Biophysical Reviews and Letters
Vol. 8, Nos. 3 & 4 (2013) 109–136
© World Scientific Publishing Company
DOI: 10.1142/S1793048013300016

World Scientific
www.worldscientific.com

MATHEMATICAL TREATMENTS THAT SOLVE SINGLE MOLECULES

OPHIR FLOMENBOM

Flomenbom-BPS Ltd, 19 Louis Marshal Street
Tel Aviv, 62668, Israel
flomenbo@flomenbom.net

Received 31 July 2013
Accepted 13 August 2013
Published 14 January 2014

In this article, we talk about the ways that scientists can solve single molecule trajectories. Solving single molecules, that is, finding the model from the data, is complicated at least as much as measuring single molecules. We must filter the noise and take care of every step in the analysis when constructing the most accurate model from the data. Here, we present valuable solutions. Ways that solve clean discrete data are first presented. We review here our reduced dimensions forms (RDFs): unique models that are canonical forms of discrete data, and the statistical and numerical toolbox that builds a RDF from *finite*, clean, two-state data. We then review our most recent filter that "tackles" the noise when measuring two state noisy photon trajectories. The filter is a numerical algorithm with various special statistical treatments that is based on a general likelihood function that we have developed recently. We show the strengths of the filter (also over other approaches) and talk about its various new variants. This filter (with minor adjustments) can solve the noise in any discrete state trajectories, yet, extensions are needed in "tackling" the noise from other data, e.g. continuous data. Only the combined procedures enable creating the most accurate model from noisy discrete trajectories from single molecules. These concepts and methods (with adjustments) are valuable also when solving continuous trajectories and fluorescence resonance energy transfer trajectories. We also present a set of simple methods that can help any scientist with treating the trajectory perhaps encouraging applying the involved methods. The involved methods will appear in software that we are developing now, helping therefore the experimentalists utilizing these methods on real data. Comparisons with other known methods in this field are made.

Keywords: Solving single molecules; FRET; AFM; enzymes; ion channels; quantum dots.

Special Issue Comment: This article about mathematical treatments when solving single molecules is related to the reviews in this Special Issue about measuring enzymes[67] and about FRET experiments[2] and about the software QUB.[6]

1. Introduction

1.1. *About the Special Issue*

In this paper, I present my results from about a decade of work in this field of solving single molecules' trajectories. These results are connected with practically all articles in this Special Issue about measuring[1–32] and solving[33–67] single molecules[a]: proteins, RNA, DNA, enzymes, ion channels, receptors, ATPses, other biological complexes and entities, quantum dots and other nanobjects are measured at the level of individual entities. Solving the data, we try constructing the mechanism of the measured entity. The required mathematical, statistical and numerical treatments that we have developed while collaborating with other scientists, and with connections with results from many scientists in this field, are presented in this paper. The field of solving single molecules is among the main ones I work at, with important results about mathematical and statistical formulations,[33,36,38,39,45,64] yet also while collaborating with experimentalists and solving data from experiments involving individual enzymes.[15,16]

Let me explain about the logic that is in the basis of the treatments that solve the data from single molecules (here, "solving the data" and "solving single molecules" refer to constructing the mechanism from the trajectory). First, we must think about the best model-type that can explain the process we measure. Then, we must think about the way we can extract the model from the noisy data. We must know about kinetic schemes (KSs) and other popular dynamical operators appearing in biophysics, about reactions and energy surfaces. We must have knowledge in statistics, in analyzing data, and in tackling noise in the data. From these we should create the algorithms that solve the data. All these issues are presented in this review. We present various difficulties with solving the data appearing with binning and thresholding, yet also a set of simple methods that can help with the basic analysis. We present our recently developed filter (with various new variants) that can "tackle" the noise from photon two state data, yet can also treat (with simple extensions) any discrete data from individual molecules. We also talk about the best way we should construct the model from the clean discrete data: we present our reduced dimensions forms (RDFs) that are canonical (unique) forms of discrete data and review the statistical and numerical toolbox that solves clean data.

This review should help experimentalists that try solving in the most accurate way the trajectories from single molecules and should supply the mathematical biophysicists a reference.

1.2. *The processes*

Since the seventies and particularly in the passing 25 years, scientists have continuously been developing smart methods that measure many processes in biology, chemistry and physics at a level of individual molecules. These smart methods enable us (in principle) to extract information about microscopic processes that is not accessible from bulk experiments. Yet, finding the model from the signal is complicated. When the noise is not solved accurately in data from individual molecules, misleading models are reported. Constructing a mechanism from the

[a] Please see the introductory article of this Special Issue where we elaborate on all articles that appear in this Special Issue.

clean data is also complicated. Here, we talk about solving ("tackling") the noise in discrete (with $m = 1, 2, 3, \ldots$ states) data and about constructing the mechanism from the derived clean data. The methods that are presented here were developed during my work in the past decade. These concepts and methods (with adjustments) are valuable also when solving continuous trajectories and fluorescence resonance energy transfer (FRET) trajectories.

When talking about relevant experiments and processes, we list: (i) the passage of ions and biopolymers through individual channels,[3–6] (ii) activity and conformational changes of biopolymers (including measurements involving spectroscopy, FRET, atomic force microscopy (AFM) and other techniques),[1,2,7–21,59] (iii) diffusion of molecules,[22–25] (iv) blinking of nanocrystals,[26–29] etc.

Signals are time trajectories. Discrete ones are made of several observed values, or states, where a popular example is of *on–off* trajectories (this is also the simplest example): trajectories that are made of *on* and *off* periods [also termed residence times or jumping times (JT); see Figs. 1(a) and 1(b)]. Extensions include FRET trajectories and even continuous trajectories. From a noisy trajectory, one aims at finding the mechanism that can generate the observed process, has a physical sense and can supply scientific insights about the observed process. In many cases, we say that the model of the observed process is a multi-substate, multi-state Markovian KS. *On–off* KSs are popular,[30–45] [see Fig. 1(e)]. The model can represent, for example, one of the following physical realizations: a discrete conformational energy landscape of a biological molecule, steps in a chemical reaction with conformational changes or environmental changes, quantum states, ion channel activity, etc.

1.3. *General comments about solving the signal*

The aim is therefore solving the trajectory: building the mechanism from the noisy data. Here, it is also important to emphasize that in many projects, we can solve the process only when we solve data from individual entities, since other (experimental or numerical) techniques are missing or are not equally informative. The case of enzymes is such an example: extracting precise information about the enzymatic mechanism is possible only when solving single enzymes because bulk measurements supply very partial information and numerical simulations (usually) only provide knowledge about the short timescale.

What are the problems and difficulties that arise when we try solving the data? First, when the noise in the data is not filtered accurately while analyzing the data (due to ignoring the noise, assuming that the noise is small, using too simple filters, etc.), misleading models and partial and questionable conclusions are reported. We have observed that in many projects, the noise is not filtered accurately. Results from binning the data and thresholding may depend on the bin size and the threshold value and may fail filtering the noise and thus lead to misleading models (see Appendix A where various examples show that binning fails yet the method presented next filters the noise). Another important issue while filtering the noise is

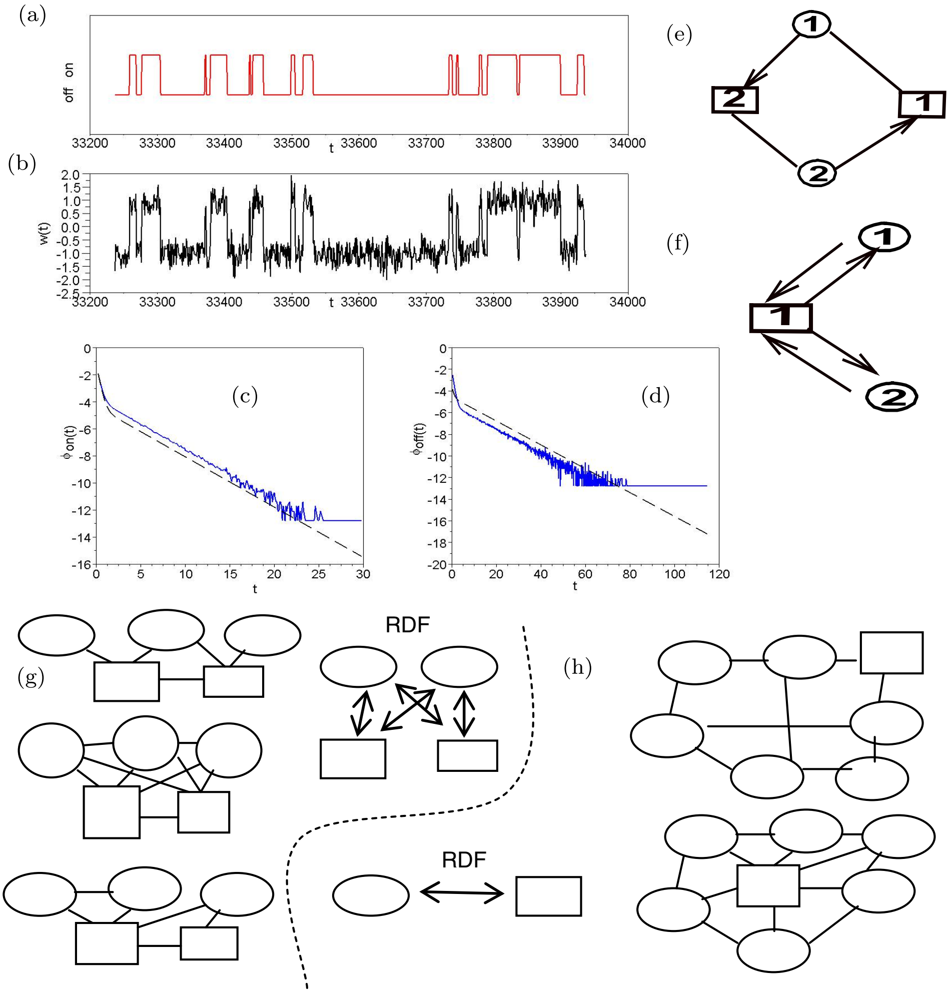


Fig. 1. A segment of a clean two-state trajectory (a), the noisy data (b), $\phi_{on}(t)$ (c), $\phi_{off}(t)$ (d), a KS (e) and its corresponding RDF (f). Trajectory (a) is generated when simulating a random walk in KS 1(e), and adding a zero mean Gaussian noise for every element of the clean data, with a variance, $\sigma_z = 0.4$ counts/bin. $\phi_{on}(t)$ and $\phi_{off}(t)$ are shown, on a log-linear scale, in panels (c) and (d), respectively. The KS 1(e) has $L_{on} = 2$ (squared substates), $L_{off} = 2$ (circled substates), and irreversible transitions. We assign numerical values for the connections among substates in the KS (the connections are termed reaction rates or transition rates or kinetic rates) when generating the data and constructing the specific RDF. The RDF is obtained from the KS with a clear-defined mathematical mapping developed in Ref. 38. (g) Three *on-off* KSs with 2 *on* substates and 3 *off* substates. The RDFs' forms of these KSs have the same structure shown on right, yet different complexity in the connection functions. Therefore: these KSs are distinguishable with respect of the *on-off* trajectory. (h) The KSs on the right in this panel have one *on* substate and 6 *off* substates. They have the same RDF shown on the left of this panel and are therefore equivalent.

utilizing the properties of the derived clean data. This is not performed in not so few cases. Yet, one of the concepts that we talk about here is indeed filters that rely also on the likelihood of the obtained cleaned data.

