

# Package ‘Household.Transmission.Chain.Data.Analysis’

October 23, 2025

**Title** Household Transmission Chain Simulation and Estimation

**Version** 0.0.0.9000

**Description** This package provides a framework for household transmission chain analysis by a streamlined pipeline to simulate household transmission data and to estimate community and within-household infection risks from either synthetic or user-supplied data. The workflow builds person-day tables, impute infection timelines with Gamma-distributed delays, and fits penalized models with optional covariates. Post-processing summarizes mean estimates, uncertainty, bias, and relative bias. Two user functions cover the main tasks: GenSyn() to simulate data and run the full estimation pipeline (with optional summaries), and TransmissionChainAnalysis() to run the same estimation pipeline on user data.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Imports** data.table,  
stats,  
utils

**Suggests** knitr,  
rmarkdown,  
testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat.edition** 3

## Contents

build_person_day_table . . . . .	2
dataframe_to_household_list . . . . .	3
data_summarization . . . . .	4
generate_synthetic_data_one . . . . .	6
generate_synthetic_data_standardized . . . . .	8
GenSyn . . . . .	10
infectious_time_imputation . . . . .	14
main_parameter_estimation_pipeline . . . . .	15
postprocessing_estimates . . . . .	18
running_parameter_estimation . . . . .	19

simulate_households . . . . .	21
summarize_individuals . . . . .	23
TransmissionChainAnalysis . . . . .	24

**Index****27****`build_person_day_table`***Construct long format person day data***Description**

Builds a person day table for likelihood based parameter estimation by expanding individual level timelines into daily observations and computing same household infectious counts by infector role.

**Usage**

```
build_person_day_table(dt, tmax, cases_t, covariate_cols = character(0))
```

**Arguments**

dt	data.table. Individual level data with infection or imputation results from <a href="#">infectious_time_imputat</a>
tmax	Integer. Maximum day index
cases_t	Numeric vector of length tmax + 1. Community intensity on days 0 to tmax.
covariate_cols	Character vector. Names of row level covariates to copy.

**Value**

A long format data.table with columns agegrp2, agegrp3, agegrp4, n\_inf, n\_inf\_infant, n\_inf\_sibling, n\_inf\_adult, n\_inf\_elder, cases, event, ID\_indiv, ID\_hh, day, and the requested covariates.

**Examples**

```
## Not run:
long_dt <- build_person_day_table(imputed, tmax = 200, cases_t = rep(0.001, 201))
head(long_dt)

## End(Not run)
```

---

**dataframe\_to\_household\_list**  
*Convert a user data frame to a list of household tables*

---

## Description

Splits a long format data frame of test observations into a list of household level data tables. The function preserves required columns and allows any extra user provided covariates.

## Usage

```
dataframe_to_household_list(
  df,
  hh_col = "HH",
  id_col = "individual_ID",
  role_col = "role",
  date_col = "test_date",
  inf_col = "infection_status",
  comm_col = "community_risk",
  keep_extra_cols = TRUE
)
```

## Arguments

<code>df</code>	Data frame. Must contain at least columns for household ID, individual ID, role, test date, infection status, and community risk.
<code>hh_col, id_col, role_col, date_col, inf_col, comm_col</code>	Character. Column names for household ID (default "HH"), individual ID ("individual_ID"), role ("role"), test date ("test_date"), infection status ("infection_status"), and community risk ("community_risk").
<code>keep_extra_cols</code>	Logical. If TRUE (default), retain additional covariate columns.

## Value

A list of data tables, one per household. Columns are ordered with household ID first.

## Examples

```
## Not run:
hh_list <- dataframe_to_household_list(df)
length(hh_list); names(hh_list)[1]; head(hh_list[[1]])

## End(Not run)
```

<code>data_summarization</code>	<i>Summarize infection episodes (per individual)</i>
---------------------------------	--

## Description

Aggregates long-format testing records into one row per individual with episode counts, first/last positive within the first episode, inferred infectious-day span, last pre-positive negative, and optional covariate summaries. Accepts either a single data frame `df` containing the six required core columns, or the six vectors individually. Optionally merges an external covariate data frame before summarizing.

## Usage

```
data_summarization(
  df = NULL,
  Household_ID = NULL,
  Individual_ID = NULL,
  Household_role = NULL,
  Sample_test_days = NULL,
  Infectious_status = NULL,
  Community_rate_infection = NULL,
  Covariate_DataFrame = NULL,
  covariate_cols = NULL
)
```

## Arguments

<code>df</code>	Data frame with at least the columns <code>HH</code> , <code>individual_ID</code> , <code>role</code> , <code>test_date</code> , <code>infection_status</code> , <code>community_risk</code> . When <code>df</code> is supplied, the vector arguments below are ignored.
<code>Household_ID</code>	Integer vector. Household identifiers; used only when <code>df</code> is <code>NULL</code> .
<code>Individual_ID</code>	Integer vector. Individual identifiers; used only when <code>df</code> is <code>NULL</code> .
<code>Household_role</code>	Character vector. Household role for each row (e.g., "infant", "sibling", "adult", "elder"); used only when <code>df</code> is <code>NULL</code> .
<code>Sample_test_days</code>	Integer or numeric vector. Testing day index for each record (relative day); used only when <code>df</code> is <code>NULL</code> .
<code>Infectious_status</code>	Integer or logical vector (0/1). Test-time infection indicator; used only when <code>df</code> is <code>NULL</code> .
<code>Community_rate_infection</code>	Numeric vector. Community risk/intensity at each record; used only when <code>df</code> is <code>NULL</code> .
<code>Covariate_DataFrame</code>	Data frame of additional covariates to merge prior to summarization. The join uses the best available keys among <code>c("HH", "individual_ID", "test_date")</code> . At minimum, <code>individual_ID</code> must be present (preferably with <code>HH</code> and/or <code>test_date</code> ).
<code>covariate_cols</code>	Character vector of covariate names to summarize (after merging). If <code>NULL</code> (default), all non-core columns are summarized.

