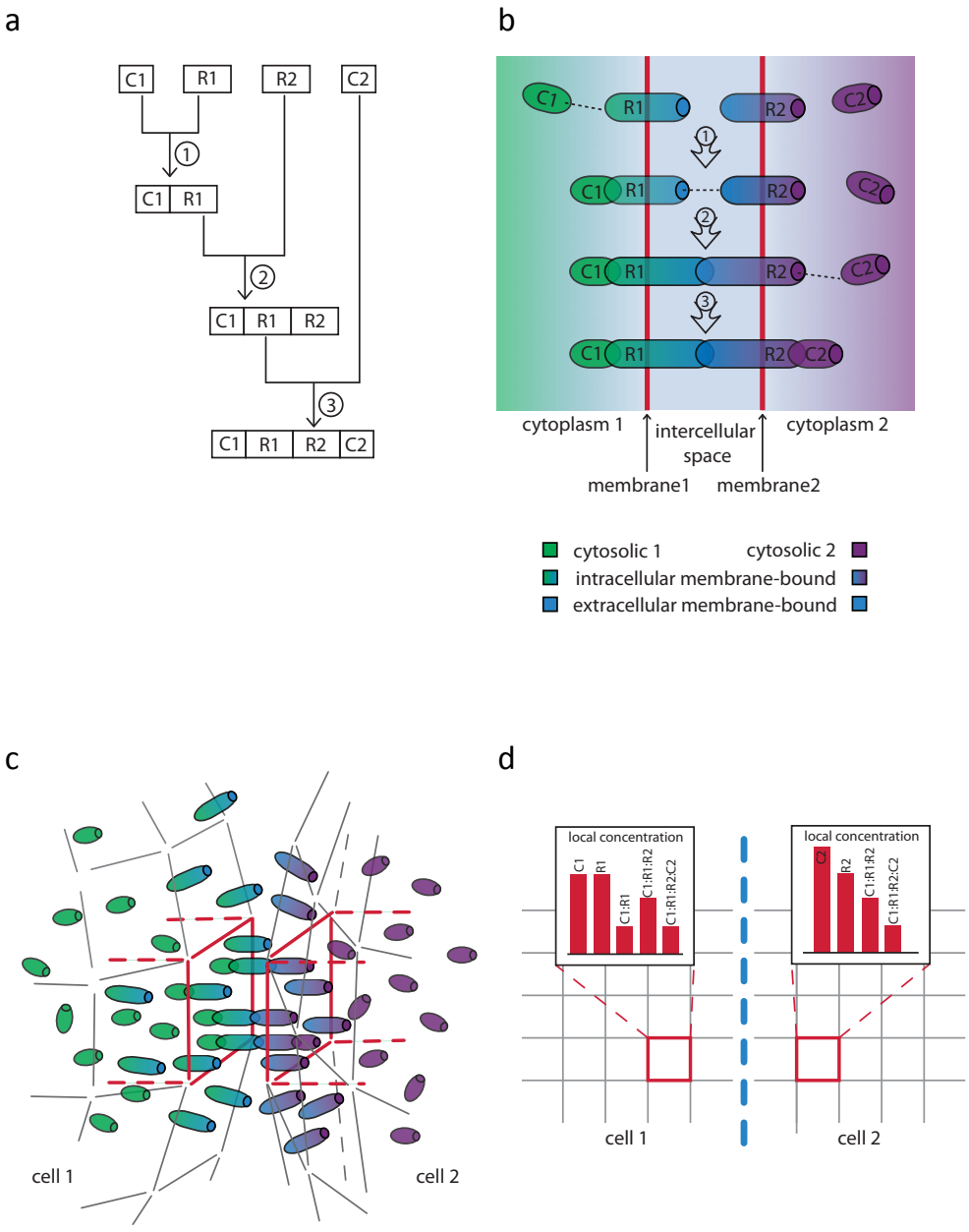
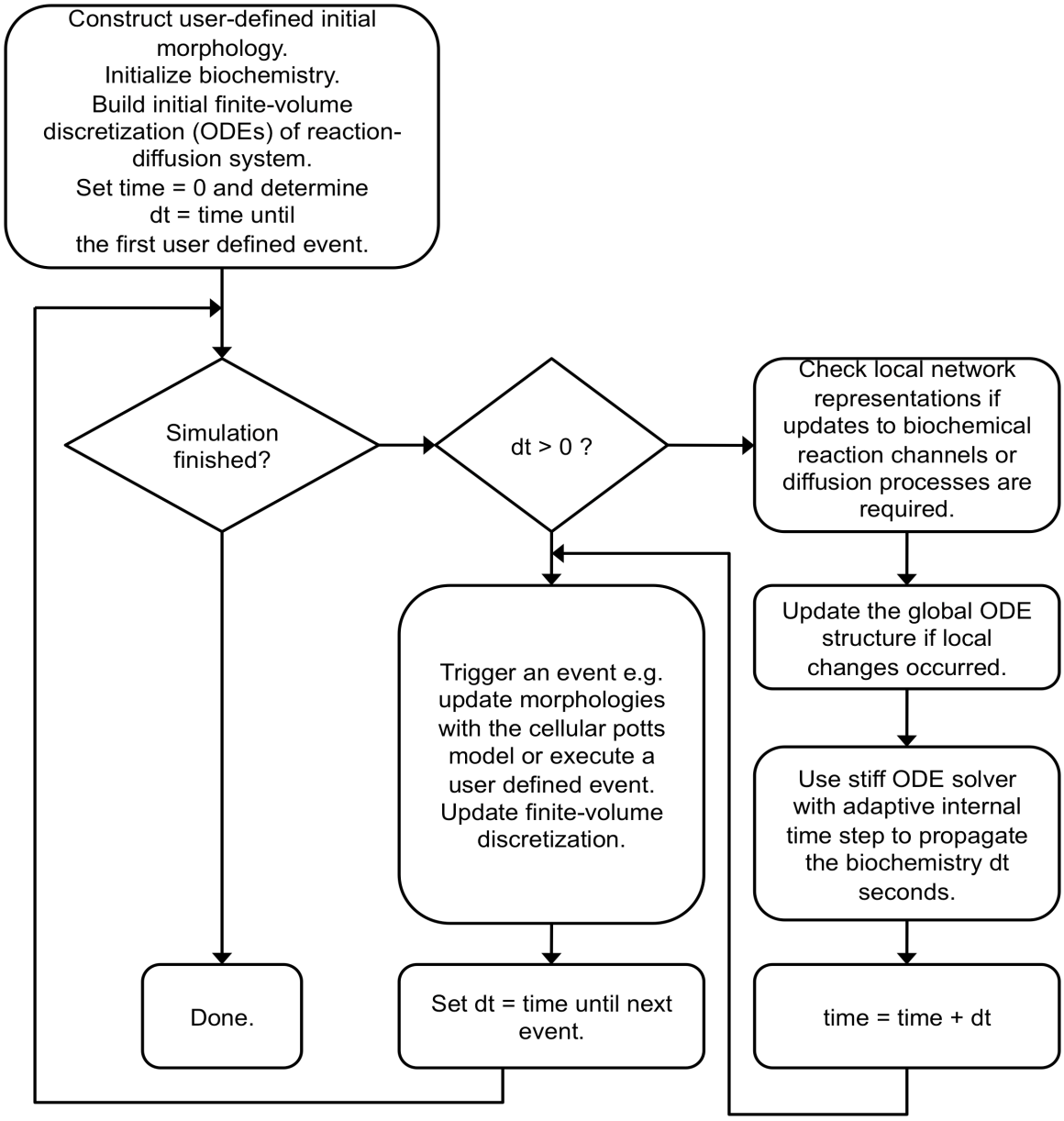
Supplementary Figure 1: Capturing spatial aspects with particle- or concentration-based approaches



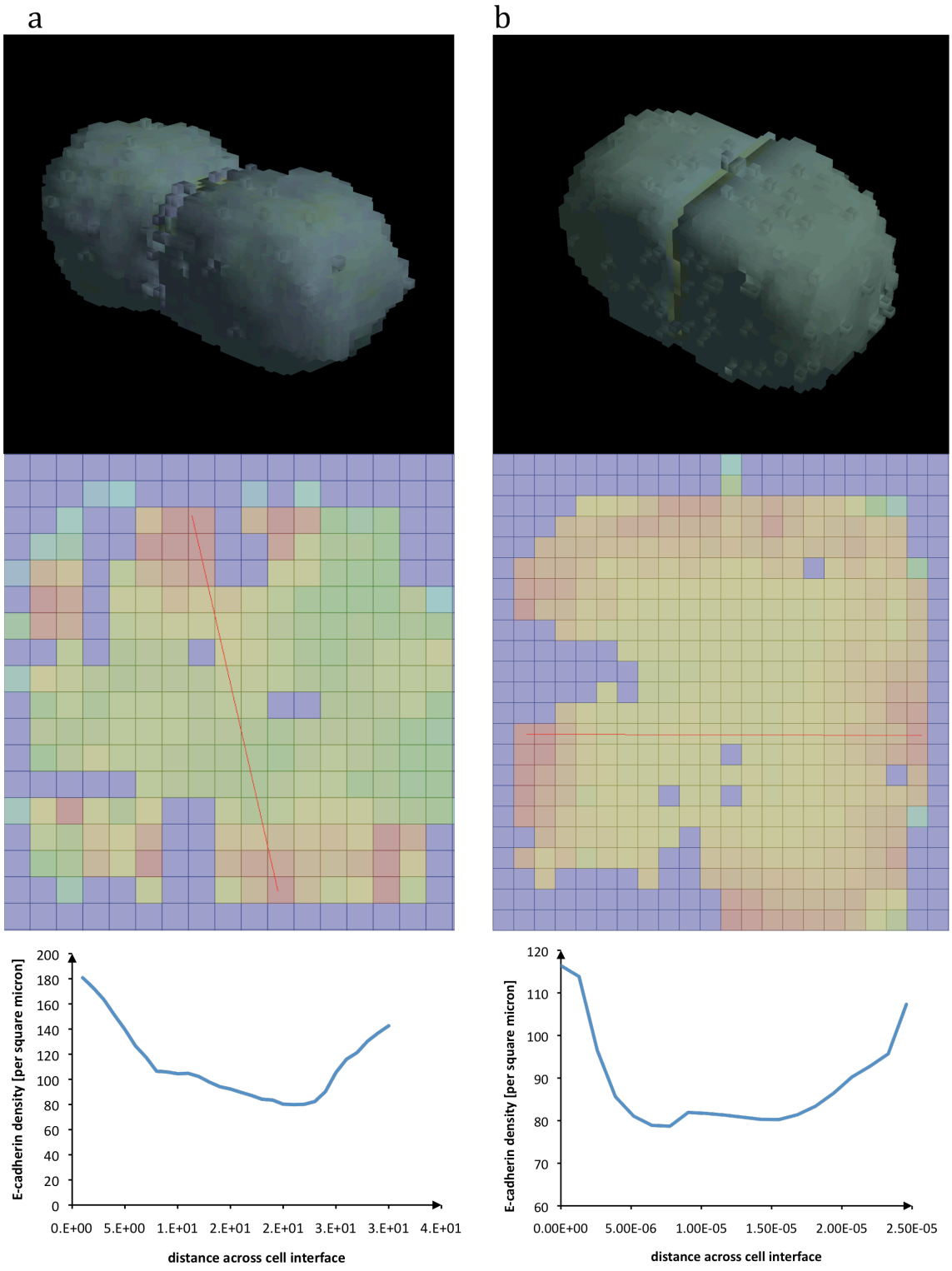
a) Non-spatial reaction network representation of a system of two receptors R_1 and R_2 and cytoplasmic components C_1 and C_2.
b) Spatial representation of the two-receptor system from (**a**). The cytoplasm of cell 1 is indicated in green, the cytoplasm of cell 2 in purple and the intercellular space in blue.
c) In a representation that tracks individual molecular complexes, the interaction possibilities are determined by spatial proximity: intercellular complexes can form at those locations where receptors from both cells are close enough to establish bonds. Discretization of the cells' surfaces using a grid-like structure permits representing molecular complexes in terms of their local concentrations as depicted in (**d**). The concentration-based approach is in most cases more efficient than tracking individual complexes but has to simulate reactions without knowing the locations of such complexes.
d) An approach that simulates the dynamics of a reaction network based on local concentrations of molecular complexes within the elements of a grid-like representation of cellular surfaces has to keep track of the interdependencies between those concentrations in cases where they are coupled by molecular interactions (inter-membrane) and diffusion (intra-membrane). The two red squares correspond to the red surface grid elements in (**c**).

Supplementary Figure 2: The main simulation loop of Simmune



The flow diagram shows how Simmune performs the time integration. Note that multiple events could occur at the same time without integration of the biochemistry between the different events. Possible events other than performing a Potts update are updates of the user interface, export of the simulation state or other user-defined changes to the simulation. The order of simultaneous events is determined by assigning a unique priority to each type of event (see the documentation of the Simmune API for details).

Supplementary Figure 3: Changing the balance between contact growth rate and E-cadherin diffusivity towards the latter leads to accumulation of the receptors at the periphery of the cell-cell contacts.