We can suggest several basic methods that one can apply when staring the analysis of the data: in case binning is used, always scan various bin sizes and thresholds. Check where binning is not stable (short and long bins) and where it is stable. Check various thresholds and their instabilities. Compare correlation functions with bulk results and control measurements. These are all the most basic tests that should supply a basic feeling about the data. Only then we can analyze the data with advanced methods: treating the (e.g.) photons as a function of the order recorded, we compute correlation functions from this trajectory and the probability density function (PDF) of the photons. In case this trajectory is not correlated all the information is just the simple distribution of photons (otherwise we can compute higher order PDFs). In any case, we should filter the noise in this trajectory with smart methods and construct the model from the clean data. We then iterate the filtering with the constructed model until convergence. Performing these steps is in fact very complicated. Methods that can help progressing are presented in this review. General relations connecting properties of the data and the mechanism that generated the data are also presented.

Let us also emphasize that there are not so few projects with weak clean data analysis (several examples about this are presented in Ref. 33): models that are not unique are reported without care. We solve the clean data with reduced dimensions forms (RDFs)[38,39,45]: these are canonical mechanisms of discrete state trajectories. We show that an RDF is the most stable mechanisms when solving clean discrete data.

1.4. *Solving the clean data with canonical models*

The problem when solving clean two-state data that has been noticed since the 80s[33,35,40,41,42] and recently solved[38,39,45] (see also Ref. 33) is: a multi substate *on–off* KS is not uniquely obtained from (even a clean and very long) two-state trajectory, and thus we must first construct a canonical form from the data for an accurate analysis (otherwise the result is just one option from many other possibilities that are equivalent statistically). Only one canonical form is built from the data. We have developed very efficient canonical *on–off* mechanisms termed reduced dimensions forms (RDFs)[38,39,45]: these are mathematical mechanisms that are (relatively) simply built from clean data yet also from a kinetic scheme. RDFs are generalized Markov process with (usually) a multi exponential connectivity of substates, where connections are just of different states. The RDFs are advantageous in accuracy and stability over other approaches when solving the clean data[38,39,45]: an RDF having the simplest topology that can reproduce the data. The mapping: $KSs \Rightarrow RDF$ was derived mathematically. RDFs are physical models. Staring from RDF, we can produce a set of possible KSs' structures, yet also finding numerically all the rate values. In data from a complex kinetic scheme, an RDF is the only reasonable way zooming in on the solution (otherwise scanning over all kinetic schemes will take many months). That is also the case when the data was generated from a rather simple KS. RDFs are presented in Fig. 1 [see Figs. 1(f)–1(h)].

1.5. *Filters that solve the noise*

There are many methods that scientists apply when solving data from individual molecules.[29–58] But existing methods treat mainly trajectories without noise or simple noise forms, and many relay on binning.[61–63] "Too simple" filters lead to misleading results. Thus, when solving the noisy trajectory we must filter the noise with the smartest methods.

The questions are: How can we filter correctly the noise in the data? How can we extract all the information from the noisy data? How can we use the information in the noisy data while constructing the most accurate model from the data? Once we filter the noise we can combine the analysis with RDFs. Thus, in answering these questions, we write a filter (a numerical algorithm)[64] based on new numerical and statistical treatments (see codes at, http://www.flomenbom.net/codes_project12.html) and a general likelihood function: we write the likelihood function in a way that involves the raw data, and the derived clean data, see Eq. (3^*). In advanced filters, we express the clean data with our recently developed canonical forms, RDFs.[38,39,45] [In fact, we express the term involving the correlations among events in the clean data with the appropriate reduced dimensions form, see Eq. (3^*).] We show that these concepts work when solving two state noisy photon trajectories from various kinetic schemes. We find that the best specific likelihood form involves a combination of (a) the photons (the observable), (b) the derived clean data presented with *on* and *off* durations and (c) *on–off* correlation terms. In most cases, the best results are seen when all these terms in the likelihood function are included in an unbiased way after a special normalization causing all terms having an equal contribution (the form of the likelihood function determines the filter strength).

We show that partial variants lead to wrong results. In particular, let me say that the filters show that utilizing the clean data in the analysis of the noisy data is crucial, otherwise the results are not accurate. New variants of the filter are presented here in the first place.

All issues related with the noise and the filters are presented in parts 2.2 and 2.3.

1.6. *The organization of this review*

In the next part of this review, we first talk about the analysis of the clean data (also presenting RDFs), and then list noise-sources in measurements. In the main part of that chapter, filters including new variants are presented with results about several dozen systems. The third part concludes.

2. Solving Noisy Trajectories

2.1. *Analysis of clean trajectories*

We start with a short description of our toolbox for solving clean trajectories.[38,39,45]

2.1.1. *The information content in the data*

Here, we consider a trajectory that is infinitely long and without noise. Therefore, we can construct directly from the data any jumping time (JT) probability density function (PDF). These include: $\phi_x(t)$ and $\phi_{x,y}(t_1, t_2)$, where, $x, y = on, \mathit{off}$. $\phi_x(t)$ gives the probability density that an event is state x lasts time t. $\phi_{x,y}(t_1, t_2)$ gives the probability density that an event in state x lasts time t_1 and the following event in state y lasts time t_2. These JTPDFs are expressed with exponential expansions:

$$\phi_x(t) = \sum_i c_{x,i} e^{-\lambda_{x,i} t} \tag{1}$$

with L_x components, (in KS, L_x is usually the number of substates in state x in the KS) and,

$$\phi_{x,y}(t_1, t_2) = \sum_{m,n} \sigma_{x,y,mn} e^{-\lambda_{x,n} t_1 - \lambda_{y,m} t_2}. \tag{2}$$

$\phi_{x,y}(t_1, t_2)$ having $L_x L_y$ components. $\phi_{on}(t)$ and $\phi_{\mathit{off}}(t)$ are constructed from trajectory 1(a) [Fig. 1(a)] and are shown in Figs. 1(c) and 1(d), respectively.

2.1.2. *Constructing a mechanism from the clean data*

When $\phi_x(t)$ and $\phi_{x,y}(t_1, t_2)$ are known, we focus on constructing a KS from these JTPDFs. For this, we construct the likelihood function, $l(\mathbf{\Theta})$,

$$l(\mathbf{\Theta}) = \sum_{x,y} \Sigma_i \log(\phi_{x,y}(t_{1,i} t_{2,i})), \tag{3}$$

and maximize $l(\mathbf{\Theta})$ with respect to $\mathbf{\Theta}$, where $\mathbf{\Theta}$ is the set of rates in the KS. In Eq. (3), the index i represents the ith cycle in the cleaned *on–off* data. We perform the maximization with constraints: the coefficients in $\mathbf{\Theta}$ should also reproduce the coefficients of $\phi_x(t)$ and $\phi_{x,y}(t_1, t_2)$. Yet, finding the KS from $\phi_x(t)$ and $\phi_{x,y}(t_1, t_2)$ is difficult. The reasons are: (i) the number of the substates in each of the states, L_x ($x = on, \mathit{off}$), is usually large, and (ii) the connectivity among the substates is usually complex. Yet, the data has limited information content, and so not *all* the details regarding the KS are obtainable from the data. In addition, there are many local solutions in the landscape of the coefficients,[38] and many of these solutions are very different than the correct KS. We can average over many initial conditions and this exhaustive search is not efficient since convergence is not guaranteed in the space of coefficients. (iii) A fundamental difficulty in finding the correct KS arises from the equivalence of KSs; namely, there are a number of KSs with the same trajectory in a statistical sense.[33–35,38,39,45]

We solve these problems with canonical (unique) forms.[35,38,39,45,64] The space of KSs is mapped. The new space in made of canonical forms. A given KS is equivalent with a unique canonical form, yet several KSs can have the same canonical form. KSs with the same canonical form are equivalent, and cannot be discriminated based on the information in an ideal two-state trajectory. We have derived new canonical forms: RDFs[38,39,45]; see Fig. 1. RDFs are generalized Markovian models since the connections in RDFs have usually multi-exponential JTPDFs. The advantages of RDFs in solving the problem of relating a model with the time *on–off* trajectory over other approaches are numerous[38,39,45]: RDFs are physical models, and possess the simplest possible topology. They are accurately constructed from the data (and from KSs) and are accurately connected with a set of KSs, etc.

We developed a rather simple algorithm that can construct the RDF from $\phi_x(t)$ and $\phi_{x,y}(t_1, t_2)$. First, we note that the rates in the exponential expansion of the JTPDF $\phi_x(t)$ are equivalent with those in the exponential expansion of the JTPDFs, $\varphi_{x,mn}(t)$; $\varphi_{x,mn}(t)$ is the JTPDF connecting substates $n_x \to m_y$ in the RDF, and follows:

$$\varphi_{x,mn}(t) = \sum_{\nu=1}^{L_x} C_{RDF,x,m\nu n} e^{-\lambda_{x,\nu} t}. \tag{4}$$

Then, we note that the rank $R_{x,y}$ $(x \neq y)$ of the matrix $\sigma_{x,y}$ that appears in the double exponential expansion of $\phi_{x,y}(t_1, t_2)$ gives, in most systems, the number of substates in state y in the RDF. Finally, the coefficients in $\{\mathrm{C}_{RDF}\}$ are found when maximizing the likelihood function, Eq. (3), when built from the RDF.

2.1.3. *Constructing the RDF from finite length clean data*

Finding the most accurate RDF from a finite trajectory, even without noise, is a real challenge. The reason is that $\phi_x(t)$ and $\phi_{x,y}(t_1, t_2)$ are not known, and we need smart numerical procedures for extracting these PDFs from finite data. We have developed a set of procedures, forming a toolbox, for constructing accurately the RDF from finite data.[38,39] The toolbox executes the following steps: (i) The rates and the coefficients in the exponential expansion of the JTPDF $\phi_x(t)$s are found with fitting, using a new procedure based on the Padé approximation method. (ii) The ranks $R_{x,y}$s of the matrices $\sigma_{x,y}$s are found from the matrices $\phi_{x,y}(t_1, t_2)$s; any particular $\phi_{x,y}(t_1, t_2)$ and the corresponding $\sigma_{x,y}$ have the same rank, yet, the rank of $\phi_{x,y}(t_1, t_2)$ is much more accurately obtained from finite data. We have developed a new numerical procedure that computes the rank of the $\phi_{x,y}(t_1, t_2)$s from the data. (iii) The matrices $\sigma_{x,y}$ are estimated from the data while constructing special JTPDFs (of the sum of following JTs and of the sum of the square root of following JTs) with a new numerical procedure. (iv) The last step utilizes a variant of Eq. (3) when constructed from the RDF.

