Package: craftgrn (via r-universe)

June 19, 2026

Title Integrative Chromatin Accessibility and RNA Framework for Gene Regulatory Networks

Version 0.1.4

Description Provides a reproducible framework for constructing and comparing gene regulatory networks by integrating chromatin accessibility footprint scores with matched RNA expression data. It implements context-specific enhancer-gene linking, transcription factor focused network analysis, differential network analysis, and regulatory topic modeling workflows for systematic exploration of gene regulation across conditions.

License GPL (>= 3)

Depends R (>= 4.1.0)

Imports cli, cluster, config, data.table, dplyr, digest, future, future.apply, ggplot2, enrichR, jsonlite, LDAvis, methods, pheatmap, readr, Rcpp, tibble, tidyr, yaml

LinkingTo Rcpp

Suggests AnnotationDbi, arrow, EnsDb.Hsapiens.v86, EnsDb.Mmusculus.v79, fgsea, golem, ggraph, ggrepel, gtable, gridExtra, htmlwidgets, igraph, Matrix, msigdbr, parallelly, progressr, RColorBrewer, knitr, rmarkdown, rstudioapi, scales, shiny, spelling, testthat, withr

VignetteBuilder knitr

SystemRequirements Optional Shiny app support on Linux may require libuv headers, for example libuv1-dev, or cmake for bundled libuv builds in transitive dependencies.

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

URL https://oncologylab.github.io/craftgrn/, https://github.com/oncologylab/craftgrn

BugReports https://github.com/oncologylab/craftgrn/issues

Config/Needs/website r-lib/pkgdown, tidyverse/tidytemplate

Config/pak/sysreqs cmake libicu-dev libuv1-dev libssl-dev perl libx11-dev

Repository https://oncologylab.r-universe.dev

Date/Publication 2026-06-10 00:08:17 UTC

RemoteUrl https://github.com/oncologylab/craftgrn

RemoteRef HEAD

RemoteSha 881ddec9ac09276bc801ec86b331fea4e23508c5

Contents

`build_module1_qc_report` *Build a Module 1 QC HTML report*

Description

Builds a comprehensive HTML report for Module 1 run parameters, input gates, motif-supported canonical support, prediction output integrity, correlation diagnostics, condition-level CraftGRN multiomic input QC, footprint alignment summaries, warning checks, and related QC artifacts. The report can consume a ‘predict_tfbs()‘ result, a step-by-step Module 1 result list, or a Module 1 output directory.

Usage

```
build_module1_qc_report(
  module1,
  omics_data = NULL,
  output_dir = NULL,
  report_name = "module1_qc_report.html",
  scan_predicted_tfbs = TRUE,
  top_n = 20L,
  verbose = TRUE
)
```

Arguments

`module1`	Module 1 result list or Module 1 output directory.
`omics_data`	Optional CraftGRN multiomic object. Used when ‘module1‘ is an output directory or does not contain ‘omics_data‘.
`output_dir`	Directory where the HTML report should be written. If ‘NULL‘, the report is written under ‘reports‘ inside the Module 1 output directory when available.
`report_name`	HTML report filename.
`scan_predicted_tfbs`	Logical; if ‘TRUE‘, scan predicted TFBS chunks to summarize top TFs and condition support. This is comprehensive but can take extra time on full projects.
`top_n`	Number of TFs to show in top-TF summaries.
`verbose`	Emit concise progress messages.

Value

Normalized path to the HTML report.

`build_module2_qc_report` *Build a Module 2 QC HTML report*

Description

Builds a comprehensive HTML report for Module 2 run parameters, CraftGRN multiomic input handoff, TF-target and FP-target correlation filters, candidate source and distance-to-TSS evidence, final TF-FP-target links, condition activity, CraftGRN multiomic condition context, warning checks, integrity checks, and related browser reports.

Usage

```
build_module2_qc_report(
  module2,
  multiomic_data = NULL,
  output_dir = NULL,
  report_name = "module2_qc_report.html",
  scan_large_tables = TRUE,
  validate_integrity = TRUE,
  top_n = 20L,
  verbose = TRUE
)
```

Arguments

`module2`	Module 2 result list, loaded Module 2 list, or output directory.
`multiomic_data`	Optional CraftGRN multiomic object used for context.
`output_dir`	Directory where the HTML report should be written. If 'NULL', the report is written under 'reports' inside the Module 2 output directory when available.
`report_name`	HTML report filename.
`scan_large_tables`	Logical; if 'TRUE', scan candidate and link chunks for top-TF, distance, and integrity summaries.
`validate_integrity`	Logical; if 'TRUE', verify final links against passing TF-target and FP-target keys while scanning link chunks.
`top_n`	Number of TFs to show in top-TF summaries.
`verbose`	Emit concise progress messages.

Value

Normalized path to the HTML report.

`build_module3_qc_report` *Build a Module 3 QC HTML report*

Description

Writes a self-contained HTML report for Module 3 topic-model outputs. The report summarizes topic-input caches, model rows, theta separation scores, compact topic-link pass counts, and differential-link summaries when available.

