

# Package ‘tLOH’

December 6, 2024

**Version** 1.14.0

**Type** Package

**Date** 2021-05-5

**Title** Assessment of evidence for LOH in spatial transcriptomics pre-processed data using Bayes factor calculations

**Description** tLOH, or transcriptomicsLOH, assesses evidence for loss of heterozygosity (LOH) in pre-processed spatial transcriptomics data. This tool requires spatial transcriptomics cluster and allele count information at likely heterozygous single-nucleotide polymorphism (SNP) positions in VCF format. Bayes factors are calculated at each SNP to determine likelihood of potential loss of heterozygosity event. Two plotting functions are included to visualize allele fraction and aggregated Bayes factor per chromosome. Data generated with the 10X Genomics Visium Spatial Gene Expression platform must be pre-processed to obtain an individual sample VCF with columns for each cluster. Required fields are allele depth (AD) with counts for reference/alternative alleles and read depth (DP).

**License** MIT + file LICENSE

**URL** <https://github.com/USCDTG/tLOH>

**Encoding** UTF-8

**Suggests** knitr, rmarkdown

**Depends** R (>= 4.2)

**Imports** scales, stats, utils, ggplot2, data.table, purrr, dplyr, VariantAnnotation, GenomicRanges, MatrixGenerics, bestNormalize, depmixS4, naniar, stringr

**VignetteBuilder** knitr

**BugReports** <https://github.com/USCDTG/tLOH/issues>

**biocViews** CopyNumberVariation, Transcription, SNP, GeneExpression, Transcriptomics

**RoxygenNote** 7.2.1

**git\_url** <https://git.bioconductor.org/packages/tLOH>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** e205786

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-12-05

**Author** Michelle Webb [cre, aut],  
David Craig [aut]

**Maintainer** Michelle Webb <michelgw@usc.edu>

## Contents

aggregateCHRPlot . . . . .	2
alleleFrequencyPlot . . . . .	3
documentErrorRegions . . . . .	4
hiddenMarkovAnalysis . . . . .	4
humanGBMsampleAC . . . . .	5
initialStartProbabilities . . . . .	6
marginalLikelihoodM1 . . . . .	6
marginalLikelihoodM2 . . . . .	7
marginalM1Calc . . . . .	8
marginalM2CalcBHET . . . . .	8
marginalM2CalcBLOH . . . . .	9
modePeakCalc . . . . .	10
prepareHMMdataframes . . . . .	10
regionAnalysis . . . . .	11
regionFinalize . . . . .	12
removeOutlierFromCalc . . . . .	12
runHMM_1 . . . . .	13
runHMM_2 . . . . .	14
runHMM_3 . . . . .	14
splitByChromosome . . . . .	15
summarizeRegions1 . . . . .	16
summarizeRegions2 . . . . .	16
tLOHCalc . . . . .	17
tLOHCalcUpdate . . . . .	18
tLOHDataImport . . . . .	19

**Index** **20**

---

aggregateCHRPlot	<i>Visualization of data output from the tLOHCalc function, aggregated per chromosome</i>
------------------	---

---

## Description

Output is a plot of the sum of  $\text{Log}_{10}(1/K)$  values ( $K$  is a Bayes factor) per chromosome for each cluster. The dotted line at  $y=3$  represents threshold for substantial evidence toward Model 2

## Arguments

df	An input dataframe with merged cluster data output by tLOHCalc
sample	Name of sample for plot title

**Value**

Output is a plot where the y axis is sum of  $\text{Log}_{10}(1/K)$  values ( $K$  is a Bayes factor) per chromosome and the x axis is chromosome

