JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

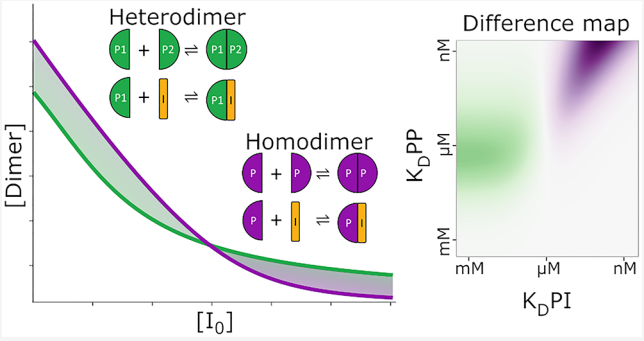
PyBindingCurve, Simulation, and Curve Fitting to Complex Binding Systems at Equilibrium

Steven Shave,* Yan-Kai Chen, Nhan T. Pham, and Manfred Auer*

Cite This: *J. Chem. Inf. Model.* 2021, 61, 2911−2915 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Understanding multicomponent binding interactions in protein−ligand, protein−protein, and competition systems is essential for fundamental biology and drug discovery. Hand-deriving equations quickly become unfeasible when the number of components is increased, and direct analytical solutions only exist to a certain complexity. To address this problem and allow easy access to simulation, plotting, and parameter fitting to complex systems at equilibrium, we present the Python package PyBindingCurve. We apply this software to explore homodimer and heterodimer formations culminating in the discovery that under certain conditions, homodimers are easier to break with an inhibitor than heterodimers and may also be more readily depleted. This is a potentially valuable and overlooked phenomenon of great importance to drug discovery. PyBindingCurve may be expanded to operate on any equilibrium binding system and allows definition of custom systems using a simple syntax. PyBindingCurve is available under the MIT license at https://github.com/stevenshave/pybindingcurve as the Python source code accompanied by examples and as an easily installable package within the Python Package Index.

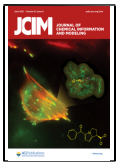


INTRODUCTION

Gaining insight into molecular interaction systems at equilibrium by simulation and by fitting of experimental data is of fundamental importance for basic biology and drug discovery. The advantages provided in experimental planning alone justify its importance, allowing expectations of signal strength and species abundance to guide experimental, assay, and instrument setup. In this manuscript, we document the creation of PyBindingCurve and apply it to the most useful protein−ligand systems in biology and drug discovery, deriving equations for the resultant population abundances at equilibrium. Beyond the simplest systems, manual derivation of direct algebraic solutions becomes difficult and we therefore turn to symbolic manipulation in software to derive solutions. These systems are described by polynomial equations with multiple solutions, presenting the problem of not only choosing the correct solution but also choosing the correct solution throughout an experiment where the physically relevant solution may change throughout a titration. In addition, there are no general direct solutions to polynomials of order greater than 4,[1,2] limiting the scope for deriving direct analytical solutions to highly complex systems. We therefore turn to minimization and root-finding techniques, transforming the problem into one of constrained optimizations. We compiled these solutions with methods to simulate titrations, plot results, and fit parameters for experimental value determination into a Python package named PyBindingCurve, allowing simple simulation, plotting, and fitting of systems at equilibrium. Multiple freely available and commercial software packages exist for plotting and simulating systems;[3−5] however, most are closed-source, hindering development and integration into existing workflows or require real solutions to be derived or transferred from the literature,[6,7] a difficult process, especially with the added problem of selecting the correct polynomial root representing physically relevant solutions. Having an open-source framework to simulate, fit, and interrogate these systems brings with it great advantages allowing automation and integration into existing software pipelines and analysis methods.

