Package 'Rchemcpp'

April 16, 2019

Type Package

Title Similarity measures for chemical compounds

Version 2.20.1

Date 2015-06-23

Author Michael Mahr, Guenter Klambauer

Maintainer Guenter Klambauer <klambauer@bioinf.jku.at>

Description The Rchemcpp package implements the marginalized graph kernel and extensions, Tanimoto kernels, graph kernels, pharmacophore and 3D kernels suggested for measuring the similarity of molecules.

biocViews ImmunoOncology, Bioinformatics, CellBasedAssays, Clustering, DataImport, Infrastructure, MicrotitrePlateAssay, Proteomics, Software, Visualization

License GPL (>= 2.1)

URL http://www.bioinf.jku.at/software/Rchemcpp

Depends R (>= 2.15.0)

Imports Rcpp (>= 0.11.1), methods, ChemmineR

Suggests apcluster, kernlab

LinkingTo Rcpp

SystemRequirements GNU make

RcppModules Rmolecule, Rmoleculeset, Relements, spectrumhelper, spectrum3Dhelper, subtreehelper

Collate 'getMoleculeNamesFromSDF.R' 'sd2gram3Dpharma.R' 'sd2gram3Dspectrum.R' 'sd2gram.R' 'sd2gramSpectrum.R' 'sd2gramSubtree.R' 'utility.R' 'zzz.R' 'roxygen.R' 'methods.R'

git_url https://git.bioconductor.org/packages/Rchemcpp

git_branch RELEASE_3_8

git_last_commit 689ea65

git_last_commit_date 2019-01-04

Date/Publication 2019-04-15

R topics documented:

Rchemcpp-package 2
createRMolecule 3
getMoleculeNamesFromSDF 4
getMoleculePropertyFromSDF 5
Rcpp_Rmolecule-class 5
Rcpp_Rmoleculeset-class 6
readRmoleculeset 8
sd2gram 8
sd2gram3Dpharma 10
sd2gram3Dspectrum 11
sd2gramSpectrum 13
sd2gramSubtree 14

Index **17**

`Rchemcpp-package` *Rchemcpp provides tools for comparing chemical compounds*

Description

Compares sets of chemical compounds given as SD/SDF/MOL- or KCF-files and returns pairwise similarities as a matrix (gram matrix). It uses the compiled-in c++ library "chemcpp" to emulate the five chemcpp tools "sd2gram", "sd2gram3Dspectrum", "sd2gramSubtree", "sd2gram3Dpharma" and "sd2gramSpectrum". The tools are made accessible as R functions.

Details

Package:	Rchemcpp
Type:	Package
Version:	1.1.1
Date:	2013-07-03
License:	GPL2.1

Author(s)

Michael Mahr and Guenter Klambauer

References

(Kashima, 2004) – H. Kashima, K. Tsuda, and A. Inokuchi. Kernels for graphs. In B. Schoelkopf, K. Tsuda, and J.P. Vert, editors, Kernel Methods in Computational Biology, pages 155-170. MIT Press, 2004.

(Mahe, 2005) – P. Mahe, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Graph kernels for molecular structure- activity relationship analysis with support vector machines. J Chem Inf Model, 45(4):939-51, 2005.

(Ralaivola, 2005) – L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. G raph kernels for chemical informatics. Neural Netw., 18(8):1093-1110, Sep 2005.

(Gaertner, 2003) – T. Gaertner, P. Flach, and S. Wrobel. On graph kernels: hardness results and efficient alternatives. In B. Schoelkopf and M. Warmuth, editors, Proceedings of the Sixteenth Annual Conference on Computational Learning Theory and the Seventh Annual Workshop on Kernel Machines, volume 2777 of Lecture Notes in Computer Science, pages 129-143, Heidelberg, 2003.

(Mahe, 2006a) – P. Mahe and J.-P. Vert. Graph kernels based on tree patterns for molecules. Technical Report, HAL:ccsd-00095488, Ecoles des Mines de Paris, September 2006.

(Mahe, 2006b) – P. Mahe, L. Ralaivola, V. Stoven, and J.-P. Vert. The pharmacophore kernel for virtual screening with support vector machines. Technical Report, HAL:ccsd-00020066, Ecole des Mines de Paris, March 2006.

