Supplementary Information: Human and machine learning of prognostic prediction for prelabor rupture of membranes and the time of delivery: a nationwide development, validation, and deployment using medical history

Contents

List of Figures

List of Tables

Please kindly see Supplemental Spreadsheet. Most are complex tables. There are 20 tables in different tabs.

In this Supplementary Information, we describe details on this study following chronological order of our analysis pipeline on causal inference or predictive modeling for prelabor rupture of membranes (PROM). There three sections
corresponding to the same sections in the main text but different order, which are respectively Methods, Results, and Discussion. Along with this PDF document, we also provide R Markdown (.Rmd) containing the same texts with this document but including the programming codes for the data analysis in-between of these texts. The codes for core steps in the analysis pipeline are also provided exclusively in an R Script (.R). The codes beyond the core steps were used for analytic decision or creating tables or figures. These are shown to provide details on how data are processed to construct all tables and figures in both the main text and this Supplementary Information, including those in Source Data Spreadsheets (.xlsx) for all figures in the main text and those in a single Supplemental Spreadsheet (.xlsx) with multiple tabs for all tables in this Supplementary Information. The complex tables were produced separately as comma-separated value (.csv) files then compiled into the single Supplemental Spreadsheet. We also provided a 5-minute Supplemental Video to briefly explain technical details on deep-insight visible neural network (DI-VNN) pipeline.

Methods

Research guidelines

We followed three research guidelines. The checklists for all the guidelines are shown (Table 1 to 3 in Supplemental Spreadsheet). To find comparable models to evaluate success criteria, we also followed other guidelines. The checklist and the comparable models are also described (Table 4 and 5 in Supplemental Spreadsheet).

Programming environment

We set up a programming environment for this study. Bioconductor was utilized as described in the main text. There were 198 R packages which are 9 base packages, 53 other packages, and 136 dependencies (Table 6 in Supplemental Spreadsheet).

Sampling procedures of the data source

The data source was a sample dataset of the whole health insurance database during 2015 and 2016 by cross-sectional design. Stratified random sampling was applied. The strata variable was constructed from 66,072 combinations of all the healthcare facilities (n=22,024) and category of family, which were: (1) a family of which members never visit the healthcare facilities; (2) a family of which members have visited only primary care; and (3) a family of which members have visited all levels of care. For each stratum, one to ten families were randomly included. This means only 10 families were randomly included if more than that number, resulting 586,969 families with 1,697,452 subjects.

The sampling procedures of the dataset in this study

We conducted non-essential data cleaning, e.g. revising the inconsistent name of states, estimating the healthcare identifiers, *et cetera*. These procedures were parts of our R package of ... (DBPR). No sampling was conducted.

After the non-essential data cleaning, we applied retrospective cohort design, as described in the main text. For pregnant women, we use several codes for determining delivery or immediately after delivery care. The 220 codes are described (Table 7 in Supplemental Spreadsheet).

Data preprocessing

We conducted data preprocessing after defining the target population and sampling it retrospectively. Demographics were included as categorical variables for causal inference. Then, we applied systematic human learning, as described in the main text, to determine what were causal factors that can be inferred from our dataset. We also computed a number of days for a code in the latest encounter before the time of prediction, including those by codes as a causal factor.

Data partition

To ensure all inference or derivation using training set only, we need to conduct data partition before continuing the downstream analysis. We described data partition for model validation in the main text (see Methods).

Causal inference

We conducted causal inference as described in the main text. This will help us to include only the confirmed causal factors as candidate predictors before conducting pre-selection of those candidates to fulfill quality control of predictors in the main text. We included causal factors of which the data were available in training set. Details on this information and ICD-10 codes or demographical variables for each candidate of causal factors are shown in the next section.

Quality control of candidate predictors

All candidate predictors, including non-demographical causal factors, have non-zero variances (Table 8 in Supplemental Spreadsheet). There were 460 candidate predictors fulfilling this criterion. We also showed in the same table that there are 426 candidate predictors without perfect separation.

We excluded the diagnosis/procedure codes that may leak the outcome information (Table 9 in Supplemental Spreadsheet). We only used the existing codes in the training set to determine outcome-leaker codes based on the previous codes for determining delivery or immediately after delivery care (Table 7 in Supplemental Spreadsheet). There were 54 codes that may leak the outcome. All of them were also irredundant (Table 10 in Supplemental Spreadsheet), as described in the main text.

By systematic human learning and causal inference using available data, we also determined causal factors as the candidate predictors (Table 11 in Supplemental Spreadsheet). There were 27 first- and 10 second-level factors of PROM. Only data for 12 out of 27 causal factors were available in training set. Either the diagnosis/procedure codes, or demographical variables (not included as candidate predictors), for causal factors are also described (Table 12 in Supplemental Spreadsheet).

Feature extraction as historical rates

We inferred the nationwide historical rates given the day number from a code encounter to current visit for each candidate predictor, as described in the main text. This used irredundant candidate predictors with non-zero variances and no perfect separation in training set only. The candidate predictors were transformed into the historical rates in all data partitions.

Feature representation as principal components by 10-fold cross validation

The historical rates of all candidate predictors were fitted to a principal component (PC) model. Only training set was used for the model fitting. We applied 10-fold cross validation to estimate weights for all candidate predictors in each PC.

Set up tuning grid and training-calibrating configuration

Previous data partition had not held out instances for calibration yet. This took 80% of training set. We also gave different weights for event and nonevent by including censored outcome, as described in the main text. For hyperparameter tuning, we applied 5-fold cross validation, instead of 10-fold as applied for PC modeling. Meanwhile, the final training and calibration for each model were conducted by bootstrapping for 30 times. The same resampling methods were applied for both classification and estimation tasks. Parallel computing by multiple central processing units (CPUs) were applied for training all models.

Hyperparameter tuning, final training, and calibrating

We applied the tuning grids and the training configurations for all models, except DI-VNN which required several modifications. This is already described clearly in the main text. More details will be described for DI-VNN in the next section.

Deep-insight visible neural network (DI-VNN)

We applied different preprocessing pipelines for feature selection and representation in DI-VNN. Instead of PCs, we used the historical rates of the candidate predictors. A 5-minute video explaining DI-VNN pipeline is available (see Supplemental Video). Only pre-calibration training set was used for the downstream pipeline.

We used nationwide, pre-calibration training set to conduct feature selection by differential analysis. Nationwide, instead of provider-wise, training set was used because we need to find the differential effect at population level as such in causal inference. First, we applied quantile-to-quantile normalization, then we conducted differential analysis as described in the main text. Only candidate predictors that showed adjusted p-values <0.05 were selected.

For 1-bit stochastic gradient descent transformation, we used the post-normalization, feature-wise average based on nationwide training set as the target of which quantile-to-quantile normalization of any subsets were applied onto. These included subsets for calibration and external validation. Instead of nationwide, these subsets were provider-wise, because we used these for prediction that likely uses medical history from a healthcare provider visited by a subject. Then, the transformation was applied as described in the main text, depending on the feature wise averages.

Demanding different statistical assumption, we used unnormalized candidate predictors for creating a feature map (i.e. ontology array) and a network architecture (i.e. ontology network). Both procedures need a distance/similarity matrix. We also used nationwide, pre-calibration training set to construct this matrix. Only the candidate features selected by differential analysis were used. Standardization was applied by subtracting each value with feature-wise average and dividing it with feature-wise standard deviation. Then, we computed a feature-to-feature Pearson correlation matrix.

For creating a feature map, we projected the filtered candidate predictors onto three dimensions, as described in the main text. The results consisted three numbers for each of the candidate predictors. We transformed two of these values as ranks. If we have 5 candidate predictors, then we order these for axes of x and y. For example, a candidate predictor is mapped at $x=1$ and $y=3$ in a two-dimensional space. This is because that predictor has values on dimensions of x and y, that rank respectively at 1^{st} and 3^{rd} positions among the five candidate predictors. Then, we split axes of x and y into 7×7 grid; thus, there might be more than one candidate predictors at the same position. We rotated the map by convex-hull algorithm, as applied in the original DeepInsight, to find smallest feature map size then re-ranking the dimensions. For any positions, we computed maximum number of candidate predictors residing on the same position. That number determined the number of the two-dimensional layers, i.e. channels. The overlapped candidate predictors were ranked based on the third dimension. The ranks determined which channels a candidate predictor residing on for each position on the two-dimensional space.

This can be considered as a three-dimensional array. In each position, a
candidate predictor may have a value of -1, 0, or 1, depending on the 1-bit stochastic gradient descent transformation. For any positions in the array, that have no candidate predictor residing on, a value of 0 is applied. The empty position may be occupied by an unfiltered candidate predictor had it surpassed filtering by the differential analysis. Therefore, zero value for the empty position has the same notion with either the unfiltered candidate predictors or those with normalized values equal to the feature-wise averages, i.e. undifferentiated features in relative to the outcome. Ranking the dimension is a novel method to reduce array dimensions, different to that applied by the previous DeepInsight transformation. Positioning of candidate predictors on the feature map was derived from pre-calibration training set. This positioning is also used for constructing the three-dimensional array of the other subsets in calibration and external validation.

