

Package: glyanno (via r-universe)

October 22, 2026

Title Hierarchical Glycan Annotation

Version 0.7.0

Description Hierarchical annotation of glycans based on molecule mass, composition, and structure. Starting from a molecule mass, glyanno calculates possible glycan compositions. Given a glycan composition, glyanno further deduces possible glycan structures. For glycan structures with generic monosaccharides (e.g., ``Hex'', ``HexNAc''), glyanno refines them into specific types (e.g., ``Glc'', ``Gal''). For structures lacking linkage information (e.g., ``Gal(??-?)GalNAc(??-)''), glyanno infers the most likely linkages (e.g., ``Gal(b1-3)GalNAc(a1-)'').

License MIT + file LICENSE

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL <https://glycoverse.github.io/glyanno/>

Imports checkmate, cli, dplyr, glue, glyparse (>= 0.5.3), glyrepr (>= 1.0.0), glymotif (>= 0.19.0), glydb (>= 0.4.0), igraph, purrr, rlang, tidyselect, tibble, httr2, lifecycle

VignetteBuilder knitr

Depends R (>= 4.1)

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev

Repository <https://glycoverse.r-universe.dev>

Date/Publication 2026-09-30 06:51:55 UTC

RemoteUrl <https://github.com/glycoverse/glyanno>

RemoteRef v0.7.0

RemoteSha 4f7c4fce7b4fedc2116fa484a431f8a1c16e6296

Contents

build_glyanno_cache	2
calculate_mz	3
comp_to_struc	5
enhance_comp	6
enhance_struc	7
enhance_struc_denovo	9
glyanno_mass_dict	10
mz_to_comp	11
ppm	12
struc_to_glytoucan	13
Index	14

build_glyanno_cache	<i>Manage the persistent annotation cache</i>
---------------------	---

Description

build_glyanno_cache() is the magic that makes glyanno fast. It precalculates database views and annotation indexes from the installed glydb. You will only need to call this function once on your computer. When glydb is updated, you'll also need to call this function to update the cache. But don't worry. We'll remind you then.

- build_glyanno_cache() reuses a valid cache unless force = TRUE.
- glyanno_cache_info() inspects it without building it, and
- clear_glyanno_cache() removes it and clears the session cache.

Usage

```
build_glyanno_cache(force = FALSE)

glyanno_cache_info()

clear_glyanno_cache()
```

Arguments

force Rebuild even when the cache is current.

Details

The cache lives in tools::R_user_dir("glyanno", "cache"). Set the glyanno.cache_dir option to use another directory. Only an explicit call to build_glyanno_cache() writes persistent data. Without a disk cache, annotation builds the required views in memory for the current session. On interactive attachment, a startup message reminds you to build a missing, outdated, or unreadable cache. No computation or questions block attachment. The cache is invalidated by a changed

glydb version, glyrepr or igraph version, R major/minor version, collation locale, or internal cache schema. Restart R after updating dependencies. Development data changes without a version bump require `force = TRUE`.

Cached data include concrete/generic compositions, intact/topological structures, confidence ranks, exact and compatible composition indexes, residue counts, species/type membership, floating-structure flags, masses for all built-in dictionaries, and local GlyTouCan accession keys. Structure vectors retain their parsed graphs. Arbitrary structure matching still uses glymotif at query time. Custom databases are never persisted.

A complete replacement is written before the previous cache is replaced; failed builds preserve the previous file. Only one completed cache is kept. Metadata inspection reads a small file header and checks the file size. The compressed annotation payload is loaded and validated on first use. Use `force = TRUE` to repair a damaged payload whose header is still valid.

Value

`build_glyanno_cache()` invisibly returns the cache file path. `glyanno_cache_info()` returns a list with path, status (one of "missing", "outdated", "unreadable", or "ready"), size_bytes, expected, and stored version metadata. `clear_glyanno_cache()` invisibly returns NULL.

Examples

```
glyanno_cache_info()
```

calculate_mz	<i>Calculate m/z values of glycans</i>
--------------	--

Description

This function calculates m/z values from glycans. Different adducts, mass types, and derivatives are supported. Custom mass dictionaries are also supported.

