# Package 'extractox'

December 3, 2024

**Title** Extract Tox Info from Various Databases

**Version** 0.1.0

**Description** Extract toxicological and chemical information from databases maintained by scientific agencies and resources, including the Comparative Toxicogenomics Database <https://ctdbase.org/>, the Integrated Chemical Environment <https://ice.ntp.niehs.nih.gov/>, the Integrated Risk Information System <https://cfpub.epa.gov/ncea/iris/>, the CompTox Chemicals Dashboard Resource Hub <https://www.epa.gov/comptox-tools/comptox-chemicals-dashboard-resource-hub>, and PubChem <https://pubchem.ncbi.nlm.nih.gov/>.

**License** MIT + file LICENSE

**URL** https://github.com/c1au6i0/extractox, https://c1au6i0.github.io/extractox/

**BugReports** https://github.com/c1au6i0/extractox/issues

**Depends** R (>= 4.1)

**Imports** cli, httr2, janitor, pingr, readxl, rvest, webchem

**Suggests** openxlsx, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Claudio Zanettini [aut, cre, cph] (<https://orcid.org/0000-0001-5043-8033>), Lucio Queiroz [aut] (<https://orcid.org/0000-0002-6090-1834>)

**Maintainer** Claudio Zanettini <claudio.zanettini@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-12-03 21:40:02 UTC

# Contents



---

`extr_casrn_from_cid` *Retrieve CASRN for PubChem CIDs*

---

## Description

This function retrieves the CASRN for a given set of PubChem Compound Identifiers (CID). It queries PubChem through the `webchem` package and extracts the CASRN from the depositor-supplied synonyms.

## Usage

```
extr_casrn_from_cid(pubchem_id)
```

## Arguments

| | |
|---|---|
| `pubchem_id` | A numeric vector of PubChem CIDs. These are unique identifiers for chemical compounds in the PubChem database. |

## Value

A data frame containing the CID, CASRN, and IUPAC name of the compound. The returned data frame includes three columns:

**CID** The PubChem Compound Identifier.

**cas_rn** The corresponding CASRN of the compound.

**IUPACName** The IUPAC name of the compound.

## See Also

PubChem

## Examples

```
# Example with formaldehyde and aflatoxin
cids <- c(712, 14434) # CID for formaldehyde and aflatoxin B1
extr_casrn_from_cid(cids)
```

---

`extr_chem_info` *Query Chemical Information from IUPAC Names*

---

## Description

This function takes a vector of IUPAC names and queries the PubChem database (using the `webchem` package) to obtain the corresponding CASRN and CID for each compound. It reshapes the resulting data, ensuring that each compound has a unique row with the CID, CASRN, and additional chemical properties.

## Usage

```
extr_chem_info(IUPAC_names, stop_on_warning = FALSE)
```

## Arguments

| | |
|---|---|
| `IUPAC_names` | A character vector of IUPAC names. These are standardized names of chemical compounds that will be used to search in the PubChem database. |
| `stop_on_warning` | Logical. If set to TRUE, the function will stop and throw an error if any substances are not found in PubChem. Defaults to FALSE, in which case a warning is issued. |

## Value

A data frame with information on the queried compounds, including:

**iupac_name** The IUPAC name of the compound.

**cid** The PubChem Compound Identifier (CID).

**isomeric_smiles** The SMILES string (Simplified Molecular Input Line Entry System).

## Examples

```
# Example with formaldehyde and aflatoxin
extr_chem_info(IUPAC_names = c("Formaldehyde", "Aflatoxin B1"))
```

extr_comptox *Download and Extract Data from CompTox Chemistry Dashboard*

## Description

This function interacts with the CompTox Chemistry Dashboard to download and extract a wide range of chemical data based on user-defined search criteria. It allows for flexible input types and supports downloading various chemical properties, identifiers, and predictive data. It was inspired by the `ECOTOXr::websearch_comptox` function.

