JOURNAL OF REPRODUCIBLE STATISTICS
FAIR Press Journals · jrepstat.fairpressjournals.com · ISSN 3043-2790
Vol. 1 (2026), Art. 11
10.68365/jrepstat.2026.11

SOFTWARE · FOUNDING ARTICLE

FlexRL: an R package for Flexible Record Linkage and Linked Data Quality Assessment

Kayané Robach*, Mark A. van de Wiel and Michel H. Hof

Amsterdam UMC, Amsterdam Public Health
*Corresponding author: k.c.robach@amsterdamumc.nl

ARTICLE INFO

Keywords
Record Linkage
Inference
Stochastic EM
False Discovery Proportion
Sampling Selection
Linked Data

History
Received 22 Oct 2026
Final version 9 Nov 2026
Accepted 28 Oct 2026
Version 1

Handling editor
Matthias Templ

ABSTRACT

FlexRL is an R package for probabilistic record linkage and inference on linked data. It provides a method to link observations from the same entities across two data sources that lack a unique identifier but share partially identifying variables. This method fits a hierarchical latent variable model of the data generation process, rather than classifying pairs of records from the comparison of their values, and accommodates registration errors and variables changing over time. Since linkage may affect subsequent analyses, the package also provides tools to assess the linked data quality as functions of the linkage decision rule. Two false discovery proportion estimators are implemented: a probabilistic approach contingent on the record linkage model and an augmented approach relying on an auxiliary estimation using synthetic data. Several measures of divergence of the linked sample from the source population are provided. These tools are agnostic to the underlying record linkage model and can be applied to any method. Together, they connect the output of the linkage to the validity of the subsequent analyses. The implementation combines standard R functions and S3 methods with Rcpp computations, resulting in fast execution times and a low memory footprint.

✓ Reproduced verified 2026-10-28 REPRODUCIBILITY RECORD

Result	Reported results reproduced. Reproduced on macOS arm64, except the BRL and multilink cells of Tables 2 and 4 (other packages' samplers) and fastLink in Table 6 (20 GB).	✓ *measured*
Entry point	`cd code/FlexRL-e4d511daaf35cb24d88480de49e4cf4e6ea87032/vignettes/article && quarto render Article-FlexRL.qmd`	✓ *executed*
Environment	R 4.5.2 on macOS arm64 (native), Tables 6 and 7 R 4.6.1 on linux/aarch64, packages as in Table 1	✓ *measured*
Determinism	Seeded.	*declared*
Archive	`flexrl-an-r-package-for-flexible-record-linkage-and-linked-data-quality-assessment-v1-repro.zip` · reviewed bundle SHA-256 `7ced60afc2cf7c0f08b53a1ec27c83753258ac1fad6e541e34fc169bade51cb3`	✓ *computed*
Data	Open.	*declared, route checked*

1 Introduction

Researchers and statistical agencies increasingly combine data sources that do not share a unique identifier (Cleaton et al., 2024; Kramer, 2026; Laidler et al., 2025). Record linkage connects records referring to the same entity across sources based on Partially Identifying Variables (PIVs, e.g. birth year, sex, postal code or product code, brand and category). Each PIV has limited discriminative power on its own and may

CITE AS Robach, K., van de Wiel, M. A., & Hof, M. H. (2026). FlexRL: an R package for Flexible Record Linkage and Linked Data Quality Assessment. Journal of Reproducible Statistics, 1, 11. https://doi.org/10.68365/jrepstat.2026.11

be missing, incorrectly recorded or change over time. Record linkage underpins the production of official statistics, enables the reuse of administrative and registry data for research (Cooney et al., 2023; Pontesilli et al., 2021; Williamson et al., 2020) and improves data quality through linkage and deduplication.

Linked data sets become the study sample for downstream analyses, where linkage errors may propagate (Doidge & Harron, 2019; Robach et al., 2026; Vo et al., 2024; Wang et al., 2022). Errors arise when links between entities that appear in both sources are missed or when records from distinct entities are falsely linked. These errors cannot be reduced simultaneously and may bias the conclusions drawn from linked data. A stricter decision rule reduces falsely linked pairs but misses more links. Practitioners therefore need not only record linkage methods, but also tools to quantify linkage errors and appraise their consequences for statistical analysis.

Two main probabilistic approaches to record linkage have developed through times. First, the traditional mixture model (Fellegi & Sunter, 1969; Fortini et al., 2001; Jaro, 1989; Larsen, 1999; Sadinle, 2017) treats linkage as a classification problem between pairs of records that are similar and pairs that are not. Second, the hierarchical entity resolution framework (Marchant et al., 2021; Marchant et al., 2023; R. C. Steorts, 2015; R. C. Steorts et al., 2016; Tancredi & Liseo, 2011) models the data generation process, considering observed values as potentially distorted versions of latent true values. The latter framework uses all the information contained in the records and is therefore more flexible but more computationally demanding. It has been implemented in `R` for entity resolution (clustering records from a single file into entities) rather than for linkage between two data sources.

We address this gap with the *FlexRL* `R` package, which makes two contributions. First, it implements the hierarchical record linkage model proposed by Robach, van der Pas, et al. (2025). The model is fitted with a Stochastic Expectation Maximisation (StEM) algorithm that accounts for registration errors and temporal changes in PIVs. Its architecture and efficient `Rcpp` backend support scalability to large data sets and a wide range of applications: health monitoring studies, survey waves, census and administrative records or data from different retailers. Second, it provides tools to assess the quality of the linked data, which quantify the linkage error and sample selection. *FlexRL* implements two estimators of the False Discovery Proportion (FDP): one derived from linkage scores (Belin & Rubin, 1995; Enamorado et al., 2019; Resnick & Asher, 2019) and one based on synthetic data (Robach, Hof, et al., 2025). *FlexRL* measures the divergence of the linked sample from the original sources using standardised mean differences, overlap of supports, maximum mean discrepancy and agreement rates. These tools only need the linked pairs and, when available, their linkage scores. They therefore apply to the output of any linkage method and *FlexRL* provides wrappers for several existing `R` packages.

The *FlexRL* package is aimed at practitioners, statistical agencies and researchers who need to link data sources without a unique identifier or who want to measure the error and the sample selection induced by record linkage. It builds on three lines of work: the biases that linkage errors cause in inference (Doidge & Harron, 2019; Lahiri & Larsen, 2005; Robach et al., 2026; Tancredi & Liseo, 2015; Wang et al., 2022); the need for linked data quality control in practice (Harron et al., 2020; Harron et al., 2017; Laidler et al., 2025; Lam et al., 2024); advances in record linkage methodology (Marchant et al., 2023; Robach, van der Pas, et al., 2025; Sadinle, 2017). By combining record linkage and diagnostics in a practical tool, it helps evaluating whether inference can be conducted on linked data.

Throughout the paper, we motivate and illustrate the main components of *FlexRL* on the 2012 and 2014 waves of the Survey on Household Income and Wealth (SHIW), conducted and made available by the Bank of Italy for research purposes. As a secondary analysis of public data, no additional ethics approval or participant consent was required. Section 2 introduces the notation, the modelling paradigms, the existing software and demonstrates the potential of *FlexRL*. Section 3 presents the record linkage model implemented in *FlexRL* and its estimation on a subset of the SHIW. On the resulting linked data, Section 4 describes the diagnostic tools *FlexRL* provides, and illustrates the impact of linkage errors on a socio-economic regression analysis. Section 5 demonstrates the scalability of *FlexRL* approach to record linkage on the SHIW. Section 6 concludes.

2 The record linkage problem

2.1 *Setup and notation*

We consider two overlapping samples $\mathcal{A}$ and $\mathcal{B}$ from the same population with $N_\mathcal{A}$ and $N_\mathcal{B} \geq N_\mathcal{A}$ records. Both sources record the same K categorical PIVs, indexed by $k = 1, \ldots, K$ with N_k possible values each. Index i refers to a record in $\mathcal{A}$ and index j to a record in $\mathcal{B}$. For PIV k, we denote by $G^\mathcal{A}_{i,k}$ the registered (observed) value and by $H^\mathcal{A}_{i,k}$ the latent true value in record $i \in \mathcal{A}$. We define $G^\mathcal{B}_{j,k}$ and $H^\mathcal{B}_{j,k}$ analogously. Registered values differ from true values because of registration errors: missing values or mistakes (e.g. typographical errors). We write $\mathbf{G}$ and $\mathbf{H}$ for the collections of all registered and all true values. True values may also differ between the two data collections. This is the case for *dynamic* PIVs like postal code, whereas *stable* PIVs like birth year cannot change. For an entity recorded in both sources, the true values of stable PIVs are therefore identical in $\mathcal{A}$ and $\mathcal{B}$ and any disagreement between their registered values is due to registration errors. We elaborate on dynamic PIVs later.

The object of interest is the linkage matrix $\mathbf{\Delta} \in \{0,1\}^{N_\mathcal{A} \times N_\mathcal{B}}$, with $\Delta_{i,j} = 1$ if records $i \in \mathcal{A}$ and $j \in \mathcal{B}$ refer to the same entity. We assume that there are no duplicates within a data source. Thus, each record forms a link with at most one record in the other data source: $\sum_j \Delta_{i,j} \leq 1$ for all i and $\sum_i \Delta_{i,j} \leq 1$ for all j. This is the one-to-one assignment constraint, also referred to as bipartite matching (Sadinle, 2017). We denote by $\gamma \in (0,1)$ the probability that a record from $\mathcal{A}$ forms a link with a record from $\mathcal{B}$, that is, the overlap between the sources as a fraction of the smaller data set.

Since record linkage is imperfect, the linked data contain two types of error: false positives (pairs declared linked while referring to distinct entities) and false negatives (true links that are missed). The non-links in the Cartesian product $\mathcal{A} \times \mathcal{B}$ are far more numerous than the links and are rarely of interest. The useful summaries are therefore the False Discovery Proportion (FDP) among the linked data (complement of precision or positive predictive value) and the sensitivity (also called recall, complement of miss rate). The two errors affect inference differently: falsely linked pairs contaminate the inference and missed links induce sample selection (Doidge & Harron, 2019; Robach et al., 2026).

2.2 *Modelling paradigms*

Linkage methods can be grouped into four paradigms (Binette & Steorts, 2022; Dusetzina et al., 2014). Deterministic (rule based) linkage declares a link when records agree on a set of variables, exactly or up to a tolerance. Supervised methods train a classifier on pairs whose linkage status is known (Mudgal et al., 2018); such labelled pairs are seldom available. The two remaining paradigms are unsupervised and probabilistic, and differ in what they model:

Paradigm 1: classification of comparison vectors
The traditional 'Fellegi-Sunter' approach (Dunn, 1946; Fellegi & Sunter, 1969; Newcombe et al., 1959; Tepping, 1968) reduces the data to comparison vectors. For each pair $(i,j) \in \mathcal{A} \times \mathcal{B}$, a vector $\boldsymbol{c}_{i,j} = (c_{i,j,1}, \ldots, c_{i,j,K})$ records whether elements $G^\mathcal{A}_{i,k}$ and $G^\mathcal{B}_{j,k}$ for $k = 1, \ldots, K$ agree or how similar they are. The pairs of records are split into the set of matched M and the set of unmatched U. The comparison vectors follow a two components mixture:

$$\begin{aligned} P(\boldsymbol{c}_{i,j}) &= \gamma m(\boldsymbol{c}_{i,j}) + (1-\gamma) u(\boldsymbol{c}_{i,j}), \\ m(\boldsymbol{c}) &= P(\boldsymbol{c} \mid M), \\ u(\boldsymbol{c}) &= P(\boldsymbol{c} \mid U), \end{aligned} \tag{1}$$

where γ in this paradigm is the proportion of links among all pairs. Under the assumption that the K comparisons are independent given the linkage status, m and u factorise over $k \in \{1, \ldots, K\}$. The parameters are then estimated with an EM algorithm (Jaro, 1989; Larsen, 1999). Pairs are classified by thresholding the ratios $m(\boldsymbol{c}_{i,j})/u(\boldsymbol{c}_{i,j})$ or the posterior linkage probabilities. Bayesian versions place a prior on the linkage matrix $\mathbf{\Delta}$ to enforce the one-to-one assignment constraint (Fortini et al., 2001; Sadinle, 2017) or a prior on a partition of the records when there are more than two sources or duplicates (Aleshin-Guendel & Sadinle, 2023). The computations are simple and fast, but memory intensive because they operate on $N_\mathcal{A} \times N_\mathcal{B}$ comparison vectors. Scaling to large data sets may therefore require blocking, which

compares only pairs that agree on some variables (R. C. Steorts et al., 2014). This paradigm treats pairs as independent although they share records, and the reduction to comparison vectors discards part of the information contained in the records.