Using the toolbox, an RDF is constructed from the data fairly accurately, and importantly, much more accurately than other mechanisms. Once the RDF is constructed, we can express this model with a set of KSs. The set usually contains the most possible KSs that are associated with the constructed RDF. Choosing from the set a particular KS requires additional information from other sources (additional experiments, etc.).

2.2. *The noise in the data*

Can we filter the noise in a way that the obtained data is accurate to an extent that enables accurate construction of a mechanism? This depends on the noise-level and on the filter. Weak filters also fail when the noise level is solvable. A fair statement is that the problem of dealing with noise is an unsolved issue in the context of data from individual molecules. In too many projects, "too" simple filters as well as filters that rely on binning are used. But these collapse much prior to the filter that is presented here. We present in the next part a filter that solves a two-state photon trajectory. The filter is applicable on any discrete data with simple extensions. Yet, we also present a general method that we can use on other data types. Next, we present all the main issues with noise in two-state data and other data-types.

2.2.1. *The type of the external noise*

The type of the external noise depends on the measurement's type. This information is utilized in the analysis of the noisy data. Examples include Poissonian noise and Gaussian noise. In particular, the time interval between successive photons is monitored in measurements that collect photons. A simple model for generating a photon two-state trajectory is shown in Fig. 2(a). The *on–off* Markovian KS has transition rates λ_{on} and λ_{off}, connecting, respectively, the *on* substate with the *off* substate and the *off* substate with the *on* substate. Once the process is in the *on* (*off*) state it emits photons with a rate, γ_{on} (γ_{off}) (in fact, noise photons are recorded also when the process is in the *on* state: this can only slightly increase the effective γ_{on}).

Two-state trajectories can have a Gaussian noise. This is observed in, e.g. ion channel recordings. We generate such data while first still generating a *clean* two-state trajectory, $u(t)$, yet here a (zero-averaged with width σ_z) Gaussian noise $z(t)$ is added every dt: the equation for the signal $w(t)$ reads: $w(t) = u(t) + z(t)$.

2.2.2. *Strong noise?*

Clearly, correctly cleaning the noise depends on how strong the noise is. Indeed, the experimentalists' interest is designing clean measurements. Yet, in the analysis, we

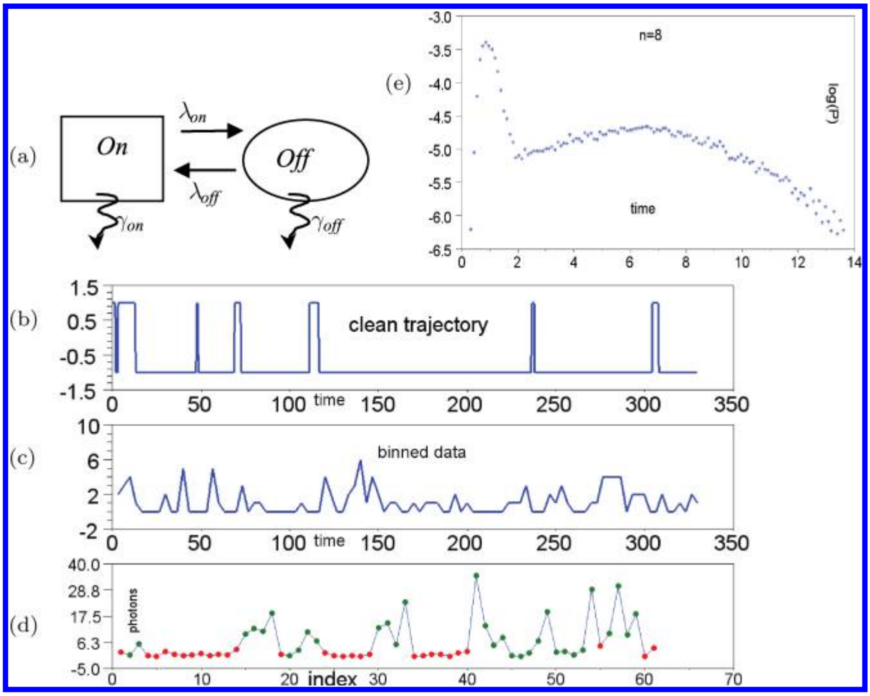


Fig. 2. A KS with a mechanism of photon emission (a), the clean *on–off* trajectory (b), the binned data (c), the photon trajectory (d), and a log-linear plot of the steady state histogram of photons (e). In (a), a curly arrow indicates on the emission of photons and a full arrow is a transition rate. The rate values are: $\lambda_{on} = 1/10$ and $\lambda_{off} = 1/99$, $\gamma_{on} = 1$ and $\gamma_{off} = 1/10$ (scaled units). The binned trajectory in (c) is very difficult to work with. The photons in (d) are plotted in the order detected, where *on* (*off*) photons are in red (green). The filter is applied on this trajectory. The histogram in (e) is the photon PDF in order $n = 8$. Its two parts are clear.

must have a way in solving any value of the ratio signal/noise, and this includes a way to identify "too" high noise levels.

2.2.3. *Internal noise: The issue of time resolution, detection efficiency, etc.*

The noise can also originate from low time-resolution of the experimental devices compared with the measured process. Say that the fastest duration of *on-off* transitions τ is smaller than the time for detecting the required amount of photons. This means that fast events can be missed. A missed fast *on* event can also originate when the number of photons that are emitted in τ is small and is the same as the number of noise photons in an *off* event. A missed *off* event occurs when the number of noise photons emitted in a fast *off* duration τ is unusually large (saying that there are just relatively fast *off* photons in the specific *off* duration) and thus bridges two *on* events. Small detection efficiency can result in similar problems.

The filter presented next can tackle such cases.

2.2.4. *Correlations in the noise*

The noise may have internal correlations, and also correlations with the state of the process. When analyzing the process, we must use the fact that the noise is correlated, that is, for discriminating signal from noise, we must include the existence of correlations in the noise. Not doing this leads to erroneous results.

The filter presented next tackles such cases.

2.2.5. *Using correlations in the clean data in solving noise*

When there are correlations among durations in the clean data, these must be taken into account when cleaning noisy data. Again, one must use this information. Otherwise, the results could be misleading.

The filter presented next tackles such cases.

2.2.6. *Other issues*

Any issue that interferes with identifying accurately the state of the process from the observable is noise. For example, entities diffusing in and out the laser spot, fluctuating coefficients (the detection efficiency) etc. We must treat all these issues when filtering the noise.

2.3. ***Filtering the noise***

In Secs. 2.1 and 2.2, we defined the problem: for solving the data in the right way while utilizing RDFs, we need solving the noise in the data. Our mathematical approach presented in Sec. 1.4 enables us in writing the likelihood function,

$$l(\boldsymbol{\Theta}) = \sum_{m,n} \log\left(P_{\text{observable}}(m, \boldsymbol{\Theta}_d)\phi_{\text{clean data}}(n, \boldsymbol{\Theta}_c)\right) + \textit{compensation terms} \qquad (3^*)$$

The symbol Sigma, $\Sigma_{m,n}$, includes all observable values and all values of the derived clean data. The first step in the algorithm is identifying the type of the observable in $P_{\text{observable}}(i, \boldsymbol{\Theta}_d)$. Yet, Eq. (3*) determines the best solution (the best identification of the observable), while maximizing $l(\boldsymbol{\Theta})$, with a fixed $\boldsymbol{\Theta}$. The set $\boldsymbol{\Theta}$ includes the coefficients representing (for example) the photons, $\boldsymbol{\Theta}_d$, and those coefficients representing the derived clean data, $\boldsymbol{\Theta}_c$. We find these coefficients from the data, namely, we write these with statistics from the data. $\phi_{\text{clean data}}(j, \boldsymbol{\Theta}_c)$ is first approximated with statistical functions. Only in an advanced step in the filtering, $\phi_{\text{clean data}}(j, \boldsymbol{\Theta}_c)$ represents the model. This model is an RDF. Therefore, in this scheme, we iterate among the identification of the photons and the identification of the best mechanism that can generate the clean data. A similar iteration scheme was suggested in Ref. 66, yet here the algorithm forming the filter having many innovative treatments, yet also the specific likelihood function, see Eq. (5).

Likelihood functions frequently appear with *compensation terms* because there are biases in the likelihood function. Yet, the *compensation terms* in Eq. (3*) are system-dependent and will determine the strength of the filter. In addition, we can perform several adjustments also on the first term in Eq. (3^*). In what follows, we show this while solving pretty complicated versions of noisy two-state photon data with the general concepts presented in Secs. 2.1. and 2.2. resulting in Eq. (3^*).

2.3.1. *Known statistical methods for noise-filtration*

Clearly, many authors have dealt with solving the actual data in relevant measurements.[46–54,51–63] General methods include: the maximum likelihood technique, a maximum entropy approach, and a Bayesian information approach. Quite a few authors have dealt with part of the specific problems that were defined in Sec. 2.2.[48–54] For example, correcting for missed fast events using a modified likelihood function was suggested in Ref. 54 when tackling ion channel data. Such methods may also be applied here, adjusted, when building the RDF from the data.

Yet, let me emphasize that techniques for solving noise in two-state data are partial: simple thresholds and smoothening are usually applied on the binned data. Correlations in the noise, or among the noise and the state of the mechanism are not taken in account in not so few cases. In addition, the methods do not use correlations among following durations in the clean data when solving the noise. In many cases, the methods assume that (due to various treatments) the noise is averaged out and does not influence the analysis. Clearly, all these assumptions are problematic and may lead to misleading conclusions. There are cases where filters that rely on binning fail yet our filter presented here removes the noise (we show such an example in the next part and in the Appendix).

Tools for solving noisy m-state trajectories, FRET trajectories, and continuous trajectories are sparser.

2.3.2. *Cleaning noisy photon two-state data*

We present here and in the next part our recently developed filters and new variants that solve noisy two-state photon trajectories where we apply Eq. (3^*) on various cases: in fact, we solve several dozen variants.

The description of the process. We say that the measured entity stays at its position while it is measured (solving diffusion is a simple extension). The detection efficiency (and quantum yield, etc.) is high. Still, the data is rather short and contains just 2700 *on–off* cycles.

We start with the KS 2(a). We set $\lambda_{\text{on}} = 1/10$, $\gamma_{\text{on}} = 1$, $\lambda_{\text{off}} = 1/99$, $\gamma_{\text{off}} = 1/10$ (all units are scaled), and generate clean data and photon data [see Figs. 2(b)–2(d)]. This design enables having 10 photons in any state, in a typical event, and in addition *on* photons are ten times faster, so in most cases there is a clear distinction among states. Nevertheless, in Fig. 2(d) we show a simple design, where there are *on* events that are fast with only one photon (photon 19 and photon 55 in 2D) and when such an event happens, it looks like one long *off* event, rather than: *off* event, short *on* event, *off* event. Our filters designed here can solve such cases. We also emphasize that we apply the filter on many rate values finding the interval where results can be accurately seen. This in fact solves the issue of signal to noise ratio, since now we can compute the rate values that generate a filterable trajectory: once we solve a particular data, consistency tests should approve that indeed we have a filterable trajectory, e.g.: in the case where KS 2(a) generated the data, filterable data is seen when *on* photon rate is at least 3 times faster than the *off* photon rate, and on average we should detect at least 5 photons in each event.