Usage

```
build_module3_qc_report(
  topic_dir,
  output_dir = file.path(topic_dir, "reports"),
  differential_links_dir = NULL,
  title = "Module 3 QC report",
  top_n = 20L,
  verbose = TRUE
)
```

Arguments

`topic_dir`	Module 3 topic output directory.
`output_dir`	Directory where the report is written. Defaults to 'topic_dir/reports'.
`differential_links_dir`	Optional Module 3 differential-link directory. If 'NULL', CraftGRN tries to detect a sibling or nested 'differential_links' directory.
`title`	Report title.
`top_n`	Number of top differential TFs retained per comparison in the QC summary CSV.
`verbose`	Emit concise progress messages.

Value

Path to the HTML report.

check_predicted_links *Perform sanity check for predicted links for Module 2 diagnostics*

Description

Perform sanity check for predicted links for Module 2 diagnostics

Usage

```
check_predicted_links(module2)
```

Arguments

module2	Module 2 result list or loaded output list.

Value

TRUE invisibly when valid.

craftgrn_demo_data_info *Return metadata for configured external CraftGRN demo data*

Description

Return metadata for configured external CraftGRN demo data

Usage

```
craftgrn_demo_data_info(demo = NULL)
```

Arguments

demo	Optional demo bundle name. No external demo bundle is currently configured.

Value

A data frame with the bundle URL, checksum, archive file name, and extracted project directory name. When no demo bundle is configured, the returned data frame has zero rows.

`download_craftgrn_demo_data` *Download and unpack configured external CraftGRN demo data*

Description

Downloads a processed demo data archive from a configured external source, verifies its MD5 checksum by default, extracts it, and returns the extracted project directory. Demo bundles are external to the R package so package installation remains small and CRAN-friendly. No external demo bundle is currently configured.

Usage

```
download_craftgrn_demo_data(
  destdir = ".",
  demo = NULL,
  overwrite = FALSE,
  checksum = TRUE,
  verbose = TRUE
)
```

Arguments

`destdir`	Directory where the archive should be downloaded and unpacked.
`demo`	Optional demo bundle name. No external demo bundle is currently configured.
`overwrite`	Logical; if 'TRUE', download the archive again and replace an existing extracted project directory.
`checksum`	Logical; if 'TRUE', verify the downloaded archive MD5.
`verbose`	Logical; if 'TRUE', emit concise status messages.

Details

If the download fails, inspect 'craftgrn_demo_data_info()' and download the configured asset manually. If checksum verification fails, rerun with 'overwrite = TRUE' to replace a stale or partial archive. The extracted project uses 'base_dir: "."', so pass the returned directory or its project config path directly to package functions after moving the folder.

Value

The normalized path to the extracted demo project directory.

Examples

```
craftgrn_demo_data_info()
```

export_predicted_tfbs_bed

Export predicted TFBS as BED files

Description

Export predicted TFBS as BED files

Usage

```
export_predicted_tfbs_bed(
  predicted_tfbs,
  out_file = NULL,
  out_dir = NULL,
  tf = NULL,
  split_by = c("none", "tf")
)
```

Arguments

predicted_tfbs	Compact predicted TFBS table or path.
out_file	BED output path. Required when split_by is none.
out_dir	Directory for split BED outputs.
tf	Optional TF subset.
split_by	One of none or tf.

Value

Output path or manifest tibble, invisibly.

export_tf_target_bedpe

Export predicted TF-target links as BEDPE

Description

Export predicted TF-target links as BEDPE

Usage

```
export_tf_target_bedpe(module2, output_file, tf = NULL)
```

Arguments

module2	Module 2 result list or loaded output list.
output_file	BEDPE output file.
tf	Optional TF subset.

Value

Output path invisibly.

---

load_config *Load a CraftGRN YAML config into an environment*

Description

Reads a YAML file and assigns each top-level key as a variable in the target environment (e.g., 'db', 'threshold_tf_expr', etc.). Also runs standard config initialization helpers when available.

Usage

```
load_config(path, env = .craftgrn_state)
```

Arguments

path	Character path to a YAML file.
env	Environment to populate. Defaults to the internal CraftGRN config state.

Value

(Invisibly) the parsed list.

Examples

```
## Not run:
load_config("craftgrn_grn.yaml")
# Config values are now available to CraftGRN helper functions.

## End(Not run)
```

load_omics_data *Load a multi-omic data object from disk*

Description

Load a multi-omic data object from disk

Usage

```
load_omics_data(file, verbose = TRUE)
```

Arguments

file	Path to an RDS file produced by save_omics_data().
verbose	Emit status messages.

Value

The loaded multi-omic data list.

load_predicted_links *Load predicted links from Module 2*

Description

Load predicted links from Module 2

Usage

```
load_predicted_links(path)
```

Arguments

path	Module 2 output directory or module2_manifest.csv path.

Value

A named list of Module 2 tables.

load_predicted_tfbs *Load TFBS predicted from Module 1*

Description

Load TFBS predicted from Module 1

Usage

```
load_predicted_tfbs(path)
```

Arguments

path	Path to a predicted TFBS manifest, Parquet file, or CSV file.

