Package ‘DESNP’

March 19, 2026

Type Package

Title Differentially Expressed Single Nucleotide Polymorphism

Version 0.1.0

Maintainer Sayanti Guha Majumdar <sayanti23gm@gmail.com>

Description Provides a framework for the identification and analysis of Differentially Expressed Single Nucleotide Polymorphisms (deSNPs) using high-throughput sequencing data. It enables users to import SNP count data from variant files, perform allele-specific read count extraction, and statistically detect SNPs showing significant differences in allele expression between biological conditions or sample groups. This package contains tools for calculating SNP-index and Delta SNP-index from VCF-derived allele depth data with statistical testing and filtering, including sliding-window analysis of genomic regions.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports vcfR, dplyr, tidyr, magrittr, tidyselect, VGAM, stats, utils, GenomicRanges, IRanges, S4Vectors, GenomeInfoDb, tools, ggplot2

NeedsCompilation no

RoxygenNote 7.3.3

Author Sayanti Guha Majumdar [aut, cre],
Saima Bano [aut],
Nandita Banerjee [aut],
Rahul Kumar Tiwari [aut],
Dipro Sinha [aut],
Subham Ghosh [aut],
Sanjeev Kumar [aut]

Repository CRAN

Date/Publication 2026-03-19 15:00:22 UTC

Contents

DeltaSNPindex *Compute Delta SNP-index*

Description

DeltaSNPindex calculates the difference in SNP-index values between two experimental conditions using output from the SNPindex() function. The Delta SNP-index (ΔSNP-index) is computed as the subtraction of SNP-index values between groups for each SNP position, representing allele frequency changes between conditions. This approach is based on the QTL-seq strategy described by Takagi et al. (2013). The function also supports automatic group detection and optional threshold-based filtering to identify SNPs showing substantial allele frequency differentiation.

Usage

```
DeltaSNPindex(
  snpindex_input,
  meta_file = NULL,
  filter_deltaSNP = TRUE,
  deltaSNP_threshold = 0.5,
  filter_direction  = c("both", "greater", "lesser"),
  column_filter     = c("all", "any")
)
```

Arguments

`snpindex_input` Either:

- Path to a tab-delimited SNP-index file (e.g., output of `SNPindex()`), or
- A data frame containing SNP-index columns.

The input must contain columns beginning with `SNPindex_`.

`meta_file` Optional path to a metadata file describing condition groupings.

The file must contain columns:

- `column` – SNP-index column name
- `condition` – condition label

Optional:

- `group` – allows multiple independent group comparisons

If `NULL`, automatic control detection mode is used.

`filter_deltaSNP`

Logical; whether to apply threshold-based filtering. Default: `TRUE`.

`deltaSNP_threshold`
Numeric threshold for filtering ΔSNP-index values. Default: `0.5`.

`filter_direction`
Filtering direction:

"both" Select SNPs with |ΔSNP-index| ≥ threshold

"greater" Select SNPs with ΔSNP-index ≥ threshold

"lesser" Select SNPs with ΔSNP-index ≤ -threshold

`column_filter` Determines filtering behavior when multiple delta SNP-index columns exist:

"any" SNP passes if any delta SNP-index column satisfies threshold

"all" SNP passes only if all delta SNP-index columns satisfy threshold

Details

ΔSNP-index is calculated as:

$$\Delta SNPindex = SNPindex_{condition2} - SNPindex_{condition1}$$

Modes of operation:

1. Metadata mode

If `meta_file` is provided:

- If a group has exactly 2 conditions → single ΔSNP-index column
- If more than 2 conditions → all pairwise comparisons are generated

2. Auto mode

If `meta_file` = `NULL`, the function:

- Automatically detects control columns (control/ctrl/c)
- Computes treatment vs control ΔSNP-index
- Falls back to pairwise comparisons if no control is detected

The function returns all computed ΔSNP-index values and optionally filtered results.