Meanwhile, using the same correlation matrix, we applied CliXO algorithm, as described in the main text. We applied α=*0.01* and β=*0.5*, as recommended. The resulting ontologies derived from pre-calibration training set were used to subset a feature map into different arrays, including in calibration and external validation. If we have 5 candidate predictors, feature 1 and 5 may be clustered into the same ontology while feature 3 and 4 into another ontology. To subset the feature map into an array for the first ontology, we multiplied all numbers in the array with 0 except the feature members of that ontology, which are feature 1 and 5. The same method is applied for the ontology including feature 3 and 4. Since CliXO is an agglomerative hierarchical clustering algorithm, both ontologies may unite into an ontology. In this example, the unified ontology includes all candidate features. But, this may not be the case. An additional ontology, called root ontology, is added to include all candidate predictors, either from the remaining candidate predictors, if any, or ones in the ontology consisting the most candidate predictors (i.e. highest ontology in the hierarchy). Eventually, we had an array for each ontology derived by CliXO algorithm.

Each ontology array was fed to a block of neural network, i.e. Inception v4-Resnet. In this architecture, an array in terminal branch of an ontology hierarchy is filtered by feeding it through Inception v4, then reconstructing the array into the same dimensions as it entered the Inception v4. Element-wise addition is applied between the filtered array and the original one. This mathematical operation is the Resnet architecture itself. To connect a child ontology array with a sibling ontology, if any, we applied depth concatenation, i.e. by the channel. The concatenated array (double or more channels) is fed to the Inception v4, then reconstructing the array into the same dimensions with the parent ontology array. The same mathematical operation is applied for the concatenated, filtered array and the parent one. Child-to-parent connection was applied between one or more arrays in non-terminal branch of an ontology hierarchy and the parent array. Each of either the terminal or non-terminal branch had a transformed array as a representation of an input array of each ontology. To get a representation, the transformation is achieved by a series of iterations using backpropagation algorithm.

Before explaining the backpropagation, we need to know how each ontology array was transformed into a single number from 0 to 1 that predicts a probability of an event, or an integer for estimation task. After transformation by a block of Inception v4-Resnet, the transformed array was fed to a block of layers for convolution. This reduced the dimensions from 7×7 in each channel into a single number by a series of mathematical operation of convolution. Therefore, we have multiple numbers showing probabilities of an event for each instance, transformed and convoluted from all ontology arrays.

For each iteration, we computed differences between the outcome, which is 0 and 1 respectively for nonevent and event, and each probability convoluted from each ontology array. For estimation task, this was the differences between the true time and the predicted time of delivery. The computation was applied by the loss function, as described in the main text. A batch size of 512 instances was computed for the loss in each iteration. The loss or error was used to update the weights. These were initiated randomly, as described in the main text, then the weights were updated via backpropagation from the root ontology to the terminal ones. This procedure is iterative using 512 instances each time until 80% instances were used in pre-calibration training set. This achieved a cycle of iterations, or epoch. At the end of each epoch, we computed the loss and AUROC using another 20% instances. Each update was multiplied by a number, called learning rate. We applied a learning rate of 2ˆ-6. For the iterations covering the first 5% of instances in each epoch, the learning rate was initiated at one thirty-second of the learning rate at the first iteration

for that epoch. This is gradually increased until reaching the learning rate at the last iteration covering the first 5% of instances, i.e. warm-up strategy. From an epoch to the next one, the learning rate for the next epoch was reduced by 4% if the AUROC of the 20%-validation subset was no higher than 0.01 in addition to that of the previous epoch. No reduction was applied if reaching the minimum, which is 1/512 of the initial learning rate. Maximum epochs of 5 and 500 was set respectively for the hyperparameter tuning and the final training. After 50% of the maximum epochs, the iterations were stopped earlier if the AUROC of the 20%-validation subset was no higher than 0.001 in addition to that of the previous epoch. Only weights of the best iteration were finally used. This was determined based on the validation AUROC. After stopping, we computed AUROCs of the 20% validation subsets by bootstrapping for 30 times. All of these procedures were applied for each trial in the hyperparameter tuning and the final training. The tuning grid was applying different λ values, as described in the main text. The best tuning parameter was determined by the bootstrapped, validation AUROC. Eventually, the same calibration and external validation procedures were applied on DI-VNN, as applied on the other models.

Evaluating the best model for classification and estimation

Model evaluation is already clearly described in the main text. We started from evaluating the calibration measures of all models for classification task. Then, we chose the best model for classification by AUROC using only internal validation (calibration split) bootstrapped for 30 times. Using the same subset, the best estimation model was also chosen.

External validation

We also confirmed the robustness of the best models using external validation sets. Uncertainty intervals were also computed by bootstrapping for 30 times. A clear description on model validation is already given in the main text.

Exploring the best model

Model exploration method is also described in the main text (see Model evaluation in Methods). In this Supplementary Information (see R Markdown and R Script), we directly explored the best model for classification task. None of the remaining models were explored.

Preparing web application

For web application, we prepared an example dataset, user interface, and processing script at the side of server computer. No line of codes for the web is included in the R Markdown or R Script. Description for this web application is already clearly described in the main text.

Result

By stratified random splitting, we provided three non-overlapping splits comprising ~20% for external validation: (1) geographical split; (2) temporal split; and (3) geotemporal split. We set these subsets challenging to predict by our models, in which the geotemporal split was the most difficult. This way we can estimate robustness of our model generalizability. But, these do not reflect common situations nationwide; thus, the ~20% of the remaining was held out by simple random splitting for external validation, which is called external random split. For causal inference and predictive modeling, including calibration, we only used the remaining ~64% of all the selected visits.

We provided Source Data Spreadsheet for Figure 1 in the main text. Up to the latest date for uncensored outcome and after splitting if >1 pregnancies, the total visit was the sum of totals from all subsets, while the total subject was the sum of totals from all subsets with attention for the overlaps (d to j in footnote of Source Data Spreadsheet of Figure 1). For causal inference, visits and subjects of the censored outcome were those after external random splitting (k and j in footnote of Source Data Spreadsheet of Figure 1). In this spreadsheet, PROM prevalence for each subset is explicitly shown.

Causal diagram

For candidate causal factors whose data were available, we created the causal diagrams (Figure 1 to 12). We excluded all common effects of PROM that we found during the systematic human learning. These are not needed for causal inference. Instead, inclusion of these variables will cause collider-stratification bias. In the causal diagrams, we apply different colors based on the types of nodes representing several factors: (1) type A is a first-level factor (with variable prefixed by A) that has a role as a confounder or common cause; (2) type I is a first-level factor (with variable also prefixed by A) that has a role as a candidate causal factor of interest; (3) type U is an unmeasured variable that can affect measure variable of type-A/I/Y variable; and (4) type Y is a target or dependent factor which is PROM. Node, of which variable denoted by asterisk, represents a type-A/I/Y variable that can be represented by several diagnosis or procedure codes. In each of the diagrams, we also show causal models or formulas which were used for causal inferences. Conceivably, all variables of type A/I/Y with asterisk may be included in each formula, but, some of these variables cannot be included because these were not available in our data, particularly in the training set (likely because of low prevalence). All measurement errors that may be affected by Type-U variables in this study were assumed as independent non-differential errors, because all data were measured from electronic medical records. As the main text, results of the causal inferences were also described (Table 13 in Supplemental Spreadsheet), either by outcome regression or inverse probability weighting (IPW).

We would describe each causal diagram. Fifty-six studies were found from PubMed (Table 11 in Supplemental Spreadsheet) (. . .). Since only some factors are possible to include in the causal inferences, only some of these studies were explained below. After verifying the assumption using our data, we constructed a final causal diagram consisting all the confirmed causal factors (Figure 2 in the main text). The Source Data Spreadsheet for Figure 2 is available.

Outcome regression showed the same ranks for chorioamnionitis and genital tract infection (GTI), while the effect estimate of intra-amniotic infection on PROM was not statistically significant by this method (OR 2.134, 95% CI 1 to 4.555). Three of 11 factors were assigned as causal factors by IPW but not by outcome regression. These included intra-amniotic infection and two variables: (1) pneumonia (OR 0.91, 95% CI 0.538 to 1.539); and (2) influenza (OR 0.957, 95% CI 0.863 to 1.061). Effects by outcome regression were mostly larger than those by IPW (Table 13 in Supplemental Spreadsheet).

Multiple pregnancy

Causal model of multiple pregnancy on PROM (ACOG, 2016a) included only maternal age as a confounder (Song J, et al 2019; Thilaganathan B, and Khalil A, 2014; Martin JA, and Osterman MJK, 2019). However, several common causes were not blocked yet: (1) Assisted reproduction (Lei LL, et al 2019; Thilaganathan B, and Khalil A, 2014); and (2) Race (Fiscella K, 1996; Martin JA, and Osterman MJK 2019). These are shown in the causal diagram (Figure 1). Assisted reproduction can be represented by diagnosis/procedure codes but unavailable in our training set. For race, it is conceivably not able being represented by those codes.