Value

A tibble.

load_prep_multiomic_data *Load and prepare the Module 1 multi-omic object*

Description

Build the rebuilt Module 1 data object from cached aligned footprints or from raw footprint overview files plus ATAC, RNA, and sample metadata inputs. The returned object is the canonical input for downstream Step 1 TFBS correlation.

Usage

```
load_prep_multiomic_data(
  config = NULL,
  genome = NULL,
  gene_symbol_col = "HGNC",
  fp_aligned = NULL,
  do_preprocess = FALSE,
  do_motif_clustering = FALSE,
  trim_hocomoco = FALSE,
  fp_root_dir = NULL,
  fp_cache_dir = NULL,
  fp_cache_tag = NULL,
  footprint_sample_scope = "metadata",
  mid_slop = 10L,
  round_digits = 1L,
```

```
  score_match_pct = 0.8,
  output_mode = c("full", "distinct"),
  write_outputs = FALSE,
  write_fp_score_qn_csv = TRUE,
  atac_data = NULL,
  rna_tbl = NULL,
  metadata = NULL,
  atac_data_path = NULL,
  rna_path = NULL,
  metadata_path = NULL,
  step1_out_dir_name = "predict_tf_binding_sites",
  label_col,
  expected_n = NULL,
  tf_list = NULL,
  motif_db = NULL,
  threshold_gene_expr = NULL,
  threshold_fp_score = NULL,
  use_parallel = TRUE,
  verbose = TRUE,
  time_log = verbose
)
```

Arguments

config	Optional YAML config path.
genome	Optional genome string used to override the config value.
gene_symbol_col	Gene-symbol column in the RNA table.
fp_aligned	Optional pre-aligned footprint object.
do_preprocess	Logical; if 'TRUE', load and align raw footprints before building the object. If 'FALSE', use cached aligned footprints.
do_motif_clustering	Logical; if 'TRUE', run motif clustering during preprocessing when available.
trim_hocomoco	Logical; trim HOCOMOCO manifests when the trimming helper is available.
fp_root_dir	Optional root directory for raw footprint overview files.
fp_cache_dir	Cache directory for aligned footprint files.
fp_cache_tag	Cache tag, typically the motif database name.
footprint_sample_scope	Footprint sample selection rule.
mid_slop, round_digits, score_match_pct	Alignment parameters passed to 'align_footprints()'.
output_mode	Output mode for aligned footprints. One of '"full"' or '"distinct"'.
write_outputs	Logical; if 'TRUE', save the prepared object as an RDS cache under 'predict_tf_binding_sites/'.

`write_fp_score_qn_csv`
: Logical; if 'TRUE' and 'write_outputs = TRUE', also save quantile-normalized footprint scores as '01_fp_scores_qn_<db>.csv' under the Module 1 output directory.

`atac_data`, `rna_tbl`, `metadata`
: Optional in-memory input tables.

`atac_data_path`, `rna_path`, `metadata_path`
: Optional explicit file paths for the input tables.

`step1_out_dir_name`
: Output folder name under 'base_dir'.

`label_col`
: Metadata column used to aggregate matched conditions.

`expected_n`
: Optional expected matched sample count.

`tf_list`
: Optional TF allowlist for downstream correlation.

`motif_db`
: Optional motif metadata table.

`threshold_gene_expr`
: Expression threshold for Step 1 expression flags.

`threshold_fp_score`
: Footprint-score threshold for Step 1 bound flags.

`use_parallel`
: Logical; if 'TRUE', allow parallel work in supported helpers.

`verbose`
: Logical; if 'TRUE', emit concise progress messages.

`time_log`
: Logical; if TRUE, emit elapsed-time messages.

Value

A rebuilt Module 1 multi-omic object.

Examples

```
## Not run:
omics_data <- load_prep_multiomic_data(
  config = "dev/config/pdac_nutrient_stress_strict_jaspar2024_demo.yaml",
  genome = "hg38",
  label_col = "strict_match_rna",
  do_preprocess = FALSE,
  verbose = TRUE
)

## End(Not run)
```

`module1_correlate_TF_to_canonical_tfbs` *Correlate TFs to their canonical TFBS*

Description

Correlate TFs to their canonical TFBS

Usage

```
module1_correlate_TF_to_canonical_tfbs(
  module1_inputs,
  r_cutoff = 0.3,
  p_cutoff = NULL,
  fdr_cutoff = NULL,
  min_non_na = 3L,
  cores = NULL,
  verbose = TRUE
)
```

Arguments

`module1_inputs`	Output from module1_prepare_tfbs_inputs.
`r_cutoff`	Minimum positive best correlation.
`p_cutoff`	Optional best-method p-value cutoff.
`fdr_cutoff`	Optional best-method FDR cutoff.
`min_non_na`	Minimum finite condition pairs required.
`cores`	Number of worker cores; NULL uses all available cores.
`verbose`	Emit concise progress messages.

