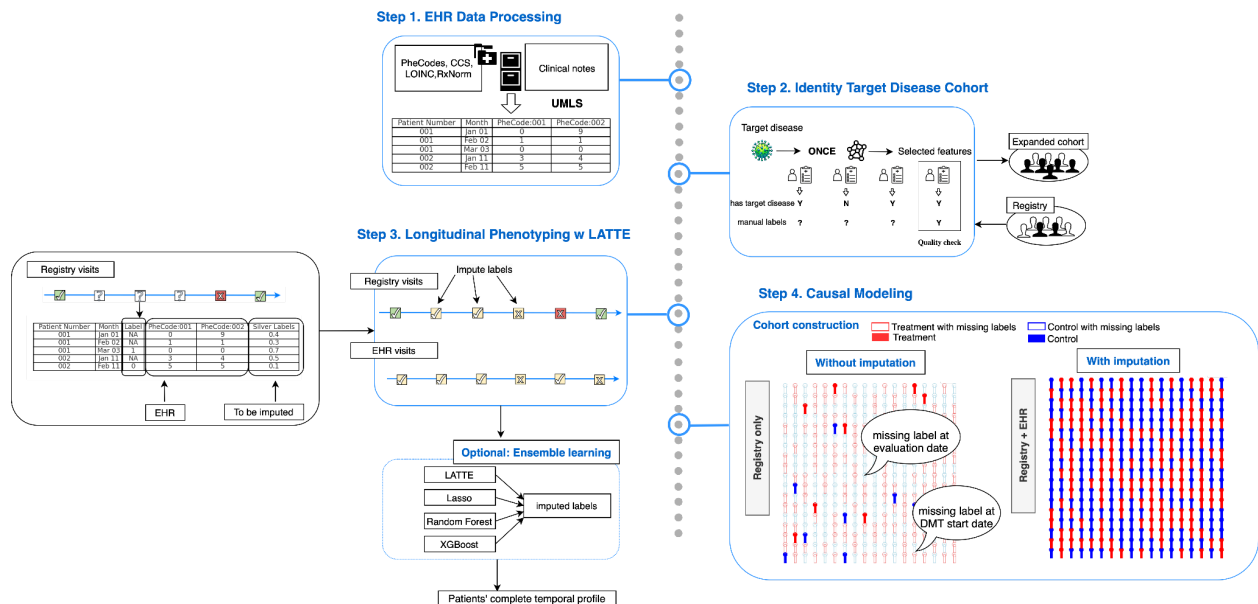


Protocol for deriving longitudinal Clinical Outcomes from electronic health records



Summary

Harnessing electronic health records (EHRs) as real-world data (RWD) for clinical research demands the curation of high quality clinical outcomes. We present a protocol to derive longitudinal clinical outcomes from structured and unstructured EHR data based on the Label efficient incident phEnotyping (LATTE) algorithm¹. We describe the steps to 1) EHR Data Processing; 2) establish an EHR cohort for the disease population; 3) prepare data for the LATTE algorithm; 4) derive the longitudinal outcome using LATTE; and 5) using LATTE derived outcome for downstream analyses.


STUDY OVERVIEW

The current study aims to conduct causal modeling using EHR-derived outcomes. As illustrated in the flowchart above, the study is divided into several key steps.

- The first step is to create an EHR data mart that includes both codified and narrative data for all patients. In this step, raw EHR concepts are aggregated and standardized to common ontologies, and NLP mentions of clinical terms are extracted and mapped to the Concept Unique Identifiers (CUIs) in the Unified Medical Language System (UMLS). These data are then aggregated into monthly counts for each patient.
- The second step involves identifying the target disease cohort through phenotyping. Using the diagnostic code of the target disease, relevant EHR features are selected via a feature selection algorithm called ONCE. The counts of the selected EHR features are then aggregated, and a weakly supervised algorithm is employed to classify patients' disease status without using any gold-standard labels. A validation step is subsequently performed using gold-standard labels for quality checking. The output from this step is an expanded target disease cohort, including patients from both the registry and EHR.
- The third step involves performing longitudinal phenotyping using the Label Efficient Incident Phenotyping (LATTE) algorithm. Proper data preparation is crucial before running the algorithm. Specifically, both the clinical outcomes and patient-level counts of selected EHR features must be calculated or aggregated for each assessment period, according to frequency of assessment of the study design. Additionally, to enable LATTE to learn from both labeled patients in the registry and unlabeled non-registry patients, silver-standard labels need to be imputed beforehand for all patients for each assessment period. By inputting these well-prepared data into LATTE, the algorithm produces imputed clinical outcomes for both registry and non-registry patients. While these outputs are sufficient to move on to the next step, it is recommended, though optional, to perform ensemble learning using the imputed scores from LATTE combined with other standard supervised methods such as LASSO, Random Forest, and XGBoost. These latter models can be trained using the gold-standard outcomes.
- Using the output data from the third step, the fourth step involves performing causal modeling with the imputed outcomes. It is important to ensure that these imputed outcomes are well-calibrated to match the gold-standard labels. The doubly robust framework can be used for causal modeling, which combines an outcome model with a propensity score model to control for confounding variables. This framework remains reliable even if either model is misspecified. Due to missing outcomes at both the DMT initialization and the evaluation end date, without imputation, causal modeling would only be possible for a small subset of labeled registry patients. This limitation typically results in wide confidence intervals, making the analysis inconclusive. By using imputed outcomes, we can effectively increase the sample size, leading to narrower confidence intervals and more informative results.


