FINAL REPORT

UTILIZING DATA FROM VARIOUS PARTNERS IN A DISTRIBUTED MANNER

Prepared by: Qoua L Her, PharmD, MS,[1] Yury Vilk, PhD,[1] Jessica Young, PhD,[1] Zilu Zhang, MS,[1] Jessica Malenfant, MPH,[1] Sarah Malek, MPPA[1], Darren Toh, ScD[1]

Author Affiliations: 1. Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute

October 24, 2018

The Sentinel System is sponsored by the U.S. Food and Drug Administration (FDA) to proactively monitor the safety of FDA-regulated medical products and complements other existing FDA safety surveillance capabilities. The Sentinel System is one piece of FDA's Sentinel Initiative, a long-term, multi-faceted effort to develop a national electronic system. Sentinel Collaborators include Data and Academic Partners that provide access to healthcare data and ongoing scientific, technical, methodological, and organizational expertise. The Sentinel Coordinating Center is funded by the FDA through the Department of Health and Human Services (HHS) Contract number HHSF223201400030I. This project was funded by the Office of the Assistant Secretary for Planning and Evaluation (ASPE) and the Food and Drug Administration (HHSF223201400030I/HHSF22301006T).

Final Report

Utilizing Data From Various Partners In A Distributed Manner

Table of Contents

ACRONYMS USED IN THIS REPORT

Abbreviation	Definition
ASPE	Office of the Assistant Secretary for Planning and Evaluation
BMI	Body Mass Index
CIDA	Cohort Identification and Descriptive Analysis
DDNs	Distributed Data Networks
DMC	DataMart Client
DP	Data Partner
DRA	Distributed Regression Analysis
FDA	Food and Drug Administration
FISMA	Federal Information Security Management Act
GLMs	Generalized linear models
GLORE	Grid Binary LOgistic REgression
HHS	Department of Health and Human Services
IRLS	Iterative reweighted least squares
NIH	National Institutes of Health
PCORnet	Patient-Centered Clinical Research Network
SAS	Statistical Analysis Software
SOC	Sentinel Operations Center
SPARK	Secure Pooled Analysis across K-sites
SSCP	Sum of squares and cross products
WebDISCO	Web-based Distributed Cox Regress
WebGLORE	Web Grid Binary LOgistic REgression

I. EXECUTIVE SUMMARY

To obtain sufficient sample sizes and valid effect estimates, many studies require pooling of individual-level data from multiple data sources. Concerns about disclosure of sensitive individual-level and institution-level information have limited this collaboration. However, data organized in distributed data networks (DDNs) combined with use of privacy-protecting analytic methods such as distributed regression analysis (DRA) may alleviate these concerns. Researchers have previously demonstrated the feasibility of using DRA to perform multivariable-adjusted regression analysis and produce results statistically equivalent to results from pooled individual-level data analysis without sharing of individual-level information in controlled or simulated DDNs. However, the implementation of DRA in practice is challenging as convergence of some regression models requires iterative information exchanges between Data Partners and an analysis center. These iterative exchanges are resource-intensive and require extensive coordination. There is a critical need for developing an automatable information exchange process that accounts for the heterogeneity of technological configurations and software requirements among Data Partners in DDNs to facilitate routine use of DRA.

In this project, we assessed the feasibility of developing a pilot, automatable DRA query workflow in PopMedNet, an open-source query distribution software application that currently supports numerous DDNs, including the Sentinel System. Through an iterative process, we analyzed the existing functionalities in PopMedNet and mapped out software designs and query workflows that would allow automatable DRA in horizontally partitioned DDNs, a data environment in which different databases include information about different individuals. We developed DRA algorithms for three regression models (linear, logistic, and Cox proportional hazards) in Base SAS and SAS/STAT and tested the algorithms and query workflow using simulated datasets, two publicly available datasets, and data from three Sentinel Data Partners. We also explored the feasibility of performing linear DRA within vertically partitioned DDNs, a data environment in which different databases include information about the same individuals, using a publicly available dataset. For the proof-of-concept vertical DRA, we assumed that a "primary key" existed to virtually link the records of the same individual between the databases.

We found that it is possible to perform automatable, routine DRA in horizontally partitioned DDNs that employ PopMedNet as their distributed data-sharing platform. A three-step process framework is required to perform DRA in these DDNs: 1) assemble a de-identified individual-level analytic dataset at each Data Partner site, which can be done using a distributed program developed by the analysis center, 2) distribute a DRA package to each Data Partner for local iterative regression analysis through PopMedNet, and 3) iteratively transfer intermediate statistics files between Data Partners and the analysis center through PopMedNet until the model converges or a pre-specified maximum number of iterations is reached. We successfully implemented a pilot DRA query workflow in the Sentinel System. Our internal and external tests consistently produced statistically equivalent regression parameter and standard error estimates to those from the pooled individual-level data analysis. External tests with the three select Sentinel Data Partners showed that DRA could be completed in under 20 minutes, excluding the time required to assemble the analytic dataset at each Data Partner. Factors that determined the execution time included the regression model type, time required to complete one iteration, and number of iterations required to reach model convergence. We also found the PopMedNet DRA query workflow can be used to conduct DRA within vertically partitioned data environments. However, additional enhancements are required to integrate vertical DRA algorithms with the workflow.

Overall, we were able to develop and pilot a PopMedNet-based DRA query workflow in the Sentinel System. The DRA query workflow poses minimal disruptions to the current workflows and require

minimal modifications to existing hardware configurations and software requirements of Data Partners. Importantly, the workflow 1) is agnostic to regression model types, 2) allows users to specify different levels of workflow automation (completely manual, semi-automated, and fully automated) to accommodate diverse perspectives towards automation, and 3) and is agnostic to statistical software.

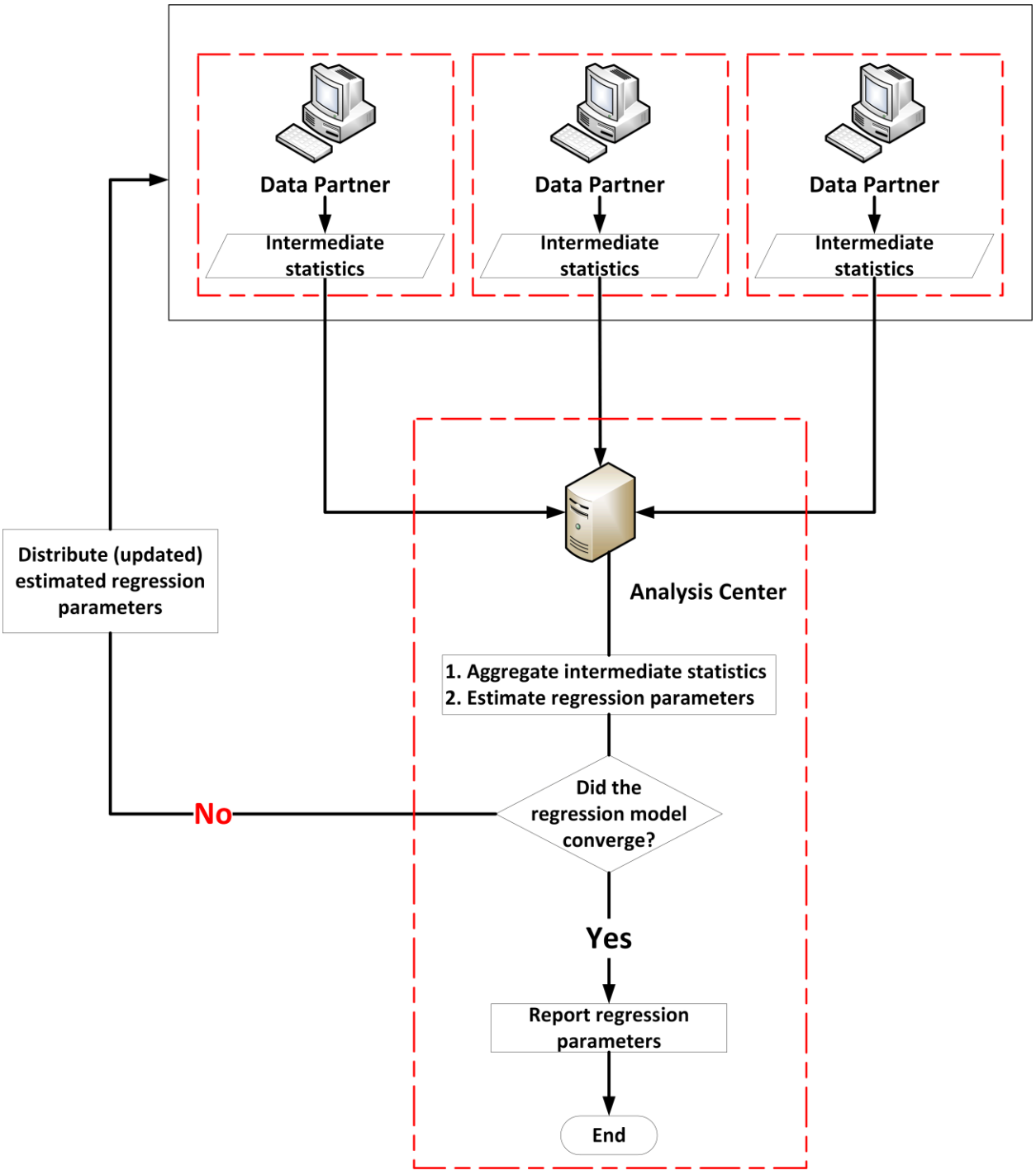
II. INTRODUCTION

Many studies require pooling of patient-level information from multiple data sources to obtain sufficient sample sizes and more generalizable findings. Concerns about data security, patient privacy, unapproved uses of data, and disclosures of proprietary information have limited multi-database collaborations [1-3]. Data organized in a distributed data network (DDN), which allows Data Partners to retain physical control of their data while making analysis more secure and feasible, serves as an alternative to mitigate these concerns [1-3]. Several DDNs already exist and have been used to investigate a wide range of clinical and scientific inquires, including the Centers for Disease Control and Prevention's Vaccine Safety Datalink [4], the National Institutes of Health (NIH)'s Health Care Systems Research Collaboratory [5], the U.S. Food and Drug Administration (FDA)'s Sentinel System [6], and the Patient-Centered Outcomes Research Institute's National Patient-Centered Clinical Research Network (PCORnet) [7].

Although simple descriptive and inferential analysis can be done with summary-level information (e.g., 2x2 tables of exposed and unexposed person-times and outcome events) in these networks, more complex statistical analysis has traditionally required sharing of patient-level information. In recent years, researchers have developed and applied several new analytic methods, including meta-analysis of site-specific effect estimates and methods that leverage confounder summary scores (e.g., propensity scores), to perform complex statistical analysis using only summary-level information [8-14]. Although these more privacy-protecting methods allow one to conduct more sophisticated analyses while preserving patient privacy and data security, they are not without limitations [9]. Specifically, none of these methods allows multivariable-adjusted outcome regression analyses that are more familiar to some stakeholders using only summary-level data pooled across databases.

Distributed regression is a suite of methods that can perform multivariable-adjusted regression analysis in a multi-database setting without sharing of patient-level information while still producing statistically equivalent results as if the databases were pooled [13 15]. A distributed regression analysis (DRA) involves participating Data Partners computing intermediate statistics (e.g. design and information matrices) required for a regression analysis that utilizes data from all participating sites and returning only these intermediate statistics to an analysis center **(Figure 1)**. The analysis center then aggregates the intermediate statistics from all participating sites and generates updated regression parameters from the aggregated intermediate statistics. If the updated regression parameter estimates fulfill the convergence criterion, they are retained as final regression parameter estimates of the model. If they do not fulfill the convergence criterion, the updated regression parameter estimates are distributed to the Data Partners to re-calculate the intermediate statistics. This iterative process of computing intermediate statistics of the distributed data, returning intermediate statistics to the analysis center for aggregation, computing regression parameter estimates, and evaluating for model convergence continues until the convergence criterion is met or a pre-specified maximum number of iterations is reached. This process is mathematically equivalent to performing regression analysis on pooled patient-level data and produces statistically equivalent results [11-17]. See Wolfson 2010 for examples about the intermediate statistics and regression parameter estimates produced and shared at each iteration [12].

Figure 1. Iterative process in distributed regression analysis



Additionally, DRA can be used to analyze both horizontally partitioned data (partition of multiple patient groups with the same data attributes, e.g., two administrative claims databases) and vertically partitioned data (partition of data attributes of one patient group, e.g., one administrative claims database and one electronic health record database), while most of the other methods are only useful for the former data environment. The ability to analyze vertically partitioned data is likely the greatest utility of DRA as the U.S. healthcare system is highly fragmented and different data sources can provide different information about the patients. For example, administrative claims databases contain longitudinal patient healthcare encounter information across different delivery systems, while electronic health record databases contain more in depth clinical data in specific delivery systems. The ability to combine these two data sources can address the limitations of each source.

These advantages make DRA a highly desirable analytic method within DDNs. However, the implementation of DRA in practice is challenging as convergence of some regression models common to biomedical assessment (e.g., logistic and Cox regression) is an iterative process that requires frequent information exchanges among Data Partners. These iterations are resource intensive and require extensive coordination among Data Partners. Routine use of DRA will require some automation of this iterative process.

To our knowledge, DRA has not been implemented and routinely used in any large, active DDNs. Previous work has largely involved simulated or controlled distributed environments [12 14 16 18 19]. This project was funded by the Office of the Assistant Secretary for Planning and Evaluation and the U.S. Food and Drug Administration to develop a prototype workflow that supports automatable DRA in PopMedNet™, an open-source query distribution software application that currently supports the FDA-funded Sentinel System and other large DDNs, including PCORnet and NIH Collaboratory.[20] The specific objectives of the project were to:

1) Develop an overall analytic/process framework for DRA in DDNs
2) Develop and test SAS DRA algorithms that are agnostic to file transfer software, and to perform distributed linear, logistic, and Cox proportional hazards regression analysis using simulated horizontally partitioned data
3) Develop a prototype workflow in PopMedNet that is automatable and can iteratively transfer DRA files between Data Partners and the analysis center in DDNs
4) Integrate the SAS algorithms and PopMedNet workflow and beta-test the integration with select Sentinel Data Partners and perform distributed linear, logistic, and Cox proportional hazards regression analysis using real-world data
5) Place the source code and documentation of the SAS algorithms and PopMedNet query workflow in public domain
6) Explore the feasibility of performing DRA with vertically partitioned data in DDNs by developing and testing SAS algorithms to perform distributed linear regression analysis with simulated vertically partitioned data

This report summarizes findings from activities designed to accomplish these objectives.

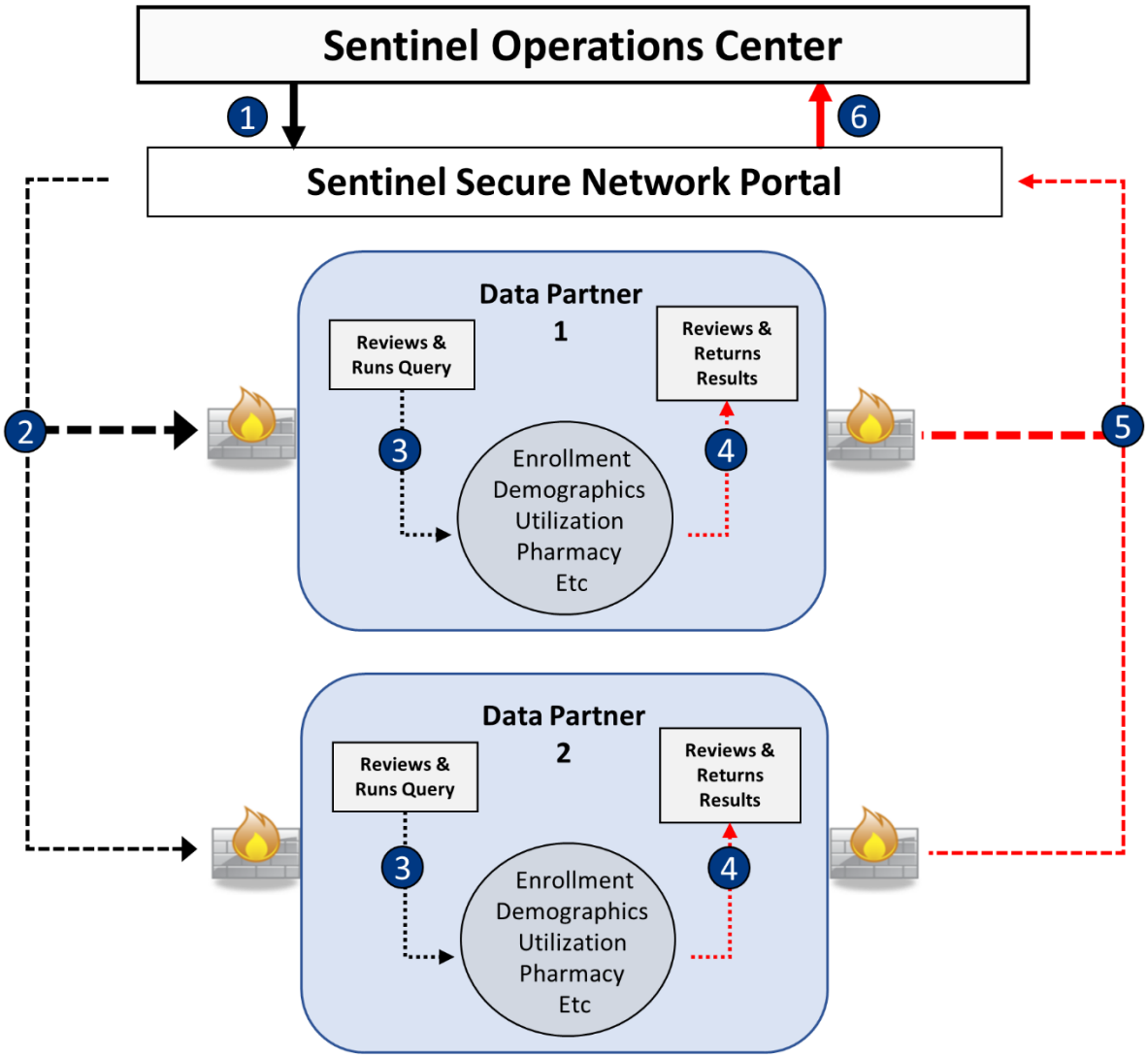
III. METHODS

A. STUDY SETTING – THE FDA SENTINEL SYSTEM

The Sentinel System served as our pilot horizontally partitioned DDN. The Sentinel System is a national surveillance system designed to monitor the safety of approved medical products using routinely collected electronic health data [6 21]. It is one of the first DDNs that employed PopMedNet as their query distribution software. Sentinel has developed a suite of re-usable analytic tools and workflows to allow rapid identification of patient cohorts and conduct of comparative safety analyses in a DDN of 17 Data Partners. The Sentinel network architecture and analytic tools have been adapted to other DDNs [22]. Thus, findings from this pilot could provide important information for the implementation of DRA beyond Sentinel.

All Sentinel Data Partners transformed their data into a common data model [23]. The Sentinel Operations Center (SOC) checks the transformed data for completeness and consistency prior to use. Sentinel has established a standard query fulfillment workflow for routine medical product safety analysis. The process begins with the FDA submitting a safety question to the SOC. A team comprised of FDA and Sentinel personnel defines query parameters such as exposures, outcomes, confounders, and inclusion and exclusion criteria based on established coding systems (e.g., ICD-9-CM, ICD-10-CM, and National Drug Codes). Using the specifications, the SOC (which also serves as the analysis center) assembles and tests a query package written in SAS (SAS Institute, Cary, North Carolina). It then securely distributes the final package to each Data Partner through PopMedNet for local execution on the transformed data **(Figure 2)**. Data Partners produce and securely transfer the requested information, usually in aggregated form, back to the SOC for final analysis through PopMedNet. Detailed patient-level data remains behind the Data Partners' firewalls, protecting patient privacy and data security. Detailed description of the Sentinel query process is available elsewhere [23].

Figure 2. Sentinel's query fulfillment process



1. Sentinel Operations Center (i.e., analysis center) creates and distributes query via the secure network portal (PopMedNet)
2. Data Partners receive notification of the query and retrieve it from the secure network portal
3. Data Partners review and execute query on their local, transformed data
4. Data partners review output
5. Data partners return output, often in aggregated form, to the secure network portal
6. Sentinel Operations Center retrieves results from the secure network portal and performs final analysis

***modified from [23]*

Sentinel's analytic capabilities largely revolve around its ability to rapidly identify cohorts of interest with its pre-tested, customizable Cohort Identification and Descriptive Analysis (CIDA) tool [24]. This tool includes a set of SAS programs that contain editable macro parameters and input files to define query parameters. It offers considerable query customization and analytic flexibility. The tool also has the ability to create a de-identified patient-level analytic dataset to be stored locally at each Data Partner site. The dataset can then serve as an input file for other re-usable Sentinel tools (e.g., propensity score matching and stratification tool) for inferential analysis.

A typical query involves four folders (*sasprogram, inputfiles, dplocal,* and *msoc*), collectively known as the common folder structure. The *sasprogram* and *inputfiles* folders contain the necessary files required for local execution of the analysis on the Data Partner's transformed data. More specifically, the *sasprogram* folder contains the SAS programs and macros while the *inputfiles* folder includes lookup tables, codes, or files used to define the covariates or other parameters of the analysis. The *dplocal* folder houses the de-identified patient-level dataset generated upon successful execution of the CIDA package; this dataset remains behind the Data Partner's firewall. The *msoc* folder stores the output files or dataset (typically summary-level) requested by the query; they are the only files that are transferred to the SOC.