Value

A tibble with Pearson, Spearman, best-method statistics, and pass flags.

`module1_filter_canonical_bound_tfbs` *Filter footprints with canonical binding for full TFBS prediction*

Description

Filter footprints with canonical binding for full TFBS prediction

Usage

```
module1_filter_canonical_bound_tfbs(
  module1_inputs,
  motif_supported_correlations,
  r_cutoff = 0.3,
  p_cutoff = NULL,
  fdr_cutoff = NULL,
  filter_to_canonical_bound = TRUE,
  verbose = TRUE
)
```

Arguments

module1_inputs	Output from module1_prepare_tfbs_inputs.
motif_supported_correlations	Output from module1_correlate_TF_to_canonical_tfbs.
r_cutoff	Minimum positive best correlation.
p_cutoff	Optional p-value cutoff.
fdr_cutoff	Optional FDR cutoff.
filter_to_canonical_bound	Keep only footprints with a passing motif-supported TF.
verbose	Emit concise progress messages.

Value

A list with canonical-bound and prediction footprint tables.

---

module1_predict_full_tfbs

Predict full TFBS for all expressed TFs

Description

Predict full TFBS for all expressed TFs

Usage

```
module1_predict_full_tfbs(
  module1_inputs,
  prediction_footprints,
  out_dir = "predict_tf_binding_sites",
  r_cutoff = 0.3,
  p_cutoff = NULL,
  fdr_cutoff = NULL,
  min_non_na = 3L,
```

```
  cores = NULL,
  write_outputs = TRUE,
  output_format = c("csv", "parquet", "auto"),
  return_prediction_stats = NULL,
  verbose = TRUE
)
```

Arguments

module1_inputs	Output from module1_prepare_tfbs_inputs.
prediction_footprints	Footprint table from module1_filter_canonical_bound_tfbs.
out_dir	Output directory.
r_cutoff	Minimum positive best correlation.
p_cutoff	Optional best-method p-value cutoff.
fdr_cutoff	Optional best-method FDR cutoff.
min_non_na	Minimum finite condition pairs required.
cores	Number of worker cores; NULL uses all available cores.
write_outputs	Write predicted TFBS outputs.
output_format	One of csv, parquet, or auto.
return_prediction_stats	Return full prediction statistics in memory.
verbose	Emit concise progress messages.

Value

A list with prediction statistics or manifests and predicted TFBS outputs.

---

module1_prepare_tfbs_inputs

Prepare Module 1 TFBS prediction inputs

Description

Prepare Module 1 TFBS prediction inputs

Usage

```
module1_prepare_tfbs_inputs(
  omics_data,
  label_col = NULL,
  tf_subset = NULL,
  verbose = TRUE
)
```

Arguments

omics_data	CraftGRN multiomic object returned by ‘load_prep_multiomic_data()‘.
label_col	Optional metadata column used to rebuild condition matrices.
tf_subset	Optional TF symbols to keep.
verbose	Emit concise progress messages.

Value

A list containing prepared data, condition columns, TFs, and footprint universe.

---

module2_correlate_fp_targets

Correlate FP score with target gene expression

Description

Correlate FP score with target gene expression

Usage

```
module2_correlate_fp_targets(
  module2_inputs,
  candidates,
  n_cores = NULL,
  verbose = TRUE
)
```

Arguments

module2_inputs	Output from module2_identify_candidate_links.
candidates	Output from module2_link_fp_targets.
n_cores	Number of worker cores; NULL uses all available cores.
verbose	Emit concise progress messages.

Value

An FP-target correlation table with pass flags.

module2_correlate_tf_targets

Correlate TF expression with target gene expression

Description

Correlate TF expression with target gene expression

Usage

```
module2_correlate_tf_targets(module2_inputs, n_cores = NULL, verbose = TRUE)
```

Arguments

module2_inputs	Output from module2_identify_candidate_links.
n_cores	Number of worker cores; NULL uses all available cores.
verbose	Emit concise progress messages.

Value

A TF-target correlation table with pass flags.

module2_identify_candidate_links

Link TFs to potential target genes based on TFBS-TSS proximity or 3D interaction data

Description

Link TFs to potential target genes based on TFBS-TSS proximity or 3D interaction data

Usage

```
module2_identify_candidate_links(
  multiomic_data,
  predicted_tfbs,
  gene_tss = NULL,
  regulatory_prior = NULL,
  project_config = NULL,
  max_distance_bp = NULL,
  verbose = TRUE
)
```

Arguments

multiomic_data	CraftGRN multiomic object.
predicted_tfbs	Predicted TFBS table or path from Module 1.
gene_tss	Optional gene TSS table or path.
regulatory_prior	Optional generic FP-target prior.
project_config	Optional project config path or list.
max_distance_bp	Maximum signed distance to TSS.
verbose	Emit concise progress messages.

Value

A list of normalized Module 2 inputs used by downstream step functions.

