From Multi-Site Observational Health Data to Real World Evidence: Privacy-Preserving Distributed Algorithms

Mackenzie Edmondson
PhD Candidate in Biostatistics
Department of Biostatistics, Epidemiology, and Informatics
University of Pennsylvania, Perelman School of Medicine

November 24, 2020 – **OHDSI Community Call**

Outline

- Background
  - Privacy challenges in multi-site studies
  - Existing approaches for privacy-preserving multi-site analysis

- Our approach: Privacy-preserving Distributed Algorithms (PDA)
  - Distributed regression; surrogate likelihood method
  - ODAP and ODAH
  - Real-world use cases

- Summary
  - Newly available R package!

Background: Privacy Challenges and Existing Approaches for Multi-Site Analysis

Privacy challenges in multi-site studies

- Multi-site studies: larger sample size, improved generalizability

- HIPAA: sharing of patient protected health information (PHI) often prohibited across institutions
  - De-identified data can be shared (e.g. “limited dataset”)

- De-identified PHI susceptible to re-identification (Benitez & Malin 2010)

- Distributed Health Data Networks: no data centralization
  - Common data model
  - Analyses performed distributively without patient-level data transfer

Existing Multi-Site Analysis Approaches: Meta-Analysis

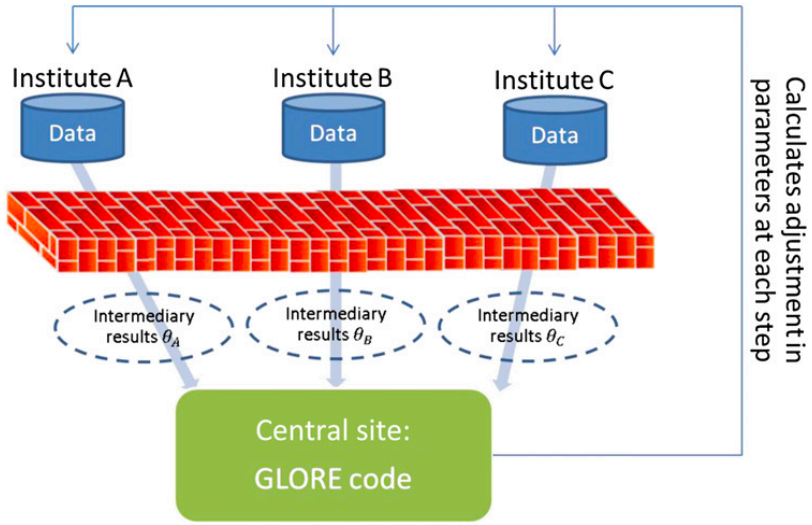
- Collaborating sites send estimated coefficients and standard errors to lead site for aggregation
- Very popular, easy to implement
  - Most common analytic method in OHDSI studies
- Biased estimation in rare-event settings
- Issues with ecological bias

Distributed Regression

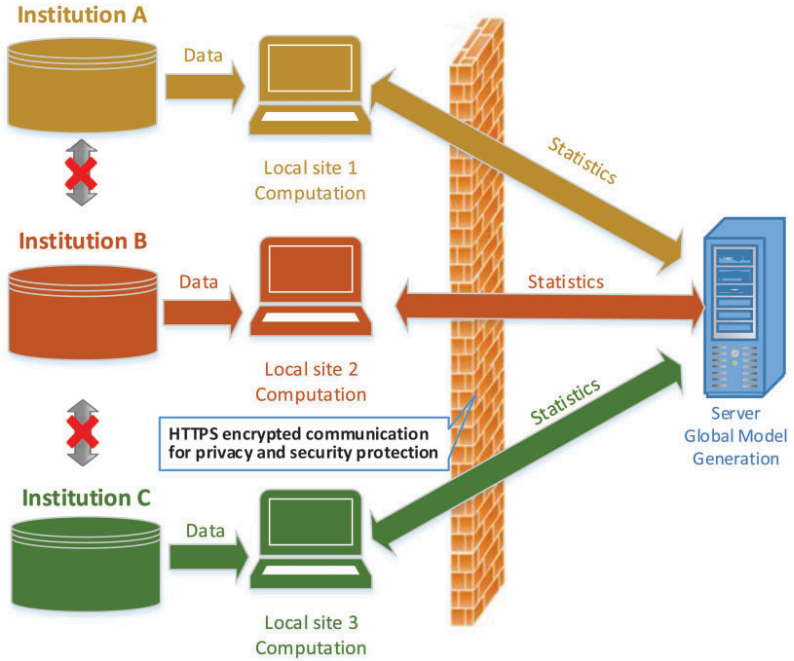
- Regression model fit in distributed fashion across sites without sharing patient-level data
- Involves aggregation of summary statistics to estimate parameters
- Multi-site distributed linear regression is **lossless** (Chen et al. 2006)
  - Estimated coefficients equivalent to those in pooled analysis
- Pooled analysis: $\widehat{\boldsymbol{\beta}}_{pooled} = (X^T X)^{-1} X^T Y$
- From each site *i*, obtain $\left(X_i^T X_i\right)$ and $X_i^T Y_i$ (aggregated summary measures)
  - $\widehat{\boldsymbol{\beta}}_{dist} = \left(\sum_i X_i^T X_i\right)^{-1} \left(\sum_i X_i^T Y_i\right) = \widehat{\boldsymbol{\beta}}_{pooled}$

Distributed Regression

- What if $\hat{\beta}$ doesn't have a closed-form solution?
- **Iterative procedures** for distributed regression
  - Newton-Raphson method
  - Also lossless
- GLORE (distributed logistic regression)
- WebDISCO (distributed Cox regression)



