

# Package: JM4QTN (via r-universe)

May 23, 2026

**Type** Package

**Title** Joint Mapping for Quantitative Trait Loci

**Version** 1.0.0

**Date** 2026-05-09

**Description** A comprehensive computational framework for joint mapping, developed by Junhui Li (2016; doi:10.11841/j.issn.1007-4333.2016.06.002), supports quantitative trait locus detection in structured genetic populations. It integrates robust phenotype summarization, computes genotype probabilities, and imputes missing markers for association and linkage mapping. Empirical significance thresholds are estimated via permutation testing coupled with stepwise regression. The framework enables genome-wide scans under both univariate and multivariate trait models, streamlining the discovery of complex genetic architectures.

**Imports** lsmeans (>= 2.25.5), StepReg (>= 1.5.0)

**Depends** R (>= 3.5.0)

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.3.2

**Config/pak/sysreqs**

libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev

**Repository** <https://junhuili1017.r-universe.dev>

**Date/Publication** 2026-05-13 21:33:45 UTC

**RemoteUrl** <https://github.com/junhuili1017/jm4qtn>

**RemoteRef** HEAD

**RemoteSha** 7ab88ec961e6568bde4317ed03c6f4cff9aca23d

## Contents

JM4QTN-package . . . . .	2
expected_genotype_dist . . . . .	3
GeneticMap . . . . .	5
GenoData . . . . .	6
GenoData_EST . . . . .	7
GenoData_S2 . . . . .	8
genotype_freq . . . . .	9
genotype_prob . . . . .	11
haldane_map . . . . .	14
joint_map . . . . .	15
permutation_test . . . . .	17
pheno_stats . . . . .	19
PhenoData . . . . .	21
skeleton_build . . . . .	22
<b>Index</b>	<b>24</b>

---

JM4QTN-package	<i>Joint Mapping for QTN</i>
----------------	------------------------------

---

## Description

The "JM4QTN" package is based on composite interval mapping and regression analysis to do association analysis and linkage analysis for multiple-line cross population and multiple traits. The package can also identify pleiotropic or linked QTL.

## Details

Package: JM4QTN  
 Type: Package  
 Version: 1.0  
 Date: 2017-03-23  
 License: GPL (>= 2)

## Author(s)

JunhuLi,GuoliangLi,HeliChen,KunCheng,WenxinLiu  
 Maintainer: JunhuiLi<junhuili@cau.edu.cn>

## References

Junhui Li, Haixiao Hu, Yujie Meng, Kun Cheng, Guoliang Li, Wenxin Liu, and Shaojiang Chen.(2016)Pleiotropic QTL detection for stalk traits in maize and related R package programming. Journal of China Agricultural University. DOI 10.11841/j.issn.1007-4333.2016.06.00(in chinese)

---

expected\_genotype\_dist

*Expected Allele Genotype Distribution Calculator*

---

## Description

Calculates expected allele genotype distribution probabilities for different marker types and cross types in genetic mapping studies. This function implements sophisticated algorithms to compute genotype probabilities based on flanking marker information and population structure.

## Usage

```
expected_genotype_dist(marType, croType, Gn = 2, x, y = 0)
```

## Arguments

marType	<p>Character string specifying the marker type. Options include:</p> <ul style="list-style-type: none"> <li>"22", "21", "20", "12", "11", "10", "02", "01", "00": Both flanking markers observed</li> <li>"N2", "N1", "N0": Only right flanking marker observed</li> <li>"2N", "1N", "0N": Only left flanking marker observed</li> </ul> <p>These codes represent different combinations of flanking marker genotypes.</p>
croType	<p>Character string specifying the cross type:</p> <ul style="list-style-type: none"> <li>"Fn": F-n generation populations</li> <li>"BCP1": Backcross to parent 1</li> <li>"BCP2": Backcross to parent 2</li> <li>"F2": F2 generation</li> <li>"DH": Doubled haploid</li> <li>"RIL": Recombinant inbred line</li> </ul>
Gn	Numeric value specifying the generation number. Must be greater than 0.
x	Numeric value representing recombination fraction between loci A and Q. Must be between 0 and 0.5.
y	Numeric value representing recombination fraction between loci Q and B. Must be between 0 and 0.5. Default is 0.