---

module2_link_fp_targets

Build restricted candidate FP-target links

Description

Build restricted candidate FP-target links

Usage

```
module2_link_fp_targets(module2_inputs, tf_target_corr, verbose = TRUE)
```

Arguments

module2_inputs	Output from internal Module 2 input preparation.
tf_target_corr	Output from module2_correlate_tf_targets.
verbose	Emit concise progress messages.

Value

A candidate table restricted by TF-target pass calls and genomic priors.

`module2_output_predicted_links` *Assemble, filter, and output final predicted TF-FP-target links*

Description

Assemble, filter, and output final predicted TF-FP-target links

Usage

```
module2_output_predicted_links(
  module2_inputs,
  candidates,
  tf_target_corr,
  fp_target_corr,
  output_dir = NULL,
  output_format = c("auto", "parquet", "csv"),
  verbose = TRUE
)
```

Arguments

`module2_inputs`	Output from [module2_identify_candidate_links()].
`candidates`	Candidate table from [module2_link_fp_targets()].
`tf_target_corr`	TF-target correlation table from [module2_correlate_tf_targets()].
`fp_target_corr`	FP-target correlation table from [module2_correlate_fp_targets()].
`output_dir`	Optional output directory.
`output_format`	One of auto, parquet, or csv.
`verbose`	Emit concise progress messages.

Value

A Module 2 result list.

`module3_construct_docs` *Construct input documents for topic modeling*

Description

Builds and caches the document-level link table, document-term table, sparse document-term matrix, and summary metadata used by Module 3 topic modeling.

Usage

```
module3_construct_docs(
  filtered_dir,
  output_dir,
  tf_cluster_map = NULL,
  check_repeated_values = FALSE,
  ...
)
```

Arguments

filtered_dir	Directory containing Module 3 filtered differential-link CSV files.
output_dir	Directory where topic input caches are written.
tf_cluster_map	Named vector mapping TF names to motif clusters.
check_repeated_values	Warn about repeated inconsistent term values. The high-throughput default is 'FALSE'; set to 'TRUE' for diagnostic audits.
...	Additional topic-document construction arguments passed to the internal Module 3 document builder.

Value

A list with cache paths and input summary counts.

---

module3_extract_topics

Extract Module 3 regulatory topics

Description

Public step function for extracting regulatory topics, pathway summaries, topic-link tables, and review outputs from trained Module 3 topic models.

Usage

```
module3_extract_topics(
  k,
  model_dir,
  output_dir,
  flatten_single_output = TRUE,
  ...
)
```

Arguments

k	Integer K selected for extraction.
model_dir	Directory containing trained topic model outputs.
output_dir	Directory to write extracted topic outputs.
flatten_single_output	Whether to write a single selected model directly under ‘output_dir‘. Defaults to ‘TRUE‘ for the public step API.
...	Additional arguments passed to the internal extraction engine, such as ‘backend‘, ‘doc_mode‘, ‘weight_label‘, and ‘topic_report_args‘.

Value

Invisibly returns TRUE when extraction completes.

---

module3_prepare_differential_links

Prepare differential links for Module 3

Description

Converts Module 2 link manifests into the filtered differential-link files consumed by CraftGRN topic-modeling utilities. This avoids writing full per-condition GRN matrices and keeps Module 3 compatible with the existing ‘*_filtered_links_up.csv‘ and ‘*_filtered_links_down.csv‘ contract.

Usage

```
module3_prepare_differential_links(
  module2,
  multiomic_data,
  compar = NULL,
  project_config = NULL,
  output_dir = NULL,
  n_cores = NULL,
  pseudocount = 1,
  rna_de_results = NULL,
  fp_signal_mode = NULL,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

module2	Module 2 object returned by [predict_tf_targets()] or a path to a Module 2 output directory containing ‘module2_manifest.csv‘.
multiomic_data	CraftGRN multiomic object returned by [load_prep_multiomic_data()].

compar	Comparison table or CSV path with ‘cond1_label‘ and ‘cond2_label‘. If ‘NULL‘, ‘data/episcope_comparisons.csv‘ under ‘base_dir‘ is used.
project_config	Project config list or YAML path.
output_dir	Directory for filtered differential links. If ‘NULL‘, ‘regulatory_topics/differential_links‘ under ‘base_dir‘ is used.
n_cores	Number of data.table threads to use while reading and joining chunks. Defaults to all available cores. Comparison-level parallelism is controlled by ‘module3_comparison_workers‘ in the project config and defaults to 1 for RAM safety.
pseudocount	Pseudocount for log2 fold-change calculations.
rna_de_results	Optional standardized RNA differential expression table or CSV. When provided, target-gene and TF log2 fold changes are read from this table and direct condition fold changes are used only for missing genes.
fp_signal_mode	FP signal used for differential FP fold changes. actual uses the measured FP score in both conditions. link_padded sets the FP score to zero in conditions where the TF-FP-gene link is not active before calculating delta_fp_score and log2FC_fp_score.
overwrite	Overwrite existing filtered link files.
verbose	Emit concise progress messages.

Value

A tibble manifest with one row per comparison.

`module3_train_topic_models`

Train Module 3 topic models

Description

Public step function for training one Module 3 topic-model setup after [module3_prepare_differential_links()] has produced filtered differential links. This is a thin Module 3-named wrapper around the internal training engine.

Usage