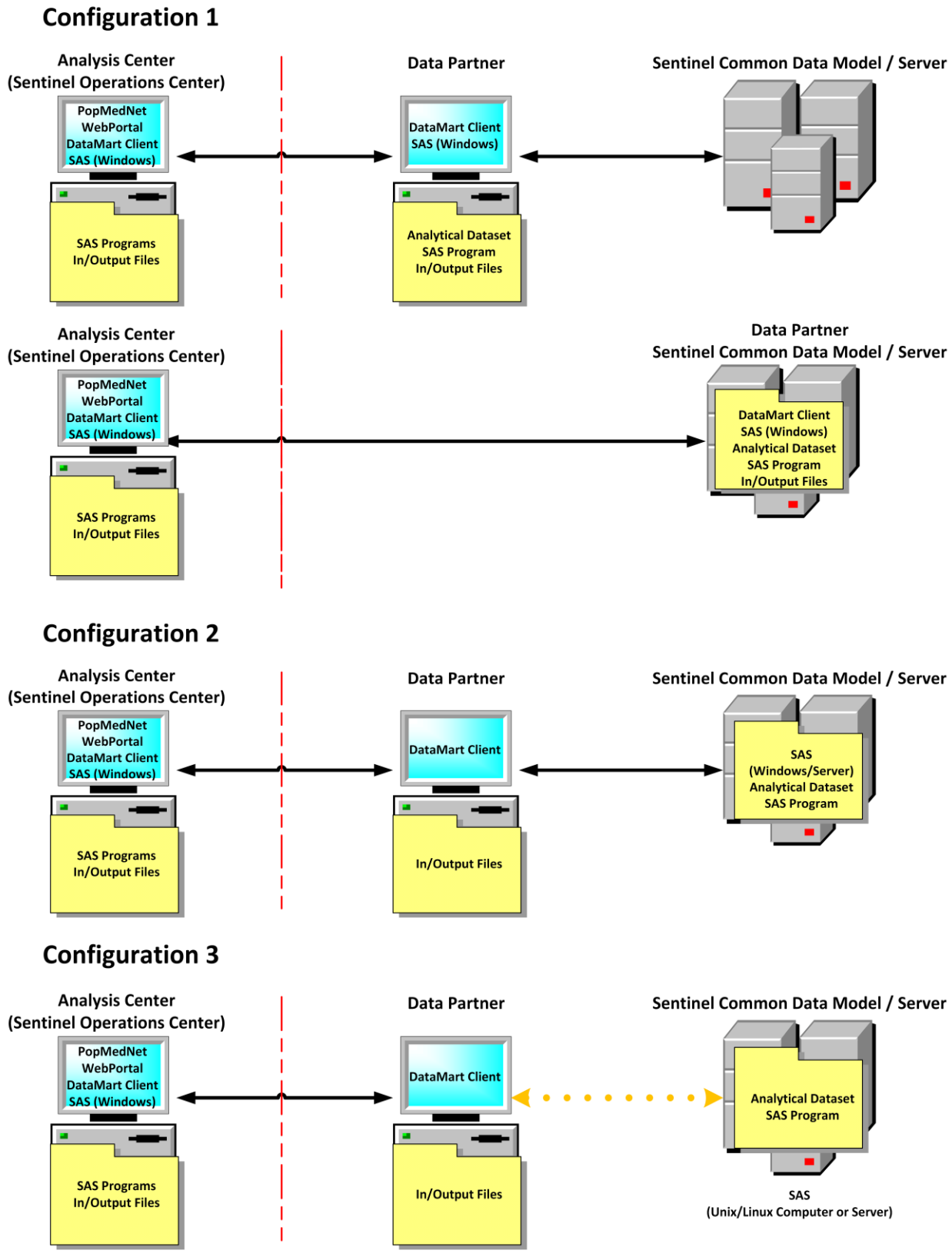
B. POPMEDNET

PopMedNet (http://www.popmednet.org) has been serving as the Sentinel data-sharing platform since 2011 [25]. Two interfaces interlink the network topology of PopMedNet: a web-based network portal and the DataMart Client (DMC). The web-based portal is typically used by the analysis center (e.g., the SOC) to create, distribute, and manage queries. The DMC is a locally installed Windows® application that acts as an inbox for Data Partners to receive query packages and transfer results to the analysis center. All file transfers (query requests and responses) between the Data Partners and the analysis center are achieved through HTTPS/SSL/TLS connections. There are no open ports, Virtual Private Networks, or any external access to Data Partner data, abating concerns about data security and ensuring only approved queries are submitted to and responses returned by participating Data Partners [25-27]. The SOC is a Federal Information Security Management Act (FISMA) compliant data center [28].

C. DATA PARTNER TECHNOLOGICAL CONFIGURATIONS

There are currently three general configurations of the components (DMC, SAS, and the common folder structure) required to fulfill a query (**Figure 3**). As part of the development process, we surveyed all Sentinel Data Partners to catalogue their hardware and software configurations to help guide our DRA query workflow design. Sixteen of the then 18 Data Partners responded to the survey. In five Data Partners, these components were available on the same Windows® desktop computer or server (**Figure 3**. **Configuration 1**). Three Data Partners housed all components on different Windows® machines (**Figure 3. Configuration 2**) and eight had the components installed on different machines with different operating systems (i.e., the DMC on a Windows® desktop computer, while SAS and the common folder structure on a Linux server) (**Figure 3. Configuration 3**). These configurations dictate Data Partners' DMC access to the contents of the common folder structure (e.g., CIDA output). Four Data Partners' DMC had direct access to these contents, while 11 implemented a manual process of transferring the common folders structure to the DMC computer or an accessible drive. Although we could not obtain information from some of the Data Partners, we expect them to fall under one of the three configurations identified.

Figure 3. General configurations of required components (DataMart Client, SAS, and common folder structure) for query fulfillment in Sentinel



D. QUERY WORKFLOW DESIGN AND FRAMEWORK FOR DISTRIBUTED REGRESSION ANALYSIS

An interdisciplinary team of Sentinel investigators that included epidemiologists, computer scientists, programmers, biostatisticians, and informaticians led the project. Through an iterative process, the team analyzed PopMedNet's existing functionalities and mapped out software designs and workflows that would allow automatable DRA process with PopMedNet in Sentinel. The workgroup presented the designs and desired functionalities to the PopMedNet software developers for consultation on feasibility, technical solutions, and timeline.

E. GUIDING PRINCIPLES FOR QUERY WORKFLOW DESIGN FOR DISTRIBUTED REGRESSION ANALYSIS

To maximize the applicability of the final DRA query workflow to other DDNs, we required our design to have minimal disruptions to the current query workflow within PopMedNet and modifications to existing hardware configurations and software requirements of data sources that employ PopMedNet. Our overall goal was to develop an automatable file transfer process that (1) accommodates commonly used regression models (e.g., linear, logistic, and Cox proportional hazards), (2) allows users to specify different levels of workflow automation (completely manual, semi-automated, and fully automated) to accommodate diverse perspectives towards automation, (3) is agnostic to statistical software, and (4) is easily implemented within current workflows.

F. EVALUATION OF THE PERFORMANCE OF THE DISTRIBUTED REGRESSION ANALYSIS QUERY WORKFLOW

We progressively developed and evaluated the DRA query workflow and algorithms in three phases (**Table 1**).

1. A Three-Phase Development and Testing Process

a) Initial Development and Testing

The first phase focused on developing the statistical algorithms (SAS programs) required to conduct DRA in horizontally partitioned data for the three regression model types (linear, logistic, and Cox proportional hazards). We developed the algorithms using Base SAS and SAS/STAT, because SAS is the approved statistical software among all Sentinel Data Partners. We developed distributed linear and logistic regression algorithms using an iteratively reweighted least squares (IRLS) algorithm [12] and distributed Cox proportional hazards algorithms using a Newton-Raphson algorithm [16]. Full details of these algorithms are available in the appendix.

Table 1. A three-phase development and testing process for automatable distributed regression analysis

Test Phase	Dataset	Model Type	Outcome Variable	Covariates	Sample Sizes
Initial development and testing	Boston Housing data* [1129]	Linear Regression	Housing price	Crime per capita, industrialization, and distance to employment centers	$n_1 = 172$ $n_2 = 182$ $n_3 = 152$
		Logistic Regression	Housing price (binary low/high)	Crime per capita, industrialization, and distance to employment centers	
	Maryland State Prison data [30]	Cox Proportional Hazards	Time to re-incarceration	Financial aid, age, and number of prior convictions	$n_1 = 134$ $n_2 = 149$ $n_3 = 149$
Data Partner testing with simulated data	Simulated bariatric surgery data†	Linear	Change in body mass index one year post-surgery	Bariatric surgery exposure, age, pre-index body mass index (BMI), combined comorbidity score, number of ambulatory, emergency department, inpatient, and other ambulatory visits, number of days between last weight or BMI measurement and index procedure, race, sex, year of surgery, and Data Partner site	$n_1 = 1,922$ $n_2 = 1,922$ $n_3 = 1,922$
		Logistic	Weight loss ≥ 20% (within one year post-surgery)		
		Cox Proportional Hazards	Time to weight loss ≥ 20% (within one year post-surgery)		
Data Partner testing with real data	CIDA-based actual bariatric surgery data	Linear	Change in body mass index one year post-surgery	Bariatric surgery exposure, age, pre-index BMI, combined comorbidity score, number of ambulatory, emergency department, inpatient, and other ambulatory visits, number of days between last weight or BMI measurement and index procedure, race, sex, year of surgery, and Data Partner site	$n_1 = 2,728$ $n_2 = 1,018$ $n_3 = 1,706$
		Logistic	Weight loss ≥ 20% (within one year post-surgery)		
		Cox Proportional Hazards	Time to weight loss ≥ 20% (within one year post-surgery)		

CIDA = Cohort Identification and Descriptive Analysis Tool

** Also used in exploration of vertical linear distributed regression analysis*

These two algorithms utilize a semi-trusted third-party as the analysis center to facilitate the required distributed computations. We define a semi-trusted third-party as a party that Data Partners trust with their summary-level data but not with their individual-level data. This analysis center does not share any data from one Data Partner with another without consent. We selected the distributed IRLS and Newton-Raphson algorithms because they both share a similar analytic framework and follow a workflow similar to currently accepted privacy-protecting analytic methods in Sentinel.

We used two different datasets in the development and initial testing of the algorithms. We selected these two datasets because they are available publicly and could be downloaded by others to test our algorithms. Importantly, the first dataset, "Boston Housing data," was originally used by Karr and colleagues to illustrate the theoretical capability of conducting distributed linear regression in a horizontally partitioned data environment [11]. This dataset included 506 observations of Boston medium housing prices and 14 housing or neighborhood characteristics [29]. To stay consistent with these authors, we also partitioned the dataset into three Data Partners of sizes $n_1 = 172$, $n_2 = 182$, and $n_3 = 152$; each dataset included the following continuous variables: housing price, crime per capita, industrialization, and distance to employment centers. Housing price served as the dependent variable while crime per capita, industrialization, and distance to employment centers were the independent covariates for the distributed linear regression model. We also dichotomized housing price into low or high (above or below the median) and used the derived binary variable as the dependent variable for distributed logistic regression analysis. We used a second dataset, "Maryland State Prison data", that included 432 Maryland convicts followed for one year post release for the development and evaluation of the distributed Cox proportional hazards algorithm [30]. We randomly partitioned the dataset into sizes of $n_1 = 134$, $n_2 = 149$, and $n_3 = 149$. Time to re-incarceration (in weeks) was the time-to-event outcome and financial aid (a binary variable), age (a continuous variable), and number of prior convictions (a continuous variable) were the independent covariates.

We created a simulated horizontally partitioned DDN of three Data Partners for internal development and testing by storing the partitioned datasets in three different directories on a Window® network drive (Configurations 1 and 2). All directories were accessible to computers typically used to perform routine Sentinel tasks. These machines operated on Windows® 7 Professional platform, with a dual-core Intel 2.7 GHz processor and 8 GB of RAM. We completed all initial development and testing in this phase with the DMC version 5.7, connected to a test version of the PopMedNet Web Portal version 6.0.

b) Data Partner Testing with Simulated Data

In the second phase, we focused on enhancing the algorithm to accommodate datasets generated from the Sentinel Common Data Model and analytic tools. We simulated a patient-level dataset typical of a CIDA output for a synthetic cohort of adults who received a primary bariatric procedure [31]. We chose the bariatric example because it provided continuous, binary, and time-to-event outcomes for linear, logistic, and Cox regression, respectively. The simulated patient-level dataset was developed to mimic the type of covariates commonly observed in Sentinel queries (e.g., a binary variable indicating whether patients have a certain condition); the values of these covariates and thus the derived parameter estimates did not have any meaningful scientific interpretation. The simulated dataset included variables indicating bariatric surgery exposure, age, pre-surgery body mass index (BMI), combined comorbidity score, number of ambulatory, emergency department, inpatient, and other ambulatory visits, number of days between the last weight or BMI measurement and the index procedure, race, sex, year of surgery, and Data Partner site. We computed a continuous outcome (change in BMI in one-year post-surgery), a binary outcome (weight loss ≥ 20% within one-year post-surgery), and a time-to-event outcome (time to weight loss ≥ 20% within one-year post-surgery) from the simulated data as dependent variables for the three regression model types. We randomly partitioned this simulated

patient-level dataset (n = 5,766) into three smaller datasets of 1,922 patients to simulate a DDN of three Data Partners. We repeated the same internal testing process described in phase one using the partitioned, synthetic datasets.

We also sent the three partitioned datasets to three Sentinel Data Partners (which coincidently represented all three configurations described above) in this phase for external end-to-end testing. We began Data Partner testing with synthetic data to get them familiarized with the automatable DRA workflow without the concerns about sharing their own data. Data Partners stored the partitioned datasets on a Windows® local or network drive accessible to their DMC. Data Partner computers used for testing were those routinely used to perform Sentinel tasks. All Data Partner testing with simulated data were completed with the DMC version 6.5, connected to a test version of the PopMedNet Web Portal version 6.6.

c) Data Partner Testing with Real Data

In the last phase, we focused on integrating the DRA query workflow and algorithms into the Sentinel query fulfillment process and addressing issues that may arise during production. We distributed an actual CIDA request to the same three Data Partners in phase two to generate a bariatric cohort using their real data. The request created three patient-level analytic datasets (n_1 = 2,728, n_2 = 1,018, and n_3 = 1,706) that were stored behind each Data Partner's firewall on a Windows® local or network drive accessible to the DMC. For final testing, the Data Partner with Configuration 3 reconfigured their hardware configuration to Configuration 1. This decision was made to simplify our external development and test environment as Configuration 3 contain additional layers of heterogeneity (sub-configurations) across the network. This workaround has previously been used with other Sentinel tools (e.g. summary tables). We conducted external end-to-end test using the actual datasets for all regression model types with the three Data Partners. All Data Partner testing in this phase was completed with the DMC version 6.5, connected to a test version of the PopMedNet Web Portal version 6.6.

2. Evaluation of Statistical and Operational Performance

We assessed the statistical performance of the DRA query workflow in all three phases of development and testing. To perform the reference analysis, we requested all Data Partners to securely transfer their CIDA output and de-identified patient-level analytic dataset to the SOC for pooling. We compared the DRA results with the results from the pooled patient-level data regression analyses. The DRA algorithms were considered successful if they produced results that were statistically equivalent to the pooled patient-level data analysis.

We documented failed end-to-end workflow tests during internal and external testing. We investigated the root causes for the failed tests and addressed them with the software developers, programmers, and statisticians. We repeated the iterative process of testing, investigating, troubleshooting, and updating the software until a successful end-to-end external test was completed. We evaluated the operational performance of the DRA query workflow for each regression model type during successful end-to-end Data Partner testing with real data in phase three. Specifically, we extracted time stamps of request and routing status changes (e.g. file download and upload, trigger file created, and SAS execution) in the PopMedNet DRA query workflow. From these time stamps, we calculated the average time to complete one DRA iteration for each model type. We also computed and averaged the time elapses to download and upload files, transfer files to the reciprocal party, and execute SAS for each regression model type.

G. EXPLORATION OF DISTRIBUTED REGRESSION ANALYSIS WITH VERTICALLY PARTITIONED DATA

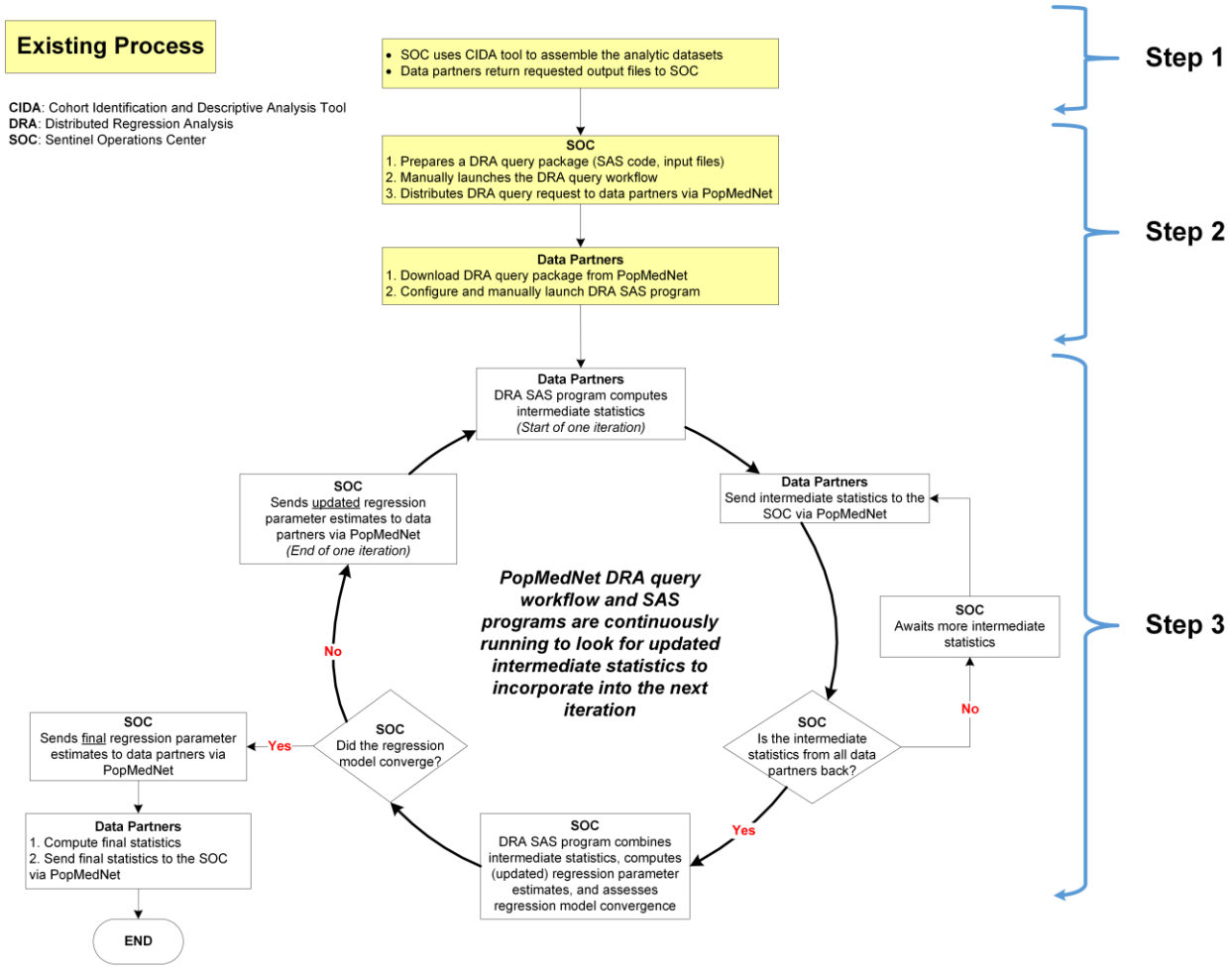
We also explored the feasibility of conducting DRA with vertically partitioned data, a setting in which information from the same individual is recorded in multiple data sources [32 33]. We reviewed different secure multiparty computation protocols for linear regression analysis with vertically partitioned data that may be integrated with PopMedNet. We chose distributed linear regression because it was computationally less complicated than other regression models. Similar to the initial development and testing phases of horizontally partitioned data, we assessed the statistical performance of the computation protocol with the Boston Housing data and the simulated bariatric surgery datasets in our simulated DDN. Both datasets were partitioned into two Data Partners, where one Data Partner held the outcome variable (housing price or change in BMI) and the other Data Partner held the covariates. For each example, a unique patient identifier existed in both Data Partners to allow virtual linkage of the partitioned datasets. The vertical DRA was a proof-of-concept analysis, its integration into the workflow developed for DRA in horizontally partitioned data was beyond the scope of the project.

IV. RESULTS

A. A THREE-STEP FRAMEWORK TO ALLOW AUTOMATABLE DISTRIBUTED REGRESSION ANALYSIS IN POPMEDNET

It is possible to perform automatable, routine DRA in Sentinel and in other horizontally partitioned DDNs that employ PopMedNet. A three-step framework is required to perform DRA in these DDNs: 1) assemble a de-identified patient-level analytic dataset at each Data Partner site, which can be done using a distributed program developed by the analysis center, 2) distribute a DRA package to each Data Partner for local iterative regression analysis through PopMedNet, and 3) iteratively transfer intermediate files between Data Partners and the analysis center through PopMedNet until the model converges or a pre-specified maximum number of iterations is reached (**Figure 4**). Our evaluation determined that the existing Sentinel query fulfillment process and PopMedNet query workflow allowed steps 1 and 2 with minimal modifications. We accomplished step 3 by enhancing the PopMedNet query workflow.

Figure 4. A 3-step framework to conduct automatable distributed regression analysis within PopMedNet™



1. Step 1: Assembling an Analytic Dataset at Each Data Partner Site

In the first step, the analysis center distributes a CIDA package via PopMedNet to assemble a de-identified patient-level analytic dataset at each Data Partner site (**Figure 4. Step 1**). The analytic dataset includes eligible patients and covariates of interest, as specified by the requester. Consistent with the existing Sentinel query fulfillment process, this dataset is stored in the *dplocal* folder and not transferred to the analysis center. This step can also accommodate additional *ad hoc* SAS code to modify or add covariates that are not part of standard CIDA output.

2. Step 2: Distributing a Distributed Regression Analysis Query Package to Data Partners for Local Iterative Execution

In the second step, the analysis center distributes a DRA package to all participating Data Partners via PopMedNet (**Figure 4. Step 2**). This package utilizes the common folder structure to organize the required analytic components for DRA. The *sasprogram* folder includes a DRA SAS program, while the *inputfiles* folder contains initial and subsequent iterative "guesses" of the regression parameter estimates and the required DRA macros. Upon receiving the package, Data Partners unzip the package and execute the SAS program on the analytic patient-level dataset created in Step 1. This SAS program runs continuously.