All binding events may be described over time by an on-rate (with units: $M^{-1}\cdot s^{-1}$), describing the rate of association, and an off-rate describing the complex falling apart into its constituent species (units: s^{-1}). The interplay between complex association and dissociation in a closed system creates a dynamic equilibrium containing a steady state of species abundances. We may combine these rate constants describing the

Received: February 24, 2021
Published: May 18, 2021



population at equilibrium into the dissociation constant, K_D (units: M), defined simply as the off-rate divided by the on-rate. Conveniently, when one binding site is present, such as in 1:1 binding, this value denotes the concentration at which 50% of a species will be in complex with its binding partner. Lower values denote higher affinity interactions and therefore tighter binding. Often, we may derive direct analytical solutions to the concentration of the complex formed. A full derivation of 1:1 binding from mass balances is available as Supporting eq 1 and can be found in the literature.[8−11] Simulation of binding curves with this equation requires no special treatment, with only one polynomial root being physically relevant across all possible experimental parameter values. This provides a direct method for calculating complex concentrations over a range of system parameters.

In addition to 1:1 binding, a common system is 1:1:1 competition, commonly used in drug discovery efforts to detect new chemical entities displacing a known binder. As an example, a fluorescently labeled ligand in complex with a target protein may have its fluorescence anisotropy measured.[12,13] Displacement of the labeled ligand by another inhibitor competing for the same binding site will remove the labeled ligand from the complex resulting in a change of anisotropy as a function of complex concentration. The 1:1:1 competition binding equation may be solved in the same manner as demonstrated in the Supporting Information for 1:1 binding with changes made to the mass balances and results in a third-order or cubic equation requiring significantly more manipulation but remaining feasible by hand. This results in three possible polynomial roots as solutions, one of which may be entirely excluded as never physically relevant, while the choice between the other two is dependent on the ligand and inhibitor K_Ds relative to each other. This solution is also readily available in the literature.[14] Additionally, the breaking of heterodimers with an inhibitor can be represented by this 1:1:1 competition system, where protein monomers can be thought of as a protein and ligand, which upon binding become a heterodimer while the inhibitor competes for a binding site on one of the protein monomers.

Studies in biology often involve oligomers, or repeating units, the conceptually simplest of which is a homodimer: two identical monomer units binding to each other. Homodimers have been characterized as having distinct properties setting them apart from heterodimers, which comprise two chemically different proteins. In general, the binding interfaces of homodimers are larger with more interacting residues, specifically enriched with hydrophobic amino acids.[15] Mathematically understanding this complex and its behavior is critically important for fundamental biology. It is therefore surprising that we were able to find only one instance in the literature of derivation of a direct, analytical solution to homodimer formation at equilibrium.[16] Upon first considering homodimer formation, it appears a simpler case than the 1:1 protein−ligand binding as only one starting species is involved, two units of which transition to become a single dimer upon complexation. An important consideration exists when a dimer undergoes dissociation and two monomers are produced, increasing the concentration of free monomer at double the dissociation rate. The same is true in reverse for complexation with two monomers consumed for the creation of one dimer. The derivation of a direct analytical solution to dimer formation can be found in Supporting eq 2. Like the solution of the quadratic equation in 1:1 binding, one polynomial root is always physically correct.

We were unable to find the literature providing direct analytical solutions for homodimer breaking with an inhibitor. This is surprising as fundamental biology and drug discovery efforts often seek to break apart homodimers with small molecules or peptides mimicking interaction surfaces. At this level of complexity, solutions can be found using symbolic manipulation with programs such as Wolfram Mathematica.[17] See the Supporting Information codes 1−4 for Wolfram Mathematica code solving the 1:1, 1:1:1 competition, dimer formation, and dimer breaking systems, respectively. In highly complex systems, such as homodimer breaking, we may observe the physically correct solution "switching" from one polynomial root to another as titrations progress. To avoid this problem, we use minimization and root-finding in an approach similar to that taken by Royer[18] to solve the systems expressed as constrained optimization problems (See the Supporting Information Code Listings 5−8). PyBindingCurve allows custom systems to be defined and solved using minimization-based techniques; simply specifying "P+P<−>PP" defines homodimer formation, while "P+L<−>PL, P+I<−>PI" defines a 1:1:1 competition. More complex systems such as ternary complex formation are also easily specified:[19] "P+C<−>PC, P+U<−>PU, PC+U<−>PCU, PU+C<−>PCU" (see the Simulation of Custom Binding Systems section in the accompanying Supporting Information). Additionally, systems may be solved kinetically as a system of ordinary differential equations (ODEs, see the Supporting Information Code Listings 9−12). PyBindingCurve automatically utilizes the fast, direct analytical solutions to systems where possible, otherwise minimization, and root-finding techniques for systems expressed as constrained optimization problems are used. Kinetic solvers are not used by default but can be specified. Below, we document the result of using PyBindingCurve to explore the striking differences between homo- and heterodimer breaking and its implications. The PyBindingCurve software package available at https://github.com/stevenshave/pybindingcurve allows simulation, fitting, and derivation of system parameters for a range of predefined and custom-definable systems.

