

Package ‘syrona’

September 4, 2026

Type Package

Title Stratified Prevalence Comparison Across OMOP CDM Datasets

Version 0.2.1

Description Derives stratified prevalence tables from the condition, procedure, and drug records in OMOP CDM (Observational Medical Outcomes Partnership Common Data Model) databases, computes log2 prevalence ratios between paired datasets, and synthesizes them via random-effects meta-analysis at multiple aggregation levels (year, age group, and sex). Between-study variance is estimated with the Paule-Mandel method, as described in Paule and Mandel (1982) <doi:10.6028/jres.087.022>.

License MIT + file LICENSE

URL <https://github.com/HealthInformaticsUT/Syrona>

BugReports <https://github.com/HealthInformaticsUT/Syrona/issues>

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports dplyr (>= 1.1.0), dbplyr (>= 2.3.0), DBI (>= 1.2.0), CDMConnector (>= 2.0.0), omopgenerics (>= 1.3.0), tibble, tidyr, cli, rlang, shiny, ggplot2, ggiraph, ggtext, scales, grDevices, stats, readr (>= 2.0.0), meta (>= 6.0.0)

Suggests testthat (>= 3.0.0), duckdb (>= 0.9.0), RPostgres, CohortConstructor (>= 0.6.0), DT, shinycssloaders, knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Maarja Pajusalu [aut, cre]

Maintainer Maarja Pajusalu <maarja.pajusalu@ut.ee>

Repository CRAN

Date/Publication 2026-09-04 20:10:02 UTC

Contents

apply_cohort_filter	2
build_forest_detail	3
build_forest_overview	4
build_forest_summary	5
build_heatmap	5
build_point_diff	6
build_pr_distribution	7
build_pr_distribution_chapters	7
build_pyramid	8
cohort_summary	9
compare_all	9
compare_meta_agegroups	10
compare_meta_by_sex	11
compare_meta_summary	11
compare_yearly	12
create_caresite_cohort	12
delete_cohort	13
extract_all	14
format_fold	14
format_pval	15
insert_cohort	15
list_care_sites	16
list_comparisons	16
list_datasets	17
load_comparison	17
load_dataset	18
load_syrona_theme	18
run_app	19
syrona_connect	20
syrona_connect_pg	20
syrona_disconnect	21
truncate_name	22
weight_color	22
Index	23

apply_cohort_filter	<i>Apply OHDSI cohort filtering to CDM table references.</i>
---------------------	--

Description

Modifies db\$cdm table references to filter by a standard OHDSI cohort table. Filters person, observation_period, death, and event tables to cohort members and their cohort windows.

Usage

```
apply_cohort_filter(db, cohort_id, cohort_schema = NULL)
```

Arguments

db	Connection list (from syrona_connect).
cohort_id	Integer cohort_definition_id to filter by.
cohort_schema	Schema containing the cohort table (NULL = default).

Value

Modified db list with filtered CDM table references.

build_forest_detail	<i>Build a per-year forest plot for a single concept</i>
---------------------	--

Description

Shows yearly points + CIs, meta-analysis diamond, and colored heatmap tile row. Faceted by sex x age group.

Usage

```
build_forest_detail(
  yearly_data,
  meta_data,
  concept_name,
  concept_code,
  name1,
  name2,
  scale_mode = "log2"
)
```

Arguments

yearly_data	Yearly comparison data for one concept.
meta_data	Meta-analysis data (across years) for one concept.
concept_name	Concept name string.
concept_code	Concept code string.
name1	Reference dataset name.
name2	Comparison dataset name.
scale_mode	"fold" or "log2".

Value

A girafe object.

`build_forest_overview` *Build an overview forest plot paired with the heatmap*

Description

Compact forest showing Female, Male, Both meta PR for every concept. Concept y-axis ordering matches the heatmap.

