Coverage generation tool – input structure and rationales

The coverage generation tool aims to generate coverage scenarios for VIMC. This will document any core coverage assumptions for future coverage projection trajectory. For historical coverage assumptions, we still need to query from data sources, mainly WHO/UNICEF and GAVI operational forecast before using the tool. This tool will also serve to generate various analysis oriented coverage scenarios.

rm(list = ls())

library("vimpact")

library("dplyr")

con = dettl:::db\_connect("production", ".")

## The tool can work on a single delivery, or a set of deliveries all together

## it works on only one country-disease combination each time

## I give three examples here

## example 1 - wuenic touchstone for gavi country (same for op touchstone) - simply binding data from two sources

## example 2 - wuenic touchstone for non-gavi country (same for op touchstone) - latest historical + future projection

## example 3 - model run touchstone or anything vimc/external users might be interested

## N.B. rules will over-write any source coverage provided,

e.g. if future is provided via src, but projection rules specified, use projection rules

e.g. if future introduction, anything in source coverage is dumped, use rules instead

## step 1: prepare source coverage

src <- vimc\_historical\_data(con, year\_cur = 2021, coverage\_src\_his = "202207wue", coverage\_src\_fut = "202303gavi")

## VIMC user extracts source coverage from montagu

## external user prepares this using provided template

## src is a list of two objects src=lis(historic, future)

## historic and future are two data frames sharing the same structure

## country, vaccine, activity\_type, year, age\_from, age\_to, gender, target, coverage, proportion\_risk

## step 2: specify input parameters and projection rules

## example 1 - mcv1 for both historical and future from source, this is like wuenic+op for gavi country

params <- list(country = "GIN",

disease = "Measles",

year\_cur = 2021,

introduction = data.frame(vaccine = c("MCV1"),

activity\_type = c("routine"),

year\_intro = c(NA))) ## year\_intro is meaningful for future intro or you want to dump historical estimates and re-project a new trajectory

proj\_rules = list(

rule1 = NULL # null means no operations needed, simply bind historical and future from src

)

## example 2 - mcv1 for historical from source + non-linear scale-up, this is like wuenic+projection for non-gavi country

params <- list(country = "GIN",

disease = "Measles",

year\_cur = 2021,

introduction = data.frame(vaccine = c("MCV1"),

activity\_type = c("routine"),

year\_intro = c(NA)))

proj\_rules = list(

rule1 = list(

non\_linear\_scale\_up = list(year\_from = 2022, year\_to = 2030, endpoint = 0.99)

) # rules are provided as a list of lists, each component is a projection rule name and its corresponding parameters for simplicity, only present one rule here

)

## next example is a complex/flexible one

## example 3 - mcv1 historical + non-linear scale-up; mcv2 manually defined future intro + various rules; sias bind past and future from source

params <- list(country = "GIN",

disease = "Measles",

year\_cur = 2021,

introduction = data.frame(vaccine = c("MCV1", "MCV2", "Measles"),

activity\_type = c("routine", "routine", "campaign"),

year\_intro = c(NA, 2024, NA)))

proj\_rules = list(

rule1 = list(non\_linear\_scale\_up = list(year\_from = 2022, year\_to = 2030, endpoint = 0.99)), # rule 1 for mcv1

rule2 = list(catch\_up\_with\_x = list(year\_from = 2024, year\_to = 2027, vaccine\_x\_level = 0.7), # rule 2 for mcv2

non\_linear\_scale\_up = list(year\_from = 2028, year\_to = 2030, endpoint = 0.9),

keep\_levels = list(year\_from = 2031, year\_to = 2035, level = 0.9),

incremental = list(year\_from = 2036, year\_to = 2040, step=0.02, cap = 0.95)),

rule3 = NULL ## rule 3 for campaign - campaign projection rules under development

)

## excute the following for each example

input <- list(params = params,

src = src,

proj\_rul = proj\_rules

)

input <- input\_check(input) # sanity check: input structure, input parameters (any conflicts in routine introduction or projection rules, etc.)

dat <- vac\_sce(input) # coverage scenario generation