## Details

- Either df or all six vector inputs must be provided; otherwise an error is thrown.
- When merging Covariate\_DataFrame, name collisions with the six core columns are avoided by renaming the colliding covariate columns with a "\_cv" suffix.
- Covariate names are normalized to snake\_case prior to creating summary columns.
- Records are internally ordered by HH, individual\_ID, test\_date.
- Episode detection is based on runs of infection\_status == 1; the first episode is used for \*\_start, \*\_end, \*\_duration, and infection.infectious.day.

## Value

A data frame with one row per individual containing:

- HH: household identifier.
- individual\_ID: individual identifier (within-household).
- n.true.infection: number of infection episodes (runs of 1's).
- n.detected.infection: number of detected episodes (same as n.true.infection given infection\_status encodes detection).
- infection.detected.start: first positive day of the first episode.
- infection.detected.end: last positive day of the first episode.
- infection.true.duration: duration (days) of the first episode based on test days (end - start + 1).
- last\_negative: last negative test day prior to the first positive (if any).
- infection.infectious.day: comma-separated list of test-day indices during the first episode.
- community.risk: mean of community\_risk across this individual's records.
- role: household role (as observed).

If covariates are present, additional columns are appended per covariate (after name normalization to snake\_case):

- <cov>\_mode: mode for categorical/logical or binary-numeric covariates.
- <cov>\_first, <cov>\_mean, <cov>\_last: first observed, mean, and last observed value for numeric covariates.
- <cov>\_timevarying: logical flag indicating whether the covariate varies over time for the individual.

## See Also

[generate\\_synthetic\\_data\\_standardized](#), [summarize\\_individuals](#), [main\\_parameter\\_estimation\\_pipeline](#)

## Examples

```
## Not run:
# Minimal example with a data frame:
df <- data.frame(
  HH = c(1,1,1,1,1),
  individual_ID = c(1,1,1,2,2,2),
  role = c("infant", "infant", "infant", "adult", "adult", "adult"),
  test_date = c(1,2,3,1,2,3),
```

```

infection_status = c(0,1,1,0,0,1),
community_risk = c(0.01,0.02,0.02,0.01,0.01,0.02)
)
data_summarization(df)

# With external covariates to merge and summarize:
covdf <- data.frame(
  HH = c(1,1), individual_ID = c(1,2),
  vacc_status = c(1,0), bmi = c(18.5, 27.2)
)
data_summarization(df, Covariate_DataFrame = covdf)

## End(Not run)

```

### generate\_synthetic\_data\_one

*Generate standardized synthetic data for one household*

### Description

Simulates testing observations for a single household across a date range with community risk using seasonal forcing, within-household transmission, adaptive testing, baseline and partial immunity, and optional covariates.

### Usage

```

generate_synthetic_data_one(
  household_id,
  hh.size = sample(3:7, 1),
  tests.per.week = 2,
  p.comm.base.infant.fix = 0.001,
  p.comm.multiplier.sibling = 1,
  p.comm.multiplier.parent = 1,
  p.comm.multiplier.elder = 1,
  p.hh.base.infant = 0.1,
  p.hh.multiplier.sibling = 1,
  p.hh.multiplier.parent = 1,
  p.hh.multiplier.elder = 1,
  p.imm.base.sibling = 1e-10,
  p.imm.base.parent = 1e-10,
  p.imm.base.elder = 1e-10,
  partial.immunity.infant = 1e-10,
  partial.immunity.sibling = 1e-10,
  partial.immunity.parent = 1e-10,
  partial.immunity.elder = 1e-10,
  duration.latent = 2,
  duration.infect.inf = 3,
  multiplier.dur.sibpar = 0.5,
  p.detect = 0.999,
  amplitude = 0,
  phase = -0.408,

```

```

    start_date = as.Date("2024-09-21"),
    end_date = as.Date("2025-04-17"),
    Covariates = FALSE,
    Covariates_list = c("Vaccination status", "Antibody Level"),
    Covariate_specifications = NULL
)

```

## Arguments

**household\_id** Integer. Household identifier used to populate the HH column.  
**hh.size** Integer. Household size.  
**tests.per.week** Integer. Tests per person per week.  
**p.comm.base.infant.fix** Numeric. Baseline community infection probability for infants per day.  
**p.comm.multiplier.sibling,** **p.comm.multiplier.parent,**  
**p.comm.multiplier.elder** Numeric multipliers relative to the infant baseline for sibling, parent (adult), and elder.  
**p.hh.base.infant** Numeric. Baseline within-household transmission probability per infectious infant contact per day.  
**p.hh.multiplier.sibling,** **p.hh.multiplier.parent,**  
**p.hh.multiplier.elder** Multipliers relative to the infant baseline for sibling, parent (adult), and elder.  
**p.imm.base.sibling, p.imm.base.parent, p.imm.base.elder** Numeric. Baseline immunity probabilities at day 1 by role.  
**partial.immunity.infant,** **partial.immunity.sibling,**  
**partial.immunity.parent, partial.immunity.elder** Numeric. Partial-immunity modifiers by role.  
**duration.latent** Integer. Mean latent period (days).  
**duration.infect.inf** Integer. Mean infectious duration for infants (days).  
**multiplier.dur.sibpar** Numeric. Infectious-duration multiplier for non-infant roles.  
**p.detect** Numeric. Probability that an infection present on a testing day is detected.  
**amplitude, phase** Numeric. Seasonality parameters for community risk.  
**start\_date, end\_date** Date. Simulation window.  
**Covariates** Logical. If TRUE, generate additional covariates.  
**Covariates\_list** Character vector. Names of covariates to generate.  
**Covariate\_specifications** List. Optional per-covariate specifications.

**Value**

A data frame with columns:

- HH household ID
- individual\_ID individual index within the household
- role one of "infant", "sibling", "adult", or "elder"
- test\_date day index where 1 maps to start\_date
- infection\_status 0/1 at the test day
- community\_risk community infection intensity on that day
- additional covariate columns when Covariates = TRUE

**Examples**

```
## Not run:
df1 <- generate_synthetic_data_one(1, hh.size = 4, tests.per.week = 2, Covariates = TRUE)
head(df1)

## End(Not run)
```

`generate_synthetic_data_standardized`

*(Deprecated) generate\_synthetic\_data\_standardized*

**Description**

This function is retained for backward compatibility. It forwards its arguments to [simulate\\_households](#) and emits a deprecation notice.

