

SOFTWARE

Open Access

APM_GUI: analyzing particle movement on the cell membrane and determining confinement

Silvia A Menchón^{1,2*}, Mauricio G Martín^{1,3} and Carlos G Dotti^{1,3*}

Abstract

Background: Single-particle tracking is a powerful tool for tracking individual particles with high precision. It provides useful information that allows the study of diffusion properties as well as the dynamics of movement. Changes in particle movement behavior, such as transitions between Brownian motion and temporary confinement, can reveal interesting biophysical interactions. Although useful applications exist to determine the paths of individual particles, only a few software implementations are available to analyze these data, and these implementations are generally not user-friendly and do not have a graphical interface.

Results: Here, we present APM_GUI (Analyzing Particle Movement), which is a MatLab-implemented application with a Graphical User Interface. This user-friendly application detects confined movement considering non-random confinement when a particle remains in a region longer than a Brownian diffusant would remain. In addition, APM_GUI exports the results, which allows users to analyze this information using software that they are familiar with.

Conclusions: APM_GUI provides an open-source tool that quantifies diffusion coefficients and determines whether trajectories have non-random confinements. It also offers a simple and user-friendly tool that can be used by individuals without programming skills.

Background

Informatics tools have become essential for biologists because they allow the analysis of a vast amount of data obtained using newly developed techniques and electronic devices. Many informatics tools are now available, which makes it possible to analyze various processes, such as the linear movement of particles, and to quantify fluorescence using easy-to-use software. However, more complex biological processes, such as the multi-directional, evanescent and at times frantic movement of proteins diffusing along the lateral plane of membranes, require sophisticated algorithms that run on complex software distributions. These applications sometimes require a thorough knowledge of programming, which makes their use difficult for biologists. The aim of this work is to provide a user-friendly, accessible

and open-source tool for the analysis of data from two-dimensional single-particle tracking (SPT) experiments.

SPT is a useful technique for the study of the dynamics and movement of individual sub-micrometer-sized particles. SPT is frequently used to study the movement of receptors on the cell surface [1]. The development of quantum dots (Qdots) and the improved sensitivity of fluorescence microscopy allow the localization of each particle with a high resolution and precision for long periods of time [2]. SPT occurs in two important and distinct stages: first, the dots (representing the particles' positions) are localized and marked, and second, the dots are tracked over time [3]. The trajectories for each particle are obtained by trace reconnection. Although many sources of noise are present throughout the process, (e.g., the blinking of Qdots, which generates dark states), some algorithms are able to label and track the marked particles. The most commonly used implementations for Qdots are MTT (a set of MatLab scripts [4]), Particle Tracker (a ImageJ plug-in [5]) and Imaris Track (Bitplane's software [6]). However, it is difficult to find a user-friendly application with a graphical interface

* Correspondence: menchon@famaf.unc.edu.ar; cdotti@cbm.uam.es

¹Department of Molecular and Developmental Genetics, VIB Center for the Biology of Disease and Center for Human Genetics, KULeuven, Campus Gasthuisberg, Herestraat 49 - bus 602, 3000 Leuven, Belgium
Full list of author information is available at the end of the article

that can analyze the dynamic properties of trajectories, such as variations in protein diffusion coefficients.

Here, an implementation for MatLab with a graphical user interface (GUI) is presented that allows the analysis of lateral diffusion properties using data obtained by SPT in two dimensions. In particular, it determines whether a particle has free Brownian motion or non-random confinement based on how long the particle stays in a given region [7,8]. When a particle trajectory shows regions with confined movement, our implementation determines where it occurs, the sizes of the confined regions, the duration of the time that a particle stays in each of these regions and the characteristic diffusion coefficients. Although this application runs on MatLab, (and saves all of the data in a workspace for MatLab users), all generated data are also exported in .dat files that can be opened with any software and subjected to statistical analyses, which allows the users to analyze and plot this information using software that they are familiar with.

Implementation

APM_GUI is a complete GUI tool created using MatLab language and GUIDE. The MatLab development is platform-independent. The installation consists of unpacking the compressed folder that contains a collection of MatLab scripts and functions to the desired location and adding this location to the MatLab path. It is possible to add directories to the MatLab path by selecting *Set Path at File* in the menu bar (and then *Add folder*). If this folder is not added permanently to the MatLab path, these steps must be performed each time MatLab is started. No manual compilation is necessary because the graphical interface appears after APM_GUI is written in the MatLab Command Windows and is composed of three panels: *Converting files*, *Analyzing* and *Exporting a trajectory*, as shown in Figure 1. APM_GUI is distributed using a short user manual (see additional file 1). The open-source nature of the software allows expert users to adjust each script according to their own needs and edit the routines running on the GUI.

Data import

Data obtained through SPT experiments must be converted to a MatLab format. The panel *Converting* calls the scripts to convert files from Imaris and ImageJ. An extra script is provided to convert files from MTT (see additional file 1). If Imaris is used to obtain the trajectories, its output has to be saved as .csv file. Then, using the “Browse” button in the panel “From Imaris (csv)”, the .csv files are selected, and their conversion starts after the “Convert” button is pressed in the “From Imaris (csv)” panel. If the Particle Tracker of ImageJ is used to obtain the trajectories, its output has to be saved as .txt

file. Then, the procedure is the same as in the case of data obtained through Imaris, but the “Browse” and “Convert” buttons in the panel “From ImageJ” are used instead.

An algorithm based on the Simson/Saxton approach

To detect the confinement zones on each trajectory, we implement an algorithm reported by Simson and co-workers [7], which is based on Saxton’s work [9] with the addition of a few modifications related to the threshold definition. This algorithm is a well-established method to determine confinement and has been used in recent publications [4,10-12]. On a particle trajectory, confinement zones are defined as the places where the particle remains for a duration of time that is considerably longer than a Brownian diffusant would stay. Thus, we first need to know how long a particle with Brownian motion would stay in a given region. This factor has already been studied by Saxton [9], who determined the following mathematical expression to describe the probability ψ that a particle, with Brownian motion and constant diffusion coefficient D , will stay in a region of radius R for a period of time t :

$$\log(\psi) = 0.2048 - 2.5117Dt/R^2. \quad (1)$$

The coefficients in Eq. (1) can be obtained from the series expansion of the analytical solution (see Appendix A in Ref. [9]).