(Leslie, 2002) – C. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: a string kernel for SVM protein clas- sification. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Kevin Lauerdale, and Teri E. Klein, editors, Proceedings of the Pacific Symposium on Biocomputing 2002, pages 564-575. World Scientific, 2002.

(Ramon, 2003) – J. Ramon and T. Gaertner. Expressivity versus efficiency of graph kernels. In T. Washio and L. De Raedt, editors, Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences, pages 65-74, 2003.

See Also

sd2gram sd2gram3Dpharma sd2gramSpectrum sd2gram3Dspectrum sd2gramSubtree

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")

sdf <- list.files(sdfolder,full.names=TRUE,pattern="small")
K1 <- sd2gram(sdf)
K2 <- sd2gramSpectrum(sdf)
K3 <- sd2gramSubtree(sdf)


sdf_tiny <- list.files(sdfolder,full.names=TRUE,pattern="tiny")
K3 <- sd2gram3Dspectrum(sdf_tiny)
K4 <- sd2gram3Dpharma(sdf_tiny)
```

`createRMolecule` *createRMolecule*

Description

Creates an \"Rmolecule\" from an atom-vector and a bond-matrix

Usage

```
createRMolecule(atoms, bonds)
```

Arguments

atoms	A vector containing the symbol names of all atoms in the molecule
bonds	A matrix with the same number of rows and columns as the atoms-vector containing the type of bonds between the atoms

Value

an instance of "molecule"

Author(s)

Michael Mahr <rchemcpp@bioinf.jku.at>

Examples

```
m <- createRMolecule(c("C","C"),matrix(c(0,3,3,0),nrow=2))
```

---

getMoleculeNamesFromSDF

getMoleculeNamesFromSDF - a helper function

Description

This function helps to extract a certain property from an SDF file. Usually the molecule class, like "active/non-active" or a property of the molecule, like "biological activity", is also stored in the SDF file. These values often serve as targets for a prediction task. This function is a small wrapper that extracts the information.

Usage

```
getMoleculeNamesFromSDF(sdfile)
```

Arguments

sdfile	A character containing the name of the SDF file.

Value

A character vector with one name per molecule.

Author(s)

Guenter Klambauer <rchemcpp@bioinf.jku.at>

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="small")
moleculeNames <- getMoleculeNamesFromSDF(sdf)
```

getMoleculePropertyFromSDF *getMoleculePropertyFromSDF - a helper function*

Description

This function helps to extract a certain property from an SDF file. Usually the molecule class, like "active/non-active" or a property of the molecule, like "biological activity", is also stored in the SDF file. These values often serve as targets for a prediction task. This function is a small wrapper that extracts the information.

Usage

```
getMoleculePropertyFromSDF(sdfile,property)
```

Arguments

sdfile	A character containing the name of the SDF file.
property	The name of the slot in the SDF.

Value

A character vector with one value per molecule.

Author(s)

Guenter Klambauer <rchemcpp@bioinf.jku.at>

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="tiny")
activity <- getMoleculePropertyFromSDF(sdf,"Activity")
```

Rcpp_Rmolecule-class *Class* "Rcpp_Rmolecule"

Description

This class is a Rcpp modules wrapper for the chemcpp c++ class "Molecule". It allows creating molecules from scratch or manipulating existing ones. Currently it exposes only a small fraction of functionality of the base class. Please note that only a part of the original chemcpp class "Molecule" is exposed until now.

Extends

chemcpp c++ class "Molecule"

Methods

`writeSD(...)`: Write molecule to sd file

`linkAtoms(...)`: Create a bond between two atoms; Atom index is zero-based

`addAtom(...)`: Add an atom by specifying its character symbol

`listStringDescriptors(...)`: Return a vector of all string descriptors of the molecule

`getStringDescriptorValue(...)`: Return the value of one string descriptor

`getStringDescriptorUnit(...)`: Return the unit of one string descriptor

`getStringDescriptorComment(...)`: Return the comment of one string descriptor

`setStringDescriptor(...)`: Create or replace a string descriptor of the molecule by specifying the name, value, unit and comment

`deleteStringDescriptor(...)`: Delete one string descriptor from the molecule

Author(s)

Michael Mahr; base class written by Jean-Luc Perret and Pierre Mahe

Examples