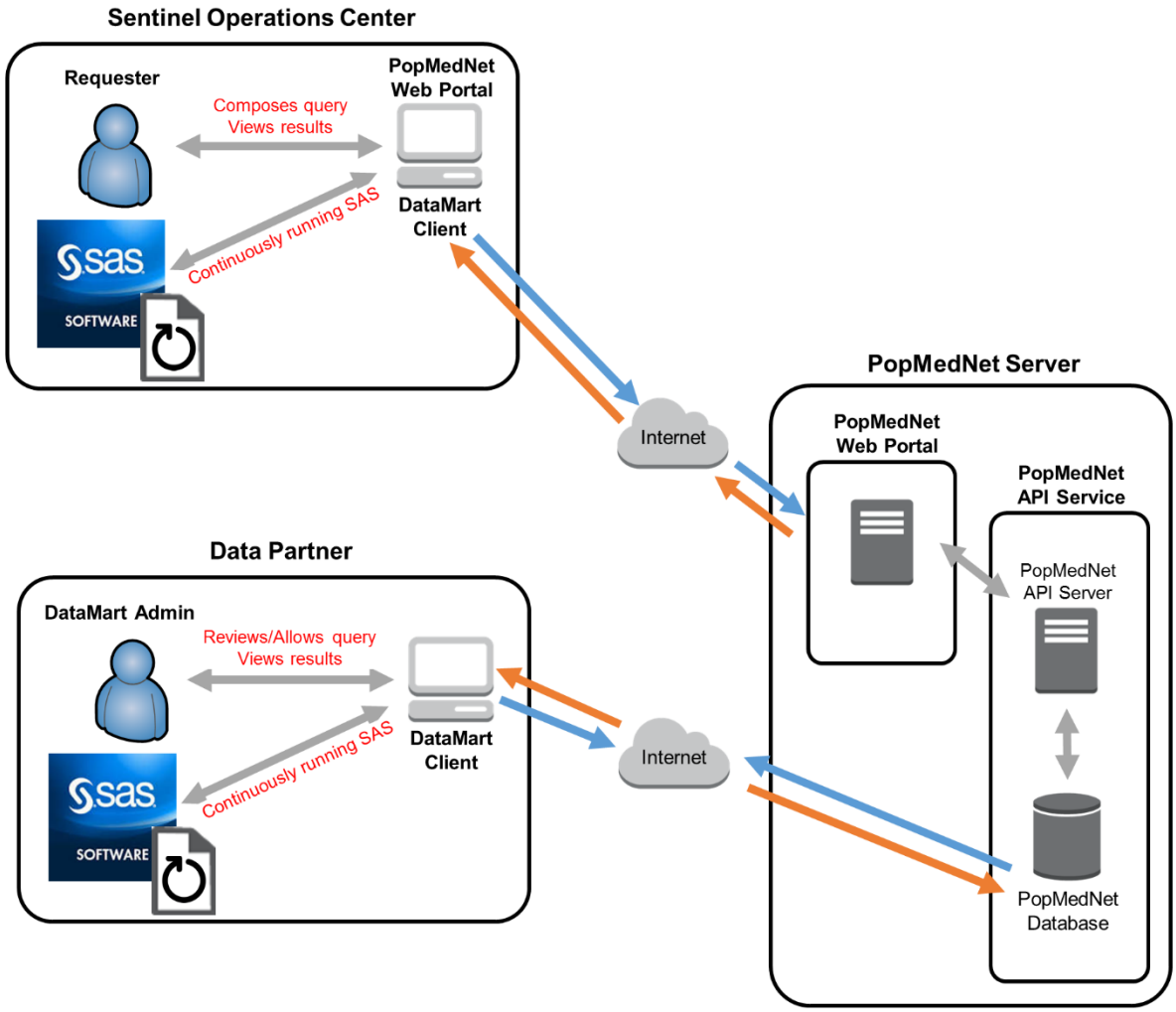
3. Step 3: Iteratively Transfer Files Between Data Partners and the Analysis Center

Successful execution of the SAS program outputs a file that contains intermediate statistics to the *msoc* folder (**Figure 4. Step 3**). Data Partners then upload and transfer the file to the analysis center via PopMedNet. A corresponding SAS program at the analysis center, also running continuously, accepts and aggregates the intermediate statistics from all participating Data Partners. Updated regression parameter estimates are computed and model convergence is evaluated in this step. If the model convergence criteria are not met, updated parameter estimates are re-distributed to the Data Partners via PopMedNet and used as new regression parameter "guesses" in the next iteration. This process of local execution and transferring files between the Data Partners and analysis center continues iteratively until the model converges or a pre-specified maximum number of iterations has been reached.

B. ENHANCEMENTS TO POPMEDNET TO ALLOW AUTOMATABLE DISTRIBUTED REGRESSION ANALYSIS

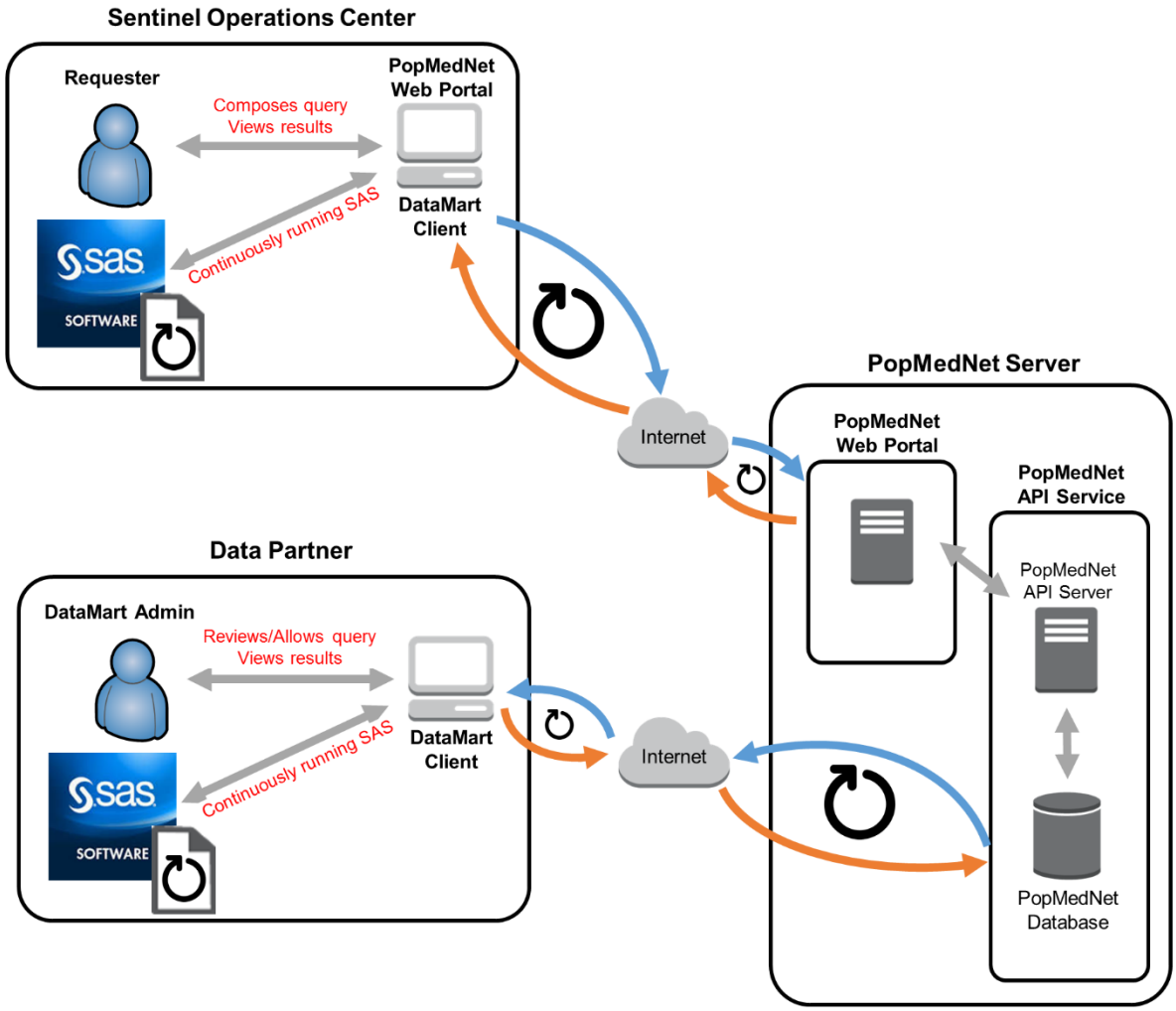
From the PopMedNet query workflow perspective, we can view DRA as a single query request that contains multiple sub-query requests and responses (iterations) looking for the "converging" intermediate statistics. The existing Sentinel query workflow manually supports one sub-query request and response **(Figure 4)**. Manual transfer of files over multiple iterations would be too resource-intensive and restrict the practicability of DRA in DDNs. Therefore, we enhanced the query workflow to allow automatable, iterative transfer of files between Data Partners and the analysis center **(Figure 5)**.

Figure 5. Existing PopMedNet™ query workflow



API: Application programming interface

Figure 6. Enhanced PopMedNet™ query workflow to support automatable distributed regression analysis

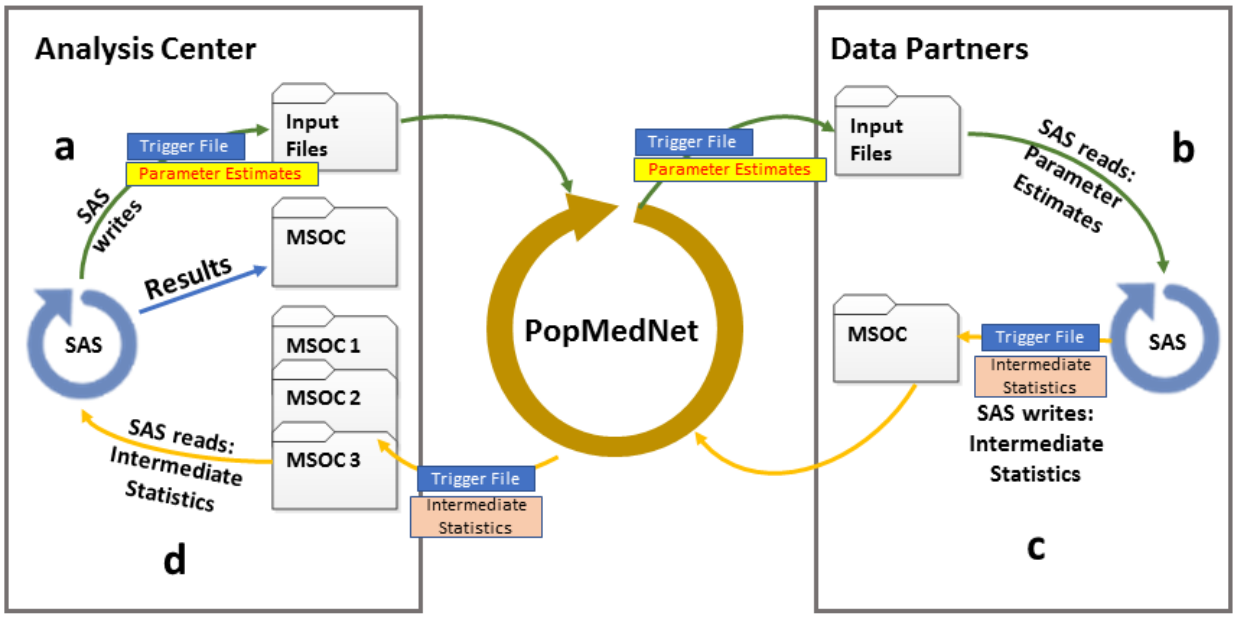


API: Application programming interface

To achieve this, we built a new back-end component referred to as an "*adapter for DRA*" in PopMedNet. This adapter allows the Data Partners and analysis center to have the option to automatically upload files from and download files to pre-defined folders in the common folder structure, when a specific trigger text file appears. To trigger these automatable processes, we built a DMC functionality that monitors pre-defined folders in the common folder structure for the appearance of trigger files into the adapter. In addition, DRA requires iterative distribution of updated regression parameters from the analysis center to the Data Partners. The existing workflow only allows one set of input files per query request. We enhanced this functionality to associate files to each sub-query request, allowing multiple sets of input files to be associated to one DRA query request.

We then integrated the new automatable iterative file transfer process – made possible by the new adapter – with the SAS-driven DRA analytic process. The integration leveraged the existing Sentinel common folder structure and used trigger text files and the newly developed DMC folder monitoring functionality to iteratively and sequentially initiate one DRA process after the other (**Figure 7**). At the beginning of each iteration, the analysis center distributes a SAS parameter dataset, which specifies all necessary parameters for distributed analysis (regression type, the names of the independent and dependent variable, iteration number, convergence status, etc.) and contains initial or new "guesses" of parameter estimates, to each Data Partners' *inputfiles* folder via PopMedNet (**Figure 7. Steps a and b**). PopMedNet then creates a trigger text file (*files_done.ok)* to signal to the continuously running DRA SAS program at each Data Partner site to incorporate the SAS parameter file and the new guesses into their local execution of the program on the de-identified patient-level dataset. Successful execution of the Data Partners' DRA SAS program outputs intermediate statistics to the *msoc* folder along with a trigger text file (*files_done.ok*) (**Figure 7. Step c**). This trigger file signals to PopMedNet that intermediate statistics are computed and ready to be uploaded and transferred to the analysis center. Upon completion of the upload to the DMC, PopMedNet deletes the trigger file from the *msoc* folder. This step ensures that the appearance of a new trigger file in the next iteration will automatically initiate a new file transfer process.

Figure 7. Trigger file and actions that allow automatable distributed regression analysis



Step	Location and appearance of trigger file in the common folder structure	Actions
a	Analysis center: *inputfiles* folder	1. Trigger file notifies PopMedNet that updated regression parameter estimates are available 2. PopMedNet uploads and transfers updated regression parameters to data partners 3. PopMedNet deletes the trigger file
b	Data partners: *inputfiles* folder	1. Trigger file notifies SAS that updated regression parameter estimates arrived from the analysis center 2. SAS inputs the updated regression parameters into the distributed regression algorithm and computes intermediate statistics 3. SAS deletes the trigger file
c	Data partners: *msoc* folder	1. Trigger file notifies PopMedNet that intermediate statistics are available 2. PopMedNet uploads and transfers intermediate statistics to the analysis center 3. PopMedNet deletes the trigger file
d	Analysis center: *msocN* folders	1. Trigger file notifies SAS that updated intermediate statistics arrived from data partners 2. SAS inputs intermediate statistics into the distributed regression algorithm and computes updated regression parameters 3. SAS deletes the trigger file

PopMedNet then transfers the output files from each Data Partner to their designated folder at the analysis center (e.g., *msoc1*, *msoc2…*) (**Figure 7. Step d**). Upon completion of the transfer, PopMedNet creates and deposits a trigger text file into the designated folder. The appearance of the trigger text files prompts the SAS program at the analysis center to: 1) perform model convergence computation using the intermediate statistics, 2) output updated parameter estimates or "guesses" to the *inputfiles* folder, and 3) delete the trigger text files (*files_done.*ok) that initiated the analysis center computation process. Again, this latter step ensures the appearance of new trigger files will automatically initiate a new analysis center computation process in the next iteration. If required, the new regression parameter "guesses" can be re-distributed to the Data Partners to fine tune the intermediate statistics using the described file transfer process.

This process of transferring files and computing statistics at the Data Partners and analysis center continues iteratively until the regression model converges or a pre-specified maximum number of iterations has been reached. When either of these two conditions is met, the SAS program at the SOC outputs a termination trigger text file (*job_done.*ok) to the *inputfiles* folder, PopMedNet then transfers this file to the Data Partners to invoke the SAS programs to compute diagnostic statistics (e.g., residuals, goodness-of-fit, and area under the receiver operating characteristics curves). Again, these statistics are returned to the analysis center in the same matter as described above. The termination trigger file also terminates all SAS programs and the DMC folder monitoring functionality.

C. STATISTICAL PERFORMANCE

We performed over 300 internal and external DRA tests. Table 2a-Table 4c summarize the statistical performance of our DRA query workflow and algorithms. In all development and testing phases the DRA algorithms produced statistically equivalent regression parameter and standard error estimates to those from the pooled patient-level analysis.

1. Phase 1: Initial Development and Testing

Table 2a. Distributed Linear Regression vs. Pooled Patient-Level Linear Regression for Horizontally Partitioned Data (Boston Housing Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Intercept	35.50548	1.57690	35.50548	1.57690	-8.38E-13	2.26E-14
Crime	-0.27283	0.04401	-0.27283	0.04401	4.44E-16	9.92E-16
Distance	-1.01582	0.23259	-1.01582	0.23259	1.09E-13	3.22E-15
Industrialization	-0.73017	0.07229	-0.73017	0.07229	3.54E-14	1.32E-15

Outcome: Housing price; Data Partner sample sizes: $n_1 = 172$, $n_2 = 182$, $n_3 = 152$

Table 2b. Distributed Logistic Regression vs. Pooled Patient-Level Logistic Regression for Horizontally Partitioned Data (Boston Housing Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Intercept	2.49660	0.49057	2.49660	0.49060	1.33E-15	9.99E-16
Crime	-0.14465	0.03686	-0.14460	0.03690	2.04E-13	-2.97E-14
Distance	-0.14105	0.06976	-0.14100	0.06980	1.38E-14	-2.22E-16
Industrialization	-0.13889	0.02376	-0.13890	0.02380	-2.42E-14	-2.19E-16

Outcome: Housing price (low/high); Data Partner sample sizes: $n_1 = 172$, $n_2 = 182$, $n_3 = 152$

Table 2c. Distributed Cox Proportional Hazards Regression vs. Pooled Patient-Level Cox Proportional Hazards Regression for Horizontally Partitioned Data (Maryland State Prison Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Age	-0.06692	0.02084	-0.06692	0.02084	-1.39E-16	2.78E-17
Financial Aid	-0.34644	0.19024	-0.34644	0.19024	2.22E-16	-2.78E-17
Prior Arrest	0.09653	0.02724	0.09653	0.02724	-1.80E-16	1.73E-17

Outcome: Time to re-incarceration; Data Partner sample sizes: $n_1 = 134$, $n_2 = 149$, $n_3 = 149$

2. Phase 2: Data Partner Testing with Simulated Data

Table 3a. Distributed Linear Regression vs. Pooled Patient-Level Linear Regression for Horizontally Partitioned Data (Simulated Bariatric Surgery Data)

Covariates	DRA		Pooled Patient-Level		Difference in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Intercept	-32.02957	0.12094	-32.02957	0.12094	-1.35E-13	-5.07E-14
Exposure	-4.97100	0.02627	-4.97100	0.02627	-9.68E-14	-1.26E-14
Age	0.20075	0.00134	0.20075	0.00134	6.00E-15	-6.44E-16
Pre-index BMI	-0.00110	0.00145	-0.00110	0.00145	2.67E-14	-5.71E-16
Combined Comorbidity Score	0.29462	0.00707	0.29462	0.00707	1.88E-14	-3.40E-15
No. Ambulatory Visits	0.99778	0.00198	0.99778	0.00198	-2.59E-14	-9.55E-16
No. Emergency Department Visits	5.01483	0.01299	5.01483	0.01299	-8.79E-14	-6.26E-15
No. Inpatient Visits	3.02073	0.01315	3.02073	0.01315	-1.10E-13	-6.33E-15
No. Non-Acute Institutional Stay	3.99424	0.01267	3.99424	0.01267	4.31E-14	-6.10E-15
No. Other Ambulatory Visits	1.99394	0.00714	1.99394	0.00714	-8.35E-14	-3.44E-15
Days Between BMI Measurement and Index Procedure	0.20004	0.00025	0.20004	0.00025	1.33E-15	-1.22E-16
Race (Unknown)	1.01827	0.04977	1.01827	0.04977	-3.07E-13	-2.40E-14
Race (American Indian or Alaska Native)	2.03444	0.05008	2.03444	0.05008	-2.75E-14	-2.41E-14
Race (Asian)	2.99764	0.05350	2.99764	0.05350	-1.93E-13	-2.58E-14
Race (Black or African American)	4.08558	0.05331	4.08558	0.05331	-2.18E-13	-2.57E-14
Race (Native Hawaiian or Other Pacific Islander)	5.08583	0.05339	5.08583	0.05339	-1.87E-13	-2.57E-14
Female	2.03663	0.03750	2.03663	0.03750	-3.04E-13	-1.80E-14
Surgery Year (2011)	-0.05015	0.04489	-0.05015	0.04489	-3.11E-13	-2.16E-14
Surgery Year (2012)	-0.05955	0.04455	-0.05955	0.04455	-2.24E-13	-2.14E-14
Surgery Year (2013)	-0.01499	0.04470	-0.01499	0.04470	-2.81E-13	-2.15E-14
Surgery Year (2014)	-0.02904	0.04374	-0.02904	0.04374	-2.61E-13	-2.11E-14
Surgery Year (2015)	-0.00116	0.04798	-0.00116	0.04798	-3.89E-13	-2.31E-14
Data Partner Site (2)	0.03099	0.03219	0.03099	0.03219	-1.46E-13	-1.55E-14
Data Partner Site (3)	-0.01559	0.03221	-0.01559	0.03221	-7.58E-14	-1.55E-14

Reference Groups: Race (White), Surgery Year (2010), and Data Partner Site (1)
Outcome: Change in body mass index one-year post-surgery; Data Partner sample sizes: n_1 = 1,922, n_2 = 1,922, n_3 = 1,922

