Package 'SQUIRE'

November 19, 2025

Type Package

Title Statistical Quality-Assured Integrated Response Estimation

Version 1.0.0

Author Richard A. Feiss [aut, cre] (ORCID: <https://orcid.org/0009-0008-0409-6042>)

Maintainer Richard A. Feiss <feiss026@umn.edu>

Description Implements statistically validated biological parameter optimization that combines automated parameter type detection with rigorous statistical quality assurance. Unlike conventional optimizers that fit parameters to any data, 'SQUIRE' first validates whether statistically significant biological effects exist before proceeding with parameter estimation. Uses trust region methods from Conn et al. (2000) <doi:10.1137/S1052623497325107>, ANOVA-based validation following Fisher (1925) <doi:10.1007/978-1-4612-4380-9_6>, and effect size calculations per Cohen (1988, ISBN:0805802835). Automatically distinguishes rate-based, positive-constrained, and unconstrained variables, applying geometry-appropriate optimization methods while preventing over-fitting to noise through built-in statistical validation, effect size assessment, and data quality requirements. Designed for complex biological and environmental models including germination studies, dose-response curves, and survival analysis. Enhanced successor to the 'GALAHAD' optimization framework with integrated statistical gatekeeping. Developed at the Minnesota Center for Prion Research and Outreach at the University of Minnesota.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.2.0)

Imports stats

Suggests testthat (>= 3.0.0)

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2025-11-19 18:00:02 UTC

Contents

`assess_data_quality` *Assess Data Quality*

Description

Check data quality requirements for optimization

Usage

```
assess_data_quality(data, treatments, min_timepoints, min_replicates)
```

Arguments

`data`	Experimental data frame
`treatments`	Treatment names vector
`min_timepoints`	Minimum required timepoints
`min_replicates`	Minimum required replicates

Value

List with data quality assessment

`calculate_response_metric` *Calculate Response Metric*

Description

Calculate appropriate response metric for statistical testing

Usage

```
calculate_response_metric(data)
```

Arguments

`data`	Experimental data frame

Value

Vector of response metrics

`calibrate_galahad_parameters` *Calibrate GALAHAD Parameters*

Description

Two-cycle parameter calibration for geometry-adaptive optimization

Usage

```
calibrate_galahad_parameters(
  data,
  treatments,
  response_type,
  validation_results,
  verbose = FALSE
)
```

Arguments

`data`	Experimental data frame
`treatments`	Treatment names vector
`response_type`	Type of biological response
`validation_results`	Results from statistical validation
`verbose`	Print progress messages

Value

Calibrated GALAHAD configuration

calibrate_optimization_parameters

Calibrate Optimization Parameters

Description

Calibrate numerical parameters for optimization

Usage

```
calibrate_optimization_parameters(data, geometry_config)
```

Arguments

`data`	Experimental data frame
`geometry_config`	Geometry configuration

Value

Optimization configuration

compile_optimization_results

Compile Optimization Results

Description

Compile results from multiple treatment optimizations

Usage

```
compile_optimization_results(treatment_parameters, treatments)
```

Arguments

`treatment_parameters`	List of treatment-specific parameters
`treatments`	Vector of treatment names

Value

Compiled optimization results

`describe_treatment_effects` *Describe Treatment Effects*

Description

Describe how treatments affect each parameter

Usage

```
describe_treatment_effects(param_values, param_name)
```

Arguments

`param_values`	Parameter values for different treatments
`param_name`	Parameter name

Value

Treatment effect description

`determine_geometry_partitioning` *Determine Geometry Partitioning*

Description

Determine parameter types for geometry-adaptive optimization

Usage

```
determine_geometry_partitioning(data, response_type)
```

Arguments

`data`	Experimental data frame
`response_type`	Type of biological response

Value

Geometry configuration

fit_biological_model *Fit Biological Model*

Description

Fit biological model to single treatment data

Usage

```
fit_biological_model(data, response_type, galahad_config, verbose = FALSE)
```

Arguments

data	Treatment-specific data frame
response_type	Type of biological response
galahad_config	GALAHAD configuration
verbose	Print progress messages

Value

Model fitting results

fit_germination_model *Fit Germination Model*

Description

Fit germination-specific model

Usage

```
fit_germination_model(data, galahad_config, verbose = FALSE)
```

Arguments

data	Germination data frame
galahad_config	GALAHAD configuration
verbose	Print progress messages

Value

Germination model results

generate_biological_interpretation

Generate Biological Interpretation

Description

Create biological interpretation of optimization results

Usage

```
generate_biological_interpretation(
  optimization_results,
  parameter_validation,
  response_type
)
```

Arguments

`optimization_results`	Optimization results
`parameter_validation`	Parameter validation results
`response_type`	Type of biological response

Value

Biological interpretation

generate_recommendations

Generate Recommendations

Description

Provide methodological and experimental recommendations

Usage

```
generate_recommendations(validation_results, parameter_validation)
```

Arguments

`validation_results`	Statistical validation results
`parameter_validation`	Parameter validation results

Value

Recommendations list

`get_biological_meaning`

Get Biological Meaning

Description

Translate parameter names to biological interpretation

Usage

```
get_biological_meaning(param_name, response_type)
```

Arguments

`param_name`	Parameter name
`response_type`	Type of biological response

Value

Biological meaning description

`perform_validated_optimization`

Perform Validated Optimization

Description

Execute parameter optimization with statistical validation

Usage

```
perform_validated_optimization(
  data,
  treatments,
  galahad_config,
  validation_results,
  response_type,
  verbose = FALSE
)
```

Arguments

`data`	Experimental data frame
`treatments`	Treatment names vector
`galahad_config`	GALAHAD configuration
`validation_results`	Statistical validation results
`response_type`	Type of biological response
`verbose`	Print progress messages

Value

Optimization results

SQUIRE *SQUIRE: Statistical Quality-Assured Integrated Response Estimation*

Description

Geometry-adaptive biological parameter estimation with built-in statistical validation. Implements two-cycle optimization: statistical validation followed by GALAHAD-calibrated parameter estimation.

