Multi-agent model of hepatitis C virus infection

Szymon Wasik[a,*], Paulina Jackowiak[b], Marek Figlerowicz[a,b], Jacek Blazewicz[a,b]

[a] *Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland*

[b] *Institute of Bioorganic Chemistry, Polish Academy of Sciences, Z. Noskowskiego 12/14, 61-704 Poznan, Poland*

Abstract

Objectives: The objective of this study is to design a method for modeling hepatitis C virus (HCV) infection using multi-agent simulation and to verify it in practice.

Methods and materials: In this paper, first, the modeling of HCV infection using a multi-agent system is compared with the most commonly used model type, which is based on differential equations. Then, the implementation and results of the model using a multi-agent simulation is presented. To find the values of the parameters used in the model, a method using inverted simulation flow and genetic algorithm is proposed. All of the data regarding HCV infection are taken from the paper describing the model based on the differential equation to which the proposed method is compared.

Results: Important advantages of the proposed method are noted and demonstrated; these include flexibility, clarity, re-usability and the possibility to model more complex dependencies. Then, the simulation framework that uses the proposed approach is successfully implemented in C++ and is verified by comparing it to the approach based on differential equations. The verification proves that an objective function that performs the best is the function that minimizes the maximal differences in the data. Finally, an analysis of one of the already known models is performed, and it is proved that it incorrectly models a decay in the hepatocytes number by 40%.

Conclusions: The proposed method has many advantages in comparison to the currently used model types and can be used successfully for analyzing

[*] Corresponding author. Tel: 4861 665 3032, Fax: 4861 877 1525

Email address: `szymon.wasik@cs.put.poznan.pl` (Szymon Wasik)

HCV infection. With almost no modifications, it can also be used for other types of viral infections.

Keywords: multi-agent systems, multi-agent simulation, differential equations, genetic algorithm, HCV infection

1. Introduction

Hepatitis C virus (HCV) infections are among the major global health problems and concern 3% of the human population [1]. Most of these infections become chronic and lead to liver failure, including fibrosis, cirrhosis and hepatocellular carcinoma [2]. The prevention and treatment of HCV infections is a significant challenge because the elaboration of efficient vaccines and antiviral drugs is hampered by a very high genetic variability of the virus. There are six major genotypes and more than 80 subtypes of HCV around the world. In a single infected individual, HCV exists as a quasispecies a population of genetically related but slightly different variants [3, 4]. Some of the quasispecies variants appear to be superior in terms of viral fitness. They dominate the populations and can survive despite the pressure of the immune system and therapeutic agents [5–7]. Current standard therapy, which involves pegylated interferon and ribavirin, is effective in only approximately 40% of the patients infected with the highly prevalent genotype 1 [8]. The efficiency of the therapy increases considerably when the interferon and ribavirin are accompanied by one of the recently approved direct-acting antiviral agents, boceprevir or telaprevir [9, 10]. Despite this substantial progress, there remain certain concerns that are connected with anti-HCV treatment. First, the novel virus-specific compounds have only been approved for use against the HCV genotype 1. Second, all of the treatment options available thus far are connected with severe side effects, which can lead to discontinuation of therapy or dose reduction and, eventually, a decrease in the overall treatment efficacy. Consequently, a better understanding of the molecular mechanisms that underlie HCV pathogenesis is essential to further improve the therapeutic regimens, to achieve a higher rate of virus eradication and eliminate adverse effects [11].

Mathematical models of HCV infection have been valuable tools in addressing biologically important questions that are concerned with crucial features of viral dynamics and the mode of action of the therapeutic agents (reviewed in [12]). The initial model was applied to explain a biphasic decline of the serum HCV RNA level upon the interferon monotherapy. This approach allowed the rate of virion production and virion half-life to be established as well as elucidating the effect exerted by interferon [13]. Modifications and extensions of the original model enabled the description of other patterns of HCV kinetics, including a triphasic decline in the serum RNA level, null response and viral breakthrough. Along with the progress in the understanding of the HCV-host interaction, the models have incorporated further parameters, such as: (i) effects of ribavirin and novel virus-specific drugs, (ii) pharmacokinetic and pharmacodynamic properties of antivirals, (iii) emergence of drug resistant variants, (iv) level of alanine aminotransferase (ALT) activity and (v) patient genetic polymorphism, which also affects the response to therapy (reviewed in [12, 14]). Nevertheless, as more new aspects of HCV infection are being unraveled, there remains a demand for more expanded models that can incorporate many parameters and integrate complex clinical and experimental data.

The most commonly used mathematical models that simulate viral infections are models that are based on differential equations. During the last 20 years, since the first differential model of the human immunodeficiency virus (HIV) kinetics was defined [15], many modifications and enhancements have been introduced. First, these models were used to describe infections of HIV, which was the most intensively studied virus in the 1990s, which is why many models that attempted to explain its behavior were designed [16]. During the last 10 years, similar models of other viral infections were created. Because HCV infection is still inaccurately understood, HCV is currently one of the most popular modeling targets [12] next to HIV. However, there are also examples of other viruses, for example, the hepatitis B virus (HBV) [17].

Although differential equations are a very well known and versatile methodology of modeling, they have some disadvantages. For example, they do not allow the modeling of space, they add custom attributes, and their analysis involves advanced mathematical theory. For this reason, some other modeling techniques have recently been designed [18, 19]. One of them is an approach that is based on multi-agent simulations (MAS) [20]. There are already some researchers who attempted to utilize MAS in biology. An extended review of their results can be found in [21] and [22]. These results include applications of agents and multi-agent systems in modeling cells [23], signaling pathways [24–26] or studying tumor growth [27, 28], morphogenesis [29], chemotaxis [30], immune responses [31, 32], granuloma formation [33] and other phenomena [34–36]. There were even some attempts to define or create an agent-based framework for systems biology [37, 38]. In [39], the authors presented how agents can be used to facilitate the creation of multi-scale biological models. There are also many papers on the application of agents in health care (for example, [40–43]). However, they describe the application of agents in medicine on a higher level of medical services and, hence, the majority do not provide methods that can be used to model viral infections of a single body.

Although the topic of multi-agent systems is popular in biological applications, there is a very limited number of their applications in the field of viral infections modeling. There are few authors who have attempted to utilize agents to model mainly HIV infection [44–46], but only one group [46] assumes that multi-agent simulations can be used for HCV. Moreover, all of the existing papers are rather concentrated on basic models of viral infection. The proposed approaches lack clearly defined goals and algorithms for setting values of unknown parameters in a process that is similar to data fitting. Only [44] presents the algorithm for a verification of some commonly assumed theories about HIV infection in the general case.