Table 3b. Distributed Logistic Regression vs. Pooled Patient-Level Logistic Regression for Horizontally Partitioned Data (Simulated Bariatric Surgery Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Intercept	-5.53801	0.32834	-5.53800	0.32830	-1.83E-10	3.71E-12
Exposure	-0.10418	0.06480	-0.10420	0.06480	-3.14E-12	4.40E-13
Age	0.00095	0.00330	0.00095	0.00330	-5.28E-14	2.23E-14
BMI (Pre-Index)	0.13680	0.00505	0.13680	0.00505	4.55E-12	1.01E-13
Comorbidity Score	0.00777	0.01743	0.00777	0.01740	-1.90E-13	1.17E-13
No. Ambulatory Visits	-0.00632	0.00487	-0.00632	0.00487	-1.30E-13	3.24E-14
No. Emergency Department Visits	0.01418	0.03182	0.01420	0.03180	1.64E-13	2.09E-13
No. Inpatient Visits	-0.01667	0.03247	-0.01670	0.03250	-4.91E-13	2.19E-13
No. Non-Acute Institutional Stay	-0.00752	0.03120	-0.00752	0.03120	-9.00E-13	2.07E-13
No. Other Ambulatory Visits	0.01480	0.01749	0.01480	0.01750	-3.41E-13	1.13E-13
Days Prior (Pre-Surgery Vitals Measurement)	0.00009	0.00062	0.00009	0.00062	9.02E-16	4.04E-15
Race (Unknown)	0.24711	0.12055	0.24710	0.12060	3.68E-12	7.86E-13
Race (American Indian or Alaska Native)	0.15906	0.12061	0.15910	0.12060	1.99E-12	7.81E-13
Race (Asian)	0.24098	0.13017	0.24100	0.13020	2.54E-12	8.56E-13
Race (Black or African American)	0.38481	0.13162	0.38480	0.13160	7.65E-12	9.04E-13
Race (Native Hawaiian or Other Pacific Islander)	0.26005	0.12914	0.26000	0.12910	3.70E-12	8.25E-13
Female	-0.03201	0.09265	-0.03200	0.09260	-4.63E-13	6.24E-13
Surgery Year (2011)	0.15400	0.11179	0.15400	0.11180	5.23E-12	7.70E-13
Surgery Year (2012)	0.12471	0.11058	0.12470	0.11060	3.36E-12	7.49E-13
Surgery Year (2013)	0.04451	0.10998	0.04450	0.11000	2.73E-12	7.37E-13
Surgery Year (2014)	-0.13413	0.10636	-0.13410	0.10640	-1.32E-12	6.85E-13
Surgery Year (2015)	-0.19318	0.11596	-0.19320	0.11600	-1.64E-12	7.41E-13
Data Partner Site (2)	-0.05598	0.07970	-0.05600	0.07970	-1.16E-12	5.46E-13
Data Partner Site (3)	-0.04758	0.07923	-0.04760	0.07920	-1.28E-12	5.25E-13

Reference Groups: Race (White), Surgery Year (2010), and Data Partner Site (1)
Outcome: Weight loss ≥ 20% (within one-year post-surgery); Data Partner sample sizes: n_1 = 1,922, n_2 = 1,922, n_3 = 1,922

Table 3c. Distributed Cox Proportional Hazards Regression vs. Pooled Patient-Level Cox Proportional Hazards Regression for Horizontally Partitioned Data (Simulated Bariatric Surgery Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Exposure	-0.04440	0.03128	-0.04440	0.03128	-3.40E-16	-2.08E-17
Age	0.00237	0.00157	0.00237	0.00157	-9.19E-17	-1.52E-18
Pre-index BMI	0.05887	0.00162	0.05887	0.00162	-3.47E-17	2.06E-17
Combined Comorbidity Score	-0.00470	0.00835	-0.00470	0.00835	-1.02E-16	-6.94E-18
No. Ambulatory Visits	0.00003	0.00237	0.00003	0.00237	-1.06E-17	4.34E-19
No. Emergency Department Visits	0.02120	0.01533	0.02120	0.01533	-6.11E-16	2.43E-17
No. Inpatient Visits	0.00101	0.01544	0.00101	0.01544	2.22E-16	-1.39E-17
No. Non-Acute Institutional Stay	0.01487	0.01513	0.01487	0.01513	3.38E-16	1.73E-17
No. Other Ambulatory Visits	-0.00140	0.00854	-0.00140	0.00854	-1.68E-16	0.00E+00
Days Between BMI Measurement and Index Procedure	-0.00004	0.00030	-0.00004	0.00030	-5.90E-18	3.79E-19
Race (Unknown)	0.12243	0.06012	0.12243	0.06012	-1.39E-17	-9.71E-17
Race (American Indian or Alaska Native)	0.06439	0.06076	0.06439	0.06076	2.64E-16	-2.78E-17
Race (Asian)	0.13966	0.06455	0.13966	0.06455	-8.33E-17	-9.71E-17
Race (Black or African American)	0.21981	0.06380	0.21981	0.06380	1.39E-16	-1.25E-16
Race (Native Hawaiian or Other Pacific Islander)	0.14498	0.06440	0.14498	0.06440	1.67E-16	-2.78E-17
Female	-0.01256	0.04451	-0.01256	0.04451	-7.34E-16	7.63E-17
Surgery Year (2011)	0.04461	0.05290	0.04461	0.05290	-3.05E-16	-1.04E-16
Surgery Year (2012)	0.02460	0.05255	0.02460	0.05255	-2.84E-16	-1.11E-16
Surgery Year (2013)	0.02640	0.05303	0.02640	0.05303	-1.04E-17	-1.25E-16
Surgery Year (2014)	-0.09727	0.05246	-0.09727	0.05246	-3.61E-16	-4.86E-17
Surgery Year (2015)	-0.06395	0.05813	-0.06395	0.05813	-6.94E-17	-1.39E-17
Data Partner Site (2)	0.00145	0.03823	0.00145	0.03823	3.72E-16	6.94E-17
Data Partner Site (3)	-0.01885	0.03841	-0.01885	0.03841	5.97E-16	4.16E-17

Reference Groups: Race (White), Surgery Year (2010), and Data Partner Site (1)
Outcome: Time to weight loss ≥ 20% (within one-year post-surgery); Data Partner sample sizes: n_1 = 1,922, n_2 = 1,922, n_3 = 1,922

3. Phase 3: Data Partner Testing with Real Data

Table 4a. Distributed Linear Regression vs. Pooled Patient-Level Linear Regression for Horizontally Partitioned Data (CIDA-based Actual Bariatric Surgery Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Intercept	34.03935	0.61075	34.03935	0.61075	3.66E-12	-9.14E-13
Exposure	2.04714	0.28723	2.04714	0.28723	-4.15E-13	-4.30E-13
Age	-0.03334	0.00837	-0.03334	0.00837	-3.68E-14	-1.25E-14
Pre-Index BMI	-0.99983	0.00050	-0.99983	0.00050	-6.00E-15	-7.44E-16
Combine Comorbidity Score	0.04388	0.06949	0.04388	0.06949	3.59E-15	-1.04E-13
No. Ambulatory Visits	-0.03068	0.01008	-0.03068	0.01008	-6.59E-17	-1.51E-14
No. Emergency Department Visits	0.10329	0.08749	0.10329	0.08749	-2.79E-14	-1.31E-13
No. Inpatient Visits	0.88725	0.25976	0.88725	0.25976	-6.51E-13	-3.89E-13
No. Non-Acute Institutional Stay	1.32338	1.79056	1.32338	1.79056	4.21E-13	-2.68E-12
No. Other Ambulatory Visits	0.02159	0.00873	0.02159	0.00873	1.22E-14	-1.31E-14
Days Between BMI Measurement and Index Procedure	0.01207	0.00567	0.01207	0.00567	3.92E-15	-8.48E-15
Race (Unknown)	0.94212	0.26841	0.94212	0.26841	-4.16E-13	-4.02E-13
Race (American Indian or Alaska Native)	-0.30948	0.69817	-0.30948	0.69817	-2.39E-13	-1.04E-12
Race (Asian)	-0.16853	0.63001	-0.16853	0.63001	-4.52E-13	-9.42E-13
Race (Black or African American)	1.51961	0.29206	1.51961	0.29206	-9.95E-14	-4.37E-13
Race (Native Hawaiian or Other Pacific Islander)	-1.22315	1.04973	-1.22315	1.04973	-4.11E-13	-1.57E-12
Female	-1.22366	0.23205	-1.22366	0.23205	-5.33E-13	-3.47E-13
Surgery Year (2011)	0.15150	0.30361	0.15150	0.30361	-5.94E-13	-4.54E-13
Surgery Year (2012)	-0.24904	0.30372	-0.24904	0.30372	-6.47E-13	-4.54E-13
Surgery Year (2013)	-0.02308	0.30223	-0.02308	0.30223	-6.08E-13	-4.52E-13
Surgery Year (2014)	0.32767	0.30609	0.32767	0.30609	-5.93E-13	-4.58E-13
Surgery Year (2015)	-0.25767	0.33352	-0.25767	0.33352	-6.18E-13	-4.99E-13
Data Partner Site (2)	-1.10559	0.31373	-1.10559	0.31373	2.89E-15	-4.69E-13
Data Partner Site (3)	-0.10990	0.30341	-0.10990	0.30341	-2.07E-13	-4.54E-13

Reference Groups: Race (White), Surgery Year (2010), and Data Partner Site (1)
Outcome: Change in body mass index one-year post-surgery; Data Partner sample sizes: n_1 = 2,728, n_2 = 1,018, n_3 = 1,706

Table 4b. Distributed Logistic Regression vs. Pooled Patient-Level Logistic Regression for Horizontally Partitioned Data (CIDA-based Actual Bariatric Surgery Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors Estimates
	Estimates	Standard Errors	Estimate	Standard Errors		
Intercept	2.11573	0.22833	2.11570	0.22830	-2.27E-13	-4.55E-14
Exposure	-1.06711	0.09895	-1.06710	0.09890	4.03E-13	-1.59E-14
Age	-0.01606	0.00316	-0.01610	0.00316	4.84E-15	-6.85E-16
BMI (Pre-Index)	0.00003	0.00020	0.00003	0.00020	-8.23E-18	-9.49E-19
Comorbidity Score	-0.02623	0.02561	-0.02620	0.02560	1.33E-14	-5.20E-15
No. Ambulatory Visits	0.01155	0.00447	0.01150	0.00447	-4.08E-14	-2.80E-15
No. Emergency Department Visits	-0.06230	0.03132	-0.06230	0.03130	6.99E-14	-6.82E-15
No. Inpatient Visits	-0.12098	0.08940	-0.12100	0.08940	1.39E-13	-1.16E-14
No. Non-Acute Institutional Stay	0.42510	0.78809	0.42510	0.78810	-3.52E-12	-1.18E-12
No. Other Ambulatory Visits	0.00381	0.00340	0.00381	0.00340	-6.49E-16	-1.28E-15
Days Prior (Pre-Surgery Vitals Measurement)	-0.00266	0.00201	-0.00266	0.00201	6.93E-15	-6.26E-16
Race (Unknown)	-0.39685	0.09485	-0.39690	0.09480	4.32E-14	-8.81E-15
Race (American Indian or Alaska Native)	-0.13938	0.26230	-0.13940	0.26230	-3.44E-14	-5.51E-14
Race (Asian)	-0.37257	0.22341	-0.37260	0.22340	1.12E-13	-2.03E-14
Race (Black or African American)	-0.29617	0.10507	-0.29620	0.10510	8.03E-14	-1.20E-14
Race (Native Hawaiian or Other Pacific Islander)	-0.02910	0.40543	-0.02910	0.40540	2.07E-13	-5.88E-14
Female	0.19993	0.08422	0.19990	0.08420	1.58E-14	-1.49E-14
Surgery Year (2011)	-0.10269	0.11683	-0.10270	0.11680	-4.35E-14	-2.02E-14
Surgery Year (2012)	0.05547	0.11897	0.05550	0.11900	-5.02E-14	-2.21E-14
Surgery Year (2013)	-0.11956	0.11382	-0.11960	0.11380	5.00E-14	-1.77E-14
Surgery Year (2014)	-0.10956	0.11617	-0.10960	0.11620	1.33E-13	-2.05E-14
Surgery Year (2015)	0.03701	0.12798	0.03700	0.12800	3.25E-13	-2.02E-14
Data Partner Site (2)	-0.10433	0.11751	-0.10430	0.11750	2.99E-13	-2.19E-14
Data Partner Site (3)	0.75506	0.12577	0.75510	0.12580	-2.82E-12	-8.75E-14

Reference Groups: Race (White), Surgery Year (2010), and Data Partner Site (1)

Outcome: Weight loss ≥ 20% (within one-year post-surgery); Data Partner sample sizes: n_1 = 2,728, n_2 = 1,018, n_3 = 1,706

Table 4c. Distributed Cox Proportional Hazards Regression vs. Pooled Patient-Level Cox Proportional Hazards Regression for Horizontally Partitioned Data (CIDA-based Actual Bariatric Surgery Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimate	Standard Errors		
Exposure	-0.58160	0.05275	-0.58160	0.05275	6.66E-16	-8.33E-17
Age	-0.01107	0.00146	-0.01107	0.00146	1.39E-17	-9.11E-18
BMI (Pre-Index)	-0.00006	0.00009	-0.00006	0.00009	2.85E-19	-1.49E-19
Comorbidity Score	-0.00787	0.01205	-0.00787	0.01205	-3.64E-17	-1.04E-17
No. Ambulatory Visits	0.00584	0.00158	0.00584	0.00158	-2.95E-17	1.08E-18
No. Emergency Department Visits	-0.01873	0.01679	-0.01873	0.00158	1.56E-16	-2.43E-17
No. Inpatient Visits	-0.08587	0.04580	-0.08587	0.04580	-9.58E-16	-1.25E-16
No. Non-Acute Institutional Stay	0.06626	0.29266	0.06626	0.29266	3.75E-16	-3.33E-16
No. Other Ambulatory Visits	0.00279	0.00134	0.00279	0.00134	4.03E-17	-1.52E-18
Days Prior (Pre-Surgery Vitals Measurement)	-0.00221	0.00096	-0.00221	0.00096	2.39E-17	-2.17E-18
Race (Unknown)	-0.18898	0.04765	-0.18898	0.04765	5.27E-16	0.00E+00
Race (American Indian or Alaska Native)	-0.07476	0.12019	-0.07476	0.12019	1.25E-16	2.78E-17
Race (Asian)	-0.22309	0.10933	-0.22309	0.10933	-2.78E-17	6.94E-17
Race (Black or African American)	-0.18457	0.05116	-0.18457	0.05116	1.94E-16	-1.39E-17
Race (Native Hawaiian or Other Pacific Islander)	-0.19748	0.17333	-0.19748	0.17333	1.42E-15	2.78E-17
Female	-0.00887	0.04052	-0.00887	0.04052	-1.24E-15	-3.47E-17
Surgery Year (2011)	-0.08021	0.05176	-0.08021	0.05176	8.60E-16	1.11E-16
Surgery Year (2012)	-0.02547	0.05136	-0.02547	0.05136	4.61E-16	7.63E-17
Surgery Year (2013)	-0.09519	0.05195	-0.09519	0.05195	1.17E-15	4.86E-17
Surgery Year (2014)	-0.16866	0.05235	-0.16866	0.05235	8.60E-16	1.18E-16
Surgery Year (2015)	0.24763	0.05640	0.24763	0.05640	3.89E-16	1.04E-16
Data Partner Site (2)	-0.15270	0.05188	-0.15270	0.05188	2.11E-15	-6.94E-18
Data Partner Site (3)	0.33440	0.05161	0.33440	0.05161	8.33E-16	2.08E-17

Reference Groups: Race (White), Surgery Year (2010), and Data Partner Site (1)

Outcome: Time to weight loss ≥ 20% (within one-year post-surgery); Data Partner sample sizes: n_1 = 2,728, n_2 = 1,018, n_3 = 1,706

D. OPERATIONAL PERFORMANCE

For each regression model type, we required at least one successful external, phase three, end-to-end test with the three select Data Partners to classify the DRA query workflow and algorithm as functional. As expected, distributed linear regression analysis required two iterations, while the distributed logistic and Cox proportional hazards regression analyses each required six iterations for model convergence during phase three testing. In total, we extracted 111, 271, and 271 time stamps from successful distributed linear, logistic, and Cox end-to-end tests, respectively (**Table 4**).

Table 4. Summary of Phase Three Tests with Data Partners

	Linear regression	Logistic regression	Cox proportional hazards regression
Total number of Data Partners	3	3	3
Required number of iterations for model convergence	2	6	6
Total time stamps extracted	111	271	271

Table 5 summarizes the operational performance of the PopMedNet DRA query workflow. It took an average of 102.4 seconds to complete one DRA iteration across all three regression model types. The file transfer processes (file upload, download, and transfer to the reciprocal party) accounted for 89% of the iteration time. Specifically, downloading and uploading DRA files at the analysis center required an average of 28.6 and 9.8 seconds, respectively. File transfer from the analysis center to the Data Partners took on average 9.4 seconds. Downloading and uploading DRA files at the Data Partners required an average of 10.1 and 15.5 seconds, respectively. File transfer from the Data Partners to the analysis center took on average 22.1 seconds. SAS execution required an average of 8.0 seconds to compute intermediate statistics at the Data Partners and 3.8 seconds to compute regression parameter estimates at the analysis center.

The distributed Cox regression required the greatest amount of iteration time (113.5 seconds), followed by logistic regression (95.0 seconds), and then linear regression (91.5 seconds). Overall, distributed linear regression analysis with our bariatric surgery test case required 440.7 seconds to complete, while logistic and Cox proportional hazards regression analysis required 925.5 and 1,016 seconds, respectively.

Table 5. Operational Performance of the PopMedNet Distributed Regression Analysis query workflow (for Horizontally Partitioned Data, CIDA-based Actual Bariatric Surgery Data)

	Linear	Logistic	Cox	Overall
	Mean Time Elapses in seconds (Standard Error)			
Average Iteration Time	**91.5 (10.5)**	**95 (3.1)**	**113.5 (5.2)**	**102.4 (3.8)**
Analysis Center				
Download Time	20.5 (5.4)	20.6 (1.3)	39.4 (4)	28.6 (3.2)
SAS Execution Time	4.3 (2.6)	3 (1.1)	4.4 (0.4)	3.8 (0.6)
Upload Time	8.4 (1.1)	10.2 (0.7)	9.9 (0.6)	9.8 (0.4)
File Transfer Time (to Data Partners)	10.5 (0.4)	9.1 (0.5)	9.4 (0.5)	9.4 (0.3)
Data Partners				
Download Time	8.6 (1.2)	10.3 (0.6)	10.3 (0.8)	10.1 (0.4)
SAS Execution Time	8.2 (0.8)	7.9 (0.4)	8 (0.3)	8 (0.2)
Upload Time	15.6 (1.2)	15.9 (0.6)	15.1 (0.3)	15.5 (0.3)
File Transfer Time (to Analysis Center)	20 (0.8)	21.8 (1.9)	23.1 (1.2)	22.1 (1.0)
Total End to End Run Time	**440.7**	**925.5**	**1016**	

E. DISTRIBUTED REGRESSION ANALYSIS WITH VERTICALLY PARTITIONED DATA

Our exploratory assessment concluded that it would also be feasible to perform DRA with vertically partitioned data in DDNs that utilizes PopMedNet as its data-sharing platform. Several secure multiparty computation protocols have been discussed in the literature for vertically partitioned data. We found the Beaver et al's secure matrix multiplication protocol to be best suited for potential integration into the PopMedNet DRA query workflow [34 35]. We explored this protocol with distributed linear regression analysis using a modified Commodity Server Approach [36]. Full details of this protocol and the mathematical derivation are available in the Appendix: Linear Regression on Vertically Partitioned Data.

Table 6 and **Table 7** show the results from our proof-of-concept analysis. The algorithm produced statistically equivalent regression parameter and standard error estimates to the pooled individual-level data analysis. Additional work is needed to integrate this algorithm into the PopMedNet DRA query workflow and additional internal and external testing is required to assess operational performance.

Table 6. Distributed Linear Regression vs. Pooled Patient-Level Linear Regression for Vertically Partitioned Data (Boston Housing Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Errors
	Estimates	Standard Errors	Estimates	Standard Errors		
Intercept	35.50548	1.57690	35.50548	1.57690	-7.82E-13	1.22E-14
Crime	-0.27283	0.04401	-0.27283	0.04401	4.66E-15	9.02E-16
Distance	-1.01582	0.23259	-1.01582	0.23259	1.05E-13	2.08E-15
Industrialization	-0.73017	0.07229	-0.73017	0.07229	2.91E-14	9.30E-16

Table 7. Distributed Linear Regression vs. Pooled Patient-Level Linear Regression for Vertically Partitioned Data (Simulated Bariatric Surgery Data)

Covariates	DRA		Pooled Patient-Level		Differences in Regression Parameter Estimates	Differences in Standard Error Estimates
	Estimates	Standard Errors	Estimates	Standard Errors		
Intercept	-32.02612	0.11958	-32.02612	0.11958	-1.32E-11	-8.56E-13
Exposure	-4.97118	0.02627	-4.97118	0.02627	1.58E-12	-1.90E-13
Age	0.20074	0.00134	0.20074	0.00134	-2.78E-17	-9.67E-15
BMI (Pre-Index)	-0.00106	0.00145	-0.00106	0.00145	-2.97E-15	-1.04E-14
Comorbidity Score	0.29456	0.00707	0.29456	0.00707	7.48E-14	-5.11E-14
No. Ambulatory Visits	0.99774	0.00198	0.99774	0.00198	-2.51E-14	-1.43E-14
No. Emergency Department Visits	5.01466	0.01298	5.01466	0.01298	4.74E-13	-9.38E-14
No. Inpatient Visits	3.02110	0.01315	3.02110	0.01315	-2.13E-14	-9.50E-14
No. Non-Acute Institutional Stay	3.99434	0.01267	3.99434	0.01267	7.11E-14	-9.15E-14
No. Other Ambulatory Visits	1.99393	0.00714	1.99393	0.00714	-5.73E-14	-5.15E-14
Days Prior (Pre-Surgery Vitals Measurement)	0.20004	0.00025	0.20004	0.00025	2.00E-15	-1.83E-15
Race (Unknown)	1.01773	0.04977	1.01773	0.04977	1.18E-11	-3.59E-13
Race (American Indian or Alaska Native)	2.03351	0.05008	2.03351	0.05008	6.70E-12	-3.62E-13
Race (Asian)	2.99682	0.05349	2.99682	0.05349	1.73E-11	-3.86E-13
Race (Black or African American)	4.08526	0.05329	4.08526	0.05329	9.69E-12	-3.85E-13
Race (Native Hawaiian or Other Pacific Islander)	5.08544	0.05339	5.08544	0.05339	1.37E-11	-3.86E-13
Female	2.03652	0.03750	2.03652	0.03750	4.76E-12	-2.71E-13
Surgery Year (2011)	-0.05022	0.04489	-0.05022	0.04489	-3.96E-12	-3.24E-13
Surgery Year (2012)	-0.05880	0.04453	-0.05880	0.04453	3.52E-12	-3.22E-13
Surgery Year (2013)	-0.01432	0.04470	-0.01432	0.04470	-3.92E-12	-3.23E-13
Surgery Year (2014)	-0.02875	0.04373	-0.02875	0.04373	-4.21E-13	-3.16E-13
Surgery Year (2015)	-0.00121	0.04797	-0.00121	0.04797	1.75E-12	-3.47E-13

Reference Groups: Race (White), Surgery Year (2010)
Outcome: Change in body mass index one-year post-surgery

We also found that DRA with vertically partitioned data required a similar three-step process framework as with horizontally partitioned data: 1) assemble a patient-level analytic dataset at each Data Partner, 2) distribute a DRA package to each Data Partner for local iterative analysis, which could be done through PopMedNet, and 3) if necessary, iteratively transfer intermediate files between Data Partners and the analysis center until the model converges or until a pre-specified maximum number of iterations is reached. Distinct to vertically partitioned data, the analytic datasets held locally by participating Data Partners must share a primary key (e.g., a de-identified unique patient ID), be sorted by the primary key, and have the same primary key and number of records with non-missing independent and dependent variables. Without a shared primary key, the disjointed intermediate statistics cannot be concatenated with the secure matrix multiplication protocol. This requirement is somewhat analogous to the need for harmonizing datasets in DRA with horizontally partitioned data – each dataset is required to have the same set of covariates.

