npj | Systems Biology and Applications

www.nature.com/npjsba

ARTICLE OPEN

The FLAME-accelerated signalling tool (FaST) for facile parallelisation of flexible agent-based models of cell signalling

Gavin Fullstone[1,2]✉, Cristiano Guttà[1], Amatus Beyer[1] and Markus Rehm[1,2]✉

Agent-based modelling is particularly adept at modelling complex features of cell signalling pathways, where heterogeneity, stochastic and spatial effects are important, thus increasing our understanding of decision processes in biology in such scenarios. However, agent-based modelling often is computationally prohibitive to implement. Parallel computing, either on central processing units (CPUs) or graphical processing units (GPUs), can provide a means to improve computational feasibility of agent-based applications but generally requires specialist coding knowledge and extensive optimisation. In this paper, we address these challenges through the development and implementation of the FLAME-accelerated signalling tool (FaST), a software that permits easy creation and parallelisation of agent-based models of cell signalling, on CPUs or GPUs. FaST incorporates validated new agent-based methods, for accurate modelling of reaction kinetics and, as proof of concept, successfully converted an ordinary differential equation (ODE) model of apoptosis execution into an agent-based model. We finally parallelised this model through FaST on CPUs and GPUs resulting in an increase in performance of 5.8× (16 CPUs) and 53.9×, respectively. The FaST takes advantage of the communicating X-machine approach used by FLAME and FLAME GPU to allow easy alteration or addition of functionality to parallel applications, but still includes inherent parallelisation optimisation. The FaST, therefore, represents a new and innovative tool to easily create and parallelise bespoke, robust, agent-based models of cell signalling.

npj Systems Biology and Applications (2020)6:10 ; https://doi.org/10.1038/s41540-020-0128-x

INTRODUCTION

Cellular signalling is essential in translating extrinsic and/or intrinsic chemical and physical stimuli into diverse cell responses such as proliferation, cell migration or cell death. The duration of stimuli, concentration of stimuli, concentration of signalling components, the reaction kinetics in signalling pathways and subcellular localisation of components can drastically affect downstream outcomes of cell signalling pathways. Moreover, cell-signalling pathways, typically, are highly complex with redundancy, cross-talk between different signals and numerous levels of regulation complicating our understanding of how cellular decisions are made. Systems biology approaches have increasingly been used to better understand and predict outcomes from cell signalling processes[1,2]. The most commonly used approach is ordinary differential equation (ODE) modelling that uses a series of differential equations to define how the concentrations of reactants change over time. This has been used effectively to describe a number of cell signalling pathways, including the NFκB pathway[3,4], the intrinsic apoptosis pathway[5–7] and the cell cycle[8]. However, biological systems are characterised by, complex structural organisation, a great level of heterogeneity and physical phenomena, such as molecular crowding, that are not adequately included in ODE models. Furthermore, stochastic effects in biological settings can have profound knock-on effects on cell signalling outcomes[9,10]. A variety of methods have been developed to better implement stochastic and spatial information in computational models with many methods being implementable by rule-based modelling platforms such as BioNetGen[11], Kappa[12] and PySB[13] for standardised, user-friendly deployment (summarised in Table 1). However, improved methods are required for deeper understanding of how complex events at the individual molecule scale underlie system level effects in cell signalling.

Agent-based modelling (ABM) is a type of bottom up systems modelling that has recently gained popularity in the study of cell signalling pathways and other biological processes[14]. ABM of cell signalling models behaviour of individual particles and their interactions. ABM is a powerful tool for modelling cell signalling as complex geometry is easily included and behaviour is naturally stochastic. However, ABM is computationally prohibitive, as the actions and interactions of potentially millions of individual signalling molecules over considerable periods of times must be considered. Furthermore, for ABM to be truly reflective of the modelled system it should be able to robustly model reaction kinetics. A number of ABM methods have been applied to cellular signalling previously, also giving rise to formal simulators such as Smoldyn, eGFRD, ChemCell and MCell[12,15–20]. These simulators offer highly robust and user-intuitive ABM of cell signalling pathways but still contain limits in scale-up of simulations, as well as, flexibility in their manipulation beyond the inbuilt functionality.

Parallel computing, the distribution of computational work across multiple central processor units (CPUs) or on graphical processing units (GPUs), offers the possibility to improve scale up of ABM simulations. FLAME (Flexible Large-scale Agent-based Modelling Environment) and FLAME GPU are generalised ABM platforms that are used to create ABM applications that can be easily parallelised on CPUs and GPUs, respectively. FLAME and FLAME GPU use a communicating X-machine approach to parallelisation, where the user declares discrete functions with input and output communication messages in an X-machine Markup Language (XMML) file and then the functions are declared in C. This allows FLAME and FLAME GPU to build the discrete functions into a parallel model therefore removing the necessity

[1]Institute for Cell Biology and Immunology, University of Stuttgart, Allmandring 31, 70569 Stuttgart, Germany. [2]Stuttgart Research Center Systems Biology (SRCSB), University of Stuttgart, Nobelstrasse 15, 70569 Stuttgart, Germany. ✉email: gavin.fullstone@izi.uni-stuttgart.de; markus.morrison@izi.uni-stuttgart.de

Published in partnership with the Systems Biology Institute

npj nature partner journals

Table 1. Simulation methods for stochastic and spatial modelling of signalling networks.

Simulation method	Software	Spatial handling	Temporal handling	Level
Gillespie	StochKit2[54]	Well mixed	Event-based	Population
Reaction-diffusion	URDME[55]	Lattice-based	Discrete or event-based	Population
	SIMMUNE[56]			
	Spatiocyte[57]			
Network free	NFsim[58]	Individual molecules	Event-based	Individual molecule
Particle simulations	Smoldyn[15,16]	Continuous	Discrete	Individual molecule
	MCell[21]			
	ChemCell[17]			
Event-based particle simulations	eGFRD[12]	Continuous	Event-based	Individual molecule
Interacting particle simulator	ReaDDY[43]	Continuous	Discrete	Individual molecule
	LAMMPS[44]			
	ESPResSo[12]			

of user knowledge of message passing interface (MPI) or CUDA coding, respectively[21,22]. Furthermore, they contain intrinsic parallelisation optimisation, even when including new functionality (for a short summary of FLAME's approach to parallelisation, see Supplementary Note 1, for a full technical report of FLAME's and FLAME GPU's approaches to parallelisation, see the reports in[21,22]).

In this paper we establish and validate new methods for accurate ABM of cell signalling. We implement these methods into the FaST (FLAME-accelerated Signalling Tool), which creates ABM models from reaction networks that can be easily customised and parallelised on CPUs or GPUs using FLAME and FLAME GPU. We then demonstrate that this tool can convert a previously established ODE model of apoptosis execution into an ABM simulation that reliably reproduces the kinetics of the original ODE model. Moreover, the performance of this simulation could be vastly improved by CPU parallelisation and GPU-acceleration.