## Usage

```
extr_comptox(
  ids,
  download_items = c("DTXCID", "CASRN", "INCHIKEY", "IUPAC_NAME", "SMILES",
    "INCHI_STRING", "MS_READY_SMILES", "QSAR_READY_SMILES", "MOLECULAR_FORMULA",
    "AVERAGE_MASS", "MONOISOTOPIC_MASS", "QC_LEVEL", "SAFETY_DATA", "EXPOCAST",
   "DATA_SOURCES", "TOXVAL_DATA", "NUMBER_OF_PUBMED_ARTICLES", "PUBCHEM_DATA_SOURCES",
    "CPDAT_COUNT", "IRIS_LINK", "PPRTV_LINK", "WIKIPEDIA_ARTICLE", "QC_NOTES",
   "ABSTRACT_SHIFTER", "TOXPRINT_FINGERPRINT", "ACTOR_REPORT", "SYNONYM_IDENTIFIER",
    "RELATED_RELATIONSHIP", "ASSOCIATED_TOXCAST_ASSAYS",
     "TOXVAL_DETAILS",
    "CHEMICAL_PROPERTIES_DETAILS", "BIOCONCENTRATION_FACTOR_TEST_PRED",
    "BOILING_POINT_DEGC_TEST_PRED", "48HR_DAPHNIA_LC50_MOL/L_TEST_PRED",
    "DENSITY_G/CM^3_TEST_PRED", "DEVTOX_TEST_PRED",
    "96HR_FATHEAD_MINNOW_MOL/L_TEST_PRED", "FLASH_POINT_DEGC_TEST_PRED",
    "MELTING_POINT_DEGC_TEST_PRED", "AMES_MUTAGENICITY_TEST_PRED",
    "ORAL_RAT_LD50_MOL/KG_TEST_PRED", "SURFACE_TENSION_DYN/CM_TEST_PRED",
    "THERMAL_CONDUCTIVITY_MW/(M*K)_TEST_PRED",
   "TETRAHYMENA_PYRIFORMIS_IGC50_MOL/L_TEST_PRED", "VISCOSITY_CP_CP_TEST_PRED",

    "VAPOR_PRESSURE_MMHG_TEST_PRED", "WATER_SOLUBILITY_MOL/L_TEST_PRED",
    "ATMOSPHERIC_HYDROXYLATION_RATE_(AOH)_CM3/MOLECULE*SEC_OPERA_PRED",
    "BIOCONCENTRATION_FACTOR_OPERA_PRED",
   "BIODEGRADATION_HALF_LIFE_DAYS_DAYS_OPERA_PRED", "BOILING_POINT_DEGC_OPERA_PRED",
    "HENRYS_LAW_ATM-M3/MOLE_OPERA_PRED", "OPERA_KM_DAYS_OPERA_PRED",
    "OCTANOL_AIR_PARTITION_COEFF_LOGKOA_OPERA_PRED",
    "SOIL_ADSORPTION_COEFFICIENT_KOC_L/KG_OPERA_PRED",
    "OCTANOL_WATER_PARTITION_LOGP_OPERA_PRED", "MELTING_POINT_DEGC_OPERA_PRED",

   "OPERA_PKAA_OPERA_PRED", "OPERA_PKAB_OPERA_PRED", "VAPOR_PRESSURE_MMHG_OPERA_PRED",
    "WATER_SOLUBILITY_MOL/L_OPERA_PRED",
    "EXPOCAST_MEDIAN_EXPOSURE_PREDICTION_MG/KG-BW/DAY", "NHANES",
    "TOXCAST_NUMBER_OF_ASSAYS/TOTAL", "TOXCAST_PERCENT_ACTIVE"),
  mass_error = 0,
  verify_ssl = FALSE,
  ...
```

)

## Arguments

| | |
|---|---|
| ids | A character vector containing the items to be searched within the CompTox Chemistry Dashboard. These can be chemical names, CAS Registry Numbers (CASRN), InChIKeys, or DSSTox substance identifiers (DTXSID). |
| download_items | A character vector of items to be downloaded. This includes a comprehensive set of chemical properties, identifiers, predictive data, and other relevant information. By Default, it download all the info |

**DTXCID** The unique identifier for a chemical in the EPA's CompTox Chemicals Dashboard.

**CASRN** The Chemical Abstracts Service Registry Number, a unique numerical identifier for chemical substances.

**INCHIKEY** The hashed version of the full International Chemical Identifier (InChI) string.

**IUPAC_NAME** The International Union of Pure and Applied Chemistry (IUPAC) name of the chemical.

**SMILES** The Simplified Molecular Input Line Entry System (SMILES) representation of the chemical structure.

**INCHI_STRING** The full International Chemical Identifier (InChI) string.

**MS_READY_SMILES** The SMILES representation of the chemical structure, prepared for mass spectrometry analysis.

**QSAR_READY_SMILES** The SMILES representation of the chemical structure, prepared for quantitative structure-activity relationship (QSAR) modeling.