Chorioamnionitis

There were two assumptions regarding relationship between chorioamnionitis and PROM. Chorioamnionitis may be a causal factor or an effect of PROM. We applied chorioamnionitis as the causal factor (Figure

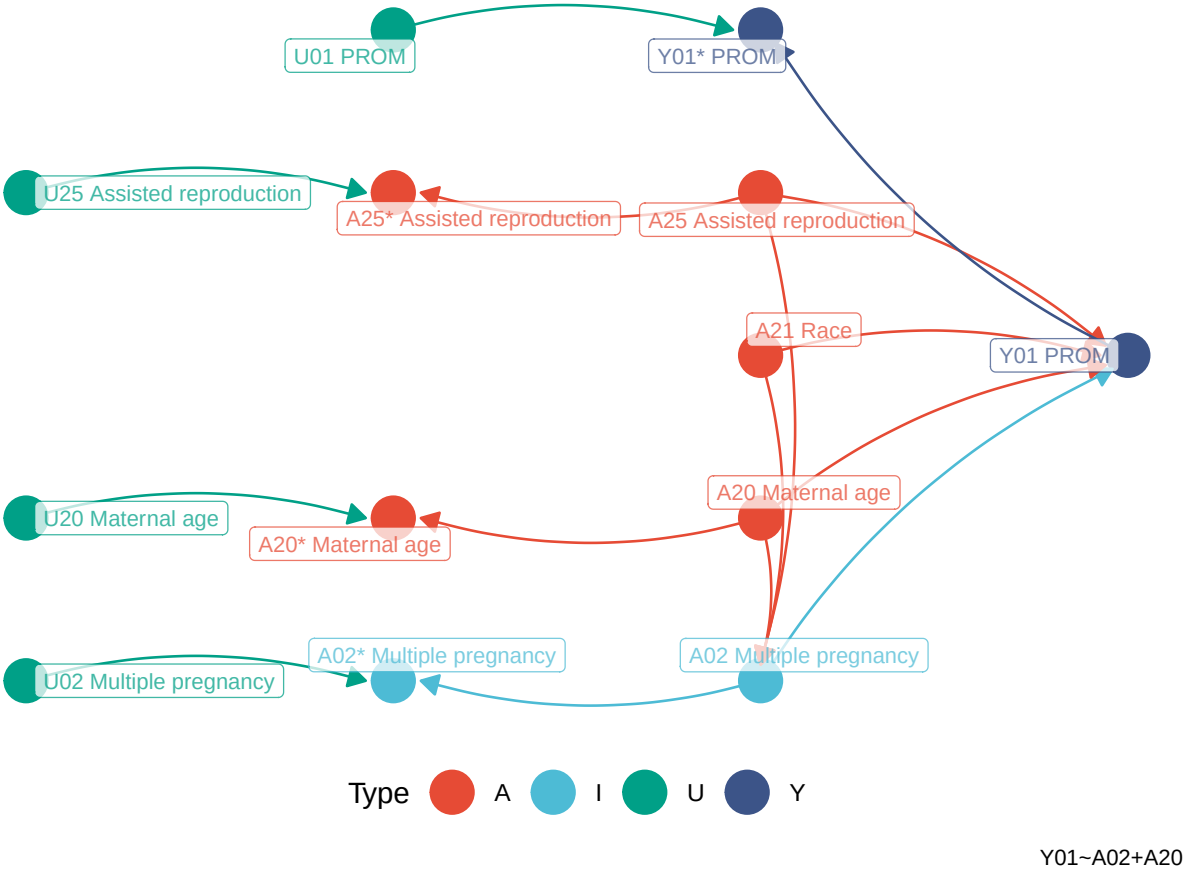

Figure 1: Multiple pregnancy

2), as previously demonstrated (Fukami T, et al 2017). Similar assumptions were also regarded between chorioamnionitis and intra-amniotic infection (IAI), but, we only treated chorioamnionitis as an effect of IAI (Tantengco OAG, et al, 2019). Except cigarette smoking (ACOG, 2016a; Kim CJ, et al, 2015), data for all common causes were available in the training set: (1) influenza (Littauer EQ, et al, 2017; Kim CJ, et al, 2015); (2) asthma (Baghlaf H, et al, 2019); and (3) IAI (ACOG, 2016a; Tantengco OAG, et al, 2019).

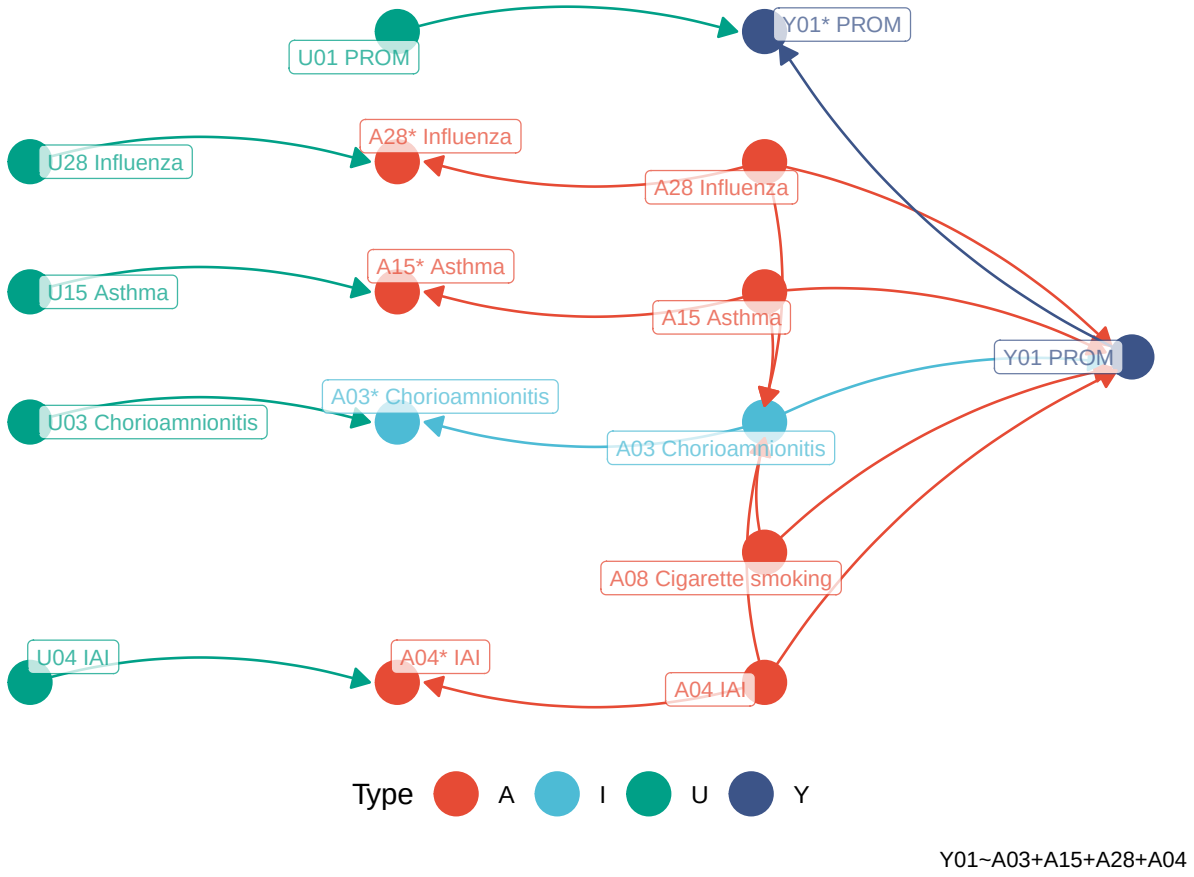

Figure 2: Chorioamnionitis

Intra-amniotic infection (IAI)

Similar to chorioamnionitis, IAI may be a causal factor or an effect of PROM. Consistently, we treated IAI as the causal factor (ACOG, 2016a) (Figure 3). We did not have data for these common causes, especially in training set: (1) cervical shortening (ACOG, 2016a; Kiefer DG, et al, 2009); and (2) race (Fiscella K, 1996; Menon R, et al, 2011). Therefore, we used these factors as the confounders: (1) genital tract infection (GTI) (Pandey D, et al, 2019; Yan JJ, et al, 2016; Romero R, et al, 2019; Tantengco OAG, et al, 2019); (2) periodontal disease (Figueiredo MGOP, et al, 2019; Stinson LF, et al, 2019); (3) pneumonia (Getahun D, et al, 2007; Stinson LF, et al, 2019); and (4) multiple pregnancy (ACOG, 2016a; Lee SM, et al, 2020).