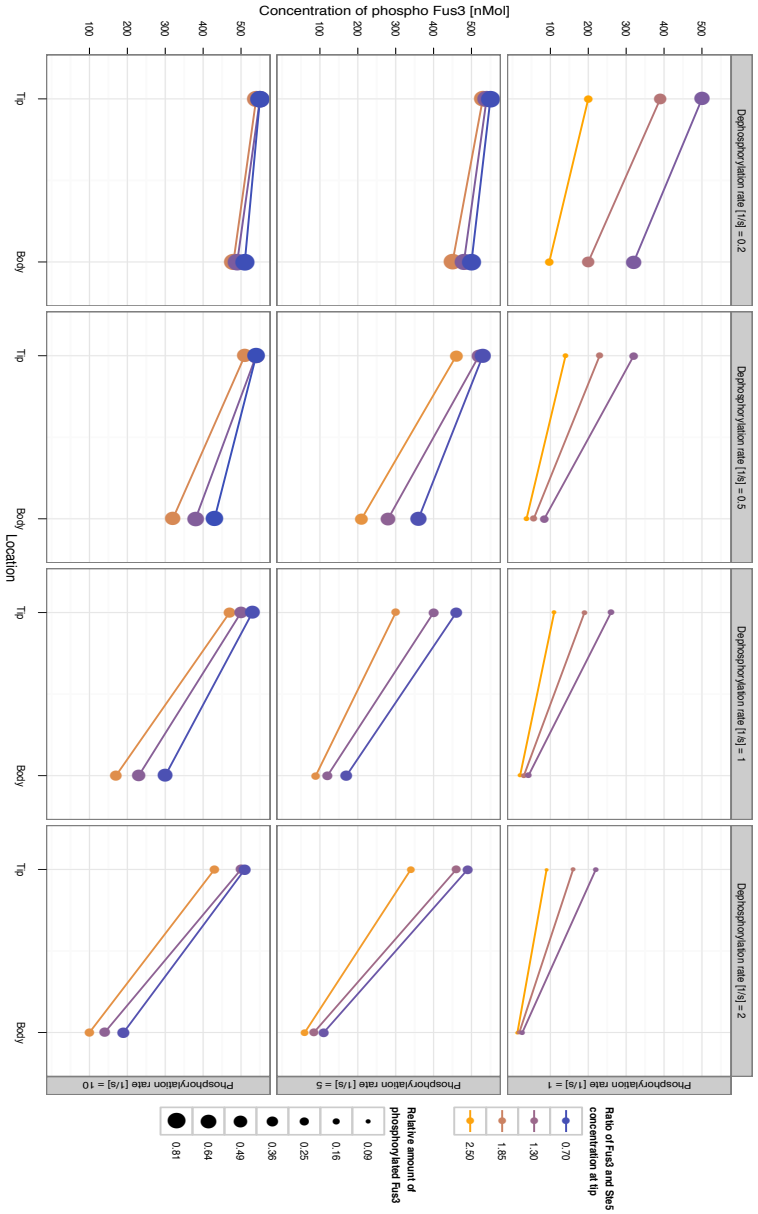


E-cadherin concentration profile (lower panels) at the dynamically grown contact interface (middle panels, with color-coded density of E-cadherin molecules: red (high concentration) to blue (low concentration)) between two cells (upper panels). The red lines across the cuts through the interfaces show the line profile paths used to generate the plots showing the concentration of E-cadherin as a function of the distance along the interface.

a) For this simulation, the diffusion coefficient of E-cadherin was set to 0.001 square microns s^{-1}, a value that is ten-fold lower than the experimentally reported value. Due to the slow diffusion, E-cadherin accumulated at a later stage in the contact formation and could therefore become trapped at the periphery of the contact, in contrast to the simulations with a physiological value for the diffusion coefficient of E-cadherin that led to central accumulation expanding with the growing contact region (see **Fig. 4d** in the main text). The contact zone E-cadherin profile was generated at 1.5 h simulated biological time.

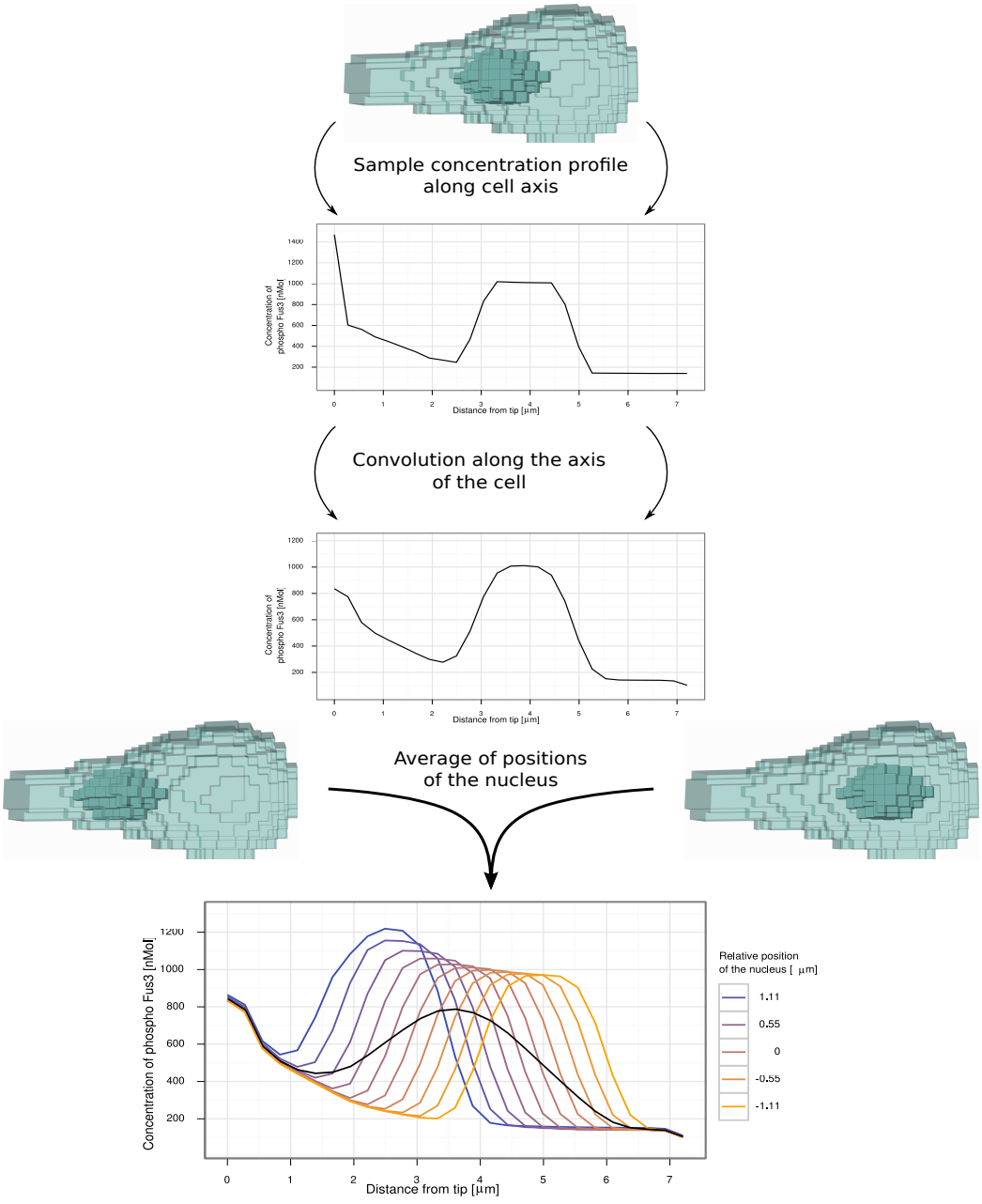
b) The Potts parameters for this simulation were set to perform updates of all volume elements at every time point (in contrast to a fraction of 0.1 used in all other Potts model simulation reported here). In addition, the volume conservation constraint was released from 1 to 0.5. This led to the formation of a broad (> cell diameter) cell-cell contact within less than 20 minutes simulated biological time. The contact growth rate was thus increased ~ five-fold. The early formation of a broad contact then allowed the edges of the contact to act as diffusional traps for E-cadherin molecules becoming ligated to binding partners on the adjacent membrane.

Supplementary Figure 4: A model that does not include shmoo tip localized Fus3 interaction partners other than Ste5 and Ste7 cannot generate a sufficiently steep Fus3/pFus3 gradient while at the same time phosphorylating 40% of the cellular Fus3 pool



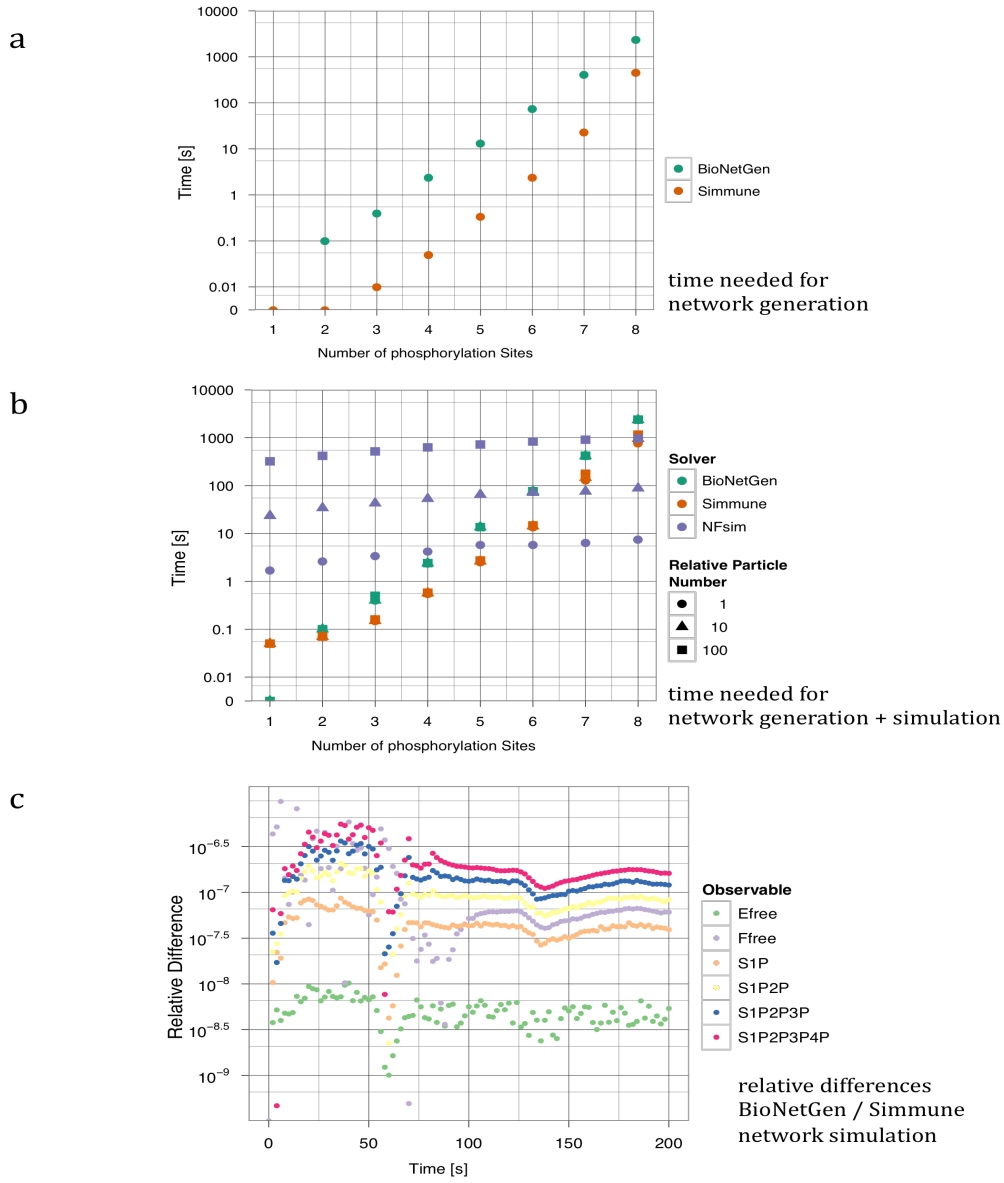
Intracellular gradient (difference between tip and body) and total production of phosphorylated Fus3 (expressed as fraction of the total pool, shown as diameter of line-end markers) as a function of Ste7 kinase activity (k_{cat}), Msg5 phosphatase activity (k_{cat}) and Ste5 recruitment to the tip (expressed as ratio of Fus3 and Ste5, color-coded blue to orange) in a model that does not include pFus3 interaction partners (other than Ste5) at the tip. With increasing phosphatase activity, the intracellular activity gradient becomes steeper. At the same time, however, the total production of pFus3 decreases. Very strong accumulations of Ste5 at the tip can generate steep gradients (4:1 and higher) and lead to almost 40% phosphorylated Fus3 but such accumulations of Ste5 relative to Fus3 (Fus3:Ste5=0.65) were ruled out by experimental measurements showing that the concentration of Fus3 in the tip is 1.7 fold higher than that of Ste5. The parameter scan for which the results are shown here can be performed using the files provided in the 'YeastScan_Kinase_Phosphatase_Recruitment' folder.

Supplementary Figure 5: Calculation of the pFus3 concentration profile from multiple simulation runs with varying nuclear position



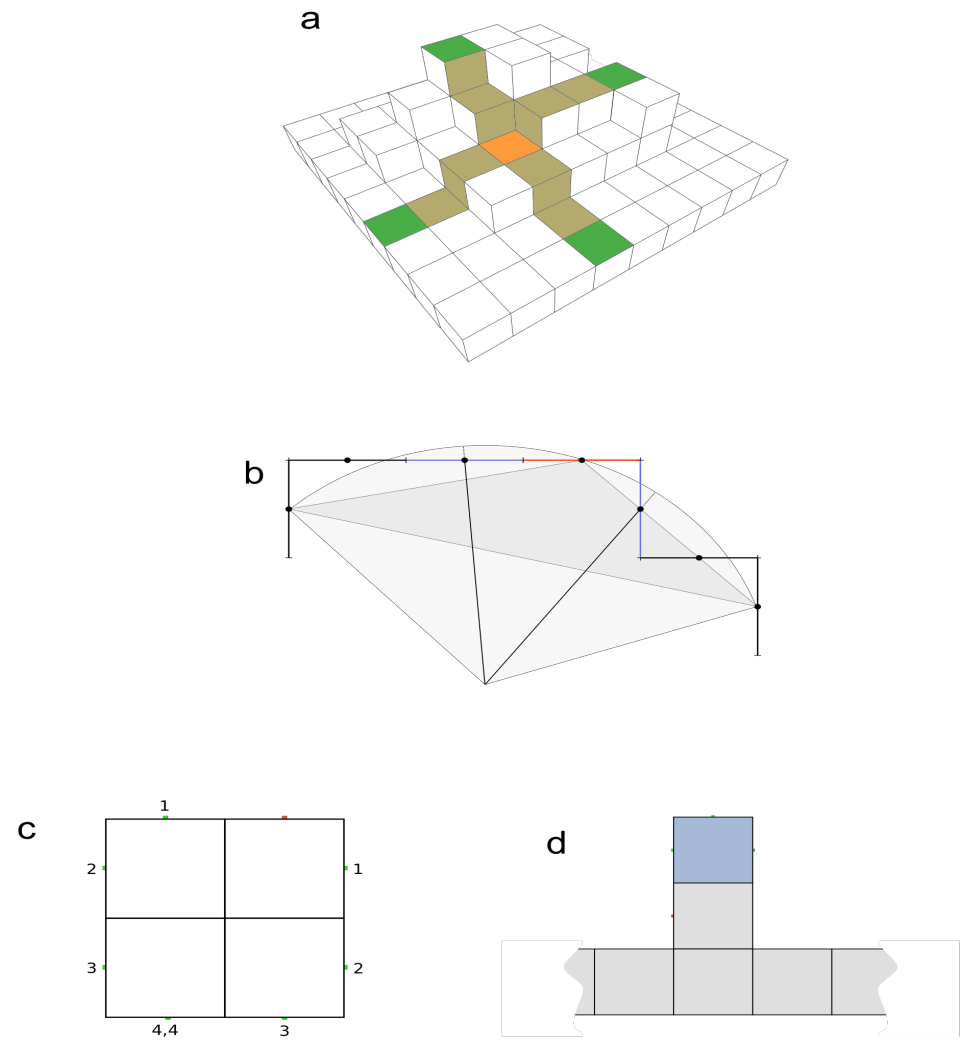
The intracellular concentration profile of phosphorylated Fus3 is strongly modulated by the position of the nucleus that accumulates a high concentration of this activated kinase. The limited resolution of an optical microscope is mimicked by sampling volume elements perpendicular to the section using a gaussian weight distribution with a FWHM of 1.37 microns. Within the section we assume a better resolution corresponding to a FWHM of 0.82 microns. Since the position of the nucleus varies among different cells (see Fig. 5d and reference [21]), our simulations had to simulate yeast cells with varying nuclear positions. The results from those simulations were then combined (a uniform probability distribution for the nuclear position was assumed) to generate the pFus3 profile for comparison with experimental data.

