Package ‘visualfrsr’

November 12, 2025

Title Visual Toolkit for Fast Reciprocal Square Root Experiments

Version 0.0.0.9000

Description Provides data pipelines, sampling utilities, and plotting helpers for investigating Fast Reciprocal Square Root (FRSR) approximations. The package wraps the exploratory scripts from the FastInverseSqrt-Visualized project into structured modules that expose reproducible datasets and ggplot2-based visualizations.

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports Matrix, Rglpk, dplyr, ggplot2, tibble, tidyr, scales, purrr, frsrr, viridis, rlang

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Adam Hyland [aut, cre]

Maintainer Adam Hyland <0x5f37642f@protonmail.com>

Contents

`visualfrsr-package` *visualfrsr: Visual Explorations of Fast Reciprocal Square Root Approximations*

Description

Provides data generators, binning utilities, clustering helpers, and standardized plotting functions that were previously scattered across scripts in the FastInverseSqrt-Visualized project. The package exposes those capabilities as reusable functions so datasets and visualizations can be reproduced programmatically or integrated into pipelines.

Author(s)

Maintainer: Adam Hyland <`0x5f37642f@protonmail.com`>

`binning_plots` *Binning visualization builders.*

Description

Assemble ggplot layers for the binned datasets and helpers. These functions expect inputs prepared by transform_binning and ultimately draw from the artifacts refreshed via `run_binned_pipeline()`.

Usage

```
bucket_heatmap_plot(bucket_df = load_bucket_sweep())

plot_bucket_widths(selection_df = prep_bucket_selection())

plot_bucket_widths_horizontal(selection_df = prep_bucket_selection())

plot_bucket_rectangles(selection_df = prep_bucket_selection())

pair_plot(bucket_df, spread_df)
```

```
plot_bucket(selection_df = prep_bucket_selection())

plot_multiple_n(binned, n_values = unique(binned$N))

plot_normalized_error_curve(binned, bins = 4:36)

compare_n_values(binned, n_small, n_large)

compare_buckets(bucket1, bucket2, error = "max")
```

Arguments

`bucket_df`	Bucket definitions for the binned view.
`selection_df`	Data frame created by `prep_bucket_selection()`.
`spread_df`	Error spread tibble for overlayed scatter/smooth layers.
`binned`	Dataset with multiple bucket counts.
`n_values`	Bucket counts retained in the facets.
`bins`	Candidate bin counts for evaluation.
`n_small`	Smaller bucket count for comparison.
`n_large`	Larger bucket count for comparison.
`bucket1`	First bucket selection tibble.
`bucket2`	Second bucket selection tibble.
`error`	Error column to compare (`"max"` or `"avg"`).

Value

A ggplot2::ggplot heatmap of bucket selections.

See Also

`run_binned_pipeline()` for producing the datasets that feed these plots and transform_binning for the supporting data wrangling helpers.

`bin_samples` *Binned optimisation artifacts*

Description

Outputs from `run_binned_pipeline()` summarising the optimisation sweeps and post-processing used in the bucket selection visuals.

Usage

```
data(bin_samples)
data(bucket_summary)
data(oob_performance)
```

Format

A tibble combining sweeps across requested bin counts with columns:

N Number of bins in the sweep.

Range_Min Lower bound of the bin's input interval.

Range_Max Upper bound of the bin's input interval.

Magic Magic constant expressed as a hexadecimal string.

Avg_Relative_Error Average relative error over the interval.

Max_Relative_Error Worst-case relative error over the interval.

Details

Additional columns emitted by `frsrr::frsr_bin()` are preserved.
`bucket_summary` adds derived metrics such as `Depth`, `Variance`, and `Rarity`.
`oob_performance` augments `bin_samples` with `OOB_Error` measurements.

Source

'`data-raw/pipelines/binned.R`'

See Also

`run_binned_pipeline()` for regeneration guidance and `prep_bucket_selection` for transformation helpers.

`build_heatmap_data` *Heatmap-ready data summarizing location, bin count, and magic.*

Description

Heatmap-ready data summarizing location, bin count, and magic.

Usage

```
build_heatmap_data(bucket_df = load_bucket_sweep())
```

Arguments

`bucket_df`	Data frame returned by `load_bucket_sweep()`.

Value

Tidy data frame.

`cluster_bands` *Cluster exploration artifacts*

Description

Datasets produced by `run_cluster_pipeline()` for the clustering visuals.

Usage

```
data(cluster_bands)
data(representative_clusters)
```

Format

A tibble with columns:

cluster Cluster identifier produced by `stats::kmeans()`.

statistic Whether the row captures the minimum or maximum error.

value Magic constant associated with the statistic.