Wu et al. 2012, *JAMIA*

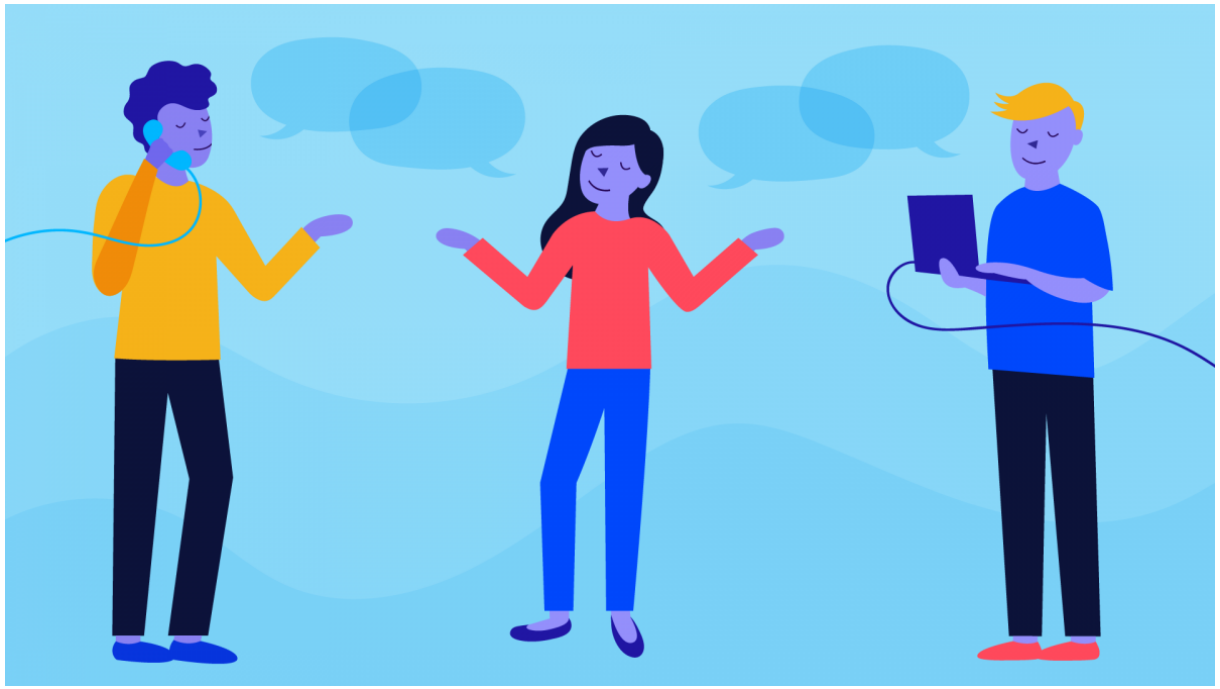


Lu et al. 2015, *JAMIA*

Distributed Regression: Limitations in Existing Approaches

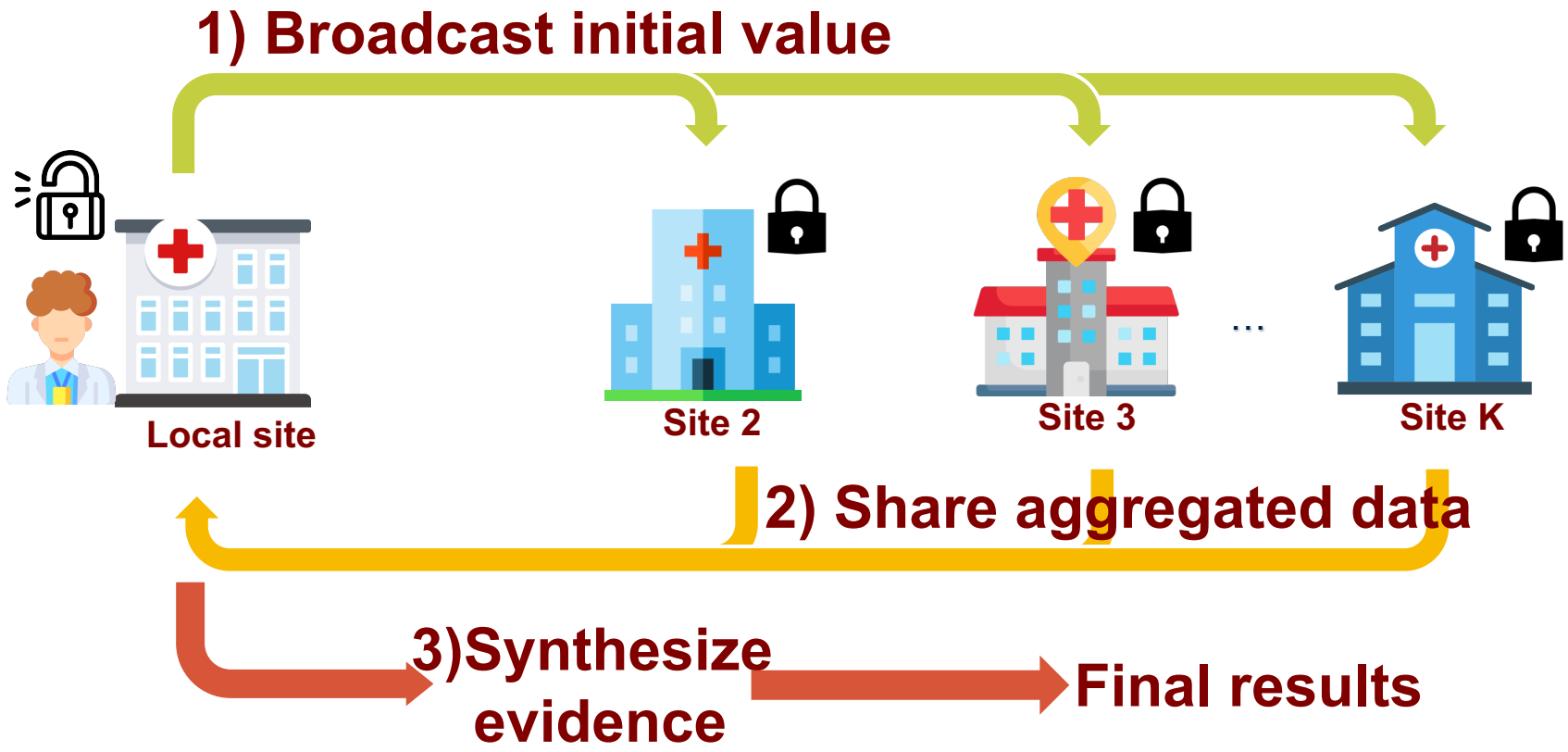
- Iterative procedures may require several rounds of communication
  - Privacy risk, even with aggregate data transfer (especially for very small data sets)
  - Inefficient, communication takes time!

- **Goal**: Can we perform distributed regression without using iterative procedure?



Our Approach: Privacy-preserving Distributed Algorithms (PDA)

PDA: Privacy-preserving Distributed Algorithms



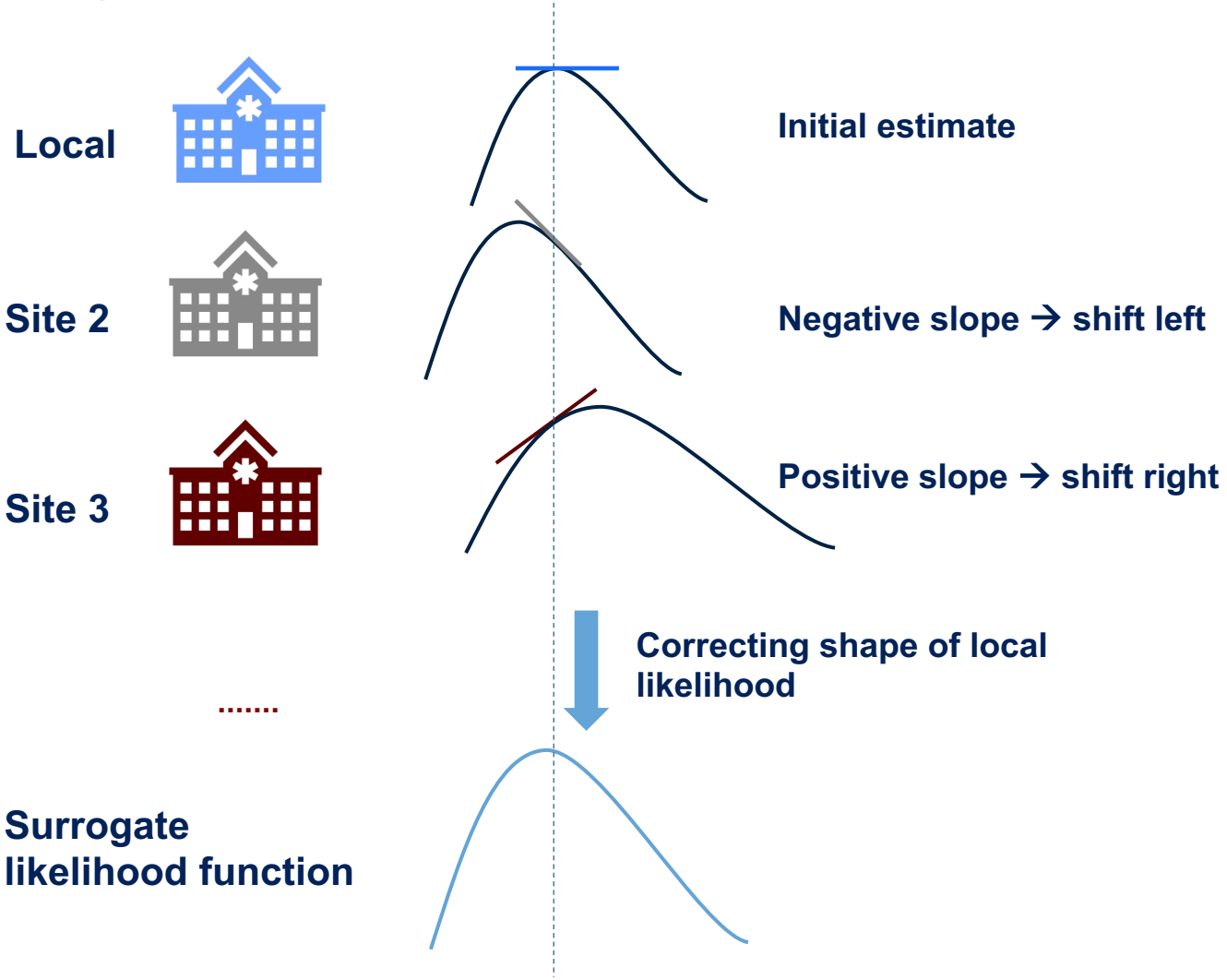
Surrogate Likelihood Estimation

- **Communication-efficient** distributed inference (Jordan et al. 2018)
  - **One-shot:** Non-iterative communication among sites

- Approximates complete data (pooled) log-likelihood using **patient-level data at only one site (local site)**
  - Aggregate information obtained from collaborating (non-local) sites
  - Not lossless, but typically closer approximation than meta-analysis
  - Uses Taylor series expansion of complete data log-likelihood
  - First-order surrogate likelihood function:

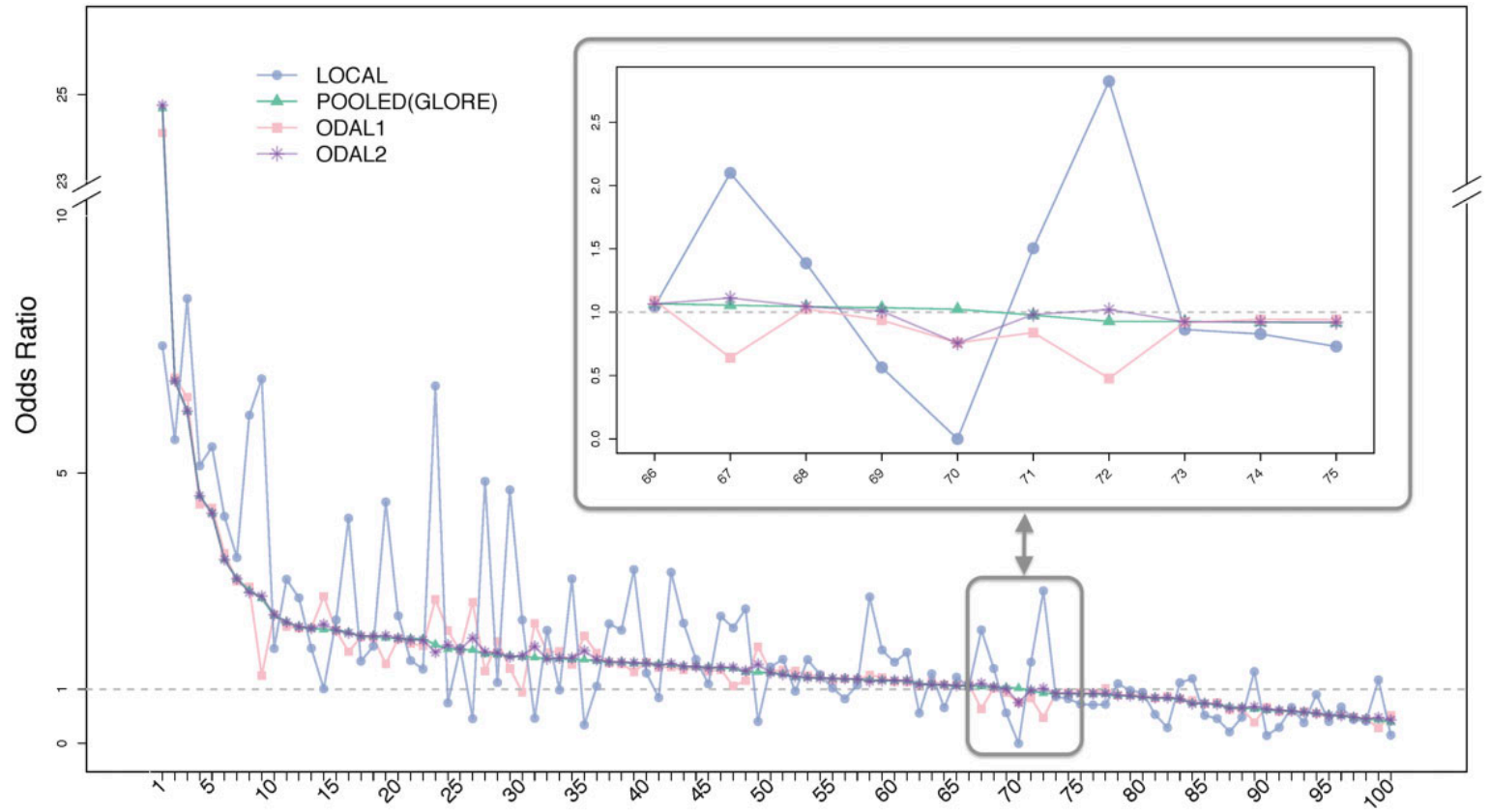
$$\tilde{L}(\beta) = L_1(\beta) + \left\{\nabla \mathrm{L}\left(\bar{\beta}\right) - \nabla L_1\left(\bar{\beta}\right)\right\}\beta$$