Paradigm 2: hierarchical modelling
The second family models how the data were generated (Marchant et al., 2021; Marchant et al., 2023; Robach, van der Pas, et al., 2025; R. C. Steorts, 2015; R. C. Steorts et al., 2016; Tancredi & Liseo, 2011). Registered values $\mathbf{G}$ are seen as distorted versions of latent true values $\mathbf{H}$, which define the linkage structure $\mathbf{\Delta}$. The joint model reads

$$P(\mathbf{G}, \mathbf{H}, \mathbf{\Delta}) = P(\mathbf{G} \mid \mathbf{H})P(\mathbf{H} \mid \mathbf{\Delta})P(\mathbf{\Delta}), \tag{2}$$

with a model for registration errors $P(\mathbf{G} \mid \mathbf{H})$, a model for underlying true values $P(\mathbf{H} \mid \mathbf{\Delta})$ and a model for the linkage structure $P(\mathbf{\Delta})$. Since records are never reduced to pairwise comparisons, the model can separate missing values, mistakes and changes over time, and it does not need blocking to fit in memory. However, it is more computationally demanding.

2.3 *Existing software*

In `Python`, *recordlinkage* (Bruin, 2019), *Splink* (Linacre et al., 2022) and *NameMatch* (Jelveh et al., 2022) implement Paradigm 1. We list the main `R` packages for the task considered here in Table 1. Paradigm 2 is only available in `R` through methods for entity resolution (Marchant et al., 2021; Marchant et al., 2023; R. Steorts et al., 2014). These methods cluster records from a single file, possibly containing duplicates, into entities.

Table 1 Record linkage packages on `CRAN` compared with *FlexRL*. The *FlexRL* package diagnostic tools apply to all these packages.

Package	Paradigm	Estimation	Linkage structure	Record deduplication	PIV dynamics	Quality tools
fedmatch 2.1.0 (Cohen et al., 2021; Friedrichs et al., 2023)	comparison vectors	push relabel maximum flow	many-to-many	no	no	no
diyar 0.5.1 (Nsonwu, 2023)	comparison vectors	multistage weights comparisons, fixed score level parameters	clusters	yes	no	no
reclin2 0.6.0 (van der Laan, 2025)	comparison vectors, supervised	EM, machine learning	one-to-one post hoc	no	no	no
fastLink 0.6.1 (Enamorado et al., 2019; Ted Enamorado, 2023)	comparison vectors	EM	one-to-one post hoc	no	no	no
BRL 0.1.0 (Sadinle, 2017; Sadinle, 2020)	comparison vectors	Bayesian, bipartite prior, Gibbs sampler	one-to-one in model	no	no	no
multilink 0.1.1 (Aleshin-Guendel, 2023; Aleshin-Guendel & Sadinle, 2023)	comparison vectors	Bayesian, partition prior, Gibbs sampler	clusters, one-to-one in model	yes	no	no
FlexRL 2.0.0 (Robach & Hof, 2026)	hierarchical	stochastic EM	one-to-one in model	no	yes	yes

To our knowledge, no `R` package implements Paradigm 2 for the linkage of two sources under the one-to-one assignment constraint or models the dynamics of PIVs over time. While the *postlink* package (Bukke et al., 2026) develops statistical tools for post-linkage data analysis, no record linkage package provides tools to assess the quality of the linked data and their suitability for inference.

The implementation of *FlexRL* relies on `Rcpp` (potentially parallelised with `OpenMP`) for the backend samplers, *Matrix* for the sparse linkage matrix, and other standard `R` packages. The diagnostic tools apply to any record linkage output in the required format, with built-in wrappers for *BRL*, *diyar*, *fastLink*, *fedmatch*, *multilink* and *reclin2*. Figure 1 shows the workflow, which follows the structure of Section 3 and Section 4: the data are prepared, linked with `StEM()` and the linked data quality is assessed with `RL_diagnostics()`. We elaborate on each component along the article.

2.4 *Demonstration*

Simulation study

As a preliminary demonstration of the functionalities of *FlexRL*, we simulate data with the package `simulate_data()` function. This function synthesises two files with a given overlap, given rates of mistakes and missing values per PIV and file, and given hazard coefficients for structured PIVs, useful for exploring and comparing record linkage methods. Here, we generate two data sets of 250 and 300

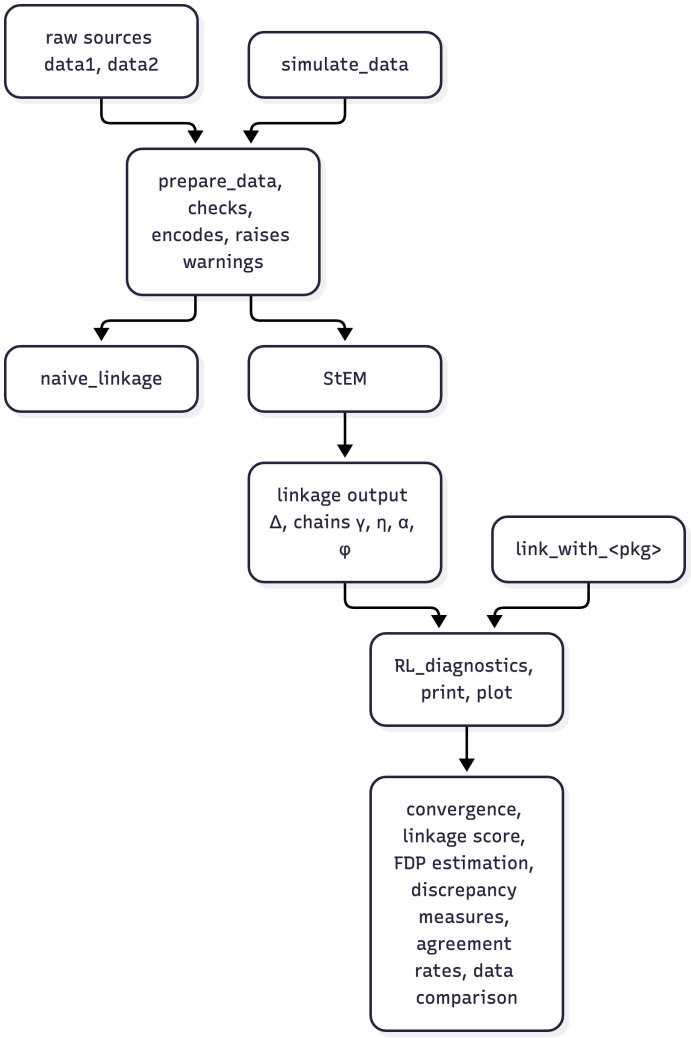


Figure 1 Workflow of *FlexRL*.

records, with 200 entities common to both. Entities are represented by four independent discrete non-uniform PIVs. We distort the data in each file uniformly with 2% of mistakes and 0.5% of missing values; one of the PIVs undergoes temporal changes.

Note that this data generation process, on which all packages cited in Table 1 are evaluated, is in line with the hierarchical model *FlexRL* fits and represents a simple setting. The fourth PIV follows a dynamic process whose hazard is piecewise constant over time and modulated by a Gaussian covariate. We fit these changes with an exponential survival model in *FlexRL*, illustrating the benefits of modelling PIV dynamics, even under model misspecification, rather than treating them as registration errors. Later in the article, we evaluate all packages on real data for a fairer comparison.

	Naive	BRL	multilink	fastLink	reclin2	diyar	fedmatch	FlexRL (dynamics)	FlexRL (no dynamics)
TP	118.40 (5.13)	80.40 (5.04)	80.60 (5.95)	101.70 (6.48)	114.10 (6.38)	34.90 (20.24)	122.10 (6.33)	114.00 (7.92)	87.50 (8.30)
FP	83.10 (11.02)	12.30 (3.23)	14.20 (4.13)	43.10 (5.59)	68.30 (8.82)	3.30 (2.91)	125.80 (6.32)	25.90 (5.15)	19.00 (4.67)
FN	81.60 (5.13)	119.60 (5.04)	119.40 (5.95)	98.30 (6.48)	85.90 (6.38)	165.10 (20.24)	77.90 (6.33)	86.00 (7.92)	112.50 (8.30)
sensitivity	0.59 (0.03)	0.40 (0.03)	0.40 (0.03)	0.51 (0.03)	0.57 (0.03)	0.17 (0.10)	0.61 (0.03)	0.57 (0.04)	0.44 (0.04)
FDP	0.41 (0.03)	0.13 (0.03)	0.15 (0.04)	0.30 (0.03)	0.37 (0.03)	0.08 (0.05)	0.51 (0.03)	0.18 (0.03)	0.18 (0.04)
time (secs)	0.27 (0.04)	0.97 (0.05)	18.57 (1.48)	1.29 (0.24)	0.29 (0.01)	1.17 (0.06)	1.44 (0.01)	4.59 (0.02)	3.72 (0.01)
memory (Mb)	1.04 (0.05)	78.60 (10.08)	105.68 (10.18)	24.85 (4.42)	41.52 (0.18)	104.31 (10.84)	42.14 (2.59)	102.11 (9.03)	70.18 (6.42)

Linkage performance of competing record linkage methods on simulated data for the default linkage decision rule; see Section 3.3. We report average performance over 10 replications, with standard deviations in parentheses.

In (**tbl-demonstration?**), we compare the methods presented in Table 1 to the *FlexRL* record linkage algorithm with and without modelling PIV dynamics in terms of sensitivity and FDP. When appropriate, modelling PIV dynamics improves record linkage by increasing sensitivity (more true links are recovered) while keeping the FDP under control. The original methodological article provides a more comprehensive study of the behaviour of FlexRL with and without modelling PIV dynamics (Robach, van der Pas, et al., 2025). (**tbl-demonstration?**) also shows the differences in computational time and in memory usage between the methods. More details on this demonstration showcasing the utility of accounting for PIV dynamics are given in the package vignette 'Linkage and Diagnostics with FlexRL' available through `vignette('FlexRL-RL-vignette', 'FlexRL')` or the online documentation.

The results presented in this article are obtained on a MacBook Pro (Apple M1 Max ARM chip architecture, 32GB memory, 8 cores) with `R` version 4.6.1. The random seed is set to 1 in each code cell that produces output (some compared packages call `set.seed()` internally). The computations are robust to different operating systems (MacOS, Linux, Windows). We measure the memory usage using the `R` garbage collector. External allocations are therefore likely underestimated. All methods compared in this article rely on `Rcpp` or `C`, except for *diyar* which is fully implemented in `R`. Time and memory usage may vary depending on the operating system, hardware, and system configuration.

3 The hierarchical architecture

3.1 *Data preparation and PIVs configuration*

The record linkage model requires each PIV k to be categorical, encoded by integers $1, \ldots, N_k$ with missing values coded as 0. The PIVs must cover a common support across the two sources. The smaller source should be labelled as $\mathcal{A}$.

As a first step in the workflow represented in Figure 1, the `prepare_data()` function which we will describe validates the PIVs configuration, labels the data sources and encodes the PIVs. When a unique identifier is available, like in simulation studies or in benchmark record linkage data such as the SHIW, it also returns the set of true links. This set may be used to evaluate linkage performance.

The function `naive_linkage()` links records that agree exactly on all PIVs (a missing value matching any value when `na_match = TRUE`), without enforcing the one-to-one constraint. When the true links are known, the performance of this deterministic linkage indicates the discriminative power of the PIVs and the difficulty of the task. The FDP of the 'Naive' linkage reflects how many records share the same combination of PIV values.

The package distinguishes different kinds of PIVs, which determine how the components of the complete data likelihood of Equation (3) are used when fitting the model in Section 3.2. Candidate pairs of records are formed from records whose latent true values agree on every PIV for which temporal changes are not modelled; whether a candidate pair is linked is then decided from its contribution to the complete likelihood. We elaborate on candidate pairs and linked pairs in Section 3.2.

- *Stable*: the latent true values do not change (e.g. birth year). Such PIV is used to form candidate pairs and its probability of mistake may be bounded (for example by 0.10) for algorithm stability (Robach, van der Pas, et al., 2025).
- *Dynamic*: the true values may change (e.g. postal code between two survey waves):
  - *Flexible*: changes cannot be modelled, typically because registration dates are not available. The PIV is still used to form candidate pairs, but its probability of mistake should not be bounded, so that the registration model absorbs both mistakes and changes.
  - *Structured*: changes are modelled by the survival function of Equation (4). Such PIV is not used to form candidate pairs. Registration dates are needed in both sources and covariates for the hazard may be supplied. Mistakes cannot be distinguished from temporal changes, so the probability of mistake should be fixed (for example at 0.02), unless enough linked pairs have time gaps near zero, where changes are not plausible and mistakes can be estimated without confusion. Estimating the changes also requires variation in the time gaps among linked pairs.

