

Package: inough (via r-universe)

May 14, 2026

Version 0.1.0

Title Inattention Detection Pipeline for Psychophysical Tasks

Description Three-stage pipeline for detecting inattention episodes in long psychophysical tasks (200+ trials). Uses accuracy residuals and response pattern signals to locate, sharpen, and formally test candidate inattention regions at trial-level precision.

License GPL (>= 3)

URL <https://github.com/pawlenartowicz/inough>

BugReports <https://github.com/pawlenartowicz/inough/issues>

Encoding UTF-8

Language en-US

Depends R (>= 3.5.0)

LazyData true

RoxygenNote 7.3.3

Imports lme4, ggplot2, patchwork, rlang, jsonlite

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs cmake make

Repository <https://pawlenartowicz.r-universe.dev>

Date/Publication 2026-05-09 14:53:53 UTC

RemoteUrl <https://github.com/pawlenartowicz/inough>

RemoteRef HEAD

RemoteSha 7ad3c7fa2ecb548c6553b3be503641772491a2a5

Contents

flags	2
inough_control	2
inough_detect	3
inough_heuristics	4
inough_signals	5
lz_complexity	6
plot.inough_detected	7
report	7
task_example	8
Index	10

flags	<i>Extract flagged trials</i>
-------	-------------------------------

Description

Returns a data frame of all trials flagged by the detection pipeline, suitable for downstream filtering via `dplyr::anti_join` or similar.

Usage

```
flags(x)
```

Arguments

`x` An `inough_detected` object.

Value

Data frame with columns `id`, `trial_idx`, `flag_type` ("bailout" or "chunk"), `chunk_id` (integer, NA for bailout), `p_adj` (numeric, NA for bailout).

inough_control	<i>Pipeline tuning parameters</i>
----------------	-----------------------------------

Description

Pipeline tuning parameters

Usage

```
inough_control(
  lz_threshold = 0.2,
  window_size = 3,
  sd_threshold = 2,
  window_weight = "uniform",
  min_chunk = 6,
  comparison = "clean"
)
```

Arguments

lz_threshold	LZ complexity below this triggers bail-out (default 0.2).
window_size	Half-width of rolling window; total window = 2*w+1 (default 3, giving 7-trial windows).
sd_threshold	Number of SDs above chance that roll_resp must exceed to flag a candidate region. Under H0 of balanced random responding, $SD = \sqrt{\sum(w^2)} / \sum(w)$ where w are the window weights. For uniform weights this simplifies to $1/\sqrt{2*window_size+1}$. Default 2 (about 5% false-positive rate per window under H0).
window_weight	Weighting scheme for the rolling window: "uniform" (default) or "triangular" (center trials weighted more; reduces edge sensitivity). The SD is computed analytically for both.
min_chunk	Minimum chunk length in trials to retain after merging (default 6).
comparison	t-test comparison set: "clean" (chunk vs all non-flagged trials) or "rest" (chunk vs everything except that chunk). Default "clean".

Value

An inough_control object with a pre-computed screening_threshold based on sd_threshold and the window.

inough_detect	<i>Detect inattention episodes</i>
---------------	------------------------------------

Description

Two-stage pipeline: (1) dual-track screening with chunk filtering, (2) formal t-test with FDR correction. Participants with extremely stereotyped responses or chance-level accuracy are bailed out first.

Usage

```
inough_detect(
  signals,
  fdr_alpha = 0.2,
  control = inough_control(),
  heuristics = inough_heuristics()
)
```

Arguments

signals	An inough_signals object from inough_signals .
fdr_alpha	FDR significance level for BH correction (default 0.05).
control	An inough_control object (see inough_control).
heuristics	An inough_heuristics object (see inough_heuristics). Controls boundary extension mode and spurious-accuracy trimming.

Value

An inough_detected object.

inough_heuristics	<i>Post-detection heuristics</i>
-------------------	----------------------------------

Description

Configures optional refinements applied after chunk screening: boundary extension mode and spurious-accuracy trimming.

Usage

```
inough_heuristics(
  boundary_mode = "heuristic",
  spurious = TRUE,
  spurious_n = 6L,
  spurious_k = NULL,
  min_left = 7L
)
```

Arguments

boundary_mode	How to extend chunk boundaries after detection: "heuristic" (default) walks backwards from chunk start to find where stereotyped responding actually began. "fixed" extends symmetrically (amount depends on window_weight in inough_control), "heuristic" walks backwards from chunk start to find where stereotyped responding actually began. End boundary always uses fixed extension.
---------------	--

spurious	Logical; enable spurious-start accuracy trimming (default TRUE). When TRUE, the first spurious_n trials of each chunk are checked: if accuracy is suspiciously high ($\geq k$ correct), those trials are excluded from the t-test.
spurious_n	Number of trials at chunk start to inspect (default 6).
spurious_k	Explicit threshold: flag if $\geq k$ correct out of spurious_n. When NULL (default), computed as $\text{ceiling}((0.5 + 1.5 * \text{sqrt}(0.25 / n)) * n)$.
min_left	Minimum trials remaining after spurious trimming (default 7). Chunks shorter than this after trimming are dropped.