BEFORE YOU BEGIN (to be moved to a supplementary file)

Study specification

 Timing: 2 weeks

1. Specify the target disease, clinical outcomes and temporal window of interest. A preliminary filter for patients potentially with the target disease should be defined according to codified identifiers of diagnosis. An interactive web-based visualization for the mapping from disease to International Classification of Diseases (versions 9 and 10) based on PheWAS catalog² is available at <https://shiny.parse-health.org/phecodemap/>. For numeric or ordinal target clinical outcomes, review their role in existing clinical studies for dichotomization into binary outcomes. The study team should specify the temporal window for determining: a) the baseline time (initial diagnosis, progression to a certain stage, or initiation of specific interventions); b) the maximal follow-up time from baseline; c) the typical frequency of assessment on the clinical outcomes in clinical practice (e.g. every 3 or 6 months).
2. Summarize the source of information for target clinical outcomes. Identification of likely sources of information for the target clinical outcomes ensures the request of such key information from the vast EHR database. Practicing clinical experts should be consulted to provide the typical documentation pattern of relevant information. Attention should be paid to non-codified EHR sources, such as numeric lab test results, assessment documented in specific types of notes (e.g. radiology reports, pathology reports) or study registries within the healthcare system.


Institutional permissions

 Timing: 2-4 weeks

EHRs contain protected health information (PHI) that is regulated under laws. Use of EHRs for research requires institutional permissions in most cases.

3. Obtain institutional approval for the use of EHRs in the study. Develop a study protocol with focus on the access and protection plan of participants' PHI. Submit the study protocol to the Institutional Review Board (IRB) and revise as instructed. Complete required training for all study team members with access to EHRs.

Installation

 Timing: <1 Hours


4. Create the Python (version 3.7.4) Conda environment with required packages (tensorflow version 2.3.0, pandas, numpy, click, scipy, scikit-learn, pyhere).

```
module load gcc/6.2.0 R/4.0.1
module load conda2/4.2.13
conda create -n env_LATTE python=3.7.4
source activate env_LATTE
pip install tensorflow==2.3.0
pip install pandas
pip install numpy
pip install click
pip install scipy
pip install scikit-learn # changed this line of code
pip install pyhere
```

Download the LATTE source codes from <https://github.com/celehs/LATTE>. For other machine learning methods to be ensemble with LATTE, refer to their documentation for installation.

