# Package 'AgeTopicModels'

February 28, 2026

**Title** Inferring Age-Dependent Disease Topic from Diagnosis Data

**Version** 0.3.0

**Description** We propose an age-dependent topic modelling (ATM) model, providing a low-rank representation of longitudinal records of hundreds of distinct diseases in large electronic health record data sets. The model assigns to each individual topic weights for several disease topics; each disease topic reflects a set of diseases that tend to co-occur as a function of age, quantified by age-dependent topic loadings for each disease. The model assumes that for each disease diagnosis, a topic is sampled based on the individual's topic weights (which sum to 1 across topics, for a given individual), and a disease is sampled based on the individual's age and the age-dependent topic loadings (which sum to 1 across diseases, for a given topic at a given age). The model generalises the Latent Dirichlet Allocation (LDA) model by allowing topic loadings for each topic to vary with age.
References: Jiang (2023) <doi:10.1038/s41588-023-01522-8>.

**License** MIT + file LICENSE

**Depends** R (>= 3.5)

**Imports** data.table, dplyr, ggplot2, ggrepel, grDevices, gtools, magrittr, pROC, reshape2, rlang, stats, stringr, tibble, tidyr, utils

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Author** Xilin Jiang [aut, cre] (ORCID: <https://orcid.org/0000-0001-6773-9182>)

**Maintainer** Xilin Jiang <jiangxilin1@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-02-28 12:20:02 UTC

# Contents



---

`age_imputation` *imputing missing age if you can't find some of them The function does two stage imputation: i. if the individual has other age label – use the mean, min, or max of other age labels for the missing ones. ii. if the individual has no age label – use the mean, min, max for all the diagnosis codes iii. if there is no age info available for any of this code, we will impute it as the mean of all age codes in the data*

---

## Description

imputing missing age if you can't find some of them The function does two stage imputation: i. if the individual has other age label – use the mean, min, or max of other age labels for the missing ones. ii. if the individual has no age label – use the mean, min, max for all the diagnosis codes iii. if there is no age info available for any of this code, we will impute it as the mean of all age codes in the data

## Usage

```
age_imputation(rec_data_missing_age, method = "mean")
```

### Arguments

| | |
|---|---|
| rec_data_missing_age | a data frame with missing age info |
| method | use one of the three choices "mean", "min", "max" |

### Value

a data frame that is imputed and ready for wrapper_ATM

### Examples

```
rec_data_missing_age <- HES_age_example
 rec_data_missing_age$age_diag[1:10000] <- NA
 rec_data_imputed <- age_imputation(rec_data_missing_age, method= "mean")
 cor(rec_data_imputed$age_diag[1:10000], HES_age_example$age_diag[1:10000])
 rec_data_imputed <- age_imputation(rec_data_missing_age, method= "min")
 cor(rec_data_imputed$age_diag[1:10000], HES_age_example$age_diag[1:10000])
 rec_data_imputed <- age_imputation(rec_data_missing_age, method= "max")
 cor(rec_data_imputed$age_diag[1:10000], HES_age_example$age_diag[1:10000])
```

---

## diseasematrix2longdata

*Disease matrix reformatting for ATM*

---

### Description

Disease matrix reformatting for ATM

### Usage

```
diseasematrix2longdata(disease_matrix)
```

### Arguments

| | |
|---|---|
| disease_matrix | a disease matrix with the first column name "eid", other column are disease names. Disease should be coded as 0,1. |

### Value

a data frame which can be feed into wrapper_ATM

### Examples

```
disease_matrix <- longdata2diseasematrix(HES_age_example)
diseasematrix2longdata(disease_matrix)
```

## disease_info_phecode_icd10 *Disease information linking PheCodes and ICD-10*

### Description

A helper table with disease metadata to support mapping between PheCodes and ICD-10.

