

# Package ‘CGHnormaliter’

May 5, 2024

**Type** Package

**Title** Normalization of array CGH data with imbalanced aberrations.

**Version** 1.58.0

**Date** 2012-06-12

**Author** Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling

**Maintainer** Bart P.P. van Houte <bvhoute@few.vu.nl>

**Depends** CGHcall (>= 2.17.0), CGHbase (>= 1.15.0)

**Imports** Biobase, CGHbase, CGHcall, methods, stats, utils

**Description** Normalization and centralization of array comparative genomic hybridization (aCGH) data. The algorithm uses an iterative procedure that effectively eliminates the influence of imbalanced copy numbers. This leads to a more reliable assessment of copy number alterations (CNAs).

**License** GPL (>= 3)

**biocViews** Microarray, Preprocessing

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

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** b59cc3d

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

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

## Contents

CGHnormaliter-package . . . . .	2
CGHnormaliter . . . . .	3
CGHnormaliter-internal . . . . .	4
CGHnormaliter.write.table . . . . .	5
Leukemia . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

CGHnormaliter-package *This package implements the CGHnormaliter algorithm which is a strategy for improved normalization of array Comparative Genomic Hybridization (aCGH) data*

---

## Description

CGHnormaliter is an iterative algorithm for normalization of aCGH data displaying imbalanced aberrations. First, provisory balanced copy numbers are identified and subsequently used for normalization based on LOWESS. These two steps are then iterated to refine the normalization. The assumption here is that the temporary exclusion of aberrations allows for a more appropriate calculation of the LOWESS regression curve. As a result, after normalization, the log<sub>2</sub> intensity ratios of the normals will generally be closer to zero and better reflect the biological reality.

## Details

Package: CGHnormaliter  
Type: Package  
Version: 1.5.6  
Date: 2011-04-06  
License: GPL (>= 3)

The package contains two public functions. The function `CGHnormaliter` performs the iterative normalization of aCGH data, while the function `CGHnormaliter.write.table` prints normalized aCGH data to a file. See function documentation for details.

## Author(s)

Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling

## References

Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling, Walter Pirovano and Jaap Heringa. CGH-normaliter: an iterative strategy to enhance normalization of array CGH data with imbalanced aberrations. *BMC Genomics*, 10:401, 2009.

## See Also

CGHcall, DNACopy

---

CGHnormaliter	<i>Iterative normalization of aCGH data</i>
---------------	---

---

**Description**

Implementation of an iterative algorithm for normalization of aCGH data displaying imbalanced aberrations.

**Usage**

```
CGHnormaliter(data, nchrom = 24, cellularity = 1, max.losses = 0.3, plot.MA = TRUE, ...)
```

**Arguments**

<code>data</code>	either a dataframe or character string containing a filename. See section Details for the format.
<code>nchrom</code>	number of chromosomes.
<code>cellularity</code>	vector of cellularities ranging from 0 to 1 to define the contamination of your sample with healthy cells (1 = no contamination). See Details for more information.
<code>max.losses</code>	vector of maximum fraction of losses in your samples, ranging from 0 to 1. In case this fraction is exceeded after calling, losses are deemed normals and centralization will be around the (lower) peak at the left of the log2 ratio distribution. See Details for more information.
<code>plot.MA</code>	logical. If 'plot.MA = TRUE', MA-plots before and after normalization are generated of each sample. The plots are saved into a PDF file.
<code>...</code>	arguments for segment (from package DNACopy) and/or CGHcall.

**Details**

The input should be either a `data.frame` or the file name of a tabseparated text file (text files must contain a header). The first four columns should contain the name, chromosome and the start and end position in bp for each array target respectively. The position columns must contain numbers only. Following these are two columns with the raw test and reference intensities for each of your samples. These intensities must be numeric as well. If the input type is a text file, missing values should be represented as 'NA' or an empty field.

The `cellularity` and `max.losses` parameters should both be a vector of length `n` where `n` is the number of samples in your dataset. Each vector is recycled if there are not enough values in it, or truncated if there are too many.

There is a `CGHnormaliter.write.table` method that prints the results in a tabular format.

**Value**

This function returns a matrix of objects of class `cghCall` with dimension (number of clones) \* (number of samples). Each object contains the following components (See section Examples on how to access them):

normalized data	A matrix with the normalized log2 intensity ratios for each profile.
segments	A matrix with the segments for each profile.
calls	A data.frame with the calls for each profile. Values are -1 (loss), 0 (normal) or 1 (gain).
probabilities	A data.frame with 3 columns of probe information (name, chromosome and position), followed by 3 columns with aberration probabilities for each sample.