Surrogate Likelihood Estimation: Intuition



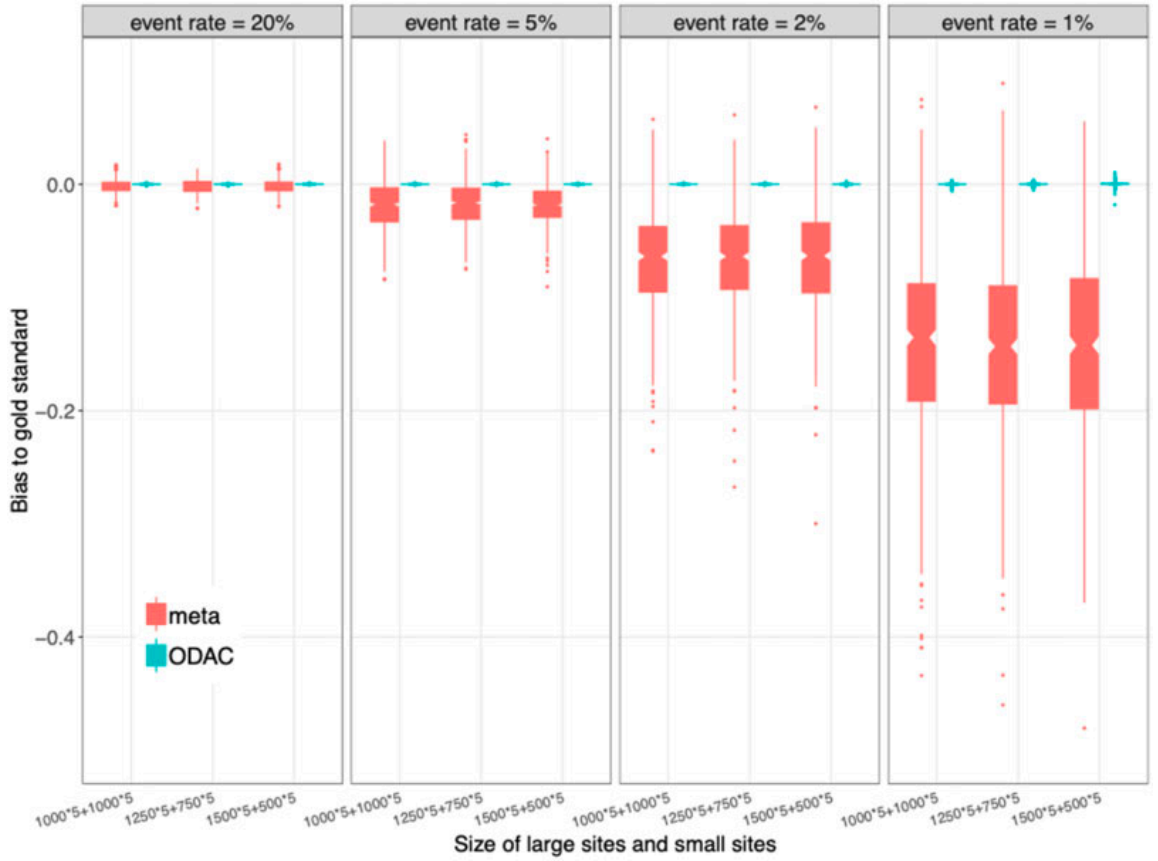
Surrogate Likelihood Estimation

- ODAL: Algorithm for performing distributed logistic regression (Duan et al. 2020)



Surrogate Likelihood Estimation: Distributed Cox Regression

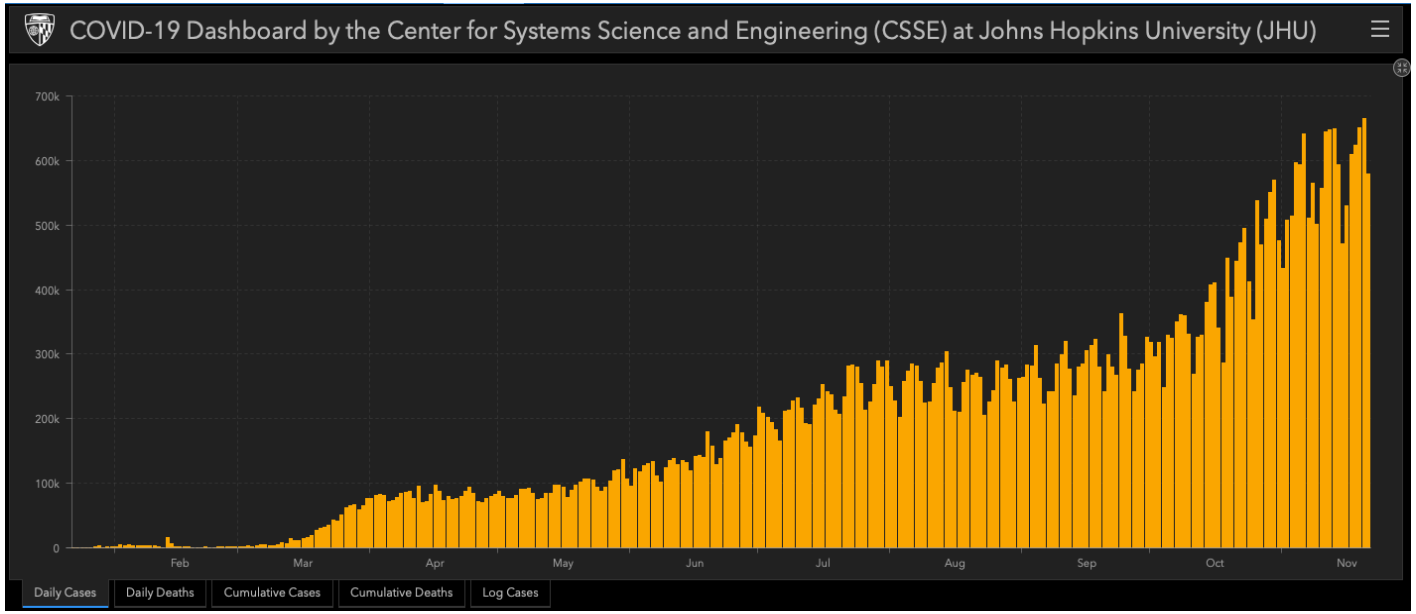
- ODAC: Algorithm for performing distributed Cox regression (Duan et al. 2020)



Communication-Efficient Distributed Regression for Count Outcomes (ODAP and ODAH)

ODAP Motivation: COVID-19 Hospitalization

- As of Sunday 11/22: > 58 million confirmed cases, 1.38 million deaths across 191 countries and territories (JHU COVID-19 Dashboard)
- Estimating demand for hospital beds crucial for contingency planning
- Length of stay (LoS) dependent on disease severity
  - Highly variable



RWD Motivation: COVID-19 Hospitalization

- Interest in characterizing association between LoS and patient characteristics
- Many individual sites with COVID-19 patient data, but typically too small for proper inference
- In a pandemic, many institutions willing to collaborate!
- **Goal:** Devise communication-efficient algorithm for modeling LOS in COVID-19 patients using data at several collaborating sites
- **Contribution:** ODAP (One-Shot Distributed Algorithm for performing Poisson regression)

ODAP: Distributed Poisson Regression Algorithm

- Count outcomes commonly modeled using Poisson regression
  - Assumption: mean = variance
  - Often in practice: mean < variance (overdispersion)
    - Poisson regression of overdispersed data results in biased standard errors (Cox 1984)

- Quasi-Poisson: account for extra variation in outcome by estimating dispersion and scaling variance
  - $E(Y_i|X_i) = \exp(X_i^T \beta) = \mu_i$
  - $Var(Y_i|X_i) = \phi\mu_i, \quad \phi > 0$

ODAP: Distributed Poisson Regression Algorithm

- Clinical data at K sites, j^{th} site has n_j unique patient records, $N = \sum_{j=1}^{K} n_j$ total patient records
- (Y_{ij}, X_{ij}): outcome, covariate vector for i^{th} subject at j^{th} site
- Pooled, complete data log-likelihood function

$$L_N(\beta) = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{n_j} Y_{ij} X_{ij}^T \beta - \exp(X_{ij}^T \beta)$$