```
set = new (Rmoleculeset)
mol = new (Rmolecule)
mol$addAtom("H")
set$addMoleculeCopy(mol)
```

Rcpp_Rmoleculeset-class

Class "Rcpp_Rmoleculeset"

Description

This class is a Rcpp modules wrapper for the chemcpp c++ class "MoleculeSet". It allows reading molecule-files and computing simple comparison-matrices. When calling the function "setComparisonSet" however, the argument object is copied (instead of storing a reference). Please note that only a part of the original chemcpp class "MoleculeSet" is exposed until now.

Extends

chemcpp c++ class "MoleculeSet"

Methods

`writeSelfKernelList(...)`: Write self kernel list

`writeGramMatrix(...)`: Write the gram matrix to a file, if one has been computed

`setMorganLabels(...)`: Set Morgan labels

`setMorganChargesLabels(...)`: Set Morgan Charges label

`setKashimaKernelParam(...)`: Set Kashima kernel parameter

`setComparisonSetSelf(...)`: Set the comparison set to be the set itself; NOTE: this is the preferred way to compare a set with itself, because faster implementations are used for comparison this way

`setComparisonSetCopy(...)`: Set the comparison set to be a different set of molecules; NOTE: this function copies the object specified as argument

`readPartialCharges(...)`: Add partial charges from file

`numMolecules(...)`: Returns the number of contained molecules

`normalizeTanimoto_raw(...)`: Normalize Tanimoto kernel

`normalizeTanimotoMinMax(...)`: Normalize Tanimoto min-max kernel

`normalizeTanimoto(...)`: Normalize Tanimoto kernel

`normalizeGram_raw(...)`: Normalize the gram-matrix

`normalizeGram(...)`: Normalize the gram-matrix

`noTottersTransform(...)`: Transform to avoid totters

`initializeSelfKernel(...)`: Initialize the self-kernel

`initializeGram(...)`: Initialize the gram matrix

`hideHydrogens(...)`: Hide hydrogen atoms in all contained molecules

`gramCompute3D(...)`: Compute 3D gram

`gramCompute(...)`: Compute gram

`getGramNormal(...)`: Return the normalized gram matrix, if one has been computed

`getGram(...)`: Return the gram matrix, if one has been computed

`getComparisonSet(...)`: Return A POINTER to the comparison set contained in the set; NOTE: this pointer expires when the set is destroyed or a different comparison set is set

`bondsListing(...)`: Return a list of all bonds which are present in the set

`atomsLabelsListing(...)`: Return a list of all atom symbols which are present in the set

`addSD2(...)`: Load a file containing molecules

`addSD(...)`: Load a file containing molecules

`addMoleculeCopy(...)`: Add a copy of a molecule object to the set

`addKCF2(...)`: Load a file containing molecules

`addKCF(...)`: Load a file containing molecules

`getMolByIndex(...)`: Return A POINTER to the molecule specified by the Index (zero-based); NOTE: this pointer expires when the set or the molecules in the set are destroyed

`length(...)`: Return the number of molecules in the molecule set

Author(s)

Michael Mahr; base class written by Jean-Luc Perret and Pierre Mahe

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="small")
set <- new(Rmoleculeset)
set$addSD(sdf, FALSE)
```

`readRmoleculeset` *Generating an Rmoleculeset from an SDF file*

Description

This function uses the ChemmineR package to read an SDF file and converts it into an `Rmoleculeset` that can be used as input for the kernel functions `sd2gram`, `sd2gramSpectrum`, ..., `sd2gram3Dpharma`.

Usage

```
readRmoleculeset(sdfFileName, detectArom = TRUE,
  bound = Inf, type = 2)
```

Arguments

`sdfFileName`	The name of the SDF file containing the molecules.
`detectArom`	If the molecules in the SDF file have no annotated aromatic bonds, the ChemmineR function `rings` is used for detecting aromaticity. (Default = TRUE).
`bound`	Detection of aromaticity can be time consuming if the molecules are large. Detection is only done if the number of atoms is below the given number. (Default = Inf).
`type`	Experimental parameter to switch between to types of the function. (Default = 2).