## Details

This function calculates expected genotype probabilities using different approaches:

### Marker Type Classification:

- **Double flanking markers** ("22", "21", "20", etc.): Both left and right flanking markers are observed, providing maximum information
- **Single right flanking marker** ("N2", "N1", "N0"): Only the right flanking marker is observed
- **Single left flanking marker** ("2N", "1N", "0N"): Only the left flanking marker is observed

### Calculation Methods by Cross Type:

- **F<sub>n</sub> populations**: Uses [genotype\\_freq](#) for complex calculations involving multiple genotype classes
- **F<sub>2</sub>, DH, RIL**: Uses analytical formulas based on classical genetics
- **BCP1, BCP2**: Uses [genotype\\_freq](#) with specific genotype indices

**Mathematical Framework:** The function uses different probability calculations depending on the available information:

- When both flanking markers are observed, it uses conditional probabilities
- When only one flanking marker is observed, it uses marginal probabilities
- The calculations incorporate recombination fractions and population-specific parameters

**Genotype Coding:** The marker types use a coding system where:

- 2: Homozygous for alternative allele
- 1: Heterozygous
- 0: Homozygous for reference allele
- N: Missing or unknown genotype

## Value

A numeric value representing the expected probability of the specified genotype. The result is typically between -1 and 1, where positive values indicate higher probability of the target genotype.

## References

Haldane, J.B.S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics*, 8(3), 299-309.

## See Also

[genotype\\_freq](#) for genotype frequency calculations, [genotype\\_prob](#) for missing genotype imputation

## Examples

```
## Not run:
# Calculate probability for F2 population with both flanking markers
prob_f2 <- expected_genotype_dist("22", "F2", Gn = 2, x = 0.1, y = 0.2)

# Calculate probability for BCP1 with only right flanking marker
prob_bcp1_n2 <- expected_genotype_dist("N2", "BCP1", Gn = 3, x = 0.15, y = 0.25)

# Calculate probability for DH population with only left flanking marker
prob_dh_2n <- expected_genotype_dist("2N", "DH", Gn = 2, x = 0.1, y = 0)

# Calculate probability for Fn population with both flanking markers
prob_fn <- expected_genotype_dist("21", "Fn", Gn = 4, x = 0.2, y = 0.3)

# Calculate probability for RIL population
prob_ril <- expected_genotype_dist("00", "RIL", Gn = 2, x = 0.1, y = 0.2)

# Example with different recombination fractions
prob_low_rec <- expected_genotype_dist("22", "F2", Gn = 2, x = 0.05, y = 0.05)
prob_high_rec <- expected_genotype_dist("22", "F2", Gn = 2, x = 0.3, y = 0.4)

## End(Not run)
```

---

GeneticMap

*Data about genetic map*

---

## Description

A data frame with 1722 observations on the following 3 variables, where the first column is marker name, the second column is chromosome numeric ID and the last column is marker genetic distance position

## Usage

```
data("GeneticMap")
```

## Format

A data frame with 1722 observations on the following 3 variables.

marker a vector for marker name

chr a numeric vector for chromosome id

pos a numeric vector for marker genetic distance position

## Details

The name of marker must be the same with marker's name of data in GenoData, and the chromosome ID and genetic distance position are numeric, otherwise an error will occur.

## References

Hu, H., Meng, Y., Wang, H., Liu, H., and Chen, S. (2012). Identifying quantitative trait loci and determining closely related, stalk traits for rind penetrometer resistance in a high-oil maize, population. *Theoretical and Applied Genetics*, 124(8), 1439-1447.

Hu, H., Liu, W., Fu, Z., Homann, L., Technow, F., and Wang, H., et al. (2013). Qtl mapping of stalk bending strength in a recombinant inbred line maize population. *Theoretical and Applied Genetics*, 126(9), 2257-66.

## Examples

```
data(GeneticMap)
```

---

GenoData

*Data about molecular markers genotype*

---

## Description

A data frame with 647 individual observations on the following 1722 variables, where 2, 1 and 0 stands for different markers genotype AA AB and BB, NA for missing marker genotype

## Usage

```
data("GenoData")
```

## Format

A data frame with 647 individual observations on the following 1722 marker genotype variables

## Details

Missing markers are substituted only by NA, otherwise an error will occur.

## References

Hu, H., Meng, Y., Wang, H., Liu, H., and Chen, S. (2012). Identifying quantitative trait loci and determining closely related, stalk traits for rind penetrometer resistance in a high-oil maize, population. *Theoretical and Applied Genetics*, 124(8), 1439-1447.

Hu, H., Liu, W., Fu, Z., Homann, L., Technow, F., and Wang, H., et al. (2013). Qtl mapping of stalk bending strength in a recombinant inbred line maize population. *Theoretical and Applied Genetics*, 126(9), 2257-66.

