Phylodynamics Simulation of Phylogenetic Trees - Mbindingavidity

Hsiang-Yu Yuan
Version: 1
Date: May 26, 2015

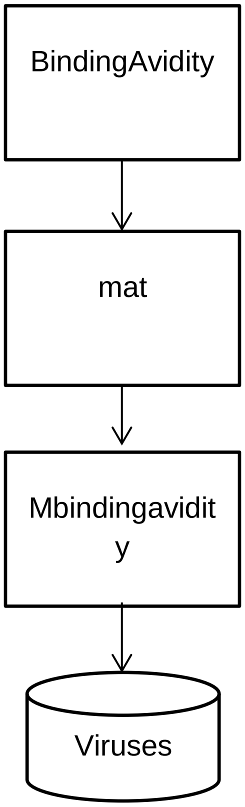
Packages for modules developing:

- BindingAvidity/mat/Mbindingavidity: Phylodynamic simulation of viral infection using Individual-based model.

- System Architecture
- System Design
- Program Flow

System Architecture (for modules developing)

This shows the file structures for the BindingAvidity package. The package simulate disease antigenic change while keep tracking individual viruses binding avidity.



```
System produces 3 objects into 'out' folder:
   1. dat_x_trans_tmp.mat: to reconstruct the transmission tree
   2. DataTLSIR.mat: to store SIR dynamics
   3. virus_traits.mat: to store virus traits
```

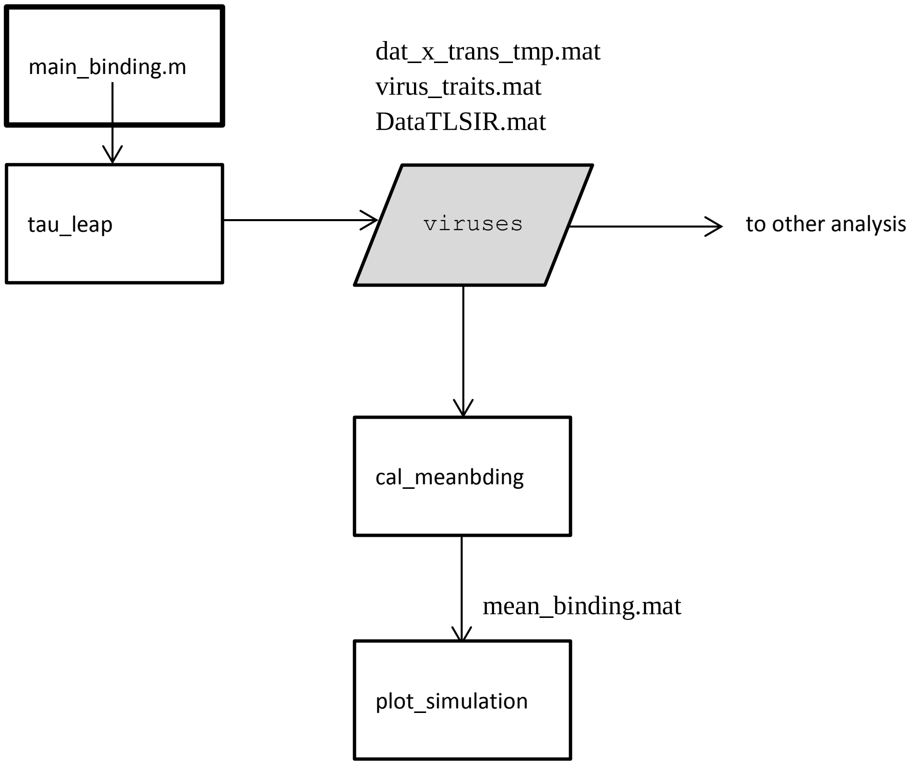
To run the code:
Go to the Mbindingavidity folder and run main_binding.m.

System Design

The folder contains the following mat files. main_binding.m is the main function which initialize the parameters and initial status then calls the tauleap function.

main_binding.m
tauleap_singlesir_ibm_matrix.m
readparams.m
get_beta.m
get_beta_list.m
cal_meanbding.m
plot_simulation_log_i.m

Mbindingavidity



Program Flow for generating virus transmission, SIR, and their traits

1. main_binding
2. Tau-leap
    a. Update virus binding avidity v and transmission rate beta
    b. Calculate number of events happened
    c. Update the epidemiological status
    d. Updating transmission for each viral strain
        i. insert new viruses into CurrentViruses
        ii. insert new viruses into VirusesArray
    e. Update the death and recovery for current viruses
3. Produce output files
    a. output_virus_transmission(VirusesArray)
    b. output_sir(DataXt);
    c. output_traits(CurrTime);
    d. Produce output files
4. calculate mean binding and plot_simulation