  - Requires sharing of patient-level data across sites

ODAP: Distributed Poisson Regression Algorithm

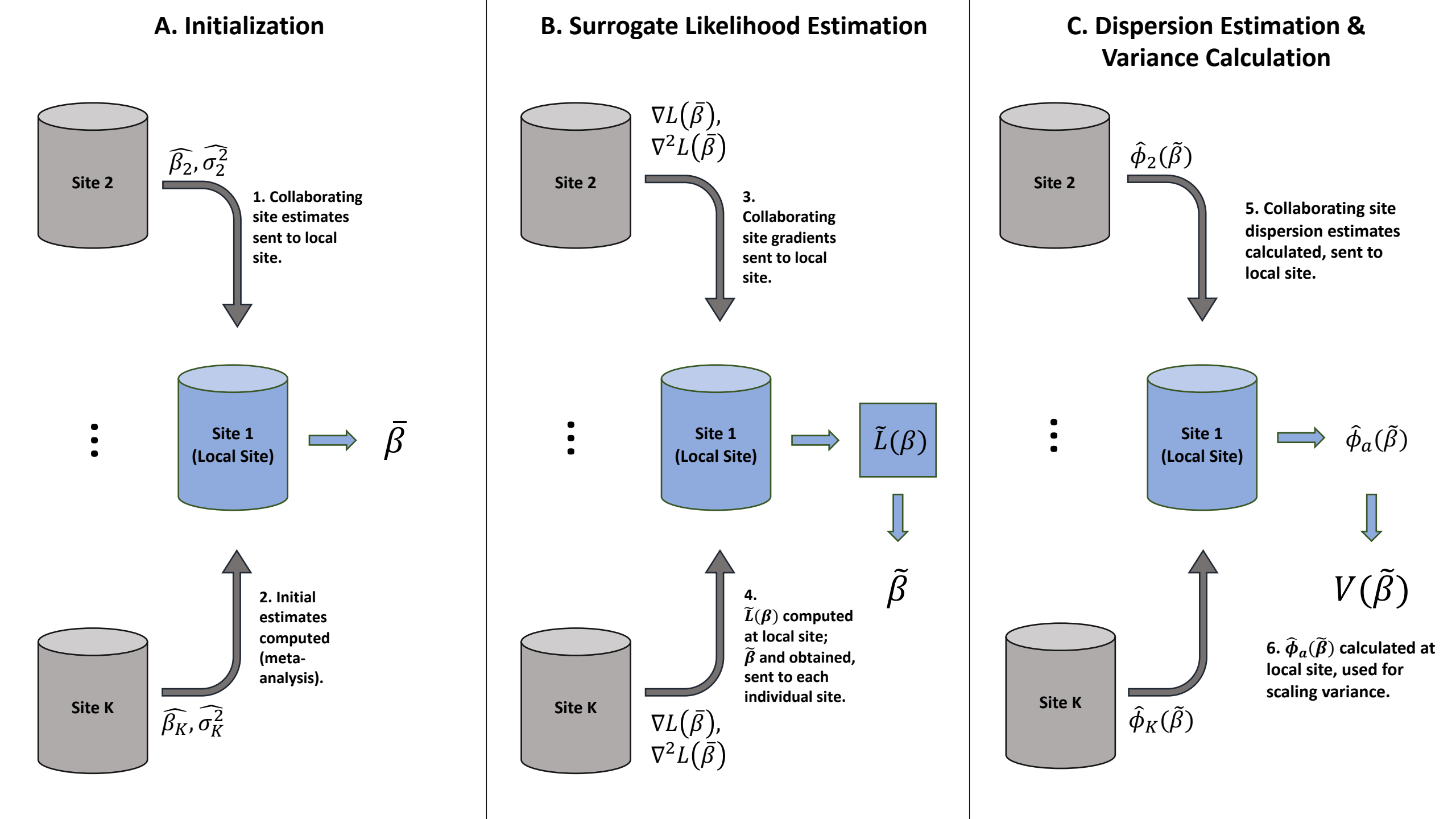
- Distributed data network: Assume we only have patient-level data at local site

- Surrogate log-likelihood function (second-order):

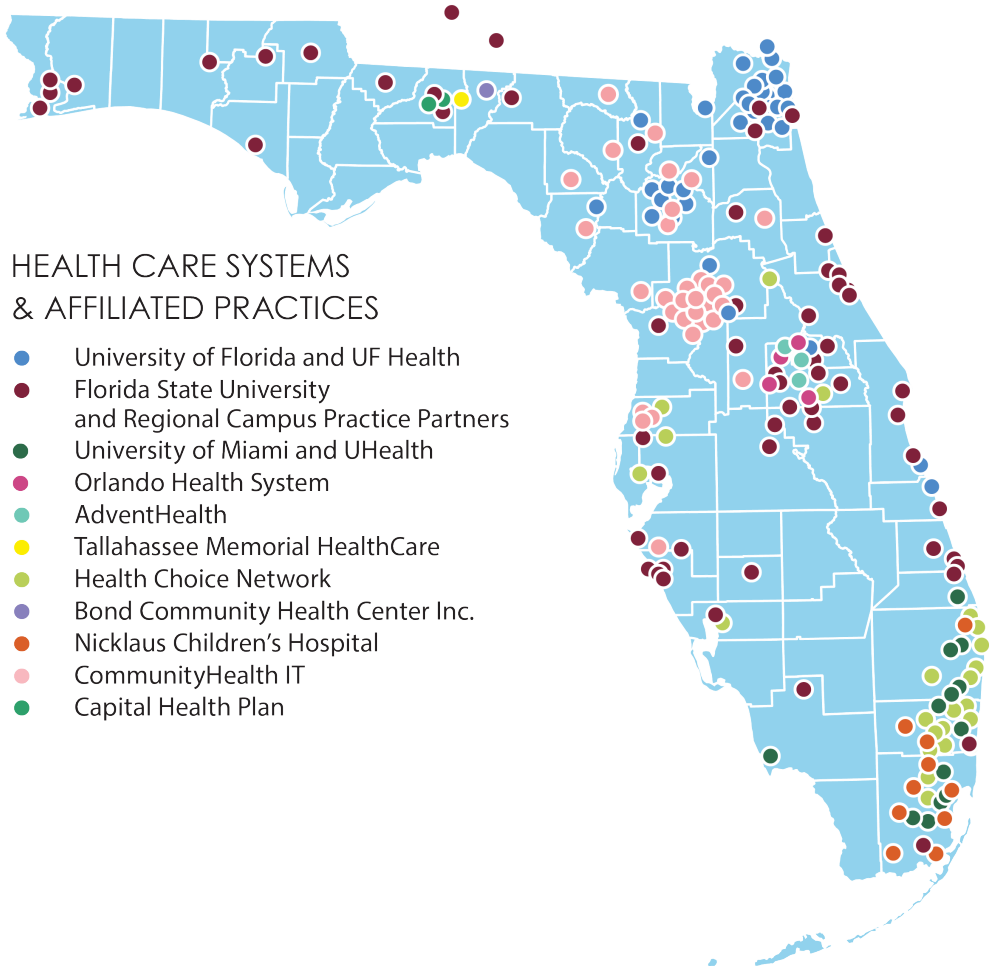
$$\tilde{L}(\boldsymbol{\beta}) = L_1(\boldsymbol{\beta}) + \{\nabla \mathrm{L_N}(\overline{\boldsymbol{\beta}}) - \nabla L_1(\overline{\boldsymbol{\beta}})\}\boldsymbol{\beta} + \frac{1}{2}(\boldsymbol{\beta} - \overline{\boldsymbol{\beta}})^T\{\nabla^2 L_N(\overline{\boldsymbol{\beta}}) - \nabla^2 L_1(\overline{\boldsymbol{\beta}})\}(\boldsymbol{\beta} - \overline{\boldsymbol{\beta}})$$

  - ∇, ∇^2: first- and second-order gradients of log-likelihood
  - $\nabla \mathrm{L_N}$: weighted average of individual site gradients
  - $\overline{\boldsymbol{\beta}}$: initial estimate for algorithm (e.g. meta-analysis estimate, local estimate)

- ODAP estimator: $\widetilde{\boldsymbol{\beta}} = \mathrm{argmax}_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta})$
- $V(\widetilde{\boldsymbol{\beta}})$: inverse Hessian scaled by overdispersion estimate $\hat{\phi}_a(\tilde{\beta})$