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsampleAC')
df <- tLOHCalc(humanGBMsampleAC)
aggregateCHRPlot(df, "Example")
```

---

alleleFrequencyPlot     *Visualization of data output from the tLOHCalc function*

---

**Description**

Creates a plot with panels for each cluster. The x-axis is chromosome, y-axis is allele frequency. Point color is  $\text{Log}_{10}(1/K)$  where  $K$  is a Bayes factor

**Arguments**

df	An input dataframe with merged cluster data
sample	Name of sample for plot title

**Value**

Output is a plot of allele frequency for each cluster. Can be assigned to object and visualized individually. For each panel, the y axis has a min of 0 and max of 1

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsampleAC')
df <- tLOHCalc(humanGBMsampleAC)
alleleFrequencyPlot(df, "Example")
```

documentErrorRegions *Identification of Errors*

---

**Description**

Generates dataframes equal to the length of original data containing NA if errors were found during the run HMM process. Important for final overview of results

**Usage**

```
documentErrorRegions(a,b)
```

**Arguments**

a List of dataframes from prepareHMMdataframes  
b List of HMM state determination dataframes from the run\_HMM series of functions

**Value**

Output is a list of dataframes to be used by the regionAnalysis function

**Author(s)**

Michelle Webb

**Examples**

```
estimatedStates <- data.frame(state = c(1,1,1), S1 = c(1,1,1), S2 = c(1,1,1))  
sampleDataFrame <- data.frame(data = c(1,1,1))  
list1 <- list(sampleDataFrame, sampleDataFrame)  
list2 <- list(estimatedStates, estimatedStates)  
documentErrorRegions(list1, list2)
```

---

hiddenMarkovAnalysis *Run Multi-Step HMM Analysis on tLOHCalcUpdate output*

---

**Description**

Applies the depmixS4 method of HMM Analysis on tLOHCalcUpdate output to obtain segments, noted by the output 'state' column

**Usage**

```
hiddenMarkovAnalysis(df, initProbs, trProbs)
```

**Arguments**

<code>df</code>	A dataframe output by <code>tLOHCalcUpdate</code>
<code>initProbs</code>	Dataframe containing 22 rows and two columns, <code>initProb1</code> and <code>initProb2</code> . Each row represents a chromosome in sequential order, with <code>initProb1</code> being the probability of state1 and <code>initProb2</code> being the probability of state2.
<code>trProbs</code>	Matrix of transition start probabilities for HMM

**Value**

Output is a dataframe containing HMM analysis output and `tLOHCalcUpdate` output summary

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsampleAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsampleAC, 1.25, 1.25, 500, 500, 4)
trProbs <- cbind(c(0.8999, 0.1001), c(0.1001, 0.8999))
output <- hiddenMarkovAnalysis(df, initialStartProbabilities, trProbs)
```

---

<code>humanGBMsampleAC</code>	<i>Imported dataset of a human glioblastoma spatial transcriptomics sample processed with <code>tLOHImportData</code>.</i>
-------------------------------	--

---

**Description**

A dataset of a human glioblastoma sample containing the allele count (AC) information for 9 spatial transcriptomics clusters

**Usage**

```
data("humanGBMsampleAC")
```

**Format**

A data frame with 34601 rows and 7 variables:

**rsID** dbSNP rs identifier  
**CLUSTER** cluster number  
**TOTAL** total number of counts  
**REF** counts for the reference allele  
**ALT** counts for the alternative allele  
**CHR** chromosome number  
**POS** genomic position

**Source**

Craig Lab data repository

**Examples**

```
data("humanGBMsampleAC")
```

---

```
initialStartProbabilities
```

*Imported dataset of sample start probabilities for hiddenMarkovAnalysis*

---

**Description**

A dataset of initial start probabilities to use with the HMM analysis. Users may create their own dataset using the same format

**Usage**

```
data("initialStartProbabilities")
```

**Format**

A data frame with 22 rows and 2 variables:

**initProb1** Initial Probability 1

**initProb2** Initial Probability 2

**Source**

Craig Lab data repository

**Examples**

```
data("initialStartProbabilities")
```

---

```
marginalLikelihoodM1
```

*Marginal M1 Calculation*

---

**Description**

Calculation of the marginal likelihood of Model 1, LOH

**Arguments**

x	Dataframe output by tLOHDataImport
a	Alpha value
b	Beta value

**Details**

The reference and total counts should come from a .csv output by the spatial LOH pre-processing pipeline. The recommended values for both Alpha1 and Beta1 is 1.25.

**Value**

The value returned from marginalLikelihoodM1 is numeric

**Author(s)**

Michelle Webb

**Examples**

