

COMPUTER SIMULATION OF RECEPTOR CLUSTERING

Dr J. P. Goldman

Department of Biosciences
University of Kent, Canterbury
CT2 7NJ

ABSTRACT

Due to the complexity of biological systems, it is often difficult to dissect and analyse the myriad biochemical interactions that occur. Computer simulations have been used to predict the outcome of biochemical processes as well as to shed light on the complex mechanisms which govern them. One such system is the interaction of growth factors and their receptors.

The aim of this project was to model a small aspect of this process, namely the ligand-dependent aggregation of Type 1 growth factor receptors which has been observed experimentally. An object-oriented design has been used to produce a simulation of growth factor receptor clustering. This program allows the user to watch the process of molecular clustering as an animation and also to view plots of data generated by the simulation. Another form of the program runs the simulation without the animated display. This allows scaling of the simulation to represent the receptors on an entire cell, and allows the number of iterations to be increased to represent several minutes in real time.

The data generated by these simulations may be useful in future for analysis of clustering behaviour under different conditions to aid in determining the characteristics of molecules which are important for clustering and to give insight into the underlying mechanism.

INTRODUCTION

Growth factors are hormone-like substances which play an important role in the maintenance and development of normal tissues and in human cancer. These bind to receptor tyrosine kinases (RTKs) present on the cell surface. This causes a cascade of protein interactions within the cell which can ultimately result in cell growth, differentiation, migration or death.

The erbB family of growth factor receptors consist of four receptors, c-erbB-1 (also known as epidermal growth factor receptor (EGFR)) c-erbB-2, c-erbB-3, and c-erbB-4, which in addition to their role in normal development, can be important in the genesis of several common solid tumour types. Because these proteins make ideal targets for anti-cancer drugs (1), a detailed understanding of the molecular interactions resulting in receptor activation and intracellular signalling is needed.

Binding of ligand to EGFR-family receptors induces dimerisation, required for the activation of the intracellular tyrosine kinase domain. Phosphorylated signalling domains act as docking sites for intracellular signalling proteins. Once bound, these can act on their target proteins, and so on, until the signal is propagated to the nucleus. It is possible to have perturbations in this cycle, which can lead to uncontrolled cell growth. These include overexpression of normal growth factor receptor, often due to gene amplification, rearrangement resulting in overproduction of ligand, or expression of a constitutively activated mutant form of receptor or (2).

Understanding the molecular interactions of Type 1 growth factor receptors is difficult due to the complexity of the receptor-ligand interactions. This family of growth factor receptor genes comprises four genes (EGFR (c-erbB-1), c-erbB-2, -3 and -4), which generate seven proteins (including 4 splice variants of c-erbB-4). There are a set of at least 12 different ligands, each of which is capable of binding to a subset of these protein molecules (3). Thus, in this system it is possible to generate 28 different homo- and heterodimers of these proteins.

A light-based system for tracking intracellular movement of receptors and second messengers has been developed in the laboratory of William Gullick. Films and still images have been produced using this system which show, for example, that EGF receptors form clusters of approximately 1000 molecules within five minutes after treatment of EGF-expressing cells with growth factors (Fig 1). We have access to antibodies against most of the receptors and all of the ligands with the exception of epiregulin, and we are in the process of making GFP- and /or RFP-tagged constructs

of all of the receptors. The fate of proteins generated by these constructs can be followed using high resolution imaging software.

We believe it would be useful to generate a computer simulation of the movement and interactions of cell surface receptors. We could then use this to generate data about cell surface receptors under different conditions, and by comparing this data to actual experimental data, refine the simulation. Similarly, we could use the simulation to determine parameters affecting receptor clustering which may be more sensitive to changing conditions, and use this information to direct experiments to areas which are more likely to generate useful information.

The main focus of this report is to describe in some detail the computer simulation in its current state.

DESIGN

Object-oriented modelling is a powerful technique for modelling complex systems. Designing an algorithm which defines the outcome of a simulation based on such a system would be very difficult, if not impossible. Instead, objects are used to represent real-world entities which are components of the system. Objects encapsulate the state and behaviour of real-world entities. This behaviour can include interactions with other objects, which in turn may influence the state and behaviour of these objects. Complexity emerges from these interactions.

Classes are used to implement object-oriented designs. Each class represents a blueprint containing a set of characteristics (attributes) and behaviour (methods) which define objects of that class. It is then possible to create many objects of a class, each with different values for its attributes, thus producing a population of individual objects.

The primary objects in modelling growth factor receptor aggregation are the molecules themselves. In order to create objects which represent molecules, it was necessary to decide which characteristics of molecules are important in the context of the simulation. In reality, cell surface molecules exist somewhere in the cell membrane and they exhibit Brownian motion. Molecules such as growth factor receptors have a certain affinity for other receptors of the same family to form dimers, which is greatly increased by the addition of ligand. Dimers form larger clusters as well. 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, its size and whether or not it has ligand bound. Thus, the attributes representing the state of a `Molecule` object include `x`- and `y`-coordinates, to indicate location, and values for `dx` and `dy` to indicate changes to `x` and `y` caused by moving (i.e. to define the direction of its next move). A `Molecule`'s size is represented by a diameter which is proportional to the number of subunits it contains. A `Molecule`'s type contains information about the number of subunits and whether or not it has ligand bound. The behaviour of `Molecule` objects includes, for example, the ability to move, bind ligand and to form aggregates with other `Molecules`.

Dimers and higher-order oligomers 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`. It also has the ability to dissociate, a behaviour that is not allowed in `Molecule` objects.

A `CellSurface` object acts as the simulation engine. This 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 insures that no `Molecules` are lost and that their density remains constant throughout the simulation.

Each `CellSurface` object is associated with an `AffinityTable` object. This provides integer constants corresponding to different `Molecule` types. It also provides a two dimensional array of values for association 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. `AffinityTable` also has an array for dissociation probabilities in which the indices correspond to type.

For a summary of how the classes in the simulation interact, refer to the UML class diagram in Figure 2.

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. During this loop, ligand may be added to the `CellSurface`, which initiates the aggregation process. The user can vary the quantity of ligand added by determining the percentage of the existing unliganded receptor monomers to change their type to liganded (Fig 3). 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. 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. It generates an animation by depicting the state of the `CellSurface` after each iteration (Fig 4). It is also possible to display two graphs which quantify the data generated by the simulation. One of these shows the state of receptor monomers over time (i.e. the proportion which remain monomers, or which form part of a dimer or larger cluster,

Fig 5, top graph). The other graph displays information about changes in cluster sizes over time (Fig 5, bottom graph).

HOW MOLECULES MOVE

We have incorporated into the design of the simulation the assumption that cell surface molecules move according to the laws governing Brownian motion. When a `Molecule` in the simulation moves, the value for `dx` is added to `x` and `dy` to `y`, resulting in a new location for the `Molecule`. Before each move, a new direction is recalculated by randomly generating new values for `dx` and `dy` based on the overall distance the `Molecule` is allowed to move per timestep. The velocity at which a `Molecule` moves is inversely proportional to the square root of its molecular weight. Thus, “speed” (i.e. the distance moved in any one step) is a constant dependent on the number of subunits.

Brownian motion of a population of particles moving in two dimensions will fit the equation $\langle r^2 \rangle = 4Dt$, where $\langle r^2 \rangle$ is the mean square distance travelled per unit time (t), given a diffusion coefficient D . D for receptor monomers was assigned a value of $5 \times 10^{-9} \text{cm}^2/\text{s}$, which is estimated to be the diffusion coefficient for a small protein in a lipid bilayer (5). In order to test for Brownian motion of `Molecules` in the simulation, 10,000 `Molecules` (monomers) were created and moved for increasing number of iterations (as number of iterations is proportional to time), and the results plotted as $\langle r^2 \rangle$ vs. t . As predicted, the plots were linear. When the tests were repeated using `Multimers` with different numbers of subunits, the slope of the lines varied in proportion to the molecular weight as expected (i.e. the square of the speed of the `Molecules` is inversely proportional to number of subunits) (Fig. 6).

However, these tests were run with the `Molecules` moving in isolation from each other, which is not what occurs during the simulation. Therefore, it was necessary to perform tests similar to the ones described above in which collisions between `Molecules` would be taken into account. No receptor aggregation took place as this would complicate the analysis by introducing the additional factor of the presence of `Molecules` of different sizes in the same test. The density of receptor `Molecules` is the same as that used in the simulation itself. The results demonstrate that overall, at this relatively sparse concentration of objects, collisions between them had no effect (Fig. 7). It would be interesting at some point to determine the density at which the motion of `Molecules` becomes non-ideal.

CONTROLLING RECEPTOR INTERACTIONS

Receptor aggregation is controlled by altering the values in the `AffinityTable`. The values for these probabilities determine how the individual `Molecule` objects interact with each other. There are values for binding, which determine the likelihood of a particular species forming, and dissociation values, which determine the stability of the complex formed. For example, using the default values, two liganded monomers have a very high probability of forming dimers, which have an extremely low probability of dissociation, rendering them very stable (Fig. 8A). Large clusters can be built up in two ways: by the sequential addition of other dimers to a cluster (Fig. 8B) or by the interaction and binding of two smaller clusters (Fig. 8C). When a cluster with an even number of subunits has a dissociation event, a dimer is released, leaving a cluster whose size is reduced by two subunits (Fig 8B,C).