To illustrate the usage of the *FlexRL* package throughout the paper, we consider the Emilia-Romagna regional subset of the SHIW data. The record linkage task can be validated in this example where the linkage structure can be reconstructed. After preparation, the 2012 and 2014 waves cover 1479 and 1524 individuals, with 898 individuals observed in both waves. In Section 4, we analyse a regression of the expected age of retirement in 2014 on the expected age of retirement in 2012.

To link the two survey waves, we use the following variables: year and region of birth, sex, marital status, working status and branch of activity. The strong association between working status and branch of activity violates the PIV independence assumption of the method and may degrade the record linkage performance (Robach, van der Pas, et al., 2025). We thus combine these variables into a single variable 'occupation'. Although marital status and occupation are 'dynamic' PIVs, we do not have enough information to model their temporal changes and set them as 'flexible' in the configuration:

```
PIVs_config <- list(BirthYear     = list(dynamics        = "stable",
                                         bound_mistakes  = c(0.10, 0.10),
                                         fix_mistakes    = c(NA, NA)),
                    Sex           = list(dynamics        = "stable",
                                         bound_mistakes  = c(0.10, 0.10),
                                         fix_mistakes    = c(NA, NA)),
                    MaritalStatus = list(dynamics        = "flexible",
                                         bound_mistakes  = c(NA, NA),
                                         fix_mistakes    = c(NA, NA)),
                    BirthRegion   = list(dynamics        = "stable",
                                         bound_mistakes  = c(0.10, 0.10),
                                         fix_mistakes    = c(NA, NA)),
                    Occupation    = list(dynamics        = "flexible",
                                         bound_mistakes  = c(NA, NA),
                                         fix_mistakes    = c(NA, NA)))

PIVs <- names(PIVs_config)
```

We prepare the two data sources for the `StEM()` record linkage algorithm with the `prepare_data()` function, which takes as inputs the data sources and their labels `df2012`, `df2014`, `'2012'`, `'2014'`; the PIVs configuration `PIVs_config`; a logical `same_mistakes` describing whether the parameters fitting the probability of mistakes among PIVs should be the same between data sources (`TRUE` when the data are sampled from a same source population); a potential record identifier variable `uniq_id` when available; a logical `restrict_support_intersection` to harmonise the support of the PIVs in the observed data. In the application, records with birth years outside of the range shared by the sources are excluded.

```
prep_data <- prepare_data(df2012, df2014, "2012", "2014", PIVs_config,
                          same_mistakes = TRUE, uniq_id = "ID")
```

```
'2014' is the larger source, saved as B; '2012' saved as A.
```

This function outputs a list suited for `StEM()`, where data sources have been encoded and labelled as `encodedA` and `encodedB`. It additionally includes the number of unique values per PIV `n_values`, the validated `PIVs_config`, the logical `same_mistakes` and a data set of indices of true links `true_pairs` when `uniq_id` was provided. It alerts the user to potential issues, including inconsistent configurations, strongly associated PIVs and values outside the common support. For instance, the following warnings appear on the Emilia-Romagna subsets of the 2012 and 2014 waves of the SHIW data:

```
PIV 'BirthYear': values out of the common support in data1 [1920, 1912] in data2 [2013,
2014].
```

```
PIVs 'Occupation' and 'Sex' are associated (Cramer's V = 0.32 in data1, 0.35 in data2);
FlexRL assumes conditional independence between PIVs, consider merging or dropping one.
```

Among the true links, known in this application, there are some registration errors or temporal changes in the PIVs. As a consequence the true links do not necessarily carry the exact same information in their records. With the `RL_agreement` function, the user can see the proportion of agreements among pairs of records (for example among linked pairs of records or, in the case presented below, among true links).

```
    BirthYear           Sex MaritalStatus   BirthRegion    Occupation
         0.99          1.00          0.98          0.99          0.82
```

In (**tbl-illustration?**), we compare the record linkage methods mentioned in Table 1 on this regional subset of the SHIW data. Memory usage is of the same order of magnitude on these relatively small data sets for all methods. The figures and tables based on the SHIW data reflect a single random realisation. They depend on the seed we use in the script and on the seed some other packages use internally.

	Naive	BRL	multilink	fastLink	reclin2	diyar	fedmatch	FlexRL
TP	710.00	158.00	160.00	467.00	721.00	20.00	510.00	297.00
FP	3494.00	24.00	72.00	850.00	339232.00	0.00	969.00	103.00
FN	188.00	740.00	738.00	431.00	177.00	878.00	388.00	601.00
sensitivity	0.79	0.18	0.18	0.52	0.80	0.02	0.57	0.33
FDP	0.83	0.13	0.31	0.65	1.00	0.00	0.66	0.26
time (secs)	0.37	17.78	1118.04	33.69	0.84	11.04	8.88	22.57
memory (Mb)	8.30	523.20	1901.90	2436.20	1180.20	699.50	213.20	263.20

Comparison of record linkage methods on the SHIW in Emilia-Romagna (for the default linkage decision rule, see Section 3.3).

3.2 *StEM estimation*

The *FlexRL* hierarchical model for record linkage adapts Equation (2) to two sources under the one-to-one assignment constraint. For non-linked records, the true values are independent draws from the population distribution. For a linked pair (i, j), $H^{\mathcal{B}}_{j,k}$ is a copy of $H^{\mathcal{A}}_{i,k}$ except when k is a dynamic PIV, whose values may have changed between the two registrations. The registered values are versions of the true values distorted by registration errors.

With parameters $\theta = (\gamma, \eta, \alpha, \phi)$, the complete data likelihood factorises as

$$\begin{aligned}\mathcal{L}_\theta\left(\mathbf{G}^{\mathcal{A}},\mathbf{G}^{\mathcal{B}},\mathbf{H}^{\mathcal{A}},\mathbf{H}^{\mathcal{B}},\boldsymbol{\Delta}\right) = &\underbrace{\mathcal{L}_{\boldsymbol{\phi}^{\mathcal{A}}}\left(\mathbf{G}^{\mathcal{A}}\mid\mathbf{H}^{\mathcal{A}}\right)\mathcal{L}_{\boldsymbol{\phi}^{\mathcal{B}}}\left(\mathbf{G}^{\mathcal{B}}\mid\mathbf{H}^{\mathcal{B}}\right)}_{\text{registration errors}}\\ &\underbrace{\mathcal{L}_{\boldsymbol{\alpha},\boldsymbol{\eta}}\left(\mathbf{H}^{\mathcal{B}}\mid\mathbf{H}^{\mathcal{A}},\boldsymbol{\Delta}\right)}_{\text{changes over time}}\underbrace{\mathcal{L}_{\boldsymbol{\eta}}\left(\mathbf{H}^{\mathcal{A}}\right)}_{\text{true values}}\\ &\underbrace{\mathcal{L}_{\gamma}\left(\boldsymbol{\Delta}\right)}_{\text{linkage}}.\end{aligned} \tag{3}$$

The parameter $\boldsymbol{\phi}$ governs the possible distortion mechanisms (mistakes, missing values) at the source of the differences between registered and latent true values. The parameter $\boldsymbol{\alpha}$ shapes the model for temporal changes between latent true values in data sources depending on the chosen 'structure' (see the different survival models for Equation (4)). The parameter $\boldsymbol{\eta}$ characterises the distributions of the PIVs. The parameter γ represents the probability that an observation in $\mathcal{A}$ forms a link with an observation in $\mathcal{B}$. Each component of Equation (3) and each parameter is developed in the original methodological paper (Robach, van der Pas, et al., 2025).

Notably, the probability that true values have not changed between data collections is modelled with a survival function. By default, the package uses an exponential model with proportional hazards. For a dynamic *structured* PIV k whose changes are modelled and a linked pair (i, j) with time gap $t_{i,j} = |t_j^{\mathcal{B}} - t_i^{\mathcal{A}}|$:

$$P\left(H_{j,k}^{\mathcal{B}} = H_{i,k}^{\mathcal{A}} \mid t_{i,j}, \boldsymbol{x}_{i,j}, \Delta_{i,j} = 1\right) = S_{\boldsymbol{\alpha}_k}(\boldsymbol{x}_{i,j}, t_{i,j}) = \exp\left\{-\exp(\boldsymbol{x}_{i,j}^T\boldsymbol{\alpha}_k)t_{i,j}\right\}, \tag{4}$$

where $\boldsymbol{x}_{i,j}$ contains an intercept and, optionally, covariates from either data source that explain the hazard of change (for example age or income for a relocation) and $\boldsymbol{\alpha}_k$ contains the log hazard coefficients. The user can choose another survival model, such as a Weibull, a Gompertz, a piecewise constant hazard or any custom model.

The likelihood of the observed data is obtained by summing Equation (3) over all values of the latent $\mathbf{H}$ and $\boldsymbol{\Delta}$. *FlexRL* maximises it with the `StEM()` function, a Stochastic Expectation Maximisation (StEM) algorithm (Celeux & Diebolt, 1986). In a classical EM, the E-step computes the conditional expectation of the complete likelihood given the observed data. In the stochastic version, this expectation is approximated by Monte Carlo: a Gibbs sampler simulates the latent variables given the current parameters. The averages of the summary statistics collected after burn-in estimate the conditional expectation from which the M-step derives the parameter updates. Algorithm 1 summarises the procedure, detailed hereafter.

Algorithm 1

```
Input: encoded sources G_A, G_B; PIV configuration;
       N_StEM iterations (burn-in N0_StEM); N_Gibbs iterations (burn-in N0_Gibbs)
Initialise θ = (γ, η, α, φ)
for StEM iter = 1, ..., N_StEM do
    Δ ← 0
    # stochastic E-step
    for Gibbs iter = 1, ..., N_Gibbs do
        draw H_A, H_B | G_A, G_B, Δ, θ, x, t
        form candidate pairs (i,j) such that H_A[i,k] = H_B[j,k] for all stable or
flexible k
        for each non-linked candidate pair: propose to link
        for each linked candidate pair: propose to break the link
        accept each proposal with the likelihood of the model
        if Gibbs iter > N0_Gibbs then
            accumulate summary statistics:
              number of linked pairs,
              latent true values distribution,
              temporal changes in linked (H_A = H_B, x, t),
              registration errors in linked and non-linked (G = H)
        end if
    end for
```

```
    average the accumulated summary statistics # estimate E[ H, Δ | G; θ]
    # M-step
    update and store θ = (γ, η, α, φ)
end for
θ_hat ← average of θ over StEM iter > N0_StEM
run further Gibbs iterations at θ_hat to build D_hat:
D_hat[i,j] ← proportion of iterations in which (i,j) is linked
```

Stochastic EM for record linkage.

Initialisation
The probability γ is drawn uniformly in $(0.2, 0.8)$, the distribution of each PIV k, $\boldsymbol{\eta}_k$, is uniform over the N_k values, the probabilities of mistakes are drawn in $(0.03, 0.20)$ and the probabilities of missing values are set to the observed proportions. For structured PIVs, the log baseline hazard is drawn between 0.01 and 1 per time unit and the covariate coefficients near 0. The user can supply other starting values.

Stochastic E-step
Given $\boldsymbol{\theta}$, the Gibbs sampler alternates two blocks. First, the latent true values are drawn. For a non-linked record $H^{\mathcal{A}}_{i,k}$ or $H^{\mathcal{B}}_{j,k}$ is drawn from $\boldsymbol{\eta}_k$ weighted by the registration model. For a linked pair $(H^{\mathcal{A}}_{i,k}, H^{\mathcal{B}}_{j,k})$ is drawn jointly among the $N_k \times N_k$ possible values, from $\boldsymbol{\eta}_k$ weighted by the registration model and the survival model.

Second, the linkage is updated. Candidate pairs are formed from matching sampled latent true values: records $i \in \mathcal{A}$ and $j \in \mathcal{B}$ are candidates when $H^{\mathcal{A}}_{i,k} = H^{\mathcal{B}}_{j,k}$ for every non-structured PIV k (stable or flexible). For each candidate pair (i, j), we evaluate the log likelihood contribution to Equation (3) when the records are linked, $\log \mathcal{L}^{\text{linked}}_{i,j}$, and when they are not, $\log \mathcal{L}^{\mathcal{A}}_i + \log \mathcal{L}^{\mathcal{B}}_j$. A sequential sweep over the candidate pairs then proposes to create or break links. A proposal is accepted by comparing the proposed 'new' state of Equation (3) to the current 'old' state with probability $\mathcal{L}_{\text{new}}/(\mathcal{L}_{\text{old}} + \mathcal{L}_{\text{new}})$; the two states differ by $\log \mathcal{L}^{\text{linked}}_{i,j} - \log \mathcal{L}^{\mathcal{A}}_i - \log \mathcal{L}^{\mathcal{B}}_j$ and by the contributions from changes in the linkage matrix.

