2
base2num . . . . .	3
crossover . . . . .	3
gamete11 . . . . .	4
gmt . . . . .	4
hapiAssemble . . . . .	4
hapiAssembleEnd . . . . .	5
hapiAutoPhase . . . . .	6
hapiBlockMPR . . . . .	7
hapiCVCluster . . . . .	8
hapiCVDistance . . . . .	9
hapiCVMap . . . . .	10
hapiCVResolution . . . . .	11
hapiFilterError . . . . .	11
hapiFrameSelection . . . . .	12
hapiGameteView . . . . .	13
hapiIdentifyCV . . . . .	14
hapiImupte . . . . .	15
hapiPhase . . . . .	16
hg19 . . . . .	16
num2base . . . . .	17
<b>Index</b>	<b>18</b>

---

Hapi-package

*Hapi is a novel easy-to-use package that only requires 3 to 5 gametes to reconstruct accurate and high-resolution haplotypes of an individual. The gamete genotype data may be generated from various platforms including genotyping arrays and next generation sequencing even with low-coverage. Hapi simply takes genotype data of known hetSNPs in single gamete cells as input and report the high-resolution haplotypes as well as confidence level of each phased hetSNPs. The package also includes a module allowing downstream analyses and visualization of crossovers in the gametes.*

---

### Description

Hapi is a novel easy-to-use package that only requires 3 to 5 gametes to reconstruct accurate and high-resolution haplotypes of an individual. The gamete genotype data may be generated from various platforms including genotyping arrays and next generation sequencing even with low-coverage. Hapi simply takes genotype data of known hetSNPs in single gamete cells as input and report the

high-resolution haplotypes as well as confidence level of each phased hetSNPs. The package also includes a module allowing downstream analyses and visualization of crossovers in the gametes.

---

base2num                      *Convert genotype coded in A/T/C/G to 0/1*

---

### Description

Convert base (A/T/C/G) coded genotype to numeric (0/1) coded

### Usage

```
base2num(gmt, ref, alt)
```

### Arguments

gmt                      a dataframe of genotype data of gamete cells  
ref                      a character represents reference allele  
alt                      a character represents alternative allele

### Value

a dataframe containing converted genotype

### Author(s)

Ruidong Li

### Examples

```
ref <- sample(c('A', 'T'), 500, replace=TRUE)
alt <- sample(c('C', 'G'), 500, replace=TRUE)

gmt <- data.frame(chr=rep(1, 500), pos=seq_len(500),
  ref=ref, alt=alt, gmt1=ref, gmt2=alt, gmt3=ref,
  gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
  stringsAsFactors = FALSE)

gmtDa <- base2num(gmt=gmt[5:9], ref=ref, alt=alt)
```

---

crossover                      *Crossover information across all gamete cells*

---

### Description

Crossover information across all gamete cells

---

gamete11	<i>Haplotypes of a single gamete cell for visualization</i>
----------	---

---

**Description**

Haplotypes of a single gamete cell for visualization

---

gmt	<i>Raw genotyping data</i>
-----	----------------------------

---

**Description**

Raw genotyping data

---

hapiAssemble	<i>Consensus haplotype assembly</i>
--------------	-------------------------------------

---

**Description**

Assemble the consensus high-resolution haplotypes

**Usage**

```
hapiAssemble(gmt, draftHap, keepLowConsistency = TRUE,
              consistencyThresh = 0.85)
```

**Arguments**

gmt	a dataframe of genotype data of gamete cells
draftHap	a dataframe with draft haplotype information
keepLowConsistency	logical, if low-consistent gamete cells should be kept
consistencyThresh	a numeric value of the threshold determining low-consistent gamete cells compared with the draft haplotype. Default is 0.85

**Value**

a dataframe containing phased haplotypes

**Author(s)**

Ruidong Li

**Examples**

```

finalDraft <- rep(0,500)
names(finalDraft) <- seq_len(500)

ref <- rep(0,500)
alt <- rep(1,500)

gmtDa <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)

idx1 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx2 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx3 <- sort(sample(seq_len(500), 30, replace = FALSE))

gmtDa[idx1,1] <- NA
gmtDa[idx2,2] <- NA
gmtDa[idx3,3] <- NA

consensusHap <- hapiAssemble(draftHap = finalDraft, gmt = gmtDa)