Binding and dissociation probabilities can be manipulated to prevent unwanted events from occurring. For example, there is little evidence of dimerisation between unliganded monomers. `Molecules` of this type therefore have a moderate probability of binding to form unliganded dimers, which have a moderate probability of dissociation, rendering them relatively unstable (Fig. 8D). Because receptor aggregation is a downstream event of receptor dimerisation, it seem unlikely that there would be many clusters with odd-numbers of subunits. Thus, the probability of a monomer binding to a dimer (or a larger cluster with an even number of subunits) is extremely low, and the probability of a monomer dissociating from a cluster so formed is very high, making it very unstable (Fig. 8E). Thus, `Molecules` of this type are unlikely to play a large role in the receptor aggregation modelled in this simulation.

RELATING SIMULATION TO REAL PARAMETERS

In order for the simulation to be accurate, it was necessary to correlate simulation measurements (in pixels) to distances on a cell surface (in metres). We estimate that the total area of a cultured cell is approximately $4 \times 10^{-10} \text{m}^2$. The simulation display covers an area of 600×700 pixels ($4.2 \times 10^5 \text{ pixel}^2$) and contains 500 receptor monomers. We assume that the `CellSurface` represents approximately 1/200 of

the total cell membrane, based on the fact that the average cell will contain approximately 50,000-100,000 EGFR. This means the CellSurface represents approximately $2 \times 10^{-12} \text{m}^2$. Thus, a unit of area represented by a pixel is $4.8 \times 10^{-18} \text{m}^2$ and the distance represented by a pixel in one dimension is equivalent to $2.2 \times 10^{-9} \text{m}$. If we use a value of one pixel for the diameter of a monomer, this translates to an area occupied by a monomer of $(\pi r^2) = 3.8 \times 10^{-18} \text{m}^2$. This is a fairly accurate representation of an EGFR monomer, which has an area of approximately $4 \times 10^{-18} \text{m}^2$.

In order to compare our simulation results to experimental results, it is necessary to convert the time measurement from number of iterations to units of real time. For this, we return to $\langle r^2 \rangle = 4Dt$. For monomers, we know that $\langle r^2 \rangle = 1.2 \times 10^{-12} \text{cm}^2$, because this is the value set as a constant for the distance travelled per move. Plugging this and the value for D ($5 \times 10^{-9} \text{cm}^2/\text{s}$) into the equation, we calculate that one complete iteration represents $60 \mu\text{s}$. In the course of a complete iteration, all Molecules in the simulation have moved once.

SIMULATION OUTPUT

We can demonstrate using the simulation that ligand concentration has an effect on aggregation. At saturating concentration of ligand (i.e. 100% receptor occupation) after a very short time, virtually all receptor monomers have become bound up in aggregates with cluster size increasing over time (Fig. 9C). As expected, at lower ligand concentrations, the number of receptor monomers which remain in this state is proportional to the amount of ligand added (Fig. 9A,B, upper graphs), as unliganded monomers have a much lower probability of aggregating. This leads to overall lower numbers of clusters and smaller cluster sizes after the simulation is complete (Fig 9A, B, lower graphs).

Running the simulation under different conditions shows that other parameters can affect receptor aggregation. Simulations were run in which the value of one entry in the AffinityTable was changed to investigate the effect on receptor aggregation. For example, three simulations were run with different values for the probability of two clusters aggregating. Using saturating concentration of ligand and varying the probability between 0 (i.e. clusters are only formed by sequential aggregation of dimers) up to a relatively high probability of 0.35 demonstrates that this parameter does affect the output. At all three probabilities for cluster + cluster

aggregation, most monomers rapidly aggregate into clusters (Fig 10, upper graphs). However, the rate at which cluster size increases is affected by this value. As expected, at the highest value, large clusters form more rapidly. However, with each of the three values, the pattern of changes in cluster size is fairly similar and clearly has not reached equilibrium at the end of the simulation (Fig10, lower graphs). It may be possible, therefore, that the final state of receptor aggregation at equilibrium may be the same regardless of how the large clusters are allowed to form.

In contrast to this, changing the value for the probability of cluster dissociation (i.e. release of a dimer from a larger aggregate) has a significant effect on the results. When no dissociation is allowed, virtually all monomers aggregate rapidly, and no dimers or isolated monomers remain (Fig 11A, upper graph). Cluster size is still on the increase at the end of the simulation (Fig 11A, lower graph). If however, dissociation probability is increased to 0.02, the rate at which monomers and dimers aggregate is significantly decreased (Fig 11B, upper graph). As in the case of cluster aggregation, this may be merely a kinetic effect, with equivalent equilibrium states. However, increasing the probability of dissociation to 0.35 has a profound effect. By the end of the simulation, 80% of monomers have dimerised, but very few aggregates have formed (Fig. 11C, upper graph). This appears unlikely to reach an equilibrium state similar to simulations using the first two values, as throughout the course of the simulation, cluster size does not appear to increase (Fig. 11C, lower graph).

SCALING UP

In order to be able to relate the results of a simulation run to experimental results, it was necessary to scale up the simulation to represent the receptor molecules on an entire cell. Simply, this can be done by initialising the cell surface with 50-100K monomers, for example, and increasing the size of the `CellSurface` proportionally. However, in order to make this computationally efficient and to enable the entire simulation to be run in a reasonable amount of time, it was necessary to separate the simulation engine from the display (which in the first version are linked in the same class). The display is quite useful, both in terms of allowing the user to get an intuitive feel for how the clustering happens and as a development tool to allow visualisation of the motion and interaction of receptors, insuring that changes to the program have the desired effect. However, the scaled-up version in its current

form runs from a command line, and lacks an animated display. When the simulation is complete, the data graphs are shown.

To reduce the time needed to run a large scale simulation, it was necessary to find a more efficient way to deal with collisions. When the simulation only involves 500 Molecules, it is not unreasonable to check every other Molecule in the simulation for collision. However, when the number of objects in the simulation is increased significantly, it is unfeasible to proceed in this way.

The solution to this problem is to associate the CellSurface with a Grid object. This consists of a lattice of points covering the entire area corresponding to the CellSurface. Each point is “responsible” for the rectangle for which it is the upper left corner point. Each Molecule keeps a reference to the GridPoint(s) to which it belongs, and each GridPoint keeps references to the Molecule(s) it contains. This information is automatically updated every time a Molecule moves. After a Molecule has moved and updated its GridPoints, it makes a collection of other Molecules which share these GridPoints to check for collision.

Tests were run to compare the efficiency of the two methods. Each test involved moving a given number of Molecules through a set number of moves (including a collision check) and measuring execution time. When a Grid was used, the execution time remained more or less constant for a given number of moves regardless of the number of Molecules in the test. In contrast, the execution time increased exponentially if every other Molecule was checked for collision (Fig. 12A).

Tests were also run to determine the ideal Grid tile size. Too large a tile size would necessitate checking Molecules not in reasonable colliding distance of the Molecule in question. In the case of excessively small tiles, A Molecule, particularly a large Multimer, may belong to several GridPoints which it would fill entirely, but which would nonetheless need to be checked for the presence of others. To determine ideal tile size, 1000 Molecules were moved 10^5 times, using a Grid for collision checking. Square tiles were used, and in each test, length of each tile dimension was a multiple of the diameter of the Molecules in the test. Execution times were compared running this test with different size tiles. The results show that the ideal Grid tile dimension is between 2-4 times the Molecule diameter (Fig. 12B). We have not had a chance to put this to use in the simulation

due to the fact that aggregation causes the average `Molecule` size to increase during the course of the simulation, making the determination of ideal tile size difficult.

FULL-SCALE SIMULATION RESULTS

A simulation was run using a `CellSurface` initialised with 100,000 `Molecules` (monomers). The area of the `CellSurface` was equivalent to $4.3 \times 10^{-10} \text{m}^2$. After addition of 100% ligand, the simulation was run for 10^6 iterations, which represents approximately 60s in real time. By the end of the simulation, there were 500 clusters with an average size of 200 subunits (Fig.13, upper graph). This accounts for virtually all of the initial monomers in the simulation, indicating that they have all aggregated. Looking at the lower graph, it would appear that the system has reached equilibrium, as all the curves are flat. In contrast to this, the upper graph shows that average cluster size is still on the increase. This curve is levelling off as well, so it appears that the simulation is approaching, but has not yet reached equilibrium. This discrepancy can be easily corrected by changing the ranges for cluster sizes in the lower graph, raising the upper limit (currently at 120 subunits) so that this curve reaches equilibrium at the same point as average cluster size.