Over the Gibbs iterations, the sampler stochastically explores the latent space of possible linkage matrices and PIVs true values. At each iteration after burn-in we store the number of pairs that were linked, the distribution of latent true values, the temporal changes (disagreements between the linked records latent true values) and the registration errors (disagreements between $\mathbf{G}$ and $\mathbf{H}$)—summary statistics required for the M-step.

Restricting the sweep to candidate pairs keeps the algorithm light in memory: likelihood contributions are computed only for these pairs and for the remaining non-linked records in each data source, which together represent the estimated unique entities in the population.

M-step
The summary statistics averaged over the Gibbs draws after burn-in give the parameter updates. Thanks to the factorisation of Equation (3), γ, $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$ have separate closed form updates (Robach, van der Pas, et al., 2025), and $\boldsymbol{\alpha}$ is obtained by numerical optimisation. Optional bounds on the probabilities of a mistake are applied at this stage. A warning is displayed when a bound is reached in many iterations, suggesting that the bound is too tight or that the PIV is dynamic.

3.3 *Linkage scores and decision rule*

Based on the StEM chains of estimates, we build an estimate of the set of parameters, $\hat{\boldsymbol{\theta}}$, by averaging the post burn-in elements in the sequence. These are used to sample further linkage matrices that are averaged to return a probabilistic estimate of $\boldsymbol{\Delta}$ as a final output, namely $\hat{\boldsymbol{D}}$, whose entries $\hat{D}_{i,j}$ are the posterior probabilities for the pair (i, j) to be linked, which we call the linkage scores. A decision rule on each entry $\xi \in [0.5, 1)$ then defines the final set of linked data $\mathcal{D} = \{(i, j); \hat{D}_{i,j} > \xi\}$. Since the scores of a record sum to at most one, a threshold of at least 0.5 ensures the one-to-one assignment (Sadinle, 2017; Tancredi & Liseo, 2011). The choice of ξ is the subject of the upcoming Section 4.

Note that probabilistic record linkage methods estimate $\boldsymbol{\Delta}$ in comparable ways, by classifying pairs according to their linkage scores. Although these scores may be on different scales (e.g. likelihood ratios,

log odds, posterior probabilities), they can be mapped to a probability scale. Hereafter, we take linkage scores to be posterior linkage probabilities in $[0, 1]$ and treat all the methods listed in Table 1 in the same way. Accordingly, the default decision rule is $\xi = 0.5$. The 'FDP versus linkage decision rule' curves (see Figure 3) provide a more comprehensive assessment of record linkage performance.

3.4 *Record linkage with* `StEM()`

`StEM()` runs Algorithm 1. Its main arguments are:

- `data`: a list, typically the output of `prepare_data()`, with the encoded sources `encodedA` and `encodedB`, the number of distinct values of each PIV `n_values` (excluding missing values), a logical indicating whether both sources share the registration error parameters `same_mistakes`, and the PIVs configuration `PIVs_config`.
- `StEM_iter`, `StEM_burnin`, `gibbs_iter`, `gibbs_burnin`: the numbers of iterations of the outer StEM algorithm and of the inner Gibbs sampler, and their burn-in periods. `gibbs_burnin = 0` sets the burn-in automatically: the sampler runs until the number of links changes by fewer than 10 for 10 consecutive iterations, with a maximum of 500 iterations.
- `threads`: the number of `OpenMP` threads used by the `Rcpp` samplers, default 4. The argument is ignored on systems where the package is built without `OpenMP`.
- `model_dynamics`: the survival model for structured PIVs, by default the exponential model of Equation (4). It is only used if a PIV is declared structured in `PIVs_config`.
- `n_post_sample`: the number of posterior draws used to compute the linkage scores.
- Starting values for γ and for the agreement probabilities $1 - \boldsymbol{\phi}$ can be supplied.
- For long runs, `new_directory` and `save_info_iter = TRUE` save the environment after each StEM iteration, and `music_on = TRUE` plays a short tune when the algorithm finishes running.

The function returns `Delta`, a data frame with columns `i` (indices in $\mathcal{A}$), `j` (indices in $\mathcal{B}$) and `x` (scores $\hat{D}_{i,j}$) for every pair with a positive score, and the four chains: `gamma` (a vector of length `StEM_iter`), `eta` (one `StEM_iter`$\times N_k$ matrix per PIV), `alpha` (one matrix per structured PIV, with `StEM_iter` rows and one column per parameter of the survival model) and `phi` (one `StEM_iter`$\times 4$ matrix per PIV: agreement in $\mathcal{A}$, in $\mathcal{B}$, missing in $\mathcal{A}$, in $\mathcal{B}$).

We run the *FlexRL* `StEM` algorithm for record linkage as follows:

```
fit <- StEM(data = prep_data)
```

3.5 *Computational aspects*

The two latent variable samplers are implemented with `Rcpp`. To draw the true values, `sampleH()` goes through the N_k^2 joint values once per PIV k for linked pairs and through the N_k values once per PIV k for non-linked records. `sampleD()` performs the sequential sweep over candidate pairs, recording the current linkage in two vectors that give, for each record of $\mathcal{A}$ and of $\mathcal{B}$, the index of the record it is linked to (or a negative value when not linked). Reading and updating the linkage takes a fixed number of operations, so the sweep is linear in the number of candidate pairs and the memory is linear in the number of records $N_{\mathcal{A}} + N_{\mathcal{B}}$ rather than in $N_{\mathcal{A}} \times N_{\mathcal{B}}$.

Candidate pairs are found by hashing the pattern of sampled true values on the non-structured PIVs, an operation linear in $N_{\mathcal{A}} + N_{\mathcal{B}}$. Their contributions to the complete data likelihood are computed in a single pass over the records and the candidate pairs, implemented in `Rcpp`. Built-in survival models are evaluated in `Rcpp`; custom survival functions keep their `R` implementation.

The single pass over the candidate pairs likelihood contributions can be parallelised over `OpenMP` threads, each processing a disjoint set of candidates. Without `OpenMP` the code runs sequentially. All random draws occur upstream, so the threads never call the `R` random number generator. Results are therefore identical regardless of the number of threads, and `set.seed()` ensures reproducibility.

Running time grows with the number of records; the number of candidate pairs, larger when PIVs have few levels; the number of PIVs and their numbers of levels; the number of structured PIVs, whose

survival model is estimated at each iteration; the number of iterations; and decreases with the number of threads.

4 Inference on linked data

4.1 *Impact of linkage errors on linear regression*

Suppose we are interested in estimating the association between X recorded in $\mathcal{A}$ and Y recorded in $\mathcal{B}$, for instance through the slope β of the linear regression $Y = X\beta + \varepsilon$, with $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{E}[X\varepsilon] = 0$, so that

$$\hat{\beta} = \mathrm{Cov}(X, Y)/\mathrm{Var}(X). \tag{5}$$

The target is β in the population of entities present in both sources. The analyst only observes the linked data $\mathcal{D} = \{(i, j); \hat{D}_{i,j} > \xi\}$ with X_i from $\mathcal{A}$ and Y_j from $\mathcal{B}$. Y_j equals Y_i for truly linked pairs but belongs to another entity for falsely linked records. Among the linked data, a proportion $1 - \mathrm{FDP}$ (that may in reality depend on X) carries the correct Y and a proportion FDP carries a spurious one. Below we denote by $\hat{\boldsymbol{\Delta}}$ the estimator of $\boldsymbol{\Delta}$, with $\hat{\Delta}_{i,j} = 1$ for pairs (i, j) in $\mathcal{D}$. Truly linked pairs of records (true positives, TP) are thus denoted $\{\hat{\boldsymbol{\Delta}} = 1, \boldsymbol{\Delta} = 1\}$ and falsely linked pairs of records (false positives, FP) $\{\hat{\boldsymbol{\Delta}} = 1, \boldsymbol{\Delta} = 0\}$. Heuristically,

$$\begin{aligned}\mathbb{E}[Y \mid X, \hat{\boldsymbol{\Delta}} = 1] &= (1 - \mathrm{FDP}) \cdot \mathbb{E}[Y \mid X, \hat{\boldsymbol{\Delta}} = 1, \boldsymbol{\Delta} = 1] + \mathrm{FDP} \cdot \mathbb{E}[Y \mid X, \hat{\boldsymbol{\Delta}} = 1, \boldsymbol{\Delta} = 0] \\ &= (1 - \mathrm{FDP}) \cdot \mathbb{E}_{TP}[Y \mid X] + \mathrm{FDP} \cdot \mathbb{E}_{FP}[Y \mid X],\end{aligned}$$

where $\mathbb{E}_{TP}$ and $\mathbb{E}_{FP}$ denote expectations among truly and falsely linked pairs.

Let us recast $\hat{\beta}$ accordingly. If X has the same mean and variance among all data in $\mathcal{A}$, then $\hat{\beta}$ is a mixture of $\hat{\beta}_{TP}$ and $\hat{\beta}_{FP}$, defined as in Equation (5) for the subsets of TP and FP,

$$\hat{\beta} = (1 - \mathrm{FDP}) \cdot \hat{\beta}_{TP} + \mathrm{FDP} \cdot \hat{\beta}_{FP}. \tag{6}$$

Linkage errors are often assumed to occur at random, in which case $\mathbb{E}_{FP}[Y \mid X] = \mathbb{E}_{FP}[Y]$ and $\hat{\beta}_{FP} = 0$. However, the PIVs, often demographic, are likely to be associated with variables of interest in the analysis, so it is more realistic to consider an association between X and Y among falsely linked pairs.

The estimation error of the analysis can be decomposed into two parts:

$$\hat{\beta} - \beta = \underbrace{(\hat{\beta}_{TP} - \beta)}_{\text{sampling bias}} + \underbrace{(\hat{\beta} - \hat{\beta}_{TP})}_{\text{contamination bias}}.$$

Missed links may cause sample selection and falsely linked data may cause measurement error in the linked variables (Doidge & Harron, 2019; Robach et al., 2026; Wang et al., 2022).

Linkage decision rule trade-off

Raising the threshold ξ on the linkage scores reduces the number of falsely linked pairs, which lowers the FDP and hence the contamination bias. Falsely linked data would distort the signal in proportion to the FDP. Indeed, from Equation (6) we derive:

$$\underbrace{\hat{\beta} - \hat{\beta}_{TP}}_{\text{contamination bias}} = \mathrm{FDP}(\hat{\beta}_{FP} - \hat{\beta}_{TP}). \tag{7}$$

However, raising the threshold ξ also selects fewer and more atypical records, which may increase the sampling bias due to the implicit record linkage selection. Indeed, the chance of being linked may be related to X; in the context of linear regression, this yields:

$$\underbrace{\hat{\beta}_{TP} - \beta}_{\text{sampling bias}} = \frac{\mathrm{Cov}_{TP}(X, \beta X + \varepsilon)}{\mathrm{Var}_{TP}(X)} - \beta = \frac{\mathrm{Cov}_{TP}(X, \varepsilon)}{\mathrm{Var}_{TP}(X)}. \tag{8}$$

These aspects motivate the complementary tools we develop in *FlexRL* to quantify these two biases. Estimating the FDP (see Section 4.2) and evaluating the divergence of the linked sample from the data sources (see Section 4.3) over a range of values of ξ may help users appraise this trade-off.

4.2 *False discovery proportion*

For the decision rule $\xi \in [0.5, 1)$, the FDP is $\text{FDP}(\xi) = \text{FP}(\xi)/\{\text{TP}(\xi) + \text{FP}(\xi)\}$ where $\text{TP}(\xi)$ and $\text{FP}(\xi)$ are the numbers of truly and falsely linked pairs. In real data applications this quantity is unknown. Linkage errors can be estimated from the scores of the linkage model (Belin & Rubin, 1995; Enamorado et al., 2019; Resnick & Asher, 2019), from synthetic data (Lam et al., 2024; Robach, Hof, et al., 2025) or from a separate model under conditions on the distribution of the number of records per entity (Blakely & Salmond, 2002; Dasylva & Goussanou, 2020, 2022). *FlexRL* implements the first two estimation procedures. We focus on the FDP because it drives the contamination bias of Equation (7); the sampling bias is addressed by the diagnostics of Section 4.3. The estimators are provided as functions of ξ, together with the linked sample size available for inference, so that the user can choose a threshold controlling the estimated FDP (and hopefully, the true underlying FDP).

Score based estimator

If the linkage model is well calibrated, $\hat{D}_{i,j}$ should represent the probability that the pair (i, j) is a true link, and the expected number of truly linked pairs among linked data is $\sum_{(i,j)} \hat{D}_{i,j} \mathbb{1}\{\hat{D}_{i,j} > \xi\}$ (Enamorado et al., 2019; Robach, Hof, et al., 2025). Therefore

$$\widehat{\text{FDP}}(\xi) = 1 - \frac{\sum_{(i,j)} \hat{D}_{i,j} \mathbb{1}\{\hat{D}_{i,j} > \xi\}}{\sum_{(i,j)} \mathbb{1}\{\hat{D}_{i,j} > \xi\}}. \tag{9}$$

This estimator comes at no extra cost for any method that returns linkage scores, but it is only as good as the linkage model.