Although the application of agents to model viral infections has not been analyzed by many researchers yet, the authors believe that it is an approach that can be very successful. The viruses that are present in the human body are independent (decentralized), movable, can interact with cells that are located in their neighborhood (local view) and cooperate to successfully infect the organism (self-organization). The same features characterize agents in the multi-agent system [20]. Additionally, the varying human body can be easily modeled by the environment in which the agents interact. In this paper, the comprehensive algorithm for the construction and verification of a multi-agent model of HCV infection is presented. To achieve this aim, first an example is model designed by analogy with an already existing model [47], which is described in section 2. Section 3 presents a method to perform some more complex analyses using multi-agent simulation and the algorithm for finding the values of the parameters that exist in the model. These algorithms can be used to tune and verify the model for a specific patient. In section 4 the results of the computational experiment are presented, and finally, in section 5, there are some conclusions.

2. Differential equation model of HCV infection with treatment

The models of HCV infection that use differential equations have already been defined and discussed by many authors (for example, [12, 48–50]). The model described here was compared with the model defined in [51, 52]. The latter was selected because it is the standard model, which can reproduce biphasic decline kinetics that are the most common for HCV infection.

2.1. Definition

As was stated in the introduction, the model of HCV infection that uses differential equations has already been defined and discussed by many authors. The model that describes both HCV infection and the response to treatment is presented in the following equations.

$$\frac{dT}{dt} = s + r_T T\left(1 - \frac{T+I}{T_{max}}\right) - d_T T - (1-\eta)\beta V T + qI \tag{1}$$

$$\frac{dI}{dt} = r_I I\left(1 - \frac{T+I}{T_{max}}\right) + (1-\eta)\beta V T - d_I I - qI \tag{2}$$

$$\frac{dV}{dt} = (1-\epsilon)pI - cV \tag{3}$$

In the above equations, T denotes the number of uninfected hepatocytes, I denotes the number of infected hepatocytes, and V is the number of free virions. Uninfected hepatocytes are produced by the differentiation of precursors at the rate s and are infected at the rate β, which is proportional to the number of uninfected cells and free virions. Both uninfected and infected hepatocytes die at the rates d_T and d_I, respectively, and they proliferate at the maximum rates of r_T and r_I, respectively, until the maximal number of hepatocytes T_{max} is reached. Infected cells can also be cured through a noncytolytic process at the rate q [47]. Free functional or non-functional virions are produced from infected hepatocytes at the rate p and are cleared by the immune system at the rate c. The coefficients ϵ and η model the treatment with antiviral drugs (interferon and ribavirin). When no treatment is set, they equal 0. Time is measured in days, and all of the quantities are measured in one milliliter of tissue.

2.2. Example analysis of the model

The model presented in section 2.1 can be analyzed in many ways using well-known mathematical instruments. For example, one can find steady states, perform a stability analysis or calculate the condition that must be satisfied to successfully cure the patient. The detailed mathematical analysis of the model was presented in [49]. In this section, the example results for one of the patients are presented, which will be referred to later. Additionally, the methodology that was used to generate these results is briefly described to make it possible to compare it with the approach proposed in this article.

In figure 1, the example result of the simulation based on this model for one of the patients is presented. For others, it can be different because the rates of the processes modeled by the differential equations are different in the body of each patient (organisms are not identical and behave in different ways). This finding means that the parameters described using small letters in equations 1-3 will probably have different values (but in some usually known ranges) for each patient and that they have to be found at the beginning of the model analysis. This goal is usually accomplished using mathematical software for the data fitting of the differential equations (see PottersWheel [53] or Berkeley Madonna [54]). Figure 1 presents the simulation for the values found for the example patient described in [49]. The number of patients was limited to one because the main objective of this article was to prove the computational usefulness of the proposed method.

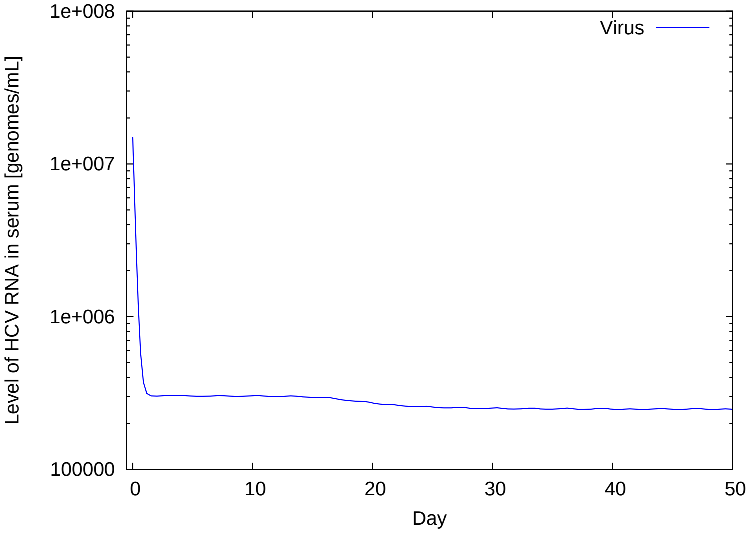


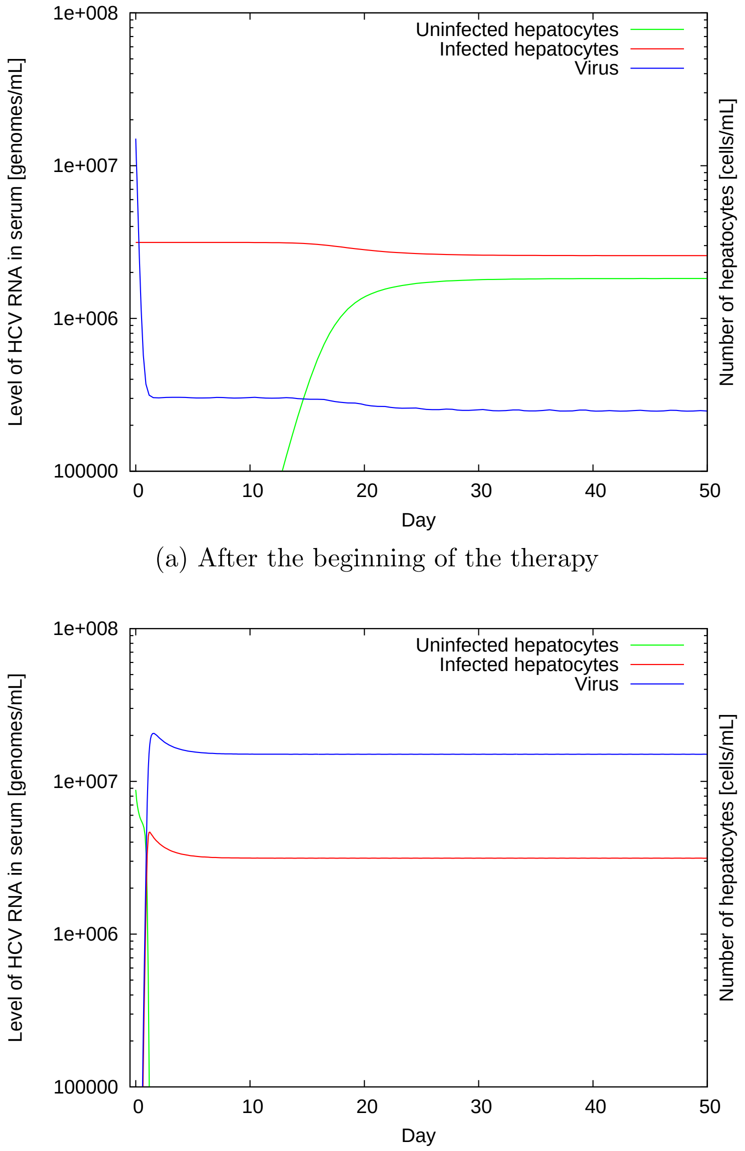
Figure 1: Simulation of the viral kinetics after the beginning of the therapy, as presented in [49]. The number of virions decreases rapidly after the beginning of the therapy. However, after two days, it stabilizes at a positive level, which means that the therapy was not successful. The parameter values are $\beta = 1.4 \cdot 10^{-6}$, $d_T = 0.012$, $d_I = 0.36$, $r_T = 3.0$, $r_I = 0.97$, $T_{max} = 5 \cdot 10^6$, $p = 28.7$, $c = 6.0$, $q = 0$, $\epsilon = 0.98$, and $\eta = 0$.