Although we did not do it in this project, steps 2 and 3 of the vertical DRA algorithm can in principle be integrated with the newly developed query workflow for horizontal DRA. We integrated the PopMedNet DRA query workflow (the iterative file transfer process) and the DRA SAS algorithms for horizontally partitioned data (analytic processes) by using a "common folder structure" and a trigger text file. Thus, the workflow is agnostic to DRA algorithms, secure multiparty protocols, and statistical software. The only requirement is that DRA algorithms monitor and output intermediate statistic files and updated parameter estimates to the pre-specified folders specified in the DMC settings (e.g. *inputfiles* and *msoc folders*).

F. PROJECT OBJECTIVES AND DELIVERABLES

Table 8 provides a list of statuses corresponding to each project objective.

Table 8. List of Project Objectives and Deliverables

Objective	Status
1) Develop an overall analytic/process framework for DRA in DDNs	*Completed* In our analysis, we found three steps were required to perform DRA. We summarize these steps in a three-step process framework (pages 13 to 15). We have also published this framework in eGEMs.[20]
2) Develop and test SAS DRA algorithms that are agnostic to file transfer software, to perform distributed linear, logistic, and Cox proportional hazards regression analysis using simulated horizontally partitioned data	*Completed* We completed the development of SAS algorithms to perform distributed linear, logistic, and Cox proportional hazards regression analysis on horizontally partitioned data. All algorithms (SAS packages), user documentation, test data, and sample of reports for each regression model type are available on the Sentinel System website at https://www.sentinelinitiative.org/sentinel/methods/utilizing-data-various-data-partners-distributed-manner. We also published technical documentations on the algorithms on arxiv.org.[37 38]

Objective	Status
	Test results with simulated horizontally partitioned data are located on pages 21 to 24 of this report.
3) Develop a prototype workflow in PopMedNet that is automatable and can iteratively transfer DRA files between Data Partners and the analysis center in DDNs	*Completed* We completed the development of a PopMedNet-based workflow that supports manual, semi-automated, and automated DRA. We report on the precision and operational performance of the automated DRA workflow on pages 21 to 28 of this report. The manual and semi-automated workflows are extensions of the automated workflow. Thus, the precision of each algorithm is the same.
4) Integrate the SAS algorithms and PopMedNet workflow and beta-test the integration with Sentinel Data Partners and perform distributed linear, logistic, and Cox proportional hazards regression analysis using real world data	*Completed* We integrated the SAS algorithms with the PopMedNet DRA query workflow and beta-tested the integration with three Sentinel Data Partners with real-world data. The precision and operational performance of the beta-tests are located on pages 25 and 28.
5) Place the source code and documentation of the SAS algorithms and PopMedNet query workflow in public domain	*Completed* We placed the source code and user documentation of the SAS algorithms on the Sentinel System website at https://www.sentinelinitiative.org/sentinel/methods/utilizing-data-various-data-partners-distributed-manner. PopMedNet source code and user documentation can be downloaded from https://www.popmednet.org.
6) Explore the feasibility of performing DRA with vertically partitioned data in DDNs by developing and testing SAS algorithms to perform distributed linear regression analysis with simulated vertically partitioned data	*Completed* We explored the feasibility of performing linear regression analysis with vertical partitioned data in DDNs. We report on this exploration on pages 29 to 31 of this report.

V. DISCUSSION

We have successfully performed automatable DRA in a large DDN with horizontally partitioned data using PopMedNet. The process required three steps: 1) assembling a de-identified patient-level analytic dataset at each Data Partner, which can be done using a distributed program developed by the analysis center, 2) distributing a DRA package developed by the analysis center for iterative local execution at each Data Partner, and 3) iteratively transferring intermediate statistics between Data Partners and the analysis center until the regression model converges or a pre-specified maximum number of iterations has been reached. In our pilot, step 1 of the framework utilized the existing Sentinel query fulfillment process to create an analytic dataset and step 2 used the existing PopMedNet query workflow to distribute the DRA package.

Step 3 required enhancement to the current PopMedNet query workflow to support an iterative, automatable file transfer process between Data Partners and the analysis center in the form of sub-query requests and responses embedded within an overall DRA query request. The introduction of trigger text files at different steps of the process integrated and automated the PopMedNet-driven file transfer process and the SAS-driven analytic process. Overall, this workflow is agnostic to statistical software (e.g., SAS and R), accommodates different regression models (e.g., linear, logistic, and Cox), and allows different levels of automation (completely manual, semi-automated, and fully automated). The source code and documentation for the algorithms and workflow are available for download on the Sentinel System website (https://www.sentinelinitiative.org/sentinel/methods/utilizing-data-various-data-partners-distributed-manner) and PopMedNet website (www.popmednet.org), respectively.[20 39]

We also successfully developed a preliminary version of the DRA algorithm for linear regression analysis in vertically partitioned data and found that vertical DRA followed a similar three-step process framework as with horizontally partitioned data. This similarity would allow us to use the PopMedNet DRA query workflow to coordinate and automate the transfer of files between the parties involved in vertical DRA.

A. PERFORMANCE OF THE DISTRIBUTED REGRESSION ANALYSIS QUERY WORKFLOW AND ALGORITHMS

We evaluated the operational and statistical performance of the pilot DRA query workflow and DRA algorithms in three different phases with four different analytic datasets. All tests produced statistically equivalent regression parameter and standard error estimates to those obtained from the pooled corresponding patient-level analyses in a simulated DDN test environment and an actual DDN with three Sentinel Data Partners.

With regards to operational performance, our tests with the three Sentinel Data Partners showed that DRA could be completed in under 20 minutes, excluding the time required to assemble an analytic dataset at each Data Partner. In our test case, two iterations were required to compute the closed-form linear regression parameters, and standard errors, while six iterations were required for logistic and Cox proportional hazards regression. The file transfer processes accounted for most of the iteration time (89%). Our test case also found distributed Cox proportional hazards regression analysis required the greatest iteration time and time to perform an end-to-end DRA. This was largely attributed to the time needed to download a greater number of files coming from the Data Partners to the analysis center. The average iteration time and time for an end-to-end distributed linear and logistic regression were similar. This was expected as distributed logistic regression analysis can be viewed as an extension of distributed linear regression, with the additional computation of the iteration's variance and weights, which is

negligible compared to file transfer process time. In practice, the execution time for a given query is expected to be dependent on several factors, including the number of Data Partners, the sample size, the type of regression model, and the complexity of the model. Indeed, Eman and colleagues reported DRA computational time increased with more covariates and Data Partners, and larger analytic datasets [40]. Although the actual execution time of future queries will likely be different than what we presented here, we expect the relative execution time – most time would be spent on file transfers – would be similar in those queries.

Overall, the performance of the DRA query workflow and algorithms demonstrates the feasibility of performing DRA in a large DDN such as Sentinel using PopMedNet. It is technically possible to further shorten the time to complete an end-to-end DRA request. However, the time required to execute the iterative process, a key component of automatable DRA, is marginal compared to the time needed to assemble an analytic dataset and a DRA package. In other words, the total response time for an end-to-end DRA request is likely comparable to the response time for a typical data request (e.g., a CIDA request) in Sentinel and possibly other DDNs if Data Partners grant permission to automate the iterative process.

Importantly, the DRA regression parameters and standard errors computed in our test cases were statistically equivalent to those that one would obtain with a pooled patient-level data analysis. Thus, the pilot DRA workflow and algorithms allow Data Partners to "share" their information for multi-center analysis with much less concern about data security, patient privacy, unapproved uses of data, and disclosures of proprietary information.

B. EXTENSION TO OTHER DISTRIBUTED DATA NETWORKS

Although we chose to develop this new PopMedNet capability within the Sentinel System, the workflow can be generalized to other PopMedNet-based DDNs, such as PCORnet and NIH Collaboratory. Most PopMedNet-based DDNs require the same components as Sentinel to fulfill a query: a DMC at each Data Partner to receive and respond to query requests, a common folder structure to manage and organize query results, and SAS to perform statistical analysis. Most of the Data Partners in other DDNs will likely have one of the three major configurations identified among the Sentinel Data Partners. Thus, we anticipate that the new DRA capability can be extended to other PopMedNet-based DDNs. Importantly, the three Data Partners that participated in this project are also members of other DDNs, such as PCORnet, thus the successful pilot of the DRA query workflow within these Data Partners will facilitate adoption of DRA in other DDNs.

Additionally, the workflow has the capability to conduct DRA in three different levels of automation: completely manual, semi-automated, and fully automated. However, the degree of automation will depend on user acceptability. In our development work survey, we found mixed perspectives towards automating part or all of the query workflow, such as automatic file uploads and downloads. Six Data Partners would be willing to automate these steps, one would require approval from their technical governing board, and eight would not be willing to automate any of these steps. Most Data Partners require or prefer the option to review all files prior to upload or download.

A manual workflow will likely impede routine use of DRA, and Data Partner participation due to its tediousness and susceptibility to human error. However, a manual DRA workflow is likely required in DDNs wishing to add DRA to their analytic capabilities, or at least available as part of the initial roll-out phase to facilitate trust and acceptability. Having an opportunity to review and confirm that the iterative process only transfers highly summarized, non-identifiable intermediate statistics may improve Data Partners' willingness to automate DRA.

Our three-step framework is also likely generalizable to other DDNs. Previous work on a pilot DRA web-service, Web Grid Binary LOgistic REgression (WebGLORE), identified four modules (steps) required for DRA 1) user registration, 2) initiator task creation, 3) user participation, and 4) collaborative model construction [18 19]. Step 1 is generally not required in established DDNs as all Data Partners are already registered. Steps 2 and 3 are embedded in step 2 of our framework, and step 4 is synonymous with step 3 of our framework.

To our knowledge, there are no reports of routine DRA in any large DDNs. Previous work has only involved simulated or controlled distributed environments [12 14 16 18 19]. Thus, it is difficult to compare our DRA workflow with other similar DDNs. However, our computational times are comparable to that of a distributed logistic regression analysis in a simulated, horizontally partitioned DDN using the Secure Pooled Analysis across K-sites (SPARK) protocol [40]. These authors reported computational time ranging from 0.024 to 1.02 minutes at each site with five to 20 covariates. Our computation time fell in the lower end of this range with 23 covariates. This difference may be attributed to the authors using homomorphic encryption to provide an additional layer of security and larger datasets (100,000 to 1 million).

Contrary to the paucity of reports on operational performance of DRA workflows and algorithms, there are more studies that have evaluated the statistical performance of DRA. Wu et al used Grid Binary LOgistic REgression (GLORE) to perform distributed logistic regression analysis on horizontally partitioned data and reported differences in the estimates between the DRA and the pooled data analysis as low as 10e-17 [14]. Eman et al's SPARK protocol reported precision as low as 10e-6 for their distributed logistic regression analysis [40]. Lu et al's performed a distributed Cox proportional hazards analysis using Web-based Distributed Cox Regress Model (WebDISCO) and reported precision as low as 10e-15. Thus, our pilot's statistical performances are consistent with comparable studies.

C. LIMITATIONS

Our pilot is not without limitations. First, we created a DRA workflow in PopMedNet, which may limit its use to only those DDNs that employ this data-sharing platform. However, PopMedNet is open source and the data-sharing platform used by several large and active DDNs (e.g., Sentinel, PCORnet, and NIH Collaboratory). Thus, our work has great potential to directly impact the conduct of large, multi-center studies with observational data. Additionally, DRA requires infrastructure and processes beyond technology that are more likely available in the aforementioned DDNs. For example, DRA with horizontally partitioned data requires harmonized datasets with the same covariates and covariate names [19]. Since its inception, Sentinel has continuously enhanced its common data model, routine analytic tools, and quality assurance processes. Thus, Sentinel Data Partners can rapidly create harmonized patient-level analytic datasets ready for DRA. DDNs without these structures and processes may not be able to perform DRA seamlessly even if our DRA workflow was adaptable to other data-sharing platforms.

Second, we successfully performed DRA in three Sentinel Data Partners with two different technological configurations. It is possible each configuration has additional layers of complexity (e.g., requiring authentication to access the network drive in Configuration 2) that were not present during testing. There may also be other configurations in Sentinel and other DDNs, making our workflow potentially inoperable or incompatible with these configurations. However, we were able to have two Data Partners re-configure their hardware configurations to Configurations 1 or 2 for initial development and testing. The process was relatively straightforward. Therefore, it is possible to have Data Partners with other configurations make relatively minor changes to their settings to facilitate implementation of DRA in large DDNs.

Lastly, our operational performances were based on a small sample of successful end-to-end tests. To increase the number of external tests would require resources beyond the scope of this project. These external tests were limited to regression models with 23 covariates, a three-Data Partner DDN, and analytic datasets of about 1,000 to 3,000 patients. In our tests, we found transfer time accounted for almost 90% of the iteration time and is dictated by the number of files transferred. Larger analytical datasets do not directly correlate with a greater number of files needed to transfer. Although we could not directly extrapolate our observed execution time to future queries, we expect much of the time to be spent on file transfer in those queries.

D. FUTURE WORK

To our knowledge, there is no published experience on addressing Data Partners' computer system failures or interruptions between DRA iterations in actual DDNs. The current Sentinel query workflow addresses interruptions by terminating the process and re-running the query. This approach is not optimal for fully automated DRA, as all Data Partners would have to re-initiate the whole iterative DRA workflow. Future enhancements will include the development of new capabilities to automatically pause the workflow or restart the analysis from the previous error-free iteration when an interruption occurs.

Future work is also needed to develop analytic algorithms for other more complicated statistical models, and more sophisticated model diagnostics that do not require the sharing of patient-level information. Thus, enhancements to our workflow to integrate other secure multiparty computation protocols may be required [15 40]. Additional work is also needed to stress test the DRA query workflow and SAS algorithms in broader and more extreme scenarios. Specifically, DRA requests with a greater number of Data Partner sites will need to be evaluated. The impact of larger analytic datasets on the operational performance of DRA should also be evaluated.

Unlike other routinely used privacy preserving analytic methods, DRA with the PopMedNet DRA query workflow require semi-synchronous collaboration of the Data Partners. In our tests, we requested the three Data Partners to execute the DRA query workflow and SAS algorithms at around the same time window. This may be the solution, but may pose some barriers to Data Partners in different time zones or Data Partners with different internal workflows. Additional work is required to develop operating procedures and policies to address these non-technical barriers to implementing DRA in practice.

We have developed the DRA capability within horizontally partitioned data environments, a setting in which information from different individuals is available in different data sources. We explored and tested distributed linear regression analysis in a simulated vertically partitioned data environment. Future work should examine other regression models and integrate vertical DRA into the PopMedNet DRA query workflow. As vertical DRA assumes the existence of a primary key, additional work is needed to facilitate the development of such a key across Data Partners who are interested in using vertical DRA in their collaborative studies.

Although we have developed the DRA algorithms and PopMedNet query workflow to be consistent with the existing Sentinel query workflows, additional work is needed to implement DRA in routine Sentinel queries. The prototype will need to be tested in all Sentinel Data Partners, which may lead to additional enhancements or modifications. The process will also need to include robust user acceptability tests as routine implementation of DRA requires automation of the file transfer process, which means that Data Partners would forego review of some of the summary-level data files before they are transferred to the SOC. Finally, proper documentation about the DRA tool would need to be developed and posted.

E. CONCLUSION

In summary, we have developed a functional prototype for conducting automatable DRA within Sentinel, a DDN that uses PopMedNet. The PopMedNet-based DRA query workflow and SAS algorithms have the potential to be integrated into Sentinel query fulfillment process for routine use and other DDNs.

VI. REFERENCES

1. Brown JS, Holmes JH, Shah K, Hall K, Lazarus R, Platt R. Distributed health data networks: a practical and preferred approach to multi-institutional evaluations of comparative effectiveness, safety, and quality of care. Medical care 2010;**48**(6 Suppl):S45-51 doi: 10.1097/MLR.0b013e3181d9919f.
2. Maro JC, Platt R, Holmes JH, et al. Design of a national distributed health data network. Annals of internal medicine 2009;**151**(5):341-4.
3. Toh S, Platt R, Steiner JF, Brown JS. Comparative-effectiveness research in distributed health data networks. Clinical pharmacology and therapeutics 2011;**90**(6):883-7 doi: 10.1038/clpt.2011.236.
4. McNeil MM, Gee J, Weintraub ES, et al. The Vaccine Safety Datalink: successes and challenges monitoring vaccine safety. Vaccine 2014;**32**(42):5390-8 doi: 10.1016/j.vaccine.2014.07.073.
5. Steiner JF, Paolino AR, Thompson EE, Larson EB. Sustaining Research Networks: the Twenty-Year Experience of the HMO Research Network. EGEMS (Washington, DC) 2014;**2**(2):1067.
6. Platt R, Carnahan RM, Brown JS, et al. The U.S. Food and Drug Administration's Mini-Sentinel program: status and direction. Pharmacoepidemiology and drug safety 2012;**21 Suppl 1**:1-8 doi: 10.1002/pds.2343.
7. Fleurence RL, Curtis LH, Califf RM, Platt R, Selby JV, Brown JS. Launching PCORnet, a national patient-centered clinical research network. Journal of the American Medical Informatics Association : JAMIA 2014;**21**(4):578-82 doi: 10.1136/amiajnl-2014-002747.
8. Rassen JA, Moran J, Toh D, et al. Evaluating strategies for data sharing and analyses in distributed data settings. Secondary Evaluating strategies for data sharing and analyses in distributed data settings 2013. https://www.sentinelinitiative.org/sites/default/files/Methods/Mini-Sentinel_Methods_Evaluating-Strategies-for-Data-Sharing-and-Analyses_0.pdf.
9. Toh S, Gagne JJ, Rassen JA, Fireman BH, Kulldorff M, Brown JS. Confounding adjustment in comparative effectiveness research conducted within distributed research networks. Medical care 2013;**51**(8 Suppl 3):S4-10 doi: 10.1097/MLR.0b013e31829b1bb1.
10. Toh S, Shetterly S, Powers JD, Arterburn D. Privacy-preserving analytic methods for multisite comparative effectiveness and patient-centered outcomes research. Medical care 2014;**52**(7):664-8 doi: 10.1097/MLR.0000000000000147.
11. Karr AF, Lin X, Sanil AP, Reiter JP. Analysis of Integrated Data without Data Integration. Chance 2012;**17**(3):26-29 doi: 10.1080/09332480.2004.10554910.
12. Wolfson M, Wallace SE, Masca N, et al. DataSHIELD: resolving a conflict in contemporary bioscience--performing a pooled analysis of individual-level data without sharing the data. International journal of epidemiology 2010;**39**(5):1372-82 doi: 10.1093/ije/dyq111.
13. Fienberg SE, Fulp WJ, Slavkovic AB, Wrobel TA. “Secure” log-linear and logistic regression analysis of distributed databases. Privacy in statistical databases: Springer, 2006:277-90.
14. Wu Y, Jiang X, Kim J, Ohno-Machado L. Grid Binary LOgistic REgression (GLORE): building shared models without sharing data. Journal of the American Medical Informatics Association : JAMIA 2012;**19**(5):758-64 doi: 10.1136/amiajnl-2012-000862.
15. Karr AF, Lin X, Sanil AP, Reiter JP. Secure Regression on Distributed Databases. Journal of Computational and Graphical Statistics 2005;**14**(2):263-79 doi: 10.1198/106186005x47714.
16. Lu CL, Wang S, Ji Z, et al. WebDISCO: a web service for distributed cox model learning without patient-level data sharing. Journal of the American Medical Informatics Association : JAMIA 2015;**22**(6):1212-9 doi: 10.1093/jamia/ocv083.
17. Wu Y, Jiang X, Ohno-Machado L. Preserving Institutional Privacy in Distributed binary Logistic Regression. AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium 2012;**2012**:1450-8.