**Usage**

```
generate_synthetic_data_standardized(
  n_households,
  hh.size = sample(3:7, 1),
  tests.per.week,
  p.comm.base.infant.fix = 0.001,
  p.comm.multiplier.sibling = 1,
  p.comm.multiplier.parent = 1,
  p.comm.multiplier.elder = 1,
  p.hh.base.infant = 0.1,
  p.hh.multiplier.sibling = 1,
  p.hh.multiplier.parent = 1,
  p.hh.multiplier.elder = 1,
  p.imm.base.sibling = 1e-10,
  p.imm.base.parent = 1e-10,
  p.imm.base.elder = 1e-10,
  partial.immunity.infant = 1e-10,
  partial.immunity.sibling = 1e-10,
  partial.immunity.parent = 1e-10,
  partial.immunity.elder = 1e-10,
```

```

duration.latent = 2,
duration.infect.inf = 3,
multiplier.dur.sibpar = 0.5,
p.detect = 0.999,
amplitude = 0,
phase = -0.408,
start_date = as.Date("2024-09-21"),
end_date = as.Date("2025-04-17"),
Covariates = FALSE,
Covariates_list = c("Vaccination status", "Antibody Level"),
Covariate_specifications = NULL
)

```

## Arguments

n\_households Integer. Number of households to simulate.

hh.size Integer. Household size applied to each simulated household.

tests.per.week Integer. Tests per person per week.

p.comm.base.infant.fix Numeric. Baseline community infection probability for infants per day.

p.comm.multiplier.sibling, p.comm.multiplier.parent,  
p.comm.multiplier.elder Numeric multipliers applied to the infant baseline for sibling, parent (adult), and elder, respectively.

p.hh.base.infant Numeric. Baseline within household transmission probability per infectious infant contact per day.

p.hh.multiplier.sibling, p.hh.multiplier.parent,  
p.hh.multiplier.elder Multipliers relative to the infant baseline for sibling, parent (adult), and elder.

p.imm.base.sibling, p.imm.base.parent, p.imm.base.elder Numeric. Baseline immunity probabilities at day 1 by role.

partial.immunity.infant, partial.immunity.sibling,  
partial.immunity.parent, partial.immunity.elder Numeric. Partial immunity by role.

duration.latent Integer. Mean latent period in days.

duration.infect.inf Integer. Mean infectious duration for infants in days.

multiplier.dur.sibpar Numeric. Infectious duration multiplier for non infant roles.

p.detect Numeric. Probability that an infection present on a testing day is detected,

amplitude, phase Numeric. Seasonality parameters for community risk.

start\_date, end\_date Date. Simulation window.

Covariates Logical. If TRUE, generate additional covariates.

Covariates\_list Character vector. Names of covariates to generate.

Covariate\_specifications List. Optional per covariate specifications.

## Details

Deprecated alias for [simulate\\_households](#).

## Value

See [simulate\\_households](#).

## See Also

[simulate\\_households](#)

## Examples

```
## Not run:
sims <- generate_synthetic_data_standardized(
  n_households = 5,
  simulation_function = generate_synthetic_data_one,
  hh.size = 4,
  tests.per.week = 2
)
length(sims); head(sims[[1]])

## End(Not run)
```

GenSyn

*Simulate households, estimate transmission, and (optionally) summarize*

## Description

Generates synthetic household transmission data and runs the full parameter estimation pipeline. By default, it prints a post-processing table comparing the mean estimates to “true” values (bias and relative bias). Optionally, it prints a per-individual data summary produced by the pipeline.

## Usage

```
GenSyn(
  synthetic_data = TRUE,
  data_summary = FALSE,
  postprocessing = TRUE,
  n_households = 10,
  n_runs = 10,
  hh.size = sample(3:7, 1),
  tests.per.week = 1,
  Covariates = FALSE,
  Covariates_list = c("Vaccination status", "Antibody Level"),
  Covariate_specifications = NULL,
  day_series_covariates = TRUE,
  series_cols = NULL,
  comm_covariate_cols = NULL,
  hh_covariate_cols = NULL,
  hh_by_role = FALSE,
```

```

hh_role_covariate_cols = NULL,
standardize_covariates = TRUE,
lambda_comm = 0.01,
lambda_hh = 0.01,
p.comm.base.infant.fix = 0.002,
p.comm.multiplier.sibling = 1,
p.comm.multiplier.parent = 1,
p.comm.multiplier.elder = 1,
p.hh.base.infant = 0.2,
p.hh.multiplier.sibling = 0.5267686,
p.hh.multiplier.parent = 0.8008933,
p.hh.multiplier.elder = 0.6008933,
p.imm.base.sibling = 1e-10,
p.imm.base.parent = 1e-10,
p.imm.base.elder = 1e-10,
partial.immunity.infant = 1e-10,
partial.immunity.sibling = 1e-10,
partial.immunity.parent = 1e-10,
partial.immunity.elder = 1e-10,
duration.latent = 1,
duration.infect.inf = 2,
multiplier.dur.sibpar = 0.5,
p.detect = 0.999,
amplitude = 2.6581 * 0,
phase = -0.408,
start_date = as.Date("2024-09-21"),
end_date = as.Date("2025-04-17"),
latent_par = list(shape = 2, scale = 1),
report_par = list(shape = 1, scale = 1.5),
infect_par = list(shape = 3, scale = 2),
start_par = c(-6, 0.02, -2, rep(0, 6)),
lambda = 0.01,
lambda0 = 0.2,
lambda_alpha = 5,
delta0_true = qlogis(0.002),
alpha0_true = qlogis(0.2),
true_values = c(delta0 = log(7.148217e-05), gamma2 = log(7.148217e-05 * 4.331956) -
log(7.148217e-05), gamma3 = log(7.148217e-05 * 1.835466) - log(7.148217e-05), gamma4 =
log(7.148217e-05 * 2) - log(7.148217e-05), alpha0 = log(0.2888953), beta2 =
log(0.2888953 * 0.5267686) - log(0.2888953), beta3 = log(0.2888953 * 0.8008933) -
log(0.2888953), beta4 = log(0.2888953 * 0.6008933) - log(0.2888953))
)