**MOLECULAR_FORMULA** The chemical formula representing the number and type of atoms in a molecule.

**AVERAGE_MASS** The average mass of the molecule, calculated based on the isotopic distribution of the elements.

**MONOISOTOPIC_MASS** The mass of the molecule calculated using the most abundant isotope of each element.

**QC_LEVEL** The quality control level of the data.

**SAFETY_DATA** Safety information related to the chemical.

**EXPOCAST** Exposure predictions from the EPA's ExpoCast program.

**DATA_SOURCES** Sources of the data provided.

**TOXVAL_DATA** Toxicological values related to the chemical.

**NUMBER_OF_PUBMED_ARTICLES** The number of articles related to the chemical in PubMed.

**PUBCHEM_DATA_SOURCES** Sources of data from PubChem.

**CPDAT_COUNT** The number of entries in the Chemical and Product Categories Database (CPDat).

**IRIS_LINK** Link to the EPA's Integrated Risk Information System (IRIS) entry for the chemical.

**PPRTV_LINK** Link to the EPA's Provisional Peer-Reviewed Toxicity Values (PPRTV) entry for the chemical.

**WIKIPEDIA_ARTICLE** Link to the Wikipedia article for the chemical.

**QC_NOTES** Notes related to the quality control of the data.

**ABSTRACT_SHIFTER** Information related to the abstract shifter.

**TOXPRINT_FINGERPRINT** The ToxPrint chemoinformatics fingerprint of the chemical.

**ACTOR_REPORT** The Aggregated Computational Toxicology Resource (ACTOR) report for the chemical.

**SYNONYM_IDENTIFIER** Identifiers for synonyms of the chemical.

**RELATED_RELATIONSHIP** Information on related chemicals.

**ASSOCIATED_TOXCAST_ASSAYS** Assays associated with the chemical in the ToxCast database.

**TOXVAL_DETAILS** Details of toxicological values.

**CHEMICAL_PROPERTIES_DETAILS** Details of the chemical properties.

**BIOCONCENTRATION_FACTOR_TEST_PRED** Predicted bioconcentration factor from tests.

**BOILING_POINT_DEGC_TEST_PRED** Predicted boiling point in degrees Celsius from tests.

**48HR_DAPHNIA_LC50_MOL/L_TEST_PRED** Predicted 48-hour LC50 for Daphnia in mol/L from tests.

**DENSITY_G/CM^3_TEST_PRED** Predicted density in g/cm³ from tests.

**DEVTOX_TEST_PRED** Predicted developmental toxicity from tests.

**96HR_FATHEAD_MINNOW_MOL/L_TEST_PRED** Predicted 96-hour LC50 for fathead minnow in mol/L from tests.

**FLASH_POINT_DEGC_TEST_PRED** Predicted flash point in degrees Celsius from tests.

**MELTING_POINT_DEGC_TEST_PRED** Predicted melting point in degrees Celsius from tests.

**AMES_MUTAGENICITY_TEST_PRED** Predicted Ames mutagenicity from tests.

**ORAL_RAT_LD50_MOL/KG_TEST_PRED** Predicted oral LD50 for rats in mol/kg from tests.

**SURFACE_TENSION_DYN/CM_TEST_PRED** Predicted surface tension in dyn/cm from tests.

**THERMAL_CONDUCTIVITY_MW_M×K_TEST_PRED** Predicted thermal conductivity in mW/m×K from tests.

**TETRAHYMENA_PYRIFORMIS_IGC50_MOL/L_TEST_PRED** Predicted IGC50 for Tetrahymena pyriformis in mol/L from tests.

**VISCOSITY_CP_CP_TEST_PRED** Predicted viscosity in cP from tests.

**VAPOR_PRESSURE_MMHG_TEST_PRED** Predicted vapor pressure in mmHg from tests.

**WATER_SOLUBILITY_MOL/L_TEST_PRED** Predicted water solubility in mol/L from tests.

**ATMOSPHERIC_HYDROXYLATION_RATE_\(AOH\)_CM3/MOLECULE*SEC_OPERA_PRE** Predicted atmospheric hydroxylation rate in cm³/molecule*sec from OPERA.