FUTURE PLANS

This simulation is still very much a work in progress. There are many aspects of the underlying structure of the simulation which still require refinement in order to improve the usefulness of the results. Presently, the simulation runs for a set number of iterations and stops. Results clearly show that this number of iterations is insufficient to achieve an equilibrium state. Ideally, we need to define a value (average cluster size, for example) which will be used as the basis for determining when equilibrium is reached. The program should automatically detect when changes to this value over time fall below a certain threshold, and stop the simulation at this point. We also need to define specifically what data is required from the simulation. This will relate to the types of measurements possible using the imaging software to allow for comparison of experimental and simulation results. Other improvements to the simulation include having more than one type of receptor in the simulation at one time to enable study of interactions between different members of the Type 1 receptor family and more realistic long-term factors such as ligand dissociation and receptor recycling.

ACKNOWLEDGEMENTS

I would like to thank Dennis Bray, Bill Gullick and Colin Johnson for helpful discussions. This work was funded in part by the Medical Research Council.

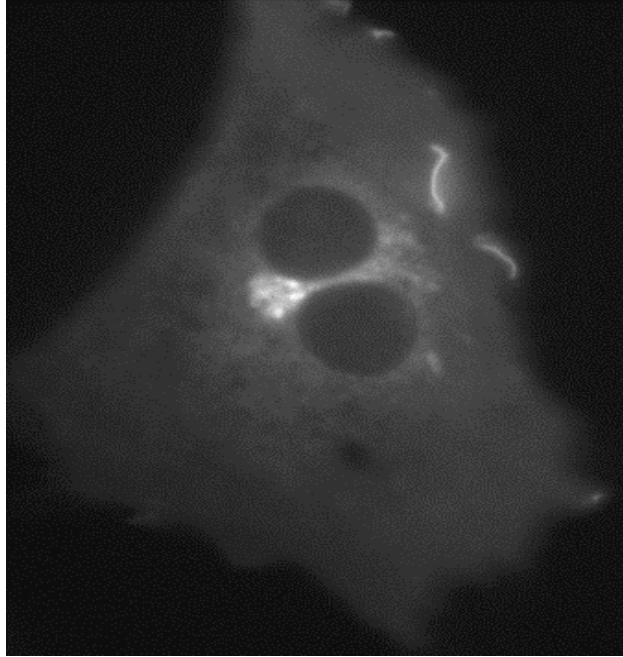
Figure 1: Growth Factor Receptor Clustering

(A) prior to ligand addition

(B) after ligand addition

(from (4), reproduced with permission of W.J.Gullick)

A.



B.

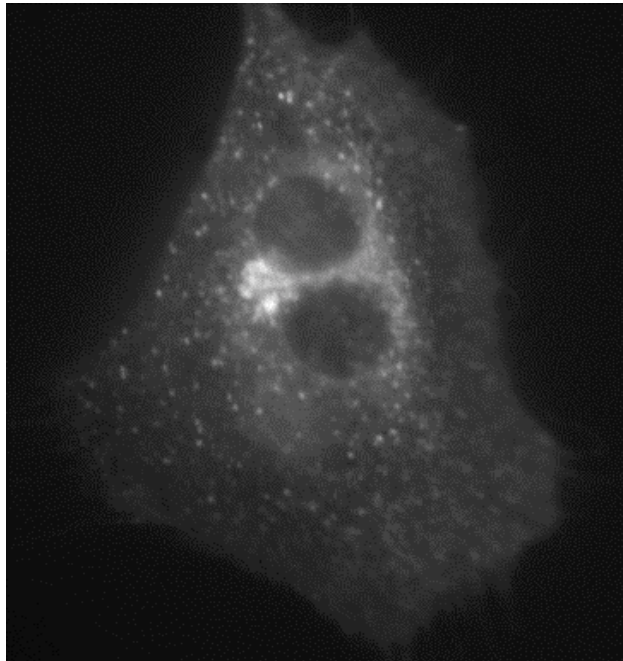


Figure 2: UML class diagram for simulation

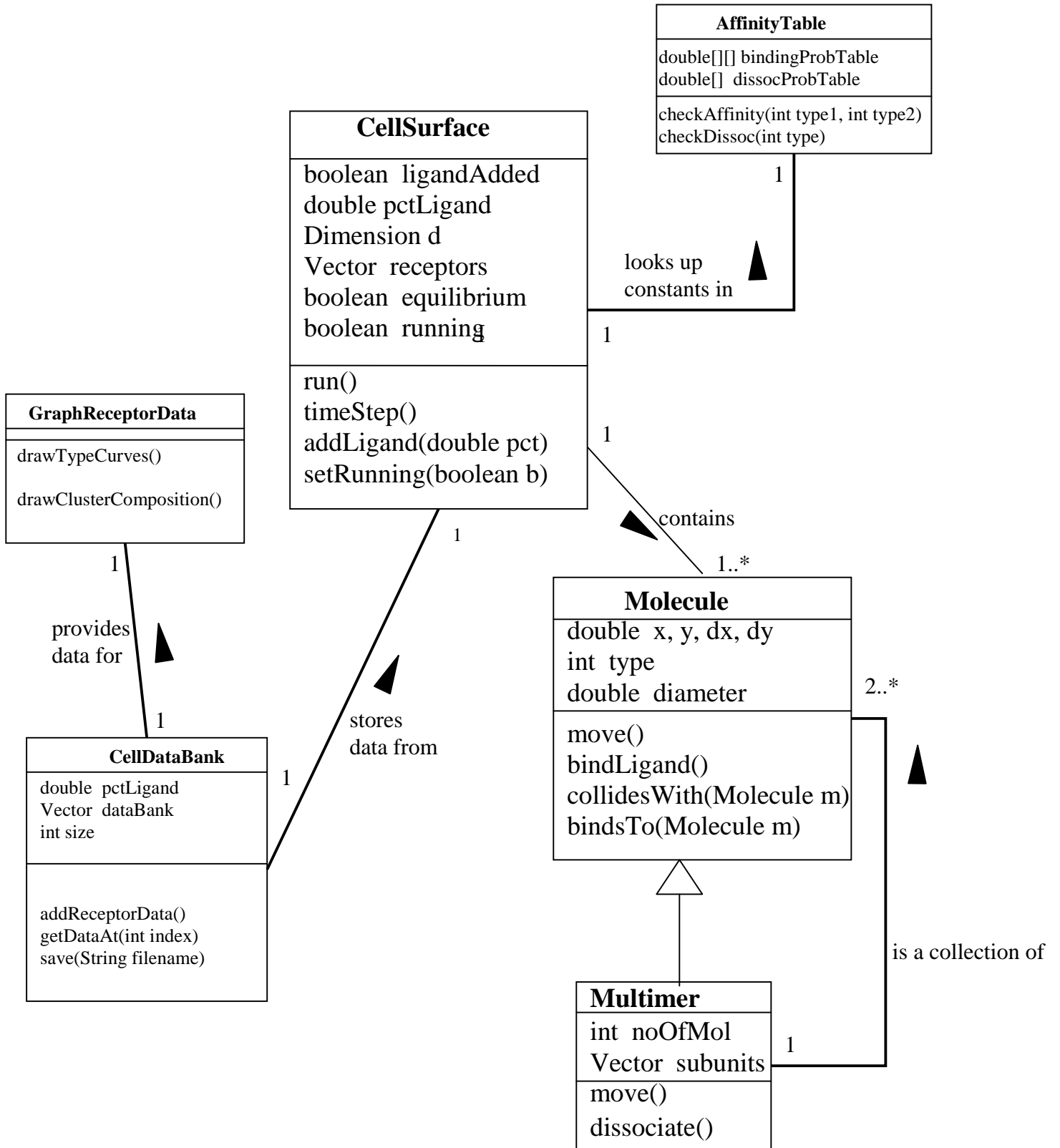
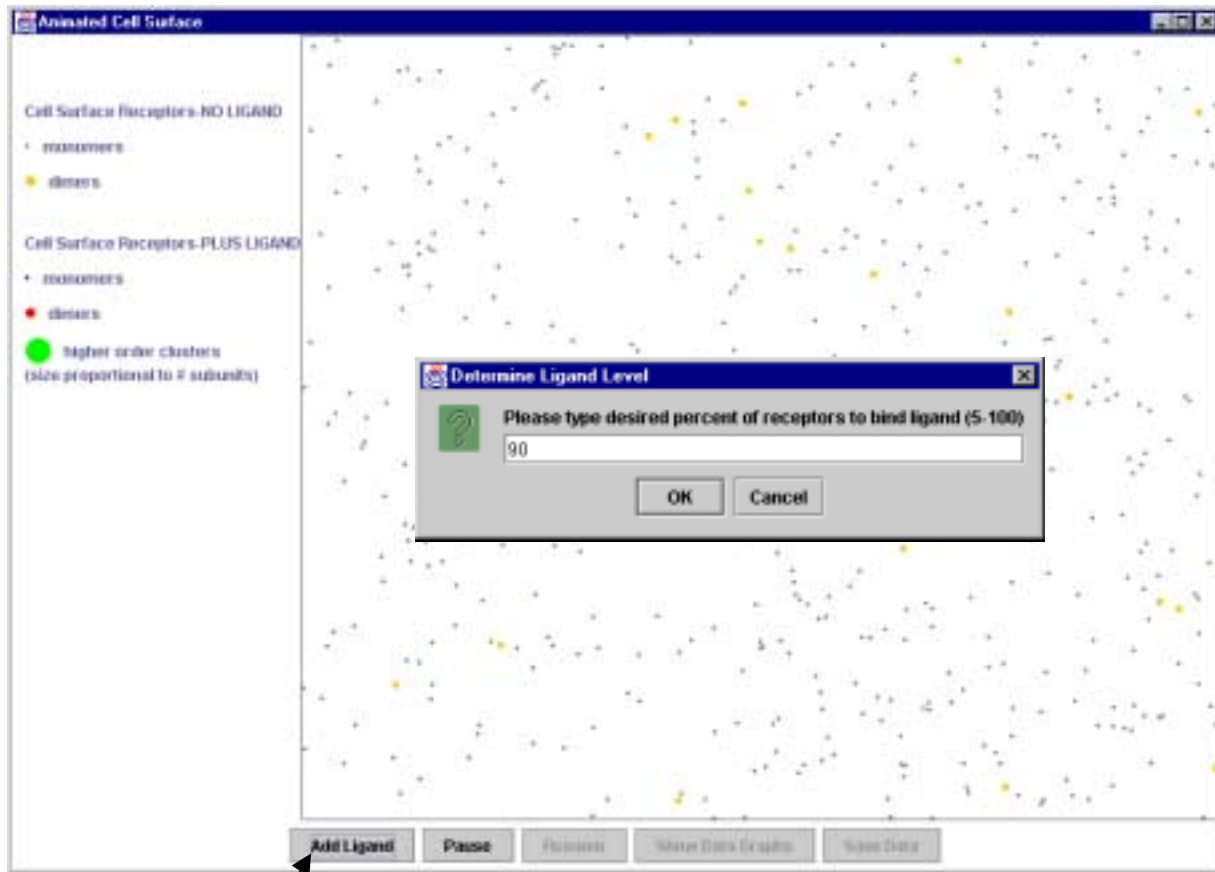


