This Agent Based Model was created as a part of a research project. The purpose of the project was to study the evolution of the Killer Cell Immunoglobulin like (KIR) receptor system which is mainly expressed in Natural Killer Cells. More specifically, to investigate whether viruses can drive the evolution and diversity of KIRs, taking into account that KIRs bind peptide presenting MHCs during the immune response. The model was written in a linux environment (relies on the linux C++ compiler) and uses Boost libraries for randomization functions.

Model Description

The model consists of two actors, hosts and viruses and three events, birth, death and infection. The basic time step of the model is one week. After the model is initiated, at every time step each host randomly faces one of the three events:

Birth events happen through sexual reproduction with a probability that depends both on the population size as well as the age of the offspring's parents. Parents transfer their genome to the offspring and there is a mutation probability for the KIRs in each birth.

Death events happen with a probability which is the sum of the intrinsic (age dependent) death rate and, if the hosts are infected, a viral load dependent death rate, which is higher during the acute phase.

Each time step, a host contacts a number of other hosts, some of which may be infected. For each contact with an infected host there is a probability that the infection will be transferred to the original host. This probability depends on which phase the infected (donor) host is: during the acute phase this probability is higher than during the chronic phase. If the virus is transferred to the (receiving) host, the host himself enters the acute phase of the infection, where death rate is higher. Furthermore, the host’s MHCs now present viral peptides instead of host peptides.

Clearance of the infection may happen after a (set) period of time if KIRs detect the change on the MHC presented peptides i.e., if all of the KIRs do not bind the new peptide MHC complexes. If at least one of the KIRs binds one or more of the new complexes (they are considered to be tricked by the virus), the virus is not cleared the host becomes chronically infected.

A number of model parameters, such as population size, rates of infection or the duration of the simulation, can be controlled by the “Parameter.data” file, while a few others are hardcoded.

The “World.cpp” and “World.h” handles the events, the main loop as well as the creation of output files.

These files are: a population file (includes birth, death, infected hosts numbers etc), a sequence file which associates peptide sequences with integers and an association file which shows how many KIRs can each MHC binds. Also two files are created periodically (period set in the parameters): a file containing the genomes of the hosts and another containing general information about the hosts.

Hosts and Viruses

Hosts are simplified diploid organisms set to emulate a human population which have a minimal genome of 1 MHC locus and 5 KIR loci and by being diploid have a max of 2 MHCs and 10 KIRs per agent. Hosts also have set of peptides. Viruses do not have a genome in this simulation, but have their own smaller set of viral peptides which are different from the host’s peptides. Both MHCs and KIRs are represented as strings of 16 characters corresponding to the one letter standard amino acid abbreviations. A peptide is represented as a string of 8 characters.

MHCs have the ability to present peptides. If a peptide is presented by an MHC molecule, 8 consecutive characters in the middle part of the MHC string (from position 5 to position 12) are replaced by the sequence of characters of the peptide string (peptide character at position 1 replaces MHC character at position 5 and so on). The new string represents the peptide - MHC complex (pMHC). A peptide binds the MHC with a binding score which is determined by the amino acid energy score in the Miyazawa and Jernigan matrix (which is provided, the model also supports

other interaction matrixes). The binding score is calculated by summing the pairwise energy scores on the 8 positions of the peptide. The peptide is considered to be presented, when the score exceeds a threshold. The same procedure is applied to determine whether a pMHC will bind to a KIR. In this case the strings are aligned from position 1 to 16 and the 16 pairwise energy score add up to a binding score. This binding score is also compared to a different threshold, which can be variable, representing higher or lower specificity for the KIRs.
Hosts and viruses are handled by “Host.cpp” and “Host.h”, their genome by “Genes.cpp” and “Genes.h” and the string interactions are handled by “Sequence.cpp” and “Sequence.h”.
Furthermore, a number of other smaller supporting files like “MathFunctions.h”, handle randomization, backup, input and output etc.

Model Initialization

At the start of the simulation, a pool of peptides is created, then split into a viral and a host pool. No MHC evolution is allowed in the model, instead an initial group of 14 MHC alleles is constructed, i.e. 14 random character strings of 16 amino acids, which remains unchanged throughout the simulation. A pool of 140 KIRs is also created, making sure that there are at least 10 “licensed” KIRs for each MHC molecule, that is each one of the MHCs must bind at least 10 out of the 140 KIRs. Finally, an initial population of hosts is created, and is left to drift for sometime before the virus is introduced. After a long period of time (at least 200000 years) the model reaches its optimum and the population enters a phase which remains stable.

This simulation creates a large amount of data, of which the population and genome files offer the most valuable information on the evolution of KIRs under viral influence. The population dynamics and changes on the genomes of the hosts over the course of the simulation can be analyzed and information can be extracted about the diversity of the genomes (MHCs, KIRs) as well as the evolved KIRs at the end of the simulation and their characteristics. The “Analysis” folder contains a number of scripts and pipelines essential for the data analysis.