Mediation of prior immunity effects on COVID-19 through neutralizing antibody titers and CD4+ and CD8+ T cells

Sydney Busch

Last updated: 9/25/25

This document gives an overview of the R code used in the natural effects analysis. Note that the "X" in the file names indicates the draft number of that file. Only the most updated file versions are published.

To run analysis:

1. Specify parameters in config_dX.R
2. Run get_all_data_and_fits_dX.R
3. Run run_jobs_dX.R
4. Run compile_jobs_dX.R
5. Run figures_report_d1.R

config_dX.R

- **Purpose:** Specify parameters for analysis.
- **Variables:**
    - stims_input
        - Specifies names of marker groups to be ran. Must correspond to groups that are listed in get_markers() fcn.
        - Note: calling this "stims" rather than marker groups so it doesn't get confused with marker families or sub-marker variables later on.
    - trt_groups_input
        - Specifies names of treatment groups to be ran, corresponding to the groups listed in clean_data_dX.R. Current options are
            - all_mRNA (arms 1-2, 4-12)
            - mRNA_moderna (arms 1-2, 5-6)
            - mRNA_pfizer (arms 7-9, 12)
            - recomb_protein (arms 13-15)
            - all_arms (arms 1-2, 3-15)
    - seed_input
        - Specifies seed to be used in effects_fcn_dX.R
    - mc_B_input
        - Specifies number of repetitions per individual for Monte Carlo integration in effects_fcn_dX.R

    - run_BS
        - Tells code whether or not you want to do bootstrapping for CI's (turn off when testing code)
    - bs_B_input
        - Specifies number of bootstrap repetitions to run
    - weights_input
        - Name of weight variable in COVAIL data for inverse probability sampling weights
    - covariates_input
        - Specifies which COVAIL variables should be adjusted for as covariates
    - exposure_input
        - Name of naïve-status variable in COVAIL data
    - outcome_time_input
        - Name of time-to-event variable in COVAIL data
    - outcome_index_input
        - Name of event index in COVAIL data
    - outcome_time_of_interest_input
        - Specifies which times to use in Cox models. Should have enough times to generate cumulative incidence curves. Also include Days 91 and 181 listed in SAP. These times are days post-D15 booster.
    - a_input
        - Exposure value. 1 = naïve / no previous infection.
    - aprime_input
        - Exposure value. 0 = non-naïve / previous infection.
- **Functions**:
    - get_markers()
        - Inputs the "stim" (marker group) for current analysis. Outputs the marker names for the analysis. If multiple markers are used in one marker family, they should be specified as mX.1, mX.2, etc.
        - Example for baseline CD4+ T cells:
            - m1.1 = Bcd4_IFNg.IL2_Wuhan.N
            - m1.2 = Bcd4_FS_Wuhan.N
            - m1.3 = Bcd4_IFNg.IL2_BA.4.5.S
            - m1.4 = Bcd4_FS_BA.4.5.S
- **Files:**
    - Stores clean_data_dX.R, fit_models_dX.R, effects_fcn_dX.R, and the COVAIL data as variables to be called in subsequent files.

get_all_data_and_fits_dX.R

- **Purpose:** Creates all datasets and fits models that are needed for running effects_fcn_dX.R.
- Calls config_dX.R to get updated variable and files names.
- Creates all combinations of stim and trt groups, then for each of those combinations, pulls the original dataset and makes bs_B number of re-sampled datasets for bootstraps.
- Fits Cox models and GLMs.
- If Cox models don't converge on the bootstrapped datasets, code will continue re-sampling until model converges.
- Saves datasets and model fits in job_output/data_and_fits
- **Functions:**
    - Calls get_data from clean_data_dX.R and fit_models from fit_model_dX.R

clean_data_dX.R

- **Purpose:** Reads in COVAIL T cell data. Subsets data to ptids that meet inclusion criteria. Creates variables for trt groups. Keeps only the variables needed to run the analysis. Renames markers from original variables names to m1.1, m1.2, etc. names.
- **Functions**
    - Create fcn get_data() to be used in get_all_data_and_fits_dX.R

fit_models_dX.R

- **Purpose:** Fits Cox models and GLMs.
- Fits Cox models for outcome Y using data subset to A = a and A = a'.
    - Uses sampling weights.
    - If Cox models don't converge, an error message is printed and fcn does not continue to GLMs.
- Fits GLMs for all markers using data subset to A = a and A = a'.
    - If the outcome sub-marker is the first marker in the family, the predictors are the covariates, exposure, and previous sub-markers from previous families (if any).
        - Examples: If outcome is m1.1, the predictors are the covariates and exposure. If outcome is m2.1, the predictors are the covariates, exposure, and m1.1, m1.2, etc.

    - If the outcome sub-marker is not the first marker in the family, the predictors are the covariates, exposure, previous sub-markers from previous families, AND previous sub-markers from that family.
        - Example: If outcome is m2.2, the predictors are the covariates, exposure, m1.1, m1.2, etc. AND m2.1.
    - Model fits are saved in a list, along with the type of model (cox or glm) and exposure (a or a') for Cox and GLMs, as well as outcome (sub-marker) and predictors for GLMs.

run_jobs_d1.R

- **Purpose**: Runs "jobs". One job is a unique combination of stim and trt groups, as well as either original or bootstrapped data.
    - Total # of jobs = (# bootstraps x # stim groups x # trt groups) + (# stim groups x # trt groups)
- Calls config_dX.R to get updated variable and files names.
- Reads in the list of data and fits used for that job from job_output/data_and_fits.
- **Function:**
    - Calls get_effects_parts() to estimate Y1 – Y8.
- Saves results from get_effects_parts() in job_output/results

effects_fcn_dX.R

- Estimates Y1 – Y8.
- **Function**: Creates get_effects_parts().
- Starts by creating a list of empty data frames to store estimated results. Each outcome time is stored in its own data frame.
- Also creates an empty list where simulated results will be stored. These correspond to each combination of counterfactual mediators listed in the Y1 – Y8 formulas in the SAP.
- Iterates over individual ptids in COVAIL data to perform Monte Carlo integration. For each individual, mediators are simulated within marker families. Previously simulated mediators are used as predictors in the simulation of subsequent mediators, according to the DAG and GLM's specified in SAP.
- Once all mc_B number of simulations for that individual are obtained, we iterate through each outcome time. Y1 – Y8 are estimated at each outcome time, using the appropriate combination of simulated data and the previously fit Cox models.
- Once Y1 – Y8 have been estimated for every individual in the COVAIL data, a weighted mean is taken using the inverse probability sampling weights, again separated by outcome time.

compile_jobs_dX.R

- **Purpose:** Compile all of the RDS files produced from run_one_job_d1.R. Compiles estimates of "original" data and bootstrapped data separately.
- Gets estimates of total and path-specific effects by adding and subtracting Y1 – Y8 according to formulas listed in SAP.
- Calculates proportion mediated (PM). Gets CI's for PM using three methods: Wald CI for raw PM using Delta method, Wald CI for log PM using Delta method, and percentile CI's for raw PM.
- Cleans and summarizes data. Final product of summarized data with CI's is called "results_summary.rds" and can be used to create figures. A file called "results_BS.rds" is also saved in case data wants to be further analyzed.

figures_report_dX.Rmd

- **Purpose:** Creates plots of cumulative incidence curves, total effect decompositions, and proportion mediated using my custom ggplot theme.