Comment: when the filter is designed inappropriately, it may fail. Not applying all steps in this filter (in the way presented next) may result in rates that are ten times different than the real rates. This filter in the stable form can tackle issues that were not solved in the past. Authors frequently assume that the noise is averaged out, or small, or not important and ignored the noise, or calculate correlations function from the binned data averaging out most of the new information that is contained in data from individual molecules, and other questionable techniques that lead to misleading results and partial results. We also show that this filter works when binning fails.

The basic filter: *the design.* The basic filter is built in the following way[b]:

(i) Create the PDF of the photons.[c] Identify the two parts (*on* and *off*). We must see these parts in this PDF in order, having the possibility of filtering the data. The intersection among the parts is a threshold, $TrSld_1$.

(ii) Compute directly from the data $1/\gamma_{on}$ and $1/\gamma_{off}$ using $TrSld_1$. Here, $t_x = 1/\gamma_x$, where t_x is the average of the x (= *on*, *off*) photons. Since there are many *off* photons left to $TrSld_1$, we use a special correction formula:

$$\tilde{t}_{off} = (1 - e^{-\frac{TrSld_1}{t_{off}}})(-TrSld_1 e^{-\frac{TrSld_1}{t_{off}}} + t_{off}(1 - e^{-\frac{TrSld_1}{t_{off}}})) + t_{off} e^{-\frac{TrSld_1}{t_{off}}}.$$

$\tilde{t}_{off}$ is the updated value and $\tilde{t}_{off} = \frac{1}{\gamma_{off}}$. (In what follows we write t_{off} also when calling the updated quantity.)

(iii) Compute a likelihood threshold using the derived rates,

$$Tr_L = 1/(1/t_{on} - 1/t_{off}) * \log(t_{off}/t_{on}).$$

This is a second threshold, Tr_L. The likelihood threshold is obtained when we set, likelihood *on* photon equal to likelihood *off* photon.

[b] The codes are presented at `http://www.flomenbom.net/codes_project12.html`.

[c] We apply the filter to the raw data. The raw data are the relative photon arrival times (to the detector). We call these relative arrival times: photons. Here "relative arrival times" is the time among two following photons' recordings.

(iv) Compute a threshold for a slow *on* photon, $TrOn_3$,

$$TrOn_3 = t_{on} * \log(N_p/10),$$

where N_p is the total number of photons. $TrOn_3$ is slowest possible *on* photon in the data in a statistical sense.

(v) Create trajectories with averaging of order n, where $n = 2, \ldots, 27$. We average any photon with n following photons e.g.: $t_i(n) = \frac{1}{n}(t_i + t_{i+1} + \cdots + t_{i+n-1})$ is photon i in trajectory of order n, where here t_i is the ith recorded photon.

(vi) Construct the photon PDF for each trajectory of order n [see Fig. 2(e)]. Compute the three thresholds in each n order trajectory: $TrSld_1$ is still the intersection of the two parts of the photon PDF. Tr_L is computed from the *on* part of the photon PDF: Tr_L is the time that maximizes the *on* part plus one standard deviation. $TrOn_3$ is computed from the *off* part of the photon PDF. It is the time that maximizes the *off* part averaged with the intersection of the parts [See Fig. 2(e)].

(vii) The algorithm determines each photon's fate in each trajectory of order n. It uses the thresholds. In a simple version, the *on* condition checks following *on* photons: at least one of the three following photons is smaller than Tr_L and the real photon is smaller than $TrOn_3$. *On* photons fulfill this condition. *Off* photons fulfill the condition: at least one of two following photons is larger than Tr_L or the photon checked is larger than $TrSld_1$.

(viii) There is a correction part in the algorithm compensating of the fact that the edges photons (when changing states) are smoother relative to the raw trajectory, particularly in higher order trajectories.

(ix) The likelihood function determines the best result. Its form determines its strength. We have derived the best form here: compute the likelihood of the photons in each trajectory, $l_{\text{photons}}(n)$,

$$l_{\text{photon}}(n) = \sum_i \log(\gamma_{on} e^{-t_{on,i}(n)\gamma_{on}}) + \sum_i \log(\gamma_{\mathit{off}} e^{-t_{\mathit{off},i}(n)\gamma_{\mathit{off}}}).$$

Here, $t_{on,i}(n)$ is *on* photon number i identified in trajectory of order n, and similarly in the *off* photons with $t_{\mathit{off},i}(n)$. The exponential function is extendable: the photon emission mechanism, $\psi_x(t)$, can result in, e.g. a multi-exponential function, yet, $\psi_x(t)$ can include other effects. We normalize $l_{\text{photon}}(n)$: $l_{\text{photon}}(n) \to l_{\text{photon}}(n)/\max_n |l_{\text{photon}}(n)|$. We then compute the likelihood of the derived clean data (this is the part of the *on and off* durations), $l_{on\ \mathit{off}}(n)$,

$$l_{on\ \mathit{off}}(n) = \sum_i \log(\lambda_{on} e^{-u_{on,i}(n)\lambda_{on}}) + \sum_i \log(\lambda_{\mathit{off}} e^{-u_{\mathit{off},i}(n)\lambda_{\mathit{off}}}).$$

Here, $u_{on,i}(n)$ is *on* duration number i identified in the trajectory with averaging of order n, and similar in the *off* durations. Again, the exponential function representing the clean data durations, $\phi_x(t)$, is extendable, and can account for any mechanism that generates these times. We normalize $l_{on\ \mathit{off}}(n)$: $l_{on\ \mathit{off}}(n) \to l_{on\ \mathit{off}}(n)/\max_n |l_{on\ \mathit{off}}(n)|$. The total likelihood is a simple combination of the two parts.

Yet, we also check about correlations in the data and when such are seen, another term, $l_c(n)/\max_n |l_c(n)|$, is included in the total likelihood function (see specification in the next part). Thus, the likelihood function in this filter follows:

$$l(\mathbf{\Theta}, n) = \tilde{l}_{ph.}(\mathbf{\Theta}_{ph.}(n), n) + \tilde{l}_{on\ \mathit{off}}(\mathbf{\Theta}_{on\ \mathit{off}}(n), n) + \tilde{l}_c(\mathbf{\Theta}_c, n), \tag{5}$$

where the symbol twiddle represents the normalization specified above, and we explicitly write here the n-dependence of the likelihood function where n represents the averaging degree.

The basic and general treatments in this filter include: (i) developing the various thresholds and (ii) the averaging of order n, (iii) developing the specific conditions in the algorithm, (iv) creating the special likelihood function from Eq. (3*). We emphasize that we can use in the filter in steps (vi)–(ix) in the algorithms other conditions and other likelihood functions. Therefore, this filter can have many additional forms. In fact, we present in what follows several additional likelihood functions (step (ix) in the filter) and several additional set of conditions (step (vii) in the filter), and study the advantages of each of these in cleaning the noise.

The basic filter: *results.* Applying the filter, we see encouraging results: we are able to extract λ_{on} and λ_{off} in various cases (see Table 1). For example, in the case specified above, the best result finds λ_{on} within 8%, λ_{off} within 7%, where *on* photons are identified correctly 86% of the time and *off* photons 94% of the time. We also find the region where the data is "too" noisy. In order to have accurate results, we must have an average of at least five *on* photons in an *on* event and γ_{on} is five times faster than λ_{off} and γ_{off}. Otherwise, filtering will fail. In Table 1, we also talk about results from three likelihood functions, showing that here Eq. (5) solves best the noise. We also apply a partial filter on several cases showing that we must use the entire filtering. In a "partial" filter, we use a simple threshold-filter with the values, $\mathit{Tr}_L(n)$ and $\mathit{TrShld}(n)$, instead of the conditioning parts (vi)–(viii) in the algorithm. In both the cases, the condition checked either the i raw photon or the i averaged photon (e.g. the condition is: $\text{photon}(n, i) \geq \mathit{Tr}_L(n) | \text{photon}(1, i) \geq \mathit{Tr}_L(1)$, when checking an *off* photon with a photon-likelihood threshold). The results show just 30% accuracy in both rates in both the cases.

The filter presented here solves cases where binning fails. We show this explicitly in the Appendix.

Results from an advanced filter. We also apply an advanced filter on the data where the conditions in part (vii) in the algorithm are changed. Here, the *on* photon-conditions are about three photons. Either photon i is faster than TrSld_1, or photon i is faster than TrOn_3 with one of the next two averaged photons is faster than

Table 1. The results when filtering the data generated from the simple mechanism in Fig 2(a): λ_{on} and λ_{off} in terms of the real values, and the percentage of correct determination of the type of the photon. We checked the results from three likelihood functions. The reported results are from likelihood function 3 (unless otherwise is indicated). Likelihood 1: the normalization of the function is performed with the number of cycles. The results are like in case 3, since in this example the main contribution in the likelihood function is from $l_{photon}(n)$. Likelihood 2: every term in the likelihood function is normalized with the number of specific events in that term. Here, in most cases the likelihood in an increasing function of n resulting in relatively large rates. There are cases where we see a maximal value at an intermediate n resulting from the patterns of the detection probability: the *off* detection probability is usually a decreasing function of n, where the *on* detection probability is an increasing function of n. Yet, there are cases where this likelihood function is better than other functions, and this is when the correlation signal is small. Likelihood 3: normalizing every term with the largest value in all n, see Eq. (5). The results are reported in the table: the best result is starred. In the text we talk about comparing results from different filters.