Ante-partum hemorrhage (APH)

Most data for common causes of ante-partum hemorrhage (APH) and PROM (ACOG, 2016a) were not available in the training set (Figure 4). Only two common causes were used as confounders: (1) low socioe-

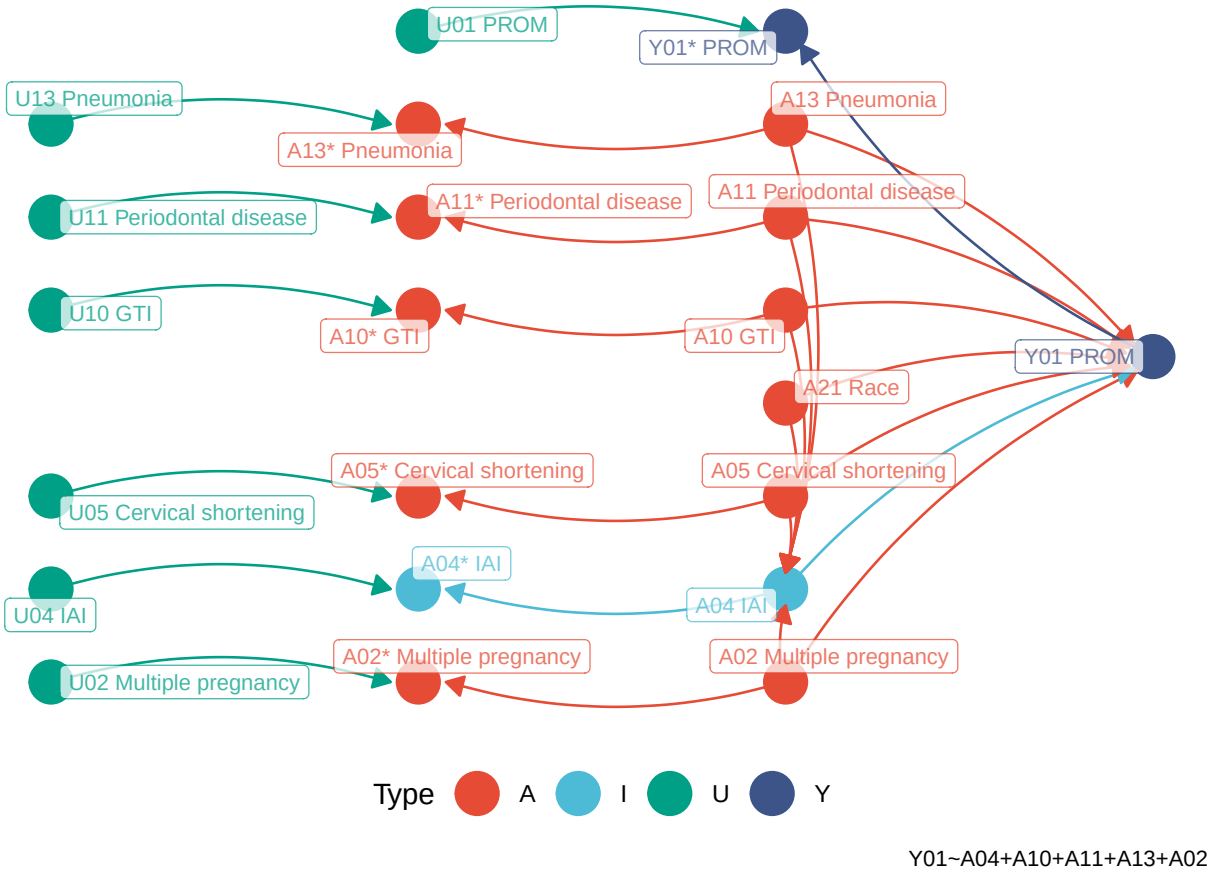

Figure 3: Intra-amniotic infection (IAI)

conomic status (SES) (ACOG, 2016a; Bhandari S, et al, 2014); and (2) maternal age (Song J, et al, 2019; Fan D, et al, 2017). The other common causes were: (1) cigarette smoking (ACOG, 2016a; Bhandari S, et al, 2014); (2) illicit drug use (ACOG, 2016a; Bhandari S, et al, 2014); (3) race (Fiscella K, 1996; Shen JJ, et al, 2005); (4) assisted reproduction (Lei LL, et al, 2019; Qin J, et al, 2016); and (5) placenta on anterior wall (Torricelli M, et al, 2015; Fan D, et al, 2017).

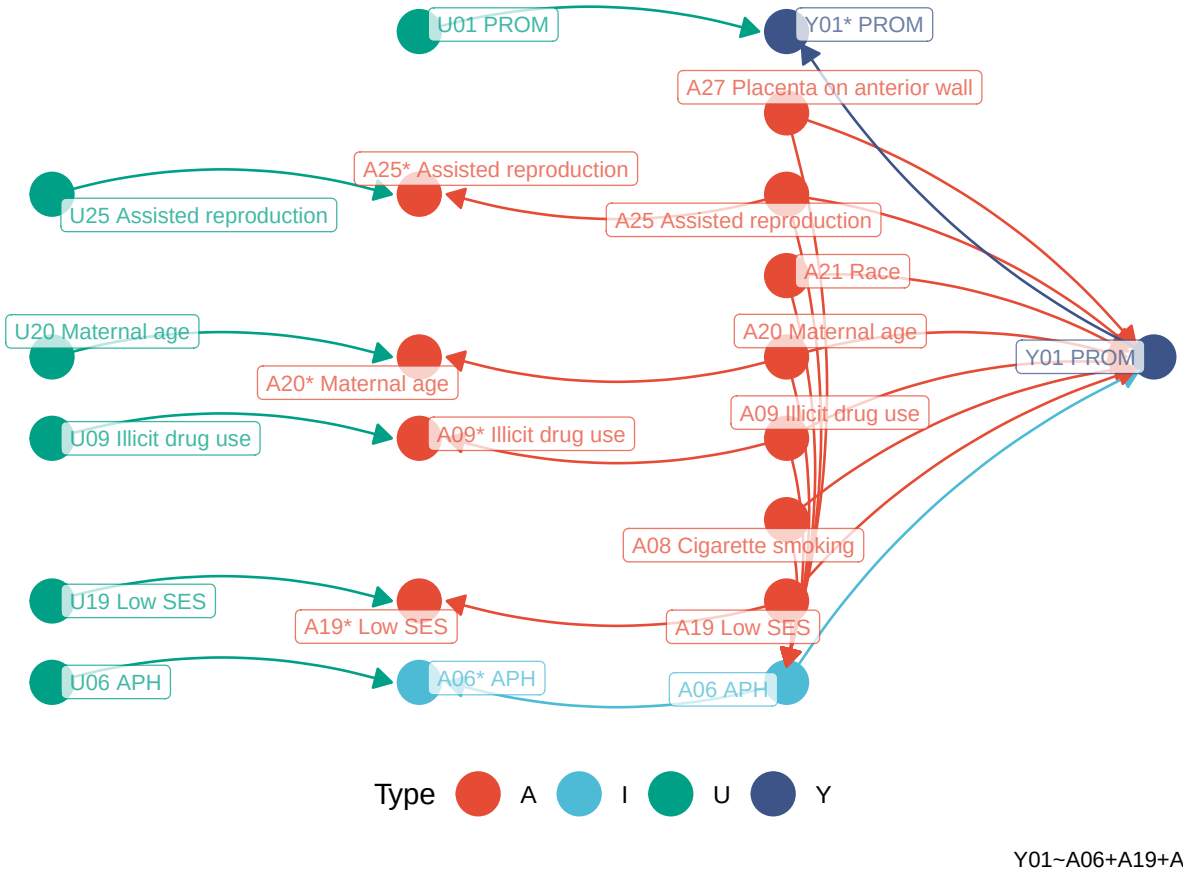

Figure 4: Ante-partum hemorrhage (APH)

Genital tract infection (GTI)

Causal relationship between GTI and PROM (Pandey D, et al, 2019; Yan JJ, et al, 2016) was only confounded by tuberculosis (Fernández AA, et al, 2017; Sharma JB, et al, 2018). We did not have data for tuberculosis in the training set. Thus, GTI is the only variable in the causal model (Figure 5).

Periodontal disease

We also found causal relationship between periodontal disease and PROM (Figueiredo MGOP, et al, 2019). A causal model was constructed by adding these common causes: (1) asthma (Baghlaf H, et al, 2019; Moraschini V, et al, 2018); and (2) maternal age (Song J, et al, 2019; Genco RJ, and Borgnakke WS, 2013). Because of data availability, we could not include these common causes into the model: (1) stress (Wang W, et al, 2020; Genco RJ, and Borgnakke WS, 2013); (2) low education (Wang W, et al, 2020; Genco RJ, and

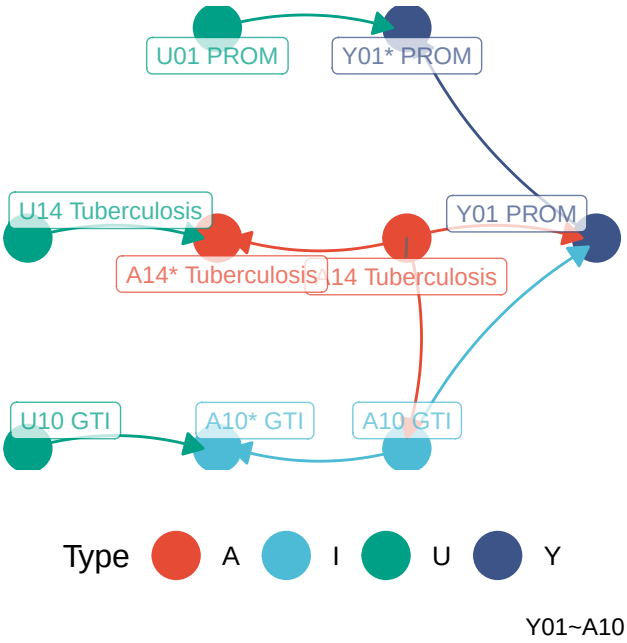

Figure 5: Genital tract infection (GTI)

Borgnakke WS, 2013); and cigarette smoking (ACOG, 2016a; Genco RJ, and Borgnakke WS, 2013) (Figure 6).

