Individual-based simulation of the clustering behaviour of epidermal growth factor receptors.

Jacki P. Goldman and William J. Gullick
Department of Biosciences
University of Kent Canterbury,
Kent, CT2 7NJ England
jpg,wjg@ukc.ac.uk

Dennis Bray
Department of Zoology
University of Cambridge
Downing Street Cambridge,
CB2 3EJ England
d.bray@zoo.cam.ac.uk

Colin G. Johnson
Computing Laboratory
University of Kent Canterbury,
Kent, CT2 7NF England
cgj@ukc.ac.uk

ABSTRACT

This paper describes ongoing work on a project to simulate the behaviour of *epidermal growth factor receptors*. These are structures which can be found on the surface of cells in the body, which receive and process chemical signals concerned with cell growth. We describe the implementation of a program which simulates the stimulation and clustering behaviour of these structures, discuss how we scale up this simulation so that we can simulate a whole cell on a tractable timescale, and discuss ongoing work in which we are calibrating our simulation against results from experiments.

Keywords

receptor biology, simulation, modelling, individual-based models, computational biology

1. BIOLOGICAL BACKGROUND.

In order for the body to function, cells need to communicate with each other. One important mechanism for this is the diffusion of chemical signalling molecules (*ligands*) in tissues, which are in turn received by *receptors* on the surface of cells [13]. Once these signalling molecules have bound to the receptors, they trigger a set of events inside the cell (figure 1).

A particular example of a system of this type is the *epidermal growth factors* (EGFs) and their receptors [25]. This consists of four different receptors (one of which can exist in four different forms) which can be bound to by at least ten different ligand types. Once the receptors have bound ligand, they form clusters which stimulate within-cell signalling events concerned with cell growth. Thus the cell can communicate with the extracellular environment without actual molecules being passed inside the cell membrane.

These structures are of importance for medicine as well as of scientific interest, as this system can contribute to cancer formation in three ways. Firstly a mutation can cause the cell to generate its own growth factor, generating unchecked cell growth. Secondly mutations in the genes for the receptor can cause it to be constantly activated regardless of the presence or absence of ligand. A third problem occurs when too many receptors are created, meaning that the cell exhibits excessive growth for small amounts of signal.

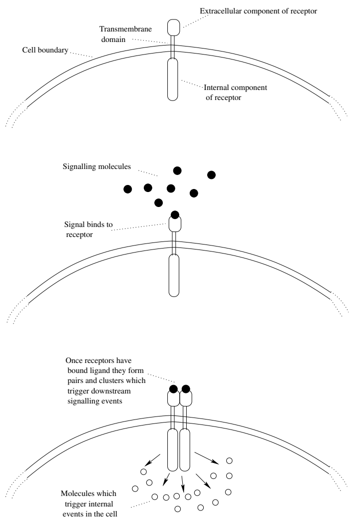


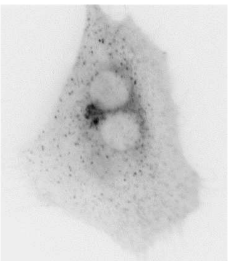
Figure 1: How cells receive signals using receptors.

We would like to understand the dynamics of this clustering process better and to understand the structure of the clusters formed. We can observe the cluster formation by tagging the receptors with a fluorescent protein and filming the motion under an optical microscope [11]. At first the fluorescence is distributed evenly around the cell, but five minutes after ligand molecules have been introduced clusters can clearly be seen (figure 2 a) and counted (figure 2b,c). However these data provide little insight into the processes which cause cluster formation. To better understand this process we have been building a computer simulation of the system.

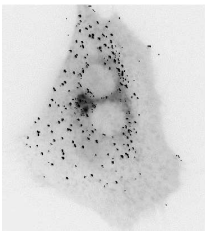
2. RELATED WORK.

Computer simulation is a technique of rapidly expanding relevance to the biological sciences. Within cellular biology a number of different techniques have been applied.

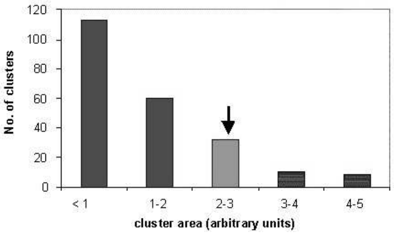
The most traditional use of computer simulation has been in computing the stoichiometrics of reactions within the cell [8]. Such approaches have used both programs specifically written to carry out these calculations, such as *Gepasi* [17] and SCAMP/*Jarnac* [22] and general purpose mathematical software tools.



