

Package ‘ciecl’

May 8, 2026

Title International Classification of Diseases 'ICD-10'/ICD-11' for Chile

Version 0.9.6

Description Tools for working with the International Classification of Diseases ('ICD-10' Chile official 'MINSAL'/DEIS' v2018). Includes optimized 'SQL' search with 'SQLite', fuzzy matching of medical terms ('Jaro-Winkler'), Charlson and Elixhauser comorbidity calculation, 'WHO' 'ICD-11' 'API' integration, and hierarchical code validation. Data from Centro FIC Chile 'DEIS' <<https://deis.minsal.cl/centrofic/>>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports DBI, RSQLite, stringdist, stringr, dplyr, tibble, tools, utils

Suggests comorbidity, gt, httr2, readxl, usethis, withr, writexl, testthat (>= 3.1.5), knitr, litedown (>= 0.9), rmarkdown

VignetteBuilder knitr

URL <https://rodotasso.github.io/ciecl/>,
<https://github.com/RodoTasso/ciecl>

BugReports <https://github.com/RodoTasso/ciecl/issues>

Config/testthat/edition 3

Config/Needs/website pkgdown

Language en, es

NeedsCompilation no

Author Rodolfo Tasso Suazo [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-1264-4933>>)

Maintainer Rodolfo Tasso Suazo <rtasso@uchile.cl>

Repository CRAN

Date/Publication 2026-04-19 14:30:02 UTC

Contents

cie10_cl	2
cie10_clear_cache	3
cie10_disconnect	4
cie10_sql	4
cie11_search	5
cie_comorbid	6
cie_expand	7
cie_guia_busqueda	8
cie_lookup	8
cie_map_comorbid	10
cie_normalizar	10
cie_search	11
cie_siglas	13
cie_table	14
cie_validate_vector	14
Index	16

cie10_cl	<i>Dataset CIE-10 Chile oficial MINSAL/DEIS v2018</i>
----------	---

Description

Dataset CIE-10 Chile oficial MINSAL/DEIS v2018

Usage

cie10_cl

Format

tibble con 39,877 filas (categorias y subcategorias):

codigo Codigo CIE-10 (ej. "E11.0")

descripcion Diagnostico en espanol chileno

categoria Categoria jerarquica

seccion Seccion dentro del capitulo

capitulo_nombre Nombre descriptivo del capitulo

inclusion Terminos incluidos

exclusion Terminos excluidos

capitulo Capitulo CIE-10 (A-Z)

es_daga Logical, codigo daga (+)

es_cruz Logical, codigo asterisco (*)

Source

<https://deis.minsal.cl/centrofic/>

Examples

```
data(cie10_cl)
head(cie10_cl)
```

cie10_clear_cache	<i>Limpiar cache SQLite (forzar rebuild)</i>
-------------------	--

Description

Limpiar cache SQLite (forzar rebuild)

Usage

```
cie10_clear_cache()
```

Value

No return value, called for side effects (deletes SQLite cache).

See Also

[cie10_sql](#), [cie10_disconnect](#)

Other sql: [cie10_disconnect\(\)](#), [cie10_sql\(\)](#)

Examples

```
# Ver ubicacion del cache
tools::R_user_dir("ciecl", "data")

cie10_clear_cache() # Elimina cie10.db local
```

cie10_disconnect	<i>Cerrar conexion pooled SQLite</i>
------------------	--------------------------------------

Description

Cierra la conexion reutilizable al archivo SQLite. Util para liberar el lock del archivo .db.

Usage

```
cie10_disconnect()
```

Value

No return value, called for side effects.

See Also

[cie10_sql](#), [cie10_clear_cache](#)

Other sql: [cie10_clear_cache\(\)](#), [cie10_sql\(\)](#)

Examples

```
# Verificar si hay conexion activa
is.null(ciecl:::ciecl_env$con)
```

```
cie10_disconnect()
```

cie10_sql	<i>Ejecutar consultas SQL sobre CIE-10 Chile</i>
-----------	--

Description

Ejecutar consultas SQL sobre CIE-10 Chile

Usage

```
cie10_sql(query, close = TRUE)
```

Arguments

query	String SQL valido SQLite (SELECT/WHERE/JOIN)
close	Deprecated. Ignorado — la conexion es pooled y se gestiona automaticamente. Sera eliminado en una version futura.