RESULTS AND DISCUSSION

Using the PyBindingCurve package, we investigated the large and surprising differences between homo- and heterodimer formations. Understanding of these differences has the potential to impact fundamental biology, with better use of tool compounds and an improved system biology-based understanding of biological pathways. We must first consider dimer formation, best illustrated by a theoretical experiment, where an increasing concentration of monomer is titrated. While experimentally it is not easy to increase the monomer concentration, the experiment could be performed in reverse with a buffer titrated into a known starting amount of monomer and dimer concentrations monitored. A dimer half-life would need to be considered, ensuring an equilibrium is achieved before each dimer measurement is recorded. Performing the experiment for heterodimers is the simplest conceptually, whereby a known and equal starting concentration of each monomer "A" and "B" is diluted, giving a total number of particles in a constant volume of N. To directly compare homodimer complexation with this system, we must

https://doi.org/10.1021/acs.jcim.1c00216
J. Chem. Inf. Model. 2021, 61, 2911−2915

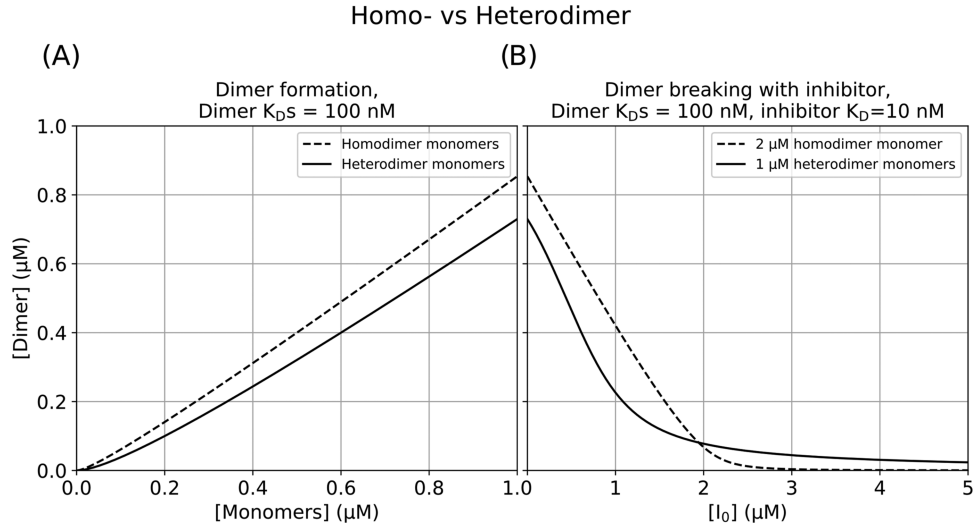


Figure 1. Homo-vs-Heterodimer making and breaking. (A) Dimer formation with a K_D of 100 nM as a function of monomer concentration. As a homodimer (broken line) contains two copies of the same monomer, the total monomer concentration is twice that shown on the *x*-axis. Heterodimer formation (solid line) contains two different monomers with the concentration for each monomer given by the *x*-axis. (B) After monomer titration to 1 μM as shown in (A), or effectively 2 μM in the homodimer case, the resultant complex has an inhibitor (I_0) titrated against it. The inhibitor has a K_D of 10 nM to one heterodimer monomer (solid line) and the same K_D to homodimer monomers.