```

---

hapiAssembleEnd

*Assembly of haplotypes in regions at the end of a chromosome*


---

**Description**

Assembly of haplotypes in regions at the end of a chromosome

**Usage**

```
hapiAssembleEnd(gmt, draftHap, consensusHap, k = 300)
```

**Arguments**

gmt	a dataframe of genotype data of gamete cells
draftHap	a dataframe with draft haplotype information
consensusHap	a dataframe of the consensus haplotype information
k	a numeric value for the number of hetSNPs that will be combined with markers beyond the framework for assembly. Default is 300

**Value**

a dataframe containing phased haplotypes

**Author(s)**

Ruidong Li

**Examples**

```

finalDraft <- rep(0,500)
names(finalDraft) <- seq_len(500)

ref <- rep(0,500)
alt <- rep(1,500)

gmtDa <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)

idx1 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx2 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx3 <- sort(sample(seq_len(500), 30, replace = FALSE))

gmtDa[idx1,1] <- NA
gmtDa[idx2,2] <- NA
gmtDa[idx3,3] <- NA

consensusHap <- data.frame(hap1=rep(0,500),hap2=rep(1,500),
total=rep(5,500),rate=rep(1,500),
confidence=rep('F',500),
stringsAsFactors = FALSE)
rownames(consensusHap) <- seq_len(500)

consensusHap <- hapiAssembleEnd(gmt = gmtDa, draftHap = finalDraft,
consensusHap = consensusHap, k = 300)

```

---

hapiAutoPhase

*Automatic inference of haplotypes*


---

**Description**

Automatic inference of haplotypes

**Usage**

```
hapiAutoPhase(gmt, code = "atcg")
```

**Arguments**

gmt	a dataframe of genotype data of gamete cells
code	a character indicating the code style of genotype data. One of 'atcg' and '01'. Default is 'atcg'

**Value**

a dataframe of inferred consensus haplotypes

**Author(s)**

Ruidong Li

**Examples**

```
ref <- sample(c('A','T'),500, replace=TRUE)
alt <- sample(c('C','G'),500, replace=TRUE)

gmt <- data.frame(chr=rep(1,500), pos=seq_len(500),
  ref=ref, alt=alt, gmt1=ref, gmt2=alt, gmt3=ref,
  gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
  stringsAsFactors = FALSE)

hapOutput <- hapiAutoPhase(gmt=gmt, code='atcg')
```

---

hapiBlockMPR	<i>Maximum Parsimony of Recombination (MPR) for proofreading of draft haplotypes</i>
--------------	--

---

**Description**

Maximum Parsimony of Recombination (MPR) for proofreading of draft haplotypes

**Usage**

```
hapiBlockMPR(draftHap, gmtFrame, cvlink = 2, smallBlock = 100)
```

**Arguments**

draftHap	a dataframe with draft haplotype information
gmtFrame	a dataframe of raw genotype data in the framework
cvlink	a numeric value of number of cvlinks. Default is 2
smallBlock	a numeric value determining the size of small blocks that should be excluded from the draft haplotypes

**Value**

a dataframe of draft haplotypes after proofreading

**Author(s)**

Ruidong Li

**Examples**

```

ref <- rep(0,500)
alt <- rep(1,500)

gmtFrame <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)

idx1 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx2 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx3 <- sort(sample(seq_len(500), 30, replace = FALSE))

gmtFrame[idx1,1] <- NA
gmtFrame[idx2,2] <- NA
gmtFrame[idx3,3] <- NA

imputedFrame <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)

draftHap <- hapiPhase(imputedFrame)

finalDraft <- hapiBlockMPR(draftHap, gmtFrame, cvlink=2, smallBlock=100)

```

---

hapiCVCluster

*Filter out hetSNPs in potential complex regions*


---

**Description**

Filter out hetSNPs in potential complex regions

**Usage**

```
hapiCVCluster(draftHap, minDistance = 1e+06, cvlink = 2)
```

**Arguments**

draftHap	a dataframe with draft haplotype information
minDistance	a numeric value of the distance between two genomic positions with cv-links. Default is 1000000
cvlink	a numeric value of number of cvlinks. Default is 2

**Value**

a dataframe of regions to be filtered out

**Author(s)**

Ruidong Li



### Examples

```
ref <- rep(0,500)
alt <- rep(1,500)

imputedFrame <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)

draftHap <- hapiPhase(imputedFrame)
cvCluster <- hapiCVCluster(draftHap = draftHap, cvlink=2)
```