Usage

```
calculate_mz(glycans, charge = 1, adduct = "H+", mass_dict = NULL, safe = TRUE)
```

Arguments

glycans	Glycans to calculate m/z values from. Valid inputs include: - A <code>glyrepr::glycan_structure()</code> vector. - A <code>glyrepr::glycan_composition()</code> vector. - Byonic style composition strings (e.g. Hex(5)HexNAc(2)). - Simple style composition strings (e.g. H5N4F1S1). - Any structure strings supported by <code>glyparse::auto_parse()</code>.
charge	Charge to use. Can be 0, 1, 2, 3, -1, -2, -3, etc. 0 means neutral. Default is 1.
adduct	Adduct to use. Can be "H+", "K+", "Na+", "NH4+", "H-", "Cl-", "HCO3-". Default is "H+".

	• When charge is 0, adduct is ignored. • When charge is positive, adduct can only be "H+", "K+", "Na+", "NH4+". • When charge is negative, adduct can only be "H-", "Cl-", "HCO3-".
mass_dict	A named numeric vector of the mass of each monosaccharide residue. Default is <code>glyanno_mass_dict(deriv = "none", mass_type = "mono")</code> . If a custom mass dictionary is provided, please make sure the names of the vector are the same as the names in <code>glyanno_mass_dict()</code> .
safe	Whether to raise an error when unsupported monosaccharides or substituents are found. • If TRUE (default), an error will be raised. • If FALSE, a warning will be raised and m/z values for invalid glycans will be set to NA.

Value

A numeric vector of m/z values.

See Also

[glyanno_mass_dict\(\)](#)

Examples

```
library(glyrepr)

# Different input types
calculate_mz(glycan_composition(c(Gal = 1, GalNAc = 1)), charge = 0)
calculate_mz("Gal(1)GalNAc(1)", charge = 0)
calculate_mz(as_glycan_structure("Gal(b1-3)GalNAc(a1-)", charge = 0)
calculate_mz("Gal(b1-3)GalNAc(a1-", charge = 0)

# For common situation in MALDI-TOF MS
calculate_mz("Man(5)GlcNAc(2)", charge = 1, adduct = "Na+")

# For common situation in ESI MS
calculate_mz("Man(5)GlcNAc(2)", charge = 1, adduct = "H+")

# Calculate permethylated m/z values
calculate_mz("Man(5)GlcNAc(2)", charge = 0, mass_dict = glyanno_mass_dict(deriv = "permethyl"))

# Calculate average mass
calculate_mz("Man(5)GlcNAc(2)", charge = 0, mass_dict = glyanno_mass_dict(mass_type = "average"))

# Vectorization
calculate_mz(c("Man(5)GlcNAc(2)", "Gal(1)GalNAc(1)"), charge = 1, adduct = "Na+")
```

comp_to_struct

Convert glycan composition to glycan structure

Description

Given glycan compositions, this function matches them to all compatible glycan structures in the glydb database. Generic, concrete, and mixed residue identities are matched residue by residue.

Usage

```
comp_to_struct(
    comps,
    db = lifecycle::deprecated(),
    return_best = FALSE,
    glycan_type = NULL,
    species = NULL,
    structure_level = "intact",
    mono_type = "concrete",
    mono_range = NULL
)
```

Arguments

comps	Glycan compositions to match against. Can be either: - A glyrepr::glycan_composition() vector. - Byonic style composition strings (e.g. Hex(5)HexNAc(2)). - Simple style composition strings (e.g. H5N4F1S1).
db	[Deprecated] Glycan structures to match against. Can be a glyrepr::glycan_structure() vector or any structure strings supported by glyparse::auto_parse() . Structures with unresolved floating parts or substituents are excluded silently. Use the filtering arguments instead. This argument cannot be combined with them.
return_best	If TRUE, only return the highest confidence match for each composition. A custom db must have a confidence attribute. Default is FALSE.
glycan_type	Glycan type to select from glydb::glydb_structures() .
species	Species to select, matched without regard to letter case.
structure_level	Structure resolution, "intact" (default) or "topological".
mono_type	Monosaccharide resolution, "concrete" (default) or "generic".
mono_range	Named list of monosaccharide count ranges; see glydb::glydb_structures() .

Details

Filter the built-in database with glycan_type, species, structure_level, mono_type, and mono_range.

Value

If return_best=TRUE: A `glyrepr::glycan_structure()` vector with the same length as comps. Unmatched compositions are returned as NA. If return_best=FALSE: A tibble with the following columns:

- composition: The glycan compositions, as `glyrepr::glycan_composition()` vector.
- structure: The possible glycan structures, as `glyrepr::glycan_structure()` vector.
- confidence: The database confidence score for each structure, or NA when no score is available. Note that one glycan composition can have multiple rows in the result, corresponding to different possible glycan structures.