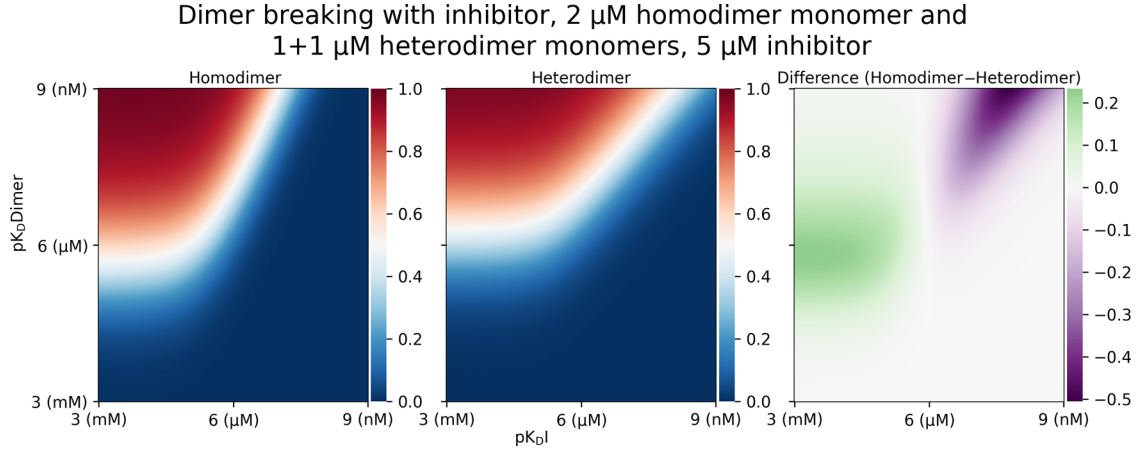


Figure 2. Homo-vs-heterodimer dissociation heatmaps showing dimer concentration between 1 μM (red) and 0 μM (blue) as a function of changing dimer and inhibitor affinity expressed as pK$_D$ between 1 nM and 1 mM. Right panel shows the difference between the homodimer and heterodimer abundance heatmaps, with magenta representing less homodimer than heterodimer at a constant 5 μM inhibitor concentration.

use twice the homodimer "H" monomer concentration to achieve the same number of particles as N.

Figure 1A shows the dimer formation as a function of monomers for dimers with a dissociation constant (K_D) of 100 nM. It is evident that with the same number of particles present, more homodimers than heterodimers are formed. This is expected as homodimer monomers can form a dimer with any other monomer. Heterodimers are only formed when two complimentary monomers come together, effectively halving the concentration of binding partners that a heterodimer monomer encounters. Figure 1B illustrates profound differences observed upon dimer breaking with an inhibitor, the understanding of which is of crucial importance for drug discovery and fundamental biology involving the application of drugs and tool compounds for dimer breaking. We can observe a system where dimer complexation starting with a total of 2 μM homodimer monomer with a dimerization K_D of 100 nM has an inhibitor titrated into it with a K_D of 10 nM to homodimer monomer. Comparing this to a similar system containing 1 μM of both components A and B of a heterodimer monomer (total 2 μM monomer concentration) with a similar dimerization inhibitor with a K_D of 10 nM to A, we observe striking differences. First, as expected from the dimer formation plot in Figure 1A, the starting dimer concentration of a homodimer is higher than that of a heterodimer. Again, this can be explained simply by the fact that a homodimer monomer can bind any other homodimer monomer, whereas, with a heterodimer monomer, a monomer must encounter a complimentary monomer for complexation to occur. As the titration continues, an interesting crossover occurs at around 2 μM inhibitor concentration with an equal amount of homo- and heterodimers present. As inhibitor concentration increases, the amount of heterodimer present is greater than that of homodimer. This effect can be explained by the increased abundance of inhibitor binding partners in the case of homodimers. As in the previous explanation, particles of homodimers are encountered at twice the rate of particles of appropriate heterodimer monomers by the inhibitor diffusing

https://doi.org/10.1021/acs.jcim.1c00216
J. Chem. Inf. Model. 2021, 61, 2911−2915