Details

`representative_clusters` captures sampled inputs, their approximation error, and the associated cluster label.

Source

‘`data-raw/pipelines/clusters.R`’

See Also

`run_cluster_pipeline()` for regeneration instructions and `prep_cluster_samples` for transformations used in the visuals.

`compute_result_block` *Sample Newton-Raphson parameter grids.*

Description

Generates randomized `(A, B)` parameter pairs across the input domain and records error ranks for downstream optimization visuals.

Usage

```
compute_result_block(slices = 1e+05, GRID_SIZE = 50)
```

Arguments

`slices`	Number of float samples drawn from `frsr_sample()`.
`GRID_SIZE`	Resolution of the ranking grid applied to `A`, `B`, and error columns.

Value

Tibble with ranked parameter/error combinations.

`create_slices` *Slice frsr sample data into equal-width bins.*

Description

Slice frsr sample data into equal-width bins.

Usage

```
create_slices(df, N, min_input = 0.5, max_input = 2)
```

Arguments

`df`	Data frame containing an `input` column.
`N`	Number of slices.
`min_input`	Minimum input to keep.
`max_input`	Maximum input to keep.

Value

Tidy data frame with `slice` column.

`deconstructed` *Core deconstruction samples*

Description

These tibbles capture the reciprocal square-root exploration datasets generated by `run_deconstruction_pipeline()`. Each object represents a different sampling strategy used throughout the package visuals.

Usage

```
data(deconstructed)
data(widened)
data(narrowed)
```

Format

Tibbles with the following columns:

input Sampled floating-point inputs in $[0.5, 2)$.
reference True reciprocal square-root for each input.
initial Initial fast-inverse square-root estimate.
after_one Value after a single Newton-Raphson refinement.
final Final estimate returned by `frsrr::frsr_sample()`.
iters Factor describing the number of iterations until convergence.
magic Integer interpretation of the chosen magic constant.
input_rank Quantile bin for the sampled input value.
magic_rank Quantile bin for the selected magic constant.
initial_rank Quantile bin for the initial approximation error.
after_rank Quantile bin for the post-iteration error.
iter_rank Binned iteration counts for plotting facets.

Source

‘`data-raw/pipelines/deconstruction.R`’

See Also

`run_deconstruction_pipeline()` for regeneration instructions and `prep_deconstruction_combined` for the companion transformations.

`deconstruction_plots` *Deconstruction visualization builders.*

Description

These plotting helpers pair with the datasets returned by `run_deconstruction_pipeline()`. Each function expects prepped data from the `prep_deconstruction_*()` family and returns a ggplot2::ggplot object ready for rendering.

Usage

```
deconstruction_zoom_plot(narrowed)

deconstruction_rate_plot(deconstructed)

deconstruction_magic_span_plot(deconstructed)

deconstruction_quadratic_plot(deconstructed)

deconstruction_improvement_plot(deconstructed)

deconstruction_painterly_plot(deconstructed)

deconstruction_polar_plot(deconstructed)

deconstruction_combined_plot(deconstructed, widened, iter_levels = 1:5)
```

Arguments

`narrowed`	Zoomed-in deconstruction tibble.
`deconstructed`	Core deconstruction tibble.
`widened`	Wide sweep deconstruction tibble.
`iter_levels`	Iteration levels retained when spotlighting subsets.

Value

A ggplot2::ggplot object.

See Also

`run_deconstruction_pipeline()` for regenerating the source datasets and transform_deconstruction for the companion transformation helpers.

`equal_bins_data` *Reference bin catalogs derived from CSV exports.*

Description

These datasets mirror the CSV artifacts stored under `inst/extdata/` and are produced by `data-raw/prepare_binned_c` Each table captures a fixed binning strategy so downstream visuals can be reproduced without rerunning the expensive samplers.

Usage

```
equal05_20bins

equal_bins

manybins

too_many_bins

varied_bins
```

Format

A data frame with the following columns:

Location Ordinal identifier of the bin within the sweep.

Range_Min Lower bound of the input interval.

Range_Max Upper bound of the input interval.

Bin_Type Factor describing how the bins were created (equal width, varied bottoms, etc.).

Magic Integer representation of the restoring constant.

Avg_Relative_Error Average relative error over the interval.

Max_Relative_Error Worst-case relative error within the interval.

N Number of bins in the sweep.

An object of class `data.frame` with 1272 rows and 9 columns.

An object of class `data.frame` with 888 rows and 9 columns.