### Usage

```
disease_info_phecode_icd10
```

### Format

A data frame/tibble. Common columns include:

**phecode** PheCode as character.

**ICD10** Alternative PheCode column name (if present).

**exclude_range** ancestor PheCode range (character).

**phenotype** Human-readable phenotype/label (character), if available.

**exclude_name** ancestor PheCode name (character).

### Examples

```
head(disease_info_phecode_icd10)
```

## HES_age_example *Example HES diagnosis ages*

### Description

A realistic sized simulated Hospital Episode Statistics (HES) data with participant IDs and ages at diagnosis, used in examples and tests. You would expect the run time of AgeTopicModels on these data is similar to what you face in real life

### Usage

```
HES_age_example
```

### Format

A data frame/tibble with example rows. Typical columns include:

**eid** Participant identifier (integer or character).

**age_diag** Age at diagnosis (numeric).

**diag_icd10** ICD-10 diagnosis code (character).

### Examples

```
head(HES_age_example)
```

---

`HES_icd10_example` *Example HES ICD-10 diagnoses*

---

### Description

A realistic sized simulated HES diagnoses with participant IDs and ICD-10 codes.

### Usage

```
HES_icd10_example
```

### Format

A data frame/tibble with example rows. Typical columns include:

**eid** Participant identifier (integer or character).

**diag_icd10** ICD-10 diagnosis code (character).

**age_diag** ICD-10 diagnosis age point (double).

### Examples

```
head(HES_icd10_example)
```

---

`icd2phecode` *Mapping the disease code from icd10 to phecode*

---

### Description

Mapping the disease code from icd10 to phecode; the mapping are based on https://phewascatalog.org/phecodes; The input if using ICD-10 should be a string f numbers and capital letters only. For example, "I25.1" should be "I251".

### Usage

```
icd2phecode(rec_data)
```

### Arguments

| | |
|---|---|
| `rec_data` | input data which use ICD10 encoding; please refer to the internal example data `HES_icd10_example` for the formatting of the data. |

## Value

a data frame where most entries are mapped from ICD10 code to phecode

## Examples

```
phecode_data <- icd2phecode(HES_icd10_example)
```

---

`loading2weights` *Mapping individuals to fixed topic loadings.*

---

## Description

Mapping individuals to fixed topic loadings.

## Usage

```
loading2weights(data, ds_list = UKB_349_disease, topics = UKB_HES_10topics)
```

## Arguments

| | |
|---|---|
| `data` | the set of diseases, formatted same way as HES_age_example |
| `ds_list` | a list of diseases that correspond to the topic loadings that patients are mapped to formatted as UKB_349_disease; default is set to be UKB_349_disease. |
| `topics` | The topics that are used to map patients. Default is set to be UKB_HES_10topics, which are the inferred topics from 349 Phecodes from the UK Biobank HES data. Details of these topics are available in the paper "Age-dependent topic modelling of comorbidities in UK Biobank identifies disease subtypes with differential genetic risk". |

## Value

a list with two dataframes: the topic_weights dataframe has the first column being the individual id, the other columns are the patient topic weights mapped to the topic loadings; The second dataframe column incidence_weight_sum is eid and the cumulative topic weights across all disease diagnoses.

## Examples

```
set.seed(1)
new_weights <- loading2weights(HES_age_example[1:1000,])
```

## longdata2diseasematrix *Title*

### Description

Title

### Usage

```
longdata2diseasematrix(rec_data)
```

### Arguments

| | |
|---|---|
| rec_data | A diagnosis data frame with three columns; format data as HES_age_example; first column is individual ids (eid), second column is the disease code (diag_icd10); third column is the age at diagnosis (age_diag). Note for each individual, we only keep the first onset of each diseases. Therefore, if there are multiple incidences of the same disease within each individual, the rest will be ignored. |

### Value

a disease matrix with first column being the individual ids, columns follows are diseases with 0,1 coding.

### Examples

```
disease_matrix <- longdata2diseasematrix(HES_age_example)
```

## phecode_icd10 *ICD-10 <-> PheCode mapping*

### Description

Mapping table between ICD-10 codes and PheCodes.

### Usage

```
phecode_icd10
```

### Format

A data frame/tibble. Common columns include:

**ICD10** ICD-10 code (character).

**PheCode** PheCode (character).

**Excl..Phecodes** ancestor PheCode range (character).

**Excl..Phenotypes** ancestor PheCode name (character).

## Examples

```
head(phecode_icd10)
```

---

`phecode_icd10cm` *ICD-10-CM <-> PheCode mapping*

---

## Description

Mapping table between ICD-10-CM codes and PheCodes.

## Usage

```
phecode_icd10cm
```

## Format

A data frame/tibble. Common columns include:

**ICD10** ICD-10-CM code (character).

**phecode** PheCode (character).

**exclude_range** ancestor PheCode range (character).

**exclude_name** ancestor PheCode name (character).

## Examples

```
head(phecode_icd10cm)
```

---

`plot_age_topics` *Title plot the topic loadings across age.*

---

## Description

Title plot the topic loadings across age.