We are interested in regions where a particle with Brownian motion has a low probability of staying for a period of time t , i.e., regions characterized by the lowest values of ψ . Therefore, it is possible to define a confinement index, L , by [7]:

$$L = \begin{cases} -\log(\psi) - 1 & \psi \leq 0.1, \\ 0 & \psi > 0.1. \end{cases} \quad (2)$$

The index L is always a positive number, and non-random confinement zones are correlated with its highest values. To determine the confinement index for an experimental trajectory, which can be represented as a sequence of points, we proceed by breaking it down into shorter segments. Thus, each point on the trajectory is taken as a starting point for a series of segments with variable length, from four to S_m . Then, to calculate the index L , we define R as the largest displacement from the starting point and t as the duration of the segment, (if the segment has n frames, $t = (n - 1)\Delta$, where Δ is the time between frames). We also take the diffusion coefficient in Eq. (1), D , for each trajectory from the highest instantaneous diffusion coefficient. Because each point on the trajectory is included in many segments, the final value for L at a given point is the average over all segments that it belongs to. Regions for non-random

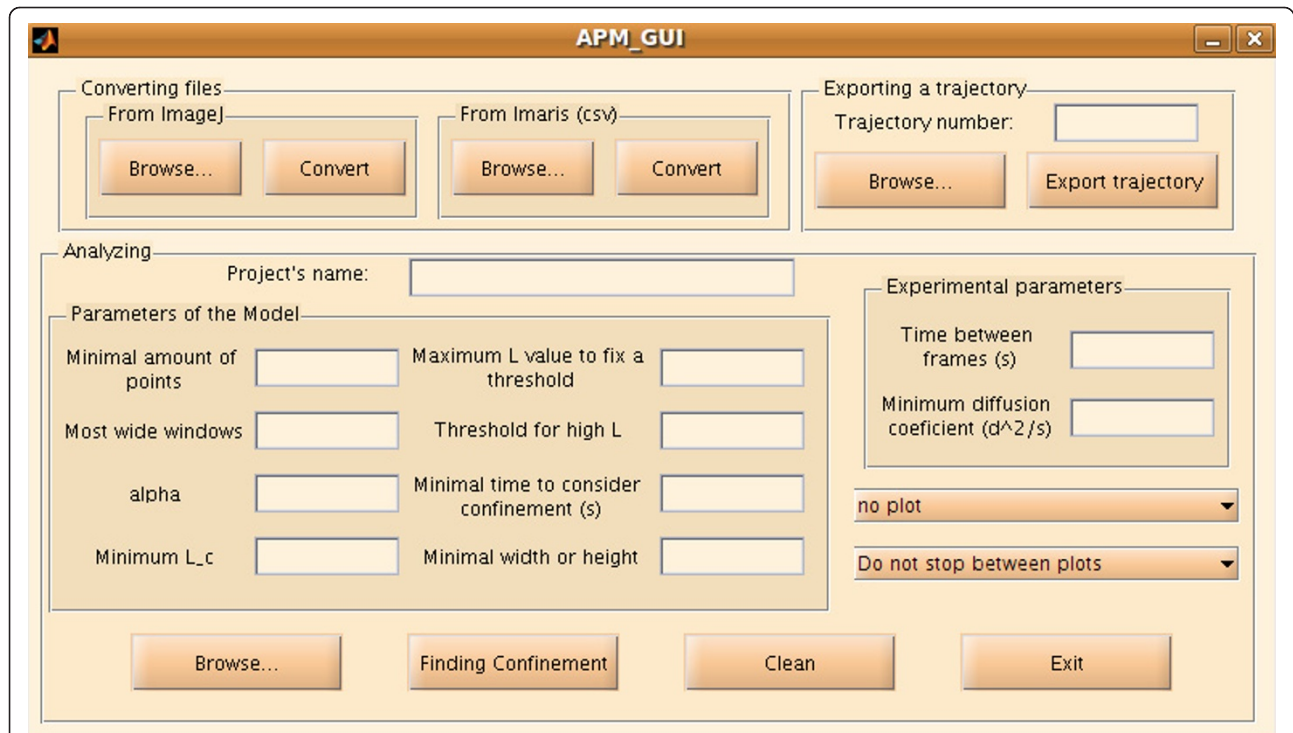


Figure 1 APM_GUI window. The APM_GUI window contains three panels: *Converting files*, *Analyzing* and *Exporting a trajectory*. Information about process status is displayed in the MatLab Command Window.

confinement are associated with the highest values of L . Thus, a period of confined diffusion is defined by the positions where L increases above a critical threshold L_c for a duration of time longer than a critical time t_c . A confinement zone position is defined by the center, which is the average position of all points with $L \geq L_c$, and the distance from the center to the furthest point into the confinement zone. Because the diffusion coefficient of a particle can change along the trajectory, we define L_c similarly to Meilhac *et al.* [8]. In our algorithm, $L_c = \alpha \langle L \rangle$, where $\langle \cdot \rangle$ represents the average over all the points along the trajectory. Because Brownian diffusants do not have non-random confined movement, we define a minimum possible value for L_c . However, to prevent mistakes in defining the size of the confinement zone, we fix the value of the threshold L_c if L has very high values. This value is fixed to ensure well-defined confinement zone sizes in the cases of experimental trajectories that are confined most of the time or in cases in which there is a very large difference between the outer and inner diffusion coefficients. Thus, for each trajectory, the threshold L_c is generally defined as $\alpha \langle L \rangle$. However, if $\alpha \langle L \rangle$ is less than the minimum possible value, L_c is defined as the minimum possible value, and if L has very large values, L_c assumes another predefined value (see the section “The parameters”).