Polyhydramnios

Polyhydramnios was also a causal factor of PROM (Odibo IN, et al, 2016). The common causes were: (1) assisted reproduction (Thilaganathan B, and Khalil A, 2014; Lei LL, et al, 2019); and (2) multiple pregnancy (ACOG, 2016a; Moise KJ, 1997). Only multiple pregnancy data were available in our training set; thus, a PROM causal model was constructed using polyhydramnios and multiple pregnancy (Figure 7).

Pneumonia

Causal model of pneumonia on PROM (Getahun D, et al, 2007) was confounded by two common causes. The first common cause was influenza (Littauer EQ, et al, 2017; Goodnight WH, and Soper DE, 2005), while the second one was asthma (Baghlaf H, et al, 2019; Goodnight WH, and Soper DE, 2005). Both were included in the causal model (Figure 8).

Asthma

As shown in the causal model of pneumonia and PROM, asthma was also a causal factor of PROM (Baghlaf H, et al, 2019). Influenza was the only common cause (Littauer EQ, et al, 2017; Murphy VE, et al, 2017). Therefore, we included this common cause in the causal model of asthma on PROM (Figure 9.

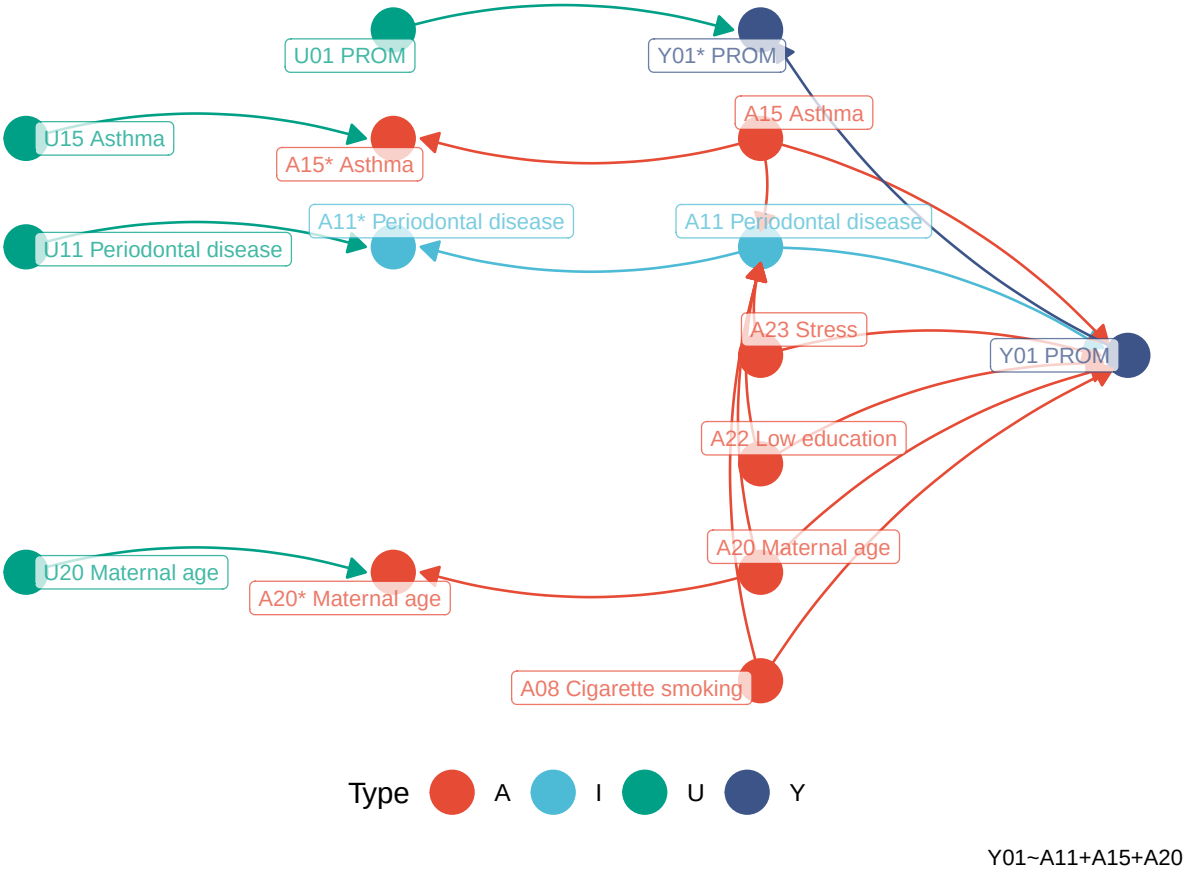

Figure 6: Periodontal disease

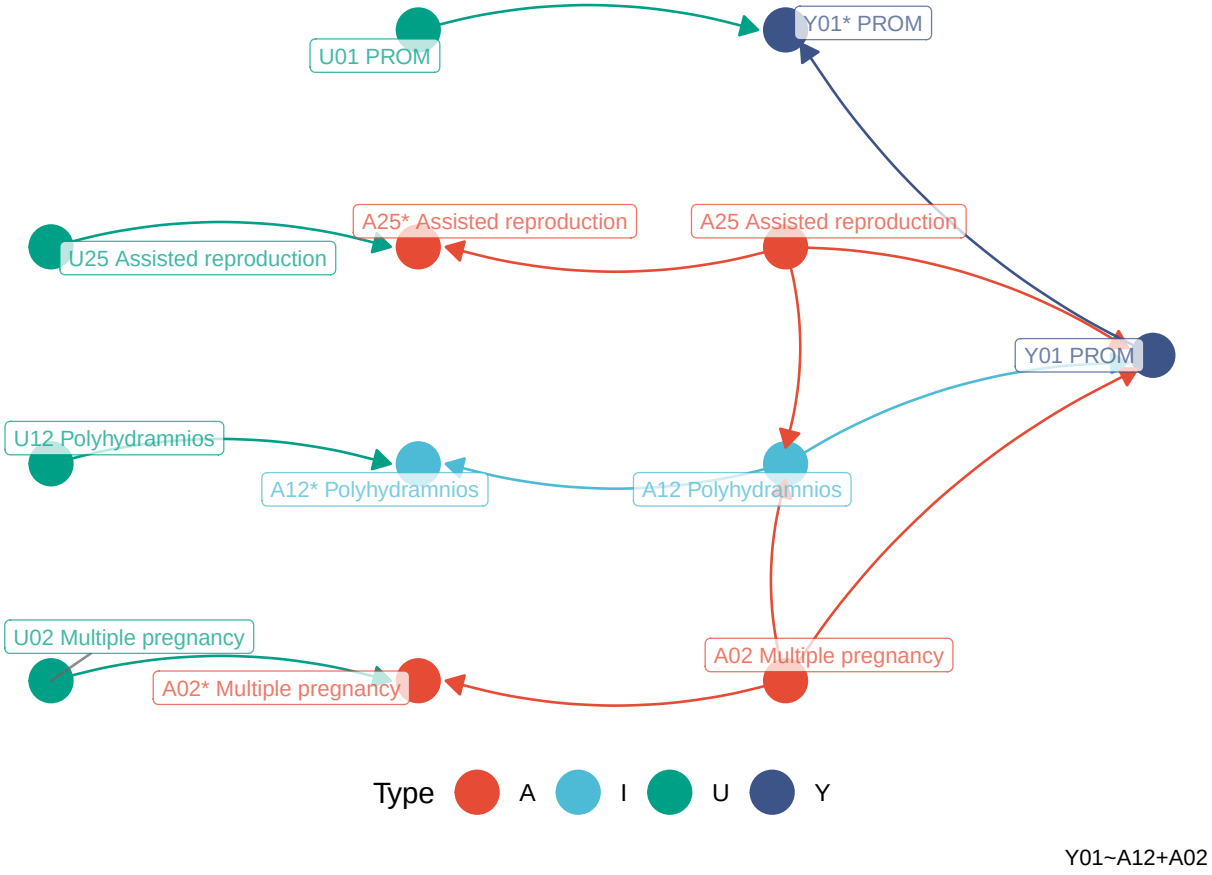

Figure 7: Polyhydramnios

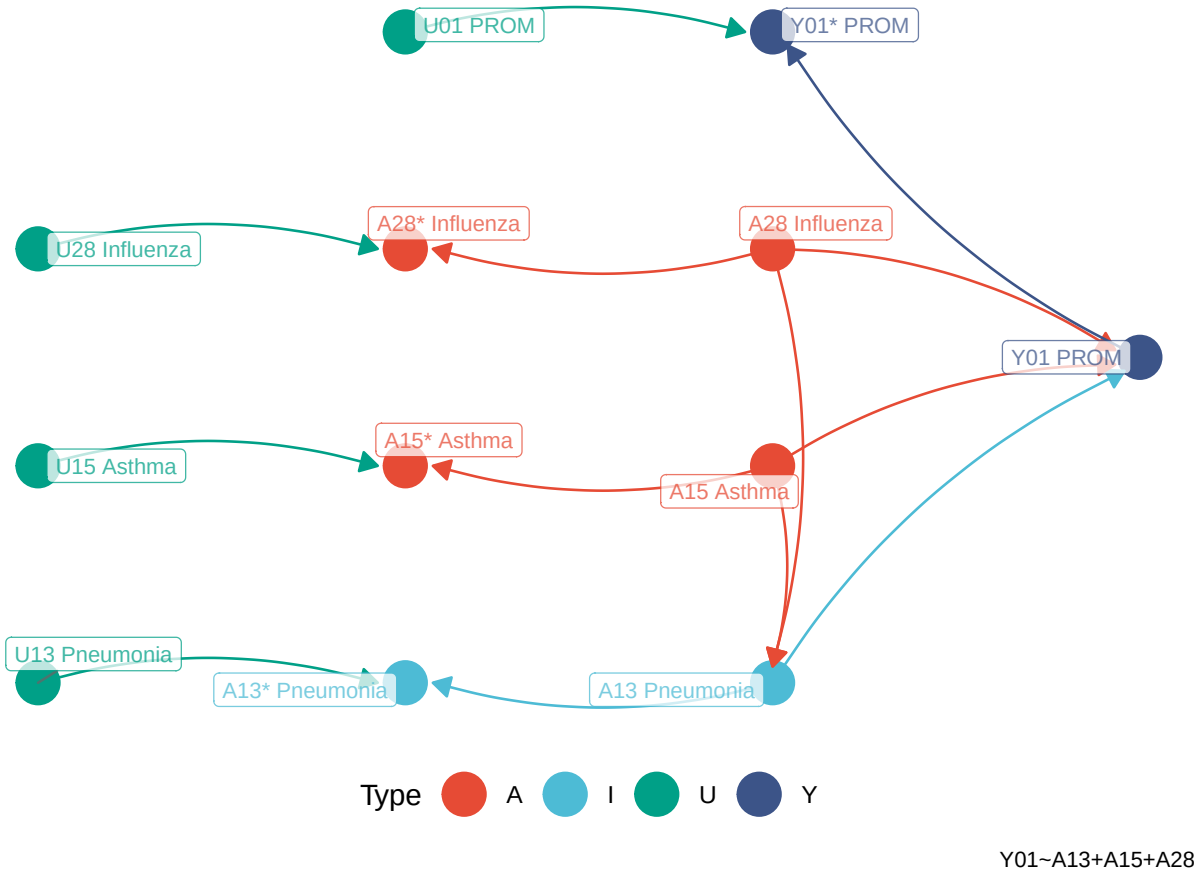

Figure 8: Pneumonia

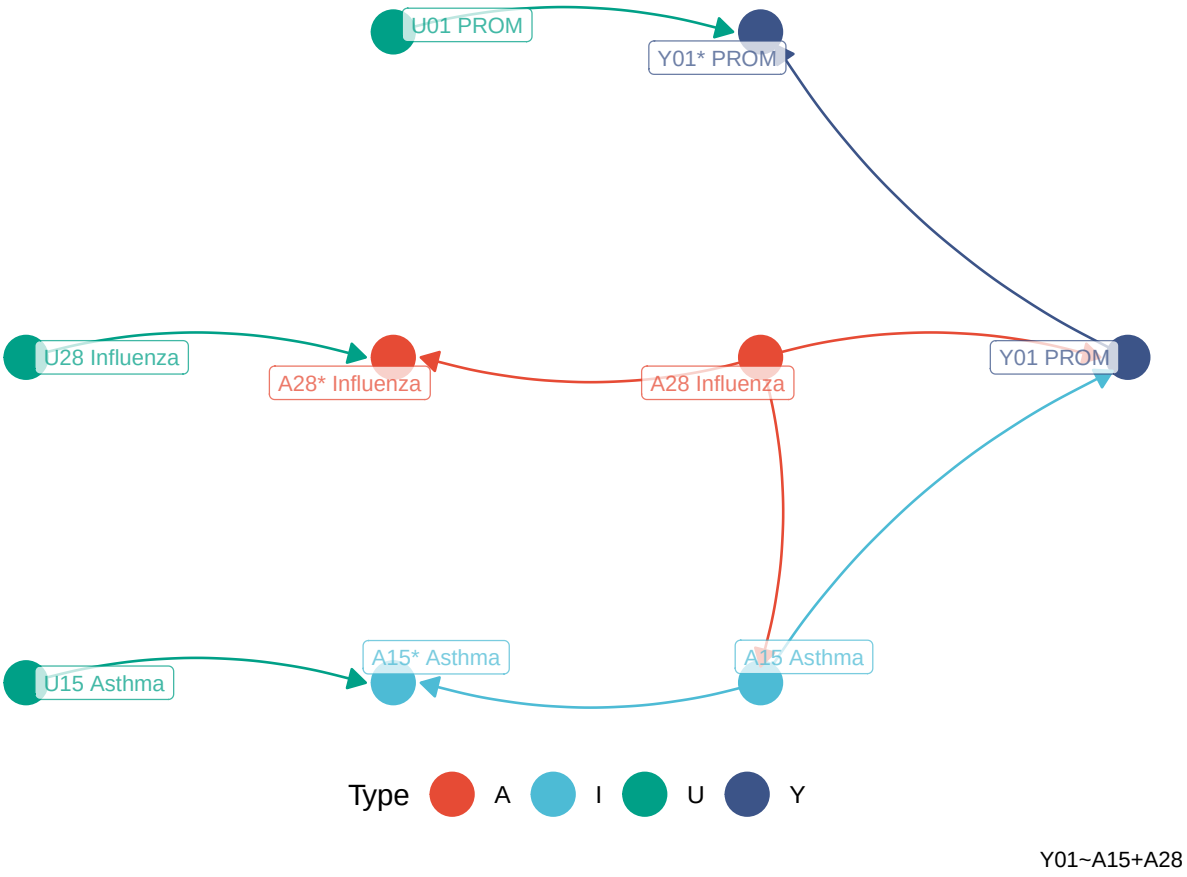

Figure 9: Asthma