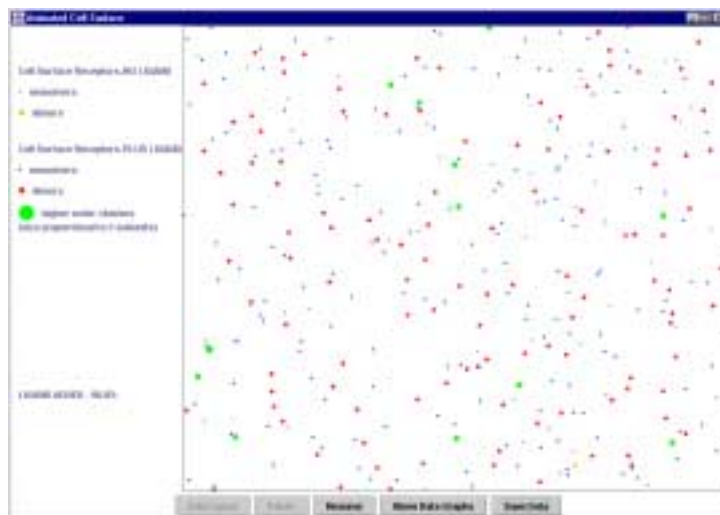
Figure 3: Adding Ligand to a CellSurface



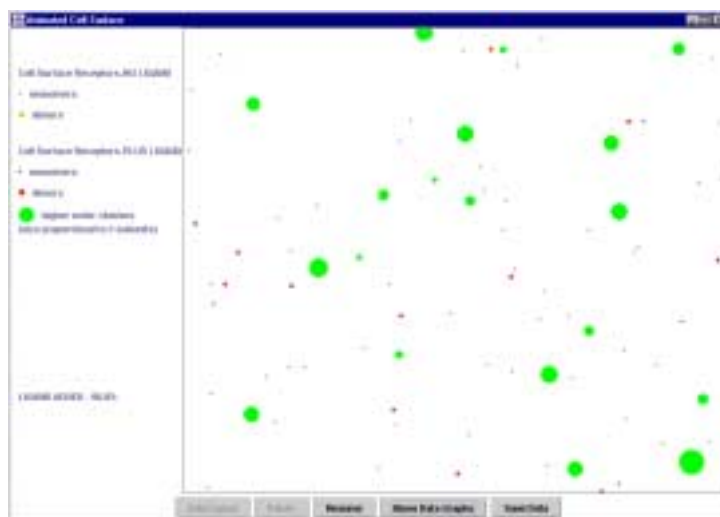
Click here to bring up dialog box "Determine Ligand Level"

Figure 4: Still images from simulation with 90% ligand. (A) Shortly after ligand addition (B) Approximately halfway through the simulation (~12,000 iterations) (C) Simulation complete (25,000 iterations)

A.



B.



C.

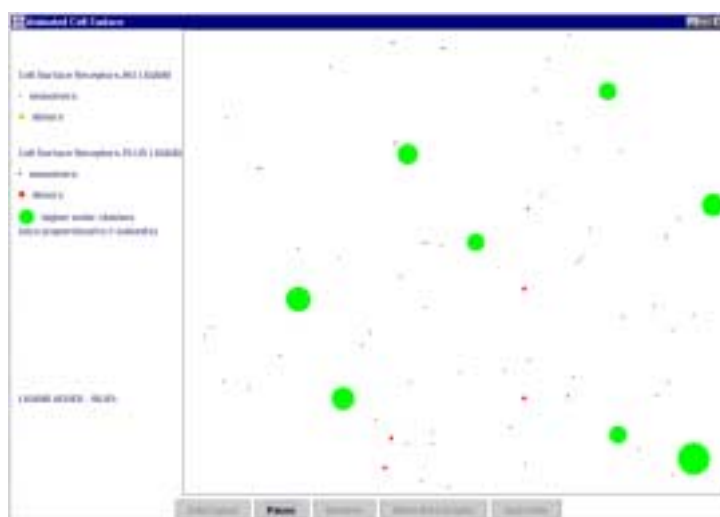


Figure 5: Graphical display of simulation data.
Simulation initialised with 500 monomers and run for 25,000 iterations after addition of 90% ligand

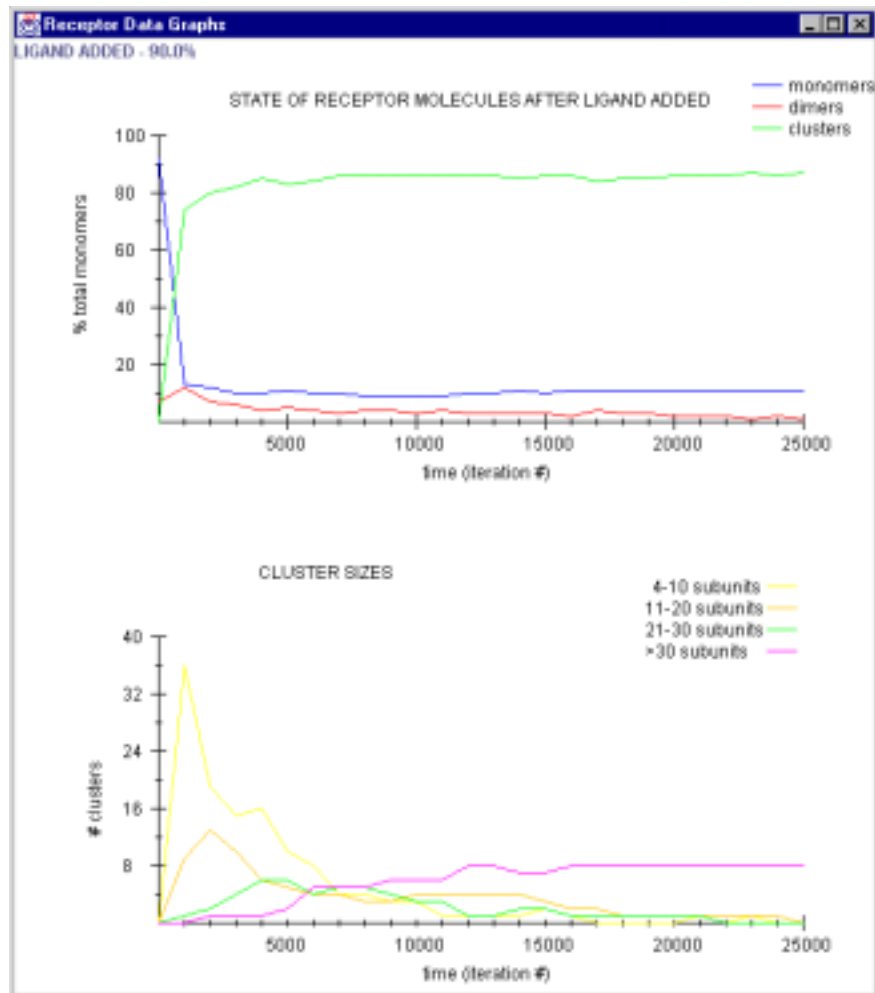


Figure 6: Plots from tests of Brownian Motion.
 $\langle r^2 \rangle$ vs. t for (A) 1mers-10mers, and (B) 20mers-50mers.