Diffusion coefficients

The diffusion coefficients were quantified by analyzing the mean square displacement (MSD). The MSD (ρ) is the average value for the square of the distance between the initial and final positions of a particle for all time lags. Using all available displacements of a given duration $n\Delta$, we have:

$$\rho_n = \frac{1}{N-n} \sum_{i=1}^{N-n} (\vec{r}_{n+i} - \vec{r}_i)^2, \quad n = 1, \dots, N-1, \quad (3)$$

where N is the total number of points in the trajectory. For small n , the MSD values are well averaged. However, longer time lags have greater uncertainty in the MSD value. Diffusion coefficient values can be obtained from the evaluation of a plot of the MSD vs. time. However, the localization uncertainty and finite camera exposure introduce an *offset* to the MSD curve. Michalet considered all of these effects (see Section III Ref. [13]) and proposed an algorithm to determine the optimal number of points to be considered when fitting the MSD curve, (see Section V Ref. [13]).

The characteristic diffusion coefficient of a given region is calculated as described by Michalet [13] using a sub-trajectory containing only the points within the region under consideration. Instantaneous diffusion

coefficients are calculated by computing an MSD curve for each sub-trajectory of duration 10Δ . Although the uncertainty in the diffusion coefficient can be relatively large for such a short sub-trajectory, instantaneous diffusion coefficients are useful in determining slow and fast diffusion regimes with durations equal to or greater than 10Δ (the used time window).

The parameters

The parameters S_m , t_c and all of the parameters involved in the definition of L_c must be optimized. A random walk can temporarily mimic confinement. Then, the L -profile would not be a horizontal line, even for a Brownian diffusant. However, the peaks for a random walk with mimicked confinement would be lower and narrower than those for non-random confinement. Then, the minimum value for L_c has to ensure that the L -profile for any Brownian diffusant would be below it (or above it, but only for durations shorter than t_c). This parameter has to be introduced in the “Minimum L_c ” text box. However, to prevent mistakes in the determination of confinement zone sizes, if L has a greater value than the parameter introduced in the “Maximum L value to fix a threshold” text box, then the threshold L_c takes the value that has been introduced in the “Threshold for high L ” text box. These two parameters cannot affect most trajectories; they are defined only for specific cases and obviously do not affect any trajectory when the parameter entered in the “Maximum L value to fix a threshold” text box is sufficiently high. The value of S_m changes the L -profile. If S_m is too low, the L -profile presents many peaks. Increasing the maximum segment size S_m smooths the L -profile. However, if S_m is too large, the L -profile is very smooth, and information about confinement is lost. This loss is caused by taking the average over very large segments of the trajectory. Then, the value of S_m must be sufficiently high to suppress the mimicked confinement of Brownian motion. However, it cannot be too high because the L -profile resulting from non-random confinement may also be suppressed. The optimization of the time t_c depends on S_m and L_c . However, depending on the biological sample, it can sometimes be estimated from previous knowledge. For a more detailed description about how to optimize the parameter S_m , see Ref. [7], as we kept the same notation.

Inputs and outputs

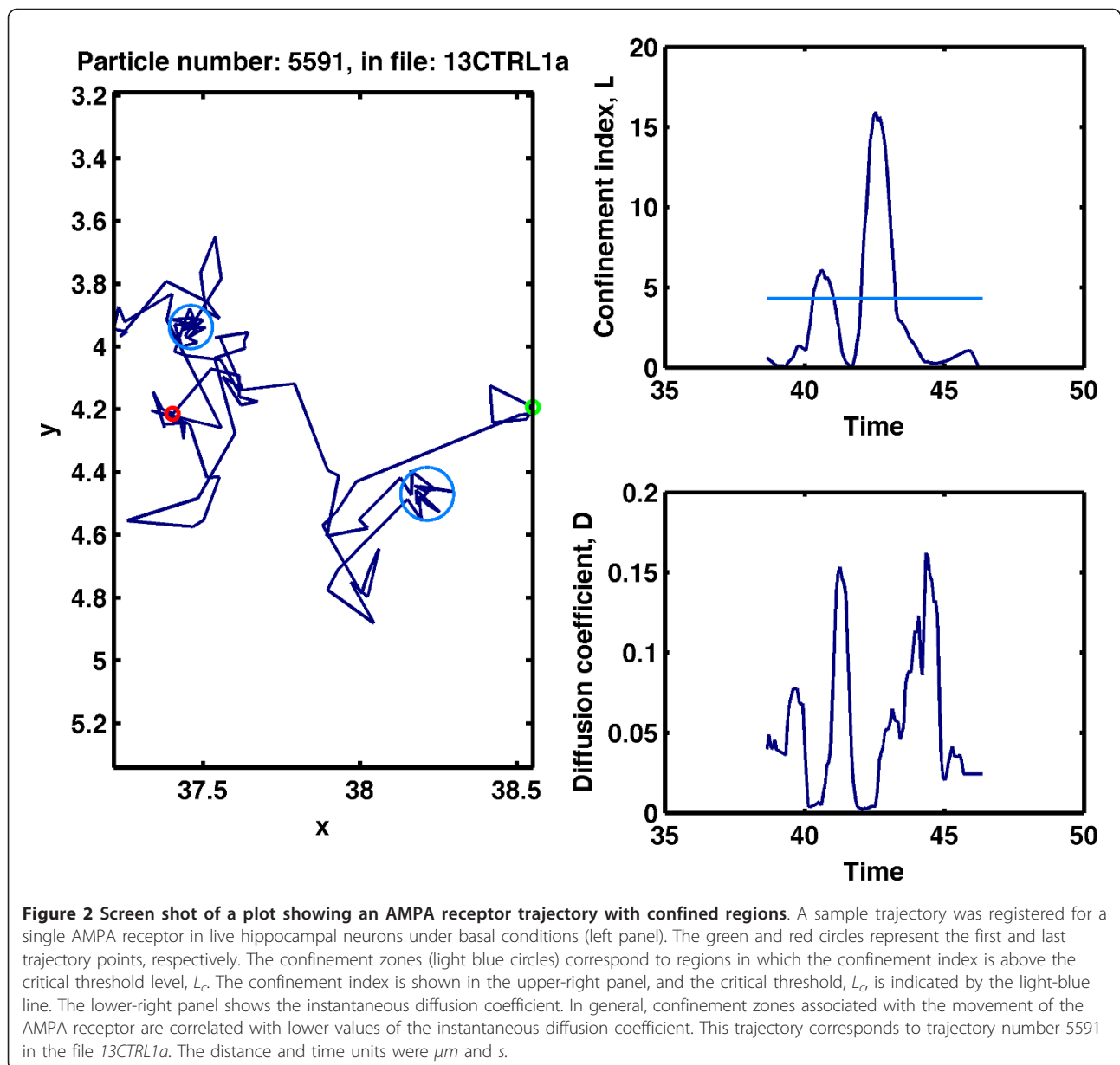
The files to analyze can be selected using the browser for the folder dialog box that appears after the “Browse” button in the panel “Analyzing” is pressed. For the analysis, APM_GUI uses the files that contain trajectory information that has been generated by the functions in the panel “Converting files”, (see section Data import).