```
test <- data.frame(REF=c(10,2,3,4,5,10),TOTAL=c(20,20,20,20,20,20))
apply(test, MARGIN = 1, FUN = marginalLikelihoodM1, a = 1.25, b = 1.25)
```

---

marginalLikelihoodM2 *Marginal M2 Calculation*

---

**Description**

Calculation of the marginal likelihood of Model 2, HET

**Arguments**

x	Dataframe output by tLOHDataImport
a	Alpha value
b	Beta value

**Details**

The reference and total counts should come from a .csv output by the spatial LOH pre-processing pipeline. The recommended values for both Alpha2 and Beta2 is 500.

**Value**

The value returned from marginalLikelihoodM1 is numeric

**Author(s)**

Michelle Webb

**Examples**

```
test <- data.frame(REF=c(10,2,3,4,5,10),TOTAL=c(20,20,20,20,20,20))
apply(test, MARGIN = 1, FUN = marginalLikelihoodM1, a = 500, b = 500)
```

---

marginalM1Calc	<i>Calculate marginal of Model 1</i>
----------------	--------------------------------------

---

**Description**

This function takes the number of counts for a reference allele as x, and the number of total allele counts as y.

**Arguments**

x	Number of counts for the reference allele.
y	Number of counts total at this SNP position.

**Details**

The reference and total counts should come from a .csv output by the spatial LOH pre-processing pipeline.

**Value**

The value returned from marginalM1Calc is numeric

**Author(s)**

Michelle Webb

**Examples**

```
marginalM1Calc(10, 0.5)
```

---

marginalM2CalcBHET	<i>Calculation of marginal M2 het</i>
--------------------	---------------------------------------

---

**Description**

Calculation of marginal M2 het

**Usage**

```
marginalM2CalcBHET(x, a, b)
```

**Arguments**

x	Number of counts for the reference allele
a	Alpha value
b	Beta value

**Value**

The value returned from marginalM2CalcBHET is numeric



**Author(s)**

Michelle Webb

**Examples**

```
save <- data.frame(REF=c(10,2,3,4,5,10),TOTAL=c(20,20,20,20,20,20))
apply(save, MARGIN = 1, FUN = marginalM2CalcBHET, a = 10,b = 10)
```

---

marginalM2CalcBLOH	<i>Marginal M2 Calculation</i>
--------------------	--------------------------------

---

**Description**

Calculation of the marginal for Model 2

**Usage**

```
marginalM2CalcBLOH(x, a, b)
```

**Arguments**

- x Counts for the reference allele
- a Alpha value
- b Beta value

**Value**

The value returned from marginalM2CalcBLOH is numeric

**Author(s)**

Michelle Webb

**Examples**

```
test <- data.frame(REF=c(10,2,3,4,5,10),TOTAL=c(20,20,20,20,20,20))
apply(test, MARGIN = 1, FUN = marginalM2CalcBLOH, a = 10,b = 10)
```

modePeakCalc

*Calculation of mode peak*

---

**Description**

This function takes a set of numbers and outputs a mode peak value. To be used in a larger function that will be updated.

**Arguments**

x                      List of allele fraction values

**Details**

List of values should be the allele fractions of SNPs with the top 25 percent of counts in a region. If only one value is input, that value is returned.

**Value**

The value returned is numeric

**Author(s)**

Michelle Webb

**Examples**

```
test <- c(1,2,3,4,5)
modePeakCalc(test)
```

---

prepareHMMdataframes*Prepare dataframes for HMM analysis*

---

**Description**

Split output from tLOHCalc or tLOHCalcUpdate into a list of cluster and chromosome separated dataframes. Applies an ordered quantile normalization on the bayes factor K values in each dataset.

**Usage**