Usage

```
build_forest_overview(  
  meta_sex_df,  
  meta_summary_df,  
  concept_ids,  
  concept_info,  
  name1,  
  name2,  
  scale_mode = "log2",  
  concept_order = NULL  
)
```

Arguments

<code>meta_sex_df</code>	Meta-analysis by sex data (F/M rows).
<code>meta_summary_df</code>	Meta-analysis summary data (Both rows).
<code>concept_ids</code>	Vector of <code>concept_ids</code> to include.
<code>concept_info</code>	Data frame with <code>concept_id</code> , <code>concept_name</code> , <code>concept_code</code> , <code>pop_weight</code> .
<code>name1</code>	Reference dataset name.
<code>name2</code>	Comparison dataset name.
<code>scale_mode</code>	"fold" or "log2".
<code>concept_order</code>	Vector of <code>concept_ids</code> in desired display order.

Value

A girafe object.

build_forest_summary	<i>Build a compact F/M/Both forest plot for a single concept</i>
----------------------	--

Description

Build a compact F/M/Both forest plot for a single concept

Usage

```
build_forest_summary(  
  meta_sex_data,  
  meta_summary_data,  
  concept_name,  
  concept_code,  
  name1,  
  name2,  
  scale_mode = "log2"  
)
```

Arguments

meta_sex_data	Meta-analysis data by sex (F, M) for one concept.
meta_summary_data	Meta-analysis summary (Both) for one concept.
concept_name	Concept name string.
concept_code	Concept code string.
name1	Reference dataset name.
name2	Comparison dataset name.
scale_mode	"fold" or "log2".

Value

A girafe object.

build_heatmap	<i>Build an interactive heatmap of prevalence ratios</i>
---------------	--

Description

Build an interactive heatmap of prevalence ratios

Usage

```
build_heatmap(
  data,
  name1,
  name2,
  scale_mode = "log2",
  concept_weights = NULL,
  concept_order = NULL
)
```

Arguments

data	Data frame with meta_agegroups data (concept_id, concept_name, concept_code, sex, age_group, log2_pr, ci_low, ci_high, p_value).
name1	Reference dataset name.
name2	Comparison dataset name.
scale_mode	"fold" or "log2" for axis labels.
concept_weights	Data frame with concept_id, concept_name, pop_weight.
concept_order	Vector of concept_ids in desired display order.

Value

A girafe object.

build_point_diff	<i>Build a dumbbell chart showing absolute prevalence values</i>
------------------	--

Description

Complements the forest detail (relative -> absolute). Same facet structure: sex x age_group.

Usage

```
build_point_diff(yearly_data, concept_name, concept_code, name1, name2)
```

Arguments

yearly_data	Yearly comparison data for one concept.
concept_name	Concept name string.
concept_code	Concept code string.
name1	Reference dataset name.
name2	Comparison dataset name.

Value

A girafe object.

build_pr_distribution *Build a PR distribution histogram from meta_summary data*

Description

Build a PR distribution histogram from meta_summary data

Usage

```
build_pr_distribution(  
  data,  
  d1_name,  
  d2_name,  
  scale_mode = "fold",  
  fold_thresh = FOLD_THRESHOLD,  
  domain_label = "conditions"  
)
```

Arguments

data	Filtered meta_summary rows (one per concept). Expected columns: log2_pr, fold_diff, sig, concept_id.
d1_name	Name of reference dataset (shown on left side).
d2_name	Name of comparison dataset (shown on right side).
scale_mode	"fold" or "log2" - controls x-axis labels only.
fold_thresh	Fold difference threshold for zone coloring.
domain_label	Label for domain (e.g. "conditions", "drugs").

Value

A ggplot object.

build_pr_distribution_chapters
Build a faceted PR distribution - one small histogram per chapter

Description

Interactive (ggiraph): clicking a chapter facet returns its chapter_id.