```

## Arguments

<code>synthetic_data</code>	Logical. Must be TRUE. If FALSE, an error is thrown with guidance to use <a href="#">TransmissionChainAnalysis</a> .
<code>data_summary</code>	Logical. If TRUE, prints <code>results\$summarized_data</code> . Default FALSE.
<code>postprocessing</code>	Logical. If TRUE, prints <code>postprocessing_estimates</code> . Default TRUE.
<code>n_households</code>	Integer. Number of households to simulate. Default 10.
<code>n_runs</code>	Integer. Repeated estimation runs. Default 10.

**hh.size** Integer. Household size (constant across HHs unless a random draw is passed). Default `sample(3:7, 1)`.

**tests.per.week** Integer. Tests per person per week (1–3). Default 1.

**Covariates** Logical. Generate synthetic covariates. Default FALSE.

**Covariates\_list** Character vector of covariate names to generate when Covariates=TRUE. Default `c("Vaccination status", "Antibody Level")`.

**Covariate\_specifications** List with per-covariate generation specs (type, dist, time\_varying, params).

**day\_series\_covariates** Logical. If TRUE, builds day-series list-columns for covariates in the person–day table. Default TRUE.

**series\_cols** Character vector of covariate names (pre-normalization) to build as day-series; if NULL, all detected covariates are used.

**comm\_covariate\_cols** Character vector of community-level covariates (no intercept) for the likelihood.

**hh\_covariate\_cols** Character vector of household covariates shared across roles.

**hh\_by\_role** Logical. If TRUE, fit role-specific household covariates.

**hh\_role\_covariate\_cols** Named list with elements `infant`, `sibling`, `adult`, `elder` providing covariate names per role when `hh_by_role`=TRUE.

**standardize\_covariates** Logical. Z-score non-binary numeric columns in model matrices. Default TRUE.

**lambda\_comm, lambda\_hh** Numeric. L2 penalties for community and household covariate coefficients.

**p.comm.base.infant.fix, p.comm.multiplier.sibling, p.comm.multiplier.parent, p.comm.multiplier.elder** Community infection baseline and role multipliers used in simulation.

**p.hh.base.infant, p.hh.multiplier.sibling, p.hh.multiplier.parent, p.hh.multiplier.elder** Household transmission baseline and role multipliers used in simulation.

**p.imm.base.sibling, p.imm.base.parent, p.imm.base.elder** Baseline immunity probabilities.

**partial.immunity.infant, partial.immunity.sibling, partial.immunity.parent, partial.immunity.elder** Breakthrough modifiers for partially immune individuals.

**duration.latent, duration.infect.inf, multiplier.dur.sibpar** Natural-history durations used in simulation.

**p.detect** Detection probability in testing algorithm.

**amplitude, phase** Seasonal modulation of community risk.

**start\_date, end\_date** Simulation window as Date.

**latent\_par, report\_par, infect\_par** Lists with Gamma(shape, scale) parameters for imputation of latent delay, reporting delay, and infectious period.

**start\_par** Numeric vector of initial parameters (auto-expanded if length mismatches model).

lambda	Base L2 penalty for slope-like parameters (age multipliers and household role offsets).
lambda0	Penalty anchoring $\delta_0$ near $\delta_0_{\text{true}}$ .
lambda_alpha	Penalty anchoring $\alpha_0$ near $\alpha_0_{\text{true}}$ .
$\delta_0_{\text{true}}, \alpha_0_{\text{true}}$	Numeric anchors (logit scale) for community baseline and household baseline.
true_values	Optional named numeric vector of reference values. Names should match columns in <code>theta_mat</code> . Unmatched names are ignored; missing references yield NA bias/relative bias.

## Details

This function is a thin wrapper around `main_parameter_estimation_pipeline` with `synthetic_data=TRUE`. Any extra columns generated (e.g., covariates) are automatically carried into the summarization and may be used in the likelihood via the covariate mapping arguments.

## Value

(Invisibly) a list with elements:

- **results**: Output of `main_parameter_estimation_pipeline` (raw simulations, summaries, person-day, estimates).
- **postprocessing**: The table returned by `postprocessing_estimates` (or NULL if `postprocessing=FALSE`).

## Workflow

1. Simulate households and test records.
2. Summarize individuals and impute infection timelines.
3. Build a person–day table and run repeated ML estimation.
4. (Optional) Print data summary and post-processing table.

## See Also

`TransmissionChainAnalysis`, `main_parameter_estimation_pipeline`, `postprocessing_estimates`

## Examples

```
## Not run:
out <- GenSyn(n_households = 50, n_runs = 20, Covariates = TRUE)
names(out)
head(out$results$person_day)

## End(Not run)
```

**infectious\_time\_imputation***Impute infection timelines from delay distributions***Description**

Imputes infection date, infectious start and end, and component delays using gamma distributions, optionally scaled by user functions of covariates.

**Usage**

```
infectious_time_imputation(
  dt,
  study_start,
  latent_par,
  report_par,
  infect_par,
  latent_scale_fn = NULL,
  report_scale_fn = NULL,
  infect_scale_fn = NULL
)
```

**Arguments**

<code>dt</code>	data.table. Individual level summary from <a href="#">summarize_individuals</a> .
<code>study_start</code>	Date. Reference origin for relative day indices.
<code>latent_par, report_par, infect_par</code>	Lists with elements <code>shape</code> and <code>scale</code> for the latent, reporting, and infectious period gamma distributions.
<code>latent_scale_fn, report_scale_fn, infect_scale_fn</code>	Optional functions that take <code>dt[idx]</code> for infected rows and return numeric multipliers for the corresponding scales.