```
prepareHMMdataframes(importedData)
```

**Arguments**

importedData      Input dataframe generated from the tLOHCalc or tLOHCalcUpdate function

**Value**

Output is a list of dataframes separated by chromosome and cluster

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsamplAC')
df <- tLOHCalcUpdate(humanGBMsamplAC,1.25,1.25,500,500,4)
output <- prepareHMMdataframes(df)
```

---

regionAnalysis	<i>Summary of HMM Regions</i>
----------------	-------------------------------

---

**Description**

Generates summary metrics for HMM regions

**Usage**

```
regionAnalysis(originalDF,dataframeList)
```

**Arguments**

originalDF	Original imported dataframe from tLOHCalcUpdated
dataframeList	List of HMM state determination dataframes from the run_HMM series of functions

**Value**

Output is a dataframe containing region metrics and data for each HMM segment

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsamplAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsamplAC,1.25,1.25,500,500,4)
dataframeList <- prepareHMMdataframes(df)
trProbs <- cbind(c(0.8999,0.1001),c(0.1001,0.8999))
dataframeList2 <- runHMM_1(dataframeList, initialStartProbabilities, trProbs)
dataframeList3 <- runHMM_2(dataframeList2)
output <- runHMM_3(dataframeList3)
final <- regionAnalysis(dataframeList,output)
```

---

regionFinalize	<i>Summary of HMM Regions</i>
----------------	-------------------------------

---

**Description**

Final metrics and summary for regions

**Usage**

```
regionFinalize(finalList1)
```

**Arguments**

finalList1      List of dataframes output by the regionAnalysis function

**Value**

Output is a table containing all calculations from the bayes factor and HMM analysis

**Author(s)**

Michelle Webb

**Examples**

```
## Not run:
data('humanGBMsamplAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsamplAC,1.25,1.25,500,500,4)
dataframeList <- prepareHMMdataframes(df)
trProbs <- cbind(c(0.8999,0.1001),c(0.1001,0.8999))
dataframeList2 <- runHMM_1(dataframeList, initialStartProbabilities, trProbs)
dataframeList3 <- runHMM_2(dataframeList2)
output <- runHMM_3(dataframeList3)
intermediate <- regionAnalysis(dataframeList,output)
final <- regionFinalize(intermediate)

## End(Not run)
```

---

removeOutlierFromCalc	<i>Removes outliers</i>
-----------------------	-------------------------

---

**Description**

Take rows with a total count greater than 2000 and sets to NA

**Arguments**

dataframe	input dataframe
cols	which column
rows	which row
newValue	what to replace

**Value**

Dataframe returned

**Author(s)**

Michelle Webb

**Examples**

```
test <- data.frame(TOTAL=c(2000,20,20,20,20,20))
removeOutlierFromCalc(test,"TOTAL",test[test$TOTAL > 2000,],NA)
```

---

runHMM\_1

*Step 1 of HMM process*

---

**Description**

Applies the depmixS4 method depmix on normalized K values.

**Usage**

```
runHMM_1(dataframeList, initProbs, trProbs)
```

**Arguments**

`dataframeList` List of dataframes separated by cluster and chromosome from the prepareHMMdataframes function.

`initProbs` Dataframe containing 22 rows and two columns, `initProb1` and `initProb2`. Each row represents a chromosome in sequential order, with `initProb1` being the probability of state1 and `initProb2` being the probability of state2.

`trProbs` Matrix of transition start probabilities for HMM

**Value**

Output is a list of depmixS4 depmix class objects for each input dataframe

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsamplAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsamplAC,1.25,1.25,500,500,4)
dataframeList <- prepareHMMdataframes(df)
trProbs <- cbind(c(0.8999,0.1001),c(0.1001,0.8999))
output <- runHMM_1(dataframeList, initialStartProbabilities, trProbs)
```

---

runHMM\_2                      *Step 2 of HMM process*

---

**Description**

Applies the depmixS4 method fit on .