Usage

```
build_pr_distribution_chapters(  
  data,  
  d1_name,  
  d2_name,  
  scale_mode = "fold",  
  fold_thresh = FOLD_THRESHOLD,  
  domain_label = "conditions"  
)
```

Arguments

data	Filtered meta_summary joined with chapters (one row per concept x chapter). Expected columns: log2_pr, fold_diff, sig, concept_id, chapter_name, chapter_id.
d1_name	Name of reference dataset.
d2_name	Name of comparison dataset.
scale_mode	"fold" or "log2".
fold_thresh	Fold difference threshold.
domain_label	Label for domain.

Value

A girafe object. Selection data_id = chapter_id (character).

build_pyramid	<i>Build an interactive population pyramid</i>
---------------	--

Description

Build an interactive population pyramid

Usage

```
build_pyramid(demo_df, dataset_name)
```

Arguments

demo_df	Demographics data frame with birth_year, sex, patient_count.
dataset_name	Name of the dataset (used as title).

Value

A girafe object.

cohort_summary	<i>Get summary statistics for a cohort.</i>
----------------	---

Description

Get summary statistics for a cohort.

Usage

```
cohort_summary(con, cohort_id, cohort_schema = NULL)
```

Arguments

con	DBI connection.
cohort_id	Integer cohort_definition_id.
cohort_schema	Schema containing the cohort table (NULL = default).

Value

A tibble with n_entries, n_persons, min_start, max_end.

compare_all	<i>Run the full comparison pipeline for two datasets across available domains.</i>
-------------	--

Description

Loads both datasets, runs the 4-step comparison for each domain present in both, and optionally saves to CSV.

Usage

```
compare_all(  
  d1_name,  
  d2_name,  
  domains = c("conditions", "procedures", "drugs"),  
  save = TRUE  
)
```

Arguments

d1_name	Name of dataset 1 (reference).
d2_name	Name of dataset 2 (comparison).
domains	Character vector of domains to compare.
save	If TRUE (default), writes CSV output.

Value

Named list of domain results (invisible).

Examples

```
# Run the bundled demo (a curated 32-concept subset) end to end.
# Copy it to a writable folder first, since the installed copy is read-only.
base <- tempdir()
file.copy(system.file("extdata", "demo", package = "syrona"),
           base, recursive = TRUE)
dir <- file.path(base, "demo")
old <- options(syrona.data_dir = dir)
# One domain keeps the example quick; drop `domains` to compare all three.
res <- compare_all("demo_population", "demo_selected", domains = "conditions")
res$condition_meta_summary
options(old)
# Then explore interactively: run_app(data_dir = dir)
```

compare_meta_agegroups

Meta-analysis across years: one row per concept_id x sex x age_group.

Description

Meta-analysis across years: one row per concept_id x sex x age_group.

Usage

```
compare_meta_agegroups(yearly_df)
```

Arguments

yearly_df Output of compare_yearly.

Value

tibble with meta-analyzed rows.

compare_meta_by_sex	<i>Meta-analysis across age groups: one row per concept_id x sex.</i>
---------------------	---

Description

Meta-analysis across age groups: one row per concept_id x sex.

Usage

```
compare_meta_by_sex(meta_agegroups_df)
```

Arguments

meta_agegroups_df
Output of compare_meta_agegroups.

Value

tibble with meta-analyzed rows.

compare_meta_summary	<i>Meta-analysis across sexes: one row per concept_id (sex = "Both").</i>
----------------------	---

Description

Meta-analysis across sexes: one row per concept_id (sex = "Both").

Usage

```
compare_meta_summary(meta_by_sex_df)
```

Arguments

meta_by_sex_df Output of compare_meta_by_sex.

Value

tibble with one summary row per concept.

compare_yearly	<i>Compare two datasets at the per-year stratum level.</i>
----------------	--

Description

Matches on concept_id x year x sex x age_group (inner join). Only keeps strata where both datasets have prevalence > 0.

Usage

```
compare_yearly(d1, d2, prev_table = "condition_prevalence")
```

Arguments

d1	Named list from load_dataset (reference).
d2	Named list from load_dataset (comparison).
prev_table	Name of the prevalence table to compare.