RESULTS

Simulation of the random walk

In order to establish methods for ABM of cell signalling, we started by focusing on the movement of individual molecules within suspension. Particles in suspension undergo Brownian motion, a random walk caused by collisions with molecules of the solvent[23]. This can be simulated by implementing the polar form of the Box-Muller transformation of uniformly distributed random numbers into normally distributed random numbers[24,25]. These are then scaled to fit the Gaussian distribution for the change in x, y or z (Δx, Δy, Δz):

$$f(\Delta x) \equiv \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\Delta x^2}{2\sigma^2}} \quad (1)$$

where σ is calculated from the translational diffusion coefficient D_T in $m^2{\cdot}s^{-1}$ and the time step Δt in s:

$$\sigma = \sqrt{2D_T\Delta t} \quad (2)$$

as demonstrated previously[15,26]. Three particles, with diffusion coefficients of 1 $\mu m^2{\cdot}s^{-1}$, 5 $\mu m^2{\cdot}s^{-1}$, and 10 $\mu m^2{\cdot}s^{-1}$, reflective of diffusion coefficients of proteins in biological membranes and under molecular crowding[27–29], were simulated for 5 min and the 3D traces are shown in Fig. 1a. The implementation of Brownian motion was assessed using the mean squared displacement (MSD):

$$\mathrm{MSD} = \sum \frac{\Delta x^2 + \Delta y^2 + \Delta z^2}{t} \quad (3)$$

which is related to the diffusion coefficient, such that:

$$\mathrm{MSD} = \lambda D_T \quad (4)$$

where λ is a constant of dimensionality equal to 2, 4 or 6 for 1D, 2D and 3D, respectively. The observed MSD was compared to the expected MSD calculated with Eq. 4 in Fig. 1b. The observed MSD shows excellent agreement with the expected MSD over the 5 minute simulation with observed diffusion coefficients calculated from Eq. 4 of 1.00 $\mu m^2{\cdot}s^{-1}$, 4.99 $\mu m^2{\cdot}s^{-1}$ and 9.99 $\mu m^2{\cdot}s^{-1}$.

Simulation of first order reactions

First order reactions, such as degradation, dissociation and catalysis, form integral parts of cell signalling pathways. Therefore, we next set out to establish and validate methods to simulate first order reactions by ABM. First order reactions of the forms:

$$A \xrightarrow{k_{deg}} \text{Degraded}$$
$$AB \xrightarrow{k_r} A + B$$
$$ES \xrightarrow{k_{cat}} ES\prime$$

can be simulated by calculating a probability P that a single molecule will react within a single discrete time step of time Δt, equal to:

$$P = 1 - e^{-k\Delta t} \quad (5)$$

A simulated molecule will react if a randomly generated number is less than the probability calculated by Eq. 5 from its own k value, given in s^{-1}. The degradation of 100 nM of molecule A, with k_r values of $10^{-1}\,s^{-1}$, $10^{-2}\,s^{-1}$, $10^{-3}\,s^{-1}$, $10^{-4}\,s^{-1}$ and $10^{-5}\,s^{-1}$, was simulated by this method and compared to the expected kinetics from the ODE rate equation:

$$\frac{d[A]}{dt} = -k[A] \quad (6)$$

in Fig. 1c. Figure 1c shows excellent agreement with the expected kinetics generated from the ODE reaction with an almost exact overlay, thus demonstrating that ABM can effectively model first order reactions.

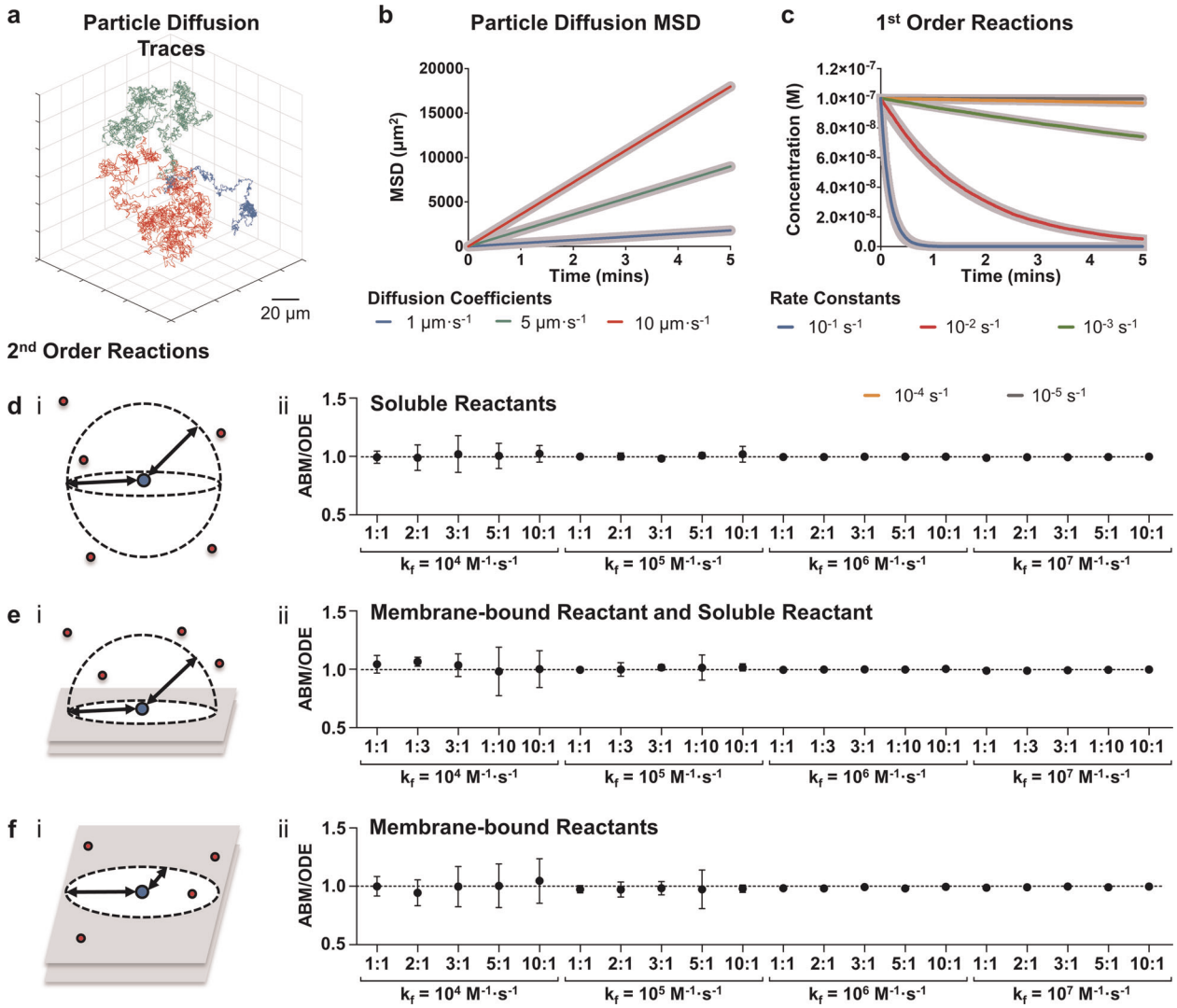

Fig. 1 Agent-based models are able to reproduce mass action kinetics. Particles with diffusion coefficients of 1 $\mu m^2 \cdot s^{-1}$ (blue), 5 $\mu m^2 \cdot s^{-1}$ (green) and 10 $\mu m^2 \cdot s^{-1}$ (red) were simulated (**a**) and the observed MSD was compared to the expected MSD (thick grey lines) (**b**). First order reactions with k values of 10^{-1} s^{-1}, 10^{-2} s^{-1}, 10^{-3} s^{-1}, 10^{-4} s^{-1} and 10^{-5} s^{-1} were simulated for 5 min with a time step of 0.05 s by ABM (thin lines) and compared to equivalent ODE models (thick grey lines) (**c**). The second order reactions of two soluble molecules (**d**), a soluble molecule with a membrane-bound molecule (**e**) and two membrane-bound molecules (**f**) were simulated with ABM and the ABM:ODE Score between the ABM simulations and equivalent ODE models were plotted for k_f values of 10^4 $M^{-1} \cdot s^{-1}$, 10^5 $M^{-1} \cdot s^{-1}$, 10^6 $M^{-1} \cdot s^{-1}$ and 10^7 $M^{-1} \cdot s^{-1}$ and different indicated concentration ratios of A to B ([A]:[B]). Dashed lines indicate the perfect agreement between ABM and ODE of 1. All simulations were for 5 min, the time step Δt for particle diffusion in all simulations was 0.0001 s and for reactions was 0.05 s. The diffusion coefficients used in **d–f** were 30 $\mu m^2 \cdot s^{-1}$ for soluble molecules and 0.3 $\mu m^2 \cdot s^{-1}$ for membrane-bound molecules.

