



US 20170083683A1

(19) **United States**

(12) **Patent Application Publication**

Dutta-Moscato et al.

(10) **Pub. No.: US 2017/0083683 A1**

(43) **Pub. Date: Mar. 23, 2017**

(54) "METHODS FOR MODELING HEPATIC INFLAMMATION"

(60) Provisional application No. 61/352,116, filed on Jun. 7, 2010.

(71) Applicant: **University of Pittsburgh - Of the Commonwealth System of Higher Education**, Pittsburgh, PA (US)

Publication Classification

(51) **Int. Cl.**
G06F 19/00 (2006.01)
G06F 19/28 (2006.01)

(72) Inventors: **Joyeeta Dutta-Moscato**, Pittsburgh, PA (US); **Liang-I Kang**, Pittsburgh, PA (US); **Qi Mi**, Pittsburgh, PA (US); **Kari Nejak-Bowen**, White Oak, PA (US); **Yoram Vodovotz**, Sewickley, PA (US)

(52) **U.S. Cl.**
CPC *G06F 19/3437* (2013.01); *G06F 19/28* (2013.01)

(21) Appl. No.: **15/223,367**

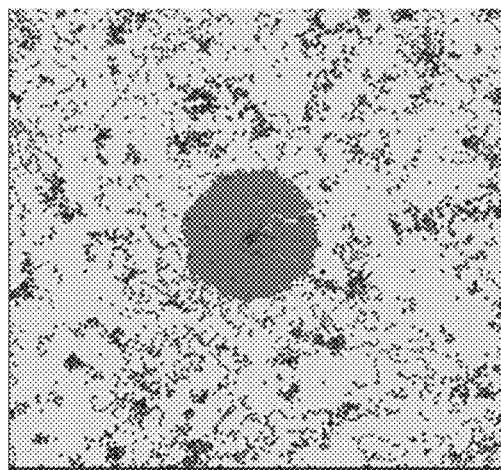
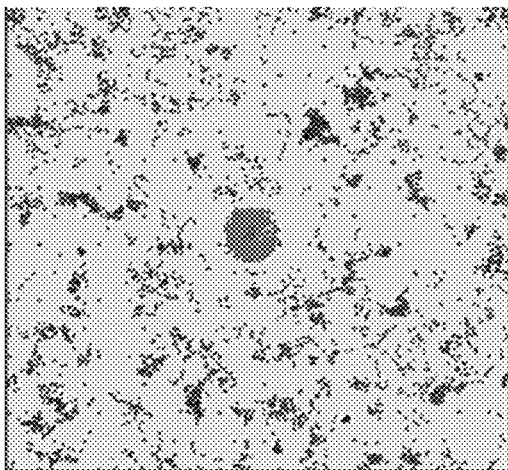
(57) **ABSTRACT**

(22) Filed: **Jul. 29, 2016**

Provided herein are in silico methods of modeling hepatic inflammation, fibrosis/cirrhosis, and cancer. The models are computer-implemented agent-based models and are useful in determining patient prognoses in hepatic conditions, including viral infections, damage, inflammation, and cancer. The modeling system also is useful in modeling the effects of active agents on normal hepatic tissue or hepatic tissue perturbed by inflammation, infection, damage, fibrosis/cirrhosis, and cancer.

Related U.S. Application Data

(63) Continuation of application No. 13/700,244, filed on Feb. 13, 2013, now abandoned, filed as application No. PCT/US2011/039400 on Jun. 7, 2011.



- Hepatocytes
- Stellate cell
- Kupffer cell
- Collagen deposit (fibrosis)
- Tumor