## Usage

```
plot_age_topics(
  disease_names,
  trajs,
  plot_title = "",
  start_age = 30,
  top_ds = 10
)
```

### Arguments

| | |
|---|---|
| disease_names | the list of disease names, ordered as the topic. |
| trajs | one disease topic, which should be a matrix of age-by-disease. |
| plot_title | the title of the figure. |
| start_age | starting age of the matrix, default 30. |
| top_ds | How many disease to show, default is 10. This will filter the disease by the average topic laodings across age and pick the top. |

### Value

a ggplot object of the topic loading.

### Examples

```
disease_list <- UKB_349_disease %>%
dplyr::left_join(disease_info_phecode_icd10, by = c("diag_icd10"="phecode" )) %>%
dplyr::pull(phenotype)
topic_id <- 1 # plot the first topic
plot_age_topics(disease_names = disease_list,
        trajs = UKB_HES_10topics[30:80,,topic_id],
        plot_title = paste0("topic ", topic_id),
        top_ds = 7)
```

---

## plot_lfa_topics *Title plot topic loadings for LFA.*

---

### Description

Title plot topic loadings for LFA.

### Usage

```
plot_lfa_topics(disease_names, beta, plot_title = "")
```

### Arguments

| | |
|---|---|
| disease_names | the list of disease names, ordered as the topic. |
| beta | disease topics, which should be a matrix of K-by-disease. |
| plot_title | the title of the figure. |

### Value

a ggplot object of the topic loading.

## Examples

```
disease_list <- UKB_349_disease$diag_icd10[1:50]
topics <- matrix(rnorm(10*length(UKB_349_disease)), nrow = length(UKB_349_disease), ncol = 10)
plot_lfa_topics(disease_names = disease_list,
        beta = topics,
        plot_title = "Example noisy topics presentation")
```

---

`prediction_OR` *Title Compute prediction odds ratio for a testing data set using pre-training ATM topic loading. Note only diseases listed in the ds_list will be used. The prediction odds ratio is the odds predicted by ATM versus a naive prediction using disease probability.*

---

## Description

Title Compute prediction odds ratio for a testing data set using pre-training ATM topic loading. Note only diseases listed in the ds_list will be used. The prediction odds ratio is the odds predicted by ATM versus a naive prediction using disease probability.

## Usage

```
prediction_OR(testing_data, ds_list, topic_loadings, max_predict = NULL)
```

## Arguments

| | |
|---|---|
| `testing_data` | A data set of the same format as HES_age_example; Note: for cross-validation, split the training and testing based on individuals (eid) instead of diagnosis to avoid using training data for testing. Note the test data that has diagnosis age outside the topic loading is disgarded, as we don't recommend extrapolate topic loadings outside the training data. |
| `ds_list` | The order of disease code that appears in the topic loadings. This is a required input as the testing data could miss some of the records. The first column should be the disease code, second column being the occurrence (to serve as the baseline for prediction odds ratio). See AgeTopicModels::UKB_349_disease as an example. |
| `topic_loadings` | A three dimension array of topic loading in the format of AgeTopicModels::UKB_HES_10topics; |
| `max_predict` | The logic of prediction is using 1,..N-1 records to predict the Nth diagnosis; we perform this prediction in turn, starting from using first disease to predict sencond.... for the max_predict^th disease, we will just predict all diseases afterwards, using only 1,..(max_predict-1) diseseas to learn the topic weights; default is set to be 11 (using 1,...10 disease to predict). |

## Value

The returned object has four components: OR_top1, OR_top2, OR_top5 is the prediction odds ratio using top 1%, top 2%, or top 5% of ATM predicted diseases as the target set; the fourth component prediction_precision is as list, with first element saves the prediction probability for 1%, 2%, 5% and 10%; additional variables saves the percentile of target disease in the ATM predicted set; for example 0.03 means the target disease ranked at 3% of the diseases ordered by ATM predicted probability.

## Examples

```
set.seed(1)
testing_data <- HES_age_example %>% dplyr::slice(1:1000)
new_output <- prediction_OR(testing_data, ds_list = UKB_349_disease,
                  topic_loadings =  UKB_HES_10topics, max_predict = 5)
```

---

# short_icd10 *Short labels (at most first for letters/digits) for ICD-10 codes*

---

## Description

A lookup table mapping ICD-10 codes to concise human-readable labels.