2.3. Analysis of the differential equations model

Usually, during the analysis of the viral infection based on a model that uses differential equations, the available clinical data are limited to the level of HCV RNA in serum. The acquisition of more detailed data, such as the number of uninfected and infected hepatocytes, requires the invasive procedure of liver biopsy, followed by sophisticated analyses. This requirement probably explains why authors of mathematical models of HCV kinetics usually pay attention only to the equation that describes the level of the viral RNA in serum and use this equation to fit their model results to data. This approach can give rise to misleading conclusions because one cannot state that a model's solution correctly fits to the data unless equations that model the number of hepatocytes have at least reasonable results. In most of the models, it is assumed that the number of hepatocytes is constant during the first days of therapy, and this fact is not verified in the simulation results. However, the solution of the equation that models the number of virions can fit perfectly, but unless the solution of equations that model the number of hepatocytes fits the data, the model does not necessarily explain the data correctly.

As an example, the model described in [49] can be analyzed. The result of the simulation of the model that was proposed in that paper is presented in figure 1. Figure 2a presents the same results but with the number of uninfected and infected hepatocytes included. It can be noted that the total number of hepatocytes after approximately 30 days decreases to approximately $4.4 \cdot 10^6$ cells/mL. This number is approximately 12% less than the number of hepatocytes in the liver that is not infected, as defined by the number of hepatocytes in the stable, uninfected state, which is equal to $4.98 \cdot 10^6$ cells/mL. This reduction in the number of hepatocytes is highly improbable.

Moreover, a greater problem arises during the analysis of what occurs before the beginning of the treatment. The model assumes that the parameters ϵ and η model the treatment correctly and that other parameters model the remaining significant aspects of the infection in the patient's body. According to this assumption, after setting ϵ and η to 0, the equations should model how the number of infected hepatocytes changes before the beginning of the treatment. This case is presented in figure 2b, and it shows that the number of hepatocytes decreases to $3.14 \cdot 10^6$ cells/mL after only 10 days of infection. That finding means that the volume of the liver decreases by almost 40% in 10 days after having been infected, which is evidently not true. In addition, after three days, the number of uninfected cells decreases to approximately $3 \cdot 10^3$ cells/mL. This finding means that almost all of the cells become very quickly infected, which does not commonly occur [55].



(a) After the beginning of the therapy

(b) Before the therapy

Figure 2: Simulation of the viral kinetics and the number of infected and uninfected hepatocytes after the beginning of the therapy (a) and before the beginning of the therapy (b). $(\eta, \epsilon = 0)$

The analysis that is described in this section shows that even using very well known and commonly applied tools and algorithms, some important analyses can be omitted. In this article, another more versatile method of modeling is presented and used to analyze problems in the above-described model more deeply.

3. HCV infection model based on multi-agent simulations

As an alternative to the model based on differential equations, the model using multi-agent simulation (MAS) will be presented in this section. Such a model uses the concept of agents cooperating in some environment and usually gives many more analytical possibilities. The definition of the model is presented in section 3.1; then, there is a brief presentation of the interactions between the agents in section 3.1, and finally, the solution to the problem of finding the parameter values is discussed in section 3.2. The aim of this section is to give the reader the overview that is required to understand the described work. More details on the implementation that allows us to reproduce the results are presented later in section 4. A detailed description of the advantages of using a multi-agent model is presented in section 5.

3.1. Model architecture

The multi-agent model uses the concept of agents interacting among themselves and the environment in which they are placed. In biological simulations that involve many cells, it is often required to simulate interactions between millions of agents that represent cells. Such a complex system is sometimes called a massively multi-agent system (MMAS) [56, 57]. In the case of the HCV model, there are three types of agents: infected hepatocytes, uninfected hepatocytes and free virions interacting in the environment of the human body. However, each of these types can occur in the simulation in hundreds of thousands of copies.

The diagram presented in figure 3 shows the interactions between the agents and the environment that are simulated iteratively while presenting the flow of time. Each circle represents all of the copies of agents of a specific type; hence, the arrows represent the interaction between the selected representative agents of that type and some other agent. Hepatocytes can proliferate (reproduce), and they die after some time. The death is modeled as being killed by the environment, and in the process of proliferation, only the single agent takes part. These processes are almost identical for infected and uninfected cells. Only the rate of the processes is different. Uninfected cells can be infected by virions and then emit progeny virions that can be cleared by the immune system (again modeled by being killed by the environment).

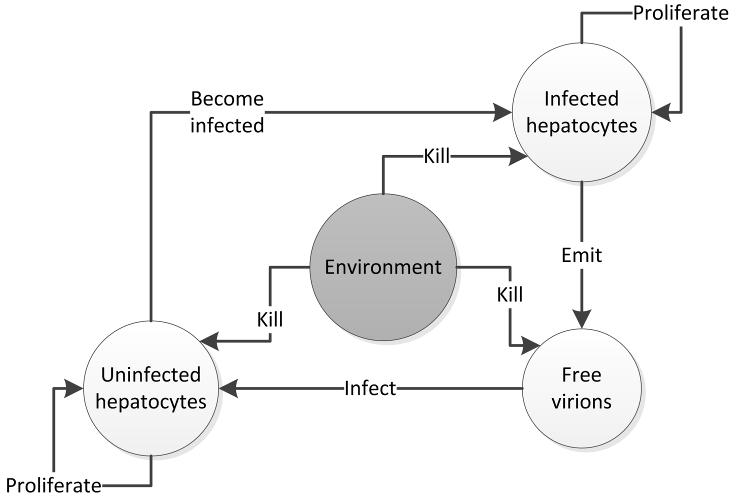


Figure 3: Interactions between agents and the environment.

Each interaction is described by one or more parameters, for example the proliferation or clearance rate. All of the parameters except for s and q are equivalents of coefficients that are used in the differential equation presented in section 2.1, and all of them are presented in table 1. In the multi-agent model, the parameters s and q are omitted because they describe processes in the human body that are present in very limited quantities and do not influence the process of infection [49].