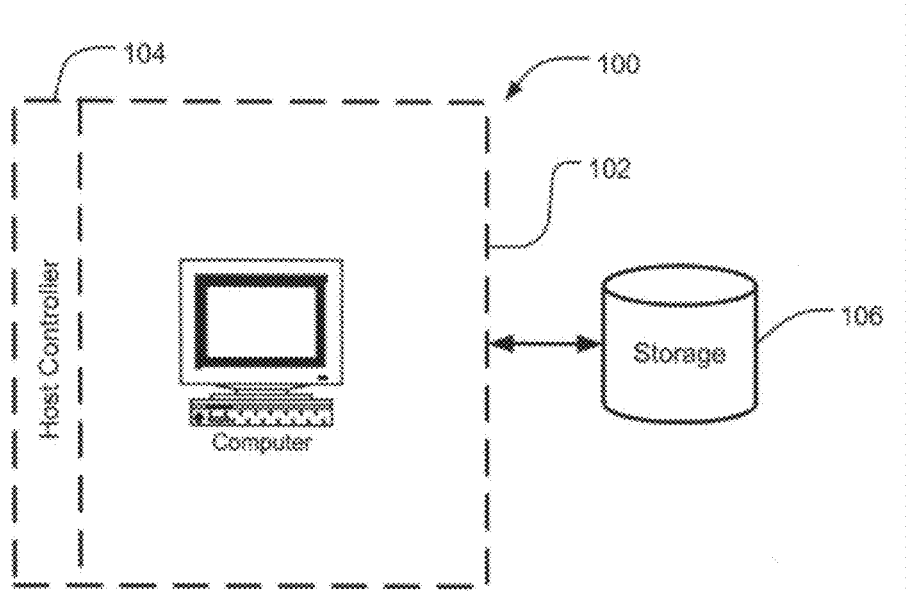


Fig. 1

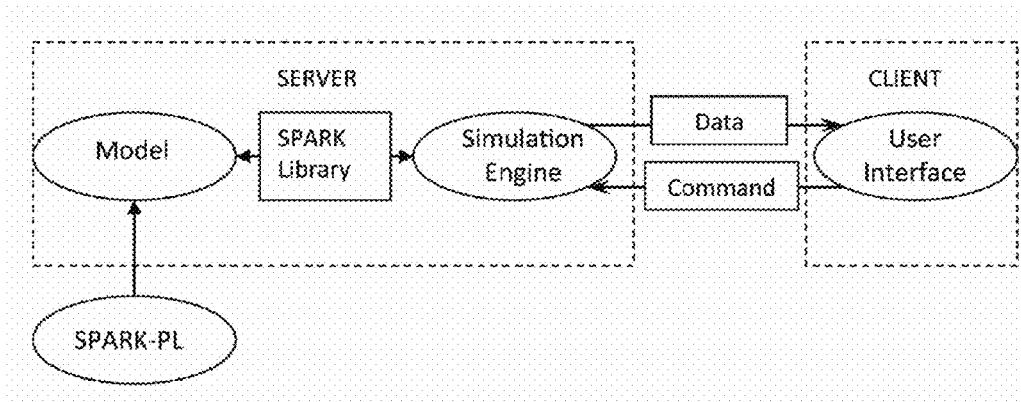


Fig. 2

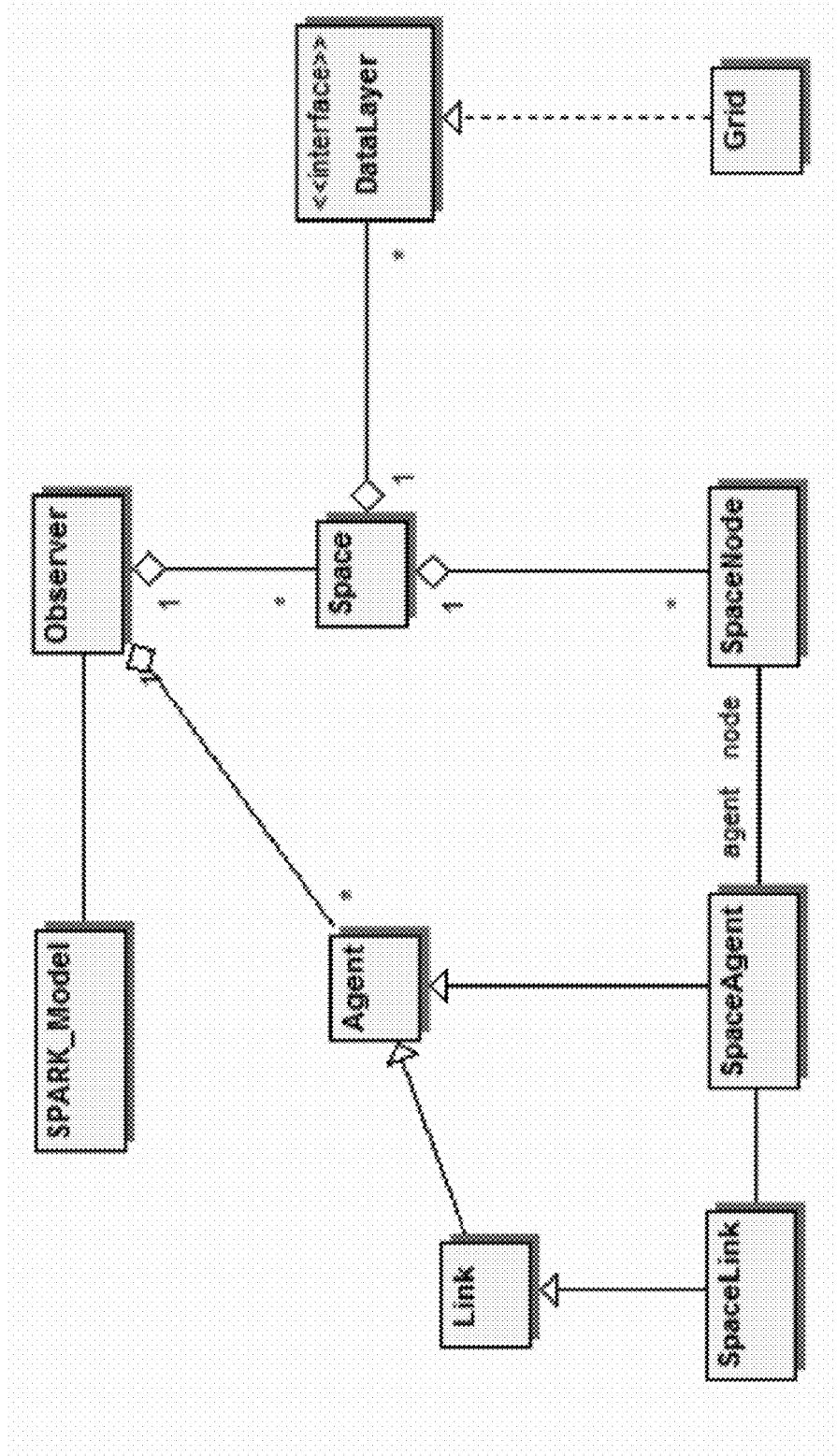


Fig. 3

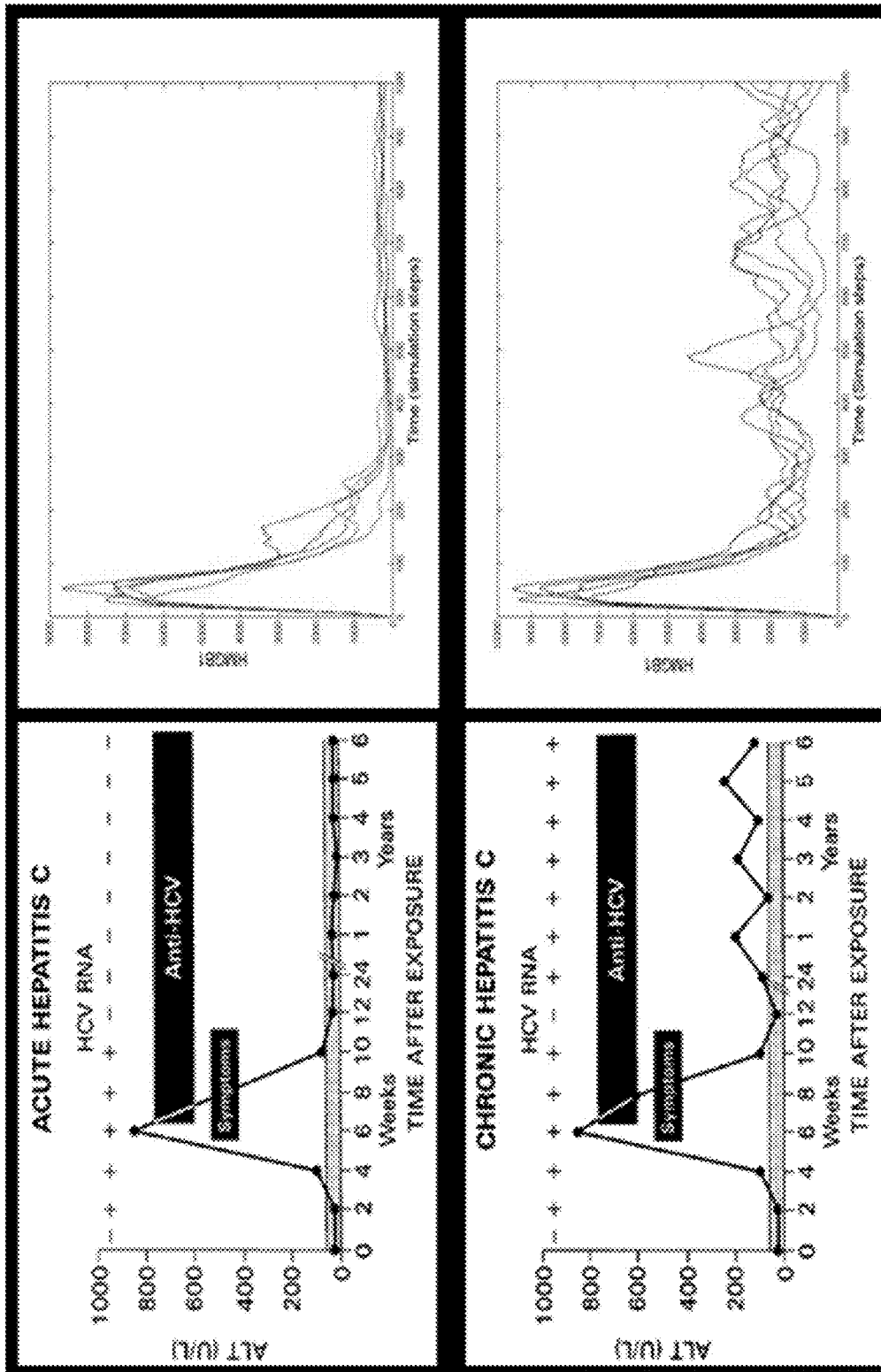


Fig. 5

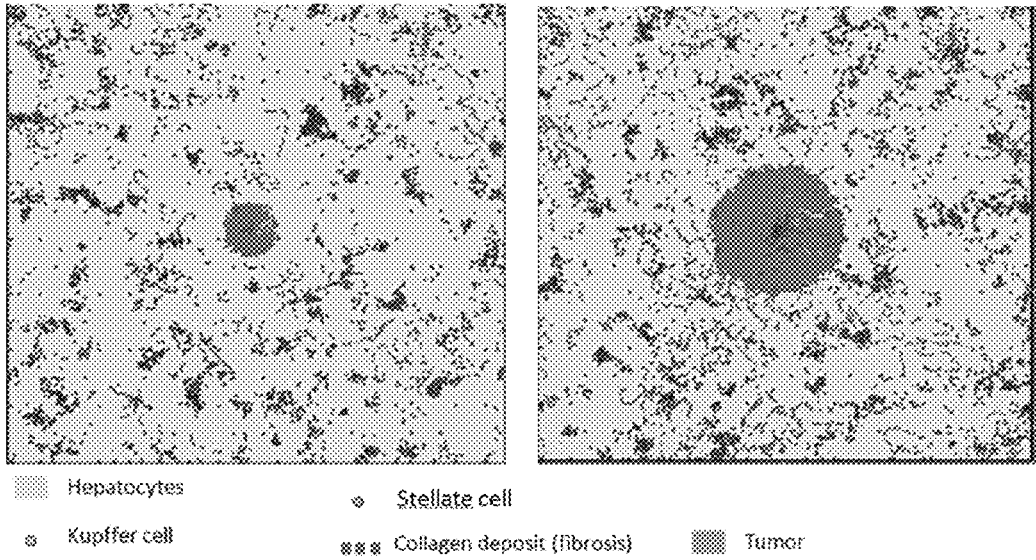


Fig. 6

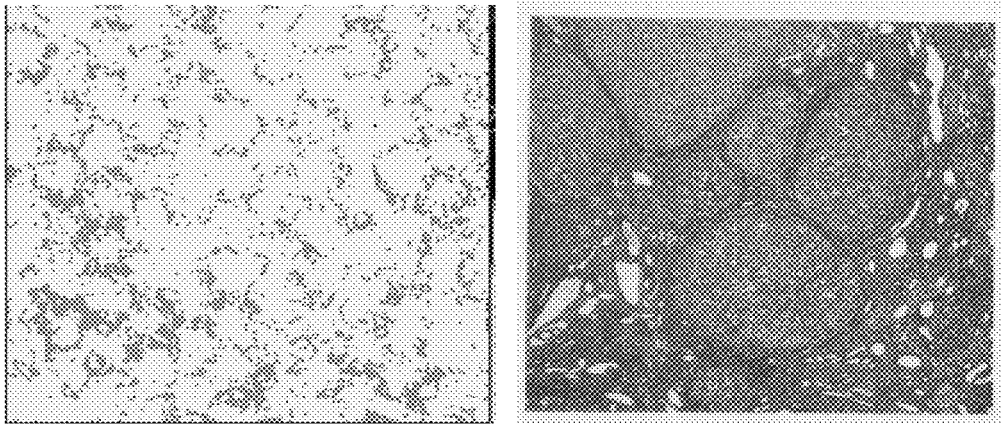


Fig. 7

**"METHODS FOR MODELING HEPATIC
INFLAMMATION"**

CROSS REFERENCE TO RELATED
APPLICATIONS

[0001] This application is a continuation of U.S. patent application Ser. No. 13/700,244, which is a national stage of International Patent Application No. PCT/US2011/039400, filed Jun. 7, 2011, which claims the benefit of U.S. Provisional Patent Application No. 61/352,116, filed Jun. 7, 2010, each of which is incorporated herein by reference in its entirety.

BACKGROUND

[0002] This invention relates to the development of computer models for simulating and creating simulated depictions of inflammation of hepatic tissue and of hepatocellular carcinoma (HCC).

[0003] Various approaches have been used to construct simulations of complex biologic processes. All these methods have distinct advantages and disadvantages. In vitro biological systems work well in many situations, but require physical facilities and often are not complex enough or accurate enough to effectively model in vivo systems. In silico (computer-simulated) systems are becoming more sophisticated and, as described below, are becoming increasingly able to model biological systems. Two modeling systems are commonly, but not exclusively, utilized to model biological systems: equation-based modeling (EBM), or more specifically ordinary differential equations (ODE), and agent-based modeling (ABM).

[0004] The ODE type of modeling is a type of equation-based modeling that consists of establishing a series of differential equations that describe the sequential change in the states of the components of the system over time. The differential equations are derived from known and hypothesized kinetics of the components of the biologic system. This approach has been used for many years to describe chemical systems, for example Michaelis-Menten kinetics. The variables of the equations generally represent average concentrations of the various components. These systems of equations are generally most accurate in settings in which large numbers of individuals of these components are assumed to exist and to exert their effects in aggregate. When the numbers become small, differential equation descriptions break down. The behavior of the system with limited spatial information (e.g. compartments) can be characterized with ODE; if more precise spatial resolution is desired, partial differential equations (PDE) are more commonly used. If simple enough, ODE can be solved analytically. If not, they can be easily solved computationally using a variety of commercially available and free software, as well as proprietary software designed for specific implementations of ODE models. Additionally, methods from nonlinear analysis can explore the properties of ODE without completely solving them. Because these equations are based on and describe biologic interactions, these models can potentially predict outcomes beyond the range of data on which these models were initially calibrated. In this latter aspect, EBM is different from statistical models. Furthermore, manipulation of a biologic mechanism can be entered into the model and an outcome derived (predicted).

[0005] The ABM type of modeling focuses on the rules and mechanisms of behavior of the individual components of a system, and may be more accurate than EBM in settings in which the stochastic actions of these agents is a better approximation of biological reality as compared to the actions of these components in aggregate. The components of a system are classified into types of "agents" by virtue of shared mechanisms that have been identified experimentally. The mechanisms are expressed as a series of conditional ("if-then") statements, and computer programs are written to describe the rules of behavior. An example would be the sequence of receptor activation involved in neutrophil adhesion. The model defines a "virtual world" based on characteristics of the reference system and generates populations of the various types of agents. The agents interact based on responses (defined by their rule systems) to inputs and outputs from their environment. For example, simulated cells would respond to variables in their immediate neighborhood, representing the extent of a cell's interaction with its extracellular milieu. The agents run in a parallel fashion to simulate simultaneous behavior, and the dynamics of the system are allowed to emerge from the multiple interactions among the agents over time. Consequently, all measured parameters and outcomes from the model are generated by the actions of the agents. The rules governing the behavior of agents should ideally be well-vetted, simple rules. Because ABMs are mechanistic models, any intervention that deals with a defined mechanism in the model can be simulated. Because they are based on rules, ABMs are often more intuitive to non-mathematicians than EBM (Ermentrout, G. B., et al. (1993). *Journal of Theoretical Biology*, 160(1), 97-133; An G. Agent-based computer simulation and SIRS: building a bridge between basic science and clinical trials. *Shock* 2001; 16(4):266-73; Vodovotz Y, Clermont G, Chow C, An G. Mathematical models of the acute inflammatory response. *Curr Opin Crit Care* 2004; 10:383-90; Vodovotz, Y., Csete, M.; Bartels, J.; Chang, S.; An, G. Translational systems biology of inflammation. *PLoS Comput. Biol.* 2008. 4:1-6; An, G.; Mi, Q.; Dutta-Moscato, J.; Solovyev, A.; Mikheev, M.; Vodovotz, Y. Agent-based models in translational systems biology. *Wiley Interdisciplinary Reviews—Systems Biology*. 2009. 1:159-171).

[0006] Inflammation and cancer are often interlinked. In the liver, hepatic inflammation can lead to fibrosis, cirrhosis, and cancer, e.g., hepatocellular carcinoma (HCC). For example, the failure of a patient's immune system to adequately clear an infection, such as in a chronic infection by Hepatitis C virus (HCV), results in liver inflammation and in some cases HCC. Development and calibration of a realistic model of liver inflammation, fibrosis, cirrhosis, and cancer development is hampered by the long-term nature of the related diseases and with respect to obtaining early-stage disease and multiple biopsies over time. Liver biopsies pose hazard to the patient, and are only taken when it is necessary. As such, early-stage biopsy data and multiple time point samples over the course of the disease are very rare.

SUMMARY

[0007] Provided herein are systems and methods of modeling hepatic inflammation, fibrosis and cancer. The systems and methods are useful for determining independent patient prognoses, as well as for drug development and testing and integrated digital pathology.

[0008] According to one embodiment, methods and systems of modeling progression of hepatic inflammation in a patient is provided. The method comprises inputting test data into an agent-based computer-implemented modeling system of hepatic inflammation, running the model, and producing output of the modeling system that comprises one or more of the following for one or more future time points: inflammation state, viral load, viral damage, presence of a tumor, size of a tumor, presence of cancer (metastasis), size or extent of metastasis, state of fibrosis and/or eventual cirrhosis. The agent-based computer-implemented modeling system comprises a plurality of agents and measurable factors, the agents, including biological elements of liver inflammation, fibrosis and cancer, such as hepatocytes, macrophages, stellate cells and tumor cells; and measurable factors, such as the inflammatory cytokine tumor necrosis factor- α (TNF- α), the cytokine transforming growth factor- β 1 (TGF- β 1), and the damage-associated molecular pattern molecule (“danger signal”) high-mobility group box-1 (HMGB1).