(a)



(b)



(c)

Figure 2: Image analysis of EGFR clustering in a cell microinjected with a GFP-tagged EGFR cDNA. (a) Original image obtained 5 minutes after addition of 500nm EGF. (b) Image after analysis using SimplePCI software, with clusters highlighted. (c) Data on cluster area generated using the SimplePCI software.

Techniques such as these cannot be used to model phenomena which depend on the spatial distribution of individual molecules, where the complete mixing assumption which underlies the above projects cannot be applied. Instead we can take an approach to simulation which is based on creating interacting models of individual components in the simulation; such models have a long history of success in ecological research [7, 16] but have seen fewer applications at the cellular level.

Nonetheless there have been some successful applications of this type of modelling to the analysis of systems such as cell-signalling networks [3, 4], the G-protein cascade [14] and the microphysiology of synaptic transmission [2].

Other systems have been developed which attempt to simulate the whole cell, e.g. the *E-Cell* project [24]. These are exciting and ambitious projects; however to use such a system requires much of knowledge about the particular type of cell in question, in particular detailed sequence- and structure- data about its proteins. In problems such as the one studied in this paper, we have limited amounts of data; therefore we have created a focussed tool which allows us to exploit what data we have and actively apply the simulation system to discovering conjectures for parameters in the real system.

3. SIMULATION AIMS AND DESIGN.

Our basic techniques in simulating this system was to create an individual-based object-oriented model of the cell surface. This is a powerful technique for this type of system with a heterogeneity of objects in the cell and no simple algorithm for global behaviour. To create an object-oriented model we need to characterise the *state* and repertoire of *behaviours* available to a particular object.

The primary classes of objects used in modelling growth factor receptor aggregation are the receptor molecules themselves. Receptors exist in the cell membrane, and exhibit Brownian motion. Growth factor receptors (represented by a `Molecule` class in the system) have a certain affinity for other receptors of the same family to form dimers, which is greatly increased by the addition of ligand. These dimers then form larger clusters.

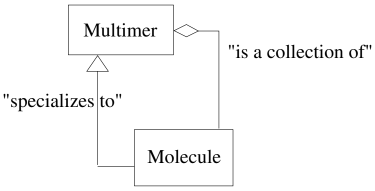


Figure 3: Modelling the multimer class as simultaneously a collection of and special case of the molecule class.

The `Molecule` class was written to contain information about individual receptor monomers in the simulation. The *state* of a `Molecule` object can be represented by its position in the cell membrane, the direction and speed with which it diffuses in the cell membrane (all represented as floating-point values), its size, and whether or not it has ligand bound. A `Molecule`'s size is represented by a diameter which is proportional to the number of subunits it contains. The *behaviour* of `Molecule` objects includes, for example, the ability to move, to bind ligand, and to form aggregates with other `Molecules`.

Dimers and higher-order oligomers (clusters of receptors) have many of the same characteristics as receptor monomers and behave much the same way on the cell surface. They have a location, size, they move, collide and bind to each other. One powerful feature of object-oriented design is the use of inheritance, which allows a subclass to inherit the attributes and behaviour of another while allowing added functionality not included in the superclass. The `Multimer` class is a subclass of `Molecule` to represent aggregates of one or more receptor monomers. It inherits most the functionality of `Molecule`, but has an additional collection attribute to keep references to the individual `Molecules` of which it is composed. In this way, a `Multimer` is both a `Molecule` and a collection of `Molecules` (figure 3). It also has the ability to dissociate, a behaviour that is not allowed in `Molecule` objects.

This property of `Multimers`, where they are both a type of `Molecule` in their own right and a collection of `Molecules` demonstrates a pattern which might well be found in other scientific models. This might be illustrative of a kind of *design pattern* which is science-specific. Could such science-specific patterns guide the design of simulations, in as general design patterns [10] have helped in programming more generally?

An object of the `CellSurface` class acts as the simulation engine. This object correlates to a rectangular portion of a cell membrane, in which the `Molecules` move and interact. The area it covers is represented by an attribute which is a two dimensional Cartesian plane with continuous values, so that the location of the `Molecules` can be determined precisely. Although a cell is a three dimensional entity, cell surface molecules are embedded in or attached to the cell membrane, which is essentially a planar surface, particularly over small areas. The motion of these molecules is therefore constrained to two dimensions, and can be modelled in this way. `Molecules` move on the `CellSurface` in a toroidal fashion. As a `Molecule` leaves the confines of the `CellSurface`, its position is reset such that it re-enters on the opposite side. This models the fact that in a typical region on the surface the inward flux of material is equal to the outward flux.