Value

An instance of Rmoleculeset.

Author(s)

Guenter Klambauer <rchemcpp@bioinf.jku.at>

`sd2gram` *sd2gram - Similarity of molecules by the marginalized kernel and proposed extensions.*

Description

This tools compute the marginalized kernel (*Kashima, 2004*) and its proposed extensions (*Mahe, 2005).*

Usage

```
sd2gram(sdf, sdf2, stopP = 0.1, filterTottering = FALSE,
  converg = as.integer(1000), atomKernelMatrix = "",
  flagRemoveH = FALSE, morganOrder = as.integer(0),
  silentMode = FALSE, returnNormalized = TRUE,
  detectArom = FALSE)
```

Arguments

sdf	File containing the molecules. Must be in MDL file format (MOL and SDF files). For more information on the file format see http://en.wikipedia.org/wiki/Chemical_table_file.
sdf2	A second file containing molecules. Must also be in SDF format. If specified the molecules of the first file will be compared with the molecules of this second file. Default = "missing".
stopP	The probability that a random walk stops. The higher the value the more weigth is put on shorter walks. Default = 0.1.
filterTottering	A logical specifying whether tottering paths should be removed. Default = FALSE.
converg	A numeric value specifying when convergence is reached. The algorithm stops when the kernel value does not change by more than 1/c, where c is the value specified by the converg option. Default = 1000.
atomKernelMatrix	A string that sets the similarity measure between atoms that should be used. Default = "missing".
flagRemoveH	A logical that indicates whether H-atoms should be removed or not. Default = FALSE.
morganOrder	The order of the DeMorgan indices to be used. If set to zero, no DeMorgan indices are used. The higher the order the more types of atoms exist and consequently the more dissimilar will be the molecules. Default = 0.
silentMode	Whether or not the program should print progress reports to the standart output. Default = FALSE.
returnNormalized	A logical specifying whether a normalized kernel matrix should be returned. Default = TRUE.
detectArom	Whether aromatic rings should be detected and aromatic bonds should a special bond type. If large molecules are in the data set the detection of aromatic rings can be very time-consuming. (Default = FALSE).

Value

A numeric matrix containing the similarity values between the molecules.

Author(s)

Michael Mahr <rchemcpp@bioinf.jku.at> c++ function written by Jean-Luc Perret and Pierre Mahe.

References

(Kashima, 2004) – H. Kashima, K. Tsuda, and A. Inokuchi. Kernels for graphs. In B. Schoelkopf, K. Tsuda, and J.P. Vert, editors, Kernel Methods in Computational Biology, pages 155-170. MIT Press, 2004.

(Mahe, 2005) – P. Mahe, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Graph kernels for molecular structure- activity relationship analysis with support vector machines. J Chem Inf Model, 45(4):939-51, 2005.

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="small")
K <- sd2gram(sdf)
```

`sd2gram3Dpharma` *sd2gram3Dpharma - Similarity of molecules by the exact pharmacophore kernel.*

Description

This tool implements the (exact version of) pharmacophore kernel for 3D structures of molecules (*Mahe, 2006*).

Usage