in the solution, causing more homodimer monomer−inhibitor complex than heterodimer monomer−inhibitor complex. For the same amount of inhibitors in the solution, a greater proportion complexes with monomers in the case of homodimers, removing a free monomer capable of complexing with another monomer into a dimer. This is interesting as, initially, the increased dimerization of homodimer is greater than the effect of increased homodimer monomer−inhibitor formation. There is, however, a point in inhibitor titration where this balance shifts, and the effect of the increased homodimer monomer−inhibitor formation is greater than the increased dimerization, shifting the dynamic such that homodimers may be more easily dissociated apart by inhibitors. A final observation can be made, in that the depletion of homodimers is more complete than that of heterodimers at high concentrations of inhibitors. Again, this can be explained by the abundance of inhibitor binding partners, each monomer in the case of homodimers vs one monomer in the case of heterodimers. Further exploring the switch of homo- vs heterodimer ease of dissociation, we may visualize areas of affinity space where breaking one is easier than the other. The left panel of Figure 2 shows the concentration of homodimer formed (red is 1 μM, blue is zero homodimer) with a starting concentration of 2 μM monomer and a range of inhibitor and dimerization K_Ds along the x- and y-axes, respectively, expressed as pK_Ds. The center panel of Figure 2 shows the same for heterodimer, with 1+1 μMs of the two monomers as a starting concentration and a range of inhibitor K_Ds. The right panel of Figure 2 shows the difference between homodimers and heterodimers, green indicating that the homodimer was harder to break and magenta indicating that it was easier to break.

In the development of PyBindingCurve, we iterated through many approaches to simulating protein−ligand systems as their complexity increased. Starting with hand-crafted direct analytical solutions, increased system complexity led to a computer-generated code requiring tracing approaches to choose the correct solution. As complexity increased and solutions to high-order polynomials were no longer found,[1,2] such as in 1:4 protein−ligand binding, we transitioned to first using iterative kinetic models describing systems as ODEs to minimization problems expressed as constrained system optimization problems. PyBindingCurve automatically chooses the most appropriate method to solve common binding systems. The use of constrained optimization led to methods capable of solving user-defined binding systems specified in the simple text, allowing applicability of PyBindingCurve to practically all biological systems. Having such capabilities present in an open-source package promotes the freedom to use, integrate, and improve PyBindingCurve.

The striking difference in the behavior of homodimers vs heterodimers could have a significant impact on drug discovery efforts. Simply dissemination of the knowledge that near-complete depletion of homodimers is easier than with heterodimers is valuable, before making any numerical predictions or analysis. We would like to emphasize the take-home message from our findings: if faced with a choice, drug discovery programs may wish to prioritize homodimers over heterodimers as targets as a route to more efficacious therapies. Most drugs in clinical development fail from the lack of efficacy, which is strongly connected to achieving the necessary drug concentration at the site of the target in human tissues. If a lower concentration of a drug acting on a homodimer can achieve equal or better effects, this might increase drug efficacy and open a new avenue in drug discovery. We believe a major strength of PyBindingCurve is direct programmatic access for exploration of these binding systems in Python, currently one of the most popular and fastest-growing programming languages. This helps allow insights as demonstrated in the homo- vs heterodimer formation example, which would not have been easily discovered using existing offerings of the traditional curve fitting and simulation software.

We envision the continued growth and development of PyBindingCurve. A detailed user guide with tutorials and API documentation is available in the Supporting Information accompanying this manuscript and online (https://stevenshave.github.io/pybindingcurve/).

METHODS

We implemented PyBindingCurve in Python (version 3.6.8), developing methods capable of simulating, plotting, and fitting parameters to experimental results for 1:1, 1:n (where n is 1−5), 1:1:1 competition, and homodimer and heterodimer systems. In the testing and validation process for these analytically solved complex systems with direct methods, we encountered considerable numerical instability[20] and so use arbitrary precision arithmetic to a high degree of accuracy using the mpmath (version 1.1.0) package.[21] Internally, PyBindingCurve uses a combination of direct analytical solutions for simple binding systems (computationally quick to simulate) and minimization and root-finding to solve constrained systems when dealing with more complex systems. To derive the Python code for simple systems with direct analytical solutions, we used Wolfram Mathematica to derive solutions from sets of mass balances and binding equations (see the Supporting Information Code Listings 1−4). These solutions were written out, and the program MathematicaEquationToPython was used to convert the equations to Python functions (available at https://github.com/stevenshave/MathematicaEquationToPython). Systems that proved unsolvable using symbolic manipulation in Mathematica are integrated into PyBindingCurve using constrained optimization approaches. The Supporting Information Code Listings 5−8 illustrate the Python code used to construct these constrained optimization systems and simulate 1:1, 1:1:1 competition, homodimer formation, and homodimer breaking. The code present within PyBindingCurve is also capable of parsing custom-defined systems and transforming them into constrained systems that are easily solvable. These systems are solved using the find_roots method from the mpmath package. In addition to system simulation at a single set of starting conditions, we created helper functions to enable plotting over a range of species concentrations. Parameter fitting is achieved using the LMFit package,[22] enabling the calculation of parameters such as K_D from experimental data.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c00216.