18. Jiang W, Li P, Wang S, et al. WebGLORE: a web service for Grid LOgistic REgression. Bioinformatics (Oxford, England) 2013;**29**(24):3238-40 doi: 10.1093/bioinformatics/btt559.
19. Jiang X, Wu Y, Marsolo K, Ohno-Machado L. Development of a web service for analysis in a distributed network. EGEMS (Washington, DC) 2014;**2**(1):1053 doi: 10.13063/2327-9214.1053.
20. Her Q, Malenfant J, Malek S, et al. A query workflow design to perform automatable distributed regression analysis in large distributed data networks. EGEMS (Washington, DC) 2018;**6**(1):11 doi: http://doi.org/10.5334/egems.209.
21. Ball R, Robb M, Anderson SA, Dal Pan G. The FDA's sentinel initiative--A comprehensive approach to medical product surveillance. Clinical pharmacology and therapeutics 2016;**99**(3):265-8 doi: 10.1002/cpt.320.
22. Curtis LH, Brown J, Platt R. Four health data networks illustrate the potential for a shared national multipurpose big-data network. Health affairs (Project Hope) 2014;**33**(7):1178-86 doi: 10.1377/hlthaff.2014.0121.
23. Curtis LH, Weiner MG, Boudreau DM, et al. Design considerations, architecture, and use of the Mini-Sentinel distributed data system. Pharmacoepidemiology and drug safety 2012;**21**(S1):23-31.
24. Sentinel System. Routine Querying System. Secondary Routine Querying System 2018. https://www.sentinelinitiative.org/sentinel/surveillance-tools/routine-querying-tools/routine-querying-system.
25. Davies M, Erickson K, Wyner Z, Malenfant J, Rosen R, Brown J. Software-Enabled Distributed Network Governance: The PopMedNet Experience. EGEMS (Washington, DC) 2016;**4**(2):1213 doi: 10.13063/2327-9214.1213.
26. Klann JG, Buck MD, Brown J, et al. Query Health: standards-based, cross-platform population health surveillance. Journal of the American Medical Informatics Association : JAMIA 2014;**21**(4):650-6 doi: 10.1136/amiajnl-2014-002707.
27. McGlynn EA, Lieu TA, Durham ML, et al. Developing a data infrastructure for a learning health system: the PORTAL network. Journal of the American Medical Informatics Association : JAMIA 2014;**21**(4):596-601 doi: 10.1136/amiajnl-2014-002746.
28. Davies M, Wyner Z. System Security - Documentation - PopMedNet Wiki. Secondary System Security - Documentation - PopMedNet Wiki 2018. https://popmednet.atlassian.net/wiki/display/DOC/System+Security.
29. Harrison D, Rubinfeld DL. Hedonic housing prices and the demand for clean air. Journal of environmental economics and management 1978;**5**(1):81-102.
30. Rossi PH, Henry JP. Seriousness: A measure for all purposes. Handbook of criminal justice evaluation 1980:489-505.
31. Arterburn D, Powers JD, Toh S, et al. Comparative effectiveness of laparoscopic adjustable gastric banding vs laparoscopic gastric bypass. JAMA surgery 2014;**149**(12):1279-87 doi: 10.1001/jamasurg.2014.1674.
32. Reiter JP, Kohnen CN, Karr AF, Lin X, Sanil AP. Secure regression for vertically partitioned, partially overlapping data. ASA Proceedings 2004. Research Triangle Park, NC: National Institute of Statistical Sciences, 2004.
33. Slavkovic AB, Nardi Y, Tibbits MM. "Secure" logistic regression of horizontally and vertically partitioned distributed databases. Seventh IEEE International Conference on Data Mining Workshops; 2007; Omaha, NE, USA. Institute of Electrical and Electronics Engineers.
34. Beaver D. Commodity-based cryptography (extended abstract). Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. El Paso, Texas, USA: ACM, 1997:446-55.
35. Beaver D. Server-assisted cryptography. Proceedings of the 1998 workshop on New security paradigms. Charlottesville, Virginia, USA: ACM, 1998:92-106.

36. Du W, Han YS, Chen S. Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification. Proceedings of the 2004 SIAM International Conference on Data Mining:222-33.
37. Her QL, Vilk Y, Young J, et al. A distributed regression analysis application based on SAS software. Part I: Linear and logistic regression. ArXiv e-prints 2018. https://ui.adsabs.harvard.edu/#abs/2018arXiv180802387H (accessed August 01, 2018).
38. Vilk Y, Zhang Z, Young J, et al. A distributed regression analysis application based on SAS software Part II: Cox proportional hazards regression. ArXiv e-prints 2018. https://ui.adsabs.harvard.edu/#abs/2018arXiv180802392V (accessed August 01, 2018).
39. PopMedNet. PopMedNet. Secondary PopMedNet 2018. https://www.popmednet.org/.
40. El Emam K, Samet S, Arbuckle L, Tamblyn R, Earle C, Kantarcioglu M. A secure distributed logistic regression protocol for the detection of rare adverse drug events. Journal of the American Medical Informatics Association : JAMIA 2013;**20**(3):453-61 doi: 10.1136/amiajnl-2011-000735.
41. McCullagh P, Nelder JA. *Generalized Linear Models, Second Edition*: Taylor & Francis, 1989.
42. Breslow N. Covariance analysis of censored survival data. Biometrics 1974;**30**(1):89-99.
43. Efron B. The Efficiency of Cox's Likelihood Function for Censored Data. Journal of the American Statistical Association 1977;**72**(359):557-65 doi: 10.1080/01621459.1977.10480613.
44. Her Q, Vilk Y, Young J, et al. A distributed regression analysis application based on SAS software Part I: Linear and logistic regression. 2018 Submitted;**77**(13):22 doi: 10.18637/jss.v077.i13.

VII. ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the contributions of the following organizations and individuals:

Affiliation	Name
Kaiser Permanente Colorado	David Tabano
Kaiser Permanente Norther California	Jack Hamilton
Kaiser Permanente Washington Health Research Institute	Ron Johnson

VIII. APPENDICES

A. DISTRIBUTED REGRESSION ALGORITHM FOR LINEAR AND LOGISTIC REGRESSION

1. Overview

Linear and logistic regression models, along with other popular models such as Poisson regression, are special cases of generalized linear models (GLMs) [41]. Maximum likelihood estimators of GLM regression coefficients can be obtained via an iteratively reweighted least squares (IRLS) algorithm when the link function for the GLM is chosen as the canonical link. In this case IRLS is equivalent to Newton-Raphson [41].

In this section, we give a distributed version of the IRLS algorithm for GLMs such that individual-level data from a given site need not be shared with other sites nor with a central analysis center. We show how this distributed algorithm can be implemented with standard SAS procedures.

Let K denote the number of sites, and n_k the number of subjects at site $= 1, \dots, K$. Further let$(Y_{i,k}, \boldsymbol{X}_{i,k}, w_{i,k})$, $i = 1, \dots, n_k$, denote the observed data for subject i at site k, with $Y_{i,k}$ the outcome, $\boldsymbol{X}_{i,k}$ a vector of p covariate values for subject *i*, and $w_{i,k}$ a subject-level weight. Let $\boldsymbol{Z}_{i,k} = \mathbf{1}||\boldsymbol{X}_{i,k}$ and $N = \sum_{k=1}^{K} n_k$ denote the sum of all observations. The input dataset at site k has the following structure:

$$\begin{matrix} w_{1,k} & X_{1,k,1} & \dots & X_{1,k,p} & Y_{1,k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{N,k} & X_{N,k,1} & \dots & X_{N,k,p} & Y_{N,k} \end{matrix} \quad (1)$$

A GLM assumes that $Y_{i,k}$ is distributed according to an exponential family (e.g. normal, binomial, Poisson) with

$$E\left[Y_{i,k}\middle|\mathbf{Z}_{i,k}\right] = \mu(\boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{Z}_{i,k})$$
$$var\left[Y_{i,k}\middle|\mathbf{Z}_{i,k}\right] = v(\boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{Z}_{i,k})$$

where $\boldsymbol{\beta}$ is a p+1 length vector of unknown regression coefficients. Before we consider how to estimate $\boldsymbol{\beta}$ generally via IRLS in this setting, we first consider the special case where we select the GLM as a linear regression model; i.e., where $Y_{i,k}$ is assumed to follow a normal distribution with $\mu(\boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{Z}_{i,k}) = \boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{Z}_{i,k}$ and $(\boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{Z}_{i,k}) = v = \sigma^2$. It follows from standard theory that a maximum likelihood estimate of $\boldsymbol{\beta}$ in this special case can be obtained by solving the (possibly weighted) least squares equations:

$$\sum_{k=1}^{K}\sum_{i=1}^{n_k} w_{i,k}(Y_{i,k} - \boldsymbol{\beta}^{T}\mathbf{Z}_{i,k})\mathbf{Z}_{i,k} = 0 \quad (2)$$

with respect to $\boldsymbol{\beta}$. In this case, an exact solution exists which is:

$$\widehat{\boldsymbol{\beta}} = \left(\sum_{k=1}^{K}\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Z}_k\right)^{-1}\left(\sum_{k=1}^{K}\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Y}_k\right) \quad (3)$$

with $\mathbf{Y}_k$a vector of length $n_k$with elements $Y_{i,k}$, $\mathbf{Z}_k$ a matrix of dimension $n_k * (p+1)$ with rows $\mathbf{Z}_{i,k}$ and $\mathbf{W}_k$ a diagonal matrix of dimension $n_k * n_k$ with diagonal elements $w_{i,k}$ $i = 1, \dots n_k$. Importantly, the matrices $\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Z}_k$ and $\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Y}_k$ can be calculated separately at each site k. These matrices are highly summarized and can be brought to the analytic center with minimal privacy risk. For example, the

dimension of $\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Z}_k$ is $(\mathrm{p}+1)*(\mathrm{p}+1)$ which is much smaller than the dimension of individual level matrix $\mathbf{Z}_k$ $(n_k*(p+1))$.

From a computational point of view, it is rather inefficient to calculate expressions like $\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Z}_k$ and $\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Y}_k$ as written because this requires transposing a large matrix $\mathbf{Z}_k$. A more efficient way of calculating the above expressions is by calculating them as weighted cross products of columns. Let us define a function $\boldsymbol{SSCP}(\mathbf{A},\mathbf{W})$ of matrix **A** with arbitrary dimensions and diagonal matrix **W** of dimension $r*r$ (with r the number of rows of **A**) as follow:

$$\left(\boldsymbol{SSCP}(\mathbf{A},\mathbf{W})\right)_{s,s'} = \sum_i w_i A_{i,s} A_{i,s'} = \sum_i w_i (\boldsymbol{A}^T)_{s,i} A_{i,s'} = (\mathbf{A}^T\mathbf{W}\mathbf{A})_{s,s'} \tag{4}$$

Here s and i are indices for a column and a row of matrix **A**, respectively. The function $\boldsymbol{SSCP}$ (Sum of Squares and Cross Products) is similar to a covariance function except that one does not need to subtract the column mean before multiplying columns. In SAS, the sums of squares and cross products (SSCP) matrix can be easily calculated using PROC CORR with option SSCP.

We can calculate matrices $\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Z}_k$ and $\mathbf{Z}_k^T\mathbf{W}_k\mathbf{Y}_k$ by applying the $\boldsymbol{SSCP}$ function to a matrix that concatenates the columns of $\mathbf{Z}_k$ and $\mathbf{Y}_k$:

$$\boldsymbol{SSCP}(\mathbf{Z}_k\,||\boldsymbol{Y}_k,\mathbf{W}_k) = \begin{pmatrix} \sum_i w_{i,k}\boldsymbol{Z_{i,k}^T}\,\mathbf{Z}_{i,k} & \sum_i w_{i,k}\boldsymbol{Z_{i,k}^T}\,Y_{i,k} \\ \sum_i w_{i,k}\mathbf{Z}_{i,k}Y_{i,k} & \sum_i w_{i,k}Y_{i,k}^2 \end{pmatrix} \tag{5}$$

Each $\boldsymbol{SSCP}(\mathbf{Z}_k\,||\boldsymbol{Y}_k,\mathbf{W}_k)$ in (5) can be easily computed at site k from the individual-level data at that site. These highly summarized data sets can then be transferred to the analytic center to compute the combined SSCP dataset:

$$\boldsymbol{SSCP}(\mathbf{Z}\,||\boldsymbol{Y},\mathbf{W}) = \sum_k \boldsymbol{SSCP}(\mathbf{Z}_k\,||\boldsymbol{Y}_k,\mathbf{W}_k) \tag{6}$$

The dataset in (6) is created with the property TYPE explicitly set to SSCP (the property TYPE is a part of a SAS dataset metadata). This dataset can then be fed directly into the REG procedure in lieu of an individual level dataset to obtain the solution (3). Once the combined SSCP matrix is fed into PROC REG at the analytic center, the procedure automatically calculates many desired statistics. These include, not only regression coefficient estimates $\widehat{\boldsymbol{\beta}}$, but also the variance estimate

$$\hat{\sigma}^2 = \frac{1}{N-p}\sum_{k=1}^{K}\left[\sum_{i=1}^{n_k} w_{i,k}\left(Y_{ik} - \widehat{\boldsymbol{\beta}}^{\boldsymbol{T}}\boldsymbol{Z}_{i,k}\right)^2\right] \tag{7}$$

inverse matrix $(\mathbf{Z}^T\mathbf{W}\mathbf{Z})^{-1}$ and the estimated covariance matrix

$$\widehat{cov}\left(\widehat{\boldsymbol{\beta}}\right) = \hat{\sigma}^2\,(\mathbf{Z}^T\mathbf{W}\mathbf{Z})^{-1} \tag{8}$$

along with collinearity diagnostic and a number of goodness of fit measures.

The above procedure is a special case of a more general IRLS algorithm for estimating the regression parameter $\boldsymbol{\beta}$ of a GLM. This general algorithm allows alternative choices of distribution, $\mu(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})$ and $v(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})$. For example, logistic regression is a GLM under a binomial outcome distribution with $\mu(\boldsymbol{\beta}^T\mathbf{Z}_{i,k}) = \frac{\exp(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})}{1+\exp(\beta^T\mathbf{Z}_{i,k})}$; $v(\boldsymbol{\beta}^T\mathbf{Z}_{i,k}) = \mu(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})[1-\mu(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})]$. Poisson regression is another example under a Poisson outcome distribution with $\mu(\boldsymbol{\beta}^T\mathbf{Z}_{i,k}) = \exp(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})$ and $v(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})$= $\mu(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})$. For any canonical link $v = \mu' = d\mu(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})/d(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})$

Unlike the special case of linear regression, IRLS for fitting a general GLM does not have an exact solution but iterates until convergence. Specifically, at each iteration m+1 until a convergence criterion is met, IRLS solves:

$$\sum_{k=1}^{K}\sum_{i=1}^{n_k}\widetilde{w}_{i,k}\left(\tilde{Y}_{i,k} - \boldsymbol{\beta}_{m+1}^{\boldsymbol{T}}\mathbf{Z}_{i,k}\right)\mathbf{Z}_{i,k} = 0 \tag{9}$$

for $\boldsymbol{\beta}_{m+1}$ where

$$\widetilde{w}_{i,k}(\boldsymbol{\beta}_m^{\boldsymbol{T}}) \equiv w_{i,k}v(\boldsymbol{\beta}_m^{\boldsymbol{T}}\mathbf{Z}_{i,k}), \tag{10}$$

$$\tilde{Y}_{i,k}(\boldsymbol{\beta}_m^{\boldsymbol{T}}) \equiv \frac{Y_{i,k} - \mu\left(\boldsymbol{\beta}_m^{\boldsymbol{T}}\mathbf{Z}_{i,k}\right)}{v\left(\boldsymbol{\beta}_m^{\boldsymbol{T}}\mathbf{Z}_{i,k}\right)} + \boldsymbol{\beta}_m^{\boldsymbol{T}}\mathbf{Z}_{i,k} \tag{11}$$

and $\boldsymbol{\beta}_m$the solution from the previous iteration m (with $\boldsymbol{\beta}_0$specified starting values). Both the redefined weight and outcome in (10) and (11), respectively, change at each iteration, but the covariate vector $\mathbf{Z}_{i,k}^T$ remains the same. For the special case of linear regression $\tilde{y}_{i,k} = Y_{i,k}$ and $\widetilde{w}_{i,k} = w_{i,k}$ and thus do not depend on $\boldsymbol{\beta}_m$. As expected, the algorithm reduces to standard linear regression that does not require iterative process.

In the more general case, the following describes a general implementation of IRLS to obtain an estimate of the regression coefficient $\boldsymbol{\beta}$ of a GLM using SSCP matrices and PROC REG in SAS. The resulting estimate is an MLE under distributional assumptions. This algorithm is implemented in the macro ***%distributed_regression*** which we describe in the next section.

1) For each iteration $m+1$ at each site k calculate the SSCP matrix

$$\boldsymbol{S_{SSCP}}\left(\mathbf{Z}_k \,||\widetilde{\boldsymbol{Y}}_{km}(\boldsymbol{\beta_m}), \widetilde{\mathbf{W}}_{km}\,(\boldsymbol{\beta_m})\right.$$

Bring these SSCP matrices from each site to the analytic center and calculate the combined SSCP matrix:

$$\boldsymbol{SSCP}\left(\mathbf{Z}\,||\widetilde{\boldsymbol{Y}}_m(\boldsymbol{\beta_m}), \widetilde{\mathbf{W}}_m(\boldsymbol{\beta_m})\right) = \sum_k \boldsymbol{S_{SSCP}}\left(\mathbf{Z}_k\,||\widetilde{\boldsymbol{Y}}_{km}(\boldsymbol{\beta_m}), \widetilde{\mathbf{W}}_{km}\,(\boldsymbol{\beta_m})\right. \tag{12}$$

2) Feed the combined SSCP matrix from (12) into PROC REG to solve for $\boldsymbol{\beta}_{m+1}$
3) Repeat until convergence is achieved (see below). On the iteration $m+1$ that meets the convergence criterion, $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}_{m+1}$

After convergence is achieved, an additional iteration of steps 1-3 will output the inverse of the matrix $\mathbf{Z}^T\widetilde{\mathbf{W}}\,(\widehat{\boldsymbol{\beta}})\mathbf{Z}$. The extra iteration is necessary because at iteration $m+1$ we do not know the matrix $\mathbf{Z}^T\widetilde{\mathbf{W}}\,(\boldsymbol{\beta_{m+1}})\mathbf{Z}$. We only know the matrix $\mathbf{Z}^{\boldsymbol{T}}\widetilde{\mathbf{W}}\,(\boldsymbol{\beta_m})\mathbf{Z}$. Note that for linear regression the weight does not depend on $\boldsymbol{\beta}$ and the extra step is not necessary.

The covariance matrix can be calculated as:

$$\widehat{cov}(\hat{\boldsymbol{\beta}}) = \mathbf{H}^{-1}(\hat{\boldsymbol{\beta}}) = \phi\left[\sum_{k=1}^{K}\sum_{i=1}^{n_k} w_{i,k}\mu'(\hat{\boldsymbol{\beta}}^T\mathbf{Z}_{i,k})\mathbf{Z}_{i,k}\mathbf{Z}_{i,k}^T\right]^{-1} = \phi\left(\mathbf{Z}^T\widetilde{\mathbf{W}}(\hat{\boldsymbol{\beta}})\mathbf{Z}\right)^{-1} \quad (13)$$

where $\mathbf{H}(\boldsymbol{\beta}) = -\frac{\partial^2 l}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^T}$ is the Hessian matrix defined by

$$\mathbf{H}(\boldsymbol{\beta}) = \sum_{k=1}^{K}\sum_{i=1}^{n_k} w_{i,k}\frac{\mu'(\boldsymbol{\beta}^T\mathbf{Z}_{i,k})}{\phi}\mathbf{Z}_{i,k}\mathbf{Z}_{i,k}^T \quad (14)$$

and ϕ a constant; for linear regression $\phi = \hat{\sigma}^2$; for logistic and Poisson regression ϕ =1. The above expression for $\widehat{cov}(\hat{\boldsymbol{\beta}})$ requires that the assumed probability distribution is correctly specified. The alternative sandwich variance estimator is robust to this assumption:

$$\widehat{cov}(\hat{\beta}) = \mathbf{H}^{-1}\boldsymbol{H_1}\mathbf{H}^{-1} \quad (15)$$

Where $\mathbf{H}(\boldsymbol{\beta})$ is as in 14 and the matrix $\boldsymbol{H_1}(\hat{\boldsymbol{\beta}})$can be calculated as:

$$\boldsymbol{H_1}(\hat{\boldsymbol{\beta}}) = \frac{N}{N-p}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{w_{i,k}^2\left(Y_{i,k} - \mu(\hat{\boldsymbol{\beta}}^T\mathbf{Z}_{i,k})\right)^2}{\phi^2}\mathbf{Z}_{i,k}\mathbf{Z}_{i,k}^T \quad (16)$$

The factor $\frac{N}{N-p}$ corresponds to the definition HC1 for the robust estimator for linear regression in PROC REG. The expression can be evaluated at each site as $\boldsymbol{SSCP}(\mathbf{Z}_k, \mathbf{W_k^H})$ where diagonal matrix of weights $\mathbf{W_k^H}$ has elements:

$$w_{i,k}^H = \frac{w_{i,k}^2\left(Y_{i,k} - \mu(\hat{\boldsymbol{\beta}}^T\mathbf{Z}_{i,k})\right)^2}{\phi^2} \quad (17)$$

After matrices $\boldsymbol{SSCP}(\mathbf{Z}_k, \mathbf{W_k^H})$ are brought to the analytic center the matrix $\boldsymbol{H_1}$ can be calculated as a sum of these matrices:

$$\boldsymbol{H_1} = \sum_k \boldsymbol{SSCP}(\mathbf{Z}_k, \mathbf{W_k^H}) \quad (18)$$

Once the covariance matrix $\widehat{cov}(\hat{\boldsymbol{\beta}})$ is calculated, the standard errors of $\hat{\boldsymbol{\beta}}$ can be calculated by taking a square root of corresponding diagonal elements of the matrix.

2. Convergence criteria

We use the relative convergence criteria which is identical to the SAS relative convergence criteria specified by option XCONV. Let $\beta_s^m\ m$ be the estimate of the parameter $s = 1, \ldots, p+1$ at iteration m.

The regression criterion is satisfied if:

$$\max_s|\delta_s^{m+1}| < xconv\_value$$

where

$$\delta_s^{m+1} = \begin{cases} \beta_s^{m+1} - \beta_s^m, & |\beta_s^m| < 0.01 \\ \frac{\beta_s^{m+1} - \beta_s^m}{\beta_s^m}, & else \end{cases}$$

B. DISTRIBUTED COX PROPORTIONAL HAZARD REGRESSION ANALYSIS SPECIFICATIONS

1. Overview

In this appendix, we describe the underlying algorithm implemented by our DRA application, which is a distributed version of the Newton-Raphson algorithm implemented to solve for the parameter estimates of a stratified Cox model based on the Breslow approximation to the partial likelihood for tied event times [42]. The algorithm fits a non-stratified Cox model when the number of strata is set to 1. This algorithm avoids sharing individual-level data by a given Data Partner with other sites and with the analysis center. As described in **Appendix C**, our algorithm can also implement the Efron approximation [43].