## Examples

```
data(GenoData)
```

---

GenoData_EST	<i>Data about information of molecular markers chromosome id, genetic distance position and genotype with estimated missing markers</i>
--------------	---

---

**Description**

A data frame with 647 individual observations, chromosome id and genetic distance position on the following 1722 molecular maker variables

**Usage**

```
data("GenoData_EST")
```

**Format**

A data frame with 649 observations on the following 1722 variables

A data frame with 649 observations on the following 1722 maker variables, the rows 'chr' and 'pos' is the chromosome id and genetic distance position of markers, other rows are the marker genotype on 647 individuals, where 0 and 2 stands for genotype AA and BB respectively, decimal is missing marker conditional probability genotype.

**Details**

This data frame is combined from geneticMap and genotype data with estimated missing markers, which can be calculated by function calGenoProb(GeneticMap, GenoData, steps=0, croType, Gn)

**References**

Hu, H., Meng, Y., Wang, H., Liu, H., and Chen, S. (2012). Identifying quantitative trait loci and determining closely related, stalk traits for rind penetrometer resistance in a high-oil maize, population. *Theoretical and Applied Genetics*, 124(8), 1439-1447.

Hu, H., Liu, W., Fu, Z., Homann, L., Technow, F., and Wang, H., et al. (2013). Qtl mapping of stalk bending strength in a recombinant inbred line maize population. *Theoretical and Applied Genetics*, 126(9), 2257-66.

**Examples**

```
data(GenoData_EST)
```

---

GenoData_S2	<i>Data about information of molecular markers chromosome id, genetic distance position and estimated missing markers genotype with steps = 2</i>
-------------	---

---

### Description

A data frame with 647 individual observations, chromosome id and genetic distance position on the following 2431 molecular maker variables

### Usage

```
data("GenoData_S2")
```

### Format

A data frame with 649 observations on the following 2431 variables

A data frame with 649 observations on the following 2431 variables, the rows 'chr' and 'pos' is the chromosome id and genetic distance position of markers, other rows are the marker genotype on 647 individuals, where 0 and 2 stands for genotype AA and BB respectively, decimal is marker conditional probability genotype with steps = 2.

### Details

This data frame is combined from geneticMap and genotype data with estimated missing markers, which can be calculated by function calGenoProb(GeneticMap, GenoData, steps=2, croType, Gn)

### References

Hu, H., Meng, Y., Wang, H., Liu, H., and Chen, S. (2012). Identifying quantitative trait loci and determining closely related, stalk traits for rind penetrometer resistance in a high-oil maize, population. *Theoretical and Applied Genetics*, 124(8), 1439-1447.

Hu, H., Liu, W., Fu, Z., Homann, L., Technow, F., and Wang, H., et al. (2013). Qtl mapping of stalk bending strength in a recombinant inbred line maize population. *Theoretical and Applied Genetics*, 126(9), 2257-66.

### Examples

```
data(GenoData_S2)
```

**Description**

Calculates genotype frequencies for different cross types and generations in genetic mapping studies. This function implements complex recurrence relations to compute genotype frequencies in various genetic populations including Fn, backcross, and advanced generation populations.

**Usage**

```
genotype_freq(
  cross_type,
  generation = 2,
  genotype_index,
  recomb_aq,
  recomb_qb = 0
)
```

**Arguments**

cross_type	Population type used in the calculation: <ul style="list-style-type: none"> <li>• "Fn": F-n generation populations (<math>n &gt; 2</math>)</li> <li>• "BCP1": Backcross to parent 1 populations</li> <li>• "BCP2": Backcross to parent 2 populations</li> <li>• "F2": F2 generation populations</li> <li>• "DH": Doubled haploid populations</li> <li>• "RIL": Recombinant inbred line populations</li> </ul>
generation	Generation number (for example, 2 for F2, 3 for F3). Must be greater than 1.
genotype_index	Index of the genotype class to evaluate. Valid range depends on 'cross_type': <ul style="list-style-type: none"> <li>• "Fn": 1-20 (20 different genotype classes)</li> <li>• "BCP1/BCP2": 1-8 (8 genotype classes)</li> <li>• "F2": 1-9 (9 genotype classes)</li> </ul>
recomb_aq	Recombination fraction between loci A and Q. Use a value between 0 and 0.5.
recomb_qb	Recombination fraction between loci Q and B. Use a value between 0 and 0.5. Default is 0.

