

# Package ‘VarCon’

November 4, 2025

**Type** Package

**Title** VarCon: an R package for retrieving neighboring nucleotides of an SNV

**Version** 1.19.0

**Description** VarCon is an R package which converts the positional information from the annotation of a single nucleotide variation (SNV) (either referring to the coding sequence or the reference genomic sequence). It retrieves the genomic reference sequence around the position of the single nucleotide variation. To assess, whether the SNV could potentially influence binding of splicing regulatory proteins VarCon calculates the HEXplorer score as an estimation. Besides, VarCon additionally reports splice site strengths of splice sites within the retrieved genomic sequence and any changes due to the SNV.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**Imports** methods, stats, IRanges, shiny, shinycssloaders, shinyFiles, ggplot2

**Depends** Biostrings, BSgenome, GenomicRanges, R (>= 4.1)

**Suggests** testthat, knitr, rmarkdown

**biocViews** FunctionalGenomics, AlternativeSplicing

**RoxygenNote** 7.1.1

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

**git\_branch** devel

**git\_last\_commit** 479604c

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2025-11-03

**Author** Johannes Ptak [aut, cre]

**Maintainer** Johannes Ptak <Johannes.Ptak@posteo.de>

Contents

|                                    |   |
|------------------------------------|---|
| calculateHZEIperNT . . . . .       | 2 |
| calculateMaxEntScanScore . . . . . | 3 |
| gene2transcript . . . . .          | 3 |
| generateHEXplorerPlot . . . . .    | 4 |
| getMaxEntInfo . . . . .            | 5 |
| getSeqInfoFromVariation . . . . .  | 5 |
| hbg . . . . .                      | 6 |
| hex . . . . .                      | 7 |
| prepareReferenceFasta . . . . .    | 7 |
| referenceDnaStringSet . . . . .    | 8 |
| startVarConApp . . . . .           | 8 |
| transCoord . . . . .               | 9 |

|       |    |
|-------|----|
| Index | 10 |
|-------|----|

---

|                    |                                                                       |
|--------------------|-----------------------------------------------------------------------|
| calculateHZEIperNT | <i>Generates table with HZEI scores per nucleotide of a sequence.</i> |
|--------------------|-----------------------------------------------------------------------|

---

Description

This function generates a table with HZEI scores per index nucleotide.

Usage

```
calculateHZEIperNT(seq)
```

Arguments

|     |                                                                                    |
|-----|------------------------------------------------------------------------------------|
| seq | Nucleotide sequence longer than 11nt and only containing bases "A", "G", "C", "T". |
|-----|------------------------------------------------------------------------------------|

Value

Dataframe with HZEI value per index position.

Examples

```
calculateHZEIperNT("TTCCAAACGAACTTTGTAGGGA")
```

---

calculateMaxEntScanScore

*Calculate MaxEntScan score of a splice site sequence*


---

### Description

This function calculates the MaxEntScan score of either splice donor or acceptor sequences.

### Usage

```
calculateMaxEntScanScore(seqVector, ssType)
```

### Arguments

|           |                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| seqVector | Character vector of nucleotide sequence of a splice site sequences. SA sequences should be 23nt long (20 intronic, 3 exonic) and splice donor sequences should be 9nt long (3 exonic, 6 intronic) only contain bases "A", "G", "C", "T". |
| ssType    | Numeric indicator, if the entered sequence is a splice donor (5) or acceptor (3)                                                                                                                                                         |

### Value

Character vector of the MaxEntScan scores generated from the entered seqVector.

### Examples

```
calculateMaxEntScanScore("TTCCAAACGAACTTTGTAGGGA",3)
calculateMaxEntScanScore("GAGGTAAGT",5)
```

---

|                 |                                                                                      |
|-----------------|--------------------------------------------------------------------------------------|
| gene2transcript | <i>Small data frame specifying a transcript to certain genes for synonymous use.</i> |
|-----------------|--------------------------------------------------------------------------------------|

---

### Description

Small data frame specifying a transcript to certain genes for synonymous use.