## Usage

```
short_icd10
```

## Format

A data frame/tibble. Common columns include:

**ICD10** ICD-10 code (character).

**parent_phecode** phecode of parent node (character).

**Excl..Phecodes** ancestor PheCode range (character).

**Excl..Phenotypes** ancestor PheCode name (character).

**occ** number of distinct patient in UKB

## Examples

```
head(short_icd10)
```

## short_icd10cm *Short labels (at most first for letters/digits) for ICD-10-CM codes*

### Description

A lookup table mapping ICD-10-CM codes to concise human-readable labels.

### Usage

```
short_icd10cm
```

### Format

A data frame/tibble. Common columns include:

**ICD10** ICD-10 code (character).

**parent_phecode** phecode of parent node (character).

**exclude_range** ancestor PheCode range (character).

**exclude_name** ancestor PheCode name (character).

**occ** number of distinct patient in UKB

### Examples

```
head(short_icd10cm)
```

## simulate_genetic_disease_from_topic *Simulate genetic-disease-topic structure (step 2)*

### Description

Second step of the two-step simulation. Consumes outputs from `simulate_topics()` and generates disease outcomes under several genetic/topic-effect configurations.

### Usage

```
simulate_genetic_disease_from_topic(
  para,
  genetics_population,
  causal_disease,
  disease_number,
  ds_per_idv = 6.1,
  itr_effect = 0,
  topic2disease = 2,
  v2t = 20,
  liability_thre = 0.8
)
```

## Arguments

| | |
|---|---|
| `para` | Simulated topic parameters; the first element returned by `simulate_topics()`. |
| `genetics_population` | Simulated genotypes; the second element returned by `simulate_topics()`. |
| `causal_disease` | Simulated causal disease; the third element returned by `simulate_topics()`. |
| `disease_number` | Number of additional diseases to simulate from the topic. The total number of diseases will be `disease_number + 5`. |
| `ds_per_idv` | Mean number of diseases per individual (default `6.1`, as observed in UKB). |
| `itr_effect` | Interaction effect size to simulate (default `0`). |
| `topic2disease` | Topic-to-disease effect size (default `2`). |
| `v2t` | Number of variants that affect topic 1 (must match the value used in `simulate_topics()`). |
| `liability_thre` | Liability threshold for simulating disease: the proportion set to *healthy*. For example, `0.8` means the top 20% of liability are set to *diseased* (default `0.8`). |

## Details

Five configurations across three SNP sets:

1. **SNP -> disease -> topic**: SNP IDs 1-20; disease ID `para$D + 1`; topic ID 1.
2. **SNP * topic -> disease**: SNP IDs 41-60; disease ID `para$D + 2`; topic ID 1.
3. **SNP -> topic -> disease; SNP -> disease**: SNP IDs 21-(20 + v2t); disease ID `para$D + 3`; topic ID 1.
4. **SNP -> topic -> disease; SNP + SNP^2 -> disease**: SNP IDs 21-(20 + v2t); disease ID `para$D + 4`; topic ID 1.
5. **SNP -> topic + topic^2 -> disease; SNP -> disease**: SNP IDs 21-(20 + v2t); disease ID `para$D + 5`; topic ID 1.

## Value

A list with four elements:

- `rec_data`: Simulated disease records (primary output).
- `ds_list`: Auxiliary data objects used in the simulation.
- `interact_disease`: Binary disease outcomes for configuration 2.
- `pleiotropy_disease`: Binary disease outcomes for configuration 3.

## See Also

`simulate_topics()`

## Examples

```
set.seed(1)
# Minimal, fast example
rslts <- simulate_topics(topic_number = 2, pop_sz = 1000,
                         disease2topic = 0.1, v2t = 20)
para_sim           <- rslts[[1]]
genetics_population <- rslts[[2]]
causal_disease     <- rslts[[3]]

reslt_ds <- simulate_genetic_disease_from_topic(
  para_sim, genetics_population, causal_disease,
  disease_number = 20, itr_effect = 1,
  topic2disease = 2, v2t = 20
)
rec_data <- reslt_ds[[1]]
```

---

# simulate_topics *Simulate genetic-disease-topic structure (step 1)*

---

## Description

First step of a two-step procedure to simulate a genetic-disease-topic structure. In this step, all genetic effects act on topic 1.