For $k = 1, \dots, K$, $m = 1, \dots, M$, and $i = 1, \dots, N_{m,k}$, let K denote the number of sites, M the number of strata, and $N_{m,k}$ the number of subjects at site k in strata m. Suppose, among all $N_m = \sum_{k=1}^{K} N_{m,k}$ patients in strata m, there are J_m unique event times, $t_{m,1} < t_{m,2} < \cdots < t_{m,J_m}$. Denote $(w_{i,m,k}, T_{i,m,k}, \Delta_{i,m,k}, \boldsymbol{Z}_{i,m,k})$ as the observed data for subject i at site k in stratum m, with $T_{i,k,m}$ representing the observed follow-up time, $\Delta_{i,m,k}$ the censoring indicator (1 if $T_{i,m,k}$ corresponds to the event time and 0 if the censoring time), $w_{i,m,k}$ an individual-level weight and $\boldsymbol{Z}_{i,m,k}$ a $p * 1$ vector of covariates. Define $d_{m,j} = \sum_{k=1}^{K} \sum_{i=1}^{N_{m,k}} I(T_{i,m,k} = t_{m,j}, \Delta_{i,m,k} = 1)$ as the number of events at time $t_{m,j}$ from all sites. Here the function $I(a = b, c = d, \dots)$ is defined to be equal to 1 when all conditions are true and 0 otherwise.

The input dataset at site k has the following structure for stratum m:

$$\begin{matrix} w_{1,m,k} & T_{1,m,k} & Z_{1,m,k,1} & \dots & Z_{1,m,k,p} & \Delta_{1,m,k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{N_{m,k},m,k} & T_{N_{m,k},m,k} & Z_{N_{m,k},m,k,1} & \dots & Z_{N_{m,k},m,k,p} & \Delta_{N_{m,k},m,k} \end{matrix} \quad (19)$$

Under a stratified Cox model, the hazard function for subjects at site k within stratum m for covariate level $\boldsymbol{Z}_{i,m,k}$ is assumed to have the following form:

$$h_m(t|\boldsymbol{Z}_{i,m,k}) = \exp(\boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k})\ h_m^{(0)}(t) \quad (20)$$

where $\boldsymbol{\beta}$ is an unknown $p * 1$ vector of regression coefficients. In the special case of $M = 1$, the model in Equation (2) reduces to a non-stratified Cox model. Another important special case occurs when Data Partner identifier is one of the stratification variables. We consider this case further below.

We use a Newton-Raphson algorithm to calculate the partial likelihood estimator of the regression coefficients $\boldsymbol{\beta}$. To apply this algorithm in DDNs, the log-likelihood $l(\boldsymbol{\beta})$, gradient $\boldsymbol{g}(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$, and the Hessian matrix $\boldsymbol{H}(\boldsymbol{\beta}) = \frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}$ must be expressed in terms of aggregated quantities from each Data Partner. Let's first define the quantities that have to be calculated at each site k in each stratum m.

Define:

$$\boldsymbol{d}_{m,j,k}^{(l)} = \sum_{i=1}^{N_{m,k}} I(T_{i,m,k} = t_{m,j}, \Delta_{i,m,k} = 1)\, w_{i,m,k} \boldsymbol{Z}_{i,m,k}^{l} \tag{21}$$

$$\boldsymbol{S}_{m,j,k}^{(l)}(\boldsymbol{\beta}) = \sum_{i=1}^{N_{m,k}} I\left(T_{i,m,k} \geq t_{m,j}\right) w_{i,m,k} \exp\left(\boldsymbol{\beta}^{T} \boldsymbol{Z}_{i,m,k}\right) \boldsymbol{Z}_{i,m,k}^{l} \tag{22}$$

Where $l = 0{,}1$ for $\boldsymbol{d}_{m,j,k}^{(l)}$ and $l = 0{,}1{,}2$ for $\boldsymbol{S}_{m,j,k}^{(l)}(\boldsymbol{\beta})$, $\boldsymbol{Z}_{i,m,k}^{0} = 1$, $\boldsymbol{Z}_{i,m,k}^{1} = \boldsymbol{Z}_{i,m,k}$, $\boldsymbol{Z}_{i,m,k}^{2} = \boldsymbol{Z}_{i,m,k}\, \boldsymbol{Z}^{\boldsymbol{T}}{}_{i,m,k}$ such that $\boldsymbol{d}_{m,j,k}^{(l)}$ and $\boldsymbol{S}_{m,j,k}^{(l)}(\boldsymbol{\beta})$ are scalars when $l = 0$, vectors with dimension p when $l = 1$ and $\boldsymbol{S}_{m,j,k}^{(2)}(\boldsymbol{\beta})$ is a matrix with dimensions $p * p$ when $l = 2$.

Below we use a notation in which an absence of an index in a matrix implies summation over that index. For example,

$$\boldsymbol{d}_{m,j}^{(l)} = \sum_{k} \boldsymbol{d}_{m,j,k}^{(l)} \qquad \boldsymbol{S}_{m,j}^{(l)} = \sum_{k} \boldsymbol{S}_{m,j,k}^{(l)} \tag{23}$$

Note, that when the list of stratification variables (index m) includes a Data Partner identifier represented by index k, the summation over k does not change the results in Equation (23) because there is only one possible value of k at a given m. However, in the more general case, summation over site index k is necessary.

The log-likelihood $l(\boldsymbol{\beta})$, gradient $\boldsymbol{g}(\boldsymbol{\beta})$ and the Hessian matrix $\boldsymbol{H}(\boldsymbol{\beta})$ can be written in terms of these summarized quantities:

$$l(\boldsymbol{\beta}) = \sum_{m=1}^{M} l_m(\boldsymbol{\beta}) \qquad \boldsymbol{g}(\boldsymbol{\beta}) = \sum_{m=1}^{M} \boldsymbol{g}_m(\boldsymbol{\beta}) \qquad \boldsymbol{H}(\boldsymbol{\beta}) = \sum_{m=1}^{M} \boldsymbol{H}_m(\boldsymbol{\beta}) \tag{24}$$

where

$$l_m(\boldsymbol{\beta}) = \sum_{j} \left\{ \boldsymbol{\beta}^{T} \boldsymbol{d}_{m,j}^{(\mathbf{1})} - d_{m,j}^{(0)} \log S_{m,j}^{(0)}(\boldsymbol{\beta}) \right\} \tag{25}$$

$$\boldsymbol{g}_m(\boldsymbol{\beta}) = \sum_{j} \left\{ \boldsymbol{d}_{m,j}^{(\mathbf{1})} - d_{m,j}^{(0)} \frac{\boldsymbol{S}_{m,j}^{(1)}(\boldsymbol{\beta})}{S_{m,j}^{(0)}(\boldsymbol{\beta})} \right\} \tag{26}$$

$$\boldsymbol{H}_m(\boldsymbol{\beta}) = -\sum_{j} d_{m,j}^{(0)} \left\{ \frac{\boldsymbol{S}_{m,j}^{(2)}(\boldsymbol{\beta})}{S_{m,j}^{(0)}(\boldsymbol{\beta})} - \frac{\boldsymbol{S}_{m,j}^{(1)}(\boldsymbol{\beta}) * \left[\boldsymbol{S}_{m,j}^{(1)}(\boldsymbol{\beta})\right]^{\boldsymbol{T}}}{\left[S_{m,j}^{(0)}(\boldsymbol{\beta})\right]^{2}} \right\} \tag{27}$$

In general, these representations imply that the Newton-Raphson algorithm can be executed such that the summarized matrices $\boldsymbol{d}_{m,j,k}^{(l)}$ and $\boldsymbol{S}_{m,j,k}^{(l)}(\boldsymbol{\beta})$ are computed at each Data Partner site and transferred to the analysis center. The size of a dataset to store these matrices for all j depends on the number of event times J_m for stratum m. For example, the dataset to store all matrices $\boldsymbol{S}_{m,j,k}^{(2)}(\boldsymbol{\beta})$ for stratum m has $p * p * J_m$ data elements. This can result in the need to transfer a significant amount of data when

the number of event times is large. There may also be a significant risk from a privacy perspective because the number of observations contributing to the computations for each event time can be small.

However, the calculations can be done much more efficiently and with much smaller privacy risk when one stratifies on a set of variables that includes the Data Partner identifier. In multi-database studies, a stratified Cox model, stratified by Data Partner identifier, is more realistic than assuming a common baseline hazard function for all Data Partners. In this case, the summation over time event index j in Equations (25) – (27) can be done at the Data Partners, resulting in the transfer of much smaller datasets to the analysis center. Specifically, only a dataset with matrices $l_m(\boldsymbol{\beta})$, $\boldsymbol{g}_m(\boldsymbol{\beta})$, and $\boldsymbol{H}_m(\boldsymbol{\beta})$, which are not dependent on the number of events times, need to be transferred to the analysis center. For example, the contribution to the Hessian matrix $\boldsymbol{H}_m(\boldsymbol{\beta})$ has dimension $p * p$.

The partial likelihood estimator of $\boldsymbol{\beta}$ is obtained by the Newton-Raphson algorithm, which on each iteration n solves:

$$-\boldsymbol{H}(\boldsymbol{\beta}_n)(\boldsymbol{\beta}_{n+1} - \boldsymbol{\beta}_n) = \boldsymbol{g}(\boldsymbol{\beta}_n) \tag{28}$$

for $\boldsymbol{\beta}_{n+1}$ such that

$$\boldsymbol{\beta}_{n+1} = \boldsymbol{\beta}_n - \boldsymbol{H}^{-1}(\boldsymbol{\beta}_n)\boldsymbol{g}(\boldsymbol{\beta}_n) \tag{29}$$

based on an initial starting value $\boldsymbol{\beta}_0$ and iterating until a convergence criterion is met. Our goal is to solve these equations using only Base SAS and SAS/STAT modules as the SAS matrix manipulation module SAS/IML is licensed separately. From a computational perspective, the main challenge for solving Equation (28) for $\boldsymbol{\beta}_{n+1}$ is matrix inversion. Below we will illustrate how PROC REG (part of the SAS/STAT modules) can be used to solve the system of the Newton-Raphson linear Equation (28). Let us consider a system of linear equations

$$A\boldsymbol{b} = \mathbf{c} \tag{30}$$

where $\mathbf{A}$ is a symmetric, positive definite matrix with dimensions $p * p$, c is a vector with dimension p and b, an unknown coefficient vector. PROC REG can be used to solve a system of linear equations of the form of Equation (30) for $\boldsymbol{b}$ by passing in a SSCP TYPE dataset in the form of:

$$\boldsymbol{SSCP} = \begin{pmatrix} \boldsymbol{A} & \mathbf{c}^{\mathbf{T}} \\ \mathbf{c} & const \end{pmatrix} \tag{31}$$

When we pass a dataset in the form of Equation (31) into PROC REG, the solution to the system of linear equations of the form of Equation (30), is $\boldsymbol{b} = \mathbf{A}^{-1}\,\mathbf{c}$ which can be obtained by specifying the output dataset option in the PROC REG procedure. The diagonal element $const$ in row $p + 1$ and column $p + 1$ only affects the "regression" R-squared and has no effect on deriving $\boldsymbol{b}$. We use a very large number for this diagonal element ($const = 10^{12}$) to ensure that PROC REG does not produce a note in the log that R-squared is negative.

In our companion paper [44], we showed how this general capability of PROC REG with input dataset TYPE=SSCP can be used to implement linear regression and iteratively reweighted least squares for generalized linear models without the need for PROC IML. Although we cannot use iteratively reweighted least squares for Cox regression, we can exploit the capability of PROC REG to solve a system of linear equations. Specifically, the gradient vector $\boldsymbol{g}(\boldsymbol{\beta}_n)$ has length p and the negative of the Hessian matrix $\mathbf{H} = -\mathbf{I}$ is symmetric and positive definite with dimension $* p$, which is close to the partial

likelihood estimate $\widehat{\boldsymbol{\beta}}$. Thus, we can use the above described approach to solve the Newton-Raphson Equation (28) by setting $\mathbf{A} = \mathbf{I}(\boldsymbol{\beta}_n) = -\boldsymbol{H}(\boldsymbol{\beta}_n)$ and $\mathbf{c} = \boldsymbol{g}(\boldsymbol{\beta}_n)$. For a given iteration n, the solution produced by PROC REG is:

$$\boldsymbol{b}_n = \mathbf{I}^{-1}(\boldsymbol{\beta}_n)\boldsymbol{g}(\boldsymbol{\beta}_n) \tag{32}$$

where $\boldsymbol{b}_n = \boldsymbol{\beta}_{n+1} - \boldsymbol{\beta}_n$. Using this solution, the next iteration of $\boldsymbol{\beta}^{n+1}$ can be computed as:

$$\boldsymbol{\beta}_{n+1} = \boldsymbol{\beta}_n + \boldsymbol{b}_n \tag{33}$$

In addition to coefficients $\boldsymbol{b}_n$, PROC REG also outputs the inverse, $\mathbf{I}^{-1}$. The value of the matrix $\mathbf{I}^{-1}(\boldsymbol{\beta}) = -\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}$ evaluated at the final partial likelihood estimate $\boldsymbol{\beta} = \widehat{\boldsymbol{\beta}}$ gives us the estimated covariance matrix:

$$\widehat{cov}(\widehat{\boldsymbol{\beta}}) = \mathbf{I}^{-1}(\widehat{\boldsymbol{\beta}})$$

Below we summarize our computational algorithm. The algorithm uses two different computational paths, which we refer to as *Case (a)* and *Case (b)*. *Case (a)* is implemented when the user specifies a stratified Cox model ($M > 1$) and the Data Partner identifier variable *dp_cd* is included in the list of stratification variables. *Case (b)* is implemented when the user specifies a non-stratified Cox model ($M = 1$), or a stratified Cox model ($M > 1$) but the Data Partner identifier variable *dp_cd* is not included in the list of stratification variables.

4) For each iteration $n + 1$ at each Data Partner site k, calculate matrices $\boldsymbol{d}^{(l)}_{m,j,k}(\boldsymbol{\beta}_n)$ and $\boldsymbol{S}^{(l)}_{m,j,k}(\boldsymbol{\beta}_n)$ using Equations (21) and (22) based on initial starting value $\boldsymbol{\beta}_0$.

   For *Case (a):* Calculate stratum-specific contributions to the log-likelihood $l_m(\boldsymbol{\beta}_n)$, gradient $\boldsymbol{g}_m(\boldsymbol{\beta}_n)$, and Hessian matrix $\boldsymbol{H}_m(\boldsymbol{\beta}_n)$ using Equations (25)- (27). In this case, these contributions can be calculated separately at each site and transferred to the analysis center, because the variable *dp_cd* is a stratification variable.

   For *Case (b)*: Bring matrices $\boldsymbol{d}^{(l)}_{m,j,k}(\boldsymbol{\beta}_n)$ and $\boldsymbol{S}^{(l)}_{m,j,k}(\boldsymbol{\beta}_n)$ from each site k to the analysis center.

5) For each iteration $n + 1$ at the analysis center.

   For *Case (a)*: Sum the stratum-specific contributions $l_m(\boldsymbol{\beta}_n)$, $\boldsymbol{g}_m(\boldsymbol{\beta}_n)$, and $\boldsymbol{H}_m(\boldsymbol{\beta}_n)$ to obtain the log-likelihood $l(\boldsymbol{\beta}_n)$, gradient $\boldsymbol{g}(\boldsymbol{\beta}_n)$, and Hessian matrix $\boldsymbol{H}(\boldsymbol{\beta}_n)$ using Equation (24).

   For *Case (b)*: Sum contributions from all sites to obtain $\boldsymbol{d}^{(l)}_{m,j}$ and $\boldsymbol{S}^{(l)}_{m,j}$ using Equation (23). Calculate the stratum-specific contributions to the log-likelihood $l_m(\boldsymbol{\beta}_n)$, gradient $\boldsymbol{g}_m(\boldsymbol{\beta}_n)$, and Hessian matrix $\boldsymbol{H}_m(\boldsymbol{\beta}_n)$ using Equations (25) - (27). Then sum the stratum-specific contributions $l_m(\boldsymbol{\beta}_n)$, $\boldsymbol{g}_m(\boldsymbol{\beta}_n)$, and $\boldsymbol{H}_m(\boldsymbol{\beta}_n)$ to obtain the log-likelihood $l(\boldsymbol{\beta}_n)$, gradient $\boldsymbol{g}(\boldsymbol{\beta}_n)$, and Hessian matrix $\boldsymbol{H}(\boldsymbol{\beta}_n)$ using Equation (24).

6) At the analysis center, construct the SSCP matrix as shown in Equation (31) using $\mathbf{A} = \mathbf{I}(\boldsymbol{\beta}_n) = -\boldsymbol{H}(\boldsymbol{\beta}_n)$ and $\mathbf{c} = \boldsymbol{g}(\boldsymbol{\beta}_n)$ and solve the system of linear equations of the form in Equation (30) using PROC REG as described above. This involves a series of steps implemented in the utility macro ***%solve_linear_equations_reg*** (part of the package for the analysis center). The macro takes datasets with $\mathbf{I}(\boldsymbol{\beta}_n)$ and $\boldsymbol{g}(\boldsymbol{\beta}_n)$ as inputs, constructs the appropriate SSCP-type dataset, feeds it into PROC REG, and outputs two datasets: one with the solution to the Newton-Raphson equation for

$\boldsymbol{b}_n = \boldsymbol{\beta}_{n+1} - \boldsymbol{\beta}_n$ and one containing the inverse matrix $\mathbf{I}^{-1}(\boldsymbol{\beta}_n)$. The latter is only used in the final iteration to estimate the covariance matrix.

7) Repeat steps 1 to 3 until convergence is achieved at the iteration $n+1$ that meets the convergence criterion, $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}_{n+1}$

After convergence is achieved, an additional iteration of steps 1 to 4 is necessary to calculate the covariance matrix of parameter estimates $\widehat{cov}(\widehat{\boldsymbol{\beta}}) = \mathbf{I}^{-1}(\boldsymbol{\beta}_{n+1})$. The additional iteration is required because at iteration n we do not know the matrix $\mathbf{I}(\boldsymbol{\beta}_{n+1})$, we only know the matrix $\mathbf{I}(\boldsymbol{\beta}_n)$.

2. Residuals and Survival Function

Akaike Information Criterion is defined as:

$$AIC = -2l(\widehat{\boldsymbol{\beta}}) + 2p \tag{34}$$

Bayesian Information Criterion is defined as:

$$BIC = -2l(\widehat{\boldsymbol{\beta}}) + p\ln(\sum_{i,m,k} \Delta_{i,m,k}) \tag{35}$$

Estimators for cumulative baseline hazard function (minus log of baseline survival function).

Breslow estimator:

$$h_{cum}^{(m,0)}(T) = \sum_{\boldsymbol{j}} \frac{d_{m,j}}{S_{m,j}} I(T \geq t_{m,j}) \tag{36}$$

Fleming-Harrington Estimator for Efron approximation:

$$h_{cum}^{(m,0)}(T) = \sum_{\boldsymbol{j}} \sum_{s=1}^{d_{j,m}} \frac{1}{S_{m,j,s}^{(E)}} I(T \geq t_{m,j}) \tag{37}$$

Note that both cumulative baseline hazard estimators change only at event times $T = t_{m,j}$ and stay constant in between event times.

Cumulative hazard function (minus log of survival function):

$$h_{cum}^{m}(T_{i,m,k}, \boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k}) = \exp(\boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k})\ h_{cum}^{(m,0)}(T_{i,m,k}) \tag{38}$$

Survival function:

$$S_{surv}^{m}(T_{i,m,k}) = \exp\left(-h_{cum}^{m}(T_{i,m,k}, \boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k})\right) \tag{39}$$

Martingale residuals:

$$M_{i,m,k} = \Delta_{i,m,k} - h_{cum}^{m}(T_{i,m,k}, \boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k}) \tag{40}$$

Deviance residuals:

$$D_{i,m,k} = sign\left(M_{i,m,k}\right)\sqrt{2\left[-M_{i,m,k} - \Delta_{i,m,k,} log\left(\Delta_{i,m,k} - M_{i,m,k}\right)\right]} \quad (41)$$

It is useful to plot Martingale or Deviance residuals against $\boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k}$ or some continuous independent variable using some smoothing algorithm (Loess, average by some interval). The random scatter plot around 0 with no trend indicates that linear assumption for the log of hazard as function of independent variables is reasonable.

C. EFRON APPROXIMATION FOR COX DISTRIBUTED REGRESSION ANALYSIS

The Efron approximation provides a correction to the Breslow approximation when there are a relatively large number of ties. For this approximation, in addition to site-specific matrices $\boldsymbol{d}_{m,j,k}^{(l)}(\boldsymbol{\beta}_n)$ and $\boldsymbol{S}_{m,j,k}^{(l)}(\boldsymbol{\beta}_n)$ defined in the context of the Breslow approximation in Appendix B, one also needs to also calculate the following matrices:

$$\boldsymbol{Q}_{m,j,k}^{(l)}(\boldsymbol{\beta}) = \sum_{i=1}^{n_{m,k}} I(T_{i,m,k} = t_{m,j}, \Delta_{i,m,k} = 1)\, w_{i,m,k} \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k}\right) \boldsymbol{Z}_{i,m,k}^{l} \quad (42)$$

$$\boldsymbol{Q}_{m,j,k}^{(l)}(\boldsymbol{\beta}) = \sum_{i=1}^{n_{m,k}} I(T_{i,m,k} = t_{m,j}, \Delta_{i,m,k} = 1)\, w_{i,m,k} \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_{i,m,k}^{l}\right)$$

where, for $l = 0,1,2$, $\boldsymbol{Z}_{i,m,k}^{0} = 1$, $\boldsymbol{Z}_{i,m,k}^{1} = \boldsymbol{Z}_{i,m,k}$, $\boldsymbol{Z}_{i,m,k}^{2} = \boldsymbol{Z}_{i,m,k}\, \boldsymbol{Z}^{\boldsymbol{T}}{}_{i,m,k}$. such that $\boldsymbol{Q}_{m,j,k}^{(l)}(\boldsymbol{\beta})$ is a scalar when $l = 0$, a vector with dimension p when $l = 1$ and a matrix with dimensions $p * p$ when $l = 2$.