Simulation of second order reactions

Many important biological reactions can be described by second order reaction kinetics where two molecules react together. Therefore, we next looked to establish and validate ABM methodology describing second order reactions of the form:

$$A + B \xrightarrow{k_f} AB.$$

Simulation of second order reactions of two soluble reactants

Pogson and colleagues previously described a method for ABM of second order reactions that is valid when both reactants are freely soluble[19]. A molecule of A reacts with a molecule of B if the molecule of B ends the iteration within an interaction volume V_i around A, calculated from the k_f in $M^{-1} \cdot s^{-1}$ and the time step Δt. Assuming V_i is distributed as a sphere around the centre of mass of A then the two molecules react if they end an iteration separated by less than an interaction distance d_i:

$$d_i = \sqrt[3]{\frac{3k_f \Delta t}{4\pi 10^3 N_A}} \quad (7)$$

where N_A is Avogadro's constant (Fig. 1d.i). Full derivation of the V_i and d_i is well described and illustrated in the publication of Pogson and co-workers.

Simulation of second order reactions of a membrane-bound reactant and a soluble reactant

Whilst Pogson and colleagues consider reactions involving two soluble reactants, they do not explicitly address the interaction of a soluble reactant with a membrane-bound reactant. However, these types of reactions are often an integral part of many cell-signalling pathways, for example, in receptor-ligand binding. Therefore, we next set out to extend these methods to include this

type of second order reaction. In reactions involving a membrane-bound and a soluble reactant, in most cases, the soluble reactant is unable to freely cross the membrane. Consequently, by assuming a sufficiently small V_i limits the impact of membrane curvature, it can be assumed that V_i is distributed as a hemisphere around the receptor (Fig. 1e.i). The interaction distance d_i for R to react with S is therefore equal to:

$$d_i = \sqrt[3]{\frac{3k_f\Delta t}{2\pi 10^3 N_A}} \tag{8}$$

Simulation of second order reactions of two membrane-bound reactants

Membrane-bound molecules also can participate in second order reactions, for example, in receptor clustering and dimerisation. Therefore, we next established ABM methodology for the second order reaction of two membrane-bound reactants. When considering the reaction of two membrane-bound molecules the general principles are the same as for soluble interactions except that as reactions take place on a planar membrane, molecules have an interaction area A_i rather than an interaction volume (Fig. 1f.i). In order to hold true for receptor-receptor interactions, receptor levels should be measured in density with units m^{-2} and k_f in $m^2{\cdot}s^{-1}$ giving an interaction distance d_i of:

$$d_i = \sqrt{\frac{k_f\Delta t}{\pi 10^3 N_A}} \tag{9}$$

However, in many cases concentration is still measured in M and the k_f determined using soluble forms of the membrane protein in the units of $M^{-1}{\cdot}s^{-1}$. In these situations, a modified value k_f, called k_f', can be calculated that is scaled to the surface area to volume ratio. In internal cellular reactions this is the cell membrane surface area A_c and the cell cytosolic volume V_c so that:

$$k_f\prime = k_f\frac{A_C}{V_C} \tag{10}$$

The substitution of Eq. 10 into Eq. 9 gives the interaction distance when k_f is in the units of $M^{-1}{\cdot}s^{-1}$ as:

$$d_i = \sqrt{\frac{k_f A_C\Delta t}{\pi 10^3 V_C N_A}} \tag{11}$$

Agent-based modelling can reproduce mass action kinetics for second order reactions

In order to validate that these methods for ABM of second-order reactions are capable of reproducing mass action kinetics, we conducted a series of simulations by ABM for all three methods and compared these to equivalent ODE models. The rate equations used were:

$$\frac{d[A]}{dt} = -k_f[A][B] \tag{12}$$

$$\frac{d[B]}{dt} = -k_f[A][B] \tag{13}$$

$$\frac{d[AB]}{dt} = k_f[A][B] \tag{14}$$

with different k_f values (10^4 $M^{-1}{\cdot}s^{-1}$, 10^5 $M^{-1}{\cdot}s^{-1}$, 10^6 $M^{-1}{\cdot}s^{-1}$, 10^7 $M^{-1}{\cdot}s^{-1}$) and different ratios of reactants. The progress of each reaction was plotted against time for soluble-soluble (Supplementary Fig. 1), membrane-soluble (Supplementary Fig. 2) and membrane-membrane reactions (Supplementary Fig. 3). These figures show a good overlay of the ABM over the ODE curves for each of the second order reaction methods described across a range of different conditions. We then went further in numerically assessing the accuracy of each individual ABM simulation by taking the ratio of the concentration of the product, *AB*, in ABM simulations ($[AB(t)]_{ABM}$) against the concentration in ODE simulations ($[AB(t)]_{ODE}$) at individual time points ($[AB(t)]_{ABM}$:$[AB(t)]_{ODE}$). We did this every 0.05 s for 5 min, or until reaction completion, and then calculated the average ratio (ABM:ODE Score) as a score with an idealised value of 1 representing perfect agreement. Each reaction was repeated in three independent ABM simulations and the calculated ABM:ODE Scores were plotted in Fig. 1d.ii, 1e.ii, and 1f.ii for two soluble reactants, a membrane-bound to a soluble reactant and two membrane-bound reactants reactions, respectively.

The data in Fig. 1d–f show good agreement between the ABM and ODE with all mean values centred on, or proximal, to the perfect agreement ratio of 1. When the reaction is slower, due to the low k_f value and low levels of reactants, the amount of noise increases due to natural stochastic effects having greater weight relative to the mean. However, the mean values still demonstrate excellent agreement with the ODE even under these conditions. It may be expected that when the binding interaction distance and concentration of reactants is high that these methods will undervalue the reaction kinetics because of the increasing probability of multiple substrates falling within the interaction distance in a single iteration. In these cases, a decision is made on which substrate to bind based on proximity. The risk of this can be minimised by reducing the time step accounting for the k_f and concentration of reactants. In this section we have shown new ABM methods and demonstrated that they are able to reproduce second-order reaction kinetics successfully across a wide range of scenarios.

Simulation of reversible reactions

Reversible reactions form an integral part of many cell-signalling pathways, with dynamic forward and reverse reactions occurring simultaneously even in steady-state conditions. The combination of the methods described above for the different forward and reverse reactions can be combined together to give reversible reactions. However, when using a second order forward reaction, as reactions occur according to proximity, this can lead to a problem where two molecules are highly to react immediately after their dissociation, a phenomenon termed as germinate recombination in the work of Andrews and Bray[15]. We limit this effect by the introduction of an unbinding distance d_u equal to:

$$d_u = 4d_i \tag{15}$$

The unbinding distance is an arbitrary distance used to separate two reactive molecules after dissociation, thus preventing their immediate reassociation. Therefore, by combining this with the second order reaction and first order reaction methodology presented previously, we can model reversible reactions by ABM.