Value

An inough_heuristics object.

inough_signals	<i>Extract inattention signals</i>
----------------	------------------------------------

Description

Fits a probit GLMM on accuracy and computes response bias indicators. This is the first step in the inough pipeline: call inough_signals, then pass the result to [inough_detect](#).

Usage

```
inough_signals(
  df,
  correct,
  response,
  id = "ID",
  learning_effect = TRUE,
  participant_effect = TRUE,
  trial_transform = "sqrt"
)
```

Arguments

df	A data frame with rows ordered by trial within each participant.
correct	Formula. LHS names the accuracy column (0/1 integer). RHS names design predictors that explain correctness (e.g., <code>correct ~ Stim + Weight + Orient + Block</code>).
response	Formula identifying the response column. Use a two-sided formula where the RHS names the column (e.g., <code>response ~ answ</code>). Must have exactly 2 distinct values; auto-encoded to 0/1 by sorted order.
id	String naming the participant identifier column (default "ID").
learning_effect	Logical. If TRUE (default), adds n_trial as fixed effect and random slope per participant.

`participant_effect`
 Logical. If TRUE (default), adds random intercept per participant.

`trial_transform`
 Transformation applied to trial index before rescaling to [-1, 1]. One of "sqrt" (default), "log", "linear", or a function.

Value

An `inough_signals` object.

<code>lz_complexity</code>	<i>Normalized Lempel-Ziv Complexity (LZ76)</i>
----------------------------	--

Description

Computes the normalized LZ76 complexity of a binary sequence. Higher values indicate more random/complex sequences; lower values indicate more predictable/repetitive patterns.

Usage

```
lz_complexity(x)
```

Arguments

`x` Integer vector of 0s and 1s.

Value

Numeric scalar in $\setminus[0, 1]$. Normalized complexity where 1 = maximally complex (random) and values near 0 = highly predictable.

Examples

```

lz_complexity(c(0, 0, 0, 0, 0))      # low
lz_complexity(c(0, 1, 0, 1, 0, 1))  # low-medium
lz_complexity(sample(0:1, 100, TRUE)) # near 1

```

plot.inough_detected *Plot per-participant diagnostic panels*

Description

Produces a stacked 4-panel visualization: accuracy strip, accuracy residuals, lag-1 response, and dual-track z-scores. Flagged chunks are highlighted as red shaded regions.

Usage

```
## S3 method for class 'inough_detected'  
plot(x, id, ...)
```

Arguments

x	An inough_detected object.
id	Character scalar — participant ID to plot.
...	Ignored.

Value

A patchwork object (invisibly).

report *Generate interactive HTML report*

Description

Creates a self-contained HTML file with a participant browser, diagnostic plots, chunk details, and summary statistics.

Usage

```
report(x, ...)  
  
## S3 method for class 'inough_detected'  
report(x, file = NULL, custom_plot = NULL, ...)
```

Arguments

x	An inough_detected object.
...	Arguments passed to methods.
file	Output file path. If NULL (default), uses a tempfile and opens in the browser.
custom_plot	Optional per-trial variable to show as an extra panel in the participant view. A list with three fields: <ul style="list-style-type: none"> • data: a data.frame with columns id, trial_idx, value. trial_idx is 1-based within participant, matching the trial order fed to inough_signals. • title: string shown as the panel label. • description: string shown as the panel blurb.

Value

Invisibly returns the file path.

task_example

Example dual-task data with diverse inattention profiles

Description

A minimal anonymized subset of the Dual Task (Gabor orientation under motor interference) dataset, intended for demonstrating the inough pipeline. Twenty participants were sampled to span a range of attention profiles: clean performers, two bail-out cases (one for response stereotypy, one for chance-level accuracy), participants with localized inattention chunks, and participants with extended inattention periods.

Usage

task_example

Format

A data frame with one row per trial and the following columns:

participant Anonymized participant identifier (factor, P01–P20).

block Block index within the session (integer, >= 1; practice block excluded).

trial Trial index within the participant's session (integer).

stim Stimulus identifier (integer).

weight Stimulus weight / contrast (numeric).

orient Gabor orientation code (integer).

cue_type Cue type code (integer).

response Participant's response (integer, two unique values).

correct Trial accuracy (integer, 0 or 1).

Details

Participant identifiers have been replaced with arbitrary codes (P01–P20) and any session timing information has been removed.

Source

A subset of the Dual Task (s_9) data collected in the COST/Kraken consciousness study (Krakow site). Participant IDs have been re-coded for anonymity.

Examples

```
data(task_example)
head(task_example)

signals <- inough_signals(
  task_example,
  correct = correct ~ stim + weight + orient + cue_type + block,
  response = response ~ response,
  id      = "participant"
)
det <- inough_detect(signals)
summary(det)
```

Index

* datasets

task_example, 8

flags, 2

inough_control, 2, 4

inough_detect, 3, 5

inough_heuristics, 4, 4

inough_signals, 4, 5

lz_complexity, 6

plot.inough_detected, 7

report, 7

task_example, 8