Parameter	Description
β	infection rate
d_T	death rate of uninfected hepatocytes
d_I	death rate of infected hepatocytes
r_T	proliferation rate of uninfected hepatocytes
r_I	proliferation rate of infected hepatocytes
T_{max}	maximal capacity of liver (number of all hepatocytes)
p	production rate of free virions
c	clearance rate of virions by the immune system
ϵ	decrease in the production rate p when using therapy
η	decrease in the infection rate β when using therapy

Table 1: Parameters used in the model

3.2. Finding the parameter values

The largest difficulty during the analysis of the model was finding the values of the parameters presented in table 1 for a specific patient. In the case of differential equations, broadly applied algorithms can be used for data fitting. Unfortunately, multi-agent simulation requires the use of nonstandard, more complex algorithms that cannot be easily described using mathematical equations. For this reason, the values of the parameters cannot be determined before the simulation. To find them, the inverted simulation method can be used [58]. The general idea of this method is presented in figure 4.

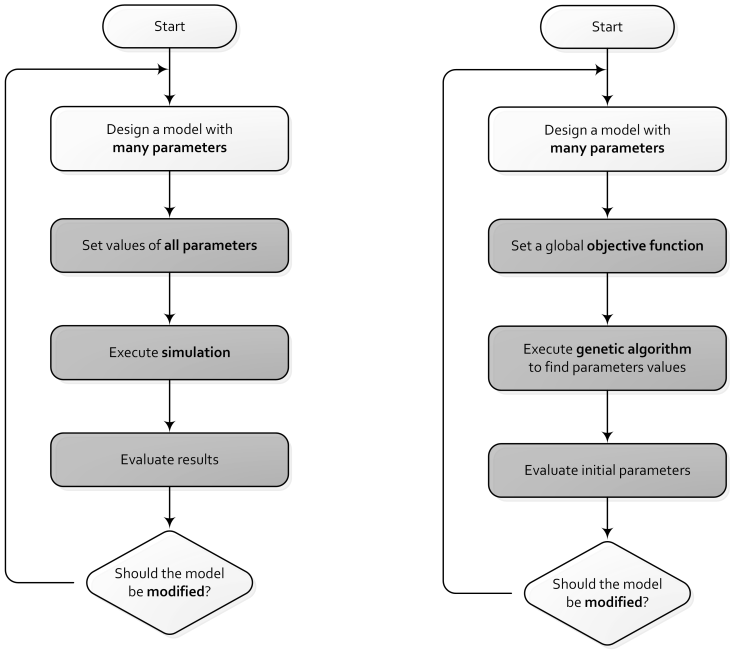

Start
Design a model with many parameters
Set values of all parameters
Execute simulation
Evaluate results
Should the model be modified?
Start
Design a model with many parameters
Set a global objective function
Execute genetic algorithm to find parameters values
Evaluate initial parameters
Should the model be modified?

Figure 4: Methods for simulating the models. The left diagram presents the commonly used simulation flow. The right diagram presents the modified, inverted simulation method.

In the case of the simulation of a complex model that is characterized by many parameters using a regular simulation method, the values of the parameters are always known from the beginning of the simulation or can be easily determined, for example, using data fitting. When the simulation completes, the results and the execution of the simulation are evaluated. If they do not reflect the reality precisely enough, then the model can be modified.

In the inverted simulation method, the objective function is defined (for example, a best fit of the results to the data), and then, a genetic algorithm or other metaheuristic is used to optimize this function and obtain the values of the parameters. Finally, the found values of the parameters and the execution of the simulation for this set of parameters are evaluated. If required, the model is further modified. This method also allows us to perform the analysis of the solution sensitivity. It is possible that by running simulations multiple times, we receive different values of the parameters. By checking their distribution, we can decide which parameters are important when designing the antiviral therapy. The significant parameters are those that have a low standard deviation.

4. Computational experiment

4.1. Computer implementation of the GA

The inspiration for the implementation of the genetic algorithm was provided by the research on real parameter optimization using the GA [59]. The suggestions proposed in this paper were implemented, and the variant of the algorithm for which the performance was the best was chosen. According to this paper, the real coding of the parameters was used, which means that each parameter was encoded as a real number scaled to the $(0, 1)$ interval in such a way that the chromosome was represented as a sequence of real numbers. The algorithm used fitness proportionate (roulette-wheel) selection and one-point crossover that divides each chromosome at the same point between two numbers that represent successive parameters and exchanges them. The mutation rate was 0.1, which means that it changed the value v of the real parameter by choosing randomly the value from the range $(max(0, v-0.1), v)$ or $(v, min(1, v + 0.1))$, depending on the randomly chosen mutation direction. The mutation probability was 0.05, which also, according to [59], was the best choice in most cases.

One of the most important steps during the design and implementation of the algorithm was to define and verify the optimized objective function. The goal was to fit the results of the simulation to clinical data. Let X denote the values that were generated during the simulation, and $\hat{X}$ denotes values obtained from clinical experiments. The following three objective functions that were minimized were considered:

$$max_{x_i \in X, \hat{x_i} \in \hat{X}} |x_i - \hat{x_i}| \tag{4}$$

$$max_{x_i \in X, \hat{x_i} \in \hat{X}} \frac{|x_i - \hat{x_i}|}{\hat{x_i}} \tag{5}$$

$$\sum_{x_i \in X, \hat{x_i} \in \hat{X}} |x_i - \hat{x_i}| \tag{6}$$

The results obtained using function (4) were the best. All of the results that are presented in the further sections were generated using this function.

When using function (5), it happens that the number of virions was very small in the clinical data and the number of virions calculated by the simulation was several times larger. In such a scenario, the ratio defined by function (5) was large; however, the difference was biologically irrelevant. Function (6) does not distinguish cases in which a single measurement has a very large difference from the case where all of the measurements have average differences, the latter case being more desirable.

4.2. Computer implementation of the MAS

4.2.1. Algorithm

The environment in which agents interact (the liver) was modeled using a rectangular grid. Although there is some precedent to use a hexagonal grid [60], it was not chosen because it has worse performance and tests showed that, in this case, the characteristics of the rectangular grid are sufficient. Each agent in this environment has its own position, which is set at the initialization phase randomly (using a uniform distribution). Each field of the grid can be occupied by many cells. The neighborhood that was used was composed of the eight nearest neighbors. Then, the simulation begins and is composed of five phases:

1. **Moving agents.** In this phase, each movable agent is moved. In this case, only free virions can move by flowing with the blood through the liver. The direction of the movement is chosen randomly; however, the movement in the direction that is consistent with the direction of the blood flow is much more probable.
2. **Interactions between agents and the environment.** In this phase, interactions between agents and the environment are simulated. Interactions executed in this phase model remove cells from the liver, create new hepatocytes by proliferation and emit new free virions into the environment from the infected cells according to the definitions from section 3.1. The position of the new cell is chosen randomly in the close neighborhood of the source cell.
3. **Updating the environment.** After modifying the numbers and positions of the cells in the previous phases, some update of the environment model is needed. This update does not model any biological process but instead rebuilds some data structures that are required to make the subsequent phases run faster.