```
sd2gram3Dpharma(sdf, sdf2, chargesFileName = "",
  chargesFileName2 = "",
  edgeKernelType = c("RBF", "triangular"),
  edgeKernelParameter = 1, atomKernelMatrix = "",
  flagRemoveH = FALSE, morganOrder = as.integer(0),
  morganChargesThreshold = 0, silentMode = FALSE,
  returnNormalized = TRUE, detectArom = FALSE)
```

Arguments

`sdf`	File containing the molecules. Must be in MDL file format (MOL and SDF files). For more information on the file format see http://en.wikipedia.org/wiki/Chemical_table_file.
`sdf2`	A second file containing molecules. Must also be in SDF format. If specified the molecules of the first file will be compared with the molecules of this second file. Default = "missing".
`chargesFileName`	A character with the name of the file containing the atom charges. Default = missing.
`chargesFileName2`	A character with the name of the file containing the atom charges. Default = missing.
`edgeKernelType`	Options to specify the kernel function comparing distances between atoms. Choices are "RBF" or "triangular". Default = "RBF".
`edgeKernelParameter`	Specifies the parameter associated to these kernels. Either the bandwith of the RBF kernel or the cut-off of the triangular kernel. Default = 1.
`atomKernelMatrix`	A string that sets the similarity measure between atoms that should be used. Default = "missing".
`flagRemoveH`	A logical that indicates whether H-atoms should be removed or not. Default = FALSE.
`morganOrder`	The order of the DeMorgan Indices to be used. If set to zero no DeMorgan Indices are used. The higher the order the more types of atoms exist and consequently the more dissimilar will be the molecules. Default = 0.

`morganChargesThreshold`
: specifies a threshold above which partial Morgan charges are considered as positive/negative. By default this threshold is zero, and every positive (resp. negative) partial charge is seen as a positive (resp. negative) charge. However, it might be interesting to consider a threshold of 0.2 for example, in which case only partial charges greater than 0.2 (resp. smaller than -0.2) would be seen as positive (resp. negative). Default = 0.

`silentMode`
: Whether or not the program should print progress reports to the standart output. Default = TRUE.

`returnNormalized`
: A logical specifying whether a normalized kernel matrix should be returned. Default = TRUE.

`detectArom`
: Whether aromatic rings should be detected and aromatic bonds should a special bond type. If large molecules are in the data set the detection of aromatic rings can be very time-consuming. (Default = FALSE).

Value

A numeric matrix containing the similarity values between the molecules.

Author(s)

Michael Mahr <rchemcpp@bioinf.jku.at> c++ function written by Jean-Luc Perret and Pierre Mahe

References

(Mahe, 2006) – P. Mahe, L. Ralaivola, V. Stoven, and J.-P. Vert. The pharmacophore kernel for virtual screening with support vector machines. Technical Report, HAL:ccsd-00020066, Ecole des Mines de Paris, March 2006.

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="tiny")
K <- sd2gram3Dpharma(sdf)
```

`sd2gram3Dspectrum` *sd2gram3Dspectrum - Similarity of molecules by fast approximations of the pharmacophore kernel*

Description

This tool implements the six discrete approximations of the pharmacophore kernel presented in "The pharmacophore kernel for virtual screening with support vector machines" (*Mahe, 2006*).

Usage

```
sd2gram3Dspectrum(sdf, sdf2, chargesFileName = "",
  chargesFileName2 = "",
 kernelType = c("3Pspectrum", "3Pbinary", "3Ptanimoto", "2Pspectrum", "2Pbinary", "2Ptanimoto")
  depthMax = as.integer(3), nBins = as.integer(20),
  distMin = 0, distMax = 20, flagRemoveH = FALSE,
  morganOrder = as.integer(0), chargesThreshold = 0,
  silentMode = FALSE, returnNormalized = TRUE,
  detectArom = FALSE)
```

Arguments

sdf	File containing the molecules. Must be in MDL file format (MOL and SDF files). For more information on the file format see http://en.wikipedia.org/wiki/Chemical_table_file.
sdf2	A second file containing molecules. Must also be in SDF format. If specified the molecules of the first file will be compared with the molecules of this second file. Default = "missing".
chargesFileName	A character with the name of the file containing the atom charges. Default = missing.
chargesFileName2	A character with the name of the file containing the atom charges. Default = missing.
kernelType	Type of kernel to be used. Possible choices are 3-points spectrum kernel ("3Pspectrum"), 3-points binary kernel ("3Pbinary"), 3-points Tanimoto kernel ("3Ptanimoto"), 2-points spectrum kernel ("2Pspectrum"), 2-points binary kernel ("2Pbinary"), 2-points Tanimoto kernel ("2Ptanimoto"). Default = "3Pspectrum".
depthMax	The maximal length of the molecular fragments. Default = 3.
nBins	number of bins used to discretize the inter-atomic lengths. An adequate value for the number of bins is between 20 and 30. Default = 20.
distMin	minimum distance for inter-atomic distance range. Default = 0.
distMax	maximum distance in angstrom for inter-atomic distance range. Default = 20.
chargesThreshold	specifies a threshold above which partial charges are considered as positive/negative. By default this threshold is zero, and every positive (resp. negative) partial charge is seen as a positive (resp. negative) charge. However, it might be interesting to consider a threshold of 0.2 for example, in which case only partial charges greater than 0.2 (resp. smaller than -0.2) would be seen as positive (resp. negative). Default = 0.
flagRemoveH	A logical that indicates whether H-atoms should be removed or not. Default = FALSE.
morganOrder	The order of the DeMorgan Indices to be used. If set to zero, no DeMorgan indices are used. The higher the order the more different types of atoms exist and consequently the more dissimilar will be the molecules.
silentMode	Whether or not the program should print progress reports to the standart output. Default = FAlSE.
returnNormalized	A logical specifying whether a normalized kernel matrix should be returned. Default = TRUE.