**Details**

This function calculates genotype frequencies using sophisticated mathematical models:

**For Fn Populations ( $y > 0$ ):**

- Handles 20 different genotype classes with complex recurrence relations
- Accounts for recombination between three loci (A, Q, B)

- Uses the relationship  $z = x + y - 2xy$  for flanking marker recombination
- Implements generation-by-generation frequency calculations

**For BCP1 Populations:**

- Handles 8 genotype classes
- Backcross to the first parent (AAQQBB)
- Specific recurrence relations for each genotype class

**For BCP2 Populations:**

- Handles 8 genotype classes
- Backcross to the second parent (aaqqbb)
- Different initial conditions and recurrence relations

**For F2 Populations (y = 0):**

- Handles 9 genotype classes
- Simplified two-locus model
- Standard F2 generation frequencies

**Mathematical Foundation:** The function uses recurrence relations to calculate genotype frequencies across generations. For each generation, the frequency of each genotype class is computed based on the frequencies in the previous generation and the recombination fractions between loci.

**Value**

A numeric value representing the frequency of the specified genotype in the given generation. The result is always between 0 and 1.

**References**

Haldane, J.B.S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics*, 8(3), 299-309.

**See Also**

[expected\\_genotype\\_dist](#) for expected allele genotype distribution calculations

**Examples**

```
## Not run:
# Calculate frequency for F2 generation, genotype 1, with recombination fraction 0.1
freq_f2 <- genotype_freq("Fn", generation = 2, genotype_index = 1, recomb_aq = 0.1, recomb_qb = 0)

# Calculate frequency for BCP1 generation, genotype 5, with recombination fractions
freq_bcp1 <- genotype_freq(
  "BCP1", generation = 3, genotype_index = 5, recomb_aq = 0.15, recomb_qb = 0.25
)

# Calculate frequency for Fn generation, genotype 10, with recombination fractions
```

```

freq_fn <- genotype_freq(
  "Fn", generation = 4, genotype_index = 10, recomb_aq = 0.2, recomb_qb = 0.3
)

# Calculate frequency for BCP2 generation, genotype 3
freq_bcp2 <- genotype_freq(
  "BCP2", generation = 2, genotype_index = 3, recomb_aq = 0.1, recomb_qb = 0.2
)

# Example with different generation numbers
freq_gen3 <- genotype_freq("Fn", generation = 3, genotype_index = 1, recomb_aq = 0.1, recomb_qb = 0)
freq_gen5 <- genotype_freq("Fn", generation = 5, genotype_index = 1, recomb_aq = 0.1, recomb_qb = 0)

## End(Not run)

```

---

genotype\_prob

*Calculate Genotype Probabilities and Impute Missing Data*


---

### Description

Calculates genotype probabilities and imputes missing genotype data for genetic mapping studies. Supports both Association Mapping (AM) and Linkage Mapping (LM) methods with comprehensive missing data handling and virtual marker creation capabilities.

### Usage

```
genotype_prob(GeneticMap, GenoData, method, croType = NULL, steps = 0, Gn = 2)
```

### Arguments

GeneticMap	A data frame containing genetic map information with columns: <ul style="list-style-type: none"> <li>• marker: Marker names (character)</li> <li>• chr: Chromosome numbers (numeric or factor)</li> <li>• pos: Genetic positions in centimorgans (numeric)</li> </ul>
GenoData	A matrix or data frame containing genotype data with individuals as rows and markers as columns. Genotype codes: <ul style="list-style-type: none"> <li>• 0: Homozygous for reference allele</li> <li>• 1: Heterozygous</li> <li>• 2: Homozygous for alternative allele</li> <li>• 9 or NA: Missing data</li> </ul>
method	Character string specifying the mapping method: <ul style="list-style-type: none"> <li>• "AM": Association Mapping - combines map with genotype data</li> <li>• "LM": Linkage Mapping - performs missing data imputation</li> </ul>
croType	Character string specifying the cross type for LM method. Required when method="LM":

	<ul style="list-style-type: none"> <li>• "RIL": Recombinant Inbred Lines</li> <li>• "DH": Doubled Haploids</li> <li>• "F2": F2 population</li> <li>• "BCP1": Backcross to Parent 1</li> <li>• "BCP2": Backcross to Parent 2</li> <li>• "Fn": Advanced generation (<math>n &gt; 2</math>)</li> </ul>
steps	Numeric value specifying the step size (in cM) for virtual marker creation in LM. If 0, no virtual markers are created. Default is 0.
Gn	Numeric value specifying the generation number for advanced populations. Must be greater than 1. Default is 2.