See Also

`glyparse::auto_parse()`

Examples

```
comp_to_struc("H5N2")
```

enhance_comp

Enhance glycan composition

Description

Given a generic or mixed glycan composition (e.g. Hex(5)HexNAc(2)), this function gives all possible compatible concrete glycan compositions (e.g. Man(5)GlcNAc(2)).

Usage

```
enhance_comp(
  comps,
  db = lifecycle::deprecated(),
  return_best = FALSE,
  glycan_type = NULL,
  species = NULL,
  mono_type = "concrete",
  mono_range = NULL
)
```

Arguments

comps	A <code>glyrepr::glycan_composition()</code> vector, or a character vector of glycan composition strings of Byonic or simple style (e.g. "Hex(5)HexNAc(2)", "H5N4F1S1"). Generic and mixed compositions are matched to compatible concrete compositions in db. Concrete compositions are returned as is.
-------	--

db	[Deprecated] A <code>glydb::glydb_compositions()</code> vector, or a character vector of glycan composition strings of Byonic or simple style (e.g. "Man(5)GlcNAc(2)", "H5N4F1S1"). All compositions in db must be concrete (e.g. Man(5)GlcNAc(2)). Use the filtering arguments instead. This argument cannot be combined with them.
return_best	Logical. If TRUE, only return the highest confidence match for each input composition. A custom db must have a confidence attribute. Defaults to FALSE.
glycan_type	Glycan type to select from <code>glydb::glydb_compositions()</code> .
species	Species to select, matched without regard to letter case.
mono_type	Monosaccharide resolution; only "concrete" is supported for enhancement.
mono_range	Named list of monosaccharide count ranges; see <code>glydb::glydb_compositions()</code> .

Details

Filter the built-in database with `glycan_type`, `species`, `mono_type`, and `mono_range`.

Value

If `return_best=TRUE`: A `glyrepr::glycan_composition()` vector with the same length as `comps`. Unmatched compositions are returned as NA. If `return_best=FALSE`: A tibble with the following columns:

- `raw`: The original compositions.
- `enhanced`: The enhanced compositions.
- `confidence`: The database confidence score for each enhanced composition, or NA when no score is available. Note that one raw composition can have different enhanced compositions as multiple rows in the result.

Examples

```
enhance_comp("Hex(5)HexNAc(2)")
```

enhance_struc	<i>Enhance glycan structure</i>
---------------	---------------------------------

Description

Given a glycan structure vector of any resolution level (see `glyrepr::get_structure_level()` for details), this function gives compatible structures with more specific residue identities or linkage information.

Usage

```
enhance_struc(
  strucs,
  db = lifecycle::deprecated(),
  return_best = FALSE,
  glycan_type = NULL,
  species = NULL,
  structure_level = "intact",
  mono_type = "concrete",
  mono_range = NULL
)
```

Arguments

strucs	A glyrepr::glycan_structure() vector, or a character vector of glycan structure strings supported by glyparse::auto_parse() . Inputs with unresolved floating parts or substituents are excluded with a warning.
db	[Deprecated] A glydb::glydb_structures() vector, or a character vector of glycan structure strings supported by glyparse::auto_parse() . Structures with unresolved floating parts or substituents are excluded silently. Use the filtering arguments instead. This argument cannot be combined with them.
return_best	Logical. If TRUE, only return the best matching structure (highest confidence) for each input structure. A custom db must have a confidence attribute. Default is FALSE.
glycan_type	Glycan type to select from glydb::glydb_structures() .
species	Species to select, matched without regard to letter case.
structure_level	Structure resolution, "intact" (default) or "topological".
mono_type	Monosaccharide resolution, "concrete" (default) or "generic".
mono_range	Named list of monosaccharide count ranges; see glydb::glydb_structures() .

Details

Input and database vectors may mix generic, concrete, and mixed residues, as well as topological, partial, and intact structures. Each database candidate is matched against each input independently.

Value

If return_best=TRUE: An unnamed [glyrepr::glycan_structure\(\)](#) vector with the same length as strucs. Unmatched structures are returned as NA. If return_best=FALSE: A tibble with the following columns:

- raw: The original glycan structures.
- enhanced: The enhanced glycan structures.
- confidence: The database confidence score for each enhanced structure, or NA when no score is available. Note that one raw glycan structure can have different enhanced structures as multiple rows in the result.