Value

tibble with one row per matched stratum, full PR column set.

create_caresite_cohort	<i>Create an OHDSI cohort from a care site.</i>
------------------------	---

Description

For each person who visited the specified care site, their cohort window runs from their first visit start date to their last visit end date at that site. Optionally clips to observation period overlap.

Usage

```
create_caresite_cohort(
  con,
  care_site_id,
  cohort_id,
  cohort_schema = NULL,
  cdm_schema,
  overwrite = FALSE,
  restrict_to_observation = TRUE,
  collapse_strategy = c("person_span", "visit_occurrence")
)
```

Arguments

con	DBI connection to the OMOP CDM database.
care_site_id	Integer care_site_id to define the cohort.
cohort_id	Integer cohort_definition_id to assign.
cohort_schema	Schema for the cohort table (NULL = default).
cdm_schema	Schema containing OMOP CDM tables.
overwrite	If TRUE, deletes existing entries for this cohort_id before inserting.
restrict_to_observation	If TRUE (default), clips cohort intervals to observation period overlap.
collapse_strategy	How to collapse visits per person: "person_span" (default) = first visit start to last visit end; "visit_occurrence" = one cohort entry per visit.

Details

This uses server-side INSERT...SELECT for performance on large databases (avoids downloading + re-uploading patient-level data).

Value

Number of rows inserted (invisible).

delete_cohort	<i>Delete a cohort from the cohort table.</i>
---------------	---

Description

Delete a cohort from the cohort table.

Usage

```
delete_cohort(con, cohort_id, cohort_schema = NULL)
```

Arguments

con	DBI connection.
cohort_id	Integer cohort_definition_id to delete.
cohort_schema	Schema containing the cohort table (NULL = default).

Value

Number of rows deleted (invisible).

extract_all	<i>Run the full source data extraction pipeline.</i>
-------------	--

Description

Derives stratified prevalence tables from the records in an OMOP CDM database for one or more clinical domains. Applies k-anonymity suppression and optionally saves results to CSV.

Usage

```
extract_all(
  dataset_name,
  db,
  domains = c("conditions", "procedures", "drugs"),
  cohort_id = NULL,
  cohort_schema = NULL,
  save = TRUE
)
```

Arguments

dataset_name	Short label for the dataset (e.g. "EH30", "EstBB").
db	Either a DuckDB file path (string) or an existing connection list from syrona_connect / syrona_connect_pg.
domains	Character vector of domains to extract. Options: "conditions", "procedures", "drugs".
cohort_id	Integer cohort_definition_id to filter by. If NULL (default), extracts the full dataset.
cohort_schema	Schema containing the cohort table.
save	If TRUE (default), saves CSV to data/sources/<dataset_name>/.

Value

Named list of tibbles matching the Syrona schema (invisible).

format_fold	<i>Format a log2 prevalence ratio as a fold difference string</i>
-------------	---

Description

Vectorized. For values < 1, shows reciprocal in parentheses.

Usage

```
format_fold(log2_val)
```

Arguments

log2_val Numeric vector of log2 PR values.

Value

Character vector of formatted fold strings.

format_pval	<i>Format p-values in Nature style</i>
-------------	--

Description

Vectorized. Shows < 0.001 for very small values.

Usage

```
format_pval(p)
```

Arguments

p Numeric vector of p-values.

Value

Character vector of formatted p-value strings.

insert_cohort	<i>Insert a cohort from a local data frame.</i>
---------------	---

Description

Uploads a data frame to the database via `omopgenerics::insertTable` and marks it as a cohort table with `omopgenerics::newCohortTable`. Use this when cohort membership has already been computed locally (e.g. from a CSV or programmatic cohort definition).