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 illustrates one embodiment of a computer system for implementing a modeling system as described below.

[0010] FIG. 2 is a flow diagram of the SPARK agent-based modeling software application as implemented in a client-server environment.

[0011] FIG. 3 is a diagram showing connection between main SPARK model components

[0012] FIG. 4 is a flow diagram showing the inter-relationship of elements of the SPARK-implemented ABM described in Example 1.

[0013] FIG. 5: Left panel: The time course of serum alanine aminotransferase (ALT) levels in a typical case of acute, self-limited hepatitis C (top) and chronic hepatitis C (bottom) [Hepatitis C: The Clinical Spectrum of Disease, Hoofnagle J H (1997) Hepatology 26(Suppl 1):155-205]. Right panel: The time course of HMGB1 levels in our simulated liver section, for several independent viral inoculations of Hepatitis C. Some instances of infection led to acute, self-limited hepatitis C (top) while others resulted in chronic hepatitis C (bottom)

[0014] FIG. 6: Initial tumor formation (left) and progression (right) over the following 12 years in one case of chronic hepatitis C.

[0015] FIG. 7: Appearance of periportal and bridging fibrosis in a liver ABM (left) similar to patterns of early cirrhosis as seen clinically in Masson-Trichrome stained liver samples (right).

DETAILED DESCRIPTION

[0016] The use of numerical values in the various ranges specified in this application, unless expressly indicated otherwise, are stated as approximations as though the minimum and maximum values within the stated ranges are both preceded by the word “about”. In this manner, slight variations above and below the stated ranges can be used to achieve substantially the same results as values within the ranges. Also, unless indicated otherwise, the disclosure of these ranges is intended as a continuous range including every value between the minimum and maximum values.

For definitions provided herein, those definitions refer to word forms, cognates and grammatical variants of those words or phrases

[0017] As used herein, the terms “comprising,” “comprise” or “comprised,” and variations thereof, are meant to be open ended. The terms “a” and “an” are intended to refer to one or more.

[0018] As used herein, the term “patient” or “subject” refers to members of the animal kingdom including but not limited to human beings and “mammal” refers to all mammals, including, but not limited to human beings.

[0019] As indicated above, a system and method are provided for modeling progression of the inter-related conditions of hepatic inflammation, fibrosis and cancer in a patient. The systems and methods are computer-implemented. The method comprises inputting test data into an agent-based computer-implemented modeling system of hepatic inflammation, running the model, and producing output of the modeling system that comprises one or more of the following for one or more future time points: hepatic tissue structure, hepatic inflammation state, inflammation-induced hepatic damage, hepatic viral load, virally-induced hepatic damage, hepatic tumor presence, hepatic tumor size, hepatic cancer presence, hepatic metastasis size, hepatic metastasis extent, hepatic fibrosis state, or any combination thereof. The agent-based computer-implemented modeling system is implemented on at least one computer with a computer readable medium having programming instructions stored thereon. In the system, the programming instructions, when executed by a processor of the at least one computer, cause the processor to: receive or define at least one of the following: agent data, global data, initialization process data, or any combination thereof, the agent data comprising at least one of the following: definition data, behavior data, class data, internal state data, function data, link data, space data, or any combination thereof; model the at least one hepatic condition associated with at least one agent based at least in part upon at least a portion of the agent data corresponding to the at least one agent; and generate modeling output data representative of at least one of the following: hepatic tissue structure, hepatic inflammation state, inflammation-induced hepatic damage, hepatic viral load, virally-induced hepatic damage, hepatic tumor presence, hepatic tumor size, hepatic cancer presence, hepatic metastasis size, hepatic metastasis extent, hepatic fibrosis state, or any combination thereof. The agent data, global data, initialization process data, or any combination thereof represent biological elements of liver structure, such as portal triads and/or hepatic lobules, biological elements of liver inflammation, fibrosis and cancer, such as hepatocytes, macrophages, stellate cells and/or tumor cells; and measurable factors, such as TNF- α , TGF- β 1, and/or HMGB1. The model includes relevant cells, cytokines, and processes, including the response to tissue damage, the process by which cell death leads to a hypoxic core, and the process by which a hypoxic region stimulates tumor angiogenesis. The system and methods facilitate identification and testing of active agents (e.g., drugs) that prevent inflammation, tissue damage, fibrosis or cancer based on their impact on the model. Likewise, the systems and methods facilitate generation of exemplary representative digital representations of cell and tissue structure at any stage of the modeling process, permitting characterization of images of a sample biopsy obtained from a patient, and determining a patient's

prognosis by comparison of the patient's test image with a library of the exemplary images.

[0020] The inflammation, fibrosis and cancer modeling platform described herein is generally useful since it allows for simulating the responses of a slice of liver to diverse insults that, in turn, lead to inflammation, which in turn leads to tissue damage and/or dysfunction, which with some probability will in turn heal appropriately or become fibrotic or cirrhotic, and which, with some probability, can allow for the uncontrolled proliferation of cancer stem cells (or cancer progenitor cells, which are functionally equivalent in this ABM). These cancer stem cells/progenitor cells will lead to the formation of a tumor, which can develop a hypoxic core that will develop new blood vessels and hence lead to further tumor growth as well as metastasis. This general-purpose nature of the simulation will allow for developing general therapeutic strategies for inflammation, fibrosis, cirrhosis, and cancer in the liver, under conditions specific to individuals or groups (e.g. situations in which there is a high level of baseline inflammation due to alcoholism [a risk factor for cirrhosis and liver cancer]), high cholesterol/obesity, overdose of acetaminophen, exposure to liver toxins, chronic infections, or other diseases that affect the liver).

[0021] Based on the above attributes of the simulation, this simulation may be used for drug development and testing. The current methodology for the design of therapies for HCV-induced HCC is rooted in basic science and pre-clinical studies related to cancer treatment. The basic science studies are generally carried out in cell culture, which cannot reproduce the complex environment of cancer in vivo or the interaction between cancerous cells and surrounding healthy tissues. Preclinical studies are generally carried out in rodents that are either subjected to artificially administered doses of cancerous agents or genetically modified to develop HCC. These studies suffer from the limitations that the damage is acute and not chronic, that it occurs not through HCV infection but through alternate pathways to HCC that are chemically, genetically, or radioactively induced instead of virally induced, and that the progression of cancer in these animals differs in some respect from the human scenario. The strategy to be used for rational design of anti-liver cancer drugs, targeting various aspects of the inflammatory cascade, the fibrosis/cirrhosis process, the carcinogenesis process, or the tumor angiogenesis process embodied in the computational simulation, is to minimize overall tissue damage/dysfunction. A computerized algorithm would search the parameter space of the computational model of inflammation/cancer, in order to determine what changes to the parameters characteristic of the inflamed, fibrotic, cirrhotic, or cancerous state (in which damage/dysfunction is high) will result in reducing this damage/dysfunction to levels characteristic of a healthy tissue (and, concomitantly, to restore the healthy appearance of tissue biopsies). The iterative strategy would include verification of the effects of the drug on the various parameters both in vitro and in vivo, with verification of the reduction of damage/dysfunction in vivo. A therapeutic agent obtained via this strategy would be any agent that is derived from a computational simulation describing the inter-related cascade of inflammation, fibrosis/cirrhosis, and liver cancer and that culminates in tissue damage/dysfunction, and that is subsequently vetted in screening assays carried out both in vitro and in vivo. The efficacy of this agent is then subsequently tested in a simulated clinical trial, in which a variable population is

generated using variation in insult size/inoculum as well as variation in inflammatory products. Below is an example of such a therapeutic strategy:

[0022] An automated search of the parameter space of the computational model described above suggests that a drug candidate that will reduce liver damage/dysfunction sufficiently to increase survival in a rat model of liver cancer will have the following properties: in vitro reduction of the responsiveness of liver macrophages (Kupffer cells) to HMGB1 as well as a reduction in the capacity of macrophages to produce TNF- α , in vitro elevation of the capacity of liver stellate cells to produce active TGF- β 1, in vivo reduction in tissue levels of TNF- α and HMGB1, and an overall reduction in liver fibrosis observed histologically. This strategy would allow for the simulation of the effects of the proposed agent in a randomized, placebo-controlled clinical trial prior to carrying out such a trial, in order to further verify the clinical efficacy of the proposed drug.

[0023] One use of the ABM leverages the fact that this type of simulation can often reproduce a visual pattern of cells and tissue architecture, which in turn can be compared to a photomicrograph. Moreover, since the ABM recapitulates a biological process, additional information such as distribution of given molecules (e.g. inflammatory cytokines) can be depicted. This pattern can be compared to that seen upon immunostaining of tissue sections for such molecules.

[0024] The pattern observed in the ABM simulation is derived from a simulated biological process, and thus calibrating the ABM for the pattern observed in actual photomicrographs or immunostained tissue sections can yield a quantitatively predictive (in both time and levels of cells/mediators) that essentially leads to a patient-specific, personalized simulation of the cancer process. In addition, this mechanistic foundation of the simulation means that simulated therapies can be run in silico on a patient-specific basis. Thus, this platform allows for a "smart diagnostic", which not only can lead to better prediction of the course of disease, but that can also be used to suggest specific therapeutic options.

[0025] As used herein "agent-based modeling" is a rules-based, stochastic modeling framework in which agents obey simple rules that, as a whole, generate complex system behavior.

[0026] As used herein, a "differential equation" is an equation in which the derivatives of a function appear as variables. In equation-based modeling (EBM), the model is a set of equations, and execution consists of evaluating them. An ordinary differential equation (or ODE) is a relation that contains functions of only one independent variable, and one or more of its derivatives with respect to that variable.

[0027] As used herein a "biological system" is a subset or microcosm of the physical structure and/or activities that are present in a living organism, such as a cell, plant, animal, vertebrate or invertebrate. A biological system may be, without limitation, an organ, a biochemical or biological process, a biochemical process present in an organ, tissue or organism, a pathogen and infected tissue, diseased tissue, etc. In the context of the present disclosure, biological systems include, for example and without limitation, cells, tissue, organs, cells and factors (chemical compounds, proteins such as cytokines, DNA, RNA, and other cellular and non-cellular elements) involved in any fashion with the

process of hepatic inflammation, fibrosis and cancer in any context. As such, “hepatic inflammation, fibrosis and cancer” refers to the process of liver inflammation caused, e.g., by hepatic viral infection or injury, and the progression of the inflammatory state to fibrosis (including, for example, cirrhosis) and/or cancer. Elements thereof include organs, tissue, cells and factors that are physically present and/or active in the biological system. As used herein, hepatic inflammation, fibrosis and cancer refer to hepatic inflammation, fibrosis and cancer, no matter the cause of the inflammation.

[0028] As used herein a “modeling system” is a computational framework by which a biological system is modeled. A modeling system embodies various elements of a biological system. As used herein an “element,” in the context of modeling a biological system, means any components of the model, as described above, including, without limitation, agent data, global data, initialization process data, data layers, representative grids and images, and input parameters. For example, elements of the SPARK platform, described below include, for example: agents, spaces, data layers, the observer, links, etc.

[0029] A modeling system may be “calibrated” by assigning, relating, comparing, or otherwise comporting elements of a modeled system with actual data values obtained directly or in any other manner from actual data values, for example and without limitation, by statistical computation methods or other mathematical methods. Calibration typically involves iterative adjustment of model parameters needed to achieve a situation in which the output of a mathematical model matches the experimentally determined time courses of the analytes (variables) in the model. The quality of the model and the calibration procedure are typically assessed by a process of “validation”, in which predictions of the model are tested against data withheld from the calibration procedure.