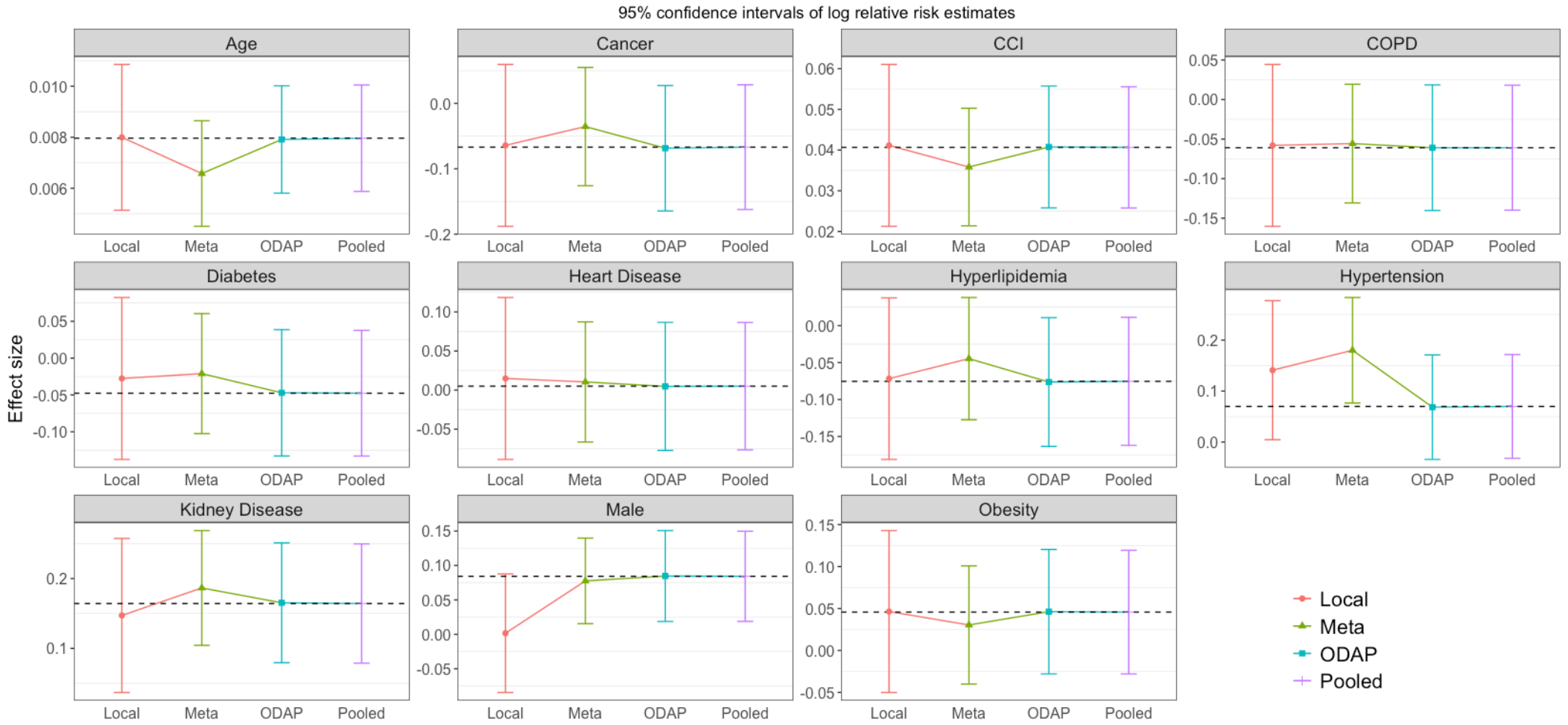

A. Initialization
Site 2
$\widehat{\beta_2}, \widehat{\sigma_2^2}$
1. Collaborating site estimates sent to local site.
Site 1 (Local Site)
$\bar{\beta}$
2. Initial estimates computed (meta-analysis).
Site K
$\widehat{\beta_K}, \widehat{\sigma_K^2}$
B. Surrogate Likelihood Estimation
Site 2
$\nabla L(\bar{\beta}), \nabla^2 L(\bar{\beta})$
3. Collaborating site gradients sent to local site.
Site 1 (Local Site)
$\tilde{L}(\beta)$
$\tilde{\beta}$
4. $\tilde{L}(\beta)$ computed at local site; $\tilde{\beta}$ and obtained, sent to each individual site.
Site K
$\nabla L(\bar{\beta}), \nabla^2 L(\bar{\beta})$
C. Dispersion Estimation & Variance Calculation
Site 2
$\hat{\phi}_2(\tilde{\beta})$
5. Collaborating site dispersion estimates calculated, sent to local site.
Site 1 (Local Site)
$\hat{\phi}_a(\tilde{\beta})$
$V(\tilde{\beta})$
6. $\hat{\phi}_a(\tilde{\beta})$ calculated at local site, used for scaling variance.
Site K
$\hat{\phi}_K(\tilde{\beta})$

Application: OneFlorida Clinical Research Consortium



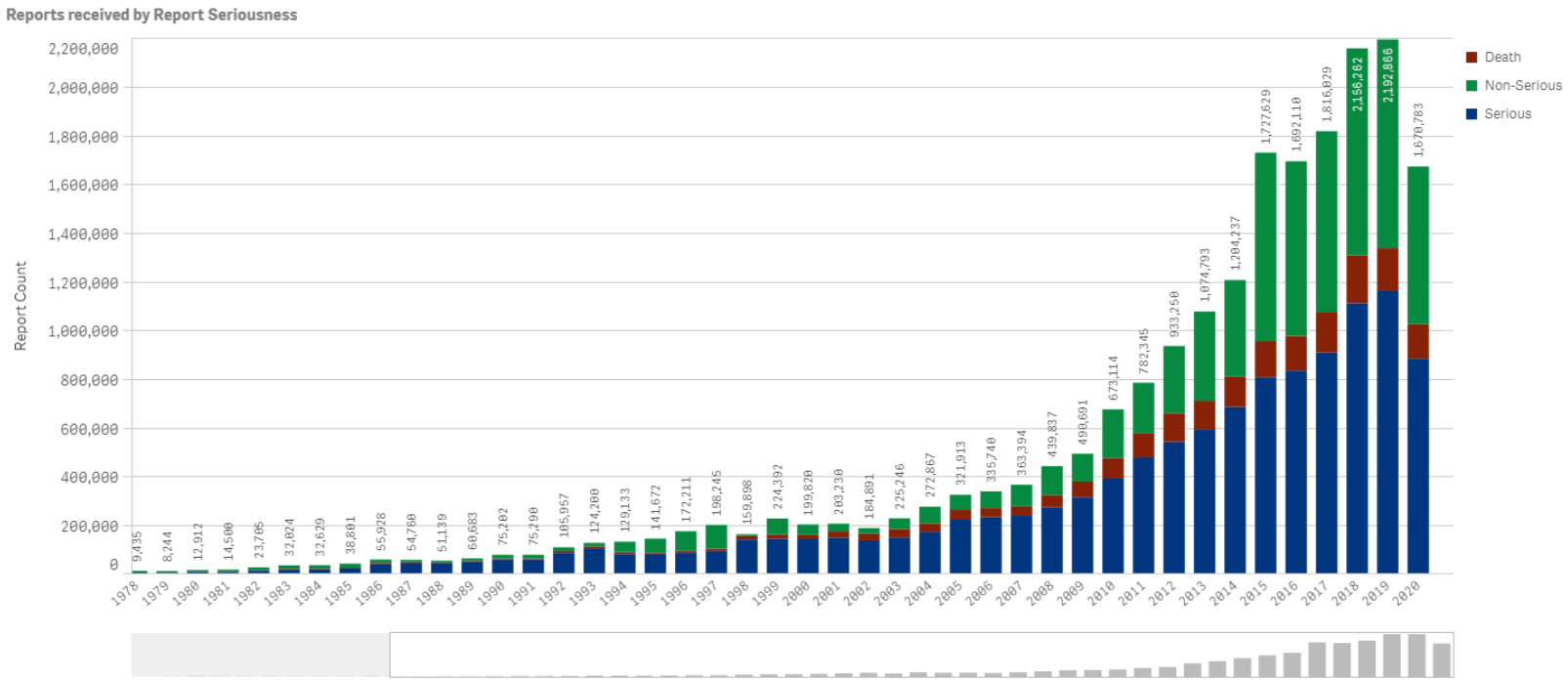
- Limited dataset, patient-level RWD of 15 million Floridians (> 50% state population)
- Centralized data
- Q: In patients hospitalized with COVID-19, which risk factors are most associated with length of stay?
- Study data: 4,212 COVID-19 patients from 4 clinical sites
- High overdispersion: $\hat{\phi} \approx 10$ ($\hat{\phi} = 1$: no dispersion)

ODAP RWD Application: Results



ODAH Motivation: Serious Adverse Events (Pharmacovigilance)

- Post-market drug safety evaluated via adverse event reporting
- Real-world example: FOLFIRI chemotherapy treatment for colorectal cancer (CC)
- Interest in modeling serious adverse event (SAE) frequency for CC patients taking FOLFIRI



FDA Adverse Events Reporting System (FAERS) Public Dashboard, obtained on 11/22/20.

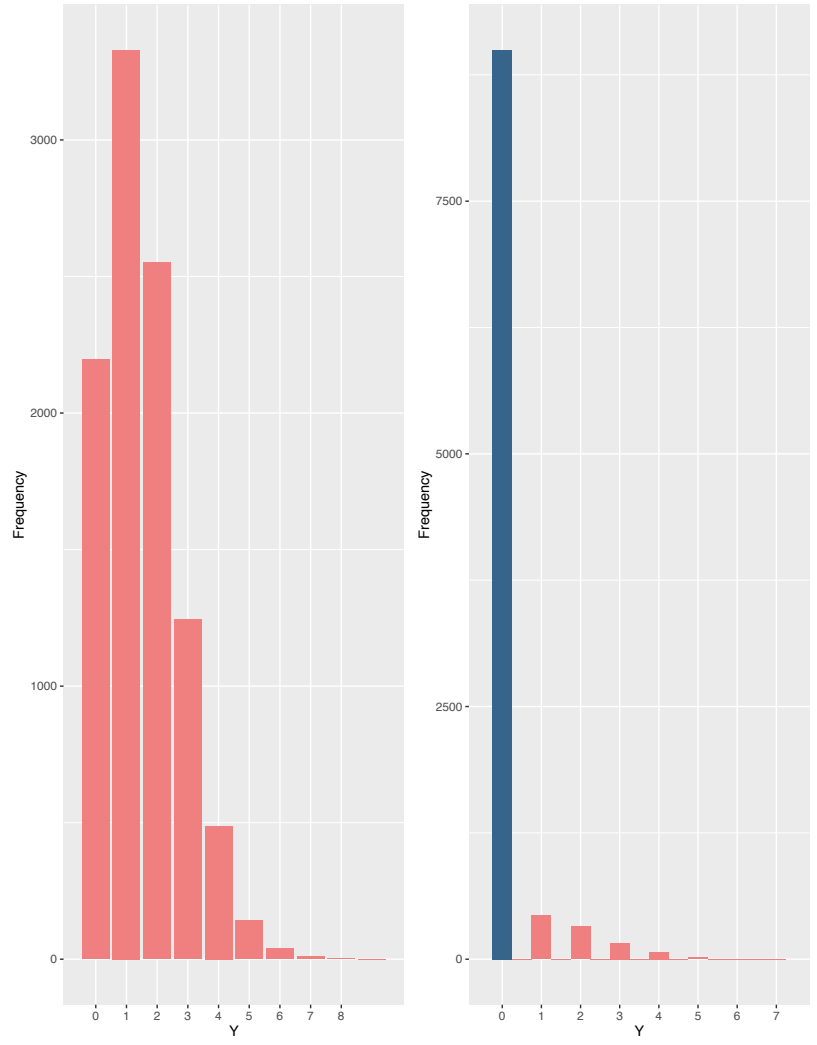
RWD Motivation: Severe Adverse Events (Pharmacovigilance)

- Most patients do not have SAE → many zero counts (e.g. > 80%)
- High variance in quantity and quality of adverse event reporting
- **Goal:** Devise communication-efficient algorithm to model SAE frequency using data at several collaborating sites
- **Contribution:** ODAH (One-Shot Distributed Algorithm for performing Hurdle regression)