**Author(s)**

Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling

**References**

Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling, Walter Pirovano and Jaap Heringa. CGH-normaliter: an iterative strategy to enhance normalization of array CGH data with imbalanced aberrations. *BMC Genomics*, 10:401, 2009.

**Examples**

```
data(Leukemia)
## Normalize the raw intensity values of the first 4 chromosomes.
result <- CGHnormaliter(Leukemia, nchrom=4)
## Get the normalized log2 intensity ratios, segments and calls
normalized.data <- copynumber(result)
segmented.data <- segmented(result)
called.data <- calls(result)
## Plot the normalization result of sample 2
plot(result[, 2])
## Write the normalized log2 intensity ratios to file
CGHnormaliter.write.table(result)
```

---

CGHnormaliter-internal

*Internal CGHnormaliter functions*

---

**Description**

Internal CGHnormaliter functions.

**Details**

These are not to be called by the user.

---

`CGHnormaliter.write.table`*Print normalized aCGH data to a file*

---

## Description

This function stores the results of a CGHnormaliter normalization into a plain text file.

## Usage

```
CGHnormaliter.write.table(input, data.type=c("normalized", "segmented", "called"), file=paste(data.ty
```

## Arguments

<code>input</code>	result object of a CGHnormaliter normalization.
<code>data.type</code>	type of aCGH data to store.
<code>file</code>	file name to store the data in.

## Details

Available aCGH data types are:

- `normalized`: The normalized log<sub>2</sub> intensity ratios
- `segmented`: The segmented log<sub>2</sub> intensity ratios (after normalization)
- `called`: The calls (after normalization)

The selected data are stored into a tabseparated file with the specified file name.

## Author(s)

Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling

## References

Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling, Walter Pirovano and Jaap Heringa. CGH-normaliter: an iterative strategy to enhance normalization of array CGH data with imbalanced aberrations. BMC Genomics, 10:401, 2009.

## See Also

[CGHnormaliter](#)

**Examples**

```

data(Leukemia)
## Normalize the intensity ratios
## Not run: result <- CGHnormaliter(Leukemia)
## Write the result to a file
## Not run: CGHnormaliter.write.table(result)
## Not run: CGHnormaliter.write.table(result, type="segmented")
## Not run: CGHnormaliter.write.table(result, type="called", file="mycalls.txt")

```

---

Leukemia	<i>Array CGH experiment data on childhood acute lymphoblastic leukemia (ALL) in humans</i>
----------	--

---

**Description**

A dataframe containing 30180 rows and 10 columns, representing the array CGH data of 3 ALL samples.

**Usage**

Leukemia

**Format**

A dataframe containing the following 10 columns:

- CloneID** The unique identifiers of array elements.
- Chromosome** Chromosome number of each array element.
- Start** Chromosomal start position in bp of each array element.
- End** Chromosomal end position in bp of each array element.
- Case1.test** Background corrected test intensity values for sample 1.
- Case1.ref** Background corrected reference intensity values for sample 1.
- Case2.test** Background corrected test intensity values for sample 2.
- Case2.ref** Background corrected reference intensity values for sample 2.
- Case3.test** Background corrected test intensity values for sample 3.
- Case3.ref** Background corrected reference intensity values for sample 3.

**Source**

Provided by the authors (see references).

**References**

Paulsson K, Heidenblad M, Morse H, Borg A, Fioretos T, Johansson B: Identification of cryptic aberrations and characterization of translocation breakpoints using array CGH in high hyperdiploid childhood acute lymphoblastic leukemia. *Leukemia* 2006, 20:2002-2007.

# Index

- \* **IO**
  - CGHnormaliter.write.table, 5
- \* **datasets**
  - Leukemia, 6
- \* **file**
  - CGHnormaliter.write.table, 5
- \* **internal**
  - CGHnormaliter-internal, 4
- \* **iteration**
  - CGHnormaliter, 3
- \* **methods**
  - CGHnormaliter, 3
- \* **package**
  - CGHnormaliter-package, 2
  - .calculateMA (CGHnormaliter-internal), 4
  - .localLowess (CGHnormaliter-internal), 4
  - .plotMA (CGHnormaliter-internal), 4
  - .readCghRaw (CGHnormaliter-internal), 4
  - .runCGHcall (CGHnormaliter-internal), 4
- CGHnormaliter, 3, 5
- CGHnormaliter-internal, 4
- CGHnormaliter-package, 2
- CGHnormaliter.write.table, 5
- Leukemia, 6