[0030] Modeling systems are implemented on a computing device (computer) as processes. In the context of computing, a process is, broadly speaking any computer-implemented activity that generates an outcome, such as implementation of a mathematical or logical formula or operation, algorithm, etc. FIG. 1 illustrates one embodiment of a system 100 for implementing a modeling system. The system 100 may include a device 102 operating under the command of a controller 104. Device 102 may be referred to herein, without limitation, as a computer or computing device. The broken lines are intended to indicate that in some implementations, the controller 104, or portions thereof considered collectively, may instruct one or more elements of the device 102 to operate as described. Accordingly, the functions associated with the modeling methods (e.g., processes, software, programs) described herein may be implemented as software executing in the system 100 and controlling one or more elements thereof. An example of a device 102 in accordance with one embodiment of the present invention is a general-purpose computer capable of responding to and executing instructions in a defined manner. Other examples include a special-purpose computer including, for example, a personal computer (PC), a workstation, a server, a laptop computer, a web-enabled telephone, a web-enabled personal digital assistant (PDA), a microprocessor, an integrated circuit, an application-specific integrated circuit, a microprocessor, a microcontroller, a network server, a Java™ virtual machine, a logic array, a

programmable logic array, a micro-computer, a mini-computer, or a large frame computer, or any other component, machine, tool, equipment, or some combination thereof capable of responding to and executing instructions.

[0031] In one non-limiting embodiment, system 100 is implemented as a PC. Furthermore, the system 100 may include a central processing engine including a baseline processor, memory, and communications capabilities. The system 100 also may include a communications system bus to enable multiple processors to communicate with each other. In addition, the system 100 may include storage 106 in the form of computer readable medium/media, such as a disk drive, optical drive, a tape drive, flash memory (e.g., a non-volatile computer storage chip), cartridge drive, and control elements for loading new software. In embodiments of the invention, one or more reference values may be stored in a memory associated with the device 102. Data, such as images produced by the methods and systems described herein may be organized on computer readable media in a database, which is an organized collection of data for one or more purposes, usually in digital form

[0032] Embodiments of the controller 104 may include, for example, a program, code, a set of instructions, or some combination thereof, executable by the device 102 for independently or collectively instructing the device 102 to interact and operate as programmed, referred to herein as “programming instructions”. One example of a controller 104 is a software application (for example, operating system, browser application, client application, server application, proxy application, on-line service provider application, and/or private network application) installed on the device 102 for directing execution of instructions. In one embodiment, the controller 104 may be a Windows™ based operating system. The controller 104 may be implemented by utilizing any suitable computer language (e.g., C/C++, UNIX SHELL SCRIPT, PERL, JAVA™, JAVASCRIPT, HTML/DHTML/XML, FLASH, WINDOWS NT, UNIX/LINUX, APACHE, RDBMS including ORACLE, INFORMIX, and MySQL) and/or object-oriented techniques.

[0033] In one embodiment, the controller 104 may be embodied permanently or temporarily in any type of machine, component, physical or virtual equipment, storage medium, or propagated signal capable of delivering instructions to the device 102. In particular, the controller 104 (e.g., software application, and/or computer program) may be stored on any suitable computer readable media (e.g., disk, device, or propagated signal), readable by the device 102, such that if the device 102 reads the storage medium, the functions described herein are performed. For example, in one embodiment, the controller 104 may be embodied in various computer-readable media for performing the functions associated with processes embodying the modeling methods.

[0034] Liver models described herein may be implemented in a number of programming platforms for implementing Agent-based modeling applications for creating Agent-Based Models (ABMs) by use of Agent-Based Model Modules (ABMMs), which are computer processes for running ABMs. In the examples below, a custom-built programming platform, the Simple Platform for Agent-based Representation of Knowledge (SPARK) was used. SPARK is a stand-alone application written in the Java™ programming language. It provides a user-friendly interface, and a simple programming language for developing agent-based models

(ABMs). SPARK has the following features specialized for modeling biomedical systems: 1) continuous space that can simulate real physical space; 2) flexible agent size and shape that can represent the relative proportions of various cell types; 3) multiple spaces that can concurrently simulate and visualize multiple scales in biomedical models; 4) a convenient graphical user interface. Other Agent-based modeling software include: StarLogo®, NetLogo®, Swarm (www.swarm.org; Swarm Development Group wild), Repast, MASON, and Ascape. Among these, NetLogo® is one of the most popular for biomedical modeling, particularly by non-formally-trained programmers, due to its user-friendly interface and the natural language-like syntax of its Logo-based programming language. SPARK, along with the SPARK-PL language was designed with Logo-like syntax, so that users could benefit from some capabilities currently not found in NetLogo® and similar software. These include the special features of SPARK cited earlier, as well as the ability to organize code into modules that can map to biological processes, as well as offering the potential for parallelization in distributed computer architectures. Non-limiting examples of ODE software programs include: XPPAUT (www.math.pitt.edu/~phase; an ODE simulation program), Mathematics® and MATLAB®; an ODE simulation program. Elements of EBM or ODE modeling can be readily implemented within ABM software.

[0035] The elements of the ABM models described herein are termed “agent data, global data, initialization process data, or any combination thereof” as generic for the various elements of each model, irrespective of the software package used. For instance in the context of SPARK, agent data, global data, initialization process data, or any combination thereof refer to agents, spaces (e.g., grids), data layers, links, the observer, etc.

[0036] One embodiment of a useful Agent-Based Modeling software platform is SPARK, which is described in detail for illustrative purposes. SPARK is implemented as a client-server application. There are five main components which constitute SPARK (FIG. 2), including a specialized SPARK programming language (SPARK-PL) that can be used for model development. The use of SPARK-PL is optional; SPARK models can be written directly in Java™ as well. User models are created with the functionality provided by the SPARK library, which contains definitions of the main SPARK concepts: Agents, Spaces, Data Layers and the Observer. The simulation engine runs user models. It uses classes from the SPARK library to communicate with user models.

[0037] The user interface allows working with SPARK models interactively. Users can load models, set parameters of their choice, run simulations, and visualize results. The simulation engine is completely independent from the user interface. The user interface (which runs on a client machine) communicates with the simulation engine (which runs on the server) by sending commands to the server and by receiving data from the server. The received data can be saved on a disk, analyzed, or visualized by the client. In general, data sent by the server contains information about the state of a simulation: values of variables, positions of agents in a space, etc.

[0038] There are several possible implementations of the SPARK client and server. One option is to implement them inside the same program. In this case, the communication process will be very fast, but the client and server need to

share the resources of the same machine. Another option is to implement the client and server as two separate programs, and establish the communication between them using a network interface.

[0039] In this situation, it is optimal not to send all data after each simulation step. Instead, data can be sent at specific time intervals using data buffers, or data can be collected and sent only after the end of a simulation. The two versions of the implementations of client and server are designed for different purposes of simulations. The first version can be applied for models with relatively short simulation time but requiring quick visualization of output. The latter version is suitable for large scale simulations where the users may want to view the results after thousands of simulation steps. In the standard SPARK distributions, both versions of the client and the server are available.

[0040] From a modeler’s point of view, the most important components of SPARK are 1) the SPARK library; 2) SPARK-PL; and 3) the SPARK user interface. Key concepts of the SPARK library, the SPARK programming language, and the user interface are described below.

[0041] SPARK models are created by using basic components of the SPARK library. A SPARK model consists of the following elements: definitions of agents and a global model description. A model description includes the initialization process for a model (the “setup” method) and contains the global variables and parameters. It also includes methods which are called before and after each simulation step, and which are primarily used for changing global variables and collecting statistics.

[0042] The main components of SPARK models are Space, Data Layers, Agents, Links, and the Observer (FIG. 3). Space is analogous to the physical space, and provides a context within which the model evolves. For example, the ABM contains spaces representing liver tissue as a contiguous layer of hepatocytes. Data Layers provide a means of tracking multiple variables within the same model. For example data layers can be used to track the amounts of the inflammatory cytokine tumor necrosis factor- α (TNF- α). Agents can move, perform functions, interact with each other, and also interact with the space they occupy. Classes of agents can correspond to different cell types involved in inflammation, fibrosis and cancer. Each agent has its own set of behaviors and rules of action. For example, inflammatory cells (macrophages) are chemo-attracted and activated by platelets.

[0043] Links can be used to connect two or more agents together with various strengths and distances. Not all biomedical systems can be modeled using a number of independent agents. In some systems, agents are expected to link together. For instance, in normal situation, platelets in the blood can be represented as separate agents in a SPARK model; once these platelets are activated, however, they would change their shape and tightly bind together and also to the injured tissue. In this case, we need a way to build the connection among these agents. The Observer contains information about spaces and all agents in the model and also controls the simulation process.

[0044] A SPARK model consists of agents that have some predefined behavioral rules, and a global description of the model itself: classes of agents in the model, initialization process, global variables. Generally, the simulation proceeds as follows. First, the model is initialized. Then, the simulation loop begins. A global function, defined in a model class,

is called at the beginning of each simulation step. Next, all agents are asked to perform their steps (execute their behavior function). After that, a global function for finalizing a step is called. The simulation ends when a predefined number of simulation steps is achieved, or when some user-defined conditions are met.

[0045] Agents are the central components of a SPARK model; the modeling process consists of creating agents and programming their behavior. Each agent can be characterized by two main features: its class (like inflammatory cell class) and its internal state. The internal state of an agent is the collection of variables and their values, which completely describe the agent. Agents of the same class have identical rules (the same functionality). In keeping with object oriented specifications, classes of agents form a hierarchical structure, with the base class called, simply, “Agent”. The class is the first thing that is defined for any agent. Many operations in SPARK work directly with “Agent” classes.

[0046] For example, there are functions that return all agents of a specific class possessing some special properties (e.g., occupying some location in a space). One feature of an agent class is that all agents of the same class can be scheduled to act at specific time points (see the description of the Observer for more details). The behavior of each agent is specified in the “step” function defined for any agent. This function is the same for all agents of the same class. The only parameter of the “step” function is the information about time passed from the beginning of a simulation.

[0047] Another function defined for each agent is the function “die”. It kills the agent and removes it from a simulation. The exact behavior of that function depends on the current execution mode and is described later. Agents can change their own state variables. If an agent wants to change some variables of another agent, it can do so directly, but this is not recommended. If an agent changes variables of another agent, it can create inconsistencies in a parallelized environment. Agents can change states of other agents indirectly. For example, an agent can change some value in a data layer or a global variable and this, in turn, will change the state of other agents. Agents may read the state of another agent, though care should be taken in the case of parallelized simulation, since several agents that read the same variable of another agent during a simulation step can get distinct values depending on the time when the variable was read.

[0048] Links are special agents that can connect two other agents, so that each agent knows to which other agents it is connected. Links are agents, so they have the “step” function and they have specific behavioral rules. One class of links is the “SpaceLink” which can be used to connect SpaceAgents (which are described later) in a space. SpaceLinks have a special function that returns its length, e.g., the distance between two connected agents.

[0049] One particular example of a link is a model for a physical spring that connects two agents. In this case, the behavior of a link (spring) is the application of forces to connected agents if the length of the link (spring) has been changed. For instance, in vascular tissues, smooth muscle cells are normally attached to each other, which can be simulated by these physical springs. We can adjust the spring coefficient and the rest length of the spring to fit the desired behavior of smooth muscle cells. When there are forces applied to these smooth muscle cells, these springs may

change their lengths and also generate forces against the original force. In this case, we would be able to see the dynamic changes of smooth muscle cells under internal and external forces.