Supplementary Figure 6: Assessing performance and accuracy of Simmune when compared to other approaches



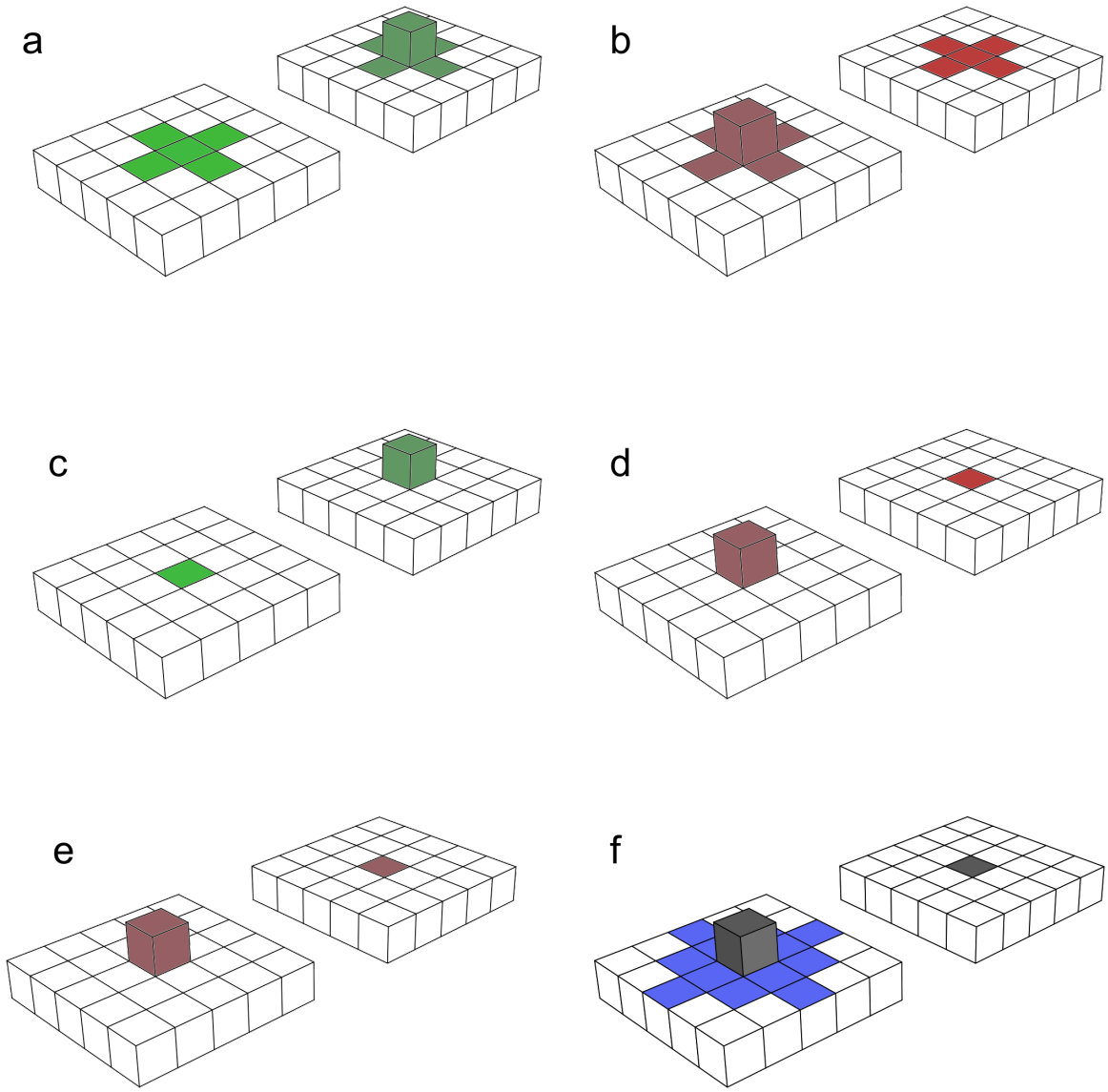
a) Comparison of the times needed to generate networks of various complexity based on the multi-site phosphorylation model suggested by Sneddon et al. (Nat. Meth. 2011 Feb;8(2):177-83). The number of different molecular complexes that need to be built for a model with n phosphorylation sites is 4n+2. For the tested model sizes (up to eight phosphorylation sites corresponding to up to 65538 automatically built complexes) Simmune is consistently faster (up to 20 times) but scales less well than BioNetGen.
b) Comparison of the times needed to generate and simulate (for 200 seconds) networks of various complexity based on the multi-site phosphorylation model suggested by Sneddon et al. Only for large networks and low particle numbers is NFSim as fast or faster than BioNetGen or Simmune.
c) Assessment of the relative differences between the numerical solutions calculated for a set of molecule complex patterns of the four-site phosphorylation model when simulated with BioNetGen or Simmune. The differences correspond to solver tolerances, suggesting that both approaches correctly generate and simulate the model network.

Supplementary Figure 7: Details of the calculation of effective surface geometries



a) Illustration of the iteration over element edges to find distant neighbors. The four elements shown in green are at a distance of Δk=4 steps from the central element highlighted in orange. The paths leading to the distant elements are indicated in beige.
b) Determination of the arc lengths using the circumcircle. The surface element under consideration is shown in red. The other two vertices of the triangle spanning the circumcircle are Δk=3 steps from the central element. The arc lengths are given by the length of the section of the circumcircle between the center of the element shown in red and intersection of the radius of the circumcircle going through the centers the neighboring elements (shown in blue).
c) Illustration of the stepping algorithm wrapping around a small detail. Numbers show the number of steps taken from element under consideration. After four steps the algorithm encounters the same element.
d) Example of a situation where the stepping algorithm encountering a small detail.

Supplementary Figure 8: Effects of the different redistribution algorithms of membrane molecules



a) Surface elements used for the redistribution of membrane molecules when extending a volume element with the scheme conserving the number of molecules, using the neighbors of changing elements.
b) Surface elements used for the redistribution of membrane molecules when retracting a volume element with the scheme conserving the number of molecules, using the neighbors of changing elements.
c) Surface elements used for the redistribution of membrane molecules when extending a volume element with the scheme conserving the number of molecules, using only changing elements.
d) Surface elements used for the redistribution of membrane molecules when retracting a volume element with the scheme conserving the number of molecules, using only changing elements.
e) Surface elements used for calculating the concentration average when conserving concentrations.
f) Neighbors of changing surface elements, for which the concentration instead of particle number is conserved, when the concentration conservation is enabled for an extended neighborhood.

Supplementary Note 1

Comparison of features among different modeling approaches and tools

Features	Tools that implement them
(a) Automated network generation	(1)(2)(4)(7)(8)(12)
(b) Spatially-resolved simulation	(1)(5)(6)(9)(10)(11)(12)(13)(14)
(c) Spatially-resolved automated network generation	(1)(12 *see explanation of features)
(d) Stochastic simulation possibility	(2)(3)(4)(5)(6)(8)(9)(10)(11)(12)(13)(14)
(e) Stochastic spatially-resolved simulation	(3)(5)(6)(9)(10)(11)(12)(13)
(f) Particle-based simulations	(3)(4)(5)(6)(8)(10)(11)(12)
(g) (Deterministic) differential equations	(1)(2)(3)(7)
(h) Automatic dynamic adaptation of networks to changing geometry	(1)
(i) GUI for defining reaction rules as components for automated network generation	(1)(2)(4)
(j) GUI for defining spatial aspects of reaction rules	(1)
(k) SBML export of automatically generated networks	(1)(2)(4)(7)

(1) Simmune 2.0 (this manuscript); (2) BioNetGen [1]; (3) E-cell [2]; (4) Kappa [3]; (5) MCell [4]; (6) Meredys [5]; (7) Moleculizer [6]; (8) NFSim [7]; (9) Smart Cell [8]; (10) Smoldyn [9]; (11) Spatiocyte [10]; (12) SRSim [11]; (13) STEPS [12]; (14) Virtual Cell [13].

Explanation of the features:

(a) Automated network generation: ability to generate the resulting network of (multi-molecular) complexes and their reactions (associations, dissociations, transformations (enzyme-dependent or spontaneous)) based on the definition of interactions between pairs of molecules. In many cases, the results of molecular interactions or even whether the interactions take place or not depend on the properties of the interacting molecules. For example, the binding of an enzyme to its substrate may only happen if this substrate has the right phosphorylations and is in an 'open' configuration, allowing access to the binding sites used by the enzyme. Such conditional interactions are frequently referred to as 'interaction rules'.
(b) Spatially-resolved simulation: ability to simulate reaction-diffusion of interacting molecular complexes using a 2D or 3D computational representation of the simulated geometry of the biological system being modeled. Note that this is different from far simpler compartmental representations that use, for instance, one cytosolic compartment and one nuclear compartment and simulated exchange between the two but does not actually simulate diffusion of molecules.
(c) Spatially-resolved automated network generation: ability to use the combination of interaction rules and simulated geometry to generate reaction-diffusion networks that take into account the influence of local features of the simulated geometry (e.g., the presence of membranes) on the reaction-diffusion network. *The SRSim approach has a different focus: it allows the modeler to include information about how the spatial structures of

molecules influence their interaction behaviors by imposing constraints on possible relative orientations of interacting molecules.
(d) Stochastic simulation possibility: ability to calculate the time evolution of the simulated biological system using stochastic techniques.
(e) Stochastic spatially-resolved simulations: ability to combine stochastic simulation and spatial resolution of the simulated system.
(f) Particle-based simulations simulate the motion and interactions of discrete, individual particle representations of the modeled biochemistry as opposed to local concentrations.
(g) Deterministic differential equations can be used as opposed to stochastic approaches.
(h) Automatic dynamic adaptation of networks to changing geometry: ability to simulate reaction-diffusion processes under conditions in which the simulated space (e.g., the shape of a cell) changes over time.
(i) GUI for defining reaction rules as components for automated network generation: users don't have to write scripts when specifying the molecular interactions in the simulated system.
(j) GUI for defining spatial aspects of reaction rules: users can employ graphical symbols when defining how molecular interactions are influenced by, for instance, the position of molecular components relative to a membrane.
(k) SBML export of automatically generated networks: ability to generate a reaction network automatically, based on the specification of molecular interactions, and then create a representation of the network as SBML file.

When does a modeling project need which features?

Systems with molecular interactions that can lead to large complicated networks are difficult to translate into quantitative simulations using 'hand-written' equations [3, 14] In those cases, the ability of a modeling program to create the reactions networks automatically (feature (a)) can be very helpful. For projects that aim at modeling biological systems in which the assumption of a well-stirred homogeneous biochemistry is unrealistic, for instance because of the formation of concentration gradients or the existence of membranes, the simulation approach has to be able to represent the spatial structure of the underlying biological system (feature (b)). The inclusion of this spatial aspect into the model may require defining molecular properties and interaction modes that go beyond conventional interaction rules. As is illustrated in the E-cadherin example system, switching from a non-spatial to a spatial representation of a signaling network can introduce considerable complexity. A modeling tool implementing feature (c) can reduce the error potential when transitioning from a non-spatial to a spatial network. In particular for large reaction networks, the translation of their dynamics into spatially-resolved simulations may become prohibitively difficult. Note that simulations that include changing geometries necessarily need the ability to generate such networks automatically to be able to adjust to such changes (feature (h)).

All biochemical processes in a living cell are stochastic at the single-molecular level. In systems whose behavior is governed by processes that involve very low molecular concentrations, stochastic effects play an important role and modeling such systems requires tools capable of simulating these effects (feature (d)). Feature (e) permits

stochastic simulations of systems that have spatial characteristics that need to be accounted for while stochastic effects involving high inhomogeneity with regard to the localization of molecules and the clusters they may form in addition may require single-particle representations (feature (f)). However, many biological processes involve molecular concentrations that permit neglecting the underlying stochasticity because the stochastic effects average out at high particle numbers. Since stochastic simulations are typically orders of magnitude slower than deterministic simulations (see Supplementary Note 3b), simulation tools can, in those cases, take advantage of the ability to perform deterministic simulations (feature (g)).

For large models that contain many molecule types with interactions depending on specific states of these molecules and involving reaction-induced state changes, even just writing down the definition of all the interactions can be very time consuming and error prone. In those cases a graphical user interface (GUI) (feature (i)) can be helpful since it avoids errors that may occur when interaction rules are formulated as computer scripts and because graphical representations of molecules and their interactions are more intuitive than textual code.