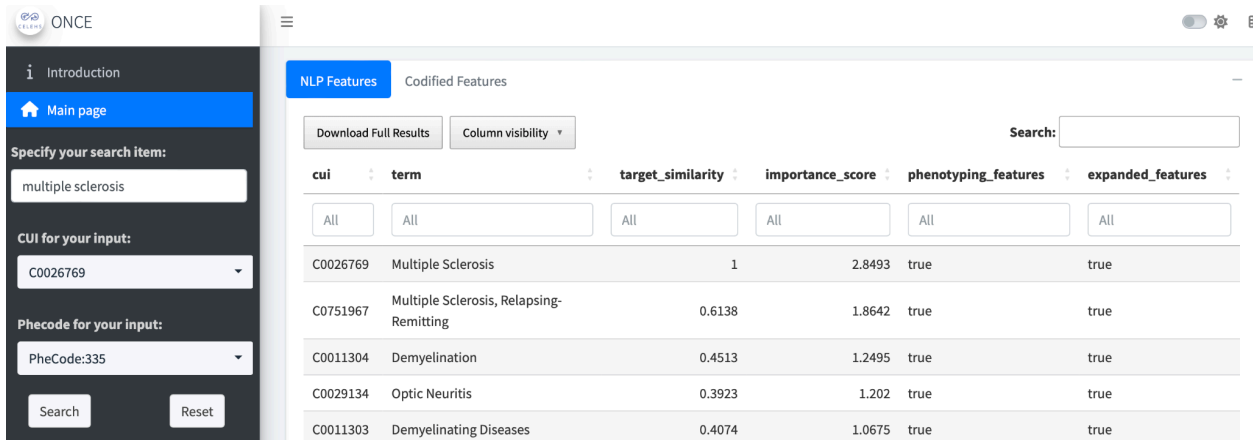
STEP-BY-STEP METHOD DETAILS

EHR Data Processing

 Timing: 2 weeks

To study the target clinical outcome for a disease of interest, the first step is to create an EHR data mart. EHR data consists of longitudinal patient visits, with each visit recording the observations of both structured EHR concepts (codes for diagnosis, procedures, medication prescriptions, laboratory test orders, and results), and unstructured narrative clinical notes. Since clinical outcomes such as disease activity and relevant documentation are largely embedded in clinical notes, it is important to include clinical notes in the datamart. The initial datamart should include codified and narrative data for all patients according to the preliminary disease filter defined during study design (at least 1 or 2 diagnostic codes of the disease).

5. **Compile Variable Dictionary for Study:** Since the number of total variables (codified and narrative) in the EHRs is monstrous, an effective filtering strategy is necessary for controlling storage and computation cost, as well as protecting patient privacy according to the standard for minimum necessary uses. Existing studies on knowledge networks have provided the tools for evaluating the relevance of EHR variables to disease of interest and target clinical outcomes.³⁻⁵ These tools provide linkage of a disease phenotype to related EHR codes and clinical terms. For example concepts important for rheumatoid arthritis may include ICD codes for rheumatoid arthritis, NLP mentions of rheumatoid arthritis, methotrexate prescription, CRP laboratory test, as well as NLP mentions of joint pain and erosion. For the knowledge network developed by Xiong et al.,⁵ a shiny app web service is available at <https://shiny.parse-health.org/ONCE/>. Users can input the names of the disease of interest at “Specify Your Search Item” and download the variable dictionaries by clicking “Download Full Results” under “NLP Features” and “Codified Features”.

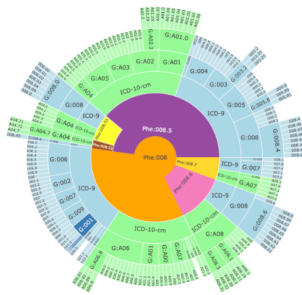


cui	term	target_similarity	importance_score	phenotyping_features	expanded_features
C0026769	Multiple Sclerosis	1	2.8493	true	true
C0751967	Multiple Sclerosis, Relapsing-Remitting	0.6138	1.8642	true	true
C0011304	Demyelination	0.4513	1.2495	true	true
C0029134	Optic Neuritis	0.3923	1.202	true	true
C0011303	Demyelinating Diseases	0.4074	1.0675	true	true

6. **Aggregate Codified Data:** In the variable dictionary for codified features, raw EHR concepts are rolled up to higher level concepts according to common ontology as in previous studies: Diagnosis (ICD) codes are grouped into the established Phenome-wide association studies (PheWAS) catalogs (PheCodes);² Procedure codes are grouped into categories according to the Clinical Classifications Software for Services and Procedures (CCS), medications are mapped to ingredient level RxNorm codes, and laboratory tests are

ICD, LOINC, and RxNORM Hierarchies

The Phecode to ICD map showcases the knowledge graph mapping the relationship between International Classification of Diseases (ICD) coding system and Phecode system.



LOINC (Logical Observation Identifiers Names and Codes) is a standardized coding system used globally to report laboratory and other clinical observations. The LOINC database contains over 71,000 terms, each described by a six-part structure that includes component, property, time aspect, system, scale, and method.



This tool illustrates the hierarchical structure of medications and their variants in brand and dosage form denoted in RxNorm systems. It also showcases the knowledge graph mapping medicine in RxNorm code to other common medicine coding systems, such as ATC and VA.



Both codified and NLP data are aggregated longitudinally into monthly counts for each patient, starting from the baseline time determined in study design.

-ala	AB00-ala	C0002563	acronym	DB00855
-mea	AB00-mea	C0010648	acronym	DB00847
-mpt	AB00-mpt	C0086584	acronym	DB00765
-tea	AB00-tea	C1455006	acronym	CDR0000763148
0.2g	AB000.2g	C0470519	acronym	wx07z
0.3g	AB000.3g	C0733017	acronym	wx07E
0.5g	AB000.5g	C0470520	acronym	wx080
0/5	AB0000/5	C0444499	acronym	X80bv
1 h	AB0001 h	C0700308	acronym	C-10101

```
0|0|time|placeholder|
The patient is hemodynamically stable
[report_end]
1|1|time|placeholder|
The patient is hemodynamically unstable.
[report_end]
2|2|time|placeholder|
The patient is in pain.
[report_end]
```