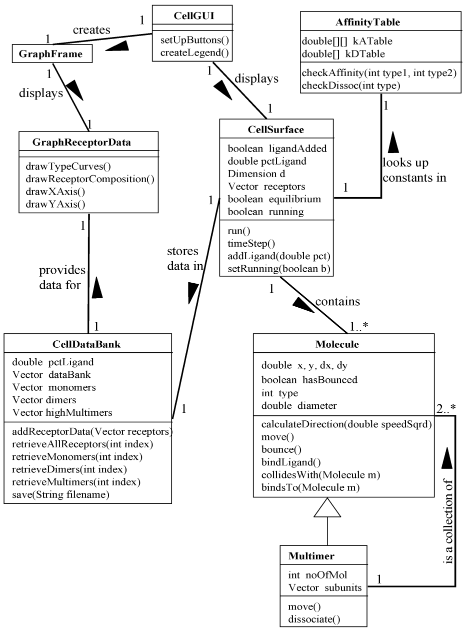


Figure 4: A UML diagram to summarize the design of the simulation.

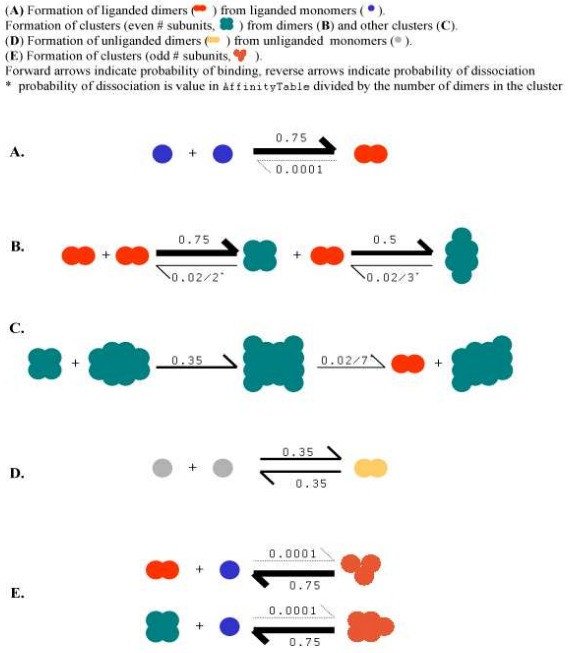


Figure 5: An example of transition probabilities between various states.

Each `CellSurface` object is associated with an `AffinityTable` object. This provides constants corresponding to different `Molecule` types. It also provides a two dimensional array of values for association and disassociation probabilities, the indices of which correspond to the `Molecule` types. Thus, when two `Molecules` collide, their affinity can be looked up in this table based on their values for type.

The design is summarized in a UML [9] diagram in figure 4.

The simulation begins with the initialisation of the cell surface. This involves creating a `CellSurface` object and populating it with unliganded monomers in random locations. The program then runs as a loop in which the `Molecules` are moved in a manner which approximates Brownian motion. At any point, the user can choose to convert any proportion of cells to a liganded state, which initiates the aggregation process.

When the positions of two `Molecules` indicate that a collision between the two has occurred the `CellSurface` uses the `AffinityTable` to look up values for binding and dissociation constants (figure 5). The sequence is as follows: a `Molecule` moves and its position is checked with regard to other `Molecules` in the simulation. If there is any overlap between the area covered by that `Molecule` and another `Molecule`, a collision is deemed to have taken place. If affinity is sufficient, a `Multimer` is formed whose subunits consist of the monomers from the colliding `Molecules`. If, however, the affinity calculation is not favourable, the moving `Molecule` stops at the point in its trajectory just outside the area of the other `Molecule`. Thus, `Molecules` cannot move through each other, and no two `Molecules` can occupy the same space.

The graphical display of the `CellSurface` draws each `Molecule` as a circle whose area is proportional to the number of subunits it contains. This can be used either to generate a computergraphical animation of the changing state of the cell surface (figure 6) or to generate statistics about the changing structure of the clusters with time.