Supporting mass balances, diagrams, and Wolfram Mathematica code used to derive direct analytical solutions to binding systems; and Python code for minimizer-based and kinetic interrogation of systems, PyBindingCurve User Guide, and API reference (PDF)

https://doi.org/10.1021/acs.jcim.1c00216
J. Chem. Inf. Model. 2021, 61, 2911−2915

AUTHOR INFORMATION

Corresponding Authors

Steven Shave − *University of Edinburgh, School of Biological Sciences, University of Edinburgh, Edinburgh, Scotland EH9 3BF, United Kingdom;* orcid.org/0000-0001-6996-3663; Email: s.shave@ed.ac.uk

Manfred Auer − *University of Edinburgh, School of Biological Sciences, University of Edinburgh, Edinburgh, Scotland EH9 3BF, United Kingdom;* orcid.org/0000-0001-8920-3522; Email: manfred.auer@ed.ac.uk

Authors

Yan-Kai Chen − *University of Edinburgh, School of Biological Sciences, University of Edinburgh, Edinburgh, Scotland EH9 3BF, United Kingdom;* orcid.org/0000-0001-7161-9503

Nhan T. Pham − *University of Edinburgh, School of Biological Sciences, University of Edinburgh, Edinburgh, Scotland EH9 3BF, United Kingdom;* orcid.org/0000-0003-1620-2910

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c00216

Author Contributions

S.S. programmed PyBindingCurve with contributions from Y.-K.C. S.S. and N.T.P. contributed to fitting techniques and management of solution switching. M.A. and S.S. conceived the project. The manuscript was written through contributions of S.S., N.T.P., and M.A. All authors have given approval to the final version of the manuscript.

Notes

The authors declare no competing financial interest.

Full source code along with examples for every system is available in the public GitHub repository accessible at https://github.com/stevenshave/pybindingcurve. Additionally, the PyBindingCurve package has also been submitted to the Python Package Index (https://pypi.org/project/pybindingcurve/), which may be installed via pip using the command "pip install pybindingcurve". Online documentation along with a tutorial is also available at https://stevenshave.github.io/pybindingcurve/.

ACKNOWLEDGMENTS

This manuscript is dedicated to the memory of the late Dr. Kurt Müller, a great scientist and a warm-hearted colleague, whose investigation into the automated derivation of system equations from mass balances and binding equations made this work possible. The authors would also like to acknowledge the extensive work performed by Prof. Nicole Meisner-Kober in experimental validation of Dr. Müller's mathematical derivations. S.S., N.T.P., and M.A. acknowledge financial support from the Scottish Universities Life Sciences Alliance (SULSA, http://www.sulsa.ac.uk) and the Medical Research Council (MRC, www.mrc.ac.uk, J54359) Strategic Grant. M.A. and N.T.P. acknowledge financial support from the Wellcome Trust (Grant 201531/Z/16/Z).

REFERENCES

(1) Ayoub, R. Paolo Ruffini's contributions to the quintic. *Arch. Hist. Exact Sci.* **1980**, *23*, 253−277.

(2) Rosen, M. Niels Henrik Abel and the equation of the fifth degree. *Am. Math. Mon.* **1995**, *102*, 495−505.

(3) Royer, C. A. Improvements in the numerical analysis of thermodynamic data from biomolecular complexes. *Anal. Biochem.* **1993**, *210*, 91−97.

