

# Package ‘rgoslin’

October 18, 2022

**Type** Package

**Title** Lipid Shorthand Name Parsing and Normalization

**Version** 1.0.99

**Date** 2022-05-19

**Description** The R implementation for the Grammar of Succinct Lipid Nomenclature parses different short hand notation dialects for lipid names. It normalizes them to a standard name. It further provides calculated monoisotopic masses and sum formulas for each successfully parsed lipid name and supplements it with LIPID MAPS Category and Class information. Also, the structural level and further structural details about the head group, fatty acyls and functional groups are returned, where applicable.

**License** MIT + file LICENSE

**Imports** Rcpp (>= 1.0.3), dplyr

**LinkingTo** Rcpp

**Suggests** testthat (>= 2.1.0), BiocStyle, knitr, rmarkdown, kableExtra, BiocManager, stringr, ggplot2, tibble, lipidr

**RoxygenNote** 7.1.2

**Encoding** UTF-8

**VignetteBuilder** knitr

**biocViews** Software, Lipidomics, Metabolomics, Preprocessing, Normalization, MassSpectrometry

**BugReports** <https://github.com/lifs-tools/rgoslin/issues>

**URL** <https://github.com/lifs-tools/rgoslin>

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

**git\_branch** RELEASE\_3\_15

**git\_last\_commit** b7d7eff

**git\_last\_commit\_date** 2022-10-16

**Date/Publication** 2022-10-18

**Author** Nils Hoffmann [aut, cre] (<<https://orcid.org/0000-0002-6540-6875>>), Dominik Kopczynski [aut] (<<https://orcid.org/0000-0001-5885-4568>>)

**Maintainer** Nils Hoffmann <[nils.hoffmann@cebitec.uni-bielefeld.de](mailto:nils.hoffmann@cebitec.uni-bielefeld.de)>

**R topics documented:**

isValidLipidName . . . . .	2
listAvailableGrammars . . . . .	2
parseLipidNames . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

isValidLipidName	<i>Check lipid name.</i>
------------------	--------------------------

**Description**

isValidLipidName checks the provided lipid name against the built-in grammars. Will return FALSE if none of the parsers was able to parse the provided name successfully. Will stop execution via stop if non character input is detected.

**Usage**

```
isValidLipidName(lipidName)
```

**Arguments**

lipidName	The lipid name to check.
-----------	--------------------------

**Value**

TRUE if the lipidName could be parsed, FALSE otherwise.

**Examples**

```
isValidLipidName("PC 32:1")
isValidLipidName("PC(32:1)")
isValidLipidName("PCX(32:1)")
```

listAvailableGrammars	<i>Return the list of grammars supported by goslin.</i>
-----------------------	---

**Description**

listAvailableGrammars returns the list of grammars that the underlying cppgoslin library supports.

**Usage**

```
listAvailableGrammars()
```

**Value**

the list of grammars

**Examples**

```
listAvailableGrammars()
```

---

parseLipidNames	<i>Parse multiple lipid names and return a data frame with the results.</i>
-----------------	---

---

**Description**

parseLipidNames reads the provided lipid names vector and returns structural information as a data frame. Will return a cell with the "Grammar" column set to "NOT\_PARSEABLE" if none of the parsers was able to parse the provided name successfully. Will stop execution via stop if invalid non character input is detected or fatal errors are encountered during parsing.

**Usage**

```
parseLipidNames(lipidNames, grammar = NULL)
```

**Arguments**

lipidNames	The vector of lipid names to parse.
grammar	The grammar to use. One of "Goslin", "GoslinFragments", "SwissLipids", "LipidMaps", "HMDB", "FattyAcids". Call listAvailableGrammars() for a complete list of available grammars. If grammar is omitted or NULL is passed as a parameter, all available grammars / parsers will be tested. The first successful one will win. If all parsers fail, the "Messages" column in the returned data frame will contain the last parsers message.

**Value**

Data frame where each row reports the parsing result of each element in lipidNames.

**Examples**

```
parseLipidNames(c("PC 32:1", "LPC 34:1", "TG(18:1_18:0_16:1)"))  
parseLipidNames(c("Cer(d18:1(8Z)/24:0)", grammar = "LipidMaps"))
```

# Index

`isValidLipidName`, [2](#)

`listAvailableGrammars`, [2](#)

`parseLipidNames`, [3](#)