Usage

```
SQUIRE(
  data,
  treatments,
  control_treatment = treatments[1],
  response_type = c(”germination”, ”growth”, ”survival”),
  validation_level = 0.05,
  min_timepoints = 5,
  min_replicates = 3,
  galahad_config = NULL,
  verbose = TRUE
)
```

Arguments

`data`	Data frame with columns: time, response, treatment, replicate
`treatments`	Character vector of treatment names
`control_treatment`	Name of control treatment for comparisons
`response_type`	Type of response: "germination", "growth", "survival"
`validation_level`	Statistical significance level (default: 0.05)

`min_timepoints` Minimum timepoints required for fitting (default: 5)

`min_replicates` Minimum replicates per treatment (default: 3)

`galahad_config` Optional pre-calibrated GALAHAD parameters

`verbose` Logical, print progress messages

Details

SQUIRE implements a two-stage validation process:

Stage 1: Statistical Validation

- Tests for significant treatment effects using ANOVA
- Checks data quality requirements (timepoints, replication)
- Only proceeds to optimization if biological signals detected

Stage 2: Validated Optimization

- Calibrates GALAHAD geometry parameters on significant effects
- Applies optimized parameters with uncertainty quantification
- Validates that optimized parameters are statistically meaningful

Value

List with statistical validation results, optimized parameters, and biological interpretation (only if statistically justified)

Examples

```
# Quick data setup example (fast execution)
n_time <- 5
n_rep <- 3

# Simulate example data
example_data <- data.frame(
  time = rep(1:n_time, times = 3 * n_rep),
  treatment = rep(c(”Control”, ”Treatment_A”, ”Treatment_B”),
                  each = n_time * n_rep),
  replicate = rep(rep(1:n_rep, each = n_time), times = 3),
  response = c(
    cumsum(rbinom(n_time * n_rep, 1, 0.1)),  # Control
    cumsum(rbinom(n_time * n_rep, 1, 0.15)), # Treatment A
    cumsum(rbinom(n_time * n_rep, 1, 0.2))   # Treatment B
  )
)

# Inspect data structure (this runs quickly)
head(example_data)
table(example_data$treatment)


# Full analysis (longer computation)
```

```
results <- SQUIRE(
  data = example_data,
  treatments = c("Control", "Treatment_A", "Treatment_B"),
  control_treatment = "Control",
  response_type = "germination",
  verbose = FALSE
)

# Check results
if(results$optimization_performed) {
  print("Optimization was justified")
  print(results$parameters)
} else {
  print("No significant effects detected")
  print(results$statistical_advice)
}
```

`summarize_biological_effects` *Summarize Biological Effects*

Description

Create overall biological summary

Usage

```
summarize_biological_effects(interpretation, response_type)
```

Arguments

`interpretation`	Parameter interpretation results
`response_type`	Type of biological response

Value

Biological summary text

`summary.SQUIRE` *SQUIRE Summary Method*

Description

Print method for SQUIRE results

Usage

```
## S3 method for class 'SQUIRE'
summary(object, ...)
```

Arguments

`object`	SQUIRE results object
`...`	Additional arguments

Value

No return value, called for side effects (prints summary to console)

`test_treatment_effects` *Test Treatment Effects*

Description

Statistical test for treatment differences

Usage

```
test_treatment_effects(data, treatments, alpha = 0.05, verbose = FALSE)
```

Arguments

`data`	Experimental data frame
`treatments`	Treatment names vector
`alpha`	Significance level
`verbose`	Print progress messages

Value

List with statistical test results

validate_biological_effects

Validate Biological Effects

Description

Test for statistically significant treatment effects before optimization

Usage

```
validate_biological_effects(
  data,
  treatments,
  control_treatment,
  response_type,
  alpha,
  min_timepoints,
  min_replicates,
  verbose
)
```

Arguments

`data`	Data frame with experimental data
`treatments`	Vector of treatment names
`control_treatment`	Name of control treatment
`response_type`	Type of biological response
`alpha`	Statistical significance level
`min_timepoints`	Minimum required timepoints
`min_replicates`	Minimum required replicates
`verbose`	Print progress messages

Value

List with validation results

validate_optimized_parameters *Validate Optimized Parameters*

Description

Test statistical significance of optimized parameters

Usage

```
validate_optimized_parameters(
  optimization_results,
  treatments,
  control_treatment,
  alpha = 0.05,
  verbose = FALSE
)
```

Arguments

`optimization_results`	Results from optimization
`treatments`	Vector of treatment names
`control_treatment`	Name of control treatment
`alpha`	Significance level
`verbose`	Print progress messages

Value

Parameter validation results

Index