When the files to analyze have been selected, and the required parameters have been given, the analysis will start after the “Finding confinement” button in the panel “Analyzing” is pressed.

The information from the analysis is saved in a *.mat* file. Users familiar with MatLab can then work on the project (performing statistical interpretation) after the analysis is complete. Information related to trajectories that do not have confinement zones is saved in the vectors with names containing “*nonConf*”. Information for trajectories with confinement zones is divided into three groups: inside (including data about confinement zones characterized by a diffusion coefficient greater than the value that has been introduced in the “Minimum diffusion coefficient (d^2/s)” text box); immobile (including information about confinement zones characterized by a diffusion coefficient less than the value that has been introduced in the “Minimum diffusion coefficient (d^2/s)” text box); and outside (characterizing the trajectories outside confinement zones). All relevant data are also available in four files with a *.dat* extension: one with general information and three representing the inside, immobile and outside groups, as explained above. These files often have small sizes and are easily transported, which allows the statistical analyses to be performed using the user’s preferred computer and/or software. Other options available in the panel “Analyzing” are referred to as plotting options. All trajectories, trajectories with non-confinement zones or trajectories with confinement zones can be selected for plotting. Confinement zones are represented by light and dark blue circles, which indicate mobile and immobile particles, respectively. The small green and red circles indicate the first and last points of the trajectory, respectively. The file name and the trajectory number are indicated at the top of the plot. An example of a plot is shown in Figure 2. For a detailed description, see the additional file 1.

The panel: Exporting a trajectory

Using the panel “Exporting a trajectory”, it is possible to export the spatial coordinates, the confinement index L and the instantaneous diffusion coefficient over time as a *.dat* file. In this case, the number of the selected trajectory has to be entered, and the file that contains it has to be selected using the browser for the folder dialog box that appears after the “Browse” button is pressed in the “Exporting a trajectory” panel. The parameters required in the empty boxes in the panel “Analyzing” also have to be entered. After the “Export trajectory” button is pressed, a new *.dat* file is created and saved in the same folder where the selected file is located.



Results and Discussion

Testing Simson/Saxton approach with simulated data

To test our implementation and analyze its power of detection, we used simulated random walks. Particle trajectories were generated by using a jump size ℓ and choosing a random direction for each jump. After 100 jumps, the particle position was recorded. Each simulated trajectory consisted of 100 frames (i.e., the particle position was recorded 100 times), and for each setting (i.e., for each ℓ), 1000 particles were simulated. The fixed jump length was determined by $\ell = \sqrt{4Ddt}$, where D was the diffusion coefficient applied and dt was the time interval of a basic simulation step. To simulate the

confined motion, we assumed that the particles were entrapped in circular domains of radius r_d . The exit from the domain was restricted by rigid walls. To test the effects of different noise levels, we displaced every trajectory point of a simulated trajectory by a random distance that was generated based on Gaussian white noise with a standard deviation of σ_n . Because we also tested our implementation with experimental data obtained from AMPA receptors (see next sub-section), we used a similar setting for the simulations. Then, $dt = 0.067/100 s$ and for the analysis, S_m , α and t_c were selected as 25, 0.5 and 0.67s (10 frames), respectively. The minimal value for L_c was 4.3, and if $L \geq 100$, L_c took the value of 19.

First, we analyzed a set of pure random trajectories, considering eight different diffusion coefficients between $0.01 \mu\text{m}^2/\text{s}$ and $0.1 \mu\text{m}^2/\text{s}$. In all cases without noise, the false positive rate was less than 2.5%. However, this false positive rate decreased when noise was added. Simulations with noise were performed using standard deviations between $0.01 \mu\text{m}$ and $0.06 \mu\text{m}$. In particular, all of the cases tested had a false positive rate of less than 1% if $\sigma_n \geq 0.05 \mu\text{m}$.

To determine the power of detection of our implementation, we simulated trajectories with confinement zones. Two different types of mixed trajectories were generated. First, we considered that a particle started at the center of a confinement zone with a rigid wall and that the wall vanished after M frames. In other words, the particle was allowed only to jump inside the confinement zone for M frames, and afterward, it had a free Brownian motion without being confined. Second, we assumed that a particle with free Brownian motion entered a confinement zone through its edge at frame number $100 - M$ and remained in the confinement region until the last frame (100). In this case, the rigid wall appeared at the moment that the particle was entrapped. We called these schemes Confined-Free Motion and Free-Confined Motion, respectively. In Figure 3, the power of detection vs. r_{dl}/ℓ is plotted for different M -values for both types of mixed trajectories, without considering noise. As expected, better detection was provided with a smaller ratio r_{dl}/ℓ and a higher M (because the particles reached the boundary of the confinement zone more often). According to our settings, it is not possible to detect confinement if $M \leq 10$ and the last S_m positions (25 points) are not averaged well. Therefore, the power of detection for free-confined motion is much lower than that for confined-free motion. The simulations were performed for $D = 0.05 \mu\text{m}^2/\text{s}$ (open circles) and $D = 0.1 \mu\text{m}^2/\text{s}$ (asterisk). When noise was added, the power of detection changed, as shown in Figure 4. Although Figure 4 shows the results for $D = 0.05 \mu\text{m}^2/\text{s}$, similar results were obtained for $D = 0.1 \mu\text{m}^2/\text{s}$. Considering that $\ell = 0.012 \mu\text{m}$ for $D = 0.05 \mu\text{m}^2/\text{s}$, Figure 4 shows that the power of detection of our implementation is not strongly affected by noise levels that are similar to ℓ .