NILE output

```

EMPI|Report_Number|Report_Date_Time|CUIs
0|0|time|C0002962:Y,C2024883:Y,C2926611:Y
1|1|time|C0439064:Y,AB000PES:Y
2|2|time|AB000CAD:Y
3|3|time|AB00LESS:Y,C0239110:Y,C0718338:Y,C2004580:Y,C3665546:Y,C3669039:Y,C4553014:Y,C2987125:Y
4|4|time|C2987125:Y,C0239110:Y,C0718338:Y,C2004580:Y,C3665546:Y,C3669039:Y,C4553014:Y,C1704322:Y,C1961028:Y
5|5|time|C0233407:Y,C0004093:Y,C3714552:Y
6|6|time|C0262926:Y,C0600260:Y
7|7|time|C0205307:Y,C0558145:Y,C1550457:Y,C1550469:Y,C1551394:Y,C1553386:Y,C1553399:Y,C1553402:Y,C1553406:Y,C1704701:Y,C1873497:Y,C2698497:Y,C0024109:Y

```


Code block

```

java -Xmx8000m -Dfile.encoding=UTF-8 -cp ".\NLP_linux.jar:.\commons-net-3.6.jar:.\commons-lang3-3.5.jar:.\commons-collections4-4.1.jar" NLP_LOCAL.NILE_linux

```

Identify target disease cohort via phenotyping

 Timing: 2 weeks

Identification of patients with target disease has been thoroughly studied in phenotyping literature.⁷ Scalable methods include semi-supervised algorithms such as the PheCAP detailed in Zhang et al.⁸ and weakly supervised methods such as the PheNorm⁹, Multimodal Automated Phenotyping (MAP)¹⁰ and the knowledge-driven online multi-modal phenotyping system.⁵ We recommend using weakly supervised methods due to their ability to train these algorithms without the need for any gold standard labels although labels are needed for training the algorithms.

8. **Assemble Patient Level Features for Phenotyping to Classify Disease Status:** the most important features needed for classifying disease status include the relevant diagnosis codes (ICD) and mention of clinical terms for the disease in notes (NLP) as well as a healthcare utilization measure such as the number of visit days to the health system (Visit). Weakly supervised phenotyping can be performed using these three variables.^{9,10} When the weakly supervised phenotyping has poor performance in validation (Step 7), other phenotyping methods using additional features from Steps 1-3 should be considered. See Yang et al.⁷ for a summary of phenotyping methods.

ID	ICD	NLP	Visit
1	1	0	13
2	5	2	21
3	1	3	5
...
500	20	6	17

Input data for MAP.

9. **Perform weakly supervised phenotyping:** most of these algorithms require specifications of surrogate features which can be treated as silver standard labels on the disease of interest. We recommend using diagnostic code of the disease and NLP mention of the disease concept as the two key surrogates if clinical text is included as part of the study. With specified surrogate features along with healthcare utilization as well as supporting features, one may deploy a weakly supervised algorithm to classify the disease status. The MAP

algorithm only requires the key surrogates and the healthcare utilization without additional features. These algorithms typically output a probability of having the disease for each patient and one may choose a threshold to classify a patient as having the condition if the probability exceeds a threshold.

```
install.packages("MAP")
library(MAP)

MAP.input = read.csv("MAPinput.csv")
MAP.fit = MAP(mat = Matrix(data = cbind(MAP.input$ICD, # Counts of diagnosis code
                                         MAP.input$NLP)), # Counts of mention in notes
              note = Matrix(MAP.input$Visit, ncol=1)) # Counts of all visits
MAP.fit$score # The estimated probability of disease status
```

Running MAP in R.

```
library(MAP)
```

```
## Loading required package: flexmix
```

```
## Loading required package: lattice
```

```
## Loading required package: Matrix
```

simulate data to test the algorithm

```
set.seed(123)
n = 400
ICD = c(rpois(n/4,10), rpois(n/4,1), rep(0,n/2) )
NLP = c(rpois(n/4,10), rpois(n/4,1), rep(0,n/2) )
mat = Matrix(data=cbind(ICD,NLP),sparse = TRUE)
note = Matrix(rpois(n,10)+5,ncol=1,sparse = TRUE)
res = MAP(mat = mat, note=note)
```

```
## #####
## MAP only considers pateints who have note count data and
##      at least one nonmissing variable!
## ####
## Here is a summary of the input data:
## Total number of patients: 400
##   ICD NLP note Freq
## 1 YES YES  YES  400
## ####
```

```
head(res$scores)
```




```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##
## [1,] 0.9174454
## [2,] 0.9214842
## [3,] 0.9222160
## [4,] 0.9219585
## [5,] 0.9205031
## [6,] 0.9221740
```

```
res$cut.MAP
```

```
## [1] 0.2824703
```

10. **Create gold-standard labels on target disease status for validation.** To ensure the quality of the target disease cohort, the study team should include a clinically qualified annotator to manually review a subset of the patients and label their onset status of the target disease. At least 50 patients should be randomly sampled to validate the phenotyping methods. More labels should be created for alternative phenotyping methods involving model-training with gold-standard labels.
11. **Perform algorithm validation and establish the disease cohort:** With gold standard labels, one may calculate the area under the Receiver Operating Characteristic Curve (AUC) along with sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV) across various threshold values. Alternative phenotyping methods should be considered if $AUC < 0.800$. We recommend choosing threshold values that maximize sensitivity while maintaining a PPV higher than a desired level such as 90%.