**BIOCONCENTRATION_FACTOR_OPERA_PRED** Predicted bioconcentration factor from OPERA.

**BIODEGRADATION_HALF_LIFE_DAYS_DAYS_OPERA_PRED** Predicted biodegradation half-life in days from OPERA.

**BOILING_POINT_DEGC_OPERA_PRED** Predicted boiling point in degrees Celsius from OPERA.

**HENRYS_LAW_ATM-M3/MOLE_OPERA_PRED** Predicted Henry's law constant in atm-m³/mole from OPERA.

**OPERA_KM_DAYS_OPERA_PRED** Predicted Km in days from OPERA.

**OCTANOL_AIR_PARTITION_COEFF_LOGKOA_OPERA_PRED** Predicted octanol-air partition coefficient (log Koa) from OPERA.

**SOIL_ADSORPTION_COEFFICIENT_KOC_L/KG_OPERA_PRED** Predicted soil adsorption coefficient (Koc) in L/kg from OPERA.

**OCTANOL_WATER_PARTITION_LOGP_OPERA_PRED** Predicted octanol-water partition coefficient (log P) from OPERA.

**MELTING_POINT_DEGC_OPERA_PRED** Predicted melting point in degrees Celsius from OPERA.

**OPERA_PKAA_OPERA_PRED** Predicted pKa (acidic) from OPERA.

**OPERA_PKAB_OPERA_PRED** Predicted pKa (basic) from OPERA.

**VAPOR_PRESSURE_MMHG_OPERA_PRED** Predicted vapor pressure in mmHg from OPERA.

**WATER_SOLUBILITY_MOL/L_OPERA_PRED** Predicted water solubility in mol/L from OPERA.

**EXPOCAST_MEDIAN_EXPOSURE_PREDICTION_MG/KG-BW/DAY** Predicted median exposure from ExpoCast in mg/kg-bw/day.

**NHANES** National Health and Nutrition Examination Survey data.

**TOXCAST_NUMBER_OF_ASSAYS/TOTAL** Number of assays in ToxCast.

**TOXCAST_PERCENT_ACTIVE** Percentage of active assays in ToxCast.

`mass_error` Numeric value indicating the mass error tolerance for searches involving mass data. Default is `0`.

`verify_ssl` Logical value indicating whether SSL certificates should be verified. Default is `FALSE`. Note that this argument is not used on linux OS.

`...` Additional arguments passed to `httr2::req_options()`. Note that this argument is not used on linux OS.

## Details

Please note that this function, which pulls data from EPA servers, may encounter issues on some Linux systems. This is because those servers do not accept secure legacy renegotiation. On Linux systems, the current function depends on `curl` and `OpenSSL`, which have known problems with unsafe legacy renegotiation in newer versions. One workaround is to downgrade to `curl v7.78.0` and `OpenSSL v1.1.1`. However, please be aware that using these older versions might introduce potential security vulnerabilities. Refer to this gist for instructions on how to downgrade `curl` and `OpenSSL` on Ubuntu.

## Value

A cleaned data frame containing the requested data from CompTox.

## See Also

CompTox Chemicals Dashboard Resource Hub

## Examples

```
# Example usage of the function:
extr_comptox(ids = c("Aspirin", "50-00-0"))
```

---

`extr_ctd` *Extract Data from the CTD API*

---

## Description

This function queries the Comparative Toxicogenomics Database API to retrieve data related to chemicals, diseases, genes, or other categories.

## Usage

```
extr_ctd(
  input_terms,
  category = "chem",
  report_type = "genes_curated",
  input_term_search_type = "directAssociations",
  action_types = NULL,
  ontology = NULL,
  verify_ssl = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| `input_terms` | A character vector of input terms such as CAS numbers or IUPAC names. |
| `category` | A string specifying the category of data to query. Valid options are "all", "chem", "disease", "gene", "go", "pathway", "reference", and "taxon". Default is "chem". |
| `report_type` | A string specifying the type of report to return. Default is "genes_curated". Valid options include:<br>**"cgixns"** Curated chemical-gene interactions. Requires at least one `action_types` parameter.<br>**"chems"** All chemical associations.<br>**"chems_curated"** Curated chemical associations.<br>**"chems_inferred"** Inferred chemical associations. |

**"genes"** All gene associations.

**"genes_curated"** Curated gene associations.

**"genes_inferred"** Inferred gene associations.

**"diseases"** All disease associations.

**"diseases_curated"** Curated disease associations.

**"diseases_inferred"** Inferred disease associations.

**"pathways_curated"** Curated pathway associations.

**"pathways_inferred"** Inferred pathway associations.

**"pathways_enriched"** Enriched pathway associations.

**"phenotypes_curated"** Curated phenotype associations.

**"phenotypes_inferred"** Inferred phenotype associations.

**"go"** All Gene Ontology (GO) associations. Requires at least one `ontology` parameter.

**"go_enriched"** Enriched GO associations. Requires at least one `ontology` parameter.

`input_term_search_type`
: A string specifying the search method to use. Options are "hierarchicalAssociations" or "directAssociations". Default is "directAssociations".

`action_types`
: An optional character vector specifying one or more interaction types for filtering results. Default is "ANY". Other acceptable inputs are "abundance", "activity", "binding", "cotreatment", "expression", "folding", "localization", "metabolic processing"...See https://ctdbase.org/tools/batchQuery.go for a full list.

`ontology`
: An optional character vector specifying one or more ontologies for filtering GO reports. Default NULL.

`verify_ssl`
: Boolean to control of SSL should be verified or not.

`...`
: Any other arguments to be supplied to `req_option` and thus to `libcurl`.

## Value

A data frame containing the queried data in CSV format.

## References


- Davis, A. P., Grondin, C. J., Johnson, R. J., Sciaky, D., McMorran, R., Wiegers, T. C., & Mattingly, C. J. (2019). The Comparative Toxicogenomics Database: update 2019. Nucleic acids research, 47(D1), D948–D954. doi:10.1093/nar/gky868


## See Also

Comparative Toxicogenomics Database

## Examples