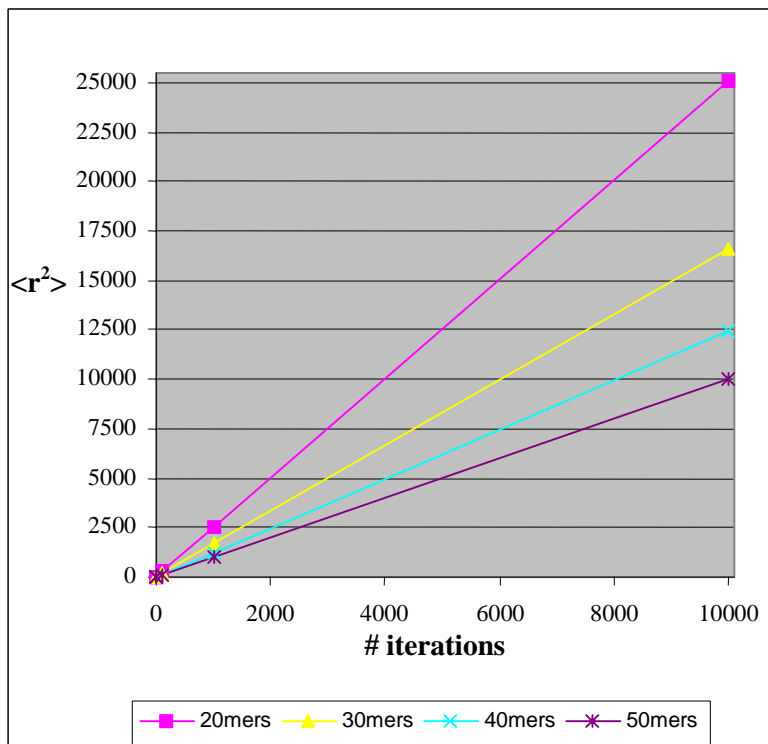
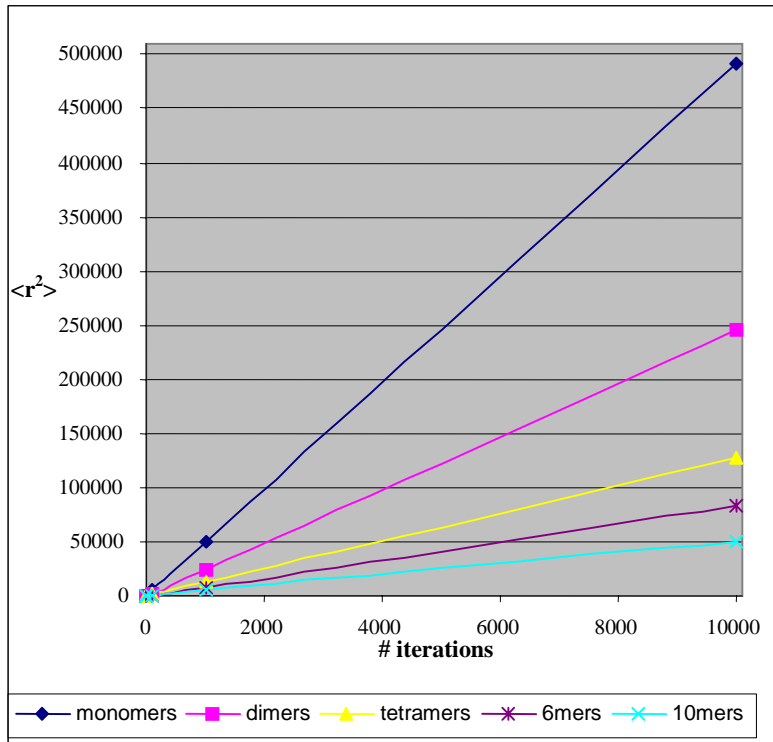


Figure 7: Test for Brownian motion with collisions. Results shown are for monomers. Ideal curve shows results from Fig 6A (for monomers). Avg curve represents the plot of average $\langle r^2 \rangle$ from six independent tests (each using 1000 Molecules) vs. t. Comparable results were produced running tests using larger Molecules (2mers-50mers, data not shown).

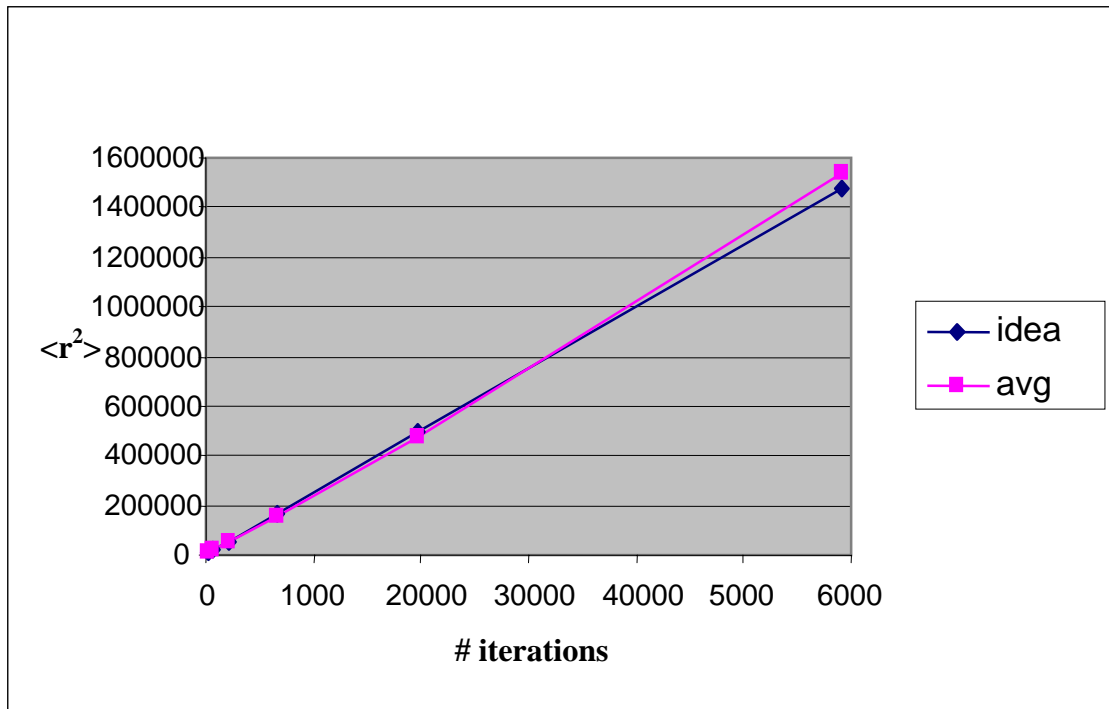


Figure 8: Examples of Binding and Dissociation Probabilities.

(A) Formation of liganded dimers (●●) from liganded monomers (●).

Formation of clusters (even # subunits, ●●●●) from dimers (B) and other clusters (C).

(D) Formation of unliganded dimers (●●) from unliganded monomers (●).

(E) Formation of clusters (odd # subunits, ●●●).

Forward arrows indicate probability of binding, reverse arrows indicate probability of dissociation

* probability of dissociation is value in `AffinityTable` divided by the number of dimers in the cluster