```
module3_train_topic_models(
  k_grid,
  filtered_dir,
  output_dir,
  flat_output = TRUE,
  ...
)
```

Arguments

k_grid	Integer vector of K values for training.
filtered_dir	Directory containing Module 3 filtered differential-link files.
output_dir	Directory to write topic model outputs.
flat_output	Whether to write this selected setup directly under ‘output_dir‘. Defaults to ‘TRUE‘ for the public step API.
...	Additional arguments passed to the internal training engine, such as ‘doc_design‘, ‘fp_term_mode‘, ‘backend‘, and ‘local_threads‘.

Value

Invisibly returns TRUE when training completes.

---

output_predicted_tfbs *Output predicted TFBS*

Description

Output predicted TFBS

Usage

```
output_predicted_tfbs(
  prediction_stats,
  out_dir = NULL,
  output_format = c("auto", "parquet", "csv"),
  include_support = TRUE
)
```

Arguments

prediction_stats	Module 1 TFBS prediction statistic table.
out_dir	Optional output directory. If supplied, a predicted TFBS table and manifest are written for Module 2.
output_format	Output format: auto, parquet, or csv.
include_support	Include compact condition support when available.

Value

A predicted TFBS tibble when ‘out_dir‘ is NULL; otherwise a list with output paths and row counts.

predict_tf_targets *Predict TF targets through TFBS-target and TF-target correlations*

Description

Predict TF targets through TFBS-target and TF-target correlations

Usage

```
predict_tf_targets(
  multiomic_data,
  predicted_tfbs,
  gene_tss = NULL,
  regulatory_prior = NULL,
  project_config = NULL,
  output_dir = NULL,
  max_distance_bp = NULL,
  n_cores = NULL,
  output_format = c("auto", "parquet", "csv"),
  verbose = TRUE,
  write_qc_report = TRUE,
  qc_report_validate = FALSE
)
```

Arguments

multiomic_data	CraftGRN multiomic object returned by 'load_prep_multiomic_data()'.
predicted_tfbs	Compact Module 1 predicted TFBS table or manifest path.
gene_tss	Optional gene TSS annotation table or path. If 'NULL', the table is resolved from 'project_config$gene_tss' or generated from the configured 'ref_genome'.
regulatory_prior	Optional generic FP-target regulatory prior.
project_config	Optional project YAML path or list.
output_dir	Optional output directory.
max_distance_bp	Maximum signed distance to TSS for window candidates.
n_cores	Number of CPU cores.
output_format	Output format: auto, parquet, or csv.
verbose	Emit concise progress messages.
write_qc_report	Write a Module 2 HTML QC report when 'output_dir' is supplied.
qc_report_validate	Run relational integrity checks in the automatic QC report.

Value

Compact Module 2 relational result list.

`predict_tfbs` *Predict transcription factor binding sites from matched footprint and RNA data*

Description

Run the Module 1 TFBS workflow as one user-facing operation. The function first uses motif-supported FP-TF correlations to define high-confidence footprints, then predicts sparse FP-TF binding events for expressed TFs.

Usage

```
predict_tfbs(
  omics_data,
  out_dir = "predict_tf_binding_sites",
  db = "JASPAR2024",
  label_col = NULL,
  r_cutoff = 0.3,
  p_cutoff = NULL,
  fdr_cutoff = NULL,
  filter_to_canonical_bound = TRUE,
  tf_subset = NULL,
  write_outputs = TRUE,
  write_stats = FALSE,
  write_bed = FALSE,
  write_qc_report = TRUE,
  qc_report_scan = FALSE,
  output_format = c("csv", "parquet", "auto"),
  return_prediction_stats = NULL,
 prediction_return_limit = getOption("craftgrn.module1_prediction_return_limit", 5e+06),
  min_non_na = 3L,
  cores = NULL,
  verbose = TRUE,
  time_log = verbose
)
```

Arguments

`omics_data`	CraftGRN multiomic object returned by ‘load_prep_multiomic_data()‘.
`out_dir`	Output directory.
`db`	Motif database label used in output metadata.
`label_col`	Metadata column used to build condition-level matrices when missing from ‘omics_data‘.

r_cutoff	Minimum positive correlation used for motif-supported and prediction calls.
p_cutoff	Optional best-method p-value cutoff. If 'NULL', p-value filtering is disabled.
fdr_cutoff	Optional best-method adjusted p-value cutoff. If 'NULL', FDR filtering is disabled.
filter_to_canonical_bound	Logical; if 'TRUE', only footprints with at least one motif-supported TF passing the cutoffs are used for the all-expressed-TF prediction stage.
tf_subset	Optional TF subset.
write_outputs	Write Module 1 output files.
write_stats	Retain and write full FP-TF correlation statistics.
write_bed	Write optional BED-like browser files for high-confidence footprints and in-memory TFBS prediction statistics.
write_qc_report	Write a Module 1 HTML QC report when outputs are written.
qc_report_scan	Scan predicted TFBS chunks for top-TF summaries in the QC report.
output_format	Output format for large streamed TFBS prediction statistic chunks.
return_prediction_stats	Return the TFBS prediction statistic table in memory. If 'NULL', large output-writing runs are streamed to disk and return a manifest.
prediction_return_limit	Maximum number of predicted events to keep in memory when 'return_prediction_stats = NULL' and 'write_outputs = TRUE'.
min_non_na	Minimum finite condition pairs required for correlation.
cores	Number of worker cores for the dense prediction correlation step. If 'NULL', use available cores.
verbose	Emit concise progress messages.
time_log	Logical; if TRUE, emit elapsed-time messages.

Value

A list containing 'omics_data', 'high_confidence_footprints', 'motif_supported_correlations', 'prediction_stats', 'prediction_stats', 'reports', and 'parameters'.

`query_predicted_links` *Query specific links by TF(s) and/or distance to TSS*

Description

Query specific links by TF(s) and/or distance to TSS

Usage

