Package: DEploid (via r-universe)

June 26, 2024

Type Package

Title Deconvolute Mixed Genomes with Unknown Proportions

Version 0.5.3

Description Traditional phasing programs are limited to diploid organisms. Our method modifies Li and Stephens algorithm with Markov chain Monte Carlo (MCMC) approaches, and builds a generic framework that allows haplotype searches in a multiple infection setting. This package is primarily developed as part of the Pf3k project, which is a global collaboration using the latest sequencing technologies to provide a high-resolution view of natural variation in the malaria parasite Plasmodium falciparum. Parasite DNA are extracted from patient blood sample, which often contains more than one parasite strain, with unknown proportions. This package is used for deconvoluting mixed haplotypes, and reporting the mixture proportions from each sample.

URL https://github.com/DEploid-dev/DEploid-r

BugReports https://github.com/DEploid-dev/DEploid-r/issues

License GPL (>= 3)

Depends R (>= 3.1.0)

Imports Rcpp (>= 0.11.2), scales (>= 0.4.0), plotly (>= 4.7.1), magrittr (>= 1.5), rmarkdown(>= 1.6), htmlwidgets (>= 1.0)

Suggests knitr, testthat (>= 0.9.0)

SystemRequirements C++11

VignetteBuilder knitr

LinkingTo Rcpp

RoxygenNote 7.1.0

Date 2020-04-21

Repository https://plasmogenepi.r-universe.dev

RemoteUrl https://github.com/DEploid-dev/DEploid-r

RemoteRef HEAD

RemoteSha 9f0b8e38d73cc6ef717fabbed6b2ea52f7b8e8e9

Contents

`DEploid-package` *Deconvolute Mixed Genomes with Unknown Proportions*

Description

Traditional phasing programs are limited to diploid organisms. Our method modifies Li and Stephens algorithm with Markov chain Monte Carlo (MCMC) approaches, and builds a generic framework that allows haplotype searches in a multiple infection setting. This package is primarily developed as part of #' the Pf3k project, which is a global collaboration using the latest sequencing technologies to provide a high-resolution view of natural variation in the malaria parasite Plasmodium falciparum. Parasite DNA are extracted from patient blood sample, which often contains more than one parasite strain, with unknown proportions. This package is used for deconvoluting mixed haplotypes, #' and reporting the mixture proportions from each sample.

Author(s)

Zhu Sha

Maintainer: Joe Zhu <sha.joe.zhu@gmail.com>

computeObsWSAF *Compute observed WSAF*

Description

Compute observed allele frequency within sample from the allele counts.

Usage

```
computeObsWSAF(alt, ref)
```

Arguments

alt	Numeric array of alternative allele count.
ref	Numeric array of reference allele count.

Value

Numeric array of observed allele frequency within sample.

See Also

histWSAF for histogram.

Examples

```
# Example 1
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390CoverageT = extractCoverageFromTxt(refFile, altFile)
obsWSAF = computeObsWSAF(PG0390CoverageT$altCount, PG0390CoverageT$refCount)

# Example 2
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
obsWSAF = computeObsWSAF(PG0390CoverageV$altCount, PG0390CoverageV$refCount)
```

dEploid *Deconvolute mixed haplotypes*

Description

Deconvolute mixed haplotypes, and reporting the mixture proportions from each sample This function provieds an interface for calling *dEploid* from R. The command line options are passed via the `args` argument

Usage

```
dEploid(args)
```

Arguments

`args`	String of dEploid input.

Value

A list with members of haplotypes, proportions and log likelihood of the MCMC chain.

- `Haps` Haplotypes at the final iteration in plain text file.
- `Proportions` MCMC updates of the proportion estimates.
- `llks` Log likelihood of the MCMC chain.

Seeding

The R version of DEploid uses random number from R's random generator. Therefore, the '-seed' argument of the command line version will be ignored, and no seed is given in the output. Use the R function 'set.seed' prior to calling this function to ensure reproduciblity of results.

See Also

- `vignette('dEploid-Arguments')` for an overview of commandline arguments

Examples

```
## Not run:
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
plafFile = system.file("extdata", "labStrains.test.PLAF.txt", package = "DEploid")
set.seed(1234)
PG0390.deconv = dEploid(paste("-vcf", vcfFile, "-plaf", plafFile, "-noPanel"))

## End(Not run)
```

extractCoverageFromTxt

Extract read counts from plain text file

Description

Extract read counts from tab-delimited text files of a single sample.