[0050] A space is a structure which groups SPARK agents based on their positions. Each SPARK model may contain several spaces with different properties; however, one “base” space needs to be defined as the default space for that model. A space facilitates interactions between agents, and provides additional organization and structure for their interactions. It is not required for an agent to exist in a space; it is possible to define agents that have no spatial properties. Examples of such agents include those that modify values in a data layer at each simulation step, or those that collect some statistics.

[0051] Agents that can exist in a space are derived from a SPARK class called “SpaceAgent”. All SpaceAgents can freely move in a space where they are located, and they can change their size and shape. When a new SpaceAgent is created, it is automatically put into the default space of a model. Any SpaceAgent should be located in only one space at one time step, but they can move from one space to another.

[0052] Technically, SpaceAgents are not directly contained inside a space. Instead, each Space-Agent contains a variable of type “SpaceNode”, which associates an agent and a space. In other words, SpaceAgents are attached to SpaceNodes, and SpaceNodes are entities that exist in a space. A SpaceNode contains all information about “physical” appearance of an agent in a space, such as its coordinates, size and shape. An agent can be thought of as a brain that controls a body (SpaceNode). This approach allows changing SpaceNodes for an agent dynamically. For instance, all SpaceNodes have a fixed shape, and if an agent wants to change its shape, then it is enough to create a new SpaceNode that has a desired shape and attach an agent to this new node. There are circular SpaceNodes, square SpaceNodes; this is done for efficient collision detection algorithms. Each space has some topological structure. It is not required to have a Euclidean coordinate system in a space, but in the current version of SPARK only Euclidean spaces are available. There are two main types of spaces in SPARK: a continuous space and a grid (discrete) space. Each type can be either two-dimensional or three-dimensional. The basic topologies of these spaces are rectangular, cylindrical, or toroidal.

[0053] A continuous space has the usual Euclidean coordinate system. In a grid (discrete) space, some operations have special meaning. A grid space is divided into cells—squares in a two-dimensional space, cubes in a three-dimensional space—and all agents in the same cell are considered to be at the same location on the grid. There are several operations available in a space itself. One can get all agents at a specific location, get all agents of a given class at a specific location, or all agents that overlap with another specified agent type. Using space allows for more efficient operation than iteration over all agents. Other operations in space include getting space size and space topology.

[0054] Data layers are used to define a scalar field in a space. A data layer associates a numerical value with each point of a space, making it easy to work with these values and to perform operations on them. Data layers can be implemented in different ways. The simplest way is to define a grid structure in the bounded space. Each cell of the grid

keeps some numerical value, and all points in the space inside this cell are associated with this value. Computations involving grids can be performed quite efficiently since each grid stores numeric values in a rectangular table.

[0055] The data layers can be used to represent the location specific values which can have smaller scale than agents. For example, if agents represent cells and a space represents a tissue then data layers represent chemical compounds in the tissue. In SPARK, data layers can be of different scale. Some data layers can be coarse, other can be fine. Each data layer is associated with one space for which the data layer is defined. Each space contains a complete list of all data layers inside it. SpaceAgents can read and change values of any data layer at any position of a space. In particular, it is common to work with data layers at the location of an agent itself. There are special operations defined for data layers which allow reading and modifying values at the location of a specific agent. Reading and changing values at a specific point of a data layer are local operations for a data layer.

[0056] Global operations defined for data layers include diffusion and evaporation. There are two implementations of a diffusion operation in SPARK. One is a simple diffusion (similar to NetLogo), which diffuses a value of a data layer cell to all adjacent neighbors. Another version of a diffusion operation is an improved version of a simple diffusion that allows diffusion to all neighbors at a specific distance. Evaporation implements degradation of the value of a data layer at the specified location.

[0057] The Observer has two main functions: it is the main container for all other SPARK components and it schedules and executes actions of other components. Agents and spaces are contained directly in the Observer. Basic functions of the Observer allow retrieval of information about created agents (for example, a number of agents of a given class) and sets of agents. When a new agent is created it is automatically added to the Observer. When an agent dies, it is removed from the Observer. The Observer also can be used to get information about spaces defined in a model; data layers are stored in spaces, so to get information about a data layer, it is required to get space first, and then retrieve a data layer.

[0058] There are two main modes in which the Observer executes actions of agents. The first mode is called the serial mode. In this mode, agents carry out their steps one after another and all actions of agents have an immediate effect. For example, if an agent hatches another agent, then the hatched agent is immediately added to the set of existing agents and all other agents are able to see it. The second operation mode is called the concurrent mode. In this mode, many actions of agents are postponed until the end of a simulation step, and executed only after all agents have made their steps. For instance, if the Observer runs in the concurrent mode, then a newly created agent will not be added to the set of agents immediately. It will be added only after all agents finish their steps. The main idea of the concurrent mode is to make the order of agents' actions irrelevant. This mode is particularly useful for implementation of a parallelized version of SPARK because it prevents synchronization problems. In many cases no changes are required for a serial model to be able to run in the concurrent mode.

[0059] The Observer also schedules agent actions. The scheduler works at the level of agent classes, so agents of the

same class are scheduled in the same way. The basic operation of the scheduler is to specify at which simulation steps agents of a particular class will act. For example, if some agents can act at each simulation step, while others only each third step, a convenient way to schedule actions is by using the notion of simulation time. In this case each simulation step has a time value associated with it; therefore each simulation step represents the smallest unit of time for the simulation. For each agent class, it is possible to define the time interval between actions of agents of that class. For instance, there could be agents who act after two time units and agents who act after one and half time units (fractional time is also possible). Then, a simulation will proceed as follows: nobody will act during the first simulation step, agents of the second type (one and half time units) will act during the second simulation step, agents of the first type will act during the third simulation step, and after them agents of the second type will also act (three time units passed), etc. Besides time, agents are assigned priorities. If there are several agents acting at the same time point, priority tells the Observer which agent will act first. If no priority is specified explicitly, or if two agents have the same priority, then lexicographic order on agent class names determines the order of agents' actions.

[0060] The SPARK programming language (SPARK-PL) was developed to simplify the process of model creation by researchers not experienced in computer programming. Its syntax is derived primarily from the Logo programming language (the inspiration for NetLogo language) and the Java™ programming language. All models written in SPARK-PL are translated into Java™ source code first, and then a Java™ compiler is used to produce the byte code that can be executed by the SPARK simulation engine. Because of this two-step process, it is convenient to extend SPARK-PL with native Java™ constructions. As a result, SPARK-PL harnesses the full power of the Java™ language for ABM development. The main features of this language include object oriented constructions, a static type system with type inference, close ties with SPARK, as well as the concise and expressive syntax of the Logo language.

[0061] SPARK-PL includes several object-oriented concepts such as classes and inheritance similar to mainstream general purpose programming languages (such as C++, Java™, C#). These hierarchical properties allow for greater ease of organization and scheduling code blocks, which facilitates the development of complex, large-scale models. The source code of each model in SPARK-PL can be separated into several files, and can also be easily organized in a hierarchical format.

[0062] SPARK-PL static type system eliminates many programmers' errors concerning an incorrect use of variables of incompatible type. Type inference feature of the SPARK Language makes it possible to implement models in a way similar to Logo-like languages, such as NetLogo, in which dynamic type system is used. With type inference, it is not required to provide an explicit type for every new variable. The translator is capable of finding the correct type of many variables automatically by looking at the expressions. Another advantage of static type system is that it leads to efficient and fast code, because it is not required to check type consistency in runtime (Solovyev, A, et al. SPARK: A Framework for Multi-Scale Agent-Based Biomedical Modeling International Journal of Agent Technologies and Systems, 2(3), 18-30, Jul.-Sep. 2010).

[0063] SPARK is implemented in the Java™ programming language. SPARK code can run on all machines with Java™ Standard Edition runtime environment version 1.5 or 1.6. The SPARK source code is freely available and can be retrieved from the SPARK repository (<http://code.google.com/p/sparkabm/>). The compiled SPARK packages, along with the tutorials, can be downloaded from the official SPARK website (www.pitt.edu/~cirm/spark). There are several third-party libraries used in SPARK: 1) JFreeChart (<http://www.jfree.org/jfreechart/>); 2) Java™ OpenGL (JOGL, <https://jogl.dev.java.net/>); and 3) Colt (<http://acs.lbl.gov/~hoschek/colt/>). JFreeChart is used for creating and visualizing line plots and histograms in real time. JOGL is used for visualization of a simulation. Colt is a library for high performance scientific and technical computing in Java™.

[0064] According to one embodiment of the methods described herein, a system and a computer-implemented method of modeling progression of hepatic inflammation, fibrosis and cancer in a patient are provided. Generally-speaking, the method comprises first defining an agent-based model (ABM) in accordance with obtained data, the ABM adapted to simulate hepatic tissue, the ABM comprising agent data, global data, initialization process data, or any combination thereof. The agent data, global data, initialization process data, or any combination thereof comprises representations of biological elements of liver inflammation, fibrosis and cancer, for example chosen from hepatocytes, macrophages (e.g., Kupffer cells), stellate cells, and cancer cells and measurable factors relating to liver inflammation, fibrosis and cancer chosen from TNF- α , TGF- β 1, and HMGB1. The defining step includes entering agent data, global data, initialization process data, or any combination thereof into the ABM that acts as an initial state of the simulation. Next, the method comprises simulating the ABM. The simulation may be run a fixed number of cycles to represent a passage of a defined time period. Lastly, the method comprises generating output data, representing simulated hepatic conditions at one or more time points, such as hepatic tissue structure, hepatic inflammation state, inflammation-induced hepatic damage, hepatic viral load, virally-induced hepatic damage, hepatic tumor presence, hepatic tumor size, hepatic cancer presence, hepatic metastasis size, hepatic metastasis extent, hepatic fibrosis state, or any combination thereof. The output data for any one or more time point may be stored on computer readable media. The simulating step may be run multiple (two or more) times to generate a stochastic range of multiple different simulation outputs, given the stochastic nature of the model. Initial values for the agent data, global data, initialization process data, or any combination thereof may be altered, or additional values added to match outcomes obtained from a patient, in order to predict an outcome for another patient for whom only partial knowledge of certain parameters or dynamics (i.e. time-dependent changes in relevant, measurable variables) are known. Likewise, initial values for the agent data, global data, initialization process data, or any combination thereof may be altered, or additional values added in order to simulate the impact of an active agent (e.g., a drug or other physiologically-active substance).

[0065] The model may comprise a therapeutic or diagnostic agent as agent data, global data, initialization process data, or any combination thereof in order to simulate the action of that agent in the simulation. By accurately simu-

lating the impact of such an agent on a biological system, much of the cost, time, labor and resources, such as laboratory animals involved with early-stage research can be avoided by the *in silico* modeling. Thus, the model can simulate the impact of a therapeutic strategy for a disease or condition involving the interrelations among hepatic inflammation, fibrosis/cirrhosis, and cancer. In another non-limiting embodiment, the model is used to rationally design a drug, device, diagnostic, prophylaxis, or therapeutic strategy for a disease or condition involving the interrelations among hepatic inflammation, fibrosis and cancer. In yet another embodiment, the method is used in the diagnosis of a disease involving the interrelations among hepatic inflammation, fibrosis/cirrhosis, and cancer.

[0066] An ideal therapy for inflammation-based cancer should suppress excessive anti-inflammatory responses (which promote cancer growth) while enhancing the production of cytokines and other factors that promote pro-inflammatory responses (which promote cancer apoptosis and death). This therapy would not over-induce the pro-inflammatory response, so that bystander tissue damage could be minimized.

[0067] According to one embodiment of the methods and systems described herein, an agent-based computer simulation of the progression of hepatocellular carcinoma (HCC) is provided, including and dependent on pro- and anti-inflammatory responses present in chronic infections such as Hepatitis C virus (HCV). These simulations were calibrated on published literature data regarding the dynamics of cell populations and cytokines and their influence on inflammation and the early stages of cancer caused by chronic infection. The ABM could reproduce acute inflammation that leads to a normal healing response as well as HCC following a chronic inflammatory response, including features such as a hypoxic core and tumor angiogenesis.