4. **Interactions between agents.** In this phase, interactions between agents are simulated. In the described model, the only interaction is the hepatocytes' infection by free virions. The cell can be infected by cells in its neighborhood with some probability.
5. **Updating statistics.** Finally, certain statistics must be updated and saved to enable analysis of the simulation after its completion.

4.2.2. Complexity

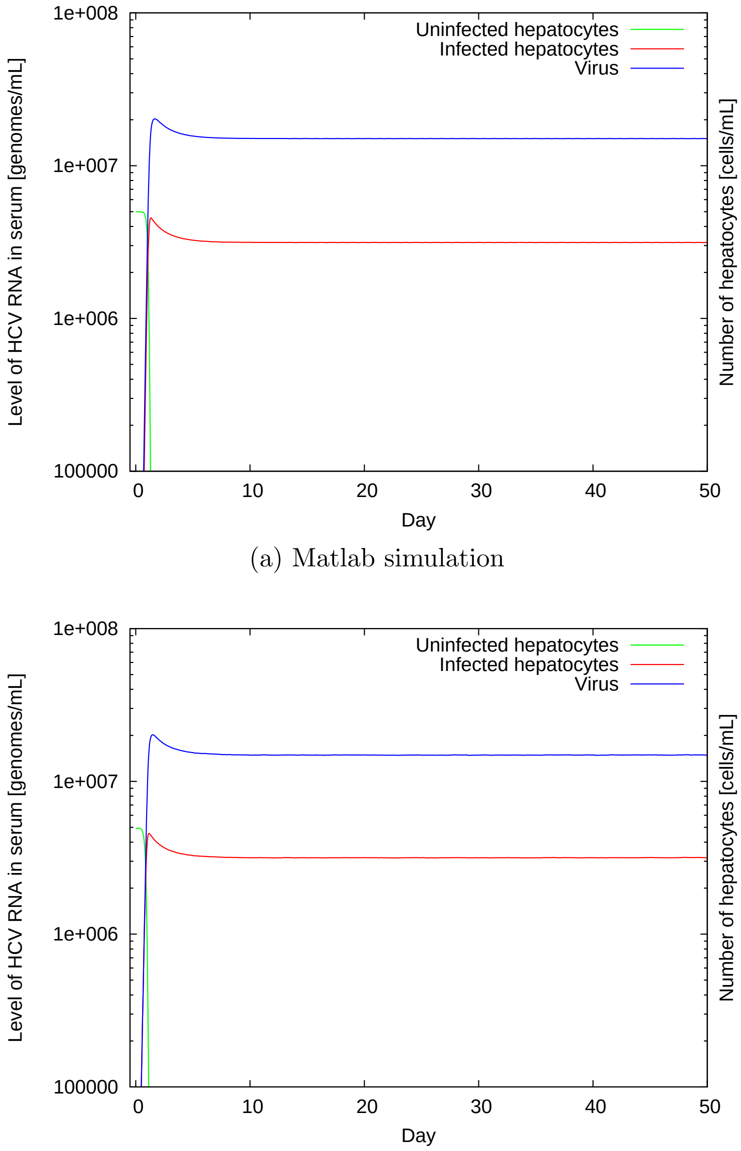
The computational complexity of the single simulation is $O(d \cdot (n + m))$, where d denotes the number of iterations that are simulated, n denotes the maximal number of all agents in some iteration, and m denotes the number of cells in the grid that models the environment. Considering that both n and m can be equal to several hundreds of thousands and d to several thousands, a single simulation can be executed even for approximately 5 min. Considering that the evaluation of a single population of chromosomes in the genetic algorithm requires execution of 20 or 30 simulations and that finding parameter values requires the evaluation of several hundreds of populations, it can in total require many hours or even days of the computations to complete a single experiment. To shorten the time of computations, the following optimizations were used:

1. The simulation was implemented very efficiently in the C++ language, which is much faster than, for example, Java or C#. All of the critical and frequently executed fragments of code were carefully optimized.
2. The OpenMP library was used to execute many simulations in parallel to make use of all four cores of the Intel i7 processor that were used to conduct the experiments. The OpenMP 3.0 implementation that is part of the g++ 4.5 compiler was used.

The above optimization decreased the execution time of a single experiment to below 6 h, which is an acceptable amount of time.

4.2.3. Verification

The correctness of the implementation was verified by comparing it with the Matlab simulation of the model based on differential equations. The most complicated step of the algorithm was to find values of the parameters. For this reason the goal of the verification was primarily to verify this procedure. To achieve this goal, the data for all of the patients presented in [49] were taken, and the objective of the algorithm was to find the values of the parameters that characterize these specific infections. As a result, the correct values of the parameters were found, and the multi-agent simulation was as successful in simulating the course of infection as the original approach based on the differential equations. The results of the verification for one of the patients are presented in figure 5. It can be noted that both charts are very similar.



(a) Matlab simulation

(b) Multi-agent simulation

Figure 5: Exemplary comparison of Matlab and multi-agent simulation.

4.3. Verification of the differential equation model

To verify if the initial model that was criticized in section 2.3 is inaccurate or if it only has incorrect parameter values, an analysis was performed by the proposed multi-agent simulation. The objective of the simulation was to check whether it is possible to find values of parameters that will generate biologically reasonable results. Two inconsistencies that had been noted in section 2.3 were verified:

1. Total number of hepatocytes being too low is verified in section 4.3.1.
2. Total number of uninfected hepatocytes being too low is verified in section 4.3.2.

For each inconsistency, two cases were verified:

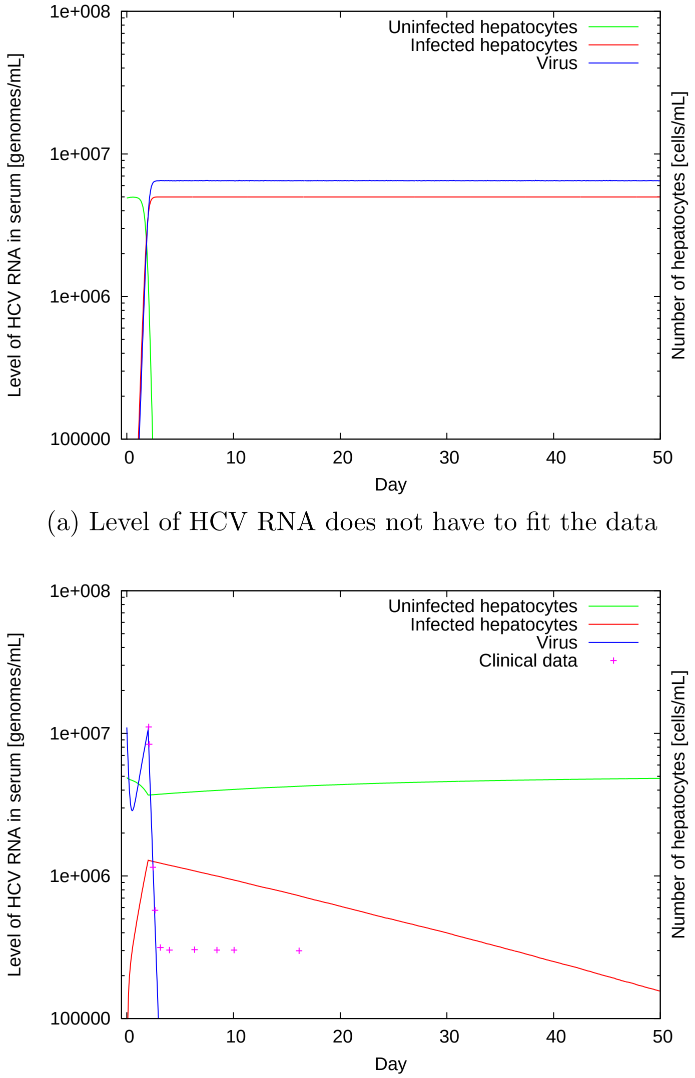
1. Basic verification when the simulated number of free virions is not fitted to clinical data that describes the level of the virus. From the biological point of view, such a verification is pointless. However, from the computational point of view, it is a good initial selection. If this verification cannot succeed, then we can be sure that a more complex verification in which the level of the virions is considered will also fail.
2. A more difficult, but realistic, experiment, apart from satisfying the condition verified in the previous point, is that the number of virions is required to fit the clinical data.