---

hapiCVDistance      *Histogram of crossover distance*

---

### Description

Histogram of crossover distance

### Usage

```
hapiCVDistance(cv)
```

### Arguments

cv                    a dataframe of crossover information

### Value

a histogram

### Author(s)

Ruidong Li

### Examples

```
data(crossover)
hapiCVDistance(cv=crossover)
```

---

hapiCVMap                      *Visualization of crossover map*

---

**Description**

Visualization of crossover map

**Usage**

```
hapiCVMap(cv, chr = hg19, step = 5, gap = gap.hg19, x.limits = 6,  
          y.breaks = NULL, y.labels = NULL)
```

**Arguments**

cv	a dataframe of crossover information
chr	a dataframe of chromosome information, including length, and centrometric regions
step	a numeric value of genomic interval in Mb. Default is 5
gap	a dataframe of unassembled regions with the first column is chromosome, the second column is start position, and third column is the end position of the gap. Default is gap for hg19. If no gap region is provided, use gap=NULL
x.limits	a numeric value of limits on x axis
y.breaks	a vector of positions to show labels on y axis. Default is NULL
y.labels	a vector of labels on the y axis. Default is NULL

**Value**

a plot of crossover map on all the chromosomes

**Author(s)**

Ruidong Li

**Examples**

```
data(crossover)  
hapiCVMap(cv=crossover)
```

---

hapiCVResolution      *Histogram of crossover resolution*

---

**Description**

Histogram of crossover resolution

**Usage**

```
hapiCVResolution(cv)
```

**Arguments**

cv                      a dataframe of crossover information

**Value**

a histogram

**Author(s)**

Ruidong Li

**Examples**

```
data(crossover)
hapiCVResolution(cv=crossover)
```

---

hapiFilterError      *Filter out hetSNPs with potential genotyping errors*

---

**Description**

Filter out hetSNPs with potential genotyping errors

**Usage**

```
hapiFilterError(gmt, hmm = NULL)
```

**Arguments**

gmt                      a dataframe of genotype data of gamete cells  
hmm                      a list containing probabilities of a HMM. Default is NULL

**Value**

a dataframe of genotype data of gamete cells

**Author(s)**

Ruidong Li

**Examples**

```
ref <- rep(0,500)
alt <- rep(1,500)

gmt <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
  gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
  stringsAsFactors = FALSE)

idx <- sort(sample(seq_len(500), 10, replace = FALSE))
gmt[idx,1] <- 1

gmtDa <- hapiFilterError(gmt = gmt)
```

---

hapiFrameSelection      *Selection of hetSNPs to form a framework*

---

**Description**

Selection of hetSNPs to form a framework

**Usage**

```
hapiFrameSelection(gmt, n = 3)
```

**Arguments**

gmt	a dataframe of genotype data of gamete cells
n	a numeric value of the minimum number of gametes with observed genotypes at a locus

**Value**

a dataframe of genotype data of gamete cells

**Author(s)**

Ruidong Li

**Examples**

```

ref <- rep(0,500)
alt <- rep(1,500)

gmt <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)

idx <- sort(sample(seq_len(500), 10, replace = FALSE))

gmt[idx,1] <- NA
gmt[idx,2] <- NA
gmt[idx,3] <- NA

gmtFrame <- hapiFrameSelection(gmt = gmt, n = 3)

```

---

hapiGameteView

*Visualization of haplotypes in a single gamete cell*


---

**Description**

Visualization of haplotypes in a single gamete cell

**Usage**

```

hapiGameteView(hap, chr = hg19, hap.color = c("deepskyblue2",
"darkorange2"), centromere.fill = "black", x.breaks = NULL,
x.labels = NULL, y.breaks = NULL, y.labels = NULL)

```

**Arguments**

hap	a dataframe of all the phased hetSNPs in all chromosomes
chr	a dataframe of chromosome information, including length, and centrometric regions
hap.color	a vector of colors for the two haplotypes. Default is c('deepskyblue2', 'darkorange2')
centromere.fill	a character of the color for the centromeres. Default is 'black'
x.breaks	a vector of positions to show labels on x axis. Default is NULL
x.labels	a vector of labels on the x axis. Default is NULL
y.breaks	a vector of positions to show labels on y axis. Default is NULL
y.labels	a vector of labels on the y axis. Default is NULL

**Value**

a plot of haplotypes in a single gamete cell

**Author(s)**

Ruidong Li

**Examples**

```
data(gamete11)
hapiGameteView(hap=gamete11)
```

---

`hapiIdentifyCV`*Identify crossovers in gamete cells*

---

**Description**

Identify crossovers in gamete cells

**Usage**

```
hapiIdentifyCV(hap, gmt, hmm = NULL)
```

**Arguments**

hap	a dataframe of the two haplotypes
gmt	a dataframe of genotype data of gamete cells
hmm	a list containing probabilities of a HMM. Default is NULL