**Value**

The input `dt` with added or imputed columns: `latent_delay`, `report_delay`, `infect_period`, `inf_date`, `inf_start_date`, `inf_end_date`, and relative day versions `inf_day_rl`, `infectious_day_rl`, `infectious_end_day_rl`.

**Examples**

```
## Not run:
imputed <- infectious_time_imputation(
  dt = ind,
  study_start = as.Date("2024-09-21"),
  latent_par = list(shape = 2, scale = 1),
  report_par = list(shape = 1, scale = 1.5),
  infect_par = list(shape = 3, scale = 2)
)

## End(Not run)
```

---

**main\_parameter\_estimation\_pipeline**  
*Main parameter estimation pipeline*

---

**Description**

End-to-end pipeline that (i) obtains household testing data (either simulated or user-supplied), (ii) summarizes individuals, (iii) imputes infection timelines, (iv) expands to a person-day table, and (v) runs repeated maximum-likelihood estimation with penalties.

**Usage**

```
main_parameter_estimation_pipeline(
  user_data = NULL,
  synthetic_data = TRUE,
  n_households = 10,
  n_runs = 10,
  hh.size = sample(3:7, 1),
  tests.per.week = 1,
  Covariates = FALSE,
  Covariates_list = c("Vaccination status", "Antibody Level"),
  Covariate_specifications = NULL,
  day_series_covariates = TRUE,
  series_cols = NULL,
  comm_covariate_cols = NULL,
  hh_covariate_cols = NULL,
  hh_by_role = FALSE,
  hh_role_covariate_cols = NULL,
  standardize_covariates = TRUE,
  lambda_comm = 0.01,
  lambda_hh = 0.01,
  p.comm.base.infant.fix = 0.002,
  p.comm.multiplier.sibling = 1,
  p.comm.multiplier.parent = 1,
  p.comm.multiplier.elder = 1,
  p.hh.base.infant = 0.2,
  p.hh.multiplier.sibling = 0.5267686,
  p.hh.multiplier.parent = 0.8008933,
  p.hh.multiplier.elder = 0.6008933,
  p.imm.base.sibling = 1e-10,
  p.imm.base.parent = 1e-10,
  p.imm.base.elder = 1e-10,
  partial.immunity.infant = 1e-10,
  partial.immunity.sibling = 1e-10,
  partial.immunity.parent = 1e-10,
  partial.immunity.elder = 1e-10,
  duration.latent = 1,
  duration.infect.inf = 2,
  multiplier.dur.sibpar = 0.5,
  p.detect = 0.999,
  amplitude = 2.6581 * 0,
```

```

phase = -0.408,
start_date = as.Date("2024-09-21"),
end_date = as.Date("2025-04-17"),
latent_par = list(shape = 2, scale = 1),
report_par = list(shape = 1, scale = 1.5),
infect_par = list(shape = 3, scale = 2),
start_par = c(-6, 0.02, -2, rep(0, 6)),
lambda = 0.01,
lambda0 = 0.2,
lambda_alpha = 5,
delta0_true = qlogis(0.002),
alpha0_true = qlogis(0.2)
)

```

## Arguments

<code>user_data</code>	List or <code>data.frame</code> . User-provided observations. If a list, it should contain one <code>data frame</code> per household; if a single <code>data.frame</code> , it must contain at least columns <code>HH</code> , <code>individual_ID</code> , <code>role</code> , <code>test_date</code> , <code>infection_status</code> , and <code>community_risk</code> . Ignored when <code>synthetic_data</code> = <code>TRUE</code> . Default <code>NULL</code> .
<code>synthetic_data</code>	Logical. If <code>TRUE</code> (default), simulate data using <code>generate_synthetic_data_standardized</code> via <code>simulate_households</code> ; otherwise, convert <code>user_data</code> with <code>dataframe_to_household_list</code> .
<code>n_households</code>	Integer. Number of households to simulate (used only when <code>synthetic_data</code> = <code>TRUE</code> ). Default 10.
<code>n_runs</code>	Integer. Number of repeated estimation runs. Default 10.
<code>hh.size</code>	Integer. Household size for simulation. Default <code>sample(3:7, 1)</code> .
<code>tests.per.week</code>	Integer. Tests per person per week (1–3). Default 1.
<code>Covariates</code>	Logical. If <code>TRUE</code> , generate additional covariates in simulation.
<code>Covariates_list</code>	Character vector. Names of covariates to generate.
<code>Covariate_specifications</code>	List. Per-covariate specifications; see <code>generate_synthetic_data_one</code> for schema and defaults.
<code>day_series_covariates</code>	Logical. If <code>TRUE</code> , build day-series list-columns for covariates during summarization (see <code>summarize_individuals</code> ).
<code>series_cols</code>	Character vector or <code>NULL</code> . Names (pre-normalization) of covariates to include as day-series; if <code>NULL</code> , uses all detected covariates.
<code>comm_covariate_cols</code>	Character vector. Column names included in the community risk model (no intercept).
<code>hh_covariate_cols</code>	Character vector. Household covariates shared across roles.
<code>hh_by_role</code>	Logical. If <code>TRUE</code> , allow role-specific household covariates.
<code>hh_role_covariate_cols</code>	List. Named list with elements <code>infant</code> , <code>sibling</code> , <code>adult</code> , <code>elder</code> , each a character vector of columns; falls back to <code>hh_covariate_cols</code> when missing.
<code>standardize_covariates</code>	Logical. Z-score non-binary numeric columns in model matrices.

```

lambda_comm, lambda_hh
    Numeric. L2 penalties for community and household covariate coefficients. Defaults 0.01.
p.comm.base.infant.fix
    Numeric. Baseline community infection probability per day for infants.
p.comm.multiplier.sibling, p.comm.multiplier.parent,
p.comm.multiplier.elder
    Numeric multipliers applied to p.comm.base.infant.fix by role.
p.hh.base.infant
    Numeric. Baseline within-household transmission probability per infectious infant contact (per day).
p.hh.multiplier.sibling, p.hh.multiplier.parent,
p.hh.multiplier.elder
    Multipliers relative to infant baseline by role.
p.imm.base.sibling, p.imm.base.parent, p.imm.base.elder
    Numeric. Baseline (day-1) immunity probabilities by role.
partial.immunity.infant, partial.immunity.sibling,
partial.immunity.parent, partial.immunity.elder
    Numeric. Breakthrough scaling (0–1) if previously immune, by role.
duration.latent
    Integer. Mean latent period (days).
duration.infect.inf
    Integer. Mean infectious duration for infants (days). Other roles use duration.infect.inf * multiplier.dur.sibpar.
multiplier.dur.sibpar
    Numeric. Infectious duration multiplier for non-infant roles.
p.detect
    Numeric. Detection probability on testing days.
amplitude, phase
    Numeric. Seasonality parameters (cosine forcing) for community risk.
start_date, end_date
    Date. Simulation/analysis window (inclusive of start_date).
latent_par, report_par, infect_par
    Lists with elements shape and scale for the latent delay, reporting delay, and infectious period gammas.
start_par
    Numeric vector. Initial parameter vector. If not tailored to the covariate layout, it is resized internally when passed to the optimizer.
lambda
    Numeric. Base L2 penalty for gamma (age) and z_* (role offsets).
lambda0, lambda_alpha
    Numeric. Penalties pulling delta0 and alpha0 toward delta0_true and alpha0_true.
delta0_true, alpha0_true
    Numeric anchors for the intercept penalties (logit scale).