Usage

```
extractCoverageFromTxt(refFileName, altFileName)
```

Arguments

refFileName	Path of the reference allele count file.
altFileName	Path of the alternative allele count file.

Value

A data.frame contains four columns: chromosomes, positions, reference allele count, alternative allele count.

Note

The allele count files must be tab-delimited. The allele count files contain three columns: chromosomes, positions and allele count.

Examples

```
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390 = extractCoverageFromTxt(refFile, altFile)
```

extractCoverageFromVcf

Extract read counts from VCF

Description

Extract read counts from VCF file of a single sample.

Usage

```
extractCoverageFromVcf(vcfFileName, ADFieldIndex = 2)
```

Arguments

vcfFileName	Path of the VCF file.
ADFieldIndex	Index of the AD field of the sample field. For example, if the format is "GT:AD:DP:GQ:PL", the AD index is 2 (by default).

Value

A data.frame contains four columns: chromosomes, positions, reference allele count, alternative allele count.

Note

The VCF file should only contain one sample. If more samples present in the VCF, it only returns coverage for of the first sample.

Examples

```
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390 = extractCoverageFromVcf(vcfFile)
```

---

extractPLAF *Extract PLAF*

Description

Extract population level allele frequency (PLAF) from text file.

Usage

```
extractPLAF(plafFileName)
```

Arguments

plafFileName	Path of the PLAF text file.

Value

A numeric array of PLAF

Note

The text file must have header, and population level allele frequency recorded in the "PLAF" field.

Examples

```
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
  package = "DEploid")
plaf = extractPLAF(plafFile)
```

`extractVcf` *Extract VCF information*

Description

Extract VCF information

Usage

```
extractVcf(filename)
```

Arguments

`filename`	VCF file name.

Value

A dataframe list with members of haplotypes, proportions and log likelihood of the MCMC chain.

- `CHROM` SNP chromosomes.
- `POS` SNP positions.
- `refCount` reference allele count.
- `altCount` alternative allele count.

See Also

- `extractCoverageFromVcf`
- `extractCoverageFromTxt`

Examples

```
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
vcf = extractVcf(vcfFile)
```

haplotypePainter *Painting haplotype according the reference panel*

Description

Plot the posterior probabilities of a haplotype given the refernece panel.

Usage

```
haplotypePainter(
  posteriorProbabilities,
  title = "",
  labelScaling,
  numberOfInbreeding = 0
)
```

Arguments

posteriorProbabilities	Posterior probabilities matrix with the size of number of loci by the number of reference strain.
title	Figure title.
labelScaling	Scaling parameter for plotting.
numberOfInbreeding	Number of inbreeding strains copying from.

histWSAF *WSAF histogram*

Description

Produce histogram of the allele frequency within sample.

Usage

```
histWSAF(
  obsWSAF,
  exclusive = TRUE,
  title = "Histogram 0<WSAF<1",
  cex.lab = 1,
  cex.main = 1,
  cex.axis = 1
)
```

Arguments

obsWSAF	Observed allele frequency within sample
exclusive	When TRUE 0 < WSAF < 1; otherwise 0 <= WSAF <= 1.
title	Histogram title
cex.lab	Label size.
cex.main	Title size.
cex.axis	Axis text size.

Value

histogram

Examples

```
# Example 1
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390Coverage = extractCoverageFromTxt(refFile, altFile)
obsWSAF = computeObsWSAF(PG0390Coverage$altCount, PG0390Coverage$refCount)
histWSAF(obsWSAF)
myhist = histWSAF(obsWSAF, FALSE)

# Example 2
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
obsWSAF = computeObsWSAF(PG0390CoverageV$altCount, PG0390CoverageV$refCount)
histWSAF(obsWSAF)
myhist = histWSAF(obsWSAF, FALSE)
```

---

plotAltVsRef *Plot coverage*

Description

Plot alternative allele count vs reference allele count at each site.

Usage

```
plotAltVsRef(
  ref,
  alt,
  title = "Alt vs Ref",
  exclude.ref = c(),
  exclude.alt = c(),
  potentialOutliers = c(),
```

```
  cex.lab = 1,
  cex.main = 1,
  cex.axis = 1
)
```

Arguments

ref	Numeric array of reference allele count.
alt	Numeric array of alternative allele count.
title	Figure title, "Alt vs Ref" by default
exclude.ref	Numeric array of reference allele count at sites that are not deconvoluted.
exclude.alt	Numeric array of alternative allele count at sites that are not deconvoluted
potentialOutliers	Index of potential outliers.
cex.lab	Label size.
cex.main	Title size.
cex.axis	Axis text size.