Prepare data for LATTE

 Timing: 4 weeks

12. **Obtain gold-standard clinical outcomes for a subset of patients:** gold standard labels on a small subset of patients are needed for downstream modeling. This can be achieved by either linking to disease registries where clinical outcomes have been collected or performing manual chart review. Labels should be curated for 200 or more patients to enable reliable training and validation of the algorithm.

ID	Date	Outcome
1	YYYY-MM-DD	1
1	YYYY-MM-DD	0
3	YYYY-MM-DD	1
...
200	YYYY-MM-DD	0

Gold-standard label for target clinical outcome.

13. **Knowledge Guided Feature Selection and Aggregation:** Due to the limited availability of gold standard labels and the complexity of modeling longitudinal clinical outcomes, it is desirable to perform further feature selection according to knowledge sources. Moreover,

many related EHR variables are highly correlated, which motivates dimension reduction according to their correlation structure. To perform knowledge guided feature selection and aggregation, we again utilize the large scale knowledge networks. The ONCE variable dictionary compiled in Step 1 has an “importance score” metric, based on which we suggest further filter the variables at >0.1 threshold. For selected EHR variables, extract the semantic embeddings from the corresponding knowledge network to guide aggregation in LATTE. One may augment these knowledge sources with domain expert input to finalize the input features.

	0.1	0.2	0.3	...	0.200
EHR Var:1	-0.46159	0.217159	-0.14798	...	-0.29897
EHR Var:2	-0.4147	0.061663	0.525976	...	0.207914
...
EHR Var:50	-0.50107	0.084842	0.667013	...	-0.18562

Format for embedding file.