Notably, the calibration of the scores and the record linkage performance are affected by missing values and mistakes, the size of the overlap between the sources, the number of entities in the population, the discriminative power of the PIVs, their changes over time, and their marginal and joint distributions. The literature discusses the strengths and limitations of different record linkage paradigms in detail, including comparisons of *FlexRL* and *BRL* across a range of configurations (Robach, van der Pas, et al., 2025; Robach, Hof, et al., 2025).

Synthetic data estimator

An estimator that does not rely on linkage scores can be computed with synthetic data (Robach, Hof, et al., 2025). Source $\mathcal{B}$ is augmented with N_{synth} synthetic records, drawn from a generative model fitted on the PIVs of $\mathcal{B}$—for example *arf* (Watson et al., 2023), *synthpop* (Nowok et al., 2016) or *mice* (Buuren & Groothuis-Oudshoorn, 2011). Synthetic records by definition do not form a link in $\mathcal{A}$, so synthetic links can represent false positives when links and non-links have similar distributions (Robach, Hof, et al., 2025). Then $\text{FP}_{\text{synth}}(\xi)$ is rescaled by $N_{\mathcal{B}}/N_{\text{synth}}$, which follows from equating $\text{FP}(\xi)/(N_{\mathcal{A}} \cdot N_{\mathcal{B}})$ with $\text{FP}_{\text{synth}}(\xi)/(N_{\mathcal{A}} \cdot N_{\text{synth}})$. As for the score based estimator, $\text{TP}(\xi) + \text{FP}(\xi)$ corresponds to the number of (real) linked pairs $N_{\text{real}}(\xi) = \sum_{(i,j)} \mathbb{1}\{\hat{D}_{i,j} > \xi\}$:

$$\widehat{\text{FDP}}_{\text{synth}}(\xi) = \frac{\text{FP}_{\text{synth}}(\xi) \cdot N_{\mathcal{B}}/N_{\text{synth}}}{N_{\text{real}}(\xi)}. \tag{10}$$

The procedure is repeated over several synthetic draws and the results are averaged. The estimator does not depend on the linkage model, but it depends on the quality of the data synthesis.

When the two estimators disagree, a practitioner should rely on the one whose assumptions are more plausible. The score based estimator relies on the correct specification of the linkage model, including its specific assumptions. It reflects the performance the model would achieve under correct specification. The synthetic data estimator relies on the fidelity of the synthesiser and on the assumption that links and non-links have similar distributions. Earlier work assessed both estimators robustness to the degree of overlap between the data sources, the discriminative power of the PIVs and differences between the PIV distributions of true links and non-links (Robach, Hof, et al., 2025), but not to deviations from PIV independence. In the SHIW in Emilia-Romagna application, the PIV independence is violated (occupation and sex are associated; see the warning in Section 3.1). For models assuming independent PIVs, such as *FlexRL* and Fellegi-Sunter mixture models (see Equation (1)), we expect the score based estimator to

be biased, as its validity rests on the violated assumption. The synthetic data estimator though, should remain valid provided the synthesiser captures the dependencies among PIVs. This is in line with what we observe in the output of (**fig-print-output?**).

4.3 *Representativeness of the linked sample*

As explained in Section 4.1, the linked sample may differ from the data sources in ways the analyst cannot see (Doidge & Harron, 2019; Robach et al., 2026; Wang et al., 2022). In particular, we appraise the sampling bias of Equation (8) by comparing the distributions of variables of interest between the linked data ('selected' through record linkage) and the original data sources. For any decision rule $\xi \in [0.5, 1)$, *FlexRL* compares each source with its linked subset using the following measures, computed on sets of variables $\mathcal{V}_{\mathcal{A}}$ and $\mathcal{V}_{\mathcal{B}}$ available in sources $\mathcal{A}$ and $\mathcal{B}$ respectively:

- *Agreement rates*: the proportion of linked pairs whose records agree on v, for each variable $v \in \mathcal{V}_{\mathcal{A}} \cap \mathcal{V}_{\mathcal{B}}$ (such as the PIVs).
- *Standardised Mean Differences* (SMD): $(\text{mean}(v_{\text{linked}}) - \text{mean}(v_{\text{source}}))/\text{sd}(v_{\text{source}})$, for a continuous variable v in a given set; the same quantity is computed for the indicator of each level when v is categorical.
- *Overlap of supports* (support IoU): the Intersection over Union (IoU) of the sets of values taken in the linked sample and in the source, for a variable v in a given set.
- *Maximum Mean Discrepancy* (MMD): a multivariate measure of the difference between the linked sample and the source, computed jointly over a given set of variables, with a Gaussian kernel and the median heuristic for the bandwidth (Gretton et al., 2012).

These quantities are computed over a grid of values of ξ. They are built on the declared pairs and, when available, their linkage scores; they therefore apply to any linkage method.

In (**tbl-demonstration-2?**) we proceed with the demonstration of Section 2.4 showcasing the functionalities of *FlexRL* on a simulation study. The synthetic data FDP estimates ('est. FDP* synth') are not reliable for *fedmatch* and *FlexRL (dynamics)*, possibly because of the specific linkage procedures these packages implement. For *FlexRL (dynamics)*, the cause is likely the misspecified PIV dynamics model, or the added difficulty of estimating the FDP under modelling of PIV dynamics, since the estimate for *FlexRL (no dynamics)* is reliable.

	Naive	BRL	multilink	fastLink	reclin2	diyar	fedmatch	FlexRL (dynamics)	FlexRL (no dynamics)
TP	118.40 (5.13)	80.40 (5.04)	80.60 (5.95)	101.70 (6.48)	114.10 (6.38)	34.90 (20.24)	122.10 (6.33)	114.00 (7.92)	87.50 (8.30)
FP	83.10 (11.02)	12.30 (3.23)	14.20 (4.13)	43.10 (5.59)	68.30 (8.82)	3.30 (2.91)	125.80 (6.32)	25.90 (5.15)	19.00 (4.67)
FN	81.60 (5.13)	119.60 (5.04)	119.40 (5.95)	98.30 (6.48)	85.90 (6.38)	165.10 (20.24)	77.90 (6.33)	86.00 (7.92)	112.50 (8.30)
sensitivity	0.59 (0.03)	0.40 (0.03)	0.40 (0.03)	0.51 (0.03)	0.57 (0.03)	0.17 (0.10)	0.61 (0.03)	0.57 (0.04)	0.44 (0.04)
FDP	0.41 (0.03)	0.13 (0.03)	0.15 (0.04)	0.30 (0.03)	0.37 (0.03)	0.08 (0.05)	0.51 (0.03)	0.18 (0.03)	0.18 (0.04)
est. FDP score	NA	0.19 (0.02)	NA	0.37 (0.02)	0.44 (0.03)	NA	0.10 (0.01)	0.20 (0.01)	0.18 (0.02)
est. FDP* score	NA	0.21 (0.02)	NA	0.39 (0.02)	0.43 (0.07)	NA	0.10 (0.01)	0.21 (0.01)	0.19 (0.02)
est. FDP* synth	NA	0.18 (0.10)	0.20 (0.05)	0.33 (0.07)	0.45 (0.14)	0.14 (0.07)	0.08 (0.03)	0.37 (0.11)	0.27 (0.08)
max SMD	NA	0.22 (0.05)	0.22 (0.06)	0.17 (0.03)	0.24 (0.04)	0.68 (0.31)	0.15 (0.03)	0.17 (0.04)	0.22 (0.04)
min support IoU	NA	0.80 (0.01)	0.80 (0.01)	0.80 (0.01)	0.80 (0.01)	0.76 (0.10)	0.82 (0.03)	0.82 (0.03)	0.81 (0.02)
max MMD	NA	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.08 (0.05)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
time (secs)	0.27 (0.04)	0.97 (0.05)	18.57 (1.48)	1.29 (0.24)	0.29 (0.01)	1.17 (0.06)	1.44 (0.01)	4.59 (0.02)	3.72 (0.01)
memory (Mb)	1.04 (0.05)	78.60 (10.08)	105.68 (10.18)	24.85 (4.42)	41.52 (0.18)	104.31 (10.84)	42.14 (2.59)	102.11 (9.03)	70.18 (6.42)

Linkage performance of competing record linkage methods on simulated data for the default decision rule $\xi = 0.5$, with linked data quality assessed through *FlexRL*. The * symbol denotes the record linkage augmented task, see Section 4.2. We report average performance over 10 replications, with standard deviations in parentheses.

This short simulation study demonstrates the potential of the *FlexRL* package. Since *FlexRL* provides linked data quality diagnostics for any record linkage method, we report three estimates of the FDP along with measures of the divergence of the linked sample from the data sources. Section 4.2 presented the FDP estimators implemented in the package and Section 4.3 discussed the measures of divergence. The FDP estimators are averaged over 5 procedures in each of the 10 replications.

4.4 *Diagnostics*

`RL_diagnostics()` gathers the tools of Section 4.2 and Section 4.3, summarised in Table 5, and returns an S3 object of class `RL_diagnostics`, with `print` and `plot` methods. All measures are computed over a grid of thresholds $\xi \in [0.5, 1)$ (when the linkage method allows it by returning scores) or for a fixed threshold when specified. Each diagnostic can also be computed on its own. We present all commands hereafter but show only some outputs. Further details on these diagnostic tools are given in the package vignette 'Linkage and Diagnostics with FlexRL' available through `vignette('FlexRL-RL-vignette', 'FlexRL')` or the online documentation.

Table 5 Diagnostics performed by `RL_diagnostics()`.

Diagnostic	Computation	Display
Convergence	`plot_StEM_convergence()` on the chains returned by `StEM()`	`plot(x, 'convergence')`
Linkage scores	`plot_linkage_scores()`	`plot(x, 'scores')`
FDP estimation	`compute_RL_FDP_score()` (Equation (9)), `compute_augmRL_FDP_synth()` (Equation (10))	`plot(x, 'FDP')`
Discrepancy measures	`smd()`, `support_iou()`, `mmd()`, `RL_agreement()`	`plot(x, 'discrepancy')`
Data comparison	`plot_distributions()` on each source and its linked subset	`plot(x, 'distributions')`

The function takes several arguments as input. The record linkage output `fit`, the data sources `encodedA` and `encodedB`, the set of variables of interest to compare `compare_vars`, their type `vars_type_cont` (`TRUE` if continuous) and the `PIVs` are passed to `RL_diagnostics()`, which computes FDP estimates and discrepancy measures that characterise the linked sample across a grid of thresholds ξ. The `true_pairs` enables reporting the true FDP and sensitivity when known.

```
compare_vars <- c("BirthYear", "Sex", "MaritalStatus",
                  "BirthRegion", "Occupation")

vars_type_cont <- list(BirthYear = TRUE, Sex = FALSE, MaritalStatus = FALSE,
                       BirthRegion = FALSE, Occupation = FALSE)

diag <- RL_diagnostics(fit, prep_data$encodedA, prep_data$encodedB,
                       compare_vars, vars_type_cont, PIVs,
                       true_pairs = prep_data$true_pairs, FDP_estimation = TRUE,
                       RL_method = "FlexRL", data = prep_data)
```

The FDP estimation is optional (`FDP_estimation = TRUE`) since it can be time consuming. For a given `RL_method`, it is repeated `n_repeats` times and the estimates are averaged. `compute_RL_FDP_score()` returns the score based estimate on the original task. `compute_augmRL_FDP_synth()` adds `n_synth` records generated by `synth_method` to $\mathcal{B}$, links the augmented sources as an auxiliary task and computes both the score based and the synthetic data estimates on it. Similar score based estimates on the original and augmented tasks support the use of the synthetic data estimator. (**tbl-demonstration-2?**) reported these three estimates as 'est. FDP score', 'est. FDP* score', 'est. FDP* synth'.