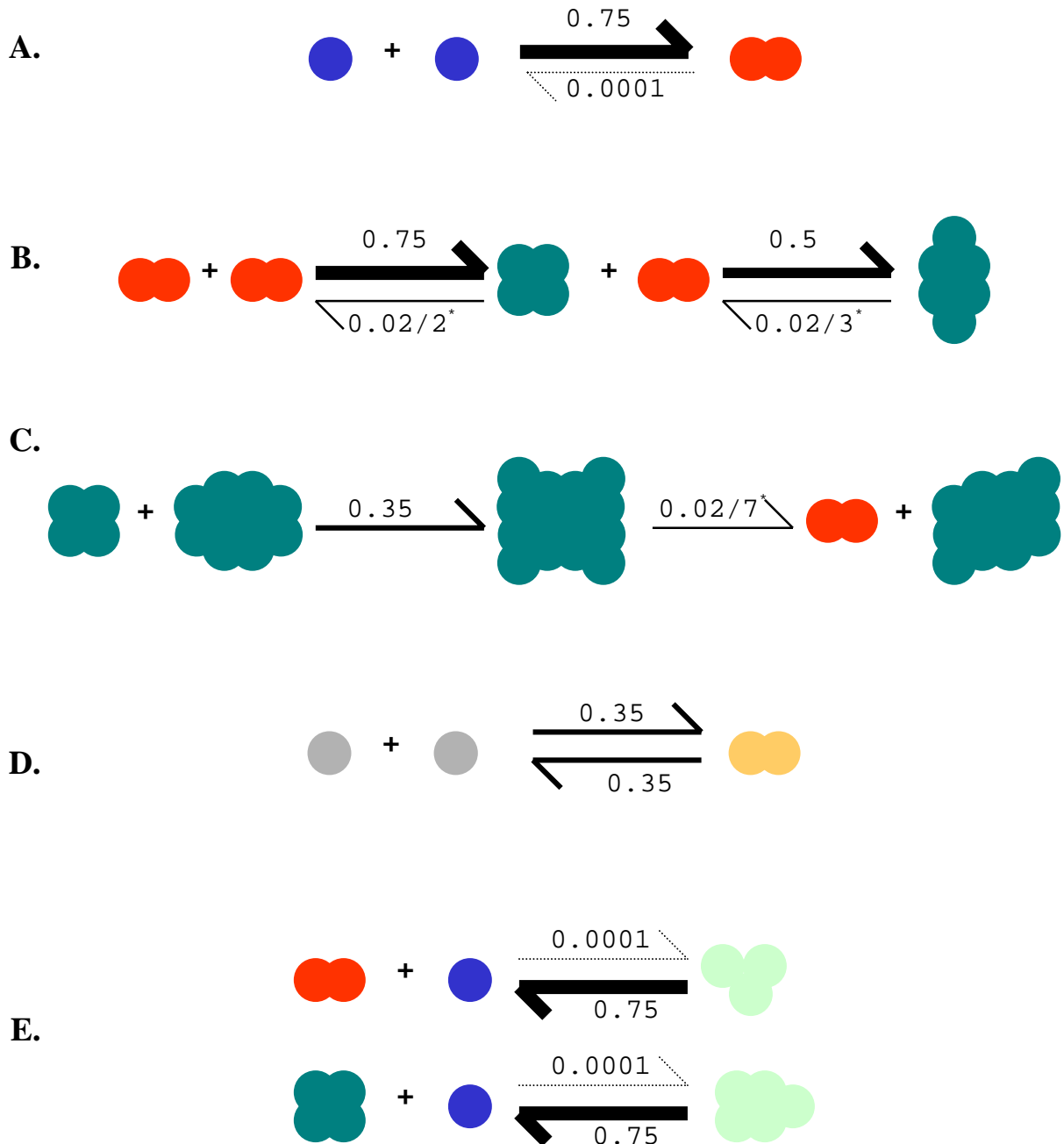
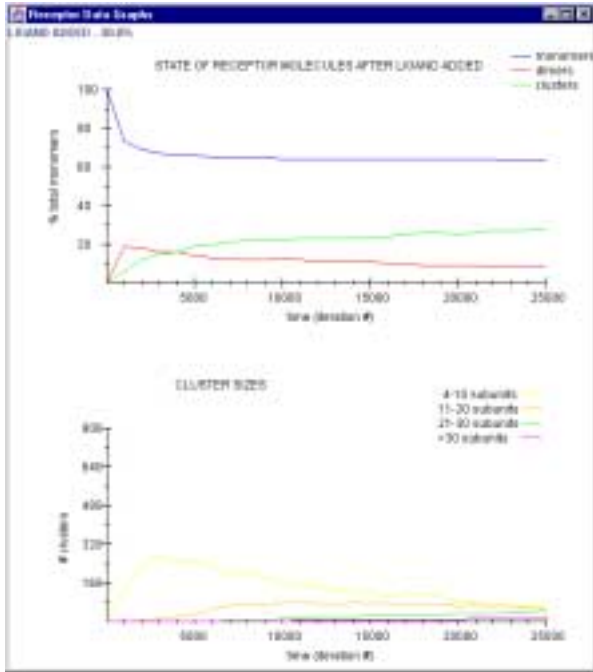
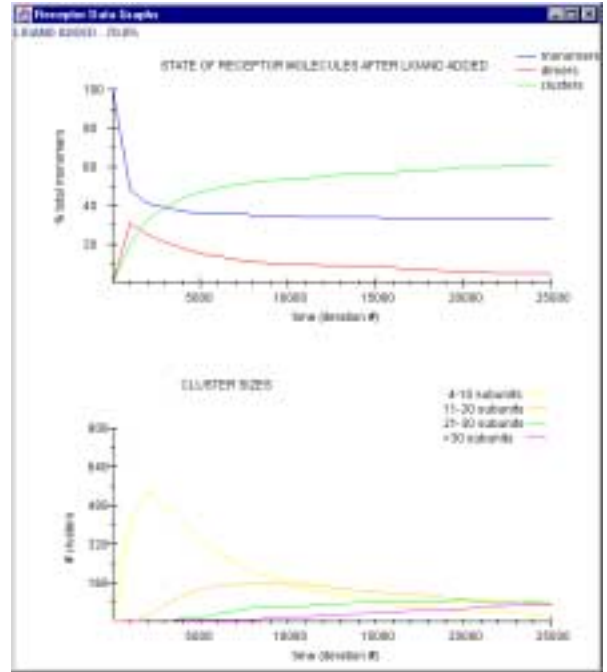


Figure 9: Effect of ligand concentration on Molecule aggregation.
 Data generated from simulations run with ligand concentrations of 40% (A), 70% (B) or 100% (C)

A.



B.



C.

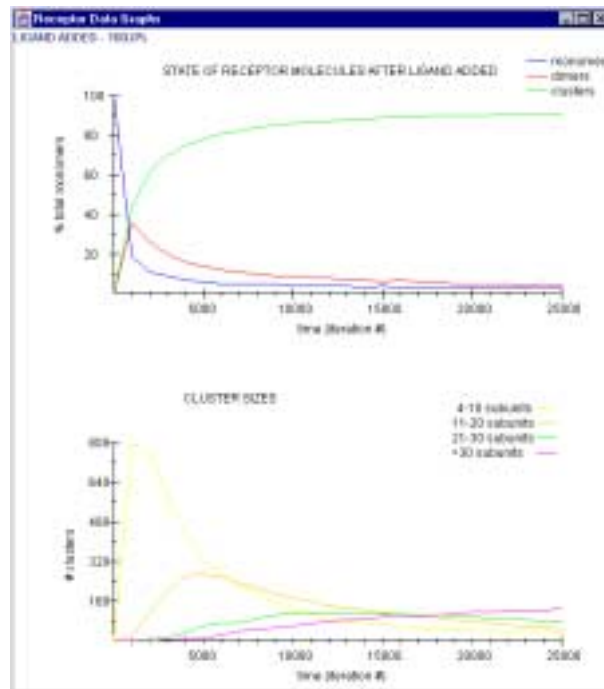
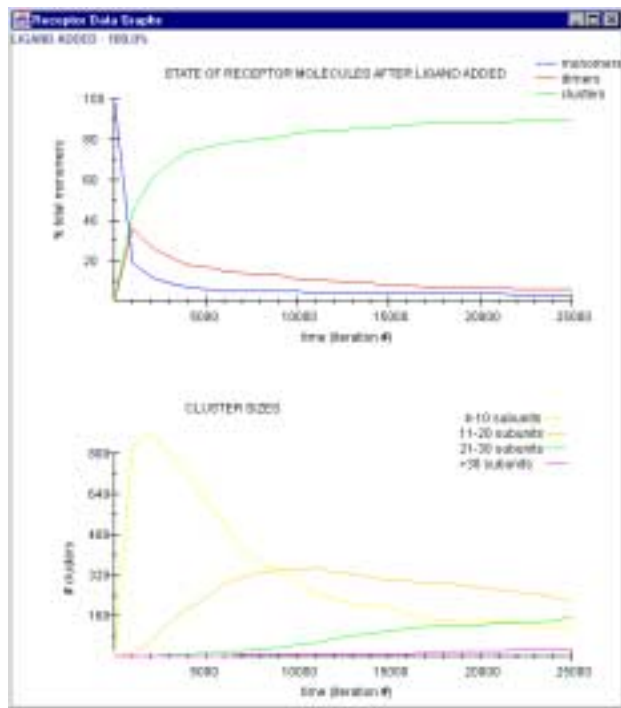
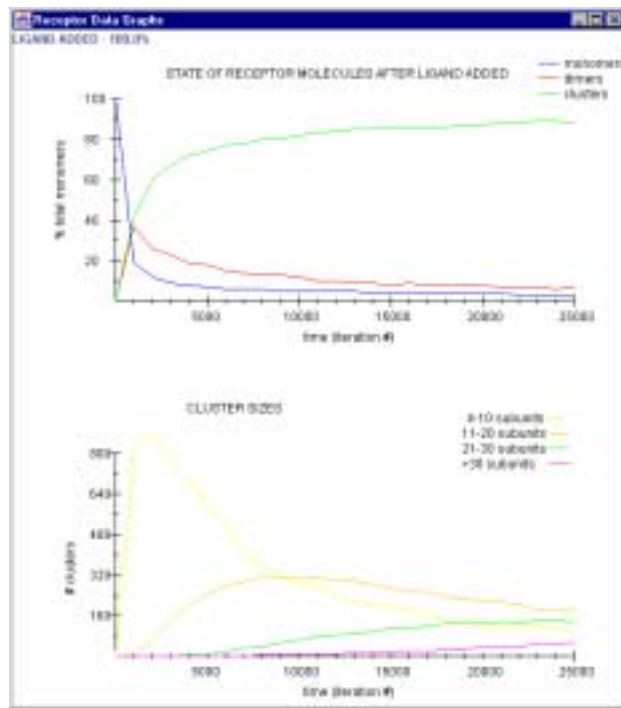


Figure 10: Effect of varying probability of cluster+cluster binding (see fig 8C, first step). Data generated from simulations in which this probability was set to 0 (A), 0.02(B) or 0.35(C).