## Usage

```
simulate_topics(
  topic_number,
  num_snp = 100,
  pop_sz = 10000,
  disease2topic = 0,
  v2t = 20,
  snp2t = 0.04,
  snp2d = 0.15,
  liability_thre = 0.8
)
```

## Arguments

| | |
|---|---|
| `topic_number` | Number of topics to simulate. |
| `num_snp` | Total number of SNPs (default 100; must be >= 60). |
| `pop_sz` | Number of individuals (default 10,000). |
| `disease2topic` | Disease-to-topic effect size (default 0). |
| `v2t` | Number of variants affecting topic 1 (0-20; default 20). |
| `snp2t` | SNP-to-topic effect size (default 0.04; informed by UKB). |
| `snp2d` | SNP-to-disease effect size (default 0.15). |
| `liability_thre` | Liability threshold: proportion set to *healthy*. For example, `0.8` means the top 20% of liability are set to *diseased* (default 0.8). |

## Details

Five configurations across three SNP sets:

1. **SNP -> disease -> topic**: SNP IDs 1-20; disease ID `para$D + 1`; topic ID 1.
2. **SNP * topic -> disease**: SNP IDs 41-60; disease ID `para$D + 2`; topic ID 1.
3. **SNP -> topic -> disease; SNP -> disease**: SNP IDs 21-(20 + v2t); disease ID `para$D + 3`; topic ID 1.
4. **SNP -> topic -> disease; SNP + SNP^2 -> disease**: SNP IDs 21-(20 + v2t); disease ID `para$D + 4`; topic ID 1.
5. **SNP -> topic + topic^2 -> disease; SNP -> disease**: SNP IDs 21-(20 + v2t); disease ID `para$D + 5`; topic ID 1.

## Value

A list of length 3:

- `para`: Topic parameters.
- `genetics_population`: Simulated genotype matrix.
- `causal_disease`: One simulated binary disease caused by loading on topic 1.

## See Also

`simulate_genetic_disease_from_topic()`

## Examples

```
set.seed(1)
disease2topic <- 0
v2t_small <- 20

# Step 1: simulate topics (fast)
rslts <- simulate_topics(
  topic_number = 2, pop_sz = 1000,
  disease2topic = disease2topic, v2t = v2t_small
)
para_sim            <- rslts[[1]]
genetics_population <- rslts[[2]]
causal_disease      <- rslts[[3]]

# Step 2 (optional): generate diseases from topics
reslt_ds <- simulate_genetic_disease_from_topic(
  para_sim, genetics_population, causal_disease,
  disease_number = 20, itr_effect = 1,
  topic2disease = 2, v2t = 20
)
rec_data <- reslt_ds[[1]]
```

## SNOMED_ICD10CM *SNOMED <-> ICD-10(-CM) mapping (excerpt)*

### Description

A small mapping table used by functions such as `icd2phecode`

### Usage

```
SNOMED_ICD10CM
```

### Format

A data frame/tibble. Common columns include:

**SNOMED** SNOMED CT concept identifier (character).

**ICD10** ICD-10 code (character), and/or

**ICD10_name** ICD-10-CM code (character).

**SNOMED_description** SNOMED readable explanation

**occ** ICD10 occurence in UKB

### Examples

```
head(SNOMED_ICD10CM)
```

## UKB_349_disease *List of 349 UK Biobank diseases (example)*

### Description

A character vector or table listing the set of disease phenotypes used in examples/vignettes.

### Usage

```
UKB_349_disease
```

### Format

A data frame/tibble containing disease identifiers/names. Columns include:

**diag_icd10** Phecode (character).

**occ** number of distinct patient in UKB

@examples head(UKB_349_disease)

## UKB_HES_10topics *Example topic model output (10 topics, UKB HES)*

### Description

An illustrative result object/table from a 10-topic model fit to UKB HES-like data; used for examples, plotting, and tests.

### Usage

```
UKB_HES_10topics
```

### Format

An array for UKB topic loadings. Dimention is age, disease, topics. the ordering of disease is the same as UKB_349_disease.