[0068] One use of the ABM leverages the fact that this type of simulation can often reproduce a visual pattern of cells and tissue architecture, which in turn can be compared to a photomicrograph of a biopsy specimen from a patient or an experimental animal. Moreover, since the ABM recapitulates a biological process, additional information such as distribution of given molecules (e.g. inflammatory cytokines, pro-fibrotic molecules, or cancer-related molecules) can be depicted. These simulated patterns are comparable to those patterns seen upon immunostaining of tissue sections for such molecules. The patterns observed in the ABM simulations are derived from simulated biological processes, and thus calibrating the ABM for the pattern observed in actual photomicrographs or immunostained tissue sections can yield a quantitative prediction (in both time and levels of cells/mediators) that essentially leads to a patient-specific, personalized simulation of the inflammation, fibrosis and/or cancer process. In addition, this mechanistic foundation of the simulation means that simulated therapies can be run *in silico* on a patient-specific basis. Thus, this platform allows for a “smart diagnostic”, which not only can lead to better prediction of the course of disease, but that can also be used to suggest specific therapeutic options.

[0069] Because these simulations include a central role for molecular surrogates of tissue damage/dysfunction, any simulated therapy can be simultaneously evaluated both for efficacy (killing tumor cells) and for safety (not inducing higher levels of damage). For example, the ABM can simulate the behavior of both standard liver macrophages

(Kupffer cells) as well as tumor-infiltrating macrophages (TAM, which are known to be subverted by the tumor environment in a manner that ultimately promotes tumor growth and metastasis; A. Yuan, J. J. Chen, and P. C. Yang. Pathophysiology of tumor-associated macrophages. *Adv. Clin. Chem.* 45:199-223, 2008). The ABM has demonstrated the importance of TAM in the process of inflammation and early cancer and suggested that drugs targeting TAMs may be a promising therapeutic strategy (in support of prior studies; I. J. Fidler. Macrophages and metastasis—A biological approach to cancer therapy: Presidential address. *Cancer Res.* 45:4714-4726, 1985).

[0070] The biology encompassed in this ABM spans the range from acute HCV infection through chronic HCV infection to cancer, but can also simulate inflammation induced by non-infectious means, leading to liver fibrosis and cirrhosis. Thus, novel therapies can be explored in these settings, either on a patient-specific basis or as an *in silico* clinical trial. As an example, of such a therapy, the ABM is used to reproduce the anti-TNF- α antibody treatment known to be therapeutically relevant in a mouse model of ethanol-induced hepatic necrosis and inflammation (see, e.g., Jimuro, Y., et al., Antibodies to tumor necrosis factor α attenuate hepatic necrosis and inflammation caused by chronic exposure to ethanol in the rat. *Hepatology*, 1997. 26(6): p. 1530-7).

[0071] Another use of this ABM lies in its ability to predict novel therapeutic regimens directed at preventing the formation of cancer or treating an early-stage cancer. Simulations can be run in a state in which pre-existing cancer is assumed. In this simulation, anti-inflammatory responses precede a large decrease in pro-inflammatory responses, which drives damage/dysfunction and leads to death of the patient. Death in this model occurs not from the cancer itself, but from damage related to the dramatic increase in the inflammatory response.

[0072] This is a realistic clinical outcome. The time until patient death is inversely related to the initial cancer load. In addition, higher damage rates result in a faster progression to death along with smaller tumor size at the time of death. Thus, this model may be of use in determining the responses of patients who have had a tumor for several years.

[0073] The *in silico* methods would therefore be useful both for diagnosing the state of HCC (in combination with digital pathology specimens), for elucidating mechanisms of HCC treatment, for testing the effects of known therapies in the setting of HCC, and for developing novel therapies for HCC.

[0074] The modeling systems described herein comprise any number of: agent data, global data, initialization process data, or any combination thereof, within the program, representing cells, structures, proteins, chemicals, etc. In the model for hepatic inflammation, fibrosis and cancer (e.g., HCC), a computational simulation of the key inflammatory events triggered by hepatitis C viral infection was developed. The model probabilistically leads to a chronic condition, subsequently resulting in smoldering necrosis and fibrosis, an environment fertile for the progression of HCC from a few initial progenitor cells (transformed hepatocytes or cancer stem cells).

[0075] In the model, virus infects healthy hepatocytes; this injury activates resident Kupffer cells to produce pro-inflammatory cytokines such as TNF- α , and anti-inflammatory cytokines such as TGF- β 1. TNF- α causes acceleration in the

death of injured hepatocytes, but also has a bystander effect on local healthy hepatocytes. TNF- α also activates stellate cells, which transform into fibrosis promoting myofibroblasts, which further proliferate and increase collagen formation in response to TGF- β 1.

[0076] One element of the model is a class of molecules known as alarmins or damage-associated molecular pattern (DAMP) molecules. Commonly found in the setting of runaway inflammation induced cell death, DAMPs can cause inflammatory responses to persist, even when the original pro-inflammatory stimulus has been cleared subsequent to the actions of TH1 cytokines. One such DAMP, HMGB1 (also, HMG-1), particularly in conjunction with its receptor for advanced glycation endproducts (RAGE), has been associated with the proliferation and metastasis of many tumor types. RAGE expression is elevated in hepatitis, and even more so in HCC. As such, HMGB1 is included in the hepatic inflammation and HCC models. HMGB1 protein is an abundant component of all mammalian nuclei, and related proteins exist in all eukaryotes. HMGB1 binds with high affinity to specific DNA structures such as bent or kinked DNA. It is considered to be a structural protein of chromatin. Activated macrophages and monocytes secrete HMGB1 as a cytokine mediator of inflammation. Antibodies that neutralize HMGB1 confer protection against damage and tissue injury during arthritis, colitis, ischemia, sepsis, endotoxemia, and systemic lupus erythematosus. The mechanism of inflammation and damage is binding to TLR4, which mediates HMGB1-dependent activation of macrophage cytokine release. This positions HMGB1 at the intersection of sterile and infectious inflammatory responses (see, e.g., Lotze, M. T., Tracey, K. J. High-mobility group box 1 protein (HMGB1): nuclear weapon in the immune arsenal. *Nature Rev. Immun.* 5: 331-342, 2005).

[0077] In terms of the liver structure, all or part of which can be included as elements of an ABM, sheets of connective tissue divide the liver into thousands of small units called lobules. The hepatic lobule is the structural unit of the liver. It consists of a roughly hexagonal arrangement of plates of hepatocytes radiating outward from a central vein in the center. At the vertices of the lobule are regularly distributed portal triads, containing a bile duct and a terminal branch of the hepatic artery and portal vein. Another basic unit of liver structure is the hepatic acinus, which represents a unit that is oriented around the afferent vascular system.

[0078] The parenchymal cells of the liver are hepatocytes. Hepatocytes are the chief functional cells of the liver. Roughly 80% of the mass of the liver is contributed by hepatocytes. These polygonal cells are joined to one another in anastomosing plates, with borders that face either the sinusoids or adjacent hepatocytes. The ultrastructure appearance of hepatocytes reflects their substantial metabolic role, with abundant rough and smooth endoplasmic reticulum, and Golgi membranes. Glycogen granules and vesicles containing very low density lipoproteins are readily observed. Hepatocytes contact blood in sinusoids, which are distensible vascular channels lined with highly fenestrated endothelial cells and populated with phagocytic Kupffer cells. The space between endothelium and hepatocytes is called the Space of Disse which collects lymph for delivery to lymphatic capillaries.

[0079] Kupffer cells, also known as Browicz-Kupffer cells and stellate macrophages, are specialized macrophages located in the liver lining the walls of the sinusoids that form

part of the reticuloendothelial system (RES) (aka: mononuclear phagocyte system). Kupffer cells activation are responsible for early ethanol-induced liver injury, common in chronic alcoholics. Chronic alcoholism and liver injury deal with a two hit system. The second hit is characterized by an activation of the Toll-like receptor 4 (TLR4) and CD14, receptors on the Kupffer cell that internalize endotoxin (lipopolysaccharide or LPS). This activates the transcription of pro-inflammatory cytokines (Tumor necrosis factor-alpha or TNF- α) and production of reactive oxygen species such as hydroxyl radical, superoxide, and hydrogen peroxide (a pro-oxidant). TNF- α will then enter the stellate cell in the liver, leading to collagen synthesis and fibrosis. Fibrosis will eventually cause cirrhosis, or loss of function of the liver.

[0080] Hepatic stellate cells (stellate cells), also known as perisinusoidal cells or Ito cells (earlier lipocytes or fat-storing cells), are pericytes found in the perisinusoidal space (a small area between the sinusoids and hepatocytes) of the liver also known as the space of Disse. The stellate cell is the primary cell type involved in liver fibrosis. Stellate cells can be selectively stained with gold chloride, but their distinguishing feature in routine histological preparations is the presence of multiple lipid droplets in their cytoplasm. Reelin also is a marker in discerning them from other myofibroblasts, which is increasingly expressed after liver injury. In normal liver, stellate cells are quiescent and represent 5-8% of the total number of liver cells. When the liver is damaged, stellate cells are activated and are characterized by proliferation, contractility, and chemotaxis. Activated stellate cells also are responsible for secreting collagen scar tissue, which can lead to cirrhosis. Cirrhosis results from chronic liver disease and is characterized by fibrosis, formation of scar tissue and regenerative nodules, and is accompanied by loss of liver function. Cirrhosis is most commonly caused by alcoholism, hepatitis B and C, and fatty liver disease, but has many other possible causes and in some cases is idiopathic.

[0081] Tumor necrosis factor (TNF- α , formerly known as cachexin or cachectin) is a multifunctional pro-inflammatory cytokine secreted predominantly by monocytes/macrophages that has effects on lipid metabolism, coagulation, insulin resistance, and endothelial function. The primary role of TNF- α is in the regulation of immune cells. TNF- α is able to induce apoptotic cell death, to induce inflammation, and to inhibit tumorigenesis and viral replication.

[0082] The emergent visual pattern of changes to the liver tissue when simulated over a course of time, for example as seen in FIG. 7, described below, provides an image to be compared to images of clinical pathology. As the ABM is stochastic, multiple runs of the same initial conditions would create diverse outcomes.

[0083] Thus, according to one embodiment of the systems and methods described herein relating to digital pathology, simulated images of liver tissue are generated that are grids representative of hepatic tissue. Digital pathology is a computer-implemented, image-based process that allow for the management of information generated from a digital slide, which is produced, for example, by scanning, analyzing and storing on computer readable medium digitized photomicrograph images, e.g., from a glass slide. Those images are compared to an image of a patient's liver tissue. Because the ABM is stochastic in nature, running the ABM multiple times results in multiple outcomes that generate representative data values, such as hepatic tissue structure, inflammation state, inflammation-induced damage, viral load,

virally-induced damage, presence of a tumor, size of a tumor, presence of cancer (e.g., metastasis), size or extent of metastasis, and state of fibrosis (e.g., cirrhosis). A database of images is produced and is stored in computer storage. By comparing an image of a liver biopsy of a patient with the image representations of the multiple simulation outputs in computer storage and determining a best match with the image of a liver biopsy of a patient, an output that includes a patient prognosis can be produced because each computer-generated image stored in the library is connected with a specific series of images and outcome data corresponding to each individual run of the simulation that produced the matching computer-generated image. Also, the simulation can be run using differing initial values, parameters, agents, etc. to generate additional data images for storage in the library on the computer storage. This can be of value because in many cases, liver inflammation, fibrosis and cancer is a long-term process, and earlier exposure to one of many factors that can alter the simulation, whether known or unknown, to a patient. Thus the model also comprises both 1) simulating the ABM multiple times such that multiple different simulation outputs comprising a plurality of image representations of the grid for the plurality of time points are generated, and storing the image representations of the multiple simulation outputs on a computer readable medium (e.g., in a database) and 2) altering the agent data, global data, initialization process data, or any combination thereof from its original states and simulating the ABM multiple times such that multiple different simulation outputs comprising a plurality of image representations of the grid for the plurality of time points are generated, and storing the image representations of the multiple simulation outputs on a computer readable medium (e.g., in a database). In either case, the systems and methods comprise comparing an image of a liver biopsy of a patient with the image representations of the multiple simulation outputs the computer readable medium and determining a best match with the image of a liver biopsy of a patient; and producing an output that includes a patient prognosis based upon output data produced by the simulation that generated the best match image representation.