The `plot(x, type)` method for an object of class `RL_diagnostics` draws the plots listed in Table 5, with `type` being one of `'convergence'`, `'scores'`, `'FDP'`, `'discrepancy'` and `'distributions'`. With the `'convergence'` plots, we could verify that the StEM chains have stabilised.

```
plot(diag, 'convergence')
```

With the `'scores'` plot, we can analyse the distribution of the linkage scores of the `fit` as shown in Figure 2.

```
plot(diag, 'scores')
```

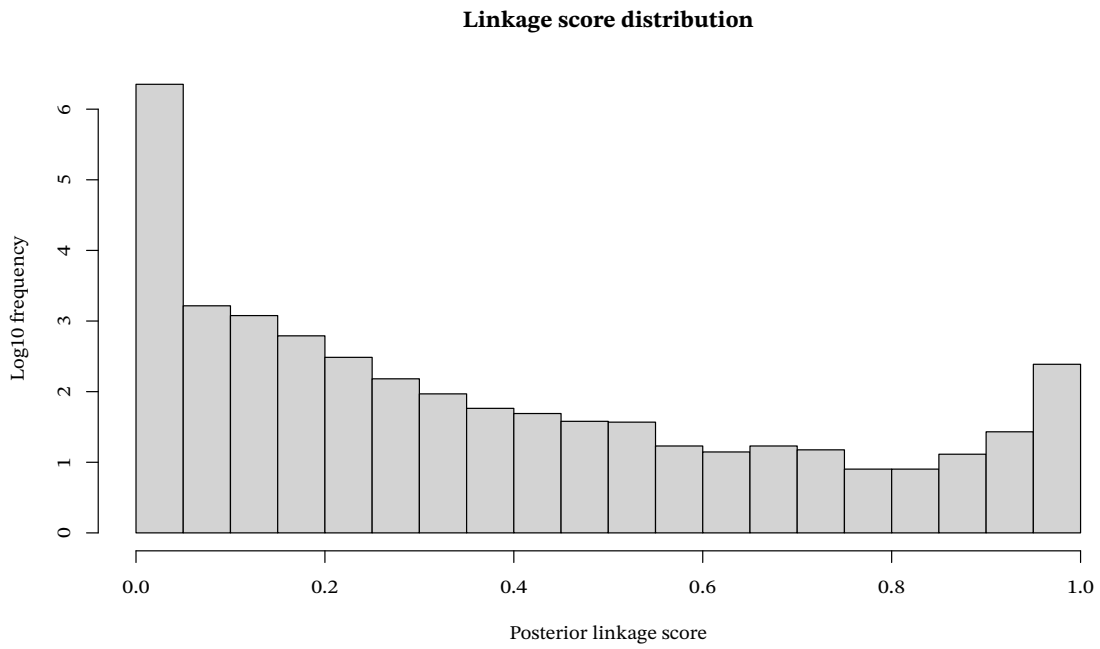
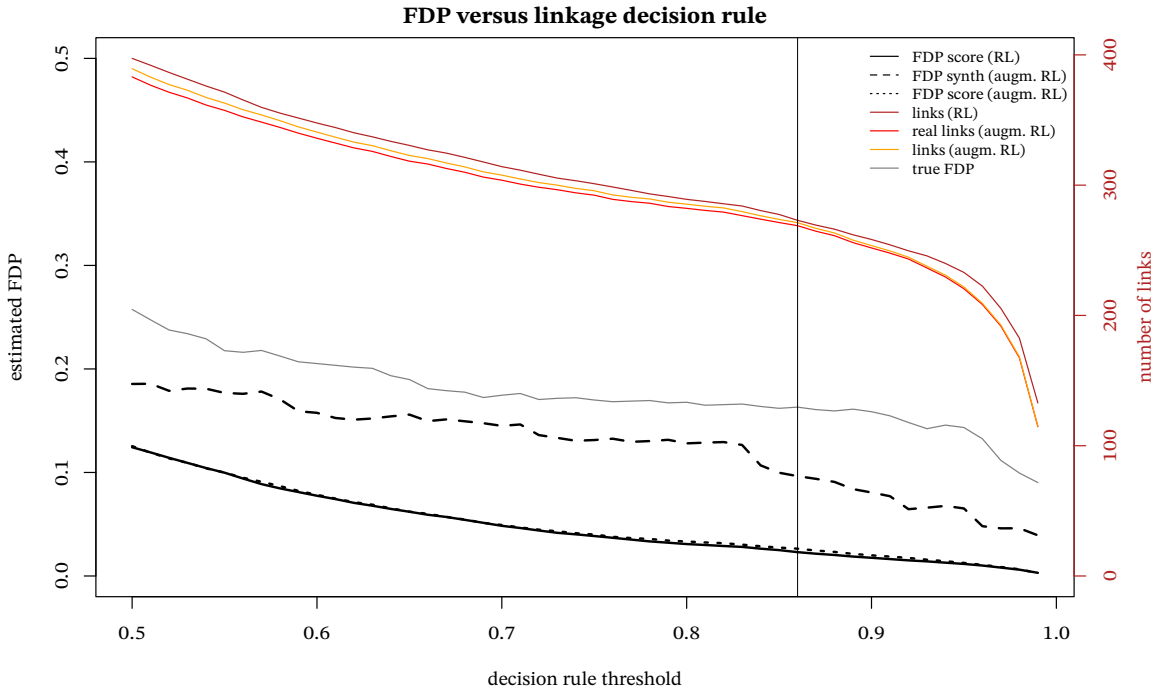


Figure 2 Distribution of the linkage scores returned by *FlexRL* `StEM()` on the SHIW in Emilia-Romagna.

Albeit bimodal, the distribution of linkage scores produced by the record linkage algorithm does not indicate where to set the threshold ξ. The FDP estimation results and the divergence plots may help further. In Figure 3 we show the `'FDP'` plot and some of the `'discrepancy'` plots for the MMD and the SMD. Similar plots for the overlap of supports and the agreement rate are usually displayed by `plot(diag, 'discrepancy')`.

```
plot(diag, 'FDP')
```



```
plot(diag, 'discrepancy')
```

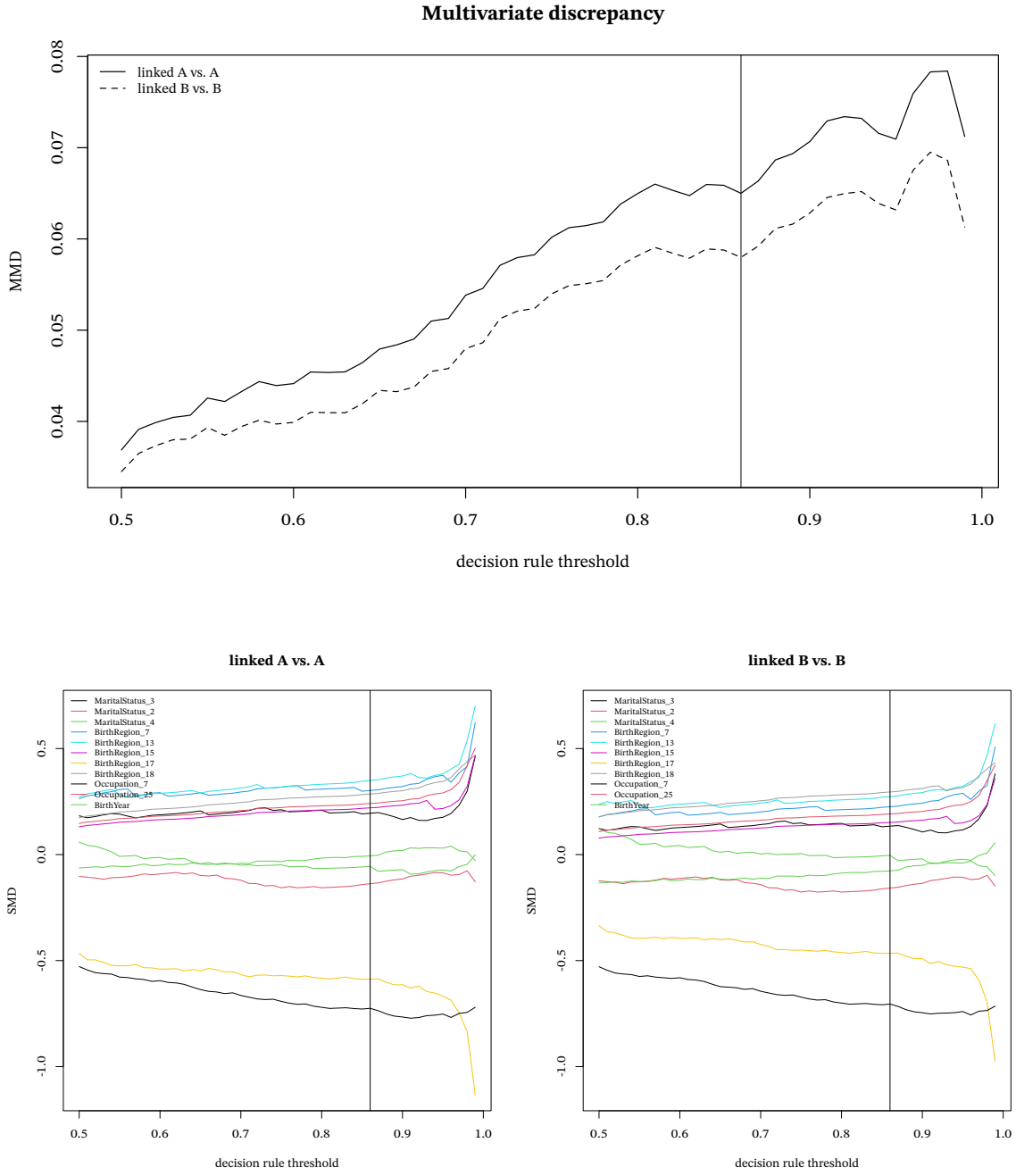


Figure 3 Estimated FDP and discrepancy measures against the linkage decision rule threshold obtained with *FlexRL* `RL_diagnostics()` on the SHIW in Emilia-Romagna.

The graphics from Figure 3 show how each linkage decision rule affects linked data quality. They illustrate the trade-off of Section 4.1: as the threshold ξ increases, the estimated FDP diminishes while the divergence of the linked sample from the sources grows. Ideally, we choose an appropriate threshold for inference at the 'elbow' of the divergence curves, provided that the estimated FDP at that level is reasonable. This decision rule is a heuristic motivated by the potential impact of linkage errors on inference (Doidge & Harron, 2019; Lahiri & Larsen, 2005; Robach et al., 2026; Tancredi & Liseo, 2015; Wang et al., 2022) and its reliability has not been assessed. Note that the divergence measures are indicators of the

sampling bias but do not estimate Equation (8). For our example, we choose $\xi = 0.86$ following this rule, as indicated by the vertical line in Figure 3.

With the `'distributions'` plots, we could compare the distribution of the linked data and of the data sources at the selected threshold—for example $\xi = 0.86$.

```
plot(diag, 'distributions', threshold = 0.86)
```

The `print()` method for an S3 object of class `RL_diagnostics` reports the linked sample size, the FDP estimates, the true FDP and sensitivity when available, and the discrepancy measures. We report the output in (**fig-print-output?**). As discussed in Section 4.2, the FDP estimators have different strengths and weaknesses. In the SHIW in Emilia-Romagna, the synthetic data estimator yields an estimate of 0.07 and the score based estimator yields an estimate of 0.01, compared with the true value of 0.15.

```
print(diag, threshold = 0.86)
```

```
  Linked pairs  (score > 0.86):  282
```

```
  FDP           (true pairs):    0.16
  Sensitivity   (true pairs):    0.26
```

```
  FDP score estimate (RL task):           min threshold 0.50: FDP ~ 0.12 (sd 0.00)
                                              threshold 0.86: FDP ~ 0.02 (sd 0.00)
                                          max threshold 0.99: FDP ~ 0.00 (sd 0.00)
  FDP score estimate (augmented RL task): min threshold 0.50: FDP ~ 0.13 (sd 0.00)
                                              threshold 0.86: FDP ~ 0.03 (sd 0.00)
                                          max threshold 0.99: FDP ~ 0.00 (sd 0.00)
  FDP synth estimate (augmented RL task): min threshold 0.50: FDP ~ 0.19 (sd 0.06)
                                              threshold 0.86: FDP ~ 0.10 (sd 0.07)
                                          max threshold 0.99: FDP ~ 0.04 (sd 0.07)
```

```
  Multivariate MMD  (linked A vs. A):  0.07
```

```
  Multivariate MMD  (linked B vs. B):  0.06
```

```
  SMD BirthYear                          (linked A vs. A):  -0.01
  SMD Sex values: {1, 2}                 (linked A vs. A):  -0.01, 0.01
  SMD MaritalStatus values: {2, 3, ...}  (linked A vs. A):  -0.14, 0.19, ...
  SMD BirthRegion values: {13, 17, ...}  (linked A vs. A):  0.35, -0.59, ...
  SMD Occupation values: {7, 25, ...}    (linked A vs. A):  -0.73, 0.24, ...
```

```
  SMD BirthYear                          (linked B vs. B):  -0.08
  SMD Sex values: {1, 2}                 (linked B vs. B):  -0.01, 0.01
  SMD MaritalStatus values: {2, 3, ...}  (linked B vs. B):  -0.16, 0.13, ...
  SMD BirthRegion values: {17, 18, ...}  (linked B vs. B):  -0.47, 0.30, ...
  SMD Occupation values: {7, 10, ...}    (linked B vs. B):  -0.71, 0.22, ...
```

Example of output from the `print()` method on *FlexRL* `RL_diagnostics` class, at threshold 0.86, on the SHIW in Emilia-Romagna.