Usage

```
insert_cohort(cdm, cohort_df, name = "cohort")
```

Arguments

cdm	A CDM reference (from <code>syrona_connect()\$cdm</code>).
cohort_df	Data frame with columns: <code>cohort_definition_id</code> , <code>subject_id</code> , <code>cohort_start_date</code> , <code>cohort_end_date</code> .
name	Name for the cohort table in the database (default "cohort").

Details

Requires that the CDM connection was created with a writeSchema.

Value

Updated CDM reference with the cohort table attached.

list_care_sites	<i>List care sites with patient counts.</i>
-----------------	---

Description

Useful for identifying care_site_id values before creating cohorts.

Usage

```
list_care_sites(con, cdm_schema, min_patients = 100)
```

Arguments

con	DBI connection.
cdm_schema	Schema containing the OMOP CDM tables.
min_patients	Minimum number of patients to include (default 100).

Value

A tibble with care_site_id, care_site_name, n_patients.

list_comparisons	<i>List available comparisons.</i>
------------------	------------------------------------

Description

List available comparisons.

Usage

```
list_comparisons()
```

Value

Character vector of comparison directory names.

list_datasets	<i>List all available datasets.</i>
---------------	-------------------------------------

Description

List all available datasets.

Usage

```
list_datasets()
```

Value

Character vector of dataset names.

load_comparison	<i>Load a previously saved comparison from CSV.</i>
-----------------	---

Description

Load a previously saved comparison from CSV.

Usage

```
load_comparison(d1_name, d2_name)
```

Arguments

d1_name	Dataset 1 name.
d2_name	Dataset 2 name.

Value

Named list of tibbles.

load_dataset	<i>Load a previously saved dataset from CSV.</i>
--------------	--

Description

Load a previously saved dataset from CSV.

Usage

```
load_dataset(dataset_name)
```

Arguments

dataset_name Short label (subfolder name under data/sources/).

Value

Named list of tibbles.

Examples

```
base <- tempdir()
file.copy(system.file("extdata", "demo", package = "syrona"), base, recursive = TRUE)
old <- options(syrona.data_dir = file.path(base, "demo"))
d <- load_dataset("demo_population")
names(d)
options(old)
```

load_syrona_theme	<i>Load all Syrona design tokens into the calling environment</i>
-------------------	---

Description

Makes color constants, thresholds, and other design tokens available as variables. Used by the Shiny dashboard's global.R.

Usage

```
load_syrona_theme(envir = parent.frame())
```

Arguments

envir Environment to load into (default: caller's environment).

Value

No return value, called for its side effect of assigning the design tokens as variables in envir.

run_app	<i>Run the Syrona dashboard</i>
---------	---------------------------------

Description

Launches the Shiny dashboard for exploring prevalence comparisons. The app looks for data/sources/ and data/comparisons/ relative to your current working directory. Run this from the directory that contains your data/ folder.

Usage

```
run_app(data_dir = getwd(), port = NULL, launch.browser = TRUE, ...)
```

Arguments

data_dir	Path to the directory containing data/sources/ and data/comparisons/. Defaults to the current working directory.
port	Port to run the app on (default: auto-select).
launch.browser	Whether to open a browser window (default: TRUE).
...	Additional arguments passed to runApp .

Value

No return value, called for its side effect of launching the Shiny dashboard.

Examples

```
# Copy the bundled demo, generate the comparison, then launch the dashboard.
if (interactive()) {
  base <- tempdir()
  file.copy(system.file("extdata", "demo", package = "syrona"), base, recursive = TRUE)
  dir <- file.path(base, "demo")
  old <- options(syrona.data_dir = dir)
  compare_all("demo_population", "demo_selected")
  run_app(data_dir = dir)
  options(old)
}
```

syrona_connect	<i>Connect to a DuckDB OMOP CDM.</i>
----------------	--------------------------------------

Description

Opens a DuckDB file and creates a CDMConnector reference. By default the connection is read-only; set `read_only = FALSE` if you need to write cohort tables or temp tables into the database.