Examples

```
# Example 1
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390CoverageT = extractCoverageFromTxt(refFile, altFile)
plotAltVsRef(PG0390CoverageT$refCount, PG0390CoverageT$altCount)

# Example 2
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
plotAltVsRef(PG0390CoverageV$refCount, PG0390CoverageV$altCount)
```

---

plotAltVsRefPlotly *Plot coverage*

Description

Plot alternative allele count vs reference allele count at each site.

Usage

```
plotAltVsRefPlotly(ref, alt, title = "Alt vs Ref", potentialOutliers = c())
```

Arguments

ref	Numeric array of reference allele count.
alt	Numeric array of alternative allele count.
title	Figure title, "Alt vs Ref" by default
potentialOutliers	Index of potential outliers.

Examples

```
# Example 1
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390CoverageT = extractCoverageFromTxt(refFile, altFile)
plotAltVsRefPlotly(PG0390CoverageT$refCount, PG0390CoverageT$altCount)

# Example 2
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
plotAltVsRefPlotly(PG0390CoverageV$refCount, PG0390CoverageV$altCount)
```

---

plotHistWSAFPlotly *WSAF histogram*

Description

Produce histogram of the allele frequency within sample.

Usage

```
plotHistWSAFPlotly(obsWSAF, exclusive = TRUE, title = "Histogram 0<WSAF<1")
```

Arguments

obsWSAF	Observed allele frequency within sample
exclusive	When TRUE 0 < WSAF < 1; otherwise 0 <= WSAF <= 1.
title	Figure title, "Histogram 0<WSAF<1" by default

Value

histogram

Examples

```
# Example 1
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390Coverage = extractCoverageFromTxt(refFile, altFile)
obsWSAF = computeObsWSAF(PG0390Coverage$altCount, PG0390Coverage$refCount)
plotHistWSAFPlotly(obsWSAF)
myhist = plotHistWSAFPlotly(obsWSAF)

# Example 2
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
obsWSAF = computeObsWSAF(PG0390CoverageV$altCount, PG0390CoverageV$refCount)
plotHistWSAFPlotly(obsWSAF)
myhist = plotHistWSAFPlotly(obsWSAF)
```

`plotObsExpWSAF` *Plot WSAF*

Description

Plot observed alternative allele frequency within sample against expected WSAF.

Usage

```
plotObsExpWSAF(
  obsWSAF,
  expWSAF,
  title = "WSAF(observed vs expected)",
  cex.lab = 1,
  cex.main = 1,
  cex.axis = 1
)
```

Arguments

`obsWSAF`	Numeric array of observed WSAF.
`expWSAF`	Numeric array of expected WSAF.
`title`	Figure title.
`cex.lab`	Label size.
`cex.main`	Title size.
`cex.axis`	Axis text size.

Examples

```
## Not run:
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
obsWSAF = computeObsWSAF(PG0390CoverageV$altCount, PG0390CoverageV$refCount)
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
 package = "DEploid")
PG0390.deconv = dEploid(paste("-vcf", vcfFile,
                                      "-plaf", plafFile, "-noPanel"))
prop = PG0390.deconv$Proportions[dim(PG0390.deconv$Proportions)[1],]
expWSAF = t(PG0390.deconv$Haps) %*% prop
plotObsExpWSAF(obsWSAF, expWSAF)

## End(Not run)
```

---

plotObsExpWSAFPlotly *Plot WSAF*

Description

Plot observed alternative allele frequency within sample against expected WSAF.

Usage

```
plotObsExpWSAFPlotly(obsWSAF, expWSAF, title = "WSAF(observed vs expected)")
```

Arguments

obsWSAF	Numeric array of observed WSAF.
expWSAF	Numeric array of expected WSAF.
title	Figure title, "WSAF(observed vs expected)" by default

Examples

```
## Not run:
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
obsWSAF = computeObsWSAF(PG0390CoverageV$altCount, PG0390CoverageV$refCount)
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
 package = "DEploid")
PG0390CoverageV.deconv = dEploid(paste("-vcf", vcfFile,
                                       "-plaf", plafFile, "-noPanel"))

prop = PG0390CoverageV.deconv$Proportions[dim(PG0390CoverageV.deconv
                                              $Proportions)[1],]

expWSAF = t(PG0390CoverageV.deconv$Haps) %*% prop
```

```
plotObsExpWSAFPlotly(obsWSAF, expWSAF)

## End(Not run)
```

`plotProportions` *Plot proportions*

Description

Plot the MCMC samples of the proportion, indexed by the MCMC chain.