Examples

```
enhance_struc("Gal(??-?)GalNAc(??-)", glycan_type = "O-GalNAc",
  species = "Homo sapiens")
```

enhance_struc_denovo *Reconstruct topological N-glycan structures de novo*

Description

[Experimental]

Reconstructs generic topological N-glycans from their core and branches. Every non-missing input must have only generic residues and no linkage information. Inputs that cannot be reconstructed de novo are matched against a fallback database at "topological" level.

Topological de-novo enhancement preserves optional core fucose and bisecting GlcNAc residues. Complex N-glycan branches are matched against the internal branch data. For hybrid N-glycans, the all-Hex arm is assigned as mannose and the HexNAc-bearing arm uses branch matching. High-mannose candidates are constrained to core-aligned subtrees of the Glc3Man9 precursor.

Usage

```
enhance_struc_denovo(strucs, fallback_db = NULL)
```

Arguments

strucs	A glyrepr::glycan_structure() vector, or a character vector of glycan structure strings supported by glyparse::auto_parse() . Every non-missing element must be generic and topological. Missing values are allowed and preserved. Inputs with unresolved floating parts or substituents are replaced with missing values and produce a warning.
fallback_db	A glydb::glydb_structures() vector, or a character vector of glycan structure strings supported by glyparse::auto_parse() . The default is glydb::glydb_structures() at "topological" level. Every non-missing candidate must have concrete residues. Linkage information is removed before fallback matching. Candidates with unresolved floating parts or substituents are excluded silently. Fallback candidates require a confidence attribute.

Value

An unnamed [glyrepr::glycan_structure\(\)](#) vector with the same length as strucs. Unmatched structures are returned as NA.

Examples

```
n_generic <- glyrepr::n_glycan_core(linkage = FALSE, mono_type = "generic")
enhance_struc_denovo(n_generic)
```

glyanno_mass_dict	<i>Default Mass Dictionary for Glycan Residues</i>
-------------------	--

Description

A named numeric vector of the mass of each monosaccharide residue. Used by default as the `mass_dict` argument in `calculate_mz()`.

The names includes:

- Monosaccharides: Hex, HexNAc, dHex, dHexNAc, ddHex, Pen, HexA, HexN, NeuAc, NeuGc, Kdn, Neu
- Ions: H+, H, H2O, K+, Na+, NH4+, H-, Cl-, HCO3-
- Substituents: S, P
- Reducing end: red_end, which is the additional mass of the reducing end of the glycan.

Usage

```
glyanno_mass_dict(deriv = "none", mass_type = "mono")
```

Arguments

<code>deriv</code>	A character scalar of the derivatization method to use. Can be "none", "permethyl", or "peracetyl". Default is "none".
<code>mass_type</code>	"mono" for monoisotopic mass, "average" for average mass. Default is "mono".

Value

A named numeric vector.

Examples

```
glyanno_mass_dict(deriv = "none", mass_type = "mono")
glyanno_mass_dict(deriv = "permethyl", mass_type = "mono")
glyanno_mass_dict(deriv = "none", mass_type = "average")
```

mz_to_comp

*Convert m/z values to glycan composition***Description**

Given m/z values, this function matches them to all possible glycan compositions in the glydb database.

Usage

```
mz_to_comp(
  mz,
  tol = ppm(10),
  db = lifecycle::deprecated(),
  return_best = FALSE,
  charge = 1,
  adduct = "H+",
  mass_dict = NULL,
  glycan_type = NULL,
  species = NULL,
  mono_type = "concrete",
  mono_range = NULL
)
```

Arguments

mz	A numeric vector of m/z values.
tol	A numeric scalar of the tolerance for the m/z value in Da or a <code>ppm()</code> object for dynamic tolerance. Default is <code>ppm(10)</code> .
db	[Deprecated] Glycan compositions to match against. Can be a <code>glyrepr::glycan_composition()</code> vector or glycan composition strings in Byonic style (e.g. <code>Hex(5)HexNAc(2)</code>) or simple style (e.g. <code>H5N4F1S1</code>). Use the filtering arguments instead. This argument cannot be combined with them.
return_best	A logical scalar. If TRUE, only the match with the highest confidence score is returned for each m/z value. A custom db must have a confidence attribute. Default is FALSE.
charge	Charge to use. Can be 0, 1, 2, 3, -1, -2, -3, etc. 0 means neutral. Default is 1.
adduct	Adduct to use. Can be "H+", "K+", "Na+", "NH4+", "H-", "Cl-", "HCO3-". Default is "H+". • When charge is 0, adduct is ignored. • When charge is positive, adduct can only be "H+", "K+", "Na+", "NH4+". • When charge is negative, adduct can only be "H-", "Cl-", "HCO3-".
mass_dict	A named numeric vector of the mass of each monosaccharide residue. Default is <code>glyanno_mass_dict(deriv = "none", mass_type = "mono")</code> . If a custom mass dictionary is provided, please make sure the names of the vector are the same as the names in <code>glyanno_mass_dict()</code> .

glycan_type	Glycan type to select from <code>glydb::glydb_compositions()</code> .
species	Species to select, matched without regard to letter case.
mono_type	Monosaccharide resolution, "concrete" (default) or "generic".
mono_range	Named list of monosaccharide count ranges; see <code>glydb::glydb_compositions()</code> .