```
input_terms <- c("50-00-0", "64-17-5", "methanal", "ethanol")
dat <- extr_ctd(
  input_terms = input_terms,
  category = "chem",
```

```
  report_type = "genes_curated",
  input_term_search_type = "directAssociations",
  action_types = "ANY",
  ontology = c("go_bp", "go_cc")
)
str(dat)

# Get expresssion data
dat2 <- extr_ctd(
  input_terms = input_terms,
  report_type = "cgixns",
  category = "chem",
  action_types = "expression"
)

str(dat2)
```

---

`extr_ice` *Extract Data from NTP ICE Database*

---

## Description

The `extr_ice` function sends a POST request to the ICE API to search for information based on specified chemical IDs and assays.

## Usage

```
extr_ice(casrn, assays = NULL, verify_ssl = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `casrn` | A character vector specifying the CASRNs for the search. |
| `assays` | A character vector specifying the assays to include in the search. Default is NULL, meaning all assays are included. |
| `verify_ssl` | Boolean to control of SSL should be verified or not. |
| `...` | Any other arguments to be supplied to `req_option` and thus to `libcurl`. |

## Value

A data frame containing the extracted data from the ICE API.

## See Also

NTP ICE database

## Examples

```
extr_ice(c("50-00-0"))
```

## extr_iris *Extract Data from EPA IRIS Database*

### Description

The `extr_iris` function sends a request to the EPA IRIS database to search for information based on a specified keywords and cancer types. It retrieves and parses the HTML content from the response. Note that if `keywords` is not provide all dataset are retrieved.

### Usage

```
extr_iris(casrn = NULL, cancer_types = c("non_cancer", "cancer"))
```

### Arguments

| | |
|---|---|
| `casrn` | A single character string specifying the CASRN for the search. |
| `cancer_types` | A character vector specifying the types of cancer to include in the search. Must be either "non_cancer" or "cancer". |

### Value

A data frame containing the extracted data.

### See Also

EPA IRIS database

### Examples

```
extr_iris(c("1332-21-4", "50-00-0"))
```

## extr_pubchem_fema *Extract FEMA from PubChem*

### Description

This function retrieves FEMA (Flavor and Extract Manufacturers Association) flavor profile information for a list of CAS Registry Numbers (CASRN) from the PubChem database using the `webchem` package. It applies the function `extr_fema_pubchem_` to each CASRN in the input vector and combines the results into a single data frame.

### Usage

```
extr_pubchem_fema(casrn)
```

### Arguments

| | |
|---|---|
| casrn | A vector of CAS Registry Numbers (CASRN) as atomic vectors. |

### Value

A data frame containing the FEMA flavor profile information for each CASRN. If no information is found for a particular CASRN, the output will include a row indicating this.

### See Also

PubChem

### Examples

```
extr_pubchem_fema(c("64-17-5", "50-00-0"))
```

---

## extr_pubchem_ghs *Extract GHS Codes from PubChem*

---

### Description

This function extracts GHS (Globally Harmonized System) codes from PubChem. It relies on the webchem package to interact with PubChem.

### Usage

```
extr_pubchem_ghs(casrn)
```

### Arguments

| | |
|---|---|
| casrn | Character vector of CAS Registry Numbers (CASRN). |

### Value

A dataframe containing GHS information.