detectArom — Whether aromatic rings should be detected and aromatic bonds should a special bond type. If large molecules are in the data set the detection of aromatic rings can be very time-consuming. (Default = FALSE).

Value

A numeric matrix containing the similarity values between the molecules.

Author(s)

Michael Mahr <rchemcpp@bioinf.jku.at> c++ function written by Jean-Luc Perret and Pierre Mahe

References

(Mahe, 2006) – P. Mahe, L. Ralaivola, V. Stoven, and J.-P. Vert. The pharmacophore kernel for virtual screening with support vector machines. Technical Report, HAL:ccsd-00020066, Ecole des Mines de Paris, March 2006.

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="tiny")
K <- sd2gram3Dspectrum(sdf)
```

---

sd2gramSpectrum — *sd2gramSpectrum - Similarity of molecules by walk-based graph kernels*

Description

This function computes several walk-based graph kernel functions based on finite length walks and a fast implementation for input SDF file(s).

Usage

```
sd2gramSpectrum(sdf, sdf2,
 kernelType = c("spectrum", "tanimoto", "minmaxTanimoto", "marginalized", "lambda"),
  margKernelEndProbability = 0.1, lambdaKernelLambda = 1,
  depthMax = as.integer(3), onlyDepthMax = FALSE,
  flagRemoveH = FALSE, morganOrder = as.integer(0),
  silentMode = FALSE, returnNormalized = TRUE,
  detectArom = FALSE)
```

Arguments

sdf — File containing the molecules. Must be in MDL file format (MOL and SDF files). For more information on the file format see http://en.wikipedia.org/wiki/Chemical_table_file.

sdf2 — A second file containing molecules. Must also be in SDF. If specified the molecules of the first file will be compared with the molecules of this second file. Default = "missing".

`kernelType` Type of kernel to be used. Options are "spectrum (Spectrum kernel) , "tanimoto" (Tanimoto kernel), "minmaxTanimoto" (MinMax Tanimoto kernel), "marginalized (Marginalized kernel approximation) and "lambda" (LambdaK kernel). See vignette for details. Default = "spectrum".

`margKernelEndProbability` The ending probability for the marginalized kernel. Default = 0.1.

`lambdaKernelLambda` The lambda parameter of the LambdaK kernel. Default = 1.0.

`depthMax` The maximal length of the molecular fragments. Default = 3.

`onlyDepthMax` Whether fragments up to the given length should be used or only fragments of the given length. Default = FALSE.

`flagRemoveH` A logical that indicates whether H-atoms should be removed or not. Default = FALSE

`morganOrder` The order of the DeMorgan indices to be used. If set to zero no DeMorgan indices are used. The higher the order the more different types of atoms exist and consequently the more dissimilar will be the molecules. Default = 0.

`silentMode` Whether or not the program should print progress reports to the standart output. Default = FALSE.

`returnNormalized` A logical specifying whether a normalized kernel matrix should be returned. Default = TRUE.

`detectArom` Whether aromatic rings should be detected and aromatic bonds should a special bond type. If large molecules are in the data set the detection of aromatic rings can be very time-consuming. (Default = FALSE).

Value

A numeric matrix containing the similarity values between the molecules.

Author(s)

Michael Mahr <rchemcpp@bioinf.jku.at> c++ function written by Jean-Luc Perret and Pierre Mahe

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="tiny")
K <- sd2gramSpectrum(sdf)
```

---

sd2gramSubtree *sd2gramSubtree - Similarity of molecules by several graph kernels based on the count of common subtrees*

Description

This tools computes several graph kernels based on the detection of common subtrees: the so-called tree-pattern graph kernels, originally introduced in (*Ramon, 2003*), and revisited in (*Mahe, 2006*).