Integration into the FLAME-accelerated signalling tool

Cell signalling networks involve complex networks of many reactants and reactions occurring simultaneously. Therefore, once we established and validated the methods described in the previous section we set out to create a tool for the facile writing of complex cell signalling networks as ABM simulations compatible with FLAME and FLAME GPU. The FaST is a Matlab-encoded tool fronted with a graphical user interface. It requires two text input files, one containing agent properties and the other containing reaction properties. The agent input file lists the agent name, its cellular localisation, diffusion coefficient and concentration. The reaction input file lists the reactants and products involved in each reaction, the type of reaction and the reaction constants. The FLAME-accelerated Signalling Tool produces FLAME and FLAME GPU simulation code from these input files along with associated tools for data retrieval and initial state generation. Furthermore,

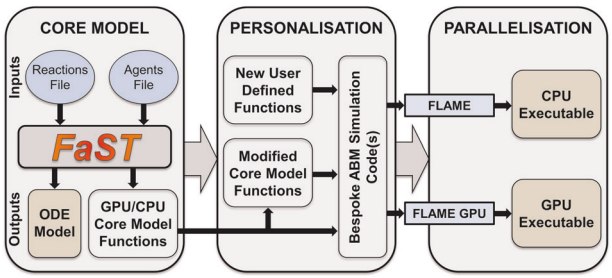

Fig. 2 Schematic of the FLAME-accelerated signalling tool. The *Agents Input File* includes information on the localisation, concentration and diffusion coefficient of the agents in the system. The *Reactions Input File* includes the reaction, reaction type and reaction kinetic data. The FaST then generates ABM files compatible with FLAME and FLAME GPU, as well as an ODE model. FLAME and FLAME GPU then convert these into ABM executable models using the appropriate compilation tools. Furthermore, the user can also modify the core ABM code and add new functions, but still easily parallelise their bespoke ABM simulation code using FLAME or FLAME GPU.

the FaST has the option to produce an equivalent ODE model for direct comparison between ABM and ODE simulations. The codes can be compiled as they are for CPUs or GPUs. However, they can also be easily modified to create new bespoke ABM codes for parallelisation on CPUs or GPUs, taking advantage of the inherent parallelisation optimisation in FLAME and FLAME GPU (Fig. 2). Example input files and the tool itself is provided in source and cost-free binary format (for windows, mac and linux) through the Zenodo platform[30].

The FLAME-accelerated signalling tool is able to convert ODE models into agent-based models

In order to test the applicability of the FaST to modelling cell-signalling processes, we used it to convert a modified form of a previously well-characterised ODE model of apoptosis execution signalling into an ABM model[7]. In apoptosis execution signalling, upstream death signals lead to mitochondrial outer membrane permeabilisation (MOMP). This permeabilisation allows the release of the mitochondrial located proteins cytochrome c and Second Mitochondria-derived Activator of Caspases (SMAC). These two proteins activate a signalling cascade that results in apoptosis. This signalling requires the formation of a protein complex called the apoptosome[31,32]. This complex is used as a platform for the activation of the inactive precursor of the initiator caspase, pro-caspase 9 (PC9), into caspase 9 (C9), which in turn activates the inactive precursor of the executioner caspase, pro-caspase 3 (PC3), into the active caspase 3 (C3). C3 then cleaves numerous downstream substrates that invoke apoptosis, but can also cleave C9[31–34]. This process is inhibited by the actions of X-linked Inhibitor of Apoptosis Protein (XIAP), which binds, inhibits and promotes ubiquitin-mediated degradation of the active form of C3 and C9[35–38]. However, XIAP is unable to bind and inhibit C3-cleaved C9 (C9P) creating a positive-feed back loop. The actions of XIAP are countered by SMAC, which, after its release from the mitochondria, binds XIAP and actively breaks up caspase-XIAP complexes[39–42]. The reaction network consists of 14 protein/protein complexes and 23 individual reactions (Fig. 3a). Full details of the model, including starting concentrations, reactions, reaction kinetics and diffusion coefficients are summarised in the Supplementary Note 2 and Supplementary Tables 1 and 2. The model was placed into the setup text files required by the FaST, which are also included with FaST. The FaST was then used to make the ABM simulations for both FLAME and FLAME GPU.

The ABM simulations were run with >20,000 agents and the progression of the reactions were compared, over 30 min, against the ODE model. The apoptosis execution model culminates with the cleavage of a substrate of C3, to reflect the C3-mediated cleavage of an experimental Förster Resonance Energy Transfer (FRET) probe. The cleavage of this substrate showed excellent agreement between the ODE and ABM simulations with an $R^2 > 0.999$ (Fig. 3b). We further investigated the dynamics of the apoptosis execution pathway by comparing time-course concentration changes of intermediates and visualisation of individual molecules over time (Fig. 4). The activity of C9, C3 and the XIAP-SMAC regulatory axis in the ODE and ABM simulations is compared in Fig. 4a, b, c, respectively. All complexes within the ABM simulation show excellent agreement with the ODE simulation, demonstrating that the ABM methods used in the FaST can indeed reproduce mass action kinetics of ODE simulations whilst including complex spatial information. Importantly, whilst the ABM simulations recreated mass action kinetics, they also included stochasticity, as evidenced by the observable variation in Fig. 4a, b, c. Taken together, this demonstrates scope for wider application of FaST where stochasticity and spatial information play a fundamental role in cellular signalling dynamics such as in cases of low reactant concentrations, compartmentalisation and subcellular localisation.

The FLAME-accelerated signalling tool can improve ABM performance by parallelisation

As the relatively poor performance of ABM simulations is the major drawback of ABM compared to other methods such as ODE, SDE or PDE models, we next looked at whether the use of GPU and CPU parallelisation could improve the speed of ABM simulations. We took the GPU and CPU versions of the apoptosis execution model and ran the simulation under the same conditions but on a GPU or on 1, 4, 6, 8, 12 or 16 CPUs in parallel, connected by 4× FDR InfiniBand interconnect. To benchmark the relative performance of FaST against other ABM software, we simulated the same apoptosis execution pathway using the established ABM-simulator MCell. The total run time, in hours, for each simulation is displayed in Fig. 5. The parallelisation of the ABM simulation on both GPUs and CPUs improved the runtime of the simulations compared to running on a CPU in serial with a speed up of 53.8× for GPU-acceleration and 5.8× for parallelisation across 16-CPUs. Whilst the simulator MCell had improved serial performance compared to FaST, it performed orders of magnitude slower than CPU-parallelised or GPU-parallelised FaST simulations. The speed up efficiency of CPU parallelisation can be calculated from the observed and theoretical speed up, for our ABM simulation this ranged from 37% (16-cores) to 90% (6-cores). Parallelisation efficiency decreased as more CPUs were added but this change was gradual suggesting that the addition of further CPUs would further increase speed up. Most notably, however, the GPU-accelerated version reduced the run time to under 3 h.

DISCUSSION

ABM is a powerful method for modelling cellular signalling as it can include stochastic effects, heterogeneity and spatiotemporal organisation. The major challenges in ABM of cellular signalling is in increasing the scale of simulations, decreasing the time taken for simulation and producing robust accurate modelling of biological systems. The FaST, described and tested in this paper, is software that is able to produce ABM codes that robustly model biological systems and are parallelisable on GPUs or CPUs, thus addressing these limitations in ABM.