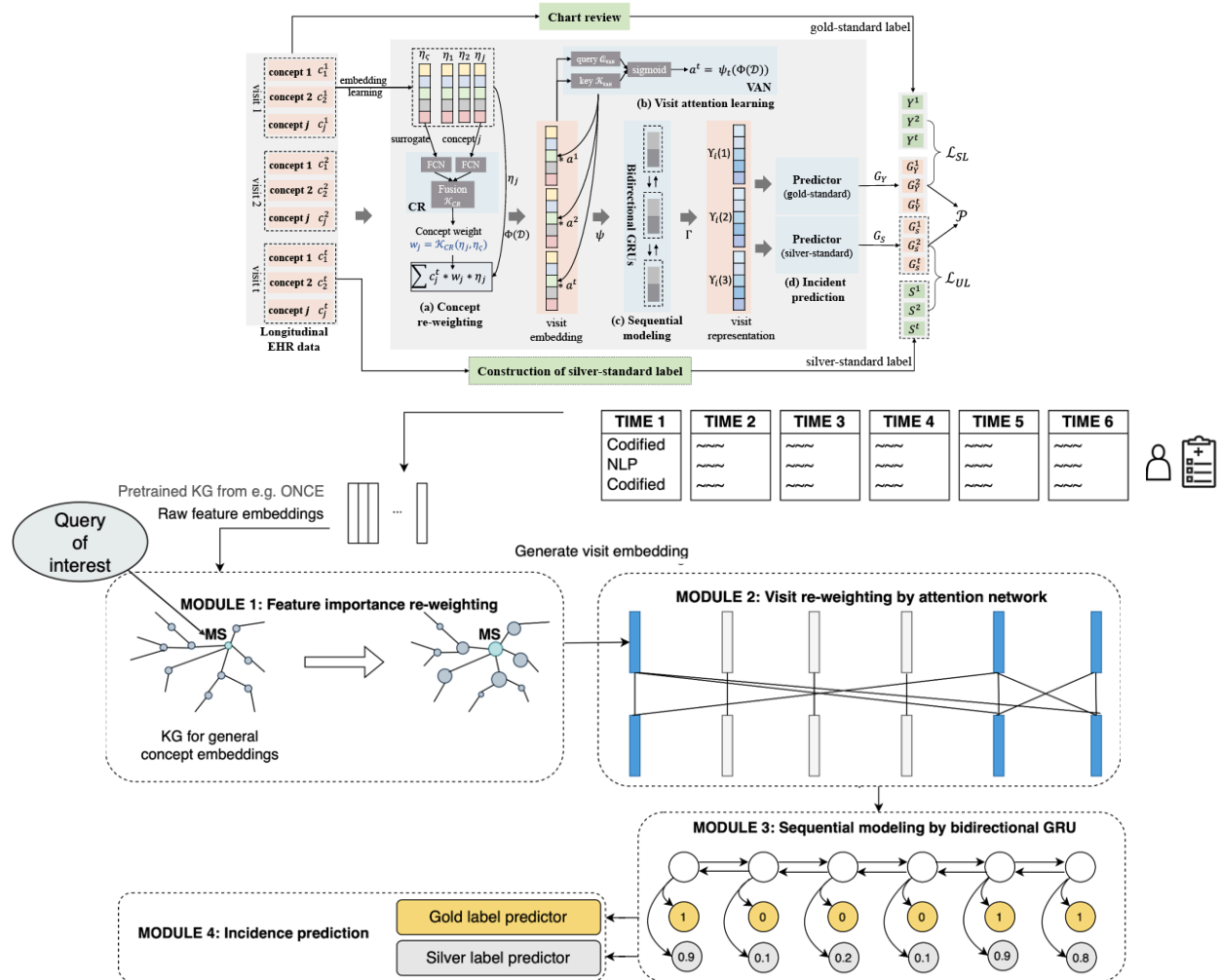
14. **Construction of the Silver Standard Labels:** A key ingredient for LATTE’s label efficient trait is the use of predictive silver standard labels for target clinical outcomes to assist training with large unlabeled data. Silver standard labels can be constructed for each patient-period (according to frequency of assessment determined in Step 1) in a few ways. First, the counts of diagnosis codes for target diseases naturally reflect the disease activities.^{1,11} Second, specialized NLP or artificial intelligence (AI) tools can partially identify unstructured documentations of clinical outcomes from narrative notes.^{12–14} Third, domain experts can propose other likely indicative EHR variables such as specialized diagnosis or treatment following poor target clinical outcome. Aggregation and preliminary calibration of these silver labels can be constructed by fitting a logistic regression using a few training labels.
15. **Formatting the Input Data:** To run LATTE, the input data needs to be formatted into several files (with corresponding filenames in example codes): i) training data (train.csv), ii) testing data (test.csv), iii) embedding (embedding.csv), iv) and v) variable lists (kn_var.csv and other_var.csv). The training/testing data should have the following named columns: “ID” – patient number, “T” – period, “bool_data” – follow-up indicator, “Y” – gold-standard labels on target clinical outcomes, “silver” – silver standard labels from Step 10, and variables selected in Step 9. Each patient should have the same number of rows, K = maximal follow-up / frequency of assessment as determined in Step 1, indexed by period variable “T” = 1, ..., K . Periods after the last EHR encounter data of the patient are indicated by “bool_data”=FALSE. Missing gold-standard labels on the target clinical outcomes during the period for the patient is marked as “Y”=NA. Selected EHR variables are aggregated over the months within each period. Two variable lists should be created: one for EHR variables selected from the knowledge network (kn_var.csv), and the other for additional variables recommended by domain experts (other_var.csv). The column names of the selected variables in training/testing data must match the names in the variable lists. The embedding file (embedding.csv) contains the semantic embeddings for EHR variables selected from the knowledge network by row with matching rownames.

	ID	T	bool_data	Y	silver	EHR Var:1	...	EHR Var:50	Other: 1	...	Other: 9		x		x
1	1	1	TRUE	0	0.204	0	...	4	0.839	...	0.884	1	EHR Var:1	1	Other:1
2	1	2	TRUE	NA	0.533	0	...	2	0.034	...	0.854	2	EHR Var:2	2	Other:2
3	1	3	TRUE	1	0.807	1	...	4	0.087	...	0.859	3	EHR Var:3	3	Other:3
4	1	4	FALSE	NA	0	0	...	8	0.087	...	0.879	4	EHR Var:4	4	Other:4
5	2	1	TRUE	NA	0.561	0	...	11	0.015	...	0.775	5	EHR Var:5	5	Other:5
...
20000	5000	4	TRUE	NA	0.473	0	...	9	0.034	...	0.527	50	EHR Var:50	9	Other:9

Formatting of training/testing data.

Formatting of variable lists.

Derive the longitudinal EHR clinical outcomes using LATTE



Timing: 1 week

As illustrated in Figure XX, with longitudinal EHR data and a small number of labels on the phenotype status over time as the input, LATTE consists of four key computational components: (a) a Concept Re-weighting module that assigns a weight for each input concept

or feature; (b) a Visit Attention Network which aims to assign higher weights to visits that are more indicative of the incident; (c) a sequential model to capture visit temporal dependency and obtain visit representations; (d) final incident predictions at each visit by the Predictor. To alleviate the need for gold-standard labels, the learning of LATTE includes two key steps. (i) LATTE constructs longitudinal silver-standard labels for the event status over time based on predictive surrogates to perform unsupervised model pre-training. (ii) LATTE is fine-tuned jointly by the gold-standard labels and silver-standard labels.