An object of class `data.frame` with 131322 rows and 8 columns.

An object of class `data.frame` with 3228 rows and 8 columns.

An object of class `data.frame` with 3520 rows and 9 columns.

Source

`data-raw/prepare_binned_data.R`

`find_optimal_magic` *Find the optimal magic constant for a slice.*

Description

Find the optimal magic constant for a slice.

Usage

```
find_optimal_magic(slice_data)
```

Arguments

`slice_data`	Slice produced by `create_slices()`.

Value

List containing the minimum magic and its objective value.

`generate_example_samples` *Generate deterministic comparison samples across classical FRSR variants.*

Description

Generate deterministic comparison samples across classical FRSR variants.

Usage

```
generate_example_samples(slices = 16384, float_tol = 0.0004882812)
```

Arguments

`slices`	Base number of slices for the log-stratified samples.
`float_tol`	Tolerance passed to `frsrr::frsr()`.

Value

Tidy data frame combining four approximation families.

history_plots *Historical approximation visualization builders.*

Description

Render comparisons across classical fast inverse square root implementations using the outputs of `run_deconstruction_pipeline()` and helper transforms such as `prep_example_samples()`.

Usage

```
plot_example_comparison(approximated = prep_example_samples())

plot_iteration_histogram(approximated = prep_example_samples())

nrplot(df = prep_example_samples(), approx = "QuakeIII")
```

Arguments

`approximated`	Tibble generated by `prep_example_samples()`.
`df`	Tibble from `prep_example_samples()`.
`approx`	Which approximation(s) to highlight in `nrplot()`.

Value

A ggplot2::ggplot object.

See Also

`run_deconstruction_pipeline()` for regenerating the historical comparison datasets and `prep_example_samples()` for the transformation helper mentioned throughout this module.

load_bucket_sweep *Load precomputed bucket sweeps.*

Description

Load precomputed bucket sweeps.

Usage

```
load_bucket_sweep(filename = "too_many_bins.csv")
```

Arguments

`filename`	Name of the CSV stored under `inst/extdata`.

Value

Data frame.

`load_varied_bins` *Load the varied bin dataset with ordered Bottom factor.*

Description

Load the varied bin dataset with ordered Bottom factor.

Usage

```
load_varied_bins()
```

Value

Data frame.

`logStratifiedSampler` *Log-stratified sampler for floats.*

Description

Log-stratified sampler for floats.

Usage

```
logStratifiedSampler(min, max, n)
```

Arguments

`min`	Minimum sample value.
`max`	Maximum sample value.
`n`	Number of points to draw.

Value

Numeric vector of length `n`.

optimization_plots *Optimization visualization builders.*

Description

Optimization visualization builders.

Usage

```
plot_optimized_tile(optimized = prep_optimized_results())

plot_optimized_paths(optimized = prep_optimized_results())

plot_optimized_pairs(optimized = prep_optimized_results())
```

Arguments

optimized	Tibble produced by prep_optimized_results().

Value

A ggplot2::ggplot object summarizing optimization landscapes.

See Also

run_binned_pipeline() for regenerating the optimization landscapes and prep_optimized_results() for producing the tidy search results consumed by these plots.

plot_sampled_clusters *Cluster visualization builders.*

Description

Plot sampled clusters of magic constants across the input range. These layers assume data produced by run_cluster_pipeline() and tidied by helpers in R/transform-clusters.R.

Usage

```
plot_sampled_clusters(clusters = prep_cluster_samples())
```

Arguments

clusters	Tibble produced by prep_cluster_samples().

Value

A ggplot2::ggplot object visualising representative clusters.

See Also

run_cluster_pipeline() for regenerating the source datasets and prep_cluster_samples() for data preparation helpers.

`prep_cluster_samples` *Prepare sampled clusters for plotting.*

Description

Wraps `sample_clusters()` so ggplot builders can accept either raw cluster draws or pre-computed data. The function simply forwards arguments to the sampler and returns the tidy tibble, bridging from `run_cluster_pipeline()` to `plot_sampled_clusters()`.

Usage

```
prep_cluster_samples(
  n = 16384,
  cluster_ranges = list(c(1597622357, 1597176038), c(1598049921, 1597918602),
    c(1597884263, 1596910052))
)
```

Arguments

`n`	Number of samples per cluster.
`cluster_ranges`	List of length three containing magic ranges.

Value

Tibble of sampled clusters.

See Also

`run_cluster_pipeline()` for regenerating the underlying data and `plot_sampled_clusters()` for visualizing the prepared samples.