We extended previously established methods for ABM of cell signalling and validated their potential to reproduce mass action kinetics under a wide range of conditions[19]. Importantly, we validated these methods based on conditions where reactants were evenly distributed, a key assumption of mass action kinetics

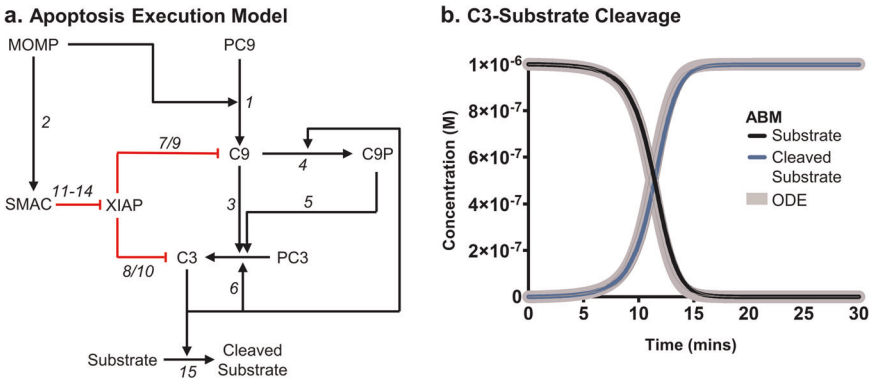

Fig. 3 Agent-based models can be used to model complex signalling networks. The network of reactions in the model of apoptosis execution is shown **a**. ABM simulations (thin solid lines) of apoptosis execution were compared to equivalent ODE simulations (thick grey lines) for final substrate cleavage (**b**). All simulations were for 30 min, the time step Δt for particle diffusion was 0.0001 s and for reactions was 0.05 s in all simulations. Data in **b** represents mean +/− range (thin grey lines) from three independent simulations.

used in ODE models. However, the added spatial component in ABM, evident in Fig. 4, provides greater scope to extend the application of FaST to more complex geometries and include uneven distribution of reactants, potentially giving rise to emergent behaviour not available through ODE-based simulation. These methods, based on the previous work of Pogson and colleagues, are similar in concept to those used in formal agent-based modelling software, such as Smoldyn, Chemcell and MCell, but operate at considerably longer time steps, therefore decreasing the amount of overall computation. Previously, it has been reported that the methods of Pogson and colleagues may not be able to reproduce kinetics observed under diffusion-limited, crowded or compartmentalised microenvironments[19,20]. We did not observe this effect in this paper (Figs. 1, 3 and 4) or in further testing, this is possibly due to the diffusion coefficients used in this paper and/or the simple geometry of the testing environment. In cases of crowded and compartmentalised environments, the accuracy of simulations can be increased by reducing the time step according to the expected concentration of reactants, kinetic rates, diffusion rates and geometry (a more detailed discussion about choosing a time step for FaST is provided in the Supplementary Note 3). In circumstances where diffusion-limited behaviour becomes problematic, alternative particle-based solvers such as the enhanced Green's Function Reaction Dynamics (eGFRD) may be more accurate but are computationally more prohibitive[12]. Alternatively, the correction suggested in the work of Klann and colleagues using collision rates is easily implementable in the framework we have presented[20]. In our ABM method, the reaction is driven entirely by proximity, rather than by collisions and activation energy that occur at the individual particle level. Several papers have suggested adapting the methodology to more accurately reflect these events using an increased binding radius, calculated from the collision rate, and a probability of reaction[19,20]. The probability of reaction can be related to the activation energy, calculable from the rate constant through the Arrhenius equation. This is easily included within the methods presented here. However, such an inclusion would increase the amount of computation due to the increased binding radius and in most circumstances, this is unlikely to significantly change outcomes in the simulation. In very crowded environments, alternative simulators such as LAMMPS or ReaDDY may be more appropriate as they include more detailed particle-particle interactions that become prominent under these circumstances[43,44]. However, with all simulations, inclusion of increased detail leads to a greater computational burden that restricts their usage to the sort of time scales (seconds/minutes) that are less relevant in biology. We believe that FaST demonstrates sufficient robustness for use in most signalling pathways whilst offering favourable computational performance to reach the longer time scales relevant for most biological processes.

We integrated the validated ABM methodology into the FaST, software that can take standard input files and generate ABM simulations compatible with FLAME and FLAME GPU. FLAME and FLAME GPU offer alternative approaches to parallelisation, increasing the feasibility of large-scale ABM simulations of cell signalling, as demonstrated in Fig. 5. Importantly, FaST makes ABM more computationally competitive compared to other approaches, including stochastic simulations and reaction-diffusion simulations, which have improved computational performance but lower spatial resolution and reduced ability to uncover emergent behaviour compared to agent-based modelling[45–47]. The CPU parallelised version of the apoptosis model increased speed up of simulation by 5.8-fold when we used 16 CPUs. The speed up by CPU parallelisation is highly dependent on the amount of inter-agent communications, as it requires messaging through the message-passing interface. Agent-based modelling of cell signalling is communication heavy, as each individual protein (potentially millions in a single cell) has to communicate its own location, traditionally making it poorly suited for CPU parallelisation. Here we reported a speed-up efficiency for parallelisation of >90% up until 6 cores, with efficiency dropping off with increased numbers of cores (37% for 16 cores), compared to optimal speed up. The lag, caused from invoking the message passing interface, is dependent on the interconnect between individual CPU units. In this paper, we used a high-performance system using a 4× Fourteen Data Rate (FDR) InfiniBand interconnect. However, recent developments of Enhanced Data Rate (EDR) and High Data Rate (HDR) systems offer improved performance in interconnect, potentially reducing overheads associated with CPU parallelisation of agent-based applications.

Message heavy applications are generally more suited to GPUs, architecture specifically designed for massively parallel processing. We reported here that our GPU-accelerated ABM simulation of apoptosis increased the speed up of simulations by 53.9-fold, compared to the serial CPU version. This presents a major improvement in feasibility of undertaking ABM simulations of cell signalling pathways. Furthermore, we envisage that our FaST GPU-accelerated ABM simulations make the application of systems theoretical approaches, including parameter estimation, sensitivity analysis and uncertainty analysis, more feasible in ABM. These techniques, frequently used in other modelling approaches, can massively improve our understanding of signalling networks but often require highly iterative running of simulations, traditionally making them ill-suited for the high computational burden of ABM[48,49]. Previously, GPU-implementations of the formal-simulator Smoldyn were shown to speed up simulations by up to 130-fold, although the GPU-implementation has reduced