Supplementary Note 2

a) Generating reaction networks based on local molecular interaction possibilities

Local reaction networks are built incrementally by assembling complexes with increasing numbers of molecular components based on the user-defined interaction possibilities between molecules or multi-molecular complexes. The interaction possibilities (rules) defined by the user specify (1) the interacting molecules (or complexes), (2) the states the interaction partners are in when performing the interaction and (3) the states the resulting complex(es) are in after the interaction. The states of the complexes can encode properties such as phosphorylation, steric conformations, etc. A prototypical example of a binding-induced state change is the activation of a receptor's intracellular domain as a result of the interaction of its extracellular domain with its ligand. The interaction between the receptor and the ligand, in turn, may be modulated by the state of the receptor. Intracellular ligation ('inside-out signaling') or phosphorylation may have an influence on the kinetics of the receptor ectodomain-ligand interaction. In this case, multiple rules for the different initial states of the receptor have to be specified. Once all the interaction rules for a system of interacting molecules and complexes have been specified the resulting complexes can be generated. Ligand-bound receptors (two molecular components: receptor and ligand), for instance, may bind to intracellular adaptors thereby generating three component complexes that then could 'recruit' additional binding partners.

One fundamental challenge for algorithms implementing such incremental complex building cascades is the unique identification of the complexes. The problem arises from the fact that complexes frequently can be generated in multiple ways. For instance, the

adaptor:receptor:ligand complex could in principle (unless the interaction rules suppress one or the other possibility) be built by the receptor:ligand complex binding to the adaptor or by the adaptor:receptor complex binding to the ligand. If the results of these two reactions are equivalent in terms of binding structure and state, the algorithm building the complexes has to be able to identify them as being equivalent and assign to both reactions the same result complex before continuing to build larger complexes involving the adaptor:receptor:ligand complex. For situations in which components of multi-molecular complexes differ not with regard to their biochemical identity but only with regard to their location of origin, this task is more difficult than for reaction networks that do not take into account spatial aspects. Consider the E-cadherin example discussed in the main text: here, the interaction partners all have the same biochemical identity: 'E-cadherin'. However, for the simulation of the reaction kinetics and the associated concentration changes in two adjacent membrane regions with inter-cellular complexes, it is important to keep track of the location of the components. Consider, for instance, the E-cadherin trimer consisting of two *trans*-bound monomers with one additionally *cis*-bound to another E-cadherin located in the same membrane for which the creation process is discussed in **Fig. 3** of the main text. When this trimer decays – whether along a *trans* or a *cis* interaction – the resulting concentration changes for the two membranes involved depend on in which of the two membranes the two *cis*-bound molecules of the trimer are located. A *trans* dissociation will increase the concentration of *cis*-dimers in that membrane, whereas the other membrane will experience an increase in the concentration of its monomers.
After all complex reactions in the volume and membrane elements have been identified in this way, the simulation can use this information to determine how the local concentrations of the different complexes in the volume element change over time by constructing the equivalents of an ODE right-hand side and Jacobian.

b) System of differential equations describing E-cadherin complex formation for two adjacent locations

To illustrate the complexity arising from the simple case of two adjacent membrane locations with E-cadherin, the molecule species on the two sides are called E1 and E2, respectively. This makes it possible to create a 'human-readable' representation of the differential equations describing the reactions resulting from *cis* and *trans* associations between E-cadherin monomers and multimers. In this example, E-cadherin has only one interface for *trans* and one for *cis* interactions. Therefore, the system does not include complexes such as *cis* trimers.

E1 and E2 (equations (1) and (2)) are E-cadherin monomers, E1E1 and E2E2 are *cis* dimers, E1E2 is a *trans* dimer, E1E1E2 consists of an E1 *cis* dimer *trans*-bound to E2 (E1E2E2 vice versa). $\{E_1E_2E_1E_2\}_V^{1-1}$ corresponds to complex *V* from **Fig. 2b** with the closed *cis* bond on the 1-side. Complexes *VI* and *VII* are depicted in **Fig. 2b** as well. Note that, in contrast to the simulated models in the main text (that permit the formation of *cis* bonds only within complexes possessing *trans* bonds), the reactions below incorporate the direct formation of *cis* dimers for didactical purposes.

It should be emphasized that the simple strategy of hand-assigning indices (or labels) to molecules or complexes in order to encode their spatial location quickly becomes unwieldy in cases with many interacting membrane elements and/or cells and impossible in cases that involve dynamic self-interactions of membranes (for instance in models of endocytosis).

$$\begin{aligned}\frac{dE_1}{dt} &= -k_{on}^{cis}\cdot E_1\cdot\left(2\cdot E_1 + E_1E_2 + E_1E_2E_2\right) + k_{off}^{cis}\cdot\left(2\cdot E_1E_1 + E_1E_1E_2 + \left\{E_1E_2E_1E_2\right\}_{VI}\right)\\ &\quad -k_{on}^{trans}\cdot E_1\cdot\left(E_2 + 2\cdot E_2E_2 + E_1E_2E_2\right) + k_{off}^{trans}\left(E_1E_2 + E_1E_2E_2 + 2\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{2-2}\right)\end{aligned} \quad (1)$$

$$\begin{aligned}\frac{dE_2}{dt} &= -k_{on}^{cis}\cdot E_2\cdot\left(2\cdot E_2 + E_1E_2 + E_1E_1E_2\right) + k_{off}^{cis}\cdot\left(2\cdot E_2E_2 + E_1E_2E_2 + \left\{E_1E_2E_1E_2\right\}_{VI}\right)\\ &\quad -k_{on}^{trans}\cdot E_2\cdot\left(E_1 + 2\cdot E_1E_1 + E_1E_1E_2\right) + k_{off}^{trans}\left(E_1E_2 + E_1E_1E_2 + 2\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{1-1}\right)\end{aligned} \quad (2)$$

$$\begin{aligned}\frac{dE_1E_1}{dt} &= k_{on}^{cis}\cdot E_1\cdot E_1 - k_{off}^{cis}\cdot E_1E_1 - 2\cdot k_{on}^{trans}\cdot E_1E_1\cdot\left(E_2 + 2\cdot E_2E_2\right)\\ &\quad + k_{off}^{trans}\cdot\left(E_1E_1E_2 + \left\{E_1E_2E_1E_2\right\}_{VI}\right)\end{aligned} \quad (3)$$

$$\begin{aligned}\frac{dE_2E_2}{dt} &= k_{on}^{cis}\cdot E_2\cdot E_2 - k_{off}^{cis}\cdot E_2E_2 - 2\cdot k_{on}^{trans}\cdot E_2E_2\cdot\left(E_1 + 2\cdot E_1E_1\right)\\ &\quad + k_{off}^{trans}\cdot\left(E_1E_2E_2 + \left\{E_1E_2E_1E_2\right\}_{VI}\right)\end{aligned} \quad (4)$$

$$\begin{aligned}\frac{dE_1E_2}{dt} &= -k_{on}^{cis}\cdot E_1E_2\cdot\left(E_1 + E_2 + 2\cdot E_1E_2\right) + k_{off}^{cis}\cdot E_1E_1E_2 + k_{off}^{cis}\cdot E_1E_2E_2\\ &\quad + 2\cdot k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{1-1} + 2\cdot k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{2-2} + k_{on}^{trans}\cdot E_1\cdot E_2\\ &\quad - k_{off}^{trans}\cdot E_1E_2\end{aligned} \quad (5)$$

$$\begin{aligned}\frac{dE_1E_1E_2}{dt} &= k_{on}^{cis}\cdot E_1\cdot E_1E_2 - k_{on}^{cis}\cdot E_2\cdot E_1E_1E_2 + k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{VI} - k_{off}^{cis}\cdot E_1E_1E_2\\ &\quad + 2\cdot k_{on}^{trans}\cdot E_1E_1\cdot E_2 - k_{on}^{trans}\cdot E_1E_1E_2\cdot E_2 + 2\cdot k_{off}^{trans}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{1-1}\\ &\quad - k_{off}^{trans}\cdot E_1E_1E_2\end{aligned} \quad (6)$$

$$\frac{dE_1E_2E_2}{dt} = k_{on}^{cis}\cdot E_1E_2\cdot E_2 - k_{on}^{cis}\cdot E_1\cdot E_1E_2E_2 + k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{VI} - k_{off}^{cis}\cdot E_1E_2E_2$$
$$+2\cdot k_{on}^{trans}\cdot E_1\cdot E_2E_2 - k_{on}^{trans}E_1\cdot E_1E_2E_2 + 2\cdot k_{off}^{trans}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{2-2} \quad (7)$$
$$-k_{off}^{trans}\cdot E_1E_2E_2$$

$$\frac{d\left\{E_1E_2E_1E_2\right\}_{V}^{1-1}}{dt} = k_{on}^{cis}\cdot E_1E_2\cdot E_1E_2 - k_{on}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{1-1} + k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{VII} \quad (8)$$
$$-k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{1-1} + k_{on}^{trans}\cdot E_1E_1E_2\cdot E_2 - 2\cdot k_{off}^{trans}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{1-1}$$

$$\frac{d\left\{E_1E_2E_1E_2\right\}_{V}^{2-2}}{dt} = k_{on}^{cis}\cdot E_1E_2\cdot E_1E_2 - k_{on}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{2-2} + k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{VII} \quad (9)$$
$$-k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{2-2} + k_{on}^{trans}\cdot E_1E_2E_2\cdot E_1 - 2\cdot k_{off}^{trans}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{2-2}$$

$$\frac{d\left\{E_1E_2E_1E_2\right\}_{VI}}{dt} = k_{on}^{cis}\cdot E_1\cdot E_1E_2E_2 + k_{on}^{cis}\cdot E_2\cdot E_1E_1E_2 - 2\cdot k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{VI}$$
$$+4\cdot k_{on}^{trans}E_1E_1\cdot E_2E_2 - k_{on}^{trans}\cdot\left\{E_1E_2E_1E_2\right\}_{VI} + 2\cdot k_{off}^{trans}\left\{E_1E_2E_1E_2\right\}_{VII} \quad (10)$$
$$-k_{off}^{trans}\left\{E_1E_2E_1E_2\right\}_{VI}$$

$$\frac{d\left\{E_1E_2E_1E_2\right\}_{VII}}{dt} = k_{on}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{1-1} + k_{on}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{V}^{2-2} - 2\cdot k_{off}^{cis}\cdot\left\{E_1E_2E_1E_2\right\}_{VII} \quad (11)$$
$$+k_{on}^{trans}\left\{E_1E_2E_1E_2\right\}_{VI} - 2\cdot k_{off}^{trans}\left\{E_1E_2E_1E_2\right\}_{VII}$$

Adding diffusion terms to equation (9) would yield:

$$\frac{d\left\{E_1^iE_2^iE_1^iE_2^i\right\}_{V}^{2-2}}{dt} = k_{on}^{cis}\cdot E_1^iE_2^i\cdot E_1^iE_2^i - k_{on}^{cis}\cdot\left\{E_1^iE_2^iE_1^iE_2^i\right\}_{V}^{2-2} + k_{off}^{cis}\cdot\left\{E_1^iE_2^iE_1^iE_2^i\right\}_{VII}$$
$$-k_{off}^{cis}\cdot\left\{E_1^iE_2^iE_1^iE_2^i\right\}_{V}^{2-2} + k_{on}^{trans}\cdot E_1^iE_2^iE_2^i\cdot E_1^i - 2\cdot k_{off}^{trans}\cdot\left\{E_1^iE_2^iE_1^iE_2^i\right\}_{V}^{2-2}$$
$$-D^*\cdot\left(N\cdot\left\{E_1^iE_2^iE_1^iE_2^i\right\}_{V}^{2-2} - \sum_{j}^{\substack{N(\#\,neighbor\\ space-elements)}}\left\{E_1^jE_2^jE_1^jE_2^j\right\}_{V}^{2-2}\right)$$

Below is the output generated by the simulator application in response to a user request to provide information about the local reaction networks at the interface of two cells expressing either E1 or E2 from the reaction system specified above.
The application first lists all complexes with information about the binding state of their molecular constituents. The selected membrane element belongs to the cell expressing E1. For the 'Trans Dimer' E1 is therefore listed as a molecule with location 'on membrane' and bound via its first binding site to E2 'on contact membrane'. The binding code '**0:0:1/_:_:_**' for E1 specifies that the binding that involves the first binding site binds to binding site 0 (= first binding site, indices start at 0) of component 0 of molecule 1 in the complex (E2 is molecule 1 in the complex; E1 is molecule 0). The part of the binding code describing '**_:_:_**' the binding for the second binding site simply states that this site is not bound in the complex.
After the complexes are listed and numbered the reactions are described using these numbers to identify the complexes. For reactions that involve reaction partners that are entirely located on the contact membrane a brief structure description is provided instead of a number. The simulator lists reactions with the numerical value of the reaction rates. To facilitate reading the automated output the following rates were used <u>for this illustration only</u>:

$$k_{on}^{trans} = 1; k_{on}^{cis}(E1E1) = 2; k_{on}^{trans}(\text{intra}) = 3; k_{on}^{cis}(E1E1,\text{intra}) = 4; k_{on}^{cis} = 5; k_{on}^{cis}(E2E2,\text{intra}) = 6;$$

$$k_{off}^{trans} = 7; k_{off}^{cis}(E1E1) = 8; k_{off}^{trans}(\text{intra}) = 9; k_{off}^{cis}(E1E1,\text{intra}) = 10; k_{off}^{cis} = 11; k_{off}^{cis}(E2E2,\text{intra}) = 12;$$

```
************************
*** Membrane network: ***
************************

**********
Complexes:
**********
(0):  Cis  Dimer  1{E1<>(on  membrane)/_:_:_/1:0:1  |  E1<>(on
membrane)/_:_:_/1:0:0}
concentration: 0.000103797 per square micron.
(1): E1{E1<>(on membrane)/_:_:_/_:_:_}
concentration: 50.0069 per square micron.
(2):  Trans  Dimer{E1<>(on  membrane)/0:0:1/_:_:_  |  E2<>(on  contact
membrane)/0:0:0/_:_:_}
concentration: 1.97712e-05 per square micron.
(3): {E1<>(on membrane)/0:0:2/_:_:_ | E1<>(on membrane)/0:0:3/_:_:_ |
E2<>(on  contact  membrane)/0:0:0/1:0:3  |  E2<>(on  contact
membrane)/0:0:1/1:0:2}
concentration: 1.22622e-10 per square micron.
(4):  {E1<>(on  membrane)/0:0:2/_:_:_  |  E2<>(on  contact
membrane)/_:_:_/1:0:2 | E2<>(on contact membrane)/0:0:0/1:0:1}
concentration: 0 per square micron.
(5): {E1<>(on membrane)/0:0:3/1:0:1 | E1<>(on membrane)/0:0:2/1:0:0 |
E2<>(on  contact  membrane)/0:0:1/1:0:3  |  E2<>(on  contact
membrane)/0:0:0/1:0:2}
concentration: 1.50264e-12 per square micron.
(6): {E1<>(on membrane)/0:0:3/1:0:1 | E1<>(on membrane)/0:0:2/1:0:0 |
E2<>(on  contact  membrane)/0:0:1/_:_:_  |  E2<>(on  contact
membrane)/0:0:0/_:_:_}
```

```
concentration: 0 per square micron.
(7): {E1<>(on membrane)/_:_:_/1:0:1 | E1<>(on membrane)/0:0:2/1:0:0 |
E2<>(on contact membrane)/0:0:1/_:_:_}
concentration: 1.43465e-10 per square micron.
(8): {E1<>(on membrane)/_:_:_/1:0:1 | E1<>(on membrane)/0:0:3/1:0:0 |
E2<>(on contact membrane)/_:_:_/1:0:3 | E2<>(on contact
membrane)/0:0:1/1:0:2}
concentration: 0 per square micron.
```