(4) Royer, C. A.; Smith, W. R.; Beechem, J. M. Analysis of binding in macromolecular complexes: A generalized numerical approach. *Anal. Biochem.* **1990**, *191*, 287−294.

(5) Munson, P. J. LIGAND: a computerized analysis of ligand binding data. *Methods Enzymol.* **1983**, *92*, 543−576.

(6) Bronstein, I. N.; Semendjajew, K. A.; Grosche, G.; Ziegler, V.; Ziegler, D. *Taschenbuch der Mathematik*; Deutsch, 1989.

(7) Thrall, S. H.; Reinstein, J.; Wöhrl, B. M.; Goody, R. S. Evaluation of human immunodeficiency virus type 1 reverse transcriptase primer tRNA binding by fluorescence spectroscopy: specificity and comparison to primer/template binding. *Biochemistry* **1996**, *35*, 4609−4618.

(8) Hulme, E. C.; Trevethick, M. A. Ligand binding assays at equilibrium: validation and interpretation. *Br. J. Pharmacol.* **2010**, *161*, 1219−37.

(9) Green, N. M. A Spectrophotometric Assay for Avidin and Biotin Based on Binding of Dyes by Avidin. *Biochem. J.* **1965**, *94*, 23C−24C.

(10) Inglese, J.; Blatchly, R. A.; Benkovic, S. J. A multisubstrate adduct inhibitor of a purine biosynthetic enzyme with a picomolar dissociation constant. *J. Med. Chem.* **1989**, *32*, 937−40.

(11) Wang, Z. X.; Kumar, N. R.; Srivastava, D. K. A novel spectroscopic titration method for determining the dissociation constant and stoichiometry of protein-ligand complex. *Anal. Biochem.* **1992**, *206*, 376−81.

(12) Weber, G. Rotational Brownian Motion and Polarization of the Fluorescence of Solutions. *Adv. Protein Chem.* **1953**, *8*, 415−459.

(13) Lakowicz, J. R. *Principles of Fluorescence Spectroscopy*, 3rd ed.; Springer: New York, 2006; 954 p.

(14) Teukolsky, S. A.; Flannery, B. P.; Press, W.; Vetterling, W. Numerical recipes in C. *SMR* **1992**, *693*, 59−70.

(15) Zhanhua, C.; Gan, J. G.; Lei, L.; Sakharkar, M. K.; Kangueane, P. Protein subunit interfaces: heterodimers versus homodimers. *Bioinformation* **2005**, *1*, 28−39.

(16) Benfield, C. T.; Mansur, D. S.; McCoy, L. E.; Ferguson, B. J.; Bahar, M. W.; Oldring, A. P.; Grimes, J. M.; Stuart, D. I.; Graham, S. C.; Smith, G. L. Mapping the IκB kinase β (IKKβ)-binding interface of the B14 protein, a vaccinia virus inhibitor of IKKβ-mediated activation of nuclear factor κB. *J. Biol. Chem.* **2011**, *286*, 20727−35.

(17) Wolfram Research, I.. *Mathematica Version 12*, 12.0; Champaign: Illinois, 2019.

(18) Royer, C. A.; Smith, W. R.; Beechem, J. M. Analysis of binding in macromolecular complexes: a generalized numerical approach. *Anal. Biochem.* **1990**, *191*, 287−94.

(19) Han, B. A suite of mathematical solutions to describe ternary complex formation and their application to targeted protein degradation by heterobifunctional ligands. *J. Biol. Chem.* **2020**, *295*, 15280−15291.

(20) Goldberg, D. What Every Computer Scientist Should Know About Floating-Point Arithmetic. *ACM Comput. Surv.* **1991**, *23*, 5−48.

(21) Johansson, F. *mpmath: A Python Library for Arbitrary-precision Floating-pointarithmetic (version 1.0.0), 1.1.0*, 2018.

(22) Newville, M.; Stensitzki, T.; Allen, D. B.; Rawlik, M.; Ingargiola, A.; Nelson, A. LMFIT: Non-linear least-square minimization and curve-fitting for Python.*Astrophysics Source Code Library*, 2016.