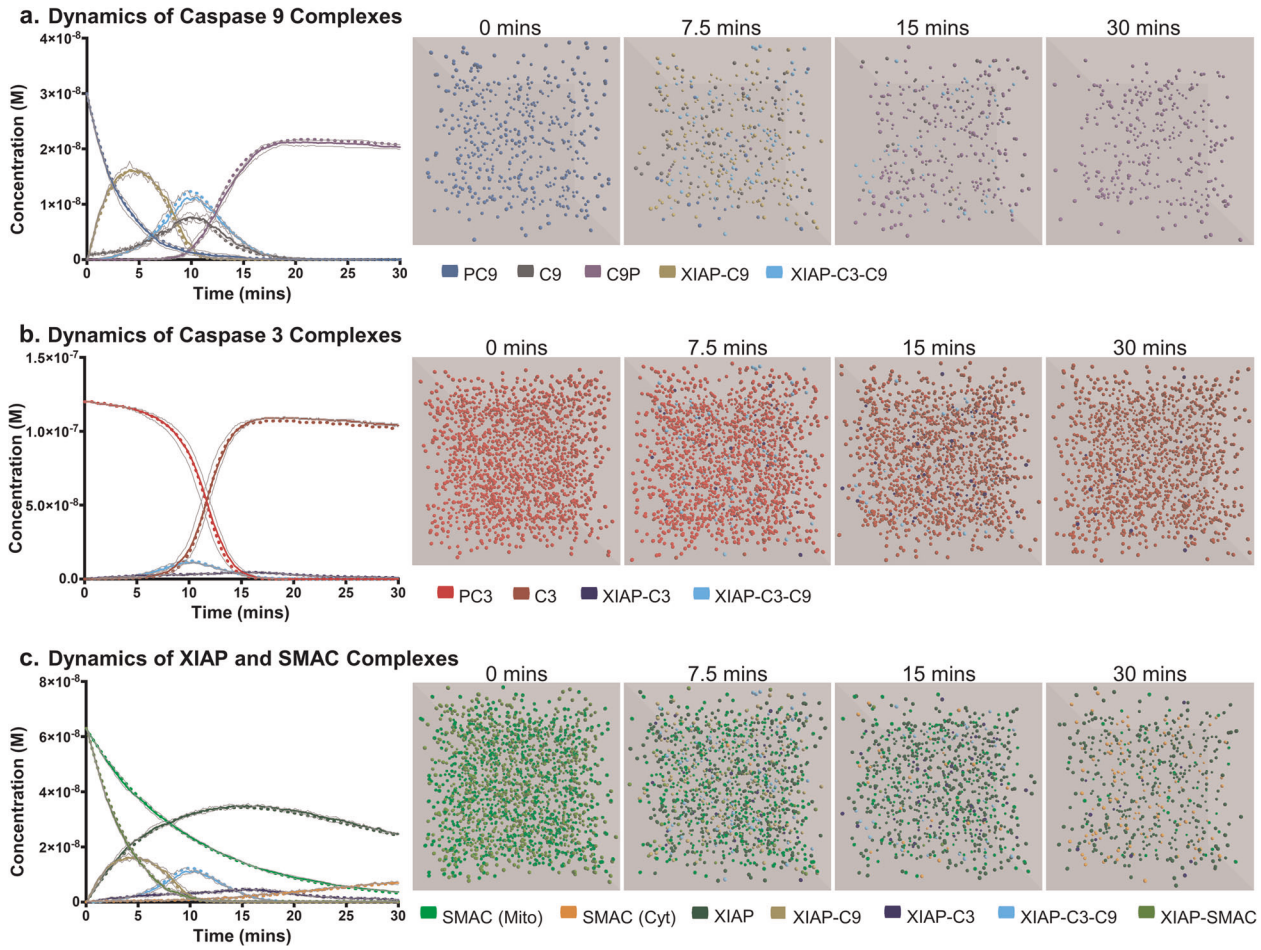

Fig. 4 Agent-based models reproduce complex pathway dynamics whilst incorporating spatial information. ABM simulations (solid lines) of apoptosis execution were compared to equivalent ODE simulations (dashed lines) for pro, activated or complex forms of caspase 9 (**a**); pro, activated or complex forms of caspase 3 (**b**); and XIAP or SMAC complexes (**c**). Visualisations of a single simulation were captured at 0, 7.5, 15 and 30 mins. All simulations were for 30 min, the time step Δt for particle diffusion was 0.0001 s and for reactions was 0.05 s in all simulations. Data in **a**–**c** represent results from a single simulation. Data in b represents mean +/− range (thin grey lines) from three independent simulations.

functionality compared to Smoldyn itself[50]. Importantly, our simulations were performed on a GPU with 96 cores, whereas current state-of-the-art units may have upwards of 4000 cores, offering a huge potential improvement in runtimes beyond those demonstrated in this paper. In most circumstances, GPU architecture likely represents the optimal platform for ABM simulations of cell signalling. However, GPUs are limited by their fixed amount of memory, which under certain circumstances may limit the scale of simulation compared to CPU versions of FLAME, where memory is less prohibitive[21]. One such scenario would be ABM-ODE hybrid simulations, for example, in a simulation where multiple cells undergo their own individual ODE for intracellular signalling, but simultaneously undertake intercellular signalling through ABM methods. Here, the memory required to store reactant concentrations for each individual cell may become impractical for GPUs but is well suited for CPU parallelisation.

The FaST is not designed to compete directly with formal simulators. MCell, Smoldyn and other formal agent-based modelling software packages offer optimised, accurate, user-friendly agent-based modelling with a wide-range of options in terms of geometric conditions[16–18]. Indeed, under serial conditions, MCell outperformed FaST (Fig. 5). This was expected as the communicating X-machine approach used by FLAME and FLAME GPU is optimised for deployment in parallel, but probably does not represent the most efficient method for ABM simulations run in serial[21,22]. The FaST, instead, is aimed to produce agent-based modelling code for simulating cell signalling, where greater personalisation and flexibility in functionality is required. The advantage of using FLAME and FLAME GPU is the ease-of-access to parallelisation without the requirement for detailed knowledge in MPI or CUDA coding, respectively. Moreover, both software packages offer a *plug-and-play* approach to agent-based modelling, where additional functionality can be added or removed through the use of self-contained functions, but with the inherent parallelisation optimisation used within FLAME. Therefore, code produced by the FLAME-accelerated Signalling Tool can be easily altered whilst retaining the ability to be easily parallelised (a tutorial of how to modify FaST-built simulations is available with the release of FaST on Zenodo[30]). Previously, FLAME and FLAME GPU have been used to model signalling processes including the NFκB pathway, *Escherichia coli* oxygen sensing and the mitogen-activated protein kinase pathway[51–53]. However, these models have always been based on relatively simple reaction networks as implementation requires extensive coding. The FaST offers easy creation of bespoke ABM simulations, of more extensive reaction networks, for FLAME and/or FLAME GPU.

In conclusion, we have presented methodology and a new software tool, the FLAME-accelerated signalling tool, for the building of agent-based models of cellular signalling that are

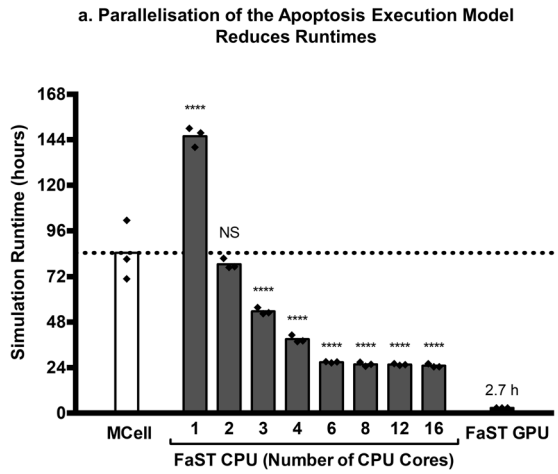

Fig. 5 Parallelisation by FaST improves runtimes of agent-based model simulations of apoptosis execution. The apoptosis execution model was run using MCell on CPU serial, FaST on CPU serial, FaST on parallel CPU or FaST on parallel GPU architecture and simulation runtimes were compared (**a**). Data represents mean and individual data points from three independent simulations. Dashed line indicates the runtime of the MCell simulation as a benchmark for FaST performance. All simulations were for 30 min, the time step Δt for particle diffusion was 0.0001 s and for reactions was 0.05 s in all FaST simulations. The time step used for MCell was 0.001 s, longer time points produced inherent errors. Asterisks (****) indicates $P < 0.0001$ from comparing FaST simulation runtimes against MCell. Significance was tested using one-way ANOVA with Dunnett's correction.

flexible and malleable but still can be easily parallelised on CPUs or GPUs using FLAME and FLAME GPU, respectively.

METHODS

Software

The FLAME-accelerated signalling tool was constructed using the Graphic User Interface tools in MATLAB 2016b, license number 886886. FLAME xparser, message libraries (libmboard) and visualiser were obtained from github (https://github.com/FLAME-HPC/) with further documentation provided at www.flame.ac.uk. The FLAME GPU Software Development Kit (SDK) was downloaded from github (https://github.com/FLAMEGPU/) with further documentation provided at www.flamegpu.com. FLAME models and analysis scripts were built and tested using GCC 4.2.1 packaged through Xcode 8.3.3 developer tools, in conjunction with OpenMPI 2.0.2 MPI libraries for parallel compilation. FLAME GPU applications were produced using the FLAME GPU SDK using NVIDIA CUDA 9. The ODE models produced by the FaST were run on MATLAB 2016b. Blender, CellBlender and MCell were downloaded from MCell's website (https://mcell.org).

