

# Package: POINT (via r-universe)

August 30, 2024

**Type** Package

**Title** Protein Structure Guided Local Test

**Version** 1.3

**Date** 2023-10-01

**Author** Rachel Marceau West, Shannon T. Holloway

**Maintainer** Shannon T. Holloway <shannon.t.holloway@gmail.com>

**Description** Provides an implementation of a rare variant association test that utilizes protein tertiary structure to increase signal and to identify likely causal variants. Performs structure-guided collapsing, which leads to local tests that borrow information from neighboring variants on a protein and that provide association information on a variant-specific level. For details of the implemented method see West, R. M., Lu, W., Rotroff, D. M., Kuenemann, M., Chang, S-M., Wagner M. J., Buse, J. B., Motsinger-Reif, A., Fourches, D., and Tzeng, J-Y. (2019) <[doi:10.1371/journal.pcbi.1006722](https://doi.org/10.1371/journal.pcbi.1006722)>.

**License** GPL-2

**Depends** methods, stats, rARPACK, Matrix, CompQuadForm

**NeedsCompilation** no

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**Collate** 'A\_Kernel.R' 'A\_BurdenKernel.R' 'A\_LinearKernel.R'  
'A\_PolyKernel.R' 'B\_PvMethod.R' 'B\_PvMethod\_Davies.R'  
'B\_PvMethod\_Liu.R' 'C\_BinomialTrait.R' 'C\_GaussianTrait.R'  
'D\_NullResult.R' 'calcLocalKernel.R' 'distanceMatrix.R'  
'mainCode.R' 'point.R' 'pvResamp.R'

**Date/Publication** 2023-10-01 18:00:02 UTC

**Repository** <https://sth1402.r-universe.dev>

**RemoteUrl** <https://github.com/cran/POINT>

**RemoteRef** HEAD

**RemoteSha** 1f5de2f0390c7d0b273d34eacf09257bc2ca502d

## Contents

point . . . . .	2
<b>Index</b>	<b>5</b>

---

point	<i>Protein Structure Guided Local Test</i>
-------	--

---

### Description

A rare variant association test that utilizes protein tertiary structure to increase signal and to identify likely causal variants. Performs structure-guided collapsing, which leads to local tests that borrow information from neighboring variants on a protein and that provide association information on a variant-specific level.

### Usage

```
point(
  yy,
  X,
  snp,
  proteinCoord,
  ...,
  trait = "binomial",
  cValues = c(0, 0.1, 0.2, 0.3, 0.4, 0.5),
  weighted = TRUE,
  weight = NULL,
  kernel = "linear",
  d = NULL,
  pvMethod = "davies",
  nperturb = 1000,
  verbose = TRUE
)
```

### Arguments

<code>yy</code>	numeric vector; phenotype values.
<code>X</code>	numeric matrix; non-genetic covariates.
<code>snp</code>	numeric matrix; genotype snp matrix (count of minor alleles). Matrix cannot contain missing values.
<code>proteinCoord</code>	numeric matrix; columns correspond to 3 dimensional coordinates (x,y,z) of each variant in the protein tertiary structure.
<code>...</code>	optional additional arguments for p-value methods <code>CompQuadForm::davies</code> and <code>CompQuadForm::liu</code> .
<code>trait</code>	character; type of phenotype data. Must be one of { 'gaussian', 'binomial' } quantitative or case control data, respectively.

cValues	numeric vector; c values from which to choose the optimal neighborhood size for borrowing significant information.
weighted	logical; whether or not to weight the local kernel test using (non-distance based) weights.
weight	numeric vector (optional) If NULL and weighted is TRUE $(1.0-MAF)^{24}$ . Ignored if weighted is FALSE.
kernel	character; type of local kernel to use; Must be one of {'burden', 'linear', 'polynomial'}.
d	numeric; If kernel = 'poly', d is the order of the polynomial kernel.
pvMethod	character; method of calculating the p-value of each single marker test for fixed c values. Must be one of {'davies', 'liu'}.
nperturb	numeric, number of perturbations/resamples (perturbed test statistics) to calculate p-value of minP statistic.
verbose	logical; generate progress screen prints.

### Value

Returns a matrix the rows of which correspond to individual markers. Columns correspond to:

- (1) minP statistic;
- (2) local kernel test p-value;
- (3) optimal scale value from input cValues;
- (4) minor allele frequency; and
- (5) single variant score test p-value.

### Examples

```
# number of subjects
nsubj <- 1000

# number of markers
nm <- 5

# generate coordinates for proteins
protein <- cbind( stats::rnorm(n = nm, mean = 17.6, sd = 6.6),
                 stats::rnorm(n = nm, mean = 1.6, sd = 13.6),
                 stats::rnorm(n = nm, mean = 22.9, sd = 10.4) )

# generate snp matrix
snp <- matrix(data = rbinom(n = nsubj*nm, size = 1, p = 0.02),
              nrow = nsubj, ncol = nm)
colnames(snp) = paste0("m",1:nm)

# generate binomial response
MAF <- colMeans(x = snp)/2
causal <- numeric(nm)
causal[c(2,4)] <- 1.0
betaG <- 0.4*abs(log10(x = MAF))*causal

#no non-genetic covariates
```

```
X <- NULL
mu <- -0.05 + snp %*% betaG

pryy <- exp(mu)/(1+exp(mu))
yy <- sapply(X = pryy, FUN = stats::rbinom, n = 1, size = 1)

res <- point(yy = yy, X = X, snp = snp, proteinCoord = protein,
             trait = 'binomial', cValues = c(0.1,0.2),
             weighted = TRUE, weight = NULL, kernel = 'linear',
             pvMethod = 'liu', nperturb = 100,
             verbose = FALSE)
```

# Index

point, [2](#)