### Details

This function provides comprehensive genotype data processing for genetic mapping:

#### Association Mapping (AM):

- Simply combines genetic map with genotype data
- Converts missing values to code 9
- No imputation performed

#### Linkage Mapping (LM):

- Validates genotype codes based on cross type expectations
- Imputes missing genotypes using flanking marker information
- Creates virtual markers at specified intervals when  $steps > 0$
- Uses Haldane mapping function for recombination calculations
- Applies cross-type specific genotype probability calculations

#### Cross Type Genotype Expectations:

- RIL/DH: Only codes 0 and 2 allowed
- F2: Codes 0, 1, and 2 allowed
- BCP1/BCP2: Codes 0, 1, and 2 with specific constraints
- Fn: Codes 0, 1, and 2 for advanced generations

**Virtual Marker Creation:** When  $steps > 0$ , virtual markers are created at regular intervals between existing markers to improve mapping resolution and handle large gaps in the genetic map.

### Value

A data frame containing the processed genotype data with genetic map information and calculated genotype probabilities. The structure depends on the method:

- **AM method:** Original data combined with genetic map
- **LM method:** Imputed genotype data with probabilities and optional virtual markers

## References

Haldane, J.B.S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics*, 8(3), 299-309.

## See Also

[haldane\\_map](#) for recombination fraction calculations, [expected\\_genotype\\_dist](#) for expected genotype probabilities

## Examples

```
## Not run:
# Example genetic map
genetic_map <- data.frame(
  marker = c("M1", "M2", "M3", "M4", "M5"),
  chr = c(1, 1, 1, 1, 1),
  pos = c(0, 10, 25, 40, 50)
)

# Example genotype data with missing values
geno_data <- matrix(
  c(2, 0, NA, 1, 2,
    1, 0, 2, NA, 0,
    2, 1, 0, 2, 1,
    0, NA, 1, 0, 2),
  nrow = 4, ncol = 5,
  dimnames = list(c("Ind1", "Ind2", "Ind3", "Ind4"),
                 c("M1", "M2", "M3", "M4", "M5"))
)

# Association mapping (no imputation)
result_am <- genotype_prob(genetic_map, geno_data, method = "AM")

# Linkage mapping for F2 population with imputation
result_lm <- genotype_prob(genetic_map, geno_data, method = "LM",
                          croType = "F2", steps = 0)

# Linkage mapping with virtual marker creation
result_lm_vm <- genotype_prob(genetic_map, geno_data, method = "LM",
                              croType = "F2", steps = 5)

# RIL population example
result_ril <- genotype_prob(genetic_map, geno_data, method = "LM",
                            croType = "RIL", steps = 0)

## End(Not run)
```

---

`haldane_map`*Haldane Mapping Function*

---

**Description**

Converts genetic distance in Morgan units to recombination fraction using the Haldane mapping function. This function implements the Haldane mapping function which assumes no interference between crossovers.

**Usage**`haldane_map(x)`**Arguments**

`x` A numeric value representing the genetic distance in Morgan units (centimorgans/100). Must be non-negative.

**Details**

The Haldane mapping function is defined as:

$$r = \frac{1}{2}(1 - e^{-2x})$$

where  $x$  is the genetic distance in Morgan units and  $r$  is the recombination fraction.

This function assumes no interference between crossovers, meaning that the occurrence of one crossover does not affect the probability of other crossovers occurring nearby.

**Value**

A numeric value representing the recombination fraction ( $r$ ) between 0 and 0.5.

**References**

Haldane, J.B.S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics*, 8(3), 299-309.

**See Also**

[genotype\\_prob](#) for genotype probability calculations

**Examples**

```
# Convert 0.1 Morgan (10 cM) to recombination fraction
haldane_map(0.1)

# Convert 0.5 Morgan (50 cM) to recombination fraction
haldane_map(0.5)

# Convert 1.0 Morgan (100 cM) to recombination fraction
haldane_map(1.0)
```

joint\_map

*Joint Mapping Analysis for Multiple Traits***Description**

Performs comprehensive joint mapping analysis for multiple traits using either Association Mapping (AM) or Linkage Mapping (LM) methods to identify QTL affecting multiple traits simultaneously. This function implements sophisticated statistical methods for multivariate QTL analysis with comprehensive cofactor selection and significance testing.