A.



B.



C.

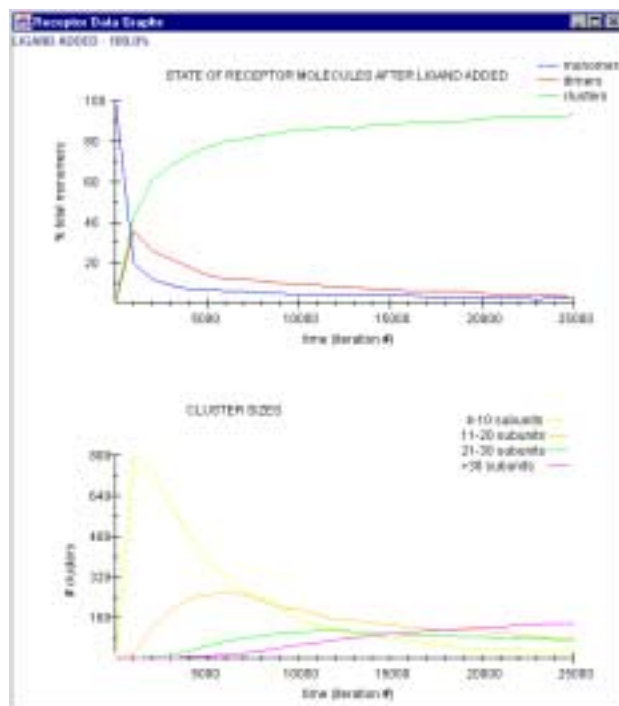
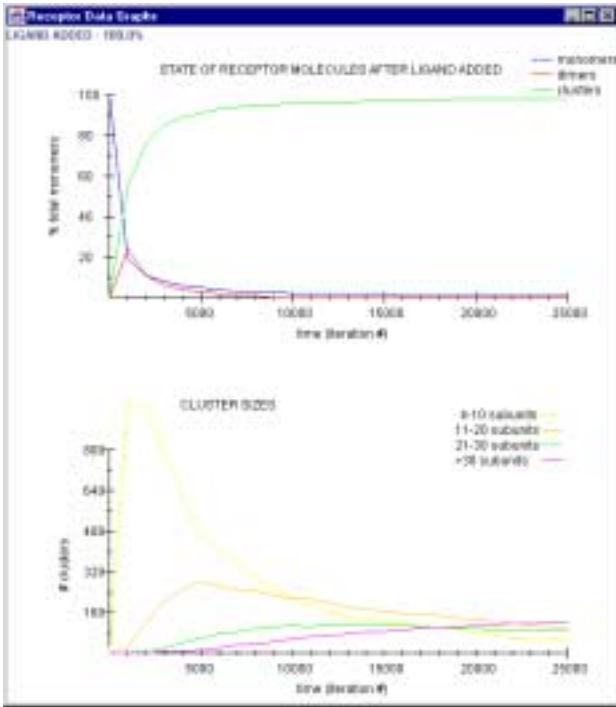
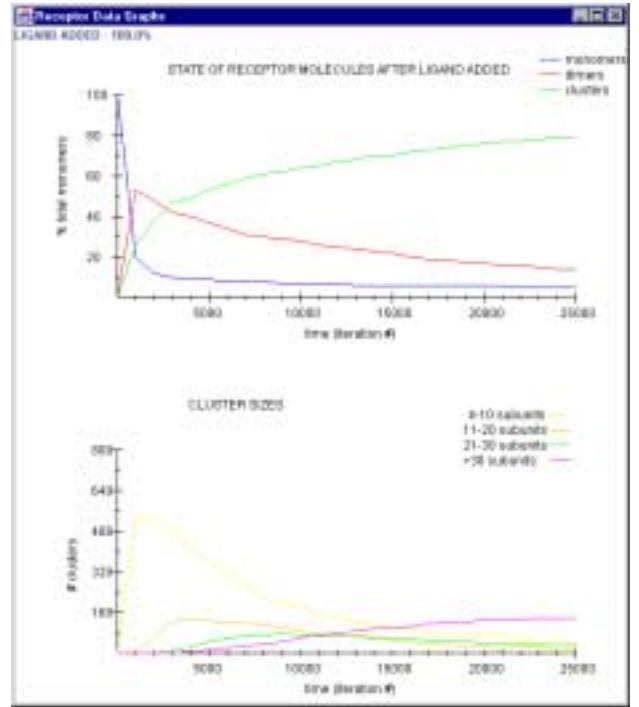


Figure 11: Effect of varying probability of cluster-dimer dissociation (see fig 8C, second step). Data generated from simulations in which this probability was set to 0 (A), 0.02(B) or 0.35(C).

A.



B.



C.

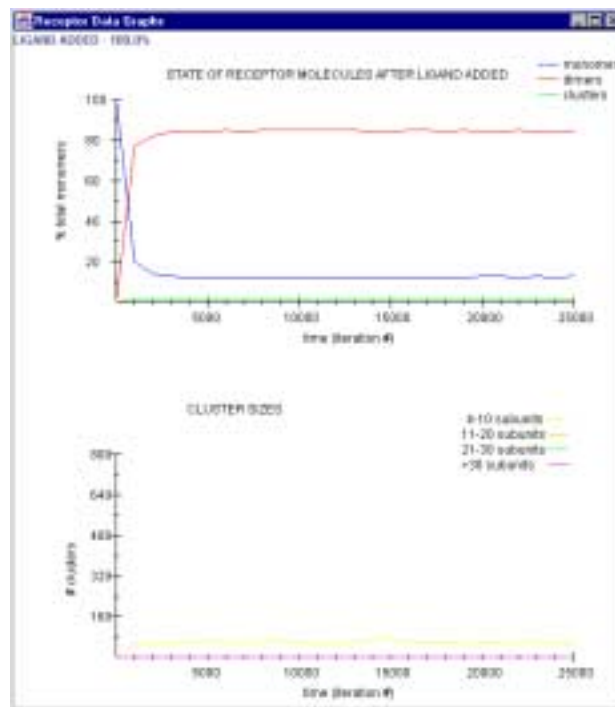


Figure 12: Using a Grid for collision detection.

(A) Execution time vs. number of Molecules for different number of moves. Each move involves choosing a Molecule at random to move and checking for collision using the indicated method.

(B) Determining ideal GridPoint tile size. See text for description of test.