```

## Details

When `synthetic_data` = TRUE, data are produced by `generate_synthetic_data_standardized` via `simulate_households`. When FALSE, `user_data` is converted with `dataframe_to_household_list`. Covariate names are normalized to snake\_case during summarization.

Community intensity `cases_t` used by the person-day builder is generated internally as a smoothed, scaled seasonal signal over the analysis window.

**Value**

A list with elements:

- `raw_simulation`: list of household data frames (simulated or user-provided).
- `summarized_data`: individual-level summary table (from `summarize_individuals` then `infectious_time_imputation`).
- `person_day`: long person-day table (from `build_person_day_table`).
- `estimates`:  $n\_runs \times p$  matrix of parameter estimates (from `running_parameter_estimation`).

**See Also**

`simulate_households`, `generate_synthetic_data_standardized`, `summarize_individuals`, `infectious_time_imputation`, `build_person_day_table`, `running_parameter_estimation`, `postprocessing_estimates`

**Examples**

```
## Not run:
res <- main_parameter_estimation_pipeline(
  synthetic_data = TRUE,
  n_households = 20, n_runs = 5,
  hh.size = 4, tests.per.week = 2,
  delta0_true = qlogis(0.002), alpha0_true = qlogis(0.2)
)
str(res$estimates)

## End(Not run)
```

**postprocessing\_estimates**

*Summarize and compare estimates against reference values*

**Description**

Computes mean, SD, SE across runs and (optionally) bias and relative bias versus supplied reference (true) values for each parameter.

**Usage**

```
postprocessing_estimates(
  theta_mat,
  true_values = c(delta0 = log(7.148217e-05), gamma2 = log(7.148217e-05 * 4.331956) -
    log(7.148217e-05), gamma3 = log(7.148217e-05 * 1.835466) - log(7.148217e-05), gamma4 =
    log(7.148217e-05 * 2) - log(7.148217e-05), alpha0 = log(0.2888953), beta2 =
    log(0.2888953 * 0.5267686) - log(0.2888953), beta3 = log(0.2888953 * 0.8008933) -
    log(0.2888953), beta4 = log(0.2888953 * 0.6008933) - log(0.2888953))
```

## Arguments

- `theta_mat` Numeric matrix. Parameter estimates from multiple runs (e.g., output of [running\\_parameter\\_estimation](#)). Must be non-empty.
- `true_values` Optional named numeric vector of reference values. Names should match columns in `theta_mat`. Unmatched names are ignored; missing references yield NA bias/relative bias.

## Details

Rows with incomplete estimates are dropped using the first column as a sentinel. If `theta_mat` lacks column names, generic names "par1", "par2", ... are assigned. The number of retained runs is stored in the "n\_runs" attribute of the returned table.

## Value

A `data.table` with one row per parameter and columns:

- Parameter: column name from `theta_mat`
- Estimate: mean across runs
- SD: standard deviation across runs
- SE: standard error ( $SD / \sqrt{n\_runs}$ )
- True: supplied reference value (if any)
- Bias: Estimate - True
- RelBias: Bias / |True| (NA when True is 0 or missing)
- Block, Role: simple parameter grouping labels for readability

## See Also

[main\\_parameter\\_estimation\\_pipeline](#), [running\\_parameter\\_estimation](#)

## Examples

```
## Not run:
est <- matrix(rnorm(50), nrow = 10)
colnames(est) <- c("delta0","gamma2","alpha0","z_sib","theta_comm_x")
postprocessing_estimates(est, true_values = c(delta0 = -9.5, alpha0 = -1.2))

## End(Not run)
```

## Description

Repeats optimization `n_runs` times from jittered starts to estimate community and household transmission parameters with optional covariate effects and quadratic penalties.

**Usage**

```
running_parameter_estimation(
  long_dt,
  n_runs,
  start_par = NULL,
  lambda,
  lambda0,
  lambda_alpha,
  delta0_true,
  alpha0_true,
  comm_covariate_cols = NULL,
  hh_covariate_cols = NULL,
  hh_by_role = FALSE,
  hh_role_covariate_cols = NULL,
  standardize_covariates = FALSE,
  lambda_comm = lambda,
  lambda_hh = lambda,
  verbose = TRUE
)
```

**Arguments**

long_dt	data.table or data.frame. Person day data from <a href="#">build_person_day_table</a> .
n_runs	Integer. Number of optimization runs for multi start.
start_par	Numeric vector or NULL. Initial parameters. If NULL, a vector of the correct length is initialized with delta0_true and alpha0_true in the intercept slots and zeros elsewhere.
lambda	Numeric. Base L2 penalty on gamma age effects and z_* role offsets.
lambda0, lambda_alpha	Numeric. Penalty strengths that pull delta0 and alpha0 toward delta0_true and alpha0_true.
delta0_true, alpha0_true	Numeric anchors for the intercept penalties on the logit scale.
comm_covariate_cols	Character vector of community risk covariate names with no intercept.
hh_covariate_cols	Character vector of household covariates shared across roles.
hh_by_role	Logical. If TRUE, allow role specific household covariates through hh_role_covariate_cols.
hh_role_covariate_cols	Named list with elements infant, sibling, adult, elder that give covariate names per role. Falls back to hh_covariate_cols if a role list is missing.
standardize_covariates	Logical. Z score non binary numeric columns in model matrices.
lambda_comm, lambda_hh	Numeric. L2 penalties for community and household covariate coefficients.
verbose	Logical. If TRUE, print notes on dropped or unknown covariates and initialization.