Value

Returns a list containing:

all_delta_SNPs Data frame containing all computed ΔSNP-index values

filtered_delta_SNPs Filtered SNPs based on threshold

summary Summary statistics of the analysis

Author(s)

Sayanti Guha Majumdar, Saima Bano, Nandita Banerjee, Rahul Kumar Tiwari, Dipro Sinha, Subham Ghosh, Sanjeev Kumar

References

Takagi, H., Abe, A., Yoshida, K., Kosugi, S., Natsume, S., Mitsui, H., Uemura, A., Utsushi, H., Tamiru, M., Takahashi, R., Goto, K., Terauchi, R. (2013). QTL-seq: rapid mapping of quantitative trait loci in rice by whole genome resequencing of DNA from two bulked populations. Plant Journal, 74(1), 174–183. doi:10.1111/tpj.12105

Examples

```
snp_file <- system.file("extdata", "All_SNPs_Index.tsv",
                        package = "DESNP")

meta_file <- system.file("extdata", "meta_columns.tsv",
                         package = "DESNP")

result_deltasnpindex <- DeltaSNPindex(
  snpindex_input = snp_file,
  meta_file = meta_file,
  deltaSNP_threshold = 0,
  filter_direction = "both",
  column_filter = "any"
)

head(result_deltasnpindex$all_delta_SNPs)
head(result_deltasnpindex$filtered_delta_SNPs)
result_deltasnpindex$summary
```

`DeltaSNPWindow` *Sliding Window Analysis of ΔSNP-index*

Description

DeltaSNPWindow compute ΔSNP-index values across chromosomes using an overlapping sliding-window approach to identify genomic regions exhibiting allele frequency differentiation between experimental conditions. The method follows the QTL-seq framework described by Takagi et al. (2013), which builds upon the SNP-index strategy introduced by Abe et al. (2012). The function computes the mean ΔSNP-index within each window to highlight consistent allele frequency differences and optionally applies a Wilcoxon signed-rank test to evaluate whether window-level deviations significantly differ from zero. Multiple testing correction is performed using the Benjamini–Hochberg method. Regions showing strong and statistically supported ΔSNP-index signals represent candidate loci enriched for biologically relevant allele frequency shifts.

Usage

```
DeltaSNPWindow(
  deltasnp_input,
  delta_cols = NULL,
  window_size = 1e6,
  step_size = 1e5,
  run_wilcoxon = TRUE
)
```

Arguments

`deltasnp_input` A file path (TSV format) or a data frame containing SNP data. The input must contain at least the following columns: `CHROM` (chromosome name) and `POS` (genomic position).

`delta_cols` Character vector or numeric indices specifying ΔSNP-index columns. If `NULL`, columns containing both "delta" and "snp" (case-insensitive) and numeric values are automatically detected.

`window_size` Size of sliding windows in base pairs. Default is `1e6` (1 Mb).

`step_size` Step size for sliding windows in base pairs. Default is `1e5` (100 kb).

`run_wilcoxon` Logical value indicating whether to perform Wilcoxon signed-rank test for each window. If `TRUE`, p-values and FDR-adjusted p-values (Benjamini-Hochberg method) are computed.

Details

The function:

- Generates sliding windows for each chromosome.
- Calculates mean ΔSNP-index within each window.
- Optionally performs Wilcoxon signed-rank test to test deviation from zero.
- Adjusts p-values using Benjamini-Hochberg FDR correction.

Value

A data frame containing:

- `seqnames` – Chromosome name
- `start` – Window start position
- `end` – Window end position
- `width` – Window width
- `strand` – Genomic strand
- Mean ΔSNP-index columns (prefixed with `mean_`)
- Wilcoxon p-values (if enabled)
- FDR-adjusted p-values (if enabled)

Author(s)

Sayanti Guha Majumdar, Saima Bano, Nandita Banerjee, Rahul Kumar Tiwari, Dipro Sinha, Subham Ghosh, Sanjeev Kumar

References

Abe, A., Kosugi, S., Yoshida, K., Natsume, S., Takagi, H., Kanzaki, H., Matsumura, H., Yoshida, K., Mitsuoka, C., Tamiru, M., Innan, H., Cano, L. M., Kamoun, S., Terauchi, R. (2012). Genome sequencing reveals agronomically important loci in rice using MutMap. Nature Biotechnology, 30(2), 174–178. doi:10.1038/nbt.2095

Takagi, H., Abe, A., Yoshida, K., Kosugi, S., Natsume, S., Mitsui, H., Uemura, A., Utsushi, H., Tamiru, M., Takahashi, R., Goto, K., Terauchi, R. (2013). QTL-seq: rapid mapping of quantitative trait loci in rice by whole genome resequencing of DNA from two bulked populations. Plant Journal, 74(1), 174–183. doi:10.1111/tpj.12105

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B, 57(1), 289–300.

