description for the optimization file

R Markdown

In the add bootstrap and add grid search file, I basically added two simple thing: 1. I added a grid search for gamma distribution using mean and sd and our zurich alpha data and it compared the rmse we get from deconvolute ww data and case data.(to do: to try with shape and scale directly) 2. I added back the bootstrap data. Now the output of get_infection_incidence_by_deconvolution would be three column, one date, one value, one replicate number. (later maybe need catchment, variant…)

I also tried with optimization using gamma and beta(from line 374 to 460), using default algorithm and SANN algorithm. and get good result. but it is not important for now LOL since we are changing it from deconvolution to convolution.

So this is the simple grid search on line 470 on the r file

```
  meanOpts = seq(0.5, 10, 0.5)
  sdOpts = seq(0.5, 5, 0.5)

  deconv_results = cbind(expand_grid(meanOpts, sdOpts),
                         'rmse_cc' = NA)

  for (row_id in 1:nrow(deconv_results)){
    #set.seed=1234   should we set seed here?
    deconv_config = try(deconvolveIncidence(zurich_alpha_ww,
                                            IncubationParams=list(shape = 0, scale = 0),
                                            getGammaParams(deconv_results[row_id, 'meanO
pts'],
                                                           deconv_results[row_id, 'sdOpt
s']), #can change for shape and scale.
                                            hypothesis="gamma", n_boot = 1 ))

    if('try-error' %in% class(deconv_config)){
      deconv_results[row_id, c('rmse_cc')] = c(Inf)
      next
    }

    deconv_results[row_id, c('rmse_cc')] = compareTracesRMSE(deconv_config, oneboot_data
#we get from decovole case data
```

)}

```
  write_csv(deconv_results, '/Users/meiyilong/Downloads/wastewaterRe-main/scan/deconv_tr
y.csv')
```

and this is the changed get_infection_incidence_by_deconvolution function adding bootstrap loop (from line 265 to line 340 on the r script)

```
### Main function to do deconvolution, get_infection_incidence_by_deconvolution.
###  Version, with boostrap
get_infection_incidence_by_deconvolution <- function(
  data_subset,# unsmoothed data with two columns, value and date
  constant_delay_distribution, # pdf for incubation + onset
  #constant_delay_distribution_incubation = c(), # pdf for incubation only
  #is_onset_data = F,# 不需要 and default
  #is_local_cases = T, # 不需要 and default
  #smooth_incidence = TRUE,# depends on the version of get_bootstrap_replicate
  days_incl,# hyperparameter
  #empirical_delays  = tibble(),# default
  n_bootstrap,# hyperparameter and 50
  days_further_in_the_past, # hyperparameter and default. this one is used in the data p
reprocessing as well as iterateRL
  #days_further_in_the_past_incubation = 5, # hyperparaeter and default. what's this fo
r: its not in our case
  threshold_chi_squared,
  is_sampling, # TODO
  max_iterations = 100) {# default verbose = FALSE

  ####### Initialization
  data_subset <- data_subset %>%   # exclude leading zeroes
    arrange(date) %>%
    filter(cumsum(value) > 0)

  #data_type_subset <- unique(data_subset$data_type)[1] # 我们是n1或者n2, 只用在命名col里
  #data_type_name <- paste0("infection_", data_type_subset) # "infection_n1" or "infecti
on_n2", 只用在命名col里

  minimal_date <- min(data_subset$date) - days_further_in_the_past
  maximal_date <- max(data_subset$date)
  all_dates <- seq(minimal_date, maximal_date, by = "days")

  ##### nested function that doesn't need to change, get_matrix_constant_waiting_time_di
str
  delay_distribution_matrix <- get_matrix_constant_waiting_time_distr(constant_delay_dis
tribution,all_dates)
  initial_delta <- min(which(cumsum(constant_delay_distribution) > 0.5)) - 1 # take medi
an value (-1 because index 1 corresponds to zero days)

  ######### Output only one deconvolution       #no bootstrap for value? might need to a
dd it.
  ###### nested functions that we dont need to change, getLOESSCases
  #added bootstrap for value
  results <- list(tibble())
  for (bootstrap_replicate_i in 0:n_bootstrap) {

    if (bootstrap_replicate_i == 0) {
      time_series <- data_subset
    } else {
      time_series <- get_bootstrap_replicate(data_subset)
    }
```

```
    smoothed_incidence_data <- time_series %>%
      complete(date = seq.Date(min(date), max(date), by = "days"), fill = list(value = 0
)) %>%
      mutate(value = getLOESSCases(dates = date, count_data = value, days_incl))

    ##### Scale the smoothed data, do we need this?
    raw_total_incidence <- sum(time_series$value, na.rm = TRUE)
    smoothed_total_incidence <- sum(smoothed_incidence_data$value, na.rm = T)

    if (smoothed_total_incidence > 0) {
      smoothed_incidence_data <- smoothed_incidence_data %>%
        mutate(value = value * raw_total_incidence / smoothed_total_incidence)
    }
    ##########################################
    # Deconvolution on smoothed data
    deconvolved_infections <-  do_deconvolution(smoothed_incidence_data,
                                                delay_distribution_matrix = delay_distri
bution_matrix,
                                                days_further_in_the_past = days_further_
in_the_past,# 30
                                                initial_delta = initial_delta,# median v
alue of gamma mixture
                                                max_iterations = max_iterations,
                                                threshold_chi_squared = threshold_chi_sq
uared)
    deconvolved_infections <- deconvolved_infections %>% slice((days_further_in_the_past
-5 + 1):n())

    ## dataframe containing results
    deconvolved_infections <- tibble(
      date = deconvolved_infections$date,
      value = deconvolved_infections$value,
      replicate = bootstrap_replicate_i
    )
    results <- c(results, list(deconvolved_infections))

 } # end of function. tibble with two columns date and value.
  return(bind_rows(results))
}
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.