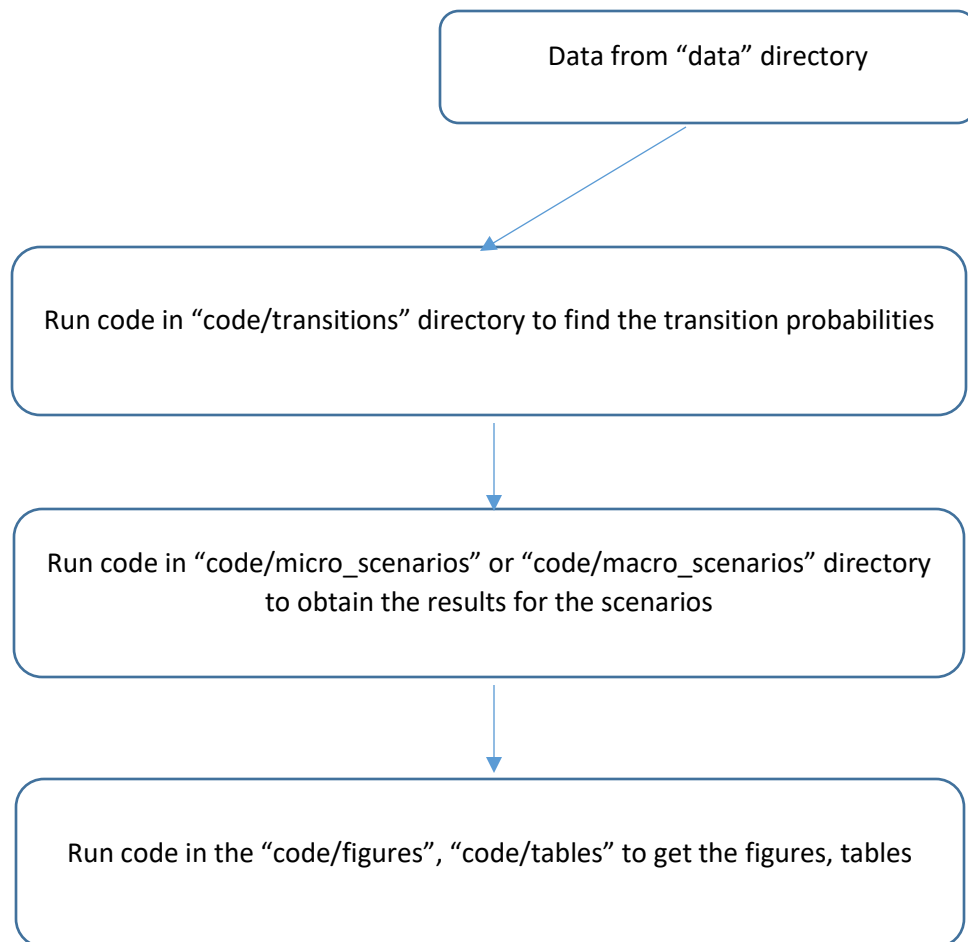


Outline



More details

Please set the working directory in R to the “ecig_code” directory (i.e. the directory containing sub-folders: code, data, references) before running the codes

Data from “data” directory

Run code in “code/transitions” directory to find the transition probabilities

For Singapore:

- I ran the code “sin_smoke_prevalence.R” in “sin_smoke_prevalence” to produce the smoking prevalence arrays which serve as input for the JAGS code in “sin_jags/Singapore_mcmc_free9/sintrans_jags.R”
- The codes in “sin_tax_coef” were ran to find the multipliers for transition probabilities under TAX scenarios (The results are presented in table S1-1 in Supplement 1 of the paper)

Run code in “code/micro_scenarios” to obtain the micro-simulation result

There are 5 directories in “code/micro_scenarios”:

- “micro_noecig”: the codes for scenarios without ecig (e.g. SQ, MLA, TAX)
- “micro_v0.2”: the codes for scenarios with ecig in the sensitivity analysis presented in Supplement 5, Figure S5-6
- “micro_v0.3”: the codes for scenarios with ecig in the main text of the paper
- “micro_v0.3_sensitivity”: the codes for other sensitivity analyses in Supplement 5
- “util”: the codes for helper functions

Run code in the “code/figures”, “code/tables” to get the figures, tables

A direction to expand the codes to find transition probabilities to account for gender and race

If you follow the approach that I used, a simplistic way to expand the codes to find transition probabilities for different genders and races is:

- Write a code which is similar to “code/transitions/sin_smoke_prevalence/sin_smoke_prevalence.R” but filter for the specific gender and race. Then use the output smoking prevalence arrays as input to a JAGS code similar to “sin_jags/Singapore_mcmc_free9/sintrans_jags.R”

If you are comfortable with hierarchical modeling, perhaps you could also explore that option.

Please feel free to come up with more reasonable/efficient approaches.

A note about the micro-simulation approach that I used:

I wrote the micro-simulation in R utilizing vectorization. The code is easy to write and able to produce results quite fast. However, it takes up quite sizable memory.

You can reduce the memory used by writing code in Rcpp.