Low socioeconomic status (SES)

Low SES was also indicated as a causal factor of PROM (ACOG, 2016a). We could not find the common cause. The causal model only included low SES. We represented several demographical factors as low SES (Figure 10).

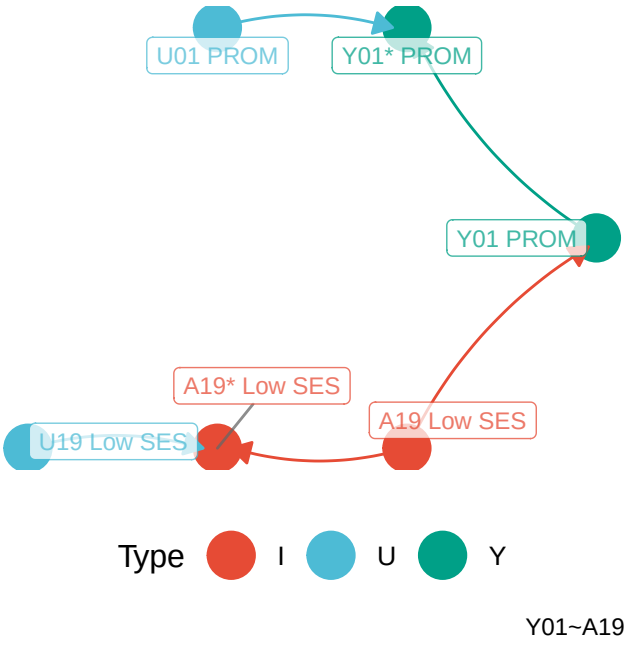

Figure 10: Low socio-economic status (SES)

Maternal age

We could find maternal age as a common cause of PROM and multiple pregnancy/APH/ periodontal disease. Obviously, there is no common cause of PROM in a causal model with maternal age as the variable of interest (Song J, et al 2019). We included this variable exclusively in the causal model (Figure 11).

Influenza

Similar to maternal age, obviously there is no common cause of PROM in a causal model with influenza as the variable of interest (Littauer EQ, et al, 2017). This disease has a specific agent. Therefore, the PROM causal model only included this disease (Figure 12).

Prognostic prediction of premature rupture of membranes

Predictive performances for classification task were already clearly described in the main text. The Source Data Spreadsheet for Figure 3 in the main text is available. In this Supplementary Information, parameter estimates in each model is described. However, these were not always straightforward because of the complexity of several models.