Testing the Simson/Saxton approach with experimental data

We also tested our application by studying the diffusion of AMPA receptors on the cell membrane of hippocampal neurons maintained for 10-14 days *in vitro* under basal conditions of activity and after the addition of the excitatory neurotransmitter glutamate. For a detailed description of the material and methods, see additional file 2. We evaluated the data from five different

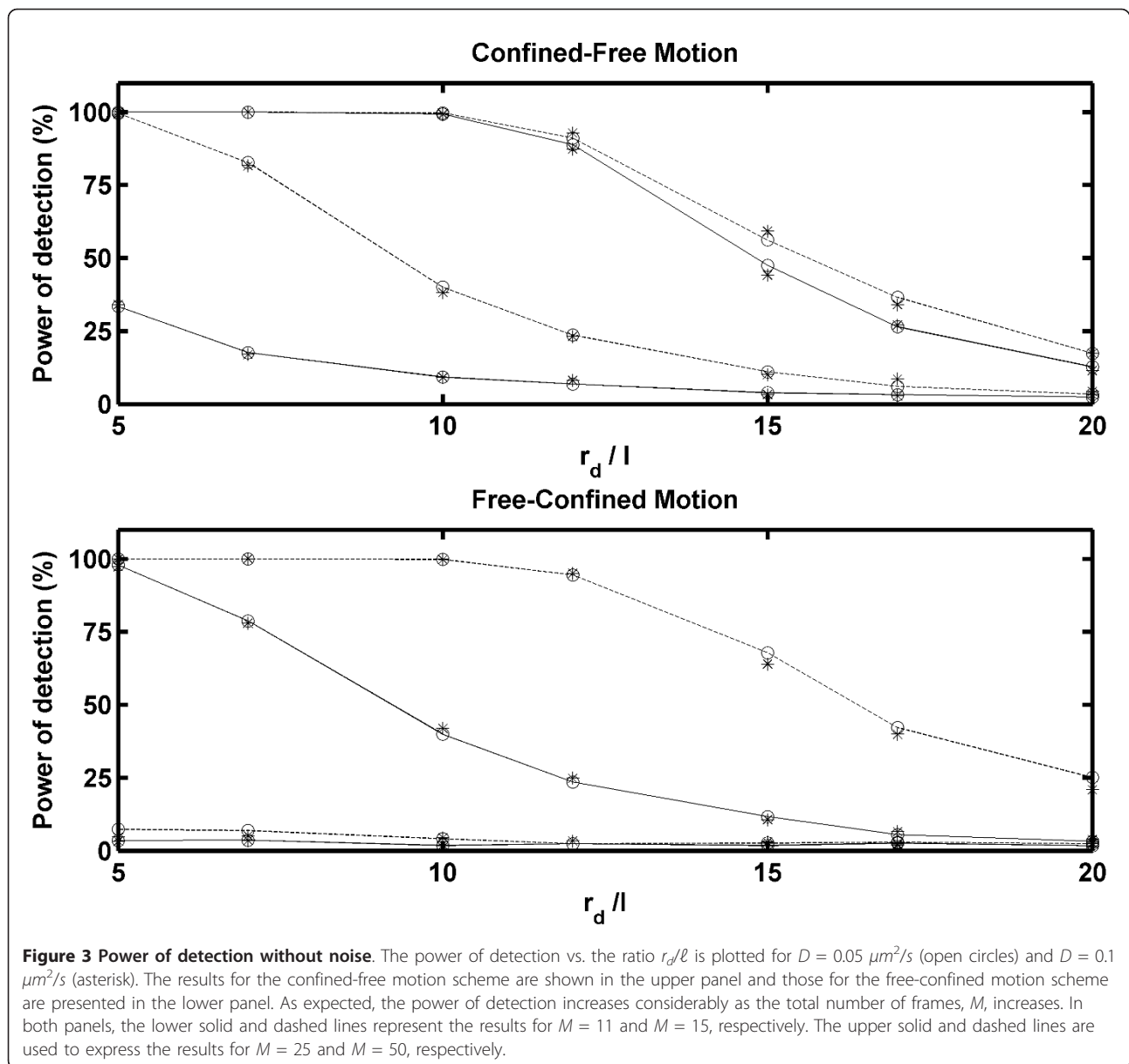
neuronal cultures. The trajectories of single molecules were obtained by tracking with the Imaris program. Only trajectories containing at least 50 points were considered for the analysis; S_m , α and t_c were selected as 25, 0.5 and 0.67s (10 frames), respectively. The minimal value for L_c was 4.3, and if $L \geq 100$, then L_c assumed the value of 19.