In the following, we use a color code to map between the differential equation for the concentration of the Ecad monomer (E_1) on the first membrane and the output generated by the simulator application.

$$\frac{dE_1}{dt} = -k_{on}^{cis} \cdot E_1 \cdot \left(2 \cdot E_1 + E_1E_2 + E_1E_2E_2\right) + k_{off}^{cis} \cdot \left(2 \cdot E_1E_1 + E_1E_1E_2 + \{E_1E_2E_1E_2\}_{VI}\right)$$
$$-k_{on}^{trans} \cdot E_1 \cdot \left(E_2 + 2 \cdot E_2E_2 + E_1E_2E_2\right) + k_{off}^{trans} \left(E_1E_2 + E_1E_2E_2 + 2 \cdot \{E_1E_2E_1E_2\}_{V}^{2-2}\right)$$

```
**********
Reactions:
**********
(0): Cis Dimer 1{E1<>(on membrane)/_:_:_/1:0:1 | E1<>(on
membrane)/_:_:_/1:0:0}
concentration: 0.000103797 per square micron.
binds to complex on contact membrane: E2{E2<>} to form (7). Rate: 1.
(7) shared with contact membrane.
binds to complex on contact membrane: E2{E2<>} to form (7). Rate: 1.
(7) shared with contact membrane.
binds to complex on contact membrane: Cis Dimer 2{E2<>|E2<>} to form
(8). Rate: 1. (8) shared with contact membrane.
binds to complex on contact membrane: Cis Dimer 2{E2<>|E2<>} to form
(8). Rate: 1. (8) shared with contact membrane.
binds to complex on contact membrane: Cis Dimer 2{E2<>|E2<>} to form
(8). Rate: 1. (8) shared with contact membrane.
binds to complex on contact membrane: Cis Dimer 2{E2<>|E2<>} to form
(8). Rate: 1. (8) shared with contact membrane.
decay into: (1) and (1). Rate: 8

(1): E1{E1<>(on membrane)/_:_:_/_:_:_}
concentration: 50.0069 per square micron.
binds to: (1) to form (0). Rate: 2
binds to: (2) to form (7). Rate: 2. Complexes (2) and (7) shared with
contact membrane.
binds to: (4) to form (3). Rate: 1. Complexes (4) and (3) shared with
contact membrane.
binds to: (4) to form (8). Rate: 2. Complexes (4) and (8) shared with
contact membrane.
binds to complex on contact membrane: E2{E2<>} to form (2). Rate: 1.
(2) shared with contact membrane.
binds to complex on contact membrane: Cis Dimer 2{E2<>|E2<>} to form
(4). Rate: 1. (4) shared with contact membrane.
binds to complex on contact membrane: Cis Dimer 2{E2<>|E2<>} to form
(4). Rate: 1. (4) shared with contact membrane.

(2): Trans Dimer{E1<>(on membrane)/0:0:1/_:_:_ | E2<>(on contact
membrane)/0:0:0/_:_:_}
```

concentration: 1.97712e-05 per square micron.
binds to: (1) to form (7). Rate: 2. Complexes (2) and (7) shared with contact membrane.
binds to: (2) to form (6). Rate: 2. All complexes shared with contact membrane.
binds to: (2) to form (3). Rate: 5. All complexes shared with contact membrane.
binds to complex on contact membrane: E2{E2<>} to form (4). Rate: 5. Startcomplex and (4) shared with contact membrane.
decay into: (1) and complex on contact membrane: E2{E2<>} rate: 7

(3): {E1<>(on membrane)/0:0:2/_:_:_ | E1<>(on membrane)/0:0:3/_:_:_ | E2<>(on contact membrane)/0:0:0/1:0:3 | E2<>(on contact membrane)/0:0:1/1:0:2}
concentration: 1.22622e-10 per square micron.
transforms into: (5). Rate: 4. Start- and Resultcomplex shared with contact membrane.
decay into: (2) and (2) rate: 11. All complexes shared with contact membrane.
decay into: (4) and (1) rate: 7. The startcomplex and (4) shared with contact membrane.
decay into: (4) and (1) rate: 7. The startcomplex and (4) shared with contact membrane.

(4): {E1<>(on membrane)/0:0:2/_:_:_ | E2<>(on contact membrane)/_:_:_/1:0:2 | E2<>(on contact membrane)/0:0:0/1:0:1}
concentration: 0 per square micron.
binds to: (1) to form (3). Rate: 1. Complexes (4) and (3) shared with contact membrane.
binds to: (1) to form (8). Rate: 2. Complexes (4) and (8) shared with contact membrane.
decay into: (1) and complex on contact membrane: Cis Dimer 2{E2<>|E2<>} rate: 7
decay into: (2) and complex on contact membrane: E2{E2<>}. Rate: 11. The startcomplex and (2) shared with contact membrane.

(5): {E1<>(on membrane)/0:0:3/1:0:1 | E1<>(on membrane)/0:0:2/1:0:0 | E2<>(on contact membrane)/0:0:1/1:0:3 | E2<>(on contact membrane)/0:0:0/1:0:2}
concentration: 1.50264e-12 per square micron.
transforms into: (8). Rate: 9. Start- and Resultcomplex shared with contact membrane.
transforms into: (3). Rate: 10. Start- and Resultcomplex shared with contact membrane.
transforms into: (8). Rate: 9. Start- and Resultcomplex shared with contact membrane.
transforms into: (6). Rate: 12. Start- and Resultcomplex shared with contact membrane.

(6): {E1<>(on membrane)/0:0:3/1:0:1 | E1<>(on membrane)/0:0:2/1:0:0 | E2<>(on contact membrane)/0:0:1/_:_:_ | E2<>(on contact membrane)/0:0:0/_:_:_}
concentration: 0 per square micron.
transforms into: (5). Rate: 6. Start- and Resultcomplex shared with contact membrane.
decay into: (2) and (2) rate: 8. All complexes shared with contact membrane.

decay into: (7) and complex on contact membrane: E2{E2<>}. Rate: 7. The startcomplex and (7) shared with contact membrane.
decay into: (7) and complex on contact membrane: E2{E2<>}. Rate: 7. The startcomplex and (7) shared with contact membrane.

(7): {E1<>(on membrane)/_:_:_/1:0:1 | E1<>(on membrane)/0:0:2/1:0:0 | E2<>(on contact membrane)/0:0:1/_:_:_}
concentration: 1.43465e-10 per square micron.
binds to complex on contact membrane: E2{E2<>} to form (6). Rate: 1. Startcomplex and (6) shared with contact membrane.
binds to complex on contact membrane: E2{E2<>} to form (8). Rate: 5. Startcomplex and (8) shared with contact membrane.
decay into: (0) and complex on contact membrane: E2{E2<>} rate: 7
decay into: (2) and (1) rate: 8. The startcomplex and (2) shared with contact membrane.

(8): {E1<>(on membrane)/_:_:_/1:0:1 | E1<>(on membrane)/0:0:3/1:0:0 | E2<>(on contact membrane)/_:_:_/1:0:3 | E2<>(on contact membrane)/0:0:1/1:0:2}
concentration: 0 per square micron.
transforms into: (5). Rate: 3. Start- and Resultcomplex shared with contact membrane.
decay into: (0) and complex on contact membrane: Cis Dimer 2{E2<>|E2<>} rate: 7
decay into: (4) and (1) rate: 8. The startcomplex and (4) shared with contact membrane.
decay into: (7) and complex on contact membrane: E2{E2<>}. Rate: 11. The startcomplex and (7) shared with contact membrane.

Supplementary Note 3

a) Technical overview: How Simmune 2.0 operates; comparison with other approaches

Note: Reading this note is not necessary for using Simmune. It provides 'under the hood' technical details.

Defining and simulating a model with Simmune involves four steps:

(1) Defining the molecular interactions that will determine the network topology and dynamics of the biochemistry in the simulated system.
(2) Defining the geometry (shape and size of the cells and extracellular space) of the simulated system.
(3) Defining the initial molecular concentrations and their spatial distributions and the mechanisms (Potts rules) determining the morphological dynamics of cells.
(4) Running the simulation.

(1) Defining the molecular interactions

Instead of using a scripting language (such as BNGL[1] or Kappa[3]), the Simmune Modeler has a graphical interface that represents molecules, molecular components, and binding sites as well as the multi-molecular complexes resulting from associations between molecules as iconographic symbols. Interactions between molecular binding sites can be defined by drawing connections between the sites and then specifying the conditions for the interactions and specifying the modifications of the molecules that result from the interaction. These steps are explained in great detail in the tutorials provided in the supplementary material. The graphical interface was chosen instead of a scripting approach because of its ease of use and because capturing spatial aspects of molecular interactions requires a greater spectrum of properties (such as the orientation of molecular components relative to membranes) than existing languages for 'rule-based' modeling currently offer. The resulting models are stored in an SQL (Structured Query Language) database to allow for flexible and efficient organization and for detailed queries about molecular properties and interactions. The database communicates with the simulation software through an API (Application Programming Interface) that can directly be accessed by computational modelers/programmers who want to implement their own simulation engines, or supply programmatically generated models to the simulator. Model databases are stored as SQLite dbf files that can easily exchanged between researchers.
Currently, Simmune does not support reactions that involve *de novo* creation of molecules or their destruction. In principle, transformations – into a 'degraded' state – could be used as a work-around. But we will add functionality for 0^{th} order reaction in the very near future.

(2) Defining the geometry of the simulated system

The geometry of the simulated system (shape of the cells, intracellular organelles, extracellular space) can be defined independently of specification of the molecular interactions. This means that a given biochemical model can be used with different geometries and different models can be simulated on the same geometry. We provide a tool, called 'SimCelldesigner', with a graphical user interface that can be used to specify geometries. It is most suitable for geometries of simple to medium complexity, including compartments containing multiple interacting cells. The cell design process is based on the concept of 'blobs' or 'meta-balls' (see: James F. Blinn. 1982. A Generalization of Algebraic Surface Drawing. ACM Trans. Graph. 1, 3 (July 1982), 235-256. http://doi.acm.org/10.1145/357306.357310). Each blob or meta-ball is a source of a spherical building block (or a sphere of influence) that can add its influence field to the final shape consisting of the sum of all building blocks with a smooth transition between blocks.

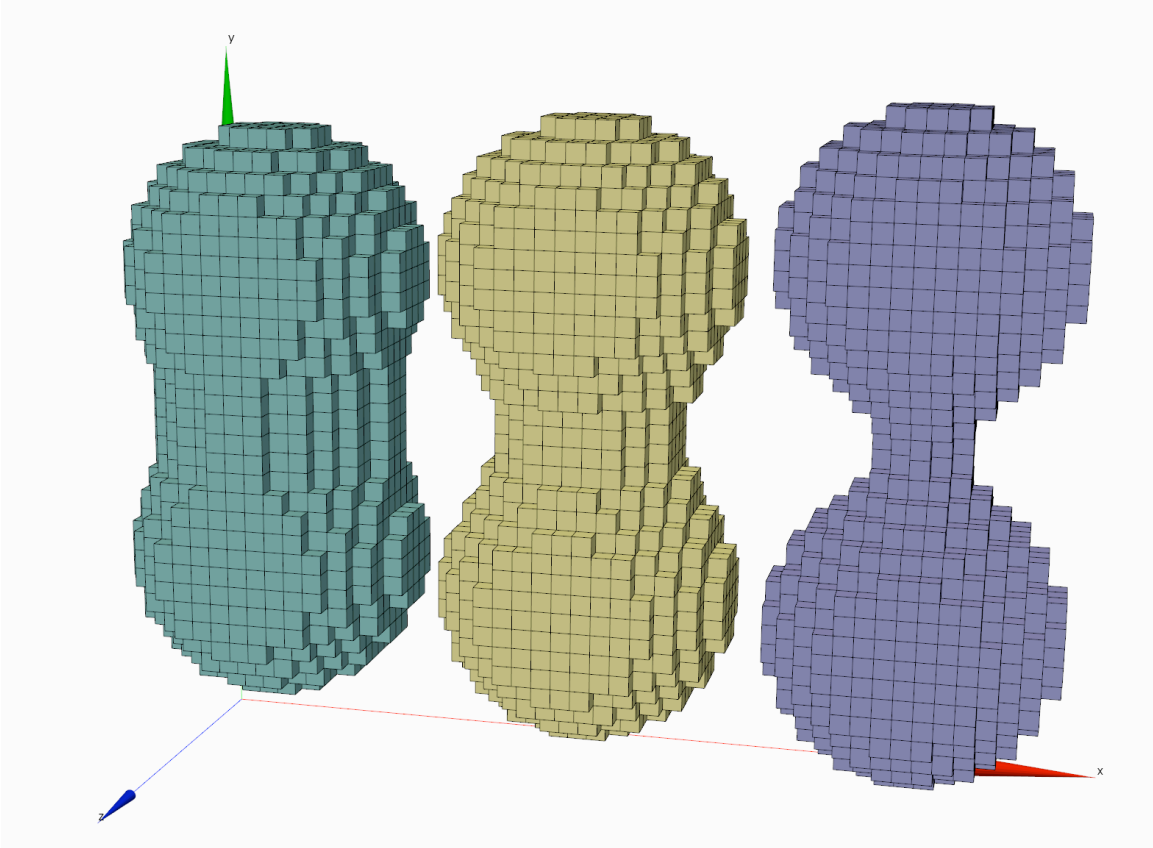


Illustration of the meta-ball concept. Each of the colored objects is composed of two meta-balls. The distance between the meta-balls within one object increases from left to right. Note the smooth transition between the meta-balls.