Both of these cases were easy to simulate when using the genetic algorithm. The only work that had to be performed was a slight modification of the fitness function.

4.3.1. Maximizing the total number of hepatocytes

The goal of this experiment was to check whether it is possible to obtain a total number of hepatocytes that is close to the number of cells in the human liver. To verify this possibility, the original clinical data set containing information about the number of virions was extended with information about the expected number of hepatocytes in the liver. The number of data points in which the level of the hepatocytes was verified was equal to the number of data points that describe the virions, to give them equal significance, and they were uniformly distributed during the period of treatment.

The results presented in figure 7 show that it is possible to model the correct level of hepatocytes only if the number of virions does not have to fit the clinical data. However, it is impossible otherwise, in the realistic case.



(a) Level of HCV RNA does not have to fit the data

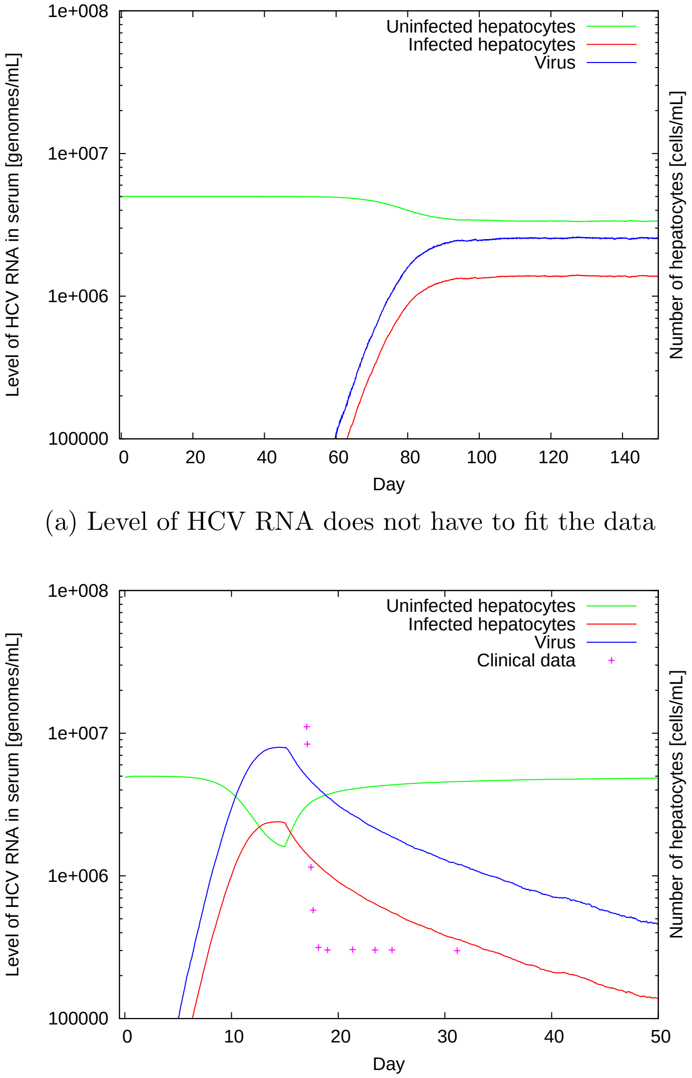
(b) Level of HCV RNA should fit the data but it does not

Figure 6: Results of the simulation that maximizes the number of hepatocytes. It is possible to receive the high number of hepatocytes only if the number of viruses does not have to fit the clinical data.

4.3.2. Maximizing the number of uninfected cells

During the analysis of the model described in section 2.3, it was noticed that the number of uninfected hepatocytes quickly falls to almost zero, which contradicts the results from the clinical research. The goal of this experiment was to check if it is possible to model a reasonable number of uninfected cells. The verification was similar to the procedure described in section 4.3.1. However, according to [55], different expected levels of uninfected hepatocytes were defined. In successive experiments, the level of uninfected hepatocytes was equal to 20%, 30%, and so on, up to 90% of all hepatocytes. According to [55], if the model can predict the correct level of uninfected hepatocytes, then it should correctly model some of the above levels.

Similar to the experiment described in section 4.3.1, the results showed that it is possible to model the correct level of hepatocytes only if the number of virions does not have to fit the clinical data. However, it is impossible otherwise, in the realistic case. The results of the example experiment in which the number of uninfected hepatocytes is equal to 90% of the hepatocytes are presented in figure 7.



(a) Level of HCV RNA does not have to fit the data

(b) Level of HCV RNA should fit the data but it does not

Figure 7: Results of the simulation that maximizes the number of uninfected cells. It is possible to receive the high number of uninfected cells only if the number of viruses does not have to fit the clinical data.

5. Conclusions

In this paper, a multi-agent model of HCV infection was presented. It was designed based on an already existing differential equation model, but the new concept of using a multi-agent simulation was applied. This approach has many advantages in comparison to currently used modeling techniques (see also [20]), such as:

1. Interactions are written using biological terms instead of mathematical variables and parameters. Consequently, they reflect the actual complexity of the interaction network better. In addition, they can be easily understood and defined by the users who do not have expert mathematical knowledge.
2. It is easier to modify the interactions. Usually, after introducing a modification, only a simple change in the simulation's definition is required instead of a repetition of a complex mathematical analysis. Moreover, some components of the simulation (for example, selected agents with associated interactions) can be easily used in other simulations in completely different research.
3. The objective function and constraints can be more complex. They do not have to be limited to data fitting but, for example, can define expected or maximal rates of some processes or change in defined moments in time.
4. It is possible to define precise spatial dependencies and distinguish between different cells of the same type. In differential equations, all of the cells of the same type are assumed to behave identically, whereas in a multi-agent system, their behavior can depend on some attributes (for example, genetic mutations and genetic code analysis [61]).
5. There are greater possibilities for analyzing the results because each cell is simulated separately. In this way, we can analyze how a single cell or a group of cells affects the results of the experiment.
6. It is easy to model randomness by introducing random variables.