Examples

```
example_file <- system.file(
  "extdata",
  "All_delta_SNPs_Index.tsv",
  package = "DESNP"
)

window_result <- DeltaSNPWindow(
  deltasnp_input = example_file,
  window_size = 5e5,
  step_size = 1e5,
  run_wilcoxon = TRUE
)

head(window_result)
```

manhattan_plot *Genome-wide Manhattan Plot for SNP Index and ΔSNP Index*

Description

Generates a customizable genome-wide Manhattan plot for SNP index (`SNPindex_`) or ΔSNP index (`delta_snp_index_`) values. The function automatically detects appropriate SNP index columns or allows the user to specify columns manually. It computes cumulative genomic positions across chromosomes and produces a `ggplot` object. Faceting is supported when multiple SNP index columns are plotted.

Usage

```
manhattan_plot(
  input_data,
  plot_cols = NULL,
  plot_title = "Genome-wide Manhattan Plot of SNP Index",
  x_label = "Chromosome",
  y_label = NULL,
  y_limits = c(0, 1),
  y_breaks = NULL,
  x_expand = 0.01,
  point_size = 0.7,
  point_alpha = 0.7,
  hline_value = 0.5,
  hline_linetype = "dashed",
  hline_color = "black",
  title_size = 14,
  title_face = "bold",
  axis_title_size = 12,
  axis_text_size = 8,
  axis_text_angle = 40,
  facet_ncol = 1,
  facet_scales = "free_y"
)
```

Arguments

`input_data` Either a path to a tab-delimited SNP result file or a data frame containing SNP data.

The data must contain:

- `CHROM` — Chromosome name
- `POS` — Genomic position
- One or more SNP index or ΔSNP index columns

Column names are case-insensitive for `CHROM` and `POS`.

`plot_cols` Optional character vector specifying which columns to plot.

If NULL (default):

- The function first searches for columns matching `^delta_snp_index(_.*)?$`
- If none found, searches for `^SNPindex_`
- Delta SNP index columns are prioritized

`plot_title` Main title of the Manhattan plot.

`x_label` Label for x-axis.

`y_label` Label for y-axis. Automatically set to ΔSNP index for delta columns, or “SNP index” for SNPindex columns if NULL.

`y_limits` Numeric vector specifying y-axis limits. Default: `c(0, 1)`

`y_breaks` Custom y-axis tick marks. If NULL, ggplot default is used.

`x_expand` Expansion factor for x-axis spacing.

`point_size` Size of plotted SNP points.

`point_alpha` Transparency level of points (0–1).

`hline_value` Y-value for horizontal reference line. Default: `0.5`

`hline_linetype` Line type for horizontal reference line.

`hline_color` Color of horizontal reference line.

`title_size` Font size of plot title.

`title_face` Font face of title (e.g., "bold").

`axis_title_size` Font size of axis titles.

`axis_text_size` Font size of axis tick labels.

`axis_text_angle` Rotation angle of chromosome labels.

`facet_ncol` Number of columns when faceting multiple SNP index types.

`facet_scales` Scaling for faceted plots ("fixed", "free", "free_y").

Details

Automatic Column Detection

If `plot_cols` = `NULL`, the function:

1. Searches for columns matching `^delta_snp_index(_.*)?$`
2. If none found, searches for `^SNPindex_`
3. Stops if no matching columns are detected.

Cumulative Genome Position

To create a genome-wide Manhattan plot across multiple chromosomes:

- Chromosome lengths are calculated using maximum POS.
- Cumulative offsets are computed.
- SNP positions are shifted accordingly.

This ensures chromosomes appear sequentially along the x-axis.

Faceting

If more than one SNP index column is plotted:

- Separate panels are created using `facet_wrap()`
- Y-axis scaling can be controlled via `facet_scales`

Plot Output

The function **returns a** `ggplot` **object**. Use `ggsave()` if you want to save PNG, PDF, or TIFF files manually.

Value

A `ggplot` object representing the Manhattan plot.

Author(s)

Sayanti Guha Majumdar, Saima Bano, Nandita Banerjee, Rahul Kumar Tiwari, Dipro Sinha, Subham Ghosh, Sanjeev Kumar

Examples

```
example_file <- system.file(
  "extdata",
  "All_delta_SNPs_Index.tsv",
  package = "DESNP"
)

manhattan_plot(input_data = example_file)

# Plot specific column
manhattan_plot(
  input_data = example_file,
  plot_cols = c("delta_snp_index_SNPindex_trt_200dai_vs_SNPindex_control_200dai")
)

# Custom y-limits and horizontal line
manhattan_plot(
  input_data = example_file,
  y_limits = c(-1, 1),
  hline_value = 0
)
```

SNPindex *SNP-index Calculation*

Description

Computes SNP-index values from allele depth (AD) fields of the VCF file, performs statistical comparison between experimental conditions, and identifies significantly differentiated SNPs following the SNP-index framework described by Abe et al. (2012). The function supports beta-binomial regression and Fisher's exact test, with automatic test selection based on replicate availability.