### Value

A numeric matrix of dimension n\_runs by n\_parameters with column names that match the parameter layout, for example delta0, gamma2, alpha0, z\_sib, and theta\_\*.

### Examples

```
## Not run:
est <- running_parameter_estimation(
  long_dt, n_runs = 5,
  lambda = 0.01, lambda0 = 0.2, lambda_alpha = 5,
  delta0_true = qlogis(0.002), alpha0_true = qlogis(0.2)
)
colnames(est)

## End(Not run)
```

**simulate\_households**    *Simulate synthetic household data (list output)*

### Description

Simulates independent households by repeatedly calling a single household simulation function (default: [generate\\_synthetic\\_data\\_one](#)). Returns a list of long format data frames, one per household.

### Usage

```
simulate_households(
  n_households,
  simulation_function = generate_synthetic_data_one,
  hh.size = sample(3:7, 1),
  tests.per.week,
  p.comm.base.infant.fix = 0.001,
  p.comm.multiplier.sibling = 1,
  p.comm.multiplier.parent = 1,
  p.comm.multiplier.elder = 1,
  p.hh.base.infant = 0.1,
  p.hh.multiplier.sibling = 1,
  p.hh.multiplier.parent = 1,
  p.hh.multiplier.elder = 1,
  p.imm.base.sibling = 1e-10,
  p.imm.base.parent = 1e-10,
  p.imm.base.elder = 1e-10,
  partial.immunity.infant = 1e-10,
  partial.immunity.sibling = 1e-10,
  partial.immunity.parent = 1e-10,
  partial.immunity.elder = 1e-10,
  duration.latent = 2,
  duration.infect.inf = 3,
  multiplier.dur.sibpar = 0.5,
  p.detect = 0.999,
```

```

  amplitude = 0,
  phase = -0.408,
  start_date = as.Date("2024-09-21"),
  end_date = as.Date("2025-04-17"),
  Covariates = FALSE,
  Covariates_list = c("Vaccination status", "Antibody Level"),
  Covariate_specifications = NULL
)

```

## Arguments

**n\_households** Integer. Number of households to simulate.  
**simulation\_function** Function. The single household simulation function to call.  
**hh.size** Integer. Household size applied to each simulated household.  
**tests.per.week** Integer. Tests per person per week.  
**p.comm.base.infant.fix** Numeric. Baseline community infection probability for infants per day.  
**p.comm.multiplier.sibling**, **p.comm.multiplier.parent**,  
**p.comm.multiplier.elder** Numeric multipliers applied to the infant baseline for sibling, parent (adult), and elder, respectively.  
**p.hh.base.infant** Numeric. Baseline within household transmission probability per infectious infant contact per day.  
**p.hh.multiplier.sibling**, **p.hh.multiplier.parent**,  
**p.hh.multiplier.elder** Multipliers relative to the infant baseline for sibling, parent (adult), and elder.  
**p.imm.base.sibling**, **p.imm.base.parent**, **p.imm.base.elder** Numeric. Baseline immunity probabilities at day 1 by role.  
**partial.immunity.infant**, **partial.immunity.sibling**,  
**partial.immunity.parent**, **partial.immunity.elder** Numeric. Partial immunity by role.  
**duration.latent** Integer. Mean latent period in days.  
**duration.infect.inf** Integer. Mean infectious duration for infants in days.  
**multiplier.dur.sibpar** Numeric. Infectious duration multiplier for non infant roles.  
**p.detect** Numeric. Probability that an infection present on a testing day is detected,  
**amplitude, phase** Numeric. Seasonality parameters for community risk.  
**start\_date, end\_date** Date. Simulation window.  
**Covariates** Logical. If TRUE, generate additional covariates.  
**Covariates\_list** Character vector. Names of covariates to generate.  
**Covariate\_specifications** List. Optional per covariate specifications.

### Value

A list of length n\_households. Each element is a data frame with columns HH, individual\_ID, role, test\_date, infection\_status, community\_risk, and any generated covariates.

### See Also

[generate\\_synthetic\\_data\\_one](#), [generate\\_synthetic\\_data\\_standardized](#)

### Examples

```
## Not run:
sims <- simulate_households(5, hh.size = 4, tests.per.week = 2)
length(sims); head(sims[[1]])

## End(Not run)
```

**summarize\_individuals** *Summarize individual level infection data*

### Description

Produces one row per individual with infection windows, index case flags, observation bounds, and covariate summaries. Optionally builds day series list columns for selected covariates.

### Usage

```
summarize_individuals(
  raw_dt,
  study_start,
  study_end,
  day_series_covariates = TRUE,
  series_cols = NULL
)
```

### Arguments

raw_dt	List of data frames or data tables, one per household.
study_start, study_end	Date. Analysis window used for relative day indexing.
day_series_covariates	Logical. If TRUE, attach day series list columns for covariates.
series_cols	Character vector or NULL. Which covariates to build day series for. If NULL, all detected covariates are considered.

### Value

A `data.table` with one row per individual including infection detection start and end, inferred infectious window using relative day indices, index case flag, observation window, role and age category, and per person covariate summaries.