**Usage**

```
joint_map(formula, data, skeleton, include, cut_off_list)
```

**Arguments**

formula	A model formula defining the full set of candidate linear-model terms (for example, factor population effects and marker-by-population interactions). Term labels from this formula are compared to the selected model in skeleton.
data	A data frame containing all variables used in formula and in skeleton\$call\$formula.
skeleton	A fitted model object (typically the final model from stepwise selection) that contains skeleton\$call\$formula. Commonly produced by skeleton_build() using the same formula, data, include, and cut_off_list.
include	Optional character vector of predictor names that should be treated as grouping variables (coerced to factor when formula contains : interactions). If NULL, interaction-based inclusion may be inferred from term labels. Matches the include argument used when fitting skeleton.
cut_off_list	A list whose component cut_off includes a named element "pvalue" (for example, output from <a href="#">permutation_test</a> ). Used upstream with skeleton_build() to obtain skeleton; it is part of the API for a consistent joint-mapping workflow but is not read inside joint_map().

**Value**

A list containing pvalue and LOD for each term in the formula:

- pvalue: P-value for each term
- lod: LOD score for each term

## References

Jiang, C. and Zeng, Z.B. (1995). Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics*, 140(3), 1111-1127.

## See Also

[permutation\\_test](#) for permutation thresholds, [genotype\\_prob](#) for genotype probability calculations

## Examples

```
## Not run:
# Example phenotype data
pheno_data <- data.frame(
  Trait1 = rnorm(100, mean = 100, sd = 15),
  Trait2 = rnorm(100, mean = 50, sd = 8),
  Popu = rep(c("Pop1", "Pop2"), each = 50)
)

# Example genotype data
geno_data <- matrix(sample(c(0,1,2), 100*50, replace = TRUE),
  nrow = 200, ncol = 50)
colnames(geno_data) <- paste0("M", 1:50)

data1 <- cbind(pheno_data,geno_data)

terms <- c("Popu", paste0(colnames(geno_data), ":Popu"))
formula1 <- reformulate(terms, response = "Trait1")

cut_off_list <- permutation_test(formula1, data1, n=100,
  alpha = 0.1)

skeleton <- skeleton_build(
  formula1, data1, strategy = "bidirection", metric = "SL",
  cut_off_list = cut_off_list
)

results <- joint_map(formula1, data1, skeleton, include = "Popu", cut_off_list = cut_off_list)

print(results)

formula2 <- reformulate(terms, response = "cbind(Trait1,Trait2)")

cut_off_list <- permutation_test(formula2, data1, n = 100, alpha = 0.1)

skeleton <- skeleton_build(
  formula2, data1, strategy = "bidirection", metric = "SL",
  cut_off_list = cut_off_list
)

results <- joint_map(formula2, data1, skeleton, include = "Popu", cut_off_list = cut_off_list)
```

```
print(results)

## End(Not run)
```

---

permutation\_test      *Permutation Test for Stepwise Regression*

---

## Description

Performs permutation tests for stepwise regression to determine empirical significance thresholds for QTL detection using stepwise regression. This function implements a comprehensive permutation testing framework for stepwise regression.

## Usage

```
permutation_test(
  formula,
  data,
  n = 1000,
  alpha = 0.1,
  include = NULL,
  strategy = "bidirection",
  metric = "SBC",
  type = "linear"
)
```

## Arguments

formula	A model formula defining the response and candidate predictors for permutation testing.
data	A data frame containing all variables referenced in formula, including response trait(s) and predictor variables.
n	Integer number of permutations to run. Larger values provide more stable empirical thresholds but increase computation time. Default is 1000.
alpha	Numeric significance level used to extract empirical cutoff values from the permutation distributions. Must be between 0 and 1. Default is 0.1.
include	Optional character vector of variable names that should always be included during stepwise model selection. If NULL, inclusion terms can be inferred from interaction terms in formula. Default is NULL.
strategy	Stepwise selection strategy passed to <code>StepReg::stepwise()</code> . Typical values include "forward", "backward", and "bidirection". Default is "bidirection".
metric	Model selection metric used inside stepwise regression. Typical values include "AIC", "BIC", or "SBC". Default is "SBC".
type	Type of regression model. Typical values include "linear", "logistic", and "poisson". Default is "linear".

## Details

This function implements a comprehensive permutation testing framework.