Case	Likelihood 3
(1) $\lambda_{on} = 1/10$, $\gamma_{on} = 1$, $\lambda_{off} = 1/99$, $\gamma_{off} = 1/10$	$p(\lambda_{on}) = 108\%$, $p(\lambda_{off}) = 93\%$, 86% *on*, 94% *off*
Advanced filter on case 1	$p(\lambda_{on}) = 98\%$, $p(\lambda_{off}) = 88\%$, 85% *on*, 94% *off*
*Zooming filter on case 1	$p(\lambda_{on}) = 106\%$, $p(\lambda_{off}) = 113.9$, 93% *on*, 92% *off*
(2) $\lambda_{on} = 1/10$, $\gamma_{on} = 1$, $\lambda_{off} = 1/33$, $\gamma_{off} = 1/10$	$p(\lambda_{on}) = 139\%$, $p(\lambda_{off}) = 193\%$, 91% *on*, 92% *off*
*Advanced filter on case 2	$p(\lambda_{on}) = 123\%$, $p(\lambda_{off}) = 103\%$, 86% *on*, 91% *off*
Zooming filter on case 2	$p(\lambda_{on}) = 121\%$, $p(\lambda_{off}) = 145\%$, 94% *on*, 84% *off*
(3) $\lambda_{on} = 1/10$, $\gamma_{on} = 1$, $\lambda_{off} = 1/5$, $\gamma_{off} = 1/10$	$p(\lambda_{on}) = 416\%$, $p(\lambda_{off}) = 163\%$, 96% *on*, 74% *off*
*Advanced filter on case 3	$p(\lambda_{on}) = 198\%$, $p(\lambda_{off}) = 123\%$, 89% *on*, 72% *off*
Zooming filter on case 3	$p(\lambda_{on}) = 169\%$, $p(\lambda_{off}) = 440\%$, 67% *on*, 94% *off*
(4) $\lambda_{on} = 1/49$, $\gamma_{on} = 1$, $\lambda_{off} = 1/49$, $\gamma_{off} = 1/10$	$p(\lambda_{on}) = 113\%$, $p(\lambda_{off}) = 103$, 98% *on*, 93% *off*
Advanced filter on case 4	$p(\lambda_{on}) = 87\%$, $p(\lambda_{off}) = 116\%$, 96% *on*, 98% *off*
*Zooming filter on case 4	$p(\lambda_{on}) = 90\%$, $p(\lambda_{off}) = 105\%$, 98% *on*, 92% *off*
*(5) $\lambda_{on} = 1/49$, $\gamma_{on} = 1$, $\lambda_{off} = 1/49$, $\gamma_{off} = 1/3.77$	$p(\lambda_{on}) = 103\%$, $p(\lambda_{off}) = 63$, 86% *on*, 92% *off*
Advanced filter on case 5	$p(\lambda_{on}) = 113\%$, $p(\lambda_{off}) = 42\%$, 84% *on*, 98% *off*
Zooming filter on case 5	$p(\lambda_{on}) = 55\%$, $p(\lambda_{off}) = 44\%$, 55% *on*, 92% *off*
*(6) $\lambda_{on} = 1/10$, $\gamma_{on} = 1$, $\lambda_{off} = 1/60$, $\gamma_{off} = 1/10$	$p(\lambda_{on}) = 126\%$, $p(\lambda_{off}) = 103$, 97% *on*, 99% *off*
Advanced filter on case 6	$p(\lambda_{on}) = 157\%$, $p(\lambda_{off}) = 94$, 92% *on*, 98% *off*
Zooming filter on case 6	$p(\lambda_{on}) = 113.8\%$, $p(\lambda_{off}) = 123\%$, 94% *on*, 89% *off*
*(7) $\lambda_{on} = 1/6.1$, $\gamma_{on} = 1$, $\lambda_{off} = 1/60$, $\gamma_{off} = 1/10$	$p(\lambda_{on}) = 134\%$, $p(\lambda_{off}) = 106\%$, 84% *on*, 95% *off*
Advanced filter on case 7	$p(\lambda_{on}) = 164\%$, $p(\lambda_{off}) = 119\%$, 94% *off*, 78% *on*
Zooming filter on case 7	$p(\lambda_{on}) = 121\%$, $p(\lambda_{off}) = 139\%$, 92% *on*, 87% *off*
(8) $\lambda_{on} = 1/2.77$, $\gamma_{on} = 1$, $\lambda_{off} = 1/60$, $\gamma_{off} = 1/10$	$p(\lambda_{on}) = 207\%$, $p(\lambda_{off}) = 123\%$, 75% *on*, 94% *off*
Advanced filter on case 8	$p(\lambda_{on}) = 233\%$, $p(\lambda_{off}) = 179\%$, 72% *on*, 89% *off*
*Zooming filter on case 8	$p(\lambda_{on}) = 161\%$, $p(\lambda_{off}) = 187\%$, 84% *on*, 85% *off*

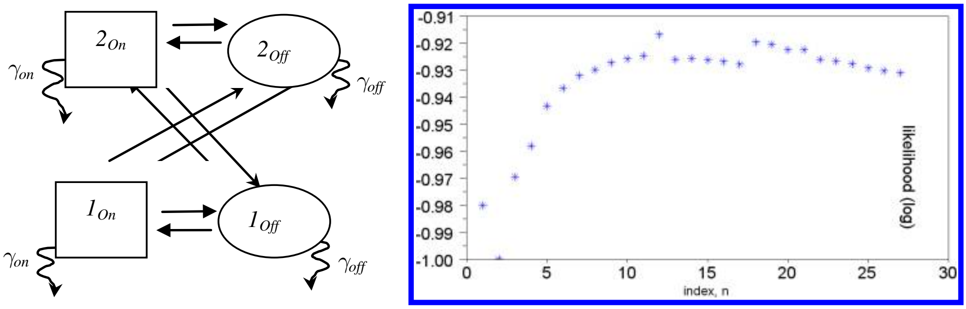
$Tr_L(n)$ and the other photon in the couple is faster than $TrSld_1$. The *off* photon-condition is about five photons, either: (i) the i photon is larger than $TrSld_1$, (ii) one in three averaged photons $[t_i(n), t_{i+1}(n), t_{i+2}(n)]$ are larger than $TrSld_1(n)$, (iii) photon $t_{i+3}(n)$ is larger than $TrSld_1(n)$ and the average of the previous three photons is larger than $Tr_L(n)$ and (iv) photon $t_{i+4}(n)$ is larger than $TrSld_1(n)$ and the average of the previous four photons is larger than $Tr_L(n)$.

Table 1 also reports on the results with the advanced filter. In various cases, the results are better with the advanced filter than the simple filter. This is valid in cases where the number of *off* photons detected is relative small, yet the difference among the average durations of *on* and *off* photons is a constant and large: γ_{on}/γ_{off} is large, and, $\gamma_{on}/\lambda_{off} =>$ decreases from a large number toward one. When many photons are detected, yet the difference among the average *on* and *off* photon times is relatively small (γ_x/λ_x is large, and, $\gamma_{on}/\gamma_{off} =>$ decreases toward one), the simpler algorithm is better.

The logic in the basis of the filters. We expect seeing many short photons in the *on* state and many slow, long, photons in the *off* state. In principle, a simple threshold can discriminate *on* and *off* photons in a deterministic photon emission case. Yet, the emission of the photons is also a stochastic process. Namely, in the *on* event we may see several photons that are slow and in an *off* event we may see several fast photons. The second case will occur often when the *off* photon emission PDF is exponential, where the first case is relatively rare. In particular, the probability that an *on* photon is larger than any cut-off c is given by: $p_{c,on} = \int_c^\infty \psi_{on}(t)dt$. Therefore, in the case where, $\psi_{on}(t) = \gamma_{on}e^{-\gamma_{on}t}$, with $\gamma_{on} = 1$ and $c = 3.33$, we have, $p_{c,on} = e^{-\gamma_{on}c} \approx 0.033$. Therefore, in an *on* state, two consecutive slow photons signaling a jump from the *on* state with a probability of 99.9%. In this way, we can choose the conditions in the algorithm. What about the *off* state? $p_{c,off} = \int_0^c \psi_{off}(t)dt$ is the probability that an *off* photon is faster than c. In the cases where $\psi_{off}(t) = \gamma_{off}e^{-\gamma_{off}t}$, with $\gamma_{off} = 1/10$ and $c = 3.33$, we have, $p_{c,off} = 1 - e^{-\gamma_{off}c} \approx 1/3$. Thus, even three fast consecutive *off* photons can appear in the trajectory with a relatively large probability. Just the sixth fast consecutive photon may signal on a jump from the *off* state. Again, we see that the conditions in the algorithm are adjustable depending on $p_{c,off}$. When we study the following fast photons until we see a slow one (in an *off* state), we check the average of these fast photons. The average should exceed (at least) the likelihood threshold: the average of m *off* photons is $1/\gamma_{off}$ with a width of $1/(\gamma_{off}\sqrt{m})$. This average minus the width is much larger than the likelihood cut-off also when m is 5 (and even smaller) in the case specified above.

2.3.3. *Cleaning noisy two-state photon data with correlations in the cleaned data*

The data is generated from a KS with four substates: two *on* substates and two *off* states, see Fig. 3(a). This generates a trajectory with correlations among durations



(a) (b)

Fig. 3. The data was generated from the KS (a) with the following rates (scaled units) ($\lambda_{x,21}$ connecting substates 1 in x and 2): $\lambda_{\mathit{off},21} = 0.85/99$, $\lambda_{\mathit{off},11} = 0.15/99$, $\lambda_{\mathit{off},12} = 0.85/499$, $\lambda_{\mathit{off},22} = 0.15/499$, and, $\lambda_{on,21} = 0.85/9$, $\lambda_{on,11} = 0.15/9$, $\lambda_{on,12} = 0.85/99$, $\lambda_{on,22} = 0.15/99$. Once the process is in any of the *off* substates, it emits photons with a rate, $\gamma_{\mathit{off}} = 1/10$, where when staying in the *on* substates, photons are emitted with a rate, $\gamma_{on} = 1$. The likelihood function is presented in (b), on a log linear scale. Here, the maximal likelihood value is when $n = 12$.

in the clean data. The filter presented in the previous part is applied also here. The only modification includes another term, $l_c(n)$,

$$l_c(n) = \sum_{x,y,i} \log(N_{x,y} \exp\{-(\sqrt{u_{x,i}(n)u_{y,i'}(n)} - \langle\sqrt{u_x u_y}\rangle)^2/2\tilde{\sigma}^2_{x,y}\}. \qquad (6)$$

Here, $x, y = on, \mathit{off}$, and i is cycle i ($i' = i + 1$, where when $x = on$ and $y = \mathit{off}$, $i' = i$), $N_{x,y}$ is a normalization constant, and, $\tilde{\sigma}^2_{x,y} = \langle u_x u_y\rangle$. $l_c(n)$ is included in the likelihood function, Eq. (5). This can account for correlations in the data. In the likelihood function, we include a normalized term: $l_c(n)/\max_n |l_c(n)|$. This term is included when the correlation coefficient $C_{x,y}$ shows correlations in the clean data that are larger than 10%. The correlation coefficient follows:

$$C_{x,y} = \left|\frac{\langle u_x u_y\rangle}{\langle u_x\rangle\langle u_y\rangle} - 1\right|.$$

Here, $x, y = on, \mathit{off}$. The results are encouraging in the various cases checked (results for about eight cases are presented in Table 2). Only when the correlation term is included [in a normalized, unbiased, way: see Eq. (5)], we see accurate results. The likelihood function is presented in Fig. 3(b), representing case 1 in Table 2. At the solution that maximizes the likelihood function, we find: (i) 99% *on* photon identification and 86% *off* photons, (ii) the average duration in the *on* state is found within 24%, and 45% in the *off* state and (iii) correlations coefficients in durations (derived from the cleaned data) are found within 67%. Positive correlation condition is seen. These results are from the simple filter, where better ones are derived from advanced filter presented next, see Table 2.