**Usage**

```
runHMM_2(dataframeList)
```

**Arguments**

dataframeList    List of depmixS4 depmix class objects generated from the runHMM\_1 step

**Value**

Output is a list of depmixS4 depmix.fitted class output for each input dataframe

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsamplAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsamplAC,1.25,1.25,500,500,4)
dataframeList <- prepareHMMdataframes(df)
trProbs <- cbind(c(0.8999,0.1001),c(0.1001,0.8999))
dataframeList2 <- runHMM_1(dataframeList, initialStartProbabilities, trProbs)
output <- runHMM_2(dataframeList2)
```

---

runHMM\_3                      *Step 3 of HMM process*

---

**Description**

Applies the depmixS4 method posterior on normalized K values.

**Usage**

```
runHMM_3(dataframeList)
```

**Arguments**

dataframeList    List of depmixS4 depmix.fitted class output generated from the runHMM\_2 step

**Value**

Output is a list of depmixS4 posterior state classifications for each input dataframe

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsampleAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsampleAC,1.25,1.25,500,500,4)
dataframeList <- prepareHMMdataframes(df)
trProbs <- cbind(c(0.8999,0.1001),c(0.1001,0.8999))
dataframeList2 <- runHMM_1(dataframeList, initialStartProbabilities, trProbs)
dataframeList3 <- runHMM_2(dataframeList2)
output <- runHMM_3(dataframeList3)
```

---

splitByChromosome      *Split dataframe into individual chromosome dataframes*

---

**Description**

Creates individual chromosome dataframes

**Usage**

```
splitByChromosome(listOfDataframes,numberOfDataframes)
```

**Arguments**

listOfDataframes  
Input dataframe generated from the tLOHDataImport function

numberOfDataframes  
Number of dataframes in list

**Value**

Output is a list of dataframe separated by chromosome

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsampleAC')
df <- tLOHCalcUpdate(humanGBMsampleAC,1.25,1.25,500,500,4)
output <- splitByChromosome(list(df),1)
```

---

summarizeRegions1      *Step 1 of region summary*

---

**Description**

Function used by regionFinalize to group segments

**Usage**

```
summarizeRegions1(x)
```

**Arguments**

x                      A list of dataframes containing calculated values and JMM state determinations

**Value**

Output is a list of dataframes

**Author(s)**

Michelle Webb

**Examples**

```
## Not run: data('humanGBMsamplAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsamplAC,1.25,1.25,500,500,4)
dataframeList <- prepareHMMdataframes(df)
trProbs <- cbind(c(0.8999,0.1001),c(0.1001,0.8999))
dataframeList2 <- runHMM_1(dataframeList, initialStartProbabilities, trProbs)
dataframeList3 <- runHMM_2(dataframeList2)
output <- runHMM_3(dataframeList3)
intermediate <- regionAnalysis(dataframeList,output)
finalList1 <- purrr::map(intermediate,
~dplyr::mutate(.x, state = as.character(state)))
sampleValues <- as.data.frame(purrr::reduce(finalList1,full_join))
sampleData <- summarizeRegions1(finalList1)
## End(Not run)
```

---

summarizeRegions2      *Step 2 of region summary*

---

**Description**

Function used by regionFinalize to identify segment start and end positions

**Usage**

```
summarizeRegions2(finalTable)
```



**Arguments**

`finalTable` A dataframe containing metrics from the Bayes Factor and HMM analysis

**Value**

Output is a dataframe

**Author(s)**

Michelle Webb

**Examples**

```
## Not run:
data('humanGBMsamplesAC')
data('initialStartProbabilities')
df <- tLOHCalcUpdate(humanGBMsamplesAC,1.25,1.25,500,500,4)
dataframeList <- prepareHMMdataframes(df)
trProbs <- cbind(c(0.8999,0.1001),c(0.1001,0.8999))
dataframeList2 <- runHMM_1(dataframeList, initialStartProbabilities, trProbs)
dataframeList3 <- runHMM_2(dataframeList2)
output <- runHMM_3(dataframeList3)
intermediate <- regionAnalysis(dataframeList,output)
finalList1 <- purrr::map(intermediate,
~dplyr::mutate(.x, state = as.character(state)))
sampleValues <- as.data.frame(purrr::reduce(finalList1,full_join))
sampleData <- summarizeRegions1(finalList1)
sampleData$lengthOfInterval <- sampleData$intervalEnd - sampleData$intervalStart
sampleDF <- summarizeRegions2(sampleValues)
## End(Not run)
```

---

tLOHCalc	<i>Assessment of evidence for LOH in clusters from spatial transcriptomics data</i>
----------	---

---

**Description**

Calculates Bayes factors for allele fractions at each SNP position. Uses dataframe output by tLOHDataImport

**Usage**