### Output Interpretation:

- **P-values:** Empirical significance thresholds for p-value
- **LOD scores:** Empirical significance thresholds for LOD score

## Value

A list containing the empirical significance thresholds for p-value and LOD score with:

- **cut\_off:** A vector containing the empirical significance thresholds for p-value and LOD score
- **pvalue:** A vector containing the p-values for each permutation
- **lod:** A vector containing the LOD scores for each permutation

## Examples

```
## Not run:
# Example phenotype data
pheno_data <- data.frame(
  Trait1 = rnorm(200, mean = 100, sd = 15),
  Trait2 = rnorm(200, mean = 50, sd = 8),
  Popu = rep(c("Pop1", "Pop2"), each = 100)
)

# Example genotype data
geno_data <- matrix(sample(c(0,1,2), 100*50, replace = TRUE),
  nrow = 200, ncol = 50)
colnames(geno_data) <- paste0("M", 1:50)

data1 <- cbind(pheno_data,geno_data)

terms <- c("Popu", paste0(colnames(geno_data), ":Popu"))
formula1 <- reformulate(terms, response = "Trait1")

cut_off_list <- permutation_test(formula1, data1, n=100,
  alpha = 0.1)

formula2 <- reformulate(terms, response = "cbind(Trait1,Trait2)")

cut_off_list <- permutation_test(formula1, data1, n=100,
  alpha = 0.1)

## End(Not run)
```

---

 pheno\_stats

*Statistical Analysis of Phenotype Data*


---

### Description

Performs comprehensive statistical analysis on phenotype data including normality tests, analysis of variance (ANOVA), and least squares means calculations for genetic studies.

### Usage

```
pheno_stats(phenoData, defineForm = NULL, effNotation = "G")
```

### Arguments

phenoData	A data frame containing phenotype data with at least 5 columns. The first 4 columns must be: Environment (E), Block (B), Repetition (R), and Genotype (G). Additional columns contain trait measurements for statistical analysis.
defineForm	Optional character vector of custom formula strings for statistical analysis. If NULL, default formulas are automatically generated based on the data structure.
effNotation	Character string specifying the effect notation for least squares means. Default is "G" for genotype effect.

### Details

This function performs a comprehensive statistical analysis pipeline:

1. **Normality Test:** Uses Shapiro-Wilk test to assess normality of each trait
2. **Model Selection:** Automatically generates appropriate statistical models based on data structure
3. **ANOVA:** Performs analysis of variance to test significance of effects
4. **Least Squares Means:** Calculates adjusted means for genotypes

**Automatic Model Generation:** The function automatically generates appropriate formulas based on the experimental design:

- Multiple environments and blocks:  $\text{trait} \sim E + G + E:G + B\%in\%E$
- Multiple environments only:  $\text{trait} \sim E*G$
- Custom formulas: User-defined formulas when `defineForm` is provided

**Data Requirements:** The phenotype data must have the following structure:

- Column 1: Environment (E) - factor variable
- Column 2: Block (B) - factor variable
- Column 3: Repetition (R) - factor variable
- Column 4: Genotype (G) - factor variable
- Columns 5+: Trait measurements - numeric variables

**Value**

A list containing comprehensive statistical analysis results for each trait:

normality_test	Results of Shapiro-Wilk normality test
formula	Formula used for the statistical model
ANOVA	Complete analysis of variance results
lsmeans	Least squares means for genotypes with standard errors

**References**

Shapiro, S.S. and Wilk, M.B. (1965). An analysis of variance test for normality. *Biometrika*, 52(3-4), 591-611.

**See Also**

[joint\\_map](#) for joint mapping analysis

**Examples**

```
## Not run:
# Example with multiple environments and blocks
pheno_data <- data.frame(
  E = rep(c("Env1", "Env2", "Env3"), each = 60),
  B = rep(c("Block1", "Block2"), each = 30, times = 3),
  R = rep(1:5, 36),
  G = rep(1:12, 15),
  Height = rnorm(180, 175, 8),
  Weight = rnorm(180, 75, 12)
)

# Perform statistical analysis with default formulas
results <- pheno_stats(pheno_data)

# View normality test results
results$Height$normality_test

# View ANOVA results
results$Height$ANOVA

# View least squares means
results$Height$lsmeans

# Example with custom formulas
custom_formulas <- c(
  "Height ~ E + G + E:G + B%in%E",
  "Weight ~ E + G + E:G"
)
results_custom <- pheno_stats(pheno_data, defineForm = custom_formulas)

## End(Not run)
```

---

PhenoData	<i>Data about all phenotype</i>
-----------	---------------------------------

---

**Description**

A data frame with 647 observations on the following 10 variables, the phenotype value is calculated by best linear unbiased estimates.