In Table 2, we present various additional results about this process. One important conclusion is that when the correlation signal is small (that is, smaller than 0.1, and there are cases here where this is also the mathematical value), a different

Table 2. Results about filtering the trajectories from Kinetic Scheme 3A. In all cases: $\gamma_{on} = 1$, $\gamma_{\mathit{off}} = 1/10$ (scaled units). In the first 4 cases, the jumping probabilities from substates of the same states are 15%, where in cases 5 until 8 these probabilities equal 33%. e.g.: in cases 1 through 4, the probability of jumping from substate 1_{on} and reaching substate 1_{off} is 85%, where the jumping times are exponentially distributed with a rate $\lambda_{on,1}$. Likelihoods are similar with those presented in the previous table. Here, $p(t_{on})$ is the computed average *on* durations in terms of the mathematical value. The term *time*: $\mathit{time} = [p(t_{on}), p(t_{\mathit{off}})]$. The term *Id.* follows: $\mathit{Id.} = [\%\mathit{on}, \%\mathit{off}]$. The term Pc containing the averages $\langle t_{on} t_{\mathit{off}} \rangle$ and $\langle t_{\mathit{off}} t_{on} \rangle$ in terms of the mathematical values. SC are the correlation conditions in all combinations of x, $y = \mathit{on}, \mathit{off}$ (*on on*, *on off*, *off on*, *off off*). We see that when the correlation signal is low, likelihood 2 is better in various cases. The best result in each case is starred.

Case	Likelihood 3
(1) $\lambda_{on,1} = 1/9$, $\lambda_{on,2} = 1/99$, $\lambda_{\mathit{off},1} = 1/99$, $\lambda_{\mathit{off},2} = 1/499$	$\mathit{time} = [124, 145]\%$, $\mathit{Id.} = [99, 86]\%$, $Pc = [69, 71]\%$, $SC = [9, 9, 6, 1]\%$
Advanced filter on case 1	$\mathit{time} = [135, 134]\%$, $\mathit{Id.} = [96, 98]\%$, $Pc = [64, 78]\%$, $SC = [4, 25, 3, 1]\%$
*Zooming filter on case 1	$\mathit{time} = [96, 113]\%$, $\mathit{Id.} = [99, 88]\%$, $Pc = [104, 106]\%$, $SC = [3, 16, 13, 13]\%$
(2) $\lambda_{on,1} = 1/19$, $\lambda_{on,2} = 1/81$, $\lambda_{\mathit{off},1} = 1/49$, $\lambda_{\mathit{off},2} = 1/199$	$\mathit{time} = [108, 132]\%$, $\mathit{Id.} = [99, 74]\%$, $Pc = [78, 78]\%$, $SC = [8, 9, 8, 4]\%$
*Advanced filter on case 2	$\mathit{time} = [121, 117]\%$, $\mathit{Id.} = [98, 93]\%$, $Pc = [80, 98]\%$, $SC = [10, 14, 12, 5]\%$
Zooming filter on case 2	$\mathit{time} = [119, 140]\%$, $\mathit{Id.} = [99, 82]\%$, $Pc = [62, 64]\%$, $SC = [9, 16, 12, 8]\%$
(3) $\lambda_{on,1} = 1/5$, $\lambda_{on,2} = 1/19$, $\lambda_{\mathit{off},1} = 1/49$, $\lambda_{\mathit{off},2} = 1/99$	$\mathit{time} = [147, 249]\%$, $\mathit{Id.} = [99, 72]\%$, $Pc = [33, 33]\%$, $SC = [1, 0, 1.9, 2.3]\%$
Advanced filter on case 3	$\mathit{time} = [183, 203]\%$, $\mathit{Id.} = [93, 83]\%$, $Pc = [29, 30]\%$, $SC = [2, 3, 0, 2.7]\%$
*Zooming filter on case 3	$\mathit{time} = [99, 104]\%$, $\mathit{Id.} = [94, 95]\%$, $Pc = [111, 111]\%$, $SC = [5, 5, 5, 2]\%$
(4) $\lambda_{on,1} = 1/5$, $\lambda_{on,2} = 1/5$, $\lambda_{\mathit{off},1} = 1/49$, $\lambda_{\mathit{off},2} = 1/199$	$\mathit{time} = [167, 450]\%$, $\mathit{Id.} = [97, 72]\%$, $Pc = [13, 14]\%$, $SC = [0, 0.6, 1, 2]\%$
Advanced filter on case 4 (likelihood 2)	$\mathit{time} = [184, 119]\%$, $\mathit{Id.} = [75, 95]\%$, $Pc = [45, 45]\%$, $SC = [1, 1, 2, 5]\%$
*Zooming filter on case 4	$\mathit{time} = [119, 126]\%$, $\mathit{Id.} = [84, 94]\%$, $Pc = [63, 64]\%$, $SC = [1, 0.7, 3, 7]\%$
(5) $\lambda_{on,1} = 1/9$, $\lambda_{on,2} = 1/81$, $\lambda_{\mathit{off},1} = 1/49$, $\lambda_{\mathit{off},2} = 1/199$	$\mathit{time} = [99, 116]\%$, $\mathit{Id.} = [69, 99]\%$, $Pc = [97, 98]\%$, $SC = [3, 3, 1, 0]\%$
Advanced filter on case 5	$\mathit{time} = [125, 127]\%$, $\mathit{Id.} = [98, 98]$, $Pc = [69, 68]\%$, $SC = [1, 6, 5, 0]\%$
*Zooming filter on case 5	$\mathit{time} = [98, 116]\%$, $\mathit{Id.} = [99, 88]$, $Pc = [106, 109]\%$, $SC = [5, 6, 2, 1]\%$
(6) $\lambda_{on,1} = 1/19$, $\lambda_{on,2} = 1/81$, $\lambda_{\mathit{off},1} = 1/49$, $\lambda_{\mathit{off},2} = 1/199$	$\mathit{time} = [111, 133]\%$, $\mathit{Id.} = [69, 99]\%$, $Pc = [69, 69]\%$, $SC = [3, 2, 1, 2]\%$
Advanced filter on case 6	$\mathit{time} = [119, 116]\%$, $\mathit{Id.} = [96, 98]\%$, $Pc = [77, 76]\%$, $SC = [3, 3, 4, 0.67]\%$
*Zooming filter on case 6	$\mathit{time} = [101, 116]\%$, $\mathit{Id.} = [99, 92]\%$, $Pc = [95, 93]\%$, $SC = [8, 10, 8, 4]\%$
(7) $\lambda_{on,1} = 1/5$, $\lambda_{on,2} = 1/19$, $\lambda_{\mathit{off},1} = 1/49$, $\lambda_{\mathit{off},2} = 1/99$	$\mathit{time} = [184, 135]\%$, $\mathit{Id.} = [99, 97]\%$, $Pc = [42, 42]\%$, $SC = [0, 0, 0, 1.5]\%$
Advanced filter on case 7	$\mathit{time} = [167, 174]\%$, $\mathit{Id.} = [94, 87]\%$, $Pc = [36, 35]\%$, $SC = [0, 3.8, 1, 2.61]\%$
*Zooming filter on case 7	$\mathit{time} = [106, 109]\%$, $\mathit{Id.} = [94, 93]\%$, $Pc = [93, 92]\%$, $SC = [6, 3, 2.4, 2]\%$
(8) $\lambda_{on,1} = 1/5$, $\lambda_{on,2} = 1/5$, $\lambda_{\mathit{off},1} = 1/49$, $\lambda_{\mathit{off},2} = 1/199$	$\mathit{time} = [600, 136]\%$, $\mathit{Id.} = [99, 67]\%$, $Pc = [13, 12]\%$, $SC = [1.4, 3, 4.9, 5]\%$
Advanced filter on case 8 (likelihood 2)	$\mathit{time} = [144, 123]\%$, $\mathit{Id.} = [81, 96]\%$, $Pc = [58, 57]\%$, $SC = [1.4, 1.8, 1, 1.3]\%$
*Zooming filter on case 8	$\mathit{time} = [116, 121]\%$, $\mathit{Id.} = [87, 94]\%$, $Pc = [74, 75]\%$, $SC = [0.3, 3, 5, 8]\%$

normalization than that in Eq. (5) is found to be better (that is, we find better results with another normalization). This operation normalizing each term with the number of events (for example, the *on* photon-likelihood is normalized with the number of *on* photon detected, etc.) rather than with the largest absolute value (depending on n) of the term.

2.3.4. *A three scale filter*

We study here a new variant of the filter. We develop a new set of conditions in step (vii) in the algorithm. We suggest a general new concept. Scan the photon trajectory with three scales: the local scale, the intermediate one, and the possibly large scale. These are length scales in the photon index. This filter should increase the accuracy in determining a jump among states in a relatively small additional time investment (about a 33% increase in a very fast basic algorithm that solves in several minutes 2700 cycle data (with a 10 photon average in each event)).

In the three-scale filter, the local scale is similar with the scale appearing in the conditions suggested in the previous filters (these check about five photons in determining the type of the specific photon). The intermediate scale is used in ambiguous cases where the local scale does not result in a clear answer about "jumping" or "staying". There, we check a bunch of eight following photons after the photon that may imply a jump: we compute statistics that should confirm the jump, e.g.: we compute the average and the variance of the photons in the group and detect patterns that may show a jump: we check whether the number of following photons faster than a threshold, what is the slowest photon in the group, and additional patterns. In cases where the eight photon analysis does not supply a clear answer about a jump, we check a larger group. We will scan the trajectory until a clear pattern signifying a jump is seen (that is, instead of working with eight photons we may work with 25 photons or 33 photons and even more).

Specific three-scale filter. Here, we specify a three-scale filter.

In the *on* state: the basic condition is like in the advanced filter. When a jump is implied, the zooming part is considered with four and five photons. A jump is implied when any of the following conditions is fulfilled: (i) the average of either groups is larger than Tr_L. (ii) The maximal photon value is larger than $TrOn_3$. (iii) The standard deviation is larger than $TrSld_1$. (iv) In addition, we also check eight photons, when three from eight are larger than $TrSld_1$: this implies a jump.

In the *off* state: the basic condition is like in the advanced filter. When a jump is implied, the zooming part is considered with groups with four and five and six and seven and eight photons: (i) in case where the average is larger than Tr_L and the next photon is larger than the $TrSld_1$: this too is the *off* state continuation. (ii) When the maximal photon is larger than $TrOn_3$: this is the *off* state continuation. (iii) When the standard deviation is larger than $TrSld_1$: this is the *off* state continuation.

(iv) In addition, we also check eight photons: when three from eight are larger than $TrSld_1$: this is the *off* state continuation.

Results from the three scale filter: *the simple system.* Here, we talk about the results from the specific three-scale filter when applied on the simple KS in Fig. 2(a). Results appear in Table 1. We see that in all cases where the advanced filter having better results than the simple filter, the filter with the zooming in part can also improve the results.

Results from the three-scale filter: *the correlated data.* Here, we talk about the results from the three-scale filter when applied on the KS in Fig. 3(a). Results appear in Table 2. The zooming in filter is almost always much better than the other filters.

3. Concluding Comments

This article presented treatments required when solving single molecules.

3.1. *Solving single molecule trajectories*

The filters explaining: we must invest in smart filters. Noise in the data has not been treated in the most accurate way in many projects. This may result in misleading conclusions. Solving the noise in relevant data will help significantly many researchers. (i) Here, we talked about our recently developed likelihood function, Eq. (3*), and its specific forms in discrete data. We showed here that in two-state noisy photon trajectories, this form is a combination of three terms: the photon data, the clean *on–off* durations, and the correlations. Each term is normalized in a special way: in most cases, the normalization follows: $l_*(n)/\max_n |l_*(n)|$ where * representing the various term types. There are cases where we normalize with the specific number of events per term. We see in preliminary studies of a different filter that the likelihood function should also contain specially designed compensating terms in order having even more accurate results. (ii) The iterative scheme that we talked about in the area of Eq. (3*) was suggested in Ref. 66. Yet, the uniqueness here is in the statistical and technical fronts and the algorithms' forms that is based on many new treatments presented here and in Ref. 64. (iii) The noise filters combined with the toolbox RDF that solves clean discrete data will form the most accurate way that can solve single molecules.