In Figure 5, the histograms for the diffusion coefficients inside confined regions, the durations of confinement and the radii of the confined zones for a culture after the addition of the excitatory neurotransmitter glutamate are shown. The particles were considered to be immobile in a given region if the median of their instantaneous diffusion coefficients in that region was below $0.0075 \mu\text{m}^2/\text{s}$. The median was used because the diffusion coefficients for each culture had a long-tail distribution. Trajectories with confined zones were split into confined and non-confined regions by APM_GUI. For each type of region, a diffusion coefficient was assigned for computing the MSD curve. Non-confined regions were referred to as regions with free Brownian motion. Under basal conditions, the diffusion coefficient values for the mobile particles were significantly reduced in the confined zones, $0.0176 \mu\text{m}^2/\text{s}$, compared with free Brownian motion, $0.0732 \mu\text{m}^2/\text{s}$ ($p < 0.001$, t-test, five cultures). After the application of glutamate, the confined and free diffusion coefficients for the AMPA receptors significantly increased: $0.0208 \mu\text{m}^2/\text{s}$ ($p < 0.05$, t-test, 5 cultures) and $0.1097 \mu\text{m}^2/\text{s}$ ($p < 0.05$, t-test, five cultures), respectively, compared with the basal conditions. This result suggests that the activation of hippocampal neurons with glutamate increases AMPA receptor diffusion. Significant variations were not observed in the size of the confined zones, the duration of permanence within the confined zones or the proportion of immobile particles in the confined zones. These results are consistent with previous studies [14-16].

Conclusions

In conclusion, our application provides a robust tool to study confined movement and diffusion dynamics on the cell membrane. Although some parameters depend on the experimental equipment, many others have to be optimized according to the instructions given in the section "The parameters". This application is programmed in MatLab and has a GUI that allows it to be used even without any programming knowledge. However, expert users can modify the scripts according to their needs.

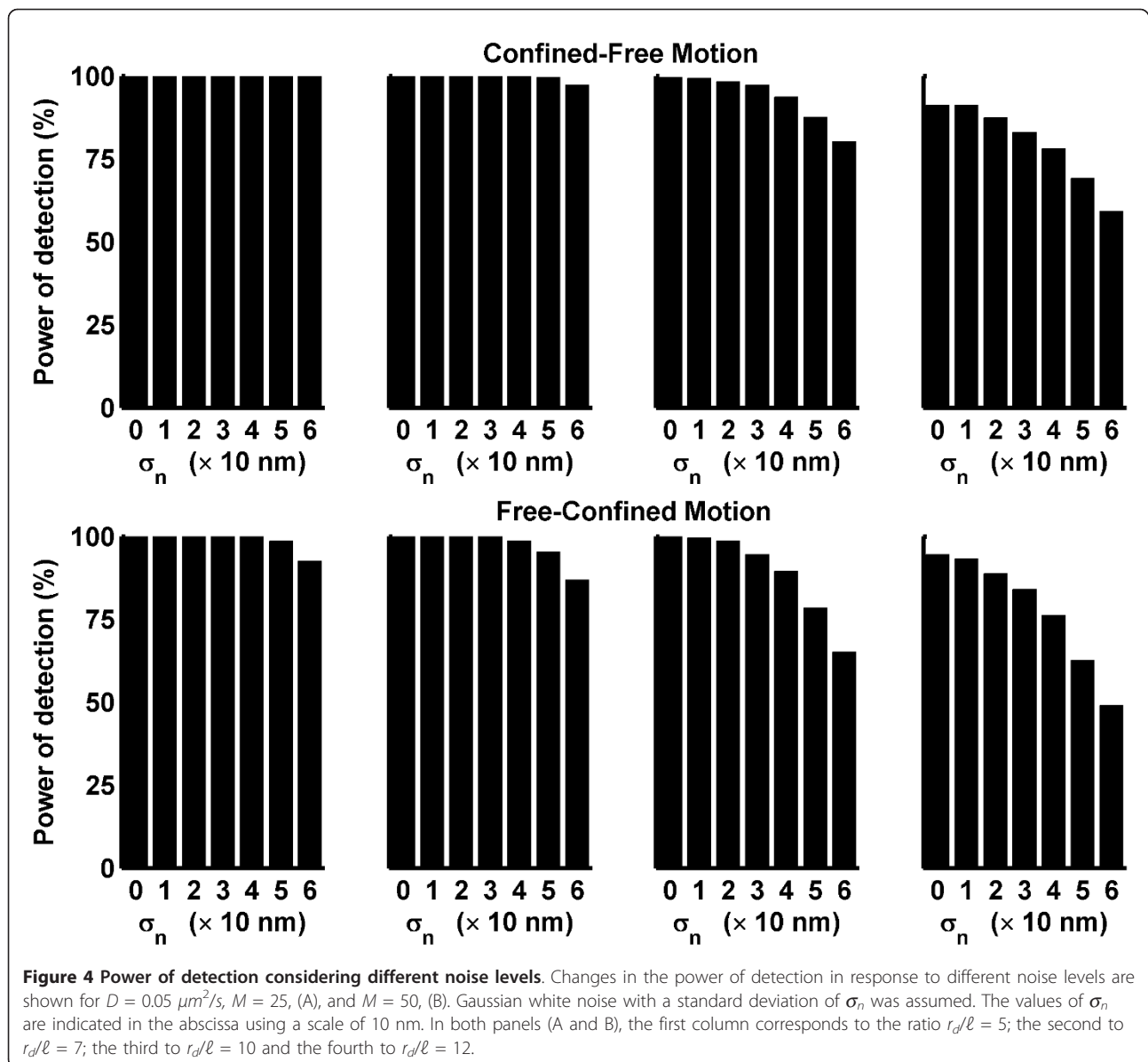
In general, the trajectories obtained by SPT are analyzed by fitting the MSD vs. time plot or by using a maximum likelihood estimator [17]. Both approaches give reliable diffusion coefficient values when they are used properly [13,17]. However, to identify standard types of motion, such as pure diffusion, anomalous



diffusion, directed motion or confined movement, a specific algorithm should be provided. The test presented here is an algorithm based on the Simson/Saxton approach. This test has as its null hypothesis, *the particle has a pure random walk*, and as its alternative hypothesis, *the trajectory has confinement zones*. Thus, our implementation tests whether a trajectory corresponds to a pure random walk or if it has confinement zones. This type of algorithm is generally used to analyze biological data, when trapping is caused not only by the particles encountering a more viscous region but also by some type of barrier, such as a membrane skeleton fence [18]. Therefore, confined motion may result, for instance, from cytoskeletal corrals or from

restrictions to motion that can be imposed by lipid domains [1]. Our implementation determines the confinement zones and characteristic diffusion coefficients very well. Although the power of detection decreases when the confinement region is at the end of the trajectory, a high power of detection can be obtained with relatively short trajectories, even in the presence of noise.