For example, during the computational experiment, it was demonstrated that it is trivial to define a custom objective function. In this case, the maximization of some of the values was performed to verify one of the already existing models. The experiment proved that the genetic algorithm can be successfully used to optimize this function and draw valuable results.

Moreover, the inverted simulation method was proposed. This method solves the problem of finding values of parameters used in models for some specific patient. Using this method, anyone can analyze a viral infection using techniques that are similar to models based on differential equations but making use of all of the advantages of a multi-agent simulation. This method is an important tool that was not described in any earlier work related to the use of multi-agent systems in viral infections modeling.

Acknowledgments

This study was partially supported by the Polish National Science Center under grant 2012/05/B/ST6/03026. In addition, SW was also supported by the scholarship within the project "Scholarship support for PH.D. students specializing in majors strategic for Wielkopolska's development", Submeasure 8.2.2 Human Capital Operational Programme, co-financed by the European Union under the European Social Fund.

Bibliography

[1] World Health Organization . Hepatitis C – global prevalence (update). Weekly Epidemiological Record 2000;75(3):18–9.

[2] Chen SL, Morgan TR. The natural history of hepatitis C virus (HCV) infection. International Journal of Medical Sciences 2006;3(2):47–52.

[3] Figlerowicz M, Alejska M, Kurzynska-Kokorniak A, Figlerowicz M. Genetic variability: the key problem in the prevention and therapy of RNA-based virus infections. Medicinal Research Reviews 2003;23:488–518.

[4] Domingo E, Gomez J. Quasispecies and its impact on viral hepatitis. Virus Research 2007;127(2):131 –50.

[5] Figlerowicz M, Formanowicz P, Kedziora P, Alejska M, Jackowiak P, Blazewicz J, et al. The clinical consequences of changes occurring in HCV population during the first weeks of chronic hepatitis C treatment with interferon and ribavirin. Przeglad Epidemiologiczny 2005;59(2):581–90.

[6] Figlerowicz M, Jackowiak P, Formanowicz P, Kedziora P, Alejska M, Malinowska N, et al. Hepatitis C virus quasispecies in chronically infected children subjected to interferon-ribavirin therapy. Archives of Virology 2010;155(12):1977–87.

[7] Jackowiak P, Kowala-Piaskowska A, Figlerowicz M, Alejska M, Malinowska N, Figlerowicz M. Evolution of hepatitis C virus hypervariable region 1 in chronically infected children. Virus Research 2012;167:380–4.

[8] Manns MP, McHutchison JG, Gordon SC, Rustgi VK, Shiffman M, Reindollar R, et al. Peginterferon alfa-2b plus ribavirin compared with interferon alfa-2b plus ribavirin for initial treatment of chronic hepatitis C: a randomised trial. Lancet 2001;358(9286):958–65.

[9] McHutchison JG, Everson GT, Gordon SC, Jacobson IM, Sulkowski M, Kauffman R, et al. Telaprevir with peginterferon and ribavirin for chronic HCV genotype 1 infection. New England Journal of Medicine 2009;360(18):1827–38.

[10] Poordad F, McCone Jr J, Bacon BR, Bruno S, Manns MP, Sulkowski MS, et al. Boceprevir for untreated chronic HCV genotype 1 infection. New England Journal of Medicine 2011;364(13):1195–206.

[11] Jackowiak P, Figlerowicz M, Kurzynska-Kokorniak A, Figlerowicz M. Mechanisms involved in the development of chronic hepatitis C as potential targets of antiviral therapy. Current Pharmaceutical Biotechnology 2011;12(11):1774–80.

[12] Dahari H, Guedj J, Perelson A, Layden T. Hepatitis C Viral Kinetics in the Era of Direct Acting Antiviral Agents and Interleukin-28B. Current Hepatitis Reports 2011;10:214–27. 10.1007/s11901-011-0101-7.

[13] Neumann AU, Lam NP, Dahari H, Gretch DR, Wiley TE, Layden TJ, et al. Hepatitis C viral dynamics in vivo and the antiviral efficacy of interferon-alpha therapy. Science 1998;282(5386):103–7.

[14] Perelson AS, Herrmann E, Micol F, Zeuzem S. New kinetic models for the hepatitis C virus. Hepatology 2005;42(4):749–54.

[15] Perelson AS. Mathematical and statistical approaches to AIDS epidemiology; chap. Modeling the interaction of the immune system with HIV. New York, NY, USA: Springer-Verlag New York, Inc. ISBN 0-387-52174-7; 1989, p. 350–70.

[16] Wodarz D, Nowak MA. Mathematical models of HIV pathogenesis and treatment. Bioessays 2002;24(12):1178–87.

[17] Sypsa V, Hatzakis A. Modelling of viral dynamics in hepatitis B and hepatitis C clinical trials. Statistics in Medicine 2008;27(30):6505–21.

[18] Mogilner A, Allard J, Wollman R. Cell Polarity: Quantitative Modeling as a Tool in Cell Biology. Science 2012;336(6078):175–9.

[19] Wasik S, Jackowiak P, Krawczyk JB, Kedziora P, Formanowicz P, Figlerowicz M, et al. Towards prediction of HCV therapy efficiency. Computational and Mathematical Methods in Medicine 2010;11(2):185–99.

[20] An G, Mi Q, Dutta-Moscato J, Vodovotz Y. Agent-based models in translational systems biology. Wiley Interdisciplinary Reviews: Systems Biology and Medicine 2009;1(2):159–71.

[21] Merelli E, Armano G, Cannata N, Corradini F, d'Inverno M, Doms A, et al. Agents in bioinformatics, computational and systems biology. Briefings in Bioinformatics 2007;8(1):45–59.

[22] Machado D, Costa RS, Rocha M, Ferreira EC, Tidor B, Rocha I. Modeling formalisms in systems biology. AMB Express 2011;1(1):1–14.

[23] Webb K, White T. Cell modeling using agent-based formalisms. In: Jennings NR, editor. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3. AAMAS '04; Washington, DC, USA: IEEE Computer Society. ISBN 1-58113-864-4; 2004, p. 1190–6.

[24] González PP, Cárdenas M, Camacho D, Franyuti A, Rosas O, Lagúnez-Otero J. Cellulat: an agent-based intracellular signalling model. Biosystems 2003;68(2):171–85.

[25] Pogson M, Smallwood R, Qwarnstrom E, Holcombe M. Formal agent-based modelling of intracellular chemical interactions. Biosystems 2006;85(1):37–45.

[26] Pogson M, Holcombe M, Smallwood R, Qwarnstrom E. Introducing spatial information into predictive NF-κB modelling–an agent-based approach. PLoS One 2008;3(6):2367–.