### Usage

```
gene2transcript
```

### Format

data frame

**gene\_name** HGNC gene name

**gene\_ID** Ensembl gene ID

**transcript\_ID** Ensembl transcript ID

**Examples**

```
gene2transcript
```

---

`generateHEXplorerPlot` *Generates plot with HZEI values and splice site strengths from a list holding information about an SNV.*

---

**Description**

This function generates a plot depicting the HZEI score changes and changes in the HBS or Max-EntScan score, from a sequence variation.

**Usage**

```
generateHEXplorerPlot(variationInfoList, ntWindow)
```

**Arguments**

`variationInfoList`      Output from the `getSeqInfoFromVariation` function.

`ntWindow`              Numeric value defining the sequence surrounding of interest.

**Value**

Plot stating the HZEI values per nt and splice site strength with and without the SNV.

**Examples**

```
#Defining exemplary input data
transcriptTable <- transCoord    # Using pseudo transcript table
transcriptID <- "pseudo_ENST00000650636"    # Using pseudo transcript
variation <- "c.412C>G/p.(T89M)"
ntWindow <- 20
gene2transcript <- data.frame(gene_name = "Example_gene",
gene_ID = "pseudo_ENSG00000147099", transcript_ID = "pseudo_ENST00000650636")

results <- getSeqInfoFromVariation(referenceDnaStringSet, transcriptID, variation, ntWindow=ntWindow, transcriptID=transcriptID)

generateHEXplorerPlot(results)
```

---

|               |                                                                          |
|---------------|--------------------------------------------------------------------------|
| getMaxEntInfo | <i>Generates table with MaxEntScan scores per potential SA position.</i> |
|---------------|--------------------------------------------------------------------------|

---

### Description

This function generates a table with MaxEntScan scores per potential SA position.

### Usage

```
getMaxEntInfo(seq)
```

### Arguments

|     |                                                                                    |
|-----|------------------------------------------------------------------------------------|
| seq | Nucleotide sequence longer than 22nt and only containing bases "A", "G", "C", "T". |
|-----|------------------------------------------------------------------------------------|

### Value

Dataframe of potential acceptor index positons and corresponding MaxEntScan scores.

### Examples

```
getMaxEntInfo("TTCCAAACGAACTTTGTAGGGA")
```

---

|                         |                                                                         |
|-------------------------|-------------------------------------------------------------------------|
| getSeqInfoFromVariation | <i>Collects information about genomic context of sequence variants.</i> |
|-------------------------|-------------------------------------------------------------------------|

---

### Description

This function collects information about genomic context of sequence variants.

### Usage

```
getSeqInfoFromVariation(referenceDnaStringSet, transcriptID,  
variation, ntWindow=20, transcriptTable, gene2transcript=gene2transcript)
```

### Arguments

|                       |                                                                                                             |
|-----------------------|-------------------------------------------------------------------------------------------------------------|
| referenceDnaStringSet | DNAStringset from the reference genome fasta file.                                                          |
| transcriptID          | Ensembl ID of the transcript of interest.                                                                   |
| variation             | A sequence variation either referring to coding sequence or the genomic sequence (c.12A>T, or g.182284A>T). |

ntWindow            Numeric value defining the sequence surrounding of interest.  
transcriptTable    Table of transcripts and their exon coordinates and CDS coordinates.  
gene2transcript    Gene to transcript conversion table with the gene name in the first column and  
                     the gene ID in the second and the transcript ID in the third column.

**Value**

List of informations about the entered variation.