```
query_predicted_links(
  module2,
  tf = NULL,
  fp_id = NULL,
  target_gene = NULL,
  max_distance_to_tss = NULL,
  pass_only = TRUE
)
```

Arguments

module2	Module 2 result list or loaded output list.
tf	Optional TF filter.
fp_id	Optional FP filter.
target_gene	Optional target-gene filter.
max_distance_to_tss	Optional maximum absolute distance to TSS.
pass_only	Keep only passing links.

Value

A tibble of matching final links.

---

report_direct_tf_tf_regulations

Export an interactive HTML browser of direct TF-TF regulations

---

Description

Export an interactive HTML browser of direct TF-TF regulations

Usage

```
report_direct_tf_tf_regulations(
  module2,
  output_dir,
  multiomic_data = NULL,
  k_values = c(5L, 7L, 10L),
  verbose = TRUE
)
```

Arguments

module2	Module 2 result list, loaded output list, or output directory.
output_dir	Output directory.
multiomic_data	Optional CraftGRN multiomic object for condition-filtered reports.
k_values	Cluster counts.
verbose	Emit concise progress messages.

Value

A tibble report manifest.

---

report_tf_tf_coregulations

Export an interactive HTML browser of TF-TF co-regulatory activities

---

Description

Export an interactive HTML browser of TF-TF co-regulatory activities

Usage

```
report_tf_tf_coregulations(
  module2,
  output_dir,
  multiomic_data = NULL,
  k_values = c(5L, 7L, 10L),
  verbose = TRUE
)
```

Arguments

module2	Module 2 result list, loaded output list, or output directory.
output_dir	Output directory.
multiomic_data	Optional CraftGRN multiomic object for condition-filtered reports.
k_values	Cluster counts.
verbose	Emit concise progress messages.

Value

A tibble report manifest.

report_top_tf_targets *Export an interactive HTML browser of individual TF regulons*

Description

Export an interactive HTML browser of individual TF regulons

Usage

```
report_top_tf_targets(module2, output_dir, tfs, top_n = 100L, verbose = TRUE)
```

Arguments

module2	Module 2 result list, loaded output list, or output directory.
output_dir	Output directory.
tfs	TFs to report.
top_n	Number of top targets per TF.
verbose	Emit concise progress messages.

Value

A tibble report manifest.

run_app *Run the Shiny Application*

Description

Run the Shiny Application

Usage

```
run_app(
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

Arguments

onStart	A function that will be called before the app is actually run. This is only needed for shinyAppObj, since in the shinyAppDir case, a global.R file can be used for this purpose.
options	Named options that should be passed to the runApp call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify width and height parameters which provide a hint to the embedding environment about the ideal height/width for the app.
enableBookmarking	Can be one of "url", "server", or "disable". The default value, NULL, will respect the setting from any previous calls to enableBookmarking(). See enableBookmarking() for more information on bookmarking your app.
uiPattern	A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
...	arguments to pass to golem_opts. See '?golem::get_golem_options' for more details.

---

run_topic_modeling *Run topic modeling*

Description

Wrapper function to conduct the full regulatory topic-modeling workflow for one selected topic-document construction method.

Usage

```
run_topic_modeling(
  filtered_dir,
  multiomic_data = NULL,
  comparisons,
  output_dir,
  project_config = NULL,
  method = NULL,
  k_grid = NULL,
  warplda_iterations = NULL,
  topic_link_output = NULL,
  vae_device = NULL,
  vae_batch_size = NULL,
  pathway_backend = NULL,
  ...
)
```