ODAH: Distributed Hurdle Regression Algorithm

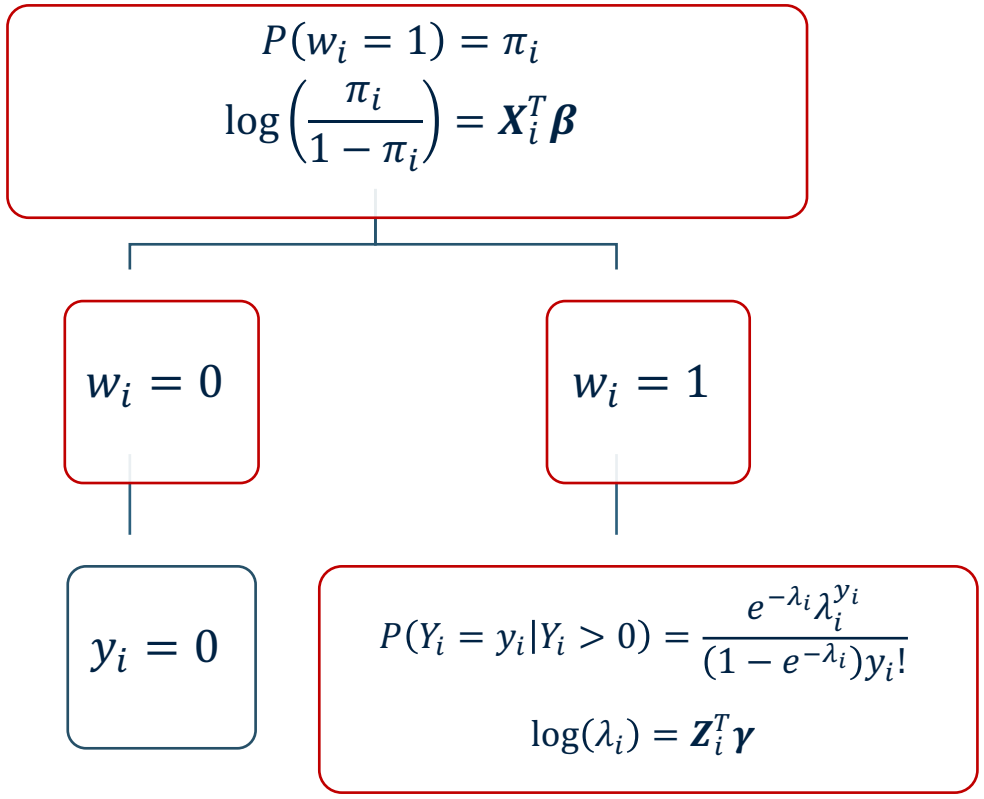
- Zero-inflation: another common feature of count data in practice
  - Excess zero counts -- more than would be expected under traditional count distribution (e.g. Poisson or Negative Binomial)
  - Often zero-inflated: length of stay, number of hospitalizations, lab tests ordered

- Methods for handling zero-inflated counts
  - Zero-inflated regression model
  - **Hurdle model**



ODAH: Distributed Hurdle Regression Algorithm

- Hurdle model: two-part model
  - "Zero" part: logistic regression
  - "Non-zero" part: zero-truncated count model (**Poisson**/Negative Binomial)
- No shared parameters: completely independent
- Interpretation differs from zero-inflated model
  - One source of zeros (sample) vs two (structural)



ODAH: Distributed Hurdle Regression Algorithm

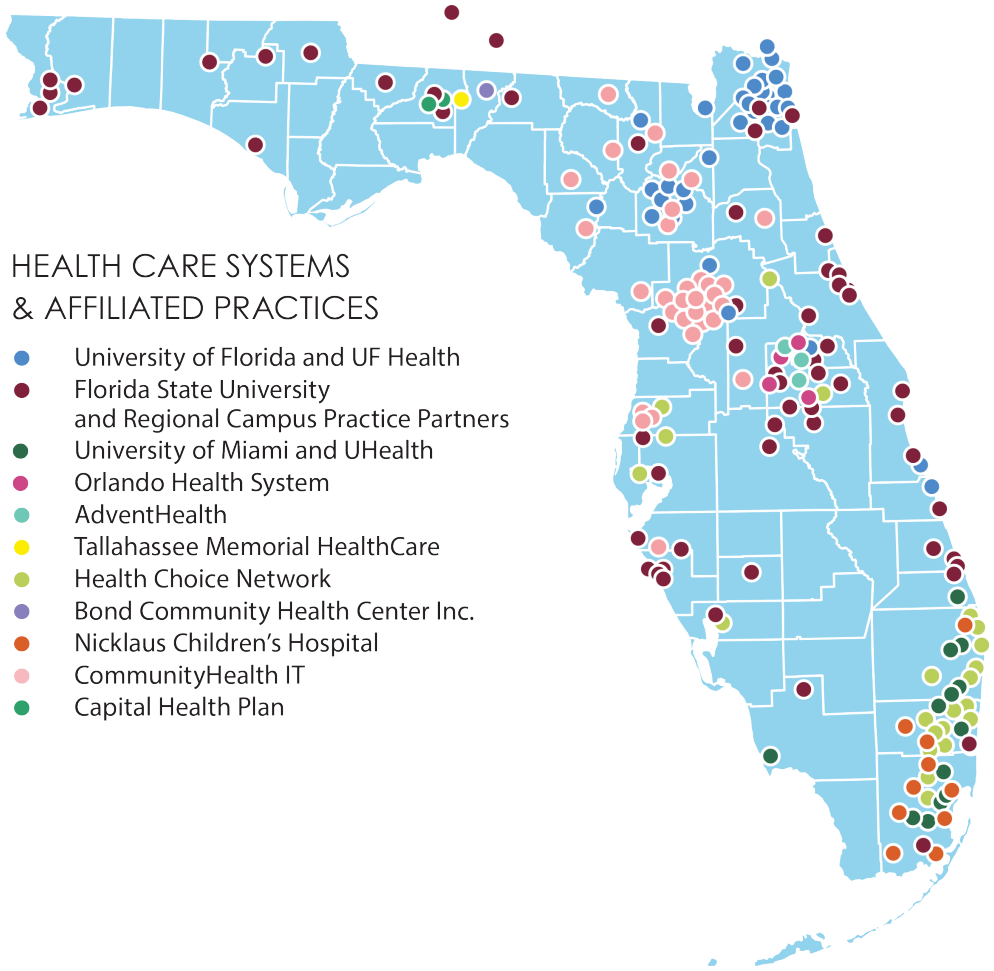
- Log-likelihood of Poisson-Logit hurdle: sum of Binomial, zero-truncated Poisson log-likelihoods

$$L(\beta, \gamma) = L_1(\beta) + L_2(\gamma)$$

- Surrogate likelihood analogous to that for ODAP
  - Now have two surrogate log-likelihood functions, one for each component

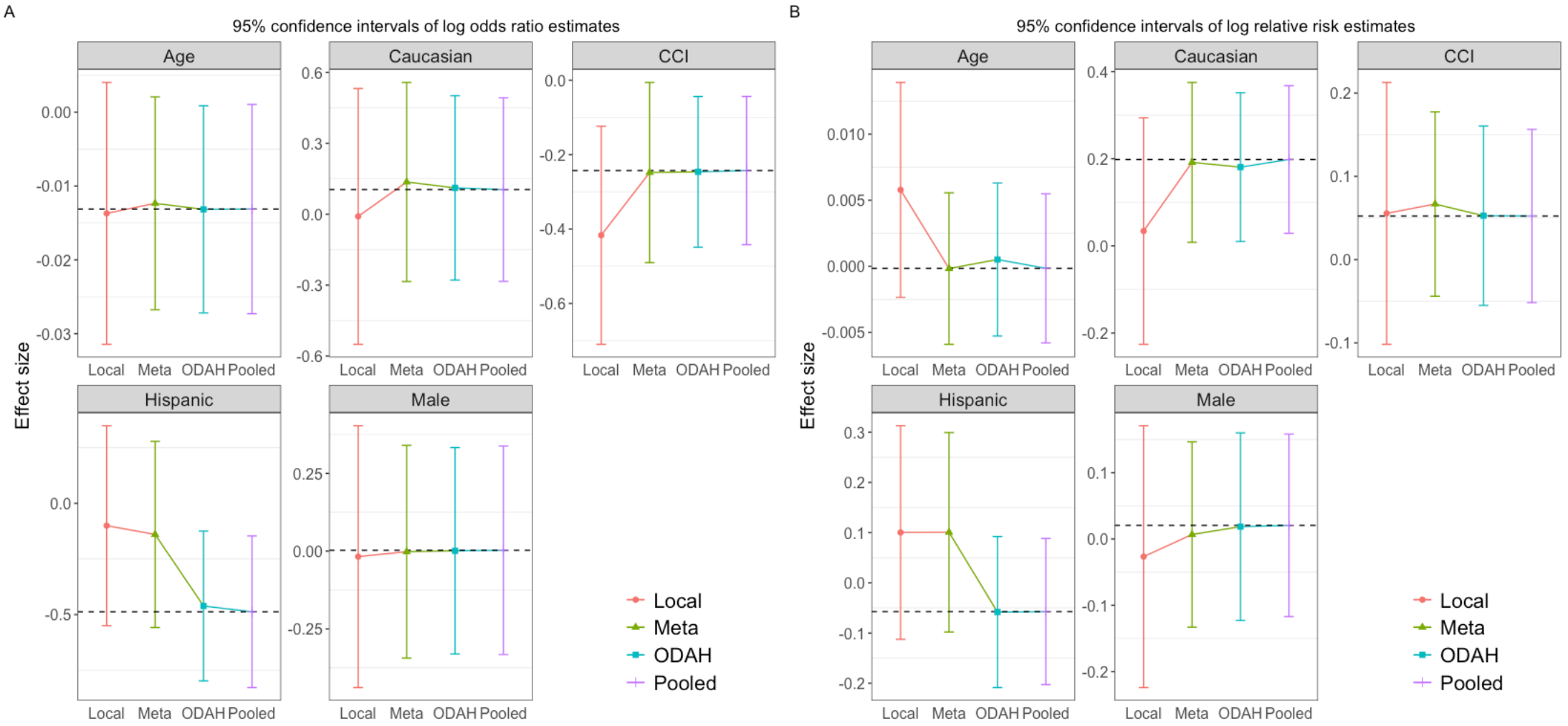
- *At most* three rounds of non-iterative communication among sites
  - Depends on choice of initial value, estimation of dispersion

Real-World Data Application: OneFlorida CRC



- Goal: Assess drug safety in terms of severe adverse event (SAE) frequency
- Q: Given demographics and risk factors, how many SAEs are expected for colorectal cancer patient receiving FOLFIRI?
- Data: 660 colorectal cancer patients taking FOLFIRI from three clinical sites

ODAH RWD Application: Results



Summary

- PDA methods are:
  - **Accurate:** High accuracy relative to pooled estimates; large advantage over meta-analysis in rare-outcome settings
  - **Safe:** At-most three rounds of communicating aggregate data; as little as one round
  - **Efficient:** Non-iterative communication among collaborating sites
- ODAP/ODAH currently assume homogeneity, where statistical model is the same across all sites
  - Estimating dispersion helps capture heterogeneity in outcome
  - Future extension: modify algorithms to further account for heterogeneity
- R package on CRAN soon and currently on GitHub; website in the works!