Test models

The test models in Figs. 1 and S2–4 were produced in FLAME and ran in serial on a 4 GHz iMac Intel Core i7. All testing in Figs. 1, 3–5 Supplementary Figs. 2–4 and 6 were performed in a 3 μm × 3 μm × 3 μm square test environment, with the bottom edge treated as a planar membrane. Concentrations of all reactants, including membrane-bound molecules, were calculated relative to the fixed volume. All boundaries within the testing environment are treated as reflective boundaries. All simulations presented in this paper used a time step for particle diffusion of 0.0001 s and 0.05 s for reactions. MCell simulations were performed using a time step of 0.001 s, longer time steps in line with those used by FaST (0.05 s) introduced unacceptable error rates.

Speed testing

The simulations in Figs. 3b and 4 and speed testing in Fig. 5 were performed either on the Baden-Württemberg Tier 3 High Performance Computing Uni Cluster (bwUniCluster) for CPU simulations with FLAME and MCell or on an NVIDIA GeForce 630 (96-cores), 2 GB graphics card for GPU simulations. Simulation code was compiled using standard GNU compilers with the parallel FLAME message board libraries (libmboard) and message passing interface libraries (OpenMPI). Parallel CPU simulations were run on Intel Zeon E5-2670 processors with a 4× FDR InfiniBand interconnect. Partitioning was performed using a round robin approach offered by the FLAME software. Alternatively, FLAME is able to undergo geometric partitioning, where partitioning agents on separate CPUs is performed according to position. The theoretical run time for a given number (N) of CPUs (CPU_N) was calculated relative to the run time when the number of CPUs is equal to 1 ($N = 1$):

$$\text{Run time}(CPU_N) = \frac{\text{Run time}(CPU_{N=1})}{N} \quad (16)$$

The speed up of parallelised simulations was calculated relative to the CPU serial model ($CPU_{N=1}$) by the relation:

$$\text{Speed up} = \frac{\text{Run time}(CPU_{N=1})}{\text{Run time}} \quad (17)$$

The efficiency of parallelisation on CPUs was calculated from the observed speed up and the theoretical speed up of N:

$$\text{Parallelisation efficiency} = \frac{\text{Speed up}(CPU_N)}{N} \times 100\% \quad (18)$$

Reporting summary

Further information on research design is available in the Nature Research Reporting Summary linked to this article.

DATA AVAILABILITY

The FaST tool and all the models used in this study are available in the Zenodo repository[30]. All datasets generated in this study can be reproduced using the setup files and FaST provided through Zenodo. The specific datasets generated and/or analysed during the current study are available from the corresponding author on reasonable request.

CODE AVAILABILITY

The FaST tool code is hosted as Matlab source code and compiled binaries through Zenodo[30] under a Creative Commons Attribution 4.0 International license. All codes used in this manuscript are generatable through FaST. Setup files, instructions and tutorials are hosted there for reproduction of the data in Figs. 1, 3–5. MCell scripts are available from the corresponding author on request.

Received: 15 October 2019; Accepted: 17 March 2020;
Published online: 20 April 2020

REFERENCES

1. Ideker, T., Galitski, T. & Hood, L. A NEW APPROACH TO DECODING LIFE: systems biology. *Annu. Rev. Genomics Hum. Genet.* **2**, 343–372 (2001).
2. Chuang, H.-Y., Hofree, M. & Ideker, T. A decade of systems biology. *Annu. Rev. Cell Dev. Biol.* **26**, 721–744 (2010).
3. Ashall, L. et al. Pulsatile stimulation determines timing and specificity of NF-κB-dependent transcription. *Science* **324**, 242 LP–242246 (2009).
4. Nelson, D. E. et al. Oscillations in NF-κB signaling control the dynamics of gene expression. *Science* **306**, 704 LP–704708 (2004).
5. Spencer, S. L. & Sorger, P. K. Measuring and modeling apoptosis in single cells. *Cell* **144**, 926–939 (2011).
6. Würstle, M. L. & Rehm, M. A systems biology analysis of apoptosome formation and apoptosis execution supports allosteric procaspase-9 activation. *J. Biol. Chem.* **289**, 26277–26289 (2014).
7. Rehm, M., Huber, H. J., Dussmann, H. & Prehn, J. H. M. Systems analysis of effector caspase activation and its control by X-linked inhibitor of apoptosis protein. *EMBO J.* **25**, 4338 LP–4334349 (2006).
8. Ferrell, J. E., Tsai, T. Y.-C. & Yang, Q. Modeling the cell cycle: why do certain circuits oscillate? *Cell* **144**, 874–885 (2011).
9. Lipniacki, T., Paszek, P., Brasier, A. R., Luxon, B. A. & Kimmel, M. Stochastic regulation in early immune response. *Biophys. J.* **90**, 725–742 (2018).