### Examples

```
head(UKB_HES_10topics)
```

## wrapper_ATM *Run ATM on diagnosis data.*

### Description

Run ATM on diagnosis data to infer topic loadings and topic weights. Note one run of ATM on 100K individuals would take ~30min (defualt is 5 runs and pick the best fit); if the data set is small and the goal is to infer patient-level topic weights (i.e. assign comorbidity profiles to individuals based on the disedases), please use loading2weights.

### Usage

```
wrapper_ATM(
  rec_data,
  topic_num = 10,
  degree_free_num = 3,
  CVB_num = 5,
  save_data = FALSE
)
```

### Arguments

| | |
|---|---|
| rec_data | A diagnosis data frame with three columns; format data as HES_age_example; first column is individual ids (eid), second column is the disease code (diag_icd10); third column is the age at diagnosis (age_diag). Note for each individual, we only keep the first onset of each diseases. Therefore, if there are multiple incidences of the same disease within each individual, the rest will be ignored. If there is no age variation in the third column, LDA (no age information) will be run instead of ATM. |
| topic_num | Number of topics to infer. Default is 10 but we highly recommend running multiple choices of this number. |
| degree_free_num | control the parametric for of topic loadings: Degrees of freedom (d.f.) from 2 to 7 represent linear, quadratic polynomial, cubic polynomial, spline with one knot, spline with two knots, and spline with three knots. Default is set to 3. |
| CVB_num | Number of runs with random initialization. The final output will be the run with highest ELBO value. |
| save_data | A flag which determine whether full model data will be saved. If TRUE, a Results/ folder will be created and full model data will be saved. Default is set to be FALSE. |

### Value

Return a list object with topic_loadings (of the best run), topic_weights (of the best run), ELBO_convergence (ELBO until convergence), patient_list (list of eid which correspond to rows of topic_weights), ds_list (gives the ordering of diseases in the topic_loadings object), disease_number (number of total diseases), patient_number(total number of patients), topic_number (total number of topic), topic_configuration (control the parametric for of topic loadings: Degrees of freedom (d.f.) from 2 to 7 represent linear, quadratic polynomial, cubic polynomial, spline with one knot, spline with two knots, and spline with three knots. Default is set to 3.), multiple_run_ELBO_compare (ELBO of each runs).

### Examples

```
# minimal, always-run example (tiny data/iterations)
set.seed(1)
inference_results <- wrapper_ATM(HES_age_example[1:500,], topic_num = 2, CVB_num = 1)
```

---

## wrapper_LFA *Run LFA on diagnosis data.*

---

### Description

Run LFA on diagnosis data to infer topic loadings and topic weights. Note one run of LFA on 100K individuals would take ~30min (defualt is 5 runs and pick the best fit); if the data set is small and the goal is to infer patient-level topic weights (i.e. assign comorbidity profiles to individuals based on the disedases), please use loading2weights.

## Usage

```
wrapper_LFA(
  rec_data,
  topic_num,
  CVB_num = 5,
  save_data = FALSE,
  beta_prior_flag = FALSE,
  topic_weight_prior = NULL
)
```

## Arguments

| | |
|---|---|
| rec_data | A diagnosis data frame with three columns; format data as HES_age_example; first column is individual ids, second column is the disease code; third column is the age at diagnosis. Note for each individual, we only keep the first onset of each diseases. Therefore, if there are multiple incidences of the same disease within each individual, the rest will be ignored. |
| topic_num | Number of topics to infer. |
| CVB_num | Number of runs with random initialization. The final output will be the run with highest ELBO value. |
| save_data | A flag which determine whether full model data will be saved. If TRUE, a Results/ folder will be created and full model data will be saved. Default is set to be FALSE. |
| beta_prior_flag | A flag if true, will use a beta prior on the topic loading. Default is set to be FALSE. |
| topic_weight_prior | prior of individual topic weights, default is set to be a vector of one (non-informative) |

## Value

Return a list object with topic_loadings (of the best run), topic_weights (of the best run), ELBO_convergence (ELBO until convergence), patient_list (list of eid which correspond to rows of topic_weights), ds_list (gives the ordering of diseases in the topic_loadings object), disease_number (number of total diseases), patient_number(total number of patients), topic_number (total number of topic), ,multiple_run_ELBO_compare (ELBO of each runs).

## Examples

```
  HES_age_small_sample <- HES_age_example[1:100,]
inference_results <- wrapper_LFA(HES_age_small_sample, topic_num = 3, CVB_num = 1)
```

# Index