4. SPEEDING UP THE SIMULATION.

The use of floating-point coordinates in this model contrasts with the common use of grid models in individual-based modelling (e.g. [15]). In grid models each component is placed at a pair of integer coordinates on a grid, which can lead to artifacts which do not represent the real world; an example would be that (given a simple move up-down-left-right model of motion) there is a bias toward motion in the grid directions which does not reflect the uniformity of the real world. It also contrasts with continuum models based upon density gradients of molecules (e.g. [20, 6, 1]). In summary the objects which are discrete in the real world (i.e. the individual molecules) are represented by discrete computational objects, whilst the parts of the system which are continuous (i.e. the ambient medium in which the molecules move) are represented by a continuum of values in the computer. There are some similarities with particle-in-cell methods in computational physics [21].

One reason why grid models are typically used in individual based spatial modelling is to simplify the problem of collision detection between particles in the system. However in our model we have introduced a technique which allows fast collision detection whilst retaining the idea of each object's position being represented by floating-point coordinates.

This system exploits the ability of OO systems to link two pieces of information together so that each of the two pieces is mutu-

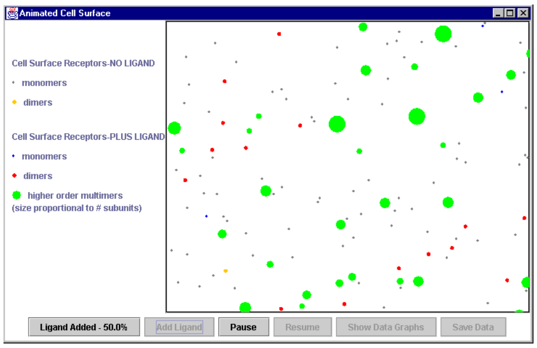


Figure 6: A snapshot from the animation of a portion of the cell surface.

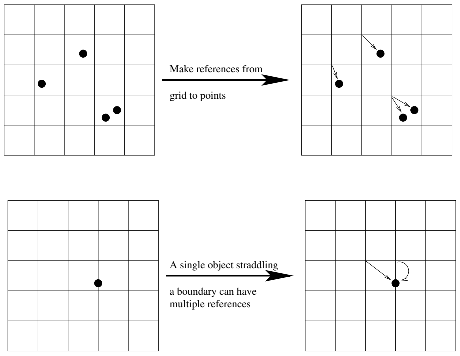


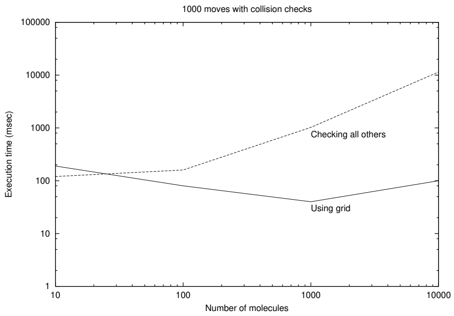
Figure 7: Clamping the objects to the corners of the grids which represent them.

ally aware of the other. We use this to combine the advantages of the grid and continuum representations. The main representation places the receptors on the 2-dimensional floating-point continuum; all motion takes place in this space. However in addition to this space a grid spans the space.

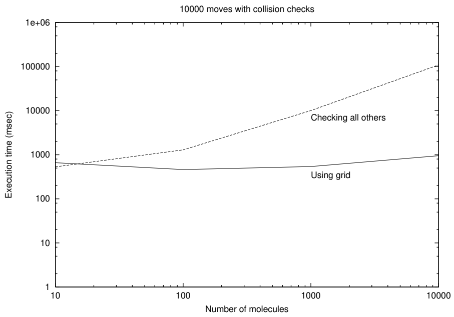
Each point on this grid "owns" the rectangle of the surface for which it is the top left-hand corner. At each crossing point on the grid there is a list (a variable-length array type such as a Java `ArrayList`) of references (C-style pointers) to objects which are of the class of the items on the surface. This is easily extended to multiple types of object moving on the surface, by having the pointers point to a Java-style interface type, which is implemented by all of the points on the surface.

When we begin the simulation we iterate through the objects on the surface, work out which region(s) they belong to, and add a reference from the appropriate grid point to the object (figure 7). We then make a list of references (C-style pointers again, or just coordinates) *back* from the object to the grid points which contain them. This ability for two objects to both know data about each other is a powerful technique in object-oriented modelling, and is a powerful way to get away from an oversimple hierarchical view where each piece of data contains other data, and not *vice versa*.