**Examples**

```
#Defining exemplary input data
transcriptTable <- transCoord
transcriptID <- "pseudo_ENST00000650636"
variation <- "c.412C>G/p.(T89M)"
gene2transcript <- data.frame(gene_name = "Example_gene",
gene_ID = "pseudo_ENSG00000147099", transcriptID = "pseudo_ENST00000650636")

results <- getSeqInfoFromVariation(referenceDnaStringSet, transcriptID,
variation, ntWindow=20, transcriptTable, gene2transcript=gene2transcript)

#Using a predefined gene to transcript conversion
transcriptID <- "Example_gene"
results <- getSeqInfoFromVariation(referenceDnaStringSet, transcriptID,
variation, ntWindow=20, transcriptTable, gene2transcript=gene2transcript)
```

---

|     |                                      |
|-----|--------------------------------------|
| hbg | <i>Donor sequences and their HBS</i> |
|-----|--------------------------------------|

---

**Description**

Donor sequences and their HBS

**Usage**

hbg

**Format**

A data frame with columns:  
**seq** 11nt long donor sequence  
**hbs** HBS of the donor sequence

**Examples**

hbg

---

|     |                              |
|-----|------------------------------|
| hex | <i>Hexamers and Z scores</i> |
|-----|------------------------------|

---

**Description**

Hexamers and Z scores

**Usage**

hex

**Format**

A data frame with columns:

**seq** Sequence of the hexamer.

**value** ZEI-score of the hexamer from HEXplorer.

**first** First codon within the hexamer.

**second** Second codon within the hexamer.

**first\_AA** First encoded amino acid within the hexamer (three letter code).

**second\_AA** Second encoded amino acid within the hexamer (three letter code).

**AA** Both encoded amino acid within the hexamer

**Examples**

hex

---

prepareReferenceFasta *Imports Fasta file from filepath.*

---

**Description**

This function imports Fasta file of the reference genome into R environment as DNASTringset.

**Usage**

prepareReferenceFasta(filepath)

**Arguments**

filepath            R conform filepath to the fasta file of the reference genome to use.

**Value**

Creates new DNStringSet from the object stated by the entered filepath.

**Examples**

```
## Loading exemplary DNStringSet
filepath <- system.file("extdata", "fastaEx.fa", package="Biostrings")
referenceDnaStringSet <- prepareReferenceFasta(filepath)
```

---

referenceDnaStringSet *Small DNStringset as exemplary reference genome sequence*

---

**Description**

Small DNStringset as exemplary reference genome sequence

**Usage**

```
referenceDnaStringSet
```

**Format**

DNStringset

**width** Length of feature sequence

**seq** Sequence of the feature

**names** Name of the feature

**Examples**

```
referenceDnaStringSet
```

---

startVarConApp *Start GUI of VarCon.*

---

**Description**

Start graphical user interface for the VarCon application.

**Usage**

```
startVarConApp()
```

**Value**

Shiny app

**Examples**

```
## Not run:
startVarConApp()

## End(Not run)
```

---

transCoord

*Small table as exemplary transcript table with exon coordinates*


---

**Description**

Small table as exemplary transcript table with exon coordinates.

**Usage**

```
transCoord
```

**Format**

data frame

**Gene.stable.ID** Ensembl gene ID

**Transcript.stable.ID** Ensembl Transcript ID

**Strand** Strand of the feature

**Exon.region.start..bp.** Smalles coordinate of the exon end coordinates of a specific exon

**Exon.region.end..bp.** Largest coordinate of the exon end coordinates of a specific exon

**cDNA.coding.start** Start of the coding sequence

**cDNA.coding.end** End of the coding sequence

**CDS.start** Covered coding nucleotides start

**CDS.end** Covered coding nucleotides end

**Exon.rank.in.transcript** Rank of the exon within the respective transcript

**Exon.stable.ID** Ensembl exon ID

**Chromosome.scaffold.name** Name of the chromosome

**Examples**

```
transCoord
```

# Index

## \* **datasets**

- gene2transcript, [3](#)
- hbg, [6](#)
- hex, [7](#)
- referenceDnaStringSet, [8](#)
- transCoord, [9](#)

- calculateHZEIperNT, [2](#)
- calculateMaxEntScanScore, [3](#)

- gene2transcript, [3](#)
- generateHEXplorerPlot, [4](#)
- getMaxEntInfo, [5](#)
- getSeqInfoFromVariation, [5](#)

- hbg, [6](#)
- hex, [7](#)

- prepareReferenceFasta, [7](#)

- referenceDnaStringSet, [8](#)

- startVarConApp, [8](#)

- transCoord, [9](#)