16. **Training LATTE:** To run LATTE using the formatted data, put all 5 csv files created in Step 12 in the chosen “input” folder. Then, call LATTE from the installed environment set up during preparation.

```
module load gcc/6.2.0 conda2/4.2.13 # Load GCC and Python Conda
source activate env_LATTE # Load the environment

python3 LATTE.py --train_dfname "input/train.csv" \ # Training data directory
--test_dfname "input/test.csv" \ # Testing data directory
--ftsname "input/kn_var.csv" \ # Variable list from Knowledge Network
--other_ftsname "input/other_var.csv" \ # Other variable list
--embed_dim 200 \ # Dimension of the embedding vectors
--embed_fname "input/embedding.csv" \ # Directory of embedding vectors
--key_code "EHR Var:1" \ # Main PheCode for the disease
--output_directory "output/" \ # Output directory
--output_fname "latte_pred" \ # Output file
--epochs 50 \ # Numbers of iterations
--max_visits 4 \ # Maximal periods K = maximal follow-up / frequency of assessment
--layers_incident "120" \ # Number of layers for LATTE
--weight_prevalence 0.3 \ # Weight for cumulative loss
--weight_unlabel 0.3 \ # Weight for unlabeled data
```

The output file for LATTE contains named columns “ID”, “T”, “bool_data”, “Y” (described in Step 11 for training/test data) plus a “Y_pred” column of imputed probability for “Y”=1. LATTE has 4 tuning parameters: i) “epochs” – number of iterations, ii) “layers_incidence” – depth of the neural network, iii) “weight_prevalence” – contribution of the loss for prevalence, iv) “weight_unlabel” – contribution of the loss from unlabeled data. The loss for prevalence aggregates longitudinal clinical outcomes to the binary “ever Y=1” vs “never Y=1”, which is most effective for progressive clinical outcome that develops from 0 to 1 overtime and rarely reverts back. The loss from unlabeled data utilizes the silver standard labels to incorporate unlabeled data for training the LATTE model and is most effective with high-quality silver standard labels. Due to the random nature of stochastic algorithms, average over repeated cross validation (e.g. 10 times 5-fold cross-validation) is recommended for tuning the parameters.

	ID	T	bool_data	Y	Y_Pred
1	1	1	TRUE	0	0.102
2	1	2	TRUE	NA	0.240
3	1	3	TRUE	1	0.910
4	1	4	FALSE	NA	0
5	2	1	TRUE	NA	0.322
...
20000	5000	4	TRUE	NA	0.671

Output table of LATTE.

17. **(Optional) Ensemble Learning to Combine Other Machine Learning Methods:** The LATTE imputed probability “Y_pred” can be combined with imputation from other machine learning methods. Standard (supervised) machine learning such as LASSO,^{15,16} random forest,¹⁷ boosting^{18,19} can be used to train imputation models by regressing clinical outcomes “Y” to predictors “silver”, “EHR VAR:x” and “Other:x” over patient-period with gold-standard labels. To reduce overfitting bias, we recommend aggregation of multiple imputation models through a cross-fitted logistic regression.

- Split the labeled data into 5 folds.
- For each fold, obtain the imputed logit probabilities from cross-fitted LATTE and machine learning models trained with out-of-fold labels.
- Construct the ensemble model by regressing “Y” on to cross-fitted imputed logit probabilities via logistic regression.

EXPECTED OUTCOMES

The output is the imputed probabilities for longitudinal clinical outcomes over i) unlabeled patients and ii) periods with no observed assessment for labeled patients. The imputed probability can be used in downstream analysis in two different ways.

Apply to downstream analyses

- Derive EHR clinical outcomes according to gold-standard labels.** Imputed probabilities can be used to classify clinical outcomes by chosen thresholds. To ensure consistency of prevalence between derived EHR outcomes and annotated gold-standard outcomes, the prevalence under a series of thresholds (e.g. 0.01, 0.02, ..., 0.99) should be computed to select the best threshold producing the consistent prevalence.
- Augment gold-standard labels in surrogate assisted semi-supervised learning.** The imputed probabilities can also be used as the predictive surrogate in surrogate assisted semi-supervised learning for risk prediction²⁰ or estimation of average treatment effect.²¹