**Value**

a dataframe containing crossover information in each gamete cell

**Author(s)**

Ruidong Li

**Examples**

```
ref <- sample(c('A','T'),500, replace=TRUE)
alt <- sample(c('C','G'),500, replace=TRUE)

hap <- data.frame(hap1=ref, hap2=alt, stringsAsFactors = FALSE)
rownames(hap) <- seq_len(500)

gmt <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
  gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
  stringsAsFactors = FALSE)

cvOutput <- hapiIdentifyCV(hap=hap, gmt=gmt)
```

---

`hapiImupte`*Imputation of missing genotypes in the framework*

---

**Description**

Imputation of missing genotypes in the framework

**Usage**

```
hapiImupte(gmt, nSPT = 2, allowNA = 0)
```

**Arguments**

<code>gmt</code>	a dataframe of genotype data of gamete cells in the framework
<code>nSPT</code>	a numeric value of the minimum number of supports for an imputation
<code>allowNA</code>	a numeric value of the maximum number of gametes with NA at a locus

**Value**

a dataframe of imputed genotypes in the framework

**Author(s)**

Ruidong Li

**Examples**

```
ref <- rep(0,500)
alt <- rep(1,500)

gmtFrame <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)

idx1 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx2 <- sort(sample(seq_len(500), 30, replace = FALSE))
idx3 <- sort(sample(seq_len(500), 30, replace = FALSE))

gmtFrame[idx1,1] <- NA
gmtFrame[idx2,2] <- NA
gmtFrame[idx3,3] <- NA
imputedFrame <- hapiImupte(gmtFrame, nSPT=2, allowNA=0)
```

---

hapiPhase	<i>Phase draft haplotypes by majority voting</i>
-----------	--

---

**Description**

Phase draft haplotypes by majority voting

**Usage**

```
hapiPhase(gmt)
```

**Arguments**

gmt                    a dataframe of imputed genotype data of gamete cells

**Value**

a dataframe of inferred draft haplotypes

**Author(s)**

Ruidong Li

**Examples**

```
ref <- rep(0,500)
alt <- rep(1,500)
imputedFrame <- data.frame(gmt1=ref, gmt2=alt, gmt3=ref,
gmt4=ref, gmt5=c(alt[1:250], ref[251:500]),
stringsAsFactors = FALSE)
draftHap <- hapiPhase(gmt=imputedFrame)
```

---

hg19	<i>Chromosome information of hg19</i>
------	---------------------------------------

---

**Description**

Chromosome information of hg19



---

num2base	<i>Convert genotype coded in 0/1 to A/T/C/G</i>
----------	---

---

**Description**

Convert numeric (0/1) coded genotype to base (A/T/C/G) coded

**Usage**

```
num2base(hap, ref, alt)
```

**Arguments**

hap	a dataframe of consensus haplotypes
ref	a character represents reference allele
alt	a character represents alternative allele

**Value**

a dataframe containing converted haplotypes

**Author(s)**

Ruidong Li

**Examples**

```
ref <- sample(c('A','T'),500, replace=TRUE)
alt <- sample(c('C','G'),500, replace=TRUE)

consensusHap <- data.frame(hap1=rep(0,500),hap2=rep(1,500),
  total=rep(5,500),rate=rep(1,500),
  confidence=rep('F',500),
  stringsAsFactors = FALSE)
rownames(consensusHap) <- seq_len(500)

hap <- num2base(hap=consensusHap, ref=ref, alt=alt)
```

# Index

## \*Topic **datasets**

crossover, [3](#)

gamete11, [4](#)

gmt, [4](#)

hg19, [16](#)

base2num, [3](#)

crossover, [3](#)

gamete11, [4](#)

gmt, [4](#)

Hapi (Hapi-package), [2](#)

Hapi-package, [2](#)

hapiAssemble, [4](#)

hapiAssembleEnd, [5](#)

hapiAutoPhase, [6](#)

hapiBlockMPR, [7](#)

hapiCVCluster, [8](#)

hapiCVDistance, [9](#)

hapiCVMap, [10](#)

hapiCVResolution, [11](#)

hapiFilterError, [11](#)

hapiFrameSelection, [12](#)

hapiGameteView, [13](#)

hapiIdentifyCV, [14](#)

hapiImupte, [15](#)

hapiPhase, [16](#)

hg19, [16](#)

num2base, [17](#)