Value

tibble resultado query

See Also

[cie10_clear_cache](#), [cie10_disconnect](#), [cie_search](#)

Other sql: [cie10_clear_cache\(\)](#), [cie10_disconnect\(\)](#)

Examples

```
# Buscar diabetes
cie10_sql("SELECT codigo, descripcion FROM cie10 WHERE codigo LIKE 'E11%'")
```

```
# Contar por capitulo
cie10_sql("SELECT capitulo, COUNT(*) n FROM cie10 GROUP BY capitulo")
```

cie11_search

Buscar codigos CIE-11 via API OMS

Description

Buscar codigos CIE-11 via API OMS

Usage

```
cie11_search(
  texto,
  api_key = NULL,
  lang = "es",
  max_results = 10,
  release = "2024-01"
)
```

Arguments

texto	String termino busqueda espanol/ingles
api_key	String opcional, Client ID + Secret OMS separados ":" Obtener en: https://icd.who.int/icdapi
lang	Character, idioma respuesta ("es" o "en")
max_results	Integer, maximo resultados (default 10)
release	Character, version de release CIE-11 a consultar (default "2024-01"). Ver releases disponibles en la API OMS.

Value

tibble con codigos CIE-11 + titulos o vacio si error

See Also

[cie_search](#), [cie_lookup](#)

Examples

```
# Ver parametros disponibles
args(cie11_search)

# Requiere credenciales OMS gratuitas (https://icd.who.int/icdapi)
Sys.setenv(ICD_API_KEY = "client_id:client_secret")
cie11_search("depression mayor")
```

cie_comorbid

Calcular comorbilidades Charlson/Elixhauser para Chile

Description

Calcular comorbilidades Charlson/Elixhauser para Chile

Usage

```
cie_comorbid(data, id, code, map = c("charlson", "elixhauser"), assign0 = TRUE)
```

Arguments

data	data.frame con columnas id paciente + codigos CIE-10
id	String nombre columna identificador paciente
code	String nombre columna con codigos CIE-10 (uno por fila)
map	Character, esquema comorbilidad ("charlson" o "elixhauser")
assign0	Logical, asignar 0 si sin comorbilidad (default TRUE)

Value

data.frame ancho con scores comorbilidad por paciente

See Also

[cie_map_comorbid](#), [cie_normalizar](#)

Other comorbilidades: [cie_map_comorbid\(\)](#)

Examples

```
# Ver documentacion de parametros
args(cie_comorbid)

df <- data.frame(
  id_pac = c(1, 1, 2, 2),
  diag = c("E11.0", "I21.0", "C50.9", "E10.9")
)
cie_comorbid(df, id = "id_pac", code = "diag", map = "charlson")
```

cie_expand	<i>Expandir codigo jerarquico (ej. E11 -> E11.0-E11.9)</i>
------------	---

Description

Expandir codigo jerarquico (ej. E11 -> E11.0-E11.9)

Usage

```
cie_expand(codigo)
```

Arguments

codigo String codigo padre (ej. "E11")

Value

Character vector con todos los codigos hijos del codigo padre. Vector vacio si el codigo no existe en la base de datos.

See Also

[cie_normalizar](#), [cie_lookup](#)

Other validacion: [cie_normalizar\(\)](#), [cie_validate_vector\(\)](#)

Examples

```
cie_expand("E11")
```

cie_guia_busqueda *Guia de funciones de busqueda CIE-10*

Description

Muestra tabla comparativa de cuando usar cada funcion de busqueda.

Usage

```
cie_guia_busqueda()
```

Value

tibble con guia comparativa de funciones de busqueda

See Also

[cie_search](#), [cie_lookup](#), [cie_siglas](#)

Other busqueda: [cie_lookup\(\)](#), [cie_search\(\)](#), [cie_siglas\(\)](#)

Examples

```
cie_guia_busqueda()
```

cie_lookup *Busqueda exacta por codigo CIE-10*

Description

Busqueda exacta por codigo CIE-10

Usage

```
cie_lookup(  
  codigo,  
  expandir = FALSE,  
  normalizar = TRUE,  
  descripcion_completa = FALSE,  
  extract = FALSE,  
  check_siglas = FALSE  
)
```

Arguments

codigo	Character vector de codigos (ej. "E11", "E11.0", c("E11.0", "Z00")) o rango (ej. "E10-E14"). Acepta vectores. Soporta formatos: con punto (E11.0), sin punto (E110), o solo categoria (E11).
expandir	Logical, expandir jerarquia completa (default FALSE)
normalizar	Logical, normalizar formato de codigos automaticamente (default TRUE)
descripcion_completa	Logical, agregar columna descripcion_completa con formato "CODIGO - DESCRIPCION" (default FALSE)
extract	Logical, extraer codigo CIE-10 de texto con prefijos/sufijos (default FALSE). IMPORTANTE: Solo usar con codigo ESCALAR (longitud 1). Ejemplo: "CIE:E11.0" -> "E11.0", "E11.0-confirmado" -> "E11.0". Para vectores multiples usar extract=FALSE (default).
check_siglas	Logical, buscar siglas medicas comunes (default FALSE). Ejemplo: "IAM" -> I21.0 (Infarto agudo miocardio)

Value

tibble con codigo(s) matcheado(s)

See Also

[cie_search](#), [cie_normalizar](#), [cie_expand](#)