Usage

```
plotProportions(
  proportions,
  title = "Components",
  cex.lab = 1,
  cex.main = 1,
  cex.axis = 1
)
```

Arguments

`proportions`	Matrix of the MCMC proportion samples. The matrix size is number of the MCMC samples by the number of strains.
`title`	Figure title.
`cex.lab`	Label size.
`cex.main`	Title size.
`cex.axis`	Axis text size.

Examples

```
## Not run:
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
    package = "DEploid")
panelFile = system.file("extdata", "labStrains.test.panel.txt",
    package = "DEploid")
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390CoverageT = extractCoverageFromTxt(refFile, altFile)
PG0390Coverage.deconv = dEploid(paste("-ref", refFile, "-alt", altFile,
    "-plaf", plafFile, "-noPanel"))
plotProportions(PG0390Coverage.deconv$Proportions, "PG0390-C proportions")

## End(Not run)
```

plotWSAFvsPLAF *Plot WSAF vs PLAF*

Description

Plot allele frequencies within sample against population level.

Usage

```
plotWSAFvsPLAF(
  plaf,
  obsWSAF,
  expWSAF = c(),
  potentialOutliers = c(),
  title = "WSAF vs PLAF",
  cex.lab = 1,
  cex.main = 1,
  cex.axis = 1
)
```

Arguments

plaf	Numeric array of population level allele frequency.
obsWSAF	Numeric array of observed altenative allele frequencies within sample.
expWSAF	Numeric array of expected WSAF from model.
potentialOutliers	Index of potential outliers.
title	Figure title, "WSAF vs PLAF" by default
cex.lab	Label size.
cex.main	Title size.
cex.axis	Axis text size.

Examples

```
# Example 1
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390CoverageT = extractCoverageFromTxt(refFile, altFile)
obsWSAF = computeObsWSAF(PG0390CoverageT$altCount, PG0390CoverageT$refCount)
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
  package = "DEploid")
plaf = extractPLAF(plafFile)
plotWSAFvsPLAF(plaf, obsWSAF)

# Example 2
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
```

```
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
obsWSAF = computeObsWSAF(PG0390CoverageV$altCount, PG0390CoverageV$refCount)
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
  package = "DEploid")
plaf = extractPLAF(plafFile)
plotWSAFvsPLAF(plaf, obsWSAF)
```

`plotWSAFVsPLAFPlotly` *Plot WSAF vs PLAF*

Description

Plot allele frequencies within sample against population level.

Usage

```
plotWSAFVsPLAFPlotly(
  plaf,
  obsWSAF,
  ref,
  alt,
  title = "WSAF vs PLAF",
  potentialOutliers = c()
)
```

Arguments

`plaf`	Numeric array of population level allele frequency.
`obsWSAF`	Numeric array of observed altenative allele frequencies within sample.
`ref`	Numeric array of reference allele count.
`alt`	Numeric array of alternative allele count.
`title`	Figure title, "WSAF vs PLAF" by default
`potentialOutliers`	Index of potential outliers.

Examples

```
# Example 1
refFile = system.file("extdata", "PG0390-C.test.ref", package = "DEploid")
altFile = system.file("extdata", "PG0390-C.test.alt", package = "DEploid")
PG0390CoverageT = extractCoverageFromTxt(refFile, altFile)
obsWSAF = computeObsWSAF(PG0390CoverageT$altCount, PG0390CoverageT$refCount)
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
  package = "DEploid")
plaf = extractPLAF(plafFile)
plotWSAFVsPLAFPlotly(plaf, obsWSAF, PG0390CoverageT$refCount,
```

```
                PG0390CoverageT$altCount)

# Example 2
vcfFile = system.file("extdata", "PG0390-C.test.vcf.gz", package = "DEploid")
PG0390CoverageV = extractCoverageFromVcf(vcfFile)
obsWSAF = computeObsWSAF(PG0390CoverageV$altCount, PG0390CoverageV$refCount)
plafFile = system.file("extdata", "labStrains.test.PLAF.txt",
  package = "DEploid")
plaf = extractPLAF(plafFile)
plotWSAFVsPLAFPlotly(plaf, obsWSAF, PG0390CoverageV$refCount,
                PG0390CoverageV$altCount)
```

Index