Usage

```
sd2gramSubtree(sdf, sdf2,
  kernelType = c("sizebased", "branchingbased"),
  branchKernelUntilN = FALSE, lambda = 1,
  depthMax = as.integer(3), flagRemoveH = FALSE,
  filterTottering = FALSE, morganOrder = as.integer(0),
  silentMode = FALSE, returnNormalized = TRUE,
  detectArom = FALSE)
```

Arguments

sdf	File containing the molecules. Must be in MDL file format (MOL and SDF files). For more information on the file format see http://en.wikipedia.org/wiki/Chemical_table_file.
sdf2	A second file containing molecules. Must also be in SDF. If specified the molecules of the first file will be compared with the molecules of this second file. Default = "missing".
kernelType	Determines whether subtrees of the molecule are penalized size-based or branching-based. Default = "sizebased".
branchKernelUntilN	Logical whether tree patterns of until N should be considered. Default = FALSE.
lambda	Weighted contribution of tree-patterns depending on their sizes Default = 1.
depthMax	tree-patterns of depth. Default = 3.
flagRemoveH	A logical that indicates whether H-atoms should be removed or not. Default = FALSE.
filterTottering	A logical that indicates whether tottering walks should be removed. Default = FALSE.
morganOrder	The order of the DeMorgan indices to be used. If set to zero no DeMorgan indices are used. The higher the order the more different types of atoms exist and consequently the more dissimilar will be the molecules. Default = 0.
silentMode	Whether the program should print progress reports to the standart output. Default = FALSE.
returnNormalized	A logical specifying whether a normalized kernel matrix should be returned. Default = TRUE.
detectArom	Whether aromatic rings should be detected and aromatic bonds should a special bond type. If large molecules are in the data set the detection of aromatic rings can be very time-consuming. (Default = FALSE).

Value

A numeric matrix containing the similarity values between the molecules.

Author(s)

Michael Mahr <rchemcpp@bioinf.jku.at> c++ function written by Jean-Luc Perret and Pierre Mahe

References

(Mahe, 2006) – P. Mahe and J.-P. Vert. Graph kernels based on tree patterns for molecules. Technical Report, HAL:ccsd-00095488, Ecoles des Mines de Paris, September 2006. (Ramon, 2003) – J. Ramon and T. Gaertner. Expressivity versus efficiency of graph kernels. In T. Washio and L. De Raedt, editors, Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences, pages 65-74, 2003.

Examples

```
sdfolder <- system.file("extdata",package="Rchemcpp")
sdf <- list.files(sdfolder,full.names=TRUE,pattern="small")
K <- sd2gramSubtree(sdf)
```

Index

∗Topic **classes**
Rcpp_Rmolecule-class, 5
Rcpp_Rmoleculeset-class, 6
∗Topic **package**
Rchemcpp-package, 2

createRMolecule, 3

getMoleculeNamesFromSDF, 4
getMoleculePropertyFromSDF, 5

length,Rcpp_Rmoleculeset-method
*(*Rcpp_Rmoleculeset-class*)*, 6

Rchemcpp *(*Rchemcpp-package*)*, 2
Rchemcpp-package, 2
Rcpp_Rmolecule-class, 5
Rcpp_Rmoleculeset-class, 6
readRmoleculeset, 8
Rmolecule *(*Rcpp_Rmolecule-class*)*, 5
Rmoleculeset *(*Rcpp_Rmoleculeset-class*)*,
6

sd2gram, *3*, 8
sd2gram3Dpharma, *3*, 10
sd2gram3Dspectrum, *3*, 11
sd2gramSpectrum, *3*, 13
sd2gramSubtree, *3*, 14