For causal ridge regression (RR), the estimates were weights or beta values, as commonly described in a regression model (Table 14 in Supplemental Spreadsheet). For classification task, top three highest weights

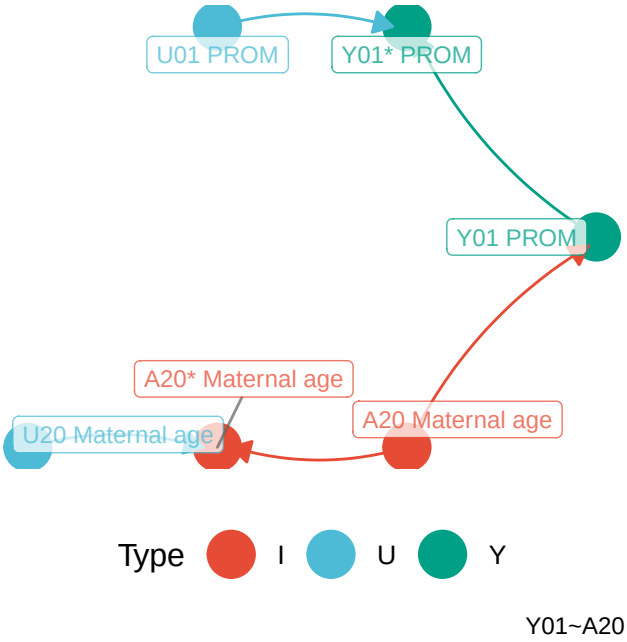

Figure 11: Maternal age

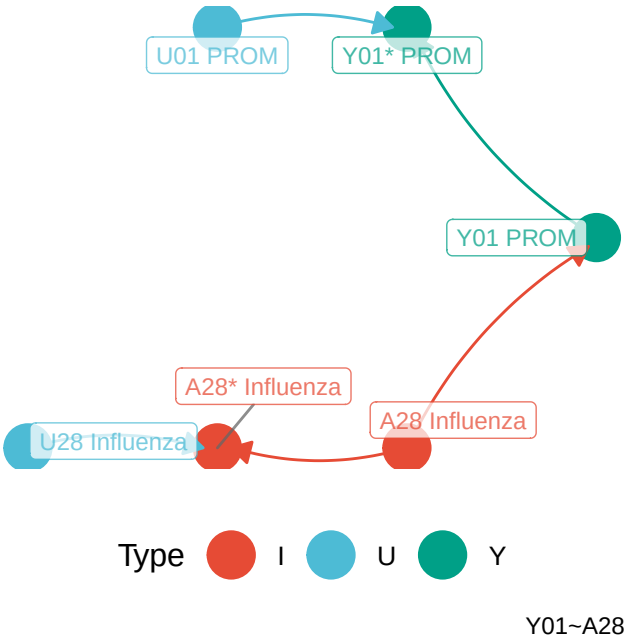

Figure 12: Influenza

were assigned for chorioamnionitis, IAI, and GTI. This is similar to the effect ranks in causal inferences by IPW.

Three models used principal components (PCs), including PC elastic net regression (PC-ENR). To transform predictors into a PC, each predictor has a weight to multiply with. These weights were also shown (Table 16 in Supplemental Spreadsheet). For the PCs themselves, the parameter estimates in PC-ENR were similar to those in causal RR. The weights in PC-ENR were also shown (Table 16 in Supplemental Spreadsheet).

For PC random forest (PC-RF) and PC gradient boosting machine (PC-GBM), the parameter estimates may be represented by variable importance. It is a proportion of learners (trees) that include a predictor. These numbers were also shown (Table 17 and 18 in Supplemental Spreadsheet).

For DI-VNN, we filtered predictors by differential analysis which consisted of multiple univariable linear regressions. The parameter estimates were expressed as log of fold changes. These were equivalent to log of odds ratios. This number and others, including false discovery rate (FDR), were also shown for the selected predictors by FDR <0.05 (Table 19 in Supplemental Spreadsheet).

Parameter estimates of the DI-VNN were extremely enormous. To get similar sense with those of the other models, we used an intermediate output at a layer after being fed to Inception v4 for each ontology. This showed a learning representation by DI-VNN on a predictor. We have described how these were computed in the main text. The intermediate outputs were shown (Table 20 in Supplemental Spreadsheet). We have already pointed out several meanings of these outputs in the main text for both population and individual levels. In addition, connections between ontologies are also described (Table 21 in Supplemental Spreadsheet).

For comparison of our models with those of previous studies, we applied methods in the preferred reporting items for systematic reviews and meta-analyses (PRISMA) 2020 expanded checklist (Table 4 in Supplemental Spreadsheet). From three literature databases and several steps, we found two prediction models as described in the main text (Table 5 in Supplemental Spreadsheet). The steps are shown (Figure 13).

Estimation of the time of delivery

Estimation performances were already clearly described in the main text. The Source Data Spreadsheet for Figure 4 in the main text is also available. Parameter estimates of the estimation models were also shown in the same tables with those of the classification models (Table 14 in Supplemental Spreadsheet). For estimation task, which implicitly related to preterm delivery, the lowest (negative) weights in causal RR were also assigned to chorioamnionitis, IAI, and GTI, but, the second lowest rank was shifted up by multiple pregnancy. Lower-rank weight means earlier time of delivery, which is, likely having more chance to be a preterm delivery. It turns out our decision to choose IAI and chorioamnionitis as the causal factors, instead of the effects of PROM, is consistent with these findings.

Although causal RR mostly had achieved highest proportion of criteria fulfilled among the models by external validation sets, we noticed this only applies for the predicted nonevents based on visual assessment. The causal factors may be well-generalized to estimate the time of delivery under nonevents predicted (modeled) by DI-VNN. Nevertheless, we selected PC-RF as the best model for estimation task of the time of delivery.

Challenge on estimation task is reasonable considering different distributions among internal and external validation sets. In internal validation set, of which we only utilized the calibration set for model evaluation, predicted events by DI-VNN happened 15 weeks (95% CI 11 to 18; n=760) on average from the time of prediction. This was similar to those in external random (18, 95% CI 14 to 21; n=973) and temporal splits (15, 95% CI 12 to 19; n=687), but later than those in external geographical (9, 95% CI 6 to 13; n=500) and geotemporal splits (3, 95% CI 1 to 5; n=157). Meanwhile, the pregnancies predicted as nonevents by DI-VNN ended at 40 weeks (95% CI 39 to 41; n=20,746) on average from time of prediction. This was earlier than those in external random (42, 95% CI 40 to 43; n=25,959), temporal (52, 95% CI 50 to 53; n=17,533), and geotemporal splits (60, 95% CI 54 to 65; n=2,067), but later than that in external geographical split (37, 95% CI 35 to 38; n=15,318).

Exploring deep-insight visible neural network

Population-level data exploration is briefly described in the main text. For interactive figure and table of DI-VNN, we provide these in our web application (https://predme.app/promtime). In this Supplementary Information, we described more technical details on the DI-VNN exploration. The Source Data Spreadsheet for Figure 5 in the main text is available.

Unlike other agglomerative hierarchical clustering algorithm, a child ontology term in CliXO may connect to more than one parent ontology term. For each edge, we applied several blocks of state-of-the-art convolutional neural network (CNN) architecture, i.e. Inception v4-Resnet. From deep (bottom) to surface layer (nodes), the AUROCs were getting higher. We computed this metric using parts of the architecture up to each node. This number also contributed to the parameter updates of this model (Equation 5 in the main text). A node is an ontology term in the form of a three-dimensional array of which non-zero values are initially coming from the feature members of that ontology term.

A child ontology array has a similar value distribution with the parent one is because a neural network applies a backpropagation algorithm that updates the model parameters consecutively from the surface to the deeper layer following the path, which is the ontology network of DI-VNN. But, unlike other neural network models, DI-VNN isolates the backpropagation effect; therefore, we can trace the array values to interpret the possible meaning.

The dimensional reduction algorithm, which is t-moderated stochastic network embedding (t-SNE) using Barnes-Hut approximation, mapped features on high dimensional to lower dimensional space, as multiple localities. This algorithm spreads small clusters instead of making a large bubble of clusters; thus, we expected our CNN algorithm can be better extracting predictive features from these localities. If a feature nearer to one than another, this means there is a closer relationship between both features. The localities clustered by t-SNE are grouped together at the root node on the most superficial layer (Figure 5b in the main text). Deeper layers have different subsets of features separated by the ontology grouping.

A node on more superficial layer, which is ONT:155, consisted a feature that prefers the same outcome that N760 (acute vaginitis) and causal_A03 (chorioamnionitis) prefer, which is 598 (urethral catheterization). This feature is semantically related to acute vaginitis because the anatomical sites are adjacent. But, since ONT:155 is on a more superficial layer, this node will connect to the same node with many features from other ontology terms. This means more factors may need to interact with urethral catheterization (598) to be predictive for PROM. In addition, within the same ontology term, there is also 8602 (injection or tattooing of skin lesion or defect). Apparently, the CliXO algorithm have clustered these features together semantically, which are similarly invasive procedures.

From ONT:167 and ONT:149 on the deeper layer, we can find unusual features in the context of PROM, which are H527 (unspecified disorder of refraction) preferring nonevents while 734 (flat foot), H521 (myopia), and H522 (astigmatism) preferring events. These codes might be responses to the subject symptoms of edema in the feet and blurry vision. Both symptoms in a pregnant woman may be typically associated with severe preeclampsia. But, a doctor may avoid this association if the context does not support the symptoms, e.g. symptoms by a non-pregnant subject. This may lead a doctor to assign these codes responding to those symptoms. For each prediction, a human user may need to explore the model to avoid misclassification by ignoring the prediction if it counters the clinical reasoning. More pragmatically at individual level, the predictive value may not be sufficient or the estimation may not be quite precise based on the corresponding subpopulation data with respectively the same predicted outcome or estimated time of delivery. In addition, albeit all of the population-level patterns from this exploration, every subject may reveal a different pattern using the same DI-VNN model, as described in the next section.