Other busqueda: [cie_guia_busqueda\(\)](#), [cie_search\(\)](#), [cie_siglas\(\)](#)

Examples

```
# Busqueda directa por codigo
cie_lookup("E11.0")

cie_lookup("E110")      # Sin punto
cie_lookup("E11")      # Solo categoria
cie_lookup("E11", expandir = TRUE) # Todos E11.x
# Vectorizado - multiples codigos y formatos
cie_lookup(c("E11.0", "Z00", "I10"))
# Con descripcion completa
cie_lookup("E110", descripcion_completa = TRUE)
# Extraer codigo de texto con ruido (solo codigo escalar)
cie_lookup("CIE:E11.0", extract = TRUE)
cie_lookup("E11.0-confirmado", extract = TRUE)
# Buscar por siglas medicas
cie_lookup("IAM", check_siglas = TRUE)
cie_lookup("DM2", check_siglas = TRUE)
```

cie_map_comorbid	<i>Mapeo manual grupos comorbilidad Chile-especifico</i>
------------------	--

Description

Agrupar códigos CIE-10 chilenos en categorías comorbilidad MINSAL. Basado en Decreto 1301/2016 MINSAL + icd::icd10_map_charlson.

Usage

```
cie_map_comorbid(codigos)
```

Arguments

codigos	Character vector códigos CIE-10
---------	---------------------------------

Value

tibble con código + categoría_comorbilidad

See Also

[cie_comorbid](#), [cie_normalizar](#)

Other comorbilidades: [cie_comorbid\(\)](#)

Examples

```
cie_map_comorbid(c("E11.0", "I50.9", "C50.9"))
```

cie_normalizar	<i>Normalizar códigos CIE-10 a formato con punto</i>
----------------	--

Description

Convierte códigos CIE-10 de diferentes formatos al formato estándar (con punto). Maneja múltiples variaciones de entrada comunes en datos clínicos.

Usage

```
cie_normalizar(codigos, buscar_db = TRUE)
```

Arguments

codigos	Character vector de códigos en cualquier formato
buscar_db	Logical, buscar código en base de datos si no se encuentra exacto (default TRUE)

Details

La normalizacion incluye:

- Conversion a mayusculas
- Eliminacion de espacios (inicio, fin e internos)
- Eliminacion de simbolos daga y asterisco (codificacion dual)
- Conversion de guiones a puntos (I10-0 -> I10.0)
- Eliminacion de puntos iniciales (.I10 -> I10)
- Correccion de puntos multiples (E..11 -> E.11)
- Eliminacion de sufijo X en codigos cortos (I10X -> I10)
- Preservacion de X en codigos largos (placeholder 7o caracter)
- Agregado de punto en posicion correcta (E110 -> E11.0)

El sistema de daga/asterisco indica codificacion dual donde la daga marca la enfermedad subyacente y el asterisco la manifestacion. Ambos simbolos se eliminan para normalizacion.

Value

Character vector con codigos normalizados al formato con punto

See Also

[cie_validate_vector](#), [cie_expand](#), [cie_lookup](#)

Other validacion: [cie_expand\(\)](#), [cie_validate_vector\(\)](#)

Examples

```

cie_normalizar("E110")      # Retorna "E11.0"
cie_normalizar("E11")      # Retorna "E11" (categoria)
cie_normalizar("I10X")     # Retorna "I10" (elimina X)
cie_normalizar("E 11 0")   # Retorna "E11.0" (espacios internos)
cie_normalizar("I10-0")    # Retorna "I10.0" (guion a punto)
cie_normalizar("A17.0\u2020") # Retorna "A17.0" (elimina daga)
cie_normalizar("G01*")     # Retorna "G01" (elimina asterisco)
cie_normalizar(c("E110", "I10X", "Z00")) # Vectorizado

```

cie_search

Busqueda difusa (fuzzy) de terminos medicos CIE-10

Description

Busca en descripciones CIE-10 usando multiples estrategias:

1. Expansion de siglas medicas (IAM, TBC, DM, etc.)
2. Busqueda exacta por subcadena (mas rapida)
3. Busqueda fuzzy con Jaro-Winkler (tolera typos)

Usage

```

cie_search(
  texto,
  threshold = 0.7,
  max_results = 50,
  campo = c("descripcion", "inclusion"),
  solo_fuzzy = FALSE,
  verbose = TRUE
)

```

Arguments

texto	String termino medico en espanol o sigla (ej. "diabetes", "IAM", "TBC")
threshold	Numeric entre 0 y 1, umbral similitud Jaro-Winkler (default 0.70)
max_results	Integer, maximo resultados a retornar (default 50)
campo	Character, campo busqueda ("descripcion" o "inclusion")
solo_fuzzy	Logical, usar solo busqueda fuzzy sin busqueda exacta (default FALSE)
verbose	Logical, mostrar mensajes informativos (default TRUE). Usar FALSE en scripts.