Usage

```
syrona_connect(
  db_path,
  cdm_schema = "main",
  write_schema = cdm_schema,
  read_only = TRUE
)
```

Arguments

<code>db_path</code>	Path to the DuckDB file.
<code>cdm_schema</code>	Schema containing OMOP CDM tables (default "main").
<code>write_schema</code>	Schema for writing temp/cohort tables. Defaults to <code>cdm_schema</code> . Set to a different schema if the CDM schema is read-only (common in production setups).
<code>read_only</code>	Logical. Open DuckDB in read-only mode? Default TRUE. Set to FALSE when you need to create cohort tables or use <code>omopgenerics::insertTable</code> .

Value

A list with components:

con DBI connection object.

cdm CDMConnector CDM reference (`cdm_reference`).

syrona_connect_pg	<i>Connect to a PostgreSQL OMOP CDM.</i>
-------------------	--

Description

For remote databases, typically accessed via SSH tunnel: `ssh -L 5432:localhost:5432 user@server`.

Usage

```
syrona_connect_pg(
  host = "localhost",
  port = 5432,
  dbname,
  user,
  password = NULL,
  cdm_schema = "public",
  write_schema = cdm_schema
)
```

Arguments

host	Hostname (default "localhost" for SSH tunnel).
port	Port number (default 5432).
dbname	Database name.
user	PostgreSQL username.
password	Password. If NULL, uses .pgpass or PGPASSWORD env var.
cdm_schema	Schema with OMOP CDM tables (e.g. "cdm").
write_schema	Schema for cohort/temp tables (e.g. "results_maarja").

Value

Same structure as syrona_connect.

syrona_disconnect	<i>Disconnect from an OMOP CDM database.</i>
-------------------	--

Description

Disconnect from an OMOP CDM database.

Usage

```
syrona_disconnect(db)
```

Arguments

db	Connection list returned by syrona_connect or syrona_connect_pg.
----	--

Value

No return value, called for its side effect of closing the database connection.

truncate_name	<i>Truncate a concept name to a maximum number of characters</i>
---------------	--

Description

Adds "..." if truncated.

Usage

```
truncate_name(name, max = CONCEPT_NAME_MAX)
```

Arguments

name	Character vector of names.
max	Maximum character length (default CONCEPT_NAME_MAX).

Value

Character vector of (possibly truncated) names.

weight_color	<i>Map population weight (%) to a hex color on a log-scale gradient</i>
--------------	---

Description

Light gray for rare concepts, dark charcoal for common ones.

Usage

```
weight_color(pct)
```

Arguments

pct	Numeric vector of population weight percentages (0-100).
-----	--

Value

Character vector of hex color strings.

Index

`apply_cohort_filter`, [2](#)

`build_forest_detail`, [3](#)
`build_forest_overview`, [4](#)
`build_forest_summary`, [5](#)
`build_heatmap`, [5](#)
`build_point_diff`, [6](#)
`build_pr_distribution`, [7](#)
`build_pr_distribution_chapters`, [7](#)
`build_pyramid`, [8](#)

`cohort_summary`, [9](#)
`compare_all`, [9](#)
`compare_meta_agegroups`, [10](#)
`compare_meta_by_sex`, [11](#)
`compare_meta_summary`, [11](#)
`compare_yearly`, [12](#)
`create_caresite_cohort`, [12](#)

`delete_cohort`, [13](#)

`extract_all`, [14](#)

`format_fold`, [14](#)
`format_pval`, [15](#)

`insert_cohort`, [15](#)

`list_care_sites`, [16](#)
`list_comparisons`, [16](#)
`list_datasets`, [17](#)
`load_comparison`, [17](#)
`load_dataset`, [18](#)
`load_syrona_theme`, [18](#)

`run_app`, [19](#)
`runApp`, [19](#)

`syrona_connect`, [20](#)
`syrona_connect_pg`, [20](#)
`syrona_disconnect`, [21](#)

`truncate_name`, [22](#)
`weight_color`, [22](#)