3.2. *Further research*

The filter reviewed here cleans the noise. Once we have the clean data, we can use our toolbox to find the model from the clean data[39] (see also Sec. 2.1). Else, we can apply another filter's type that uses the results from the clean data of the first filter. We can apply this filter in an iterative way (the input coefficients are updated after the filter is applied, and the filter is applied again until convergence is seen). Our focus is in finalizing such a filter. We emphasize that tackling other trajectory-types will demand further development: Extensions will treat continuous trajectories, FRET data, quantum dot data, and other data variants that extend the discrete data type. Further related filters will appear in future publications.

An important part in this project continuation will develop software that will enable experimentalists solving the trajectories easily without bother about all the math. We will collaborate with Fred Sachs about combining and improving the toolbox RDF and the software QUB.[30,31,66] We plan also collaborating with experimentalists when developing stable software.

3.3. *Further concluding comments about this review and the Special Issue*

The field of biophysics is an established one. We usually utilize dynamics and thermodynamics in explaining biological processes in this field. In this review, we talked about solving data from the most advanced experiments in biophysics: experiments measuring individual biological entities. We must apply smart statistical methods and smart numerical algorithms when creating an accurate model from the data. The model is a KS: a KS is valuable in explaining the dynamical process and is valuable in expressing the energy surface of the process. The KS can account for reaction pathways and structural dynamics, simultaneously. A complex KS can have time dependent rates representing changing environments. Here, any KS also represents the way the observable is generated. Solving the data requires smart filtering methods for solving the noise and ways about constructing the mechanism from the clean data. All these are talked about in this paper. Related reviews that appear in this Special Issue about measuring and solving single molecules include: the review from Fred Sachs about the software QUB.[6] The review from Hatzakis and collaborators about measuring individual enzymes.[67] The review from Orte and collaborators about measuring bioprocesses with FRET.[2]

Appendix A. Comparisons with Binning and Thresholding

In the Appendix, we show that the filter presented in the main text solves cases where binning the data does not work and the obtained binned trajectory is "too" noisy and all the methods that are used on the binned data will not help (the methods in Ref. 50 do not filter the noise: these calculate correlation functions from the raw data (and from slightly smoothed data) but this does not clean the data and the extracted information is rather poor in content).

In Figs. A.1–A.3, we simply show that when the noise increases there are cases where binning results in trajectories where an *off* event might contain a peak that is identical with *on* events and that there are *on* events that contain "too" few photons, identical with an *off* event. Thus, any method that is applied on the binned data will not filter such cases. In such cases we must apply the information from various bins and also global information. Yet, rather than using the binned values we should employ the raw data, e.g.: the photons. Just an algorithm that can check every photon when utilizing the local information of consecutive photons and the global information (thresholds extracted from the entire data) can help. The algorithms of the main text therefore constitute the best way that can enable solving the data photon after photon.

We highlight the point that since the noise appears also in the *off* state, when photons are recorded with an exponential rate, many will have fast values. Applying the Fourier-transform on the binned data will not help much here, in particular, in the cases specified in the next figures. (The Fourier transform method will help when each point in the trajectory is a combination of real signal and a Gaussian noise.)

A.1. *A simple case*

In Fig. A.1 in the Appendix, we present data that is generated from the mechanism in Fig. 2(a) in the main text. The rate values are: $\lambda_{on} = 1/10$, $\gamma_{on} = 1$, $\lambda_{off} = 1/99$, $\gamma_{off} = 1/10$ (all units are scaled). Figure A.1 showing various panels: the upper panel shows the clean *on-off* data and the binned data, with the number of photons in a bin of size 1.99. The other (lower) panel is the photon durations in the order recorded. We also plot the real identification. The larger value there is the value of *about* $1/\gamma_{off}$ and the smaller value is the value of about $1/\gamma_{on}$.

In this setup, we can generate very clear two-state data. We see very clearly two-state trajectory also in the binned data (middle of the figure, blue), and the binned data and the clean data (red curve) coincide. Thus, in this example, cleaning

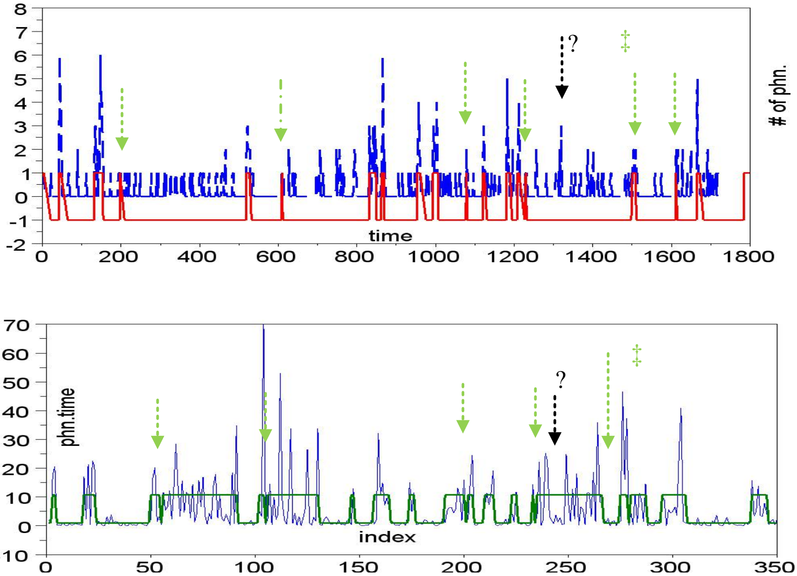


Fig. A.1. (Color online) The binned data (upper panel), and the raw data (lower panel): the photon durations. In both panels, we show also the clean data. We show also misinterpretations with dashed arrows (black: *off*). Further description is presented in the text.

the noise also from the binned data is simple. We show with arrows the events that are missed (green: missed *on* event, black: missed *off* event) when thresholding the binned data, with: *threshold*= 3. In the lower panel we show the raw data: the photon durations in the order recorded (blue) with the real identification (in green). Clearly, this trajectory is smoother than the binned data. Nevertheless, in this case also the binned data is not "too" noisy. We show in the plots cases where filtering the raw data is simple but the binned data is "too" noisy. These are labeled with, "?" and "‡" in both panels.

A.2. *A more complicated case*

In this case the data was also generated from KS 2A, yet here the rate values are (scaled units): $\lambda_{on} = 1/10$, $\gamma_{on} = 1$, $\lambda_{off} = 1/49$, $\gamma_{off} = 1/10$. Here, the rate is changed for controlling the *off* durations. In this case, the rate value is just 1/49 where in the previous example $\lambda_{off} = 1/99$. Since in this example λ_{off} is faster, there are many *off* events with few photons. Many fast *off* events increases the noise: there are not so few *off* events that do not have the required amount of photons needed in order that in that event we will detect a slow photon. We see in this example that the binned data is rather noisy in comparison with the raw data representation of photon durations in the order recorded. We solved such data with the methods of the main text (see Table 1). In Fig. A.2, we show with arrows

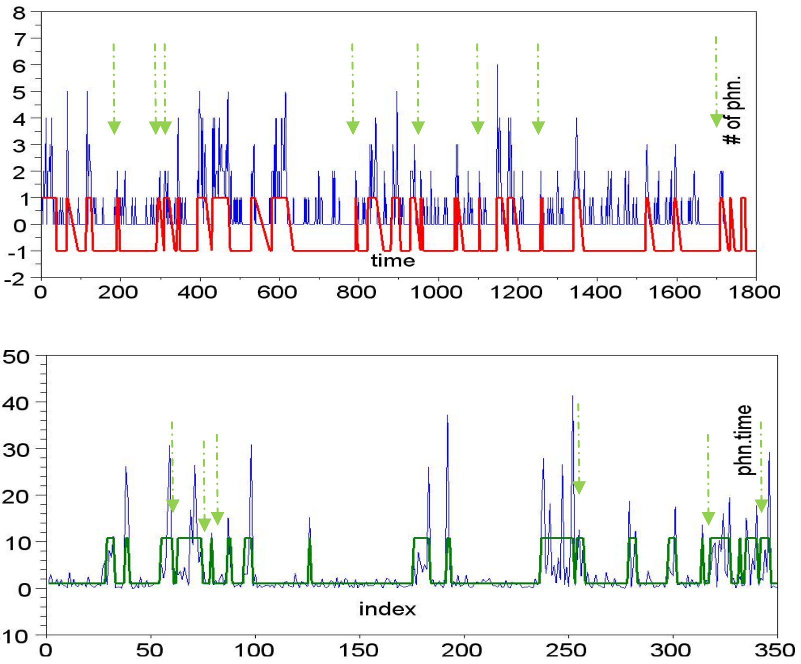
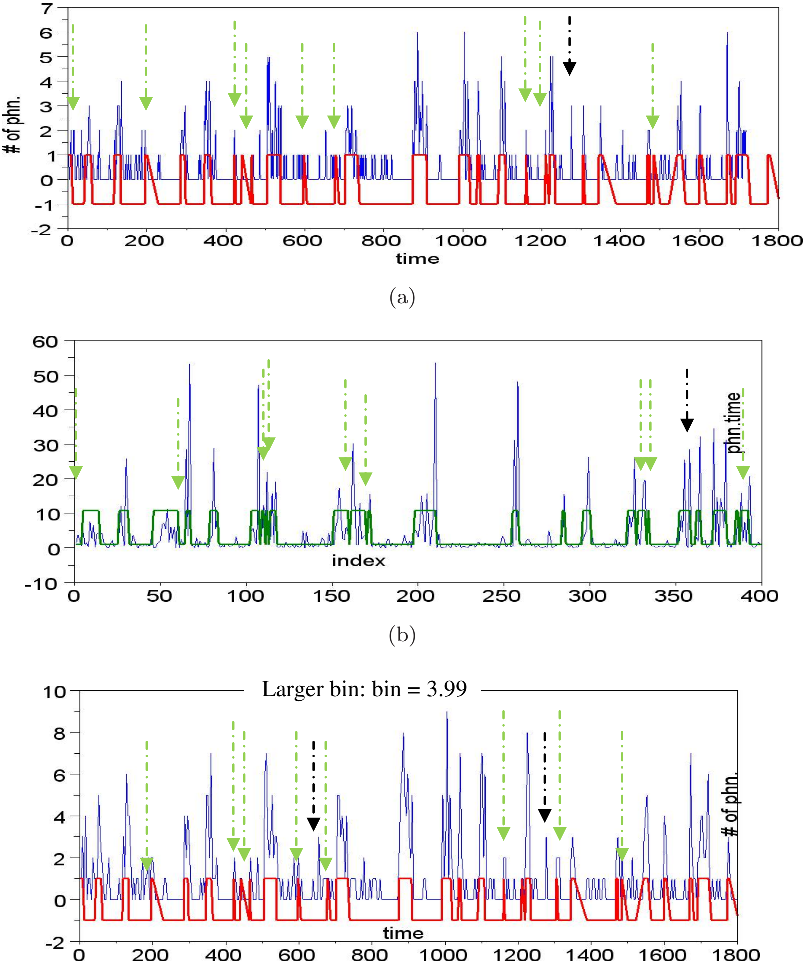


Fig. A.2. Trajectories from a more complicated case. Panels are similar with those in the previous figure. The description is presented in the text.

all the cases where the real binned data is *on* but the recorded binned data is smaller than the threshold (*threshold* = 3). We show these cases also in the raw data representation. At least cases 2 and 3 (in chronological order) are solved with the filter presented in the main text but are missed in the binned data with any filtering technique. The new filter is therefore at least 25% better.