In the formulas below, a variable without a subscript implies a sum over that subscript. In particular, the absence of the *k* index implies summation over all Data Partners. For example:

$$\boldsymbol{Q}_{m,j}^{(l)}(\boldsymbol{\beta}) = \sum\nolimits_{k} \boldsymbol{Q}_{m,j,k}^{(l)}(\boldsymbol{\beta}) \quad (43)$$

We also define the following additional matrices:

$$\boldsymbol{S}_{m,j,k,s}^{(l,E)}(\boldsymbol{\beta}) = \boldsymbol{S}_{m,j,k}^{(l)}(\boldsymbol{\beta}) - \frac{s-1}{d_{m,j}}\ \boldsymbol{Q}_{m,j,k}^{(l,E)}(\boldsymbol{\beta}) \quad (44)$$

Using these matrices and $\boldsymbol{d}_{m,j,k}^{(l)}$ from Equation (21) the partial log-likelihood $l^{(E)}(\boldsymbol{\beta})$, gradient vector $\boldsymbol{g}^{(E)}(\boldsymbol{\beta})$, and the Hessian matrix $\boldsymbol{H}^{(E)}(\boldsymbol{\beta})$ can be calculated under the Efron approximation as follows:

$$l^{(E)}(\boldsymbol{\beta}) = \textstyle\sum_{m=1}^{M} l_m^{(E)}(\boldsymbol{\beta}) \quad (45)$$

$$\boldsymbol{g}^{(E)}(\boldsymbol{\beta}) = \textstyle\sum_{m=1}^{M} \boldsymbol{g}_m^{(E)}(\boldsymbol{\beta})$$

$$\boldsymbol{H}^{(E)}(\boldsymbol{\beta}) = \sum_{m=1}^{M} \boldsymbol{H}_m^{(E)}(\boldsymbol{\beta})$$

$$l_m^{(E)}(\boldsymbol{\beta}) = \sum_j \left\{ \boldsymbol{\beta}^T \boldsymbol{d}_{m,j}^{(1)} - \frac{d_{m,j}^{(0)}}{d_{m,j}} \sum_{s=1}^{d_{m,j}} \log S_{m,j,s}^{(0,E)}(\boldsymbol{\beta}) \right\} \quad (46)$$

$$\boldsymbol{g}_m^{(E)}(\boldsymbol{\beta}) = \sum_j \left\{ \boldsymbol{d}_{m,j}^{(1)} - \frac{d_{m,j}^{(0)}}{d_{m,j}} \sum_{s=1}^{d_{m,j}} \frac{\boldsymbol{S}_{m,j,s}^{(1,E)}(\boldsymbol{\beta})}{S_{m,j,s}^{E}(\boldsymbol{\beta})} \right\} \quad (47)$$

$$\boldsymbol{H}_m^{(E)}(\boldsymbol{\beta}) = -\sum_j \frac{d_{m,j}^{(0)}}{d_{m,j}} \sum_{s=1}^{d_{m,j}} \left\{ \frac{\boldsymbol{S}_{m,j,s}^{(2,E)}(\boldsymbol{\beta})}{S_{m,j,s}^{E}(\boldsymbol{\beta})} - \frac{\boldsymbol{S}_{m,j,s}^{(1,E)}(\boldsymbol{\beta}) * \left[\boldsymbol{S}_{m,j,s}^{(1,E)}(\boldsymbol{\beta})\right]^T}{\left[S_{m,j,s}^{E}(\boldsymbol{\beta})\right]^2} \right\} \quad (48)$$

Note that the only difference between the scalar $d_{m,j}^{(0)}$ and the number of events $d_{m,j}$ used in the above equations is that the former is calculated using weights (see Equation 21). When $w_{i,m,k} = 1$ these quantities are the same for all subjects.

The main steps of our DRA computational algorithm using the Efron approximation are similar to the ones described in the context of the Breslow approximation Appendix B. The only difference is that one needs to use matrices $\boldsymbol{S}_{m,j,k,s}^{(l,E)}(\boldsymbol{\beta})$ instead of matrices $\boldsymbol{S}_{m,j.k}^{(l)}(\boldsymbol{\beta})$ and perform an additional summation over the index s when calculating the log-likelihood, gradient, and Hessian matrix using equations (46) – (48).

D. LINEAR REGRESSION ON VERTICALLY PARTITIONED DATA

Let assume for simplicity that the data are partitioned vertically between only two sites (Data Partners): DP1 and DP2 (dp_cd=1, 2). This seems to be the most likely application of the vertical regression. The theoretical generalization to multiple sites is straightforward. The site DP1 has analytic dataset with n independent variables $X_{1,i,\alpha}$ and the site DP2 has m independent variables $X_{2,i,\alpha}$ and dependent variable Y. Here $i = 1, \dots, N$ denote a subject (observation), α is a variable index: α=1,…n for DP1 and α=(n+1)…(n+1+m) for DP2. Below we will use vector notation $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ with the matrix $\boldsymbol{Z}_1 = (\boldsymbol{1} \,||\, \boldsymbol{X_1})$ including the intercept (column of 1) and the matrix $\boldsymbol{Z}_2 = (\boldsymbol{X_2} \,||\, \boldsymbol{Y})$ including the dependent variable (the operator || performs vertical concatenation of matrices). The first matrix has dimension $\boldsymbol{Z}_1 \in M_{N\times(n+1)}(\mathrm{R})$ and the second matrix has dimension $\boldsymbol{Z}_2 \in M_{N\times(m+1)}(\mathrm{R})$. In a typical application, the number of parameters n, m is orders of magnitude smaller than the number of observations N. In addition, we assume that: a) analytic datasets share a primary key (e.g. PatID, SSN or their hashed/encrypted values) b) the datasets are sorted on the primary key variable and c) have the same number of records with non-missing independent and dependent variables. It is unlikely that raw DPs data satisfy all these conditions but we assume that these issues are resolved when the analytic datasets are prepared for regression.

We are interested in performing linear regression analysis on the datasets that combines data from both Data Partners:

$$\boldsymbol{Z} = \boldsymbol{Z}_1 \,||\, \boldsymbol{Z}_2 = \begin{pmatrix} 1 & X_{1,1} & \dots & X_{1,n} & X_{1,n+1} & \dots & X_{1,n+m} & Y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{N,1} & \dots & X_{N,n} & X_{N,n+1} & \dots & X_{N,n+m} & Y_N \end{pmatrix}$$

The problem of linear regression analysis can be reduced to the calculation of SSCP matrix:

$$\boldsymbol{SSCP}(\boldsymbol{Z}) = \boldsymbol{Z}'\boldsymbol{Z} = \begin{pmatrix} N & \sum_i \boldsymbol{X_i} & \sum_i \boldsymbol{Y_i} \\ \sum_i \boldsymbol{X_i} & \sum_i \boldsymbol{X_i'X_i} & \sum_i \boldsymbol{X_i'Y_i} \\ \sum_i \boldsymbol{Y_i} & \sum_i \boldsymbol{X_i'Y_i} & \sum_i \boldsymbol{Y_i'Y_i} \end{pmatrix}$$

The dimensions of SSCP matrix are much smaller than the dimensions of the matrix $\boldsymbol{Z}$. The dimensions of the former are determined only by the number of parameters $\boldsymbol{SSCP}(\boldsymbol{Z}) \in M_{(n+m+2)\times(n+m+2)}(\mathrm{R})$ while the dimensions of the later depend on the number of observations $\boldsymbol{Z} \in M_{N\times(n+m+2)}(\mathrm{R})$. Thus, the SSCP matrix is highly summarized and it can be shared by Data Partners with little risk of revealing the patient level information. In addition, since the matrix is relatively small, computationally intensive matrix manipulations like matrix inversion can be performed easily on virtually any modern computer. By contrast, matrix inversion involving matrices with dimensions of order N can quickly run into the computer memory (REM) limitations even for datasets of moderate size (above 10,000 obs).

Once SSCP is known one can easily calculate virtually all standard results including regression coefficients, covariance matrix, standard errors, various goodness of fit measures and statistical tests. In SAS this is particular easy to do using PROC REG which accepts the SSCP matrix as an input dataset.

For two DPs the SSCP matrix can be calculated as:

$$\boldsymbol{SSCP}(\boldsymbol{Z}_{1,} \,||\, \boldsymbol{Z}_2) = (\boldsymbol{Z}_1 \,||\, \boldsymbol{Z}_2)'\,(\boldsymbol{Z}_1 \,||\, \boldsymbol{Z}_2) = \begin{Bmatrix} \boldsymbol{Z}_1' & \boldsymbol{Z}_1 & \boldsymbol{Z}_1' & \boldsymbol{Z}_{,2} \\ \boldsymbol{Z}_2' & \boldsymbol{Z}_1 & \boldsymbol{Z}_2' & \boldsymbol{Z}_2 \end{Bmatrix}$$

The calculation of matrices on the diagonal of the SSCP matrix is straightforward. It can be easily done at each site separately using a standard procedure (e.g. SAS PROC CORR with SSCP option). The challenge is calculating the off-diagonal element $\boldsymbol{Z}_1'\,\boldsymbol{Z}_2$. Due to privacy considerations one cannot just bring individual level data elements without performing some kind of transformation to obscure them. Several methods have been proposed to achieve secure matrix multiplication. We will discuss pros and cons of various approaches later in this document. Here we choose an approach that utilizes a semi-trusted third party which we will call the Analytic Center. The main idea of this approach was proposed by Beaver. In the context of linear regression, it was used by Du et al. 2004 (Commodity Server approach). We have modified it to simplify some of the calculations. The main difference between our approach and Du et al. 2004 is that we start with the expression for the SSCP matrix while Du et al. 2004 start with the following final expression for the regression coefficient estimates:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{Y}$$

The advantage of starting with the SSCP matrix is that it requires only one secure matrix multiplication: calculation of $\boldsymbol{Z}_1'\,\boldsymbol{Z}_2$. By contrast, starting with the expression for $\widehat{\boldsymbol{\beta}}$ requires secure matrix multiplication of several matrices and also secure matrix inversion. The secure versions of matrix operations are significantly more complicated than regular matrix operations. They also require more transfers of large datasets (datasets of order $N * n$). In addition, knowledge of the SSCP matrix allows calculation of other standard regression statistics and tests beyond just the estimates of the regression coefficients $\widehat{\boldsymbol{\beta}}$.

Below we describe the main steps for secure matrix multiplication using our proposed Analytic Center approach. The first 3 steps are the same as in Du et al 2004. Note that the Analytic Center has no access to any individual level data. Its main role is to help DPs to obscure their individual data before they exchange any data elements between themselves. The Analytic Center creates four random matrices: $\boldsymbol{R}_1 \in M_{N\times(n+1)}(\mathrm{R})$, $\boldsymbol{R}_2 \in M_{N\times(m+1)}(\mathrm{R})$, $\boldsymbol{r}_1 \in M_{(n+1)\times(m+1)}(\mathrm{R})$, $\boldsymbol{r}_2 \in M_{(n+1)\times(m+1)}(\mathrm{R})$. The first

three of these random matrices are generated independently and the fourth $\boldsymbol{r}_2$ is calculated in such a way that the matrices satisfy the following condition:

$$\boldsymbol{R}_1' \boldsymbol{R}_2 = \boldsymbol{r}_1 + \boldsymbol{r}_2$$

It is important that $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are randomly generated. It, obviously, does not matter if the matrix $\boldsymbol{r}_2$ is calculated and the matrix $\boldsymbol{r}_1$ is generated.

1. The Analytic Center sends datasets $\boldsymbol{R}_1$, $\boldsymbol{r}_1$ only to the DP1 and the datasets $\boldsymbol{R}_2$, $\boldsymbol{r}_2$ only to the DP2. As Du et al 2004 pointed out, it is possible to minimize the amount of data transferred in this step. Instead of sending large datasets $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$, the Analytic Center can just send two seed numbers which were used to generate these matrices using a pseudo random number generator. The only important thing here is that one seed number is only shared with DP1 and another only with DP2.
2. The Data Partner DP1 creates a scrambled dataset $\widehat{\boldsymbol{Z}_1} = \boldsymbol{Z}_1 + \boldsymbol{R}_1$ and sends it to the DP2 and the Data Partner DP2 creates a scrambled dataset $\widehat{\boldsymbol{Z}_2} = \boldsymbol{Z}_2 + \boldsymbol{R}_2$ and sends to the DP1.
3. The Data Partners DP1 and DP2 create new summarized datasets $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ using the following formulas:

$$\boldsymbol{t}_1 = \boldsymbol{r}_1 - \boldsymbol{R}_1' \widehat{\boldsymbol{Z}_2}$$
$$\boldsymbol{t}_2 = \boldsymbol{r}_2 + \widehat{\boldsymbol{Z}_1'} \boldsymbol{Z}_2$$

It is important to note here that at this stage in the process, the individual level data are eliminated. The resulting datasets $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ are summarized over all observations. Their dimensions are determined by the number of parameters at each DP $\boldsymbol{t}_1$, $\boldsymbol{t}_2 \in M_{(n+1)\times(m+1)}(\mathrm{R})$ rather than the number of observations N. Thus, there is relatively little risk to share these data either with one another or with the Analytic Center. In our approach, these datasets are sent to the Analytic Center and the final calculations are done there. In the paper by Du et al 2004 the individual DPs exchange the summarized datasets until they calculate all necessary quantities to get $\widehat{\boldsymbol{\beta}}$**.** In our judgment, the use of the Analytic Center at this stage is preferable. It significantly simplifies the process and reduces the number of data transfers with little risk to data privacy.

4. The Analytic Center receives the following summarized datasets from the Data Partners: $\boldsymbol{t}_1$, $\boldsymbol{t}_2$, $\boldsymbol{Z}_1' \boldsymbol{Z}_1$and $\boldsymbol{Z}_2' \boldsymbol{Z}_2$. It is easy to verify by direct substitution that

$$\boldsymbol{t}_1 + \boldsymbol{t}_2 = \boldsymbol{r}_1 - \boldsymbol{R}_1' \widehat{\boldsymbol{Z}_2} + \boldsymbol{r}_2 + \widehat{\boldsymbol{Z}_1'} \boldsymbol{Z}_2 = \boldsymbol{Z}_1' \boldsymbol{Z}_2 + \boldsymbol{r}_1 + \boldsymbol{r}_2 - \boldsymbol{R}_1' \boldsymbol{R}_2 = \boldsymbol{Z}_1' \boldsymbol{Z}_2$$

The last step is true because of constrain used in constriction of matrices: $\boldsymbol{R}_1$, $\boldsymbol{r}_1$, $\boldsymbol{R}_2$, $\boldsymbol{r}_2$. At this point, the Analytic Center has all necessary components to calculate SSCP matrix and perform complete linear regression analysis using standard regression procedure like SAS PROC REG with SSCP input.

Let's now consider the computational costs of the above secure matrix protocol. It is useful to compare it to the optimal possible cost of a matrix product calculation for vertically partitioned data. The latter is defined as a cost of computing matrix product without the privacy constrains. In that case, the data transfer cost is just the cost of downloading the dataset $\boldsymbol{Z}_1$ to DP2 (or $\boldsymbol{Z}_2$ to DP1). For $\boldsymbol{Z}_1$ the number of data elements is of order nN. By comparison, the data transfer cost of the secure matrix product with the help of Analytic Center is about twice that $(n+m)N$ (sending $\widehat{\boldsymbol{Z}_1}$ to DP2 and $\widehat{\boldsymbol{Z}_2}$ to DP1). In addition, the secure matrix product requires two matrix multiplications of large (order of $N * n$) matrices $\boldsymbol{R}_1' \widehat{\boldsymbol{Z}_2}$ and $\widehat{\boldsymbol{Z}_1'} \boldsymbol{Z}_2$ instead of just one $\boldsymbol{Z}_1' \boldsymbol{Z}_2$. However, the memory required for calculating matrix products like $\widehat{\boldsymbol{Z}_1'} \boldsymbol{Z}_2$ is the same as in the optimal case $\boldsymbol{Z}_1' \boldsymbol{Z}_2$. The important point here is that in both cases the required memory does not scale with N. This is because the matrix products like $\widehat{\boldsymbol{Z}_1'} \boldsymbol{Z}_2$

can be calculated as cross-product of all $\widehat{\boldsymbol{Z_1}}$columns with all $\boldsymbol{Z}_2$ columns in the dataset $\widehat{\boldsymbol{Z_1}}$ || $\boldsymbol{Z}_2$. In such approach the matrix $\widehat{\boldsymbol{Z_1}}$ does not need to be transposed and matrices $\widehat{\boldsymbol{Z_1}}$ and $\boldsymbol{Z}_2$ do not need to be loaded into memory. Instead, the calculations can process row by row: multiply data elements on the same row i and add them up $(\widehat{\boldsymbol{Z_1}}\,\boldsymbol{Z}_2)_{\alpha,\beta} = \sum_i (Z_{i,\alpha,1} * Z_{i,\beta,2})$. As the result, the required memory depends only on the size of the final matrix $(n * m)$ which is relatively small. The cost of downloading small matrices of order $n * m$ can be neglected in comparison to the matrices of size $N * n$. To summarize: the communication cost of the secure protocol with Analytic Center is only about twice of the optimal cost while required memory is about the same.

Let's now compare the above approach with approaches that do not involve the Analytic Center (Commodity Server). Several such approaches were suggested: Du et al 2004 (two party protocol), Karr et al 2009. The obvious advantage of these approaches is that DPs can setup the process without a third party. However, there are significant computational disadvantages. The main challenge is that these approaches scale non-linearly with the number of observations both in terms of required memory and the volume of data that must be transferred. The computational costs of the two-party approaches in Du et al 2004 and in Karr at al 2009 are similar. Below, we focus on the two-party solution described by Du et al 2004.

Their initial step requires that both parties generate a random matrix $\boldsymbol{M}$ which has not only N rows but also N columns. In addition, one of the Data Partners must invert matrix **M**. The latter has to be done in memory and both memory and CPU scale faster than N^2 for matrix inversion. After this initial step, one of the DPs has to split matrix **M** in half vertically while the other DP has to split the inverse matrix $\boldsymbol{M^{-1}}$ horizontally. The next step involves calculating matrix products: $\boldsymbol{Z}_1'\,\mathbf{M}_{top}$, $\boldsymbol{Z}_1'\,\mathbf{M}_{bottom}$, $\boldsymbol{Z}_2'\,\mathbf{M}_{right}^{-1}$, $\boldsymbol{Z}_2'\,\mathbf{M}_{left}^{-1}$ and then transferring them to the other party. The total number of data elements in these matrices is $4 * N * \frac{N}{2} = 2N^2$. The N^2scaling of the computational costs including data transfers, memory and CPU makes it difficult to apply this approach even to moderately sized datasets. To put things in prospective, consider a dataset with 20,000 observations and 10 variables at each DP. The amount of data transferred in that case would be $2 * \frac{N}{m} = 4000$ larger than the minimal required amount (total $2N^2 * 8\ Bytes = 1.6\ GB$); the amount of required memory for matrix inversion will be more than $\frac{N^2}{(n+m)^2} \approx \left(\frac{N}{(n+m)}\right)^2 = 1{,}000{,}000$ larger than minimally necessary (total $N^2 * 8 = 0.8\ GB$). I don't have experience with inverting matrices of this scale but when you Google "How to invert large matrix…" the typical recommendation is "Don't do it…, find the other way to get what you need".

The above two-party approach can be improved somewhat by splitting the data into blocks and in effect doing a combination of horizontal and vertical distributed regression. The main advantage of such an approach is that matrix inversion $\boldsymbol{M_{block}^{-1}}$ could be done on a smaller matrix. The disadvantage is that it will make the algorithm significantly more complicated. Also, there is a limit on how small a block can be used. This is because the number of records in each block must be much larger than the number of parameters $N_{block} \gg n$. Otherwise, matrix **M** will not be effective in obscuring individual level data. Thus, one still needs to do an inversion of a matrix that is at least an order of magnitude larger than in the optimal case and the same is true for the communication cost.

In my judgment, the advantages of using the Analytic Center approach for regression with vertically partitioned data should outweigh its disadvantages in most potential applications and for large analytic datasets this seems to be the only viable option.

It is instructive to compare privacy preserving distributed regression on vertically partitioned data with distributed regression on horizontally partitioned data. In both cases, the goal is to do the regression

without revealing individual level data. In the horizontal case, this can be achieved by exchanging highly summarized data. By contrast, in the vertical case one always has to exchange the individual level data and then merge them on the primary key. The best that one can do in this case is to transform individual level data in such a way that the individual data cannot be unscrambled back but necessary summarized data can be still be calculated. As a result, the communication cost of distributed regression on vertically partitioned data is significantly higher than that for horizontally partitioned data. At best, the former scales linearly with the number of observation while the latter does not depend on N at all.

Both the vertical and horizontal cases can be done with and without the Analytic Center. However, from a computational perspective, the advantages of using the Analytic Center are much more significant in the vertical case. It allows orders of magnitude improvements in all computational aspects of the process: communication costs, memory and CPU. By contrast, in the horizontal case, the computational costs with and without Analytic Center are similar and the main advantage of the Analytic Center is that it facilitates the process.

In our implementation of horizontal distributed regression, the Analytic Center and the intermediate server for data transfer (PopMedNet portal) are used together. One can use a similar approach for vertical regression as well. However, there are important differences. In the horizontal case, the data are exchanged only between the Analytic Center and DPs. There is no data exchange between DPs. In the vertical case, the scrambled individual level data ($\widehat{\boldsymbol{Z}_1}$, $\widehat{\boldsymbol{Z}_2}$) are exchanged only between DPs and it is very important that the Analytic Center cannot read these data on the portal. This is because it can unscramble the datasets $\widehat{\boldsymbol{Z}_1}$, $\widehat{\boldsymbol{Z}_2}$ using datasets $\boldsymbol{R}_1, \boldsymbol{R}_2$. One way of dealing with this issue is for DPs to encrypt the datasets $\widehat{\boldsymbol{Z}_1}$, $\widehat{\boldsymbol{Z}_2}$ using a password that they share between themselves but do not share with the Analytic Center. SAS has a built-in encryption capability that can be used for this purpose. Another approach is to not use an intermediate server for data transfer at all and, instead, use direct data transfer between DPs and also between DPs and the Analytic Center. The presence of a network firewall makes such a setup more complicated but it is doable. Most companies have so called “staging servers” to get data from their clients. For example, the HPHC receives enrollment data from their clients (employers). A staging server is accessible from the outside and it works in combination with the automatic script that moves data from a staging server to the production server inside the company’s firewall.