However, if a particle has a pure random walk, and if we wish to test if its diffusion coefficient changes along the trajectory, the Simson/Saxton approach would fail. In this case, an approach such as that presented by Montiel and co-authors should be used [17]. The Montiel test *always* assumes pure random walks, and the



question behind it is: *are there changes in the diffusion coefficient values?* A statistical analysis of Montiel's test is presented in Ref. [17]. Although it is possible to obtain a high power of detection for diffusion coefficients differing by one order of magnitude, for smaller differences, considerably long trajectories are needed, which can be a limitation for biological samples on the cell membrane (because of endocytosis).

APM_GUI is based on the Simson/Saxton approach, and it is an open-source application that is publicly available directly from its home page and it is also included in the additional file 1. It also allows an easy analysis of all results because of the generation of files with a universal format and relative small size, which

can be imported by any software for statistical analyses and plotting.

Availability and requirements

Project name: AMP_GUI

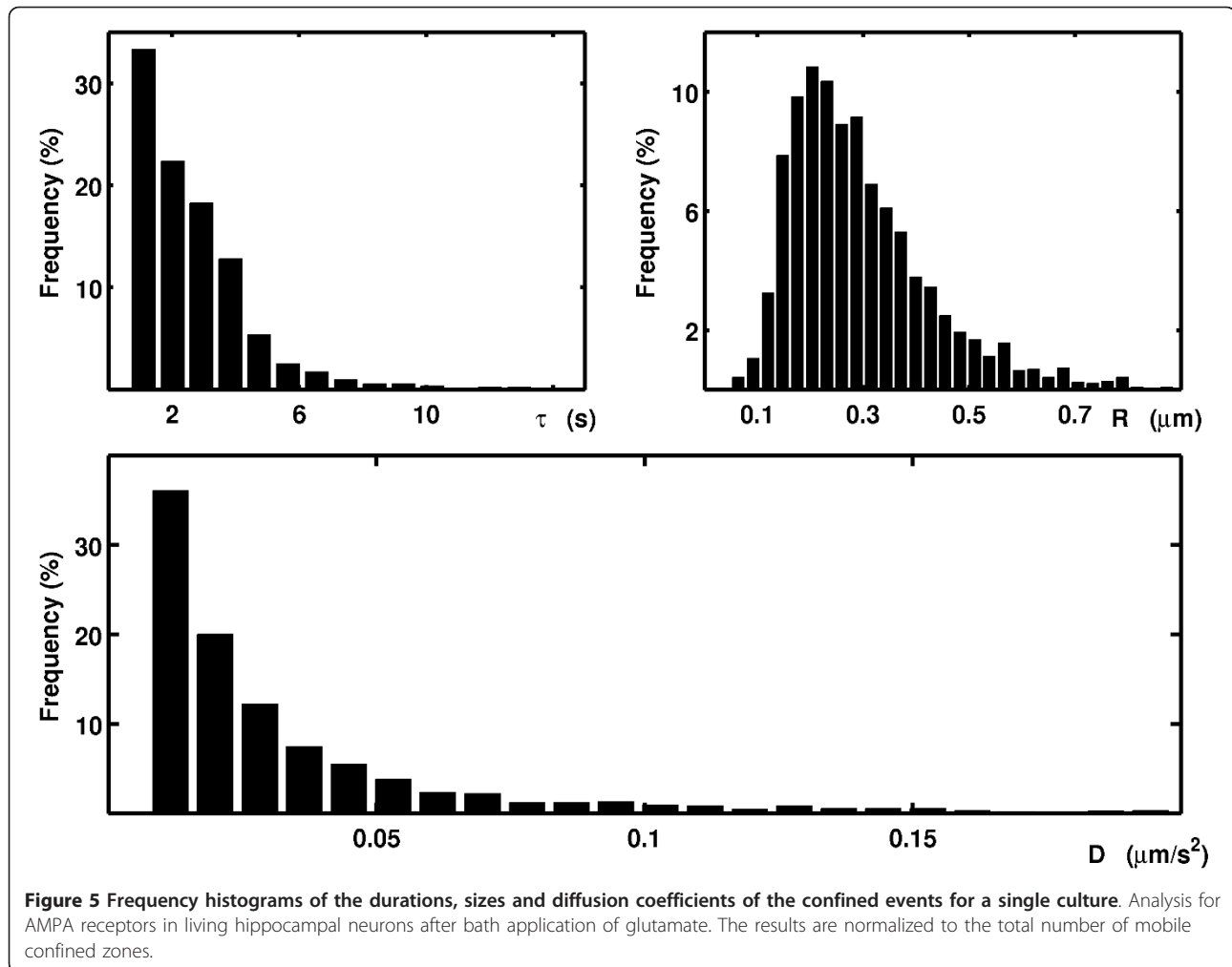
Project home page: <http://www.bits.vib.be/> under *BioInfo at VIB - Developed at VIB*.

Operating system(s): platform-independent.

Other requirements: MatLab.

License: APM_GUI is distributed under the terms of the GNU General Public License, as published by the Free Software Foundation, version 3.

Further information: a short manual with a user's guide and installation instructions is available with



the code source, which is also included in the additional file 1.

Additional material

Additional file 1: Source code of APM_GUI. The file should be extracted using a suitable program (e.g. Winzip, 7-Zip or File-roller). The extracted folder Scripts_APM_GUI should be placed in MatLab's path. Then, the application can be started by typing APM_GUI in MatLab's Command Window. The folder Examples has a few trial files for the software, and the file instructions_and_installation provides a short manual.