10. Lipniacki, T., Paszek, P., Marciniak-Czochra, A., Brasier, A. R. & Kimmel, M. Transcriptional stochasticity in gene expression. *J. Theor. Biol.* **238**, 348–367 (2006).
11. Harris, L. A. et al. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics* **32**, 3366–3368 (2016).
12. Sokolowski, T. R. et al. eGFRD in all dimensions. *J. Chem. Phys.* **150**, 54108 (2019).
13. Lopez, C. F., Muhlich, J. L., Bachman, J. A. & Sorger, P. K. Programming biological models in Python using PySB. *Mol. Syst. Biol.* **9**, 646 (2013).
14. An, G., Mi, Q., Dutta-Moscato, J. & Vodovotz, Y. Agent-based models in translational systems biology. *WILEY Interdiscip. Rev. Biol. Med.* **1**, 159–171 (2009).
15. Andrews, S. S. & Bray, D. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Phys. Biol.* **1**, 137 (2004).
16. Andrews, S. S. Smoldyn: particle-based simulation with rule-based modeling, improved molecular interaction and a library interface. *Bioinformatics* **33**, 710–717 (2017).
17. Slepoy, S. J. P. & A Microbial cell modeling via reacting diffusive particles. *J. Phys. Conf. Ser.* **16**, 305 (2005).
18. Kerr, R. et al. Fast Monte Carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces. *SIAM J. Sci. Comput.* **30**, 3126–3149 (2008).
19. Pogson, M., Holcombe, M., Smallwood, R. & Qwarnstrom, E. Introducing spatial information into predictive NF-kappa B modelling-an agent-based approach. *PLoS ONE* **3**, 2367 (2008).
20. Klann, M. T., Lapin, A. & Reuss, M. Agent-based simulation of reactions in the crowded and structured intracellular environment: Influence of mobility and location of the reactants. *BMC Syst. Biol.* **5**, 71 (2011).
21. Richmond, P., Walker, D., Coakley, S. & Romano, D. High performance cellular level agent-based simulation with FLAME for the GPU. *Brief. Bioinform.* **11**, 334–347 (2010).
22. Chin, L. et al. FLAME: An approach to the parallelisation of agent-based applications. *Rutherford Applet. Lab. Tech. Reports* **501**, 63259 (2012).
23. Ribeiro, A. S. Stochastic and delayed stochastic models of gene expression and regulation. *Math. Biosci.* **223**, 1–11 (2010).
24. Knop, R. Remark on Algorithm 334 [G5]: normal random deviates. *Commun. ACM* **12**, 281– (1969).
25. Bell, J. R. Algorithm 334: normal random deviates. *Commun. ACM* **11**, 498– (1968).
26. Fullstone, G., Wood, J., Holcombe, M. & Battaglia, G. Modelling the transport of nanoparticles under blood flow using an agent-based Approach. *Sci. Rep.* **5**, 10649 (2015).
27. Banks, D. S. & Fradin, C. Anomalous diffusion of proteins due to molecular crowding. *Biophys. J.* **89**, 2960–2971 (2005).
28. Knight, J. D., Lerner, M. G., Marcano-Velázquez, J. G., Pastor, R. W. & Falke, J. J. Single molecule diffusion of membrane-bound proteins: window into lipid contacts and bilayer dynamics. *Biophys. J.* **99**, 2879–2887 (2010).
29. Weiß, K. et al. Quantifying the diffusion of membrane proteins and peptides in black lipid membranes with 2-focus fluorescence correlation spectroscopy. *Biophys. J.* **105**, 455–462 (2013).
30. Fullstone, G., Guttà, C., Beyer, A. & Rehm, M. FaST. *Zenodo* https://doi.org/10.5281/zenodo.2620047
31. Li, P. et al. Cytochrome c and dATP-dependent formation of Apaf-1/caspase-9 complex initiates an apoptotic protease cascade. *Cell* **91**, 479–489 (1997).
32. Zou, H., Henzel, W. J., Liu, X. S., Lutschg, A. & Wang, X. D. Apaf-1, a human protein homologous to C-elegans CED-4, participates in cytochrome c-dependent activation of caspase-3. *Cell* **90**, 405–413 (1997).
33. Janicke, R. U., Sprengart, M. L., Wati, M. R. & Porter, A. G. Caspase-3 is required for DNA fragmentation and morphological changes associated with apoptosis. *J. Biol. Chem.* **273**, 9357–9360 (1998).
34. Slee, E. A. et al. Ordering the cytochrome c-initiated caspase cascade: Hierarchical activation of caspases-2, -3, -6, -7, -8, and -10 in a caspase-9-dependent manner. *J. Cell Biol.* **144**, 281–292 (1999).
35. Yoo, S. J. et al. Hid, Rpr and Grim negatively regulate DIAP1 levels through distinct mechanisms. *Nat. Cell Biol.* **4**, 416 (2002).
36. Deveraux, Q. L. & Reed, T. C. IAP family proteins-suppressors of apoptosis. *Genes Dev.* **13**, 239–252 (1999).
37. Suzuki, Y., Nakabayashi, Y. & Takahashi, R. Ubiquitin-protein ligase activity of X-linked inhibitor of apoptosis protein promotes proteasomal degradation of caspase-3 and enhances its anti-apoptotic effect in Fas-induced cell death. *Proc. Natl Acad. Sci. USA* **98**, 8662–8667 (2001).
38. MacFarlane, M., Merrison, W., Bratton, S. B. & Cohen, G. M. Proteasome-mediated degradation of Smac during apoptosis: XIAP promotes Smac ubiquitination in vitro. *J. Biol. Chem.* **277**, 36611–36616 (2002).
39. Wu, G. et al. Structural basis of IAP recognition by Smac/DIABLO. *Nature* **408**, 1008–1012 (2000).
40. Rehm, M., Dussmann, H. & Prehn, J. H. M. Real-time single cell analysis of Smac/DIABLO release during apoptosis. *J. Cell Biol.* **162**, 1031–1043 (2003).
41. Du, C. Y., Fang, M., Li, Y. C., Li, L. & Wang, X. D. Smac, a mitochondrial protein that promotes cytochrome c-dependent caspase activation by eliminating IAP inhibition. *Cell* **102**, 33–42 (2000).
42. Verhagen, A. M. et al. Identification of DIABLO, a mammalian protein that promotes apoptosis by binding to and antagonizing IAP proteins. *Cell* **102**, 43–53 (2000).
43. Hoffmann, M., Fröhner, C. & Noé, F. ReaDDy 2: fast and flexible software framework for interacting-particle reaction dynamics. *PLOS Comput. Biol.* **15**, e1006830 (2019).
44. Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **117**, 1–19 (1995).
45. Bonabeau, E. Agent-based modeling: methods and techniques for simulating human systems. *Proc. Natl Acad. Sci. USA* **99**, 7280 LP–7287287 (2002).
46. Figueredo, G. P., Siebers, P.-O., Owen, M. R., Reps, J. & Aickelin, U. Comparing stochastic differential equations and agent-based modelling and simulation for early-stage cancer. *PLoS One* **9**, e95150 (2014).
47. Parunak, V., Savit, R. & Riolo, R. Agent-Based Modeling vs. Equation-Based Modeling: a Case Study and Users' Guide. In *Proceedings of the 1998 Workshop on Multi-Agent Systems and Agent-Based Simulation*, Springer (2000).
48. Banga, J. R. Optimization in computational systems biology. *BMC Syst. Biol.* **2**, 47 (2008).
49. Moles, C. G., Mendes, P. & Banga, J. R. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.* **13**, 2467–2474 (2003).
50. Gladkov, D. V., Alberts, S., D'Souza, R. M. & Andrews, S. Accelerating the Smoldyn Spatial Stochastic Biochemical Reaction Network Simulator Using GPUs. In *Proceedings of the 19th High Performance Computing Symposia* 151–158 (Society for Computer Simulation International, 2011).
51. Rhodes, D. M., Smith, S. A., Holcombe, M. & Qwarnstrom, E. E. Computational modelling of NF-κB activation by IL-1Rl and its co-receptor TILRR, predicts a role for cytoskeletal sequestration of IκBα in inflammatory signalling. *PLoS One* **10**, e0129888 (2015).
52. Bai, H. et al. Agent-based modeling of oxygen-responsive transcription factors in Escherichia coli. *PLOS Comput. Biol.* **10**, e1003595 (2014).
53. Shuaib, A., Hartwell, A., Kiss-Toth, E. & Holcombe, M. Multi-compartmentalisation in the MAPK signalling pathway contributes to the emergence of oscillatory behaviour and to ultrasensitivity. *PLoS One* **11**, e0156139 (2016).
54. Sanft, K. R. et al. StochKit2: software for discrete stochastic simulation of biochemical systems with events. *Bioinformatics* **27**, 2457–2458 (2011).
55. Drawert, B., Engblom, S. & Hellander, A. URDME: a modular framework for stochastic simulation of reaction-transport processes in complex geometries. *BMC Syst. Biol.* **6**, 76 (2012).
56. Zhang, F., Angermann, B. R. & Meier-Schellersheim, M. The Simmune Modeler visual interface for creating signaling networks based on bi-molecular interactions. *Bioinformatics* **29**, 1229–1230 (2013).
57. Arjunan, S. N. V. & Tomita, M. A new multicompartmental reaction-diffusion modeling method links transient membrane attachment of E. coli MinE to E-ring formation. *Syst. Synth. Biol.* **4**, 35–53 (2010).
58. Sneddon, M. W., Faeder, J. R. & Emonet, T. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nat. Methods* **8**, 177–183 (2011).

ACKNOWLEDGEMENTS

This work has been supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreements #642295 (MEL-PLEX) and #766069 (GLIO-TRAIN) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2075—390740016. The authors would like to thank Paul Richmond, Mozhgan Kabiri Chimeh and Peter Heywood at the University of Sheffield for support and feedback with using FLAME GPU. The authors acknowledge support by the state of Baden-Württemberg through bwHPC.

AUTHOR CONTRIBUTIONS

G.F. and M.R. designed the study and wrote the manuscript. G.F. implemented the software tool, performed the testing and analysed the data. C.G. contributed to the O. D.E. implementation in FaST. A.B. performed the CPU parallel tests. All authors reviewed the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information is available for this paper at https://doi.org/10.1038/s41540-020-0128-x.

Correspondence and requests for materials should be addressed to G.F. or M.R.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.