```
tLOHCalc(forCalcDF)
```

**Arguments**

`forCalcDF` Input dataframe generated from the tLOHDataImport function

**Value**

Output is a dataframe with values that can be visualized with `alleleFrequencyPlot()` or `aggregateCHRPlot()`

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsampleAC')
df <- tLOHCalc(humanGBMsampleAC)
head(df)
```

---

tLOHCalcUpdate

*Assesment of evidence for LOH in clusters from spatial transcriptomics allele count data*


---

**Description**

Calculates Bayes factors for allele fractions at each SNP position. Uses dataframe output by tLOHDataImport.

**Usage**

```
tLOHCalcUpdate(forCalcDF, alpha1, beta1,alpha2, beta2, countThreshold)
```

**Arguments**

forCalcDF	Input dataframe generated from the tLOHDataImport function
alpha1	Model 1 alpha value
beta1	Model 1 beta value
alpha2	Model 2 alpha value
beta2	Model 2 beta value
countThreshold	Threshold for minimum number of read counts

**Value**

Output is a dataframe with Bayes Factor values

**Author(s)**

Michelle Webb

**Examples**

```
data('humanGBMsampleAC')
df <- tLOHCalcUpdate(humanGBMsampleAC, 1.25,1.25,500,500,4)
head(df)
```

---

tLOHDataImport	<i>Import VCF for tLOHCalc</i>
----------------	--------------------------------

---

**Description**

Import a VCF with per-cluster allele count information at heterozygous SNP positions for the tLOHCalc calculation function.

**Arguments**

vcf	An input VCF file. Spatial transcriptomics clusters make up the sample columns. AD and DP fields are required. Each SNP should be annotated with dbSNP rsIDs.
-----	---

**Value**

Output is a dataframe with required fields for tLOHCalc

**Author(s)**

Michelle Webb

**Examples**

```
## Not run:  
R.utils::gunzip("inst/extdata/Example.vcf.gz", "inst/extdata/Example.vcf")  
exampleDF <- tLOHDataImport("inst/extdata/Example.vcf")  
## End(Not run)
```

# Index

## \* datasets

- humanGBMsampleAC, [5](#)
- initialStartProbabilities, [6](#)

## \* marginalM2Calc

- marginalM2CalcBHET, [8](#)

aggregateCHRPlot, [2](#)

alleleFrequencyPlot, [3](#)

cols (removeOutlierFromCalc), [12](#)

dataframe (removeOutlierFromCalc), [12](#)

documentErrorRegions, [4](#)

hiddenMarkovAnalysis, [4](#)

humanGBMsampleAC, [5](#)

initialStartProbabilities, [6](#)

marginalLikelihoodM1, [6](#)

marginalLikelihoodM2, [7](#)

marginalM1Calc, [8](#)

marginalM2CalcBHET, [8](#)

marginalM2CalcBLOH, [9](#)

modePeakCalc, [10](#)

newValue (removeOutlierFromCalc), [12](#)

prepareHMMdataframes, [10](#)

regionAnalysis, [11](#)

regionFinalize, [12](#)

removeOutlierFromCalc, [12](#)

rows (removeOutlierFromCalc), [12](#)

runHMM\_1, [13](#)

runHMM\_2, [14](#)

runHMM\_3, [14](#)

splitByChromosome, [15](#)

summarizeRegions1, [16](#)

summarizeRegions2, [16](#)

tLOHCalc, [17](#)

tLOHCalcUpdate, [18](#)

tLOHDataImport, [19](#)