Arguments

`filtered_dir`	Directory containing Module 3 filtered differential-link files.
`multiomic_data`	Optional CraftGRN multiomic object. Required when ‘replicate_documents = TRUE‘.
`comparisons`	Comparison or condition grouping table, or a CSV path.
`output_dir`	Topic output directory.
`project_config`	Optional project YAML path or config list. When supplied, ‘topic_method‘, ‘topic_k‘ or ‘topic_k_grid‘, ‘warplda_iterations‘, and ‘topic_link_output‘ are used for arguments that are left as ‘NULL‘.
`method`	Single Module 3 method ID. If ‘NULL‘, read from ‘project_config‘ or use the package default.
`k_grid`	Integer topic numbers. If ‘NULL‘, read from ‘project_config‘ or use ‘10‘.
`warplda_iterations`	Number of native WarpLDA iterations. If ‘NULL‘, read from ‘project_config‘ or use ‘2000‘.
`topic_link_output`	Topic-link output mode. If ‘NULL‘, read from ‘project_config‘ or use ‘"pass"‘.
`vae_device`	VAE device, for example ‘"auto"‘, ‘"cpu"‘, or ‘"cuda"‘. If ‘NULL‘, read from ‘project_config‘ or use ‘"auto"‘.
`vae_batch_size`	VAE mini-batch size. If ‘NULL‘, read from ‘project_config‘ or use ‘64‘.
`pathway_backend`	Pathway enrichment backend. Use ‘"enrichly"‘ for local cached enrichment or ‘"enrichr"‘ for the Enrichr web API. If ‘NULL‘, read from ‘project_config‘ or use ‘"enrichly"‘.
`...`	Additional arguments passed to the internal topic-modeling wrapper.

Value

An invisible list with topic input/model/extraction paths, review outputs, and ‘qc_report‘ when requested.

`save_omics_data` *Save a multi-omic data object to disk*

Description

Save a multi-omic data object to disk

Usage

```
save_omics_data(
  omics_data,
  file = NULL,
  out_dir = NULL,
  db = NULL,
  prefix = "omics_data",
  compress = "xz",
  verbose = TRUE
)
```

Arguments

omics_data	A multi-omic data list (e.g., output of load_prep_multiomic_data()).
file	Optional full path to an RDS file. If NULL, uses out_dir/db/prefix.
out_dir	Output directory used when file is NULL.
db	Optional database tag appended to the filename when file is NULL.
prefix	Filename prefix used when file is NULL.
compress	Compression passed to saveRDS().
verbose	Emit status messages.

Value

Path to the written file (invisible).

---

validate_config *Validate config values*

Description

Ensures required config keys (e.g. thresholds and db) exist in the chosen environment before running pipelines.

Usage

```
validate_config(
  required = c("db", "ref_genome", "threshold_expr", "threshold_fp_score",
    "threshold_fp_tf_corr_r", "link_window_bp", "threshold_rna_gene_corr_r",
    "threshold_fp_gene_corr_r"),
 numeric_required = c("threshold_expr", "threshold_fp_score", "threshold_fp_tf_corr_r",
    "link_window_bp", "threshold_rna_gene_corr_r", "threshold_fp_gene_corr_r"),
  env = .craftgrn_state
)
```

Arguments

required	Character vector of required variable names.
numeric_required	Character vector of required numeric variable names.
env	Environment to check. Defaults to the internal CraftGRN config state.

Value

TRUE invisibly when validation passes.

---

visualize_differential_grns

Export an interactive HTML browser of differential GRNs

Description

Builds an interactive TF-to-gene network browser from Module 3 filtered differential links. Users can select a comparison, choose up or down differential links, adjust the number of top TFs and links to display, and inspect footprint-supported edge evidence in tooltips.

Usage

```
visualize_differential_grns(
  differential_links_dir,
  output_dir = file.path(differential_links_dir, "reports"),
  top_tf_n = 10L,
  top_link_n = 300L,
  default_direction = "up",
  browser_max_rows_per_file = 50000L,
  top_n = NULL,
  verbose = TRUE
)
```

Arguments

differential_links_dir	Module 3 differential-link directory.
output_dir	Directory where the browser HTML and CSV summaries are written.
top_tf_n	Default number of top TFs shown in the browser.
top_link_n	Default number of top TF-to-gene links shown in the browser.
default_direction	Initial direction selected in the browser.

`browser_max_rows_per_file`
: Maximum filtered-link rows read from each comparison/direction file when building the browser payload. The full filtered-link CSVs remain the authoritative data source; this cap keeps the self-contained HTML browser responsive for large projects.

`top_n`
: Deprecated compatibility alias for `top_tf_n`.

`verbose`
: Emit concise progress messages.

Value

Path to the HTML browser.

`visualize_topic_modeling_results`
Export interactive HTML browsers of topic modeling results

Description

Builds a self-contained index browser for existing Module 3 topic-modeling review outputs at the topic, condition, comparison, and pathway levels. This function organizes existing outputs and does not train or extract models.

Usage

```
visualize_topic_modeling_results(
  topic_dir,
  output_dir = file.path(topic_dir, "reports"),
  include = c("topic", "condition", "comparison", "pathway"),
  verbose = TRUE
)
```

Arguments

`topic_dir`
: Module 3 topic output directory.

`output_dir`
: Directory where the browser HTML and manifest are written.

`include`
: Existing output families to include.

`verbose`
: Emit concise progress messages.

Value

Path to the HTML browser.

Index