In biological terms, a collection of meta-balls could be a cell or an intracellular organelle. Each meta-ball by itself could be interpreted as a protrusion of a cell. Geometries are stored in an XML-based format that can be imported by the simulation GUI. More complex geometries based on meta-balls can be created by using the API provided by the cell designer. Furthermore a low-level programmatic access to individual volume elements is possible through the API of the simulator.

(3) Defining the initial molecular concentrations and their spatial distributions and the mechanisms (Potts rules) determining the morphological dynamics of cells.

When the simulator starts it prompts the user for a molecular interaction model definition (generated in step 1) and a geometry (generated in step 2). Based on the specifications provided by those two components, the simulator interface allows the user to specify initial molecular concentrations and their spatial distribution over the geometry of the simulated system. For instance, membrane-bound molecules can be inserted into cellular or subcellular membranes with spatially inhomogeneous concentrations. The simulation startup windows also permit the specification of rules for the implementation of Potts model morphological dynamics. These rules determine how receptor signals influence the shape changes the cells experience during a simulation. Details can be found in the Methods section.
The specified initial concentrations and Potts rules can be saved so the simulation can easily be restarted with the same initial conditions.

(4) Running the simulation

After the initial conditions for the simulation are specified, the simulation GUI appears. Its functionality is explained in detail in the tutorial. Here, we focus on the technical aspects of the operation of the simulator.
Based on the molecular interaction model definition and the molecular complexes used in the specification of the initial biochemistry the simulator first generates a 'template network' that does not yet incorporate spatial aspects. The generation of this network is similar to the operation of other automated network generators such as implemented in BioNetGen[1], Kappa[3] or Moleculizer[6]. Using the example of the Ecad network from the main text, the input specifications from step 1 correspond to **Fig. 2a** and the template network corresponds to the non-spatial network depicted in **Fig. 2b**. However, in contrast to approaches that do not simulate spatial aspects through differential equations, the network generator here has to keep track of how interacting molecules become mapped onto resulting complexes. This information is needed, for example, for the simulation of adjacent membranes with interacting receptors; when the local reaction networks are generated the components of multi-molecular complexes linking two adjacent membranes have to carry with them information that permits their redistribution back onto the membranes hosting them when the dissociation process is configured. This renders the network generation more expensive than for non-spatial purposes. In spite of this considerable extra cost, the network generation of Simmune compares favorably with other approaches (see **Supplementary Note 3b**, below).
When types of molecular complexes are added to the simulation – for instance as extracellular stimuli – that were not part of the initial biochemistry, the 'template network' is expanded accordingly while the simulation is running. Then the local networks extract the new information from the global 'template network' and adjust themselves to the new reaction possibilities. This may sometimes cause a noticeable delay in the response of the system after a new molecular complex has been added to a simulated system. **Supplementary Fig. 6a** gives a rough estimate of the delay: networks in systems with 5 phosphorylation sites consist of $4^5 + 2 = 1026$ complexes required less

than a second building time on our test computer whereas the next larger network (4098 complexes) would cause the simulation to be unresponsive for a couple of seconds.
Based on the geometry specifications generated in step 2, the simulated space is divided into volume elements (currently, only cubic volume elements are supported). The total number of volume elements can be as low as 1 (no spatial resolution) or as high as 100x100x100 (the ratio of the side lengths does not need to be 1: 1x100x30 would be possible as well), with the upper limit essentially depending on the biochemical complexity of the simulated model and the performance and memory (RAM) of the computer used for the simulation. Within this spatial discretization the user-defined structures are generated (cells, intracellular domains, extracellular space and the membranes separating them).

Using these structures, the initial molecule distributions, as specified by the user, are applied. The initial molecule distributions assign a set (that can be empty for some volume and membrane elements) of molecular complexes with specific concentrations to each volume element and membrane element. Each volume/membrane element then selects the reaction patterns of the template network that can be applied to its biochemistry based on the molecular complexes that it hosts. For instance, a volume element will not select any reactions that involve membrane-bound species whereas a membrane element will not generate reaction that can only take place between non-membrane-bound molecular complexes. The selected reaction patterns will then be used to generate local networks in each volume/membrane element that take into account the spatial aspects of the reactions. For volume elements, these aspects simply involve diffusional exchange between neighboring elements. This means that a volume element in contact with another element that contains molecular complexes that differ from its own biochemistry will also generate representations of those complexes to be able to simulate the result of diffusion from the neighbor element into its own space – and vice versa. Diffusional exchange between neighbor volume or membrane elements is implemented as Finite Volume scheme. Note that cubic discretizations of the Laplacian on cell surfaces will lead to severe artifacts unless locally calculated adaptive membrane curvatures are employed. This is discussed further below whereas Supplementary Note 8 also validates the precision of the bulk diffusion for various resolutions of the simple cubic spatial discretization performed by Simmune for intracellular regions.

For membrane elements, the generation of the local reaction-diffusion networks is much more complicated than for volume elements because membrane elements can be in contact with other membrane elements, leading to the formation of inter-membrane complexes. The challenges resulting from such structures are discussed in **Fig. 3** and **Supplementary Note 2a**: inter-membrane complexes are no longer uniquely specified by their composition in terms of component molecules. Note that a simple tagging approach (for two interacting cells one cell receives tag *A* while the other receives tag *B*) that for a static situation could in principle be implemented using non-spatial network generators is not sufficient since a cell could have self-interactions, for instance through adhesion receptors as part of a model for endocytosis.

The generation of these local networks is computationally rather expensive since they are typically far more complicated than non-spatial networks. This is illustrated, for the simple Ecad network, in the comparison of **Fig. 2b** and **2c** and discussed for one complex in **Fig. 3**. To render this process more efficient local networks, once generated, are stored in buffers in a manner that allows the software to re-implement them in locations that have the same biochemistry. Without these buffers, generating all local networks for a spatially highly-resolved cell with non-trivial biochemistry would take a very long time. This would effectively render simulations with dynamic morphology impossible.
The local networks are coded in a way that optimizes the generation of 'right-hand side' (RHS) and the product (JV) of a concentration test vector with the Jacobian of the total system. Each molecule complex has a list of reaction channel descriptors corresponding to the various interactions or transformations the complex can participate in based on its local biochemical environment. These descriptors contain direct pointers to the concentration entries of all complexes affected by the respective reactions. Note that this is non-trivial for reactions involving trans-membrane complexes since such reactions affect concentrations in multiple spatial regions (see **Supplementary Note 2a**).

(i) Associations 'know' which other complex concentration they have to be multiplied with to render a concentration change entry and have pointers to the concentration entries of reactants and the product.
(ii) Dissociations have pointers to the concentration entries of their fragment complexes.
(iii) Transformations have pointers to the concentration entries of the result complexes.

Such data structures are somewhat expensive to generate and expensive to adjust when the biochemical compositions of the cells change, for instance when adjacent membrane elements with trans-membrane inter-cellular complexes are separated and these complexes have to be removed from the local biochemistry. But they permit very rapid construction of the RHS and JV arrays that are repeatedly needed by the integrator for its internal time steps.

The rates of association reactions that happen at membranes can be provided as 3D mass-action rates in L/(mol*s) (in which case Simmune calculates an effective reaction volume using a reaction volume depth of 10 nanometers) or as 2D reaction rates in units (micron^2*s)$^{-1}$. Users who wish to use 3D rates but with different reaction volume depth can implement those by adjusting the interaction rate.

Simmune uses the matrix-free (no explicit Jacobian) solver ROWMAP, a Rosenbrock-type code of order 4 with Krylov techniques for large stiff ODEs[15]. The code has proven efficient for stiff systems with more than 10 million degrees of freedom, allowing us to put all local concentrations into a single concentration array even for systems with several interacting cells with medium spatial resolution and non-trivial biochemistry. A GPU parallelization is in preparation. Due to the memory constraints of GPU cards this will, however, mainly increase the performance, not the maximally possible size (degrees of freedom) of simulated systems. The overall structure of the simulation loop is shown in **Supplementary Fig. 2**.

Technically the most challenging situation is the adaptation of the local reaction networks to changing cellular morphologies. Several aspects have to be taken into account:

(i) redistribution of molecular contents when cellular volume elements disappear (the cell shrinks locally);
(ii) redistribution of molecular contents when cellular volume elements are added (the cell expands locally);
(iii) when membrane elements that are in contact with other membrane elements are retracted (removed): breaking of membrane-membrane receptor bonds and redistribution of the fragments over the neighbor membrane elements;
(iv) generation of updated local reaction networks.

Points (i) and (ii) are mainly a question of the assumed time scale of the changes in cellular morphology relative to the rates of diffusional exchange or active molecular transport. For instance, when biological cells produce lamellipodial or filopodial protrusions the newly extended membrane regions may become populated with receptor complexes that diffuse in from the membrane regions surrounding the base of the protrusions or they may come from sub-membrane reservoirs that allow the cell to retain a more or less homogeneous membrane biochemistry in the face of morphological changes, or they may be brought into the protrusions through active transports. The exact mechanisms will depend on the type of cell and the nature of the protrusions. For most biological phenomena delineating these mechanisms is still an open problem. Choosing the appropriate mechanism in a simulation is thus part of the biological model being investigated and, consequently, user specifiable in Simmune. The available options are explained in detail below in section c). It should be noted that the examples we provide here update morphology and cellular biochemistry separately: when a morphological update has been performed, the reaction networks are adapted and the simulation continues. If, for instance, the chosen redistribution mode ((i)/(ii) above) leaves new volume elements empty, they will subsequently be filled by diffusion. The generation of updated local reaction networks (point (iv), above) proceeds similarly to the *de novo* generation of networks but with increased efficiency thanks to the network buffers.

Point (iii) is somewhat subtle since the most straightforward method of simply breaking inter-membrane bonds and putting the molecular components of membrane-membrane receptor complexes back into the membranes that host them may introduce unwanted side effects. Consider, for instance, a receptor that undergoes a conformational transformation when binding to a receptor on an adjacent membrane and loses this conformation (returns to the previous state) when the binding is lost. Just cutting the bond and leaving the receptor in the 'bound' conformation would create an orphan receptor in a 'bound' state – in most cases this would be an unwanted effect. Therefore, membrane-membrane receptor complexes are dissolved along the user-specified decay reactions when Simmune breaks a membrane-membrane contact. Note that models that do, in fact, want to include receptors that keep the memory of the bound state after a contact has been broken can define state transformations that are not reverted through dissociation.

Simmune offers user defined tolerances to permit rapid, low precision simulations to explore the behavior of large systems without having to pay with small time steps for high precision. Relative and absolute error tolerances, *rtol* and *atol*, are based on ROWMAP's tolerances: ROWMAP uses a weighted root-mean-square norm to measure the size of error vectors. It is defined by wrms=sqrt(sum[(err(i)/scal(i))^2, i=1,n]/n), where scal(i)=atol+rtol*abs(u(i)) and err(i) is the error vector, u(i) the solution and i the global index counting the concentration entries of the entire simulated system.

b) Performance comparisons for network generation and simulation

Assessment of Simmune's speed/efficiency

As an ODE-based simulation tool, Simmune has to generate the reaction network of a modeled system before integrating the equations that determine its time evolution. Since no other tool currently can automatically generate networks with spatial resolution we compared the performance of the first, non-spatial network generation step of Simmune with that of the network generators of other approaches. We used the multi-site phosphorylation model suggested by Sneddon et al.[7] as a benchmark to compare Simmune's performance in this regard with BioNetGen. A comparison of the network generation step with Nfsim cannot be performed, since NFSim generates reactions between molecular complexes 'on-demand' as they occur within the particle-based simulation.

We rebuilt the multi site phosphorylation model with the Simmune modeler for s=1 to s=8 phosphorylation sites, and measured the CPU time spent on creating the network by both BioNetGen and Simmune. **Supplementary Fig. 6a** shows the median of five timings for each simulation. Simmune compares favorably with BioNetGen, performing between 5 and 20 times faster, where reliable measurements could be made. Simmune's timings for one and two phosphorylation sites are unreliable because the network generation times were shorter than the resolution of the system timer, causing the timer to report zero execution times. The figure shows that Simmune has inferior scaling behavior to BioNetGen. This leads us to expect that BioNetGen will eventually outperform Simmune when generating even larger networks (the network generated by the rule-set with eight phosphorylation sites contains 65538 complexes and 786432 reactions); however, creating networks of the multi-site model from Sneddon et al.[7] with 9 or more phosphorylation sites would be infeasible on a computer with 16 GB memory with BioNetGen.

The total time required by Simmune, BioNetGen, and NFSim to simulate the multi site phosphorylation model is shown in **Supplementary Fig. 6b**. For Simmune and BioNetGen, these simulations involve full network generation and integration of the resulting set of ODEs. The time evolution of the system was simulated for 200 seconds (simulated system evolution time, not CPU time). Again, each point represents the median of five simulations. The simulations were repeated with ten- and hundred-fold increased particle numbers, while decreasing the on-rates in the model by the same factor, to keep the steady states comparable.

Supplementary Fig. 6b illustrates the similar behavior of Simmune and BioNetGen. Overall, Simmune performs slightly better than BioNetGen, owing to the faster network generation (see **Supplementary Fig. 6a**). BioNetGen has a shorter initialization time of the ODE solver as can be seen from the better performance for the 1 site model. Comparing both BioNetGen and Simmune with NFsim highlights the fundamental difference between ODE/network-based and particle-based simulators. The particle-based simulator NFSim that creates reactions 'on-demand' and without symmetry checks outperformed the network-based simulators for simulations that involve very large networks but only require few particles to be simulated. In contrast, the ODE-based simulators performed better if the particle numbers of at least some of the reactants are in the range of 100,000 particles. Note that in many cellular signaling networks relative concentrations of pathway components differ by three or more orders of magnitude (for example due to high concentrations of MAP kinases and phospholipids in contrast to low concentrations of receptors and adaptors/scaffolds). A particle-based simulation representing the components with lowest abundance through 100 particles thus needs to represent the highest concentrations with particle numbers in the order of 10^5.