LIMITATIONS

TROUBLESHOOTING

RESOURCE AVAILABILITY

REFERENCE

1. Wen, J., Hou, J., Bonzel, C.-L., Zhao, Y., Castro, V.M., Gainer, V.S., Weisenfeld, D., Cai, T., Ho, Y.-L., Panickan, V.A., et al. (2024). LATTE: Label-efficient incident phenotyping from longitudinal electronic health records. *Patterns (N Y)* 5, 100906.
2. Denny, J.C., Ritchie, M.D., Basford, M.A., Pulley, J.M., Bastarache, L., Brown-Gentry, K., Wang, D., Masys, D.R., Roden, D.M., and Crawford, D.C. (2010). PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene–disease associations. *Bioinformatics* 26, 1205–1210.
3. Humphreys, B.L., and Lindberg, D.A. (1993). The UMLS project: making the conceptual connection between users and the information they need. *Bull. Med. Libr. Assoc.* 81, 170–177.
4. Hong, C., Rush, E., Liu, M., Zhou, D., Sun, J., Sonabend, A., Castro, V.M., Schubert, P., Panickan, V.A., Cai, T., et al. (2021). Clinical knowledge extraction via sparse embedding regression (KESER) with multi-center large scale electronic health record data. *NPJ Digit Med* 4, 151.
5. Xiong, X., Sweet, S.M., Liu, M., Hong, C., Bonzel, C.-L., Panickan, V.A., Zhou, D., Wang, L., Costa, L., Ho, Y.-L., et al. (2023). Knowledge-Driven Online Multimodal Automated Phenotyping System. *medRxiv*. 10.1101/2023.09.29.23296239.
6. Yu, S., Cai, T., and Cai, T. (2013). NILE: Fast Natural Language Processing for Electronic Health Records. *arXiv [cs.CL]*.
7. Yang, S., Varghese, P., Stephenson, E., Tu, K., and Gronsbell, J. (2023). Machine learning approaches for electronic health records phenotyping: a methodical review. *J. Am. Med. Inform. Assoc.* 30, 367–381.
8. Zhang, Y., Cai, T., Yu, S., Cho, K., Hong, C., Sun, J., Huang, J., Ho, Y.-L., Ananthakrishnan, A.N., Xia, Z., et al. (2019). High-throughput phenotyping with electronic medical record data using a common semi-supervised approach (PheCAP). *Nat. Protoc.* 14, 3426–3444.
9. Yu, S., Ma, Y., Gronsbell, J., Cai, T., Ananthakrishnan, A.N., Gainer, V.S., Churchill, S.E., Szolovits, P., Murphy, S.N., Kohane, I.S., et al. (2017). Enabling phenotypic big data with PheNorm. *J. Am. Med. Inform. Assoc.* 25, 54–60.
10. Liao, K.P., Sun, J., Cai, T.A., Link, N., Hong, C., Huang, J., Huffman, J.E., Gronsbell, J., Zhang, Y., Ho, Y.-L., et al. (2019). High-throughput multimodal automated phenotyping (MAP) with application to PheWAS. *J. Am. Med. Inform. Assoc.* 26, 1255–1262.
11. Ahuja, Y., Wen, J., Hong, C., Xia, Z., Huang, S., and Cai, T. (2022). A semi-supervised adaptive Markov Gaussian embedding process (SAMGEP) for prediction of phenotype event times using the electronic health record. *Sci. Rep.* 12, 17737.

12. Yuan, Q., Cai, T., Hong, C., Du, M., Johnson, B.E., Lanuti, M., Cai, T., and Christiani, D.C. (2021). Performance of a Machine Learning Algorithm Using Electronic Health Record Data to Identify and Estimate Survival in a Longitudinal Cohort of Patients With Lung Cancer. *JAMA Netw Open* 4, e2114723.
13. Humbert-Droz, M., Izadi, Z., Schmajuk, G., Gianfrancesco, M., Baker, M.C., Yazdany, J., and Tamang, S. (2023). Development of a natural language processing system for extracting rheumatoid arthritis outcomes from clinical notes using the national rheumatology informatics system for Effectiveness registry. *Arthritis Care Res.* 75, 608–615.
14. Fernandes, M.B., Valizadeh, N., Alabsi, H.S., Quadri, S.A., Tesh, R.A., Bucklin, A.A., Sun, H., Jain, A., Brenner, L.N., Ye, E., et al. (2023). Classification of neurologic outcomes from medical notes using natural language processing. *Expert Syst. Appl.* 214. 10.1016/j.eswa.2022.119171.
15. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Series B Stat. Methodol.* 58, 267–288.
16. Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *J. Am. Stat. Assoc.* 101, 1418–1429.
17. Breiman, L. (2001). Random Forests. *Mach. Learn.* 45, 5–32.
18. Schapire, R.E. (1990). The strength of weak learnability. *Mach. Learn.* 5, 197–227.
19. Chen, T., and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '16*. (Association for Computing Machinery), pp. 785–794.
20. Hou, J., Guo, Z., and Cai, T. (2023). Surrogate Assisted Semi-supervised Inference for High Dimensional Risk Prediction. *J. Mach. Learn. Res.* 24, 1–58.
21. Hou, J., Mukherjee, R., and Cai, T. (2021). Efficient and Robust Semi-supervised Estimation of ATE with Partially Annotated Treatment and Response. *arXiv [stat.ME]*.