Additional file 2: Materials and Methods. This file has a description of the materials and methods for the biological samples that were used to test our software. Hippocampal neurons from 18- day- old rat embryos were cultured until 10-14 DIV on glass coverslips, according to the Banker technique [19] density of (~ 6000 neurons/cm²). The protocol described in Ref. [20] was used to label GluR2-containing AMPARs.

List of abbreviations

The abbreviations used throughout the article are SPT: Single Particle Tracking; Qdots: Quantum dots; GUI: Graphical User Interface; MSD: Mean Square Displacement.

Acknowledgements

Spanish Ministry of Science and Innovation: SAF 2010-14906, Consolider 2010-00045, Flamish Fund for Scientific Research (FWO), Federal Office for Scientific Affairs, IAP P6/43 and the Flemish Government for the Methusalem Funding to CGD. This project was partially supported by European Union FP7 IIF Marie Curie Actions (FP7-PIIF-GA-2009-252375).

Author details

¹Department of Molecular and Developmental Genetics, VIB Center for the Biology of Disease and Center for Human Genetics, KULeuven, Campus Gasthuisberg, Herestraat 49 - bus 602, 3000 Leuven, Belgium. ²IFEG-CONICET and FaMAF, Universidad Nacional de Córdoba, Córdoba, Argentina. ³Centro de Biología Molecular Severo Ochoa, CSIC-UAM, Campus Universidad Autónoma de Madrid, Madrid, Spain.

Authors' contributions

SAM designed and developed the software. MGM contributed to the evaluation of the software regarding the biological data management and aided in its design. CGD initiated and supervised the project. Each of the authors contributed to the drafting of the manuscript and have read and approved its final version.

Competing interests

The authors declare that they have no competing interests.

Received: 30 August 2011 Accepted: 20 February 2012

Published: 20 February 2012

References

1. Saxton MJ, Jacobson K: **Single-Particle Tracking: Applications to Membrane Dynamics.** *Annu Rev Biophys Biomol Struct* 1997, **26**:373-399.
2. Michalet X, Pinaud FF, Bentolila LA, Tsay JM, Doose S, Li JJ, Sundaresan G, Wu AM, Gambhir SS, Weiss S: **Quantum Dots for Live Cells, in Vivo Imaging, and Diagnostics.** *Science* 2005, **307**:538-544.
3. Saxton MJ: **Single-particle tracking: connecting the dots.** *Nat Methods* 2008, **5**:671-672.
4. Sergé A, Bertaux N, Rigneault H, Marguet D: **Dynamic multiple-target tracing to probe spatiotemporal cartography of cell membranes.** *Nat Methods* 2008, **5**:687-694.
5. Sbalzarini IF, Koumoutsakos P: **Feature point tracking and trajectory analysis for video imaging in cell biology.** *J Struct Biol* 2005, **151**:182-195.
6. Bitplane web page:[<http://www.bitplane.com/>].
7. Simson R, Sheets ED, Jacobson K: **Detection of Temporary Lateral Confinement of Membrane Proteins Using Single-Particle Tracking Analysis.** *Biophys J* 1995, **69**:989-993.
8. Meilhac N, Le Guyader L, Salome L, Destainville N: **Detection of confinement and jumps in single-molecule membrane trajectories.** *Phys Rev E* 2006, **73**:011915.
9. Saxton MJ: **Lateral diffusion in an archipelago. Single-particle diffusion.** *Biophys J* 1993, **64**:1766-1780.
10. Ruthardt N, Lamb DC, Brüche C: **Single-particle Tracking as a Quantitative Microscopy-based Approach to Unravel Cell Entry Mechanisms of Viruses and Pharmaceutical Nanoparticles.** *Mol Ther* 2011, **19**:1199-1211.
11. Elliott LCC, Barhoum M, Harris JM, Bohn PW: **Trajectory analysis of single molecules exhibiting non-Brownian motion.** *Phys Chem Chem Phys* 2011, **13**:4326-4334.
12. Thompson MA, Casolari JM, Badieirostami M, Brown PO, Moerner WE: **Three-dimensional tracking of single mRNA particles in *Saccharomyces cerevisiae* using a double-helix point spread function.** *PNAS* 2010, **107**:17864-17871.
13. Michalet X: **Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium.** *Phys Rev E* 2010, **82**:041914.
14. Renner ML, Cognet L, Lounis B, Triller A, Choquet D: **The excitatory postsynaptic density is a size exclusion diffusion environment.** *Neuropharmacology* 2009, **56**:30-36.
15. Frischknecht R, Heine M, Perrais D, Seidenbecher CI, Choquet D, Gundelfinger ED: **Brain extracellular matrix affects AMPA receptor lateral mobility and short-term synaptic plasticity.** *Nat Neurosci* 2009, **7**:897-904.
16. Petrini EM, Lu J, Cognet L, Lounis B, Ehlers MD, Choquet D: **Endocytosis and recycling maintain a mobile pool of surface AMPA receptors at synapses required for synaptic potentiation.** *Neuron* 2009, **63**:92-105.
17. Montiel D, Cang H, Yang H: **Quantitative characterization of changes in dynamical behavior for Single-Particle Tracking studies.** *J Phys Chem B* 2006, **110**:19763-19770.
18. Kusumi A, Sako Y: **Cell surface organization by the membrane skeleton.** *Curr Opin Cell Biol* 1996, **8**:566-574.
19. Goslin KBG: *Culturing Nerve Cells* MIT Press, Cambridge; 1991, 453.
20. Dahan M, Lévi S, Luccardini C, Rostaing P, Riveau B, Triller A: **Diffusion dynamics of glycine receptors revealed by single-quantum dot tracking.** *Science* 2003, **302**:442-445.

doi:10.1186/2046-1682-5-4

Cite this article as: Menchón *et al.*: APM_GUI: analyzing particle movement on the cell membrane and determining confinement. *BMC Biophysics* 2012 **5**:4.

Submit your next manuscript to BioMed Central
and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