R package: pda

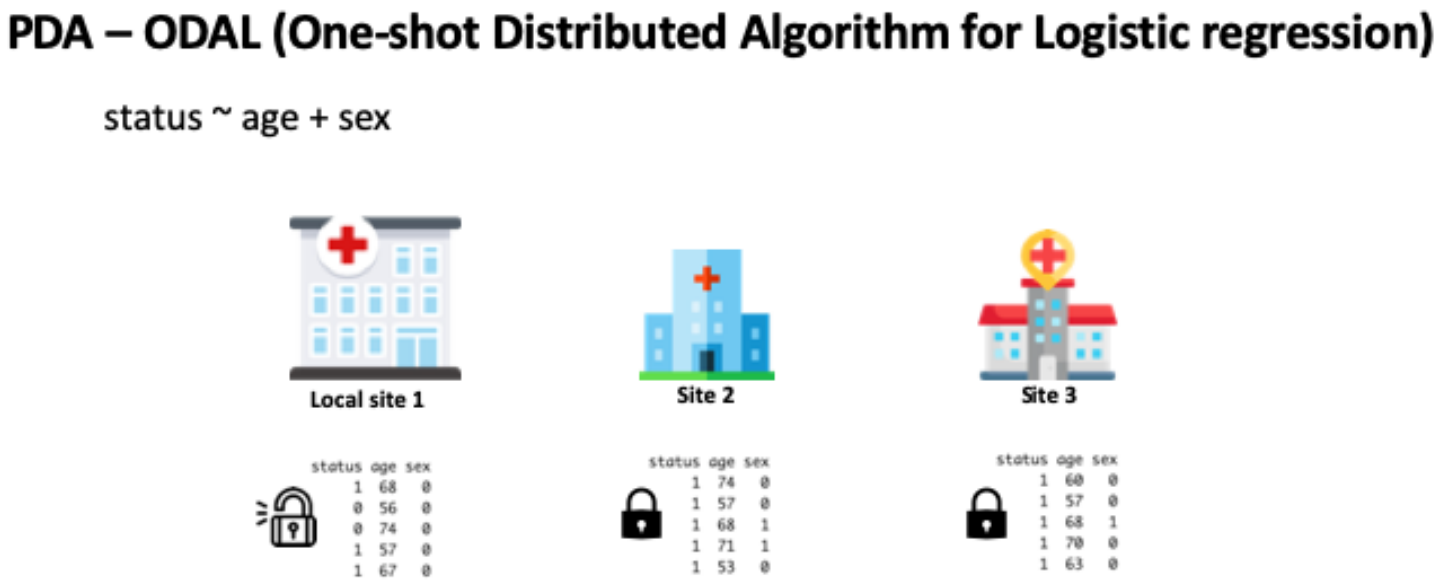
- GitHub repo: https://github.com/Penncil/pda

```r
# Install the latest version of PDA in R:
install.packages("pda")
library(pda)

# Or you can install via github:
install.packages("devtools")
library(devtools)
devtools::install_github("penncil/pda")
library(pda)
```

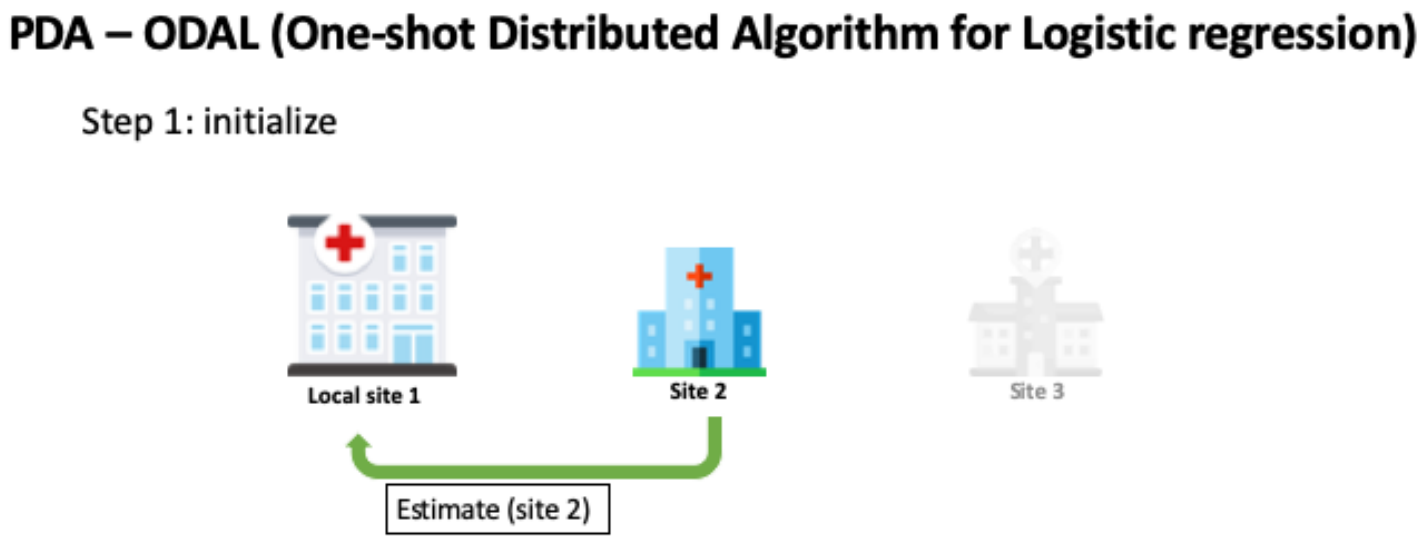
R pda Package Demo

- pda::demo(ODAL)



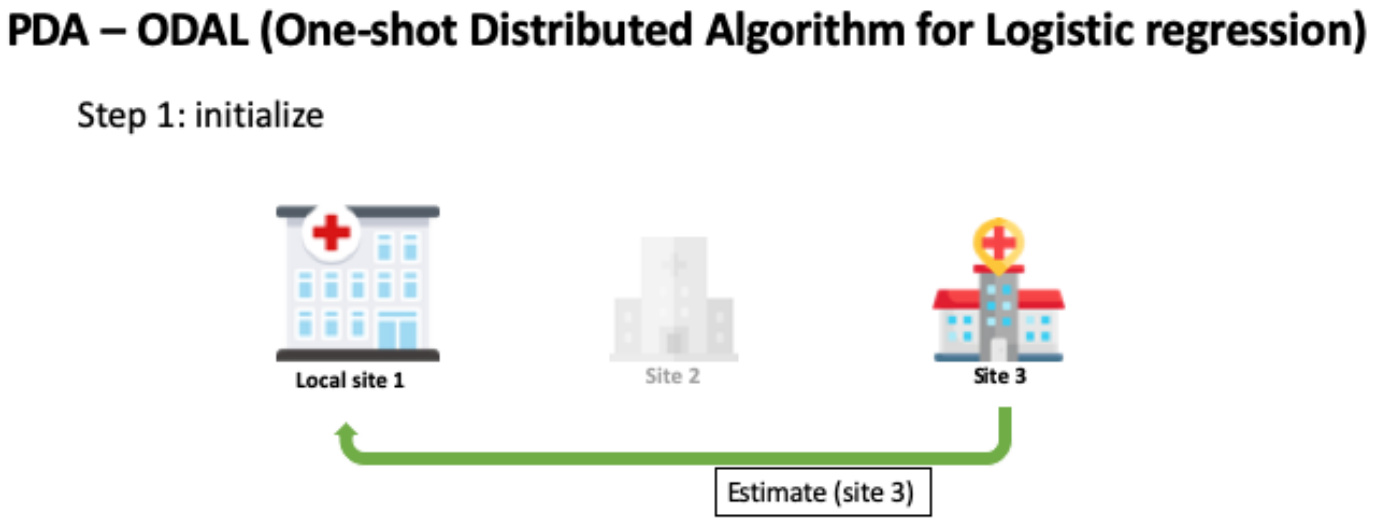
R pda Package Demo

- pda::demo(ODAL)