**Usage**

```
data("PhenoData")
```

**Format**

A data frame with 647 observations on the following 13 variables.

Indi The ID of individuals, which is a factor with levels of Pop1 with 131 individuals, Pop2 with 120 individuals, Pop3 with 200 individuals and Pop4 with 200 individuals

Popu a factor with levels Pop1 Pop2 Pop3 Pop4 for multiple-line cross population

Ma a factor with levels A D for parent 1

Pa a factor with levels B C E F for parent 2

newEC1 a numeric vector for trait newEC1

newEC2 a numeric vector for trait newEC2

newEC3 a numeric vector for trait newEC3

BM1 a numeric vector for trait BM1

BM2 a numeric vector for trait BM2

BM3 a numeric vector for trait BM3

predPH1 a numeric vector for trait predPH1

predPH2 a numeric vector for trait predPH2

predPH3 a numeric vector for trait predPH3

**Details**

Popu is the levels of multiple cross populations with different parent crosses, if it is filled with several levels, then linkage mapping or association mapping analysis is done for multiple cross populations, otherwise just for only one population

**References**

Hu, H., Meng, Y., Wang, H., Liu, H., and Chen, S. (2012). Identifying quantitative trait loci and determining closely related, stalk traits for rind penetrometer resistance in a high-oil maize, population. *Theoretical and Applied Genetics*, 124(8), 1439-1447.

Hu, H., Liu, W., Fu, Z., Homann, L., Technow, F., and Wang, H., et al. (2013). Qtl mapping of stalk bending strength in a recombinant inbred line maize population. *Theoretical and Applied Genetics*, 126(9), 2257-66.

**Examples**

```
data(PhenoData)
```

---

```
skeleton_build          Fit the stepwise "skeleton" model for joint mapping
```

---

**Description**

Runs `StepReg::stepwise()` with entry/stay levels set from a permutation p-value threshold, and returns the selected best linear model. The result is typically passed to `joint_map()` as the skeleton argument.

**Usage**

```
skeleton_build(
  formula,
  data,
  type = "linear",
  strategy = "bidirection",
  metric = "SL",
  include = NULL,
  cut_off_list
)
```

```
skeleton_build(
  formula,
  data,
  type = "linear",
  strategy = "bidirection",
  metric = "SL",
  include = NULL,
  cut_off_list
)
```

**Arguments**

formula	A model formula (same as used in <a href="#">permutation_test</a> ).
data	A data frame containing all variables in formula.
type	Model type passed to <code>StepReg::stepwise()</code> . Default is "linear".
strategy	Stepwise strategy, for example "forward", "backward", or "bidirection". Default is "bidirection".
metric	Information criterion or selection metric in <code>StepReg</code> , for example "SL". Default is "SL".
include	Optional character vector of terms to keep in the stepwise search, passed to <code>StepReg::stepwise()</code> . Default is NULL.

`cut_off_list` A list with a component `cut_off` that includes a named entry "pvalue" (for example, the return value of [permutation\\_test](#)). This value is used for both `sle` and `sls` in the stepwise call.

**Value**

The best model object for the chosen strategy and metric (an element of the `StepReg::stepwise()` result, typically with a `call` and `call$formula`).

**See Also**

[joint\\_map](#), [permutation\\_test](#)

# Index

## \* datasets

- GeneticMap, [5](#)
- GenoData, [6](#)
- GenoData\_EST, [7](#)
- GenoData\_S2, [8](#)
- PhenoData, [21](#)

## \* package JointMapping QTN

- JM4QTN-package, [2](#)

calculate\_expected\_genotype\_distribution  
(expected\_genotype\_dist), [3](#)

calculate\_genotype\_frequencies  
(genotype\_freq), [9](#)

expected\_genotype\_dist, [3](#), [10](#), [13](#)

GeneticMap, [5](#)

GenoData, [6](#)

GenoData\_EST, [7](#)

GenoData\_S2, [8](#)

genotype\_freq, [4](#), [9](#)

genotype\_prob, [4](#), [11](#), [14](#), [16](#)

haldane\_map, [13](#), [14](#)

haldane\_mapping\_function (haldane\_map),  
[14](#)

JM4QTN (JM4QTN-package), [2](#)

JM4QTN-package, [2](#)

joint\_map, [15](#), [20](#), [22](#), [23](#)

permutation\_test, [15](#), [16](#), [17](#), [22](#), [23](#)

pheno\_stats, [19](#)

PhenoData, [21](#)

skeleton\_build, [22](#)

skeleton\_build (skeleton\_build), [22](#)