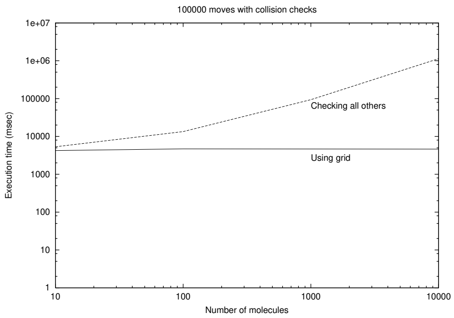
Once we have established this data structure, checking for collisions is easy. We take the object which we are interested in, and iterate through its list of grid points. This gives us a list of regions which the object occupies. We then take these grid points and iterate through its set of objects. This gives us a list of objects which occupy the region in question. We can then do an intersection check



(a)



(b)



(c)

Figure 8: Comparing the grid-based algorithm with the algorithm which checks all possibilities.

on each of those objects with the original object.

If an object moves we call a single method which updates its grid position. This method calculates which regions of the grid the object now interacts with. Using this list and the previous list, a message could be sent to each of the grid-points whose regions are not longer occupied asking that it be removed from that grid-point's list of objects. It then removes its own links to any regions which are no longer occupied, and sends messages to the grid-points belonging to any newly occupied region asking for it to be added to its list. This can be neatly contained in its own method.

Figure 8 gives results for three experiments with a different number of molecules in each experiment. The results show that using the underlying grid speeds up the program immensely, reducing the increase in computational load as we increase the scale of the experiments to almost constant regardless of the size of the experiment. This will enable the simulation to be scaled up to a whole-cell size.

Further experiments have been aimed at finding the optimal size for the grid. If the size of the squares in the grid is too large, to

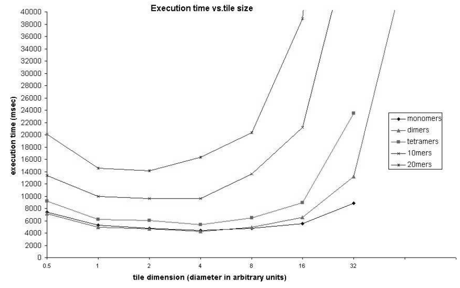


Figure 9: Results: Best grid size.

many objects need to be checked for collision. If it is too small, then each object occupies multiple grid regions. Intuitively the best size would seem to be slightly larger than the typical size of the object, as this would prevent both of these extremes—we might characterize this by saying that the typical tile should contain one object most of the time. Thus we expect a U-shaped curve. Results confirming this are given in figure 9.

5. ONGOING WORK.

We are developing this work in a number of directions. At present the system only deals with one kind of ligand binding to one receptor type. In the real system there are seven receptor types which bind at least ten ligands with different binding strengths [12, 25], and clusters containing a mixture of receptor types have been observed [19]. Another factor which might influence the dynamics of the system are *lipid rafts* which concentrate receptors in parts of the cell [5].

We are also beginning to apply the model to answer scientific questions. In particular we would like to know values for parameters which are inaccessible to direct experimental observation, such as probabilities of association and dissociation. To do this we need to calibrate the model against the real system. One way to do this is to create computergraphical “films” of models which can be adjusted by experts familiar with the system to look like films of the real system obtained by microscopy (these can be seen at `http://wwwbio.ukc.ac.uk/gullick/icrf.htm`). A second approach involves carrying out image processing on the real films to obtain statistics such as time series of changes in cluster size. We will then apply optimization techniques such as genetic algorithms and tabu search to find parameter settings which match these statistics in the output from the simulation.

Another use to which the simulation will be put is in searching for components of the system which are most sensitive to change, as such regions might be suitable targets for therapeutic intervention. Characteristics of the system such as association and dissociation probabilities can be affected by binding other molecules onto the receptors or ligands. An example of the successful therapeutic application of interventions aimed at disrupting an overexpressed growth factor receptor system is the anti-cancer drug transuzumab (Herceptin™), which blocks a form of growth factor receptor [23]. To create such treatments in a rational way we need to search the space of possible intervention sites for those where small interventions have large effects, and computational search techniques such as genetic algorithms and active nonlinear tests [18] provide a potential method for doing this in a tractable way.