## Examples

```
## Not run:
ind <- summarize_individuals(sims, as.Date("2024-09-21"), as.Date("2025-04-17"))
head(ind)

## End(Not run)
```

## TransmissionChainAnalysis

*Estimate transmission parameters from user data*

## Description

Runs the full parameter-estimation pipeline on user-provided long-format testing data. By default, prints a post-processing table (mean estimates, SD/SE, bias, relative bias). Optionally prints the per-individual summary produced by the pipeline.

## Usage

```
TransmissionChainAnalysis(
  user_data,
  data_summary = FALSE,
  postprocessing = TRUE,
  n_runs = 10,
  day_series_covariates = TRUE,
  series_cols = NULL,
  comm_covariate_cols = NULL,
  hh_covariate_cols = NULL,
  hh_by_role = FALSE,
  hh_role_covariate_cols = NULL,
  standardize_covariates = TRUE,
  lambda_comm = 0.01,
  lambda_hh = 0.01,
  start_date = as.Date("2024-09-21"),
  end_date = as.Date("2025-04-17"),
  latent_par = list(shape = 2, scale = 1),
  report_par = list(shape = 1, scale = 1.5),
  infect_par = list(shape = 3, scale = 2),
  start_par = c(-6, 0.02, -2, rep(0, 6)),
  lambda = 0.01,
  lambda0 = 0.2,
  lambda_alpha = 5,
  delta0_true = qlogis(0.002),
  alpha0_true = qlogis(0.2),
  true_values = NA
)
```

## Arguments

user_data	Data frame (preferred) with the required columns listed above; alternatively, a list of per-household data frames which will be row-bound. Must not be NULL.
-----------	--

**data\_summary** Logical. If TRUE, prints `results$summarized_data`. Default FALSE.  
**postprocessing** Logical. If TRUE (default), prints `postprocessing_estimates`.  
**n\_runs** Integer. Number of repeated estimation runs. Default 10.  
**day\_series\_covariates**  
 Logical. If TRUE, builds day-series list-columns for covariates in the person-day table. Default TRUE.  
**series\_cols** Character vector of covariate names (pre-normalization) for which to build day-series; if NULL, all detected covariates are used.  
**comm\_covariate\_cols**  
 Character vector of community-model covariates (no intercept).  
**hh\_covariate\_cols**  
 Character vector of household-model covariates shared across roles.  
**hh\_by\_role** Logical. If TRUE, use role-specific household covariates.  
**hh\_role\_covariate\_cols**  
 Named list with elements `infant`, `sibling`, `adult`, `elder` providing covariate names per role when `hh_by_role`=TRUE.  
**standardize\_covariates**  
 Logical. Z-score non-binary numeric columns in model matrices. Default TRUE.  
**lambda\_comm, lambda\_hh**  
 Numeric. L2 penalties for community and household covariate coefficients.  
**start\_date, end\_date**  
 Date. Study window used for summarization and imputation if `test_date` is provided as Date.  
**latent\_par, report\_par, infect\_par**  
 Lists with `Gamma(shape, scale)` parameters for imputation of latent delay, reporting delay, and infectious period.  
**start\_par** Numeric vector of initial parameters (auto-expanded if length mismatches model).  
**lambda** Base L2 penalty for slope-like parameters (age multipliers and household role offsets).  
**lambda0** Penalty anchoring  $\delta_0$  near  $\delta_{0\text{true}}$ .  
**lambda\_alpha** Penalty anchoring  $\alpha_0$  near  $\alpha_{0\text{true}}$ .  
**delta0\_true, alpha0\_true** Numeric anchors (logit scale) for community baseline and household baseline.  
**true\_values** Optional named numeric vector of reference values. Names should match columns in `theta_mat`. Unmatched names are ignored; missing references yield NA bias/relative bias.

## Details

This wrapper calls `main_parameter_estimation_pipeline` with `synthetic_data=FALSE` and passes `user_data` through `dataframe_to_household_list` internally. If `user_data` is a list of per-household data frames, it is row-bound before processing. Character covariates are internally factorized via model matrices when used in the likelihood.

## Value

(Invisibly) a list with elements:

- **results**: Output of `main_parameter_estimation_pipeline` (summaries, person-day, estimates).
- **postprocessing**: The table returned by `postprocessing_estimates` (or NULL if `postprocessing=FALSE`).

**Required columns in user\_data**

- HH — household identifier (integer)
- individual\_ID — individual identifier within household (integer)
- role — one of "infant", "sibling", "adult", "elder"
- test\_date — test day (integer or Date; Date will be converted to relative day)
- infection\_status — 0/1 infectious status at test\_date
- community\_risk — numeric community infection intensity at test\_date

Any additional columns are treated as candidate covariates and may be included in the likelihood via comm\_covariate\_cols, hh\_covariate\_cols, or hh\_role\_covariate\_cols.

**See Also**

[GenSyn](#), [main\\_parameter\\_estimation\\_pipeline](#), [data\\_summarization](#), [postprocessing\\_estimates](#)

**Examples**

```
## Not run:  
# Suppose df is your long-format dataset with required columns:  
fit <- TransmissionChainAnalysis(  
  user_data = df,  
  n_runs = 20,  
  comm_covariate_cols = c("cases"),  
  hh_covariate_cols = c("vaccination_status_mode")  
)  
fit$postprocessing  
  
## End(Not run)
```

# Index

build\_person\_day\_table, 2, 18, 20  
data\_summarization, 4, 26  
dataframe\_to\_household\_list, 3, 16, 17, 25  
generate\_synthetic\_data\_one, 6, 16, 21, 23  
generate\_synthetic\_data\_standardized, 5, 8, 16–18, 23  
GenSyn, 10, 26  
infectious\_time\_imputation, 2, 14, 18  
main\_parameter\_estimation\_pipeline, 5, 13, 15, 19, 25, 26  
postprocessing\_estimates, 11, 13, 18, 18, 25, 26  
running\_parameter\_estimation, 18, 19, 19  
simulate\_households, 8, 10, 16–18, 21  
summarize\_individuals, 5, 14, 16, 18, 23  
TransmissionChainAnalysis, 11, 13, 24