The particle numbers of the substrate of the multi-site model correspond to a concentration of approximately 5 nMol/l for the unscaled model with 3000 substrate molecules, assuming a cytoplasmic volume of 1 pL. In the high concentration setting with ~0.5 µMol/l substrate Simmune's performance for the largest network (8 phosphoryation sites) is comparable to Nfsim's while being considerably better for the smaller networks. Putting this network-size and concentration dependent performance comparison into the context of the yeast MAPK model in this manuscript with roughly 200 biochemical species and total particle numbers well beyond 10^5 particles, simulations with NFSim would take at least 1000 times longer than with Simmune, comparing just the non-spatial reaction aspects (NFSim cannot perform spatial simulations). This is obviously the result of the different scaling behavior of ODE/network-based simulators. If we disregard changing stiffness of the system's differential equations and changing occupation of particle states with changing system size and concentrations, we see that ODE/network based simulators -at best- scale linearly with the network size, but are independent of the simulated particle number, whereas on-demand-reacting particle-based simulators scale linearly with the particle number, but can be almost independent of the network size. We wish to point out, however, that performance comparisons between particle-based and ODE/PDE based simulations are of only rather limited value since particle-based simulations typically are not chosen because of performance (for most cell-wide biologically relevant networks and concentration ranges they are greatly outperformed by ODE techniques) but because they offer researchers the ability to investigate particular stochastic effects that ODE techniques cannot reproduce.

Assessment of the accuracy of Simmune's solver

We assessed the accuracy of Simmune's solver (and network generation) by comparing the simulated time courses of an extended set of observables in the four-site phosphorylation model to those obtained with BioNetGen. In addition to the free kinase

(Efree), free phosphatase (Ffree) and the set of substrate molecules that are phosphorylated at the first site (S1P), regardless of the phosphorylation states of the other sites we also defined observables for sets of substrate molecules requiring the first two three and all sites to be phosphorylated (S1P2P, S1P2P3P, S1P2P3P4P). **Supplementary Fig. 6c** shows the absolute value of the relative difference between the solutions produced by Simmune and BioNetGen. All differences are below 10^{-6}. The relative tolerances of the solvers were set to 10^{-4}. Therefore, the differences are consistent with the differences expected from the numerical inaccuracies of the solvers. Since BioNetGen and Simmune were developed completely independently of each other and use different network generation techniques as well as different ODE solvers the fact that the results obtained with the two approaches are identical (up to integrator tolerances) for a variety of different complex species patterns ('observables' in BioNetGen) suggests that both methods work correctly here.

c) Redistribution of membrane molecules during morphological changes

Performing simulations with dynamic morphologies raises a question that has to our knowledge so far not been discussed extensively: How are the concentrations of membrane molecules affected by changes of the morphology? The answer to this question will depend on several factors such as the time scales of the biochemical, diffusive and morphological processes as well as which phenomena are actually modeled in which detail.

The lipid bilayer of cellular membranes strongly resists changes of its area (in other words: it is not very elastic), yet cells can significantly change their apparent surface when producing membrane protrusion during processes such as migration. Thus there have to be mechanisms capable of adding lipid bilayer material to the apparent cell surface. The details of these mechanisms and their regulation are cell type dependent and not yet well understood[16, 17].

We implement four different algorithms for updating surface concentrations following changes of the membrane morphology. Two of the algorithms conserve the number of molecules on the membrane by redistributing molecules; the other two conserve the local concentrations. The number-conserving algorithms could be used when the lipid bilayer is modeled with high resolution having a one-to-one correspondence to the discretized cell surface. The model could then follow the molecular replenishment of the newly extended membrane regions through diffusion or active processes in detail. The concentration-conserving algorithms model an effective membrane which has structures below the resolution of light microscopy and thus are experimentally not accessible as is typically the case when data from live cell imaging is used as experimental reference. The model then assumes that such unresolved structures contain membrane-bound molecules that become available for signaling when surface wrinkles unfold and become part of the effective cell surface.

Whenever a volume element is added to or removed from a simulated cell, membrane elements have to be removed from the old and added to the new surface that results from

the applied changes to the cellular structure. As discussed above, there is no generally valid biological cellular mechanism that could motivate a particular choice of mapping between old and new elements for all types of models. The redistribution schemes take molecules from the old elements and redistribute them into the new ones. The schemes differ in their locality (are only membrane elements affected that are replaced or added, or do the redistribution schemes affect more elements?), handling neighboring (unchanging) elements differently. They represent different phenomenological models of convective and/or unzipping processes in the membrane in response to morphological changes. **Supplementary Figs. 8a-f** illustrate how molecules in the cell membrane are affected by the different schemes.

The molecule-number conserving algorithm that only affects the changing membrane elements first determines whether there is a membrane element that will be removed that has no neighbors that will not be removed (e.g. the membrane element at the top of the protrusion in **Supplementary Fig. 8b**). Its contents will be distributed equally amongst its neighbors. The remaining membrane elements are now guaranteed to have a neighbor that is not removed. Their molecular content is redistributed into their corresponding new membrane elements. A new membrane element has a corresponding old element if the old and new share a neighboring unchanging membrane element. If a new membrane element has no unchanging neighbors (e.g. the membrane element at the top of the protrusion in **Supplemenary Fig. 8a**) it will draw one fourth of the concentration of each of its neighbors. This scheme represents local redistribution with minimal convection and no membrane reservoirs.

The molecule-number conserving algorithm that includes the neighbors in the concentration update is essentially using the same three steps except that in the second step, when the concentration is mapped between old and new membrane elements that have unchanging neighbors, the molecules are not only distributed into the newly created membrane elements but also their unchanging neighbors.

Instead of keeping the number of molecules constant concentration-conserving algorithms preserve the average concentration between surface elements incident on a volume element that has been added or removed from the cell volume. In other words, the concentration in the new membrane elements is set to the average concentration of the removed membrane elements. The difference between the concentration-conserving algorithms lies in how local changes of the surface area (stemming from the approximation of a smooth membrane) induced by changes of the geometry in the neighborhood of an element are handled. By default, changes of the area of a surface element would lead to a change of the concentration to conserve molecule numbers. However, we can alternatively attribute these changes to the change of the approximated membrane instead and therefore conserve the concentration instead of the particle number.

d) Calculation of effective surface geometries

The surface of a cubic grid that approximates a cell does not provide a faithful approximation of the cell's surface. Therefore, the surface geometry has to be adapted to allow for a correct treatment of membrane diffusion.

We adapt the surface geometry by taking into account a neighborhood of each surface element instead of only the surface element itself for calculating gradients and the interface lengths that are used by the finite volume method discretizing membrane diffusion. The larger neighborhood permits a better estimation of the local tangents, curvatures, and distances.

We follow a previously suggested approach [18] to inscribe two circles into the surface and use their arc-lengths to define the adapted surface. This is achieved by finding a pair of surface elements separated by Δk steps along the grid (cf. **Supplementary Fig. 7a**) from the element under consideration and constructing the circle circumscribing the center of the three surface elements.

We define the distance to the direct neighbors as the arc length of the section of the circumcircle connecting the center of the surface element and the center of the direct neighbors (cf. **Supplementary Fig. 7b**).

However, instead of using the normals derived from the inscribed circumcircles to construct a Voronoi discretization as previously proposed [18], we avoid this time consuming step and use the distances to neighboring elements to derive the side lengths and the surface area of the surface elements and the distances entering the finite volume scheme for solving the diffusion equation. First, we define a side length of a surface element as one half of the sum of the distances to its opposing neighbors. The surface area is the product of the side lengths. The interface lengths and distances used for the finite volume scheme need to take into account that the distance from one element to its neighbor will differ from the reverse distance from the neighbor to the element. This is a result of the fact that the neighbor considers a different set of surface elements for estimating the radius of the circumcircle. We therefore define the distance between the centers of two surface elements and the length of the interface between these as the mean of their distances or side lengths, respectively.

There are situations in which this approach fails. The prescribed number of steps to the distant neighbors used for the construction of the circumcircle may ignore small membrane features. Or, the surface may be flat and thus the radius of the circumcircle infinite. In other situations, the center of the direct neighbor of a surface element lies close to the line passing through the center of the circumcircle and the center of the surface element resulting in a unrealistically small or vanishing distance. These situations are avoided by checking for these special cases:

- Stepping from neighbor to neighbor in opposite directions from the surface element under consideration reaches the same surface element. This occurs if the stepping algorithm wraps around a small detail (cf. **Supplementary Fig. 7c**).

- Stepping from neighbor to neighbor encounters more than two consecutive surface elements belonging to the same volume element. This occurs if a membrane has a thin protrusion (cf. **Supplementary Fig. 7d**).
- The cosine of the angle between vectors from the center of the circumcircle to two neighboring surface elements is larger than 0.995. This may occur for small values of Δk and "L" shaped surface features.

The first two cases indicate that the prescribed step number Δk is too large. In the first case the number of steps is reduced by a factor of two in order to reduce the angle spanned by the section of the circumcircle to approximately 180 degrees. In the second case a region with small membrane protrusions would have been encountered; the step number Δk is therefore reduced by two steps to exclude the protrusions from the estimation of the surface properties. The last case is handled calculating the length of the arc between the surface element and the element Δk steps away, normalizing it by the distance measured in steps Δk. If this approach fails too due to near co-linear vectors the distance is assigned by constructing a new circumcircle from the surface element and its direct neighbors.

As discussed in Supplementary Note 8 we can reproduce previously published results [18] finding the optimal value of $\Delta k = h^{-2/3}$ where h is the grid constant when sampling a sphere with radius 1. Note that Δk is of course independent of the physical size of the grid – only the ratio of the physical radius of the sphere and the grid constant are relevant. Therefore all calculations with respect to Δk are carried out in units of grid constants. Simmune uses the following heuristics to estimate an optimal value for Δk for a simulation.

- Calculate the mean volume of all cells in units of volume elements.
- Calculate the radius of a sphere enclosing the mean volume in units of grid constants
- Use the inverse of this radius to calculate $\Delta k = h^{-2/3}$

Supplementary Note 4: Numerical parameters and settings used in the simulations

a) E-cadherin model

Cell diameter: 30 microns

Density of E-cadherin molecules: 50 per square micron (ref. [19])

Diffusion coefficient of E-cadherin: 10^{-14} m^2 s^{-1} (0.01 square microns s^{-1}) (ref. [19])

E-cadherin *trans*-association rate: 9.0×10^4 l/(mol s). This value is based on experimentally reported numbers ($K_D \sim 100$ micromole) [19, 20] and references therein and then adjusted to take into account the smaller interaction volume containing the E-cadherin adhesive sites. In the simulations reported here, the membranes of two adjacent cells have a distance of 30 nm but the layer containing the interactive sites is only 10 nm wide. Multiplication of both concentrations with a factor 3 would yield an overall factor 9 that is absorbed here into the association rate.

E-cadherin *trans*-dissociation rate: 1 s^{-1}

E-cadherin *trans*-association rate (intra complex): 100 s^{-1}
E-cadherin *trans*-dissociation rate (intra complex): 0.01 s^{-1}

E-cadherin *cis*-association rate: 9.0×10^3 l/(mol s) (See above remarks on E-cadherin *trans*-association rate.)
E-cadherin *cis*-dissociation rate: 1 s^{-1}
E-cadherin *cis*-association rate (intra complex): 100 s^{-1}
E-cadherin *cis*-dissociation rate (intra complex): 0.01, 0.033, 0.1, 0.33, 1 s^{-1} (see main text and **Fig. 4d** for a discussion of the variation of the lifetime of intra-complex *cis* bond). Intra-complex associations are reactions that create additional bonds within existing complexes. The usual rate units of l/(mol s) do, therefore, not apply here and are replaced by transition rates (with unit s^{-1}). Intra-complex dissociations are reactions that break bonds within complexes without causing the complex to disintegrate (i.e., the complex is still held together by other bonds).

b) Yeast model
Cell length: 8 microns.
Cell width (max.): 4.2 microns.
Cell volume: 38.7 cubic microns.

Molecule abundances of main signaling components (per cell):
Fus3: 17,360 molecules (concentration measured in ref. [21], see also Suppl. Note 6)
Ste11: 700 molecules (concentration measured in ref. [21], see also Suppl. Note 6)
Ste7: 1930 molecules (concentration measured in ref. [21], see also Suppl. Note 6)
Ste5: 1290 molecules (concentration measured in ref. [21], see also Suppl. Note 6)
Msg5: 3200 molecules (concentration measured in ref. [21], see also Suppl. Note 6)
Cdc14: 1280 molecules (estimated): same cytoplasmic concentration as its substrate, Ste7 (a significant fraction of Ste7 is membrane bound).
MP2C: 640 molecules (estimated): same cytoplasmic concentration as its substrate, Ste11 (a significant fraction of Ste11 is membrane bound).
Ste20: 2140 molecules (estimated).
The complete list of all concentrations/abundances as well as of all reaction rates can be accessed by opening the model files with the modeler application and opening the simulation description with the simulator. Both applications run on Linux, Mac, and Windows operating systems and can be downloaded from www.niaid.nih.gov/LabsAndResources/labs/aboutlabs/lsb/Pages/simmuneproject.aspx.

In the polarized 'shmooing' yeast cell, membrane-bound adaptors such as Gβγ and Spa2p as well phospholipids (PIP_2) are concentrated in the shmoo tip to recruit Ste5, Ste7, and Ste20 [21, 22]. The tip location of these molecules is maintained through actin-dependent mechanisms. Since our model focuses on the polarized steady state, we do not model the process of accumulation of these anchors. Instead, we position them in the shmoo tip and set their diffusion coefficient to 0. The anchor for pFus3 we discuss in Supplementary Note 6 is an exception: this membrane anchor becomes activated in the tip through interaction with Gβγ but can diffuse away from the tip with a diffusion coefficient of 0.03

square microns s^{-1} while it becomes deactivated with a rate of 0.01 s^{-1}. Only in the active state it can tag pFus3 to the membrane.

Molecular concentrations of all species differ strongly between cytoplasm and nucleus due to selective import/export through nuclear pores and the presence of binding partners (such as $Dig_{1,2}$ for Fus3) in the nucleus. The cytoplasmic and nuclear concentrations of Fus3 and Msg5 are reported in Maeder et al. [21]. To reproduce those, we defined nuclear pores with the corresponding import/export ratio (3:1 for Fus3, 2:3 for Msg5). The concentration of Dig (representing Dig1 and/or Dig2), which is assumed to interact with pFus3, is adjusted so as to reproduce the nuclear part of the pFus3 concentration profile. Note that due to the exclusion of Ste7 from the nucleus [21] nuclear pFus3 is exposed to Msg5 to a much higher degree than its cytoplasmic counterpart. Without the presence of nuclear pFus3 binding partners (in the simulation: Dig), the level of pFus3 in the nucleus would be far lower than experimentally reported.