[0084] A large number of examples of image comparison software programs are available. Examples include: Image Comparer (Bolide Software, Russian Federation), Perceptual Image Diff, Imatch3 (Photools); and IRMA ((Image Retrieval in Medical Applications, Aachen University of Technology, Aachen, Germany, an example in which this is used in a clinical environment. See also Hipp et al., as an example of paper showing implementation of image comparison in the pathology setting (Spatially Invariant Vector Quantization: A pattern matching algorithm for multiple classes of image subject matter including pathology *J Pathol Inform* 2011, 2:13).

[0085] As described above, all of these methods can be simulated by modeling on a computer system. Thus, also provided is system for modeling progression of at least one hepatic condition. The system comprises on at least one computer with a computer readable medium having programming instructions stored thereon, which, when executed by a processor of the at least one computer, cause the processor to: a) receiving or defining at least one of the following: agent data, global data, initialization process data, or any combination thereof, the agent data comprising at least one of the following: definition data, behavior data,

class data, internal state data, function data, link data, space data, or any combination thereof; b) modeling the at least one hepatic condition associated with at least one agent based at least in part upon at least a portion of the agent data corresponding to the at least one agent; and c) generating modeling output data representative of at least one of the following: hepatic tissue structure, hepatic inflammation state, inflammation-induced hepatic damage, hepatic viral load, virally-induced hepatic damage, hepatic tumor presence, hepatic tumor size, hepatic cancer presence, hepatic metastasis size, hepatic metastasis extent, hepatic fibrosis state, or any combination thereof. The system may further comprise modeling the at least one hepatic condition multiple times to generate multiple different modeling output data, which can be stored on computer readable medium, e.g., in a database. The system may comprise output data from altering agent data, global data, initialization process data, or any combination thereof and modeling the at least one hepatic condition multiple times to generate multiple different modeling output data, which can be stored on computer readable medium, e.g., in a database. Examples of altering agent data, global data, initialization process data, or any combination thereof include, changing a value for one or more agents of the model, adding an additional agent to the model, etc., representing such biological elements such as viral load, type of virus, blood levels of one or more proteins, damage to the liver, inflammatory state, tumor size, vascularization of a tumor, surgical removal of a tumor, the effects of a cytotoxic drug, such as the simulated death of both tumor cells and hepatocytes, the effects of an antiviral drug, such as viral killing combined with increased inflammation, the effects of an anti-inflammatory drug, such as reduction in inflammation and inflammatory damage, immunosuppression and increased viral growth, the effects of an anti-fibrotic drug, such as a reduction in fibrosis, increased inflammatory damage and suppression of tissue healing, etc.

[0086] In one embodiment, the output data comprises a simulation of liver tissue on a grid for a plurality of time points and in one further embodiment comprises storing images representative of the grids on computer readable medium, e.g., in a database. As indicated above, a patient's prognosis or an outcome based on a therapeutic regimen may be determined by modeling the at least one hepatic condition multiple times using the same or different agent data, global data, initialization process data, or any combination thereof, to generate multiple different modeling output data comprising a plurality of image representations of the grid for the plurality of time points, which are stored on a computer readable medium. An image of a tissue sample from a liver biopsy can then be compared to the stored image representations of the grid in order to identify a best match—that is, an image in the database of stored images that best matches that of the image of the tissue sample. Because the stored image representations of the grid are representations taken from different time points in specific individual runs of the modeling step, a patient's prognosis can be determined and output by looking at output data for later time points in the particular run of the modeling step.

[0087] A method of modeling progression of at least one hepatic condition also is provided. The method is performed on any system as described above and comprises, in the at least one computer, receiving or defining at least one of the agent data, global data, initialization process data, or any combination thereof; modeling the at least one hepatic

condition; and generating output data from the modeling step representative of at least one of the following: hepatic tissue structure, hepatic inflammation state, inflammation-induced hepatic damage, hepatic viral load, virally-induced hepatic damage, hepatic tumor presence, hepatic tumor size, hepatic cancer presence, hepatic metastasis size, hepatic metastasis extent, hepatic fibrosis state, or any combination thereof. According to one embodiment, the method is performed on at least one computer with a computer readable medium having programming instructions stored thereon, which, when executed by a processor of the at least one computer, cause the processor to: a) receive or define at least one of the following: agent data, global data, initialization process data, or any combination thereof, the agent data comprising at least one of the following: definition data, behavior data, class data, internal state data, function data, link data, space data, or any combination thereof; b) model the at least one hepatic condition associated with at least one agent based at least in part upon at least a portion of the agent data corresponding to the at least one agent; and c) generate modeling output data representative of at least one of the following: hepatic tissue structure, hepatic inflammation state, inflammation-induced hepatic damage, hepatic viral load, virally-induced hepatic damage, hepatic tumor presence, hepatic tumor size, hepatic cancer presence, hepatic metastasis size, hepatic metastasis extent, hepatic fibrosis state, or any combination thereof.

EXAMPLE 1

[0088] The following are the rules created for the agent data, global data, initialization process data, or any combination thereof (agents, layers, etc. depicted in FIG. 4).

[0089] Time scale. It was assumed in the model that one unit of simulated time represents 0.21 days. This assumption results in the rise and fall of virally-driven damage levels, from initiation to resolution, in approximately 6 weeks. The dynamics of cell behavior and cytokine production and decay were calibrated proportionately towards this time course.

[0090] Initialization. The model begins with 200×200 cell grid representing a small portion of liver tissue. The cell types present in this liver tissue at baseline are hepatocytes (the main structural and functional cell of the liver), Kupffer cells (the resident macrophages [inflammatory cells] of the liver), stellate cells (cells that can drive the healing of the injury and are key players in fibrosis), cancer cells (which start as clonal progenitors or “cancer stem cells”). Portal triads (afferent liver blood vessels) are present in a regular grid among the hepatocytes, forming the vertices of hexagonal lobules grouping the hepatocytes. These portal triads form the baseline circulatory system providing nutrients to healthy liver tissue, and also provide the entry point to the liver for the blood-borne virus. The portal triads are connected by a thin layer of connective tissue comprising the border of each lobule, and a data layer with constant diffusion gradient implements zonal differentiation (Zones 1-3) for hepatocytes based on their distance from border to the center of a lobule, with a graded decrease in oxygenation levels towards the center of the lobule. The cell types and interactions are shown in Table 1, along with the initial values relevant to this particular example. When infected with virus, the initialization value is the number of viruses randomly dispersed in the simulated liver tissue slice. Only the virus whose location coincides with a portal triad sur-

vives (dispersal function), thus implementing the blood vessel into the liver as the site of infection, as well as stochastically setting the number of initial virus that arrive at the liver alive. Lifespans of the various cells and viruses are set to a baseline constant plus a random number within a specified range. Cytokines, represented as data layers, are the pro-inflammatory TNF- α (TNF), anti-inflammatory TGF- β 1 (TGF) and the damage-associated molecular pattern molecule/danger signal High Mobility Group Box 1 (HMGB1). Each of these are initialized to 0, are only produced by agents subsequent to certain activity, and are subject to steady diffusion (fraction of the value in this grid location divided between neighbors) and degradation (fraction of the value in this grid retained) rates. Another data layer, Oxygen, is representative of oxygenation from the portal triads permeating surrounding tissue. A global parameter for Injury is also tracked.

TABLE 1

| Initialization of the Model | |
|---|---|
| Number of Cells created | Initialized to |
| Resident macrophages (Kupffer cells) | 50 |
| Resident stellate cells | 50 |
| Viruses | 500 * 20, subject to dispersal function |
| Cancer cells | 4 |
| Lifespan | |
| Hepatocyte lifespan | 100 + x, where 0 < x < 500 |
| Virus lifespan | 5 + x, where 0 < x < 15 |
| Macrophage lifespan (begins after activation) | 50 + x, where 0 < x < 50 |
| Stellate lifespan (begins after activation) | 5 + x, where 0 < x < 20 |
| Cancer cell lifespan | 1000 + x, where 0 < x < 50 |
| Rates | |
| Virus proliferation rate | 15 |
| Cancer propagation probability | 0.02 |
| Macrophage activation probability | 1 |
| Stellate activation probability | 0.04 |
| Thresholds | |
| Bystander TNF threshold | 4 |
| Tumor TNF threshold | 4 |
| Virus TNF threshold | 4 |
| Data Layers | |
| TNF (TNF- α) | 0 |
| TGF (TGF- β 1) | 0 |
| HMGB1 | 0 |
| Oxygen | 3 per Portal triad position |
| Diffusion Rates | |
| TNF | 1 |
| TGF | 1 |
| HMGB1 | 1 |
| Oxygen | 1 |
| Degradation Rates | |
| TNF | 0.92 |
| TGF | 0.95 |
| HMGB1 | 0.9 |
| Oxygen | 0.996 |

[0091] Rules governing the model are shown in Table 2. At each time step, an agent takes action based on specified rules of behavior.

TABLE 2

| Rules governing the Model | |
|----------------------------------|--|
| Agents | Rules |
| Hepatocyte | Stationary, no motion. Check value of TGF at grid position. If TGF > 5, With 50% probability: Check 1 of its 8 neighboring spaces to see if there is free space, and create a new hepatocyte there Else With 20% probability: Check 1 of its 8 neighboring spaces to see if there is free space, and create a new hepatocyte there Check value of TNF at grid position. If TNF > Bystander TNF threshold, With 50% probability: Convert Hepatocyte to Dead cell Raise HMGB1 at this grid position by 1 unit |
| Infected (Subtype of Hepatocyte) | If Infected hepatocyte, Raise HMGB1 at this grid position by 1 unit Check value of TNF at grid position. If TNF > Virus TNF threshold, With 50% probability: Reduce lifespan by an additional unit Release virus (replication within infected hepatocyte) periodically (every 2 nd step) with 1% probability number = random within the range [virus-proliferation rate \pm 2] If reach end of lifespan, Convert Infected hepatocyte to Dead cell Raise HMGB1 at this grid position by 1 unit Reduce Injury at this grid position by 1 unit |
| Dead | If Dead cell, Raise HMGB1 at this grid position by 0.5 unit |
| Virus | Motion: Stochastic walk If Hepatocyte present at this grid position, Convert it to Infected hepatocytes Raise Injury at this grid position by 1 unit |
| Macrophage | Always activated Motion: Stochastic walk; chemoattracted by HMGB1 Baseline replenishment of resident macrophages (Kupffer cells): Created periodically (every step) with 50% probability number = 1 or 2 Injury-driven recruitment of circulating macrophages: Created periodically (every 5 th step), with 70% probability: number = total injury * macrophage activation probability If come in contact with a Stellate cell, activate it If come in contact with a Dead cell, phagocytose it If come in contact with an Infected hepatocyte, phagocytose it with 20% probability and reduce Injury at this grid position by 1 unit Check value of HMGB1, TNF and TGF at grid position: If HMGB1 > 0.01 [TNF += 1 / (TGF + 0.01) TGF += 0.238] |
| Stellate | If activated: Motion: Stochastic walk Check value of HMGB1 and TGF at grid position: If TGF > 1, increase its lifespan by 1 If HMGB1 > 1, create Fibrosis |
| Portal Triad | Raise Oxygen at this grid position by 3 units If Cancer Cell present at this grid position, then remove Portal Triad |
| Cancer Cell | Stationary, no motion. Raise HMGB1 at this grid position by 2 units Raise TGF at this grid position by 0.2 units With 10% probability: Reduce lifespan by 1 unit If reach end of lifespan, Convert Cancer Cell to Dead cell Raise HMGB1 at this grid position by 3 units On every 4 th step, Check value of TNF at grid position. |

