

Package: DCODE (via r-universe)

September 16, 2024

Type Package

Title List Linear n-Peptide Constraints for Overlapping Protein Regions

Version 1.0

Date 2016-03-10

Author Sophie Lebre

Maintainer Sophie Lebre <sophie.lebre@umontpellier.fr>

Depends R (>= 3.1.0), seqinr

Description Traversal graph algorithm for listing linear n-peptide constraints for overlapping protein regions. (Lebre and Gascuel, The combinatorics of overlapping genes, freely available from arXiv at : <http://arxiv.org/abs/1602.04971>).

License GPL (>= 2)

LazyLoad yes

LazyData no

NeedsCompilation no

Date/Publication 2016-03-10 17:55:18

Repository <https://sophie-lebre.r-universe.dev>

RemoteUrl <https://github.com/cran/DCODE>

RemoteRef HEAD

RemoteSha 0a53b7b9dae0a1b938760b1b27c089e407c15262

Contents

DCODE-package	2
build_data	3
DCODE-internal	4
getConstraint	4

Index	6
--------------	----------

Description

Traversal graph algorithm for listing linear n-peptide constraints for overlapping protein regions.

Author(s)

S. Lebre

Maintainer: S. Lebre <sophie.lebre@umontpellier.fr>

References

Lebre and Gascuel, The combinatorics of overlapping genes (freely available from arXiv at: <http://arxiv.org/abs/1602.04971>).

Examples

```
## Not run:
# 1) Build peptideList, used by function getConstraint.
mydata <- build_data()
aaList <- mydata$aaList
geneticCode <- mydata$geneticCode
peptideList <- mydata$peptideList

# 2) Ask for the constraint induced on a chosen peptide in a chosen overlapping frame

## amino acid constraints
getConstraint("C", frame = -2, code=geneticCode, pepList=peptideList)
getConstraint("D", frame = 1, code=geneticCode, pepList=peptideList)

## 3-peptide constraints
getConstraint("CWC", frame = -2, code=geneticCode, pepList=peptideList)

## 5-peptide constraints
getConstraint("CWCCC", frame = -2, code=geneticCode, pepList=peptideList)

## End(Not run)
```

build_data	<i>Function to build the elements aaList, geneticCode, peptideList, used by function getConstraint.</i>
------------	---

Description

To be run before the first use of function [getConstraint](#). This function generates the variables aaList, geneticCode and peptideList, a list of all the peptides (up to length 5), used by function [getConstraint](#).

Usage

```
build_data()
```

Value

aaList	Vector of all amino acids.
geneticCode	A list describing the standard genetic code.
peptideList	A list of all the peptides (up to length 5).

Author(s)

Sophie Lebre <sophie.lebre@umontpellier.fr>

References

Lebre and Gascuel, The combinatorics of overlapping genes (freely available from arXiv at: <http://arxiv.org/abs/1602.04971>).

See Also

[getConstraint](#)

Examples

```
## Not run:
# 1) Build peptideList, used by function getConstraint.
mydata <- build_data()
aaList <- mydata$aaList
geneticCode <- mydata$geneticCode
peptideList <- mydata$peptideList

# 2) Ask for the constraint induced on a chosen peptide in a chosen overlapping frame

## amino acid constraints
getConstraint("C", frame = -2, code=geneticCode, pepList=peptideList)
getConstraint("D", frame = 1, code=geneticCode, pepList=peptideList)
```

```

## 3-peptide constraints
getConstraint("CWC",frame = -2, code=geneticCode, pepList=peptideList)

## 5-peptide constraints
getConstraint("CWCCC",frame = -2, code=geneticCode, pepList=peptideList)

## End(Not run)

```

DCODE-internal *Internal DCODE Functions*

Description

Internal DCODE functions

Details

These are not to be called by the user (or in some cases are just waiting for proper documentation to be written).

getConstraint *Function giving the linear constraint associated with the chosen peptide and overlapping frame.*

Description

This function prints the linear constraint to which the chosen peptide in the reference frame pep is subject when considering gene overlap with frameshift frame.

Usage

```
getConstraint(pep, frame, code, pepList)
```

Arguments

pep	A chosen n-peptide in the reference frame (with length $n \leq 5$ if pepList is generated by function build_data).
frame	The frameshift chosen in $\{-2, -1, 0, 1, 2\}$.
code	The code used to translate codons into amino acids (which can be generated by function build_data).
pepList	A list of all the peptide of length 1, 2, 3, ...required by the graph traversal algorithm (which can be generated by function build_data).

Author(s)

Sophie Lebre <sophie.lebre@umontpellier.fr>

References

Lebre and Gascuel, The combinatorics of overlapping genes (freely available from arXiv at: <http://arxiv.org/abs/1602.04971>).

See Also

[build_data](#)

Examples

```
## Not run:
# 1) Build peptideList, used by function getConstraint.
mydata <- build_data()
aaList <- mydata$aaList
geneticCode <- mydata$geneticCode
peptideList <- mydata$peptideList

# 2) Ask for the constraint induced on a chosen peptide in a chosen overlapping frame

## amino acid constraints
getConstraint("C", frame = -2, code=geneticCode, pepList=peptideList)
getConstraint("D", frame = 1, code=geneticCode, pepList=peptideList)

## 3-peptide constraints
getConstraint("CWC", frame = -2, code=geneticCode, pepList=peptideList)

## 5-peptide constraints
getConstraint("CWCCC", frame = -2, code=geneticCode, pepList=peptideList)

## End(Not run)
```

Index

- * **amino acid composition**
 - DCODE-package, 2
- * **codon usage**
 - DCODE-package, 2
- * **double-coding sequences**
 - DCODE-package, 2
- * **genetic code**
 - DCODE-package, 2
- * **linear and logical constraints**
 - DCODE-package, 2
- * **overlapping gene**
 - DCODE-package, 2
- * **print**
 - getConstraint, 4
- * **stop codons**
 - DCODE-package, 2

build_data, 3, 4, 5

compRev (DCODE-internal), 4

DCODE (DCODE-package), 2

DCODE-internal, 4

DCODE-package, 2

deleteStops (DCODE-internal), 4

getAllSeqs (DCODE-internal), 4

getConstraint, 3, 4

getConstraint_rec1 (DCODE-internal), 4

getConstraint_rec2 (DCODE-internal), 4

pepList (DCODE-internal), 4

posLecturef1 (DCODE-internal), 4

posLecturef2 (DCODE-internal), 4

printConstraint (DCODE-internal), 4

stopCodons (DCODE-internal), 4

toPep (DCODE-internal), 4