[27] Zhang L, Athale CA, Deisboeck TS. Development of a three-dimensional multiscale agent-based tumor model: simulating gene-protein interaction profiles, cell phenotypes and multicellular patterns in brain cancer. Journal of Theoretical Biology 2007;244(1):96–107.

[28] Engelberg JA, Ropella GEP, Hunt CA. Essential operating principles for tumor spheroid growth. BMC Systems Biology 2008;2:110–.

[29] Grant MR, Mostov KE, Tlsty TD, Hunt CA. Simulating properties of in vitro epithelial cell morphogenesis. PLoS Computational Biology 2006;2(10):129–.

[30] Emonet T, Macal CM, North MJ, Wickersham CE, Cluzel P. AgentCell: a digital single-cell assay for bacterial chemotaxis. Bioinformatics 2005;21(11):2714–21.

[31] Lollini PL, Motta S, Pappalardo F. Discovery of cancer vaccination protocols with a genetic algorithm driving an agent based simulator. BMC Bioinformatics 2006;7(1):352–.

[32] Li NY, Verdolini K, Clermont G, Mi Q, Rubinstein EN, Hebda PA, et al. A patient-specific in silico model of inflammation and healing tested in acute vocal fold injury. PLoS One 2008;3(7):2789–.

[33] Segovia-Juarez JL, Ganguli S, Kirschner D. Identifying control mechanisms of granuloma formation during M. tuberculosis infection using an agent-based model. Journal of Theoretical Biology 2004;231(3):357–76.

[34] Thorne BC, Bailey AM, Peirce SM. Combining experiments with multicell agent-based modeling to study biological tissue patterning. Briefings in Bioinformatics 2007;8(4):245–57.

[35] Amigoni F, Dini M, Gatti N, Somalvico M. Anthropic agency: a multiagent system for physiological processes. Artificial Intelligence in Medicine 2003;27(3):305–34.

[36] Amigoni F, Beda A, Gatti N. Multiagent systems for cardiac pacing simulation and control. AI Communications 2005;18(3):217–28.

[37] Cannata N, Corradini F, Merelli E, Omicini A, Ricci A. An Agent-Oriented Conceptual Framework for Systems Biology. In: Priami C, Merelli E, Gonzalez P, Omicini A, editors. Transactions on Computational Systems Biology III; vol. 3737 of *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer. ISBN 978-3-540-30883-6; 2005, p. 105–22.

[38] Sneddon MW, Faeder JR, Emonet T. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. Nature Methods 2011;8(2):177–83.

[39] Qu Z, Garfinkel A, Weiss JN, Nivala M. Multi-scale modeling in biology: how to bridge the gaps between scales? Progress in Biophysics and Molecular Biology 2011;107(1):21–31.

[40] Isern D, Sánchez D, Moreno A. Agents applied in health care: A review. International Journal of Medical Informatics 2010;79(3):145–66.

[41] Moreno A, Garbay C. Editorial: Software agents in health care. Artificial Intelligence in Medicine 2003;27(3):229–32.

[42] Nealon J, Moreno A. Agent-based applications in health care. In: Moreno A, Nealon J, editors. Applications of software agent technology in the health care domain. Whitestein series on Agent Technology; Basel: Birkhauser Eds. ISBN 978-3-7643-2662-3; 2003, p. 1–18.

[43] Moreno A. Guest Editor's Introduction: On the Evolution of Applying Agent Technology to Healthcare. IEEE Intelligent Systems 2006;21(6):8–10.

[44] Zaiyi G, Kwang HH, Cing TJ. Sufficiency verification of HIV-1 pathogenesis based on multi-agent simulation. In: Beyer HG, O'Reilly UM, editors. GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation. New York, NY, USA: ACM. ISBN 1-59593-010-8; 2005, p. 305–12.

[45] Mitha F, Lucas T, Feng F, Kepler T, Chan C. The Multiscale Systems Immunology project: software for cell-based immunological simulation. Source Code for Biology and Medicine 2008;3(1):6–.

[46] Itakura J, Kurosaki M, Itakura Y, Maekawa S, Asahina Y, Izumi N, et al. Reproducibility and usability of chronic virus infection model using agent-based simulation; comparing with a mathematical model. Biosystems 2010;99(1):70–8.

[47] Dahari H, Major M, Zhang X, Mihalik K, Rice CM, Perelson AS, et al. Mathematical modeling of primary hepatitis C infection: noncytolytic

clearance and early blockage of virion production. Gastroenterology 2005;128(4):1056–66.

[48] Dahari H, Ribeiro RM, Perelson AS. Triphasic decline of hepatitis C virus RNA during antiviral therapy. Hepatology 2007;46(1):16–21.

[49] Reluga TC, Dahari H, Perelson AS. Analysis of Hepatitis C Virus Infection Models with Hepatocyte Homeostasis. SIAM Journal on Applied Mathematics 2009;69(4):999–1023.

[50] Dahari H, Lo A, Ribeiro RM, Perelson AS. Modeling hepatitis C virus dynamics: Liver regeneration and critical drug efficacy. Journal of Theoretical Biology 2007;247(2):371–81.

[51] Talal AH, Ribeiro RM, Powers KA, Grace M, Cullen C, Hussain M, et al. Pharmacodynamics of PEG-IFN alpha differentiate HIV/HCV coinfected sustained virological responders from nonresponders. Hepatology 2006;43(5):943–53.

[52] Rong L, Dahari H, Ribeiro RM, Perelson AS. Rapid emergence of protease inhibitor resistance in hepatitis C virus. Science Translational Medicine 2010;2(30):30–2.

[53] Maiwald T, Eberhardt O, Sy SK, Blumberg J. Mathematical Modeling with PottersWheel - Introduction; 2 ed.; 2010.

[54] Macey R, Oster G, Zahnley T. Berkeley Madonna Users Guide. University of California; Berkeley; 2009.

[55] Liang Y, Shilagard T, Xiao SY, Snyder N, Lau D, Cicalese L, et al. Visualizing hepatitis C virus infections in human liver by two-photon microscopy. Gastroenterology 2009;137(4):1448–58.

[56] Zhang S, Liu J. A massively multi-agent system for discovering HIV-immune interaction dynamics. Massively Multi-Agent Systems I 2005;:572–3.

[57] Zhang S, Yang J, Wu Y, Liu J. An enhanced massively multi-agent system for discovering HIV population dynamics. Advances in Intelligent Computing 2005;:988–97.

[58] Terano T. Exploring the Vast Parameter Space of Multi-Agent Based Simulation. In: Antunes L, Takadama K, editors. Multi-Agent-Based Simulation VII; vol. 4442 of *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer; 2007, p. 1–14.

[59] Wright A. Genetic algorithms for real parameter optimization. Foundations of Genetic Algorithms 1991;1:205–18.

[60] Birch CP, Oom SP, Beecham JA. Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. Ecological Modelling 2007;206(3):347–59.

[61] Lukasiak P, Antczak M, Ratajczak T, Bujnicki JM, Szachniuk M, Adamiak RW, et al. RNAlyzer – novel approach for quality analysis of RNA structural models. Nucleic Acids Research 2013;41(12):5978–90.