Usage

```
SNPindex(
  vcf_file,
  metadata_file,
  min_depth = 10,
  test_method = c("auto", "beta_binomial", "fisher_exact"),
  PVALUE = 0.05,
  Pvalue_filter = c("all", "any")
)
```

Arguments

`vcf_file`	Path to the input VCF file. The VCF file must contain allele depth (AD) values in the genotype (FORMAT) field.

`metadata_file` Path to a metadata file describing sample information. The file must be tab- or whitespace-separated with a header row.

Required columns:

- `sample` – Sample names exactly matching the genotype column names of the VCF file (case-sensitive).
- `condition` – Experimental group label (e.g., control, treatment).

Optional columns:

Any additional columns (e.g., `tissue`, `batch`, `replicate`) are automatically treated as grouping variables for stratified analysis.

Important notes:

- Sample names must match VCF file sample names exactly.
- At least two unique `condition` values are required.
- Mismatched samples will be removed during merging.

Example without grouping variables:

sample	condition
S1	control
S2	control
S3	treatment
S4	treatment

Example with grouping variables:

sample	condition	tissue	batch
S1	control	leaf	1
S2	control	root	1
S3	treatment	leaf	1
S4	treatment	root	2

`min_depth` Minimum total read depth (REF + ALT) required for a SNP to be retained. Default: `10`.

`test_method` Statistical method used to compare allele frequencies between conditions.

"auto" Automatically selects beta-binomial if $\geq$2 replicates per condition are detected; otherwise Fisher's exact test is used.

"beta_binomial" Forces beta-binomial regression using `VGAM::vglm()`.

"fisher_exact" Forces Fisher's exact test using `stats::fisher.test()`.

`PVALUE` Significance threshold for identifying significant SNPs. Default: `0.05`. Filtering is based on raw p-values.

`Pvalue_filter` Filtering behavior when grouping variables are present.

"all" SNP must be significant in all groups.

"any" SNP is significant if significant in at least one group.

Details

The SNP-index is calculated as:

$$SNPindex = ALT/(REF + ALT)$$

SNPs with total depth below `min_depth` are removed before testing.

Statistical testing strategy:

- If ≥ 2 biological replicates are detected per condition, beta-binomial regression is applied.
- If replicates are insufficient, Fisher's exact test is used.
- In `"auto"` mode, the test is selected automatically.

P-values are adjusted for multiple testing using the Benjamini–Hochberg (BH) method and reported as FDR.

Value

Returns a list containing:

all_SNPs Data frame of all analyzed SNPs including allele counts, SNP-index values, p-values, and FDR.

significant_SNPs Subset of SNPs passing the `PVALUE` threshold.

summary Summary statistics including total SNPs, number of significant SNPs, test used, depth threshold, and PVALUE.

Author(s)

Sayanti Guha Majumdar, Saima Bano, Nandita Banerjee, Rahul Kumar Tiwari, Dipro Sinha, Subham Ghosh, Sanjeev Kumar

References

Abe, A., Kosugi, S., Yoshida, K., Natsume, S., Takagi, H., Kanzaki, H., Matsumura, H., Yoshida, K., Mitsuoka, C., Tamiru, M., Innan, H., Cano, L. M., Kamoun, S., & Terauchi, R. (2012). Genome sequencing reveals agronomically important loci in rice using MutMap. Nature Biotechnology, 30(2), 174–178. doi:10.1038/nbt.2095

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate. Journal of the Royal Statistical Society: Series B, 57(1), 289–300.

See Also

`read.vcfR`, `vglm`, `fisher.test`

Examples

```
vcf_path <- system.file("extdata", "combined_top10_new_smut.vcf",
                        package = "DESNP")

meta_path <- system.file("extdata", "metadata.tsv",
                         package = "DESNP")


result_snpindex <- SNPindex(
  vcf_file = vcf_path,
  metadata_file = meta_path,
  min_depth = 10,
  test_method = "auto",
  PVALUE = 0,
  Pvalue_filter = "any"
)


head(result_snpindex$all_SNPs)
head(result_snpindex$significant_SNPs)
result_snpindex$summary
```

Index