Global simulation conventions/parameters

a) Diffusion coefficients
The simulation application calculates the diffusion coefficients of multi-molecular complexes based on the assumption that they are proportional to the cubic root of the total mass of the complex. For example, the diffusion coefficient for a bimolecular complex AB consisting of two molecules A and B with diffusion coefficients D_A and D_B is calculated according to $D_{AB} = \left(D_A^3 + D_B^3\right)^{1/3}$.

b) 3D vs. 2D reaction rates
For the interactions between membrane-bound complexes it is assumed that their effective interaction volume is reduced compared to freely diffusing cytosolic components, thereby increasing their effective local concentrations. The effective interaction volume is 10 nm x membrane surface. Note that this convention (that amounts to using higher association rates for membrane-bound complexes) still allows the user of the modeler application to choose specific reaction rates by adjusting the 3D association rate in the model. Moreover, the modeler application permits the definition of 2D rates (in square microns s^{-1}) for interactions between membrane-bound complexes.

c) Membrane-membrane interactions
Adjacent membranes of cells in contact are assumed to have a distance of 30 nm. This distance also defines the intercellular volume for extracellular molecules diffusing between the cells.

Supplementary Note 5: A step-by-step guide to building and simulating the E-cadherin model is provided as an HTML document.

Supplementary Note 6: Discussion of the modeling constraints provided by experimental data on Yeast Pheromone Signaling

Experimentally determined numerical parameters used in the simulation were taken from Maeder et al.[21] and, partially, from Good et al.[23]. Maeder et al. measured the concentrations of Ste5, Ste11, Ste7, Fus3, and Msg5, the phosphatase that dephosphorylates pFus3. In addition, fluorescence intensity measurements of the relative concentrations of Ste5, Ste11, Ste7, and Fus3 at the shmoo tip were performed in that study and the fractions of Ste5 in complex with Ste11, Ste7, or Fus3 were measured using FCS and translated into effective dissociation constants for all pairwise interactions within the MAPK scaffold.

We will now discuss the implications that the measured parameters had for the modeling process and the conclusions that can be drawn from the combination of experimental and simulated data. We will also point out the degrees of freedom that remain in spite of the stringent constraints provided by the numerous measurements in Maeder et al.

All of the experimentally measured parameters (concentrations, fractions of Ste5 in complex with the kinases Ste11, Ste7, Fus3) have equivalents in the simulation that are not given by a single species concentration but by sums of concentrations of multiple molecular complexes. To determine, for instance, the fraction of Ste7 bound to Ste5 one first has to determine the sum of the concentrations of complexes containing Ste5 (in our model: 50 different complexes in the cytoplasm and even more at the membrane) and then the total concentration of complexes containing Ste7 bound to Ste5 (40 different complexes in the cytoplasm). The modeling approach (with automated generation of complexes and reactions) used here allows us to query these complex families by using a pattern matching mechanism. The complexes in our models can carry tags specifying states, for instance whether a molecule in the complex is phosphorylated or not. In the yeast pheromone model we defined (among others) a complex pattern “Ste5_all” that carries ‘wildcards’ (“don’t care”) for all tags of this complex species, meaning that any complex containing Ste5 in any state would match the pattern “Ste5_all”. Instead of summing up concentrations of all 50 complexes containing Ste5 we can thus ask the simulation to display the concentration profile of the pattern “Ste5_all”. Similarly, we can use the pattern “Ste5_Ste7_all” that matches all complexes in which Ste7 is bound to Ste5. Note, that a pattern corresponds to a specific binding structure of the complex. If Ste5 had two binding sites for Ste7 we would have to add the concentrations of the two corresponding patterns. For more details on complexes and states please see the documentation of the modeler application or the step-by-step guides and tutorials.

a) Experimentally measured cytoplasmic concentrations after stimulation with pheromone: 29 nM Ste5, 28 nM Ste11, 63 nM Ste7, 571 nM Fus3.

Since these concentrations do not include the membrane-bound fractions of the molecules, the simulation has to be initialized with higher concentrations to account for direct (Ste5 via Gβγ, Ste7 via Spa2p) or indirect (Ste11, Ste7, Fus3 via Ste5) membrane recruitment (see also Supplementary Note 4).

b) Measured fractions of Ste5 in complex with Ste11, Ste7, or Fus3:
fraction(Ste5:Ste11 in Ste5) = 0.33, fraction(Ste5:Ste7 in Ste5) = 0.27,

fraction(Ste5:Fus3 in Ste5) = 0.19.

Since these measurements were performed using a technique (Fluorescence Cross Correlation Spectroscopy, FCCS) that cannot easily distinguish between components that are directly bound to Ste5 vs. indirectly bound ones, the fraction(Ste5:Fus3 in Ste5) contains complexes in which Fus3 is bound directly to Ste5 and such complexes in which Ste7 links Ste5 and Fus3, i.e., Fus3 is indirectly bound to Ste5 (see **Fig. 5a,b**). Adopting the K_D from Good et al. for the interaction between scaffold-bound Ste7 and Fus3 (80 nM) resulted in a considerable fraction of the measured Ste5-bound Fus3 bound to Ste5 indirectly, via Ste7. This, in turn, means that the affinity for the direct association of Fus3 to Ste5 has to be even lower than estimated in the Maeder et al. publication (see Supplementary Note 6).

c) Measured relative concentrations of Ste5, Ste11, Ste7 and Fus3 at the shmoo tip:
ratio(Ste11/Ste5) = 0.15, ratio(Ste7/Ste5) = 1.0, ratio(Fus3/Ste5) = 1.7.

The concentration of Ste11 relative to that of Ste5 at the shmoo tip (0.15) is, interestingly, lower than the fraction of Ste5 bound to Ste11 in the cytoplasm. This may indicate that the off-rate for the release of pSte11 from Ste5 is higher than the off-rate for the dissociation of unphosphorylated Ste11. Since pSte11 has a much higher concentration at the tip than in the cytoplasm, this has only a minor effect on the total fraction of cytoplasmic Ste5 in complex with Ste11 and pSte11.

As discussed in Maeder et al., the fact that Ste7 has the same concentration at the tip as Ste5 means that Ste7 has to be recruited in a secondary mechanism in addition to the recruitment via the Gβγ:Ste5 complex. As suggested by Maeder et al. we incorporated Spa2p as a component that is concentrated at the membrane in the shmoo tip in a concentration that is adjusted relative to the concentration of Gβγ that recruits Ste5.

The concentration of Fus3 at the shmoo tip is 1.7 fold higher than the concentration of Ste5. As Maeder et al. pointed out, this may mean that Fus3 is retained at the tip by binding partners other than Ste5. For a model that does not include such binding partners, **Supplementary Fig. 4** shows a how the total production of phosphorylated Fus3 and the concentration difference between shmoo tip and shmoo-distant region of the cell vary with varying concentrations of Ste5 at the tip and varying activities (k_{cat}) of Ste7 phosphorylating Fus3 and Msg5 dephosphorylating pFus3. Even though it is possible to increase the phosphorylation rate of Fus3 (and/or decrease the rate of its dephosphorylation by Msg5) to compensate for the loss of pFus3 production caused by weak accumulation of Ste5 at the tip, adjusting those rates cannot reproduce the combination of 40% pFus3 production and strong intracellular gradients of Fus3/pFus3. In the original reaction-diffusion model developed in Maeder et al. this led to a fraction of phosphorylated Fus3 of 25%, a number that is confirmed by our simulations (**Supplementary Fig. 4**). The parameter scan for which the results are shown in **Supplementary Fig. 4** can be performed using the files provided in the 'YeastScan_Kinase_Phosphatase_Recruitment' folder.

Following the suggestion from the publication by Maeder et al., we therefore modified the model in such a way that some pFus3 is retained at the tip through interactions with other molecules besides Ste5 (in the model "YeastMAPK.dbf" the name of the molecule representing those 'anchors' is simply "Fus3_anchor"). Note that, in addition to reconciling the simulated relative concentrations of Fus3 and Ste5 at the tip with the experimental observations, this also shifts the balance between the concentration of Fus3/pFus3 at the tip and in the tip-distal parts of the cell towards the tip, thus making the intracellular gradient steeper. Remarkably, a constraint about the relative concentrations of two components (Fus3, Ste5) at the tip led to a model that turned out to solve another problem: the previous inability to combine a strong gradient with a large fraction of the Fus3 pool becoming phosphorylated. Since part of the pFus3 gradient (that is steeper than that of the total Fus3) is now due to pFus3 being retained at the tip (as opposed to being due only to dephosphorylation by Msg5 as it leaves the tip), the activity of the phosphatase Msg5 can be weaker, thereby allowing a higher total production of pFus3. However, since we cannot resolve the relative contributions from the two sources of pFus3 accumulation at the tip, we are left with a degree of freedom, an unknown that requires additional experimental investigation beyond the scope of this methodological report.

Supplementary Fig. 4 also shows that in the presence of significant Msg5 phosphatase activity on pFus3, the activity (k_{cat}) of Ste7 phosphorylating Fus3, in our simulations, has to be significantly higher than the experimentally determined value ($4.3x10^{-3}$ s^{-1}) reported in ref. [23]. The maximum number of complexes producing pFus3 with scaffold support is given by the amount of Ste5 bound to Ste7, which is approximately 400 complexes per cell (30% of Ste5 is bound to Ste7 according to ref. [21]). This would yield a max. production rate of approx. 100 pFus3 molecules per minute which could, interestingly, result in the phosphorylation of a major part (>30%) of the approx. 17000 Fus3 molecules (in the cell size we used for our simulations) within an hour (40% had been reported [21]) if the phosphatase Msg5 (and all other phosphatases targeting pFus3) are switched off. If, however, Msg5 actively contributes to the pFus3 gradient one would have to suggest that the activity of scaffold bound Ste7 on Fus3 is much higher *in vivo* than reported in ref. [23]. In our simulations, we were able to reproduce experimentally observed pFus3 gradients and total amounts with k_{cat} values ranging from 3 s^{-1} to 10 s^{-1} for scaffold bound Ste7 phosphorylating Fus3. Note that for the classical 'in-scaffold' activation scheme (**Fig. 5a** main text), according to which Fus3 has to be bound to Ste5 directly, we would obtain (assuming a K_D of 1 micromole of the Ste5:Fus3 interaction and 570 nM of cytoplasmic Fus3) only about a third (145) of the max. number of pFus3 producing complexes.

Note: Supplementary Note 6 is provided as a separate document.

References

1. Blinov, M.L., Faeder, J.R., Goldstein, B. & Hlavacek, W.S. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics* **20**, 3289-3291 (2004).
2. Takahashi, K. et al. E-Cell 2: multi-platform E-Cell simulation system. *Bioinformatics* **19**, 1727-1729 (2003).
3. Feret, J., Danos, V., Krivine, J., Harmer, R. & Fontana, W. Internal coarse-graining of molecular systems. *Proc Natl Acad Sci U S A* **106**, 6453-6458 (2009).
4. Czech, J., Dittrich, M. & Stiles, J.R. Rapid creation, Monte Carlo simulation, and visualization of realistic 3D cell models. *Methods Mol Biol* **500**, 237-287 (2009).
5. Tolle, D.P. & Le Novere, N. Meredys, a multi-compartment reaction-diffusion simulator using multistate realistic molecular complexes. *BMC Syst Biol* **4**, 24 (2010).
6. Lok, L. & Brent, R. Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nat Biotechnol* **23**, 131-136 (2005).
7. 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).
8. Ander, M. et al. SmartCell, a framework to simulate cellular processes that combines stochastic approximation with diffusion and localisation: analysis of simple networks. *Syst Biol (Stevenage)* **1**, 129-138 (2004).
9. Andrews, S.S., Addy, N.J., Brent, R. & Arkin, A.P. Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Comput Biol* **6**, e1000705 (2010).
10. Arjunan, S.N. & 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).
11. Gruenert, G. et al. Rule-based spatial modeling with diffusing, geometrically constrained molecules. *BMC Bioinformatics* **11**, 307 (2010).
12. Wils, S. & De Schutter, E. STEPS: Modeling and Simulating Complex Reaction-Diffusion Systems with Python. *Front Neuroinform* **3**, 15 (2009).
13. Moraru, II et al. Virtual Cell modelling and simulation software environment. *IET Syst Biol* **2**, 352-362 (2008).
14. Hlavacek, W.S., Faeder, J.R., Blinov, M.L., Perelson, A.S. & Goldstein, B. The complexity of complexes in signal transduction. *Biotechnol Bioeng* **84**, 783-794 (2003).
15. Weiner, R., Schmitt, B.A. & Podhaisky, H. ROWMAP - a ROW code with Krylov techniques for large stiff ODEs. *Appl. Numer. Math.* **25**, 303-319 (1997).
16. Hallett, M.B., von Ruhland, C.J. & Dewitt, S. Chemotaxis and the cell surface-area problem. *Nat Rev Mol Cell Biol* **9**, 662; author reply 662 (2008).
17. Kay, R.R., Langridge, P., Traynor, D. & Hoeller, O. Changing directions in the study of chemotaxis. *Nat Rev Mol Cell Biol* **9**, 455-463 (2008).

18. Novak, I.L. et al. Diffusion on a Curved Surface Coupled to Diffusion in the Volume: Application to Cell Biology. *J Comput Phys* **226**, 1271-1290 (2007).
19. Perez, T.D., Tamada, M., Sheetz, M.P. & Nelson, W.J. Immediate-early signaling induced by E-cadherin engagement and adhesion. *J Biol Chem* **283**, 5014-5022 (2008).
20. Chen, C.P., Posy, S., Ben-Shaul, A., Shapiro, L. & Honig, B.H. Specificity of cell-cell adhesion by classical cadherins: Critical role for low-affinity dimerization through beta-strand swapping. *Proc Natl Acad Sci U S A* **102**, 8531-8536 (2005).
21. Maeder, C.I. et al. Spatial regulation of Fus3 MAP kinase activity through a reaction-diffusion mechanism in yeast pheromone signalling. *Nat Cell Biol* **9**, 1319-1326 (2007).
22. Dohlman, H.G. & Thorner, J.W. Regulation of G protein-initiated signal transduction in yeast: paradigms and principles. *Annu Rev Biochem* **70**, 703-754 (2001).
23. Good, M., Tang, G., Singleton, J., Remenyi, A. & Lim, W.A. The Ste5 scaffold directs mating signaling by catalytically unlocking the Fus3 MAP kinase for activation. *Cell* **136**, 1085-1097 (2009).