Across the diagnostics, record linkage is run through wrappers with a common form, `link_with_<pkg>(dataA, dataB, ...)`, returning the indices of linked pairs and their linkage score. Wrappers exist for *FlexRL*, *fastLink*, *BRL*, *reclin2*, *multilink*, *diyar* and *fedmatch*. That way, the diagnostic tools apply to any of these methods in a similar fashion.

On the illustrative example developed along this article, we regress the expected age of retirement in 2014 on the expected age of retirement in 2012 (originally named 'etapen' in each wave). After preparation of the SHIW in Emilia-Romagna data for record linkage (see `prepare_data()` in Section 3.1), we work

with 1479 and 1524 records in the two data sources, which share 898 true links. The solid red line in Figure 4 is the reference estimate computed on these true links.

The tools presented so far help identify a suitable linkage threshold by assessing the trade-off between contamination and sampling bias. In Figure 4, a low threshold $\xi = 0.55$ yields a linked data estimate (dotted orange) affected by contamination bias, due to noisy falsely linked data (short dashed yellow estimate); see Equation (7). Conversely, a high threshold $\xi = 0.99$ yields a linked data estimate (dotted orange) affected by sampling bias, due to atypical truly linked data (long dashed yellow estimate); see Equation (8).

Guided by the diagnostic tools, we choose $\xi = 0.86$, which yields a linked data estimate (dotted orange) closer to the reference. At this threshold, we retain 282 linked observations, corresponding to an FDP of 0.16 and a sensitivity of 0.26.

5 Scalability

In (**tbl-illustration?**) of Section 3, we compared record linkage methods on the SHIW in Emilia-Romagna. Despite heterogeneous linkage performance, running time and memory usage, all methods ran at that scale. To complete our investigation, we demonstrate the scalability of *FlexRL* and examine that of other record linkage methods on the complete SHIW data sets. After preparation, the 2012 and 2014 waves over the 20 Italian regions cover 19215 and 20002 individuals, with 10568 individuals observed in both waves. We consider two settings, without and with regional blocking. Note that blocking on the region assumes that this variable is stable and free of registration errors.

In Section 3.1, we showed the proportion of agreements between the information carried by records among true links of the SHIW in Emilia-Romagna. Below we show the proportion of agreements among true links of the complete SHIW data. In this application, the region is indeed free of registration errors and stable.

```
BirthYear           Sex MaritalStatus   BirthRegion    Occupation
     0.99          1.00          0.98          1.00          0.82
   Region
     1.00
```

Record linkage becomes harder on large data sets: the discriminative power of the PIVs is unlikely to grow with the size of the sources, whereas the number of candidate records in $\mathcal{B}$ for each record in $\mathcal{A}$ does. Without blocking, we may therefore expect higher FDPs and also higher sensitivities, since true links that blocking excludes (due to registration errors or changes between waves) become recoverable. However, in the complete SHIW the region is stable and error free so the expected gain does not apply. FDPs and sensitivities are similar without and with blocking for the comparable methods, except for *FlexRL* which has both a higher FDP and a higher sensitivity without blocking. Note that `prepare_data()` eliminates out-of-support records at regional levels, which explains the differences in the partial confusion matrix of the 'Naive' method between (**tbl-scalability?**) and (**tbl-scalability-block?**).

Below, we report the number of unique values of each PIV in the SHIW in Emilia-Romagna and in the complete SHIW. These counts indicate the discriminative power of the PIVs in the regional and national applications. In addition, we appraise the difficulty of the tasks, without and with regional blocking, through the performance of the 'Naive' approach in (**tbl-scalability?**) and (**tbl-scalability-block?**), which reflects how many records share the same combination of PIV values.

```
BirthYear           Sex MaritalStatus   BirthRegion    Occupation
       94             2             4            18            52
```

```
BirthYear           Sex MaritalStatus   BirthRegion    Occupation
       99             2             4            20            84
   Region
       20
```

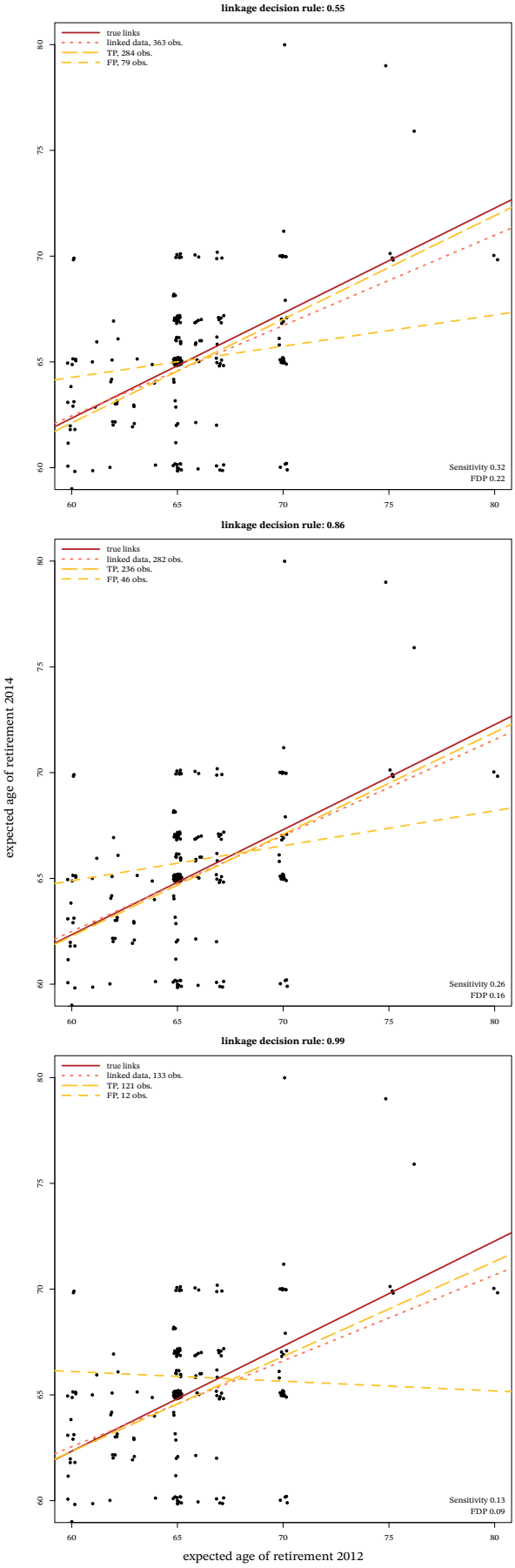


Figure 4 Regression analysis on the SHIW in Emilia-Romagna linked through *FlexRL*.

(**tbl-scalability?**) presents the comparative study without blocking; 'Not Available' results indicate memory errors. When supported, methods run with four threads (`threads` for *FlexRL*, `n.cores` for *fastLink*, `nthread` for *fedmatch*). (**tbl-scalability-block?**) shows the results with blocking on the Italian regions. More details on this data application showcasing the scalability of *FlexRL* are given in the package vignette 'Scalable Record Linkage with FlexRL' available through `vignette('FlexRL-RL-block-vignette', 'FlexRL')` or the online documentation.

	Naive	BRL	multilink	fastLink	reclin2	diyar	fedmatch	FlexRL
TP	8289.00	NA	NA	5061.00	NA	NA	5057.00	4105.00
FP	49350.00	NA	NA	11175.00	NA	NA	14154.00	5417.00
FN	2279.00	NA	NA	5507.00	NA	NA	5511.00	6463.00
sensitivity	0.78	NA	NA	0.48	NA	NA	0.48	0.39
FDP	0.86	NA	NA	0.69	NA	NA	0.74	0.57
time (mins)	0.00	NA	NA	9.00	NA	NA	5.00	6.00
memory (Mb)	123.20	NA	NA	20070.10	NA	NA	1877.00	4030.30

Comparison of record linkage methods on the complete SHIW data application for the default decision rule $\xi = 0.5$.

	Naive	BRL	multilink	fastLink	reclin2	diyar	fedmatch	FlexRL
TP	8289.00	1609.00	1771.00	4911.00	8993.00	316.00	5284.00	2916.00
FP	49346.00	449.00	1092.00	11295.00	4092774.00	12.00	13123.00	1308.00
FN	2059.00	8739.00	8577.00	5437.00	1355.00	10032.00	5064.00	7432.00
sensitivity	0.80	0.16	0.17	0.48	0.87	0.03	0.51	0.28
FDP	0.86	0.22	0.38	0.70	1.00	0.04	0.71	0.31
time (mins)	0.33	3.63	206.82	5.82	0.42	2.67	2.25	5.03
memory (Mb)	16.00	1150.00	3837.00	4716.00	2223.00	1530.00	766.00	862.00

Comparison of record linkage methods on the complete SHIW data application with regional blocking, for the default decision rule $\xi = 0.5$.

6 Discussion

This article presented *FlexRL*: an `R` package for probabilistic record linkage of two overlapping data sources using a hierarchical model that accounts for registration errors and accommodates temporal changes in Partially Identifying Variables (PIVs). *FlexRL* provides tools for assessing the quality of the resulting linked data. These diagnostics are independent of the linkage method and thus apply broadly. These tools are essential for using record linkage in practice, where potential linkage errors and induced sample selection cannot be observed directly.

To draw attention to the implications of linkage quality for statistical analysis, *FlexRL* provides two complementary tools as functions of the linkage threshold. First, the estimated False Discovery Proportion (FDP), which quantifies contamination among the linked pairs. Second, the divergence of the linked sample from the source populations, which captures potential sampling bias. Examining these quantities jointly makes the trade-off between contamination and sampling bias explicit and provides users with information to select a linkage decision rule in light of its potential consequences for subsequent inference.

Several directions follow naturally from this work. On the computational side, the memory footprint is linear in the number of records and the running time scales with the number of records, candidate pairs, PIVs, their levels and configurations. Memory and running time allow record linkage between large data sets on a standard computer. On the modelling side though, the linkage model currently assumes independence of PIVs given the latent true values, no duplicates within a source, categorical PIVs; extensions would broaden the range of applications. Moreover, the diagnostic estimators and the

proposed quality measures, whose validity rests on the assumptions underlying their estimation, would benefit from further theoretical and empirical investigation.

Notwithstanding these considerations, *FlexRL* provides a framework that connects record linkage to linked data quality assessment. It offers a solution to record linkage that is scalable in terms of memory usage and running time, and flexible in accommodating diverse scenarios. Linkage performance depends on the difficulty of the task and the chosen decision rule, controlling the FDP and recovering a fraction of the true links accordingly. It brings record linkage into practice, helping researchers and practitioners not only to link data, but also to quantify and communicate the extent to which linkage decisions may affect downstream inference.

Data and Code Availability

The *FlexRL* package is available from the Comprehensive R Archive Network (`CRAN`) or from the package GitHub repository alongside documentation and vignettes describing its functionality. *FlexRL* is also available as a module in `JASP 1.0` (JASP Team, 2026). The script is available as supplementary material and on the GitHub repository of the package.

Acknowledgements

The authors thank the anonymous reviewers for their valuable suggestions and for conducting a thorough reproducibility check.

Competing interests

The authors are the developers of *FlexRL*.

References

Aleshin-Guendel, S. (2023). *Multifile record linkage and duplicate detection* [R package version 0.1.1]. https://CRAN.R-project.org/package=multilink

Aleshin-Guendel, S., & Sadinle, M. (2023). Multifile Partitioning for Record Linkage and Duplicate Detection. *Journal of the American Statistical Association*, *118*(543), 1786–1795. https://doi.org/10.1080/01621459.2021.2013242

Belin, T. R., & Rubin, D. B. (1995). A Method for Calibrating False-Match Rates in Record Linkage. *Journal of the American Statistical Association*, *90*(430), 694–707. https://doi.org/10.1080/01621459.1995.10476563

Binette, O., & Steorts, R. C. (2022). (Almost) all of entity resolution. *Science Advances*, *8*(12). https://doi.org/10.1126/sciadv.abi8021

Blakely, T., & Salmond, C. (2002). Probabilistic record linkage and a method to calculate the positive predictive value. *International Journal of Epidemiology*, *31*(6), 1246–1252. https://doi.org/10.1093/ije/31.6.1246

De Bruin, J. (2019). *Python Record Linkage Toolkit: A toolkit for record linkage and duplicate detection in Python* [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.3559043

Bukke, P., Kamat, G., Cui, J., Gutman, R., & Slawski, M. (2026). *Postlink: Post-linkage data analysis* [R package version 0.1.2]. https://postlink-group.github.io/postlink/

Buuren, S. van, & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in*R*. *Journal of Statistical Software*, *45*(3). https://doi.org/10.18637/jss.v045.i03

Celeux, G., & Diebolt, J. (1986). L'algorithme SEM : Un algorithme d'apprentissage probabiliste pour la reconnaissance de mélange de densités. *Revue de Statistiques Appliquées*, *34*(2), 35–52. http://www.numdam.org/item/RSA_1986__34_2_35_0/

Cleaton, M., Hall, J., Shipsey, R., White, Z., & Xhaferaj, K. (2024). A case study of using Splink: Census duplicate matching. *Proceedings of Statistics Canada Symposium 2022. Data Disaggregation: Building a More Representative Data Portrait of Society*.