Details

La busqueda es tolerante a tildes: "neumonia" encuentra "neumonia". Soporta siglas medicas comunes: "IAM" busca "infarto agudo miocardio".

Value

tibble ordenado por score descendente (1.0 = coincidencia exacta). Si el texto corresponde a una sigla medica, se expande automaticamente antes de buscar.

See Also

[cie_lookup](#), [cie_siglas](#), [cie10_sql](#)

Other busqueda: [cie_guia_busqueda\(\)](#), [cie_lookup\(\)](#), [cie_siglas\(\)](#)

Examples

```

# Busqueda basica
cie_search("diabetes")

cie_search("neumonia")

# Busqueda por siglas medicas
cie_search("IAM")
cie_search("DM2")

# Tolerante a tildes y typos
cie_search("diabetis")

```

```
# Buscar en inclusiones
cie_search("bacteriana", campo = "inclusion")
```

cie_siglas	<i>Listar siglas medicas soportadas</i>
------------	---

Description

Muestra todas las siglas medicas que pueden usarse en `cie_search()`.

Usage

```
cie_siglas(categoria = NULL)
```

Arguments

`categoria` Character opcional, filtrar por categoria. Valores validos: "cardiovascular", "respiratoria", "metabolica", "gastrointestinal", "infecciosa", "oncologica", "reumatologica", "neurologica", "psiquiatrica", "traumatologica", "pediatrica", "gineco_obstetrica". Si es NULL (default), retorna todas las siglas.

Value

tibble con columnas: sigla, termino_busqueda, categoria

See Also

[cie_search](#), [cie_lookup](#)

Other busqueda: [cie_guia_busqueda\(\)](#), [cie_lookup\(\)](#), [cie_search\(\)](#)

Examples

```
# Ver todas las siglas
cie_siglas()

# Filtrar por categoria
cie_siglas("cardiovascular")
cie_siglas("oncologica")

# Buscar una sigla especifica
cie_siglas() |> dplyr::filter(sigla == "iam")
```

cie_table	<i>Generar tabla HTML interactiva GT de codigo CIE-10</i>
-----------	---

Description

Generar tabla HTML interactiva GT de codigo CIE-10

Usage

```
cie_table(codigo)
```

Arguments

codigo	String codigo (ej. "E11" muestra jerarquia)
--------	---

Value

Objeto de clase gt_tbl (tabla HTML interactiva)

See Also

[cie_search](#), [cie_lookup](#)

Examples

```
# Requiere paquete gt
requireNamespace("gt", quietly = TRUE)

cie_table("E11") # Diabetes tipo 2 completo
```

cie_validate_vector	<i>Validar vector de codigos CIE-10 formato</i>
---------------------	---

Description

Validar vector de codigos CIE-10 formato

Usage

```
cie_validate_vector(codigos, strict = FALSE)
```

Arguments

codigos	Character vector codigos (ej. c("E11.0", "Z00.0"))
strict	Logical, validar existencia en DB (default FALSE)

Value

Logical vector de la misma longitud que codigos. TRUE si el codigo tiene formato CIE-10 valido (y existe en DB si `strict = TRUE`).

See Also

[cie_normalizar](#), [cie_expand](#)

Other validacion: [cie_expand\(\)](#), [cie_normalizar\(\)](#)

Examples

```
cie_validate_vector(c("E11.0", "INVALIDO", "Z00"))
```

Index

- * **api**
 - cie11_search, 5
- * **busqueda**
 - cie_guia_busqueda, 8
 - cie_lookup, 8
 - cie_search, 11
 - cie_siglas, 13
- * **comorbilidades**
 - cie_comorbid, 6
 - cie_map_comorbid, 10
- * **datasets**
 - cie10_cl, 2
- * **sql**
 - cie10_clear_cache, 3
 - cie10_disconnect, 4
 - cie10_sql, 4
- * **validacion**
 - cie_expand, 7
 - cie_normalizar, 10
 - cie_validate_vector, 14
- * **visualizacion**
 - cie_table, 14

- cie10_cl, 2
- cie10_clear_cache, 3, 4, 5
- cie10_disconnect, 3, 4, 5
- cie10_sql, 3, 4, 4, 12
- cie11_search, 5
- cie_comorbid, 6, 10
- cie_expand, 7, 9, 11, 15
- cie_guia_busqueda, 8, 9, 12, 13
- cie_lookup, 6–8, 8, 11–14
- cie_map_comorbid, 6, 10
- cie_normalizar, 6, 7, 9, 10, 10, 15
- cie_search, 5, 6, 8, 9, 11, 13, 14
- cie_siglas, 8, 9, 12, 13
- cie_table, 14
- cie_validate_vector, 7, 11, 14