* Default\_VSV\_toolbox\_RK4 takes no input and returns a structure containing the default parameters that the toolbox uses to run. Note running the toolbox with the default parameters and an empty structure are equivalent, they are also built into the toolbox. But modifying some of the parameters in the structure then running the toolbox will pass those changes to the simulation.
* VSV\_get\_defect takes one input, the number of defects, which we have decided on 11. It returns two outputs. One is called “defect\_struct\_cutoff” which tells the skeleton about a transition in how the defects are stored (small detail pertaining to whether the defect is passed to the toolbox as a standalone parameter or in a structure). The other is a cell array, with one cell for each parameter. Each cell contains that parameters name and the relative bounds of each severity bin for that parameter (i.e. [0, .2, .4, .6, .8, 1]). The skeleton samples from within these ranges when creating defects, then multiplies by the default parameter value.
* VSV\_toolbox\_RK4 is the simulation itself. It is a model of a single cell VSV infection. Toolbox is because the simulation is made to be easy to modify parameters and experiment with fitness. RK4 stand for 4th order Runge-Kutta, the numerical method used to solve the differential equations. To run, the toolbox needs an ending time, a time step for the numerical method, a true/false whether or not to plot some variables (usually turned off for multiple runs), and an optional structure of parameter values (see Default\_VSV\_toolbox\_RK4). This uses Lim’s (2006) model of the VSV virus, but modifies the input for better use in epistasis simulations.
  + Entering VSV\_toolbox\_RK4(Default\_VSV\_toolbox\_RK4(),25\*3600,15,1) in the console will run the wild-type model as used in the paper
* Epistasis\_Skeleton is the workhorse of the group. It gathers the default parameters and defect ranges, then randomly samples a set of defects for a given severity and runs the toolbox. It then saves the results as a “.mat” file. “process\_num” and “cluster\_num” were used in the distributed computing to uniquely seed the random number generator and save the results, any integers can be used for these values. The usefulness lies in how it is set up to run many iterations for multiple numbers of defects. The epistasis tests usually originate with calling the skeleton several times for different severities and the skeleton interfaces with everything from there.
  + Entering Epistasis\_Skeleton(5,2,3,100,200,1) in the console will run a small epistasis simulation of 5 repetitions with up to 2 defects (5 repetitions with 1 defect, then 5 repetitions with 2 defects). It will use the 3rd of 5 levels of increasing severity, and the random number generator will be seeded with 100200 (which will also be in the name of the output data file). The last argument being 1 or “True” will use the light version of the model as detailed in the paper.