TABLE 2-continued

| Rules governing the Model | |
|---------------------------|---|
| Agents | Rules |
| | If $TNF > \text{Tumor TNF threshold}$, With 25% probability: Convert Cancer Cell to Dead cell Raise HMGB1 at this grid position by 2 units Check value of TGF at grid position. If $TGF > 2$, multiply propagation probability by 4. Propagation: With 50% chance of propagation probability: Check 1 of its 8 neighboring spaces to see if there no Cancer Cell in that space, and create a new Cancer Cell there. Raise HMGB1 at this grid position by 1 unit On every 8 th step, With 1% probability, if Oxygen at present grid position < 1 , Create Platelet (Angiogenesis at the hypoxic core of the tumor) |

[0092] The initial stage of the model begins with a 2-D view of a small patch (a few square mm) of liver tissue, about 40,000 simulated hepatocytes. Hepatocytes are arranged in hexagonal lobules, with portal triads present at the vertices, forming the baseline circulatory system providing nutrients to healthy liver tissue. They also serve as the entry point for the bloodborne virus. The portal triads are connected by a thin layer of connective tissue comprising the border of each lobule, and a data layer with constant diffusion gradient implements zonal differentiation (Zones 1-3) for hepatocytes based on their distance from border to the center of a lobule; moving from Zone 1 (periportal) to Zone 3 goes from higher to lower oxygen uptake and regenerative capacity.

[0093] Events take place in the model as an aggregate of rule-based behavior by each individual agent. Bloodborne virus arrives at the tissue via the portal triads. The number of viruses is a random function of user-provided inoculation. Infected hepatocytes act as a site of virus replication, and as they die are detected as tissue injury and damage. These factors drive activation of local Kupffer cells, which lead to further recruitment of activated macrophages and release of pro-inflammatory cytokines (such as $TNF-\alpha$) and anti-inflammatory cytokines (such as $TGF-\beta 1$). They also lead to activation of stellate cells, which then differentiate into myofibroblasts that ultimately drive local fibrosis. Probabilistic rules governing the biological interactions allow for stochastic outcomes that can be analyzed and compared to real clinical outcomes.

EXAMPLE 2

[0094] Our model was built using an agent-based approach, which is particularly well suited to representing transition between mechanisms at a local scale of organization, to global emergent behavior. We simulate rule-based interactions between parenchymal cells, inflammatory cells and cytokines that participate in the response to HCV infection, and may drive the initiation and progression of HCC.

[0095] One example of such emergent behavior is the course of liver damage after viral infection (FIG. 5).

[0096] As the simulation progresses, SPARK provides a visual representation of changes to the area of the liver under observation, and simultaneous measurements of the number

of active agents (macrophages, viruses, hepatocytes, etc.) and levels of data layers (cytokines, oxygen or other diffusible factors). An example of tracking the realtime measurements is shown in FIG. 5, Right panel. An example of the visual representation is shown in FIG. 6.

EXAMPLE 3

Cirrhosis Model

[0097] Cirrhosis is a common outcome of chronic hepatitis C, and 70-90% of cases of hepatocellular carcinomas are observed in cirrhotic livers. We therefore have also been exploring the emergence of cirrhosis in this liver ABM. Myofibroblasts are chemoattracted by $TGF-\beta$, and proliferate and produce collagen in response to $TGF-\beta$. By calibrating for the influence of zonal differences on collagen production, the course of pattern formation in the development of cirrhosis—periportal fibrosis, followed by bridging fibrosis, eventually disrupting the regular lobular structure with regenerative nodules—is reproduced. (FIG. 7).

[0098] The basic structure of the model is the same as in Example 1. The differences in this example are: 1) Addition of new agents: Myofibroblast, Septa, and Collagen. These develop more structural detail for the zonal behavior relevant to the lobule. Since their creation and actions are downstream of the action of agents in Example 1, the initialization of all agents for this Example remains the same as in Example 1; 2) Calibration: Oxygen originating out of Portal triads is calibrated to have a steady gradient implementing zonal differentiation centered around the periportal area; and 3) No cancer cells were seeded, therefore there was no tumor formation in this instantiation of the model.

TABLE 3

| Changes in initialization of the Model for Example 3 | |
|--|-------------------------------|
| Initialized to | |
| Data Layers | |
| Oxygen Diffusion Rates | 0.2 per Portal triad position |
| Oxygen Degradation Rates | 0.5 |
| Oxygen | 0.95 |

TABLE 4

| Changes in rules governing the Model for Example 3 | |
|--|---|
| Agents | Rules |
| Hepatocyte | Stationary, no motion. Check number of Collagen units present at grid position. If $Collagen > 5$ Create Fibrosis Check value of HMGB1 at this grid position. If $HMGB1 > 1$ Raise TGF at this grid position by 0.2 |
| Dead | If Dead cell, Raise HMGB1 at this grid position by 0.1 unit |

TABLE 4-continued

| Changes in rules governing the Model for Example 3 | |
|--|--|
| Agents | Rules |
| Macrophage | Check value of Injury, and value of HMGB1 at grid position. Activated if $HMGB1 > 0.05$ or $Injury > 0.5$ If activated: Motion: Stochastic walk; chemoattracted by HMGB1 Inactivated if $Injury = 0$ If come in contact with a Stellate cell, check value of TNF at this grid position If $TNF > 1$ With probability = stellate activation probability, Transform Stellate to Myofibroblast If activated, transform to Myofibroblast (accomplished by Macrophage rule) |
| Stellate | Unaffected by HMGB1 (unlike in Example 1) Raise Oxygen at this grid position by 0.2 units Raise Oxygen at this grid position by 0.1 units If Cancer Cell or Fibrosis present at this grid position, then remove Septa |
| Portal Triad Septa | |
| Myofibroblast | Always activated If activated: Motion: Stochastic walk; chemoattracted by TGF Check value of Oxygen and TGF at this grid position. If $Oxygen > x$, where $x = \text{random within the range } [0 \text{ to } 2]$ If $TGF > 5$ Deposit 1 unit of Collagen With 5% probability, create a new Myofibroblast |
| Collagen | If Cancer Cell or Fibrosis present at this grid position, then remove Septa |

[0099] Having described this invention above, it will be understood to those of ordinary skill in the art that the same can be performed within a wide and equivalent range of conditions, formulations and other parameters without affecting the scope of the invention or any embodiment thereof.

1-23. (canceled)

24. A computer-implemented method of modeling progression of at least one hepatic condition in a patient, comprising:

- (i) receiving, with at least one processor, data relating to at least one first parameter of a plurality of parameters associated with a hepatic condition in a patient;
- (ii) identifying, with at least one processor, at least one first rule from a rules database, the at least one first rule selected based at least in part based on the data relating to the at least one first parameter;
- (iii) generating, with at least one processor, a simulated progression of the hepatic condition by applying the at least one first rule to the data relating to the at least one first parameter, thereby generating a progression of the hepatic condition based on the at least one first rule;
- (iv) generating, using at least one processor and based on the simulated progression, image-based output providing predicted hepatic tissue structure, and storing the image-based output in a database;
- (v) repeating steps (i)-(iv) to generate a stochastic range of image-based output of the simulated progressions;
- (vi) comparing, using at least one processor, the stochastic range of image-based output of the simulated progressions with one or more liver biopsy images; and
- (vii) associating, using at least one processor, image-based output corresponding to a best match liver biopsy image and storing the associated data in the database.

25. The computer-implemented method of claim **24**, further comprising:

- (viii), associating, using at least one processor, the associated data, and diagnostic information relating to the patient; and
- (ix) storing, using at least one processor, the image-based output corresponding to the best match liver biopsy image and the diagnostic information relating to the patient.

26. The computer-implemented method of claim **24**, wherein the data comprises an image of hepatic tissue and data relating to one or more biologic factors related to the hepatic condition.

27. The computer-implemented method of claim **26**, wherein the step of generating a simulated progression of the hepatic condition comprises a step of modifying the image of hepatic tissue.

28. The computer-implemented method of claim **24**, wherein the one or more first rules comprises thresholds relating to levels of at least one first parameter.

29. The computer-implemented method of claim **24**, wherein the image-based output further comprises data relating to predicted hepatic inflammation state, inflammation-induced hepatic damage, hepatic viral load, virally-induced hepatic damage, hepatic tumor presence, hepatic tumor size, hepatic cancer presence, hepatic metastasis size, hepatic metastasis extent, hepatic fibrosis state, or any combination thereof.

30. The computer-implemented method of claim **24**, wherein the data and rule are associated with an active agent, and wherein the image-based output comprises a representation of the effect of the active agent on hepatic inflammation, fibrosis and/or cancer.

31. The computer-implemented method of claim **24**, in which the data comprises data representing hepatocytes, macrophages, stellate cells, cancer cells, $TNF-\alpha$, $TGF-\beta 1$, and/or HMGB1.

32. The computer-implemented method of claim **24**, in which the data comprises data representing one or both of a hepatic lobule and a portal triad.

33. The computer-implemented method of claim **24**, in which the data comprises data representing one or more of septa, myofibroblasts, and collagen.

34. The computer-implemented method of claim **24**, in which the data comprises data representing hepatic septa, myofibroblasts, and collagen.

35. The computer-implemented method of claim **24**, wherein the data comprises data obtained from a patient.

36. The computer-implemented method of claim **24**, wherein the data comprises data representing an effect of an active agent on the at least one parameter.

37. The computer-implemented method of claim **24**, wherein the data and rule are associated with simulating surgical removal of a simulated tumor with attendant tissue damage that stimulates further inflammation.

38. The computer-implemented method of claim **24**, wherein the data and rule are associated with simulating a chemotherapeutic cytotoxic drug, such that simulated death of both tumor cells and hepatocytes occurs.

39. The computer-implemented method of claim **24**, wherein the data and rule are associated with simulating an antiviral drug, such that both viral killing and inflammatory damage to the simulated liver tissue are simulated.

40. The computer-implemented method of claim **24**, wherein the data and rule are associated with simulating an anti-inflammatory drug, such that reduction in inflammation and subsequent inflammatory damage to simulated liver tissue, immunosuppression, and virus growth-stimulating effects, are simulated.

41. The computer-implemented method of claim **24**, wherein the data and rule are associated with simulating an anti-fibrotic drug, such that both reduction in fibrosis and subsequent inflammatory damage to simulated liver tissue, as well as suppression of tissue healing, are simulated.

42. A system comprising:

a display;

a processor; and

memory having stored thereon programming instructions that when executed by the processor cause the processor to:

- (i) receive data relating to at least one first parameter of a plurality of parameters associated with a hepatic condition;

- (ii) identify at least one first rule from a rules database, the at least one first rule selected based at least in part based on the data relating to the at least one first parameter;

- (iii) generate a simulated progression of the hepatic condition by applying the at least one first rule to the data relating to the at least one first parameter, thereby generating a progression of the hepatic condition based on the at least one first rule;

- (iv) generate, based on the simulated progression, image-based output providing predicted hepatic tissue structure, and store the image-based output in a database;

- (v) repeat steps (i)-(iv) to generate a stochastic range of image-based output of the simulated progressions;

- (vi) compare the stochastic range of image-based output of the simulated progressions with one or more liver biopsy images; and

- (vii) associate the image-based output, liver biopsy image, and actual diagnostic data, and store the associated data in the database.

* * * * *