A broader ongoing project is examining how we can create more general object-oriented modelling techniques for interacting protein systems, and more generally still how we need to adapt object-oriented techniques for the type of programming needed for science rather than business applications.

6. NOTES.

Parts of this work have been funded by the UK Medical Research Council. The current version of the model can be found on the web at:
`http://www.cs.ukc.ac.uk/people/staff/cgj/research/receptors.html`

7. REFERENCES

[1] A. Anderson, M. Chaplain, E. Newman, R. Steele, and A. Thompson. Mathematical modelling of tumour invasion and metastasis. *Journal of theoretical medicine*, 2:129–154, 2000.

[2] T. M. Bartol and J. R. Stiles. MCell: A general monte carlo simulator of cellular microphysiology. Website describing ongoing project, at `http://www.mcell.cnl.salk.edu/` (visited December 2000), 1999.

[3] D. Bray and S. Lay. Computer simulated evolution of a network of cell-signaling molecules. *Biophysical Journal*, 66:972–977, 1994.

[4] D. Bray, M. D. Levin, and C. J. Morton-Firth. Receptor clustering as a cellular mechanism to control sensitivity. *Nature*, 393:85–88, 1998.

[5] G. Carpenter. The EGF receptor: a nexus for trafficking and signalling. *Bioessays*, 22(8):697–707, 2000.

[6] P. Dale, L. Olsen, P. Maini, and J. Sherratt. Travelling waves in wound healing. *FORMA*, 10:205–222, 1995.

[7] D. DeAngelis and L. Gross, editors. *Individual-based Models and Approaches in Ecology*. Chapman and Hall, 1992.

[8] D. A. Fell. *Understanding the Control of Metabolism*. Portland Press, London, 1996.

[9] M. Fowler. *UML Distilled*. Addison-Wesley, 1997.

[10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns*. Addison-Wesley, 1994.

[11] H. Gillham, M. C. Golding, R. Pepperkok, and W. J. Gullick. Intracellular movement of green fluorescent protein-tagged phosphatidylinositol 3-kinase in response to growth factor receptor signalling. *Journal of Cell Biology*, 146(4):869–880, 1999.

[12] W. Gullick. The type 1 growth factor receptors and their ligands considered as a complex system. *Endocrine-Related Cancer*, 8:75–82, 2001.

[13] J. Hancock. *Cell Signalling*. Longman, 1997.

[14] T. D. Lamb. Stochastic simulation of activation in the G-protein cascade. *Biophysical Journal*, 67:1439–1454, 1996.

[15] C. G. Langton. Studying artificial life with cellular automata. *Physica D*, 22:120–149, 1986.

[16] C. C. Maley and H. Caswell. Implementing i-state configuration models for population dynamics : An object-oriented programming approach. *Ecological Modelling*, 68:68–75, 1993.

[17] P. Mendes. Biochemistry by numbers: simulation of biochemical pathways with Gepasi. *Trends in Biochemical Sciences*, 22:361–363, 1997.

[18] J. H. Miller. Active nonlinear tests (ANTs) of complex simulation models. *Management Science*, 44(6), 1998.

[19] C. R. Monks, B. A. Freiberg, H. Kupfer, N. Sciaky, and A. Kupfer. Three-dimensional segregation of supramolecular activation clusters in T-cells. *Nature*, 395:82–86, September 1998.

[20] J. Murray. *Mathematical Biology*. Springer, 1993.

[21] C. D. Norton, B. K. Szymanski, and V. K. Deyck. Object-oriented parallel computation for plasma simulation. *Communications of the ACM*, 38(10):88–100, 1995.

[22] H. Sauro and D. Fell. SCAMP: A metabolic simulator and control analysis program. *Mathematical and Computational Modelling*, 15(12):15–28, 1991.

[23] M. Sliwkowski. Nonclinical studies addressing the mechanism of action of trastuzumab (herceptin). *Seminars in Oncology*, 26:60–70, 1999.

[24] M. Tomita, K. Hashimoto, K. Takahashi, T. Shimizu, Y. Matsuzaki, F. Miyoshi, K. Saito, K. Yugi, J. Venter, and C. Hutchinson. E-CELL: software environment for whole cell simulation. *Bioinformatics*, 15(1):72–84, 1999.

[25] Y. Yarden and M. X. Sliwkowski. Untangling the ErbB signalling network. *Nature Reviews: Molecular Cell Biology*, 2:127–137, February 2001.