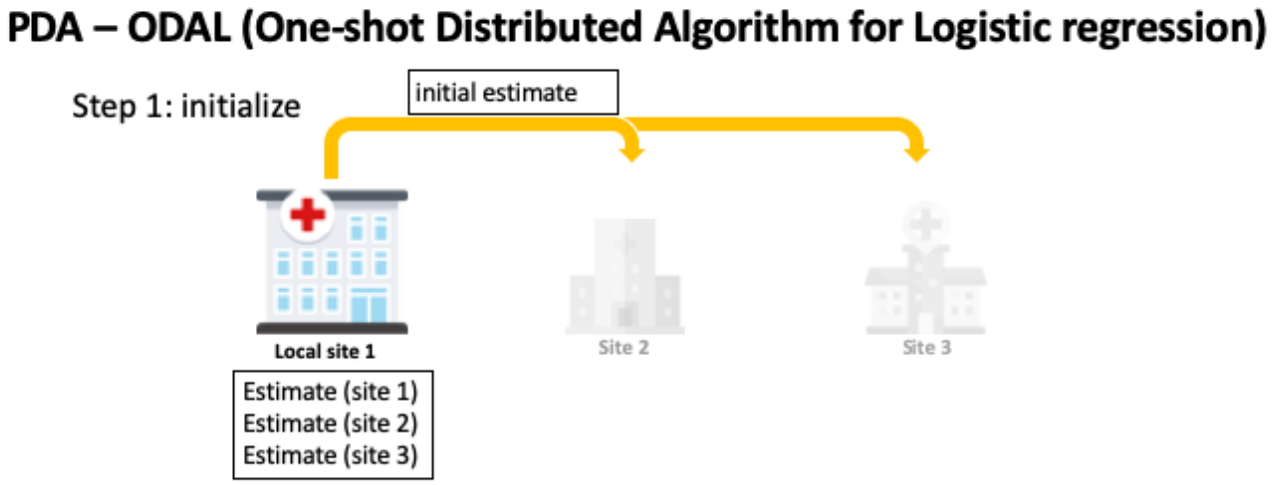
R pda Package Demo

- pda::demo(ODAL)



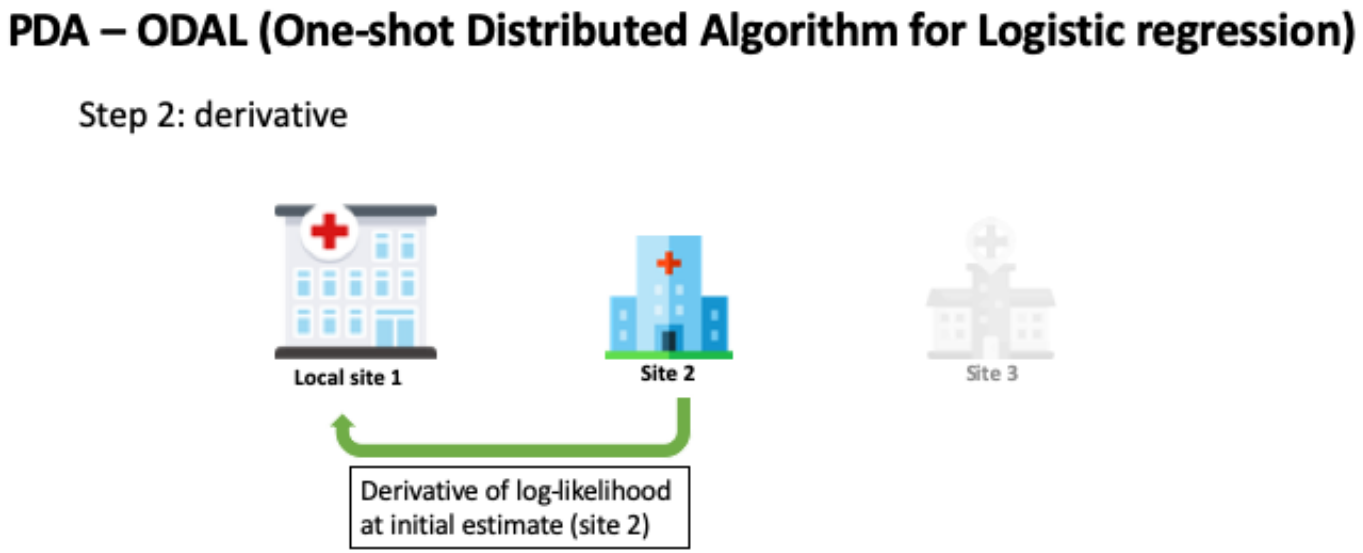
R pda Package Demo

- pda::demo(ODAL)



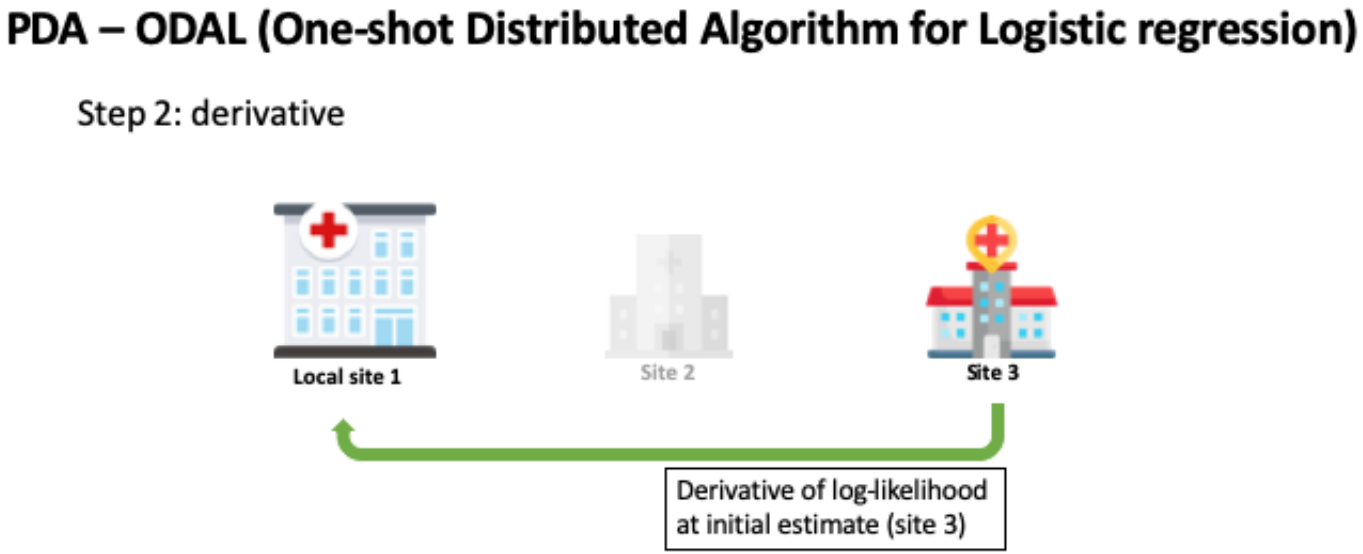
R pda Package Demo

- pda::demo(ODAL)



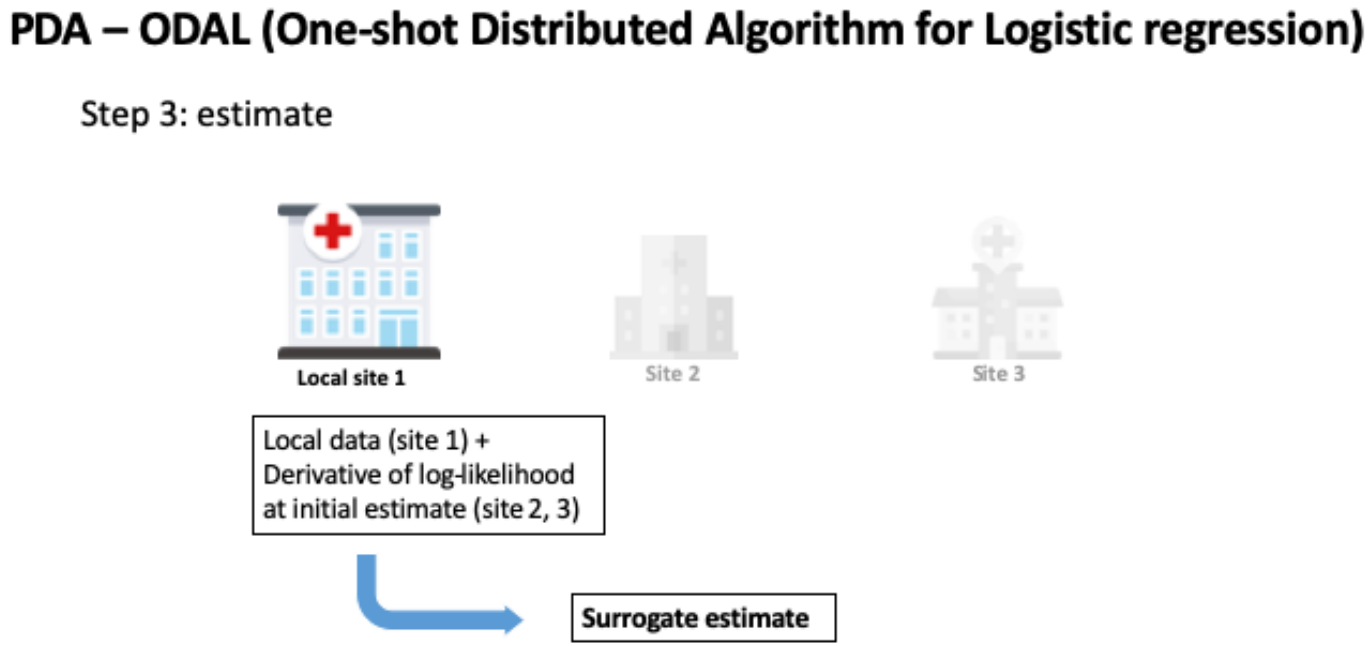
R pda Package Demo

- pda::demo(ODAL)



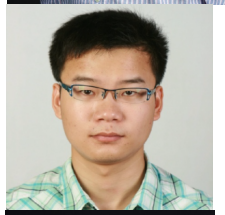
R pda Package Demo

- pda::demo(ODAL)

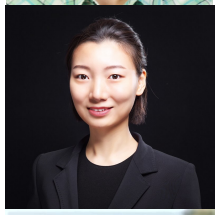


Acknowledgements

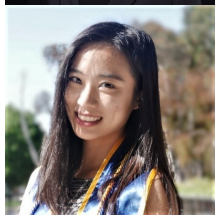
Yong Chen, PhD



Chongliang Luo, PhD



Rui Duan, PhD



Jiayi Tong, BS

ennCIL

A Computing · Inference · Learning lab at University of Pennsylvania

https://penncil.med.upenn.edu

Thank you!

Questions or ideas? Email me!
macjohn@pennmedicine.upenn.edu