Value

If `return_best=TRUE`: An unnamed `glyrepr::glycan_composition()` vector with the same length as `mz`. Unmatched `m/z` values are returned as NA. If `return_best=FALSE`: A tibble with the following columns:

- `mz`: The molecule `m/z` values, same as the input `mz`.
- `composition`: The possible glycan compositions, as `glyrepr::glycan_composition()` vector.
- `confidence`: The database confidence score for each composition, or NA when no score is available. Note that one `m/z` value can have multiple rows in the result, corresponding to different possible glycan compositions.

Filter the built-in database

Use `glycan_type`, `species`, `mono_type`, and `mono_range` to select compositions from `glydb::glydb_compositions()`. For example,

```
mz_to_comp(mz, species = "Homo sapiens", glycan_type = "N")
```

See Also

`ppm()`, `glyanno_mass_dict()`

Examples

```
mz_to_comp(933.3175, charge = 1, adduct = "Na+")
```

ppm

Calculate PPM

Description

This helper function should be used in `mz_to_comp()` as the `tol` argument. It makes the function using PPM as dynamic tolerance.

Usage

```
ppm(x)
```

Arguments

x A numeric scalar of the PPM value.

Value

A function that takes a numeric vector of m/z values and returns a numeric vector of tolerances.

Examples

```
ppm(10)(2368.84)
ppm(10)(c(2368.84, 2368.85))
```

struc_to_glytoucan	<i>Assign GlyTouCan accessions to glycan structures</i>
--------------------	---

Description

This function takes a vector of glycan structures and returns the corresponding GlyTouCan accessions. It first looks up structures from [glydb::glydb_data](#), then uses the GlycanFormatConverter API maintained by the Glycosmos project for structures not found locally.

Usage

```
struc_to_glytoucan(strucs)
```

Arguments

strucs A [glyrepr::glycan_structure\(\)](#) vector, or a character vector of glycan text representations supported by [glyparse::auto_parse\(\)](#). The glycan structure must have "concrete" monosaccharides (e.g., Gal, GalNAc). Inputs with unresolved floating parts or substituents produce a warning and return NA.

Details

For "topological" structures (e.g., "Gal(??-)GalNAc(??-)"), this function will first call [glyrepr::fill_anomer_pos\(\)](#) to fill in the missing anomeric positions before looking up accessions. This is necessary because all glycan structures in GlyTouCan must have defined anomeric positions.

Value

A character vector of GlyTouCan accessions corresponding to the input glycan structures. If a structure cannot be converted to a GlyTouCan accession, the corresponding entry will be NA.

Index

build_glyanno_cache, [2](#)

calculate_mz, [3](#)
calculate_mz(), [10](#)
clear_glyanno_cache
 (build_glyanno_cache), [2](#)
comp_to_struc, [5](#)

enhance_comp, [6](#)
enhance_struc, [7](#)
enhance_struc_denovo, [9](#)

glyanno_cache_info
 (build_glyanno_cache), [2](#)
glyanno_mass_dict, [10](#)
glyanno_mass_dict(), [4](#), [11](#), [12](#)
glydb::glydb_compositions(), [7](#), [12](#)
glydb::glydb_data, [13](#)
glydb::glydb_structures(), [5](#), [8](#), [9](#)
glyparse::auto_parse(), [3](#), [5](#), [6](#), [8](#), [9](#), [13](#)
glyrepr::fill_anomer_pos(), [13](#)
glyrepr::get_structure_level(), [7](#)
glyrepr::glycan_composition(), [3](#), [5–7](#),
 [11](#), [12](#)
glyrepr::glycan_structure(), [3](#), [5](#), [6](#), [8](#), [9](#),
 [13](#)

mz_to_comp, [11](#)
mz_to_comp(), [12](#)

ppm, [12](#)
ppm(), [11](#), [12](#)

struc_to_glytoucan, [13](#)