A.3. *The complicated case*

In this case, the data was also generated from KS 2A in the main text, yet the rate values are (scaled units): $\lambda_{on} = 1/10$, $\gamma_{on} = 1$, $\lambda_{off} = 1/33$, $\gamma_{off} = 1/10$. We have



(a)

(b)

(c)

Fig. A.3. Trajectories from KS 2A in the main text with rate values resulting in a rather noisy case. Panels are similar with those in the previous figure. Yet here we have three panels (a), (b), (c): the third panel is a trajectory with a larger bin size than in Fig. A.3(a). The description is presented in the text.

changed the rate controlling the *off* durations: here λ_{off} is just 1/33 where in the previous examples $\lambda_{off} = 1/49$ and $\lambda_{off} = 1/99$.

In this setup there are many *off* events with few photons: there are short events shorter than in the previous example. This increases the noise: there are many *off* events that do not have the required amount of photons needed in order seeing a very slow one. Again, we see here that the binned data is rather noisy, clearly, relative with the raw data representation of photon durations in the order recorded. We solved also such data with the methods of the main text (see Table 1). In Fig. A.3, we show with arrows all the cases where the binned data is *on* but smaller than the threshold; here, *threshold* = 3. These are shown in green. We show with a black arrow when an *off* event having many photons in a bin. Here, this is a bin larger than 3. We show all these cases also in the raw data representation. At least cases 1, 2 and 9 (in a chronological order) are solved with the filter presented in the main text yet are missed in the binned data with any filtering technique. Therefore here the new filer is at least 33% better than binning. We show in panel 3 that a larger bin size does not improve the situation: one additional *on* event is identified but one *off* event has many photons in a bin and is not detected.

References

1. W. E. Moerner and M. Orrit, *Science* **283**, 1670 (1999).
2. M. J. Ruedas-Rama, J. M. Alvarez-Pez and A. Orte, *Biophys. Rev. Lett.* **8**, 161–190 (2013).
3. E. Neher and B. Sakmanm, *Nature* **260**, 799 (1976).
4. M. Fahrner *et al.*, *Immunol. Rev.* **231**, 99 (2009).
5. J. J. Kasianowicz, E. Brandin, D. Branton and D. W. Deamer, *Proc. Natl. Acad. Sci. USA* **93**, 13770 (1996).
6. C. Nicolai and F. Sachs, *Biophys. Rev. Lett.* **8**, 191–211 (2013).
7. B. Schuler, E. A. Lipman and W. A. Eaton, *Nature* **419**, 743 (2002).
8. P. Narayan *et al.*, *Nat. Struct. Mol. Biol.* **19**, 79 (2012).
9. H. Yang, G. Luo, P. Karnchanaphanurach, T. Louie, I. Rech, S. Cova, L. Xun and X. S. Xie, *Science* **302**, 262 (2003).
10. E. Rhoades, E. Gussakovsky and G. Haran, *Proc. Natl. Acad. Sci. USA* **100**, 3197 (2003).
11. W. Min, G. Lou, B. J. Cherayil, S. C. Kou and X. S. Xie, *Phys. Rev. Lett.* **94**, 198302 (2005).
12. X. Zhuang, H. Kim, M. J. B. Pereira, H. P. Babcock, N. G. Walter and S. Chu, *Science* **296**, 1473 (2002).
13. H. Lu, L. Xun and X. S. Xie, *Science* **282**, 1877 (1998).
14. L. Edman, Z. Földes-Papp, S. Wennmalm and R. Rigler, *Chem. Phys.* **247**, 11 (1999).
15. K. Velonia *et al.*, *Angew. Chem. Int. Edit.* **44**, 560 (2005).
16. O. Flomenbom *et al.*, *Proc. Natl. Acad. Sci. USA* **102**, 2368 (2005); O. Flomenbom *et al.*, *Chem. Phys. Lett.* **432**, 371 (2006).
17. B. P. English *et al.*, *Nat. Chem. Biol.* **2**, 87 (2006).
18. R. J. Davenport, G. J. L. Wuite, R. Landick and C. Bustamante, *Science* **287**, 2497 (2000).
19. F. Rico, A. Oshima, P. Hinterdorfer, Y. Fujiyoshi and S. Scheuring, *J. Mol. Biol.* **412**, 72 (2011).

20. M. Ribezzi, M. Wagner and F. Ritort, *J. Non. Math. Phys.* **18**, 397 (2011).
21. C. Roduit *et al.*, *Nat. Methods* **9**, 774 (2012).
22. S. Nie, D. T. Chiu and R. N. Zare, *Science* **266**, 1018 (1994).
23. R. Shusterman, S. Alon, T. Gavrinyov and O. Krichevsky, *Phys. Rev. Lett.* **92**, 048303-1 (2004).
24. H. De Keersmaecker, S. Rocha, E. Fron *et al.*, *Biophys. Rev. Lett.* **8**, 229–242 (2013).
25. A. E. Cohen and W. E. Moerner, *Proc. Natl. Acad. Sci. USA* **103**, 4362 (2006).
26. R. M. Dickson, A. B. Cubitt, R. Y. Tsien and W. E. Moerner, *Nature* **388**, 355 (1997).
27. I. Chung and M. G. Bawendi, *Phys. Rev. B.* **70**, 165304-1 (2004).
28. R. Rossetti, S. Nakahara and L. E. Brus, *J. Chem. Phys.* **79**, 1086 (1983).
29. J. Tang and R. A. Marcus, *J. Chem. Phys.* **123**, 204511 (2005).
30. F. Qin. A. Auerbach and F. Sachs, *Biophys. J.* **79**, 1928 (2000).
31. F. Qin, A. Auerbach and F. Sachs, *Biophys. J.* **79**, 1915 (2000).
32. F. G. Ball and M. S. P. Sansom, *Proc. R. Soc. Lond. B* **236**, 385 (1989).
33. O. Flomenbom, J. Klafter and A. Szabo, *Biophys. J.* **88**, 3780 (2005).
34. J. Cao, *Chem. Phys. Lett.* **327**, 38 (2000).
35. W. J. Bruno, J. Yang and J. Pearson, *Proc. Natl. Acad. Sci. USA* **102**, 6326 (2005).
36. O. Flomenbom and J. Klafter, *Acta Phys. Pol. B* **36**, 1527 (2005); *J. Chem. Phys.* **123**, 064903 (2005).
37. J. B. Witkoskie and J. Cao, *J. Chem. Phys.* **121**, 6361 (2004).
38. O. Flomenbom and R. J. Silbey, *Proc. Natl. Acad. Sci. USA* **103**, 10907 (2006); *J. Chem. Phys.* 128, 114902 (2008).
39. O. Flomenbom and R. J. Silbey, *Phys. Rev. E* **78**, 066105 (2008).
40. R. J. Bauer, B. F. Bowman and J. L. Kenyon, *Biophys. J.* **52**, 961 (1987).
41. P. Kienker, *Proc. R. Soc. London B.* **236**, 269 (1989).
42. D. R. Fredkin and J. A. Rice, *J. Appl. Prob.* **23**, 208 (1986).
43. D. Colquhoun and A. G. Hawkes, *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **300**, 1 (1982).
44. L. Song and K. L. Magleby, *Biophys. J.* **67**, 91 (1994).
45. O. Flomenbom, *Adv. Chem. Phys.* **146**, 367 (2011).
46. B. Hille, *Ion Channels of Excitable Membranes* (USA: Sinauer Associates Inc, 2001).
47. D. R. Cox and D. V. Hinkley, *Theoretical Statistics* (USA: Chapman & Hall/CRC, 1974).
48. L. E. Baum, T. Petrie, G. Soules and N. Weiss, *Ann. Math. Stat.* **41**, 164–171 (1970).
49. R. Horn and K. Lange, *Biophys. J.* **43**, 207–223 (1983).
50. H. Yang and X. S. Xie, *J. Chem Phys.* **117**, 10965–10979 (2002).
51. D. Colquhoun, A. G. Hawkes and K. Srodzinski, *Philos. Trans. R. Soc. London, Ser. A* **354**, 2555–2590 (1996).
52. F. Qin, A. Auerbach and F. Sachs, *Biophys. J.* **70**, 264–280 (1996).
53. B. Roux and R. Sauve, *Biophys. J.* **48**, 149–158 (1985).
54. A. G. Hawkes, A. Jalali and D. Colquhoun, *Philos. Trans. R. Soc. London, Ser. A* **332**, 511 (1990).
55. C. B. Li, H. Yang and T. Komatsuzaki, *Proc. Nat. Acad. Sci. USA* **105**, 536–541 (2008).
56. L. P. Watkins and H. Yang, *Biophys. J.* **86**, 4015–4029 (2004).
57. J. Cao and R. J. Silbey, *J. Phys. Chem. B* **112**, 12876 (2008).
58. I. V. Gopich and A. Szabo, *J. Phys. Chem. B* **114**, 15221-6 (2010); *PNAS USA* **109**, 7747 (2012).
59. H. S. Chung, I. V. Gopich, K. McHale, T. Cellmer, J. M. Louis and W. A. Eaton, *J. Phys. Chem. A* **115**(16), 3642–3656 (2011).

60. R. Hurtado Guerrero *et al.*, *J. Biol. Chem.* **284**, 8461-9 (2009).
61. J. N. Taylor, D. E. Makarov and C. F. Landes, *Biophys. J.* **98**, 164–173 (2010).
62. M. A. Little and N. S. Jones, *Proc. R. Soc. A* **467**, 3088–3114 (2011); *Proc. R. Soc. A* **467**, 3115–3140 (2011); *Phil. Trans. R. Soc. A* **371**, 20110546 (2013).
63. G. Tatyana *et al.*, *ACS Nano* **6**(1), 346–354 (2012).
64. O. Flomenbom, submitted (2013). See http://arxiv.org/abs/1306.2435.
65. A. G. Hawkes, A. Jalali and D. Colquhoun, *Phil. Trans. R. Soc. Lond. B* **337**, 383–404 (1992).
66. F. Qin, *Biophys. J.* **86**(3), 1488–1501 (2004).
67. S. K. Jørgensen and N. S. Hatzakis, *Biophys. Rev. Lett.* **8**, 137–160 (2013).