Cohen, G. J., Dice, J., Friedrichs, M., Gupta, K., Hayes, W., Kitschelt, I., Lee, S. J., Marsh, W. B., Mislang, N., Shaton, M., Sicilian, M., & Webster, C. (2021). The U.S. syndicated loan market: Matching data. *Journal of Financial Research*, *44*(4), 695–723. https://doi.org/10.1111/jfir.12261

Cooney, T., Yu, S., Daral, S., Field, M., Krebs, B., Kieran, M., Manley, P., Govinda Raju, S., & Zelt, S. (2023). Healthcare resource use for pediatric low-grade glioma care: A cohort of linked electronic health records and

claims data. *Journal of Clinical Oncology*, *41*(16_suppl), 10038–10038. https://doi.org/10.1200/JCO.2023.41.16_suppl.10038

Dasylva, A., & Goussanou, A. (2020). Estimating linkage errors under regularity conditions. *Proceedings of the Survey Research Methods Section*.

Dasylva, A., & Goussanou, A. (2022). On the consistent estimation of linkage errors without training data. *Japanese Journal of Statistics and Data Science*, *5*(1), 181–216. https://doi.org/10.1007/s42081-022-00153-3

Doidge, J. C., & Harron, K. L. (2019). Reflections on modern methods: linkage error bias. *International Journal of Epidemiology*. https://doi.org/10.1093/ije/dyz203

Dunn, H. L. (1946). Record Linkage. *American Journal of Public Health and the Nations Health*, *36*(12), 1412–1416. https://doi.org/10.2105/ajph.36.12.1412

Dusetzina, S. B., Tyree, S., Meyer, A. M., Meyer, A., Green, L., & Carpenter, W. R. (2014). Linking data for health services research: A framework and instructional guide [internet]. In *Agency for Healthcare Research and Quality (US)*. https://www.ncbi.nlm.nih.gov/books/NBK253312/. https://www.ncbi.nlm.nih.gov/books/NBK253312/

ENAMORADO, T., FIFIELD, B., & IMAI, K. (2019). Using a Probabilistic Model to Assist Merging of Large-Scale Administrative Records. *American Political Science Review*, *113*(2), 353–371. https://doi.org/10.1017/s0003055418000783

Fellegi, I. P., & Sunter, A. B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*, *64*(328), 1183–1210. https://doi.org/10.1080/01621459.1969.10501049

Fortini, M., Liseo, B., Nuccitelli, A., & Scanu, M. (2001). On Bayesian record linkage. *Research in Official Statistics*, *4*(6), 185–198.

Friedrichs, M., Webster, C., Marsh, B., Dice, J., & Lee, S. (2023). *Fedmatch: Fast, flexible, and user-friendly record linkage methods* [R package version 2.0.6]. https://CRAN.R-project.org/package=fedmatch

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, *13*, 723–773.

Harron, K., Doidge, J. C., & Goldstein, H. (2020). Assessing data linkage quality in cohort studies. *Annals of Human Biology*, *47*(2), 218–226. https://doi.org/10.1080/03014460.2020.1742379

Harron, K. L., Doidge, J. C., Knight, H. E., Gilbert, R. E., Goldstein, H., Cromwell, D. A., & van der Meulen, J. H. (2017). A guide to evaluating linkage quality for the analysis of linked data. *International Journal of Epidemiology*, *46*(5), 1699–1710. https://doi.org/10.1093/ije/dyx177

Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, *84*(406), 414–420. https://doi.org/10.1080/01621459.1989.10478785

JASP Team. (2026). *JASP (Version 1.0.0)[Computer software]*. https://jasp-stats.org/

Jelveh, Z., McNeill, M., & Lin, T.-Y. (2022). *NameMatch*. https://github.com/urban-labs/namematch

Kramer, R. (2026). Towards scalable probabilistic linkage of administrative population data in Germany: Insights from a method test with Splink. *International Journal of Population Data Science*, *11*(4). https://doi.org/10.23889/ijpds.v11i4.3814

Lahiri, P., & Larsen, M. D. (2005). Regression Analysis With Linked Data. *Journal of the American Statistical Association*, *100*(469), 222–230. https://doi.org/10.1198/016214504000001277

Laidler, J., Imaz Blanco, A., & Balasubramanian, D. (2025). Probabilistic Linkage Pipeline Improving Linkage Quality and Explainability in Healthcare. *International Journal of Population Data Science*, *10*(4). https://doi.org/10.23889/ijpds.v10i4.3271

Lam, J., Boyd, A., Linacre, R., Blackburn, R., & Harron, K. (2024). Generating synthetic identifiers to support development and evaluation of data linkage methods. *International Journal of Population Data Science*, *9*(1). https://doi.org/10.23889/ijpds.v9i1.2389

Larsen, M. D. (1999). Multiple imputation analysis of records linked using mixture models. *Proceedings of the Survey Methods Section, Statistical Society of Canada Annual Meeting*, *63*, 65–71.

Linacre, R., Lindsay, S., Manassis, T., Slade, Z., & Hepworth, T. (2022). Splink: Free software for probabilistic record linkage at scale. *International Journal of Population Data Science*, *7*(3). https://doi.org/10.23889/ijpds.v7i3.1794

Marchant, N. G., Kaplan, A., Elazar, D. N., Rubinstein, B. I. P., & Steorts, R. C. (2021). d-blink: Distributed End-to-End Bayesian Entity Resolution. *Journal of Computational and Graphical Statistics*, *30*(2), 406–421. https://doi.org/10.1080/10618600.2020.1825451

Marchant, N. G., Rubinstein, B. I. P., & Steorts, R. C. (2023). Bayesian Graphical Entity Resolution using Exchangeable Random Partition Priors. *Journal of Survey Statistics and Methodology*, *11*(3), 569–596. https://doi.org/10.1093/jssam/smac030

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., & Raghavendra, V. (2018). Deep Learning for Entity Matching: A Design Space Exploration. *Proceedings of the 2018 International Conference on Management of Data*, 19–34. https://doi.org/10.1145/3183713.3196926

Newcombe, H. B., Kennedy, J. M., Axford, S. J., & James, A. P. (1959). Automatic Linkage of Vital Records: Computers can be used to extract "follow-up" statistics of families from files of routine records. *Science*, *130*(3381), 954–959. https://doi.org/10.1126/science.130.3381.954

Nowok, B., Raab, G. M., & Dibben, C. (2016). synthpop : Bespoke Creation of Synthetic Data in *R*. *Journal of Statistical Software*, *74*(11). https://doi.org/10.18637/jss.v074.i11

Nsonwu, O. (2023). *Diyar: Record linkage and epidemiological case definitions in 'r'* [R package version 0.5.1]. https://olisansonwu.github.io/diyar/index.html

Pontesilli, M., Hof, M. H., Ravelli, A. C. J., van Altena, A. J., Soufan, A. T., Mol, B. W., Kostelijk, E. H., Slappendel, E., Consten, D., Cantineau, A. E. P., van der Westerlaken, L. A. J., van Inzen, W., Dumoulin, J. C. M., Ramos, L., Baart, E. B., Broekmans, F. J. M., Rijnders, P. M., Curfs, M. H. J. M., Mastenbroek, S., … Painter, R. C. (2021). Effect of parental and ART treatment characteristics on perinatal outcomes. *Human Reproduction*, *36*(6), 1640–1665. https://doi.org/10.1093/humrep/deab008

Resnick, D., & Asher, J. (2019, September). Measurement of type i and type II record linkage error. *Using and Linking Administrative and Auxiliary Data, Government Statistics Section*. https://ww2.amstat.org/meetings/proceedings/2019/data/assets/pdf/1199462.pdf

Robach, K., & Hof, M. H. (2026). *FlexRL: Flexible record linkage and inference on linked data* [R package version 2.0.0]. https://CRAN.R-project.org/package=FlexRL

Robach, K., Hof, M. H., & van de Wiel, M. A. (2025). False Discovery Estimation in Record Linkage. *Statistics in Medicine*, *44*(23-24). https://doi.org/10.1002/sim.70292

Robach, K., van de Wiel, M. A., Hof, M. H., & van der Pas, S. L. (2026). *Causal record linkage*. https://arxiv.org/abs/2611.00760

Robach, K., van der Pas, S. L., van de Wiel, M. A., & Hof, M. H. (2025). A flexible model for record linkage. *Journal of the Royal Statistical Society Series C: Applied Statistics*, *74*(4), 1100–1127. https://doi.org/10.1093/jrsssc/qlaf016

Sadinle, M. (2017). Bayesian Estimation of Bipartite Matchings for Record Linkage. *Journal of the American Statistical Association*, *112*(518), 600–612. https://doi.org/10.1080/01621459.2016.1148612

Sadinle, M. (2020). *Beta record linkage* [R package version 0.1.0]. https://CRAN.R-project.org/package=BRL

Steorts, R. C. (2015). Entity Resolution with Empirically Motivated Priors. *Bayesian Analysis*, *10*(4), 849–875. https://doi.org/10.1214/15-ba965si

Steorts, R. C., Hall, R., & Fienberg, S. E. (2016). A Bayesian Approach to Graphical Record Linkage and Deduplication. *Journal of the American Statistical Association*, *111*(516), 1660–1672. https://doi.org/10.1080/01621459.2015.1105807

Steorts, R. C., Ventura, S. L., Sadinle, M., & Fienberg, S. E. (2014). A Comparison of Blocking Methods for Record Linkage. In *Lecture Notes in Computer Science* (pp. 253–268). Springer International Publishing. https://doi.org/10.1007/978-3-319-11257-2_20

Steorts, R., Hall, R., & Fienberg, S. (2014). SMERED: A Bayesian Approach to Graphical Record Linkage and Deduplication. In S. Kaski & J. Corander (Eds.), *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics* (Vol. 33, pp. 922–930). PMLR. https://proceedings.mlr.press/v33/steorts14.html

Tancredi, A., & Liseo, B. (2011). A hierarchical Bayesian approach to record linkage and population size problems. *The Annals of Applied Statistics*, *5*(2B), 1553–1585. https://doi.org/10.1214/10-aoas447

Tancredi, A., & Liseo, B. (2015). Regression analysis with linked data: problems and possible solutions. *Statistica*, *Vol 75*, No 1 (2015). https://doi.org/10.6092/ISSN.1973-2201/5821

Ted Enamorado, K. I., Ben Fifield. (2023). *fastLink: Fast probabilistic record linkage with missing data* [R package version 0.6.1]. https://CRAN.R-project.org/package=fastLink

Tepping, B. J. (1968). A Model for Optimum Linkage of Records. *Journal of the American Statistical Association*, *63*(324), 1321–1332. https://doi.org/10.1080/01621459.1968.10480930

van der Laan, J. (2025). *reclin2: Record linkage toolkit* [R package version 0.6.0]. https://github.com/djvanderlaan/reclin2

Vo, T. H., Garès, V., Zhang, L., Happe, A., Oger, E., Paquelet, S., & Chauvet, G. (2024). Cox regression with linked data. *Statistics in Medicine*, *43*(2), 296–314. https://doi.org/10.1002/sim.9960

Wang, Z., Ben-David, E., Diao, G., & Slawski, M. (2022). Regression with linked datasets subject to linkage error. *WIREs Computational Statistics*, *14*(4). https://doi.org/10.1002/wics.1570

Watson, D. S., Blesch, K., Kapar, J., & Wright, M. N. (2023). *Adversarial random forests for density estimation and generative modeling*. https://arxiv.org/abs/2205.09435. https://CRAN.R-project.org/package=arf

Williamson, E. J., Walker, A. J., Bhaskaran, K., Bacon, S., Bates, C., Morton, C. E., Curtis, H. J., Mehrkar, A., Evans, D., Inglesby, P., Cockburn, J., McDonald, H. I., MacKenna, B., Tomlinson, L., Douglas, I. J., Rentsch, C. T., Mathur, R., Wong, A. Y. S., Grieve, R., … Goldacre, B. (2020). OpenSAFELY: Factors associated with COVID-19-related hospital death in the linked electronic health records of 17 million adult NHS patients. *MedRxiv*, 430–436.

COLOPHON Typeset with Typst 0.15.0 from the authors' Quarto source. Template jrepstat.typ at commit 87326df. Set in STIX Two Text, STIX Two Math and JetBrains Mono. Built 9 November 2026.