### See Also

PubChem

### Examples

```
extr_pubchem_ghs(casrn = c("50-00-0", "64-17-5"))
```

`extr_tetramer` *Extract Tetramer Data from the CTD API*

## Description

This function queries the Comparative Toxicogenomics Database API to retrieve tetramer data based on chemicals, diseases, genes, or other categories.

## Usage

```
extr_tetramer(
  chem,
  disease = "",
  gene = "",
  go = "",
  input_term_search_type = "directAssociations",
  qt_match_type = "equals",
  verify_ssl = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| `chem` | A string indicating the chemical identifiers such as CAS number or IUPAC name of the chemical. |
| `disease` | A string indicating a disease term. Default is an empty string. |
| `gene` | A string indicating a gene symbol. Default is an empty string. |
| `go` | A string indicating a Gene Ontology term. Default is an empty string. |
| `input_term_search_type` | A string specifying the search method to use. Options are "hierarchicalAssociations" or "directAssociations". Default is "directAssociations". |
| `qt_match_type` | A string specifying the query type match method. Options are "equals" or "contains". Default is "equals". |
| `verify_ssl` | Boolean to control if SSL should be verified or not. Default is FALSE. |
| `...` | Any other arguments to be supplied to `req_option` and thus to `libcurl`. |

## Value

A data frame containing the queried tetramer data in CSV format.

## References


- Comparative Toxicogenomics Database: http://ctdbase.org
- Davis, A. P., Grondin, C. J., Johnson, R. J., Sciaky, D., McMorran, R., Wiegers, T. C., & Mattingly, C. J. (2019). The Comparative Toxicogenomics Database: update 2019. Nucleic acids research, 47(D1), D948–D954. doi:10.1093/nar/gky868

- Davis, A. P., Wiegers, T. C., Wiegers, J., Wyatt, B., Johnson, R. J., Sciaky, D., Barkalow, F., Strong, M., Planchart, A., & Mattingly, C. J. (2023). CTD tetramers: A new online tool that computationally links curated chemicals, genes, phenotypes, and diseases to inform molecular mechanisms for environmental health. Toxicological Sciences, 195(2), 155–168. doi:10.1093/toxsci/kfad069


## See Also

Comparative Toxicogenomics Database

## Examples

```
tetramer_data <- extr_tetramer(
  chem = c("50-00-0", "ethanol"),
  disease = "",
  gene = "",
  go = "",
  input_term_search_type = "directAssociations",
  qt_match_type = "equals"
)
str(tetramer_data)
```

---

# extr_tox *Extract Toxicological Information from Multiple Databases*

---

## Description

This wrapper function retrieves toxicological information for specified chemicals by calling several external functions to query multiple databases, including PubChem, the Integrated Chemical Environment (ICE), CompTox Chemicals Dashboard, and the Integrated Risk Information System (IRIS).

## Usage

```
extr_tox(casrn)
```

## Arguments

| | |
|---|---|
| casrn | A character vector of CAS Registry Numbers (CASRN) representing the chemicals of interest. |

## Details

Specifically, this function:

- Calls extr_pubchem_ghs to retrieve GHS classification data from PubChem.
- Calls extr_comptox to retrieve data from the CompTox Chemicals Dashboard.
- Calls extr_ice to gather assay data from the ICE database.
- Calls extr_iris to retrieve risk assessment information from the IRIS database.

## Value

A list of data frames containing toxicological information retrieved from each database:

**ghs_dat** Toxicity data from PubChem's Globally Harmonized System (GHS) classification.

**comptox_list** List of toxicity information from the CompTox Chemicals Dashboard.

**ice_dat** Assay data from the Integrated Chemical Environment (ICE) database.

**iris** Risk assessment data from the IRIS database.

## Examples

```
casrn <- c("50-00-0", "107-02-8")
extr_tox(casrn)
```

---

# write_dataframes_to_excel

*Write Dataframes to Excel*

---

## Description

This function creates an Excel file with each dataframe in a list as a separate sheet.

## Usage

```
write_dataframes_to_excel(df_list, filename)
```

## Arguments

| | |
|---|---|
| df_list | A named list of dataframes to write to the Excel file. |
| filename | The name of the Excel file to create. |

## Value

No return value. The function prints a message indicating the completion of the Excel file writing.

## Examples

```
tox_dat <- extr_tox("50-00-0")
temp_file <- tempfile(fileext = ".xlsx")
write_dataframes_to_excel(tox_dat, filename = temp_file)
```

# Index