Web application

The Source Data Spreadsheet for Figure 6 in the main text is available. After the application was done (Figure 6 in the main text), we tried to adjust the threshold to the maximum value, such that the data for population-level performances are still available and the predictive value is also maximized depending

on the prediction result. The population-level data was the same with internal validation (calibration split; n=21,506) which was used to plot the ROC curve (Figure 3b in the main text). The threshold was 0.67 with positive predictive value of 0.809 (95% CI 0.798 to 0.82). The sensitivity was reasonably low (0.107, 95% CI 0.104 to 0.11) if using this threshold. But, from a standpoint of prediction at individual level, a precise estimation is important to determine whether a decision corresponding to this prediction can be made with a good confidence. By default, the threshold is set at an optimum value of 0.14 (sensitivity 0.494, 95% CI 0.489 to 0.5). Based on the predicted probability case-by-case, a user can decide the threshold to adjust at.

From the reported timeline (Figure 6 in the main text), this subject is shown predicted to deliver on October 18ˆth 2016, approximately. If the predicted outcome is not PROM, the shaded area would be red; otherwise, turquoise color is applied to the area as shown. It depicted population-level estimation of true time of delivery for subjects that were also estimated to deliver within 11 weeks and predicted as PROM by the same threshold. The population-level data was the same with internal validation (n=107,536) which was used to plot the PC-RF estimation window (Figure 4b in the main text). By population-level estimation, the time of delivery might be at the beginning up to the end of October. Using threshold at 0.67, this population-level estimation was shifted earlier for a week compared to that at 0.14. In addition, for illustration purpose, just like a real-world setting, say we know the gestational age was 22-23 weeks' gestation based on last menstrual period and ultrasound examination. If this estimation model is precise for this case, the subject would deliver at 33-34 weeks' gestation, which is a preterm PROM.

We also saw the medical history of this subject from the reported timeline (Figure 6 in the main text). Up to the date of prediction, the prediction model used these features: (1) A09 (diarrhea and gastroenteritis of presumed infectious origin); (2) J069 (unspecified acute upper respiratory infection); (3) K30 (dyspepsia); and (4) 8878 (diagnostic ultrasound of gravid uterus). On the timeline, these were ordered from the most positive (top) to the most negative (bottom) based on each output in the ontology array.

In the prediction model, any of these features were classified in the ontology arrays of ONT:158, ONT:169, ONT:176, and root, as depicted by the ontology network (Figure 6 in the main text). We also identified the deepest ontology that includes all features in the timeline, which is ONT:169, but the predicted outcome based on this ontology is not PROM using the same threshold with that of the root. A user also can see a predictive performance of any ontologies, just like AUROC of the root (0.714, 95% CI 0.712 to 0.716). It is computed for the pre-calibrated DI-VNN only, since the calibrated one only used the predicted probability that was the output convoluted from the root ontology array.

Negative values at population level tend to event, as described in the previous section. From the ontology array (Figure 6 in the main text), J069 (unspecified acute upper respiratory infection) tends to event in that array, as shown as mostly negative outputs in the timeline. This feature was also surrounded by more negative outputs. A09 (diarrhea and gastroenteritis of presumed infectious origin) also had a negative value, but, this feature and K30 (dyspepsia) were surrounded by more positive outputs in the ontology array. If we apply the same illustrative gestational age, these infectious diseases (J069 and A09) were diagnosed respectively 10 and 4 weeks the start of pregnancy, as shown on the timeline. Root is the only ontology that predicted PROM and included all features in this subject. This is implied all of these features should be taken together for the prediction. In addition, one may question why J069 (unspecified acute upper respiratory infection) tends to event while influenza have the opposite effect (OR 0.995,95% CI 0.993 to 0.997; Table 13 in Supplemental Spreadsheet). We found that influenza, which is causal_A28, did not include J069 (Table 12 in Supplemental Spreadsheet). This implied specific acute upper respiratory infection, such influenza, may not have the same effect with that by the unspecified one on PROM.

Beyond the root, the array of ontology ONT:169 is also shown (Figure 14). A09 and J069 had negative values. As described in the main text, these features tend to an event. Respectively, the surrounding positive and negative values were subtler in this ontology array. Since this filtered array is fed forward to convolutional layers to be reduced each time passing a layer (see Supplemental Video), the value of A09 and J069 were averaged along with the adjacent values toward zero, opposite to event. In turn, this coincides with lower AUROC in ONT:169 compared to that in root.

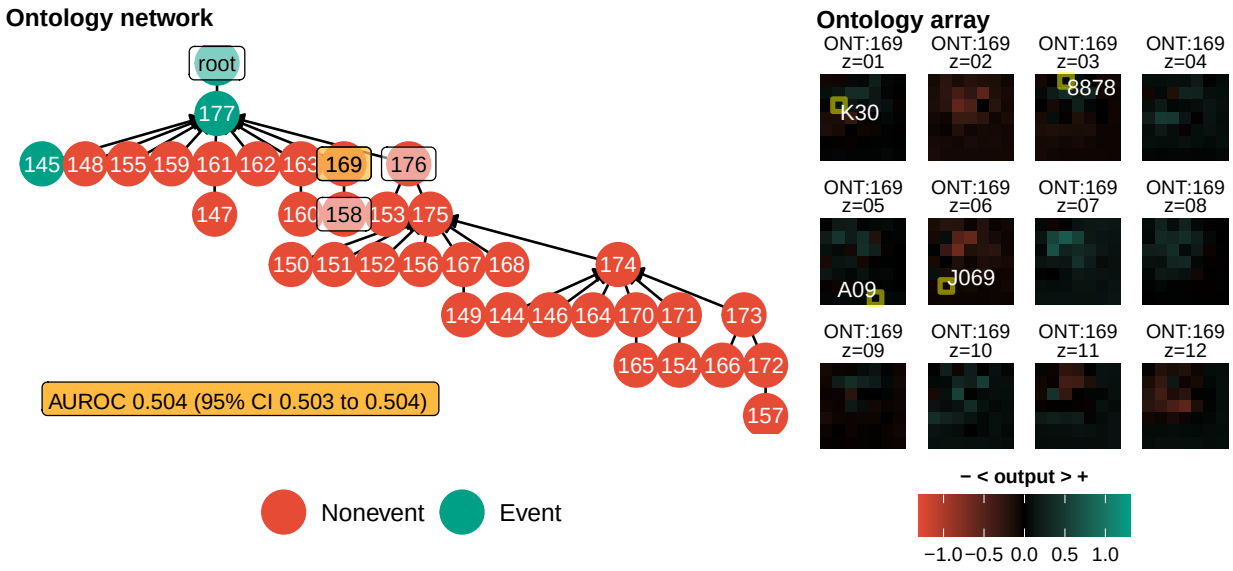

Figure 13: Ontology ONT:169

Discussion

From causal inference and feature extraction to model selection and exploration, we only used an internal validation set. But, the model is evaluated using four external validation sets with a large sample size (8,778 visits and 3,352 subjects for events only). This has found the DI-VNN prediction was robust and the PC-RF estimation was precise within a reasonable time window. All of these models used only a medical history of a patient, which is easily extracted from the electronic medical record systems of most healthcare providers worldwide. Neither a biomarker testing nor even a laboratory test is needed. Eventually, the best models are ready to use for any healthcare providers using an open-access web application without changing their electronic health record systems and revealing private data.

There were four novel approaches in this study. First, we extracted a medical history as a quantity (Equation 1 in the main text) that is sensible to be differential through time for affecting a future health state. We applied Kaplan-Meier estimator to compute a historical rate with linear interpolation. A medical history of a patient is often isolated within a healthcare provider, not allowing other healthcare providers to utilize it to improve patient outcome. By population-level historical rate, the isolated data are possible to utilize indirectly for each condition in a medical record of a patient. We also resolved a resampling method for PCs that are used for prediction (Equation 4 in the main text). To the best of our knowledge, most prediction models that used PCs, if not all, did not apply resampling when inferencing the PCs. Furthermore, it is not seldom to infer PCs before splitting the dataset into training and validation subsets, which exposed to a risk of overfitting. It may cause a prediction model, that uses PCs, is not robust for samples beyond the training set. This situation is reasonable because PCs are typically used for statistical inference and unsupervised machine learning instead of a prediction model or supervised machine learning. Resampling or validation splitting are more well-known in the latter task. Eventually, we also introduced a pipeline to construct a data-driven CNN architecture using non-image data with feature selection by differential analysis that deals high-dimensionality problem. DI-VNN also enables a human user to critically appraise and explore the prediction result case-by-case using extra data knowledge. This is similar to a common, long-standing practice for a doctor to learn and interpret clinical data for assessment of a patient's condition. In addition, we applied a systematic human learning applying a modified snowball sampling (Algorithm 1 in the main text), This was intended to draw a causal diagram of PROM as the assumption and used statistical learning to verify that assumption using our data by IPW, one of G-methods for causal inference on time-varying exposure data.