`prep_example_samples` *Prepare historical approximation datasets for plotting.*

Description

Prepare historical approximation datasets for plotting.

Usage

```
prep_example_samples(slices = 16384, float_tol = 0.0004882812)
```

Arguments

`slices`	Base number of slices for the log-stratified samples.
`float_tol`	Tolerance passed to `frsrr::frsr()`.

Value

Tidy tibble containing approximated inverse square roots.

See Also

`run_deconstruction_pipeline()` for regenerating the approximated datasets and history_plots for visual layers that consume the tidy output.

`prep_optimized_results` *Prepare optimized parameter search results for plotting.*

Description

Prepare optimized parameter search results for plotting.

Usage

```
prep_optimized_results(slices = 1e+05, GRID_SIZE = 50)
```

Arguments

`slices`	Number of float samples drawn from `frsr_sample()`.
`GRID_SIZE`	Resolution of the ranking grid applied to `A`, `B`, and error columns.

Value

Tibble describing optimization outcomes.

See Also

`run_binned_pipeline()` for regenerating the optimization sweeps and optimization_plots for plotting helpers.

`sample_clusters` *Sample representative clusters for downstream visualizations.*

Description

Sample representative clusters for downstream visualizations.

Usage

```
sample_clusters(
  n = 16384,
  cluster_ranges = list(c(1597622357, 1597176038), c(1598049921, 1597918602),
    c(1597884263, 1596910052))
)
```

Arguments

`n`	Number of samples per cluster.
`cluster_ranges`	List of length three containing magic ranges.

Value

Tibble with input/error/magic/cluster columns.

`transform_binning` *Prepare binned data products for visualization.*

Description

These helpers wrap higher level binning utilities so plots can focus on aesthetic composition. Functions return ready-to-plot data frames or lists of derived statistics produced after `run_binned_pipeline()`.

Usage

```
prep_bucket_heatmap(bucket_df = load_bucket_sweep())

prep_bucket_selection(df = load_bucket_sweep(), range = 4:100)

prep_bucket_widths(selection_df)

prep_bucket_rectangles(selection_df)

prep_bucket_pair_layout(bucket_df, spread_df)

prep_bucket_comparison(binned, n_small, n_large)

prep_bucket_error_segments(df)

prep_bucket_multiple_n(binned, n_values = unique(binned$N))

prep_bucket_error_curve(binned, bins = 4:36)

prep_bucket_compare_buckets(bucket1, bucket2, error = "max")
```

Arguments

`bucket_df`	Output from `load_bucket_sweep()`.
`df`	Binned dataset used when recomputing bucket selections.
`range`	Vector of candidate bucket counts.
`selection_df`	Data frame returned by `bucket_selection()`.
`spread_df`	Data frame containing error spreads for paired plots.
`binned`	Dataset of optimal bucket summaries.
`n_small`	Small bucket count used for comparison visuals.
`n_large`	Large bucket count used for comparison visuals.
`n_values`	Vector of `N` values to retain.
`bins`	Integer vector of bin counts to evaluate.
`bucket1`	First bucket selection tibble supplied to the comparator.
`bucket2`	Second bucket selection tibble supplied to the comparator.
`error`	Which error metric to compare (`"max"` or `"avg"`).

Value

A tibble or list ready for downstream plotting.

See Also

`run_binned_pipeline()` for regenerating the source data and binning_plots for the visual layers that consume these transforms.

`transform_deconstruction` *Prepare deconstruction-derived datasets for visualization.*

Description

These helpers enrich the raw outputs from `run_deconstruction_pipeline()` so plotting functions can focus on layer composition. Each function returns a tibble tailored to a family of deconstruction visuals.

Usage

```
prep_deconstruction_errors(deconstructed)

prep_deconstruction_zoom(narrowed)

prep_deconstruction_magic_span(deconstructed, max_iter = 4)

prep_deconstruction_subset(deconstructed, iter_levels = 1:5)

prep_deconstruction_combined(deconstructed, widened, iter_levels = 1:5)
```

Arguments

`deconstructed`	Tidy tibble describing the core deconstruction sample.
`narrowed`	Tidy tibble describing the zoomed-in deconstruction sample.
`max_iter`	Highest iteration count to include when summarising spans.
`iter_levels`	Integer vector of iteration levels to retain.
`widened`	Tidy tibble describing the wide sweep deconstruction sample.

Value

A tibble or named list ready for plotting.

See Also

`run_deconstruction_pipeline()` for refreshing the underlying datasets and deconstruction_plots for the plotting helpers that consume these transforms.

Index