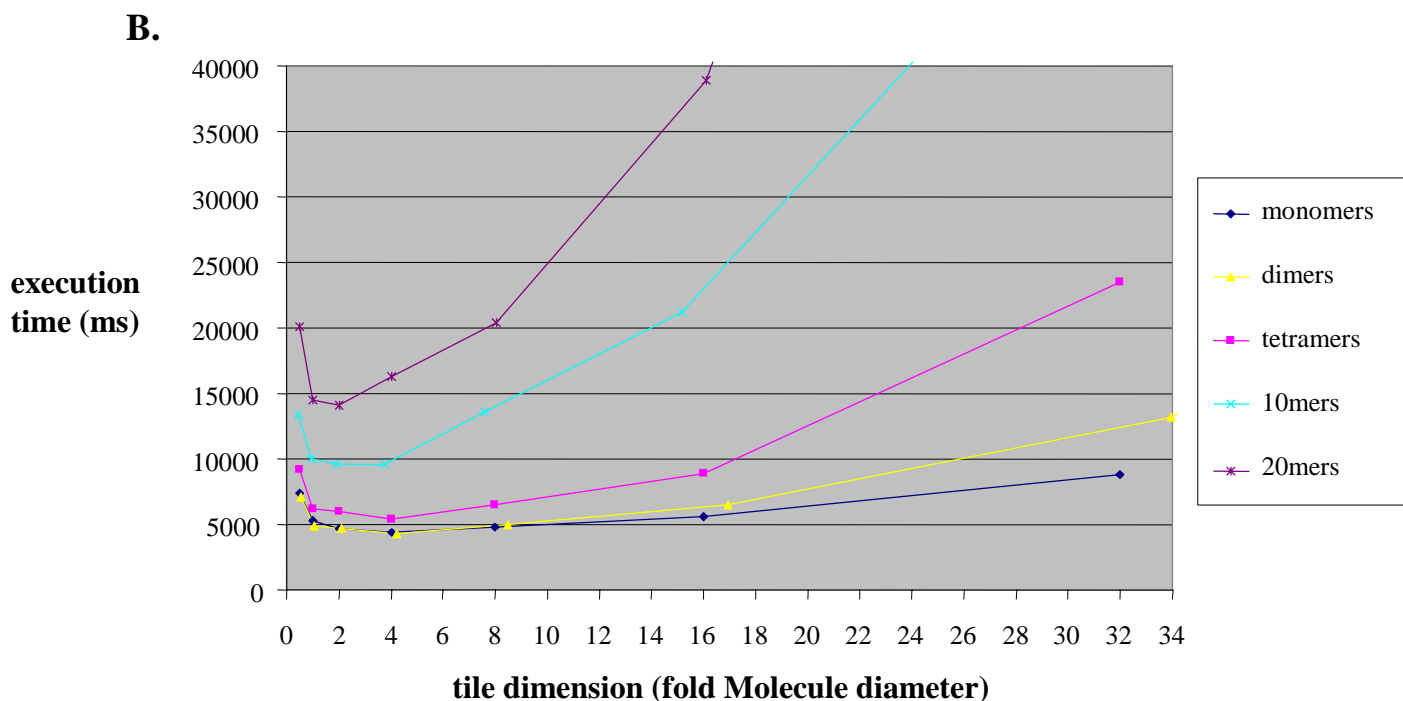
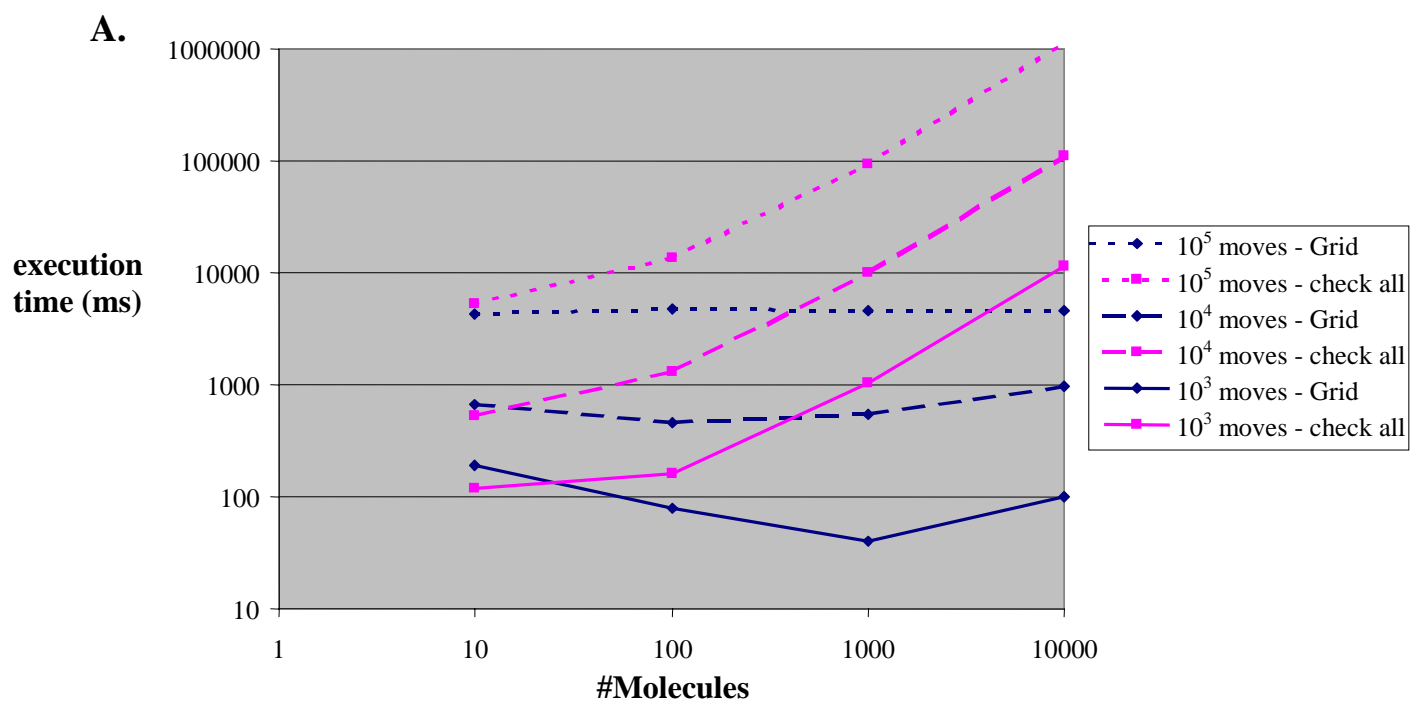
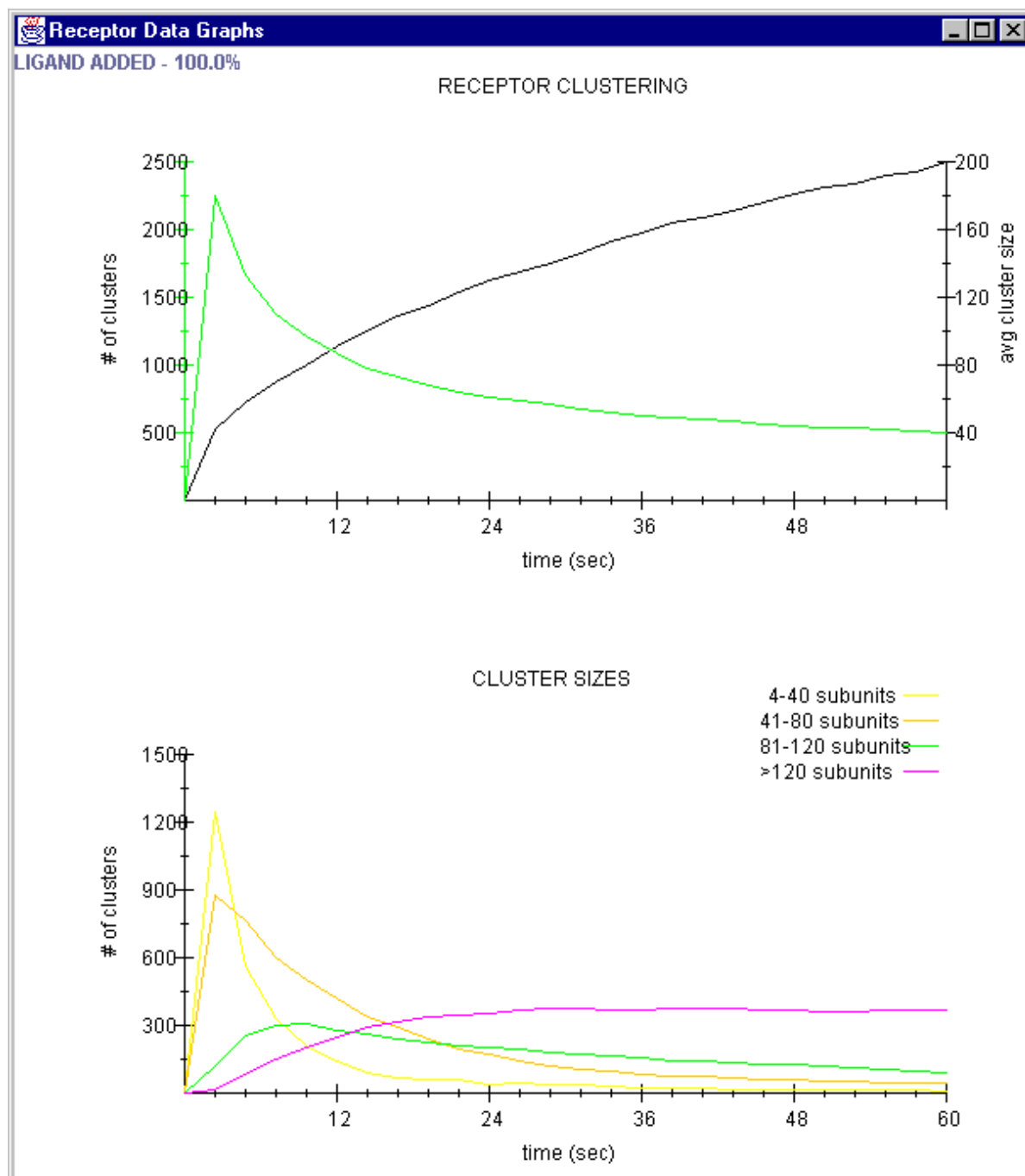


Figure 13: Results from a full-scale simulation run with 100,000 monomers for 10^6 iterations (equivalent to 60 s in real time).



REFERENCES

1. Mendelsohn, J., and Baselga, J. (2000) The EGF receptor family as targets for cancer therapy. *Oncogene* **19**, 6550-65.
2. Salomon, D.S., Brandt, R. Ciardiello, F. and Normanno, N. (1995) Epidermal growth factor-related peptides and their receptors in human malignancies. *Critical Reviews in Oncology/Hematology* **19**, 183-232.
3. Olayioye, .A., Neve, R.M., Lane, H.A., and Hynes, N.E. (2000) The ErbB signalling network: receptor heterodimerisation in development and cancer. *EMBO Journal* **19**, 3159-67.
4. Gullick, W.J. (2001) The Type 1 growth factors and their ligands considered as a complex system. *Endocrine-Related Cancer* **8**, 75-82.
5. Berg, H.C. (1993) *Random Walks in Biology*, Princeton University Press.